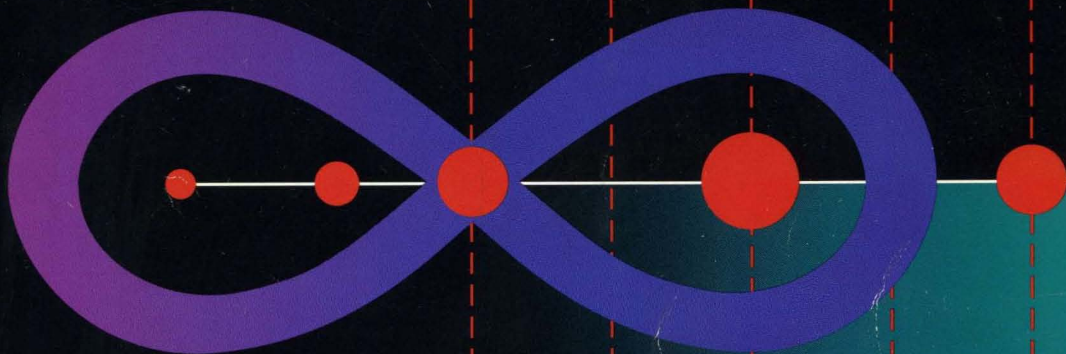
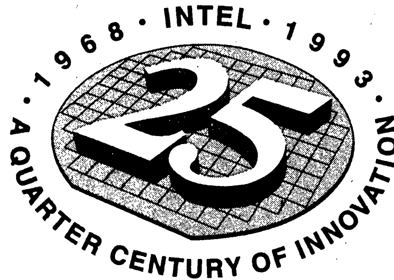
Memory Products



intel



Founded in 1968 to pursue the integration of large numbers of transistors onto tiny silicon chips, Intel's history has been marked by a remarkable number of scientific breakthroughs and innovations. In 1971, Intel introduced the 4004, the first microprocessor. Containing 2300 transistors, this first commercially-available computer on a chip is considered primitive compared with today's million-plus transistor products.

Innovations such as the microprocessor, the erasable programmable read-only memory (EPROM) and the dynamic random access memory (DRAM) revolutionized electronics by making integrated circuits the mainstay of both consumer and business computing products.

Over the last two and a half decades, Intel's business has evolved and today the company's focus is on delivering an extensive line of component, module and system-level building block products to the computer industry. The company's product line covers a broad spectrum, and includes microprocessors, flash memory, microcontrollers, a broad line of PC enhancement and local area network products, multimedia technology products, and massively parallel supercomputers. Intel's 32-bit X86 architecture, represented by the Intel386™ and Intel486™ microprocessor families, are the de facto standard of modern business computing and installed in millions of PCs worldwide.

Intel has over 25,000 employees located in offices and manufacturing facilities around the world. Today, Intel is the largest semiconductor company in the United States and the second largest in the world.

U.S. and CANADA LITERATURE ORDER FORM

NAME: ____________________

COMPANY: ____________________

ADDRESS: ____________________

CITY: ____________________ STATE: ________ ZIP: ________

COUNTRY: ____________________

PHONE NO.: (   ) ____________________

ORDER NO.	TITLE	QTY.		PRICE		TOTAL
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	

Subtotal ________

Must Add Your Local Sales Tax ________

Include postage:
Must add 15% of Subtotal to cover U.S. and Canada postage. (20% all other.)

→ Postage ________

Total ________

Pay by check, money order, or include company purchase order with this form ($200 minimum). We also accept VISA, MasterCard or American Express. Make payment to Intel Literature Sales. Allow 2-3 weeks for delivery.

☐ VISA ☐ MasterCard ☐ American Express Expiration Date ____________________

Account No. ____________________

Signature ____________________

Mail To: Intel Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

International Customers outside the U.S. and Canada should use the International order form on the next page or contact their local Sales Office or Distributor.

For phone orders in the U.S. and Canada
Call Toll Free: (800) 548-4725

Prices good until 12/31/93.
Source HB

CG/LOF1/103092

INTERNATIONAL LITERATURE ORDER FORM

NAME: ____________________

COMPANY: ____________________

ADDRESS: ____________________

CITY: ____________________ STATE: __________ ZIP: __________

COUNTRY: ____________________

PHONE NO.: ( )____________________

ORDER NO.	TITLE	QTY.		PRICE	TOTAL
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=

Subtotal __________

Must Add Your Local Sales Tax __________

Total __________

PAYMENT

Cheques should be made payable to your ***local*** Intel Sales Office (see inside back cover).

Other forms of payment may be available in your country. Please contact the Literature Coordinator at your ***local*** Intel Sales Office for details.

The completed form should be marked to the attention of the LITERATURE COORDINATOR and returned to your ***local*** Intel Sales Office.

MEMORY PRODUCTS

1993

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and may only be used to identify Intel products:

376™
Above™
ActionMedia®
BITBUS™
Code Builder™
DeskWare™
Digital Studio™
DVI®
EtherExpress™
ETOX™
ExCA™
Exchange and Go™
FaxBACK™
FlashFile™
Grand Challenge™
i®
ICE™
iLBX™
Inboard™
i287™
i386™
i387™
i486™
i487™
i750®
i860™
i960®
Intel287™
Intel386™
Intel387™
Intel486™
Intel487™
Intel®
intel inside®
Intellec®
iPSC®
iRMX®
iSBC®
iSBX™
iWARP™
LANDesk™
LANPrint®
LANProtect™
LANSelect®
LANShell®
LANSight™
LANSpace®
LANSpool®
MAPNET™
Matched™
MCS®
Media Mail™
NetPort™
NetSentry™
OpenNET™
OverDrive™
Paragon™
Pentium™
ProSolver™
RapidCAD™
READY-LAN™
Reference Point®
RMX/80™
RxServer™
SatisFAXtion®
SnapIn 386™
Storage Broker™
SugarCube™
The Computer Inside™
TokenExpress™
Visual Edge™
WYPIWYF®

MDS is an ordering code only and is not used as a product name or trademark. MDS is a registered trademark of Mohawk Data Sciences Corporation.

CHMOS and HMOS are patented processes of Intel Corp.

Intel Corporation and Intel's FASTPATH are not affiliated with Kinetics, a division of Excelan, Inc. or its FASTPATH trademark or products.

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

©INTEL CORPORATION 1993

CG-110392

DATA SHEET DESIGNATIONS

Intel uses various data sheet markings to designate each phase of the document as it relates to the product. The marking appears in the upper, right-hand corner of the data sheet. The following is the definition of these markings:

Data Sheet Marking	Description
Product Preview	Contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product becomes available.
Advanced Information	Contains information on products being sampled or in the initial production phase of development.*
Preliminary	Contains preliminary information on new products in production.*
No Marking	Contains information on products in full production.*

*Specifications within these data sheets are subject to change without notice. Verify with your local Intel sales office that you have the latest data sheet before finalizing a design.

Memory Technologies 1

Flash Overview 2

Flash Memory Components 3

Flash Memory Systems 4

Automotive Memory 5

Table of Contents

Alphanumeric Index ... xiii

CHAPTER 1

Memory Technologies

Intel Memory Technologies ... 1-1

CHAPTER 2

Flash Overview

Flash Overview ... 2-1

CHAPTER 3

Flash Memory Components

DATA SHEETS

28F256A 256K (32K x 8) CMOS Flash Memory ... 3-1
28F512 512K (64K x 8) CMOS Flash Memory ... 3-26
28F010 1024K (128K x 8) CMOS Flash Memory ... 3-54
28F020 2048K (256K x 8) CMOS Flash Memory ... 3-84
28F001BX-T/28F001BX-B 1M (128K x 8) CMOS Flash Memory ... 3-117
28F200BX-T/B, 28F002BX-T/B 2 Mbit (128K x 16, 256K x 8) Boot Block Flash Memory Family ... 3-146
28F200BX-TL/BL, 28F002BX-TL/BL 2 Mbit (128K x 16, 256K x 8) Low Power Boot Block Flash Memory Family ... 3-192
28F400BX-T/B, 28F004BX-T/B 4 Mbit (256K x 16, 512K x 8) Boot Block Flash Memory Family ... 3-233
28F008SA 8 Mbit (1 Mbit x 8) Flash Memory ... 3-281

APPLICATION NOTES

AP-316 Using Flash Memory for In-System Reprogrammable Nonvolatile Storage . 3-309
AP-341 Designing an Updatable BIOS Using Flash Memory ... 3-354
AP-357 Power Supply Solutions for Flash Memory ... 3-397
AP-359 28F008SA Hardware Interfacing ... 3-432
AP-360 28F008SA Software Drivers ... 3-443
AP-363 Extended Flash Bios Design For Portable Computers ... 3-465
AP-364 28F008SA Automation and Algorithms ... 3-494
AP-371 Designing Flash Card Readiness Into Today's System ... 3-508
AB-29 Flash Memory Applications in Laser Printers ... 3-516

ENGINEERING REPORTS

ER-20 ETOX™ II Flash Memory Technology ... 3-523
ER-24 Intel Flash Memory 28F256A, 28F512, 28F010, 28F020 ... 3-528
ER-26 The Intel 28F001BX-T and 28F001BX-B Flash Memories ... 3-551
ER-27 The Intel 28F008SA Flash Memory ... 3-566
ER-28 ETOX™ III Flash Memory Technology ... 3-591
ER-29 The Intel 2/4 Mbit Boot Block Flash Memory Family ... 3-604

RELIABILITY REPORTS

RR-60 ETOX™ II Flash Memory Reliability Data Summary ... 3-633
RR-69 28F008SA Reliability Summary ... 3-670

ARTICLE REPRINTS

AR-470 A 90 ns 100K Erase/Program Cycle Megabit Flash Memory ... 3-678
AR-472 The Memory Driver ... 3-682
AR-478 Flash Memories: The Best of Two Worlds ... 3-683
AR-486 Flash Outshines ROM & EPROM ... 3-688
AR-491 Memory Breakthrough Drives Miniaturization ... 3-694
AR-493 Look Out EPROMs, Here Comes Flash ... 3-696
AR-494 Laptop Vendors Join the Flash Bandwagon ... 3-700
AR-495 Store Data in a Flash ... 3-702

Table of Contents (Continued)

AR-497 Designing with Flash Memory 3-708

SUPPORT TOOLS

Boot Block Flash: The Next Generation White Paper 3-718

Intel FlashFile™ Memory-The Key to Diskless Mobile PCs 3-722

SOP Guide 3-728

Intel Blocked Flash Memory 28F001BX-B/28F001BX-T Product Brief 3-734

Intel 2 Megabit Boot Block Flash Memory 28F200BX-T/B, 28F002BX-T/B Product Brief 3-736

Intel 4 Megabit Boot Block Flash Memory 28F400BX-T/B, 28F004BX-T/B Product Brief 3-738

Intel 28F008SA (8 Mbit) FlashFile™ Memory Product Brief 3-740

Intel Flash Memory Evaluation Kit II Product Brief 3-742

Intel 28F008SA FlashFile Memory Evaluation Module (D,FLASHEVAL4) Product Brief 3-743

Intel 28F400BX/28F004BX Flash Memory Evaluation Module (D,FLASHEVAL5) Product Brief 3-744

CHAPTER 4

Flash Memory Systems

DATA SHEETS

iMC001FLKA 1-Megabyte Flash Memory Card 4-1

iMC002FLKA 2-Mbyte Flash Memory Card 4-31

iMC004FLKA 4-Megabyte Flash Memory Card 4-61

Series 2 Flash Memory Cards 4-91

iSM001FLKA 1 Mbyte (512K x 16) CMOS Flash SIMM 4-129

iSM002FLKA 2 Mbyte (1024K x 16) CMOS Flash SIMM 4-163

APPLICATION NOTES

AP-343 Solutions for High Density Applications Using Intel Flash Memory 4-199

AP-361 Implementing the Integrated Registers of the Series 2 Flash Memory Card. 4-267

RELIABILITY REPORT

RR-70 Flash Memory Card Quality/Reliability Data Summary 4-284

ARTICLE REPRINTS

AR-484 PC Standard in the Cards 4-295

AR-490 Flash Memory Operates 10–20 Times Longer 4-300

AR-498 Nonvolatile, In-Circuit-Reprogrammable Memories 4-302

AR-700 Flash Memory Cards 4-312

AR-701 Execute-In-Place 4-316

AR-702 Interfacing Microsoft's Flash File Systems 4-318

SUPPORT TOOLS

Flash Memory Card White Paper 4-326

Intel FlashFile™ Memory-The Key to Diskless Mobile PCs 4-331

Flash Memory Card Product Brief 4-337

Series 2 Flash Memory Card Product Brief 4-338

Intel PC CardKit Product Brief 4-340

Intel Flash Memory Card/SIMM Evaluation Module (D,FLASHEVAL3) Product Brief 4-341

CHAPTER 5

Automotive Memory

A28F256A 256K (32K x 8) CMOS Flash Memory (Automotive) 5-1

A28F512 512K (64K x 8) CMOS Flash Memory (Automotive) 5-25

A28F010 1024K (128 x 8) CMOS Flash Memory (Automotive) 5-49

A27C256 256K (32K x 8) CHMOS EPROM (Automotive) 5-72

A87C257 256K (32K x 8) CHMOS Latched EPROM (Automotive) 5-83

Alphanumeric Index

28F001BX-T/28F001BX-B 1M (128K x 8) CMOS Flash Memory 3-117
28F008SA 8 Mbit (1 Mbit x 8) Flash Memory 3-281
28F010 1024K (128K x 8) CMOS Flash Memory 3-54
28F020 2048K (256K x 8) CMOS Flash Memory 3-84
28F200BX-T/B, 28F002BX-T/B 2 Mbit (128K x 16, 256K x 8) Boot Block Flash Memory Family 3-146
28F200BX-TL/BL, 28F002BX-TL/BL 2 Mbit (128K x 16, 256K x 8) Low Power Boot Block Flash Memory Family 3-192
28F256A 256K (32K x 8) CMOS Flash Memory 3-1
28F400BX-T/B, 28F004BX-T/B 4 Mbit (256K x 16, 512K x 8) Boot Block Flash Memory Family 3-233
28F512 512K (64K x 8) CMOS Flash Memory 3-26
A27C256 256K (32K x 8) CHMOS EPROM (Automotive) 5-72
A28F010 1024K (128 x 8) CMOS Flash Memory (Automotive) 5-49
A28F256A 256K (32K x 8) CMOS Flash Memory (Automotive) 5-1
A28F512 512K (64K x 8) CMOS Flash Memory (Automotive) 5-25
A87C257 256K (32K x 8) CHMOS Latched EPROM (Automotive) 5-83
AB-29 Flash Memory Applications in Laser Printers 3-516
AP-316 Using Flash Memory for In-System Reprogrammable Nonvolatile Storage 3-309
AP-341 Designing an Updatable BIOS Using Flash Memory 3-354
AP-343 Solutions for High Density Applications Using Intel Flash Memory 4-199
AP-357 Power Supply Solutions for Flash Memory 3-397
AP-359 28F008SA Hardware Interfacing 3-432
AP-360 28F008SA Software Drivers 3-443
AP-361 Implementing the Integrated Registers of the Series 2 Flash Memory Card 4-267
AP-363 Extended Flash Bios Design For Portable Computers 3-465
AP-364 28F008SA Automation and Algorithms 3-494
AP-371 Designing Flash Card Readiness Into Today's System 3-508
AR-470 A 90 ns 100K Erase/Program Cycle Megabit Flash Memory 3-678
AR-472 The Memory Driver 3-682
AR-478 Flash Memories: The Best of Two Worlds 3-683
AR-484 PC Standard in the Cards 4-295
AR-486 Flash Outshines ROM & EPROM 3-688
AR-490 Flash Memory Operates 10–20 Times Longer 4-300
AR-491 Memory Breakthrough Drives Miniaturization 3-694
AR-493 Look Out EPROMs, Here Comes Flash 3-696
AR-494 Laptop Vendors Join the Flash Bandwagon 3-700
AR-495 Store Data in a Flash 3-702
AR-497 Designing with Flash Memory 3-708
AR-498 Nonvolatile, In-Circuit-Reprogrammable Memories 4-302
AR-700 Flash Memory Cards 4-312
AR-701 Execute-In-Place 4-316
AR-702 Interfacing Microsoft's Flash File Systems 4-318
Boot Block Flash: The Next Generation White Paper 3-718
ER-20 ETOX™ II Flash Memory Technology 3-523
ER-24 Intel Flash Memory 28F256A, 28F512, 28F010, 28F020 3-528
ER-26 The Intel 28F001BX-T and 28F001BX-B Flash Memories 3-551
ER-27 The Intel 28F008SA Flash Memory 3-566
ER-28 ETOX™ III Flash Memory Technology 3-591
ER-29 The Intel 2/4 Mbit Boot Block Flash Memory Family 3-604
Flash Memory Card Product Brief 4-337
Flash Memory Card White Paper 4-326
Flash Overview 2-1
iMC001FLKA 1-Megabyte Flash Memory Card 4-1

Alphanumeric Index (Continued)

iMC002FLKA 2-Mbyte Flash Memory Card 4-31
iMC004FLKA 4-Megabyte Flash Memory Card 4-61
Intel 2 Megabit Boot Block Flash Memory 28F200BX-T/B, 28F002BX-T/B Product Brief .. 3-736
Intel 28F008SA (8 Mbit) FlashFile™ Memory Product Brief 3-740
Intel 28F008SA FlashFile Memory Evaluation Module (D,FLASHEVAL4) Product Brief 3-743
Intel 28F400BX/28F004BX Flash Memory Evaluation Module (D,FLASHEVAL5) Product Brief 3-744
Intel 4 Megabit Boot Block Flash Memory 28F400BX-T/B, 28F004BX-T/B Product Brief .. 3-738
Intel Blocked Flash Memory 28F001BX-B/28F001BX-T Product Brief 3-734
Intel Flash Memory Card/SIMM Evaluation Module (D,FLASHEVAL3) Product Brief 4-341
Intel Flash Memory Evaluation Kit II Product Brief 3-742
Intel FlashFile™ Memory-The Key to Diskless Mobile PCs 4-331
Intel FlashFile™ Memory-The Key to Diskless Mobile PCs 3-722
Intel Memory Technologies 1-1
Intel PC CardKit Product Brief 4-340
iSM001FLKA 1 Mbyte (512K x 16) CMOS Flash SIMM 4-129
iSM002FLKA 2 Mbyte (1024K x 16) CMOS Flash SIMM 4-163
RR-60 ETOX™ II Flash Memory Reliability Data Summary 3-633
RR-69 28F008SA Reliability Summary 3-670
RR-70 Flash Memory Card Quality/Reliability Data Summary 4-284
Series 2 Flash Memory Card Product Brief 4-338
Series 2 Flash Memory Cards 4-91
SOP Guide 3-728

Memory Technologies 1

intel®

INTEL MEMORY TECHNOLOGIES

Most of this handbook is devoted to techniques and information to help you design and implement semiconductor memory in your application or system. In this section, however, the memory chip itself will be examined and the processing technology required to turn a bare slice of silicon into high performance memory devices is described. The discussion has been limited to the basics of MOS (Metal Oxide Semiconductor) technologies as they are responsible for the majority of memory devices manufactured at Intel.

There are three major MOS technology families—PMOS, NMOS, and CMOS (Figure 1). They refer to the channel type of the MOS transistors made with the technology. PMOS technologies implement p-channel transistors by diffusing p-type dopants (usually boron) into an n-type silicon substrate to form the source and drain. P-channel is so named because the channel is comprised of positively charged carriers. NMOS technologies are similar, but use n-type dopants (normally phosphorus or arsenic) to make n-channel transistors in p-type silicon substrates. N-channel is so named because the channel is comprised of negatively charged carriers. CMOS or Complementary MOS technologies combine both p-channel and n-channel devices on the same silicon. Either p- or n-type silicon substrates can be used, however, deep areas of the opposite doping type (called wells) must be defined to allow fabrication of the complementary transistor type.

Most of the early semiconductor memory devices, like Intel's pioneering 1103 dynamic RAM and 1702 EPROM were made with PMOS technologies. As higher speeds and greater densities were needed, most new devices were implemented with NMOS. This was due to the inherently higher speed of n-channel charge carriers (electrons) in silicon along with improved process margins. CMOS technology has begun to see widespread commercial use in memory devices. It allows for very low power devices used for battery operated or battery back-up applications. Historically, CMOS has been slower than any NMOS device. However, CMOS technology has been improved to produce higher speed devices. The extra cost of processing required to make both transistor types had kept CMOS memories limited to those areas where the technology's special characteristics would justify the extra cost. In the future, the learning curve for high performance CMOS costs are making a larger number of memory devices practical in CMOS.

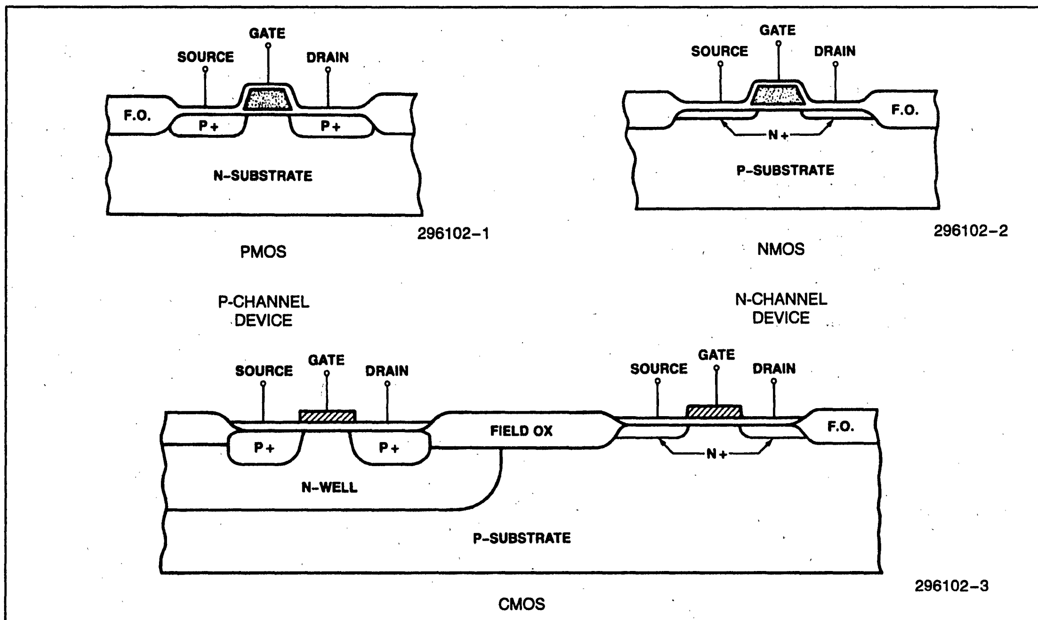


Figure 1. MOS Process Cross-sections

In the following section, the basic fabrication sequence for an HMOS circuit will be described. HMOS is a high performance n-channel MOS process developed by Intel for 5V single supply circuits. HMOS, and CHMOS, CHMOS-E (EPROM) and ETOX™ (Flash Memory), along with their evolutionary counterparts comprise the process family responsible for most of the memory components produced by Intel today.

The MOS IC fabrication process begins with a slice (or wafer) of single crystal silicon. Typically, it's 100 or 150 millimeter in diameter, about a half millimeter thick, and uniformly doped p-type. The wafer is then oxidized in a furnace at around 1000°C to grow a thin layer of silicon dioxide (SiO_2) on the surface. Silicon nitride is then deposited on the oxidized wafer in a gas phase chemical reactor. The wafer is now ready to receive the first pattern of what is to become a many layered complex circuit. The pattern is etched into the silicon nitride using a process known as photolithography, which will be described in a later section. This first pattern (Figure 2) defines the boundaries of the active regions of the IC, where transistors, capacitors, diffused resistors, and first level interconnects will be made.

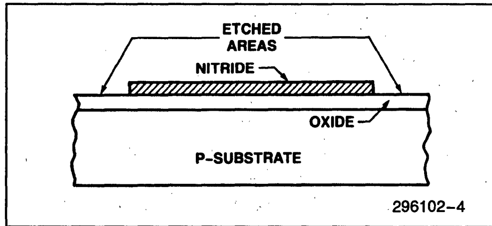


Figure 2. First Mask

The patterned and etched wafer is then implanted with additional boron atoms accelerated at high energy. The boron will only reach the silicon substrate where the nitride and oxide was etched away, providing areas doped strongly p-type that will electrically separate active areas. After implanting, the wafers are oxidized again and this time a thick oxide is grown. The oxide only grows in the etched areas due to silicon nitride's properties as an oxidation barrier. When the oxide is grown, some of the silicon substrate is consumed and this gives a physical as well as electrical isolation for adjacent devices as can be seen in Figure 3.

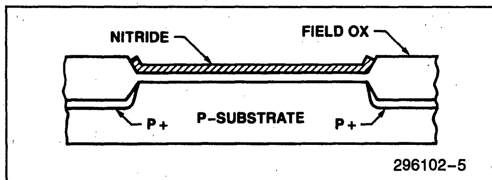


Figure 3. Post Field Oxidation

Having fulfilled its purpose, the remaining silicon nitride layer is removed. A light oxide etch follows taking with it the underlying first oxide but leaving the thick (field) oxide.

Now that the areas for active transistors have been defined and isolated, the transistor types needed can be determined. The wafer is again patterned and then if special characteristics (such as depletion mode operation) are required, it is implanted with dopant atoms. The energy and dose at which the dopant atoms are implanted determines much of the transistor's characteristics. The type of the dopant provides for depletion mode (n-type) or enhancement mode (p-type) operation.

The transistor types defined, the gate oxide of the active transistors are grown in a high temperature furnace. Special care must be taken to prevent contamination or inclusion of defects in the oxide and to ensure uniform consistent thickness. This is important to provide precise, reliable device characteristics. The gate oxide layer is then masked and holes are etched to provide for direct gate to diffusion ("buried") contacts where needed.

The wafers are now deposited with a layer of gate material. This is typically poly crystaline silicon ("poly") which is deposited in a gas phase chemical reactor similar to that used for silicon nitride. The poly is then doped (usually with phosphorus) to bring the sheet resistance down to 10–20 Ω/square. This layer is also used for circuit interconnects and if a lower resistance is required, a refractory metal/polysilicon composite or refractory metal silicide can be used instead. The gate layer is then patterned to define the actual transistor gates and interconnect paths (Figure 4).

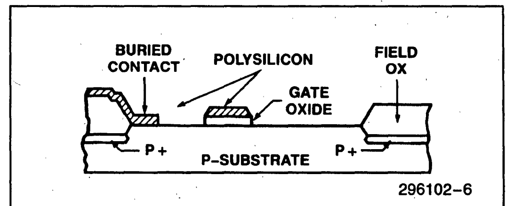


Figure 4. Post Gate Mask

The wafer is next diffused with n-type dopant (typically arsenic or phosphorus) to form the source and drain junctions. The transistor gate material acts as a barrier to the dopant providing an undiffused channel self-aligned to the two junctions. The wafer is then oxidized to seal the junctions from contamination with a layer of SiO_2 (Figure 5).

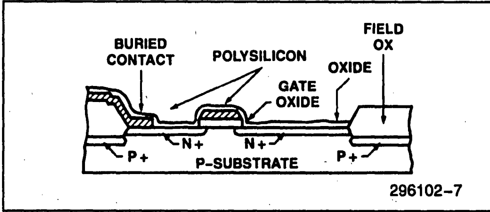


Figure 5. Post Oxidation

A thick layer glass is then deposited over the wafer to provide for insulation and sufficiently low capacitance between the underlying layers and the metal interconnect signals. (The lower the capacitance, the higher the inherent speed of the device.) The glass layer is then patterned with contact holes and placed in a high temperature furnace. This furnace step smooths the glass surface and rounds the contact edges to provide uniform metal coverage. Metal (usually aluminum or aluminum/silicon) is then deposited on the wafer and the interconnect patterns and external bonding pads are defined and etched (Figure 6). The wafers then receive a low temperature (approximately 500°C) alloy that insures good ohmic contact between the aluminum and diffusion or poly.

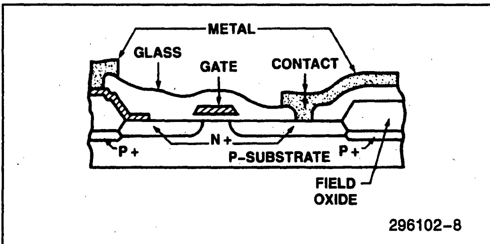


Figure 6. Complete Circuit (without passivation)

At this point the circuit is fully operational, however, the top metal layer is very soft and easily damaged by handling. The device is also susceptible to contamination or attack from moisture. To prevent this the wafers are sealed with a passivation layer of silicon nitride or a silicon and phosphorus oxide composite. Patterning is done for the last time opening up windows only over the bond pads where external connections will be made.

This completes basic fabrication sequence for a single poly layer process. Double poly processes such as those used for high density Dynamic RAMs, EPROMs, flash memories, and EEPROMs follow the same general process flow with the addition of gate, poly deposition, doping, and interlayer dielectric process modules required for the additional poly layer (Figure 7). These steps are performed right after the active areas have been defined (Figure 3) providing the capacitor or floating gate storage nodes on those devices.

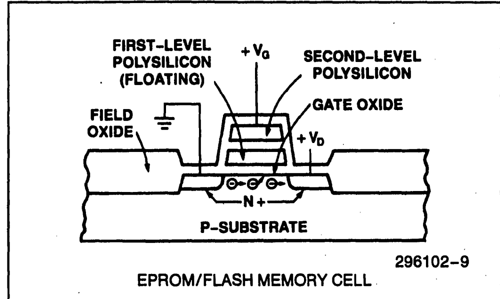


Figure 7. Double Poly Structure

After fabrication is complete, the wafers are sent for testing. Each circuit is tested individually under conditions designed to determine which circuits will operate properly both at low temperature and at conditions found in actual operation. Circuits that fail these tests are inked to distinguish them from good circuits. From here the wafers are sent from assembly where they are sawed into individual circuits with a paper-thin diamond blade. The inked circuits are then separated out and the good circuits are sent on for packaging.

Packages fall into two categories—hermetic and non-hermetic. Hermetic packages are Cerdip, where two ceramic halves are sealed with a glass fritt, or ceramic with soldered metal lids. An example of hermetic package assembly is shown in Table 1. Non-hermetic packages are molded plastics.

The ceramic package has two parts, the base, which has the leads and die (or circuit) cavity, and the metal lid. The base is placed on a heater block and a metal alloy preform is inserted. The die is placed on top of the preform which bonds it to the package. Once attached, wires are bonded to the circuit and then connected to the leads. Finally the package is placed in a dry inert atmosphere and the lid is soldered on.

The cerdip package consists of a base, lead frame, and lid. The base is placed on a heater block and the lead frame placed on top. This sets the lead frame in glass attached to the base. The die is then attached and bonded to the leads. Finally the lid is placed on the package and it is inserted in a seal furnace where the glass on the two halves melt together making a hermetic package.

In a plastic package, the key component is the lead frame. The die is attached to a pad on the lead frame and bonded out to the leads with gold wires. The frame then goes to an injection molding machine and the package is formed around the lead frame. After mold the excess plastic is removed and the leads trimmed.

After assembly, the individual circuits are retested at an elevated operating temperature to assure critical operating parameters and separated according to speed and power consumption into individual specification groups. The finished circuits are marked and then readied for shipment.

The basic process flow described above may make VLSI device fabrication sound straightforward, however, there are actually hundreds of individual operations that must be performed correctly to complete a working circuit. It usually takes well over two months to complete all these operations and the many tests and measurements involved throughout the manufacturing process. Many of these details are responsible for ensuring the performance, quality, and reliability you expect from Intel products. The following sections will discuss the technology underlying each of the major process elements mentioned in the basic process flow.

PHOTOLITHOGRAPHY

The photo or masking technology is the most important part of the manufacturing flow if for no other reason than the number of times it is applied to each wafer. The manufacturing process gets more complex in order to make smaller and higher performance circuits. As this happens the number of masking steps increases, the features get smaller, and the tolerance required becomes tighter. This is largely because the minimum size of individual pattern elements determine the size of the whole circuit, effecting its cost and limiting its potential complexity. Early MOS IC's used minimum geometries (lines or spaces) of 8–10 microns (1 micron = 10^{-6} meter ≅ 1/25,000 inch). The n-channel processes of the mid 1970's brought this down to approximately 5 microns, and today minimum geometries of one micron are in production. This dramatic reduction

Table 1. Typical Hermetic Package Assembly

Flow	Process/Materials	Typical Item	Frequency	Criteria
	Wafer			
	Die saw, wafer break			
	Die wash and plate			
	Die visual inspection	Passivation, metal	100% of die	
	QA gate		Every lot	0/76, LTPD = 5%
	Die attach (Process monitor)	Wet out	4 x /operator/shift	0/11 LTPD = 20%
	Post die attach visual		100% of devices	
	Wire bond (Process monitor)	Orientation, lead dressing, etc.	4 x /operator/ machine/shift	
	Post bond inspection		100% devices	
	QA gate	All previous items	every lot	1/129, LTPD = 3%
	Seal and Mark (Process monitor)	Cap align, glass integrity, moisture	4 x /furnace/shift	0/15, LTPD = 15%
	Temp cycle		10 x to mil std. 883 cond. C	1/11, LTPD = 20%
	Hermeticity check (Process monitor)	F/G leak	100% devices	
	Lead Trim (Process monitor)	Burrs, etc. (visual) Fine leak	4 x /station/shift 2 x /station/shift	0/15, LTPD = 15% 1/129, LTPD = 3%
	External visual	Solder voids, cap alignment, etc.	100% devices	
1.	QA gate	All previous items	All lots	1/129, LTPD = 3%
	Class test (Process monitor)	Run standards (good and reject) Calibrate every system using "autover" program	Every 48 hrs.	
2.	Mark and Pack			
	Final QA	(See attached)		

296102–11

NOTES:
1. Units for assembly reliability monitor.
2. Units for product reliability monitor.

in feature size was achieved using the newer high resolution photo resists and optimizing their processing to match improved optical printing systems.

A second major factor in determining the size of the circuit is the registration or overlay error. This is how accurately one pattern can be aligned to a previous one. Design rules require that space be left in all directions according to the overlay error so that unrelated patterns do not overlap or interfere with one another. As the error space increases the circuit size increases dramatically. Only a few years ago standard alignment tolerances were $\geq \pm 2$ microns; now advanced Intel processes have reduced this dramatically due mostly to the use of advanced projection and step and repeat exposure equipment.

The wafer that is ready for patterning must go through many individual steps before that pattern is complete. First the wafer is baked to remove moisture from its surface and is then treated with chemicals that ensure good resist adhesion. The thick photoresist liquid is then applied and the wafer is spun flat to give a uniform coating, critical for high resolution. The wafer is baked at a low temperature to solidify the resist into gel. It is then exposed with a machine that aligns a mask with the new pattern on it to a previously defined layer. The photo-resist will replicate this pattern on the wafer.

Negative working resists are polymerized by the light and the unexposed resist can be rinsed off with solvents. Positive working resists use photosensitive polymerization inhibitors that allow a chemically reactive developer to remove the exposed areas. The positive resists require much tighter control of exposure and development but yield higher resolution patterns than negative resistance systems.

The wafer is now ready to have its pattern etched. The etch procedure is specialized for each layer to be etched. Wet chemical etchants such as hydrofluoric acid for silicon oxide or phosphoric acid for aluminum are often used for this. The need for smaller features and tighter control of etched dimensions is increasing the use of plasma etching in fabrication. Here a reactor is run with a partial vacuum into which etchant gases are introduced and an electrical field is applied. This yields a reactive plasma which etches the required layer.

The wafer is now ready for the next process step. Its single journey through the masking process required the careful engineering of mechanics, optics, organic chemistry, inorganic chemistry, plasma chemistry, physics, and electronics.

DIFFUSION

The picture of clean room garbed operators tending furnace tubes glowing cherry red is the one most often associated with IC fabrication. These furnace operations are referred to collectively as diffusion because they employ the principle of sold state diffusion of matter to accomplish their results. In MOS processing, there are three main types of diffusion operations: predeps, drives, and oxidations.

Predeposition, or "predep," is an operation where a dopant is introduced into the furnace from a solid, liquid, or gaseous source and at the furnace temperature (usually 900°C–1200°C) a saturated solution is formed at the silicon surface. The temperature of the furnace, the dopant atom, and rate of introduction are all engineered to give a specific dose of the dopant on the wafer. Once this is completed the wafer is given a drive cycle where the dopant left at the surface by the predep is driven into the wafer by high temperatures. These are generally at different temperatures than the predeps and are designed to give the required junction depth and concentration profile.

Oxidation, the third category, is used at many steps of the process as was shown in the process flow. The temperature and oxidizing ambient can range from 800°C to 1200°C and from pure oxygen to mixtures of oxygen and other gases to steam depending on the type of oxide required. Gate oxides require high dielectric breakdown strength for thin layers (between 0.01 and 0.1 micron) and very tight control over thickness (typically ± 0.005 micron or less than $\pm 1/5{,}000{,}000$ inch), while isolation oxides need to be quite thick and because of this their dielectric breakdown strength per unit thickness is much less important.

The properties of the diffused junctions and oxides are key to the performance and reliability of the finished device so the diffusion operations must be extremely well controlled for accuracy, consistency and purity.

ION IMPLANT

Intel's high performance products require such high accuracy and repeatability of dopant control that even the high degree of control provided by diffusion operations is inadequate. However, this limitation has been overcome by replacing critical predeps with ion implantation. In ion implantation, ionized dopant atoms are accelerated by an electric field

and implanted directly into the wafer. The acceleration potential determines the depth to which the dopant is implanted.

The charged ions can be counted electrically during implantation giving very tight control over dose. The ion implanters used to perform this are a combination of high vacuum system, ion source, mass spectrometer, linear accelerator, ultra high resolution current integrator, and ion beam scanner. You can see that this important technique requires a host of sophisticated technologies to support it.

THIN FILMS

Thin film depositions make up most of the features on the completed circuit. They include the silicon nitride for defining isolation, polysilicon for the gate and interconnections, the glass for interlayer dielectric, metal for interconnection and external connections, and passivation layers. Thin film depositions are done by two main methods: physical deposition and chemical vapor deposition. Physical deposition is most common for deposition metal. Physical depositions are performed in a vacuum and are accomplished by vaporizing the metal with a high energy electron beam and redepositing it on the wafer or by sputtering it from a target to the wafer under an electric field.

Chemical vapor deposition can be done at atmospheric pressure or under a moderate vacuum. This type of deposition is performed when chemical gases react at the wafer surface and deposit a solid film of the reaction product. These reactors, unlike their general industrial counterparts, must be controlled on a microscale to provide exact chemical and physical properties for thin films such as silicon dioxide, silicon nitride, and polysilicon.

The fabrication of modern memory devices is a long, complex process where each step must be monitored, measured and verified. Developing a totally new manufacturing process for each new product or even product line takes a long time and involves significant risk. Because of this, Intel has developed process families, such as HMOS, on which a wide variety of devices can be made. These families are scalable so that circuits need not be totally redesigned to meet your needs for higher performance.[1] They are evolutionary so that development time of new processes and products can be reduced without compromising Intel's commitment to consistency, quality, and reliability.

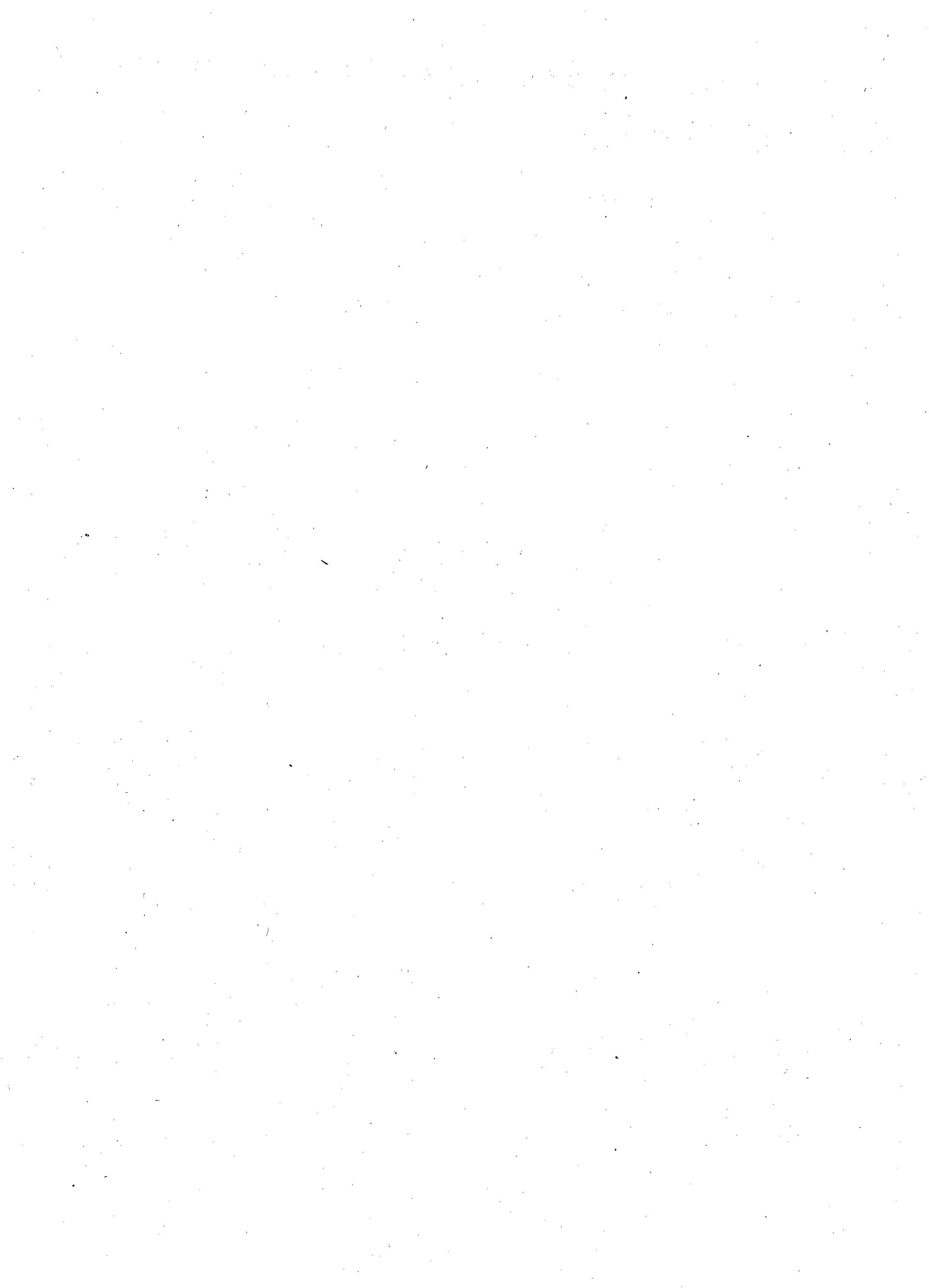
The manufacture of today's MOS memory devices requires a tremendous variety of technologies and manufacturing techniques, many more than could be mentioned here. Each requires a team of experts to design, optimize, control and maintain it. All these people and thousands of others involved in engineering, design, testing and production stand behind Intel's products.

Because of these extensive requirements, most manufacturers have not been able to realize their needs for custom circuits on high performance, high reliability processes. To address this Intel's expertise in this area is now available to industry through the silicon foundry. Intel supplies design rules and support to design and debug circuits. This includes access to Intel's n-well CHMOS technology. Users of the foundry can now benefit from advanced technology without developing processes and IC manufacturing capability themselves.

[1]R. Pashley, K. Kokkonen, E. Boleky, R. Jecmen, S. Liu, and W. Owen, "H-MOS Scales Traditional Devices to Higher Performance Level," *Electronics*, August 18, 1977.

Flash Overview

2



Flash Memory Overview

The ideal memory system optimizes density, nonvolatility, fast readability and cost effectiveness. While traditional memory technologies may individually exhibit one or more of these desired characteristics, no single memory technology has achieved all of them without major tradeoffs—until the introduction of Intel Flash Memory.

WHAT IS FLASH MEMORY?

Introduced by Intel in 1988, ETOX™ flash memory is a high-density, truly nonvolatile and high performance read-write memory solution also characterized by low power consumption and extreme ruggedness and high reliability. The cost trend of Intel Flash Memory components continues to decline sharply due to: (1) manufacturing economies inherent in ETOX, Intel's industry-standard EPROM-based flash technology, (2) increases in memory density, and (3) rapid growth in production volume.

A comparison between Intel Flash Memory and other solid-state memory technologies underscores the fact that flash offers a design solution with distinct advantages. These advantages are key to future product differentiation for many applications requiring firmware updates or compact mass storage (Figure 1).

— ROM (read-only memory) is a mature, high density, nonvolatile, reliable and low cost memory technology widely used in PC and embedded applications. Once it is manufactured however, the contents of a ROM can never be altered. Additionally, initial ROM programming involves a time-consuming mask development process that requires stable code and is most cost-effective in high volumes.

Easy updatability makes flash memory clearly more flexible than ROM in most applications.

— SRAM (static random-access memory) is a high-speed, reprogrammable memory technology which is limited by its volatility and relatively low density. As a volatile memory technology, SRAM requires constant power to retain its contents. Built-in battery backup is therefore required when the main power source is turned off. Since battery failure is an inevitable fact of life, SRAM data loss is an ever-present danger. Additionally, SRAM requires four to six transistors to store one bit of information. This becomes a significant limitation in developing higher densities—effectively keeping SRAM cost relatively high.

Figure 1. Intel Flash Memory vs Traditional Memory Technologies

Memory	Inherently Non-Volatile	High Density	Low Power	One Transistor Cell	In-System Re-Writable	Code and Data Storage	Byte Alterable	Blocking	Hands off Updates
Flash	✓	✓	✓	✓	✓	✓		✓	✓
SRAM + Battery					✓	✓	✓	✓	✓
DRAM + Disk		✓			✓	✓	✓	✓	✓
EEPROM	✓		✓		✓	✓	✓	✓	✓
OTP/EPROM	✓	✓	✓	✓					
Masked ROM	✓	✓	✓	✓					

*MS-DOS and Windows are registered trademarks of Microsoft Corporation.

Order Number: 296101-004

In contrast, Intel flash memory is inherently nonvolatile, and the single transistor cell design of Intel's ETOX™ manufacturing process is extremely scalable, allowing the development of continuously higher densities and steady cost improvement over SRAM (Figure 2).

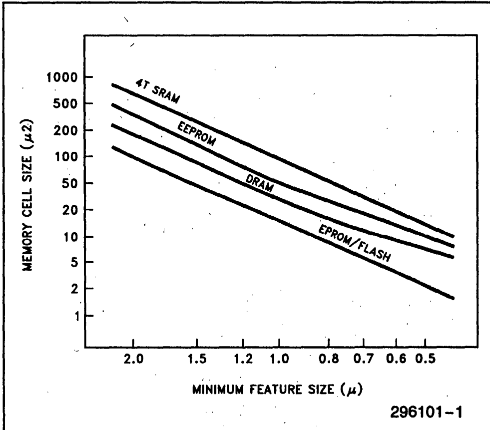


Figure 2

— EPROM (electrically programmable read-only memory) is a mature, high-density, nonvolatile technology which provides a degree of updatability not found in ROM. An OEM may program EPROM as needed to accommodate code changes or varying manufacturing unit quantities. Once programmed, however, the EPROM may only be erased by removing it from the system and then exposing the memory component to ultraviolet light—an impractical and time-consuming procedure for many OEMs and a virtually impossible task for end-users.

Unlike EPROM, flash memory is electrically re-writable within the host system, making it a much more flexible and easier to use alternative. Flash memory offers OEMs not only high density and nonvolatility, but higher functionality and the ability to differentiate their systems.

— EEPROM (electrically erasable programmable read-only memory) is nonvolatile and electrically byte-erasable. Such byte-alterability is needed in certain applications but involves a more complex cell structure, and significant trade-offs in terms of limited density, lower reliability and higher cost, making it unsuitable as a mainstream memory.

Unlike EEPROM, Intel flash memory technology utilizes a one-transistor cell, allowing higher densities, scalability, lower cost, and higher reliability, while taking advantage of in-system, electrical erasability (Figure 3).

	Intel ETOX™ Flash	EEPROM
Transistors	1	2
Cell Size (1-Micro Lithogrpahy)	15μ	38μ
Cycling Features	0.1%	5%

Figure 3

— DRAM (dynamic random access memory) is a volatile memory known for its density and low cost. Because of its volatility, however, it requires not only a constant power supply to retain data, but also an archival storage technology, such as disk, to back it up.

Partnered with hard disks for permanent mass storage, DRAM technology has provided a low-cost, yet space and power-hungry solution for today's PCs.

With ETOX process technology, Intel manufactures a flash memory cell that is 30% smaller than equivalent DRAM cells. Flash memory's scalability offers a price advantage as well, keeping price parity with DRAM, and also becoming more attractive as a hard disk replacement in portable systems as densities grow and costs decline.

Intel flash memory combines advantages from each of these memory technologies. In embedded memory applications, flash memory provides higher-performance and more flexibility than ROM and EPROM, while providing higher density and better cost effectiveness than battery-backed SRAM and EEPROM. Moreover, the true nonvolatility and low power consumption characteristics of flash memory make it a compelling alternative to DRAM in many applications.

ETOX™ III TECHNOLOGY

Unlike other approaches to flash memory, Intel ETOX is a proven technology. As its name suggests, ETOX (or "EPROM tunnel oxide") technology evolved from EPROM. With 95% process compatibility with EPROM, ETOX taps experience gained from a mature high-volume manufacturing base pioneered by Intel in the 1970s.

Data retention and lifetime reliability statistics for ETOX III flash are equivalent to those of EPROM. Representing the third generation of Intel flash memory technology, the ETOX III 0.8μ process provides <100 FITS (failures in time) @ 55°C in a specification that delivers 100,000 write cycles *per block*. This capability significantly exceeds the cycling requirements of even the most demanding applications.

For example, code storage for embedded control programs used in standard computer applications is infrequently updated. Twenty-year system lifetimes may require fewer than 100 rewrites. Even routinely-changed data tables (used in navigational computers and black box controllers) only require about 1,000 write cycles over a 20-year period. The most demanding flash memory application of all, archival data storage in PC applications, typically requires about 5,000 write cycles in 20 years.

ETOX flash memory's simple single-transistor cell structure makes it smaller than other flash cells, allowing designers to create highly integrated systems which are more reliable and cost-effective than those based on more complex and less mature flash technologies. The inherent scalability of ETOX III Flash Memory and high-volume manufacturing is enabling a corresponding downtrend in flash cost.

Flash memory has added a new dimension to nonvolatile memory applications. Embedded systems, such as PC BIOS, hard disk drive controllers and cellular telephone applications take advantage of the easy update capability, high density and high performance of Intel Flash Memory. Today's new generation of portable computers require the optimum combination of performance, size, weight, low power and shock resistance. Whether implemented in memory cards, solid state disks or at the component level, Intel's Flash Memories are also enabling a whole new generation of mobile computers.

IMPLEMENTING INTEL FLASH MEMORY

Today, Intel continues to serve both updatable nonvolatile memory applications as well as the rapidly emerging solid-state mass storage applications with flash memory solutions tailored to meet the needs of these markets.

Updatable Code Storage

Code and data storage comprise the updatable nonvolatile memory applications that require high performance, high density, and easy update capability. Because these applications are not updated as frequently as solid-state mass storage applications, erase/write cycles are not as critical as integration and performance requirements. This application segment is served effectively with full chip-erase or Boot Block products.

Intel's 28F001BX 1 Mbit Boot Block flash component, featuring a sectored architecture, has been widely accepted in embedded code storage applications, particularly in PC BIOS and cellular communications. By adopting Boot Block for their products, over 20 PC manufacturers have gained added flexibility and the ability to differentiate in a highly competitive market. End users also benefit from the ability to upgrade BIOS software quickly and securely. The blocked architecture allows the OEM to store critical system code securely in the lockable "boot block" of the device that can minimally bring up the system and download to other locations of the device to initialize the system. The hardware boot locking feature guarantees that even if the power is disrupted during a BIOS update, the system will be able to recover immediately.

In response to customer requests for speed, density, low power, surface-mount options and an industry-standard upgrade path for portable computing and telecommunications, Intel more recently introduced the 2 Mbit 28F200BX and 4 Mbit 28F400BX Boot Block products.

These products offer 60 ns performance; two surface mount packages: 40-lead TSOP (X8 only) and 44-lead PSOP; and a proprietary Boot Block architecture similar to the 1 Mbit Boot Block device. The Boot Block stores the code necessary to initialize the system, while parameter blocks can be used to store manufacturing product code, setup parameters, and frequently updated code such as system diagnostics. The main operating code is stored in the main blocks. Both devices are available in a x16/x8 ROM-compatible pin-out in 44-lead PSOP surface mount packaging. These pinouts and packages allow an easy upgrade from 2 Mb to 4 Mb, since only one address is added to the 4 Mb device.

Solid-State Mass Storage

This major application segment requires very high density memory, automated programming and high-performance erase/write capability at a very low cost per bit. Erasing and writing portions of the code or data is much more frequent in solid-state mass storage than in updatable firmware applications.

Intel's symmetrically blocked 8 Mbit 28F008SA FlashFile™ memory is the highest density nonvolatile read/write solution for solid-state mass storage. What's more, it is the first flash memory device optimized for solid-state storage of software and data files.

The 28F008SA is packaged in an advanced 40-lead TSOP (thin, small outline package) or 44-lead PSOP (plastic SOP) to provide the extremely small form factor required for today's handheld, pen-based and sub-

**Based on 10 MB card design, 5,000 cycle yields 50,000 MB of stored data, which far exceeds most usage environments and file system methodologies.*

notebook portable computers. The compactness of an 8 Mbit device in a TSOP package allows for high-density flash arrays to be included both on a system motherboard for direct execution of user programs or operating systems, as well as memory cards for transportable program and file storage.

Memory Cards

Intel's family of flash memory cards provides the most reliable and rugged form of removable memory media. High density, true nonvolatility, rewrite flexibility, and proven cost effectiveness make Intel Series 1 Flash Memory Cards the ideal medium for storing and updating application code as well as capturing data.

For file storage applications that require high performance, ruggedness, long battery life, small size and light weight, Intel's Series 2 Flash Memory Cards in 4-Mbyte, 10-Mbyte and 20-Mbyte densities provide the best solution. Based on Intel's 8-Mbit FlashFile memory components, the Series 2 card's block-erase functionality and high density take full advantage of flash filing systems like Microsoft*'s Flash File System software to provide full disk emulation in the form of removable, nonvolatile storage. The cards conform to the PCMCIA 2.0/JEIDA 4.1 68-pin standards and are compatible with Intel's Exchangeable Card Architecture (ExCA™) to ensure system-to-system interoperabililty.

THE IMPETUS BEHIND THE "SOLID-STATE" DISK

Because the disk-based PC is so prevalent and eminently familiar to both designers and end-users, many of today's portable systems still rely on it as their primary medium. At the same time, disk drive manufacturers have made great strides toward improving the reliability, size and performance of their systems, as well as the disk media themselves.

Yet the disk drive is an electro-mechanical system with inherent limitations. Any mechanical hardware is much more vulnerable than solid state semiconductor technology to the shock, vibration, and impurities that portable PCs encounter during normal use. Hard disk drives can typically withstand up to 10Gs of operating shock; Intel FlashFile memory, with no moving parts, can withstand as much as 1000Gs. Additionally, Intel's Series 2 Memory Cards feature approximately 1.6 million hours mean time between failure (MTBF). Such endurance and reliability is essential for many of today's truly mobile handheld palmtop and notebook sized PCs, particularly within applications requiring extreme data integrity.

Power consumption is another major consideration for today's mobile PC designer and user. The drive typically requires anywhere from 3 watts to as many as 8 watts of power to run—which means rapid drain of the system's batteries. Compare this to flash memory in a hard disk configuration. It consumes less than one two-hundredth the average power of a comparable magnetic disk drive based on the typical user model. At the chip level, the 8 Mbit FlashFile Memory component has a DEEP POWERDOWN mode that reduces power consumption to less than 0.2 μA.

Additional shortcomings of disk drives are their size, weight and floor costs. Magnetic drives do not scale well, that is, it becomes increasingly difficult to improve or even retain density as platter size shrinks. Thus, every reduction in drive size requires complete retooling and costly learning. Also, the complex controller circuitry provides a price floor under which magnetic drives cannot drop. Since flash is scalable, at some point in the near future, small magnetic drives are likely to become more expensive per Mbyte than flash cards and are certain to have less capacity.

	Disk/DRAM	Flash
Average Seek Time	28.0 ms	0
Latency	8.3 ms	0
Data Transfer Rate Read: Write:	 8 Mbits/sec. 8 Mbits/sec. . . . Now Read from RAM	 106.7 Mbits/sec. 1 Mbit/sec. Direct Processor Access
Total Time to Access (1 Kbyte File)	37.3 ms	0.077 ms

Figure 4

From a performance standpoint, disk-based systems still require some form of supplementary memory that is directly executable. Typically, DRAM is used for executable code storage and data manipulation. Data from the disk is downloadable into the DRAM cache before users can access the information. Then when a "save" operation is desired, the data is uploaded from DRAM back into the disk. This download/upload process slows down system throughput while the redundant memory media produce even more system overhead in the form of added space, power consumption and weight (Figure 4).

Today's PCs are typically configured with 4 Mbytes–8 Mbytes of DRAM backed up by at least a 40-Mbyte disk. FlashFile memory fully supports this system configuration when used simply as a magnetic drive replacement. Instructions and data are still swapped to DRAM, but at a faster rate. Plus execution speed can be enhanced if the DRAM is replaced with SRAM.

In the solid-state computer, the "DRAM + magnetic hard drive" is replaced by a "flash memory + SRAM". The key to this architecture is the ability to eXecute-In-Place (XIP). Program instructions stored in flash memory are read directly by the processor. Results are written directly to the flash memory. Compute-intensive operations that require the fastest memory access and byte-alterability can use high-speed SRAM or pseudo SRAM. Some of the system DRAM can be replaced by low-cost flash and a small part of the DRAM can be replaced by SRAM. The flash memory space is made even more storage efficient through the use of compression techniques which may offer up to 2:1 compression.

SOFTWARE DEVELOPMENTS POSITION FLASH FOR PORTABLE APPLICATIONS

The majority of today's portable computers and supporting software programmers are designed to run using Microsoft's MS-DOS* disk operating system. MS-DOS was developed to allow broad-based compatibility between systems and software and to optimize the sectoring scheme inherent to disk technology.

Intel's Flash Memory, based on a block-erase architecture, divides the flash memory into segments that are loosely analogous to the zones recognized in MS-DOS. Thanks to recent software developments, flash memory can effectively serve as the main memory within portable computers, providing user functions virtually identical to, and even improved over, those of disk-based systems.

Specifically, two recent developments allow this achievement: DOS in ROM-executable form (DOS was formerly designed to be stored on disk and then downloaded to/executed out of RAM); and a file system designed for flash memory technology that allows the devices to erase blocks of memory instead of the whole chip.

ROM-executable DOS provides several benefits to both system manufacturers and ultimately end users. First, since most of the operating system is composed of fixed code, the amount of system RAM required to execute DOS is reduced from 50K to 15K, thereby conserving system space and power. Secondly, DOS can now be permanently stored in and executed from a single ROM-type of device—flash memory—so systems come ready to run. Lastly, users enjoy "instant on" performance since the traditional disk-to-DRAM boot function and software downloading steps are eliminated.

For example, by storing application software and operating system code in a Resident Flash Array (RFA), users enjoy virtually instant-on performance and rapid in-place code execution. An RFA also protects against software obsolescence because it is in-system updatable. Resident software, stored in flash rather than disk, extends battery life and increases system reliability.

Since erasing and writing data to flash memory is a distinctly different operation than rewriting information to a disk, new software techniques were necessary to allow flash to emulate disk functionality. File management software like Microsoft*'s Flash File System (FFS) allows Intel's Flash Memory components and flash cards to emulate the file storage capabilities of disk. Microsoft's FFS transparently handles data swaps between flash blocks similar to the way MS-DOS handles swaps between disk sectors. Under FFS, the user may input a MS-DOS or Windows* command without regard for whether a flash memory or magnetic disk is installed in the system. The FFS also employs wear-leveling algorithms that prevent any block from being cycled excessively, thus assuring millions of hours of reliability. Flash filing systems make the management of flash memory devices completely transparent to the user.

CONCLUSION

Intel Flash Memory presents an entirely new memory technology alternative. As a high-density, nonvolatile read/write technology, it is exceptionally well-suited to serve as a solid-state disk or a cost-effective and highly reliable replacement for DRAMs and battery-backed static RAMs. Its inherent advantages over these technologies make it particularly useful in portable systems that require the utmost in low power, compact size, and ruggedness while maintaining high performance and full functionality.

Intel Flash Memory offers:

- **Inherent Nonvolatility:** Unlike static RAMs, no backup battery is required to ensure data retention. In contrast to DRAMs, flash requires no disk to provide backup storage of data, programs or files.
- **Cost-Effective High Density:** Today, Intel flash memories cost about the same as DRAMs and about one fourth of SRAMs on a per-bit basis. The broad acceptance of flash is driving manufacturing volumes up and costs down at an unprecedented rate, allowing flash to soon compete on a cost basis even with disk drive within the notebook, sub-notebook and palmtop markets.
- **Solid-State Performance:** Because it is a semiconductor technology, flash memory consumes much less power, is much lighter in weight and is smaller and more shock-resistant and reliable than disk drives. Mobile computers no longer have to drain the battery to run a disk drive motor or accommodate the disk assembly's added bulk and weight. Now users no longer have to be threatened with the possibility of losing their data after a disk crash when conditions become unusually rough.
- **Direct Execution:** Since no disk-to-DRAM download step, seek or latency times are incurred with flash memory, users enjoy significantly higher speed program and file access, as well as systems that turn on instantly to wherever the user left off.
- **Easy Updatability:** Unlike other nonvolatile memory technologies—ROM that can never be altered after it is manufactured or EPROM that can only be erased by removing it from the system and exposing it to ultraviolet light—flash can be erased and reprogrammed electrically while resident in the host system.
- **Software Compatibility:** With Flash File System software and ROM-executable versions of the disk operating system (DOS), complete software compatibility between a user's desktop and portable system is ensured.
- **Exchangeable Card Architecture (ExCA™):** Through Intel's ExCA card interface standard, Intel's flash memory cards meet all specifications of PCMCIA 2.0, ensuring interchangeability between a host of PCMCIA-compatible systems.
- **Family of Products:** Intel Flash Memory products are currently available in component densities up to 8 Mbits, and in 4-Mbyte, 10-Mbyte and 20-Mbyte memory cards. Additionally, Intel offers Boot Block devices in densities up to 4 Mbits.

Flash Memory Components

3

28F256A
256K (32K x 8) CMOS FLASH MEMORY

- **Flash Electrical Chip-Erase**
  - **— 1 Second Typical Chip-Erase**
- **Quick-Pulse Programming Algorithm**
  - **— 10 μs Typical Byte-Program**
  - **— 0.5 Second Chip-Program**
- **100,000 Erase/Program Cycles Typical**
- **12.0V ±5% V_{PP}**
- **High-Performance Read**
  - **— 120 ns Maximum Access Time**
- **CMOS Low Power Consumption**
  - **— 10 mA Typical Active Current**
  - **— 50 μA Typical Standby Current**
  - **— 0 Watts Data Retention Power**
- **Integrated Program/Erase Stop Timer**
- **Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface**
- **Noise Immunity Features**
  - **— ±10% V_{CC} Tolerance**
  - **— Maximum Latch-Up Immunity through EPI Processing**
- **ETOX™ II Flash Nonvolatile Technology**
  - **— EPROM-Compatible Process Base**
  - **— High-Volume Manufacturing Experience**
- **JEDEC-Standard Pinouts**
  - **— 32-Pin Cerdip**
  - **— 32-Lead PLCC**

(See Packaging Spec., Order #231369)

3

Intel's 28F256A CMOS flash memory offers the most cost-effective and reliable alternative for read/write random access nonvolatile memory. The 28F256A adds electrical chip-erasure and reprogramming to familiar EPROM technology. Memory contents can be rewritten: in a test socket; in a PROM-programmer socket; on-board during subassembly test; in-system during final test; and in-system after-sale. The 28F256A increases memory flexibility, while contributing to time and cost savings.

The 28F256A is a 256-kilobit nonvolatile memory organized as 32,768 bytes of 8 bits. Intel's 28F256A is offered in 32-pin plastic dip and 32-lead PLCC. Pin assignments conform to JEDEC standards.

Extended erase and program cycling capability is designed into Intel's ETOX™ II (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional EEPROMs. With the 12.0V V_{PP} supply, the 28F256A performs a minimum of 10,000 erase and program cycles well within the time limits of the Quick-Pulse Programming and Quick-Erase algorithms.

Intel's 28F256A employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 120 ns access time provides no-WAIT-state performance for a wide range of microprocessors and microcontrollers. Typical standby current of 50 μA translates into power savings when the device is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from $-1V$ to $V_{CC} + 1V$.

With Intel's ETOX II process base, the 28F256A levers years of EPROM experience to yield the highest levels of quality, reliability, and cost-effectiveness.

September 1992
Order Number: 290243-005

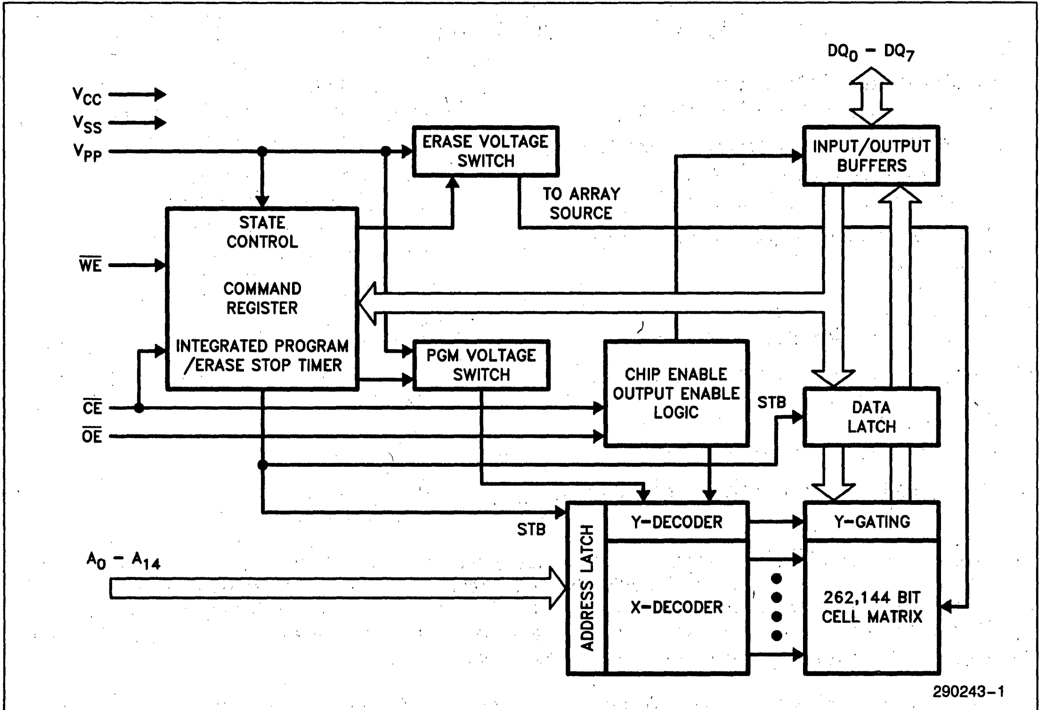


Figure 1. 28F256A Block Diagram

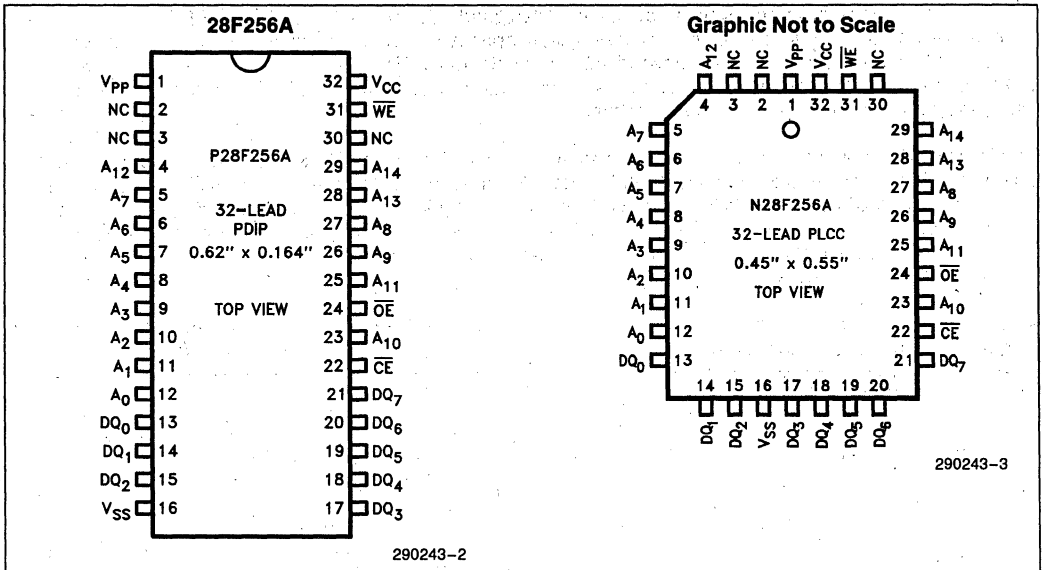


Figure 2. 28F256A Pin Configurations

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{14}	INPUT	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
DQ_0–DQ_7	INPUT/ OUTPUT	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}$	INPUT	**CHIP ENABLE** activates the device's control logic, input buffers, decoders, and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels.
$\overline{OE}$	INPUT	**OUTPUT ENABLE** gates the devices output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	INPUT	**WRITE ENABLE** controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **Note:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP}		**ERASE/PROGRAM POWER SUPPLY** for writing the command register, erasing the entire array, or programming bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY (5V ±10%).**
V_{SS}		**GROUND.**
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating.

APPLICATIONS

The 28F256A flash memory provides nonvolatility along with the capability to typically perform over 100,000 electrical chip-erasure/reprogram cycles. These features make the 28F256A an innovative alternative to disk, EEPROM, and battery-backed static RAM. Where periodic updates of code and data-tables are required, the 28F256A's reprogrammability and nonvolatility make it the obvious and ideal replacement for EPROM.

Primary applications and operating systems stored in flash eliminate the slow disk-DRAM download process. This results in a dramatic enhancement of performance and substantial reduction of power consumption—considerations particularly important in portable equipment. Flash memory increases flexibility with electrical chip-erasure and in-system update capability of operating systems and application code. With updatable BIOS, system manufacturers can easily accommodate last-minute changes as revisions are made.

In diskless workstations and terminals, network traffic reduces to a minimum and systems become instant-on. Reliability exceeds that of electromechanical media. Often in these environments, power interrupts force extended re-boot periods for all networked terminals. This mishap is no longer an issue if boot code, operating systems, communications protocols and primary applications are flash-resident in each terminal.

For embedded systems that rely on dynamic RAM/disk for main system memory or nonvolatile backup storage, the 28F256A provides higher performance, lower power consumption, instant-on capability, and allows an "execute in place" memory hierarchy for code and data table reading. Additionally, the flash memory is more rugged and reliable in harsh environments where extreme temperatures and shock can cause disk-based systems to fail.

The need for code updates pervades all phases of a system's life—from prototyping to system manufacturing to after-sale service. The electrical chip-erasure and reprogramming ability of the 28F256A allows in-circuit alterability; this eliminates unnecessary handling and less-reliable socketed connections, while adding greater test, manufacture, and update flexibility.

Material and labor costs associated with code changes increases at higher levels of system inte-

gration—the most costly being code updates after sale. Code "bugs", or the desire to augment system functionality, prompt after-sale code updates. Field revision to EPROM-based code requires the removal of EPROM components or entire boards. With the 28F256A, code updates are implemented locally via an edge-connector, or remotely over a communications link.

For systems currently using a high-density static RAM/battery configuration for data accumulation, flash memory's inherent nonvolatility eliminates the need for battery backup. The concern for battery failure no longer exists, an important consideration for portable equipment and medical instruments, both requiring continuous performance. In addition, flash memory offers a considerable cost advantage over static RAM.

Flash memory's electrical chip-erasure, byte programmability and complete nonvolatility fit well with data accumulation and recording needs. Electrical chip-erasure gives the designer a "blank slate" in which to log or record data. Data can be periodically off-loaded for analysis and the flash memory erased producing a new "blank slate".

A high degree of on-chip feature integration simplifies memory-to-processor interfacing. Figure 3 depicts two 28F256As tied to the 80C186 system bus. The 28F256A's architecture minimizes interface circuitry needed for complete in-circuit updates of memory contents.

With cost-effective in-system reprogramming, extended cycling capability, and true nonvolatility, the 28F256A is a functional superset of one or more of the alternatives: EPROMs, EEPROMs, battery backed static RAM, or disk. EPROM-compatible read specifications, straightforward interfacing, and in-circuit alterability offer designers unlimited flexibility to meet the high standards of today's designs.

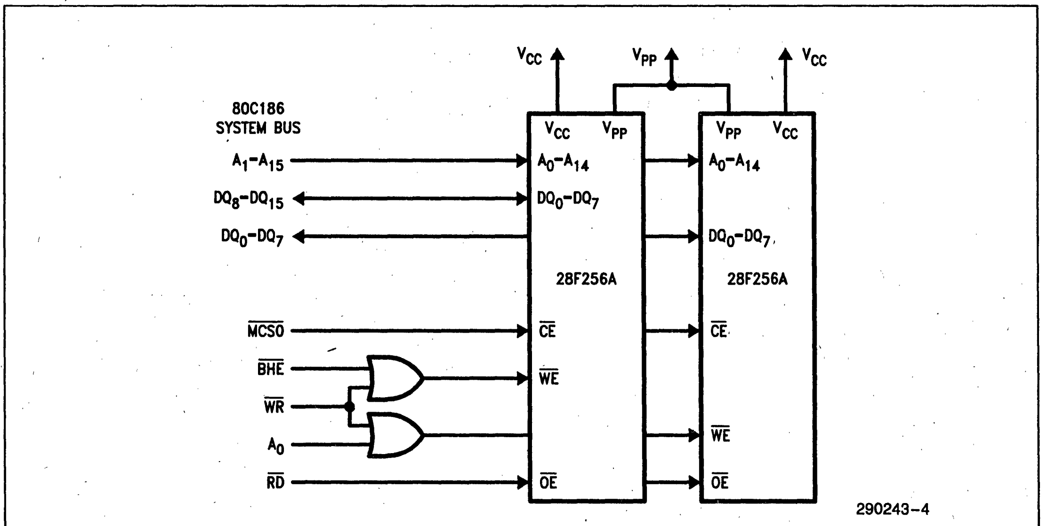


Figure 3. 28F256As in a 80C186 System

PRINCIPLES OF OPERATION

Flash memory augments EPROM functionality with in-circuit electrical erasure and reprogramming. The 28F256A introduces a command register to manage this new functionality. The command register allows for: 100% TTL-level control inputs; fixed power supply during erasure and programming; and maximum EPROM compatibility.

In the absence of high voltage on the V_{PP} pin, the 28F256A is a read-only memory. Manipulation of the external memory-control pins yields the standard EPROM read, standby, output disable, and int$_e$ligent Identifier™ operations.

The same EPROM read, standby, and output disable operations are available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} enables erasure and programming of the device. All functions associated with altering memory contents—int$_e$ligent Identifier, erase, erase verify, program, and program verify—are accessed via the command register.

Commands are written to the register using standard microprocessor write timings. Register contents serve as input to an internal state-machine which control the erase and programming circuitry. Write cycles also internally latch addresses and data needed for programming and erase operations. With the appropriate command written to the register, standard microprocessor read timings output array data, access the int$_e$ligent Identifier codes, or output data for erase and program verification.

Integrated Program/Erase Stop Timer

Successive command write cycles define the duration of program and erase operations; specifically, the program or erase time durations are normally terminated by associated program or erase verify commands. An integrated stop timer provides simplified timing control over these operations; thus eliminating the need for maximum program/erase timing specifications. Programming and erase pulse durations are minimums only. When the stop timer terminates a program or erase operation, the device enters an inactive state and remains inactive until receiving the appropriate verify or reset command.

Write Protection

The command register is only active when V_{PP} is at high voltage. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable—available only when memory updates are desired. When $V_{PP} = V_{PPL}$, the contents of the register default to the read command, making the 28F256A a read-only memory. In this mode, the memory contents cannot be altered.

Or, the system designer may choose to "hardwire" V_{PP}, making the high voltage supply constantly available. In this case, all Command Register functions are inhibited whenever V_{CC} is below the write lockout voltage V_{LKO}. (See Power Up/Down Protection). The 28F256A is designed to accommodate either design practice, and to encourage optimization of the processor-memory interface.

The two-step program/erase write sequence to the Command Register provides additional software write protection.

3

BUS OPERATIONS

Read

The 28F256A has two control functions, both of which must be logically active, to obtain data at the outputs. Chip-Enable ($\overline{CE}$) is the power control and should be used for device selection. Output-Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Refer to AC read timing waveforms.

When V_{PP} is high (V_{PPH}), the read operations can be used to access array data, to output the int$_e$ligent Identifier codes, and to access data for program/erase verification. When V_{PP} is low (V_{PPL}), the read operation can access only the array data.

Output Disable

With Output-Enable at a logic-high level (V_{IH}), output from the device is disabled. Output pins are placed in a high-impedance state.

Standby

With Chip-Enable at a logic-high level, the standby operation disables most of the 28F256A's circuitry and substantially reduces device power consumption. The outputs are placed in a high-impedance

state, independent of the Output-Enable signal. If the 28F256A is deselected during erasure, programming, or program/erase verification, the device draws active current until the operation is terminated.

int$_e$ligent Identifier Operation

The int$_e$ligent Identifier operation outputs the manufacturer code (89H) and device code (B9H). Programming equipment automatically matches the device with its proper erase and programming algorithms. With Chip-Enable and Output-Enable at a logic low level, rising A_9 to high voltage V_{ID} (see D.C. Characteristics) activates the operation. Data read from locations 0000H and 0001H represent the manufacturer's code and the device code, respectively.

The manufacturer- and device-codes can also be read via the command register, for instances where the 28F256A is erased and reprogrammed in the target system. Following a write of 90H to the command register, a read from address location 0000H outputs the manufacturer code (89H). A read from address 0001H outputs the device code (B9H).

Write

Device erasure and programming are accomplished via the command register, when high voltage is applied to the V_{PP} pin. The contents of the register serve as input to the internal state-machine. The state-machine outputs dictate the function of the device.

The command register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

The command register is written by bringing Write-Enable to a logic-low level (V_{IL}), while Chip-Enable is low. Addresses are latched on the falling edge of Write-Enable, while data is latched on the rising edge of the Write-Enable pulse. Standard microprocessor write timings are used.

Refer to AC Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the V_{PP} pin, the contents of the command register default to 00H, enabling read-only operations.

Placing high voltage on the V_{PP} pin enables read/write operations. Device operations are selected by writing specific data patterns into the command register. Table 3 defines these 28F256A register commands.

Table 2. 28F256A Bus Operations

	Pins / Operation	V_{PP}(1)	A_0	A_9	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_0–DQ_7
READ-ONLY	Read	V_{PPL}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out
	Output Disable	V_{PPL}	X(7)	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby	V_{PPL}	X	X	V_{IH}	X	X	Tri-State
	inteligent ID Manufacturer(2)	V_{PPL}	V_{IL}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = 89H
	inteligent ID Device(2)	V_{PPL}	V_{IH}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = B9H
READ/WRITE	Read	V_{PPH}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out(4)
	Output Disable	V_{PPH}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby(5)	V_{PPH}	X	X	V_{IH}	X	X	Tri-State
	Write	V_{PPH}	A_0	A_9	V_{IL}	V_{IH}	V_{IL}	Data In (6)

NOTES:
1. Refer to DC Characteristics. When $V_{PP} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes may also be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
3. V_{ID} is the int$_e$ligent Identifier high voltage. Refer to D.C. Characteristics.
4. Read operations with $V_{PP} = V_{PPH}$ may access array data or the int$_e$ligent Identifier codes.
5. With V_{PP} at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
6. Refer to Table 3 for valid Data-In during a write operation.
7. X can be V_{IL} or V_{IH}.

Table 3. Command Definitions

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	1	Write	X	00H			
Read inteligent ID Codes	3	Write	X	90H	Read	(4)	(4)
Set-Up Erase/Erase(6)	2	Write	X	20H	Write	X	20H
Erase Verify(6)	2	Write	EA	A0H	Read	X	EVD
Set-Up Program/Program(5)	2	Write	X	40H	Write	PA	PD
Program Verify(5)	2	Write	X	C0H	Read	X	PVD
Reset(7)	2	Write	X	FFH	Write	X	FFH

NOTES:
1. Bus operation are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   PA = Address of memory location to be programmed.
   Addresses are latched on the falling edge of the Write-Enable pulse.
3. ID = Data read from location IA during device identification. (Mfr = 89H, Device = B9H).
   EVD = Data read from location EA during erase verify.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of the Write-Enable.
   PVD = Data read from location PA during program verify. PA is latched on the Program command.
4. Following the Read inteligent ID command, two read operations access manufacturer and device codes.
5. Figure 4 illustrates the Quick-Pulse Programming Algorithm.
6. Figure 5 illustrates the Quick-Erase Algorithm.
7. The second bus cycle must be followed by the desired command register write.

Read Command

While V_{PP} is high, for erasure and programming, memory contents can be accessed via the read command. The read operation is initiated by writing 00H into the command register. Microprocessor read cycles retrieve array data. The device remains enabled for reads until the command register contents are altered.

The default contents of the register upon V_{PP} power-up is 00H. This default value ensures that no spurious alternation of memory contents occurs during the V_{PP} power transition. Where the V_{PP} supply is hard-wired to the 28F256A, the device powers-up and remains enabled for reads until the command register contents are changed. Refer to the AC Read Characteristics and Waveforms for specific timing parameters.

inteligent Identifier Command

Flash memories are intended for use in applications where the local CPU alters memory contents. As such, manufacturer- and device-codes must be accessible while the device resides in the target system. PROM programmers typically access signature codes by raising A_9 to a high voltage. However, multiplexing high voltage onto address lines is not a desired system-design practice.

The 28F256A contains an inteligent Identifier operation to supplement traditional PROM-programming methodology. The operation is initiated by writing 90H into the command register. Following the command write, a read cycle from address 0000H retrieves the manufacturer code 89H. A read cycle from address 0001H returns the device code B9H. To terminate the operation, it is necessary to write another valid command into the register.

Set-Up Erase/Erase Commands

Set-up Erase is a command-only operation that stages the device for electrical erase of all bytes in the array. The set-up erase operation is performed by writing 20H to the command register. To commence chip-erasure, the erase command (20H) must again be written to the register. The erase operation begins with the rising edge of the Write-Enable pulse and terminate with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, chip-erasure can only occur when high voltage is applied to the V_{PP} pin. In the absence

of this high voltage, memory contents are protected against erasure. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all of the bytes of the array in parallel. After each erase operation, all bytes must be verified. The erase verify operation is initiated by writing A0H into the command register. The address for the byte to be verified must be supplied as it is latched on the falling edge of the Write-Enable pulse. The register write terminates the erase operation with the rising edge of its Write-Enable pulse.

The 28F256A applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased.

The erase-verify command must be written to the command register prior to each byte verification to latch its address. The process continues for each byte in the array until a byte does not return FFH data, or the last address is accessed.

In the case where the data read is not FFH, another erase operation is performed. (Refer to Set-Up Erase/Erase.) Verification then resumes from the address of the last-verified byte. Once all bytes in the array have been verified, the erase step is complete. The device can be programmed. At this point, the verify operation is terminated by writing a valid command (e.g., Program Set-Up) to the command register. Figure 5, the Quick-Erase algorithm, illustrates how commands and bus operations are combined to perform electrical erasure of the 28F256A. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Set-Up Program/Program Commands

Set-up program is a command-only operation that stages the device for byte programming. Writing 40H into the command register performs the set-up operation.

Once the program set-up operation is performed, the next Write-Enable pulse causes a transition to an active programming operation. Addresses are internally latched on the falling edge of the Write-Enable pulse. Data is internally latched on the rising edge of the Write-Enable pulse. The rising edge of Write-Enable also begins the programming operation. The programming operation terminates with the next rising edge of Write-Enable, used to write the program-verify command. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Program Verify Command

The 28F256A is programmed on a byte-by-byte basis. Byte programming may occur sequentially or at random. Following each programming operation, the byte just programmed must be verified.

The program-verify operation is initiated by writing C0H into the command register. The register write terminates the programming operation with the rising edge of its Write-Enable pulse. The program-verify operation stages the device for verification of the byte last programmed. No new address information is latched.

The 28F256A applies an internally-generated margin voltage to the byte. A microprocessor read cycle outputs the data. A successful comparison between the programmed byte and true data means that the byte is successfully programmed. Programming then proceeds to the next desired byte location. Figure 4, the 28F256A Quick-Pulse Programming algorithm, illustrates how commands are combined with bus operations to perform byte programming. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or program-command sequences. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

EXTENDED ERASE/PROGRAM CYCLING

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled—an expensive solution.

Intel has designed extended cycling capability into its ETOX II flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an

advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field is one-tenth that of common EEPROMs, minimizing the probability of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 MV/cm lower then EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure—increasing time to wear out by a factor of 100,000,000.

The 28F256A is specified for a minimum of 10,000 program/erase cycles. The device is programmed and erased using Intel's Quick-Pulse Programming and Quick-Erase algorithms. Intel's algorithmic approach uses a series of operations (pulses), along with byte verification, to completely and reliably erase and program the device.

For further reliability information, see Reliability Report RR-60 (ETOX II Reliability Data Summary).

QUICK-PULSE PROGRAMMING ALGORITHM

The Quick-Pulse Programming algorithm uses programming operations of 10 μs duration. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The algorithm allows for up to 25 programming operations per byte, although most bytes verify on the first or second operation. The entire sequence of programming and byte verification is performed with V_{PP} at high voltage. Figure 4 illustrates the Quick-Pulse Programming algorithm.

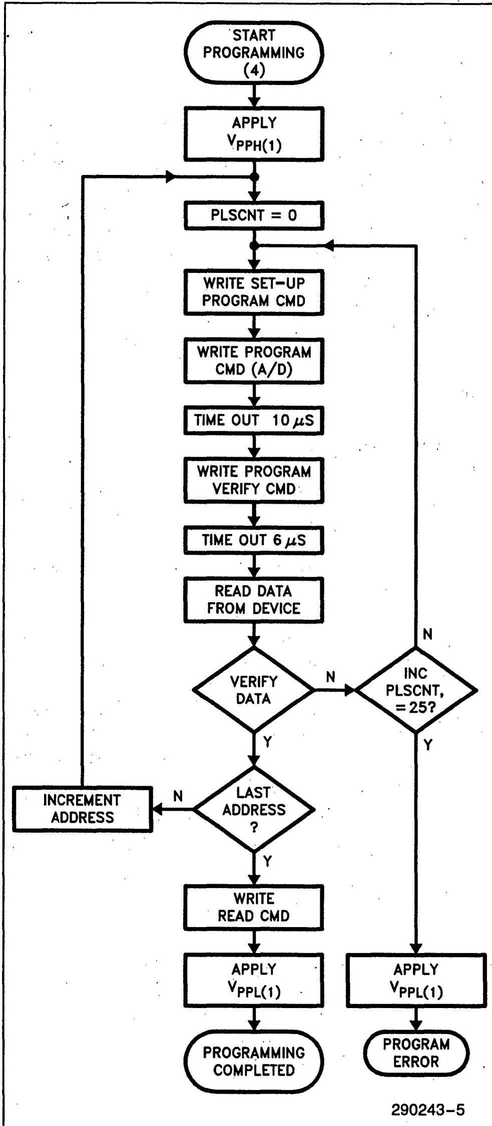
QUICK-ERASE ALGORITHM

Intel's Quick-Erase algorithm yields fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the Quick-Pulse Programming algorithm, to simultaneously remove charge from all bits in the array. Erasure begins with a read of memory contents. The 28F256A is erased when shipped from the factory. Reading FFH data from the device would immediately be followed by device programming.

For devices being erased and reprogrammed, uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (data = 00H). This is accomplished, using the Quick-Pulse Programming algorithm, in approximately one-half second.

Erase execution then continues with an initial erase operation. Erase verification (data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. With each erase operation, an increasing number of bytes verify to the erased state. Erase efficiency may be improved by storing the address of the last byte verified in a register. Following the next erase operation, verification starts at that stored address location. Erasure typically occurs in one second. Figure 5 illustrates the Quick-Erase algorithm.

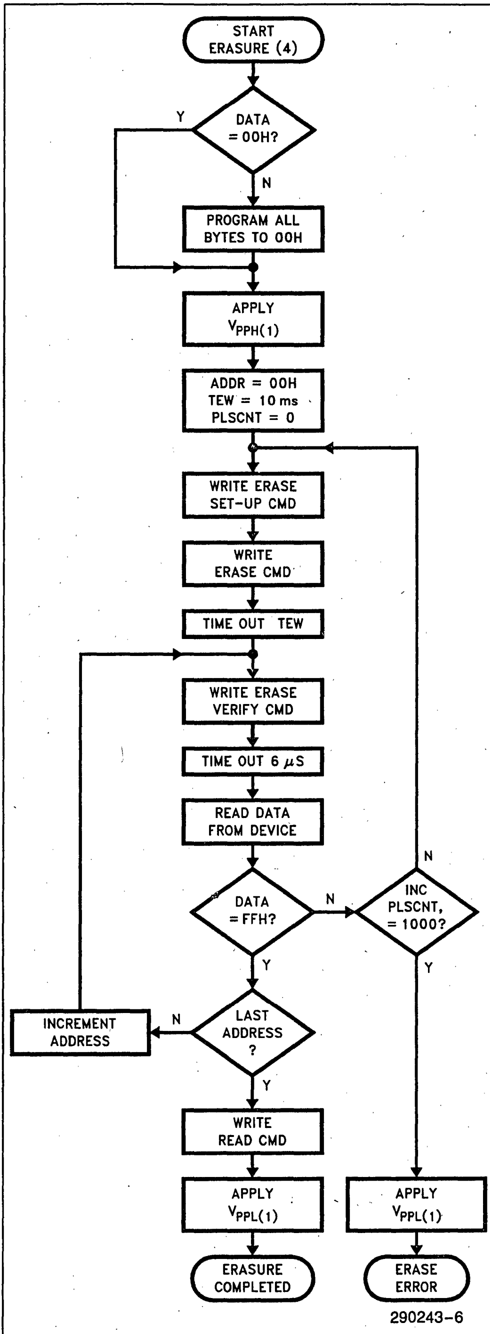
3



Bus Operation	Command	Comments
Standby		Wait for V_{PP} ramp to V_{PPH} (= 12.0V) (1) Initialize pulse-count
Write	Set-Up Program	Data = 40H
Write	Program	Valid address/data
Standby		Duration of Program operation (t_{WHWH1})
Write	Program(2) Verify	Data = C0H; Stops (3) Program Operation
Standby		t_{WHGL}
Read		Read byte to verify programming
Standby		Compare data output to data expected
Write	Read	Data = 00H, resets the register for read operations.
Standby		Wait for V_{PP} ramp to V_{PPL}(1)

NOTES:
1. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
2. Program Verify is only performed after byte programming. A final read/compare may be performed (optional) after the register is written with the Read command.
3. Refer to principles of operation.
4. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.

Figure 4. 28F256A Quick-Pulse Programming Algorithm



Bus Operation	Command	Comments
Standby		Wait for V_{PP} ramp to V_{PPH} (= 12.0V) (1) Use Quick-Pulse Programming (Fig. 4)
		Initialize Addresses, Erase Pulse Width, and Pulse Count
Write	Set-Up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase operation (t_{WHWH2})
Write	Erase Verify(2)	Addr = Byte to verify; Data = A0H; Stops Erase Operation (3)
Standby		t_{WHGL}
Read		Read byte to verify erasure
Standby		Compare output to FFH increment pulse count
Write	Read	Data = 00H, resets the register for read operations.
Standby		Wait for V_{PP} ramp to V_{PPL}(1)

NOTES:

1. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
2. Erase Verify is performed only after chip-erasure. A final read/compare may be performed (optional) after the register is written with the Read command.
3. Refer to principles of operation.
4. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.

Figure 5. 28F256A Quick-Erase Algorithm

DESIGN CONSIDERATIONS

Two-Line Output Control

Flash memories are often used in larger memory arrays. Intel provides two read-control inputs to accommodate multiple memory connections. Two-line control provides for:

a. the lowest possible memory power dissipation, and

b. complete assurance that output bus contention will not occur.

To efficiently use these two control units, an address-decoder output should drive chip-enable, while the system's read signal controls all flash memories and other parallel memories. This assures that only enabled memory devices have active outputs, while deselected devices maintain the low power standby condition.

Power Supply Decoupling

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current (I_{CC}) issues—standby, active, and transient current peaks produced by falling and rising edges of chip-enable. The capacitive and inductive loads on the device outputs determine the magnitudes of these peaks.

Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 μF ceramic capacitor connected between V_{CC} and V_{SS}, and between V_{PP} and V_{SS}.

Place the high-frequency, low-inherent-inductance capacitors as close as possible to the devices. Also, for every eight devices, a 4.7 μF electrolytic capacitor should be placed at the array's power supply connection, between V_{CC} and V_{SS}. The bulk capacitor will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

V_{PP} Trace on Printed Circuit Boards

Programming flash memories, while they reside in the target smith, requires that the printed circuit board designer pay attention to the V_{PP} pin power supply trace. Use similar trace widths and layout considerations given the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots.

Power Up/Down Protection

The 28F256A is designed to offer protection against accidental erasure or programming during power transitions. Upon power-up, the 28F256A is indifferent as to which power supply, V_{PP} or V_{CC}, powers up first. **Power supply sequencing is not required.** Internal circuitry in the 28F256A ensures that the command register is reset to the read mode upon power up.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will inhibit writes. The control register architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

28F256A Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases the usable battery life of your system because the 28F256A does not consume any power to retain code or data when the system is off. Table 4 illustrates the power dissipated when updating the 28F256A.

Table 4. 28F256A Typical Update Power Dissipation(4)

Operation	Power Dissipation (Watt-Seconds)	Notes
Array Program/Program Verify	0.043	1
Array Erase/Erase Verify	0.083	2
One Complete Cycle	0.169	3

NOTES:

1. Formula to calculate typical Program/Program Verify Power = [V_{PP} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{PP2} typical + t_{WHGL} × I_{PP4} typical)] + [V_{CC} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{CC2} typical + t_{WHGL} × I_{CC4} typical)].
2. Formula to calculate typical Erase/Erase Verify Power = [V_{PP} (I_{PP3} typical × t_{ERASE} typical + I_{PP5} typical × t_{WHGL} × # Bytes)] + [V_{CC}(I_{CC3} typical × t_{ERASE} typical + I_{CC5} typical × t_{WHGL} × # Bytes)].
3. One Complete Cycle = Array Preprogram + Array Erase + Program.
4. "Typicals" are not guaranteed, but based on a limited number of samples from production lots.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read .... 0°C to +70°C(1)
During Erase/Program .... 0°C to +70°C

Temperature Under Bias .... −10°C to +80°C

Storage Temperature .... −65°C to +125°C

Voltage on Any Pin with
Respect to Ground .... −2.0V to +7.0V(2)

Voltage on Pin A_9 with
Respect to Ground .... −2.0V to +13.5V(2, 3)

V_{PP} Supply Voltage with
Respect to Ground
During Erase/Program ..... −2.0V to +14.0(2, 3)

V_{CC} Supply Voltage with
Respect to Ground .... −2.0V to +7.0V(2)

Output Short Circuit Current............. 100 mA(4)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

***** *WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} +0.5V, which may overshoot to V_{CC} +2.0V for periods less than 20 ns.
3. Maximum DC voltage on A_9 or V_{PP} may overshoot to +14.0V for periods less than 20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature	0	70	°C	For Read-Only and Read/Write Operations
V_{CC}	V_{CC} Supply Voltage	4.50	5.50	V	

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10.0	μA	$V_{CC} = V_{CC}$ max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.3	1.0	mA	$V_{CC} = V_{CC}$max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$max $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erasure Current	1, 2		5.0	15	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10.0	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, ID Current, or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10.0		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	20	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ min
V_{OH1}	Output High Voltage		2.4			V	I_{OH} = −2.5 mA $V_{CC} = V_{CC}$min
V_{ID}	A_9 inteligent Identifier Voltage		11.50		13.00	V	
I_{ID}	A_9 inteligent Identifier Current	1, 2		90	200	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} During Read-Only Operations		0.00		6.5	V	Note: Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} During Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—CMOS COMPATIBLE

Symbol	Parameter	Notes	Limits Min	Limits Typical(4)	Limits Max	Unit	Test Conditions
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10.0	μA	$V_{CC} = V_{CC}$ max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		50	100	μA	$V_{CC} = V_{CC}$max $\overline{CE} = V_{CC} \pm 0.2V$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$max $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	15	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10.0	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, ID Current, or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10.0		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	20	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		$0.7V_{CC}$		$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ min
V_{OH1}	Output High Voltage		$0.85V_{CC}$			V	I_{OH} = −2.5 mA, $V_{CC} = V_{CC}$ min
V_{OH2}			$V_{CC} - 0.4$				I_{OH} = 100 μA, $V_{CC} = V_{CC}$ min
V_{ID}	A_9 inteligent Identifier Voltage		11.50		13.00	V	
I_{ID}	A_9 inteligent Identifier Current	1, 2		90	200	μA	$A_9 = V_{ID}$

3

DC CHARACTERISTICS—CMOS COMPATIBLE (Continued)

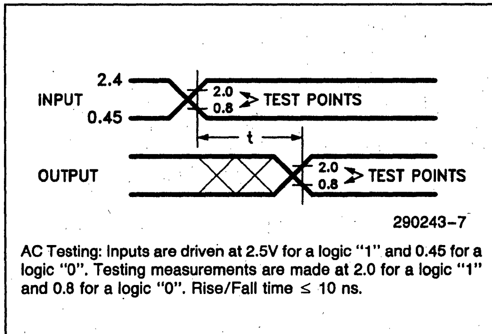
Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical(4)	Max		
V_{PPL}	V_{PP} During Read-Only Operations		0.00		6.5	V	Note: Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} During Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

CAPACITANCE(3) T = 25°C, f = 1.0 MHz

Symbol	Parameter	Notes	Limits		Unit	Conditions
			Min	Max		
C_{IN}	Address/Control Capacitance	3		6	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	3		12	pF	$V_{OUT} = 0V$

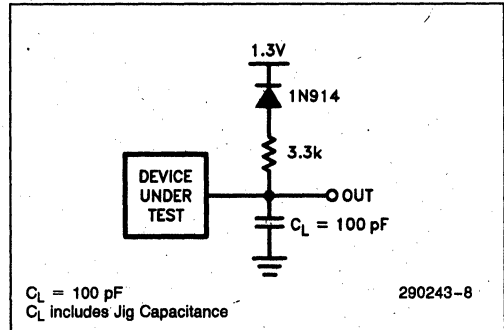
NOTES FOR DC CHARACTERISTICS AND CAPACITANCE:
1. All currents are in RMS unless otherwise noted. Typical values at $V_{CC} = 5.0V$, $V_{PP} = 12.0V$, T = 25°C. These currents are valid for all product versions (Packages and Speeds).
2. Not 100% tested: characterization data available.
3. Sampled, not 100% tested.
4. "Typicals" are not guaranteed, but based on a limited number of samples from production lots.

AC TESTING INPUT/OUTPUT WAVEFORM



AC Testing: Inputs are driven at 2.5V for a logic "1" and 0.45 for a logic "0". Testing measurements are made at 2.0 for a logic "1" and 0.8 for a logic "0". Rise/Fall time ≤ 10 ns.

AC TESTING LOAD CIRCUIT



C_L = 100 pF
C_L includes Jig Capacitance

AC Test Conditions

Input Rise and Fall Times (10% to 90%) ...... 10 ns
Input Pulse Levels .... 0.45 and 2.4
Input Timing Reference Level .... 0.8 and 2.0
Output Timing Reference Level......... 0.8 and 2.0

AC CHARACTERISTICS Read-Only Operations

Versions		Notes	28F256A-120		28F256A-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}/t_{RC}	Read Cycle Time		120		150		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time			120		150	ns
t_{AVQV}/t_{ACC}	Address Access Time			120		150	ns
t_{GLQV}/t_{OE}	Output Enable Access Time			50		55	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	2, 3	0		0		ns
t_{EHQZ}	Chip Disable to Output in High Z	2		55		55	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	2, 3	0		0		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	2		30		35	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	1, 2	0		0		ns
t_{WHGL}	Write Recovery Time before Read		6		6		µs

NOTES:
1. Whichever occurs first.
2. Sampled, not 100% tested.
3. Guaranteed by design.

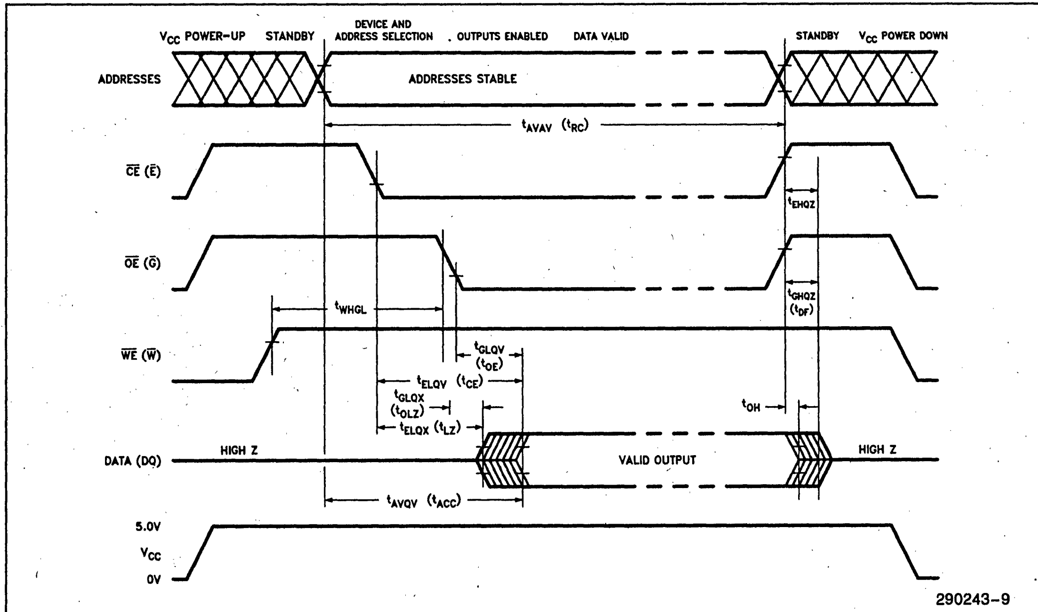


Figure 6. AC Waveform for Read Operations

AC CHARACTERISTICS—For Write/Erase/Program Operations(1)

Versions		Notes	28F256A-120		28F256A-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		120		150		ns
t_{AVWL}/t_{AS}	Address Set-Up Time		0		0		ns
t_{WLAX}/t_{AH}	Address Hold Time		60		60		ns
t_{DVWH}/t_{DS}	Data Set-Up Time		50		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		10		ns
t_{WHGL}	Write Recovery Time before Read		6		6		μs
t_{GHWL}	Read Recovery Time before Write	2	0		0		μs
t_{ELWL}/t_{CS}	Chip Enable Set-Up Time before Write		20		20		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time		0		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width		60		60		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		20		ns
t_{WHWH1}	Duration of Programming Operation	3	10		10		μs
t_{WHWH2}	Duration of Erase Operation	3	9.5		9.5		ms
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low	2	1.0		1.0		μs

NOTES:
1. Read timing parameters during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Guaranteed by design.
3. The integrated stop timer terminates the programming/erase operations, thereby eliminating the need for a maximum specification.

ERASE AND PROGRAMMING PERFORMANCE

Parameter	Notes	Limits						Unit
		28F256A-120			28F256A-150			
		Min	Typ	Max	Min	Typ	Max	
Chip Erase Time	1, 3, 4		1	10		1	10	sec
Chip Program Time	1, 2, 4		0.5	3		0.5	3	sec
Erase/Program Cycles	1, 5	10,000	100,000		10,000	100,000		cycles

NOTES:
1. "Typicals" are not guaranteed, but based on a limited number of samples taken from production lots. Data taken at 25°C, 12.0V V_{PP}, at 0 cycles.
2. Minimum byte programming time excluding system overhead is 16 μs program + 6 μs write recovery), while maximum is 400 μs/byte (16 μs x 25 loops allowed by algorithm). Max chip programming is specified lower than the worst case allowed by the programming algorithm since most bytes program significantly faster than the worst case byte.
3. Excludes 00H Programming Prior to Erasure.
4. Excludes System-Level Overhead.
5. Refer to RR-60 "ETOX" II Flash Memory Reliability Data Summary for typical cycling data and failure rate calculations.

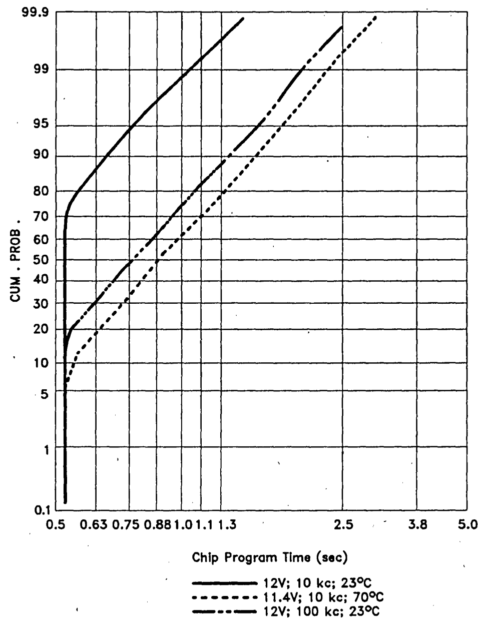


Figure 7. 28F256A Typical Programming Capability

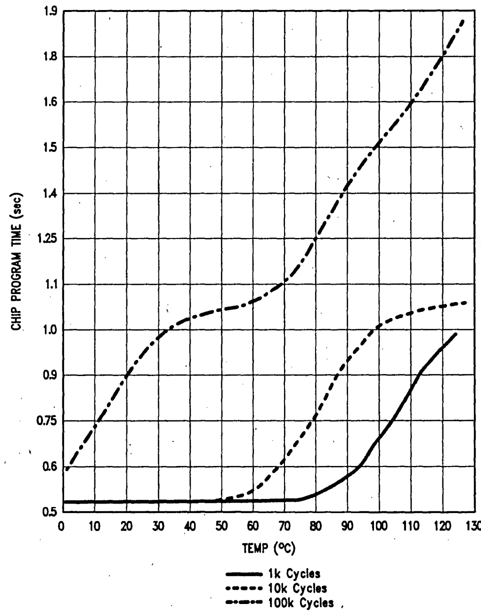


Figure 8. 28F256A Typical Program Time at 12V

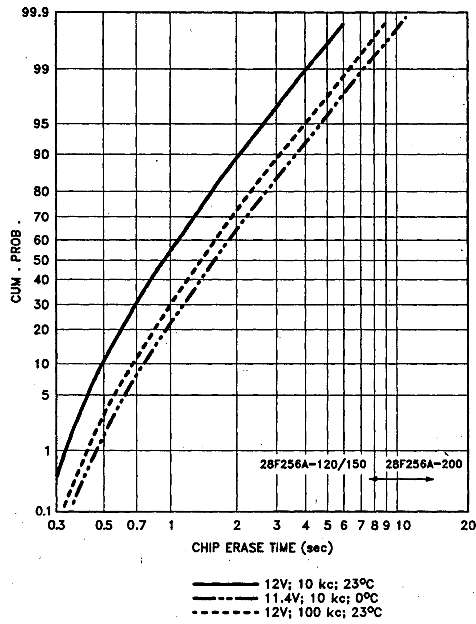


Figure 9. 28F256A Typical Erase Capability

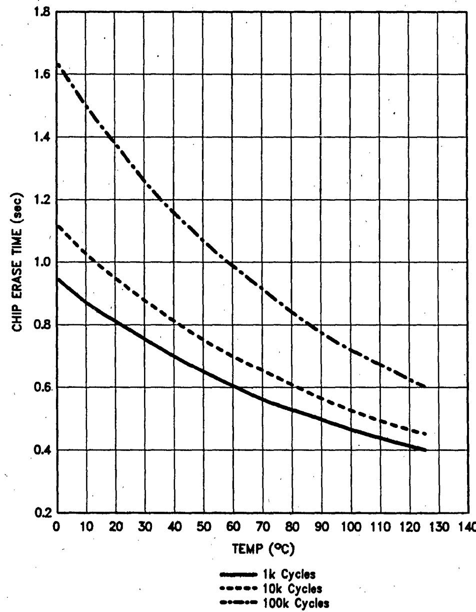


Figure 10. 28F256A Typical Erase Time at 12.0V

VCC POWER-UP & STANDBY | SET-UP PROGRAM COMMAND | PROGRAM COMMAND LATCH ADDRESS & DATA | PROGRAMMING | PROGRAM VERIFY COMMAND | PROGRAM VERIFICATION | STANDBY/ VCC POWER-DOWN

ADDRESSES

t_{AVAV} (t_{WC}) t_{AVAV} (t_{WC}) t_{AVWL} t_{WLAX} (t_{AH}) t_{AVAV} (t_{WC}) t_{AVWL} t_{WLAX} (t_{AH}) t_{AVAV} (t_{RC})

$\overline{CE}$ ($\overline{E}$)

t_{ELWL} t_{WHEH} (t_{CH}) t_{ELWL} (t_{CS}) t_{WHEH} (t_{CH}) t_{ELWL} (t_{CS}) t_{WHEH} (t_{CH}) t_{EHQZ}

$\overline{OE}$ ($\overline{G}$)

t_{GHWL} t_{WHWL} t_{WHWH1} t_{WHGL} t_{GHQZ} (t_{DF})

$\overline{WE}$ ($\overline{W}$)

t_{WLWH} t_{WLWH} t_{WLWH} t_{GLQV} (t_{OE})

t_{DVWH} t_{WHDX} t_{DVWH} t_{WHDX} t_{DVWH} t_{WHDX} (t_{DH}) t_{GLQX} (t_{OLZ}) t_{OH}

DATA (DQ) HIGH-Z DATA IN = 40H DATA IN DATA IN = C0H VALID DATA OUT

t_{ELQX} (t_{LZ}) t_{ELQV} (t_{CE})

5.0V V_{CC} 0V

t_{VPEL}

12V V_{PP} V_{PPL}

290243-14

Figure 11. AC Waveforms for Programming Operations

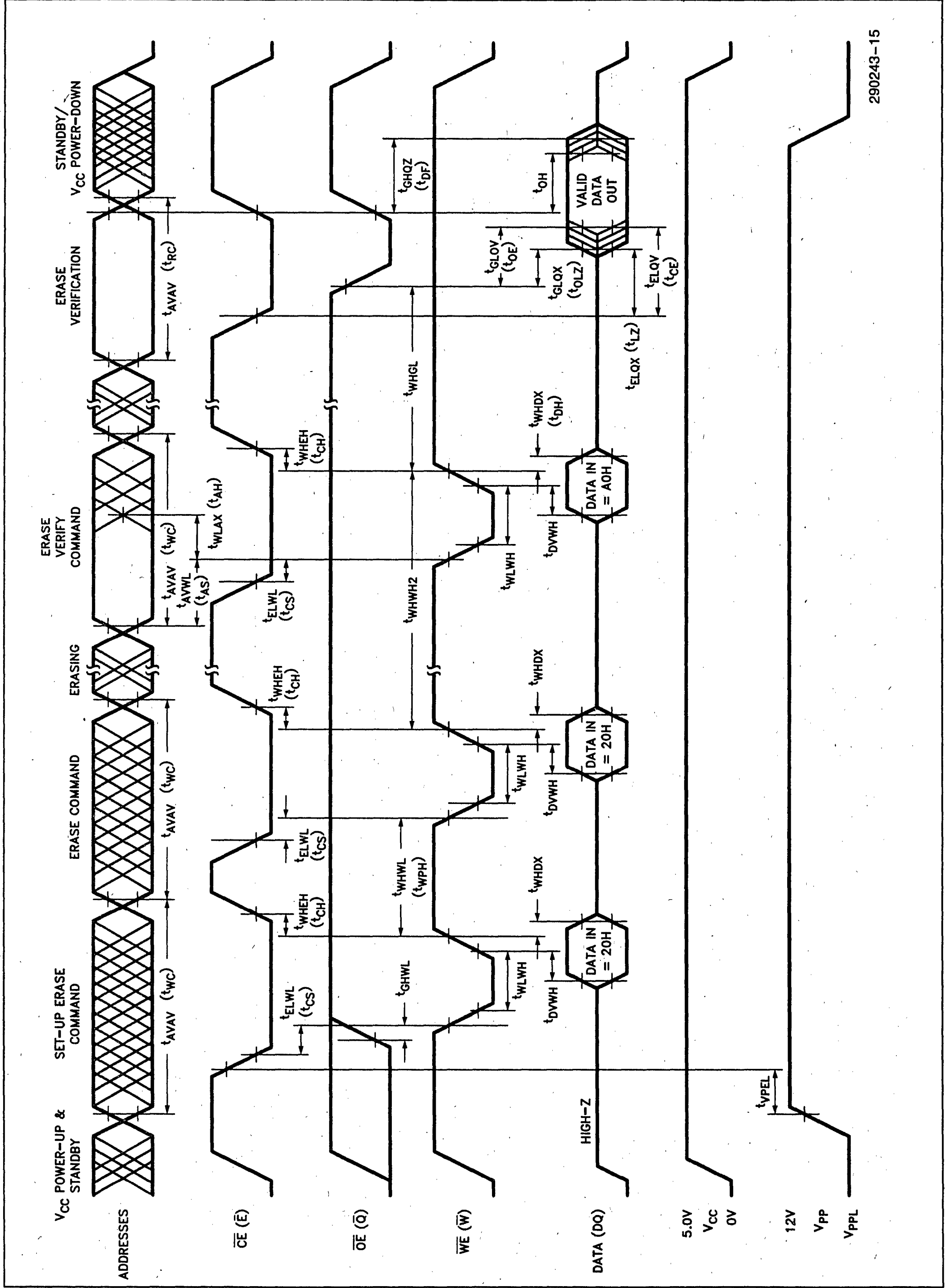


Figure 12. AC Waveforms for Erase Operations

ALTERNATIVE $\overline{CE}$-CONTROLLED WRITES

Versions		Notes	28F256A-120		28F256A-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}	Write Cycle Time		120		150		ns
t_{AVEL}	Address Set-Up Time		0		0		ns
t_{ELAX}	Address Hold Time		80		80		ns
t_{DVEH}	Data Set-Up Time		50		50		ns
t_{EHDX}	Data Hold Time		10		10		ns
t_{EHGL}	Write Recovery Time before Read		6		6		μs
t_{GHEL}	Read Recover Time before Write	2	0		0		μs
t_{WLEL}	Write Enable Set-Up Time before Chip-Enable		0		0		ns
t_{EHWH}	Write Enable Hold Time		0		0		ns
t_{ELEH}	Write Pulse Width	1	70		70		ns
t_{EHEL}	Write Pulse Width High		20		20		ns
t_{VPEL}	V_{PP} Set-Up Time to Chip-Enable Low	2	1.0		1.0		μs

NOTE:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of Chip-Enable and Write-Enable. In systems where Chip-Enable defines the write pulse width (with a longer Write-Enable timing waveform) all set-up, hold and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.
2. Guaranteed by design.

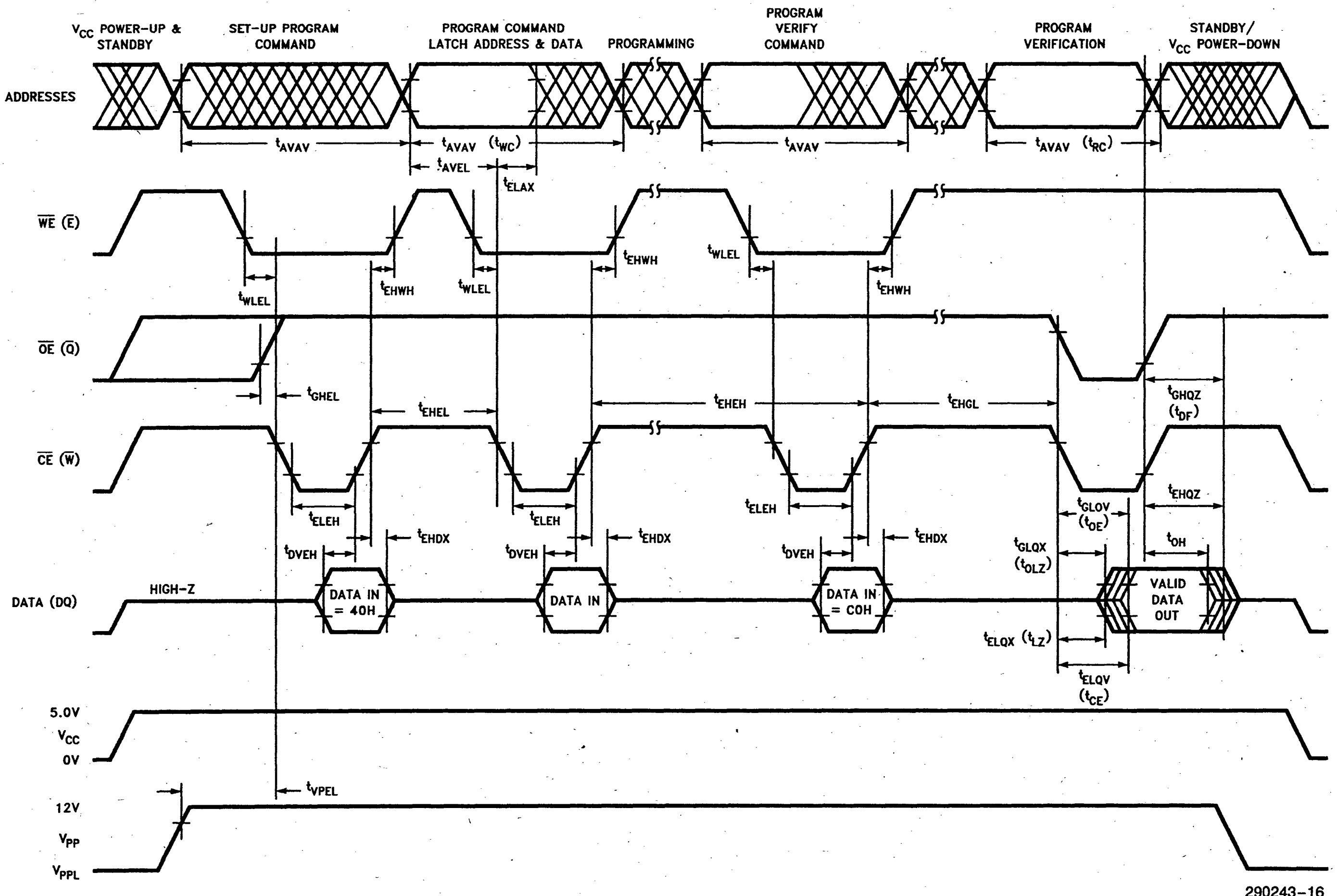
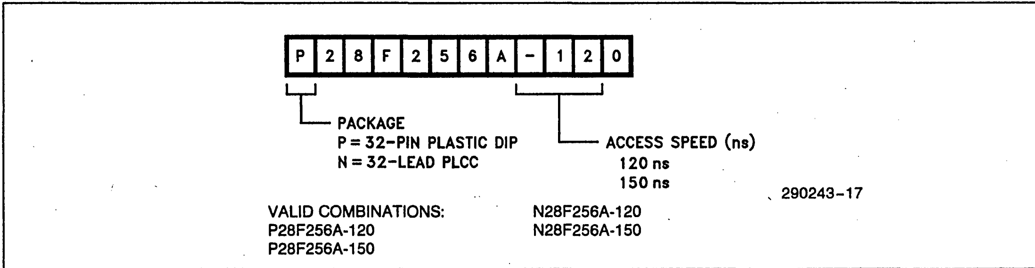


Figure 13. Alternate AC Waveforms for Programming Operations

ORDERING INFORMATION



ADDITIONAL INFORMATION

		Order Number
ER-20,	"ETOX II Flash Memory Technology"	294005
ER-24,	"Intel Flash Memory"	294008
RR-60,	"ETOX II Flash Memory Reliability Data Summary"	293002
AP-316,	"Using Flash Memory for In-System Reprogramming Nonvolatile Storage"	292046
AP-325,	"Guide to Flash Memory Reprogramming"	292059

REVISION HISTORY

Number	Description
004	Removed **Preliminary** Classification. Removed **200 ns** speed bin. Revised Erase Maximum Pulse Count for Figure 5 from **3000** to **1000**. Clarified AC and DC test conditions.
005	Corrected AC waveforms.

28F512
512K (64K x 8) CMOS FLASH MEMORY

- **Flash Electrical Chip-Erase**
  - **— 1 Second Typical Chip-Erase**
- **Quick-Pulse Programming Algorithm**
  - **— 10 μs Typical Byte-Program**
  - **— 1 Second Chip-Program**
- **100,000 Erase/Program Cycle Typical**
- **12.0V ±5% V_{PP}**
- **High-Performance Read**
  - **— 120 ns Maximum Access Time**
- **CMOS Low Power Consumption**
  - **— 10 mA Typical Active Current**
  - **— 50 μA Typical Standby Current**
  - **— 0W Data Retention Power**
- **Integrated Program/Erase Stop Timers**
- **Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface**
- **Noise Immunity Features**
  - **— ±10% V_{CC} Tolerance**
  - **— Maximum Latch-Up Immunity through EPI Processing**
- **ETOX™ II Nonvolatile Flash Technology**
  - **— EPROM-Compatible Process Base**
  - **— High-Volume Manufacturing Experience**
- **JEDEC-Standard Pinouts**
  - **— 32-Pin Plastic Dip**
  - **— 32-Lead PLCC**

  (See Packaging Spec., Order #231369)
- **Extended Temperature Options**

Intel's 28F512 CMOS flash memory offers the most cost-effective and reliable alternative for read/write random access nonvolatile memory. The 28F512 adds electrical chip-erasure and reprogramming to familiar EPROM technology. Memory contents can be rewritten: in a test socket; in a PROM-programmer socket; on-board during subassembly test; in-system during final test; and in-system after-sale. The 28F512 increases memory flexibility, while contributing to time- and cost-savings.

The 28F512 is a 512-kilobit nonvolatile memory organized as 65,536 bytes of 8 bits. Intel's 28F512 is offered in 32-pin plastic dip or 32-lead PLCC packages. Pin assignments conform to JEDEC standards for byte-wide EPROMs.

Extended erase and program cycling capability is designed into Intel's ETOX II (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional EEPROMs. With the 12.0V V_{PP} supply, the 28F512 performs a minimum of 10,000 erase and program cycles well within the time limits of the Quick-Pulse Programming™ and Quick-Erase™ algorithms.

Intel's 28F512 employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 120 nanosecond access time provides no-WAIT-state performance for a wide range of microprocessors and microcontrollers. Maximum standby current of 100 μA translates into power savings when the device is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from $-1V$ to $V_{CC} + 1V$.

With Intel's ETOX II process base, the 28F512 levers years of EPROM experience to yield the highest levels of quality, reliability, and cost-effectiveness.

November 1992
Order Number: 290204-007

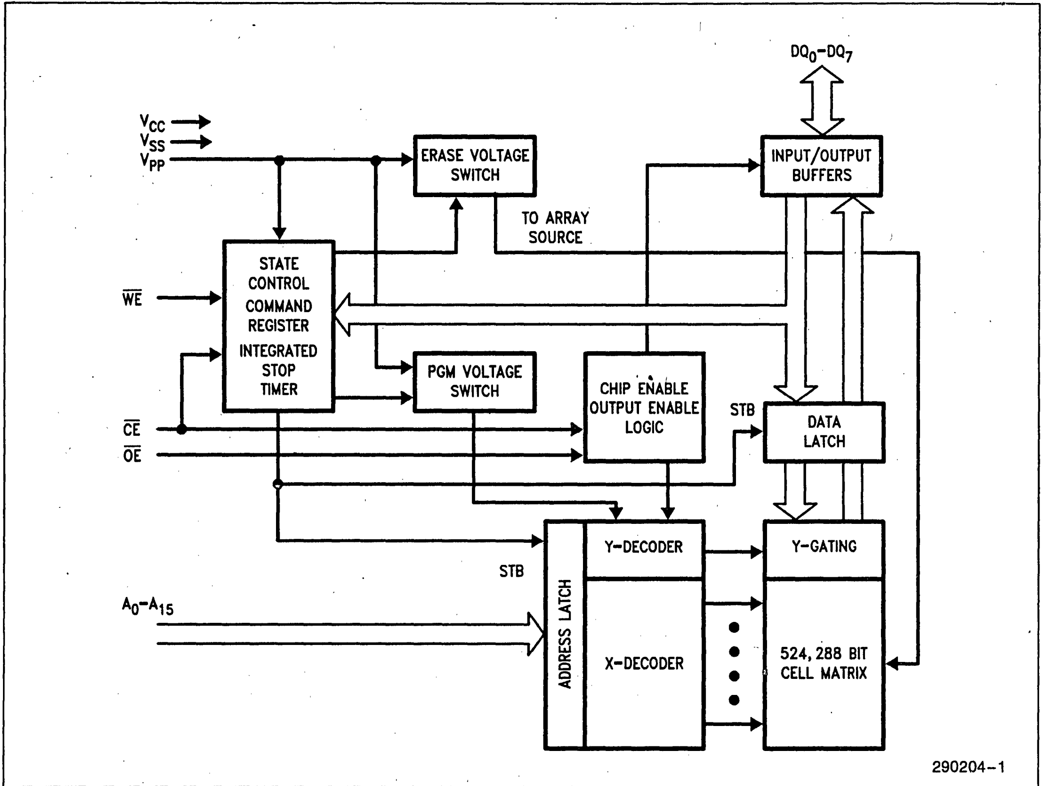


Figure 1. 28F512 Block Diagram

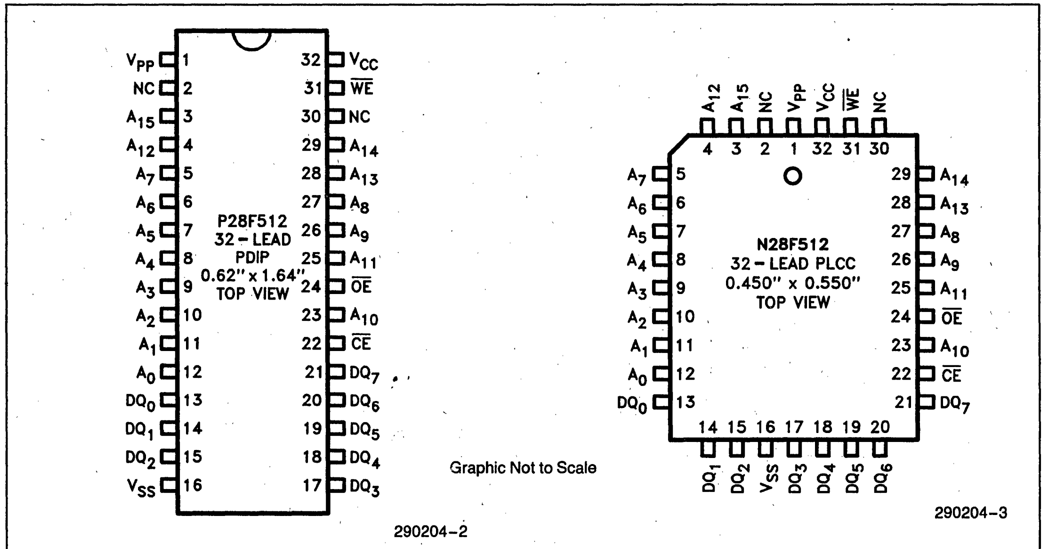


Figure 2. 28F512 Pin Configurations

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{15}	INPUT	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
DQ_0–DQ_7	INPUT/OUTPUT	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}$	INPUT	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels.
$\overline{OE}$	INPUT	**OUTPUT ENABLE:** Gates the devices output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	INPUT	**WRITE ENABLE:** Controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **Note:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP}		**ERASE/PROGRAM POWER SUPPLY** for writing the command register, erasing the entire array, or programming bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY** (5V ±10%)
V_{SS}		**GROUND**
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating.

APPLICATIONS

The 28F512 flash memory provides nonvolatility along with the capability to typically perform over 100,000 electrical chip-erasure/reprogram cycles. These features make the 28F512 an innovative alternative to disk, EEPROM, and battery-backed static RAM. Where periodic updates of code and data-tables are required, the 28F512's reprogrammability and nonvolatility make it the obvious and ideal replacement for EPROM.

Primary applications and operating systems stored in flash eliminate the slow disk-to-DRAM download process. This results in dramatic enhancement of performance and substantial reduction of power consumption — a consideration particularly important in portable equipment. Flash memory increases flexibility with electrical chip erasure and in-system update capability of operating systems and application code. With updatable BIOS, system manufacturers can easily accommodate last-minute changes as revisions are made.

In diskless workstations and terminals, network traffic reduces to a minimum and systems are instant-on. Reliability exceeds that of electromechanical media. Often in these environments, power interruptions force extended re-boot periods for all networked terminals. This mishap is no longer an issue if boot code, operating systems, communication protocols and primary applications are flash-resident in each terminal.

For embedded systems that rely on dynamic RAM/disk for main system memory or nonvolatile backup storage, the 28F512 flash memory offers a solid state alternative in a minimal form factor. The 28F512 provides higher performance, lower power consumption, instant-on capability, and allows an "execute in place" memory hierarchy for code and data table reading. Additionally, the flash memory is more rugged and reliable in harsh environments where extreme temperatures and shock can cause disk-based systems to fail.

The need for code updates pervades all phases of a system's life — from prototyping to system manufacture to after-sale service. The electrical chip-erasure and reprogramming ability of the 28F512 allows in-circuit alterability; this eliminates unnecessary handling and less-reliable socketed connections, while adding greater test, manufacture, and update flexibility.

Material and labor costs associated with code changes increases at higher levels of system integration — the most costly being code updates after sale. Code "bugs", or the desire to augment system functionality, prompt after-sale code updates. Field revisions to EPROM-based code requires the removal of EPROM components or entire boards. With the 28F512, code updates are implemented locally via an edge-connector, or remotely over a communcation link.

For systems currently using a high-density static RAM/battery configuration for data accumulation, flash memory's inherent nonvolatility eliminates the need for battery backup. The concern for battery failure no longer exists, an important consideration for portable equipment and medical instruments, both requiring continuous performance. In addition, flash memory offers a considerable cost advantage over static RAM.

Flash memory's electrical chip erasure, byte programmability and complete nonvolatility fit well with data accumulation and recording needs. Electrical chip-erasure gives the designer a "blank slate" in which to log or record data. Data can be periodically off-loaded for analysis and the flash memory erased producing a new "blank slate".

A high degree of on-chip feature integration simplifies memory-to-processor interfacing. Figure 3 depicts two 28F512s tied to the 80C186 system bus. The 28F512's architecture minimizes interface circuitry needed for complete in-circuit updates of memory contents.

With cost-effective in-system reprogramming, extended cycling capability, and true nonvolatility, the 28F512 offers advantages to the alternatives: EPROMs, EEPROMs, battery backed static RAM, or disk. EPROM-compatible read specifications, straight-forward interfacing, and in-circuit alterability offers designers unlimited flexibility to meet the high standards of today's designs.

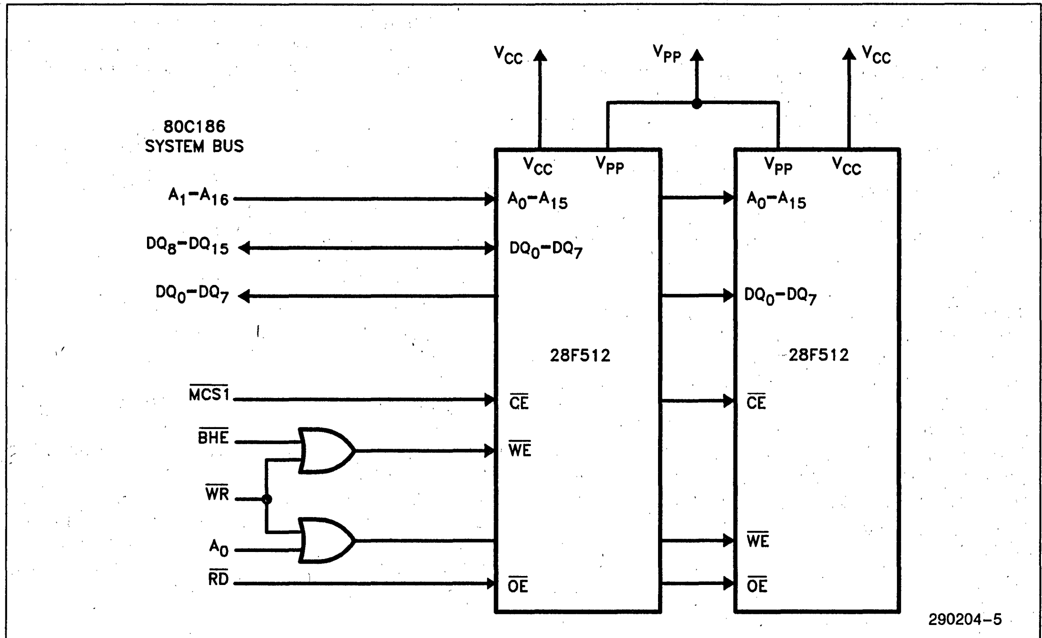


Figure 3. 28F512 in a 80C186 System

PRINCIPLES OF OPERATION

Flash-memory augments EPROM functionality with in-circuit electrical erasure and reprogramming. The 28F512 introduces a command register to manage this new functionality. The command register allows for: 100% TTL-level control inputs; fixed power supplies during erasure and programming; and maximum EPROM compatibility.

In the absence of high voltage on the V_{PP} pin, the 28F512 is a read-only memory. Manipulation of the external memory-control pins yields the standard EPROM read, standby, output disable, and Intelligent Identifier operations.

The same EPROM read, standby, and output disable operations are available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} enables erasure and programming of the device. All functions associated with altering memory contents—Intelligent Identifier, erase, erase verify, program, and program verify—are accessed via the command register.

Commands are written to the register using standard microprocessor write timings. Register contents serve as input to an internal state-machine which controls the erase and programming circuitry. Write cycles also internally latch addresses and data needed for programming or erase operations. With the appropriate command written to the register, standard microprocessor read timings output array data, access the Intelligent Identifier codes, or output data for erase and program verification.

Integrated Stop Timer

Successive command write cycles define the duration of program and erase operations; specifically, the program or erase time durations are normally terminated by associated program or erase verify commands. An integrated stop timer provides simplified timing control over these operations; thus eliminating the need for maximum program/erase timing specifications. Programming and erase pulse durations are minimums only. When the stop timer terminates a program or erase operation, the device enters an inactive state and remains inactive until receiving the appropriate verify or reset command.

Write Protection

The command register is only active when V_{PP} is at high voltage. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable—available only when memory updates are desired. When $V_{PP} = V_{PPL}$, the contents of the register default to the read command, making the 28F512 a read-only memory. In this mode, the memory contents cannot be altered.

Table 2. 28F512 Bus Operations

Operation		V_{PP}(1)	A_0	A_9	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_0–DQ_7
READ-ONLY	Read	V_{PPL}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out
	Output Disable	V_{PPL}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby	V_{PPL}	X	X	V_{IH}	X	X	Tri-State
	Intelligent Identifier (Mfr)(2)	V_{PPL}	V_{IL}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = 89H
	Intelligent Identifier (Device)(2)	V_{PPL}	V_{IH}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = B8H
READ/WRITE	Read	V_{PPH}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out(4)
	Output Disable	V_{PPH}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby(5)	V_{PPH}	X	X	V_{IH}	X	X	Tri-State
	Write	V_{PPH}	A_0	A_9	V_{IL}	V_{IH}	V_{IL}	Data In(6)

NOTES:
1. Refer to DC Characteristics. When $V_{PP} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes may also be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
3. V_{ID} is the Intelligent Identifier high voltage. Refer to DC Characteristics.
4. Read operations with $V_{PP} = V_{PPH}$ may access array data or the Intelligent Identifier codes.
5. With V_{PP} at high voltage, the standby current equals I_{CC} + I_{PP} (standby).
6. Refer to Table 3 for valid Data-In during a write operation.
7. X can be V_{IL} or V_{IH}.

Or, the system designer may choose to "hardwire" V_{PP}, making the high voltage supply constantly available. In this case, all Command Register functions are inhibited whenever V_{CC} is below the write lockout voltage V_{LKO}. (See Power Up/Down Protection). The 28F512 is designed to accommodate either design practice, and to encourage optimization of the processor-memory interface.

The two-step program/erase write sequence to the Command Register provides additional software write protection.

BUS OPERATIONS

Read

The 28F512 has two control functions, both of which must be logically active, to obtain data at the outputs. Chip-Enable ($\overline{CE}$) is the power control and should be used for device selection. Output-Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Refer to AC read timing waveforms.

When V_{PP} is high (V_{PPH}), the read operation can be used to access array data, to output the Intelligent Identifier codes, and to access data for program/erase verification. When V_{PP} is low (V_{PPL}), the read operation can **only** access the array data.

Output Disable

With Output-Enable at a logic-high level (V_{IH}), output from the device is disabled. Output pins are placed in a high-impedance state.

Standby

With Chip-Enable at a logic-high level, the standby operation disables most of the 28F512's circuitry and substantially reduces device power consumption. The outputs are placed in a high-impedance state, independent of the Output-Enable signal. If the 28F512 is deselected during erasure, programming, or program/erase verification, the device draws active current until the operation is terminated.

Intelligent Identifier Operation

The Intelligent Identifier operation outputs the manufacturer code (89H) and device code (B8H). Programming equipment automatically matches the device with its proper erase and programming algorithms.

With Chip-Enable and Output-Enable at a logic low level, raising A9 to high voltage V_{ID} (see DC Characteristics) activates the operation. Data read from locations 0000H and 0001H represent the manufacturer's code and the device code, respectively.

The manufacturer- and device-codes can also be read via the command register, for instances where the 28F512 is erased and reprogrammed in the target system. Following a write of 90H to the command register, a read from address location 0000H outputs the manufacturer code (89H). A read from address 0001H outputs the device code (B8H).

Write

Device erasure and programming are accomplished via the command register, when high voltage is applied to the V_{PP} pin. The contents of the register serve as input to the internal state-machine. The state-machine outputs dictate the function of the device.

The command register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

The command register is written by bringing Write-Enable to a logic-low level (V_{IL}), while Chip-Enable is low. Addresses are latched on the falling edge of Write-Enable, while data is latched on the rising edge of the Write-Enable pulse. Standard microprocessor write timings are used.

Refer to AC Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the V_{PP} pin, the contents of the command register default to 00H, enabling read-only operations.

Placing high voltage on the V_{PP} pin enables read/write operations. Device operations are selected by writing specific data patterns into the command register. Table 3 defines these 28F512 register commands.

Table 3. Command Definitions

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	1	Write	X	00H			
Read Intelligent Identifier Code(4)	3	Write	X	90H	Read	(4)	(4)
Set-up Erase/Erase(5)	2	Write	X	20H	Write	X	20H
Erase Verify(5)	2	Write	EA	A0H	Read	X	EVD
Set-up Program/Program(6)	2	Write	X	40H	Write	PA	PD
Program Verify(6)	2	Write	X	C0H	Read	X	PVD
Reset(7)	2	Write	X	FFH	Write	X	FFH

NOTES:
1. Bus operations are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   PA = Address of memory location to be programmed.
   Addresses are latched on the falling edge of the Write-Enable pulse.
3. ID = Data read from location IA during device identification (Mfr = 89H, Device = B8H).
   EVD = Data read from location EA during erase verify.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of Write-Enable.
   PVD = Data read from location PA during program verify. PA is latched on the Program command.
4. Following the Read Intelligent ID command, two read operations access manufacturer and device codes.
5. Figure 5 illustrates the Quick-Erase™ algorithm.
6. Figure 4 illustrates the Quick-Pulse Programming™ algorithm.
7. The second bus cycle must be followed by the desired command register write.

Read Command

While V_{PP} is high, for erasure and programming, memory contents can be accessed via the read command. The read operation is initiated by writing 00H into the command register. Microprocessor read cycles retrieve array data. The device remains enabled for reads until the command register contents are altered.

The default contents of the register upon V_{PP} power-up is 00H. This default value ensures that no spurious alteration of memory contents occurs during the V_{PP} power transition. Where the V_{PP} supply is hard-wired to the 28F512, the device powers-up and remains enabled for reads until the command-register contents are changed. Refer to the AC Read Characteristics and Waveforms for specific timing parameters.

Intelligent Identifier Command

Flash-memories are intended for use in applications where the local CPU alters memory contents. As such, manufacturer- and device-codes must be accessible while the device resides in the target system. PROM programmers typically access signature codes by raising A9 to a high voltage. However, multiplexing high voltage onto address lines is not a desired system-design practice.

The 28F512 contains an Intelligent Identifier operation to supplement traditional PROM-programming methodology. The operation is initiated by writing 90H into the command register. Following the command write, a read cycle from address 0000H retrieves the manufacturer code of 89H. A read cycle from address 0001H returns the device code of B8H. To terminate the operation, it is necessary to write another valid command into the register.

Set-up Erase/Erase Commands

Set-up Erase is a command-only operation that stages the device for electrical erasure of all bytes in the array. The set-up erase operation is performed by writing 20H to the command register.

To commence chip-erasure, the erase command (20H) must again be written to the register. The erase operation begins with the rising edge of the Write-Enable pulse and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, chip-erasure can only occur when high voltage is applied to the V_{PP} pin. In the absence of this high voltage, memory contents are protected against erasure. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all bytes of the array in parallel. After each erase operation, all bytes must be verified. The erase verify operation is initiated by writing A0H into the command register. The address for the byte to be verified must be supplied as it is latched on the falling edge of the Write-Enable pulse. The register write terminates the erase operation with the rising edge of its Write-Enable pulse.

The 28F512 applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased.

The erase-verify command must be written to the command register prior to each byte verification to latch its address. The process continues for each byte in the array until a byte does not return FFH data, or the last address is accessed.

In the case where the data read is not FFH, another erase operation is performed. (Refer to Set-up Erase/Erase). Verification then resumes from the address of the last-verified byte. Once all bytes in the array have been verified, the erase step is complete. The device can be programmed. At this point, the verify operation is terminated by writing a valid command (e.g. Program Set-up) to the command register. Figure 5, the Quick-Erase algorithm, illustrates how commands and bus operations are combined to perform electrical erasure of the 28F512. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Set-up Program/Program Commands

Set-up program is a command-only operation that stages the device for byte programming. Writing 40H into the command register performs the set-up operation.

Once the program set-up operation is performed, the next Write-Enable pulse causes a transition to an active programming operation. Addresses are internally latched on the falling edge of the Write-Enable pulse. Data is internally latched on the rising edge of the Write-Enable pulse. The rising edge of Write-Enable also begins the programming operation. The programming operation terminates with the next rising edge of Write-Enable, used to write the program-verify command. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Program-Verify Command

The 28F512 is programmed on a byte-by-byte basis. Byte programming may occur sequentially or at random. Following each programming operation, the byte just programmed must be verified.

The program-verify operation is initiated by writing C0H into the command register. The register write terminates the programming operation with the rising edge of its Write-Enable pulse. The program-verify operation stages the device for verification of the byte last programmed. No new address information is latched.

The 28F512 applies an internally-generated margin voltage to the byte. A microprocessor read cycle outputs the data. A successful comparison between the programmed byte and true data means that the byte is successfully programmed. Programming then proceeds to the next desired byte location. Figure 4, the 28F512 Quick-Pulse Programming algorithm, illustrates how commands are combined with bus operations to perform byte programming. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or program-command sequences. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

EXTENDED ERASE/PROGRAM CYCLING

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled—an expensive solution.

Intel has designed extended cycling capability into its ETOX II flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field is one-tenth that of common EEPROMs, minimizing the probability of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 MV/cm lower than EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure—increasing time to wearout by a factor of 100,000,000.

The 28F512 is specified for a minimum of 10,000 program/erase cycles. The device is programmed and erased using Intel's Quick-Pulse Programming and Quick-Erase algorithms. Intel's algorithmic approach uses a series of operations (pulses), along with byte verification, to completely and reliably erase and program the device.

For further information, see Reliability Report RR-60 (ETOX-II Reliability Data Summary).

QUICK-PULSE PROGRAMMING ALGORITHM

The Quick-Pulse Programming algorithm uses programming operations of 10 μs duration. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The algorithm allows for up to 25 programming operations per byte, although most bytes verify on the first or second operation. The entire sequence of programming and byte verification is performed with V_{PP} at high voltage. Figure 4 illustrates the Quick-Pulse Programming algorithm.

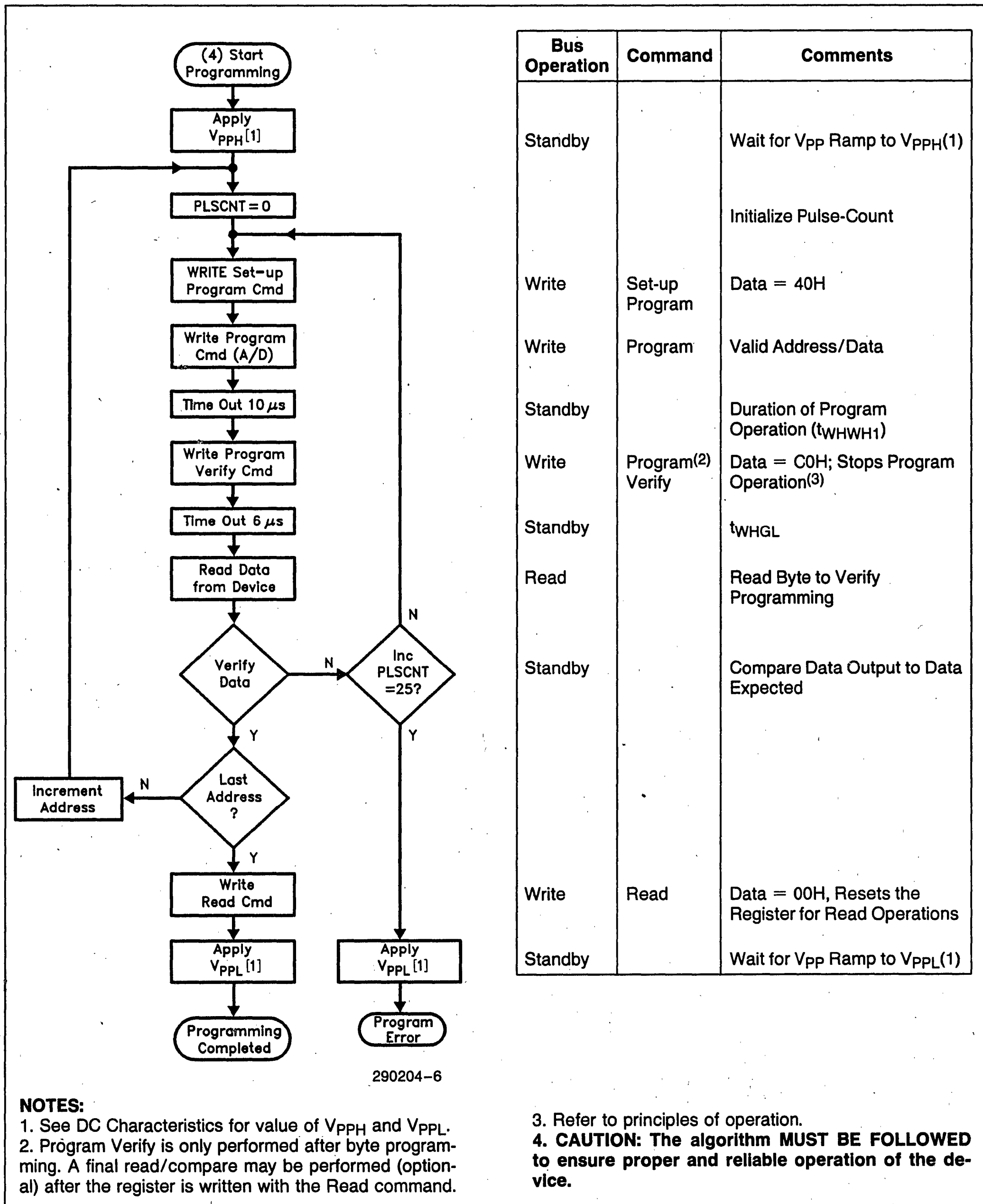
QUICK-ERASE ALGORITHM

Intel's Quick-Erase algorithm yields fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the Quick-Pulse Programming algorithm, to simultaneously remove charge from all bits in the array.

Erasure begins with a read of memory contents. The 28F512 is erased when shipped from the factory. Reading FFH data from the device would immediately be followed by device programming.

For devices being erased and reprogrammed, uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (Data = 00H). This is accomplished, using the Quick-Pulse Programming algorithm, in approximately one second.

Erase execution then continues with an initial erase operation. Erase verification (data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. With each erase operation, an increasing number of bytes verify to the erased state. Erase efficiency may be improved by storing the address of the last byte verified in a register. Following the next erase operation, verification starts at that stored address location. Erasure typically occurs in one second. Figure 5 illustrates the Quick-Erase algorithm.



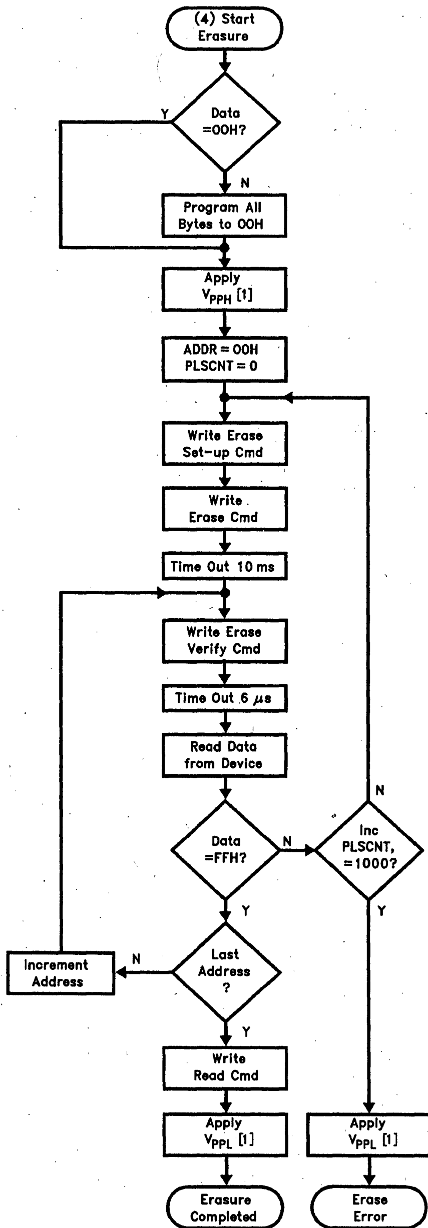
Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Pulse-Count
Write	Set-up Program	Data = 40H
Write	Program	Valid Address/Data
Standby		Duration of Program Operation (t_{WHWH1})
Write	Program(2) Verify	Data = C0H; Stops Program Operation(3)
Standby		t_{WHGL}
Read		Read Byte to Verify Programming
Standby		Compare Data Output to Data Expected
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

NOTES:

1. See DC Characteristics for value of V_{PPH} and V_{PPL}.
2. Program Verify is only performed after byte programming. A final read/compare may be performed (optional) after the register is written with the Read command.
3. Refer to principles of operation.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

Figure 4. 28F512 Quick-Pulse Programming Algorithm

3



Bus Operation	Command	Comments
		Entire memory must = 00H before erasure
		Use Quick-Pulse Programming™ Algorithm (Figure 4)
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Addresses and Pulse-Count
Write	Set-up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase(2) Verify	Addr = Byte to Verify; Data = A0H; Stops Erase Operation(3)
Standby		t_{WHGL}
Read		Read Byte to Verify Erasure
Standby		Compare Output to FFH Increment Pulse-Count
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

NOTES:

1. See DC Characteristics for value of V_{PPH} and V_{PPL}.
2. Erase Verify is performed only after chip-erasure. A final read/compare may be performed (optional) after the register is written with the read command.
3. Refer to principles of operation.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

Figure 5. 28F512 Quick-Erase Algorithm

DESIGN CONSIDERATIONS

Two-Line Output Control

Flash-memories are often used in larger memory arrays. Intel provides two read-control inputs to accommodate multiple memory connections. Two-line control provides for:

a. the lowest possible memory power dissipation and,

b. complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address-decoder output should drive chip-enable, while the system's read signal controls all flash-memories and other parallel memories. This assures that only enabled memory devices have active outputs, while deselected devices maintain the low power standby condition.

Power Supply Decoupling

Flash-memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current (I_{CC}) issues—standby, active, and transient current peaks produced by falling and rising edges of chip-enable. The capacitive and inductive loads on the device outputs determine the magnitudes of these peaks.

Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 μF ceramic capacitor connected between V_{CC} and V_{SS}, and between V_{PP} and V_{SS}.

Place the high-frequency, low-inherent-inductance capacitors as close as possible to the devices. Also, for every eight devices, a 4.7 μF electrolytic capacitor should be placed at the array's power supply connection, between V_{CC} and V_{SS}. The bulk capacitor will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

V_{PP} Trace on Printed Circuit Boards

Programming flash-memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the V_{PP} power supply trace. The V_{PP} pin supplies the memory cell current for programming. Use similar trace widths and layout considerations given the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots.

Power Up/Down Protection

The 28F512 is designed to offer protection against accidental erasure or programming during power transitions. Upon power-up, the 28F512 is indifferent as to which power supply, V_{PP} or V_{CC}, powers up first. Power supply sequencing is not required. Internal circuitry in the 28F512 ensures that the command register is reset to the read mode on power up.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will inhibit writes. The control register architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

28F512 Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases the usable battery life of your system because the 28F512 does not consume any power to retain code or data when the system is off. Table 4 illustrates the power dissipated when updating the 28F512.

Table 4. 28F512 Typical Update Power Dissipation(4)

Operation	Notes	Power Dissipation (Watt-Seconds)
Array Program/ Program Verify	1	0.085
Array Erase/ Erase Verify	2	0.092
One Complete Cycle	3	0.262

NOTES:

1. Formula to calculate typical Program/Program Verify Power = [V_{PP} × # Bytes x Typical # Prog Pulses (t_{WHWH1} × I_{PP2} Typical + t_{WHGL} × I_{PP4} Typical)] + [V_{CC} × # Bytes × Typical # Prog Pulses (t_{WHWH1} × I_{CC2} Typical + t_{WHGL} × I_{CC4} Typical).
2. Formula to calculate typical Erase/Erase Verify Power = [V_{PP}(I_{PP3} Typical × t_{ERASE} Typical + I_{PP5} Typical × t_{WHGL} × # Bytes)] + [V_{CC}(I_{CC3} Typical × t_{ERASE} Typical + I_{CC5} Typical × t_{WHGL} × # Bytes)].
3. One Complete Cycle = Array Preprogram + Array Erase + Program.
4. "Typicals" are not guaranteed, but based on a limited number of samples from production lots.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read ....0°C to +70°C(1)
During Erase/Program ....0°C to +70°C(1)

Operating Temperature
During Read ....−40°C to +85°C(2)
During Erase/Program ......−40°C to +85°C(2)

Temperature Under Bias.......−10°C to +80°C(1)

Temperature Under Bias.......−50°C to +95°C(2)

Storage Temperature ....−65°C to +125°C

Voltage on Any Pin with
Respect to Ground ....−2.0V to +7.0V(2)

Voltage on Pin A_9 with
Respect to Ground ....−2.0V to +13.5V(2, 3)

V_{PP} Supply Voltage with
Respect to Ground
During Erase/Program ....−2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage with
Respect to Ground ....−2.0V to +7.0V(2)

Output Short Circuit Current.............100 mA(4)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Operating temperature is for extended temperature product defined by this specification.
3. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
4. Maximum DC voltage on A_9 or V_{PP} may overshoot to +14.0V for periods less than 20 ns.
5. Output shorted for no more than one second. No more than one output shorted at a time.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature(1)	0	70	°C	For Read-Only and Read/Write Operations for Commercial Products
T_A	Operating Temperature(2)	−40	+85	°C	For Read-Only and Read/Write Operations for Extended Temperature Products
V_{CC}	V_{CC} Supply Voltage	4.50	5.50	V	

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE—Commercial Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10.0	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.3	1.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	15	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE—Commercial Products
(Continued)

Symbol	Parameter	Notes	Limits Min	Typ(4)	Max	Unit	Test Conditions
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10.0	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, Standby Current, or ID Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10.0		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage				0.45	V	$I_{OL} = 5.8$ mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	$I_{OH} = -2.5$ mA $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifier Voltage		11.50		13.00	V	
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	200	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—CMOS COMPATIBLE—Commercial Products

Symbol	Parameter	Notes	Limits Min	Typ(4)	Max	Unit	Test Conditions
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10.0	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		50	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC} \pm 0.2V$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, $I_{OUT} = 0$ mA

DC CHARACTERISTICS—CMOS COMPATIBLE—Commercial Products (Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ(4)	Max		
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	15	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10.0	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, ID Current, or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10.0		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		0.85 V_{CC}			V	I_{OH} = −2.5 mA, $V_{CC} = V_{CC}$ Min
V_{OH2}			V_{CC} − 0.4				I_{OH} = −100 μA, $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifier Voltage		11.50		13.00	V	$A_9 = V_{ID}$
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	200	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE—Extended Temperature Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10.0	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.3	1.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	30	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10.0	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, Standby Current, or ID Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10.0		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	I_{OH} = −2.5 mA $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifier Voltage		11.50		13.00	V	
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	500	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—CMOS COMPATIBLE—Extended Temperature Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10.0	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		50	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC} \pm 0.2V$
I_{CC1}	V_{CC} Active Read Current	1		10	50	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	15	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10.0	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, ID Current, or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10.0		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min

DC CHARACTERISTICS—CMOS COMPATIBLE—Extended Temperature Products (Continued)

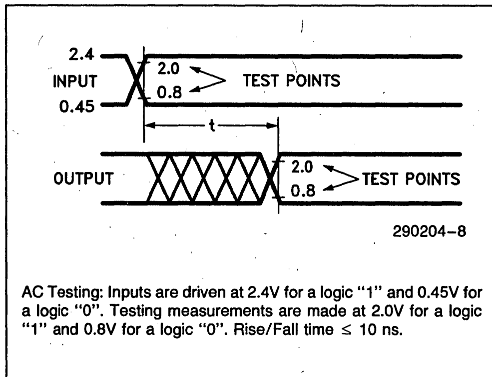
Symbol	Parameter	Notes	Limits Min	Limits Typ(4)	Limits Max	Unit	Test Conditions
V_{OH1}	Output High Voltage		0.85 V_{CC}			V	$I_{OH} = -2.5$ mA, $V_{CC} = V_{CC}$ Min
V_{OH2}			$V_{CC} - 0.4$				$I_{OH} = -100$ μA, $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifier Voltage		11.50		13.00	V	$A_9 = V_{ID}$
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	500	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

CAPACITANCE $T_A = 25°C$, f = 1.0 MHz

Symbol	Parameter	Notes	Limits Min	Limits Max	Unit	Conditions
C_{IN}	Address/Control Capacitance	3		8	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	3		12	pF	$V_{OUT} = 0V$

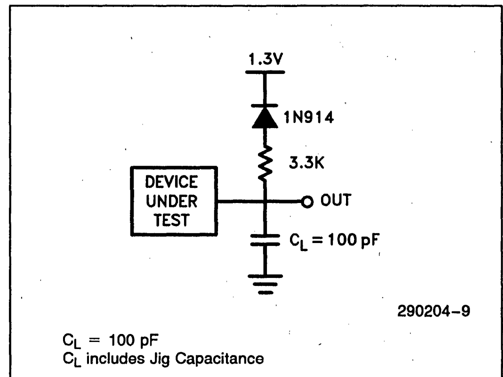
NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at $V_{CC} = 5.0V$, $V_{PP} = 12.0V$, T = +25°C. These currents are valid for all product versions (packages and speeds).
2. Not 100% tested: characterization data available.
3. Sampled, not 100% tested.
4. "Typicals" are not guaranteed, but based on a limited number of samples from production lots.

AC TESTING INPUT/OUTPUT WAVEFORM



AC Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Testing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0". Rise/Fall time ≤ 10 ns.

AC TESTING LOAD CIRCUIT



C_L = 100 pF
C_L includes Jig Capacitance

AC TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) . . . . . . 10 ns
Input Pulse Levels . . . . 0.45V and 2.4V
Input Timing Reference Level . . . . 0.8V and 2.0V
Output Timing Reference Level . . . . . . 0.8V and 2.0V

AC CHARACTERISTICS—Read-Only Operations

Versions[1]		Notes	N28F512-120 TN28F512-120 P28F512-120 TP28F512-120		N28F512-150 P28F512-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}/t_{RC}	Read Cycle Time		120		150		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time			120		150	ns
t_{AVQV}/t_{ACC}	Address Access Time			120		150	ns
t_{GLQV}/t_{OE}	Output Enable Access Time			50		55	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	2, 3	0		0		ns
t_{EHQZ}	Chip Disable to Output in High Z	2		55		55	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	2, 3	0		0		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	2		30		35	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	2, 4	0		0		ns
t_{WHGL}	Write Recovery Time before Read		6		6		µs

NOTES:
1. Model number prefixes: N = PLCC, P = PDIP, T = Extended Temperature.
2. Sampled, not 100% tested.
3. Guaranteed by design.
4. Whichever occurs first.

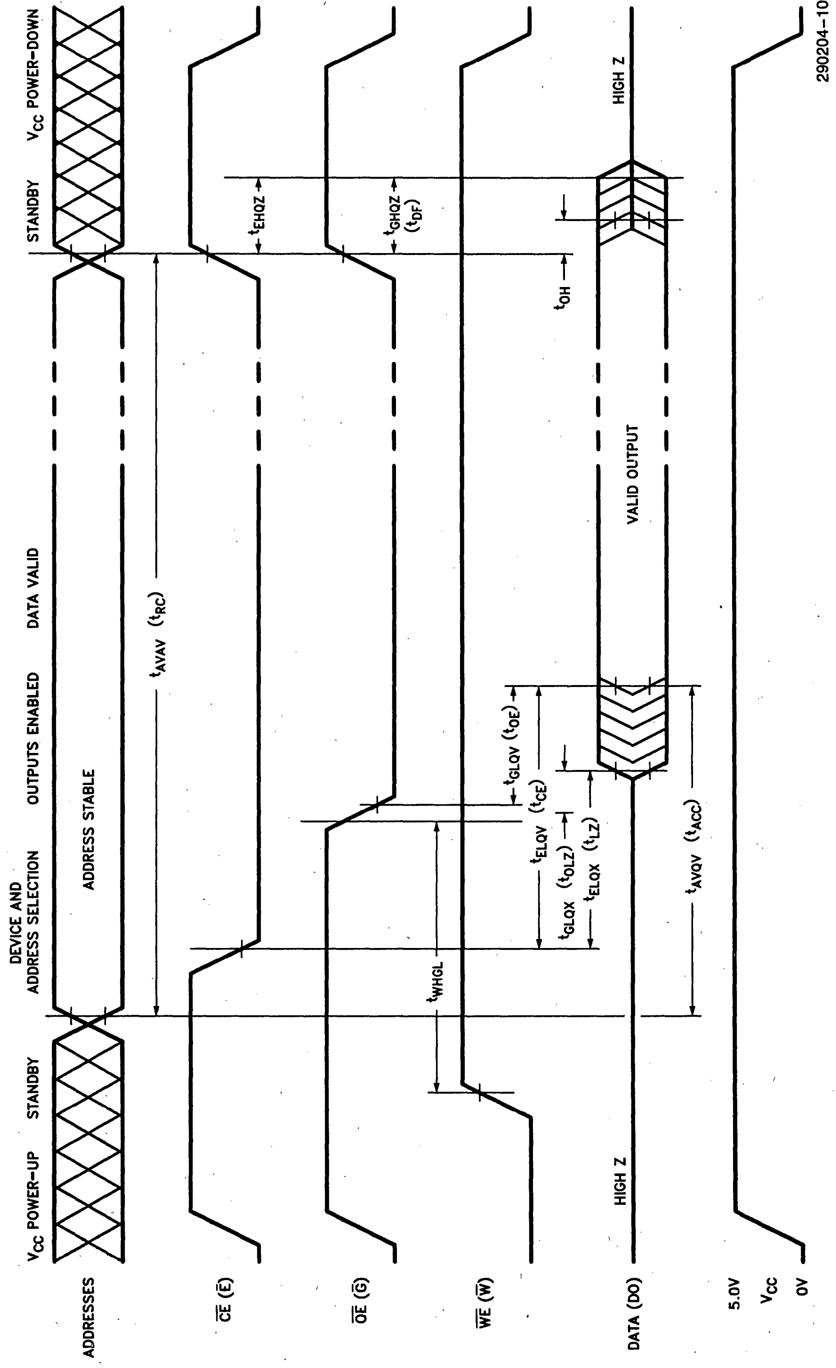


Figure 6. AC Waveforms for Read Operations

AC CHARACTERISTICS—Write/Erase/Program Operations(1)

Versions		Notes	28F512-120		28F512-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		120		150		ns
t_{AVWL}/t_{AS}	Address Set-Up Time		0		0		ns
t_{WLAX}/t_{AH}	Address Hold Time		60		60		ns
t_{DVWH}/t_{DS}	Data Set-up Time		50		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		10		ns
t_{WHGL}	Write Recovery Time before Read		6		6		μs
t_{GHWL}	Read Recovery Time before Write	2	0		0		μs
t_{ELWL}/t_{CS}	Chip Enable Set-Up Time before Write		20		20		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time		0		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width		60		60		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		20		ns
t_{WHWH1}	Duration of Programming Operation	3	10		10		μs
t_{WHWH2}	Duration of Erase Operation	3	9.5		9.5		ms
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low	2	1.0		1.0		μs

NOTES:
1. Read timing characteristics during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Guaranteed by design.
3. The integrated stop timer terminates the programming/erase operations, thereby eliminating the need for a maximum specification.

ERASE AND PROGRAMMING PERFORMANCE

Parameter	Notes	Limits						Unit
		N/P28F512-120, 150			TN/TP28F512-120(6)			
		Min	Typ	Max	Min	Typ	Max	
Chip Erase Time	1, 3, 4		1	10		1	10	Sec
Chip Program Time	1, 2, 4		1	6.25		1	6.25	Sec
Erase/Program Cycles	1, 5	10,000	100,000		1,000			Cycles

NOTES:
1. "Typicals" are not guaranteed, but based on a limited number of samples from production lots. Data taken at 25°C, 12.0V V_{PP} at 0 cycles.
2. Minimum byte programming time excluding system overhead is 16 μs (10 μs program + 6 μs write recovery), while maximum is 400 μs/byte (16 μs × 25 loops allowed by algorithm). Max chip programming time is specified lower than the worst case allowed by the programming algorithm since most bytes program significantly faster than the worst case byte.
3. Excludes 00H Programming Prior to Erasure.
4. Excludes System-Level Overhead.
5. Refer to RR-60 "ETOX II Flash Memory Reliability Data Summary" for typical cycling data and failure rate calculations.
6. Extended temperature products

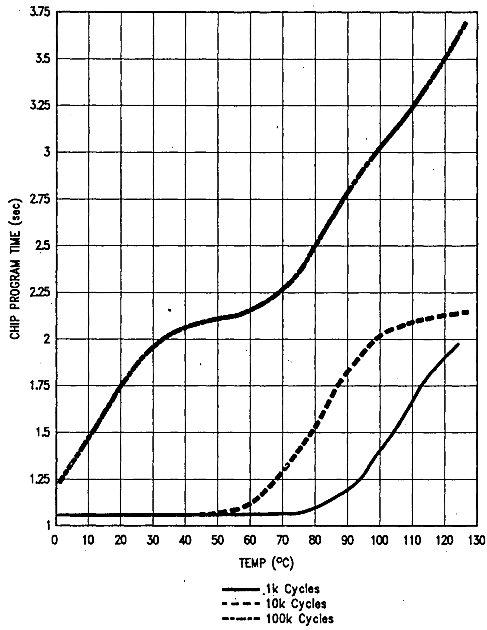


Figure 7. 28F512 Typical Program Time at 12V

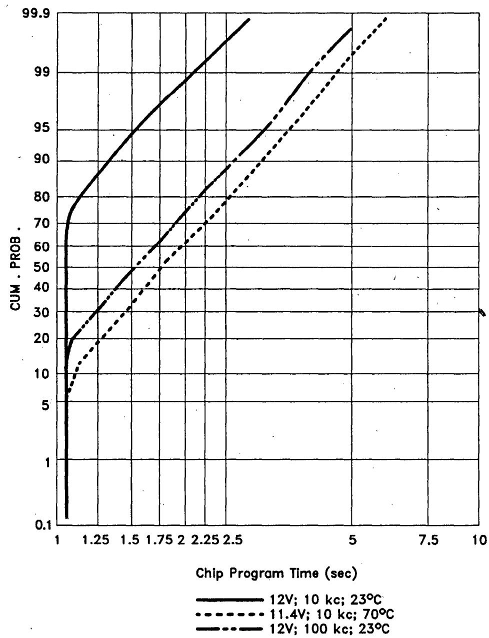
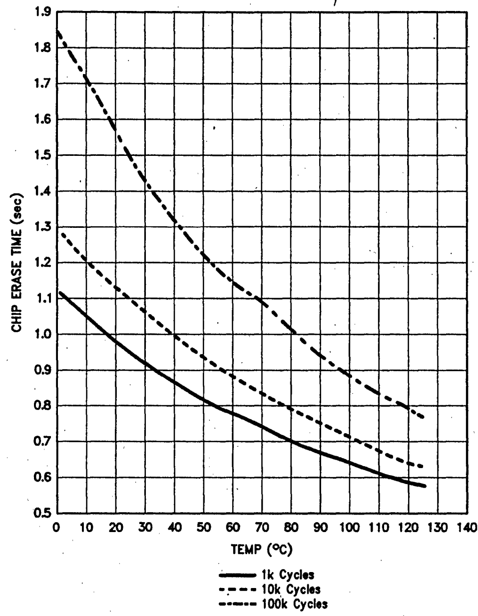


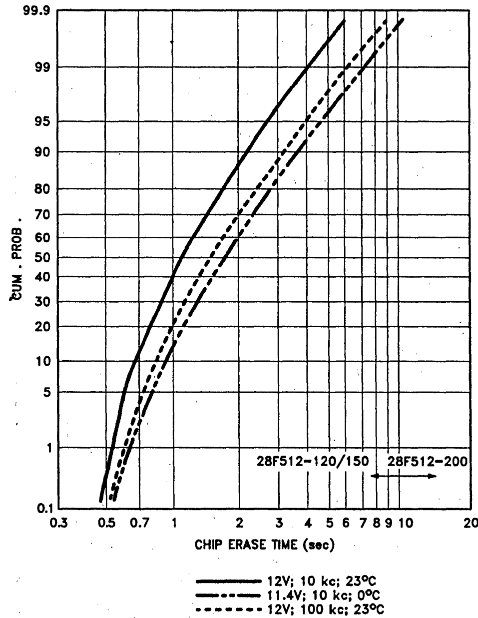
Figure 8. 28F512 Typical Programming Capability



290204-16

NOTE:
Does not include Pre-Erase program.

Figure 9. 28F512 Typical Erase Time at 12V



290204-17

NOTE:
Does not include Pre-Erase program.

Figure 10. 28F512 Typical Erase Capability

V_{CC} POWER-UP & STANDBY
SET-UP PROGRAM COMMAND
PROGRAM COMMAND LATCH ADDRESS & DATA
PROGRAMMING
PROGRAM VERIFY COMMAND
PROGRAM VERIFICATION
STANDBY/ V_{CC} POWER-DOWN

ADDRESSES V_{IH} V_{IL}
t_{AVAV} t_{AVAV} t_{AVAV} t_{AVAV}
t_{AVWL} t_{WLAX} t_{AVWL} t_{WLAX}

$\overline{CE}$ V_{IH} V_{IL}
t_{ELWL} t_{WHEH} t_{ELWL} t_{WHEH} t_{ELWL} t_{WHEH}

$\overline{OE}$ V_{IH} V_{IL}
t_{GHWL} t_{WHWL} t_{WHWH1} t_{WHGL} t_{GHQZ}

$\overline{WE}$ V_{IH} V_{IL}
t_{WLWH} t_{WHDX} t_{DVWH} t_{WLWH} t_{WHDX} t_{DVWH} t_{WLWH} t_{WHDX} t_{DVWH}
t_{OH} t_{GLQV} t_{GLQX}

DATA V_{IH} V_{IL}
HIGH Z
DATA IN = 40H
DATA IN
DATA IN = C0H
VALID DATA OUT
t_{ELQX} t_{ELQV}

V_{CC} 5.0V 0V

V_{PP} 12.0V V_{PPL}
t_{VPEL}

290204–19

Figure 11. AC Waveforms for Programming Operations

ALTERNATIVE $\overline{CE}$-CONTROLLED WRITES

Versions		Notes	28F512-120		28F512-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}	Write Cycle Time		120		150		ns
t_{AVEL}	Address Set-Up Time		0		0		ns
t_{ELAX}	Address Hold Time		80		80		ns
t_{DVEH}	Data Set-Up Time		50		50		ns
t_{EHDX}	Data Hold Time		10		10		ns
t_{EHGL}	Write Recovery Time before Read		6		6		µs
t_{GHEL}	Read Recovery Time before Write	2	0		0		µs
t_{WLEL}	Write Enable Set-Up Time before Chip Enable		0		0		ns
t_{EHWH}	Write Enable Hold Time		0		0		ns
t_{ELEH}	Write Pulse Width	1	70		70		ns
t_{EHEL}	Write Pulse Width High		20		20		ns
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low	2	1.0		1.0		µs

NOTE:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of Chip-Enable and Write-Enable. In systems where Chip-Enable defines the write pulse width (within a longer Write-Enable timing waveform) all set-up, hold and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.
2. Guaranteed by design.

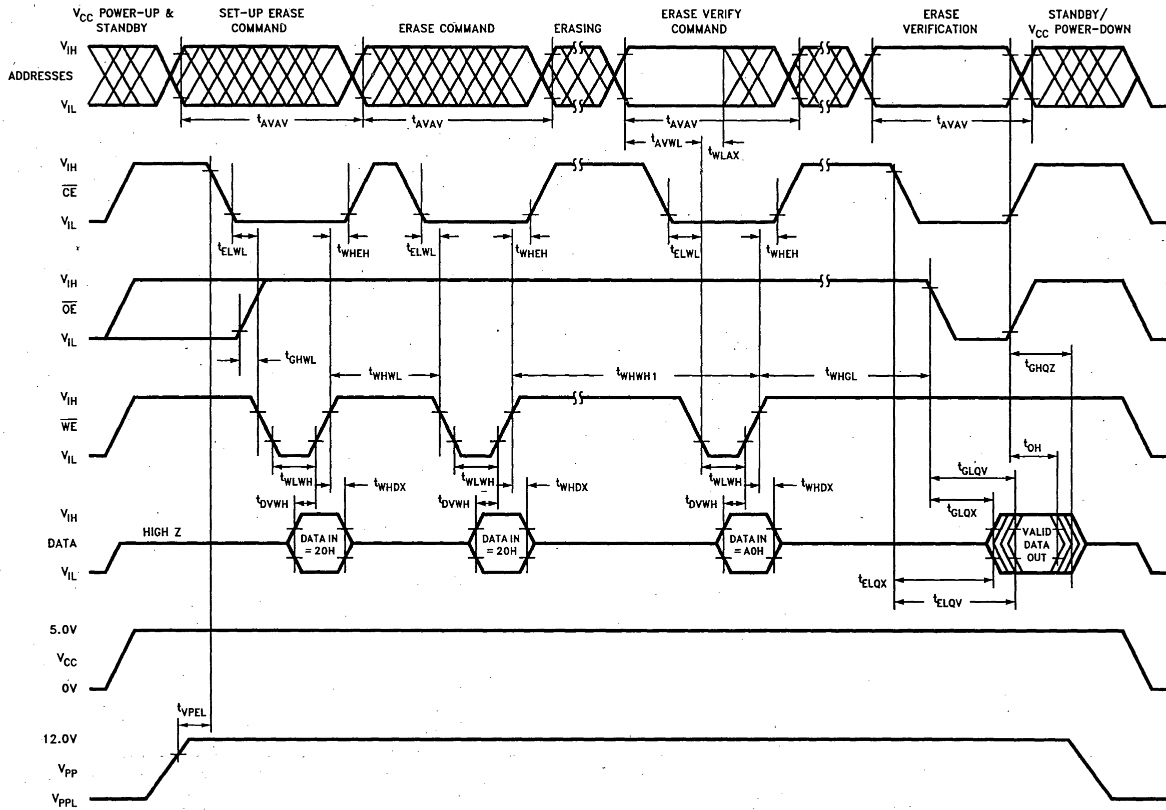


Figure 12. AC Waveforms for Erase Operations

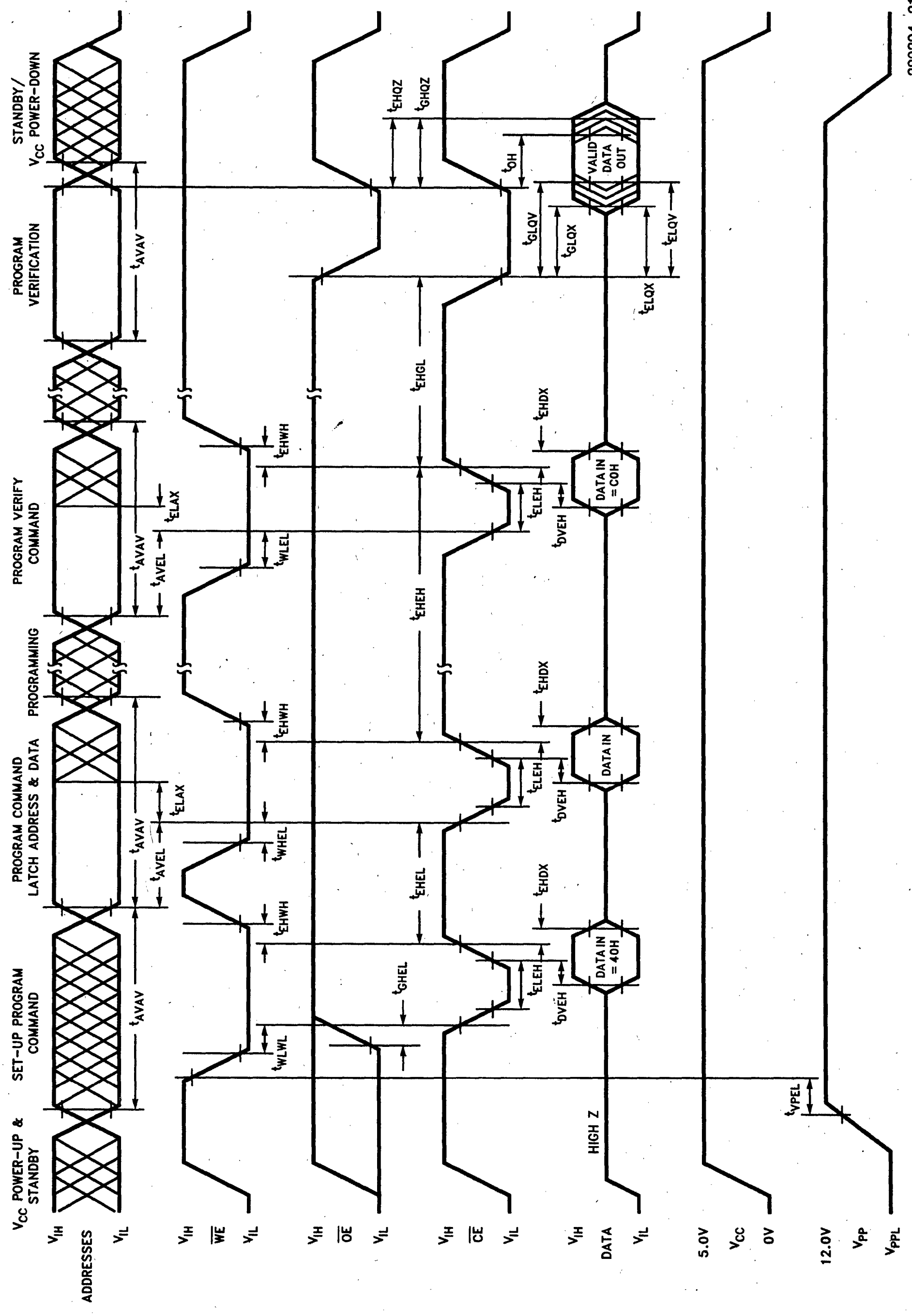
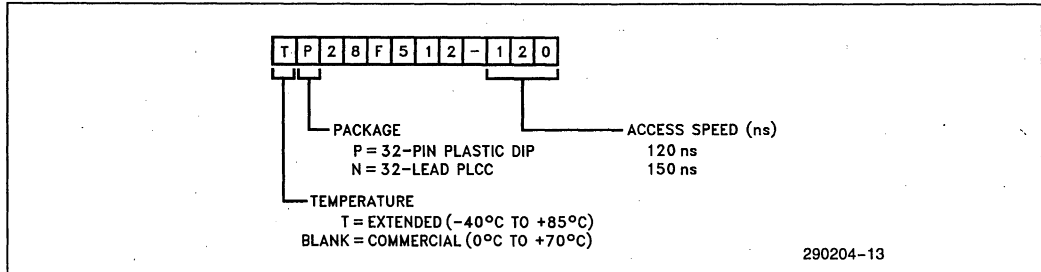


Figure 13. Alternate AC Waveforms for Programming Operations

Ordering Information



Valid Combinations:

P28F512-120	N28F512-120	TP28F512-120
P28F512-150	N28F512-150	TN28F512-120

ADDITIONAL INFORMATION

	Order Number
ER-20, "ETOX™ II Flash Memory Technology"	294005
ER-24, "Intel Flash Memory"	294008
RR-60, "ETOX™ II Flash Memory Reliability Data Summary"	293002
AP-316, "Using Flash Memory for In-System Reprogrammable Nonvolatile Storage"	292046
AP-325 "Guide to Flash Memory Reprogramming"	292059

REVISION HISTORY

Number	Description
006	Removed 200 ns speed bin Revised Erase Maximum Pulse Count for Figure 5 from 3000 to 1000 Clarified AC and DC test conditions
007	Corrected AC Waveforms Added Extended Temperature devices; TP28F512-120, TN28F512-120

28F010
1024K (128K x 8) CMOS FLASH MEMORY

- **Flash Electrical Chip-Erase**
  — 1 Second Typical Chip-Erase
- **Quick-Pulse Programming Algorithm**
  — 10 μs Typical Byte-Program
  — 2 Second Chip-Program
- **100,000 Erase/Program Cycles Typical**
- **12.0V ±5% V_{PP}**
- **High-Performance Read**
  — 90 ns Maximum Access Time
- **CMOS Low Power Consumption**
  — 10 mA Typical Active Current
  — 50 μA Typical Standby Current
  — 0 Watts Data Retention Power
- **Integrated Program/Erase Stop Timer**
- **Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface**
- **Noise Immunity Features**
  — ±10% V_{CC} Tolerance
  — Maximum Latch-Up Immunity through EPI Processing
- **ETOX™ II Nonvolatile Flash Technology**
  — EPROM-Compatible Process Base
  — High-Volume Manufacturing Experience
- **JEDEC-Standard Pinouts**
  — 32-Pin Plastic Dip
  — 32-Lead PLCC
  — 32-Lead TSOP
  (See Packaging Spec., Order #231369)
- **Extended Temperature Options**

Intel's 28F010 CMOS flash memory offers the most cost-effective and reliable alternative for read/write random access nonvolatile memory. The 28F010 adds electrical chip-erasure and reprogramming to familiar EPROM technology. Memory contents can be rewritten: in a test socket; in a PROM-programmer socket; on-board during subassembly test; in-system during final test; and in-system after-sale. The 28F010 increases memory flexibility, while contributing to time- and cost-savings.

The 28F010 is a 1024-kilobit nonvolatile memory organized as 131,072 bytes of 8 bits. Intel's 28F010 is offered in 32-pin plastic dip or 32-lead PLCC and TSOP packages. Pin assignments conform to JEDEC standards for byte-wide EPROMs.

Extended erase and program cycling capability is designed into Intel's ETOX II (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional EEPROMs. With the 12.0V V_{PP} supply, the 28F010 performs a minimum of 10,000 erase and program cycles well within the time limits of the Quick-Pulse Programming and Quick-Erase algorithms.

Intel's 28F010 employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 90 nanosecond access time provides no-WAIT-state performance for a wide range of microprocessors and microcontrollers. Maximum standby current of 100 μA translates into power savings when the device is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from −1V to V_{CC} + 1V.

With Intel's ETOX II process base, the 28F010 levers years of EPROM experience to yield the highest levels of quality, reliability, and cost-effectiveness.

November 1992
Order Number: 290207-008

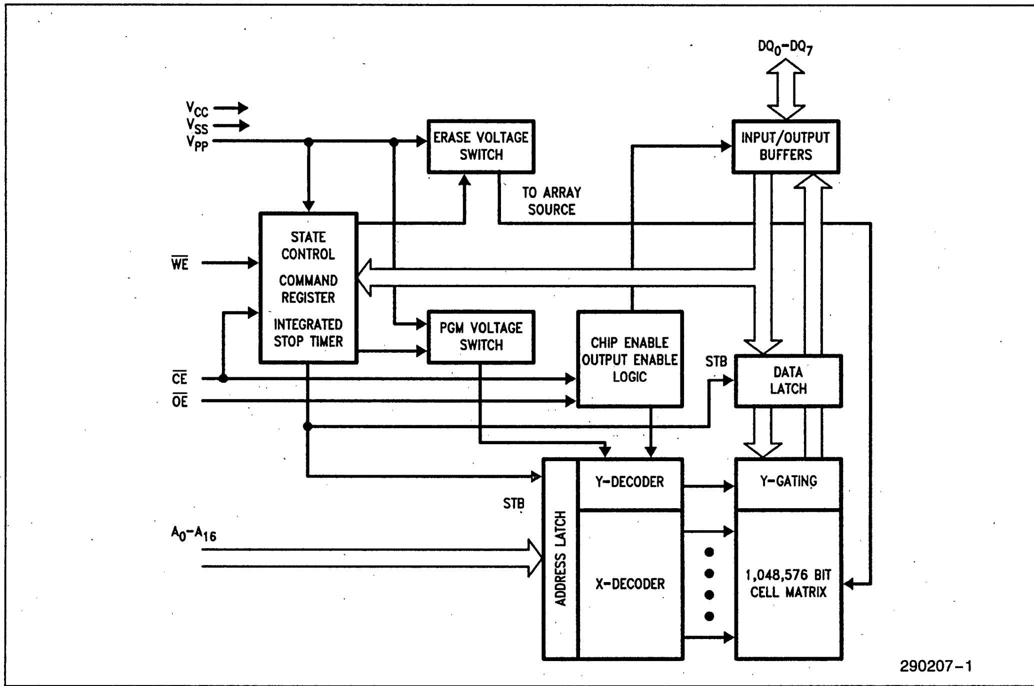


Figure 1. 28F010 Block Diagram

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{16}	INPUT	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
DQ_0–DQ_7	INPUT/OUTPUT	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}$	INPUT	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels.
$\overline{OE}$	INPUT	**OUTPUT ENABLE:** Gates the devices output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	INPUT	**WRITE ENABLE:** Controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **Note:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP}		**ERASE/PROGRAM POWER SUPPLY** for writing the command register, erasing the entire array, or programming bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY** (5V ±10%)
V_{SS}		**GROUND**
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating.

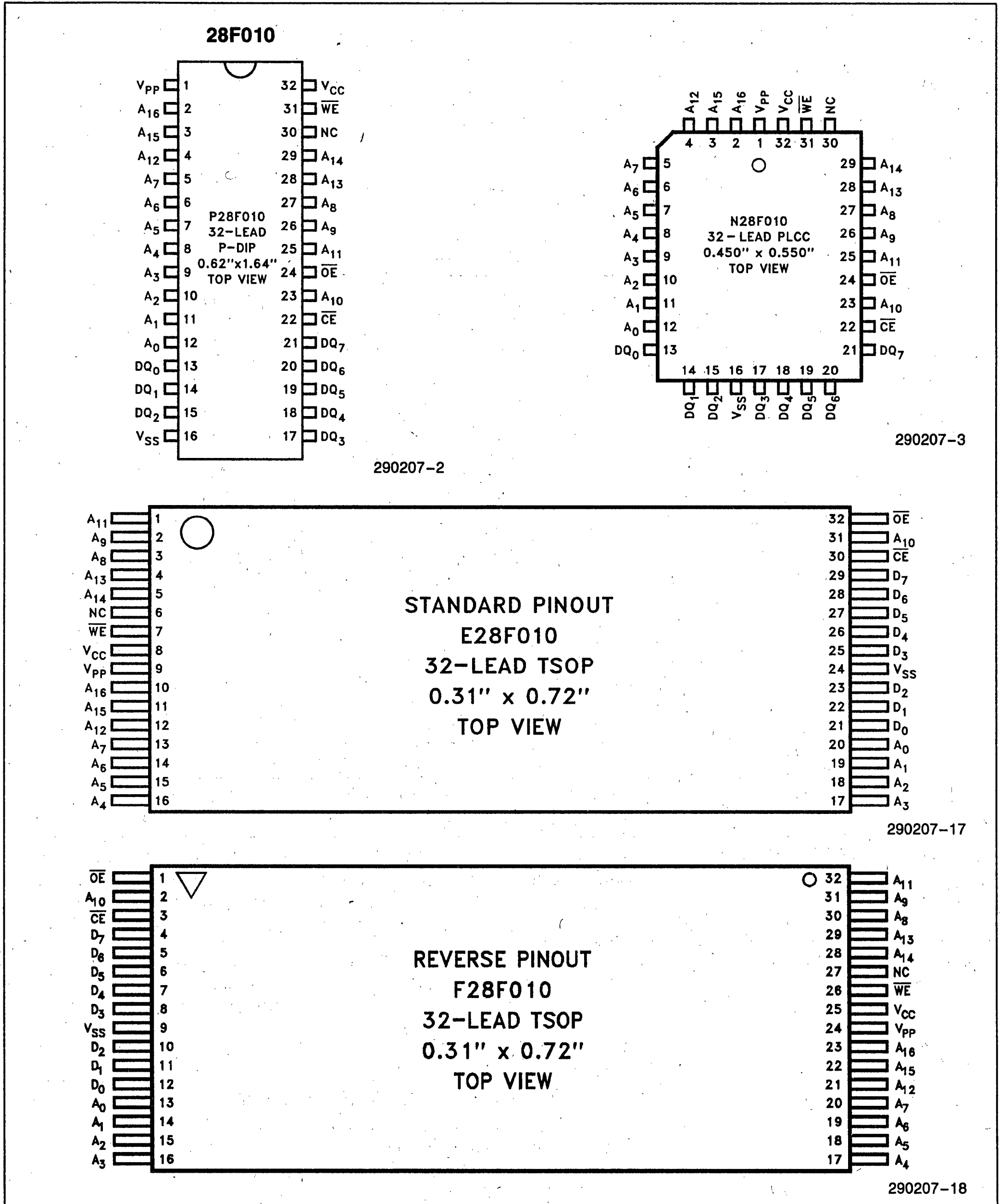


Figure 2. 28F010 Pin Configurations

APPLICATIONS

The 28F010 flash memory provides nonvolatility along with the capability to typically perform over 100,000 electrical chip-erasure/reprogram cycles. These features make the 28F010 an innovative alternative to disk, EEPROM, and battery-backed static RAM. Where periodic updates of code and datatables are required, the 28F010's reprogrammability and nonvolatility make it the obvious and ideal replacement for EPROM.

Primary applications and operating systems stored in flash eliminate the slow disk-to-DRAM download process. This results in dramatic enhancement of performance and substantial reduction of power consumption — a consideration particularly important in portable equipment. Flash memory increases flexibility with electrical chip erasure and in-system update capability of operating systems and application code. With updatable code, system manufacturers can easily accommodate last-minute changes as revisions are made.

In diskless workstations and terminals, network traffic reduces to a minimum and systems are instant-on. Reliability exceeds that of electromechanical media. Often in these environments, power interruptions force extended re-boot periods for all networked terminals. This mishap is no longer an issue if boot code, operating systems, communication protocols and primary applications are flash-resident in each terminal.

For embedded systems that rely on dynamic RAM/disk for main system memory or nonvolatile backup storage, the 28F010 flash memory offers a solid state alternative in a minimal form factor. The 28F010 provides higher performance, lower power consumption, instant-on capability, and allows an "execute in place" memory hierarchy for code and data table reading. Additionally, the flash memory is more rugged and reliable in harsh environments where extreme temperatures and shock can cause disk-based systems to fail.

The need for code updates pervades all phases of a system's life — from prototyping to system manufacture to after-sale service. The electrical chip-erasure and reprogramming ability of the 28F010 allows in-circuit alterability; this eliminates unnecessary handling and less-reliable socketed connections, while adding greater test, manufacture, and update flexibility.

Material and labor costs associated with code changes increases at higher levels of system integration — the most costly being code updates after sale. Code "bugs", or the desire to augment system functionality, prompt after-sale code updates. Field revisions to EPROM-based code requires the removal of EPROM components or entire boards. With the 28F010, code updates are implemented locally via an edge-connector, or remotely over a communcation link.

For systems currently using a high-density static RAM/battery configuration for data accumulation, flash memory's inherent nonvolatility eliminates the need for battery backup. The concern for battery failure no longer exists, an important consideration for portable equipment and medical instruments, both requiring continuous performance. In addition, flash memory offers a considerable cost advantage over static RAM.

Flash memory's electrical chip erasure, byte programmability and complete nonvolatility fit well with data accumulation and recording needs. Electrical chip-erasure gives the designer a "blank slate" in which to log or record data. Data can be periodically off-loaded for analysis and the flash memory erased producing a new "blank slate".

A high degree of on-chip feature integration simplifies memory-to-processor interfacing. Figure 4 depicts two 28F010s tied to the 80C186 system bus. The 28F010's architecture minimizes interface circuitry needed for complete in-circuit updates of memory contents.

The outstanding feature of the TSOP (Thin Small Outline Package) is the 1.2 mm thickness. With standard and reverse pin configurations, TSOP reduces the number of board layers and overall volume necessary to layout multiple 28F010s. TSOP is particularly suited for portable equipment and applications requiring large amounts of flash memory. Figure 3 illustrates the TSOP Serpentine layout.

With cost-effective in-system reprogramming, extended cycling capability, and true nonvolatility, the 28F010 offers advantages to the alternatives: EPROMs, EEPROMs, battery backed static RAM, or disk. EPROM-compatible read specifications, straight-forward interfacing, and in-circuit alterability offers designers unlimited flexibility to meet the high standards of today's designs.

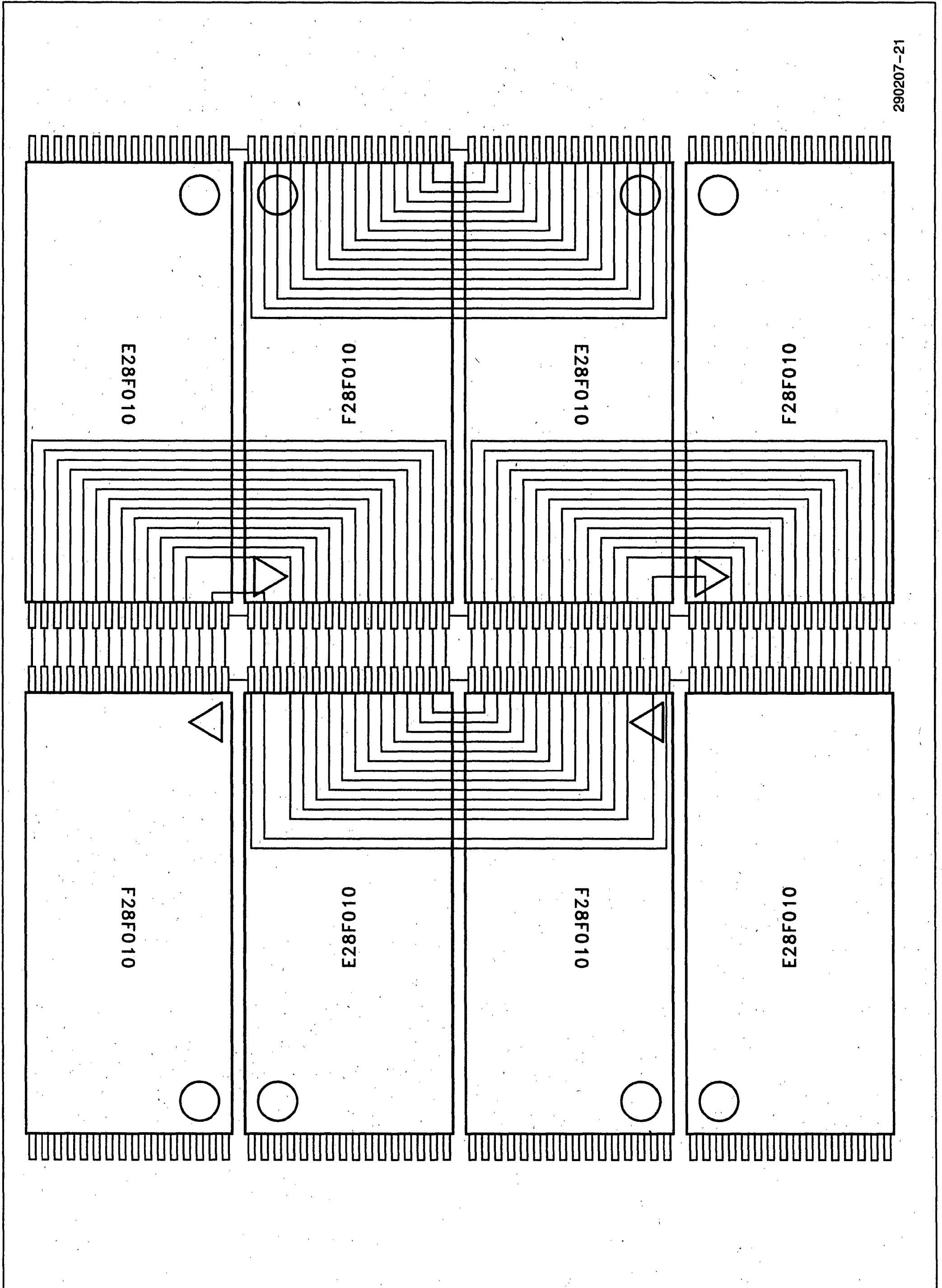


Figure 3. TSOP Serpentine Layout

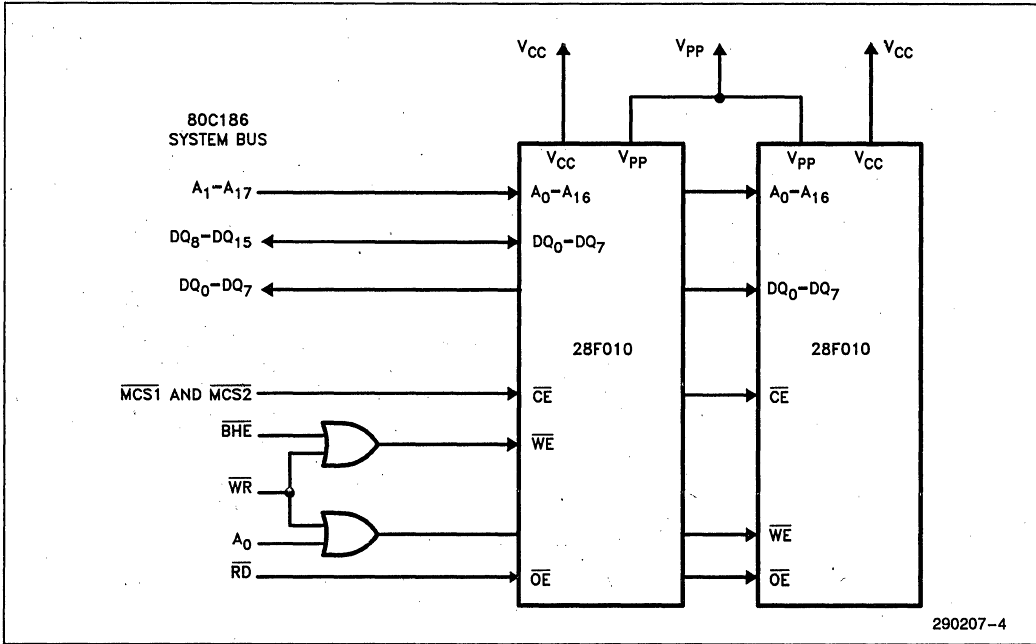


Figure 4. 28F010 in a 80C186 System

PRINCIPLES OF OPERATION

Flash-memory augments EPROM functionality with in-circuit electrical erasure and reprogramming. The 28F010 introduces a command register to manage this new functionality. The command register allows for: 100% TTL-level control inputs; fixed power supplies during erasure and programming; and maximum EPROM compatibility.

In the absence of high voltage on the V_{PP} pin, the 28F010 is a read-only memory. Manipulation of the external memory-control pins yields the standard EPROM read, standby, output disable, and int$_e$ligent Identifier™ operations.

The same EPROM read, standby, and output disable operations are available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} enables erasure and programming of the device. All functions associated with altering memory contents—int$_e$ligent Identifier, erase, erase verify, program, and program verify—are accessed via the command register.

Commands are written to the register using standard microprocessor write timings. Register contents serve as input to an internal state-machine which controls the erase and programming circuitry. Write cycles also internally latch addresses and data needed for programming or erase operations. With the appropriate command written to the register, standard microprocessor read timings output array data, access the int$_e$ligent Identifier codes, or output data for erase and program verification.

Integrated Stop Timer

Successive command write cycles define the durations of program and erase operations; specifically, the program or erase time durations are normally terminated by associated program or erase verify commands. An integrated stop timer provides simplified timing control over these operations; thus eliminating the need for maximum program/erase timing specifications. Programming and erase pulse durations are minimums only. When the stop timer terminates a program or erase operation, the device enters an inactive state and remains inactive until receiving the appropriate verify or reset command.

Write Protection

The command register is only active when V_{PP} is at high voltage. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable—available only when memory updates are desired. When $V_{PP} = V_{PPL}$, the con-

Table 2. 28F010 Bus Operations

Operation		V_{PP}(1)	A_0	A_9	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_0-DQ_7
READ-ONLY	Read	V_{PPL}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out
	Output Disable	V_{PPL}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby	V_{PPL}	X	X	V_{IH}	X	X	Tri-State
	Intelligent Identifier (Mfr)(2)	V_{PPL}	V_{IL}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = 89H
	Intelligent Identifier (Device)(2)	V_{PPL}	V_{IH}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = B4H
READ/WRITE	Read	V_{PPH}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out(4)
	Output Disable	V_{PPH}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby(5)	V_{PPH}	X	X	V_{IH}	X	X	Tri-State
	Write	V_{PPH}	A_0	A_9	V_{IL}	V_{IH}	V_{IL}	Data In(6)

NOTES:
1. Refer to DC Characteristics. When $V_{PP} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes may also be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
3. V_{ID} is the Intelligent Identifier high voltage. Refer to DC Characteristics.
4. Read operations with $V_{PP} = V_{PPH}$ may access array data or the Intelligent Identifier codes.
5. With V_{PP} at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
6. Refer to Table 3 for valid Data-In during a write operation.
7. X can be V_{IL} or V_{IH}.

tents of the register default to the read command, making the 28F010 a read-only memory. In this mode, the memory contents cannot be altered.

Or, the system designer may choose to "hardwire" V_{PP}, making the high voltage supply constantly available. In this case, all Command Register functions are inhibited whenever V_{CC} is below the write lockout voltage V_{LKO}. (See Power Up/Down Protection) The 28F010 is designed to accommodate either design practice, and to encourage optimization of the processor-memory interface.

The two-step program/erase write sequence to the Command Register provides additional software write protections.

BUS OPERATIONS

Read

The 28F010 has two control functions, both of which must be logically active, to obtain data at the outputs. Chip-Enable ($\overline{CE}$) is the power control and should be used for device selection. Output-Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Refer to AC read timing waveforms.

When V_{PP} is high (V_{PPH}), the read operation can be used to access array data, to output the Intelligent Identifier codes, and to access data for program/erase verification. When V_{PP} is low (V_{PPL}), the read operation can **only** access the array data.

Output Disable

With Output-Enable at a logic-high level (V_{IH}), output from the device is disabled. Output pins are placed in a high-impedance state.

Standby

With Chip-Enable at a logic-high level, the standby operation disables most of the 28F010's circuitry and substantially reduces device power consumption. The outputs are placed in a high-impedance state, independent of the Output-Enable signal. If the 28F010 is deselected during erasure, programming, or program/erase verification, the device draws active current until the operation is terminated.

Intelligent Identifier Operation

The intelligent Identifier operation outputs the manufacturer code (89H) and device code (B4H). Programming equipment automatically matches the device with its proper erase and programming algorithms.

With Chip-Enable and Output-Enable at a logic low level, raising A9 to high voltage V_{ID} (see DC Characteristics) activates the operation. Data read from locations 0000H and 0001H represent the manufacturer's code and the device code, respectively.

The manufacturer- and device-codes can also be read via the command register, for instances where the 28F010 is erased and reprogrammed in the target system. Following a write of 90H to the command register, a read from address location 0000H outputs the manufacturer code (89H). A read from address 0001H outputs the device code (B4H).

Write

Device erasure and programming are accomplished via the command register, when high voltage is applied to the V_{PP} pin. The contents of the register serve as input to the internal state-machine. The state-machine outputs dictate the function of the device.

The command register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

The command register is written by bringing Write-Enable to a logic-low level (V_{IL}), while Chip-Enable is low. Addresses are latched on the falling edge of Write-Enable, while data is latched on the rising edge of the Write-Enable pulse. Standard microprocessor write timings are used.

Refer to AC Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the V_{PP} pin, the contents of the command register default to 00H, enabling read-only operations.

Placing high voltage on the V_{PP} pin enables read/write operations. Device operations are selected by writing specific data patterns into the command register. Table 3 defines these 28F010 register commands.

Table 3. Command Definitions

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	1	Write	X	00H			
Read Intelligent Identifier Codes(4)	3	Write	X	90H	Read	(4)	(4)
Set-up Erase/Erase(5)	2	Write	X	20H	Write	X	20H
Erase Verify(5)	2	Write	EA	A0H	Read	X	EVD
Set-up Program/Program(6)	2	Write	X	40H	Write	PA	PD
Program Verify(6)	2	Write	X	C0H	Read	X	PVD
Reset(7)	2	Write	X	FFH	Write	X	FFH

NOTES:
1. Bus operations are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   PA = Address of memory location to be programmed.
   Addresses are latched on the falling edge of the Write-Enable pulse.
3. ID = Data read from location IA during device identification (Mfr = 89H, Device = B4H).
   EVD = Data read from location EA during erase verify.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of Write-Enable.
   PVD = Data read from location PA during program verify. PA is latched on the Program command.
4. Following the Read int$_e$ligent ID command, two read operations access manufacturer and device codes.
5. Figure 6 illustrates the Quick-Erase Algorithm.
6. Figure 5 illustrates the Quick-Pulse Programming Algorithm.
7. The second bus cycle must be followed by the desired command register write.

Read Command

While V_{PP} is high, for erasure and programming, memory contents can be accessed via the read command. The read operation is initiated by writing 00H into the command register. Microprocessor read cycles retrieve array data. The device remains enabled for reads until the command register contents are altered.

The default contents of the register upon V_{PP} power-up is 00H. This default value ensures that no spurious alteration of memory contents occurs during the V_{PP} power transition. Where the V_{PP} supply is hard-wired to the 28F010, the device powers-up and remains enabled for reads until the command-register contents are changed. Refer to the AC Read Characteristics and Waveforms for specific timing parameters.

Intelligent Identifier Command

Flash-memories are intended for use in applications where the local CPU alters memory contents. As such, manufacturer- and device-codes must be accessible while the device resides in the target system. PROM programmers typically access signature codes by raising A9 to a high voltage. However, multiplexing high voltage onto address lines is not a desired system-design practice.

The 28F010 contains an Intelligent Identifier operation to supplement traditional PROM-programming methodology. The operation is initiated by writing 90H into the command register. Following the command write, a read cycle from address 0000H retrieves the manufacturer code of 89H. A read cycle from address 0001H returns the device code of B4H. To terminate the operation, it is necessary to write another valid command into the register.

Set-up Erase/Erase Commands

Set-up Erase is a command-only operation that stages the device for electrical erasure of all bytes in the array. The set-up erase operation is performed by writing 20H to the command register.

To commence chip-erasure, the erase command (20H) must again be written to the register. The erase operation begins with the rising edge of the Write-Enable pulse and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, chip-erasure can only occur when high voltage is applied to the V_{PP} pin. In the absence of this high voltage, memory contents are protected against erasure. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all bytes of the array in parallel. After each erase operation, all bytes must be verified. The erase verify operation is initiated by writing A0H into the command register. The address for the byte to be verified must be supplied as it is latched on the falling edge of the Write-Enable pulse. The register write terminates the erase operation with the rising edge of its Write-Enable pulse.

The 28F010 applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased.

The erase-verify command must be written to the command register prior to each byte verification to latch its address. The process continues for each byte in the array until a byte does not return FFH data, or the last address is accessed.

In the case where the data read is not FFH, another erase operation is performed. (Refer to Set-up Erase/Erase). Verification then resumes from the address of the last-verified byte. Once all bytes in the array have been verified, the erase step is complete. The device can be programmed. At this point, the verify operation is terminated by writing a valid command (e.g. Program Set-up) to the command register. Figure 6, the Quick-Erase algorithm, illustrates how commands and bus operations are combined to perform electrical erasure of the 28F010. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Set-up Program/Program Commands

Set-up program is a command-only operation that stages the device for byte programming. Writing 40H into the command register performs the set-up operation.

Once the program set-up operation is performed, the next Write-Enable pulse causes a transition to an active programming operation. Addresses are internally latched on the falling edge of the Write-Enable pulse. Data is internally latched on the rising edge of the Write-Enable pulse. The rising edge of Write-Enable also begins the programming operation. The programming operation terminates with the next rising edge of Write-Enable, used to write the program-verify command. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Program-Verify Command

The 28F010 is programmed on a byte-by-byte basis. Byte programming may occur sequentially or at random. Following each programming operation, the byte just programmed must be verified.

The program-verify operation is initiated by writing C0H into the command register. The register write terminates the programming operation with the rising edge of its Write-Enable pulse. The program-verify operation stages the device for verification of the byte last programmed. No new address information is latched.

The 28F010 applies an internally-generated margin voltage to the byte. A microprocessor read cycle outputs the data. A successful comparison between the programmed byte and true data means that the byte is successfully programmed. Programming then proceeds to the next desired byte location. Figure 5, the 28F010 Quick-Pulse Programming algorithm, illustrates how commands are combined with bus operations to perform byte programming. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or program-command sequences. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

EXTENDED ERASE/PROGRAM CYCLING

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled—an expensive solution.

Intel has designed extended cycling capability into its ETOX II flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field is one-tenth that of common EEPROMs, minimizing the probability of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 MV/cm lower than EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure—increasing time to wearout by a factor of 100,000,000.

The 28F010 is specified for a minimum of 10,000 program/erase cycles. The device is programmed and erased using Intel's Quick-Pulse Programming and Quick-Erase algorithms. Intel's algorithmic approach uses a series of operations (pulses), along with byte verification, to completely and reliably erase and program the device.

For further information, see Reliability Report RR-60.

QUICK-PULSE PROGRAMMING ALGORITHM

The Quick-Pulse Programming algorithm uses programming operations of 10 μs duration. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The algorithm allows for up to 25 programming operations per byte, although most bytes verify on the first or second operation. The entire sequence of programming and byte verification is performed with V_{PP} at high voltage. Figure 5 illustrates the Quick-Pulse Programming algorithm.

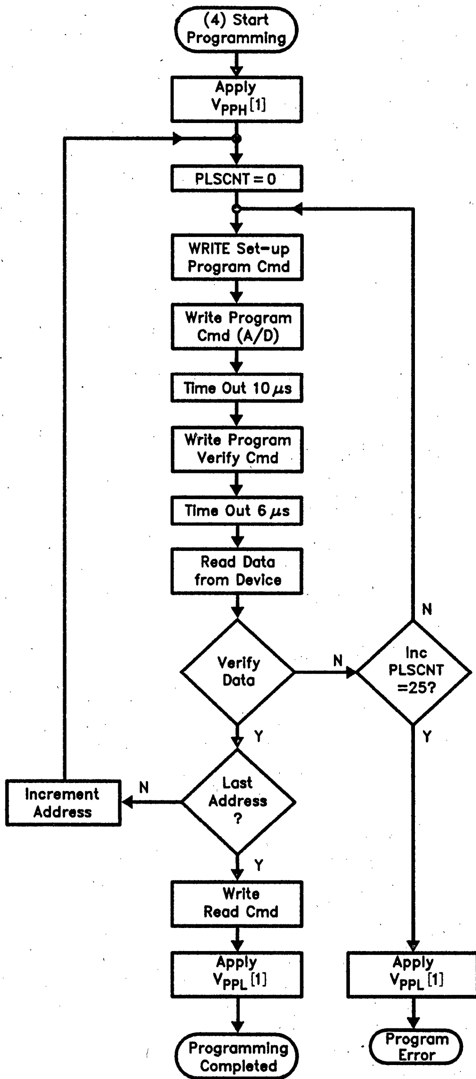
QUICK-ERASE ALGORITHM

Intel's Quick-Erase algorithm yields fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the Quick-Pulse Programming algorithm, to simultaneously remove charge from all bits in the array.

Erasure begins with a read of memory contents. The 28F010 is erased when shipped from the factory. Reading FFH data from the device would immediately be followed by device programming.

For devices being erased and reprogrammed, uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (Data = 00H). This is accomplished, using the Quick-Pulse Programming algorithm, in approximately two seconds.

Erase execution then continues with an initial erase operation. Erase verification (data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. With each erase operation, an increasing number of bytes verify to the erased state. Erase efficiency may be improved by storing the address of the last byte verified in a register. Following the next erase operation, verification starts at that stored address location. Erasure typically occurs in one second. Figure 6 illustrates the Quick-Erase algorithm.

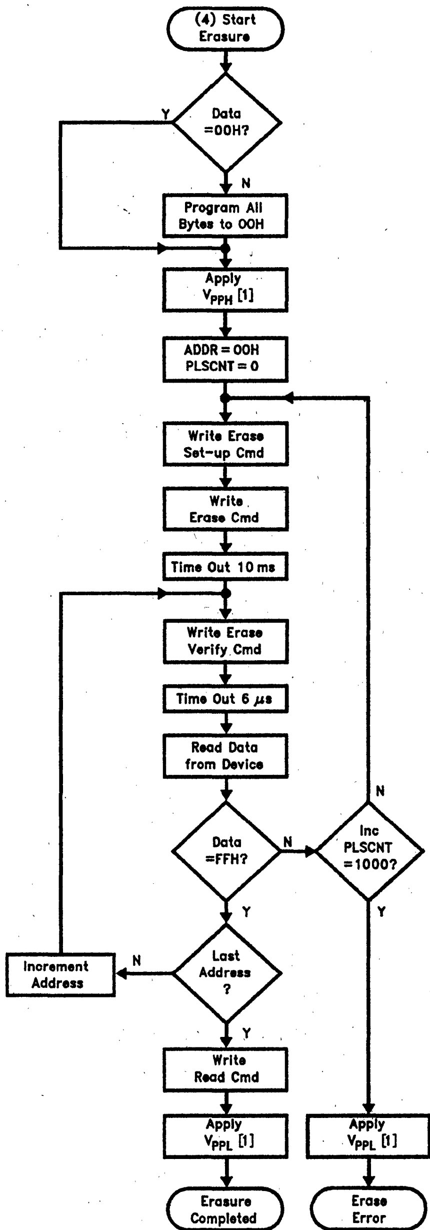


Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Pulse-Count
Write	Set-up Program	Data = 40H
Write	Program	Valid Address/Data
Standby		Duration of Program Operation (t_{WHWH1})
Write	Program(2) Verify	Data = C0H; Stops Program Operation(3)
Standby		t_{WHGL}
Read		Read Byte to Verify Programming
Standby		Compare Data Output to Data Expected
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

NOTES:

1. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
2. Program Verify is only performed after byte programming. A final read/compare may be performed (optional) after the register is written with the Read command.
3. Refer to principles of operation.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

Figure 5. 28F010 Quick-Pulse Programming Algorithm



290207-6

Bus Operation	Command	Comments
		Entire Memory Must = 00H Before Erasure
		Use Quick-Pulse Programming Algorithm (Figure 5)
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Addresses and Pulse-Count
Write	Set-up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase(2) Verify	Addr = Byte to Verify; Data = A0H; Stops Erase Operation(3)
Standby		t_{WHGL}
Read		Read Byte to Verify Erasure
Standby		Compare Output to FFH Increment Pulse-Count
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

1. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
2. Erase Verify is performed only after chip-erasure. A final read/compare may be performed (optional) after the register is written with the read command.
3. Refer to principles of operation.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

Figure 6. 28F010 Quick-Erase Algorithm

DESIGN CONSIDERATIONS

Two-Line Output Control

Flash-memories are often used in larger memory arrays. Intel provides two read-control inputs to accommodate multiple memory connections. Two-line control provides for:

a. the lowest possible memory power dissipation and,
b. complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address-decoder output should drive chip-enable, while the system's read signal controls all flash-memories and other parallel memories. This assures that only enabled memory devices have active outputs, while deselected devices maintain the low power standby condition.

Power Supply Decoupling

Flash-memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current (I_{CC}) issues—standby, active, and transient current peaks produced by falling and rising edges of chip-enable. The capacitive and inductive loads on the device outputs determine the magnitudes of these peaks.

Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 μF ceramic capacitor connected between V_{CC} and V_{SS}, and between V_{PP} and V_{SS}.

Place the high-frequency, low-inherent-inductance capacitors as close as possible to the devices. Also, for every eight devices, a 4.7 μF electrolytic capacitor should be placed at the array's power supply connection, between V_{CC} and V_{SS}. The bulk capacitor will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

V_{PP} Trace on Printed Circuit Boards

Programming flash-memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the V_{PP} power supply trace. The V_{PP} pin supplies the memory cell current for programming. Use similar trace widths and layout considerations given the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots.

Power Up/Down Protection

The 28F010 is designed to offer protection against accidental erasure or programming during power transitions. Upon power-up, the 28F010 is indifferent as to which power supply, V_{PP} or V_{CC}, powers up first. Power supply sequencing is not required. Internal circuitry in the 28F010 ensures that the command register is reset to the read mode on power up.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will inhibit writes. The control register architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

28F010 Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases the usable battery life of your system because the 28F010 does not consume any power to retain code or data when the system is off. Table 4 illustrates the power dissipated when updating the 28F010.

Table 4. 28F010 Typical Update Power Dissipation(4)

Operation	Notes	Power Dissipation (Watt-Seconds)
Array Program/Program Verify	1	0.171
Array Erase/Erase Verify	2	0.136
One Complete Cycle	3	0.478

NOTES:
1. Formula to calculate typical Program/Program Verify Power = [V_{PP} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{PP2} typical + t_{WHGL} × I_{PP4} typical)] + [V_{CC} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{CC2} typical + t_{WHGL} × I_{CC4} typical].
2. Formula to calculate typical Erase/Erase Verify Power = [V_{PP} (V_{PP3} typical × t_{ERASE} typical + I_{PP5} typical × t_{WHGL} × # Bytes)] + [V_{CC} (I_{CC3} typical × t_{ERASE} typical + I_{CC5} typical × t_{WHGL} × # Bytes)].
3. One Complete Cycle = Array Preprogram + Array Erase + Program.
4. "Typicals" are not guaranteed, but based on a limited number of samples from production lots.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read .... 0°C to +70°C(1)
During Erase/Program .... 0°C to +70°C(1)
Operating Temperature
During Read .... −40°C to +85°C(2)
During Erase/Program ...... −40°C to +85°C(2)
Temperature Under Bias .... −10°C to +80°C(1)
Temperature Under Bias .... −50°C to +95°C(2)
Storage Temperature .... −65°C to +125°C
Voltage on Any Pin with
Respect to Ground .... −2.0V to +7.0V(3)
Voltage on Pin A_9 with
Respect to Ground .... −2.0V to +13.5V(3, 4)
V_{PP} Supply Voltage with
Respect to Ground
During Erase/Program .... −2.0V to +14.0V(3, 4)
V_{CC} Supply Voltage with
Respect to Ground .... −2.0V to +7.0V(3)
Output Short Circuit Current .... 100 mA(5)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating Temperature is for commercial product defined by this specification.
2. Operating Temperature is for extended temperature products as defined by this specification.
3. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
4. Maximum DC voltage on A_9 or V_{PP} may overshoot to +14.0V for periods less than 20 ns.
5. Output shorted for no more than one second. No more than one output shorted at a time.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature	0	70	°C	For Read-Only and Read/Write Operations for Commercial Products
T_A	Operating Temperature	−40	+85	°C	For Read-Only and Read/Write Operations for Extended Temperature Products
V_{CC}	V_{CC} Supply Voltage (10%)	4.50	5.50	V	
V_{CC}	V_{CC} Supply Voltage (5%)	4.75	5.25	V	For 28F010-90V05 Only

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE—Commercial Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.3	1.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE—Commercial Products

(Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical(4)	Max		
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	15	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP}Read Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10.0		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		6.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	I_{OH} = −2.5 mA $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	200	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—CMOS COMPATIBLE—Commercial Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		50	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC}$ ±0.2V
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA

DC CHARACTERISTICS—CMOS COMPATIBLE—Commercial Products (Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical(4)	Max		
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	15	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, ID Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		6.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		0.85 V_{CC}			V	I_{OH} = −2.5 mA, $V_{CC} = V_{CC}$ Min
V_{OH2}			V_{CC} − 0.4				I_{OH} = −100 μA, $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	200	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Programs are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE—Extended Temperature Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.3	1.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	30	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10.0		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		6.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	I_{OH} = −2.5 mA $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	500	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—CMOS COMPATIBLE—Extended Temperature Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		50	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC} \pm 0.2V$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 10 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, ID Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		6.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		0.85 V_{CC}			V	I_{OH} = −2.5 mA, $V_{CC} = V_{CC}$ Min
V_{OH2}			V_{CC} − 0.4				I_{OH} = −100 μA, $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	500	μA	$A_9 = V_{ID}$

DC CHARACTERISTICS—CMOS COMPATIBLE—Extended Temperature Products (Continued)

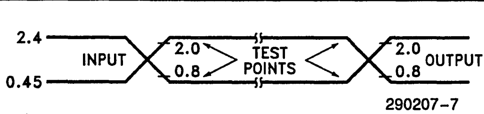
Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical(4)	Max		
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Programs are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

CAPACITANCE $T_A = 25°C$, $f = 1.0$ MHz

Symbol	Parameter	Notes	Limits		Unit	Conditions
			Min	Max		
C_{IN}	Address/Control Capacitance	3		8	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	3		12	pF	$V_{OUT} = 0V$

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at $V_{CC} = 5.0V$, $V_{PP} = 12.0V$, $T = 25°C$. These currents are valid for all product versions (packages and speeds).
2. Not 100% tested: characterization data available.
3. Sampled, not 100% tested.
4. "Typicals" are not guaranteed, but based on a limited number of samples from production lots.

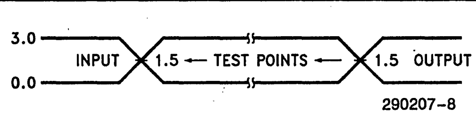
AC TESTING INPUT/OUTPUT WAVEFORM
For 28F010-120, 28F010-150



290207-7

AC test inputs are driven at V_{OH} (2.4 V_{TTL}) for a Logic "1" and V_{OL} (0.45 V_{TTL}) for a Logic "0". Input timing begins at V_{IH} (2.0 V_{TTL}) and V_{IL} (0.8 V_{TTL}). Output timing ends at V_{IH} and V_{IL}. Input rise and fall times (10% to 90%) <10 ns.

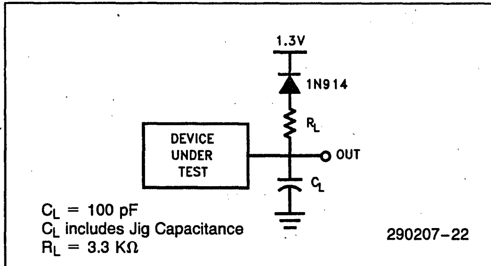
AC TESTING INPUT/OUTPUT WAVEFORM
For 28F010-90V05



290207-8

AC test inputs are driven at 3.0V for a Logic "1" and 0.0V for a Logic "0". Input timing begins, and output timing ends, at 1.5V. Input rise and fall times (10% to 90%) <10 ns.

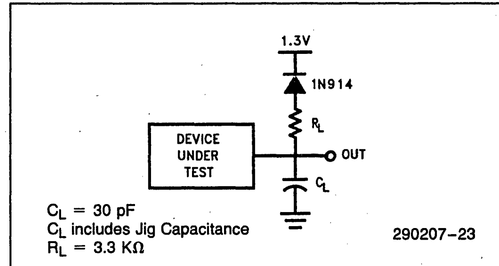
AC TESTING LOAD CIRCUIT
For 28F010-120, 28F010-150



C_L = 100 pF
C_L includes Jig Capacitance
R_L = 3.3 KΩ

290207-22

AC TESTING LOAD CIRCUIT
For 28F010-90V05



C_L = 30 pF
C_L includes Jig Capacitance
R_L = 3.3 KΩ

290207-23

AC TEST CONDITIONS
For 28F010-120, 28F010-150

Parameter	Value
Input Rise and Fall Times (10% to 90%)	10 ns
Input Pulse Levels	0.45V and 2.4V
Input Timing Reference Level	0.8V and 2.0V
Output Timing Reference Level	0.8V and 2.0V
Capacitive Load	100 pF
V_{CC} Supply Tolerance	10%

AC TEST CONDITIONS
For 28F010-90V05

Parameter	Value
Input Rise and Fall Times (10% to 90%)	10 ns
Input Pulse Levels	0.0V and 3.0V
Input Timing Reference Level	1.5V
Output Timing Reference Level	1.5V
Capacitive Load	30 pF
V_{CC} Supply Tolerance	5%

AC CHARACTERISTICS—Read-Only Operations—Commercial and Extended Temperature Products

Versions		Notes	28F010-90V05(3,4)		28F010-120		28F010-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	Min	Max	
t_{AVAV}/t_{RC}	Read Cycle Time		90		120		150		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time			90		120		150	ns
t_{AVQV}/t_{ACC}	Address Access Time			90		120		150	ns
t_{GLQV}/t_{OE}	Output Enable Access Time			40		50		55	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	2, 3	0		0		0		ns
t_{EHQZ}	Chip Disable to Output in High Z	2		55		55		55	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	2, 3	0		0		0		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	2		30		30		35	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	1, 2	0		0		0		ns
t_{WHGL}	Write Recovery Time before Read		6		6		6		μs

NOTES:
1. Whichever occurs first.
2. Sampled, not 100% tested.
3. Guaranteed by design.
4. See AC Test Conditions.

V_{CC} POWER-UP | STANDBY | DEVICE AND ADDRESS SELECTION | OUTPUTS ENABLED | DATA VALID | STANDBY | V_{CC} POWER-DOWN

ADDRESSES — ADDRESS STABLE

t_{AVAV} (t_{RC})

$\overline{CE}$ ($\overline{E}$)

t_{EHQZ}

$\overline{OE}$ ($\overline{G}$)

t_{GHQZ} (t_{DF})

t_{WHGL}

$\overline{WE}$ ($\overline{W}$)

t_{GLQV} (t_{OE})

t_{ELQV} (t_{CE})

t_{GLQX} (t_{OLZ})

t_{ELQX} (t_{LZ})

t_{OH}

DATA (DO) — HIGH Z — VALID OUTPUT — HIGH Z

t_{AVQV} (t_{ACC})

5.0V

V_{CC}

0V

290207-9

Figure 7. AC Waveforms for Read Operations

AC CHARACTERISTICS—Write/Erase/Program Operations(1)— Commercial and Extended Temperature Products

Versions		Notes	28F010-90V05(2,4)		28F010-120(4)		28F010-150(4)		Unit
Symbol	Characteristic		Min	Max	Min	Max	Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		90		120		150		ns
t_{AVWL}/t_{AS}	Address Set-Up Time		0		0		0		ns
t_{WLAX}/t_{AH}	Address Hold Time		50		60		60		ns
t_{DVWH}/t_{DS}	Data Set-Up Time		50		50		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		10		10		ns
t_{WHGL}	Write Recovery Time before Read		6		6		6		µs
t_{GHWL}	Read Recovery Time before Write	2	0		0		0		µs
t_{ELWL}/t_{CS}	Chip Enable Set-Up Time before Write		20		20		20		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time		0		0		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width2		50		60		60		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		20		20		ns
t_{WHWH1}	Duration of Programming Operation	3	10		10		10		µs
t_{WHWH2}	Duration of Erase Operation	3	9.5		9.5		9.5		ms
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low	2	1.0		1.0		1.0		µs

NOTES:
1. Read timing characteristics during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Guaranteed by design.
3. The integrated stop timer terminates the programming/erase operations, thereby eliminating the need for a maximum specification.
4. See AC Test Conditions.

ERASE AND PROGRAMMING PERFORMANCE— Commercial and Extended Temperature Products

Parameter	Notes	Limits			Unit
		Min	Typ	Max	
Chip Erase Time	1, 3, 4		1.0	10	Sec
Chip Program Time	1, 2, 4		2	12.5	Sec
Erase/Program Cycles	1, 5	10,000	100,000		Cycles
	6	1,000			

NOTES:
1. "Typicals" are not guaranteed, but based on a limited number of samples from production lots. Data taken at 25°C, 12.0V V_{PP}, at 0 cycles.
2. Minimum byte programming time excluding system overhead is 16 µsec (10 µsec program + 6 µsec write recovery), while maximum is 400 µsec/byte (16 µsec x 25 loops allowed by algorithm). Max chip programming time is specified lower than the worst case allowed by the programming algorithm since most bytes program significantly faster than the worst case byte.
3. Excludes 00H Programming prior to Erasure.
4. Excludes System-Level Overhead.
5. Refer to RR-60 "ETOX II Flash Memory Reliability Data Summary" for typical cycling data and failure rate calculations.
6. Extended Temperature Products.

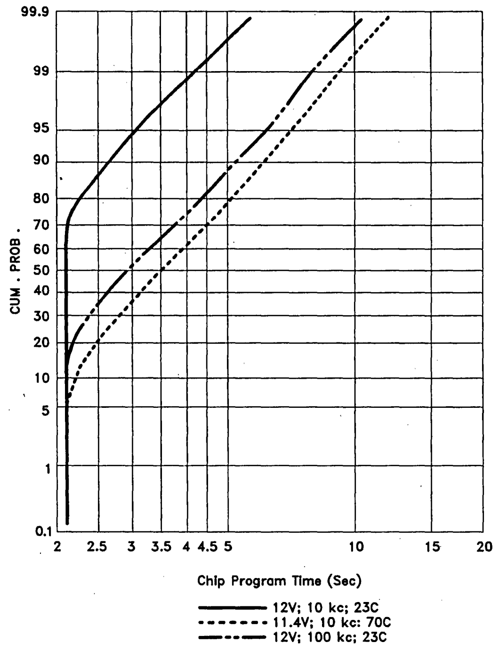


Figure 8. 28F010 Typical Programming Capability

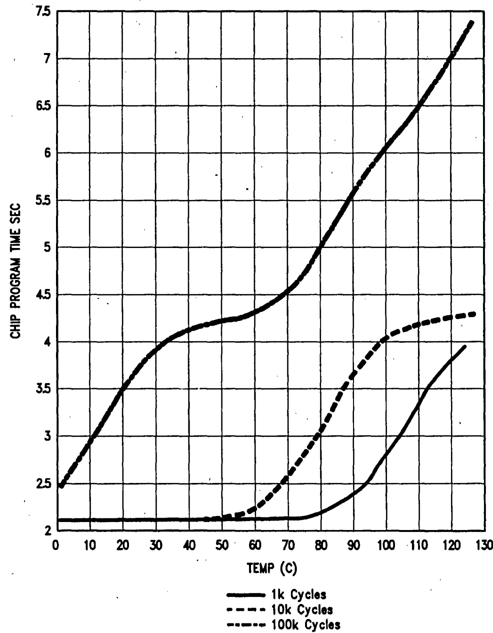


Figure 9. 28F010 Typical Program Time at 12V

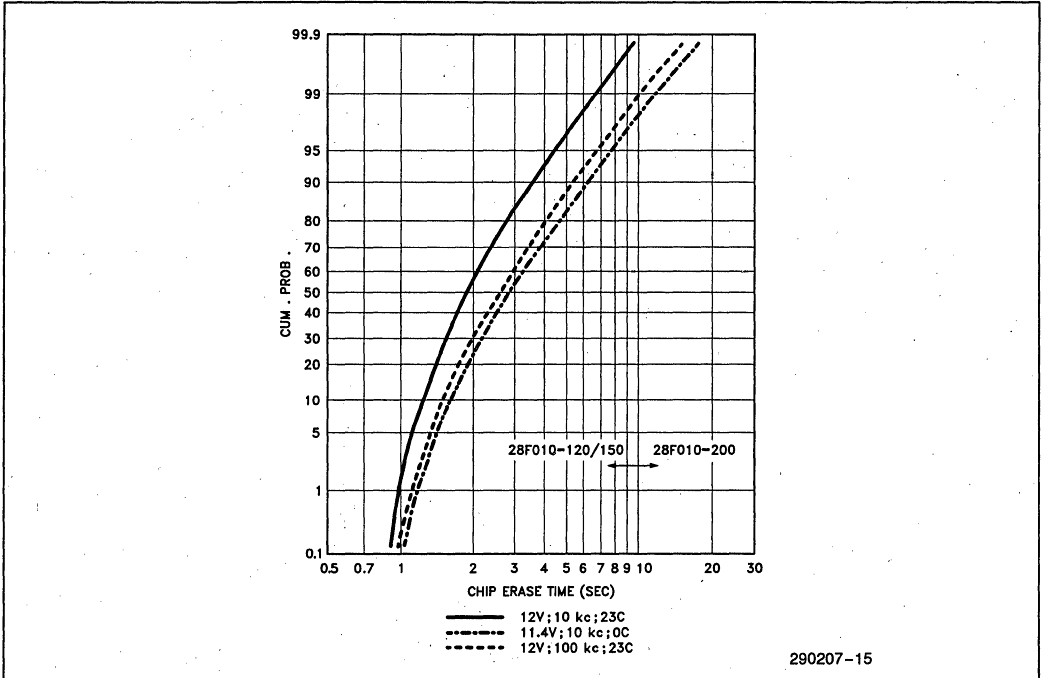


Figure 10. 28F010 Typical Erase Capability

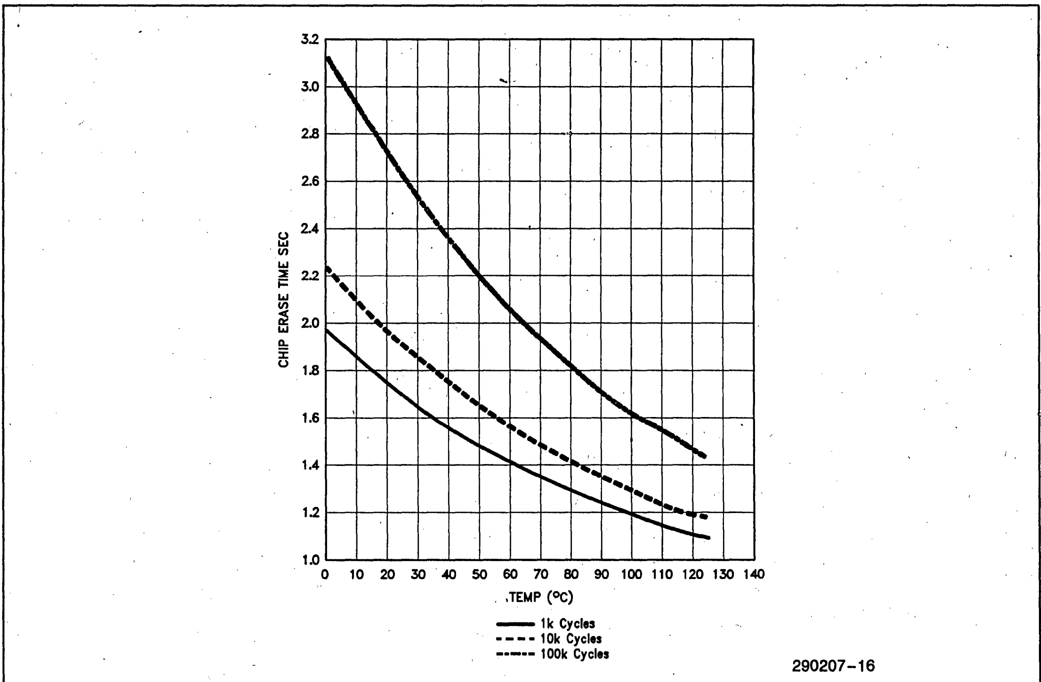


Figure 11. 28F010 Typical Erase Time at 12V

VCC POWER-UP & STANDBY
SET-UP PROGRAM COMMAND
PROGRAM COMMAND LATCH ADDRESS & DATA
PROGRAMMING
PROGRAM VERIFY COMMAND
PROGRAM VERIFICATION
STANDBY/ VCC POWER-DOWN

ADDRESSES
t_{AVAV} (t_{WC})
t_{AVAV} (t_{WC})
t_{AVWL}
t_{WLAX} (t_{AH})
t_{AVAV} (t_{WC})
t_{AVWL}
t_{WLAX} (t_{AH})
t_{AVAV} (t_{RC})

$\overline{CE}$ ($\overline{E}$)
t_{ELWL}
t_{WHEH} (t_{CH})
t_{ELWL} (t_{CS})
t_{WHEH} (t_{CH})
t_{ELWL} (t_{CS})
t_{WHEH} (t_{CH})
t_{EHQZ}

$\overline{OE}$ ($\overline{G}$)
t_{GHWL}
t_{WHWL}
t_{WHWH1}
t_{WHGL}
t_{GHQZ} (t_{DF})

$\overline{WE}$ ($\overline{W}$)
t_{WLWH}
t_{WHDX}
t_{DVWH}
t_{WLWH}
t_{WHDX}
t_{DVWH}
t_{WLWH}
t_{WHDX} (t_{DH})
t_{DVWH}
t_{GLQV} (t_{OE})
t_{OH}
t_{GLQX} (t_{OLZ})

DATA (DQ)
HIGH Z
DATA IN = 40H
DATA IN
DATA IN = C0H
VALID DATA OUT
t_{ELQX} (t_{LZ})
t_{ELQV} (t_{CE})

5.0V
V_{CC}
0V
t_{VPEL}
12.0V
V_{PP}
V_{PPL}

290207-10

Figure 12. AC Waveforms for Programming Operations

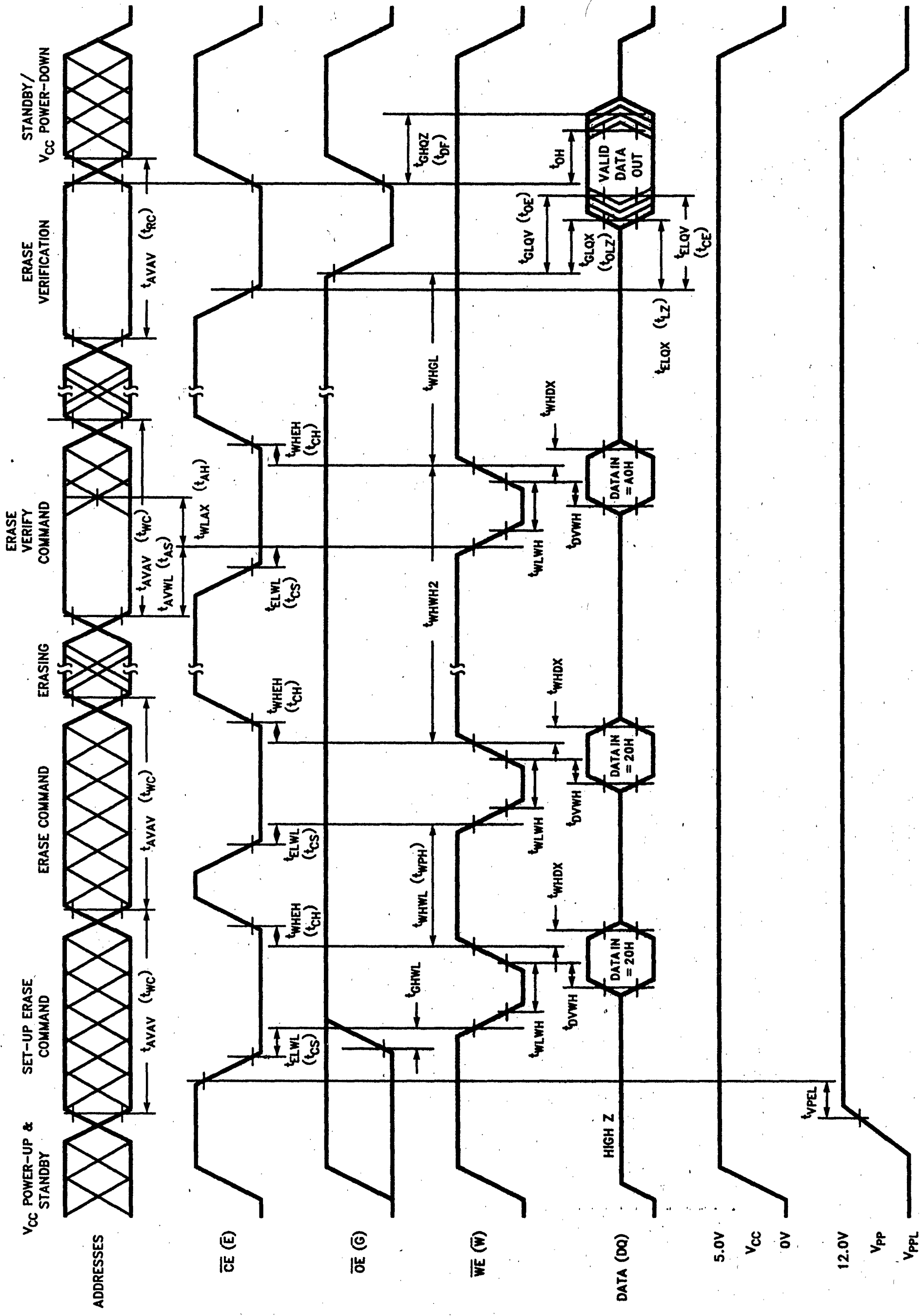


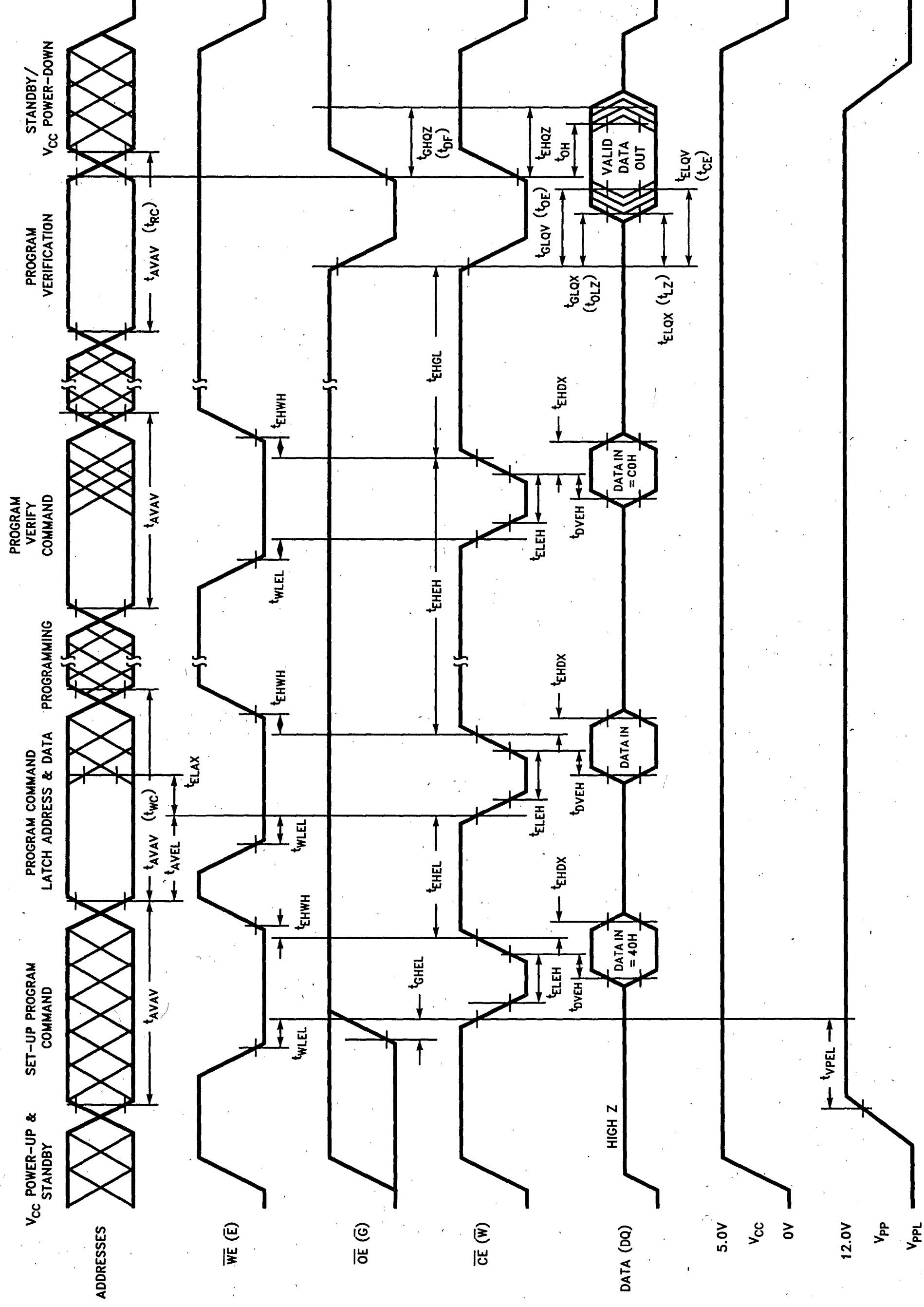
Figure 13. AC Waveforms for Erase Operations

ALTERNATIVE $\overline{CE}$-CONTROLLED WRITES

Versions		Notes	28F010-90V05(2)		28F010-120		28F010-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	Min	Max	
t_{AVAV}	Write Cycle Time		90		120		150		ns
t_{AVEL}	Address Set-Up Time		0		0		0		ns
t_{ELAX}	Address Hold Time		80		80		80		ns
t_{DVEH}	Data Set-Up Time		50		50		50		ns
t_{EHDX}	Data Hold Time		10		10		10		ns
t_{EHGL}	Write Recovery Time before Read		6		6		6		µs
t_{GHEL}	Read Recovery Time before Write	2	0		0		0		µs
t_{WLEL}	Write Enable Set-Up Time before Chip Enable		0		0		0		ns
t_{EHWH}	Write Enable Hold Time		0		0		0		ns
t_{ELEH}	Write Pulse Width	1	70		70		70		ns
t_{EHEL}	Write Pulse Width High		20		20		20		ns
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low	2	1.0		1.0		1.0		µs

NOTE:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of Chip-Enable and Write-Enable. In systems where Chip-Enable defines the write pulse width (within a longer Write-Enable timing waveform) all set-up, hold and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.
2. Guaranteed by design.

3



NOTE:
Alternative $\overline{CE}$-Controlled Write Timings also apply to erase operations.

Figure 14. Alternate AC Waveforms for Programming Operations

ORDERING INFORMATION

T	P	2	8	F	0	1	0	-	1	2	0

TEMPERATURE
T = EXTENDED (−40°C to +85°C)
BLANK = COMMERCIAL (0°C to +70°C)

PACKAGE
P = 32-PIN PLASTIC DIP
N = 32-LEAD PLCC
E = STANDARD 32-LEAD TSOP
F = REVERSE 32-LEAD TSOP

ACCESS SPEED (ns)
120 ns
150 ns

290207–20

VALID COMBINATIONS:

P28F010-120	N28F010-120	TN28F010-120
P28F010-150	N28F010-150	
N28F010-90V05		
E28F010-120	F28F010-120	TE28F010-120
E28F010-150	F28F010-150	TF28F010-120

ADDITIONAL INFORMATION

		Order Number
ER-20,	"ETOX™ II Flash Memory Technology"	294005
ER-24,	"Intel Flash Memory"	294008
RR-60,	"ETOX™ II Flash Memory Reliability Data Summary"	293002
AP-316,	"Using Flash Memory for In-System Reprogrammable Nonvolatile Storage"	292046
AP-325	"Guide to Flash Memory Reprogramming"	292059

REVISION HISTORY

Number	Description
007	Removed **200 ns** Speed Bin Revised Erase Maximum Pulse Count for Figure 5 from **3000** to **1000** Clarified AC and DC Test Conditions Added "dimple" to F TSOP Package Corrected Serpentine Layout
008	Corrected AC Waveforms Added Extended Temperature Options

28F020
2048K (256K x 8) CMOS FLASH MEMORY

- **Flash Electrical Chip-Erase**
  — 2 Second Typical Chip-Erase
- **Quick-Pulse Programming™ Algorithm**
  — 10 μs Typical Byte-Program
  — 4 Second Chip-Program
- **100,000 Erase/Program Cycles Typical**
- **12.0V ±5% V_{PP}**
- **High-Performance Read**
  — 80 ns Maximum Access Time
- **CMOS Low Power Consumption**
  — 10 mA Typical Active Current
  — 50 μA Typical Standby Current
  — 0 Watts Data Retention Power
- **Integrated Program/Erase Stop Timer**
- **Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface**
- **Noise Immunity Features**
  — ±10% V_{CC} Tolerance
  — Maximum Latch-Up Immunity through EPI Processing
- **ETOX™ Nonvolatile Flash Technology**
  — EPROM-Compatible Process Base
  — High-Volume Manufacturing Experience
- **JEDEC-Standard Pinouts**
  — 32-Pin Plastic Dip
  — 32-Lead PLCC
  — 32-Lead TSOP
  (See Packaging Spec., Order #231369)
- **Extended Temperature Options**

Intel's 28F020 CMOS flash memory offers the most cost-effective and reliable alternative for read/write random access nonvolatile memory. The 28F020 adds electrical chip-erasure and reprogramming to familiar EPROM technology. Memory contents can be rewritten: in a test socket; in a PROM-programmer socket; on-board during subassembly test; in-system during final test; and in-system after-sale. The 28F020 increases memory flexibility, while contributing to time- and cost-savings.

The 28F020 is a 2048-kilobit nonvolatile memory organized as 262,144 bytes of 8 bits. Intel's 28F020 is offered in 32-pin plastic DIP, 32-lead PLCC, and 32-lead TSOP packages. Pin assignments conform to JEDEC standards for byte-wide EPROMs.

Extended erase and program cycling capability is designed into Intel's ETOX™ (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional EEPROMs. With the 12.0V V_{PP} supply, the 28F020 performs a minimum of 10,000 erase and program cycles well within the time limits of the Quick-Pulse Programming and Quick-Erase algorithms.

Intel's 28F020 employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 80 nanosecond access time provides no-WAIT-state performance for a wide range of microprocessors and microcontrollers. Maximum standby current of 100 μA translates into power savings when the device is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from $-1V$ to $V_{CC} + 1V$.

With Intel's ETOX process base, the 28F020 levers years of EPROM experience to yield the highest levels of quality, reliability, and cost-effectiveness.

November 1992
Order Number: 290245-005

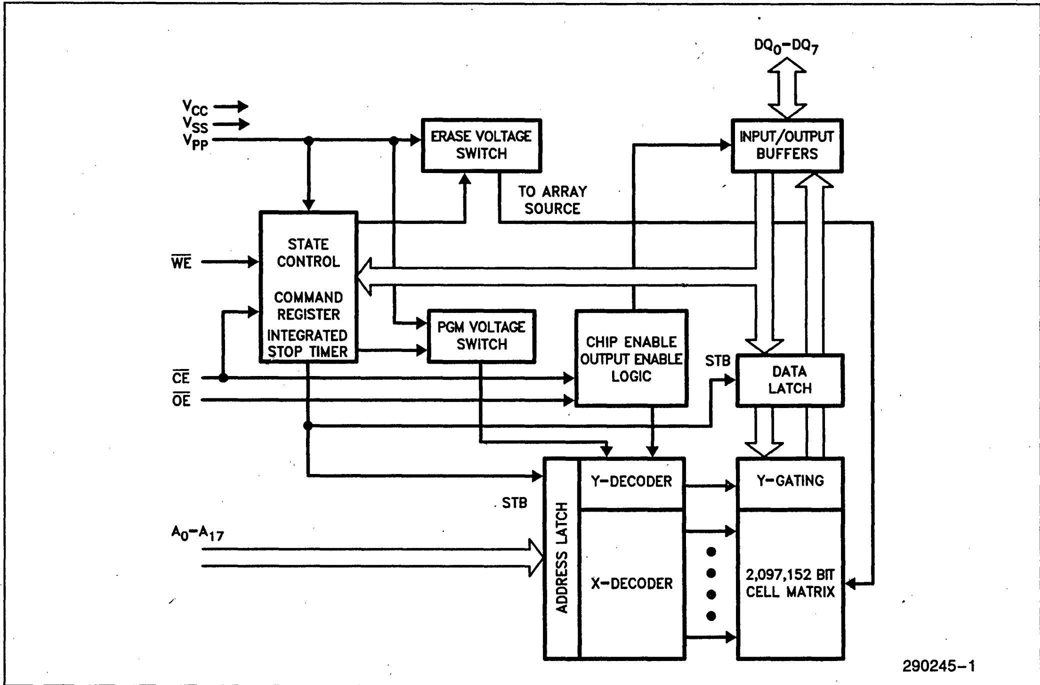
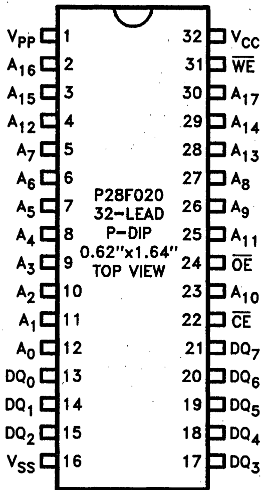


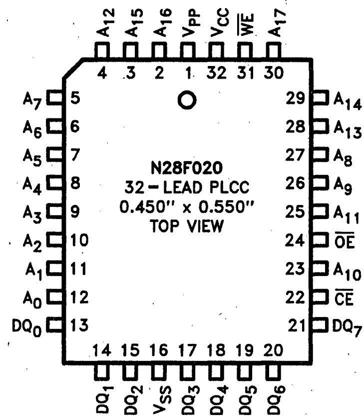
Figure 1. 28F020 Block Diagram

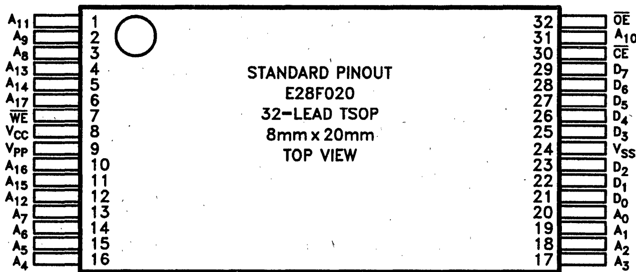
Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{17}	INPUT	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
DQ_0–DQ_7	INPUT/OUTPUT	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}$	INPUT	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels.
$\overline{OE}$	INPUT	**OUTPUT ENABLE:** Gates the devices output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	INPUT	**WRITE ENABLE:** Controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **Note:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP}		**ERASE/PROGRAM POWER SUPPLY** for writing the command register, erasing the entire array, or programming bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY** (5V ±10%)
V_{SS}		**GROUND**

3







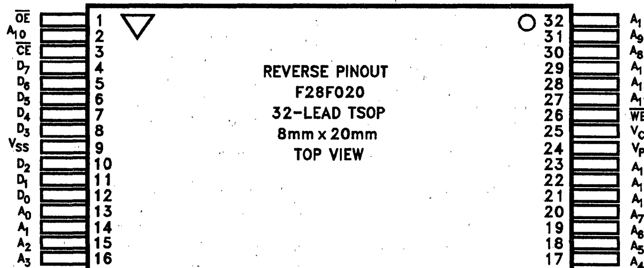


Figure 2. 28F020 Pin Configurations

APPLICATIONS

The 28F020 flash memory provides nonvolatility along with the capability to typically perform over 100,000 electrical chip-erasure/reprogram cycles. These features make the 28F020 an innovative alternative to disk, EEPROM, and battery-backed static RAM. Where periodic updates of code and datatables are required, the 28F020's reprogrammability and nonvolatility make it the obvious and ideal replacement for EPROM.

Primary applications and operating systems stored in flash eliminate the slow disk-to-DRAM download process. This results in dramatic enhancement of performance and substantial reduction of power consumption — a consideration particularly important in portable equipment. Flash memory increases flexibility with electrical chip erasure and in-system update capability of operating systems and application code. With updatable code, system manufacturers can easily accommodate last-minute changes as revisions are made.

In diskless workstations and terminals, network traffic reduces to a minimum and systems are instant-on. Reliability exceeds that of electromechanical media. Often in these environments, power interruptions force extended re-boot periods for all networked terminals. This mishap is no longer an issue if boot code, operating systems, communication protocols and primary applications are flash-resident in each terminal.

For embedded systems that rely on dynamic RAM/disk for main system memory or nonvolatile backup storage, the 28F020 flash memory offers a solid state alternative in a minimal form factor. The 28F020 provides higher performance, lower power consumption, instant-on capability, and allows an "execute in place" memory hierarchy for code and data table reading. Additionally, the flash memory is more rugged and reliable in harsh environments where extreme temperatures and shock can cause disk-based systems to fail.

The need for code updates pervades all phases of a system's life — from prototyping to system manufacture to after-sale service. The electrical chip-erasure and reprogramming ability of the 28F020 allows in-circuit alterability; this eliminates unnecessary handling and less-reliable socketed connections, while adding greater test, manufacture, and update flexibility.

Material and labor costs associated with code changes increases at higher levels of system integration — the most costly being code updates after sale. Code "bugs", or the desire to augment system functionality, prompt after-sale code updates. Field revisions to EPROM-based code requires the removal of EPROM components or entire boards. With the 28F020, code updates are implemented locally via an edge-connector, or remotely over a communcations link.

For systems currently using a high-density static RAM/battery configuration for data accumulation, flash memory's inherent nonvolatility eliminates the need for battery backup. The concern for battery failure no longer exists, an important consideration for portable equipment and medical instruments, both requiring continuous performance. In addition, flash memory offers a considerable cost advantage over static RAM.

Flash memory's electrical chip erasure, byte programmability and complete nonvolatility fit well with data accumulation and recording needs. Electrical chip-erasure gives the designer a "blank slate" in which to log or record data. Data can be periodically off-loaded for analysis and the flash memory erased producing a new "blank slate".

A high degree of on-chip feature integration simplifies memory-to-processor interfacing. Figure 4 depicts two 28F020s tied to the 80C186 system bus. The 28F020's architecture minimizes interface circuitry needed for complete in-circuit updates of memory contents.

The outstanding feature of the TSOP (Thin Small Outline Package) is the 1.2 mm thickness. With standard and reverse pin configurations, TSOP reduces the number of board layers and overall volume necessary to layout multiple 28F020s. TSOP is particularly suited for portable equipment and applications requiring large amounts of flash memory. Figure 3 illustrates the TSOP Serpentine layout.

With cost-effective in-system reprogramming, extended cycling capability, and true nonvolatility, the 28F020 offers advantages to the alternatives: EPROMs, EEPROMs, battery backed static RAM, or disk. EPROM-compatible read specifications, straight-forward interfacing, and in-circuit alterability offers designers unlimited flexibility to meet the high standards of today's designs.

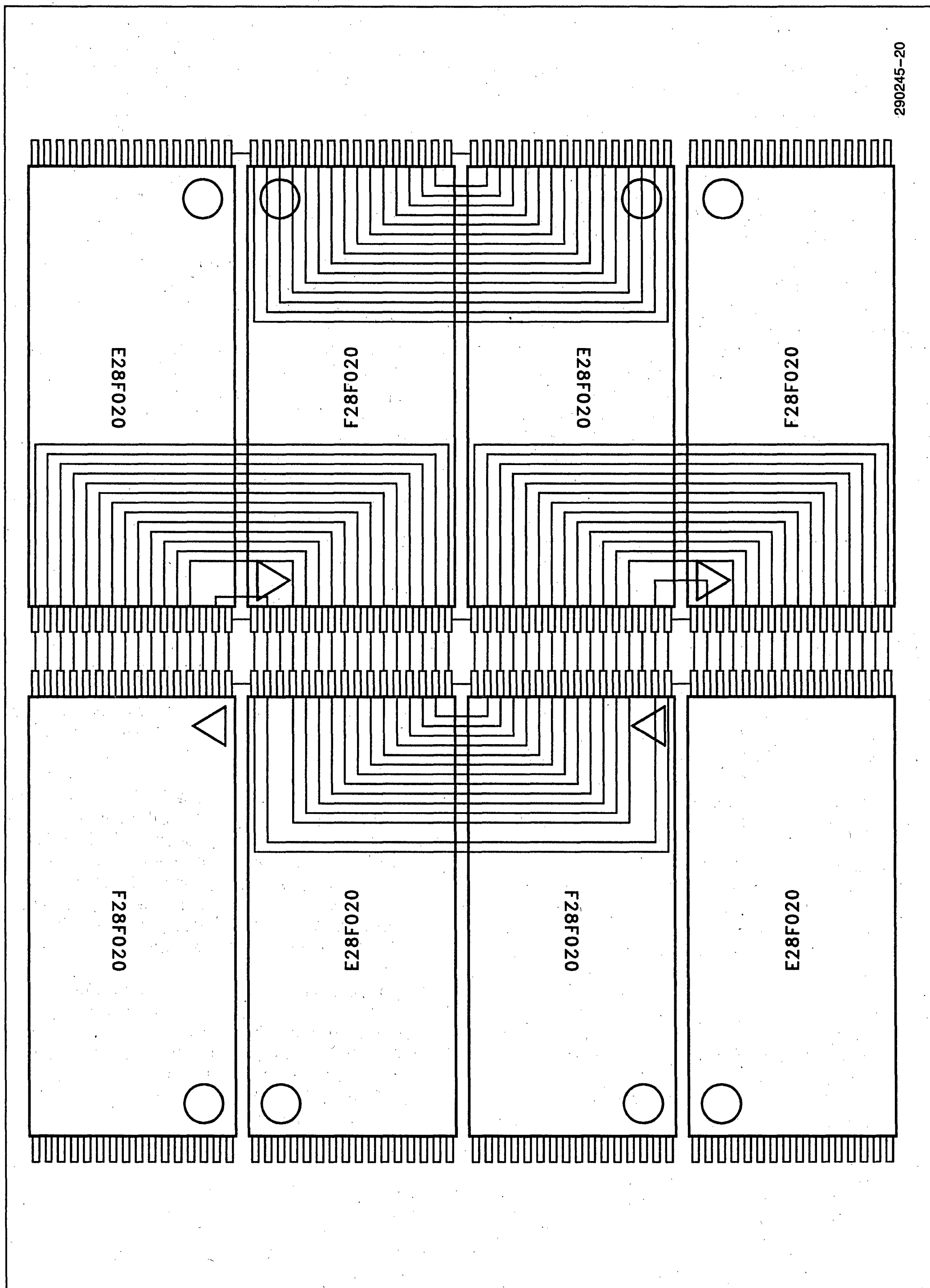


Figure 3. TSOP Serpentine Layout

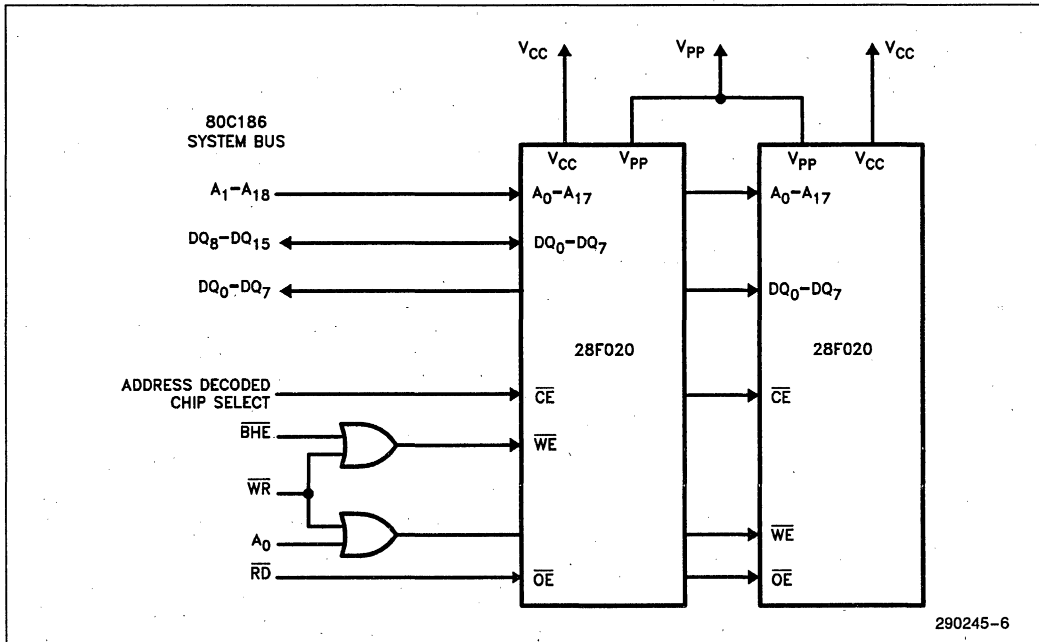


Figure 4. 28F020 in a 80C186 System

PRINCIPLES OF OPERATION

Flash-memory augments EPROM functionality with in-circuit electrical erasure and reprogramming. The 28F020 introduces a command register to manage this new functionality. The command register allows for: 100% TTL-level control inputs; fixed power supplies during erasure and programming; and maximum EPROM compatibility.

In the absence of high voltage on the V_{PP} pin, the 28F020 is a read-only memory. Manipulation of the external memory-control pins yields the standard EPROM read, standby, output disable, and Intelligent Identifier operations.

The same EPROM read, standby, and output disable operations are available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} enables erasure and programming of the device. All functions associated with altering memory contents—Intelligent Identifier, erase, erase verify, program, and program verify—are accessed via the command register.

Commands are written to the register using standard microprocessor write timings. Register contents serve as input to an internal state-machine which controls the erase and programming circuitry. Write cycles also internally latch addresses and data needed for programming or erase operations. With the appropriate command written to the register, standard microprocessor read timings output array data, access the Intelligent Identifier codes, or output data for erase and program verification.

Integrated Stop Timer

Successive command write cycles define the durations of program and erase operations; specifically, the program or erase time durations are normally terminated by associated program or erase verify commands. An integrated stop timer provides simplified timing control over these operations; thus eliminating the need for maximum program/erase timing specifications. Programming and erase pulse durations are minimums only. When the stop timer terminates a program or erase operation, the device enters an inactive state and remains inactive until receiving the appropriate verify or reset command.

Write Protection

The command register is only active when V_{PP} is at high voltage. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable—available only when memory updates are desired. When $V_{PP} = V_{PPL}$, the contents of the register default to the read command, making the 28F020 a read-only memory. In this mode, the memory contents cannot be altered.

Table 2. 28F020 Bus Operations

Operation	Pins	V_{PP}(1)	A_0	A_9	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_0–DQ_7
READ-ONLY	Read	V_{PPL}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out
	Output Disable	V_{PPL}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby	V_{PPL}	X	X	V_{IH}	X	X	Tri-State
	Intelligent Identifier (Mfr)(2)	V_{PPL}	V_{IL}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = 89H
	Intelligent Identifier (Device)(2)	V_{PPL}	V_{IH}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = BDH
READ/WRITE	Read	V_{PPH}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out(4)
	Output Disable	V_{PPH}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby(5)	V_{PPH}	X	X	V_{IH}	X	X	Tri-State
	Write	V_{PPH}	A_0	A_9	V_{IL}	V_{IH}	V_{IL}	Data In(6)

NOTES:
1. Refer to DC Characteristics. When $V_{PP} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes may also be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
3. V_{ID} is the Intelligent Identifier high voltage. Refer to DC Characteristics.
4. Read operations with $V_{PP} = V_{PPH}$ may access array data or the Intelligent Identifier codes.
5. With V_{PP} at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
6. Refer to Table 3 for valid Data-In during a write operation.
7. X can be V_{IL} or V_{IH}.

Or, the system designer may choose to "hardwire" V_{PP}, making the high voltage supply constantly available. In this case, all Command Register functions are inhibited whenever V_{CC} is below the write lockout voltage V_{LKO}. (See Power Up/Down Protection.) The 28F020 is designed to accommodate either design practice, and to encourage optimization of the processor-memory interface.

The two step program/erase write sequence to the Command Register provides additional software write protection.

BUS OPERATIONS

Read

The 28F020 has two control functions, both of which must be logically active, to obtain data at the outputs. Chip-Enable ($\overline{CE}$) is the power control and should be used for device selection. Output-Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Refer to AC read timing waveforms.

When V_{PP} is high (V_{PPH}), the read operation can be used to access array data, to output the Intelligent Identifier codes, and to access data for program/erase verification. When V_{PP} is low (V_{PPL}), the read operation can **only** access the array data.

Output Disable

With Output-Enable at a logic-high level (V_{IH}), output from the device is disabled. Output pins are placed in a high-impedance state.

Standby

With Chip-Enable at a logic-high level, the standby operation disables most of the 28F020's circuitry and substantially reduces device power consumption. The outputs are placed in a high-impedance state, independent of the Output-Enable signal. If the 28F020 is deselected during erasure, programming, or program/erase verification, the device draws active current until the operation is terminated.

Intelligent Identifier Operation

The Intelligent Identifier operation outputs the manufacturer code (89H) and device code (BDH). Programming equipment automatically matches the device with its proper erase and programming algorithms.

With Chip-Enable and Output-Enable at a logic low level, raising A9 to high voltage V_{ID} (see DC Characteristics) activates the operation. Data read from locations 0000H and 0001H represent the manufacturer's code and the device code, respectively.

The manufacturer- and device-codes can also be read via the command register, for instances where the 28F020 is erased and reprogrammed in the target system. Following a write of 90H to the command register, a read from address location 0000H outputs the manufacturer code (89H). A read from address 0001H outputs the device code (BDH).

Write

Device erasure and programming are accomplished via the command register, when high voltage is applied to the V_{PP} pin. The contents of the register serve as input to the internal state-machine. The state-machine outputs dictate the function of the device.

The command register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

The command register is written by bringing Write-Enable to a logic-low level (V_{IL}), while Chip-Enable is low. Addresses are latched on the falling edge of Write-Enable, while data is latched on the rising edge of the Write-Enable pulse. Standard microprocessor write timings are used.

Refer to AC Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the V_{PP} pin, the contents of the command register default to 00H, enabling read-only operations.

Placing high voltage on the V_{PP} pin enables read/write operations. Device operations are selected by writing specific data patterns into the command register. Table 3 defines these 28F020 register commands.

Table 3. Command Definitions

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	1	Write	X	00H			
Read Intelligent Identifier Codes(4)	3	Write	X	90H	Read	(4)	(4)
Set-up Erase/Erase(5)	2	Write	X	20H	Write	X	20H
Erase Verify(5)	2	Write	EA	A0H	Read	X	EVD
Set-up Program/Program(6)	2	Write	X	40H	Write	PA	PD
Program Verify(6)	2	Write	X	C0H	Read	X	PVD
Reset(7)	2	Write	X	FFH	Write	X	FFH

NOTES:
1. Bus operations are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   PA = Address of memory location to be programmed.
   Addresses are latched on the falling edge of the Write-Enable pulse.
3. ID = Data read from location IA during device identification (Mfr = 89H, Device = BDH).
   EVD = Data read from location EA during erase verify.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of Write-Enable.
   PVD = Data read from location PA during program verify. PA is latched on the Program command.
4. Following the Read Intelligent ID command, two read operations access manufacturer and device codes.
5. Figure 6 illustrates the Quick-Erase™ Algorithm.
6. Figure 5 illustrates the Quick-Pulse Programming™ Algorithm.
7. The second bus cycle must be followed by the desired command register write.

Read Command

While V_{PP} is high, for erasure and programming, memory contents can be accessed via the read command. The read operation is initiated by writing 00H into the command register. Microprocessor read cycles retrieve array data. The device remains enabled for reads until the command register contents are altered.

The default contents of the register upon V_{PP} power-up is 00H. This default value ensures that no spurious alteration of memory contents occurs during the V_{PP} power transition. Where the V_{PP} supply is hard-wired to the 28F020, the device powers-up and remains enabled for reads until the command-register contents are changed. Refer to the AC Read Characteristics and Waveforms for specific timing parameters.

Intelligent Identifier Command

Flash-memories are intended for use in applications where the local CPU alters memory contents. As such, manufacturer- and device-codes must be accessible while the device resides in the target system. PROM programmers typically access signature codes by raising A9 to a high voltage. However, multiplexing high voltage onto address lines is not a desired system-design practice.

The 28F020 contains an Intelligent Identifier operation to supplement traditional PROM-programming methodology. The operation is initiated by writing 90H into the command register. Following the command write, a read cycle from address 0000H retrieves the manufacturer code of 89H. A read cycle from address 0001H returns the device code of BDH. To terminate the operation, it is necessary to write another valid command into the register.

Set-up Erase/Erase Commands

Set-up Erase is a command-only operation that stages the device for electrical erasure of all bytes in the array. The set-up erase operation is performed by writing 20H to the command register.

To commence chip-erasure, the erase command (20H) must again be written to the register. The erase operation begins with the rising edge of the Write-Enable pulse and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, chip-erasure can only occur when high voltage is applied to the V_{PP} pin. In the absence of this high voltage, memory contents are protected against erasure. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all bytes of the array in parallel. After each erase operation, all bytes must be verified. The erase verify operation is initiated by writing A0H into the command register. The address for the byte to be verified must be supplied as it is latched on the falling edge of the Write-Enable pulse. The register write terminates the erase operation with the rising edge of its Write-Enable pulse.

The 28F020 applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased.

The erase-verify command must be written to the command register prior to each byte verification to latch its address. The process continues for each byte in the array until a byte does not return FFH data, or the last address is accessed.

In the case where the data read is not FFH, another erase operation is performed. (Refer to Set-up Erase/Erase). Verification then resumes from the address of the last-verified byte. Once all bytes in the array have been verified, the erase step is complete. The device can be programmed. At this point, the verify operation is terminated by writing a valid command (e.g. Program Set-up) to the command register. Figure 6, the Quick-Erase™ algorithm, illustrates how commands and bus operations are combined to perform electrical erasure of the 28F020. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Set-up Program/Program Commands

Set-up program is a command-only operation that stages the device for byte programming. Writing 40H into the command register performs the set-up operation.

Once the program set-up operation is performed, the next Write-Enable pulse causes a transition to an active programming operation. Addresses are internally latched on the falling edge of the Write-Enable pulse. Data is internally latched on the rising edge of the Write-Enable pulse. The rising edge of Write-Enable also begins the programming operation. The programming operation terminates with the next rising edge of Write-Enable, used to write the program-verify command. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Program-Verify Command

The 28F020 is programmed on a byte-by-byte basis. Byte programming may occur sequentially or at random. Following each programming operation, the byte just programmed must be verified.

The program-verify operation is initiated by writing C0H into the command register. The register write terminates the programming operation with the rising edge of its Write-Enable pulse. The program-verify operation stages the device for verification of the byte last programmed. No new address information is latched.

The 28F020 applies an internally-generated margin voltage to the byte. A microprocessor read cycle outputs the data. A successful comparison between the programmed byte and true data means that the byte is successfully programmed. Programming then proceeds to the next desired byte location. Figure 5, the 28F020 Quick-Pulse Programming™ algorithm, illustrates how commands are combined with bus operations to perform byte programming. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or program-command sequences. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

EXTENDED ERASE/PROGRAM CYCLING

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled—an expensive solution.

Intel has designed extended cycling capability into its ETOX flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field is one-tenth that of common EEPROMs, minimizing the probability of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 MV/cm lower than EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure—increasing time to wearout by a factor of 100,000,000.

The 28F020 is specified for a minimum of 10,000 program/erase cycles. The device is programmed and erased using Intel's Quick-Pulse Programming and Quick-Erase algorithms. Intel's algorithmic approach uses a series of operations (pulses), along with byte verification, to completely and reliably erase and program the device.

For further information, see Reliability Report RR-60.

QUICK-PULSE PROGRAMMING ALGORITHM

The Quick-Pulse Programming algorithm uses programming operations of 10 μs duration. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The algorithm allows for up to 25 programming operations per byte, although most bytes verify on the first or second operation. The entire sequence of programming and byte verification is performed with V_{PP} at high voltage. Figure 5 illustrates the Quick-Pulse Programming algorithm.

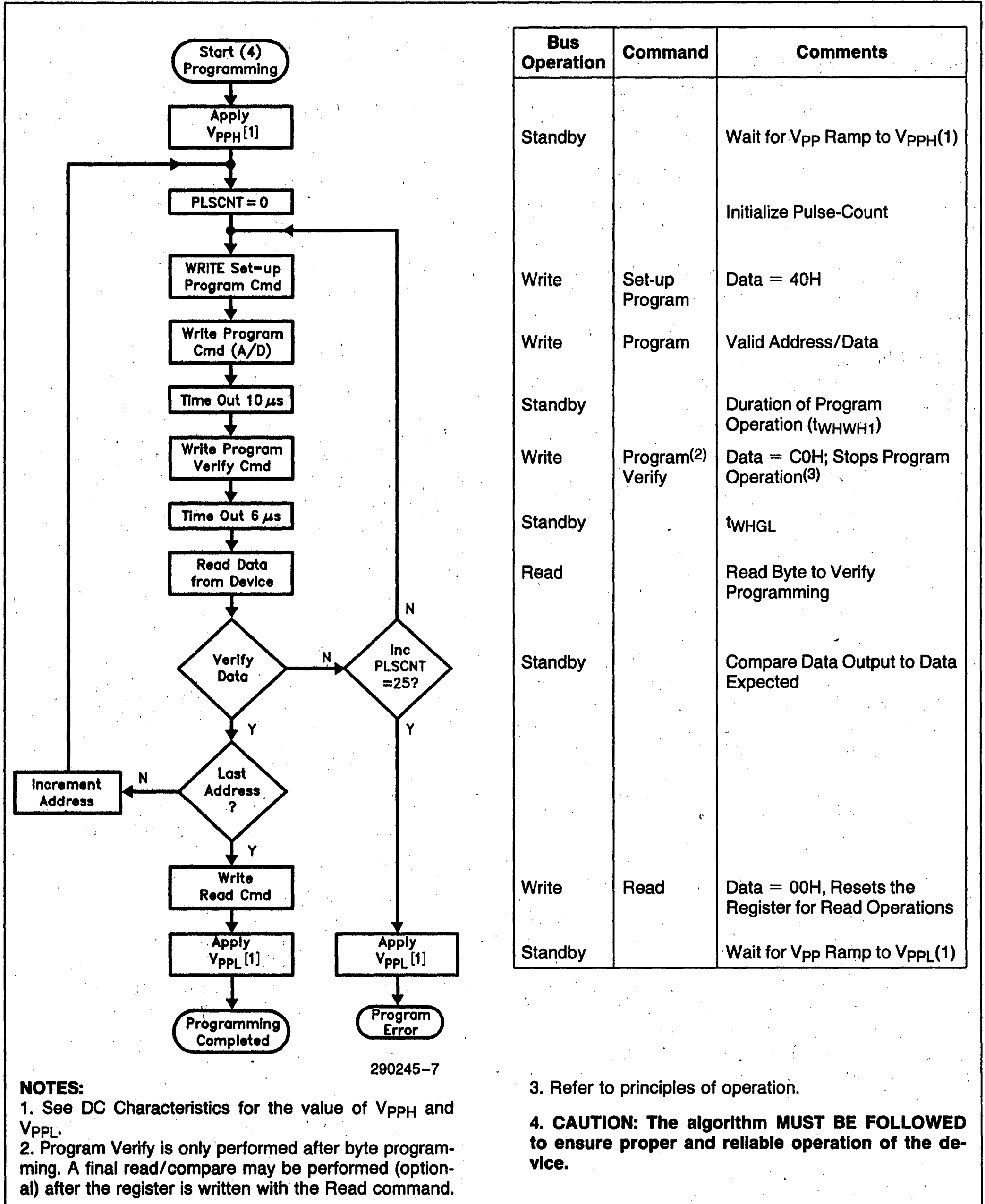
QUICK-ERASE ALGORITHM

Intel's Quick-Erase algorithm yields fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the Quick-Pulse Programming algorithm, to simultaneously remove charge from all bits in the array.

Erasure begins with a read of memory contents. The 28F020 is erased when shipped from the factory. Reading FFH data from the device would immediately be followed by device programming.

For devices being erased and reprogrammed, uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (Data = 00H). This is accomplished, using the Quick-Pulse Programming algorithm, in approximately four seconds.

Erase execution then continues with an initial erase operation. Erase verification (data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. With each erase operation, an increasing number of bytes verify to the erased state. Erase efficiency may be improved by storing the address of the last byte verified in a register. Following the next erase operation, verification starts at that stored address location. Erasure typically occurs in two seconds. Figure 6 illustrates the Quick-Erase algorithm.

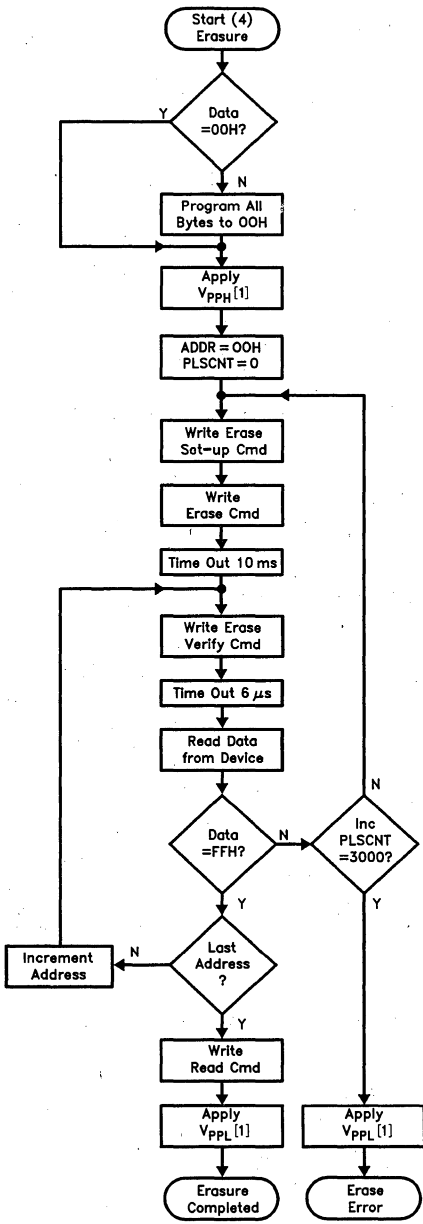


Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Pulse-Count
Write	Set-up Program	Data = 40H
Write	Program	Valid Address/Data
Standby		Duration of Program Operation (t_{WHWH1})
Write	Program(2) Verify	Data = C0H; Stops Program Operation(3)
Standby		t_{WHGL}
Read		Read Byte to Verify Programming
Standby		Compare Data Output to Data Expected
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

NOTES:

1. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
2. Program Verify is only performed after byte programming. A final read/compare may be performed (optional) after the register is written with the Read command.
3. Refer to principles of operation.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

Figure 5. 28F020 Quick-Pulse Programming Algorithm



Bus Operation	Command	Comments
		Entire Memory Must = 00H Before Erasure
		Use Quick-Pulse Programming™ Algorithm (Figure 5)
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Addresses and Pulse-Count
Write	Set-up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase(2) Verify	Addr = Byte to Verify; Data = A0H; Stops Erase Operation(3)
Standby		t_{WHGL}
Read		Read Byte to Verify Erasure
Standby		Compare Output to FFH Increment Pulse-Count
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

1. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
2. Erase Verify is performed only after chip-erasure. A final read/compare may be performed (optional) after the register is written with the read command.
3. Refer to principles of operation.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

Figure 6. 28F020 Quick-Erase Algorithm

DESIGN CONSIDERATIONS

Two-Line Output Control

Flash-memories are often used in larger memory arrays. Intel provides two read-control inputs to accommodate multiple memory connections. Two-line control provides for:

a. the lowest possible memory power dissipation and,
b. complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address-decoder output should drive chip-enable, while the system's read signal controls all flash-memories and other parallel memories. This assures that only enabled memory devices have active outputs, while deselected devices maintain the low power standby condition.

Power Supply Decoupling

Flash-memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current (I_{CC}) issues—standby, active, and transient current peaks produced by falling and rising edges of chip-enable. The capacitive and inductive loads on the device outputs determine the magnitudes of these peaks.

Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 μF ceramic capacitor connected between V_{CC} and V_{SS}, and between V_{PP} and V_{SS}.

Place the high-frequency, low-inherent-inductance capacitors as close as possible to the devices. Also, for every eight devices, a 4.7 μF electrolytic capacitor should be placed at the array's power supply connection, between V_{CC} and V_{SS}. The bulk capacitor will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

V_{PP} Trace on Printed Circuit Boards

Programming flash-memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the V_{PP} power supply trace. The V_{PP} pin supplies the memory cell current for programming. Use similar trace widths and layout considerations given the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots.

Power Up/Down Protection

The 28F020 is designed to offer protection against accidental erasure or programming during power transitions. Upon power-up, the 28F020 is indifferent as to which power supply, V_{PP} or V_{CC}, powers up first. **Power supply sequencing is not required.** Internal circuitry in the 28F020 ensures that the command register is reset to the read mode on power up.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will inhibit writes. The control register architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

28F020 Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases the usable battery life of your system because the 28F020 does not consume any power to retain code or data when the system is off. Table 4 illustrates the power dissipated when updating the 28F020.

Table 4. 28F020 Typical Update Power Dissipation(4)

Operation	Notes	Power Dissipation (Watt-Seconds)
Array Program/Program Verify	1	0.34
Array Erase/Erase Verify	2	0.37
One Complete Cycle	3	1.05

NOTES:
1. Formula to calculate typical Program/Program Verify Power = [V_{PP} × # Bytes × typical # Prog Pulse (t_{WHWH1} × I_{PP2} typical + t_{WHGL} × I_{PP4} typical)] + [V_{CC} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{CC2} typical + t_{WHGL} × I_{CC4} typical)].
2. Formula to calculate typical Erase/Erase Verify Power = [V_{PP} (I_{PP3} typical × t_{ERASE} typical + I_{PP5} typical × t_{WHGL} × # Bytes)] + [V_{CC} (I_{CC3} typical × t_{ERASE} typical + I_{CC5} typical × t_{WHGL} × # Bytes)].
3. One Complete Cycle = Array Preprogram + Array Erase + Program.
4. "Typicals" are not guaranteed but based on a limited number of samples from 28F020-150 production lots.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read ....0°C to +70°C(1)
During Erase/Program ....0°C to +70°C(1)

Operating Temperature
During Read ....−40°C to +85°C(2)
During Erase/Program ......−40°C to +85°C(2)

Temperature Under Bias.......−10°C to +80°C(1)

Temperature Under Bias.......−50°C to +95°C(2)

Storage Temperature ....−65°C to +125°C

Voltage on Any Pin with
Respect to Ground ....−2.0V to +7.0V(2)

Voltage on Pin A_9 with
Respect to Ground ....−2.0V to +13.5V(2, 3)

V_{PP} Supply Voltage with
Respect to Ground
During Erase/Program ....−2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage with
Respect to Ground ....−2.0V to +7.0V(2)

Output Short Circuit Current.............100 mA(4)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Operating temperature is for extended temperature product as defined by this specification.
3. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
4. Maximum DC voltage on A_9 or V_{PP} may overshoot to +14.0V for periods less than 20 ns.
5. Output shorted for no more than one second. No more than one output shorted at a time.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature(1)	0	70	°C	For Read-Only and Read/Write Operations for Commercial products
T_A	Operating Temperature(2)	−40	+85	°C	For Read-Only and Read/Write Operations for Extended Temperature Products
V_{CC}	V_{CC} Supply Voltage (10%)	4.50	5.50	V	
V_{CC}	V_{CC} Supply Voltage (5%)	4.75	5.25	V	For 28F020-80V05 Only

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE—Commercial Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE—Commercial Products (Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ(4)	Max		
I_{CCS}	V_{CC} Standby Current	1		0.3	1.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	15	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, ID Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		10	30	mA	$V_{PP} = V_{PPH}$
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	I_{OH} = −2.5 mA $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	200	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—CMOS COMPATIBLE—Commercial Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		50	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC} \pm 0.2V$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	15	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	15	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, ID Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8	30	mA	$V_{PP} = V_{PPH}$, Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		10	30	mA	$V_{PP} = V_{PPH}$, Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		0.85 V_{CC}			V	I_{OH} = −2.5 mA, $V_{CC} = V_{CC}$ Min
V_{OH2}			V_{CC} − 0.4				I_{OH} = −100 μA, $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	200	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Programs are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE—Extended Temperature Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.3	1.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	30	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, ID Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		10	30	mA	$V_{PP} = V_{PPH}$
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	I_{OH} = −2.5 mA $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	500	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—CMOS COMPATIBLE—Extended Temperature Products

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ(4)	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		50	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC} \pm 0.2V$
I_{CC1}	V_{CC} Active Read Current	1		10	50	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ $f = 6$ MHz, $I_{OUT} = 0$ mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	10	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current, ID Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8	30	mA	$V_{PP} = V_{PPH}$, Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		10	30	mA	$V_{PP} = V_{PPH}$, Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$, Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	$I_{OL} = 5.8$ mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		0.85 V_{CC}			V	$I_{OH} = -2.5$ mA, $V_{CC} = V_{CC}$ Min
V_{OH2}			V_{CC} − 0.4				$I_{OH} = -100$ μA, $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	
I_{ID}	A_9 Intelligent Identifier Current	1, 2		90	500	μA	$A_9 = V_{ID}$
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Programs are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

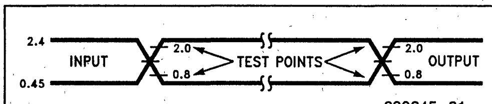
CAPACITANCE $T_A = 25°C, f = 1.0 MHz$

Symbol	Parameter	Notes	Limits		Unit	Conditions
			Min	Max		
C_{IN}	Address/Control Capacitance	3		8	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	3		12	pF	$V_{OUT} = 0V$

NOTES for DC Characteristics and Capacitance:
1. All currents are in RMS unless otherwise noted. Typical values at $V_{CC} = 5.0V$, $V_{PP} = 12.0V$, $T = 25°C$. These currents are valid for all product versions (packages and speeds).
2. Not 100% tested: Characterization data available.
3. Sampled, not 100% tested.
4. "Typicals" are not guaranteed, but based on a limited number of samples from production lots.

TESTING CHARACTERISTICS FOR 28F020-90, 28F020-150, 28F020-200

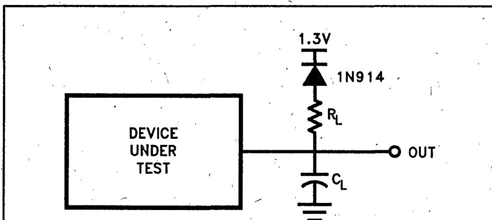
AC TESTING INPUT/OUTPUT WAVEFORM



290245-21

AC test inputs are driven at V_{OH} (2.4 V_{TTL}) for a Logic "1" and V_{OL} (0.45 V_{TTL}) for a Logic "0". Input timing begins at V_{IH} (2.0 V_{TTL}) and V_{IL} (0.8 V_{TTL}). Output timing ends at V_{IH} and V_{IL}. Input rise and fall times (10% to 90%) ≤10 ns.

AC TESTING LOAD CIRCUIT



290245-23

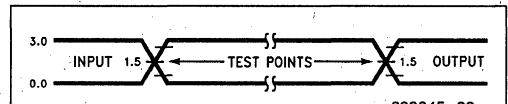
$C_L = 100$ pF
C_L includes Jig Capacitance
$R_L = 3.3$ kΩ

AC TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ...... 10 ns
Input Pulse Levels .... 0.45 and 2.4
Input Timing Reference Level .... 0.8 and 2.0
Output Timing Reference Level .... 0.8 and 2.0
Capacitive Load .... 100 pF
V_{CC} Supply Tolerance 10%

TESTING CHARACTERISTICS FOR 28F020-80V05

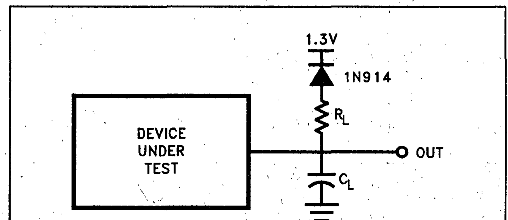
AC TESTING INPUT/OUTPUT WAVEFORM



290245-22

AC test inputs are driven at 3.0V for a Logic "1" and 0.0V for a Logic "0". Input timing begins, and output timing ends, at 1.5V. Input rise and fall times (10% to 90%) ≤10 ns.

AC TESTING LOAD CIRCUIT



290245-24

$C_L = 30$ pF
C_L includes Jig Capacitance
$R_L = 3.3$ kΩ

AC TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ...... 10 ns
Input Pulse Levels .... 0.0 and 3.0
Input Timing Reference Level 1.5
Output Timing Reference Level 1.5
Capacitive Load .... 30 pF
V_{CC} Supply Tolerance 5%

AC CHARACTERISTICS—Read-Only Operations—Commercial Products

Versions				28F020-80V05(4, 5)		28F020-90(4, 5)		28F020-150(4)		28F020-200(4)		Unit
Symbol		Parameter	Notes	Min	Max	Min	Max	Min	Max	Min	Max	
t_{AVAV}	t_{RC}	Read Cycle Time		80		90		150		200		ns
t_{ELQV}	t_{CE}	Chip Enable Access Time			80		90		150		200	ns
t_{AVQV}	t_{ACC}	Address Access Time			80		90		150		200	ns
t_{GLQV}	t_{OE}	Output Enable Access Time			35		40		55		60	ns
t_{ELQX}	t_{LZ}	Chip Enable to Output Low Z	2, 3	0		0		0		0		ns
t_{ELQZ}	t_{HZ}	Chip Disable to Output High Z	2		55		55		55		55	ns
t_{GLQX}	t_{OLZ}	Output Enable to Output Low Z	2, 3	0		0		0		0		ns
t_{GHQZ}	t_{DF}	Output Disable to Output in High Z	2		30		30		35		40	ns
t_{OH}		Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change	1, 2	0		0		0		0		ns
t_{WHGL}		Write Recovery Time before Read		6		6		6		6		µs

NOTES:
1. Whichever occurs first.
2. Sampled, not 100% tested.
3. Guaranteed by design.
4. See AC Test Conditions.
5. Preliminary Information.

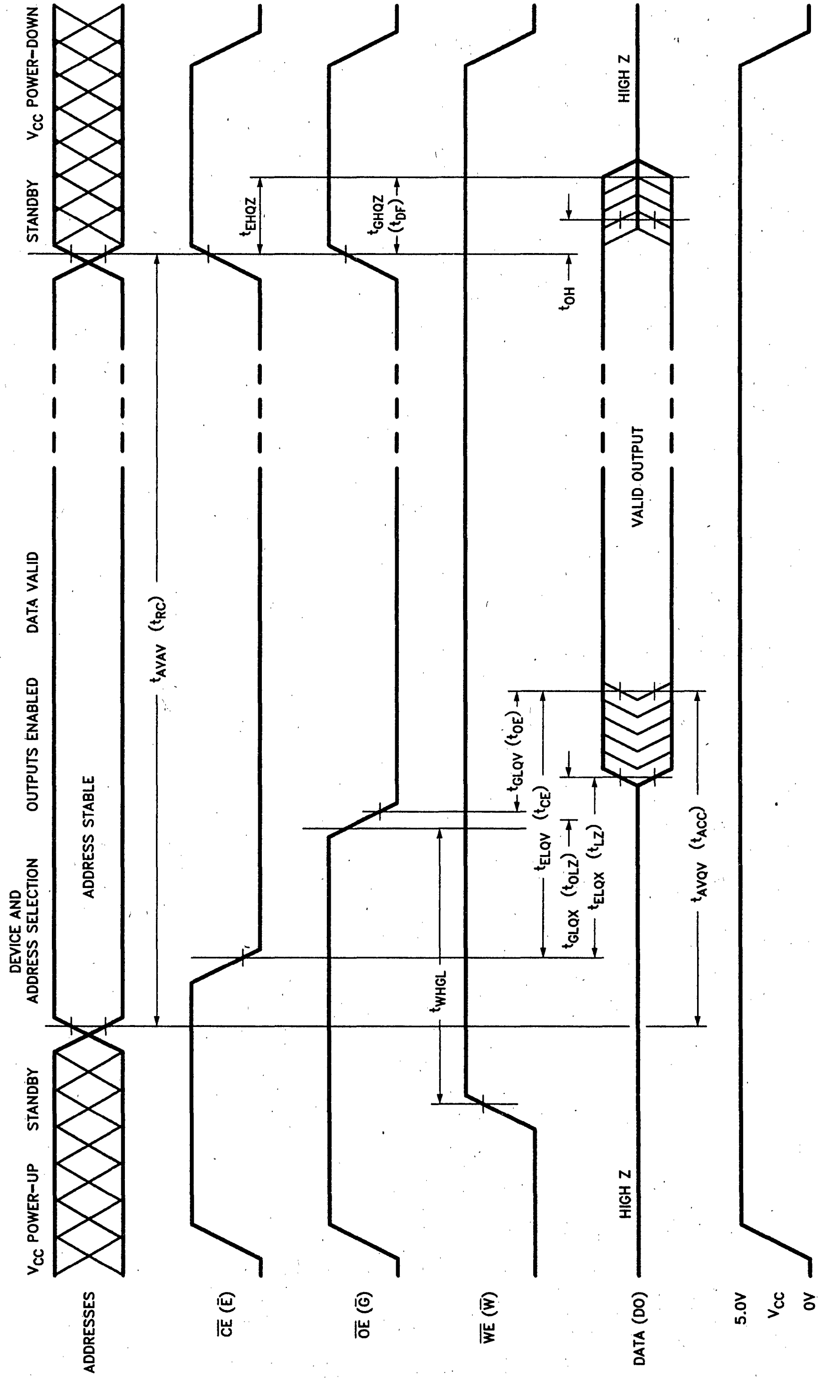


Figure 7. AC Waveforms for Read Operations

AC CHARACTERISTICS—Write/Erase/Program Operations[1]—Commercial Products

Versions		Notes	28F020-80V05[4, 5]		28F020-90[4, 5]		28F020-150[4]		28F020-200[4]		Unit
Symbol	Characteristic		Min	Max	Min	Max	Min	Max	Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		80		90		150		200		ns
t_{AVWL}/t_{AS}	Address Set-Up Time		0		0		0		0		ns
t_{WLAX}/t_{AH}	Address Hold Time		45		45		60		75		ns
t_{DVWH}/t_{DS}	Data Set-Up Time		45		45		50		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		10		10		10		ns
t_{WHGL}	Write Recovery Time before Read		6		6		6		6		μs
t_{GHWL}	Read Recovery Time before Write	2	0		0		0		0		μs
t_{ELWL}/t_{CS}	Chip Enable Set-Up Time before Write		15		20		20		20		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time		0		0		0		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width		45		45		60		60		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		20		20		20		ns
t_{WHWH1}	Duration of Programming Operation	3	10	10			10		10		μs
t_{WHWH2}	Duration of Erase Operation	3	9.5		9.5		9.5		9.5		ms
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low	2	1		1		1.0		1.0		μs

NOTES:
1. Read timing characteristics during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Guaranteed by design.
3. The integrated stop timer terminates the programming/erase operations, thereby eliminating the need for a maximum specification.
4. See AC Test Conditions.
5. Preliminary Information.

ERASE AND PROGRAMMING PERFORMANCE—Commercial Products

Parameter	Notes	Limits			Unit
		Min	Typ	Max	
Chip Erase Time	1, 3, 4		2	30	Sec
Chip Program Time	1, 2, 4		4	25	Sec
Erase/Program Cycles	1, 5	10,000	100,000		Cycles

NOTES:
1. "Typicals" are not guaranteed, but based on a limited number of samples from production lots. Data taken at 25°C, 12.0V V_{PP} at 0 cycles.
2. Minimum byte programming time excluding system overhead is 16 μsec (10 μsec program + 6 μsec write recovery), while maximum is 400 μsec/byte (16 μsec x 25 loops allowed by algorithm). Max chip programming time is specified lower than the worst case allowed by the programming algorithm since most bytes program significantly faster than the worst case byte.
3. Excludes 00H Programming prior to Erasure.
4. Excludes System-Level Overhead.
5. Refer to RR-60, 69 "ETOX Flash Memory Reliability Data Summaries" for typical cycling data and failure rate calculations.

AC CHARACTERISTICS—Read-Only Operations—Extended Temperature Products

Versions		Notes	28F020-90(4)		28F020-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}/t_{RC}	Read Cycle Time		90		150		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time			90		150	ns
t_{AVQV}/t_{ACC}	Address Access Time			90		150	ns
t_{GLQV}/t_{OE}	Output Enable Access Time			55		55	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	2, 3	0		0		ns
t_{EHQZ}	Chip Disable to Output in High Z	2		55		55	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	2, 3	0		0		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	2		35		35	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	1, 2	0		0		ns
t_{WHGL}	Write Recovery Time before Read		6		6		μs

AC CHARACTERISTICS—Write/Erase Program Operation(1)—Extended Temperature

Versions		Notes	28F020-90(4)		28F020-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		90		150		ns
t_{AVWL}/t_{AS}	Address Set-Up Time		0		0		ns
t_{WLAX}/t_{AH}	Address Hold Time		60		60		ns
t_{DVWH}/t_{DS}	Data Set-Up Time		50		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		10		ns
t_{WHGL}	Write Recovery Time before Read		6		6		µs
t_{GHWL}	Read Recovery Time before Write	2	0		0		µs
t_{ELWL}/t_{CS}	Chip Enable Set-Up Time before Write		20		20		ns
t_{WHEH}/t_{CH}	Ehip Enable Hold Time		0		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width		60		60		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		20		ns
t_{WHWH1}	Duration of Programming Operation	3	10		10		µs
t_{WHWH2}	Duration of Erase Operation	3	9.5		9.5		ns
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low	2	10.0		10.0		µs

NOTES:
1. Read timing characteristics during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Guaranteed by design.
3. The integrated stop timer terminates the programming/erase operations, thereby eliminating the need for a maximum specification.
4. Preliminary Information.

ERASE AND PROGRAMMING PERFORMANCE—Extended Temperature

Parameter	Notes	Limits						Unit
		28F020-90(6)			28F020-150			
		Min	Typ	Max	Min	Typ	Max	
Chip Erase Time	1, 3, 4		2	30		2	30	Sec
Chip Program Time	1, 2, 4		4	25		4	25	Sec
Erase/Program Cycle	1, 5	10,000	100,000		1,000			Cycles

NOTES:
1. "Typicals" are not guaranteed, but based on a limited number of samples from production lots. Data taken at 25°C, 12.0V V_{PP}.
2. Minimum byte programming time excluding system overhead is 16 µs (10 µs program + 6 µs write recovery), while maximum is 400 µs/byte (16 µs x 25 loops allowed by algorithm). Max chip programming time is specified lower than the worst case allowed by the programming algorithm, since most bytes program significantly faster than the worst case byte.
3. Excludes 00H Programming prior to Erasure.
4. Excludes System-Level Overhead.
5. Refer to RR-60, "ETOX Flash Memory Reliability Data Summary" for typical cycling data and failure rate calculations.
6. Preliminary Information.

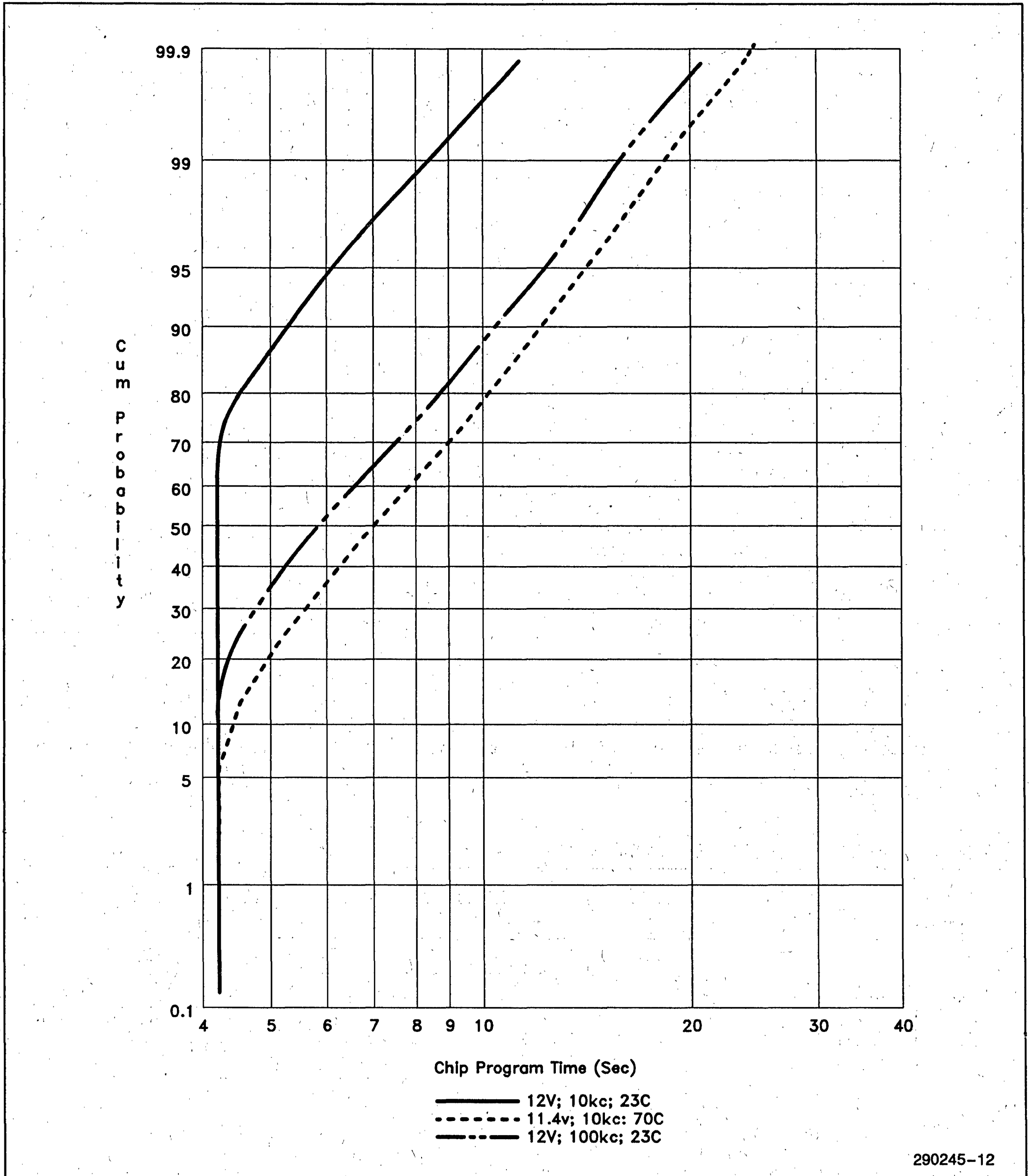


Figure 8. 28F020 Typical Programming Capability (applies to 28F020-150, 28F020-200 only)

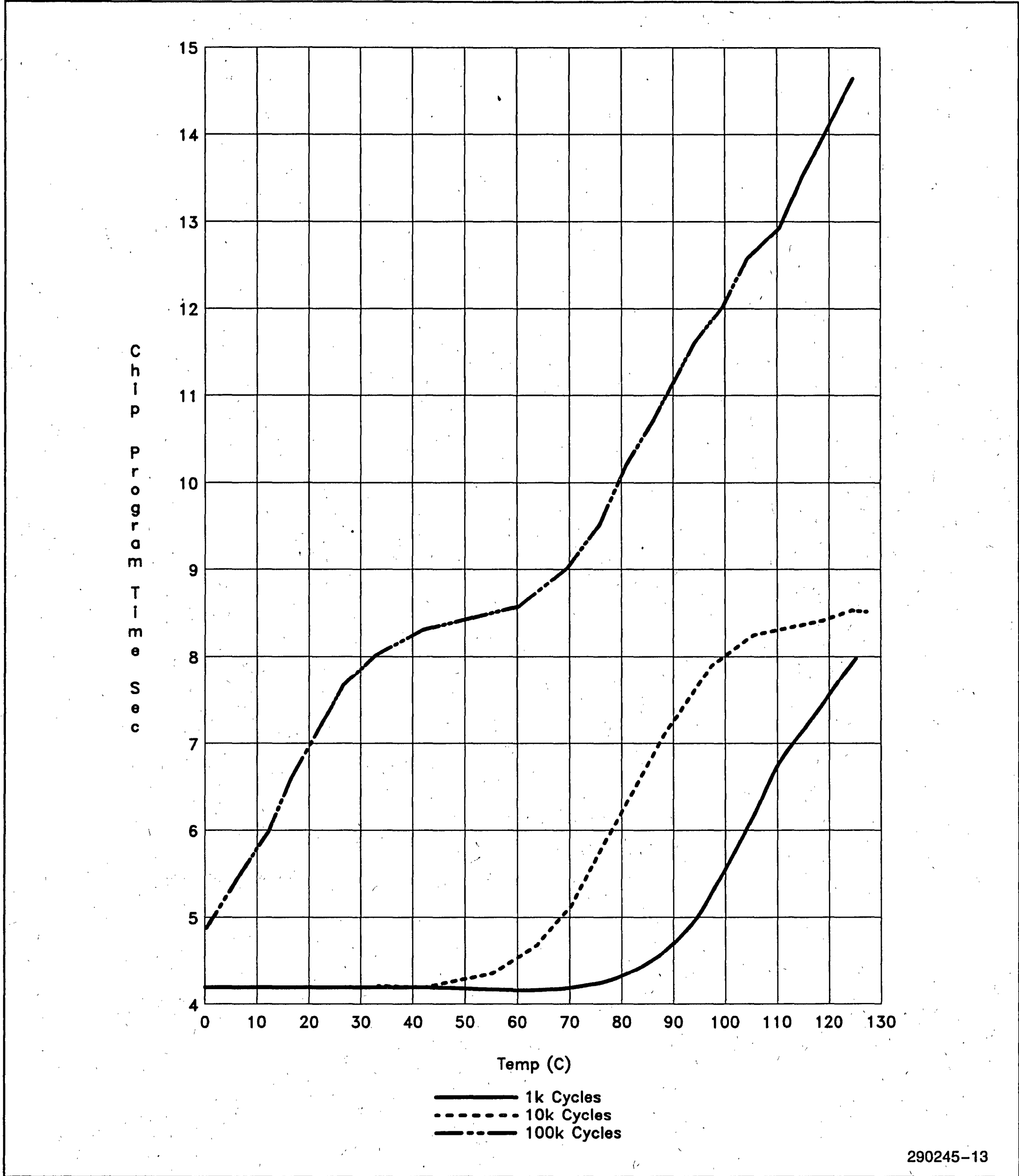
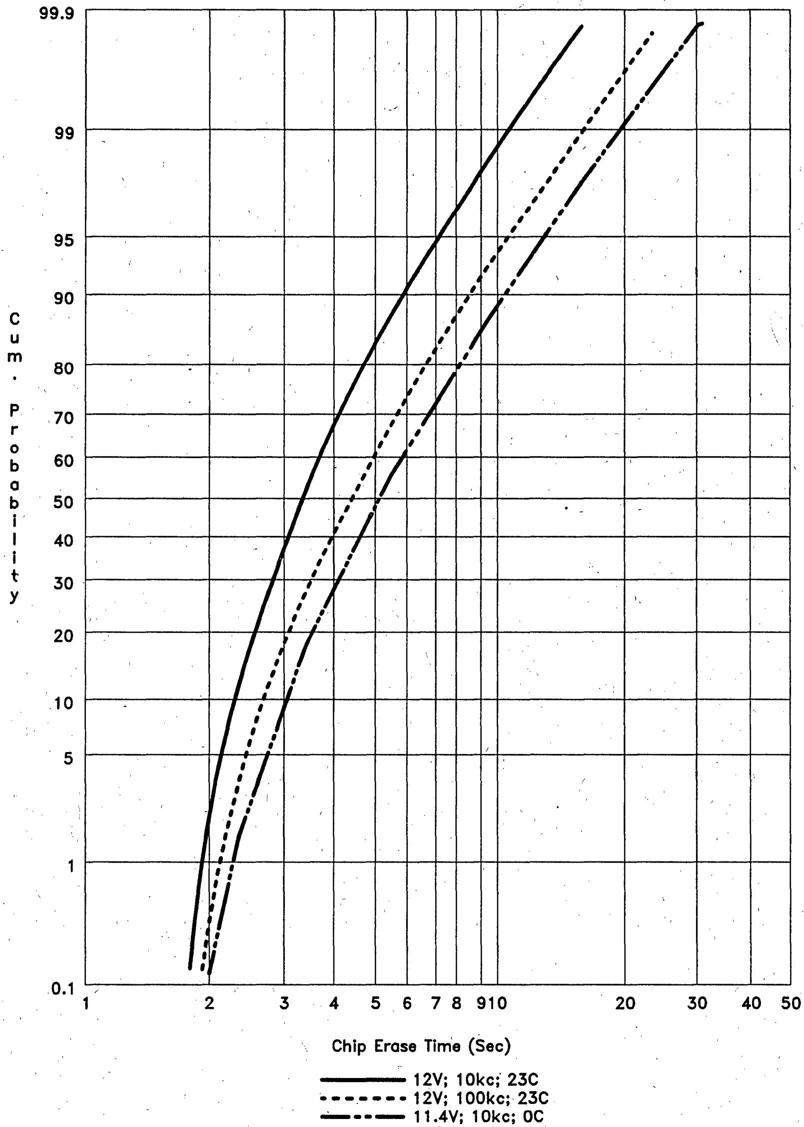


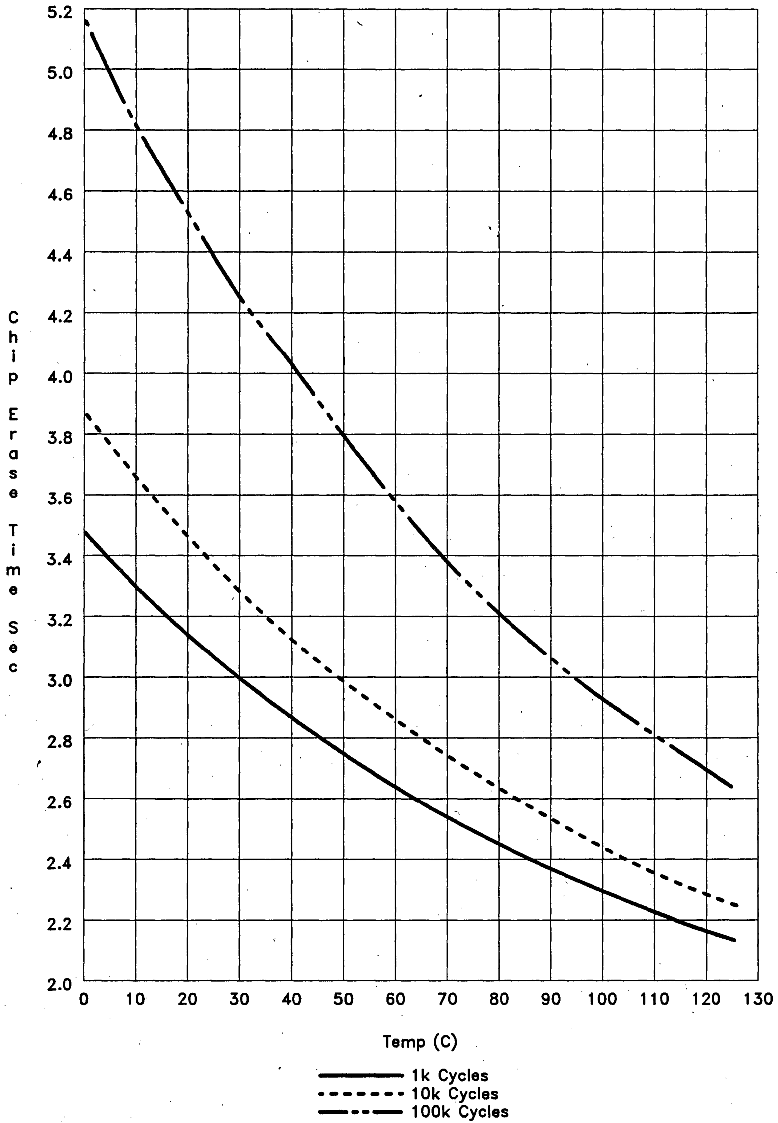
Figure 9. 28F020 Typical Program Time at 12V (applies to 28F020-150, 28F020-200 only)

3



NOTE:
Does not include Pre-Erase Program.

Figure 10. 28F020 Typical Erase Capability (applies to 28F020-150, 28F020-200 only)



NOTE:
Does not include Pre-Erase Program.

Figure 11. 28F020 Typical Erase Time at 12.0V (applies to 28F020-150, 28F020-200 only)

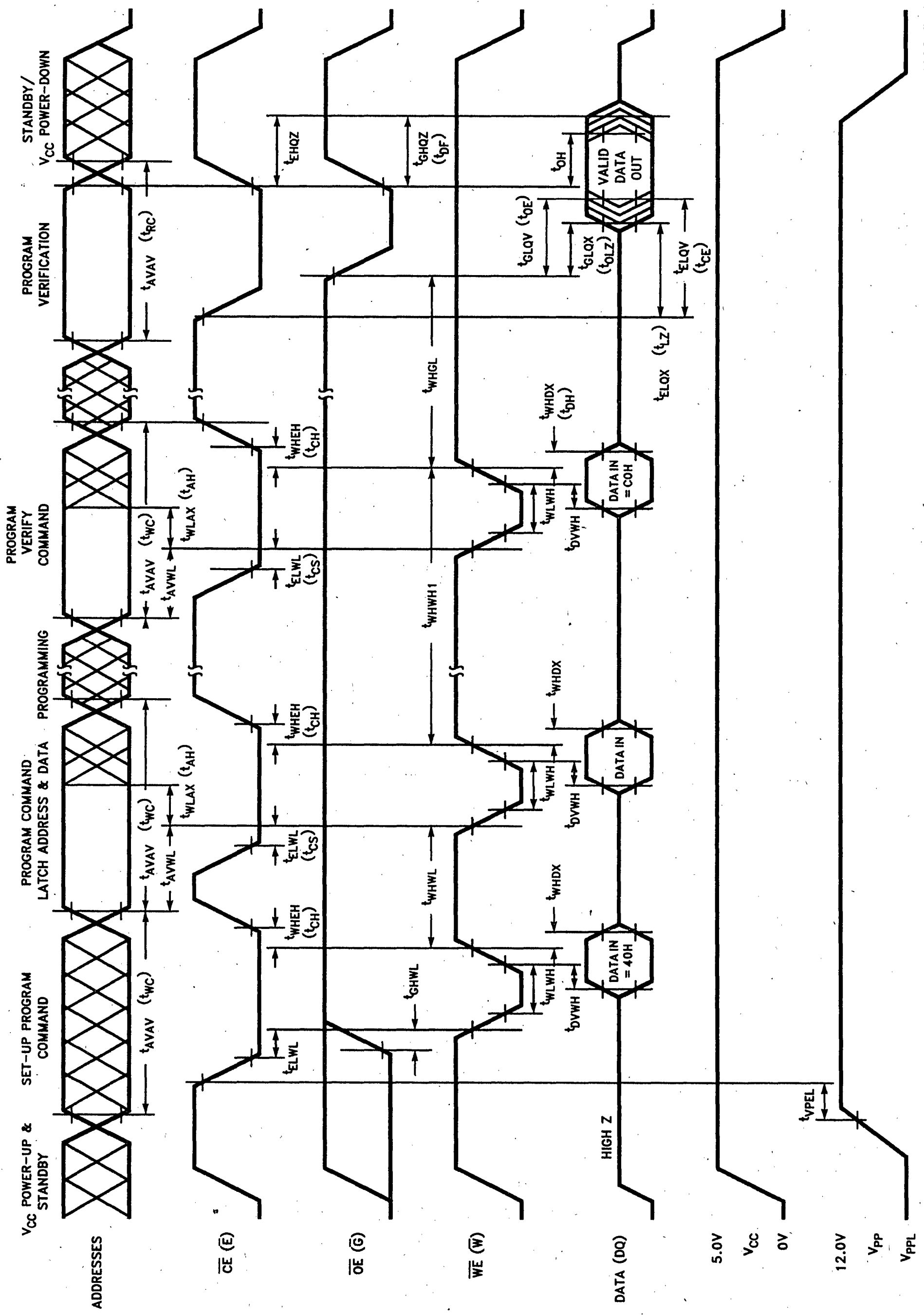


Figure 12. AC Waveforms for Programming Operations

V_{CC} POWER-UP & STANDBY
SET-UP ERASE COMMAND
ERASE COMMAND
ERASING
ERASE VERIFY COMMAND
ERASE VERIFICATION
STANDBY/ V_{CC} POWER-DOWN

ADDRESSES
$\overline{CE}$ ($\overline{E}$)
$\overline{OE}$ ($\overline{G}$)
$\overline{WE}$ ($\overline{W}$)
DATA (DQ)
5.0V
V_{CC}
0V
12.0V
V_{PP}
V_{PPL}

t_{AVAV} (t_{WC})
t_{AVAV} (t_{WC})
t_{AVAV} (t_{WC})
t_{AVWL} (t_{AS})
t_{WLAX} (t_{AH})
t_{AVAV} (t_{RC})
t_{ELWL} (t_{CS})
t_{WHEH} (t_{CH})
t_{GHWL}
t_{WHWL} (t_{WPH})
t_{WHWH2}
t_{WHGL}
t_{GHQZ} (t_{DF})
t_{WLWH}
t_{DVWH}
t_{WHDX}
t_{GLQV} (t_{OE})
t_{OH}
t_{GLQX} (t_{OLZ})
t_{ELQX} (t_{LZ})
t_{ELQV} (t_{CE})
t_{VPEL}
HIGH Z
DATA IN = 20H
DATA IN = 20H
DATA IN = A0H
VALID DATA OUT

290245–17

Figure 13. AC Waveforms for Erase Operations

ALTERNATIVE $\overline{CE}$-CONTROLLED WRITES

Versions		Notes	28F020-80V05(4, 5)		28F020-90(4, 5)		28F020-150		28F020-200		Unit
Symbol	Characteristic		Min	Max	Min	Max	Min	Max	Min	Max	
t_{AVAV}	Write Cycle Time		80		90		150		200		ns
t_{AVEL}	Address Set-Up Time		0		0		0		0		ns
t_{ELAX}	Address Hold Time		60		70		80		95		ns
t_{DVEH}	Data Set-Up Time		50		50		50		50		ns
t_{EHDX}	Data Hold Time		10		10		10		10		ns
t_{EHGL}	Write Recovery Time before Read		6		6		6		6		μs
t_{GHEL}	Read Recovery Time before Write	2	0		0		0		0		μs
t_{WLEL}	Write Enable Set-Up Time before Chip Enable		0		0		0		0		ns
t_{EHWH}	Write Enable Hold Time		0		0		0		0		ns
t_{ELEH}	Write Pulse Width	1	60		60		70		80		ns
t_{EHEH1}	Duration of Programming Operation	3	10		10		10		10		ns
t_{EHEH2}	Duration of Erase Operation	3	9.5		9.5		9.5		9.5		ns
t_{EHEL}	Write Pulse Width High		20		20		20		20		ns
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low	2	1		1		1.0		1.0		μs
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low	6	10.0		10.0		10.0		10.0		μs

NOTE:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of Chip-Enable and Write-Enable. In systems where Chip-Enable defines the write pulse width (within a longer Write-Enable timing waveform) all set-up, hold and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.
2. Guaranteed by design.
3. The integrated stop timer terminates the programming/erase operations, thereby eliminating the need for a maximum specification.
4. See AC Test conditions.
5. Preliminary Information.
6. Extended Temperature Products.

VCC POWER-UP & STANDBY | SET-UP PROGRAM COMMAND | PROGRAM COMMAND LATCH ADDRESS & DATA | PROGRAMMING | PROGRAM VERIFY COMMAND | PROGRAM VERIFICATION | STANDBY/ VCC POWER-DOWN

ADDRESSES

WE (E)

OE (G)

CE (W)

DATA (DQ)

HIGH Z

DATA IN = 40H

DATA IN

DATA IN = C0H

VALID DATA OUT

5.0V

VCC

0V

12.0V

VPP

VPPL

t_{AVAV}, t_{AVAV} (t_{WC}), t_{AVEL}, t_{ELAX}, t_{AVAV} (t_{RC}), t_{WLEL}, t_{EHWH}, t_{GHEL}, t_{EHEL}, t_{EHEH}, t_{EHGL}, t_{GHQZ} (t_{DF}), t_{ELEH}, t_{DVEH}, t_{EHDX}, t_{GLQV} (t_{OE}), t_{GLQX} (t_{OLZ}), t_{EHQZ}, t_{OH}, t_{ELQX} (t_{LZ}), t_{ELQV} (t_{CE}), t_{VPEL}

290245-18

NOTE:
Alternative $\overline{CE}$-Controlled Write Timings also apply to erase operations.

Figure 14. Alternate AC Waveforms for Programming Operations

ORDERING INFORMATION

	E	2	8	F	0	2	0	-	8	0	0	V	0	5

TEMPERATURE RANGE
T = EXTENDED (−40°C TO +55°C)
BLANK = COMMERCIAL (0°C TO +70°C)

PACKAGE
P = 32-PIN PLASTIC DIP
N = 32-LEAD PLCC
E = STANDARD 32-LEAD TSOP
F = REVERSE 32-LEAD TSOP

ACCESS SPEED (ns)
80 ns
90 ns
150 ns
200 ns

PROLIFERATION CODE

290245-19

VALID COMBINATIONS:

P28F020-150	N28F020-150	TN28F020-90	TN28F020-150
P28F020-200	N28F020-200		
E28F020-150	F28F020-150	TE28F020-90	TE28F020-150
E28F020-200	F28F020-200	TF28F020-90	TF28F020-150

ADDITIONAL INFORMATION

		Order Number
ER-20,	"ETOX™ Flash Memory Technology"	294005
ER-24,	"Intel Flash Memory"	294008
ER-28,	"ETOX™ III Flash Memory Technology"	294012
RR-60,	"ETOX™ Flash Memory Reliability Data Summary"	293002
AP-316,	"Using Flash Memory for In-System Reprogrammable Nonvolatile Storage"	292046
AP-325	"Guide to Flash Memory Reprogramming"	292059

REVISION HISTORY

Number	Description
-004	Removed **Preliminary** Classification. Clarified AC and DC test conditions. Added "dimple" to F TSOP package. Corrected serpentine layout.
-005	Added -80V05, -90 ns speed grades. Added extended temperature devices. Corrected AC Waveforms.

28F001BX-T/28F001BX-B
1M (128K x 8) CMOS FLASH MEMORY

- High Integration Blocked Architecture
  - One 8KB Boot Block w/Lock Out
  - Two 4KB Parameter Blocks
  - One 112KB Main Block
- 10,000 Erase/Program Cycles Minimum Per Block
- Simplified Program and Erase
  - Automated Algorithms via On-Chip Write State Machine (WSM)
- SRAM-Compatible Write Interface
- Deep-Powerdown Mode
  - 0.05 μA I_{CC} Typical
  - 0.8 μA I_{PP} Typical
- 12.0V ±5% V_{PP}
- High-Performance Read
  - 120 ns Maximum Access Time
  - 5.0V ±10% V_{CC}
- Hardware Data Protection Feature
  - Erase/Write Lockout during Power Transitions
- Advanced Packaging, JEDEC Pinouts
  - 32-Pin PDIP
  - 32-Lead PLCC, TSOP
- ETOX™ II Nonvolatile Flash Technology
  - EPROM-Compatible Process Base
  - High-Volume Manufacturing Experience
- Extended Temperature Options

Intel's 28F001BX-B and 28F001BX-T combine the cost-effectiveness of Intel standard Flash Memory with features that simplify write and allow block erase. These devices aid the system designer by combining the functions of several components into one, making boot block flash an innovative alternative to EPROM and EEPROM or battery-backed static RAM. Many new and existing designs can take advantage of the 28F001BX's integration of blocked architecture, automated electrical reprogramming, and standard processor interface.

The 28F001BX-B and 28F001BX-T are 1,048,576-bit nonvolatile memories organized as 131,072 bytes of 8 bits. They are offered in 32-pin plastic DIP, 32-lead PLCC and 32-lead TSOP packages. Pin assignment conform to JEDEC standards for byte-wide EPROMs. These devices use an integrated command port and state machine for simplified block erasure and byte reprogramming. The 28F001BX-T's block locations provide compatibility with microprocessors and microcontrollers that boot from high memory, such as Intel's MCS-186 family, 80286, i386™, i486™, i860™ and 80960CA. With exactly the same memory segmentation, the 28F001BX-B memory map is tailored for microprocessors and microcontrollers that boot from low memory, such as Intel's MCS-51, MCS-196, 80960KX and 80960SX families. All other features are identical, and unless otherwise noted, the term 28F001BX can refer to either device throughout the remainder of this document.

The boot block section includes a reprogramming write lock out feature to guarantee data integrity. It is designed to contain secure code which will bring up the system minimally and download code to the other locations of the 28F001BX. Intel's 28F001BX employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 120 ns access time provides no-WAIT-state performance for a wide range of microprocessors and microcontrollers. A deep-powerdown mode lowers power consumption to 0.25 μW typical through V_{CC}, crucial in laptop computer, handheld instrumentation and other low-power applications. The $\overline{PWD}$ power control input also provides absolute data protection during system powerup or power loss.

Manufactured on Intel's 1-micron ETOX II process base, the 28F001BX builds on years of EPROM experience to yield the highest levels of quality, reliability, and cost-effectiveness.

September 1992
Order Number: 290406-004

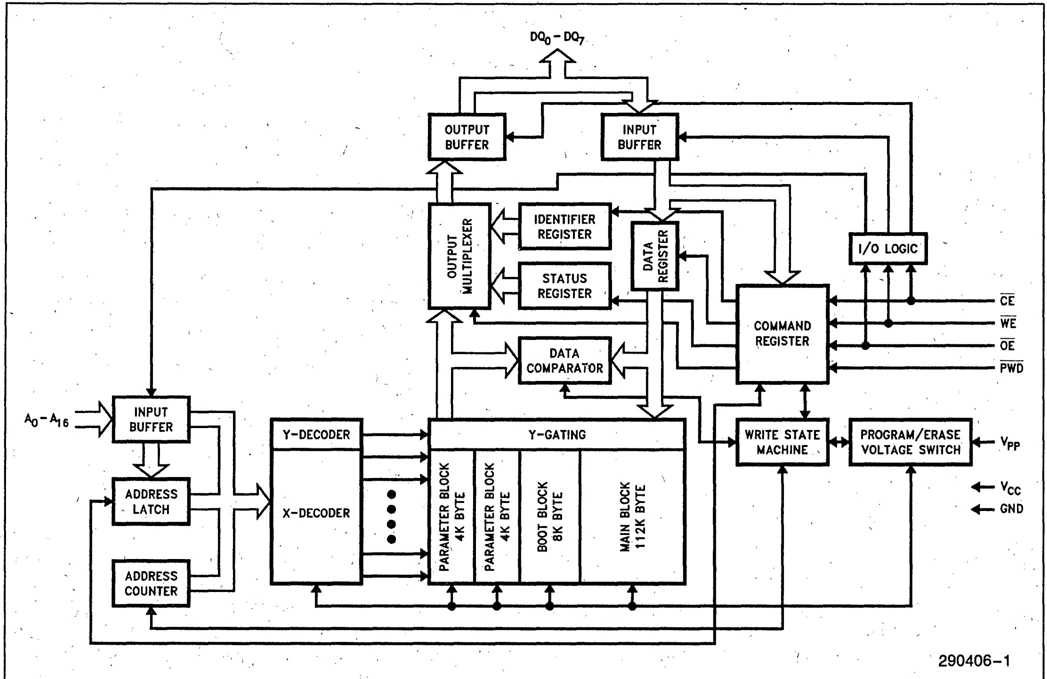


Figure 1. 28F001BX Block Diagram

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{16}	INPUT	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
DQ_0–DQ_7	INPUT/ OUTPUT	**DATA INPUTS/OUTPUTS:** Inputs data and commands during memory write cycles; outputs data during memory, Status Register and Identifier read cycles. The data pins are active high and float to tri-state off when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}$	INPUT	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels.
$\overline{PWD}$	INPUT	**POWERDOWN:** Puts the device in deep powerdown mode. $\overline{PWD}$ is active low; $\overline{PWD}$ high gates normal operation. $\overline{PWD}$ = V_{HH} allows programming of the boot block. $\overline{PWD}$ also locks out erase or write operations when active low, providing data protection during power transitions.
$\overline{OE}$	INPUT	**OUTPUT ENABLE:** Gates the device's outputs through the data buffers during a read cycle. $\overline{OE}$ is active low. $\overline{OE}$ = V_{HH} (pulsed) allows programming of the boot block.
$\overline{WE}$	INPUT	**WRITE ENABLE:** Controls writes to the Command Register and array blocks. $\overline{WE}$ is active low. Addresses and data are latched on the rising edge of the $\overline{WE}$ pulse.
V_{PP}		**ERASE/PROGRAM POWER SUPPLY** for erasing blocks of the array or programming bytes of each block. Note: With V_{PP} < V_{PPL} max, memory contents cannot be altered.
V_{CC}		**DEVICE POWER SUPPLY:** (5V ±10%)
GND		**GROUND**

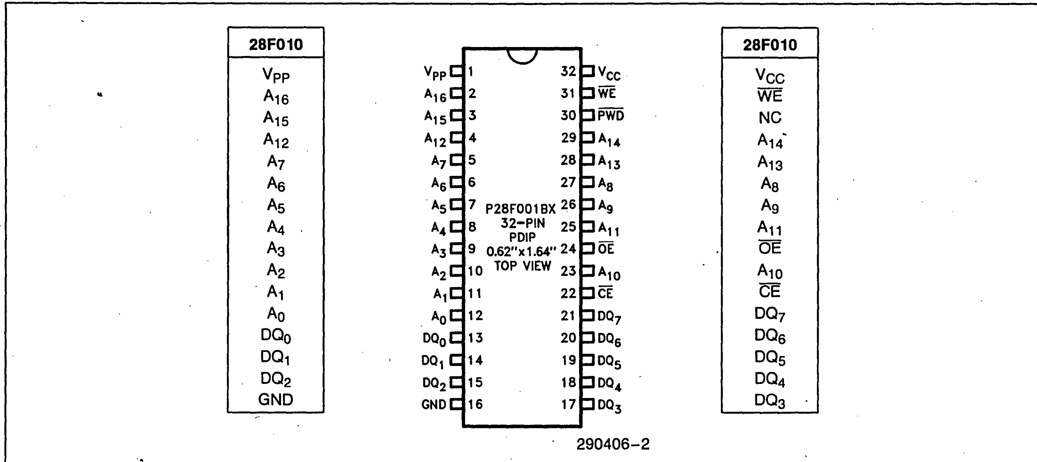


Figure 2. DIP Pin Configuration

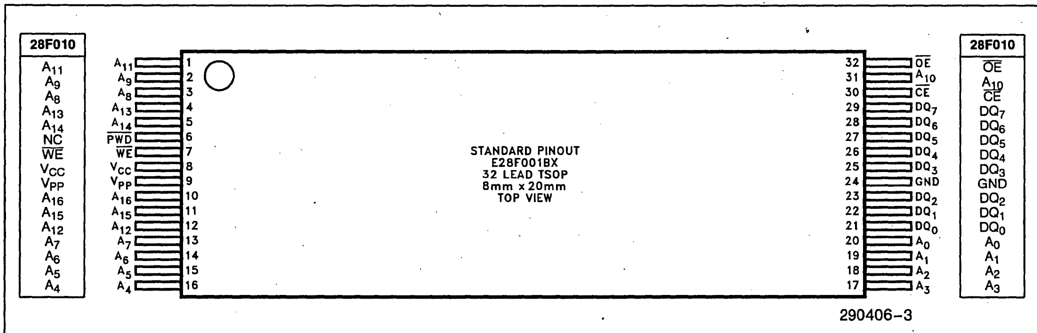


Figure 3. TSOP Lead Configuration

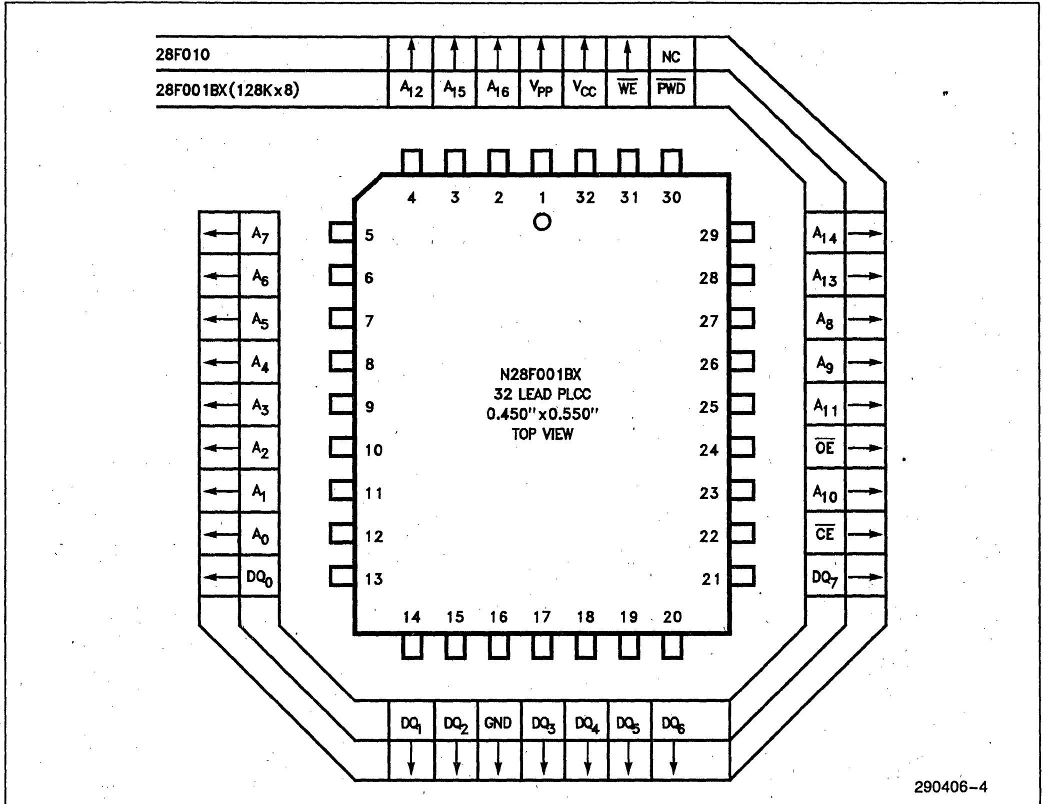


Figure 4. PLCC Lead Configuration

APPLICATIONS

The 28F001BX flash 'boot block' memory augments the non-volatility, in-system electrical erasure and reprogrammability of Intel's standard flash memory by offering four separately erasable blocks and integrating a state machine to control erase and program functions. The specialized blocking architecture and automated programming of the 28F001BX provide a full-function, non-volatile flash memory ideal for a wide range of applications, including PC boot/BIOS memory, minimum-chip embedded program memory and parametric data storage. The 28F001BX combines the safety of a hardware-protected 8-KByte boot block with the flexibility of three separately reprogrammable blocks (two 4-KByte parameter blocks and one 112-KByte code block) into one versatile, cost-effective flash memory. Additionally, reprogramming one block does not affect code stored in another block, ensuring data integrity.

The flexibility of flash memory reduces costs throughout the life cycle of a design. During the early stages of a system's life, flash memory reduces prototype development and testing time, allowing the system designer to modify in-system software electrically versus manual removal of components. During production, flash memory provides flexible firmware for just-in-time configuration, reducing system inventory and eliminating unnecessary handling and less reliable socketed connections. Late in the life cycle, when software updates or code "bugs" are often unpredictable and costly, flash memory reduces update costs by allowing the manufacturers to send floppy updates versus a technician. Alternatively, remote updates over a communication link are possible at speeds up to 9600 baud due to flash memory's fast programming time.

Reprogrammable environments, such as the personal computer, are ideal applications for the 28F001BX. The internal state machine provides SRAM-like timings for program and erasure, using the Command and Status Registers. The blocking scheme allows BIOS update in the main and parameter blocks, while still providing recovery code in the boot block in the unlikely event a power failure occurs during an update, or where BIOS code is corrupted. Parameter blocks also provide convenient configuration storage, backing up SRAM and battery configurations. EISA systems, for example, can store hardware configurations in a flash parameter block, reducing system SRAM.

Laptop BIOSs are becoming increasingly complex with the addition of power management software and extended system setup screens. BIOS code complexity increases the potential for code updates after the sale, but the compactness of laptop designs makes hardware updates very costly. Boot block flash memory provides an inexpensive update solution for laptops, while reducing laptop obsolescence. For portable PCs and hand-held equipment, the deep powerdown mode dramatically lowers system power requirements during periods of slow operation or sleep modes.

The 28F001BX gives the embedded system designer several desired features. The internal state machine reduces the size of external code dedicated to the erase and program algorithms, as well as freeing the microcontroller or microprocessor to respond to other system requests during program and erasure. The four blocks allow logical segmentation of the entire embedded software: the 8-KByte block for the boot code, the 112-KByte block for the main program code and the two 4-KByte blocks for updatable parametric data storage, diagnostic messages and data, or extensions of either the boot code or program code. The boot block is hardware protected against unauthorized write or erase of its vital code in the field. Further, the powerdown mode also locks out erase or write operations, providing absolute data protection during system powerup or power loss. This hardware protection provides obvious advantages for safety related applications such as transportation, military, and medical. The 28F001BX is well suited for minimum-chip embedded applications ranging from communications to automotive.

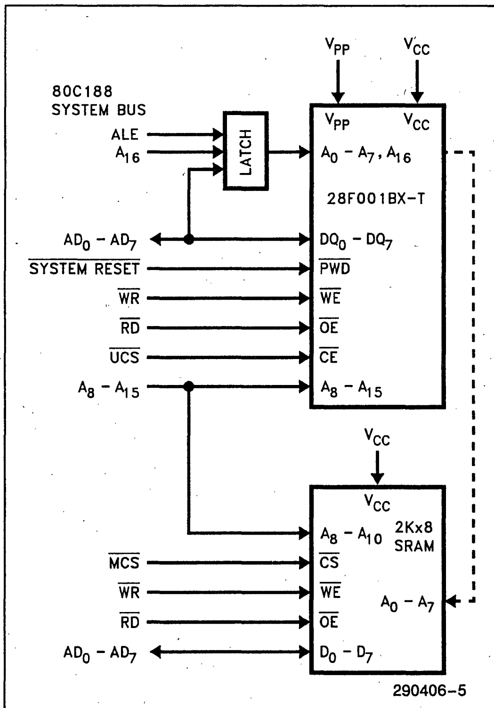


Figure 5. 28F001BX-T in a 80C188 System

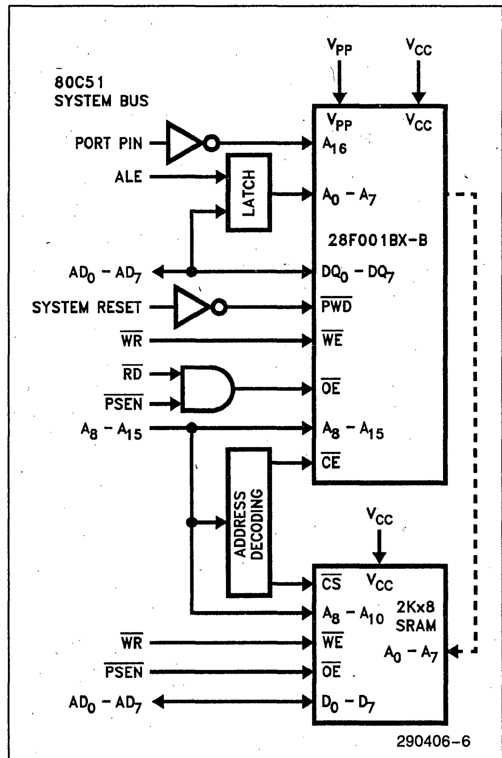


Figure 6. 28F001BX-B in a 80C51 System

PRINCIPLES OF OPERATION

The 28F001BX introduces on-chip write automation to manage write and erase functions. The write state machine allows for: 100% TTL-level control inputs; fixed power supplies during erasure and programming; minimal processor overhead with RAM-like write timings, and maximum EPROM compatiblity.

After initial device powerup, or after return from deep powerdown mode (see Bus Operations), the 28F001BX functions as a read-only memory. Manipulation of external memory-control pins yield standard EPROM read, standby, output disable or int$_e$ligent Identifier operations. Both Status Register and int$_e$ligent Identifiers can be accessed through the Command Register when $V_{PP} = V_{PPL}$.

This same subset of operations is also available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} enables successful erasure and programming of the device. All functions associated with altering memory contents—program, erase, status, and int$_e$ligent Identifier—are accessed via the Command Register and verified through the Status Register.

Commands are written using standard microprocessor write timings. Register contents serve as input to the WSM, which controls the erase and programming circuitry. Write cycles also internally latch addresses and data needed for programming or erase operations. With the appropriate command written to the register, standard microprocessor read timings output array data, access the intelligent identifier codes, or output program and erase status for verification.

Interface software to initiate and poll progress of internal program and erase can be stored in any of the 28F001BX blocks. This code is copied to, and executed from, system RAM during actual flash memory update. After successful completion of program and/or erase, code execution out of the 28F001BX is again possible via the Read Array command. Erase suspend/resume capability allows system software to suspend block erase and read data/execute code from any other block.

Command Register and Write Automation

An on-chip state machine controls block erase and byte program, freeing the system processor for other tasks. After receiving the erase setup and erase confirm commands, the state machine controls block pre-conditioning and erase, returning progress via the Status Register. Programming is similarly controlled, after destination address and expected data are supplied. The program algorithm of past Intel Flash memories is now regulated by the state machine, including program pulse repetition where required and internal verification and margining of data.

Data Protection

Depending on the application, the system designer may choose to make the V_{PP} power supply switchable (available only when memory updates are required) or hardwired to V_{PPH}. When $V_{PP} = V_{PPL}$, memory contents cannot be altered. The 28F001BX Command Register architecture provides protection from unwanted program or erase operations even when high voltage is applied to V_{PP}. Additionally, all functions are disabled whenever V_{CC} is below the write lockout voltage V_{LKO}, or when $\overline{PWD}$ is at V_{IL}. The 28F001BX accommodates either design practice and encourages optimization of the processor-memory interface.

The two-step program/erase write sequence to the Command Register provides additional software write protection.

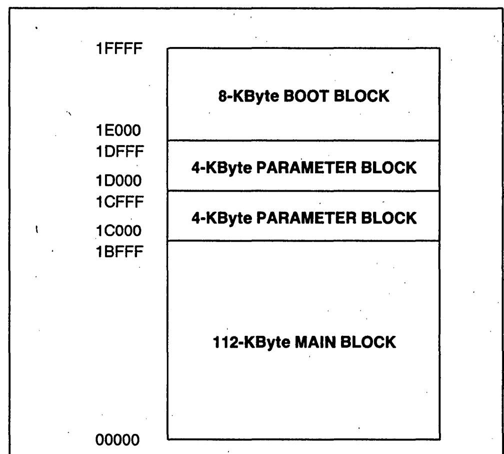


Figure 7. 28F001BX-T Memory Map

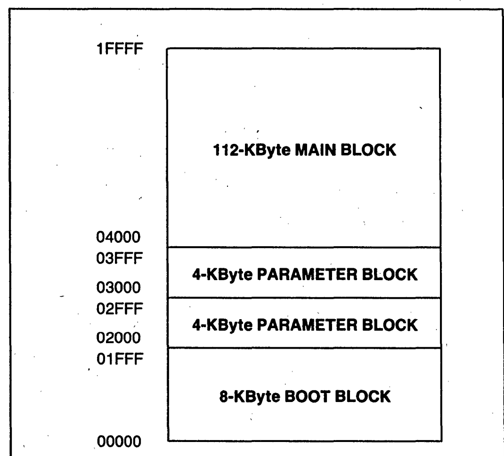


Figure 8. 28F001BX-B Memory Map

BUS OPERATION

Flash memory reads, erases and writes in-system via the local CPU. All bus cycles to or from the flash memory conform to standard microprocessor bus cycles.

Read

The 28F001BX has three read modes. The memory can be read from any of its blocks, and information can be read from the intelligent identifier or the Status Register. V_{PP} can be at either V_{PPL} or V_{PPH}.

The first task is to write the appropriate read mode command to the Command Register (array, intelligent identifier, or Status Register). The 28F001BX automatically resets to Read Array mode upon initial device powerup or after exit from deep powerdown. The 28F001BX has four control pins, two of which must be logically active to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the device selection control, and when active enables the selected memory device. Output Enable ($\overline{OE}$) is the data input/output (DQ_0–DQ_7) direction control, and when active drives data from the selected memory onto the I/O bus. $\overline{PWD}$ and $\overline{WE}$ must also be at V_{IH}. Figure 12 illustrates read bus cycle waveforms.

Output Disable

With $\overline{OE}$ at a logic-high level (V_{IH}), the device outputs are disabled. Output pins (DQ_0–DQ_7) are placed in a high-impedance state.

Standby

$\overline{CE}$ at a logic-high level (V_{IH}) places the 28F001BX in standby mode. Standby operation disables much of the 28F001BX's circuitry and substantially reduces device power consumption. The outputs (DQ_0–DQ_7) are placed in a high-impedance state independent of the status of $\overline{OE}$. If the 28F001BX is deselected during erase or program, the device will continue functioning and consuming normal active power until the operation is completed.

3

Table 2. 28F001BX Bus Operations

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_9	A_0	V_{PP}	DQ_{0-7}
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	X	D_{OUT}
Output Disable		V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	High Z
Standby		V_{IH}	V_{IH}	X	X	X	X	X	High Z
Deep Power Down		V_{IL}	X	X	X	X	X	X	High Z
Intelligent Identifier (Mfr)	4	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IL}	X	89H
Intelligent Identifier (Device)	4, 5	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IH}	X	94H, 95H
Write	6, 7, 8	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	D_{IN}

NOTES:
1. Refer to DC Characteristics. When $V_{PP} = V_{PPL}$, memory contents can be read but not programmed or erased.
2. X can be V_{IL} or V_{IH} for control pins and addresses, and V_{PPL} or V_{PPH} for V_{PP}.
3. See DC Characteristics for V_{PPL}, V_{PPH}, V_{HH} and V_{ID} voltages.
4. Manufacturer and device codes may also be accessed via a Command Register write sequence. Refer to Table 3. A_1–A_8, A_{10}–A_{16} = V_{IL}.
5. Device ID = 94H for the 28F001BX-T and 95H for the 28F001BX-B.
6. Command writes involving block erase or byte program are successfully executed only when $V_{PP} = V_{PPH}$.
7. Refer to Table 3 for valid D_{IN} during a write operation.
8. Program or erase the boot block by holding $\overline{PWD}$ at V_{HH} or toggling $\overline{OE}$ to V_{HH}. See AC Waveforms for Program/Erase Operations.

Deep Power-Down

The 28F001BX offers a 0.25 μW V_{CC} powerdown feature, entered when $\overline{PWD}$ is at V_{IL}. During read modes, $\overline{PWD}$-low deselects the memory, places output drivers in a high-impedance state and turns off all internal circuits. The 28F001BX requires time t_{PHQV} (see AC Characteristics-Read Only Operations) after return from powerdown until initial memory access outputs are valid. After this wakeup interval, normal operation is restored. The Command Register is reset to Read Array, and the Status Register is cleared to value 80H, upon return to normal operation.

During erase or program modes, $\overline{PWD}$ low will abort either operation. Memory contents of the block being altered are no longer valid as the data will be partially programmed or erased. Time t_{PHWL} after $\overline{PWD}$ goes to logic-high (V_{IH}) is required before another command can be written.

Intelligent Identifier Operation

The intelligent identifier operation outputs the manufacturer code, 89H; and the device code, 94H for the 28F001BX-T and 95H for the 28F001BX-B. Programming equipment or the system CPU can then automatically match the device with its proper erase and programming algorithms.

PROGRAMMING EQUIPMENT

$\overline{CE}$ and $\overline{OE}$ at a logic low level (V_{IL}), with A_9 at high voltage V_{ID} (see DC Characteristics) activates this operation. Data read from locations 00000H and 00001H represent the manufacturer's code and the device code respectively.

IN-SYSTEM PROGRAMMING

The manufacturer- and device-codes can also be read via the Command Register. Following a write of 90H to the Command Register, a read from address location 00000H outputs the manufacturer code (89H). A read from address 00001H outputs the device code (94H for the 28F001BX-T and 95H for the 28F001BX-B). It is not necessary to have high voltage applied to V_{PP} to read the intelligent identifiers from the Command Register.

Write

Writes to the Command Register allow read of device data and intelligent identifiers. They also control inspection and clearing of the Status Register. Additionally, when $V_{PP} = V_{PPH}$, the Command Register controls device erasure and programming. The contents of the register serve as input to the internal state machine.

The Command Register itself does not occupy an addressable memory location. The register is a latch used to store the command and address and data information needed to execute the command. Erase Setup and Erase Confirm commands require both appropriate command data and an address within the block to be erased. The Program Setup Command requires both appropriate command data and the address of the location to be programmed, while the Program command consists of the data to be written and the address of the location to be programmed.

The Command Register is written by bringing $\overline{WE}$ to a logic-low level (V_{IL}) while $\overline{CE}$ is low. Addresses and data are latched on the rising edge of $\overline{WE}$. Standard microprocessor write timings are used.

Refer to AC Write Characteristics and the AC Waveform for Write Operations, Figure 13, for specific timing parameters.

COMMAND DEFINITIONS

When V_{PPL} is applied to the V_{PP} pin, read operations from the Status Register, intelligent identifiers, or array blocks are enabled. Placing V_{PPH} on V_{PP} enables successful program and erase operations as well.

Device operations are selected by writing specific commands into the Command Register. Table 3 defines these 28F001BX commands.

Read Array Command

Upon initial device powerup and after exit from deep powerdown mode, the 28F001BX defaults to Read Array mode. This operation is also initiated by writing FFH into the Command Register. Microprocessor read cycles retrieve array data. The device remains enabled for reads until the Command Register contents are altered. Once the internal write state machine has started an erase or program operation, the device will not recognize the Read Array command, until the WSM has completed its operation. The Read Array command is functional when $V_{PP} = V_{PPL}$ or V_{PPH}.

Intelligent Identifier Command for In-System Programming

The 28F001BX contains an intelligent identifier operation to supplement traditional PROM-programming methodology. The operation is initiated by writing 90H into the Command Register. Following the command write, a read cycle from address 00000H retrieves the manufacturer code of 89H. A read cycle from address 00001H returns the device code of 94H (28F001BX-T) or 95H (28F001BX-B). To terminate the operation, it is necessary to write another valid command into the register. Like the Read Array command, the intelligent identifier command is functional when $V_{PP} = V_{PPL}$ or V_{PPH}.

Table 3. 28F001BX Command Definitions

Command	Bus Cycles Req'd	Notes	First Bus Cycle			Second Bus Cycle		
			Operation	Address	Data	Operation	Address	Data
Read Array/Reset	1	1	Write	X	FFH			
Intelligent Identifier	3	2, 3, 4	Write	X	90H	Read	IA	IID
Read Status Register	2	3	Write	X	70H	Read	X	SRD
Clear Status Register	1		Write	X	50H			
Erase Setup/Erase Confirm	2	2	Write	BA	20H	Write	BA	D0H
Erase Suspend/Erase Resume	2		Write	X	B0H	Write	X	D0H
Program Setup/Program	2	2, 3	Write	PA	40H	Write	PA	PD

NOTES:
1. Bus operations are defined in Table 2.
2. IA = Identifier Address: 00H for manufacturer code, 01H for device code.
   BA = Address within the block being erased.
   PA = Address of memory location to be programmed.
3. SRD = Data read from Status Register. See Table 4 for a description of the Status Register bits.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of $\overline{WE}$.
   IID = Data read from int$_e$ligent Identifiers.
4. Following the intelligent identifier command, two read operations access manufacture and device codes.
5. Commands other than those shown above are reserved by Intel for future device implementations and should not be used.

Read Status Register Command

The 28F001BX contains a Status Register which may be read to determine when a program or erase operation is complete, and whether that operation completed successfully. The Status Register may be read at any time by writing the Read Status Register command (70H) to the Command Register. After writing this command, all subsequent read operations output data from the Status Register, until another valid command is written to the Command Register. The contents of the Status Register are latched on the falling edge of $\overline{OE}$ or $\overline{CE}$, whichever occurs last in the read cycle. $\overline{OE}$ or $\overline{CE}$ must be toggled to V_{IH} before further reads to update the Status Register latch. The Read Status Register command functions when $V_{PP} = V_{PPL}$ or V_{PPH}.

Clear Status Register Command

The Erase Status and Program Status bits are set to "1" by the Write State Machine and can only be reset by the Clear Status Register command. These bits indicate various failure conditions (see Table 4). By allowing system software to control the resetting of these bits, several operations may be performed (such as cumulatively programming several bytes or erasing multiple blocks in sequence). The Status Register may then be polled to determine if an error occurred during that series. This adds flexibility to the way the device may be used.

Additionally, the V_{PP} Status bit (SR.3), when set to "1", MUST be reset by system software before further byte programs or block erases are attempted. To clear the Status Register, the Clear Status Register command (50H) is written to the Command Register. The Clear Status Register command is functional when $V_{PP} = V_{PPL}$ or V_{PPH}.

Table 4. 28F001BX Status Register Definitions

WSMS	ESS	ES	PS	VPPS	R	R	R
7	6	5	4	3	2	1	0

SR.7 = WRITE STATE MACHINE STATUS
1 = Ready
0 = Busy

SR.6 = ERASE SUSPEND STATUS
1 = Erase Suspended
0 = Erase In Progress/Completed

SR.5 = ERASE STATUS
1 = Error in Block Erasure
0 = Successful Block Erase

SR.4 = PROGRAM STATUS
1 = Error in Byte Program
0 = Successful Byte Program

SR.3 = V_{PP} STATUS
1 = V_{PP} Low Detect; Operation Abort
0 = V_{PP} OK

SR.2–SR.0 = RESERVED FOR FUTURE ENHANCEMENTS
These bits are reserved for future use and should be masked out when polling the Status Register.

NOTES:
The Write State Machine Status Bit must first be checked to determine program or erase completion, before the Program or Erase Status bits are checked for success.

If the Program AND Erase Status bits are set to "1s" during an erase attempt, an improper command sequence was entered. Attempt the operation again.

If V_{PP} low status is detected, the Status Register must be cleared before another program or erase operation is attempted.

The V_{PP} Status bit, unlike an A/D converter, does not provide continuous indication of V_{PP} level. The WSM interrogates the V_{PP} level only after the program or erase command sequences have been entered and informs the system if V_{PP} has not been switched on. The V_{PP} Status bit is not guaranteed to report accurate feedback between V_{PPL} and V_{PPH}.

Erase Setup/Erase Confirm Commands

Erase is executed one block at a time, initiated by a two-cycle command sequence. An Erase Setup command (20H) is first written to the Command Register, followed by the Erase Confirm command (D0H). These commands require both appropriate command data and an address within the block to be erased. Block preconditioning, erase and verify are all handled internally by the Write State Machine, invisible to the system. After receiving the two-command erase sequence, the 28F001BX automatically outputs Status Register data when read (see Figure 10; Block Erase Flowchart). The CPU can detect the completion of the erase event by checking the WSM Status bit of the Status Register (SR.7).

When the Status Register indicates that erase is complete, the Erase Status bit should be checked. If erase error is detected, the Status Register should be cleared. The Command Register remains in Read Status Register Mode until further commands are issued to it.

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, block erasure can only occur when $V_{PP} = V_{PPH}$. In the absence of this high voltage, memory contents are protected against erasure. If block erase is attempted while $V_{PP} = V_{PPL}$, the V_{PP} Status bit will be set to "1". Erase attempts while $V_{PPL} < V_{PP} < V_{PPH}$ produce spurious results and should not be attempted.

Erase Suspend/Erase Resume Commands

The Erase Suspend Command allows erase sequence interruption in order to read data from another block of memory. Once the erase sequence is started, writing the Erase Suspend command (B0H) to the Command Register requests that the WSM suspend the erase sequence at a predetermined point in the erase algorithm. The 28F001BX continues to output Status Register data when read, after the Erase Suspend command is written to it. Polling the WSM Status and Erase Suspend Status bits will determine when the erase operation has been suspended (both will be set to "1s").

At this point, a Read Array command can be written to the Command Register to read data from blocks **other than that which is suspended.** The only other valid commands at this time are Read Status Register (70H) and Erase Resume (D0H), at which time the WSM will continue with the erase sequence. The Erase Suspend Status and WSM Status bits of the Status Register will be cleared. After the Erase Resume command is written to it, the 28F001BX automatically outputs Status Register data when read (see Figure 11; Erase Suspend/Resume Flowchart).

Program Setup/Program Commands

Programming is executed by a two-write sequence. The program Setup command (40H) is written to the Command Register, followed by a second write specifying the address and data (latched on the rising edge of $\overline{WE}$) to be programmed. The WSM then takes over, controlling the program and verify algorithms internally. After the two-command program sequence is written to it, the 28F001BX automatically outputs Status Register data when read (see figure 9; Byte Program Flowchart). The CPU can detect the completion of the program event by analyzing the WSM Status bit of the Status Register. Only the Read Status Register command is valid while programming is active.

When the Status Register indicates that programming is complete, the Program Status bit should be checked. If program error is detected, the Status Register should be cleared. The internal WSM verify only detects errors for "1s" that do not successfully program to "0s". The Command Register remains in Read Status Register mode until further commands are issued to it. If byte program is attempted while $V_{PP} = V_{PPL}$, the V_{PP} Status bit will be set to "1". Program attempts while $V_{PPL} < V_{PP} < V_{PPH}$ produce spurious results and should not be attempted.

EXTENDED ERASE/PROGRAM CYCLING

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled; an expensive solution.

Intel has designed extended cycling capability into its ETOX II flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electrical field is one-tenth that of common EEPROMs, minimizing the probability of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 Mv/cm lower than EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure; increasing time to wearout by a factor of 100,000,000.

The 28F001BX-B and 28F001BX-T are specified for a minimum of 10,000 program/erase cycles on each of the two parameter blocks, main block and boot block.

ON-CHIP PROGRAMMING ALGORITHM

The 28F001BX integrates the Quick-Pulse programming algorithm of prior Intel Flash devices on-chip, using the Command Register, Status Register and Write State Machine (WSM). On-chip integration dramatically simplifies system software and provides processor-like interface timings to the Command and Status Registers. WSM operation, internal program verify and V_{PP} high voltage presence are monitored and reported via appropriate Status Register bits. Figure 9 shows a system software flowchart for device programming. The entire sequence is performed with V_{PP} at V_{PPH}. Program abort occurs when $\overline{PWD}$ transitions to V_{IL}, or V_{PP} drops to V_{PPL}. Although the WSM is halted, byte data is partially programmed at the location where programming was aborted. Block erasure or a repeat of byte programming will initialize this data to a known value.

3

ON-CHIP ERASE ALGORITHM

As above, the Quick-Erase algorithm of prior Intel Flash devices is now implemented internally, including all preconditioning of block data. WSM operation, erase success and V_{PP} high voltage presence are monitored and reported through the Status Register. Additionally, if a command other than Erase Confirm is written to the device after Erase Setup has been written, both the Erase Status and Program Status bits will be set to "1". When issuing the Erase Setup and Erase Confirm commands, they should be written to an address within the address range of the block to be erased. Figure 10 shows a system software flowchart for block erase.

Erase typically takes 1–4 seconds per block. The Erase Suspend/Erase Resume command sequence allows interrupt of this erase operation to read data **from a block other than that in which erase is being performed.** A system software flowchart is shown in Figure 11.

The entire sequence is performed with V_{PP} at V_{PPH}. Abort occurs when $\overline{PWD}$ transitions to V_{IL} or V_{PP} falls to V_{PPL}, while erase is in progress. Block data is partially erased by this operation, and a repeat of erase is required to obtain a fully erased block.

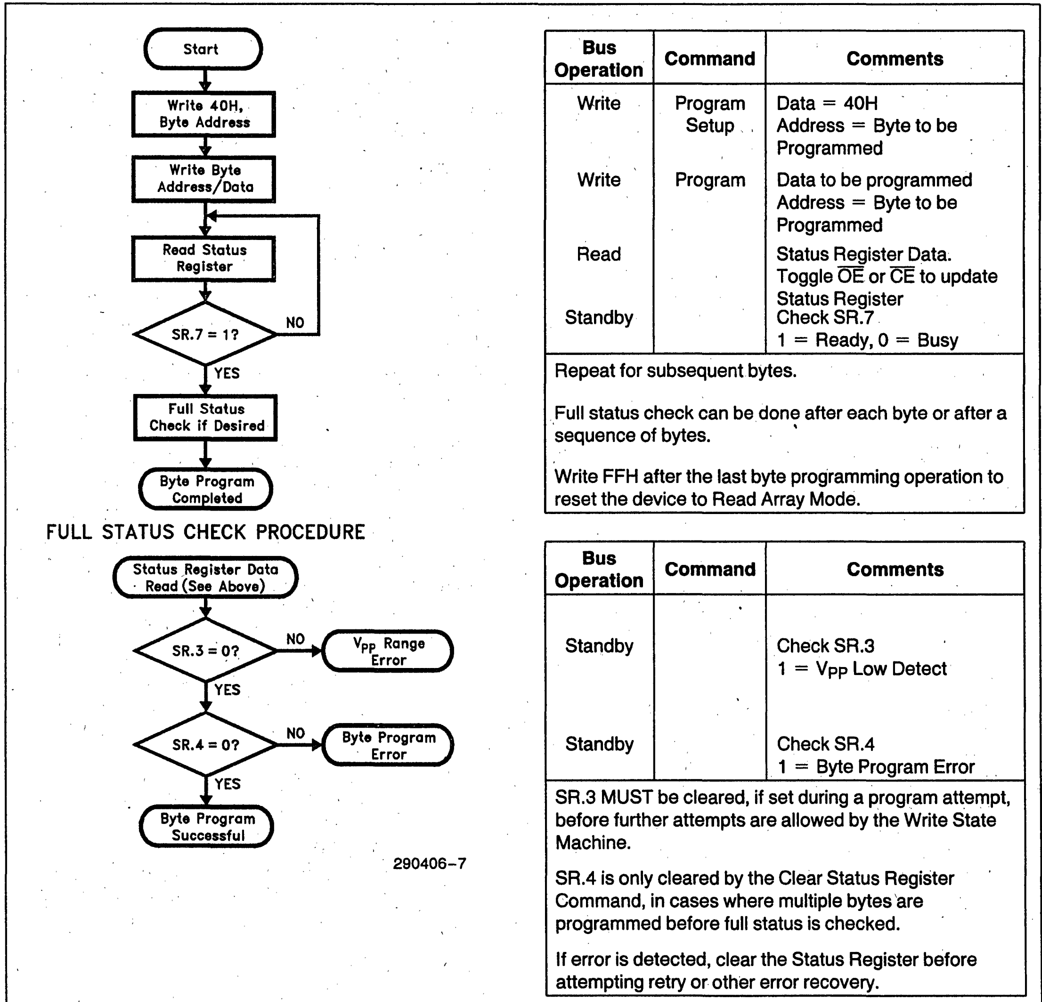
BOOT BLOCK PROGRAM AND ERASE

The boot block is intended to contain secure code which will minimally bring up a system and control programming and erase of other blocks of the device, if needed. Therefore, additional "lockout" protection is provided to guarantee data integrity. Boot block program and erase operations are enabled through high voltage V_{HH} on either $\overline{PWD}$ or $\overline{OE}$, and the normal program and erase command sequences are used. Reference the AC Waveforms for Program/Erase.

If boot block program or erase is attempted while $\overline{PWD}$ is at V_{IH}, either the Program Status or Erase Status bit will be set to "1", reflective of the operation being attempted and indicating boot block lock. Program/erase attempts while $V_{IH} < \overline{PWD} < V_{HH}$ produce spurious results and should not be attempted.

In-System Operation

For on-board programming, the $\overline{PWD}$ pin is the most convenient means of altering the boot block. Before issuing Program or Erase confirms commands, $\overline{PWD}$ must transition to V_{HH}. Hold $\overline{PWD}$ at this high voltage throughout the program or erase interval (until after Status Register confirm of successful completion). At this time, it can return to V_{IH} or V_{IL}.



Bus Operation	Command	Comments
Write	Program Setup	Data = 40H Address = Byte to be Programmed
Write	Program	Data to be programmed Address = Byte to be Programmed
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent bytes.

Full status check can be done after each byte or after a sequence of bytes.

Write FFH after the last byte programming operation to reset the device to Read Array Mode.

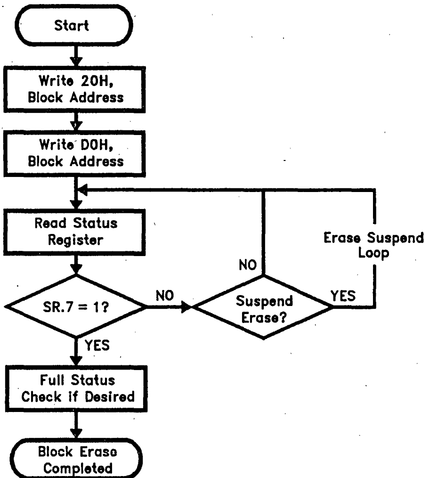
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Byte Program Error

SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple bytes are programmed before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 9. 28F001BX Byte Programming Flowchart



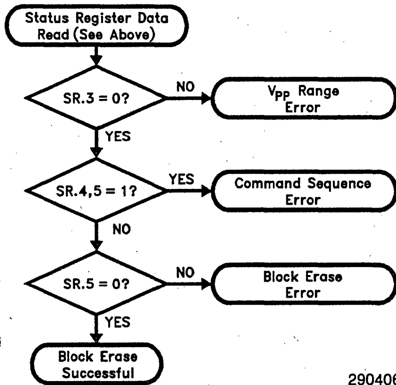
Bus Operation	Command	Comments
Write	Erase Setup	Data = 20H Address = Within Block to be erased
Write	Erase	Data = D0H Address = Within Block to be erased
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent blocks.

Full status check can be done after each block or after a sequence of blocks.

Write FFH after the last block erase operation to reset the device to Read Array Mode.

FULL STATUS CHECK PROCEDURE



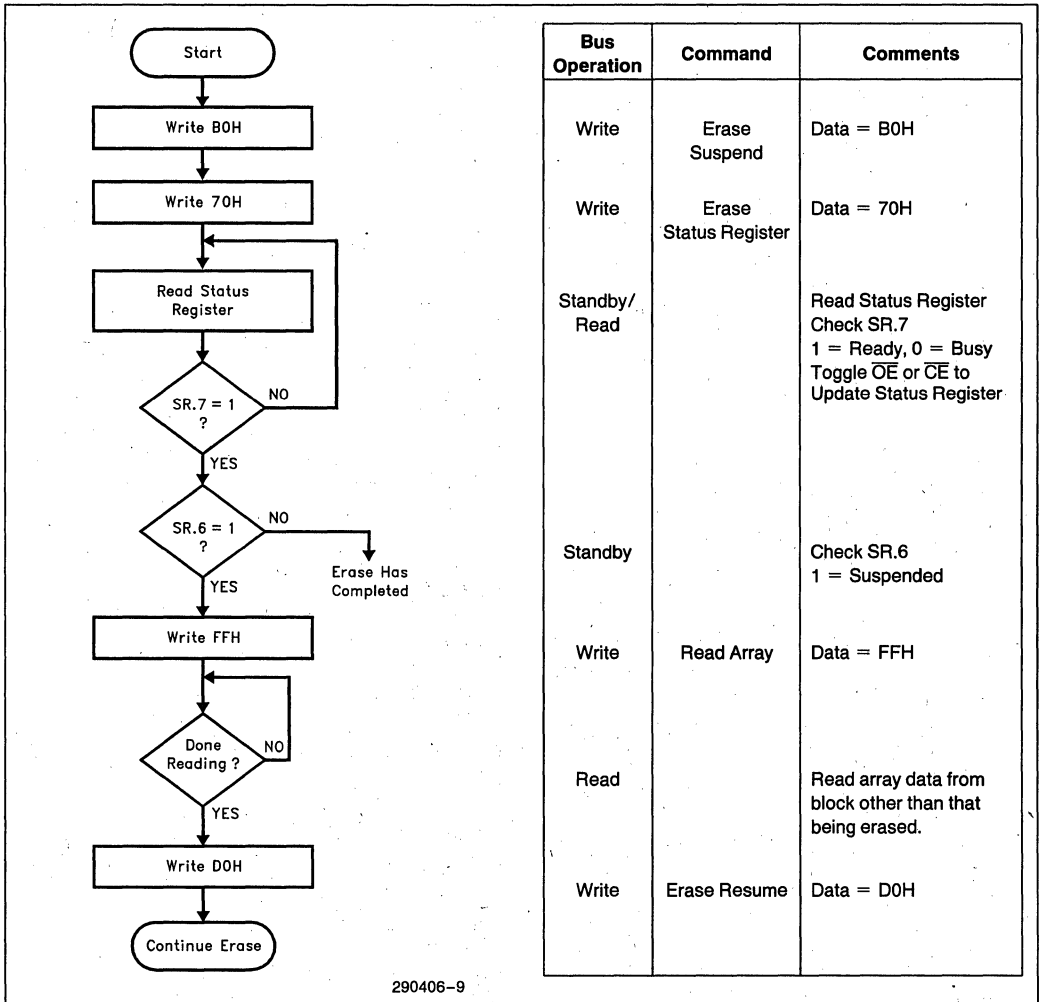
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4, 5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error

SR.3 MUST be cleared, if set during an erase attempt, before further attempts are allowed by the Write State Machine.

SR.5 is only cleared by the Clear Status Register Command, in cases where multiple blocks are erased before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 10. 28F001BX Block Erase Flowchart



Bus Operation	Command	Comments
Write	Erase Suspend	Data = B0H
Write	Erase Status Register	Data = 70H
Standby/ Read		Read Status Register Check SR.7 1 = Ready, 0 = Busy Toggle $\overline{OE}$ or $\overline{CE}$ to Update Status Register
Standby		Check SR.6 1 = Suspended
Write	Read Array	Data = FFH
Read		Read array data from block other than that being erased.
Write	Erase Resume	Data = D0H

Figure 11. 28F001BX Erase Suspend/Resume Flowchart

Programming Equipment

For PROM programming equipment that cannot bring $\overline{PWD}$ to high voltage, $\overline{OE}$ provides an alternate boot block access mechanism. $\overline{OE}$ must transition to V_{HH} a minimum of 480 ns before the initial program/erase setup command and held at V_{HH} at least 480 ns after program or erase confirm commands are issued to the device. After this interval, $\overline{OE}$ can return to normal TTL levels.

DESIGN CONSIDERATIONS

Three-Line Output Control

Flash memories are often used in larger memory arrays. Intel provides three control inputs to accommodate multiple memory connections. Three-line control provides for:

a) lowest possible memory power dissipation

b) complete assurance that data bus contention will not occur

To efficiently use these control inputs, an address decoder should enable $\overline{CE}$, while $\overline{OE}$ should be connected to all memory devices and the system's $\overline{READ}$ control line. This assures that only selected memory devices have active outputs while deselected memory devices are in Standby Mode. Finally, $\overline{PWD}$ should either be connected to V_{CC} if unused, or tied to the system $\overline{RESET}$.

Power Supply Decoupling

Flash memory power switching characteristics require careful device coupling. System designers are interested in 3 supply current issues; standby current levels (I_{SB}), active current levels (I_{CC}) and transient peaks producted by falling and rising edges of $\overline{CE}$. Transient current magnitudes depend on the device outputs' capacitive and inductive loading. Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 µF ceramic capacitor connected between its V_{CC} and GND, and between its V_{PP} and GND. These high frequency, low inherent-inductance capacitors should be placed as close as possible to the device. Additionally, for every 8 devices, a 4.7 µF electrolytic capacitor should be placed at the array's power supply connection between V_{CC} and GND. The bulk capacitor will overcome voltage slumps caused by PC board trace inductances.

V_{PP} Trace on Printed Circuit Boards

Programming flash memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the V_{PP} power supply trace. The V_{PP} pin supplies the memory cell current for programming. Use similar trace widths and layout considerations given to the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots.

V_{CC}, V_{PP}, $\overline{PWD}$ Transitions and the Command/Status Registers

Programming and erase completion are not guaranteed if V_{PP} drops below V_{PPH}. If the V_{PP} Status bit of the Status Register (SR.3) is set to "1", a Clear Status Register command MUST be issued before further program/erase attempts are allowed by the WSM. Otherwise, the Program (SR.4) or Erase (SR.5) Status bits of the Status Register will be set to "1" if error is detected. $\overline{PWD}$ transitions to V_{IL} during program and erase also abort the operations. Data is partially altered in either case, and the command sequence must be repeated after normal operation is restored. Device poweroff, or $\overline{PWD}$ transitions to V_{IL}, clear the Status Register to initial value 80H.

The Command Register latches commands as issued by system software and is not altered by V_{PP} or $\overline{CE}$ transitions or WSM actions. Its state upon powerup, after exit from Deep Power Down or after V_{CC} transitions below V_{LKO}, is FFH, or Read Array Mode.

After program or erase is complete, even after V_{PP} transitions down to V_{PPL}, the Command Register must be reset to read array mode via the Read Array command if access to the memory array is desired.

Power Up/Down Protection

The 28F001BX is designed to offer protection against accidental erasure or programming during power transitions. Upon power-up, the 28F001BX is indifferent as to which power supply, V_{PP} or V_{CC}, powers up first. Power supply sequencing is not required. Internal circuitry in the 28F001BX ensures that the Command Register is reset to Read Array mode on power up.

A system designer must guard against spurious writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will inhibit writes. The Command Register architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

Finally, the device is disabled, until $\overline{PWD}$ is brought to V_{IH}, regardless of the state of its control inputs. This provides an additional level of protection.

28F001BX Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases usable battery life because the 28F001BX does not consume any power to retain code or data when the system is off.

In addition, the 28F001BX's Deep Power Down mode ensures extremely low power dissipation even when system power is applied. For example, laptop and other PC applications, after copying BIOS to DRAM, can lower $\overline{PWD}$ to V_{IL}, producing negligible power consumption. If access to the boot code is again needed, as in case of a system $\overline{RESET}$, the part can again be accessed, following the t_{PHAV} wakeup cycle required after $\overline{PWD}$ is first raised back to V_{IH}. The first address presented to the device while in powerdown requires time t_{PHAV}, after $\overline{PWD}$ transitions high, before outputs are valid. Further accesses follow normal timing. See AC Characteristics—Read-Only Operations and Figure 12 for more information.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read .... 0°C to 70°C(1)
During Erase/Program .... 0°C to 70°C(1)

Operating Temperature
During Read .... −40°C to +85°C(2)
During Erase/Program ...... −40°C to +85°C(2)

Temperature under Bias .... −10°C to 80°C(1)

Temperature under Bias .... −20°C to +90°C(2)

Storage Temperature............. −65°C to 125°C

Voltage on Any Pin
(except A_9, $\overline{PWD}$, $\overline{OE}$, V_{CC} and V_{PP})
with Respect to GND .... −2.0V to 7.0V(3)

Voltage on A_9, $\overline{PWD}$, and $\overline{OE}$
with Respect to GND .... −2.0V to 13.5V(3, 4)

V_{PP} Program Voltage
with Respect to GND
During Erase/Program ...... −2.0V to 14.0V(3, 4)

V_{CC} Supply Voltage
with Respect to GND .... −2.0V to 7.0V(3)

Output Short Circuit Current............. 100 mA(5)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

* *WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Operating temperature is for extended temperature product defined by this specification.
3. Minimum DC voltage is −0.5V on input/output pins. During transitions, this level may undershoot to −2.0V for periods <20 ns. Maximum DC voltage on input/output pins is V_{CC} +0.5V which, during transitions, may overshoot to V_{CC} + 2.0V for periods <20 ns.
4. Maximum DC voltage on A_9 or V_{PP} may overshoot to +14.0V for periods <20 ns.
5. Output shorted for no more than one second. No more than one output shorted at a time.

OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
T_A	Operating Temperature(1)	0	70	°C
T_A	Operating Temperature(2)	−40	85	°C
V_{CC}	Supply Voltage	4.50	5.50	V

DC CHARACTERISTICS V_{CC} = 5.0V ±10%, T_A = 0°C to +70°C

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Conditions
I_{IL}	Input Load Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND
I_{CCS}	V_{CC} Standby Current			1.2	2.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{IH}$
				30	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{CC} \pm 0.2V$
I_{CCD}	V_{CC} Deep PowerDown Current	1		0.05	1.0	μA	$\overline{PWD}$ = GND ±0.2V

DC CHARACTERISTICS V_{CC} = 5.0V ±10%, T_A = 0°C to +70°C (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Conditions
I_{CCR}	V_{CC} Read Current	1		13	30	mA	V_{CC} = V_{CC} Max, $\overline{CE}$ = V_{IL} f = 8 MHz, I_{OUT} = 0 mA
I_{CCP}	V_{CC} Programming Current	1		5	20	mA	Programming in Progress
I_{CCE}	V_{CC} Erase Current	1		6	20	mA	Erase in Progress
I_{CCES}	V_{CC} Erase Suspend Current	1, 2		5	10	mA	Erase Suspended $\overline{CE}$ = V_{IH}
I_{PPS}	V_{PP} Standby Current	1		±1	±10	μA	$V_{PP} \leq V_{CC}$
				90	200	μA	$V_{PP} > V_{CC}$
I_{PPD}	V_{PP} Deep PowerDown Current	1		0.80	1.0	μA	$\overline{PWD}$ = GND ±0.2V
I_{PPP}	V_{PP} Programming Current	1		6	30	mA	V_{PP} = V_{PPH} Programming in Progress
I_{PPE}	V_{PP} Erase Current	1		6	30	mA	V_{PP} = V_{PPH} Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1		90	300	μA	V_{PP} = V_{PPH} Erase Suspended
I_{ID}	A_9 Intelligent Identifier Current	1		90	500	μA	A_9 = V_{ID}
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	V_{CC} = V_{CC} Min I_{OL} = 5.8 mA
V_{OH}	Output High Voltage		2.4			V	V_{CC} = V_{CC} Min I_{OH} = 2.5 mA
V_{ID}	A_9 Intelligent Identifier Voltage		11.5		13.0	V	
V_{PPL}	V_{PP} during Normal Operations	3	0.0		6.5	V	
V_{PPH}	V_{PP} during Prog/Erase Operations		11.4	12.0	12.6	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	
V_{HH}	$\overline{PWD}$, $\overline{OE}$ Unlock Voltage		11.4		12.6	V	Boot Block Prog/Erase

DC CHARACTERISTICS V_{CC} = 5.0V ±10%, T_A = −40°C to +85°C

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Conditions
I_{IL}	Input Load Current	1			±1.0	μA	V_{CC} = V_{CC} Max V_{IN} = V_{CC} or GND
I_{LO}	Output Leakage Current	1			±10	μA	V_{CC} = V_{CC} Max V_{OUT} = V_{CC} or GND
I_{CCS}	V_{CC} Standby Current			1.2	2.0	mA	V_{CC} = V_{CC} Max $\overline{CE}$ = $\overline{PWD}$ = V_{IH}
				30	150	μA	V_{CC} = V_{CC} Max $\overline{CE}$ = $\overline{PWD}$ = V_{CC} ±0.2V

DC CHARACTERISTICS V_{CC} = 5.0V ±10%, T_A = −40°C to +85°C (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Conditions
I_{CCD}	V_{CC} Deep PowerDown Current	1		0.05	2.0	μA	$\overline{PWD}$ = GND ±0.2V
I_{CCR}	V_{CC} Read Current	1		13	35	mA	V_{CC} = V_{CC} Max, $\overline{CE}$ = V_{IL} f = 8 MHz, I_{OUT} = 0 mA
I_{CCP}	V_{CC} Programming Current	1		5	20	mA	Programming in Progress
I_{CCE}	V_{CC} Erase Current	1		6	20	mA	Erase in Progress
I_{CCES}	V_{CC} Erase Suspend Current	1, 2		5	10	mA	Erase Suspended $\overline{CE}$ = V_{IH}
I_{PPS}	V_{PP} Standby Current	1		±1	±10	μA	$V_{PP} \leq V_{CC}$
				90	400	μA	$V_{PP} > V_{CC}$
I_{PPD}	V_{PP} Deep PowerDown Current	1		0.80	1.0	μA	$\overline{PWD}$ = GND ±0.2V
I_{PPP}	V_{PP} Programming Current	1		6	30	μA	V_{PP} = V_{PPH} Programming in Progress
I_{PPE}	V_{PP} Erase Current	1		6	30	mA	V_{PP} = V_{PPH} Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1		90	400	μA	V_{PP} = V_{PPH} Erase Suspended
I_{ID}	A_9 Intelligent Identifier Current	1		90	500	μA	A_9 = V_{ID}
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	V_{CC} = V_{CC} Min I_{OL} = 5.8 mA
V_{OH}	Output High Voltage		2.4			V	V_{CC} = V_{CC} Min I_{OH} = 2.5 mA
V_{ID}	A_9 Intelligent Identifier Voltage		11.5		13.0	V	
V_{PPL}	V_{PP} during Normal Operations	3	0.0		6.5	V	
V_{PPH}	V_{PP} during Prog/Erase Operations		11.4	12.0	12.6	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	
V_{HH}	$\overline{PWD}$, $\overline{OE}$ Unlock Voltage		11.4		12.6	V	Boot Block Prog/Erase

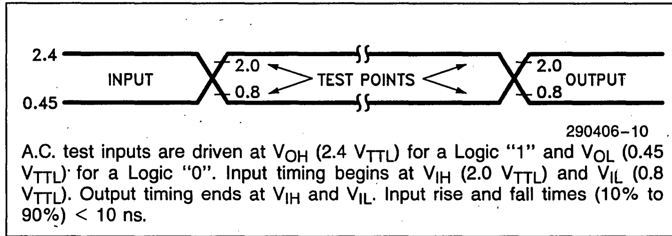
CAPACITANCE(4) T_A = 25°C, f = 1 MHz

Symbol	Parameter	Max	Unit	Conditions
C_{IN}	Input Capacitance	8	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	12	pF	V_{OUT} = 0V

NOTES:

1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T_A = 25°C. These currents are valid for all product versions (packages and speeds).
2. I_{CCES} is specified with the device deselected. If the 28F001BX is read while in Erase Suspend mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Erase/Programs are inhibited when V_{PP} = V_{PPL} and not guaranteed in the range between V_{PPH} and V_{PPL}.
4. Sampled, not 100% tested.

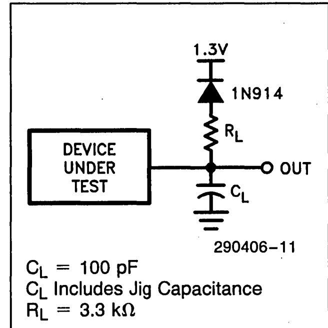
AC INPUT/OUTPUT REFERENCE WAVEFORM



290406-10

A.C. test inputs are driven at V_{OH} (2.4 V_{TTL}) for a Logic "1" and V_{OL} (0.45 V_{TTL}) for a Logic "0". Input timing begins at V_{IH} (2.0 V_{TTL}) and V_{IL} (0.8 V_{TTL}). Output timing ends at V_{IH} and V_{IL}. Input rise and fall times (10% to 90%) < 10 ns.

AC TESTING LOAD CIRCUIT



290406-11

C_L = 100 pF
C_L Includes Jig Capacitance
R_L = 3.3 kΩ

AC CHARACTERISTICS—Read-Only Operations(1)

Versions(2)		V_{CC} ± 10%		E28F001BX-120 N28F001BX-120 P28F001BX-120		E28F001BX-150 TE28F001BX-150 N28F001BX-150 TN28F001BX-150 P28F001BX-150		Unit
Symbol		Parameter	Notes	Min	Max	Min	Max	
t_{AVAV}	t_{RC}	Read Cycle Time		120		150		ns
t_{AVQV}	t_{ACC}	Address to Output Delay			120		150	ns
t_{ELQV}	t_{CE}	$\overline{CE}$ to Output Delay	3		120		150	ns
t_{PHQV}	t_{PWH}	$\overline{PWD}$ High to Output Delay			600		600	ns
t_{GLQV}	t_{OE}	$\overline{OE}$ to Output Delay	3		50		55	ns
t_{ELQX}	t_{LZ}	$\overline{CE}$ to Output Low Z	4	0		0		ns
t_{EHQZ}	t_{HZ}	$\overline{CE}$ High to Output High Z	4		55		55	ns
t_{GLQX}	t_{OLZ}	$\overline{OE}$ to Output Low Z	4	0		0		ns
t_{GHQZ}	t_{DF}	$\overline{OE}$ High to Output High Z	4		30		30	ns
	t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change, Whichever is First	4	0		0		ns

NOTES:
1. See AC Input/Output Reference Waveform for timing measurements.
2. Model Number Prefixes: E = TSOP (Standard Pinout), N = PLCC, P = PDIP, T = Extended Temperature.
3. $\overline{OE}$ may be delayed up to t_{CE}−t_{OE} after the falling edge of $\overline{CE}$ without impact on t_{CE}.
4. Sampled, not 100% tested.

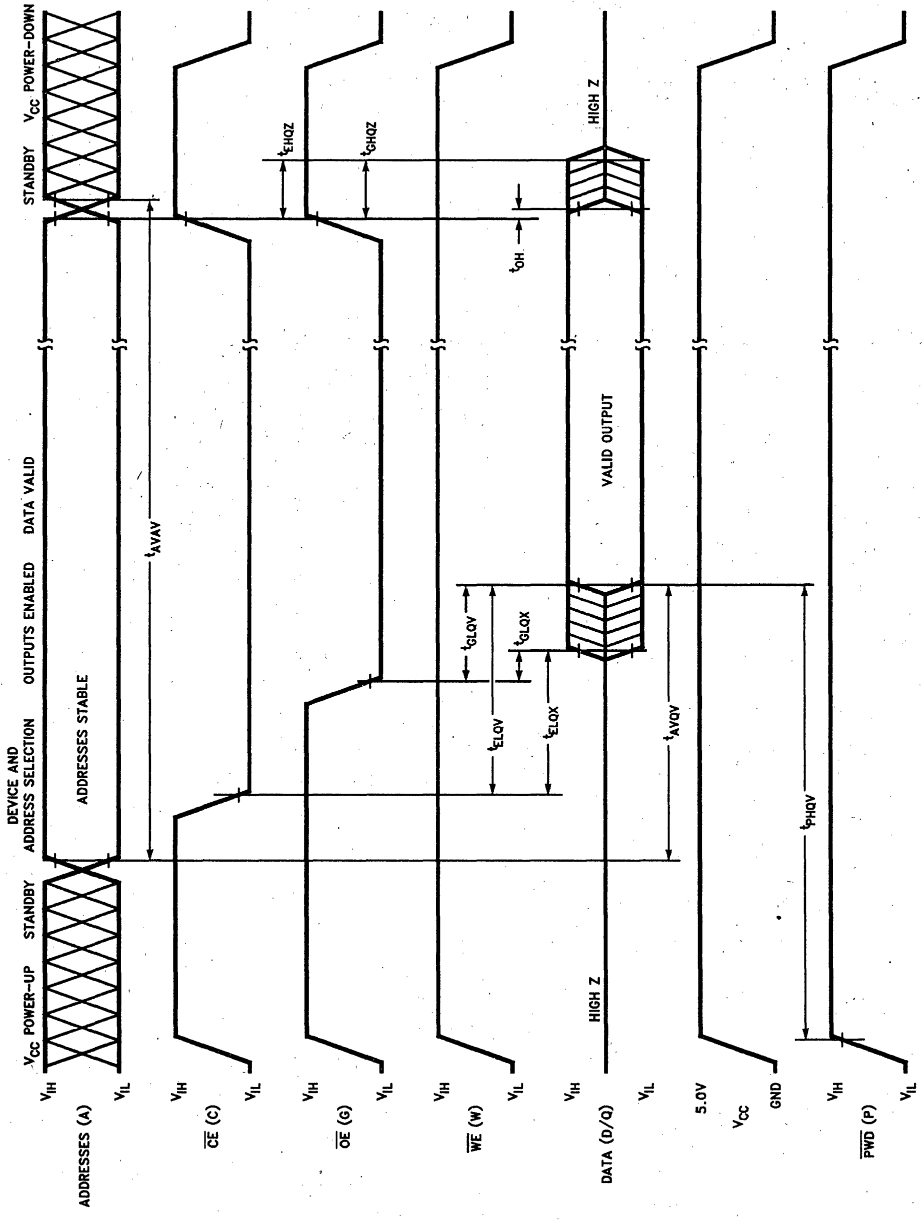


Figure 12. AC Waveform for Read Operations

AC CHARACTERISTICS—Write/Erase/Program Operations(1)

Versions		V_{CC} ±10%		28F001BX-120		28F001BX-150		Unit
Symbol		Parameter	Notes	Min	Max	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		120		150		ns
t_{PHWL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{WE}$ Going Low	2	480		480		ns
t_{ELWL}	t_{CS}	$\overline{CE}$ Setup to $\overline{WE}$ Going Low		10		10		ns
t_{WLWH}	t_{WP}	$\overline{WE}$ Pulse Width		50		50		ns
t_{PHHWH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{WE}$ Going High	2	100		100		ns
t_{VPWH}	t_{VPS}	V_{PP} Setup to $\overline{WE}$ Going High	2	100		100		ns
t_{AVWH}	t_{AS}	Address Setup to $\overline{WE}$ Going High	3	50		50		ns
t_{DVWH}	t_{DS}	Data Setup to $\overline{WE}$ Going High	4	50		50		ns
t_{WHDX}	t_{DH}	Data Hold from $\overline{WE}$ High		10		10		ns
t_{WHAX}	t_{AH}	Address Hold from $\overline{WE}$ High		10		10		ns
t_{WHEH}	t_{CH}	$\overline{CE}$ Hold from $\overline{WE}$ High		10		10		ns
t_{WHWL}	t_{WPH}	$\overline{WE}$ Pulse Width High		50		50		ns
t_{WHQV1}		Duration of Programming Operation	5, 6, 7	15		15		μs
t_{WHQV2}		Duration of Erase Operation (Boot)	5, 6, 7	1.3		1.3		sec
t_{WHQV3}		Duration of Erase Operation (Parameter)	5, 6, 7	1.3		1.3		sec
t_{WHQV4}		Duration of Erase Operation (Main)	5, 6, 7	3.0		3.0		sec
t_{WHGL}		Write Recovery before Read		0		0		μs
t_{QVVL}	t_{VPH}	V_{PP} Hold from Valid SRD	2, 6	0		0		ns
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	2, 7	0		0		ns
t_{PHBR}		Boot-Block Relock Delay	2		100		100	ns

PROM Programmer Specifications

Versions		V_{CC} ±10%		28F001BX-120		28F001BX-150		Unit
Symbol		Parameter	Notes	Min	Max	Min	Max	
t_{GHHWL}		$\overline{OE}$ V_{HH} Setup to $\overline{WE}$ Going Low	2, 8	480		480		ns
t_{WHGH}		$\overline{OE}$ V_{HH} Hold from $\overline{WE}$ High	2, 8	480		480		ns

NOTES:
1. Read timing characteristics during erase and program operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Sampled, not 100% tested.
3. Refer to Table 3 for valid A_{IN} for byte programming or block erasure.
4. Refer to Table 3 for valid D_{IN} for byte programming or block erasure.
5. The on-chip Write State Machine incorporates all program and erase system functions and overhead of standard Intel flash memory, including byte program and verify (programming) and block precondition, precondition verify, erase and erase verify (erasing).
6. Program and erase durations are measured to completion (SR.7 = 1). V_{PP} should be held at V_{PPH} until determination of program/erase success (SR.3/4/5 = 0).
7. For boot block programming and erasure, $\overline{PWD}$ should be held at V_{HH} until determination of program/erase success (SR.3/4/5 = 0).
8. Alternate boot block access method.

ERASE AND PROGRAMMING PERFORMANCE

Parameter	Notes	28F001BX-120			28F001BX-150			Unit
		Min	Typ(1)	Max	Min	Typ(1)	Max	
Boot Block Erase Time	2		2.10	14.9		2.10	14.9	Sec
Boot Block Program Time	2		0.15	0.52		0.15	0.52	Sec
Parameter Block Erase Time	2		2.10	14.6		2.10	14.6	Sec
Parameter Block Program Time	2		0.07	0.26		0.07	0.26	Sec
Main Block Erase Time	2		3.80	20.9		3.80	20.9	Sec
Main Block Program Time	2		2.10	7.34		2.10	7.34	Sec
Chip Erase Time	2		10.10	65		10.10	65	Sec
Chip Program Time	2		2.39	8.38		2.39	8.38	Sec
Erase/Program Cycles (per Block)		10,000	100,000		10,000	100,000		Cycles

NOTES:
1. 25°C, 12.0 V_{PP}.
2. Excludes System-Level Overhead.

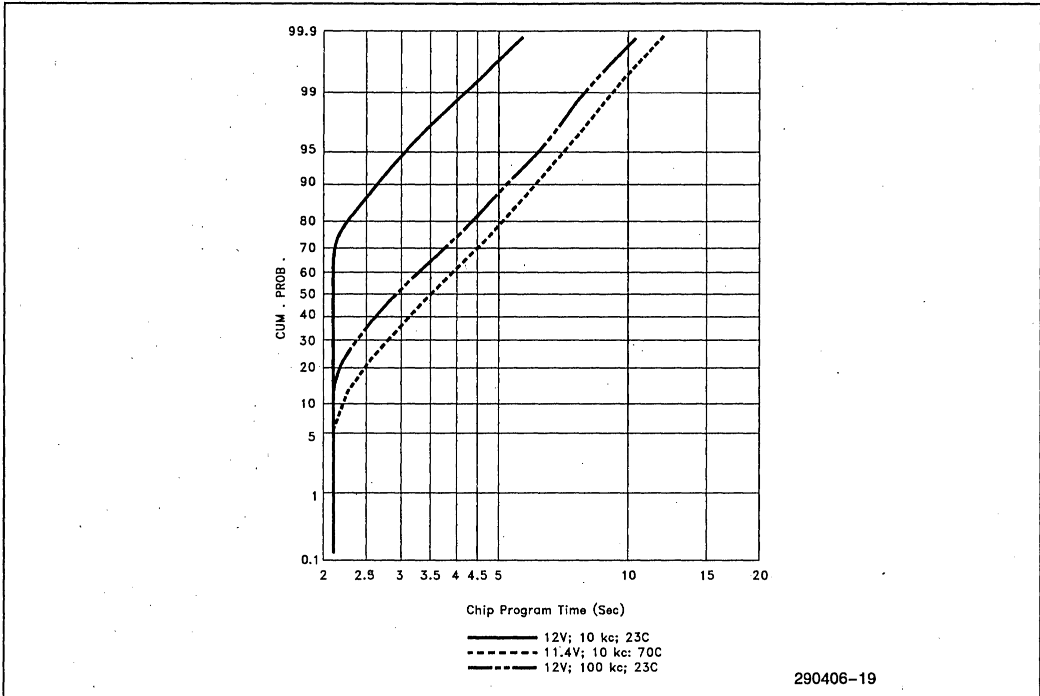


Figure 13. 28F001BX Typical Programming Capability

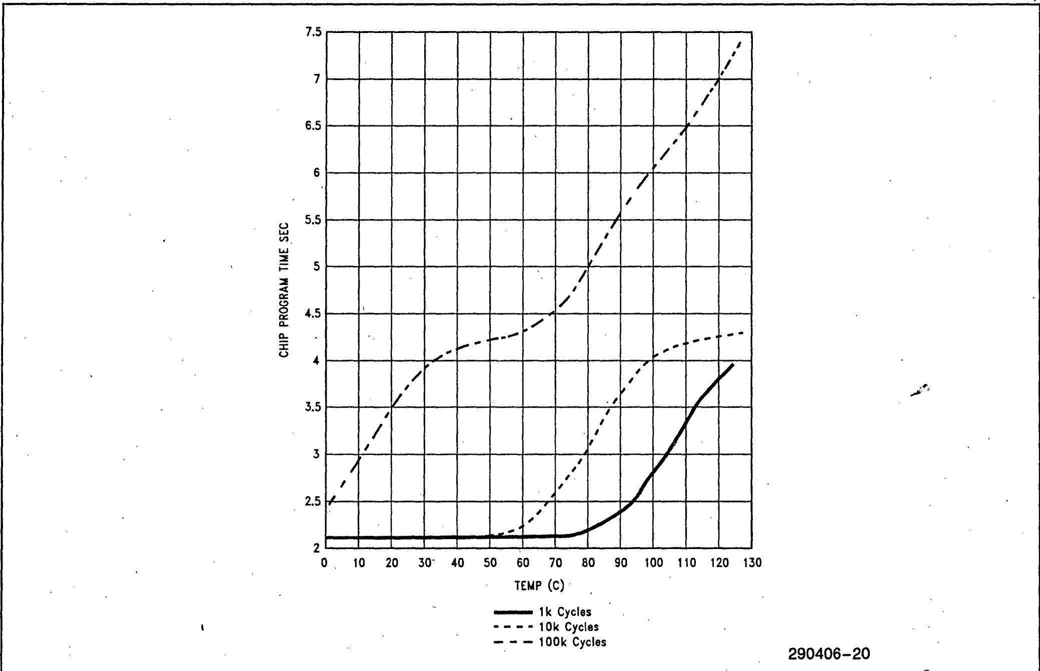


Figure 14. 28F001BX Typical Programming Time at 12V

3

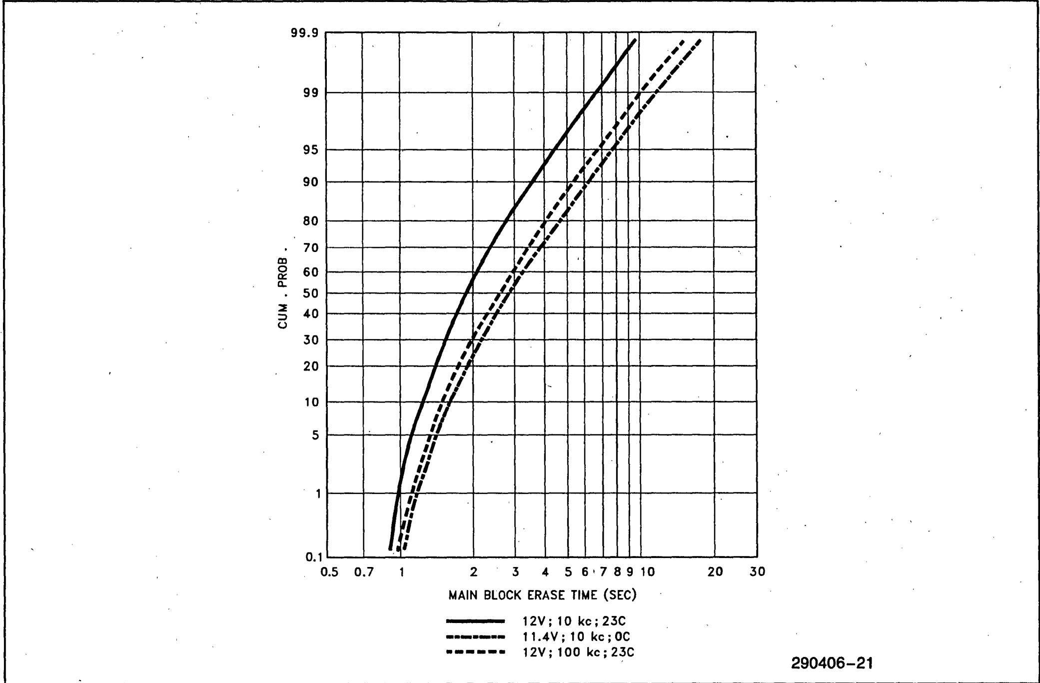


Figure 15. 28F001BX Typical Erase Capability

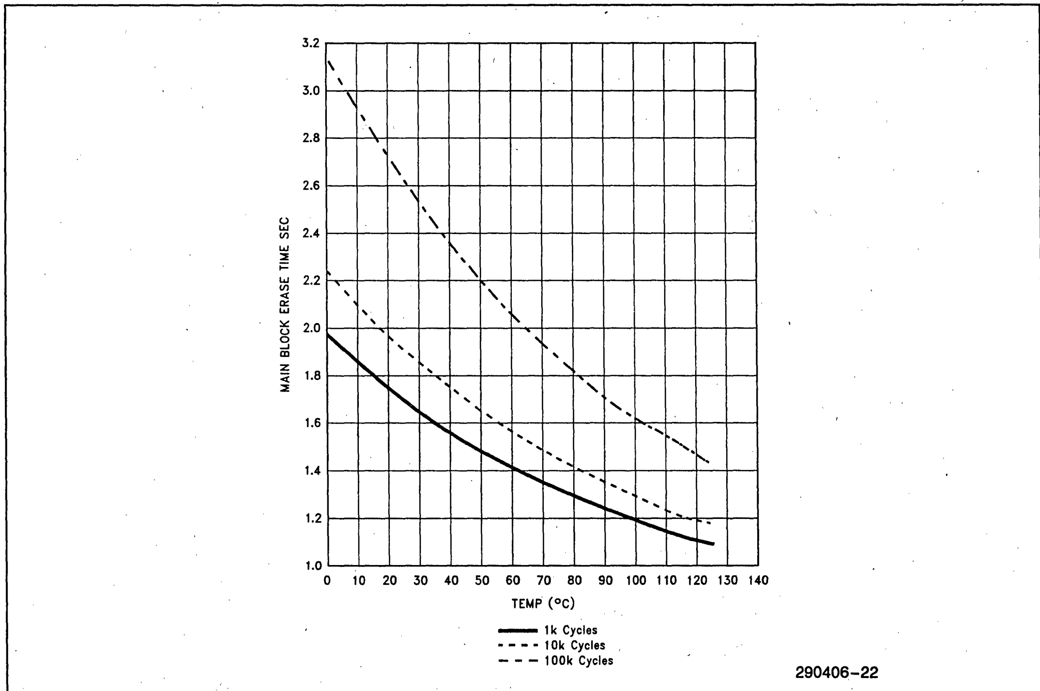


Figure 16. 28F001BX Typical Erase Time at 12V

VCC POWER-UP & STANDBY | WRITE PROGRAM OR ERASE SETUP COMMAND | WRITE VALID ADDRESS & DATA (PROGRAM) OR ERASE CONFIRM COMMAND | AUTOMATED PROGRAM OR ERASE DELAY | READ STATUS REGISTER DATA | WRITE READ ARRAY COMMAND

ADDRESSES (A)
V_{IH}
V_{IL}
A_{IN}
A_{IN}
t_{AVAV}
t_{AVWH}
t_{WHAX}

$\overline{CE}$ (E)
V_{IH}
V_{IL}
t_{ELWL}
t_{WHEH}
t_{WHGL}

$\overline{OE}$ (G)
V_{IH}
V_{IL}
t_{WHWL}
$t_{WHQV1,2,3,4}$

$\overline{WE}$ (W)
V_{IH}
V_{IL}
t_{WLWH}
t_{DVWH}
t_{WHDX}

DATA (D/Q)
V_{IH}
V_{IL}
HIGH Z
D_{IN}
D_{IN}
VALID SRD
D_{IN}
t_{PHWL}
t_{PHHWH}
t_{QVPH}

$\overline{PWD}$ (P)
V_{HH}
6.5V
V_{IH}
V_{IL}
t_{VPWH}
t_{QVVL}

V_{PP} (V)
V_{PPH}
V_{PPL}
V_{IH}
V_{IL}

290406–13

Figure 17. AC Waveform for Write Operations

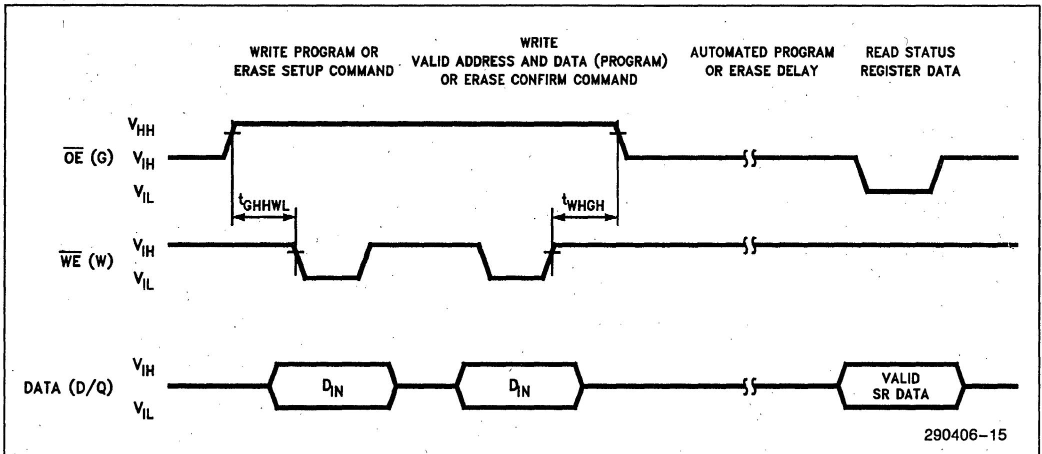


Figure 18. Alternate Boot Block Access Method Using $\overline{OE}$

ALTERNATE $\overline{CE}$-CONTROLLED WRITES(1)

Versions		V_{CC} ±10%		28F001BX-120		28F001BX-150		Unit
Symbol		Parameter	Notes	Min	Max	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		120		150		ns
t_{PHEL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{CE}$ Going Low	2	480		480		ns
t_{WLEL}	t_{WS}	$\overline{WE}$ Setup to $\overline{CE}$ Going low		0		0		ns
t_{ELEH}	t_{CP}	$\overline{CE}$ Pulse Width		70		70		ns
t_{PHHEH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{CE}$ Going High	2	100		100		ns
t_{VPEH}	t_{VPS}	V_{PP} Setup to $\overline{CE}$ Going High	2	100		100		ns
t_{AVEH}	t_{AS}	Address Setup to $\overline{CE}$ Going High	3	50		50		ns
t_{DVEH}	t_{DS}	Data Setup to $\overline{CE}$ Going High	4	50		50		ns
t_{EHDX}	t_{DH}	Data Hold from $\overline{CE}$ High		10		10		ns
t_{EHAX}	t_{AH}	Address Hold from $\overline{CE}$ High		15		15		ns
t_{EHWH}	t_{WH}	$\overline{WE}$ Hold from $\overline{CE}$ High		0		0		ns
t_{EHEL}	t_{EPH}	$\overline{CE}$ Pulse Width High		25		25		ns
t_{EHQV1}		Duration of Programming Operation	5, 6	15		15		µs
t_{EHQV2}		Duration of Erase Operation (Boot)	5, 6	1.3		1.3		sec
t_{EHQV3}		Duration of Erase Operation (Parameter)	5, 6	1.3		1.3		sec
t_{EHQV4}		Duration of Erase Operation (Main)	5, 6	3.0		3.0		sec
t_{EHGL}		Write Recovery before Read		0		0		µs
t_{QVVL}	t_{VPH}	V_{PP} Hold from Valid SRD	2, 5	0		0		ns
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	2, 6	0		0		ns
t_{PHBR}		Boot-Block Relock Delay	2		100		10	ns

PROM Programmer Specifications

Versions	V_{CC} ±10%		28F001BX-120		28F001BX-150		Unit
Symbol	Parameter	Notes	Min	Max	Min	Max	
t_{GHHEL}	$\overline{OE}$ V_{HH} Setup to $\overline{CE}$ Going Low	2, 7	480		480		ns
t_{EHGH}	$\overline{OE}$ V_{HH} Hold from $\overline{CE}$ High	2, 7	480		480		ns

NOTES:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of $\overline{CE}$ and $\overline{WE}$. In systems where $\overline{CE}$ defines the write pulse width (within a longer $\overline{WE}$ timing waveform), all set-up, hold and inactive $\overline{WE}$ times should be measured relative to the $\overline{CE}$ waveform.
2. Sampled, not 100% tested.
3. Refer to Table 3 for valid A_{IN} for byte programming or block erasure.
4. Refer to Table 3 for valid D_{IN} for byte programming or block erasure.
5. Program and erase durations are measured to completion (SR.7 = 1). V_{PP} should be held at V_{PPH} until determination of program/erase success (SR.3/4/5 = 0).
6. For boot block programming and erasure, $\overline{PWD}$ should be held at V_{HH} until determination of program/erase success (SR.3/4/5 = 0).
7. Alternate boot block access method.

3

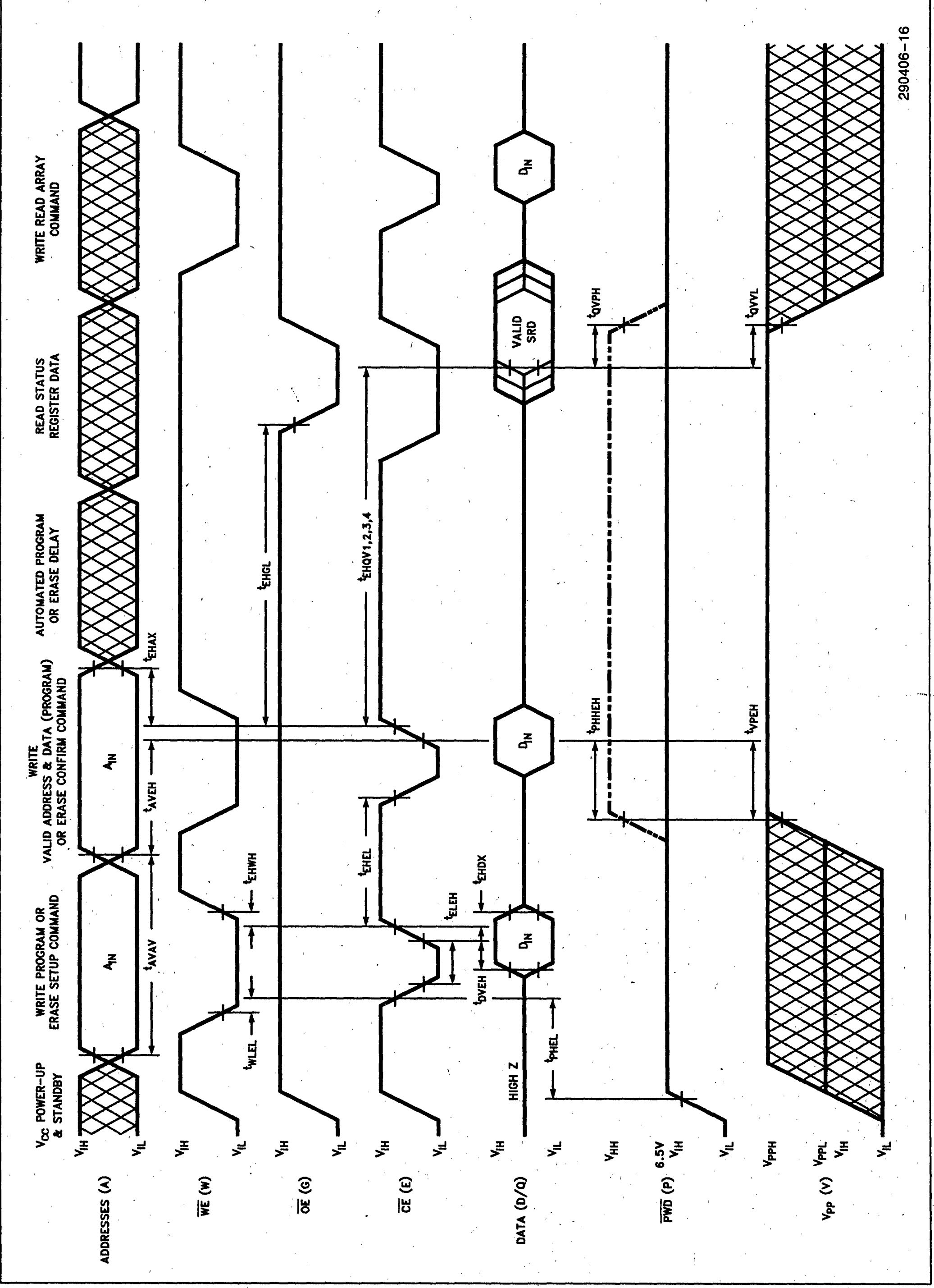


Figure 19. Alternate AC Waveform for Write Operations

ORDERING INFORMATION

T	P	2	8	F	0	0	1	B	X	-	T	1	2	0

VALID COMBINATIONS:

E28F001BX-T120	N28F001BX-T120	P28F001BX-T120
E28F001BX-T150	N28F001BX-T150	P28F001BX-T150
TE28F001BX-T150	TN28F001BX-T150	
E28F001BX-B120	N28F001BX-B120	P28F001BX-B120
E28F001BX-B150	N28F001BX-B150	P28F001BX-B150
TE28F001BX-B150	TN28F001BX-B150	

ADDITIONAL INFORMATION

		Order Number
ER-20	"ETOX™ II Flash Memory Technology"	294005
RR-60	"ETOX™ II Flash Memory Reliability Data Summary"	293002
AP-316	"Using Flash Memory for In-System Reprogrammable Nonvolatile Storage"	292046
AP-341	"Designing an Updatable BIOS Using Flash Memory"	292077

REVISION HISTORY

Number	Description
-004	Removed **Preliminary** classification. Latched address A_{16} in Figure 5. Updated Boot Block Program and Erase section: "If boot block program or erase is attempted while $\overline{PWD}$ is at V_{IH}, **either the Program Status or Erase Status bit will be set to "1", reflective of the operation being attempted and indicating boot block lock."** Updated Figure 11, 28F001BX Erase Suspend/Resume Flowchart Added DC Characteristics typical current values Combined V_{PP} Standby current and V_{PP} Read current into one V_{PP} Standby current spec with two test conditions (DC Characteristics table) Added maximum program/erase times to Erase and Programming Performance table. Added Figures 13–16 Added Extended Temperature proliferations

28F200BX-T/B, 28F002BX-T/B 2 MBIT (128K x 16, 256K x 8) BOOT BLOCK FLASH MEMORY FAMILY

- x8/x16 Input/Output Architecture
  - 28F200BX-T, 28F200BX-B
  - For High Performance and High Integration 16-bit and 32-bit CPUs
- x8-only Input/Output Architecture
  - 28F002BX-T 28F002BX-B
  - For Space Constrained 8-bit Applications
- Optimized High Density Blocked Architecture
  - One 16 KB Protected Boot Block
  - Two 8 KB Parameter Blocks
  - One 96 KB Main Block
  - One 128 KB Main Block
  - Top or Bottom Boot Locations
- Extended Cycling Capability
  - 100,000 Block Erase Cycles
- Automated Word/Byte Write and Block Erase
  - Command User Interface
  - Status Registers
  - Erase Suspend Capability
- SRAM-Compatible Write Interface
- Automatic Power Savings Feature
  - 1 mA Typical I_{CC} Active Current in Static Operation
- Hardware Data Protection Feature
  - Erase/Write Lockout during Power Transitions
- Very High-Performance Read
  - 60/80 ns Maximum Access Time
  - 30/40 ns Maximum Output Enable Time
- Low Power Consumption
  - 20 mA Typical x8 Active Read Current
  - 25 mA Typical x16 Active Read Current
- Deep Power-Down/Reset Input
  - 0.2 μA I_{CC} Typical
  - Acts as Reset for Boot Operations
- Extended Temperature Operation
  - −40°C to +85°C
- Write Protection for Boot Block
- Industry Standard Surface Mount Packaging
  - 28F200BX: JEDEC ROM Compatible 44-Lead PSOP, 56-Lead TSOP
  - 28F002BX: 40-Lead TSOP
- 12V Word/Byte Write and Block Erase
  - V_{PP} = 12V ±5% Standard
  - V_{PP} = 12V ±10% Option
- ETOX™ III Flash Technology
  - 5V Read
- Independent Software Vendor Support
  - SystemSoft* Flash BIOS

*SystemSoft is a trademark of SystemSoft Corporation.

November 1992
Order Number: 290448-002

Intel's 2 Mbit Flash Memory Family is an extension of the Boot Block Architecture which includes block-selective erasure, automated write and erase operations and standard microprocessor interface. The 2 Mbit Flash Memory Family enhances the Boot Block Architecture by adding more density and blocks, x8/x16 input/output control, very high speed, low power, an industry standard ROM compatible pinout and surface mount packaging. The 2 Mbit flash family allows for an easy upgrade to Intel's 4 Mbit Boot Block Flash Memory Family.

The Intel 28F200BX-T/B are 16-bit wide flash memory offerings. These high density flash memories provide user selectable bus operation for either 8-bit or 16-bit applications. The 28F200BX-T and 28F200BX-B are 2,097,152-bit non-volatile memories organized as either 262,144 bytes or 131,072 words of information. They are offered in 44-Lead plastic SOP and 56-Lead TSOP packages. The x8/x16 pinout conforms to the industry standard ROM/EPROM pinout.

The Intel 28F002BX-T/B are 8-bit wide flash memories with 2,097,152 bits organized as 262,144 bytes of information. They are offered in a 40-Lead TSOP package, which is ideal for space-constrained portable systems.

These devices use an integrated Command User Interface (CUI) and Write State Machine (WSM) for simplified word/byte write and block erasure. The 28F200BX-T/28F002BX-T provide block locations compatible with Intel's MCS-186 family, 80286, i386™, i486™, i860™ and 80960CA microprocessors. The 28F200BX-B/28F002BX-B provide compatibility with Intel's 80960KX and 80960SX families as well as other embedded microprocessors.

The boot block includes a data protection feature to protect the boot code in critical applications. With a maximum access time of 60 ns, these 2 Mbit flash devices are very high performance memories which interface at zero-wait-state to a wide range of microprocessors and microcontrollers. A deep power-down mode lowers the total V_{CC} power consumption to 1 μW typical. This is critical in handheld battery powered systems. For very low power applications using a 3.3V supply, refer to the Intel 28F200BX-TL/BL, 28F002BX-TL/BL 2 Mbit Boot Block Flash Memory Family datasheet.

Manufactured on Intel's 0.8 micron ETOX™III process, the 2 Mbit flash memory family provides world class quality, reliability and cost-effectiveness at the 2 Mbit density level.

1.0 PRODUCT FAMILY OVERVIEW

Throughout this datasheet the 28F200BX refers to both the 28F200BX-T and 28F200BX-B devices and 28F002BX refers to both the 28F002BX-T and 28F002BX-B devices. The 2 Mbit flash memory family refers to both the 28F200BX and 28F002BX products. This datasheet comprises the specifications for four separate products in the 2 Mbit flash memory family. Section 1 provides an overview of the 2 Mbit flash memory family including applications, pinouts and pin descriptions. Sections 2 and 3 describe in detail the specific memory organizations for the 28F200BX and 28F002BX products respectively. Section 4 combines a description of the family's principles of operations. Finally Section 5 describes the family's operating specifications.

PRODUCT FAMILY

x8/x16 Products	x8-Only Products
28F200BX-T	28F002BX-T
28F200BX-B	28F002BX-B

1.1 Main Features

The 28F200BX/28F002BX boot block flash memory family is a very high performance 2 Mbit (2,097,152 bit) memory family organized as either 128 KWords (131,072 words) of 16 bits each or 256 Kbytes (262,144 bytes) of 8 bits each.

Five Separately Erasable Blocks including a **hardware-lockable boot block** (16,384 Bytes), **two parameter blocks** (8,192 Bytes each) and **two main blocks** (1 block of 98,304 Bytes and 1 block of 131,072 Bytes) are included on the 2 Mbit family. An erase operation erases one of the main blocks in typically 2.4 seconds, and the boot or parameter blocks in typically 1.0 second. Each block can be independently erased and programmed 100,000 times.

The Boot Block is located at either the top (28F200BX-T, 28F002BX-T) or the bottom (28F200BX-B, 28F002BX-B) of the address map in order to accommodate different microprocessor protocols for boot code location. The **hardware lockable boot block** provides the most secure code storage. The boot block is intended to store the kernel code required for booting-up a system. When the $\overline{PWD}$ pin is between 11.4V and 12.6V the boot block is unlocked and program and erase operations can be performed. When the $\overline{PWD}$ pin is at or below 6.5V the boot block is locked and program and erase operations to the boot block are ignored.

The 28F200BX products are available in the ROM/EPROM compatible pinout and housed in the 44-Lead PSOP (Plastic Small Outline) package and the 56-Lead TSOP (Thin Small Outline, 1.2mm thick) package as shown in Figures 3 and 4. The 28F002BX products are available in the 40-Lead TSOP (1.2mm thick) package as shown in Figure 5.

The **Command User Interface (CUI)** serves as the interface between the microprocessor or microcontroller and the internal operation of the 28F200BX and 28F002BX flash products.

Program and Erase Automation allows program and erase operations to be executed using a two-write command sequence to the CUI. The internal Write State Machine (WSM) automatically executes the algorithms and timings necessary for program and erase operations, including verifications, thereby unburdening the microprocessor or microcontroller. Writing of memory data is performed in word or byte increments for the 28F200BX family and in byte increments for the 28F002BX family typically within 9 μs which is a 100% improvement over current flash memory products.

The **Status Register (SR)** indicates the status of the WSM and whether the WSM successfully completed the desired program or erase operation.

Maximum Access Time of **60 ns (TACC)** is achieved over the commercial temperature range (0°C to 70°C), 5% V_{CC} supply voltage range (4.75V to 5.25V) and 30 pF output load. Refer to Figure 19; T_{ACC} vs Output Load Capacitance for larger output loads. Maximum Access Time of **80 ns (TACC)** is achieved over the commercial temperature range, 10% V_{CC} supply range (4.5V to 5.5V) and 100 pF output load.

I_{PP} maximum Program current is 40 mA for x16 operation and 30 mA for x8 operation. I_{PP} Erase current is 30 mA maximum. V_{PP} erase and programming voltage is 11.4V to 12.6V (V_{PP} = 12V ±5%) under all operating conditions. As an option, V_{PP} can also vary between 10.8V to 13.2V (V_{PP} = 12V ±10%) with a guaranteed number of 100 block erase cycles.

Typical I_{CC} Active Current of 25 mA is achieved for the x16 products (28F200BX), **typical I_{CC} Active Current of 20 mA** is achieved for the x8 products (28F200BX, 28F002BX). Refer to the I_{CC} active current derating curves in this datasheet.

The 2 Mbit boot block flash family is also designed with an Automatic Power Savings (APS) feature to minimize system battery current drain and allow for very low power designs. Once the device is ac-

cessed to read array data, APS mode will immediately put the memory in static mode of operation where I_{CC} active current is typically 1 mA until the next read is initiated.

When the $\overline{CE}$ and $\overline{PWD}$ pins are at V_{CC} and the $\overline{BYTE}$ pin (28F200BX-only) is at either V_{CC} or GND the **CMOS Standby** mode is enabled where I_{CC} is typically 50 μA.

A **Deep Power-Down Mode** is enabled when the $\overline{PWD}$ pin is at ground minimizing power consumption and providing write protection during power-up conditions. **I_{CC} current** during deep power-down mode is **0.20 μA typical.** An initial maximum access time or Reset Time of 300 ns is required from $\overline{PWD}$ switching until outputs are valid. Equivalently, the device has a maximum wake-up time of 215 ns until writes to the Command User Interface are recognized. When $\overline{PWD}$ is at ground the WSM is reset, the Status Register is cleared and the entire device is protected from being written to. This feature prevents data corruption and protects the code stored in the device during system reset. The system Reset pin can be tied to $\overline{PWD}$ to reset the memory to normal read mode upon activation of the Reset pin. With on-chip program/erase automation in the 2 Mbit family and the $\overline{PWD}$ functionality for data protection, when the CPU is reset and even if a program or erase command is issued, the device will not recognize any operation until $\overline{PWD}$ returns to its normal state.

For the 28F200BX, Byte-wide or Word-wide Input/Output Control is possible by controlling the $\overline{BYTE}$ pin. When the $\overline{BYTE}$ pin is at a logic low the device is in the byte-wide mode (x8) and data is read and written through DQ[0:7]. During the byte-wide mode, DQ[8:14] are tri-stated and DQ15/A−1 becomes the lowest order address pin. When the $\overline{BYTE}$ pin is at a logic high the device is in the word-wide mode (x16) and data is read and written through DQ[0:15].

1.2 Applications

The 2 Mbit boot block flash family combines high density, high performance, cost-effective flash memories with blocking and hardware protection capabilities. Its flexibility and versatility will reduce costs throughout the product life cycle. Flash memory is ideal for Just-In-Time production flow, reducing system inventory and costs, and eliminating component handling during the production phase. During the product life cycle, when code updates or feature enhancements become necessary, flash memory will reduce the update costs by allowing either a user-performed code change via floppy disk or a remote code change via a serial link. The 2 Mbit boot block flash family provides full function, blocked flash memories suitable for a wide range of applications. These applications include **Extended PC BIOS, Digital Cellular Phone** program and data storage, **Telecommunication** boot/firmware, and various other embedded applications where both program and data storage are required.

Reprogrammable systems such as personal computers, are ideal applications for the 2 Mbit flash products. Portable and handheld personal computer applications are becoming more complex with the addition of power management software to take advantage of the latest microprocessor technology, the availability of ROM-based application software, pen tablet code for electronic hand writing, and diagnostic code. Figure 1 shows an example of a 28F200BX-T application.

This increase in software sophistication augments the probability that a code update will be required after the PC is shipped. The 2 Mbit flash products provide an inexpensive update solution for the notebook and handheld personal computers while extending their product lifetime. Furthermore, the 2 Mbit flash products' power-down mode provides added flexibility for these battery-operated portable designs which require operation at very low power levels.

The 2 Mbit flash products also provide excellent design solutions for Digital Cellular Phone and Telecommunication switching applications requiring high performance, high density storage capability coupled with modular software designs, and a small form factor package (x8-only bus). The 2 Mbit's blocking scheme allows for an easy segmentation of the embedded code with; 16 Kbytes of Hardware-Protected Boot code, 2 Main Blocks of program code and 2 Parameter Blocks of 8 Kbytes each for frequently updatable data storage and diagnostic messages (e.g. phone numbers, authorization codes). Figure 2 is an example of such an application with the 28F002BX-T.

These are a few actual examples of the wide range of applications for the 2 Mbit Boot Block flash memory family which enable system designers to achieve the best possible product design. Only your imagination limits the applicability of such a versatile product family.

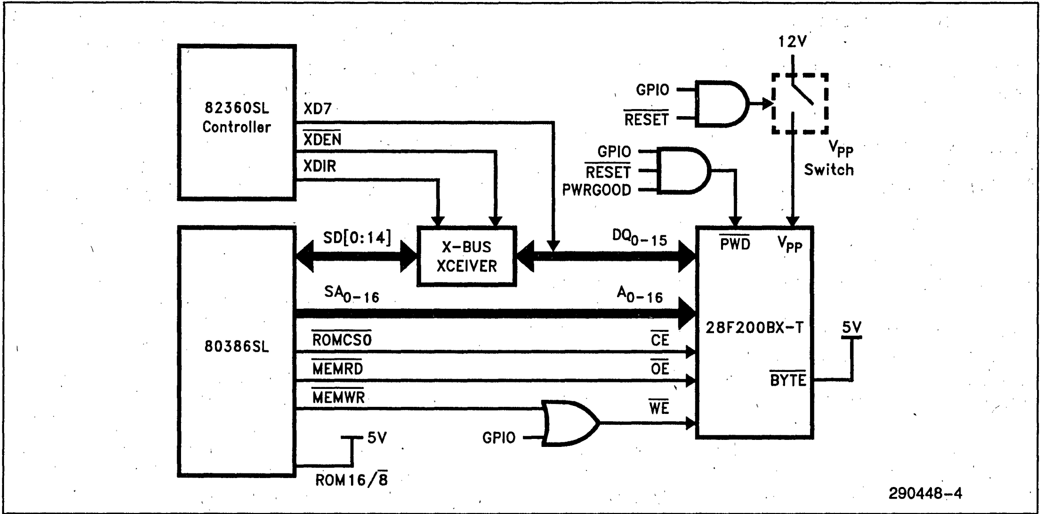


Figure 1. 28F200BX Interface to INTEL386SL™ Microprocessor Superset

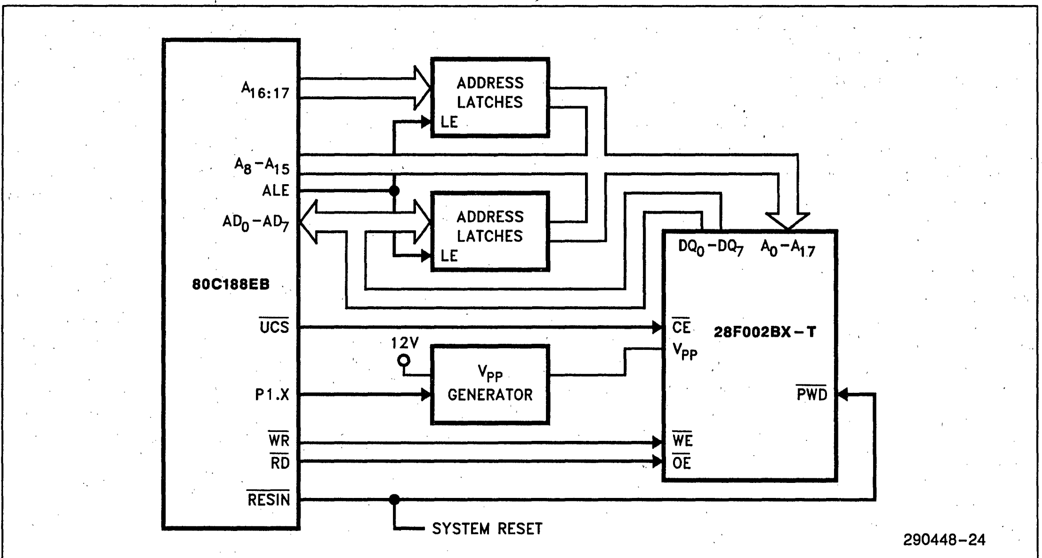


Figure 2. 28F002BX Interface to INTEL 80C188EB 8-Bit Embedded Microprocessor

1.3 Pinouts

The 28F200BX 44-Lead PSOP pinout follows the industry standard ROM/EPROM pinout as shown in Figure 3 with an upgrade to the 28F400BX (4 Mbit flash family). Furthermore, the 28F200BX 56-Lead TSOP pinout shown in Figure 4 provides density upgrades to the 28F400BX and to future higher density boot block memories.

The 28F002BX 40-Lead TSOP pinout shown in Figure 5 is 100% compatible and provides a density upgrade to the 28F004BX 4 Mbit Boot Block flash memory.

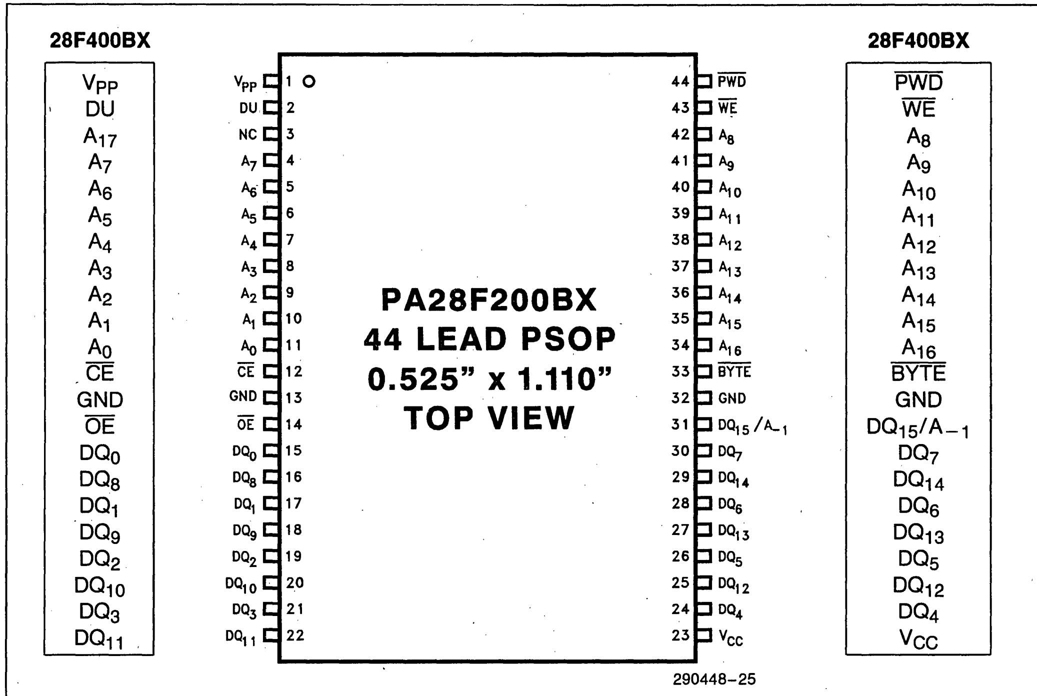


Figure 3. PSOP Lead Configuration for x8/x16 28F200BX

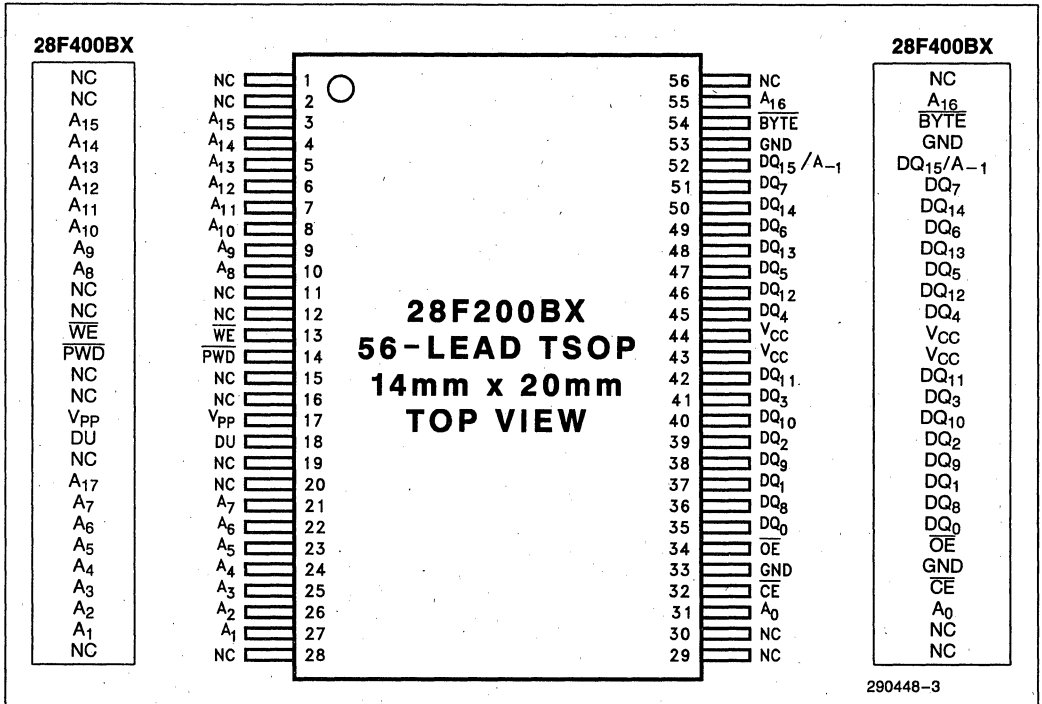


Figure 4. TSOP Lead Configuration for x8/x16 28F200BX

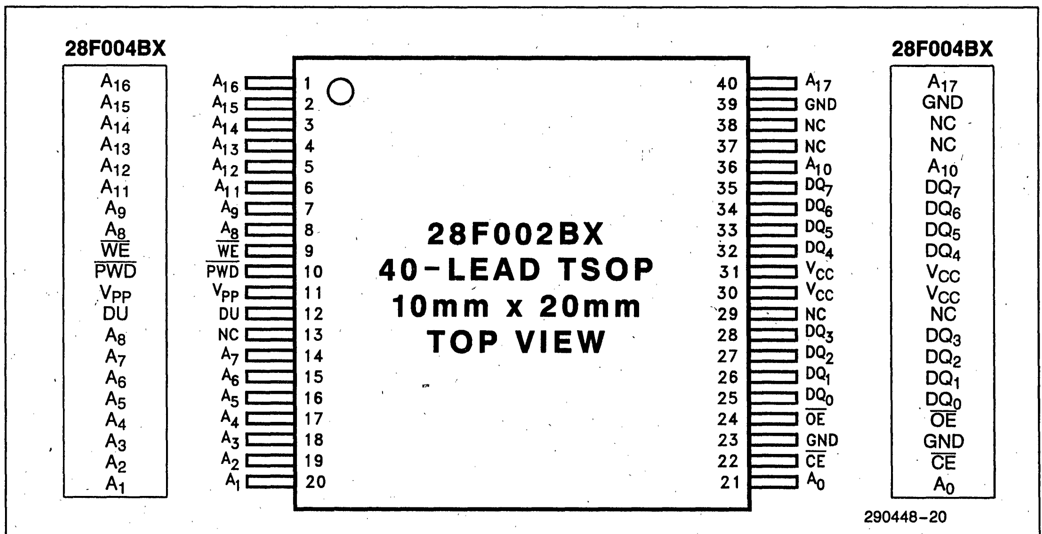


Figure 5. TSOP Lead Configuration for x8 28F002BX

1.4 Pin Descriptions for the x8/x16 28F200BX

Symbol	Type	Name and Function
A_0–A_{16}	I	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
A_9	I	**ADDRESS INPUT:** When A_9 is at 12V the signature mode is accessed. During this mode A_0 decodes between the manufacturer and device ID's. When $\overline{BYTE}$ is at a logic low only the lower byte of the signatures are read. DQ_{15}/A_{-1} is a don't care in the signature mode when $\overline{BYTE}$ is low.
DQ_0–DQ_7	I/O	**DATA INPUTS/OUTPUTS:** Inputs array data on the second $\overline{CE}$ and $\overline{WE}$ cycle during a program command. Inputs commands to the Command User Interface when $\overline{CE}$ and $\overline{WE}$ are active. Data is internally latched during the write and program cycles. Outputs array, intelligent identifier and Status Register data. The data pins float to tri-state when the chip is deselected or the outputs are disabled.
DQ_8–DQ_{15}	I/O	**DATA INPUTS/OUTPUTS:** Inputs array data on the second $\overline{CE}$ and $\overline{WE}$ cycle during a program command. Data is internally latched during the write and program cycles. Outputs array data. The data pins float to tri-state when the chip is deselected or the outputs are disabled as in the byte-wide mode ($\overline{BYTE}$ = "0"). In the byte-wide mode DQ_{15}/A_{-1} becomes the lowest order address for data output on DQ_0–DQ_7.
$\overline{CE}$	I	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels. If $\overline{CE}$ and $\overline{PWD}$ are high, but not at a CMOS high level, the standby current will increase due to current flow through the $\overline{CE}$ and $\overline{PWD}$ input stages.
$\overline{PWD}$	I	**POWER-DOWN:** Provides three-state control. Puts the device in deep power-down mode. Locks the boot block from program/erase. When $\overline{PWD}$ is at logic high level and equals 6.5V maximum the boot block is locked and cannot be programmed or erased. When $\overline{PWD}$ = 11.4V minimum the boot block is unlocked and can be programmed or erased. When $\overline{PWD}$ is at a logic low level the boot block is locked, the deep power-down mode is enabled and the WSM is reset preventing any blocks from being programmed or erased, therefore providing data protection during power transitions. $\overline{PWD}$ terminates any internally timed erase or program activities when it is taken to a logic low. $\overline{PWD}$ activates the $\overline{CE}$ input stage and requires 300 ns recovery time to transition from deep power-down to valid data on the outputs or 215 ns delay before the device can recognize valid inputs.
$\overline{OE}$	I	**OUTPUT ENABLE:** Gates the device's outputs through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	I	**WRITE ENABLE:** Controls writes to the Command Register and array blocks. $\overline{WE}$ is active low. Addresses and data are latched on the rising edge of the $\overline{WE}$ pulse.
$\overline{BYTE}$	I	**$\overline{BYTE}$ ENABLE:** Controls whether the device operates in the byte-wide mode (x8) or the word-wide mode (x16). $\overline{BYTE}$ pin must be controlled at CMOS levels to meet 100 μA CMOS current in the standby mode. $\overline{BYTE}$ = "0" enables the byte-wide mode, where data is read and programmed on DQ_0–DQ_7 and DQ_{15}/A_{-1} becomes the lowest order address that decodes between the upper and lower byte. DQ_8–DQ_{14} are tri-stated during the byte-wide mode. $\overline{BYTE}$ = "1" enables the word-wide mode where data is read and programmed on DQ_0–DQ_{15}.
V_{PP}		**PROGRAM/ERASE POWER SUPPLY:** For erasing memory array blocks or programming data in each block. **Note:** $V_{PP} < V_{PPLMAX}$ memory contents cannot be altered.
V_{CC}		**DEVICE POWER SUPPLY (5V ±10%, 5V ± 5%)**
GND		**GROUND:** For all internal circuitry.
NC		**NO CONNECT:** Pin may be driven or left floating.
DU		**DON'T USE PIN:** Pin should not be connected to anything.

3

1.5 Pin Descriptions for x8 28F002BX

Symbol	Type	Name and Function
A_0–A_{17}	I	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
A_9	I	**ADDRESS INPUT:** When A_9 is at 12V the signature mode is accessed. During this mode A_0 decodes between the manufacturer and device ID's.
DQ_0–DQ_7	I/O	**DATA INPUTS/OUTPUTS:** Inputs array data on the second $\overline{CE}$ and $\overline{WE}$ cycle during a program command. Inputs commands to the command user interface when $\overline{CE}$ and $\overline{WE}$ are active. Data is internally latched during the write and program cycles. Outputs array, intelligent identifier and status register data. The data pins float to tri-state when the chip is deselected or the outputs are disabled.
$\overline{CE}$	I	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels. If $\overline{CE}$ and $\overline{PWD}$ are high, but not at a CMOS high level, the standby current will increase due to current flow through the $\overline{CE}$ and $\overline{PWD}$ input stages.
$\overline{PWD}$	I	**POWERDOWN:** Provides Three-State control. Puts the device in deep powerdown mode. Locks the Boot Block from program/erase. When $\overline{PWD}$ is at logic high level and equals 6.5V maximum the Boot Block is locked and cannot be programmed or erased. When $\overline{PWD}$ = 11.4V minimum the Boot Block is unlocked and can be programmed or erased. When $\overline{PWD}$ is at a logic low level the Boot Block is locked, the deep powerdown mode is enabled and the WSM is reset preventing any blocks from being programmed or erased, therefore providing data protection during power transitions. $\overline{PWD}$ terminates any internally timed erase or program activities when it is taken to a logic low. $\overline{PWD}$ activates the $\overline{CE}$ input stage and requires 300 ns recovery time to transition from deep powerdown to valid data on the outputs or 215 ns delay before the device can recognize valid inputs.
$\overline{OE}$	I	**OUTPUT ENABLE:** Gates the device's outputs through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	I	**WRITE ENABLE:** Controls writes to the Command Register and array blocks. $\overline{WE}$ is active low. Addresses and data are latched on the rising edge of the $\overline{WE}$ pulse.
V_{PP}		**PROGRAM/ERASE POWER SUPPLY:** For erasing memory array blocks or programming data in each block. **Note:** $V_{PP} < V_{PPLMAX}$ memory contents cannot be altered.
V_{CC}		**DEVICE POWER SUPPLY (5V ± 10%, 5V ± 5%)**
GND		**GROUND:** For all internal circuitry.
NC		**NO CONNECT:** Pin may be driven or left floating.
DU		**DON'T USE PIN:** Pin should not be connected to anything.

2.0 28F200BX WORD/BYTE-WIDE PRODUCTS DESCRIPTION

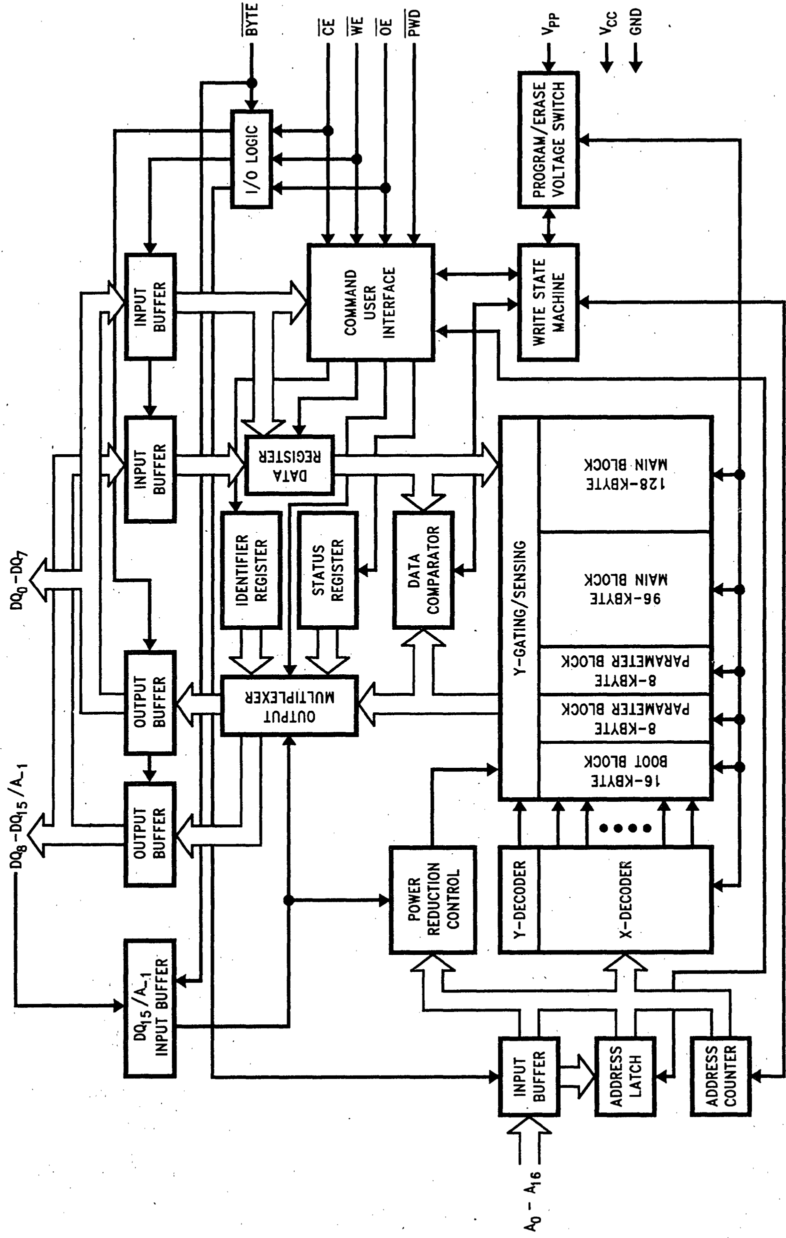


Figure 6. 28F200BX Word/Byte-Wide Block Diagram

2.1 28F200BX Memory Organization

2.1.1 BLOCKING

The 28F200BX uses a blocked array architecture to provide independent erasure of memory blocks. A block is erased independently of other blocks in the array when an address is given within the block address range and the Erase Setup and Erase Confirm commands are written to the CUI. The 28F200BX is a random read/write memory, only erasure is performed by block.

2.1.1.1 Boot Block Operation and Data Protection

The 16 Kbyte boot block provides a lock feature for secure code storage. The intent of the boot block is to provide a secure storage area for the kernel code that is required to boot a system in the event of power failure or other disruption during code update. This lock feature ensures absolute data integrity by preventing the boot block from being written or erased when $\overline{PWD}$ is not at 12V. The boot block can be erased and written when $\overline{PWD}$ is held at 12V for the duration of the erase or program operation. This allows customers to change the boot code when necessary while providing security when needed. See the Block Memory Map section for address locations of the boot block for the 28F200BX-T and 28F200BX-B.

2.1.1.2 Parameter Block Operation

The 28F200BX has 2 parameter blocks (8 Kbytes each). The parameter blocks are intended to provide storage for frequently updated system parameters and configuration or diagnostic information. The parameter blocks can also be used to store additional boot or main code. The parameter blocks however, do not have the hardware write protection feature that the boot block has. The parameter blocks provide for more efficient memory utilization when dealing with parameter changes versus regularly blocked devices. See the Block Memory Map section for address locations of the parameter blocks for the 28F200BX-T and 28F200BX-B.

2.1.1.3 Main Block Operation

Two main blocks of memory exist on the 28F200BX (1 x 128 Kbyte block and 1 x 96 Kbyte block). See the following section on Block Memory Map for the address location of these blocks for the 28F200BX-T and 28F200BX-B products.

2.1.2 BLOCK MEMORY MAP

Two versions of the 28F200BX product exist to support two different memory maps of the array blocks in order to accommodate different microprocessor protocols for boot code location. The 28F200BX-T memory map is inverted from the 28F200BX-B memory map.

2.1.2.1 28F200BX-B Memory Map

The 28F200BX-B device has the 16 Kbyte boot block located from 00000H to 01FFFH to accommodate those microprocessors that boot from the bottom of the address map at 00000H. In the 28F200BX-B the first 8 Kbyte parameter block resides in memory space from 02000H to 02FFFH. The second 8 Kbyte parameter block resides in memory space from 03000H to 03FFFH. The 96 Kbyte main block resides in memory space from 04000H to 0FFFFH. The 128 Kbyte main block resides in memory space from 10000H to 1FFFFH (word locations). See Figure 7.

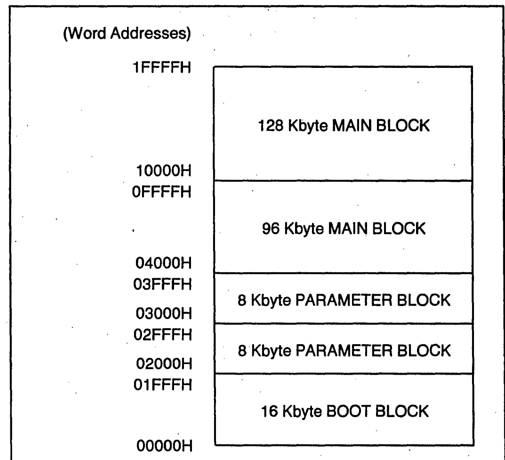


Figure 7. 28F200BX-B Memory Map

2.1.2.2 28F200BX-T Memory Map

The 28F200BX-T device has the 16 Kbyte boot block located from 1E000H to 1FFFFH to accommodate those microprocessors that boot from the top of the address map. In the 28F200BX-T the first 8 Kbyte parameter block resides in memory space from 1D000H to 1DFFFH. The second 8 Kbyte parameter block resides in memory space from 1C000H to 1CFFFH. The 96 Kbyte main block resides in memory space from 10000H to 1BFFFH. The 128 Kbyte main block resides in memory space from 00000H to 0FFFFH as shown in Figure 8.

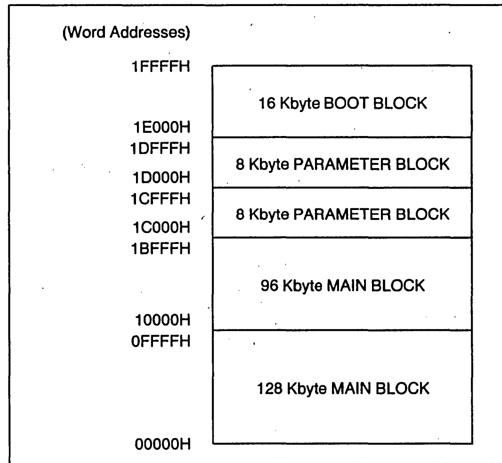


Figure 8. 28F200BX-T Memory Map

3

3.0 28F002BX BYTE-WIDE PRODUCTS DESCRIPTION

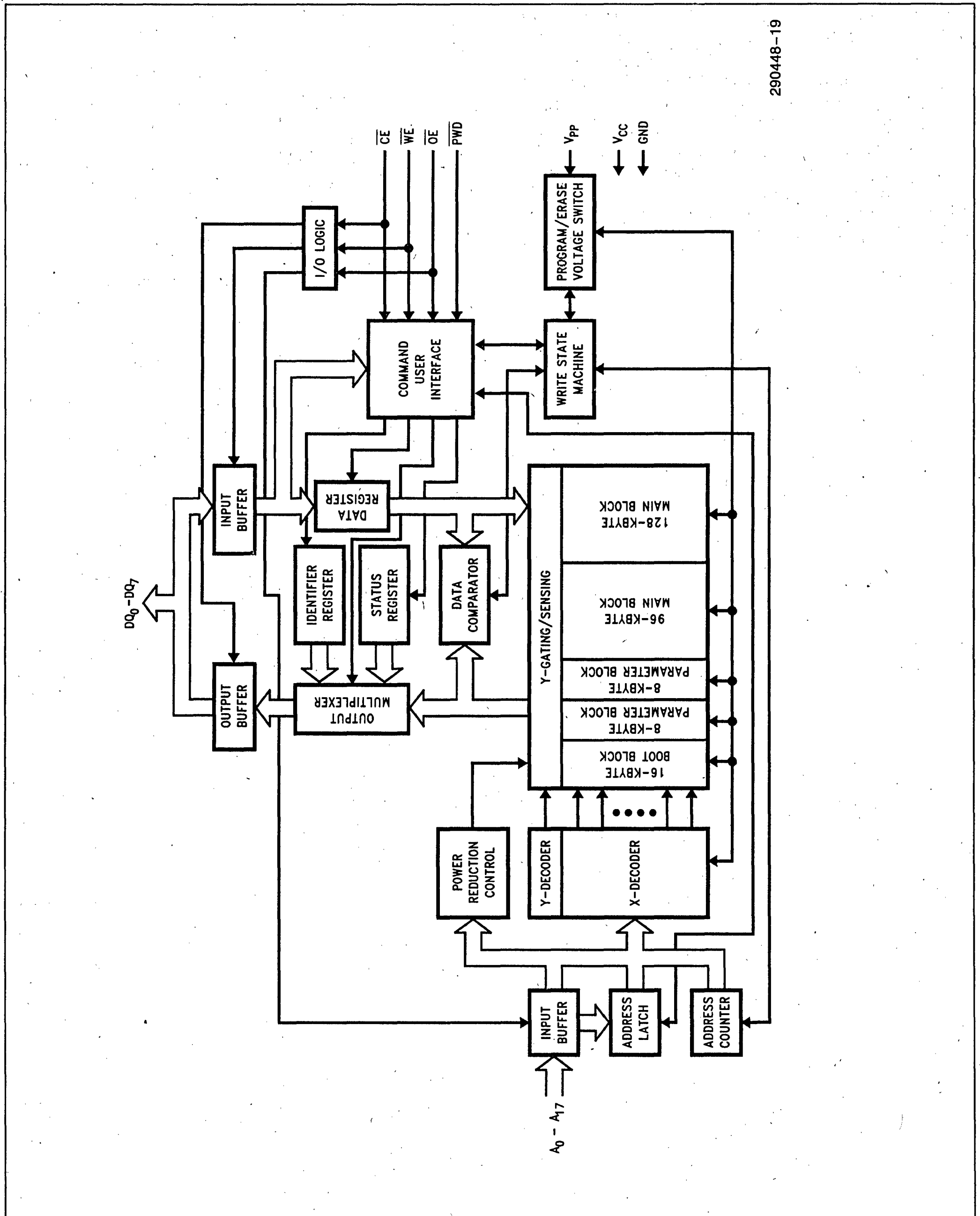


Figure 9. 28F002BX Byte-Wide Block Diagram

3.1 28F002BX Memory Organization

3.1.1 BLOCKING

The 28F002BX uses a blocked array architecture to provide independent erasure of memory blocks. A block is erased independently of other blocks in the array when an address is given within the block address range and the Erase Setup and Erase Confirm commands are written to the CUI. The 28F002BX is a random read/write memory, only erasure is performed by block.

3.1.1.1 Boot Block Operation and Data Protection

The 16 Kbyte boot block provides a lock feature for secure code storage. The intent of the boot block is to provide a secure storage area for the kernel code that is required to boot a system in the event of power failure or other disruption during code update. This lock feature ensures absolute data integrity by preventing the boot block from being programmed or erased when $\overline{PWD}$ is not at 12V. The boot block can be erased and programmed when $\overline{PWD}$ is held at 12V for the duration of the erase or program operation. This allows customers to change the boot code when necessary while still providing security when needed. See the Block Memory Map section for address locations of the boot block for the 28F002BX-T and 28F002BX-B.

3.1.1.2 Parameter Block Operation

The 28F002BX has 2 parameter blocks (8 Kbytes each). The parameter blocks are intended to provide storage for frequently updated system parameters and configuration or diagnostic information. The parameter blocks can also be used to store additional boot or main code. The parameter blocks however, do not have the hardware write protection feature that the boot block has. Parameter blocks provide for more efficient memory utilization when dealing with small parameter changes versus regularly blocked devices. See the Block Memory Map section for address locations of the parameter blocks for the 28F002BX-T and 28F002BX-B.

3.1.1.3 Main Block Operation

Two main blocks of memory exist on the 28F002BX (1 x 128 Kbyte block and 1 x 96 Kbyte block). See the following section on Block Memory Map for address location of these blocks for the 28F002BX-T and 28F002BX-B.

3.1.2 BLOCK MEMORY MAP

Two versions of the 28F002BX product exist to support two different memory maps of the array blocks in order to accommodate different microprocessor protocols for boot code location. The 28F002BX-T memory map is inverted from the 28F002BX-B memory map.

3.1.2.1 28F002BX-B Memory Map

The 28F002BX-B device has the 16 Kbyte boot block located from 00000H to 03FFFH to accommodate those microprocessors that boot from the bottom of the address map at 00000H. In the 28F002BX-B the first 8 Kbyte parameter block resides in memory from 04000H to 05FFFH. The second 8 Kbyte parameter block resides in memory space from 06000H to 07FFFH. The 96 Kbyte main block resides in memory space from 08000H to 1FFFFH. The 128 Kbyte main block resides in memory space from 20000H to 3FFFFH. See Figure 10.

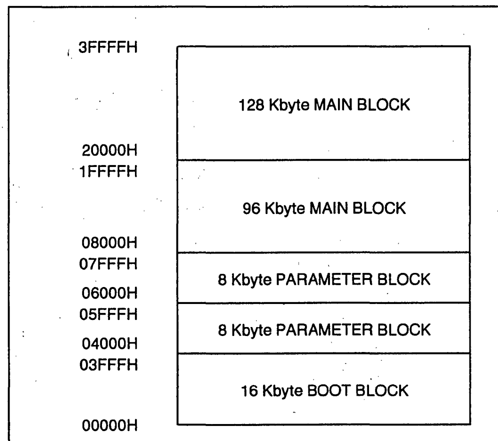


Figure 10. 28F002BX-B Memory Map

3.1.2.2 28F002BX-T Memory Map

The 28F002BX-T device has the 16 Kbyte boot block located from 3C000H to 3FFFFH to accommodate those microprocessors that boot from the top of the address map. In the 28F002BX-T the first 8 Kbyte parmeter block resides in memory space from 3A000H to 3BFFFH. The second 8 Kbyte parameter block resides in memory space from 38000H to 39FFFH. The 96 Kbyte main block resides in memory space from 20000H to 37FFFH. The 128 Kbyte main block resides in memory space from 00000H to 1FFFFH.

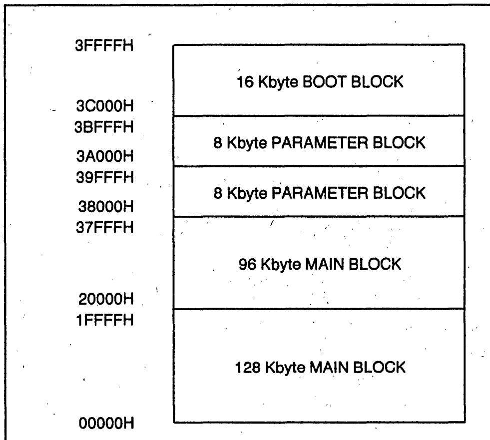


Figure 11. 28F002BX-T Memory Map

4.0 PRODUCT FAMILY PRINCIPLES OF OPERATION

Flash memory augments EPROM functionality with in-circuit electrical write and erase. The 2 Mbit flash family utilizes a Command User Interface (CUI) and internally generated and timed algorithms to simplify write and erase operations.

The CUI allows for 100% TTL-level control inputs, fixed power supplies during erasure and programming, and maximum EPROM compatibility.

In the absence of high voltage on the V_{PP} pin, the 2 Mbit boot block flash family will only successfully execute the following commands: Read Array, Read Status Register, Clear Status Register and Intelligent Identifier mode. The device provides standard EPROM read, standby and output disable operations. Manufacturer Identification and Device Identification data can be accessed through the CUI or through the standard EPROM A9 high voltage access (V_{ID}) for PROM programming equipment.

The same EPROM read, standby and output disable functions are available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} allows write and erase of the device. All functions associated with altering memory contents: write and erase, Intelligent Identifier read and Read Status are accessed via the CUI.

The purpose of the Write State Machine (WSM) is to completely automate the write and erasure of the device. The WSM will begin operation upon receipt of a signal from the CUI and will report status back through a Status Register. The CUI will handle the $\overline{WE}$ interface to the data and address latches, as well as system software requests for status while the WSM is in operation.

4.1 28F200BX Bus Operations

Flash memory reads, erases and writes in-system via the local CPU. All bus cycles to or from the flash memory conform to standard microprocessor bus cycles.

Table 1. Bus Operations for WORD-WIDE Mode ($\overline{BYTE}$ = V_{IH})

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_9	A_0	V_{PP}	DQ_{0-15}
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	X	D_{OUT}
Output Disable		V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	High Z
Standby		V_{IH}	V_{IH}	X	X	X	X	X	High Z
Deep Power-Down	9	V_{IL}	X	X	X	X	X	X	High Z
Intelligent Identifier (Mfr)	4	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IL}	X	0089H
Intelligent Identifier (Device)	4, 5	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IH}	X	2274H 2275H
Write	6, 7, 8	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	D_{IN}

Table 2. Bus Operations for BYTE-WIDE Mode ($\overline{BYTE}$ = V_{IL})

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_9	A_0	A_{-1}	V_{PP}	DQ_{0-7}	DQ_{8-14}
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	X	X	D_{OUT}	High Z
Output Disable		V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	X	High Z	High Z
Standby		V_{IH}	V_{IH}	X	X	X	X	X	X	High Z	High Z
Deep Power-Down	9	V_{IL}	X	X	X	X	X	X	X	High Z	High Z
Intelligent Identifier (Mfr)	4	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IL}	X	X	89H	High Z
Intelligent Identifier (Device)	4, 5	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IH}	X	X	74H 75H	High Z
Write	6, 7, 8	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	X	D_{IN}	High Z

NOTES:
1. Refer to DC Characteristics.
2. X can be V_{IL} or V_{IH} for control pins and addresses, V_{PPL} or V_{PPH} for V_{PP}.
3. See DC characteristics for V_{PPL}, V_{PPH}, V_{HH}, V_{ID} voltages.
4. Manufacturer and Device codes may also be accessed via a CUI write sequence. A_1–A_{17} = X.
5. Device ID = 2274H for 28F200BX-T and 2275H for 28F200BX-B.
6. Refer to Table 4 for valid D_{IN} during a write operation.
7. Command writes for Block Erase or Word/Byte Write are only executed when $V_{PP} = V_{PPH}$.
8. To write or erase the boot block, hold $\overline{PWD}$ at V_{HH}.
9. $\overline{PWD}$ must be at GND ±0.2V to meet the 1.2 μA maximum deep power-down current.

4.2 28F002BX Bus Operations

Table 3. Bus Operations

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_9	A_0	V_{PP}	DQ_{0-7}
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	X	D_{OUT}
Output Disable		V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	High Z
Standby		V_{IH}	V_{IH}	X	X	X	X	X	High Z
Deep Power-Down	9	V_{IL}	X	X	X	X	X	X	High Z
Intelligent Identifier (Mfr)	4	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IL}	X	89H
Intelligent Identifier (Device)	4, 5	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IH}	X	7CH 7DH
Write	6, 7, 8	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	D_{IN}

NOTES:
1. Refer to DC Characteristics.
2. X can be V_{IL} or V_{IH} for control pins and addresses, V_{PPL} or V_{PPH} for V_{PP}.
3. See DC characteristics for V_{PPL}, V_{PPH}, V_{HH}, V_{ID} voltages.
4. Manufacturer and Device codes may also be accessed via a CUI write sequence. A_1–A_{16} = X.
5. Device ID = 7CH for 28F002BX-T and 7DH for 28F002BX-B.
6. Refer to Table 4 for valid D_{IN} during a write operation.
7. Command writes for Block Erase or byte program are only executed when $V_{PP} = V_{PPH}$.
8. Program or erase the Boot block by holding $\overline{PWD}$ at V_{HH}.
9. $\overline{PWD}$ must be at GND ±0.2V to meet the 1.2 μA maximum deep power-down current.

4.3 Read Operations

The 2 Mbit boot block flash family has three user read modes; Array, Intelligent Identifier, and Status Register. Status Register read mode will be discussed in detail in the "Write Operations" section.

During power-up conditions (V_{CC} supply ramping), it takes a maximum of 600 ns from when V_{CC} is at 4.5V minimum to valid data on the outputs.

4.3.1 READ ARRAY

If the memory is not in the Read Array mode, it is necessary to write the appropriate read mode command to the CUI. The 2 Mbit boot block flash family has three control functions, all of which must be logically active, to obtain data at the outputs. Chip-Enable $\overline{CE}$ is the device selection control. Power-Down $\overline{PWD}$ is the device power control. Output-Enable $\overline{OE}$ is the DATA INPUT/OUTPUT (DQ[0:15] or DQ[0:7]) direction control and when active is used to drive data from the selected memory on to the I/O bus.

4.3.1.1 Output Control

With $\overline{OE}$ at logic-high level (V_{IH}), the output from the device is disabled and data input/output pins (DQ[0:15] or DQ[0:7]) are tri-stated. Data input is then controlled by $\overline{WE}$.

4.3.1.2 Input Control

With $\overline{WE}$ at logic-high level (V_{IH}), input to the device is disabled. Data Input/Output pins (DQ-[0:15] or DQ[0:7]) are controlled by $\overline{OE}$.

4.3.2 INTELLIGENT IDENTIFIERS

28F200BX Products

The manufacturer and device codes are read via the CUI or by taking the A_9 pin to 12V. Writing 90*H* to the CUI places the device into Intelligent Identifier read mode. A read of location 00000*H* outputs the manufacturer's identification code, 0089H, and location 00001*H* outputs the device code; 2274H for 28F200BX-T, 2275H for 28F200BX-B. When $\overline{BYTE}$ is at a logic low only the lower byte of the above signatures is read and DQ_{15}/A_{-1} is a "don't care" during Intelligent Identifier mode. A read array command must be written to the CUI to return to the read array mode.

28F002BX Products

The manufacturer and device codes are also read via the CUI or by taking the A9 pin to 12V. Writing 90*H* to the CUI places the device into Intelligent Identifier read mode. A read of location 00000*H* outputs the manufacturer's identification code, 89H, and location 00001*H* outputs the device code; 7CH for 28F002BX-T, 7DH for 28F002BX-B.

4.4 Write Operations

Commands are written to the CUI using standard microprocessor write timings. The CUI serves as the interface between the microprocessor and the internal chip operation. The CUI can decipher Read Array, Read Intelligent Identifier, Read Status Register, Clear Status Register, Erase and Program commands. In the event of a read command, the CUI simply points the read path at either the array, the intelligent identifier, or the status register depending on the specific read command given. For a program or erase cycle, the CUI informs the write state machine that a write or erase has been requested. During a program cycle, the Write State Machine will control the program sequences and the CUI will only respond to status reads. During an erase cycle, the CUI will respond to status reads and erase suspend. After the Write State Machine has completed its task, it will allow the CUI to respond to its full command set. The CUI will stay in the current command state until the microprocessor issues another command.

The CUI will successfully initiate an erase or write operation only when V_{PP} is within its voltage range. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable, available only when memory updates are desired. The system designer can also choose to "hard-wire" V_{PP} to 12V. The 2 Mbit boot block flash family is designed to accommodate either design practice. It is recommended that $\overline{PWD}$ be tied to logical Reset for data protection during unstable CPU reset function as described in the "Product Family Overview" section.

4.4.1 BOOT BLOCK WRITE OPERATIONS

In the case of Boot Block modifications (write and erase), $\overline{PWD}$ is set to V_{HH} = 12V typically, in addition to V_{PP} at high voltage. However, if $\overline{PWD}$ is not at V_{HH} when a program or erase operation of the boot block is attempted, the corresponding status register bit (Bit 4 for Program and Bit 5 for Erase, refer to Table 5 for Status Register Definitions) is set to indicate the failure to complete the operation.

4.4.2 COMMAND USER INTERFACE (CUI)

The Command User Interface (CUI) serves as the interface to the microprocessor. The CUI points the read/write path to the appropriate circuit block as described in the previous section. After the WSM has completed its task, it will set the WSM Status bit to a "1", which will also allow the CUI to respond to its full command set. Note that after the WSM has returned control to the CUI, the CUI will remain in its current state.

4.4.2.1 Command Set

Command Codes	Device Mode
00	Invalid/Reserved
10	Alternate Program Setup
20	Erase Setup
40	Program Setup
50	Clear Status Register
70	Read Status Register
90	Intelligent Identifier
B0	Erase Suspend
D0	Erase Resume/Erase Confirm
FF	Read Array

4.4.2.2 Command Function Descriptions

Device operations are selected by writing specific commands into the CUI. Table 4 defines the 2 Mbit boot block flash family commands.

Table 4. Command Definitions

Command	Bus Cycles Req'd	Notes	First Bus Cycle			Second Bus Cycle		
		8	Operation	Address	Data	Operation	Address	Data
Read Array/Reset	1	1	Write	X	FFH			
Intelligent Identifier	3	2, 4	Write	X	90H	Read	IA	IID
Read Status Register	2	3	Write	X	70H	Read	X	SRD
Clear Status Register	1		Write	X	50H			
Erase Setup/Erase Confirm	2	5	Write	BA	20H	Write	BA	D0H
Word/Byte Write Setup/Write	2	6, 7	Write	WA	40H	Write	WA	WD
Erase Suspend/Erase Resume	2		Write	X	B0H	Write	X	D0H
Alternate Word/Byte Write Setup/Write	2	6, 7	Write	WA	10H	Write	WA	WD

NOTES:
1. Bus operations are defined in Tables 1, 2, 3.
2. IA = Identifier Address: 00H for manufacturer code, 01H for device code.
3. SRD = Data read from Status Register.
4. IID = Intelligent Identifier Data.
Following the Intelligent Identifier Command, two read operations access manufacturer and device codes.
5. BA = Address within the block being erased.
6. PA = Address to be programmed.
PD = Data to be programmed at location PA.
7. Either 40H or 10H command is valid.
8. When writing commands to the device, the upper data bus [DQ8–DQ15] = X (28F200BX-only) which is either V_{CC} or V_{SS} to avoid burning additional current.

Invalid/Reserved

These are unassigned commands. It is not recommended that the customer use any command other than the valid commands specified above. Intel reserves the right to redefine these codes for future functions.

Read Array (FFH)

This single write command points the read path at the array. If the host CPU performs a $\overline{CE}/\overline{OE}$ controlled read immediately following a two-write sequence that started the WSM, then the device will output status register contents. If the Read Array command is given after Erase Setup the device is reset to read the array. A two Read Array command sequence (FFH) is required to reset to Read Array after Program Setup.

Inteligent Identifier (90H)

After this command is executed, the CUI points the output path to the Intelligent Identifier circuits. Only Intelligent Identifier values at addresses 0 and 1 can be read (only address A0 is used in this mode, all other address inputs are ignored).

Read Status Register (70H)

This is one of the two commands that is executable while the state machine is operating. After this command is written, a read of the device will output the contents of the status register, regardless of the address presented to the device.

The device automatically enters this mode after program or erase has completed.

Clear Status Register (50H)

The WSM can only set the Program Status and Erase Status bits in the status register, it can not clear them. Two reasons exist for operating the status register in this fashion. The first is a synchronization. The WSM does not know when the host CPU has read the status register, therefore it would not know when to clear the status bits. Secondly, if the CPU is programming a string of bytes, it may be more efficient to query the status register after programming the string. Thus, if any errors exist while programming the string, the status register will return the accumulated error status.

Program Setup (40H or 10H)

This command simply sets the CUI into a state such that the next write will load the address and data registers. Either 40H or 10H can be used for Program Setup. Both commands are included to accommodate efforts to achieve an industry standard command code set.

Program

The second write after the program setup command, will latch addresses and data. Also, the CUI initiates the WSM to begin execution of the program algorithm. While the WSM finishes the algorithm, the device will output Status Register contents. Note that the WSM cannot be suspended during programming.

Erase Setup (20H)

Prepares the CUI for the Erase Confirm command. No other action is taken. If the next command is not an Erase Confirm command then the CUI will set both the Program Status and Erase Status bits of the Status Register to a "1", place the device into the Read Array state, and wait for another command.

Erase Confirm (D0H)

If the previous command was an Erase Setup command, then the CUI will enable the WSM to erase, at the same time closing the address and data latches, and respond only to the Read Status Register and Erase Suspend commands. While the WSM is executing, the device will output Status Register data when $\overline{OE}$ is toggled low. Status Register data can only be updated by toggling either $\overline{OE}$ or $\overline{CE}$ low.

Erase Suspend (B0H)

This command only has meaning while the WSM is executing an Erase operation, and therefore will only be responded to during an erase operation. After this command has been executed, the CUI will set an output that directs the WSM to suspend Erase operations, and then return to responding to only Read Status Register or to the Erase Resume commands. Once the WSM has reached the Suspend state, it will set an output into the CUI which allows the CUI to respond to the Read Array, Read Status Register, and Erase Resume commands. In this mode, the CUI will not respond to any other commands. The WSM will also set the WSM Status bit to a "1". The WSM will continue to run, idling in the SUSPEND state, regardless of the state of all input control pins, with the exclusion of $\overline{PWD}$. $\overline{PWD}$ will immediately shut down the WSM and the remainder of the chip. During a suspend operation, the data and address latches will remain closed, but the address pads are able to drive the address into the read path.

Erase Resume (D0H)

This command will cause the CUI to clear the Suspend state and set the WSM Status bit to a "0", but only if an Erase Suspend command was previously issued. Erase Resume will not have any effect in all other conditions.

4.4.3 STATUS REGISTER

The 2 Mbit boot block flash family contains a status register which may be read to determine when a program or erase operation is complete, and whether that operation completed successfully. The status register may be read at any time by writing the Read Status command to the CUI. After writing this command, all subsequent Read operations output data from the status register until another command is written to the CUI. A Read Array command must be written to the CUI to return to the Read Array mode.

The status register bits are output on DQ[0:7] whether the device is in the byte-wide (x8) or word-wide (x16) mode for the 28F200BX. In the word-wide mode the upper byte, DQ[8:15] is set to 00H during a Read Status command. In the byte-wide mode, DQ[8:14] are tri-stated and DQ15/A−1 retains the low order address function.

It should be noted that the contents of the status register are latched on the falling edge of $\overline{OE}$ or $\overline{CE}$ whichever occurs last in the read cycle. This prevents possible bus errors which might occur if the contents of the status register change while reading the status register. $\overline{CE}$ or $\overline{OE}$ must be toggled with each subsequent status read, or the completion of a program or erase operation will not be evident.

The Status Register is the interface between the microprocessor and the Write State Machine (WSM). When the WSM is active, this register will indicate the status of the WSM, and will also hold the bits indicating whether or not the WSM was successful in performing the desired operation. The WSM sets status bits "Three" through "Seven" and clears bits "Six" and "Seven", but cannot clear status bits "Three" through "Five". These bits can only be cleared by the controlling CPU through the use of the Clear Status Register command.

4.4.3.1 Status Register Bit Definition

Table 5. Status Register Definitions

WSMS	ESS	ES	PS	VPPS	R	R	R
7	6	5	4	3	2	1	0

	NOTES:
SR.7 = WRITE STATE MACHINE STATUS 1 = Ready 0 = Busy	Write State Machine Status bit must first be checked to determine byte/word program or block erase completion, before the Program or Erase Status bits are checked for success.
SR.6 = ERASE SUSPEND STATUS 1 = Erase Suspended 0 = Erase in Progress/Completed	When Erase Suspend is issued, WSM halts execution and sets both WSMS and ESS bits to "1". ESS bit remains set to "1" until an Erase Resume command is issued.
SR.5 = ERASE STATUS 1 = Error in Block Erasure 0 = Successful Block Erase	When this bit is set to "1". WSM has applied the maximum number of erase pulses to the block and is still unable to successfully perform an erase verify.
SR.4 = PROGRAM STATUS 1 = Error in Byte/Word Program 0 = Successful Byte/Word Program	When this bit is set to "1", WSM has attempted but failed to Program a byte or word.
SR.3 = V_{PP} STATUS 1 = V_{PP} Low Detect; Operation Abort 0 = V_{PP} OK	The V_{PP} Status bit, unlike an A/D converter, does not provide continuous indication of V_{PP} level. The WSM interrogates the V_{PP} level only after the byte write or block erase command sequences have been entered and informs the system if V_{PP} has not been switched on. The V_{PP} Status bit is not guaranteed to report accurate feedback between V_{PPL} and V_{PPH}.
SR.2–SR.0 = RESERVED FOR FUTURE ENHANCEMENTS	These bits are reserved for future use and should be masked out when polling the Status Register.

4.4.3.2 Clearing the Status Register

Certain bits in the status register are set by the write state machine, and can only be reset by the system software. These bits can indicate various failure conditions. By allowing the system software to control the resetting of these bits, several operations may be performed (such as cumulatively programming several bytes or erasing multiple blocks in sequence). The status register may then be read to determine if an error occurred during that programming or erasure series. This adds flexibility to the way the device may be programmed or erased. To clear the status register, the Clear Status Register command is written to the CUI. Then, any other command may be issued to the CUI. Note again that before a read cycle can be initiated, a Read Array command must be written to the CUI to specify whether the read data is to come from the array, status register, or Intelligent Identifier.

4.4.4 PROGRAM MODE

Program is executed by a two-write sequence. The Program Setup command is written to the CUI followed by a second write which specifies the address and data to be programmed. The write state machine will execute a sequence of internally timed events to:

1. Program the desired bits of the addressed memory word (byte), and
2. Verify that the desired bits are sufficiently programmed

Programming of the memory results in specific bits within a byte or word being changed to a "0".

If the user attempts to program "1"s, there will be no change of the memory cell content and no error occurs.

Similar to erasure, the status register indicates whether programming is complete. While the program sequence is executing, bit 7 of the status register is a "0". The status register can be polled by

toggling either $\overline{CE}$ or $\overline{OE}$ to determine when the program sequence is complete. Only the Read Status Register command is valid while programming is active.

When programming is complete, the status bits, which indicate whether the program operation was successful, should be checked. If the programming operation was unsuccessful, Bit 4 of the status register is set to a "1" to indicate a Program Failure. If Bit 3 is set then V_{PP} was not within acceptable limits, and the WSM will not execute the programming sequence.

The status register should be cleared before attempting the next operation. Any CUI instruction can follow after programming is completed; however, it must be recognized that reads from the memory, status register, or Intelligent Identifier cannot be accomplished until the CUI is given the appropriate command. A Read Array command must first be given before memory contents can be read.

Figure 12 shows a system software flowchart for device byte programming operation. Figure 13 shows a similar flowchart for device word programming operation (28F200BX-only).

4.4.5 ERASE MODE

Erasure of a single block is initiated by writing the Erase Setup and Erase Confirm commands to the CUI, along with the addresses, A[12:16] for the 28F200BX or A[12:17] for the 28F002BX, identifying the block to be erased. These addresses are latched internally when the Erase Confirm command is issued. Block erasure results in all bits within the block being set to "1".

The WSM will execute a sequence of internally timed events to:

1. Program all bits within the block
2. Verify that all bits within the block are sufficiently programmed
3. Erase all bits within the block and
4. Verify that all bits within the block are sufficiently erased

While the erase sequence is executing, Bit 7 of the status register is a "0".

When the status register indicates that erasure is complete, the status bits, which indicate whether the erase operation was successful, should be checked. If the erasure operation was unsuccessful, Bit 5 of the status register is set to a "1" to indicate an Erase Failure. If V_{PP} was not within acceptable limits after the Erase Confirm command is issued, the WSM will not execute an erase sequence; instead, Bit 5 of the status register is set to a "1" to indicate an Erase Failure, and Bit 3 is set to a "1" to identify that V_{PP} supply voltage was not within acceptable limits.

The status register should be cleared before attempting the next operation. Any CUI instruction can follow after erasure is completed; however, it must be recognized that reads from the memory array, status register, or Intelligent Identifier can not be accomplished until the CUI is given the appropriate command. A Read Array command must first be given before memory contents can be read.

Figure 14 shows a system software flowchart for Block Erase operation.

4.4.5.1 Suspending and Resuming Erase

Since an erase operation typically requires 1 to 3 seconds to complete, an Erase Suspend command is provided. This allows erase-sequence interruption in order to read data from another block of the memory. Once the erase sequence is started, writing the Erase Suspend command to the CUI requests that the Write State Machine (WSM) pause the erase sequence at a predetermined point in the erase algorithm. The status register must be read to determine when the erase operation has been suspended.

At this point, a Read Array command can be written to the CUI in order to read data from blocks other than that which is being suspended. The only other valid command at this time is the Erase Resume command or Read Status Register operation.

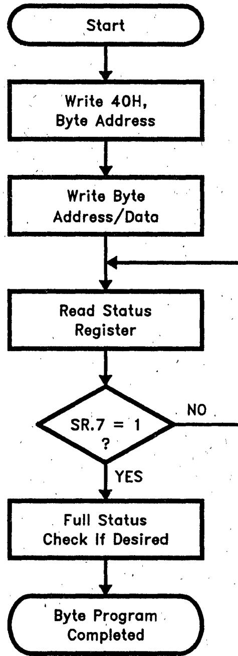
Figure 15 shows a system software flowchart detailing the operation.

During Erase Suspend mode, the chip can go into a pseudo-standby mode by taking $\overline{CE}$ to V_{IH} and the active current is now a maximum of 10 mA. If the chip is enabled while in this mode by taking $\overline{CE}$ to V_{IL}, the Erase Resume command can be issued to resume the erase operation.

Upon completion of reads from any block other than the block being erased, the Erase Resume command must be issued. When the Erase Resume command is given, the WSM will continue with the erase sequence and complete erasing the block. As with the end of erase, the status register must be read, cleared, and the next instruction issued in order to continue.

4.4.6 EXTENDED CYCLING

Intel has designed extended cycling capability into its ETOX III flash memory technology. The 2 Mbit boot block flash family is designed for 100,000 program/erase cycles on each of the five blocks. The combination of low electric fields, clean oxide processing and minimized oxide area per memory cell subjected to the tunneling electric field, results in very high cycling capability.



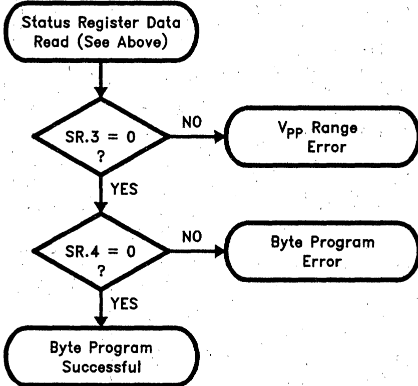
290448-6

Bus Operation	Command	Comments
Write	Setup Program	Data = 40H Address = Byte to be programmed
Write	Program	Data to be programmed Address = Byte to be programmed
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent bytes.
Full status check can be done after each byte or after a sequence of bytes.

Write FFH after the last byte programming operation to reset the device to Read Array Mode.

Full Status Check Procedure



290448-7

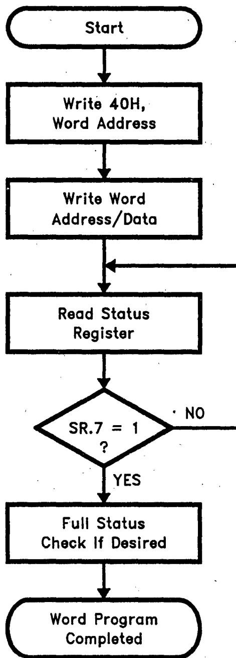
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Byte Program Error

SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple bytes are programmed before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 12. Automated Byte Programming Flowchart



290448-8

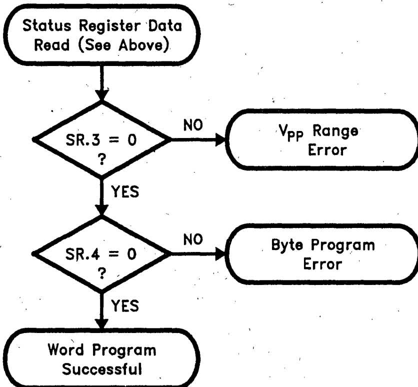
Bus Operation	Command	Comments
Write	Setup Program	Data = 40H Address = Word to be programmed
Write	Program	Data to be programmed Address = Word to be programmed
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent words.

Full status check can be done after each word or after a sequence of words.

Write FFH after the last word programming operation to reset the device to Read Array Mode.

Full Status Check Procedure



290448-9

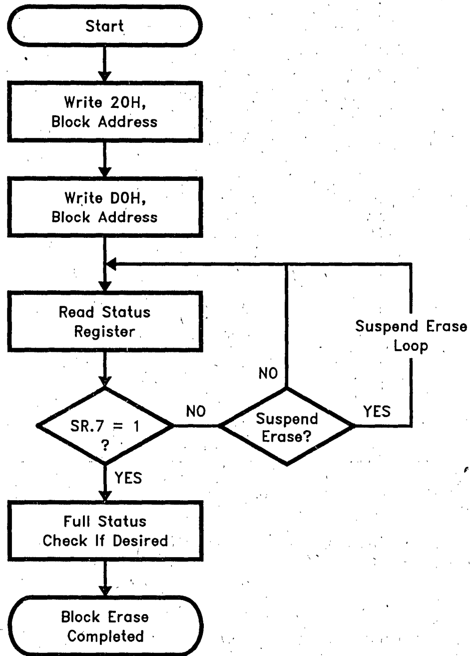
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Word Program Error

SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple words are programmed before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 13. Automated Word Programming Flowchart



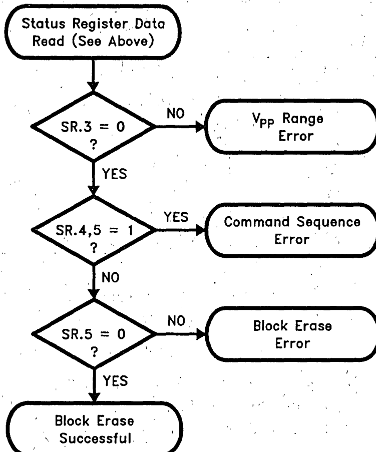
Bus Operation	Command	Comments
Write	Setup Erase	Data = 20H Address = Within block to be erased
Write	Erase	Data = D0H Address = Within block to be erased
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent blocks.

Full status check can be done after each block or after a sequence of blocks.

Write FFH after the last block erase operation to reset the device to Read Array Mode.

Full Status Check Procedure



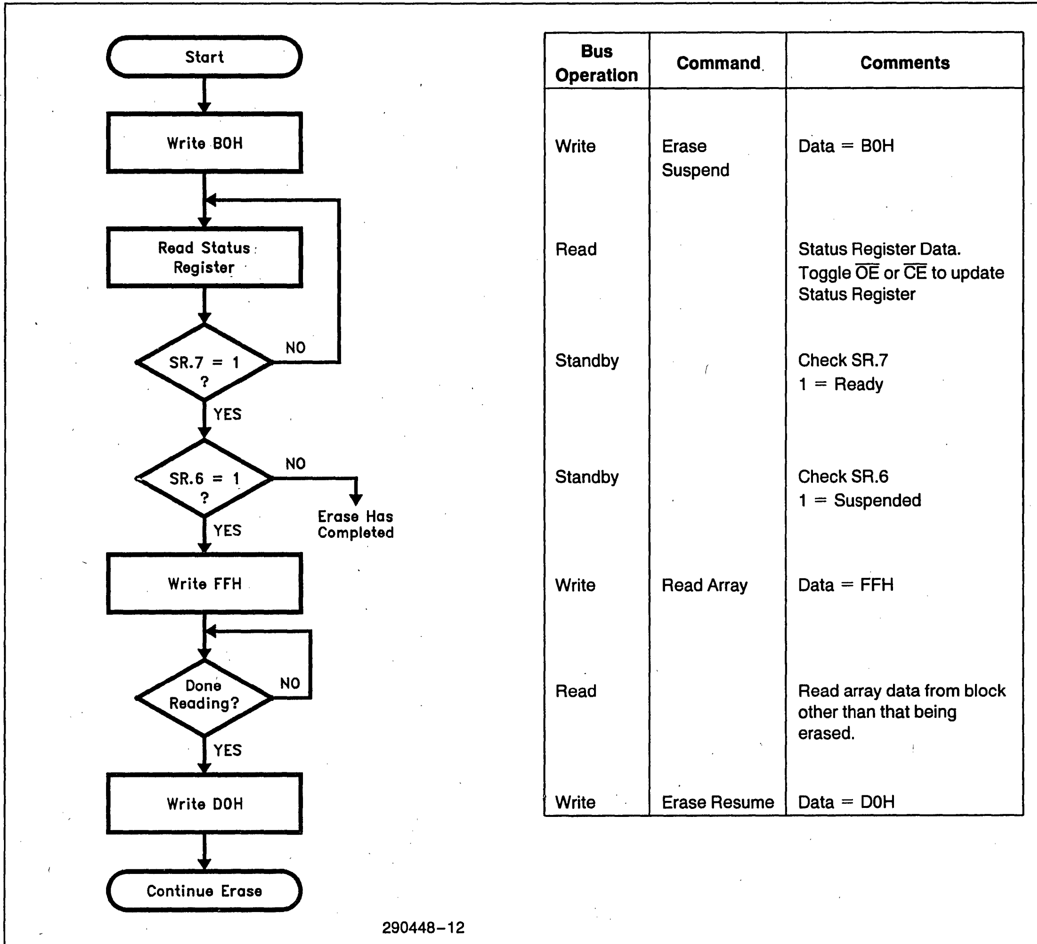
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4,5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error

SR.3 MUST be cleared, if set during an erase attempt, before further attempts are allowed by the Write State Machine.

SR.5 is only cleared by the Clear Status Register Command, in cases where multiple blocks are erased before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 14. Automated Block Erase Flowchart



Bus Operation	Command	Comments
Write	Erase Suspend	Data = B0H
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready
Standby		Check SR.6 1 = Suspended
Write	Read Array	Data = FFH
Read		Read array data from block other than that being erased.
Write	Erase Resume	Data = D0H

Figure 15. Erase Suspend/Resume Flowchart

4.5 Power Consumption

4.5.1 ACTIVE POWER

With $\overline{CE}$ at a logic-low level and $\overline{PWD}$ at a logic-high level, the device is placed in the active mode. The device I_{CC} current is a maximum of 60 mA at 10 MHz with TTL input signals.

4.5.2 AUTOMATIC POWER SAVINGS

Automatic Power Savings (APS) is a low power feature during active mode of operation. The 2 Mbit family of products incorporate Power Reduction Control (PRC) circuitry which basically allows the device to put itself into a low current state when it is not being accessed. After data is read from the memory array, PRC logic controls the device's power consumption by entering the APS mode where maximum I_{CC} current is 3 mA and typical I_{CC} current is 1 mA. The device stays in this static state with outputs valid until a new location is read.

4.5.3 STANDBY POWER

With $\overline{CE}$ at a logic-high level (V_{IH}), and the CUI in read mode, the memory is placed in standby mode where the maximum I_{CC} standby current is 100 μA with CMOS input signals. The standby operation disables much of the device's circuitry and substantially reduces device power consumption. The outputs (DQ[0:15] or DQ[0:7]) are placed in a high-impedance state independent of the status of the $\overline{OE}$ signal. When the 2 Mbit boot block flash family is deselected during erase or program functions, the devices will continue to perform the erase or program function and consume program or erase active power until program or erase is completed.

4.5.4 DEEP POWER-DOWN

The 2 Mbit boot block flash family supports a typical I_{CC} of 0.2 μA in deep power-down mode. One of the target markets for these devices is in portable equipment where the power consumption of the machine is of prime importance. The 2 Mbit boot block flash family has a $\overline{PWD}$ pin which places the device in the deep power-down mode. When $\overline{PWD}$ is at a logic-low (GND $\pm$0.2V), all circuits are turned off and the device typically draws 0.2 μA of V_{CC} current.

During read modes, the $\overline{PWD}$ pin going low deselects the memory and places the output drivers in a high impedance state. Recovery from the deep power-down state, requires a minimum of 400 ns to access valid data (t_{PHQV}).

During erase or program modes, $\overline{PWD}$ low will abort either erase or program operation. The contents of the memory are no longer valid as the data has been corrupted by the $\overline{PWD}$ function. As in the read mode above, all internal circuitry is turned off to achieve the 0.2 μA current level.

$\overline{PWD}$ transitions to V_{IL} or turning power off to the device will clear the status register.

4.6 Power-Up Operation

The 2 Mbit boot block flash family is designed to offer protection against accidental block erasure or programming during power transitions. Upon power-up the 2 Mbit boot block flash family is indifferent as to which power supply, V_{PP} or V_{CC}, powers-up first. Power suppy sequencing is not required.

The 2 Mbit boot block flash family ensures the CUI is reset to the read mode on power-up.

In addition, on power-up the user must either drop $\overline{CE}$ low or present a new address to ensure valid data at the outputs.

A system designer must guard against spurious writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either signal to V_{IH} will inhibit writes to the device. The CUI architecture provides an added level of protection since alteration of memory contents can only occur after successful completion of the two-step command sequences. Finally, the device is disabled until $\overline{PWD}$ is brought to V_{IH}, regardless of the state of its control inputs. This feature provides yet another level of memory protection.

4.7 Power Supply Decoupling

Flash memory's power switching characteristics require careful device decoupling methods. System designers are interested in 3 supply current issues:

- Standby current levels (I_{CCS})
- Active current levels (I_{CCR})
- Transient peaks produced by falling and rising edges of $\overline{CE}$.

Transient current magnitudes depend on the device outputs' capacitive and inductive loading. Two-line control and proper decoupling capacitor selection will suppress these transient voltage peaks. Each flash device should have a 0.1 μF ceramic capacitor connected between each V_{CC} and GND, and between its V_{PP} and GND. These high frequency, low-inherent inductance capacitors should be placed as close as possible to the package leads.

4.7.1 V_{PP} TRACE ON PRINTED CIRCUIT BOARDS

Writing to flash memories while they reside in the target system, requires special consideration of the V_{PP} power supply trace by the printed circuit board designer. The V_{PP} pin supplies the flash memory cell's current for programming and erasing. One should use similar trace widths and layout considerations given to the V_{CC} power supply trace. Adequate V_{PP} supply traces and decoupling will decrease spikes and overshoots.

4.7.2 V_{CC}, V_{PP} AND $\overline{PWD}$ TRANSITIONS

The CUI latches commands as issued by system software and is not altered by V_{PP} or $\overline{CE}$ transitions or WSM actions. Its state upon power-up, after exit from deep power-down mode or after V_{CC} transitions below V_{LKO} (Lockout voltage), is Read Array mode.

After any word/byte write or block erase operation is complete and even after V_{PP} transitions down to V_{PPL}, the CUI must be reset to Read Array mode via the Read Array command when accesses to the flash memory are desired.

ABSOLUTE MAXIMUM RATINGS*

Commercial Operating Temperature
- During Read ....0°C to 70°C(1)
- During Block Erase and Word/Byte Write ....0°C to 70°C
- Temperature Under Bias....... −10°C to +80°C

Extended Operating Temperature
- During Read .... −40°C to +85°C
- During Block Erase and Word/Byte Write .... −40°C to +85°C
- Temperature Under Bias....... −40°C to +85°C

Storage Temperature .... −65°C to +125°C

Voltage on Any Pin (except V_{CC} and V_{PP}) with Respect to GND .... −2.0V to +7.0V(2)

Voltage on Pin $\overline{PWD}$ or Pin A_9 with Respect to GND ..... −2.0V to +13.5V(2, 3)

V_{PP} Program Voltage with Respect to GND during Block Erase and Word/Byte Write ..... −2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage with Respect to GND .... −2.0V to +7.0V(2)

Output Short Circuit Current............. 100 mA(4)

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

3

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC voltage is −0.5V on input/output pins. During transitions, this level may undershoot to −2.0V for periods <20 ns.
Maximum DC voltage on input/output pins is V_{CC} + 0.5V which, during transitions, may overshoot to V_{CC} + 2.0V for periods <20 ns.
3. Maximum DC voltage on A_9 or V_{PP} may overshoot to 6 +14.0V for periods <20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.
5. 10% V_{CC} specifications reference the 28F200BX-60/28F002BX-60 in their standard test configuration, and the 28F200BX-80/28F002BX-80.
6. 5% V_{CC} specifications reference the 28F200BX-60/28F002BX-60 in their high speed test configuration.

OPERATING CONDITIONS

Symbol	Parameter	Notes	Min	Max	Units
T_A	Operating Temperature		0	70	°C
V_{CC}	V_{CC} Supply Voltage (10%)	5	4.50	5.50	V
V_{CC}	V_{CC} Supply Voltage (5%)	6	4.75	5.25	V

DC CHARACTERISTICS

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{LI}	Input Load Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND

DC CHARACTERISTICS (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{CCS}	V_{CC} Standby Current	1, 3			1.5	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{IH}$
					100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{CC} \pm 0.2V$ 28F200BX: $\overline{BYTE} = V_{CC} \pm 0.2V$ or GND
I_{CCD}	V_{CC} Deep Power-Down Current	1		0.20	1.2	μA	$\overline{PWD}$ = GND $\pm 0.2V$
I_{CCR}	V_{CC} Read Current for 28F200BX Word-Wide Mode	1, 5, 6			60	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 10 MHz, I_{OUT} = 0 mA CMOS Inputs
					65	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 10 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCR}	V_{CC} Read Current for 28F200BX Byte-Wide Mode and 28F004BX	1, 5, 6			55	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 10 MHz, I_{OUT} = 0 mA CMOS Inputs
					60	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 10 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCW}	V_{CC} Word Write Current	1			70	mA	Word Write in Progress
I_{CCW}	V_{CC} Byte Write Current	1			60	mA	Byte Write in Progress
I_{CCE}	V_{CC} Block Erase Current	1			30	mA	Block Erase in Progress
I_{CCES}	V_{CC} Erase Suspend Current	1, 2		5	10	mA	Block Erase Suspended, $\overline{CE} = V_{IH}$
I_{PPS}	V_{PP} Standby Current	1			± 10	μA	$V_{PP} \leq V_{CC}$
I_{PPD}	V_{PP} Deep Power-Down Current	1			5.0	μA	$\overline{PWD}$ = GND $\pm 0.2V$
I_{PPR}	V_{PP} Read Current	1			200	μA	$V_{PP} > V_{CC}$
I_{PPW}	V_{PP} Word Write Current	1			40	mA	$V_{PP} = V_{PPH}$ Word Write in Progress
I_{PPW}	V_{PP} Byte Write Current	1			30	mA	$V_{PP} = V_{PPH}$ Byte Write in Progress
I_{PPE}	V_{PP} Block Erase Current	1			30	mA	$V_{PP} = V_{PPH}$ Block Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1			200	μA	$V_{PP} = V_{PPH}$ Block Erase Suspended
I_{ID}	A_9 Intelligent Identifier Current	1			500	μA	$A_9 = V_{ID}$
V_{ID}	A_9 Intelligent Identifier Voltage		11.5		13.0	V	
V_{IL}	Input Low Voltage		-0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage				0.45	V	$V_{CC} = V_{CC}$ Min I_{OL} = 5.8 mA

DC CHARACTERISTICS (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
V_{OH}	Output High Voltage		2.4			V	$V_{CC} = V_{CC}$ Min $I_{OH} = -2.5$ mA
V_{PPL}	V_{PP} during Normal Operations	3	0.0		6.5	V	
V_{PPH}	V_{PP} during Erase/Write Operations	7	11.4	12.0	12.6	V	
V_{PPH}	V_{PP} during Erase/Write Operations	8	10.8	12.0	13.2	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.0			V	
V_{HH}	$\overline{PWD}$ Unlock Voltage		11.5		13.0	V	Boot Block Write/Erase

EXTENDED TEMPERATURE OPERATING CONDITIONS

Symbol	Parameter	Notes	Min	Max	Units
T_A	Operating Temperature		−40	+85	°C
V_{CC}	V_{CC} Supply Voltage (10%)	5	4.50	5.50	V

3

DC CHARACTERISTICS: EXTENDED TEMPERATURE OPERATION

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{LI}	Input Load Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND
I_{CCS}	V_{CC} Standby Current	1, 3			1.5	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{IH}$
					100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{CC} \pm 0.2V$ 28F200BX: $\overline{BYTE} = V_{CC} \pm 0.2V$ or GND
I_{CCD}	V_{CC} Deep Power-Down Current	1		0.20	8	μA	$\overline{PWD}$ = GND ±0.2V
I_{CCR}	V_{CC} Read Current for 28F200BX Word-Wide Mode	1, 5, 6			70	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 10 MHz, I_{OUT} = 0 mA CMOS Inputs
					75	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 10 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCR}	V_{CC} Read Current for 28F200BX Byte-Wide Mode and 28F004BX	1, 5, 6			65	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 10 MHz, I_{OUT} = 0 mA CMOS Inputs
					70	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 10 MHz, I_{OUT} = 0 mA TTL Inputs

DC CHARACTERISTICS: EXTENDED TEMPERATURE OPERATION (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{PPS}	V_{PP} Standby Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PPD}	V_{PP} Deep Power-Down Current	1			5.0	μA	$\overline{PWD}$ = GND ±0.2V
I_{PPR}	V_{PP} Read Current	1			200	μA	$V_{PP} > V_{CC}$
I_{PPW}	V_{PP} Word Write Current	1			40	mA	$V_{PP} = V_{PPH}$ Word Write in Progress
I_{PPW}	V_{PP} Byte Write Current	1			30	mA	$V_{PP} = V_{PPH}$ Byte Write in Progress
I_{PPE}	V_{PP} Block Erase Current	1			30	mA	$V_{PP} = V_{PPH}$ Block Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1			200	μA	$V_{PP} = V_{PPH}$ Block Erase Suspended
I_{ID}	A_9 Intelligent Identifier Current	1			500	μA	$A_9 = V_{ID}$
V_{ID}	A_9 Intelligent Identifier Voltage		11.5		13.0	V	
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	$V_{CC} = V_{CC}$ Min I_{OL} = 5.8 mA
V_{OH}	Output High Voltage		2.4			V	$V_{CC} = V_{CC}$ Min I_{OH} = −2.5 mA
V_{PPL}	V_{PP} during Normal Operations	3	0.0		6.5	V	
V_{PPH}	V_{PP} during Erase/Write Operations	7	11.4	12.0	12.6	V	
V_{PPH}	V_{PP} during Erase/Write Operations	8	10.8	12.0	13.2	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.0			V	
V_{HH}	$\overline{PWD}$ Unlock Voltage		11.5		13.0	V	Boot Block Write/Erase

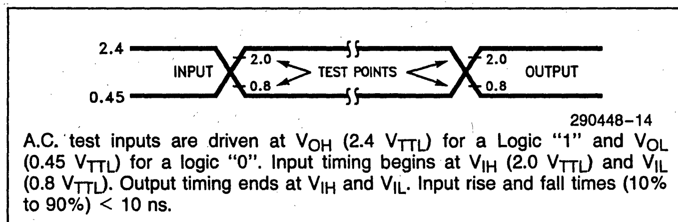
CAPACITANCE(4, 9) $T_A = 25°C, f = 1 MHz$

Symbol	Parameter	Typ	Max	Unit	Condition
C_{IN}	Input Capacitance	6	8	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	10	12	pF	$V_{OUT} = 0V$

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at $V_{CC} = 5.0V$, $V_{PP} = 12.0V$, $T = 25°C$. These currents are valid for all product versions (packages and speeds).
2. I_{CCES} is specified with the device deselected. If the device is read while in Erase Suspend Mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Block Erases and Word/Byte Writes are inhibited when $V_{PP} = V_{PPL}$ and not guaranteed in the range between V_{PPH} and V_{PPL}.
4. Sampled, not 100% tested.
5. Automatic Power Savings (APS) reduces I_{CCR} to less than 1 mA typical in static operation.
6. CMOS Inputs are either $V_{CC} \pm 0.2V$ or GND $\pm 0.2V$. TTL Inputs are either V_{IL} or V_{IH}.
7. $V_{PP} = 12.0V \pm 5\%$ for applications requiring 100,000 block erase cycles.
8. $V_{PP} = 12.0V \pm 10\%$ for applications requiring wider V_{PP} tolerances at 100 block erase cycles.
9. For the 28F002BX, address pin A_{10} follows the C_{OUT} capacitance numbers.

STANDARD TEST CONFIGURATION(1)

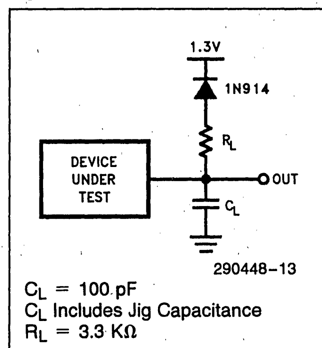
STANDARD A.C. INPUT/OUTPUT REFERENCE WAVEFORM



290448–14

A.C. test inputs are driven at V_{OH} (2.4 V_{TTL}) for a Logic "1" and V_{OL} (0.45 V_{TTL}) for a logic "0". Input timing begins at V_{IH} (2.0 V_{TTL}) and V_{IL} (0.8 V_{TTL}). Output timing ends at V_{IH} and V_{IL}. Input rise and fall times (10% to 90%) < 10 ns.

STANDARD A.C. TESTING LOAD CIRCUIT

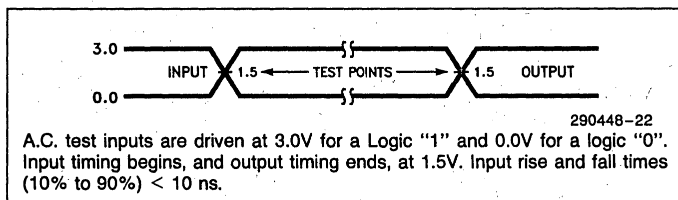


290448–13

$C_L = 100$ pF
C_L Includes Jig Capacitance
$R_L = 3.3$ KΩ

HIGH SPEED TEST CONFIGURATION(2)

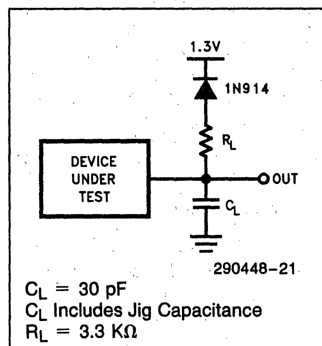
HIGH SPEED A.C. INPUT/OUTPUT REFERENCE WAVEFORM



290448–22

A.C. test inputs are driven at 3.0V for a Logic "1" and 0.0V for a logic "0". Input timing begins, and output timing ends, at 1.5V. Input rise and fall times (10% to 90%) < 10 ns.

HIGH SPEED A.C. TESTING LOAD CIRCUIT



290448–21

$C_L = 30$ pF
C_L Includes Jig Capacitance
$R_L = 3.3$ KΩ

NOTES:
1. Testing characteristics for 28F200BX-60/28F002BX-60 in standard test configuration and 28F200BX-80/28F002BX-80.
2. Testing characteristics for 28F200BX-60/28F002BX-60 in high speed test configuration.

DC CHARACTERISTICS: EXTENDED TEMPERATURE OPERATION (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{CCW}	V_{CC} Word Write Current	1			75	mA	Word Write in Progress
I_{CCW}	V_{CC} Byte Write Current	1			65	mA	Byte Write in Progress
I_{CCE}	V_{CC} Block Erase Current	1			40	mA	Block Erase in Progress
I_{CCES}	V_{CC} Erase Suspend Current	1, 2		5	10	mA	Block Erase Suspended, $\overline{CE} = V_{IH}$

AC CHARACTERISTICS—Read Only Operations(1)

Versions				V_{CC} ±5%		V_{CC} ± 10%				Unit
				28F200BX-60(4) 28F002BX-60(4)		28F200BX-60(5) 28F002BX-60(5)		28F200BX-80(5) 28F002BX-80(5)		
Symbol		Parameter	Notes	Min	Max	Min	Max	Min	Max	
t_{AVAV}	t_{RC}	Read Cycle Time		60		70		80		ns
t_{AVQV}	t_{ACC}	Address to Output Delay			60		70		80	ns
t_{ELQV}	t_{CE}	$\overline{CE}$ to Output Delay	2		60		70		80	ns
t_{PHQV}	t_{PWH}	$\overline{PWD}$ High to Output Delay			300		300		300	ns
t_{GLQV}	t_{OE}	$\overline{OE}$ to Output Delay	2		30		35		40	ns
t_{ELQX}	t_{LZ}	$\overline{CE}$ to Output Low Z	3	0		0		0		ns
t_{EHQZ}	t_{HZ}	$\overline{CE}$ High to Output High Z	3		20		25		30	ns
t_{GLQX}	t_{OLZ}	$\overline{OE}$ to Output Low Z	3	0		0		0		ns
t_{GHQZ}	t_{DF}	$\overline{OE}$ High to Output High Z	3		20		25		30	ns
	t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change, Whichever is First	3	0		0		0		ns
t_{ELFL} t_{ELFH}		$\overline{CE}$ to $\overline{BYTE}$ Switching Low or High	3		5		5		5	ns
t_{FHQV}		$\overline{BYTE}$ Switching High to Valid Output Delay	3, 6		60		70		80	ns
t_{FLQZ}		$\overline{BYTE}$ Switching Low to Output High Z	3		20		25		30	ns

NOTES:
1. See A.C. Input/Output Reference Waveform for timing measurements.
2. $\overline{OE}$ may be delayed up to $t_{CE}-t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.
3. Sampled, not 100% tested.
4. See High Speed Test Configuration.
5. See Standard Test Configuration.
6. t_{FLQV}, $\overline{BYTE}$ switching low to valid output delay, will be equal to t_{AVQV}, measured from the time DQ_{15}/A_7 becomes valid.

EXTENDED TEMPERATURE OPERATIONS
AC CHARACTERISTICS—Read Only Operations(1):

Versions				T28F200BX-90(4) T28F002BX-90(4)		Unit
Symbol		Parameter	Notes	Min	Max	
t_{AVAV}	t_{RC}	Read Cycle Time		90		ns
t_{AVQV}	t_{ACC}	Address to Output Delay			90	ns
t_{ELQV}	t_{CE}	$\overline{CE}$ to Output Delay	2		90	ns
t_{PHQV}	t_{PWH}	$\overline{PWD}$ High to Output Delay			300	ns
t_{GLQV}	t_{OE}	$\overline{OE}$ to Output Delay	2		45	ns
t_{ELQX}	t_{LZ}	$\overline{CE}$ to Output Low Z	3	0		ns
t_{EHQZ}	t_{HZ}	$\overline{CE}$ High to Output High Z	3		35	ns
t_{GLQX}	t_{OLZ}	$\overline{OE}$ to Output Low Z	3	0		ns
t_{GHQZ}	t_{DF}	$\overline{OE}$ High to Output High Z	3		35	ns
	t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change, Whichever is First	3	0		ns
t_{ELFL} t_{ELFH}		$\overline{CE}$ to $\overline{BYTE}$ Switching Low or High	3		5	ns
t_{FHQV}		$\overline{BYTE}$ Switching High to Valid Output Delay	3, 5		90	ns
t_{FLQZ}		$\overline{BYTE}$ Switching Low to Output High Z	3		35	ns

NOTES:
1. See AC Input/Output Reference Waveform for timing measurements.
2. $\overline{OE}$ may be delayed up to $t_{CE}-t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.
3. Sampled, not 100% tested.
4. See Standard Test Configuration.
5. t_{FLQV}, $\overline{BYTE}$ switching low to valid output delay, will be equal to t_{AVQV}, measured from the time DQ_5/A_1 becomes valid.

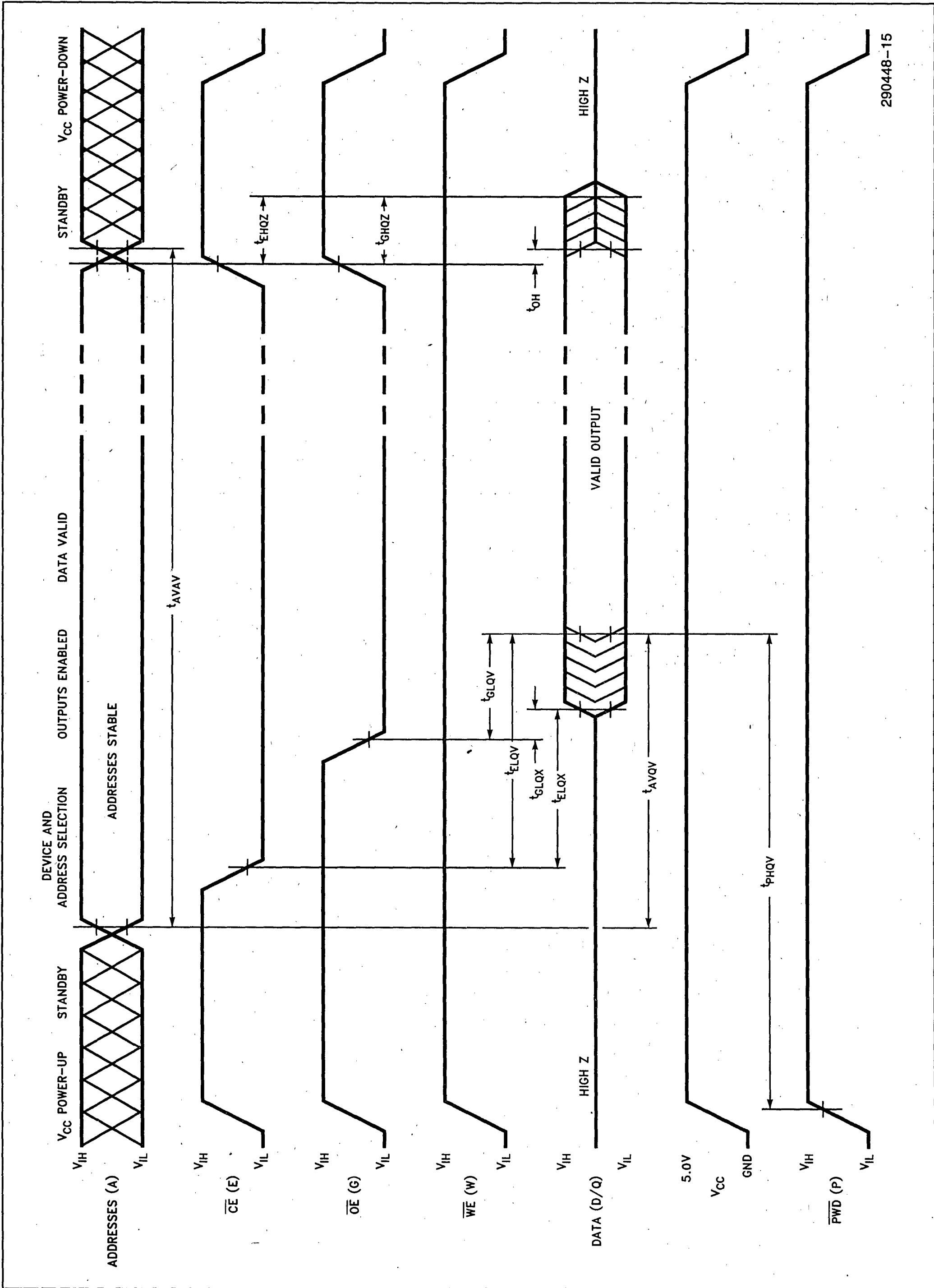


Figure 16. A.C. Waveforms for Read Operations

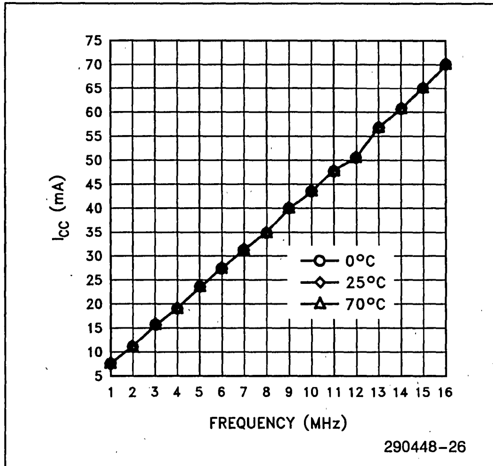


Figure 17. I_{CC} (RMS) vs Frequency (V_{CC} = 5.5V) for x16 Operation

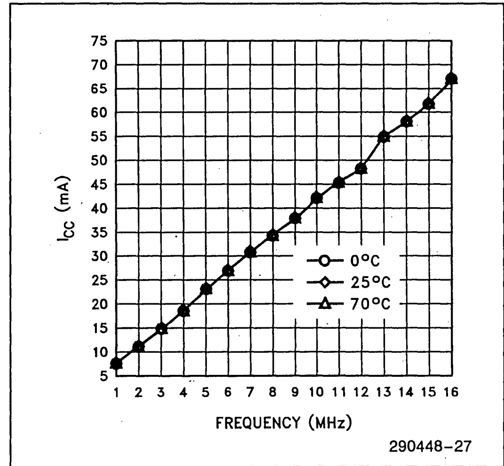


Figure 18. I_{CC} (RMS) vs Frequency (V_{CC} = 5.5V) for x8 Operation

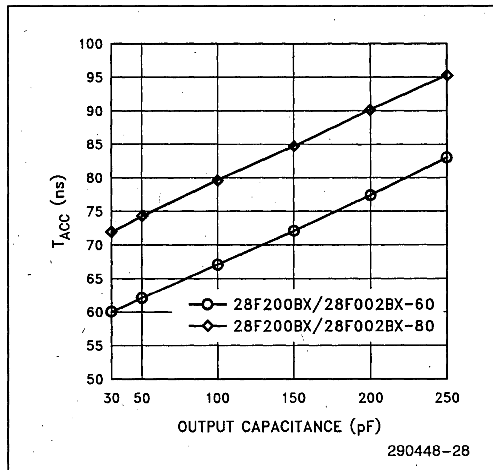


Figure 19. T_{ACC} vs Output Load Capacitance (V_{CC} = 4.5V, T = 70°C)

3

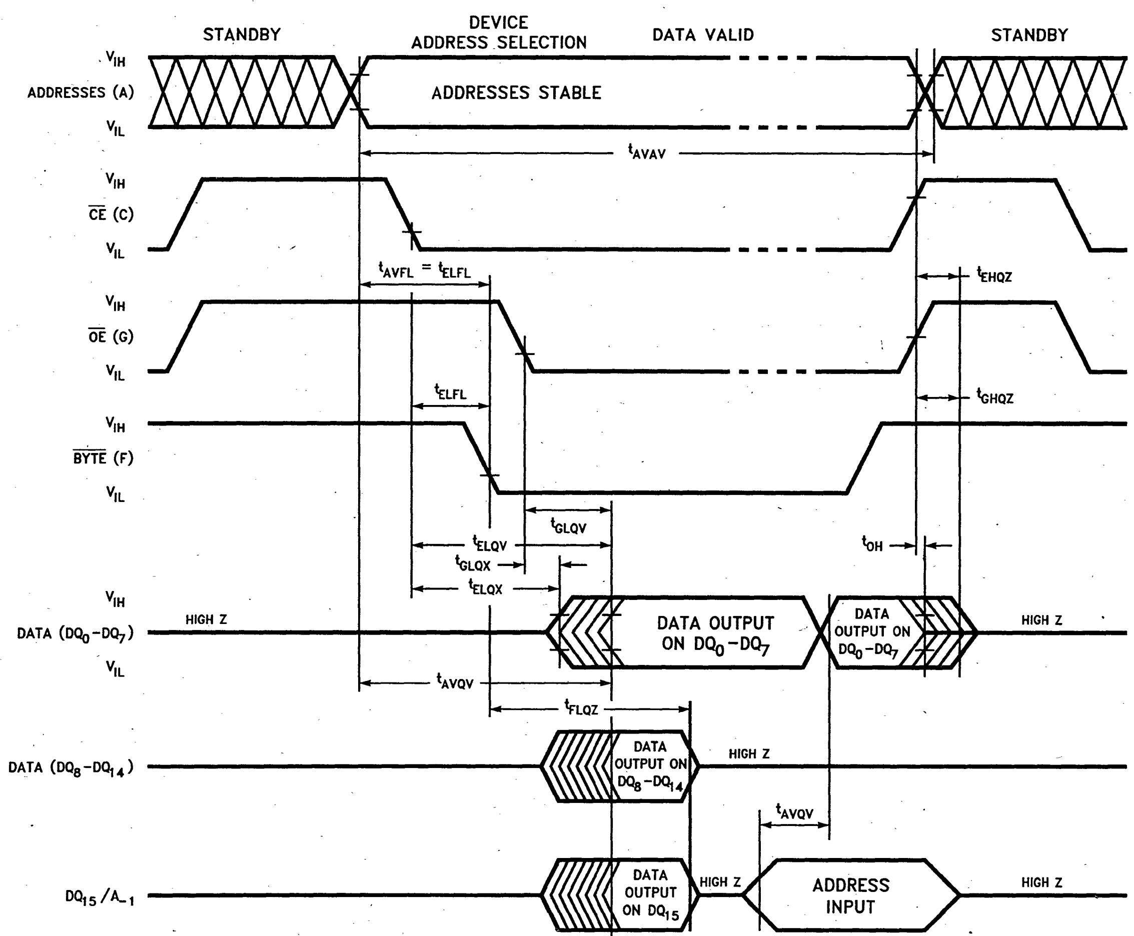


Figure 20. BYTE Timing for Both Read and Write Operations for 28F200BX

A.C. CHARACTERISTICS For $\overline{WE}$-Controlled Write Operations(1)

Versions				V_{CC} ±5%		V_{CC} ± 10%				Unit
				28F200BX-60(9) 28F002BX-60(9)		28F200BX-60(10) 28F002BX-60(10)		28F200BX-80(10) 28F002BX-80(10)		
Symbol		Parameter	Notes	Min	Max	Min	Max	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		60		70		80		ns
t_{PHWL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{WE}$ Going Low		215		215		215		ns
t_{ELWL}	t_{CS}	$\overline{CE}$ Setup to $\overline{WE}$ Going Low		0		0		0		ns
t_{PHHWH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{WE}$ Going High	6, 8	100		100		100		ns
t_{VPWH}	t_{VPS}	V_{PP} Setup to $\overline{WE}$ Going High	5, 8	100		100		100		ns
t_{AVWH}	t_{AS}	Address Setup to $\overline{WE}$ Going High	3	50		50		50		ns
t_{DVWH}	t_{DS}	Data Setup to $\overline{WE}$ Going High	4	60		60		60		ns
t_{WLWH}	t_{WP}	$\overline{WE}$ Pulse Width		50		50		50		ns
t_{WHDX}	t_{DH}	Data Hold from $\overline{WE}$ High	4	0		0		0		ns
t_{WHAX}	t_{AH}	Address Hold from $\overline{WE}$ High	3	10		10		10		ns
t_{WHEH}	t_{CH}	$\overline{CE}$ Hold from $\overline{WE}$ High		10		10		10		ns
t_{WHWL}	t_{WPH}	$\overline{WE}$ Pulse Width High		10		20		30		ns
t_{WHQV1}		Duration of Word/Byte Write Operation	2, 5	6		6		6		μs
t_{WHQV2}		Duration of Erase Operation (Boot)	2, 5, 6	0.3		0.3		0.3		s
t_{WHQV3}		Duration of Erase Operation (Parameter)	2, 5	0.3		0.3		0.3		s
t_{WHQV4}		Duration of Erase Operation (Main)	2, 5, 6	0.6		0.6		0.6		s
t_{QWL}	t_{VPH}	V_{PP} Hold from Valid SRD	5, 8	0		0		0		ns

AC CHARACTERISTICS For $\overline{WE}$-Controlled Write Operations(1) (Continued)

Versions				V_{CC} ±5%		V_{CC} ± 10%				Unit
				28F200BX-60(9) 28F002BX-60(9)		28F200BX-60(10) 28F002BX-60(10)		28F200BX-80(10) 28F002BX-80(10)		
Symbol		Parameter	Notes	Min	Max	Min	Max	Min	Max	
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	6, 8	0		0		0		ns
t_{PHBR}		Boot-Block Relock Delay	7, 8		100		100		100	ns

NOTES:
1. Read timing characteristics during write and erase operations are the same as during read-only operations. Refer to A.C. characteristics during Read Mode.
2. The on-chip WSM completely automates program/erase operations; program/erase algorithms are now controlled internally which includes verify and margining operations.
3. Refer to command definition table for valid A_{IN}.
4. Refer to command definition table for valid D_{IN}.
5. Program/Erase durations are measured to valid SRD data (successful operation, SR.7 = 1).
6. For Boot Block Program/Erase, $\overline{PWD}$ should be held at V_{HH} until operation completes successfully.
7. Time t_{PHBR} is required for successful relocking of the Boot Block.
8. Sampled but not 100% tested.
9. See High Speed Test Configuration.
10. See Standard Test Configuration.

BLOCK ERASE AND WORD/BYTE WRITE PERFORMANCE: V_{PP} = 12.0V ±5%

Parameter	Notes	28F200BX-60 28F002BX-60			28F200BX-80 28F002BX-80			Unit
		Min	Typ(1)	Max	Min	Typ(1)	Max	
Boot/Parameter Block Erase Time	2		1.0	7		1.0	7	s
Main Block Erase Time	2		2.4	14		2.4	14	s
Main Block Byte Program Time	2		1.2	4.2		1.2	4.2	s
Main Block Word Program Time	2		0.6	2.1		0.6	2.1	s

NOTES:
1. 25°C
2. Excludes System-Level Overhead.

BLOCK ERASE AND WORD/BYTE WRITE PERFORMANCE: V_{PP} = 12.0V ±10%

Parameter	Notes	28F200BX-60 28F002BX-60			28F200BX-80 28F002BX-80			Unit
		Min	Typ(1)	Max	Min	Typ(1)	Max	
Boot/Parameter Block Erase Time	2		5.8	40		5.8	40	s
Main Block Erase Time	2		14	60		14	60	s
Main Block Byte Program Time	2		6.0	20		6.0	20	s
Main Block Word Program Time	2		3.0	10		3.0	10	s

NOTES:
1. 25°C
2. Excludes System-Level Overhead.

EXTENDED TEMPERATURE OPERATION
AC CHARACTERISTICS For $\overline{WE}$-Controlled Write Operations[1]:

Versions[4]				T28F200BX-90[9] T28F002BX-90[9]		Unit
Symbol		Parameter	Notes	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		90		ns
t_{PHWL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{WE}$ Going Low		210		ns
t_{ELWL}	t_{CS}	$\overline{CE}$ Setup to $\overline{WE}$ Going Low		0		ns
t_{PHHWH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{WE}$ Going High	6, 8	100		ns
t_{VPWH}	t_{VPS}	V_{PP} Setup to $\overline{WE}$ Going High	5, 8	100		ns
t_{AVWH}	t_{AS}	Address Setup to $\overline{WE}$ Going High	3	60		ns
t_{DVWH}	t_{DS}	Data Setup to $\overline{WE}$ Going High	4	60		ns
t_{WLWH}	t_{WP}	$\overline{WE}$ Pulse Width		60		ns
t_{WHDX}	t_{DH}	Data Hold from $\overline{WE}$ High	4	0		ns
t_{WHAX}	t_{AH}	Address Hold from $\overline{WE}$ High	3	10		ns
t_{WHEH}	t_{CH}	$\overline{CE}$ Hold from $\overline{WE}$ High		10		ns
t_{WHWL}	t_{WPH}	$\overline{WE}$ Pulse Width High		30		ns
t_{WHQV1}		Duration of Word/Byte Write Operation	2, 5	7		µs
t_{WHQV2}		Duration of Erase Operation (Boot)	2, 5, 6	0.4		s
t_{WHQV3}		Duration of Erase Operation (Parameter)	2, 5	0.4		s
t_{WHQV4}		Duration of Erase Operation (Main)	2, 5, 6	0.7		s
t_{QWL}	t_{VPH}	V_{PP} Hold from Valid SRD	5, 8	0		ns
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	6, 8	0		ns
t_{PHBR}		Boot-Block Relock Delay	7, 8		100	ns

NOTES:
1. Read timing characteristics during write and erase operations are the same as during read-only operations. Refer to A.C. characteristics during Read Mode.
2. The on-chip WSM completely automates program/erase operations; program/erase algorithms are now controlled internally which includes verify and margining operations.
3. Refer to command definition table for valid A_{IN}.
4. Refer to command definition table for valid D_{IN}.
5. Program/Erase durations are measured to valid SRD data (successful operation, SR.7 = 1).
6. For Boot Block Program/Erase, $\overline{PWD}$ should be held at V_{HH} until operation completes successfully.
7. Time t_{PHBR} is required for successful relocking of the Boot Block.
8. Sampled but not 100% tested.
9. See Standard Test Configuration.

EXTENDED TEMPERATURE OPERATION
BLOCK ERASE AND WORD/BYTE WRITE PERFORMANCE: V_{PP} = 12.0V ±5%

Parameter	Notes	T28F200BX-90 T28F002BX-90			Unit
		Min	Typ(1)	Max	
Boot/Parameter Block Erase Time	2		1.5	10.5	s
Main Block Erase Time	2		3.0	18	s
Main Block Byte Program Time	2		1.4	5.0	s
Main Block Word Program Time	2		0.7	2.5	s

NOTES:
1. 25°C, 12.0V V_{PP}.
2. Excludes System-Level Overhead.

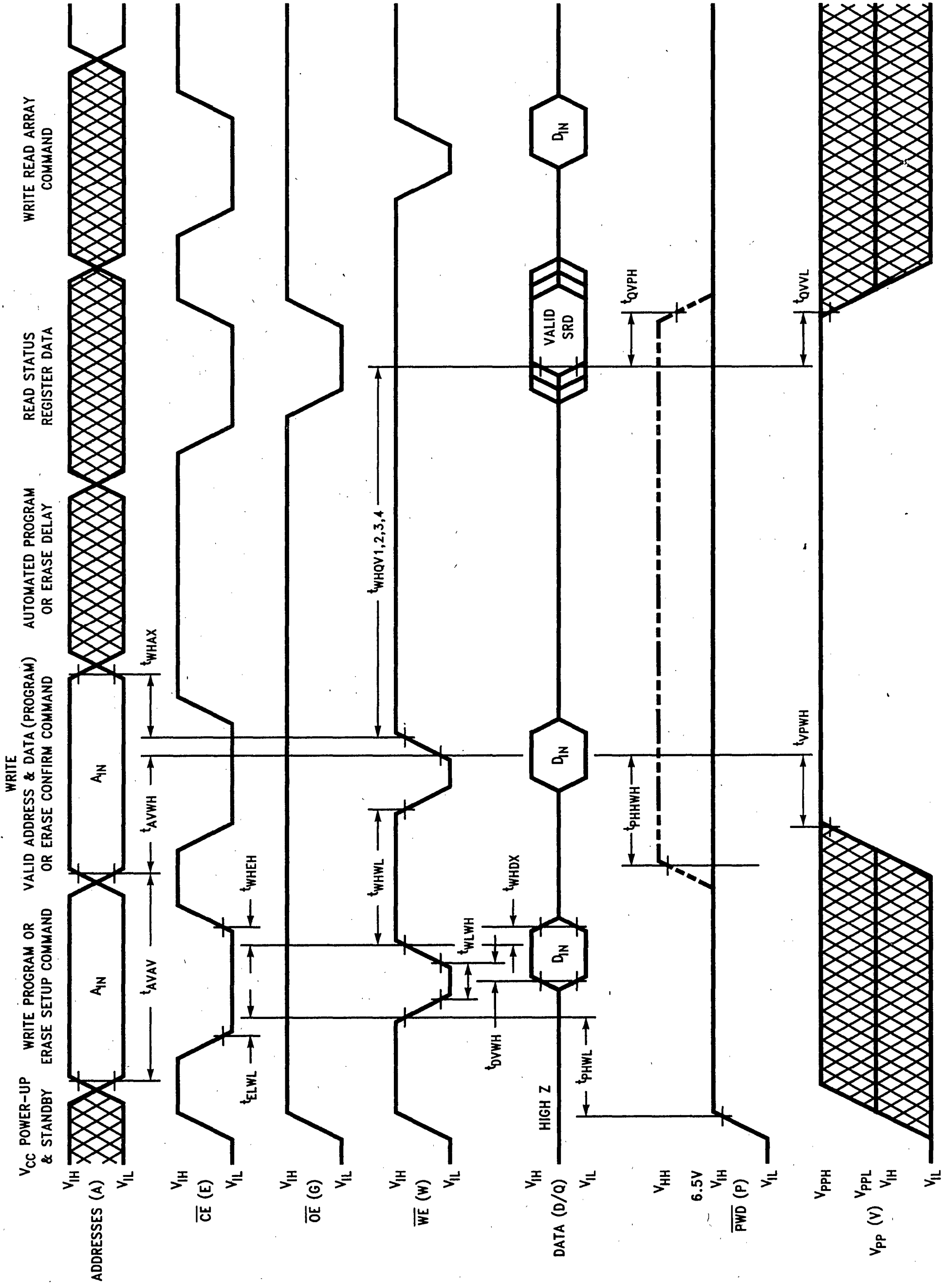


Figure 21. AC Waveforms for Write and Erase Operations ($\overline{WE}$-Controlled Writes)

AC CHARACTERISTICS FOR $\overline{CE}$-CONTROLLED WRITE OPERATIONS(1, 9)

Versions				V_{CC} ±5%		V_{CC} ±10%				Unit
				28F200BX-60(10) 28F002BX-60(10)		28F200BX-60(11) 28F002BX-60(11)		28F200BX-80(11) 28F002BX-80(11)		
Symbol		Parameter	Notes	Min	Max	Min	Max	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		60		70		80		ns
t_{PHEL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{CE}$ Going Low		215		215		215		ns
t_{WLEL}	t_{WS}	$\overline{WE}$ Setup to $\overline{CE}$ Going Low		0		0		0		ns
t_{PHHEH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{CE}$ Going High	6, 8	100		100		100		ns
t_{VPEH}	t_{VPS}	V_{PP} Setup to $\overline{CE}$ Going High	5, 8	100		100		100		ns
t_{AVEH}	t_{AS}	Address Setup to $\overline{CE}$ Going High	3	50		50		50		ns
t_{DVEH}	t_{DS}	Data Setup to $\overline{CE}$ Going High	4	60		60		60		ns
t_{ELEH}	t_{CP}	$\overline{CE}$ Pulse Width		50		50		50		ns
t_{EHDX}	t_{DH}	Data Hold from $\overline{CE}$ High	4	0		0		0		ns
t_{EHAX}	t_{AH}	Address Hold from $\overline{CE}$ High	3	10		10		10		ns
t_{EHWH}	t_{WH}	$\overline{WE}$ Hold from $\overline{CE}$ High		10		10		10		ns
t_{EHEL}	t_{CPH}	$\overline{CE}$ Pulse Width High		10		20		30		ns
t_{EHQV1}		Duration of Word/Byte Programming Operation	2, 5	6		6		6		µs
t_{EHQV2}		Duration of Erase Operation (Boot)	2, 5, 6	0.3		0.3		0.3		s
t_{EHQV3}		Duration of Erase Operation (Parameter)	2, 5	0.3		0.3		0.3		s
t_{EHQV4}		Duration of Erase Operation (Main)	2, 5	0.6		0.6		0.6		s
t_{QWL}	t_{VPH}	V_{PP} Hold from Valid SRD	5, 8	0		0		0		ns
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	6, 8	0		0		0		ns
t_{PHBR}		Boot-Block Relock Delay	7		100		100		100	ns

NOTES:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of $\overline{CE}$ and $\overline{WE}$ in systems where $\overline{CE}$ defines the write pulse-width (within a longer $\overline{WE}$ timing waveform), all set-up, hold and inactive $\overline{WE}$ time should be measured relative to the $\overline{CE}$ waveform.
2, 3, 4, 5, 6, 7, 8: Refer to A.C. Characteristics notes for $\overline{WE}$-Controlled Write Operations.
9. Read timing characteristics during write and erase operations are the same as during read-only operations. Refer to A.C. Characteristics during Read Mode.
10. See High Speed Test Configuration.
11. See Standard Test Configuration.

EXTENDED TEMPERATURE OPERATION
AC CHARACTERISTICS FOR $\overline{CE}$-CONTROLLED WRITE OPERATIONS(1, 9)

Versions				T28F200BX-90(10) T28F002BX-90(10)		Unit
Symbol		Parameter	Notes	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		90		ns
t_{PHEL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{CE}$ Going Low		210		ns
t_{WLEL}	t_{WS}	$\overline{WE}$ Setup to $\overline{CE}$ Going Low		0		ns
t_{PHHEH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{CE}$ Going High	6, 8	100		ns
t_{VPEH}	t_{VPS}	V_{PP} Setup to $\overline{CE}$ Going High	5, 8	100		ns
t_{AVEH}	t_{AS}	Address Setup to $\overline{CE}$ Going High	3	60		ns
t_{DVEH}	t_{DS}	Data Setup to $\overline{CE}$ Going High	4	60		ns
t_{ELEH}	t_{CP}	$\overline{CE}$ Pulse Width		60		ns
t_{EHDX}	t_{DH}	Data Hold from $\overline{CE}$ High	4	0		ns
t_{EHAX}	t_{AH}	Address Hold from $\overline{CE}$ High	3	10		ns
t_{EHWH}	t_{WH}	$\overline{WE}$ Hold from $\overline{CE}$ High		10		ns
t_{EHEL}	t_{CPH}	$\overline{CE}$ Pulse Width High		30		ns
t_{EHQV1}		Duration of Word/Byte Programming Operation	2, 5	7		μs
t_{EHQV2}		Duration of Erase Operation (Boot)	2, 5, 6	0.4		s
t_{EHQV3}		Duration of Erase Operation (Parameter)	2, 5	0.4		s
t_{EHQV4}		Duration of Erase Operation (Main)	2, 5	0.7		s
t_{QWL}	t_{VPH}	V_{PP} Hold from Valid SRD	5, 8	0		ns
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	6, 8	0		ns
t_{PHBR}		Boot-Block Relock Delay	7		100	ns

NOTES:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of $\overline{CE}$ and $\overline{WE}$ in systems where $\overline{CE}$ defines the write pulse-width (within a longer $\overline{WE}$ timing waveform), all set-up, hold and inactive $\overline{WE}$ time should be measured relative to the $\overline{CE}$ waveform.
2, 3, 4, 5, 6, 7, 8: Refer to AC Characteristics for $\overline{WE}$-Controlled Write Operations.
9. Read timing characteristics during write and erase operations are the same as during read-only operations. Refer to AC Characteristics during Read Mode.
10. See Standard Test Configuration.

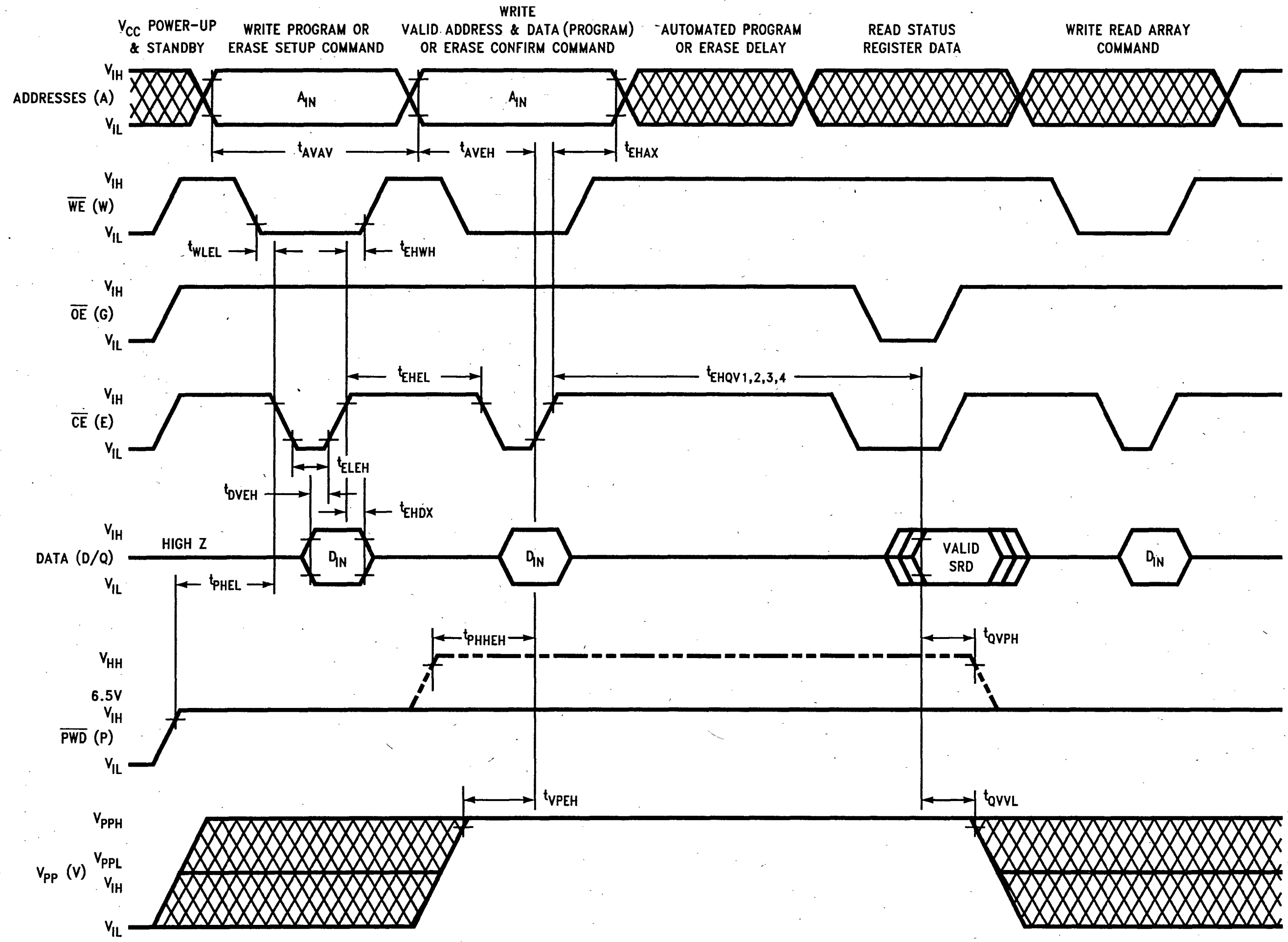


Figure 22. Alternate AC Waveforms for Write and Erase Operations ($\overline{CE}$-Controlled Writes)

ORDERING INFORMATION

	E	2	8	F	2	0	0	B	X	-	6	0

OPERATING TEMPERATURE
T = EXTENDED TEMP
BLANK = COMMERCIAL TEMP

PACKAGE
E = STANDARD 56 LEAD TSOP
PA = 44 LEAD PSOP

ACCESS SPEED (ns)
60 ns
80 ns
90 ns

290448–18

Valid Combinations:

E28F200BX-60	PA28F200BX-60	TE28F200BX-T90	TPA28F200BX-T90
E28F200BX-80	PA28F200BX-80	TE28F200BX-B90	TPA28F200BX-B90

	E	2	8	F	0	0	2	B	X	-	6	0

OPERATING TEMPERATURE
T = EXTENDED TEMP
BLANK = COMMERCIAL TEMP

PACKAGE
E = STANDARD 40 LEAD TSOP

ACCESS SPEED (ns)
60 ns
80 ns
90 ns

290448–23

Valid Combinations:

E28F002BX-60	TE28F002BX-T90
E28F002BX-80	TE28F002BX-B90

ADDITIONAL INFORMATION

	Order Number
28F400BX/28F004BX Datasheet	290451
28F200BXL/28F002BXL Datasheet	290449
28F400BXL/28F004BXL Datasheet	290450
AP-363 "Extended Flash BIOS Design for Portable Computers"	292098
ER-28 "ETOX-III Flash Memory Technology"	204012
ER-29 "The Intel 2/4-Mbit Boot Block Flash Memory Family"	294013

REVISION HISTORY

Number	Description
-002	Removed −70 speed bin Integrated −70 characteristics into −60 speed bin Added Extended Temperature characteristics Modified $\overline{BYTE}$ Timing Diagram Improved t_{PHQV}, $\overline{PWD}$ High to Output Delay and t_{PHEL}, $\overline{PWD}$ High Recovery to $\overline{CE}$ going low specifications

28F200BX-TL/BL, 28F002BX-TL/BL 2 MBIT (128K x 16, 256K x 8) LOW POWER BOOT BLOCK FLASH MEMORY FAMILY

- **Low Voltage Operation for Very Low Power Portable Applications**
  - V_{CC} = 3.3V ±0.3V
- **x8/x16 Input/Output Architecture**
  - 28F200BX-TL, 28F200BX-BL
  - For High Performance and High Integration 16-bit and 32-bit CPUs
- **x8-only Input/Output Architecture**
  - 28F002BX-TL, 28F002BX-BL
  - For Space Constrained 8-bit Applications
- **Optimized High Density Blocked Architecture**
  - One 16 KB Protected Boot Block
  - Two 8 KB Parameter Blocks
  - One 96 KB Main Block
  - One 128 KB Main Block
  - Top or Bottom Boot Locations
- **Extended Cycling Capability**
  - 10,000 Block Erase Cycles
- **Automated Word/Byte Write and Block Erase**
  - Command User Interface
  - Status Registers
  - Erase Suspend Capability
- **SRAM-Compatible Write Interface**
- **Automatic Power Savings Feature**
  - 0.8 mA Typical I_{CC} Active Current in Static Operation
- **Very High-Performance Read**
  - 150 ns Maximum Access Time
  - 65 ns Maximum Output Enable Time
- **Low Power Consumption**
  - 10 mA Typical x8 Active Read Current
  - 12 mA Typical x16 Active Read Current
- **Deep Power-Down/Reset Input**
  - 0.2 μA I_{CC} Typical
  - Acts as Reset for Boot Operations
- **Write Protection for Boot Block**
- **Hardware Data Protection Feature**
  - Erase/Write Lockout during Power Transitions
- **Industry Standard Surface Mount Packaging**
  - 28F200BX-L: JEDEC ROM Compatible 44-Lead PSOP 56-Lead TSOP
  - 28F002BX-L: 40-Lead TSOP
- **12V Word/Byte Write and Block Erase**
  - V_{PP} = 12V ±5% Standard
- **ETOX™ III Flash Technology**
  - 3.3V Read
- **Independent Software Vendor Support**
  - SystemSoft* Flash BIOS

SystemSoft is a trademark of SystemSoft Corporation.

November 1992
Order Number: 290449-002

Intel's 2 Mbit Low Power Flash Memory Family is an extension of the Boot Block Architecture which includes block-selective erasure, automated write and erase operations and standard microprocessor interface. The 2 Mbit Flash Memory Family enhances the Boot Block Architecture by adding more density and blocks, x8/x16 input/output control, very low power, very high speed, an industry standard ROM compatible pinout and surface mount packaging. The 2 Mbit Low Power Flash Family opens a new capability for 3V battery-operated portable systems and allows for an easy upgrade to Intel's 4 Mbit Low Power Boot Block Flash Memory Family.

The Intel 28F200BX-TL/BL are 16-bit wide flash memory offerings. These high density flash memories provide user selectable bus operation for either 8-bit or 16-bit applications. The 28F200BX-TL and 28F200BX-BL are 2,097,152-bit non-volatile memories organized as either 262,144 bytes or 131,072 words of information. They are offered in 44-Lead plastic SOP and 56-Lead TSOP packages. The x8/x16 pinout conforms to the industry standard ROM/EPROM pinout.

The Intel 28F002BX-TL/BL are 8-bit wide flash memories with 2,097,152 bits organized as 262,144 bytes of information. They are offered in a 40-Lead TSOP package, which is ideal for space-constrained portable systems.

These devices use an integrated Command User Interface (CUI) and Write State Machine (WSM) for simplified word/byte write and block erasure. The 28F200BX-TL/28F002BX-TL provide block locations compatible with Intel's low voltage MCS-186 family, i386™, i486™ microprocessors. The 28F200BX-BL/28F002BX-BL provide compatibility with Intel's 80960KX and 80960SX families as well as other low voltage embedded microprocessors.

The boot block includes a data protection feature to protect the boot code in critical applications. With a maximum access time of 150 ns, these 2 Mbit flash devices are very high performance low power memories which interface to a wide range of low power microprocessors and microcontrollers. A deep power-down mode lowers the total V_{CC} power consumption to 0.66 μW. This is critical in handheld battery powered systems such as Handy Phones. For very high speed applications using a 5V supply, refer to the Intel 28F200BX-T/B, 28F002BX-T/B 2 Mbit Boot Block Flash Memory Family datasheet.

Manufactured on Intel's 0.8 micron ETOX™ III process, the 2 Mbit low power flash memory family provides world class quality, reliability and cost-effectiveness at the 2 Mbit density level.

1.0 PRODUCT FAMILY OVERVIEW

Throughout this datasheet 28F200BX-L refers to both the 28F200BX-TL and 28F200BX-BL devices and 28F002BX-L refers to both the 28F002BX-TL and 28F002BX-BL devices. The 2 Mbit flash family refers to both the 28F200BX-L and 28F002BX-L products. This datasheet comprises the specifications for four separate products in the 2 Mbit flash memory family. Section 1 provides an overview of the 2 Mbit flash memory family including applications, pinouts and pin descriptions. Sections 2 and 3 describe in detail the specific memory organizations for the 28F200BX-L and 28F002BX-L products respectively. Section 4 combines a description of the family's principles of operations. Finally, section 5 describes the family's operating specifications.

PRODUCT FAMILY

x8/x16 Products	x8-Only Products
28F200BX-TL 28F200BX-BL	28F002BX-TL 28F002BX-BL

1.1 Main Features

The 28F200BX-L/28F002BX-L low power boot block flash memory family is a very low power and very high performance 2 Mbit (2,097,152 bit) memory family organized as either 128 Kwords (131,072 words) of 16 bits each or 256 Kbytes (262,144 bytes) of 8 bits each.

Five Separately Erasable Blocks including a **Hardware-Lockable boot block** (16,384 Bytes), **two parameter blocks** (8,192 Bytes each) and **two main blocks** (1 block of 98,304 Bytes and 1 block of 131,072 Bytes) are included on the 2 Mbit family. An erase operation erases one of the 5 blocks in typically 3.4 seconds and the boot or parameter blocks in typically 2.0 seconds, independent of the remaining blocks. Each block can be independently erased and programmed 10,000 times.

The Boot Block is located at either the top (28F200BX-TL, 28F002BX-TL) or the bottom (28F200BX-BL, 28F002BX-BL) of the address map in order to accommodate different microprocessor protocols for boot code location. The **hardware lockable boot block** provides the most secure code storage. The boot block is intended to store the kernel code required for booting-up a system. When the $\overline{PWD}$ pin is between 11.4V and 12.6V the boot block is unlocked and program and erase operations can be performed. When the $\overline{PWD}$ pin is at or below 4.1V the boot block is locked and program and erase operations to the boot block are ignored.

The 28F200BX-L products are available in the ROM/EPROM compatible pinout and housed in the 44-Lead PSOP (Plastic Small Outline) package and the 56-Lead TSOP (Thin Small Outline, 1.2 mm thick) package as shown in Figures 3 and 4. The 28F002BX-L products are available in the 40-Lead TSOP (1.2 mm thick) package as shown in Figure 5.

The **Command User Interface (CUI)** serves as the interface between the microprocessor or microcontroller and the internal operation of the 28F200BX-L and 28F002BX-L flash products.

Program and Erase Automation allow program and erase operations to be executed using a two-write command sequence to the CUI. The internal Write State Machine (WSM) automatically executes the algorithms and timings necessary for program and erase operations, including verifications, thereby unburdening the microprocessor or microcontroller. Writing of memory data is performed in word or byte increments for the 28F200BX-L family and in byte increments for the 28F002BX-L family typically within 11 µs.

The **Status Register (SR)** indicates the status of the WSM and whether the WSM successfully completed the desired program or erase operation.

Maximum Access Time of **150 ns (T_{ACC})** is achieved over the commercial temperature range (0°C to +70°C), over V_{CC} supply voltage range (3.0V to 3.6V, 4.5V to 5.5V) and 50 pF output load.

I_{PP} Program current is 40 mA for x16 operation and 30 mA for x8 operation. I_{PP} Erase current is 30 mA maximum. V_{PP} erase and programming voltage is 11.4V to 12.6V (V_{PP} = 12V ±5%) under all operating conditions.

Typical I_{CC} Active Current of 12 mA is achieved for the x16 products (28F200BX-L), **typical I_{CC} Active Current of 10 mA** is achieved for the x8 products (28F200BX-L, 28F002BX-L).

The 2 Mbit flash family is also designed with an Automatic Power Savings (APS) feature to minimize system battery current drain and allow for extremely low power designs. Once the device is accessed to read the array data, APS mode will immediately put the memory in static mode of operation where I_{CC} active current is typically 0.8 mA until the next read is initiated.

When the $\overline{CE}$ and $\overline{PWD}$ pins are at V_{CC} and the $\overline{BYTE}$ pin (28F200BX-L-only) is at either V_{CC} or GND the **CMOS Standby** mode is enabled where **I_{CC}** is typically **40 μA.**

A **Deep Power-down Mode** is enabled when the $\overline{PWD}$ pin is at ground minimizing power consumption and providing write protection during power-up conditions. **I_{CC} current** during deep power-down mode is **0.20 μA typical**. An initial maximum access time or Reset Time of 700 ns is required from $\overline{PWD}$ switching until outputs are valid. Equivalently, the device has a maximum wake-up time of 580 ns until writes to the Command User Interface are recognized. When $\overline{PWD}$ is at ground the WSM is reset, the Status Register is cleared and the entire device is protected from being written to. This feature prevents data corruption and protects the code stored in the device during system reset. The system Reset pin can be tied to $\overline{PWD}$ to reset the memory to normal read mode upon activation of the Reset pin. When the CPU enters reset mode, it expects to read the contents of a memory location. Furthermore, with on-chip program/erase automation in the 2 Mbit family and the $\overline{PWD}$ functionality for data protection, after the CPU is reset and even if a program or erase command is issued, the device will not recognize any operation until $\overline{PWD}$ returns to its normal state.

For the 28F200BX-L, Byte-wide or Word-wide Input/Output Control is possible by controlling the $\overline{BYTE}$ pin. When the $\overline{BYTE}$ pin is at a logic low the device is in the byte-wide mode (x8) and data is read and written through DQ[0:7]. During the byte-wide mode, DQ[8:14] are tri-stated and DQ_{15}/A_{-1} becomes the lowest order address pin. When the $\overline{BYTE}$ pin is at a logic high the device is in the word-wide mode (x16) and data is read and written through DQ[0:15].

1.2 Applications

The 2 Mbit low power boot block flash family combines high density, 3V operation, high performance, cost-effective flash memories with blocking and hardware protection capabilities. Its flexibility and versatility will reduce costs throughout the product life cycle. Flash memory is ideal for Just-In-Time production flow, reducing system inventory and costs, and eliminating component handling during the production phase. During the product life cycle, when code updates or feature enhancements become necessary, flash memory will reduce the update costs by allowing either a user-performed code change via floppy disk or a remote code change via a serial link. The 2 Mbit boot block flash family provides full function, blocked flash memories suitable for a wide range of applications. These applications include **Extended PC BIOS, Handy Digital Cellular Phone** program and data storage and various other portable embedded applications where both program and data storage are required.

Reprogrammable systems such as Notebook and Palmtop computers, are ideal applications for the 2 Mbit low power flash products. Portable and handheld personal computer applications are becoming more complex with the addition of power management software to take advantage of the latest microprocessor technology, the availability of ROM-based application software, pen tablet code for electronic handwriting, and diagnostic code. Figure 1 shows an example of a 28F200BX-TL application.

This increase in software sophistication augments the probability that a code update will be required after the PC is shipped. The 2 Mbit low power flash products provide an inexpensive update solution for the notebook and handheld personal computers while extending their product lifetime. Furthermore, the 2 Mbit flash products' deep power-down mode provides added flexibility for these battery-operated portable designs which require operation at extremely low power levels.

The 2 Mbit low power flash products also provide excellent design solutions for Handy Digital Cellular Phone applications requiring high density storage, high performance capabilities coupled with low voltage operation, and a small form factor package (x8-only bus). The 2 Mbit's blocking scheme allows for an easy segmentation of the embedded code with; 16 Kbytes of Hardware-Protected Boot code, 2 Main Blocks of program code and 2 Parameter Blocks of 8 Kbytes each for frequently updatable data storage and diagnostic messages (e.g., phone numbers, authorization codes). Figure 2 is an example of such an application with the 28F002BX-TL.

These are a few actual examples of the wide range of applications for the 2 Mbit Low Power Boot Block flash memory family which enables system designers to achieve the best possible product design. Only your imagination limits the applicability of such a versatile low power product family.

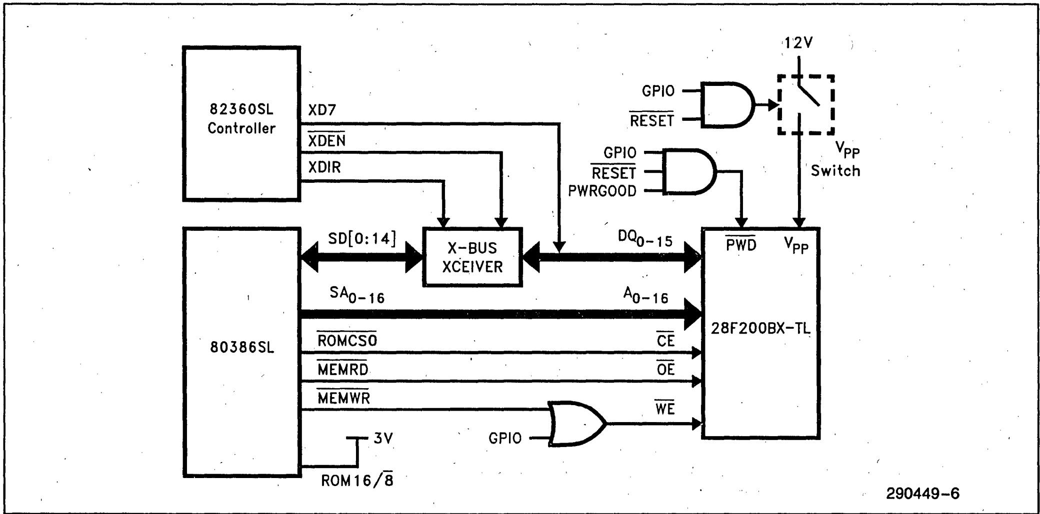


Figure 1. 28F200BX-TL Interface to INTEL386SL™ 3.3V Microprocessor Superset

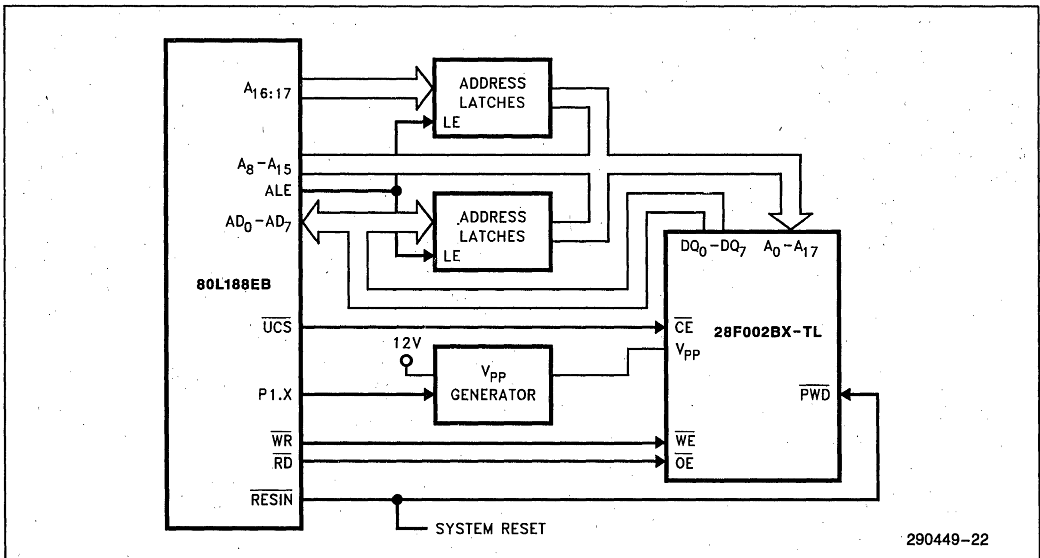


Figure 2. 28F002BX-TL Interface to INTEL 80L188EB, Low Voltage 8-Bit Embedded Microprocessor

1.3 Pinouts

The 28F200BX-L 44-Lead PSOP pinout follows the industry standard ROM/EPROM pinout as shown in Figure 3 with an upgrade to the 28F400BX-L (4 Mbit low power flash family). Furthermore, the 28F200BX-L 56-Lead TSOP pinout shown in Figure 4 provides density upgrades to the 28F400BX-L and to future higher density boot block memories.

The 28F002BX-L 40-Lead TSOP pinout shown in Figure 5 is 100% compatible and has a density upgrade to the 28F004BX-L 4 Mbit Low Power Boot Block flash memory.

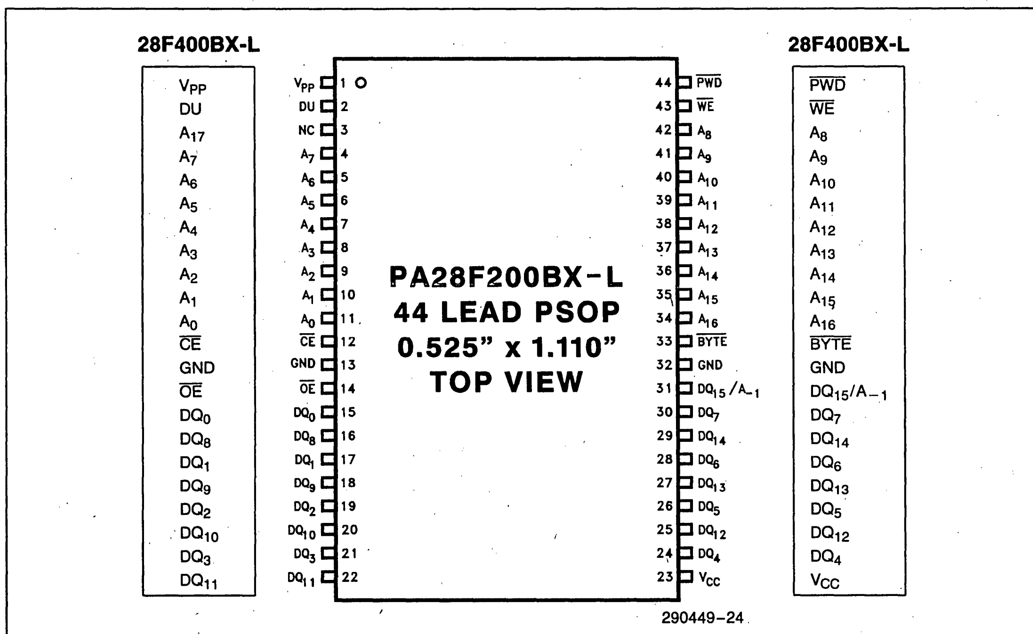


Figure 3. PSOP Lead Configuration for x8/x16 28F200BX-L

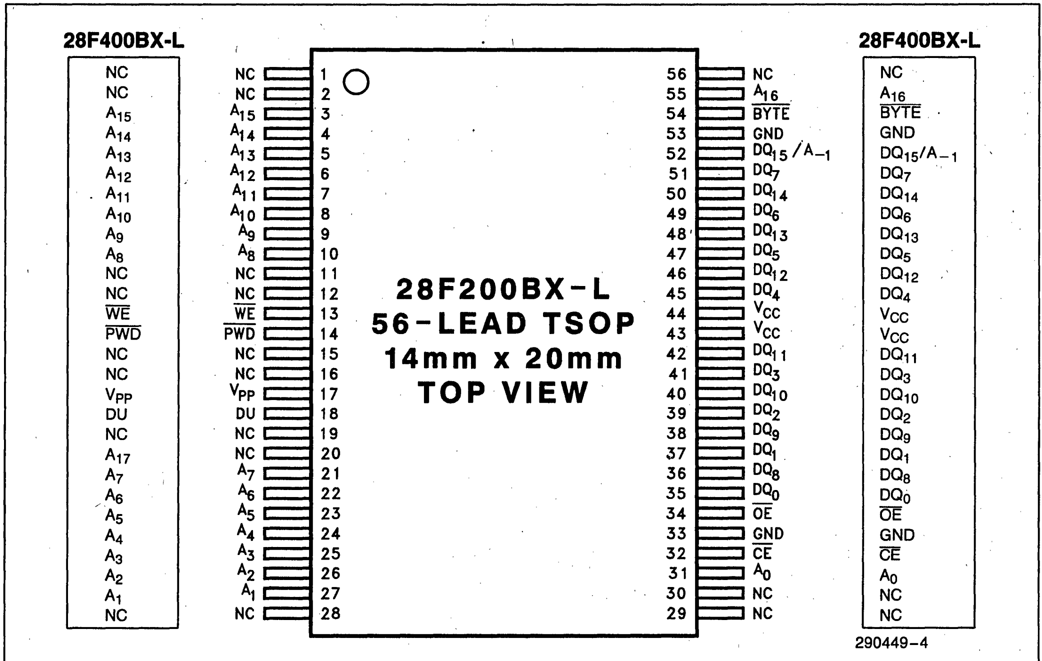


Figure 4. TSOP Lead Configuration for x8/x16 28F200BX-L

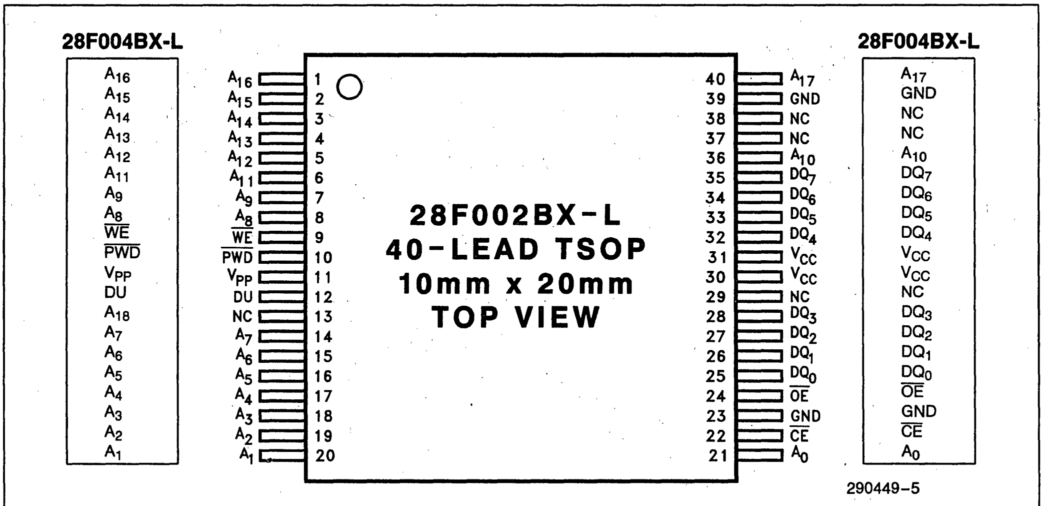


Figure 5. TSOP Lead Configuration for x8 28F002BX-L

1.4 Pin Descriptions for x8/x16 28F200BX-L

Symbol	Type	Name and Function
A_0–A_{16}	I	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
A_9	I	**ADDRESS INPUT:** When A_9 is at 12V the signature mode is accessed. During this mode A_0 decodes between the manufacturer and device ID's. When $\overline{BYTE}$ is at a logic low only the lower byte of the signatures are read. DQ_{15}/A_{-1} is a don't care in the signature mode when $\overline{BYTE}$ is low.
DQ_0–DQ_7	I/O	**DATA INPUTS/OUTPUTS:** Inputs array data on the second $\overline{CE}$ and $\overline{WE}$ cycle during a program command. Inputs commands to the command user interface when $\overline{CE}$ and $\overline{WE}$ are active. Data is internally latched during the write and program cycles. Outputs array, intelligent identifier and Status Register data. The data pins float to tri-state when the chip is deselected or the outputs are disabled.
DQ_8–DQ_{15}	I/O	**DATA INPUT/OUTPUTS:** Inputs array data on the second $\overline{CE}$ and $\overline{WE}$ cycle during a program command. Data is internally latched during the write and program cycles. Outputs array data. The data pins float to tri-state when the chip is deselected or the outputs are disabled as in the byte-wide mode ($\overline{BYTE}$ = "0"). In the byte-wide mode DQ_{15}/A_{-1} becomes the lowest order address for data output on DQ_0-DQ_7.
$\overline{CE}$	I	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels. If $\overline{CE}$ and $\overline{PWD}$ are high, but not at a CMOS high level, the standby current will increase due to current flow through the $\overline{CE}$ and $\overline{PWD}$ input stages.
$\overline{PWD}$	I	**POWER-DOWN:** Provides Three-State control. Puts the device in deep power-down mode. Locks the boot block from program/erase. When $\overline{PWD}$ is at logic high level and equals 4.1V maximum the boot block is locked and cannot be programmed or erased. When $\overline{PWD}$ = 11.4V minimum the boot block is unlocked and can be programmed or erased. When $\overline{PWD}$ is at a logic low level the boot block is locked, the deep power-down mode is enabled and the WSM is reset preventing any blocks from being programmed or erased, therefore providing data protection during power transitions. $\overline{PWD}$ terminates any internally timed erase or program activities when it is taken to a logic low. $\overline{PWD}$ activates the $\overline{CE}$ input stage and requires 700 ns recovery time to transition from deep power-down to valid data on the outputs or 580 ns delay before the device can recognize valid inputs.
$\overline{OE}$	I	**OUTPUT ENABLE:** Gates the device's outputs through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	I	**WRITE ENABLE:** Controls writes to the Command Register and array blocks. $\overline{WE}$ is active low. Addresses and data are latched on the rising edge of the $\overline{WE}$ pulse.

1.4 Pin Descriptions for x8/x16 28F200BX-L (Continued)

Symbol	Type	Name and Function
$\overline{BYTE}$	I	**$\overline{BYTE}$ ENABLE:** Controls whether the device operates in the byte-wide mode (x8) or the word-wide mode (x16). $\overline{BYTE}$ = "0" enables the byte-wide mode, where data is read and programmed on DQ_0-DQ_7 and DQ_{15}/A_{-1} becomes the lowest order address that decodes between the upper and lower byte. DQ_8-DQ_{14} are tri-stated during the byte-wide mode. $\overline{BYTE}$ = "1" enables the word-wide mode where data is read and programmed on DQ_0-DQ_{15}.
V_{PP}		**PROGRAM/ERASE POWER SUPPLY:** For erasing memory array blocks or programming data in each block. **Note:** $V_{PP} < V_{PPLMAX}$ memory contents cannot be altered.
V_{CC}		**DEVICE POWER SUPPLY (3.3V ±0.3V, 5V ±10%)**
GND		**GROUND:** For all internal circuitry.
NC		**NO CONNECT:** Pin may be driven or left floating.
DU		**DON'T USE PIN:** Pin should not be connected to anything.

1.5 Pin Descriptions for x8 28F002BX-L

Symbol	Type	Name and Function
A_0–A_{17}	I	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
A_9	I	**ADDRESS INPUT:** When A_9 is at 12V the signature mode is accessed. During this mode A_0 decodes between the manufacturer and device ID's.
DQ_0–DQ_7	I/O	**DATA INPUTS/OUTPUTS:** Inputs array data on the second $\overline{CE}$ and $\overline{WE}$ cycle during a program command. Inputs commands to the command user interface when $\overline{CE}$ and $\overline{WE}$ are active. Data is internally latched during the write and program cycles. Outputs array intelligent identifier and status register data. The data pins float to tri-state when the chip is deselected or the outputs are disabled.
$\overline{CE}$	I	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels.
$\overline{PWD}$	I	**POWER-DOWN:** Provides Three-State control. Puts the device in deep power-down mode. Locks the Boot Block from program/erase. When $\overline{PWD}$ is at logic high level and equals 4.1V maximum the Boot Block is locked and cannot be programmed or erased. When $\overline{PWD}$ = 11.4V minimum the Boot Block is unlocked and can be programmed or erased. When $\overline{PWD}$ is at a logic low level the Boot Block is locked, the deep power-down mode is enabled and the WSM is reset preventing any blocks from being programmed or erased, therefore providing data protection during power transitions. $\overline{PWD}$ terminates any internally timed erase or program activities when it is taken to a logic low. $\overline{PWD}$ activates the $\overline{CE}$ input stage and requires 700 ns recovery time to transition from deep power-down to valid data on the outputs or 580 ns delay before the device can recognize valid inputs.
$\overline{OE}$	I	**OUTPUT ENABLE:** Gates the device's outputs through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	I	**WRITE ENABLE:** Controls writes to the Command Register and array blocks. $\overline{WE}$ is active low. Addresses and data are latched on the rising edge of the $\overline{WE}$ pulse.
V_{PP}		**PROGRAM/ERASE POWER SUPPLY:** For erasing memory array blocks or programming data in each block. **Note:** $V_{PP} < V_{PPLMAX}$ memory contents cannot be altered.
V_{CC}		**DEVICE POWER SUPPLY (3.3V ±0.3V, 5V ±10%)**
GND		**GROUND:** For all internal circuitry
NC		**NO CONNECT:** Pin may be driven or left floating
DU		**DON'T USE PIN:** Pin should not be connected to anything

2.0 28F200BX-L PRODUCTS DESCRIPTION

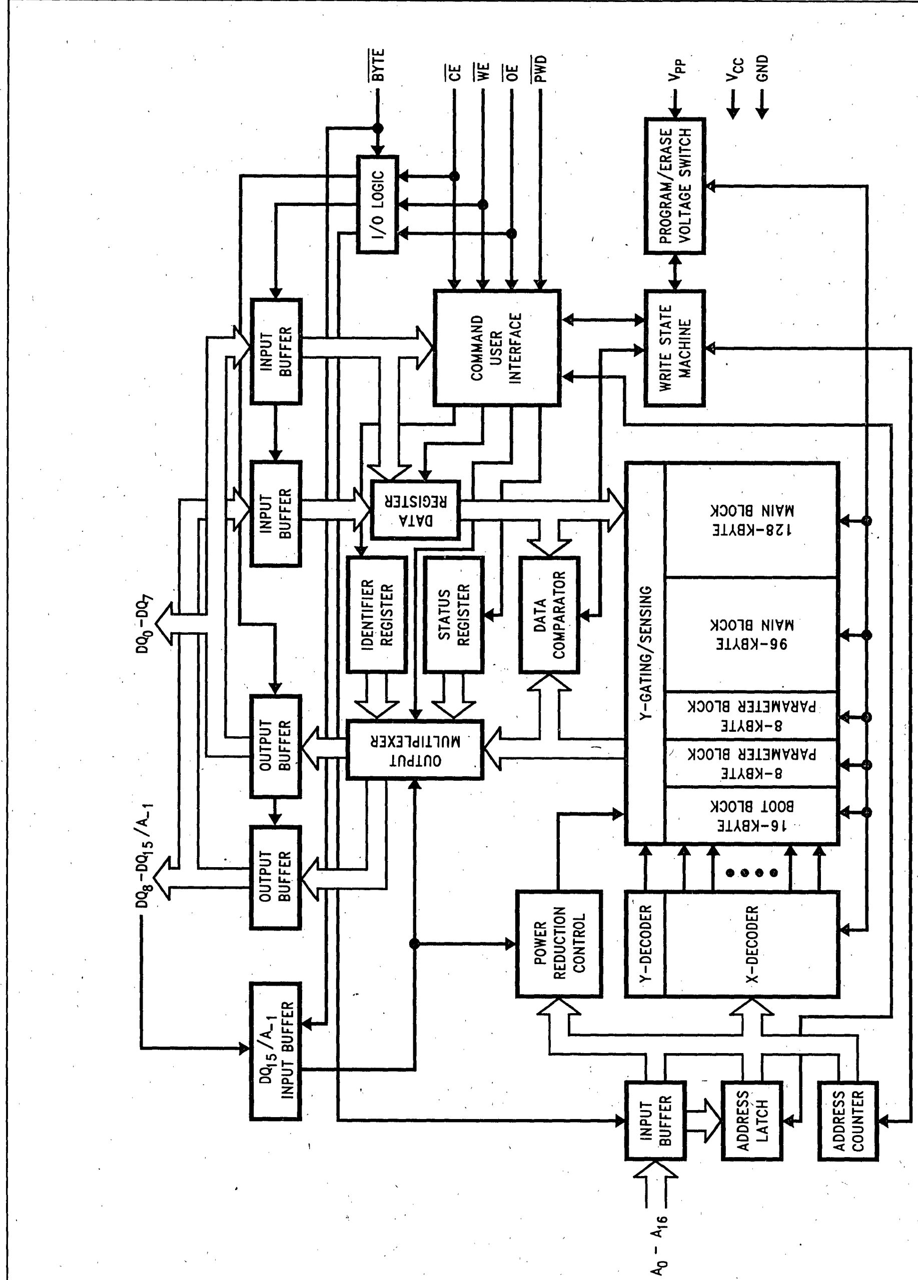


Figure 6. 28F200BX-L Word/Byte-Wide Block Diagram

2.1 28F200BX-L Memory Organization

2.1.1 BLOCKING

The 28F200BX-L uses a blocked array architecture to provide independent erasure of memory blocks. A block is erased independently of other blocks in the array when an address is given within the block address range and the Erase Setup and Erase Confirm commands are written to the CUI. The 28F200BX-L is a random read/write memory, only erasure is performed by block.

2.1.1.1 Boot Block Operation and Data Protection

The 16 Kbyte boot block provides a lock feature for secure code storage. The intent of the boot block is to provide a secure storage area for the kernel code that is required to boot a system in the event of power failure or other disruption during code update. This lock feature ensures absolute data integrity by preventing the boot block from being written or erased when $\overline{PWD}$ is not at 12V. The boot block can be erased and written when $\overline{PWD}$ is held at 12V for the duration of the erase or program operation. This allows customers to change the boot code when necessary while providing security when needed. See the Block Memory Map section for address locations of the boot block for the 28F200BX-TL and 28F200BX-BL.

2.1.1.2 Parameter Block Operation

The 28F200BX-L has 2 parameter blocks (8 Kbytes each). The parameter blocks are intended to provide storage for frequently updated system parameters and configuration or diagnostic information. The parameter blocks can also be used to store additional boot or main code. The parameter blocks however, do not have the hardware write protection feature that the boot block has. The parameter blocks provide for more efficient memory utilization when dealing with parameter changes versus reg-ularly blocked devices. See the Block Memory Map section for address locations of the parameter blocks for the 28F200BX-TL and 28F200BX-BL.

2.1.1.3 Main Block Operation

Two main blocks of memory exist on the 28F200BX-L (1 x 128 Kbyte block and 1 x 96 Kbyte blocks). See the following section on Block Memory Map for the address location of these blocks for the 28F200BX-TL and 28F200BX-BL products.

2.1.2 BLOCK MEMORY MAP

Two versions of the 28F200BX-L product exist to support two different memory maps of the array blocks in order to accommodate different microprocessor protocols for boot code location. The 28F200BX-TL memory map is inverted from the 28F200BX-BL memory map.

2.1.2.1 28F200BX-BL Memory Map

The 28F200BX-BL device has the 16 Kbyte boot block located from 00000H to 01FFFH to accommodate those microprocessors that boot from the bottom of the address map at 00000H. In the 28F200BX-BL the first 8 Kbyte parameter block resides in memory space from 02000H to 02FFFH. The second 8 Kbyte parameter block resides in memory space from 03000H to 03FFFH. The 96 Kbyte main block resides in memory space from 04000H to 0FFFFH. The 128 Kbyte main block resides in memory space from 10000H to 1FFFFH (word locations). See Figure 7.

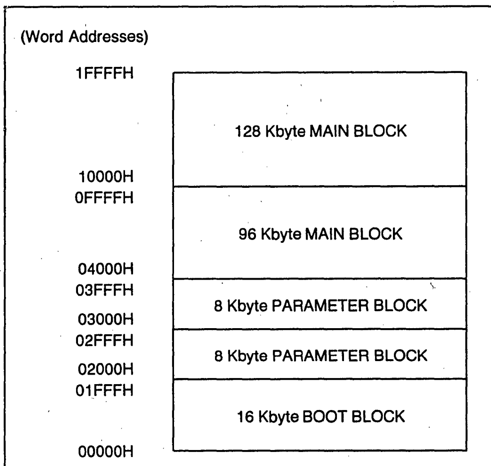


Figure 7. 28F200BX-BL Memory Map

2.1.2.2 28F200BX-TL Memory Map

The 28F200BX-TL device has the 16 Kbyte boot block located from 1E000H to 1FFFFH to accommodate those microprocessors that boot from the top of the address map. In the 28F200BX-TL the first 8 Kbyte parameter block resides in memory space from 1D000H to 1DFFFH. The second 8 Kbyte parameter block resides in memory space from 1C000H to 1CFFFH. The 96 Kbyte main block resides in memory space from 10000H to 1BFFFH. The 128 Kbyte main block resides in memory space from 00000H to 0FFFFH as shown below in Figure 8.

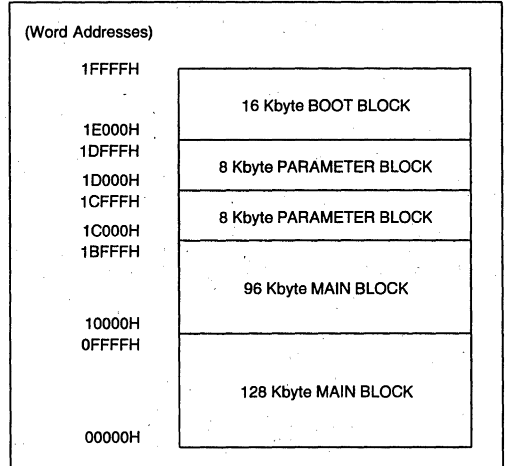


Figure 8. 28F200BX-TL Memory Map

3.0 28F002BX-L PRODUCTS DESCRIPTION

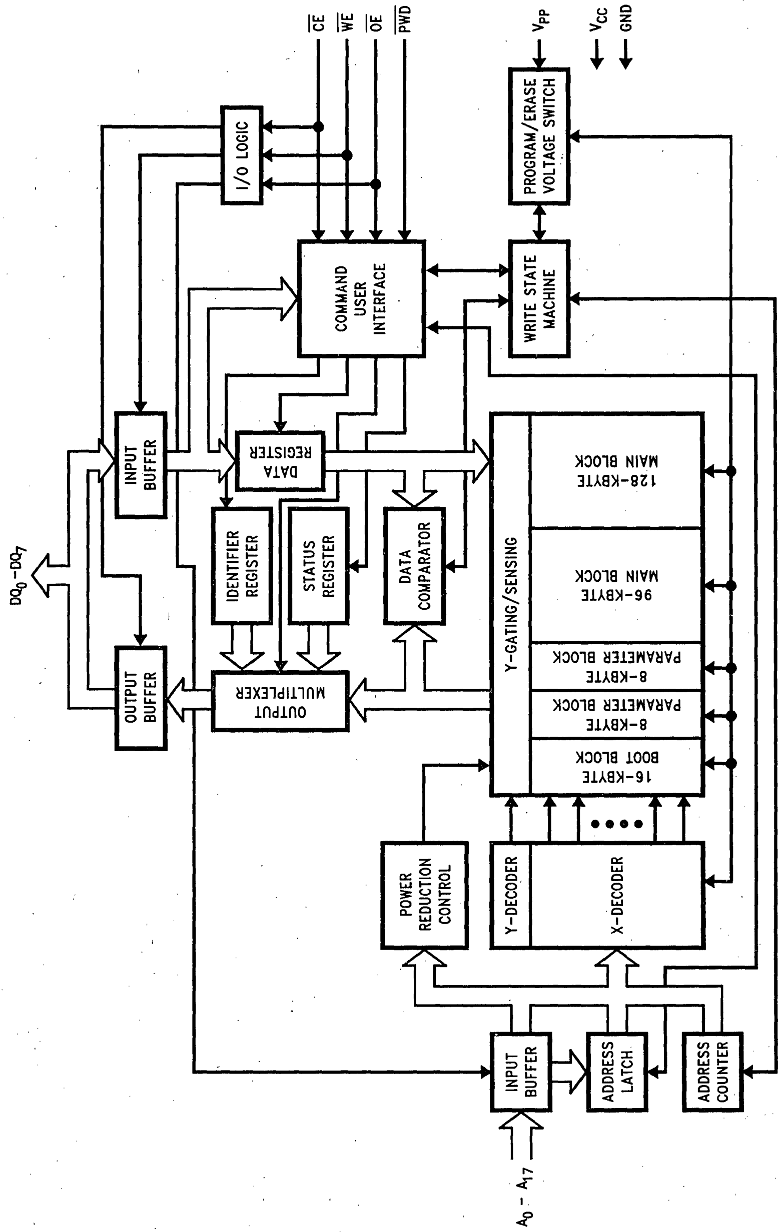


Figure 9. 28F002BX-L Byte-Wide Block Diagram

3.1 28F002BX-L Memory Organization

3.1.1 BLOCKING

The 28F002BX-L uses a blocked array architecture to provide independent erasure of memory blocks. A block is erased independently of other blocks in the array when an address is given within the block address range and the Erase Setup and Erase Confirm commands are written to the CUI. The 28F002BX-L is a random read/write memory, only erasure is performed by block.

3.1.1.1 Boot Block Operation and Data Protection

The 16 Kbyte boot block provides a lock feature for secure code storage. The intent of the boot block is to provide a secure storage area for the kernel code that is required to boot a system in the event of power failure or other disruption during code update. This lock feature ensures absolute data integrity by preventing the boot block from being programmed or erased when $\overline{PWD}$ is not at 12V. The boot block can be erased and programmed when $\overline{PWD}$ is held at 12V for the duration of the erase or program operation. This allows customers to change the boot code when necessary while still providing security when needed. See the Block Memory Map section for address locations of the boot block for the 28F002BX-TL and 28F002BX-BL.

3.1.1.2 Parameter Block Operation

The 28F002BX-L has 2 parameter blocks (8 Kbytes each). The parameter blocks are intended to provide storage for frequently updated system parameters and configuration or diagnostic information. The parameter blocks can also be used to store additional boot or main code. The parameter blocks however, do not have the hardware write protection feature that the boot block has. Parameter blocks provide for more efficient memory utilization when dealing with small parameter changes versus regularly blocked devices. See the Block Memory Map section for address locations of the parameter blocks for the 28F002BX-TL and 28F002BX-BL.

3.1.1.3 Main Block Operation

Two main blocks of memory exist on the 28F002BX-L (1 x 128 Kbyte block and 1 x 96 Kbyte block). See the following section on Block Memory Map for the address location of these blocks for the 28F002BX-TL and 28F002BX-BL.

3.1.2 BLOCK MEMORY MAP

Two versions of the 28F002BX-L product exist to support two different memory maps of the array blocks in order to accommodate different microprocessor protocols for boot code location. The 28F002BX-TL memory map is inverted from the 28F002BX-BL memory map.

3.1.2.1 28F002BX-BL Memory Map

The 28F002BX-BL device has the 16 Kbyte boot block located from 00000H to 03FFFH to accommodate those microprocessors that boot from the bottom of the address map at 00000H. In the 28F002BX-BL the first 8 Kbyte parameter block resides in memory from 04000H to 05FFFH. The second 8 Kbyte parameter block resides in memory space from 06000H to 07FFFH. The 96 Kbyte main block resides in memory space from 08000H to 1FFFFH. The 128 Kbyte main block resides in memory space from 20000H to 3FFFFH. See Figure 10.

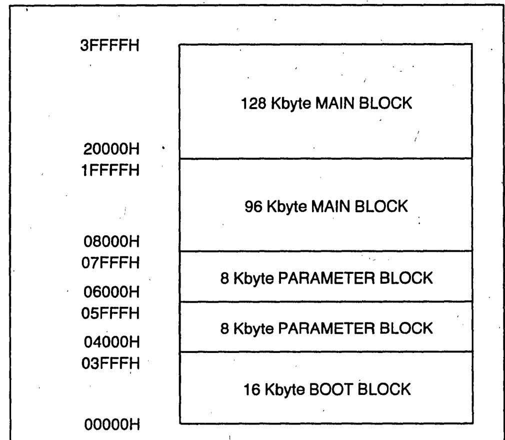


Figure 10. 28F002BX-BL Memory Map

3.1.2.2 28F002BX-TL Memory Map

The 28F002BX-TL device has the 16 Kbyte boot block located trom 3C000H to 3FFFFH to accommodate those microprocessors that boot from the top of the address map. In the 28F002BX-TL the first 8 Kbyte parameter block resides in memory space from 3A000H to 3BFFFH. The second 8 Kbyte parameter block resides in memory space from 38000H to 39FFFH. The 96 Kbyte main block resides in memory space from 20000H to 37FFFH. The 128 Kbyte main block resides in memory space from 00000H to 1FFFFH.

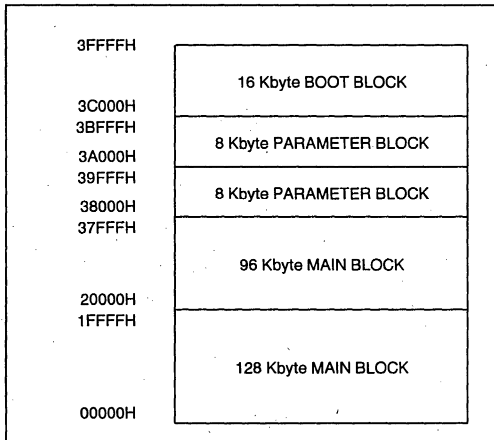


Figure 11. 28F002BX-TL Memory Map

4.0 PRODUCT FAMILY PRINCIPLES OF OPERATION

Flash memory augments EPROM functionality with in-circuit electrical write and erase. The 2 Mbit flash family utilizes a Command User Interface (CUI) and internally generated and timed algorithms to simplify write and erase operations.

The CUI allows for fixed power supplies during erasure and programming, and maximum EPROM compatibility.

In the absence of high voltage on the V_{PP} pin, the 2 Mbit flash family will only successfully execute the following commands: Read Array, Read Status Register, Clear Status Register and Intelligent Identifier mode. The device provides standard EPROM read, standby and output disable operations. Manufacturer Identification and Device Identification data can be accessed through the CUI or through the standard EPROM A9 high voltage access (V_{ID}) (for PROM programmer equipment).

The same EPROM read, standby and output disable functions are available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} allows write and erase of the device. All functions associated with altering memory contents: write and erase, Intelligent Identifier read and Read Status are accessed via the CUI.

The purpose of the Write State Machine (WSM) is to completely automate the write and erasure of the device. The WSM will begin operation upon receipt of a signal from the CUI and will report status back through a Status Register. The CUI will handle the $\overline{WE}$ interface to the data and address latches, as well as system software requests for status while the WSM is in operation.

4.1 28F200BX-L Bus Operations

Flash memory reads, erases and writes in-system via the local CPU. All bus cycles to or from the flash memory conform to standard microprocessor bus cycles.

Table 1. Bus Operations for WORD-WIDE Mode ($\overline{BYTE} = V_{IH}$)

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_9	A_0	V_{PP}	DQ_{0-15}
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	X	D_{OUT}
Output Disable		V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	High Z
Standby		V_{IH}	V_{IH}	X	X	X	X	X	High Z
Deep Power-Down	9	V_{IL}	X	X	X	X	X	X	High Z
Intelligent Identifier (Mfr)	4	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IL}	X	0089H
Intelligent Identifier (Device)	4, 5, 10	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IH}	X	2274H 2275H
Write	6, 7, 8	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	D_{IN}

Table 2. Bus Operations for BYTE-WIDE Mode ($\overline{BYTE} = V_{IL}$)

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_9	A_0	A_{-1}	V_{PP}	DQ_{0-7}	DQ_{8-14}
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	X	X	D_{OUT}	High Z
Output Disable		V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	X	High Z	High Z
Standby		V_{IH}	V_{IH}	X	X	X	X	X	X	High Z	High Z
Deep Power-Down	9	V_{IL}	X	X	X	X	X	X	X	High Z	High Z
Intelligent Identifier (Mfr)	4	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IL}	X	X	89H	High Z
Intelligent Identifier (Device)	4, 5	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IH}	X	X	74H 75H	High Z
Write	6, 7, 8	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	X	D_{IN}	High Z

NOTES:
1. Refer to DC Characteristics.
2. X can be V_{IL} or V_{IH} for control pins and addresses, V_{PPL} or V_{PPH} for V_{PP}.
3. See DC characteristics for V_{PPL}, V_{PPH}, V_{HH}, V_{ID} voltages.
4. Manufacturer and Device codes may also be accessed via a CPU write sequence. A_1–A_{16} = V_{IL}.
5. Device ID = 2274H for 28F200BX-TL and 2275H for 28F200BX-BL.
6. Refer to Table 4 for valid D_{IN} during a write operation.
7. Command writes for Block Erase or Word/Byte Write are only executed when $V_{PP} = V_{PPH}$.
8. To write or erase the boot block, hold $\overline{PWD}$ at V_{HH}.
9. $\overline{PWD}$ must be at GND $\pm$0.2V to meet the 1.2 μA maximum deep power-down current.
10. The device ID codes are identical to those of the 28F2100BX 5V versions.

4.2 28F002BX-L Bus Operations

Table 3. Bus Operations

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_9	A_0	V_{PP}	DQ_{0-7}
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	X	D_{OUT}
Output Disable		V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	High Z
Standby		V_{IH}	V_{IH}	X	X	X	X	X	High Z
Deep Power-Down	9	V_{IL}	X	X	X	X	X	X	High Z
Intelligent Identifier (Mfr)	4	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IL}	X	89H
Intelligent Identifier (Device)	4, 5	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IH}	X	7CH 7DH
Write	6, 7, 8	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	D_{IN}

NOTES:
1. Refer to DC Characteristics.
2. X can be V_{IL} or V_{IH} for control pins and addresses, V_{PPL} or V_{PPH} for V_{PP}.
3. See DC characteristics for V_{PPL}, V_{PPH}, V_{HH}, V_{ID} voltages.
4. Manufacturer and Device codes may also be accessed via a CUI write sequence. $A_1-A_{17} = V_{IL}$.
5. Device ID = 7CH for 28F002BX-TL and 7DH for 28F002BX-BL.
6. Refer to Table 4 for valid D_{IN} during a write operation.
7. Command writes for Block erase or byte program are only executed when $V_{PP} = V_{PPH}$.
8. Program or erase the Boot block by holding $\overline{PWD}$ at V_{HH}.
9. $\overline{PWD}$ must be at GND ±0.2V to meet the 1.2 μA maximum deep power-down current.
10. The device ID codes are identical to those of the 28F002BX 5V versions.

4.3 Read Operations

The 2 Mbit flash family has three user read modes; Array, Intelligent Identifier, and Status Register. Status Register read mode will be discussed in detail in the "Write Operations" section.

During power-up conditions (V_{CC} supply ramping), it takes a maximum of 700 ns from V_{CC} at 3.0V minimum to obtain valid data on the outputs.

4.3.1 READ ARRAY

If the memory is not in the Read Array mode, it is necessary to write the appropriate read mode command to the CUI. The 2 Mbit flash family has three control functions, all of which must be logically active, to obtain data at the outputs. Chip-Enable $\overline{CE}$ is the device selection control. Power-Down $\overline{PWD}$ is the device power control. Output-Enable $\overline{OE}$ is the DATA INPUT/OUTPUT (DQ[0:15] or DQ[0:7]) direction control and when active is used to drive data from the selected memory on to the I/O bus.

4.3.1.1 Output Control

With $\overline{OE}$ at logic-high level (V_{IH}), the output from the device is disabled and data input/output pins (DQ[0:15] or DQ[0:7]) are tri-stated. Data input is then controlled by $\overline{WE}$.

4.3.1.2 Input Control

With $\overline{WE}$ at logic-high level (V_{IH}), input to the device is disabled. Data Input/Output pins (DQ[0:15] or DQ[0:7]) are controlled by $\overline{OE}$.

4.3.2 INTELLIGENT IDENTIFIERS

28F200BX-L Products

The manufacturer and device codes are read via the CUI or by taking the A_9 pin to 12V. Writing 90*H* to the CUI places the device into Intelligent Identifier read mode. A read of location 00000*H* outputs the manufacturer's identification code, 0089H, and location 00001*H* outputs the device code; 2274H for 28F200BX-TL, 2275H for 28F200BX-BL. When $\overline{BYTE}$ is at a logic low only the lower byte of the above signatures is read and DQ_{15}/A_{-1} is a "don't care" during Intelligent Identifier mode. A read array command must be written to the CUI to return to the read array mode.

28F002BX-L Products

The manufacturer and device codes are also read via the CUI or by taking the A_9 pin to 12V. Writing 90*H* to the CUI places the device into Intelligent Identifier read mode. A read of location 00000*H* outputs the manufacturer's identification code, 89H, and location 00001*H* outputs the device code; 7CH for 28F002BX-TL, 7DH for 28F002BX-BL.

4.4 Write Operations

Commands are written to the CUI using standard microprocessor write timings. The CUI serves as the interface between the microprocessor and the internal chip operation. The CUI can decipher Read Array, Read Intelligent Identifier, Read Status Register, Clear Status Register, Erase and Program commands. In the event of a read command, the CUI simply points the read path at either the array, the Intelligent Identifier, or the status register depending on the specific read command given. For a program or erase cycle, the CUI informs the write state machine that a write or erase has been requested. During a program cycle, the Write State Machine will control the program sequences and the CUI will only respond to status reads. During an erase cycle, the CUI will respond to status reads and erase suspend. After the Write State Machine has completed its task, it will allow the CUI to respond to its full command set. The CUI will stay in the current command state until the microprocessor issues another command.

The CUI will successfully initiate an erase or write operation only when V_{PP} is within its voltage range. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable, available only when memory updates are desired. The system designer can also choose to "hard-wire" V_{PP} to 12V. The 2 Mbit flash family is designed to accommodate either design practice. It is recommended that $\overline{PWD}$ be tied to logical Reset for data protection during unstable CPU reset function as described in the "Product Family Overview" section.

4.4.1 BOOT BLOCK WRITE OPERATIONS

In the case of Boot Block modifications (write and erase), $\overline{PWD}$ is set to V_{HH} = 12V typically, in addition to V_{PP} at high voltage. However, if $\overline{PWD}$ is not at V_{HH} when a program or erase operation of the boot block is attempted, the corresponding status register bit (Bit 4 for Program and Bit 5 for Erase, refer to Table 5 for Status Register Definitions) is set to indicate the failure to complete the operation.

4.4.2 COMMAND USER INTERFACE (CUI)

The Command User Interface (CUI) serves as the interface to the microprocessor. The CUI points the read/write path to the appropriate circuit block as described in the previous section. After the WSM has completed its task, it will set the WSM Status bit to a "1", which will also allow the CUI to respond to its full command set. Note that after the WSM has returned control to the CUI, the CUI will remain in its current state.

4.4.2.1 Command Set

Command Codes	Device Mode
00	Invalid/Reserved
10	Alternate Program Setup
20	Erase Setup
40	Program Setup
50	Clear Status Register
70	Read Status Register
90	Intelligent Identifier
B0	Erase Suspend
D0	Erase Resume/Erase Confirm
FF	Read Array

4.4.2.2 Command Function Descriptions

Device operations are selected by writing specific commands into the CUI. Table 4 defines the 2 Mbit flash family commands.

Table 4. Command Definitions

Command	Bus Cycles Req'd	Notes	First Bus Cycle			Second Bus Cycle		
		8	Operation	Address	Data	Operation	Address	Data
Read Array	1	1	Write	X	FFH			
Intelligent Identifier	3	2, 4	Write	X	90H	Read	IA	IID
Read Status Register	2	3	Write	X	70H	Read	X	SRD
Clear Status Register	1		Write	X	50H			
Erase Setup/Erase Confirm	2	5	Write	BA	20H	Write	BA	D0H
Word/Byte Write Setup/Write	2	6, 7	Write	WA	40H	Write	WA	WD
Erase Suspend/Erase Resume	2		Write	X	B0H	Write	X	D0H
Alternate Word/Byte Write Setup/Write	2	2, 3, 7	Write	WA	10H	Write	WA	WD

NOTES:
1. Bus operations are defined in Tables 1, 2, 3.
2. IA = Identifier Address: 00H for manufacturer code, 01H for device code.
3. SRD = Data read from Status Register.
4. IID = Intelligent Identifier Data.
Following the Intelligent Identifier Command, two read operations access manufacturer and device codes.
5. BA = Address within the block being erased.
6. WA = Address to be written.
WD = Data to be written at location WA.
7. Either 40H or 10H commands is valid.
8. When writing commands to the device, the upper data bus [DQ_8–DQ_{15}] = X (28F200BX-L-only) which is either V_{CC} or V_{SS} to avoid burning additional current.

Invalid/Reserved

These are unassigned commands. It is not recommended that the customer use any command other than the valid commands specified above. Intel reserves the right to redefine these codes for future functions.

Read Array (FFH)

This single write command points the read path at the array. If the host CPU performs a $\overline{CE}/\overline{OE}$ controlled read immediately following a two-write sequence that started the WSM, then the device will output status register contents. If the Read Array command is given after Erase Setup the device is reset to read the array. A two Read Array command sequence (FFH) is required to reset to Read Array after Program Setup.

Intelligent Identifier (90H)

After this command is executed, the CUI points the output path to the Intelligent Identifier circuits. Only Intelligent Identifier values at addresses 0 and 1 can be read (only address A0 is used in this mode, all other address inputs are ignored).

Read Status Register (70H)

This is one of the two commands that is executable while the state machine is operating. After this command is written, a read of the device will output the contents of the status register, regardless of the address presented to the device.

The device automatically enters this mode after program or erase has completed.

Clear Status Register (50H)

The WSM can only set the Program Status and Erase Status bits in the status register, it can not clear them. Two reasons exist for operating the status register in this fashion. The first is a synchronization. The WSM does not know when the host CPU has read the status register, therefore it would not know when to clear the status bits. Secondly, if the CPU is programming a string of bytes, it may be more efficient to query the status register after programming the string. Thus, if any errors exist while programming the string, the status register will return the accumulated error status.

Program Setup (40H or 10H)

This command simply sets the CUI into a state such that the next write will load the address and data registers. Either 40H or 10H can be used for Program Setup. Both commands are included to accommodate efforts to achieve an industry standard command code set.

Program

The second write after the program setup command, will latch addresses and data. Also, the CUI initiates the WSM to begin execution of the program algorithm. While the WSM finishes the algorithm, the device will output Status Register contents. Note that the WSM cannot be suspended during programming.

Erase Setup (20H)

Prepares the CUI for the Erase Confirm command. No other action is taken. If the next command is not an Erase Confirm command then the CUI will set both the Program Status and Erase Status bits of the Status Register to a "1", place the device into the Read Array state, and wait for another command.

Erase Confirm (D0H)

If the previous command was an Erase Setup command, then the CUI will enable the WSM to erase, at the same time closing the address and data latches, and respond only to the Read Status Register and Erase Suspend commands. While the WSM is executing, the device will output Status Register data when $\overline{OE}$ is toggled low. Status Register data can only be updated by toggling either $\overline{OE}$ or $\overline{CE}$ low.

Erase Suspend (B0H)

This command only has meaning while the WSM is executing an Erase operation, and therefore will only be responded to during an erase operation. After this command has been executed, the CUI will initiate the WSM to suspend Erase operations, and then return to responding to only Read Status Register or to the Erase Resume commands. Once the WSM has reached the Suspend state, it will set an output into the CUI which allows the CUI to respond to the Read Array, Read Status Register, and Erase Resume commands. In this mode, the CUI will not respond to any other commands. The WSM will also set the WSM Status bit to a "1". The WSM will continue to run, idling in the SUSPEND state, regardless of the state of all input control pins, with the exclusion of $\overline{PWD}$. $\overline{PWD}$ low will immediately shut down the WSM and the remainder of the chip.

Erase Resume (D0H)

This command will cause the CUI to clear the Suspend state and set the WSM Status bit to a "0", but only if an Erase Suspend command was previously issued. Erase Resume will not have any effect in all other conditions.

4.4.3 STATUS REGISTER

The 2 Mbit flash family contains a status register which may be read to determine when a program or erase operation is complete, and whether that operation completed successfully. The status register may be read at any time by writing the Read Status command to the CUI. After writing this command, all subsequent Read operations output data from the status register until another command is written to the CUI. A Read Array command must be written to the CUI to return to the Read Array mode.

The status register bits are output on DQ[0:7] whether the device is in the byte-wide (x8) or word-wide (x16) mode for the 28F200BX-L. In the word-wide mode the upper byte, DQ[8:15] is set to 00H during a Read Status command. In the byte-wide mode, DQ[8:14] are tri-stated and DQ_{15}/A_{-1} retains the low order address function.

It should be noted that the contents of the status register are latched on the falling edge of $\overline{OE}$ or $\overline{CE}$ whichever occurs last in the read cycle. This prevents possible bus errors which might occur if the contents of the status register change while reading the status register. $\overline{CE}$ or $\overline{OE}$ must be toggled with each subsequent status read, or the completion of a program or erase operation will not be evident.

The Status Register is the interface between the microprocessor and the Write State Machine (WSM). When the WSM is active, this register will indicate the status of the WSM, and will also hold the bits indicating whether or not the WSM was successful in performing the desired operation. The WSM sets status bits "Three" through "Seven" and clears bits "Six" and "Seven", but cannot clear status bits "Three" through "Five". These bits can only be cleared by the controlling CPU through the use of the Clear Status Register command.

4.4.3.1 Status Register Bit Definition

Table 5. Status Register Definitions

WSMS	ESS	ES	PS	VPPS	R	R	R
7	6	5	4	3	2	1	0

	NOTES:
SR.7 = WRITE STATE MACHINE STATUS 1 = Ready 0 = Busy	Write State Machine Status bit must first be checked to determine byte/word program or block erase completion, before the Program or Erase Status bits are checked for success.
SR.6 = ERASE SUSPEND STATUS 1 = Erase Suspended 0 = Erase in Progress/Completed	When Erase Suspend is issued, WSM halts execution and sets both WSMS and ESS bits to "1". ESS bit remains set to "1" until an Erase Resume command is issued.
SR.5 = ERASE STATUS 1 = Error in Block Erasure 0 = Successful Block Erase	When this bit is set to "1". WSM has applied the maximum number of erase pulses to the block and is still unable to successfully perform an erase verify.
SR.4 = PROGRAM STATUS 1 = Error in Byte/Word Program 0 = Successful Byte/Word Program	When this bit is set to "1", WSM has attempted but failed to Program a byte or word.
SR.3 = V_{PP} STATUS 1 = V_{PP} Low Detect; Operation Abort 0 = V_{PP} OK	The V_{PP} Status bit unlike an A/D converter, does not provide continuous indication of V_{PP} level. The WSM interrogates the V_{PP} level only after the byte write or block erase command sequences have been entered and informs the system if V_{PP} has not been switched on. The V_{PP} Status bit is not guaranteed to report accurate feedback between V_{PPL} and V_{PPH}.

SR.2–SR.0 = RESERVED FOR FUTURE ENHANCEMENTS
These bits are reserved for future use and should be masked out when polling the Status Register.

4.4.3.2 Clearing the Status Register

Certain bits in the status register are set by the write state machine, and can only be reset by the system software. These bits can indicate various failure conditions. By allowing the system software to control the resetting of these bits, several operations may be performed (such as cumulatively programming several bytes or erasing multiple blocks in sequence). The status register may then be read to determine if an error occurred during that programming or erasure series. This adds flexibility to the way the device may be programmed or erased. To clear the status register, the Clear Status Register command is written to the CUI. Then, any other command may be issued to the CUI. Note again that before a read cycle can be initiated, a Read Array command must be written to the CUI to specify whether the read data is to come from the array, status register, or Intelligent Identifier.

4.4.4 PROGRAM MODE

Program is executed by a two-write sequence. The Program Setup command is written to the CUI followed by a second write which specifies the address and data to be programmed. The write state machine will execute a sequence of internally timed events to:

1. program the desired bits of the addressed memory word (byte), and
2. verify that the desired bits are sufficiently programmed.

Programming of the memory results in specific bits within a byte or word being changed to a "0".

If the user attempts to program "1"s, there will be no change of the memory cell content and no error occurs.

Similar to erasure, the status register indicates whether programming is complete. While the program sequence is executing, bit 7 of the status register is a "0". The status register can be polled by toggling either $\overline{CE}$ or $\overline{OE}$ to determine when the program sequence is complete. Only the Read Status Register command is valid while programming is active.

When programming is complete, the status bits, which indicate whether the program operation was successful, should be checked. If the programming operation was unsuccessful, Bit 4 of the status register is set to a "1" to indicate a Program Failure. If Bit 3 is set then V_{PP} was not within acceptable limits, and the WSM will not execute the programming sequence.

The status register should be cleared before attempting the next operation. Any CUI instruction can follow after programming is completed; however, it must be recognized that reads from the memory, status register, or Intelligent Identifier cannot be accomplished until the CUI is given the appropriate command. A Read Array command must first be given before memory contents can be read.

Figure 12 shows a system software flowchart for device byte programming operation. Figure 13 shows a similar flowchart for device word programming operation (28F200BX-L-only).

4.4.5 ERASE MODE

Erasure of a single block is initiated by writing the Erase Setup and Erase Confirm commands to the CUI, along with the addresses, A[12:16] for the 28F200BX-L or A[12:17] for the 28F002BX-L, identifying the block to be erased. These addresses are latched internally when the Erase Confirm command is issued. Block erasure results in all bits within the block being set to "1".

The WSM will execute a sequence of internally timed events to:

1. program all bits within the block
2. verify that all bits within the block are sufficiently programmed
3. erase all bits within the block and
4. verify that all bits within the block are sufficiently erased

While the erase sequence is executing, Bit 7 of the status register is a "0".

When the status register indicates that erasure is complete, the status bits, which indicate whether the erase operation was successful, should be checked. If the erasure operation was unsuccessful, Bit 5 of the status register is set to a "1" to indicate an Erase Failure. If V_{PP} was not within acceptable limits after the Erase Confirm command is issued, the WSM will not execute an erase sequence; instead, Bit 5 of the status register is set to a "1" to indicate an Erase Failure, and Bit 3 is set to a "1" to identify that V_{PP} supply voltage was not within acceptable limits.

The status register should be cleared before attempting the next operation. Any CUI instruction can follow after erasure is completed; however, it must be recognized that reads from the memory array, status register, or Intelligent Identifier can not be accomplished until the CUI is given the appropriate command. A Read Array command must first be given before memory contents can be read.

Figure 14 shows a system software flowchart for Block Erase operation.

4.4.5.1 Suspending and Resuming Erase

Since an erase operation typically requires 2 to 5 seconds to complete, an Erase Suspend command is provided. This allows erase-sequence interruption in order to read data from another block of the memory. Once the erase sequence is started, writing the Erase Suspend command to the CUI requests that the Write State Machine (WSM) pause the erase sequence at a predetermined point in the erase algorithm. The status register must be read to determine when the erase operation has been suspended.

At this point, a Read Array command can be written to the CUI in order to read data from blocks other than that which is being suspended. The only other valid command at this time is the Erase Resume command or Read Status Register operation.

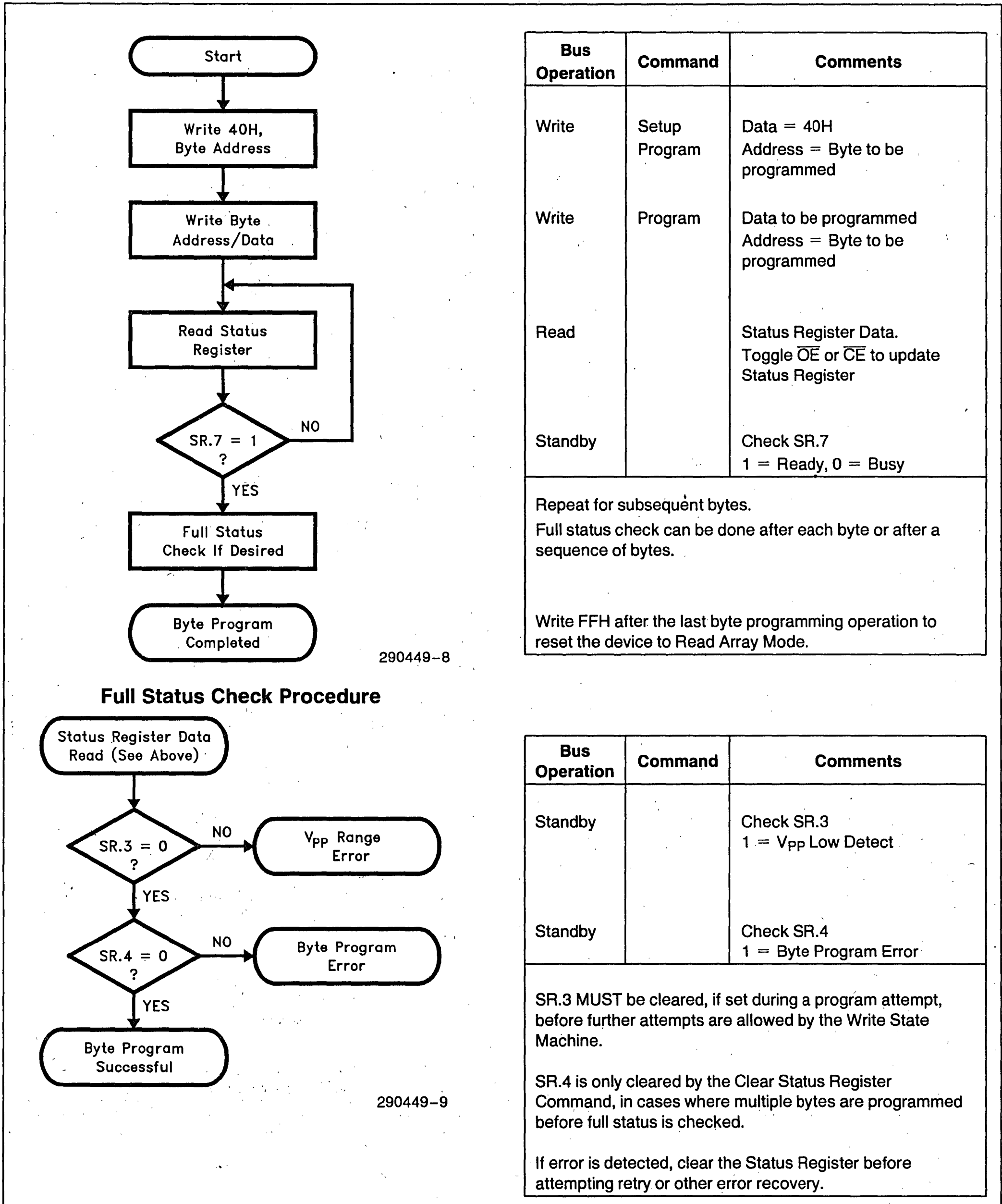
Figure 15 shows a system software flowchart detailing the operation.

During Erase Suspend mode, the chip can go into a pseudo-standby mode by taking $\overline{CE}$ to V_{IH} and the active current is now a maximum of 6 mA. If the chip is enabled while in this mode by taking $\overline{CE}$ to V_{IL}, the Erase Resume command can be issued to resume the erase operation.

Upon completion of reads from any block other than the block being erased, the Erase Resume command must be issued. When the Erase Resume command is given, the WSM will continue with the erase sequence and complete erasing the block. As with the end of erase, the status register must be read, cleared, and the next instruction issued in order to continue.

4.4.6 EXTENDED CYCLING

Intel has designed extended cycling capability into its ETOX III flash memory technology. The 2 Mbit flash family is designed for 10,000 program/erase cycles on each of the five blocks. The combination of low electric fields, clean oxide processing and minimized oxide area per memory cell subjected to the tunneling electric field, results in very high cycling capability.



Bus Operation	Command	Comments
Write	Setup Program	Data = 40H Address = Byte to be programmed
Write	Program	Data to be programmed Address = Byte to be programmed
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent bytes.

Full status check can be done after each byte or after a sequence of bytes.

Write FFH after the last byte programming operation to reset the device to Read Array Mode.

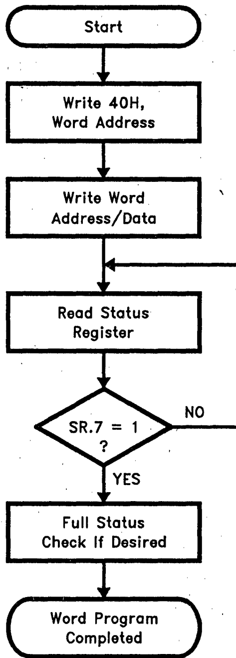
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Byte Program Error

SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple bytes are programmed before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 12. Automated Byte Programming Flowchart



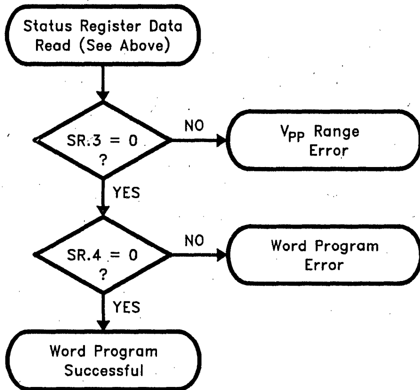
Bus Operation	Command	Comments
Write	Setup Program	Data = 40H Address = Word to be programmed
Write	Program	Data to be programmed Address = Word to be programmed
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent words.

Full status check can be done after each word or after a sequence of words.

Write FFH after the last word programming operation to reset the device to Read Array Mode.

Full Status Check Procedure



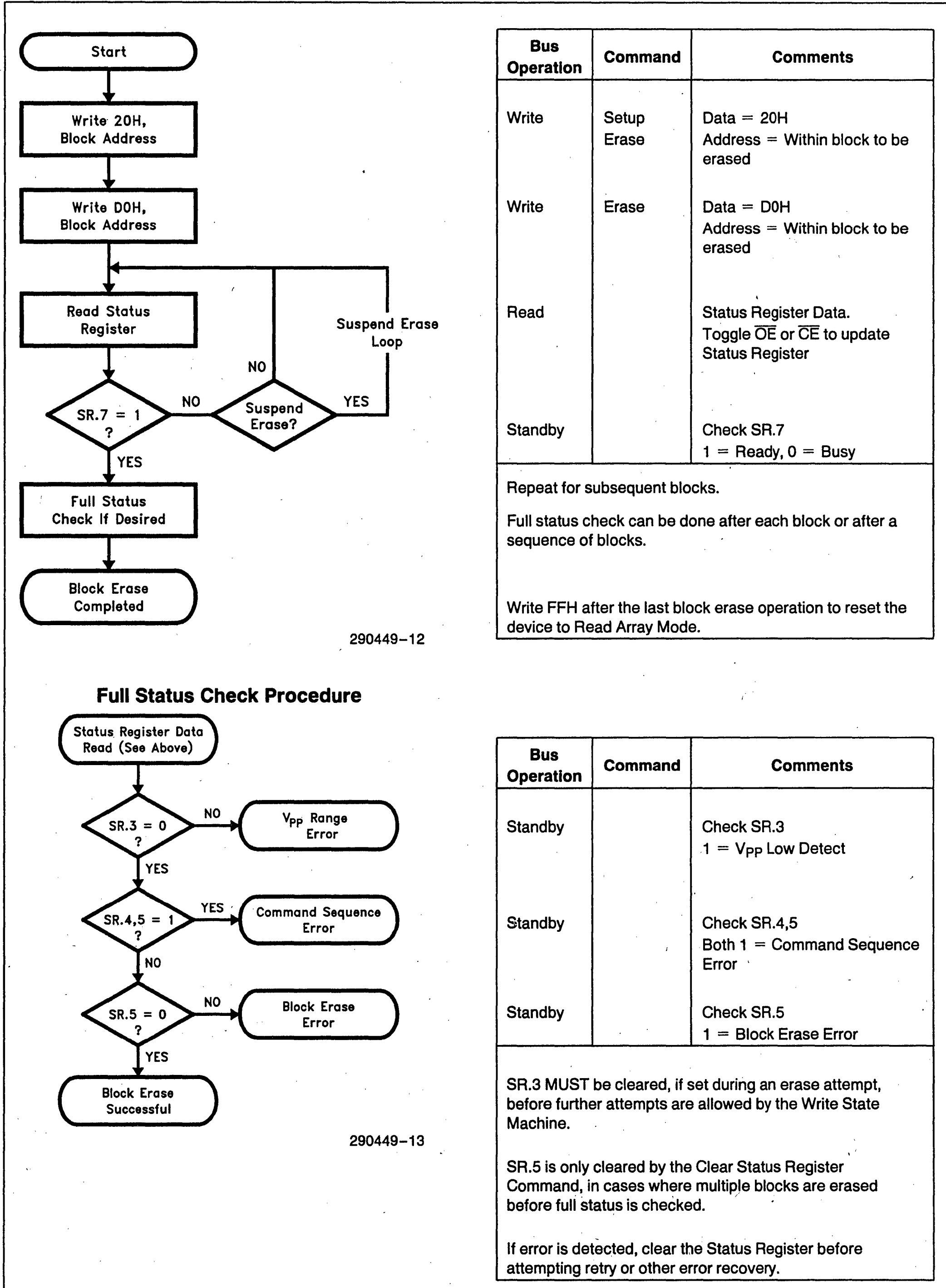
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Word Program Error

SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple words are programmed before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 13. Automated Word Programming Flowchart



Bus Operation	Command	Comments
Write	Setup Erase	Data = 20H Address = Within block to be erased
Write	Erase	Data = D0H Address = Within block to be erased
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent blocks.

Full status check can be done after each block or after a sequence of blocks.

Write FFH after the last block erase operation to reset the device to Read Array Mode.

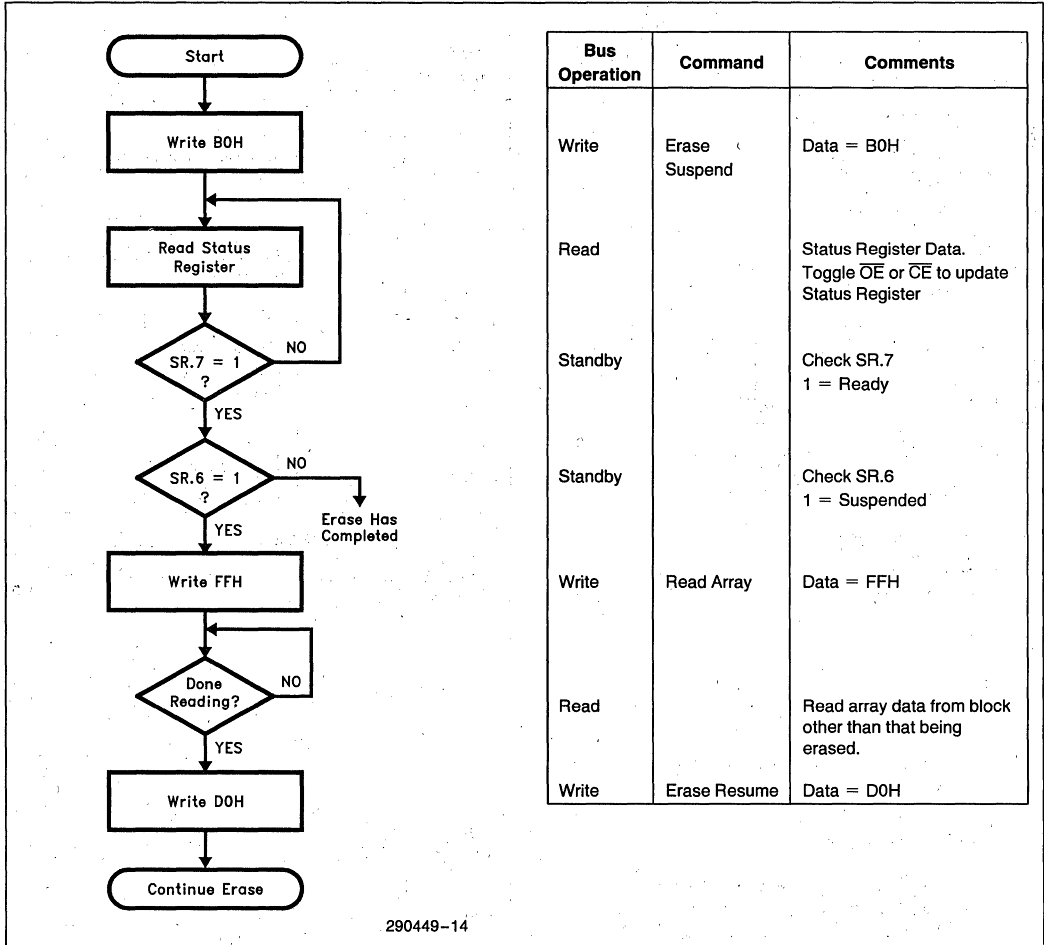
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4,5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error

SR.3 MUST be cleared, if set during an erase attempt, before further attempts are allowed by the Write State Machine.

SR.5 is only cleared by the Clear Status Register Command, in cases where multiple blocks are erased before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 14. Automated Block Erase Flowchart



Bus Operation	Command	Comments
Write	Erase Suspend	Data = B0H
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready
Standby		Check SR.6 1 = Suspended
Write	Read Array	Data = FFH
Read		Read array data from block other than that being erased.
Write	Erase Resume	Data = D0H

Figure 15. Erase Suspend/Resume Flowchart

4.5 Power Consumption

4.5.1 ACTIVE POWER

With $\overline{CE}$ at a logic-low level and $\overline{PWD}$ at a logic-high level, the device is placed in the active mode. The device I_{CC} current is a maximum of 22 mA at 5 MHz.

4.5.2 AUTOMATIC POWER SAVINGS

Automatic Power Savings (APS) is a low power feature during active mode of operation. The 2 Mbit flash family of products incorporate Power Reduction Control (PRC) circuitry which basically allows the device to put itself into a low current state when it is not being accessed. After data is read from the memory array, PRC logic controls the device's power consumption by entering the APS mode where typical I_{CC} current is 0.8 mA and maximum I_{CC} current is 2 mA. The device stays in this static state with outputs valid until a new memory location is read.

4.5.3 STANDBY POWER

With $\overline{CE}$ at a logic-high level (V_{IH}), and the CUI in read mode, the memory is placed in standby mode where the maximum I_{CC} standby current is 120 μA with CMOS input signals. The standby operation disables much of the device's circuitry and substantially reduces device power consumption. The outputs (DQ[0:15] or DQ[0:7]) are placed in a high-impedance state independent of the status of the $\overline{OE}$ sig-

nal. When the 2 Mbit flash family is deselected during erase or program functions, the devices will continue to perform the erase or program function and consume program or erase active power until program or erase is completed.

4.5.4 DEEP POWER-DOWN

The 2 Mbit flash family supports a typical I_{CC} of 0.2 µA in deep power-down mode. One of the target markets for these devices is in portable equipment where the power consumption of the machine is of prime importance. The 2 Mbit flash family has a $\overline{PWD}$ pin which places the device in the deep power-down mode. When $\overline{PWD}$ is at a logic-low (GND ±0.2V), all circuits are turned off and the device typically draws 0.2 µA of V_{CC} current.

During read modes, the $\overline{PWD}$ pin going low deselects the memory and places the output drivers in a high impedance state. Recovery from the deep power-down state, requires a minimum of 700 ns to access valid data (t_{PHQV}).

During erase or program modes, $\overline{PWD}$ low will abort either erase or program operation. The contents of the memory are no longer valid as the data has been corrupted by the $\overline{PWD}$ function. As in the read mode above, all internal circuitry is turned off to achieve the 0.2 µA current level.

$\overline{PWD}$ transitions to V_{IL} or turning power off to the device will clear the status register.

4.6 Power-Up Operation

The 2 Mbit flash family is designed to offer protection against accidental block erasure or programming during power transitions. Upon power-up the 2 Mbit flash family is indifferent as to which power supply, V_{PP} or V_{CC}, powers-up first. Power supply sequencing is not required.

The 2 Mbit flash family ensures the CUI is reset to the read mode on power-up.

In addition, on power-up the user must either drop $\overline{CE}$ low or present a new address to ensure valid data at the outputs.

A system designer must guard against spurious writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either signal to V_{IH} will inhibit writes to the device. The CUI architecture provides an added level of protection since alteration of memory contents can only occur after successful completion of the two-step command sequences. Finally the device is disabled until $\overline{PWD}$ is brought to V_{IH}, regardless of the state of its control inputs. This feature provides yet another level of memory protection.

4.7 Power Supply Decoupling

Flash memory's power switching characteristics require careful device decoupling methods. System designers are interested in 3 supply current issues:

- Standby current levels (I_{CCS})
- Active current levels (I_{CCR})
- Transient peaks produced by falling and rising edges of $\overline{CE}$.

Transient current magnitudes depend on the device outputs' capacitive and inductive loading. Two-line control and proper decoupling capacitor selection will suppress these transient voltage peaks. Each flash device should have a 0.1 µF ceramic capacitor connected between each V_{CC} and GND, and between its V_{PP} and GND. These high frequency, low-inherent inductance capacitors should be placed as close as possible to the package leads.

4.7.1 V_{PP} TRACE ON PRINTED CIRCUIT BOARDS

Writing to flash memories while they reside in the target system, requires special consideration of the V_{PP} power supply trace by the printed circuit board designer. The V_{PP} pin supplies the flash memory cell's current for programming and erasing. One should use similar trace widths and layout considerations given to the V_{CC} power supply trace. Adequate V_{PP} supply traces and decoupling will decrease spikes and overshoots.

4.7.2 V_{CC}, V_{PP} AND $\overline{PWD}$ TRANSITIONS

The CUI latches commands as issued by system software and is not altered by V_{PP} or $\overline{CE}$ transitions or WSM actions. Its state upon power-up, after exit from deep power-down mode or after V_{CC} transitions below V_{LKO} (Lockout voltage), is Read Array mode.

After any word/byte write or block erase operation is complete and even after V_{PP} transitions down to V_{PPL}, the CUI must be reset to Read Array mode via the Read Array command when accesses to the flash memory are desired.

5.0 OPERATING SPECIFICATIONS

Absolute Maximum Ratings

Operating Temperature
During Read .... 0°C to 70°C(1)
During Block Erase and
Word/Byte Write.................... 0°C to 70°C
Temperature Under Bias .... −10°C to +80°C
Storage Temperature .... −65°C to +125°C
Voltage on Any Pin
(except V_{CC} and V_{PP})
with Respect to GND .... −2.0V to +7.0V(2)
Voltage on Pin $\overline{PWD}$ or Pin A_9
with Respect to GND .... −2.0V to 13.5V(2, 3)
V_{PP} Program Voltage with Respect
to GND during Block Erase
and Word/Byte Write ..... −2.0V to +14.0V(2, 3)
V_{CC} Supply Voltage
with Respect to GND .... −2.0V to +7.0V(2)
Output Short Circuit Current............. 100 mA(4)

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC voltage is −0.5V on input/output pins. During transitions, this level may undershoot to −2.0V for periods <20 ns.
Maximum DC voltage on input/output pins is V_{CC} + 0.5V which, during transitions, may overshoot to V_{CC} + 2.0V for periods <20 ns.
3. Maximum DC voltage on A_9 or V_{PP} may overshoot to +14.0V for periods <20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.
5. AC Specifications are valid at both voltage ranges. See DC Characteristics table for voltage range-specific specifications.

OPERATING CONDITIONS

Symbol	Parameter	Notes	Min	Max	Unit
T_A	Operating Temperature		0	70	°C
V_{CC}	V_{CC} Supply Voltage	5	3.00	3.60	V
V_{CC}	V_{CC} Supply Voltage	5	4.50	5.50	V

D.C. CHARACTERISTICS V_{CC} = 3.3V ±0.3V

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{LI}	Input Load Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND
I_{CCS}	V_{CC} Standby Current	1, 3		45	120	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{CC} \pm 0.2V$
				45	120	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{IH}$
I_{CCD}	V_{CC} Deep Power-down Current	1		0.20	1.2	μA	$\overline{PWD}$ = GND ±0.2V

D.C. CHARACTERISTICS V_{CC} = 3.3V ±0.3V (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{CCR}	V_{CC} Read Current for 28F200BX-L Word-Wide Mode	1, 5, 6			22	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 5 MHz, I_{OUT} = 0 mA CMOS Inputs
					22	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 5 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCR}	V_{CC} Read Current for 28F002BX-L Byte-Wide Mode	1, 5, 6			20	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 5 MHz, I_{OUT} = mA CMOS Inputs
					20	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 5 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCW}	V_{CC} Word Write Current	1			30	mA	Word Write in Progress
I_{CCW}	V_{CC} Byte Write Current	1			30	mA	Word/Byte Write in Progress
I_{CCE}	V_{CC} Block Erase Current	1			20	mA	Block Erase in Progress
I_{CCES}	V_{CC} Erase Suspend Current	1, 2		3	6	mA	Block Erase Suspended, $\overline{CE} = V_{IH}$
I_{PPS}	V_{PP} Standby Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PPD}	V_{PP} Deep Power-down Current	1			5.0	μA	$\overline{PWD}$ = GND ±0.2V
I_{PPR}	V_{PP} Read Current	1			200	μA	$V_{PP} > V_{CC}$
I_{PPW}	V_{PP} Word Write Current	1			40	mA	$V_{PP} = V_{PPH}$ Word Write in Progress
I_{PPW}	V_{PP} Byte Write Current	1			30	mA	$V_{PP} = V_{PPH}$ Byte Write in Progress
I_{PPE}	V_{PP} Block Erase Current	1			30	mA	$V_{PP} = V_{PPH}$ Block Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1			200	μA	$V_{PP} = V_{PPH}$ Block Erase Suspended
I_{ID}	A_9 Intelligent Identifier Current	1			500	μA	$A_9 = V_{ID}$
V_{ID}	A_9 Intelligent Identifier Voltage		11.4	12.0	13.0	V	
V_{IL}	Input Low Voltage		−0.5		0.6	V	
V_{IH}	Input High Voltage		2.0		V_{CC}+0.5	V	
V_{OL}	Output Low Voltage				0.4	V	$V_{CC} = V_{CC}$ Min I_{OL} = 2 mA
V_{OH}	Output High Voltage		2.4			V	$V_{CC} = V_{CC}$ Min I_{OH} = −2 mA
V_{PPL}	V_{PP} during Normal Operations	3	0.0		4.1	V	
V_{PPH}	V_{PP} during Erase/Write Operations		11.4	12.0	12.6	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.0			V	
V_{HH}	$\overline{PWD}$ Unlock Voltage		11.4	12.0	13.0	V	Boot Block Write/Erase

CAPACITANCE(4) $T_A = 25°C$, f = 1 MHz

Symbol	Parameter	Typ	Max	Unit	Condition
C_{IN}	Input Capacitance	6	8	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	10	12	pF	$V_{OUT} = 0V$

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 3.3V, V_{PP} = 12.0V, T = 25°C. These currents are valid for all product versions (packages and speeds).
2. I_{CCES} is specified with the device deselected. If the device is read while in Erase Suspend Mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Block Erases and Word/Byte Writes are inhibited when $V_{PP} = V_{PPL}$ and not guaranteed in the range between V_{PPH} and V_{PPL}.
4. Sampled, not 100% tested.
5. Automatic Power Savings (APS) reduces I_{CCR} to less than 2 mA in static operation.
6. CMOS Inputs are either V_{CC} ±0.2V or GND ±0.2V. TTL Inputs are either V_{IL} or V_{IH}.

D.C. CHARACTERISTICS V_{CC} = 5.0V ±10%(4)

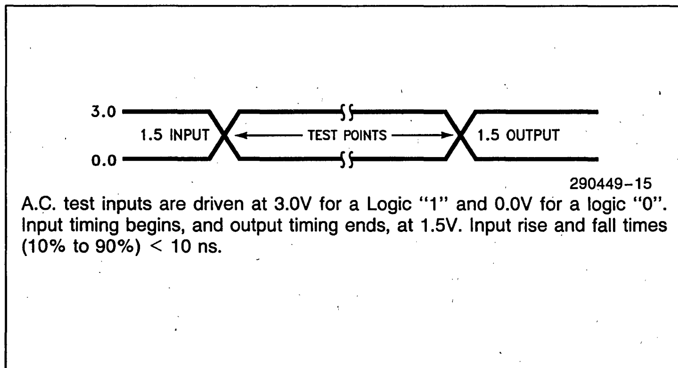
Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{LI}	Input Load Current	1			±1.0	mA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1			±10	mA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND
I_{CCS}	V_{CC} Standby Current				1.5	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{IH}$
					100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{CC}$ ±0.2V
I_{CCD}	V_{CC} Deep Power-down Current	1			1.2	μA	$\overline{PWD}$ = GND ±0.2V
I_{CCR}	V_{CC} Read Current for 28F200BX-L Word-Wide Mode	1			45	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 5 MHz, I_{OUT} = 0 mA CMOS Inputs
					45	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 5 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCR}	V_{CC} Read Current for 28F200BX-L Byte-Wide Mode	1			40	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 5 MHz, I_{OUT} = 0 mA CMOS Inputs
					40	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 5 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCW}	V_{CC} Word Write Current	1			70	mA	Word Write in Progress
I_{CCW}	V_{CC} Byte Write Current	1			60	mA	Byte Write in Progress
I_{CCE}	V_{CC} Block Erase Current	1			30	mA	Block Erase in Progress
I_{CCES}	V_{CC} Erase Suspend Current	1, 2			10	mA	Block Erase Suspended, $\overline{CE} = V_{IH}$
I_{PPS}	V_{PP} Standby Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PPD}	V_{PP} Deep Power-down Current	1			5.0	μA	$\overline{PWD}$ = GND ±0.2V

D.C. CHARACTERISTICS V_{CC} = 5.0V ±10% (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{PPR}	V_{PP} Read Current	1			200	μA	$V_{PP} > V_{CC}$
I_{PPW}	V_{PP} Word Write Current	1			40	mA	$V_{PP} = V_{PPH}$ Word Write in Progress
I_{PPW}	V_{PP} Byte Write Current	1			30	mA	$V_{PP} = V_{PPH}$ Byte Write in Progress
I_{PPE}	V_{PP} Block Erase Current	1			30	mA	$V_{PP} = V_{PPH}$ Block Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1			200	μA	$V_{PP} = V_{PPH}$ Block Erase Suspended
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC}+0.5	V	
V_{OL}	Output Low Voltage				0.45	V	$V_{CC} = V_{CC}$ Min I_{OL} = 5.8 mA
V_{OH}	Output High Voltage		2.4			V	$V_{CC} = V_{CC}$ Min I_{OH} = −2.5 mA
V_{PPL}	V_{PP} during Normal Operations	3	0.0		6.5	V	
V_{PPH}	V_{PP} during Erase/Write Operations		11.4	12.0	12.6	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.2			V	
V_{HH}	$\overline{PWD}$ Unlock Voltage		11.4	12.0	13.0	V	Boot Block Write/Erase

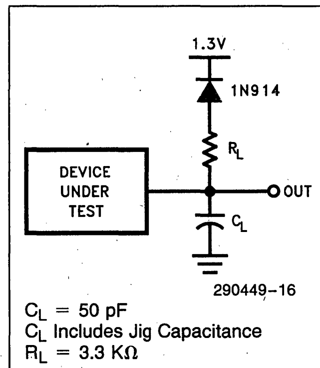
NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C. These currents are valid for all product versions (packages and speeds).
2. I_{CCES} is specified with the device deselected. If the device is read while in Erase Suspend Mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Block Erase/Byte Writes are inhibited when $V_{PP} = V_{PPL}$ and not guaranteed in the range between V_{PPH} and V_{PPL}.
4. All parameters are sampled, not 100% tested.

A.C. INPUT/OUTPUT REFERENCE WAVEFORM



A.C. test inputs are driven at 3.0V for a Logic "1" and 0.0V for a logic "0". Input timing begins, and output timing ends, at 1.5V. Input rise and fall times (10% to 90%) < 10 ns.

A.C. TESTING LOAD CIRCUIT



C_L = 50 pF
C_L Includes Jig Capacitance
R_L = 3.3 KΩ

3

A.C. CHARACTERISTICS—Read-Only Operations(1) V_{CC} = 3.3V ±0.3V, 5.0V ±10%(3)

Versions				28F200BX-L150 28F002BX-L150		Unit
Symbol		Parameter	Notes	Min	Max	
t_{AVAV}	t_{RC}	Read Cycle Time		150		ns
t_{AVQV}	t_{ACC}	Address to Output Delay			150	ns
t_{ELQV}	t_{CE}	$\overline{CE}$ to Output Delay	2		150	ns
t_{PHQV}	t_{PWH}	$\overline{PWD}$ High to Output Delay			700	ns
t_{GLQV}	t_{OE}	$\overline{OE}$ to Output Delay	2		65	ns
t_{ELQX}	t_{LZ}	$\overline{CE}$ to Output Low Z	3	0		ns
t_{EHQZ}	t_{HZ}	$\overline{CE}$ High to Output High Z	3		55	ns
t_{GLQX}	t_{OLZ}	$\overline{OE}$ to Output Low Z	3	0		ns
t_{GHQZ}	t_{DF}	$\overline{OE}$ High to Output High Z	3		45	ns
	t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change, Whichever is First	3	0		ns
t_{ELFL} t_{ELFH}		$\overline{CE}$ to $\overline{BYTE}$ Switching Low to High	3		5	ns
t_{FHQV}		$\overline{BYTE}$ Switching High to Valid Output Delay	3, 4		150	ns
t_{FLQZ}		$\overline{BYTE}$ Switching Low to Output High Z	3		45	ns

NOTES:
1. See A.C. Input/Output Reference Waveform for timing measurements.
2. $\overline{OE}$ may be delayed up to $t_{CE}-t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.
3. Sampled, not 100% tested.
4. t_{FLQV}, $\overline{BYTE}$ switching low to valid output delay will be equal to t_{AVQV}, measured from the time DQ_{15}/A_{-1} becomes valid.

V_{CC} POWER-UP
STANDBY
DEVICE AND ADDRESS SELECTION
OUTPUTS ENABLED
DATA VALID
STANDBY
V_{CC} POWER-DOWN

ADDRESSES (A) V_{IH} V_{IL}
ADDRESSES STABLE
t_{AVAV}

$\overline{CE}$ (E) V_{IH} V_{IL}
t_{EHQZ}

$\overline{OE}$ (G) V_{IH} V_{IL}
t_{GHQZ}

$\overline{WE}$ (W) V_{IH} V_{IL}
t_{GLQV}
t_{ELQV}
t_{GLQX}
t_{ELQX}
t_{OH}

DATA (D/Q) V_{IH} V_{IL}
HIGH Z
VALID OUTPUT
HIGH Z
t_{AVQV}

V_{CC} 5.0V GND
t_{PHQV}

$\overline{PWD}$ (P) V_{IH} V_{IL}

290449-17

Figure 16. A.C. Waveforms for Read Operations

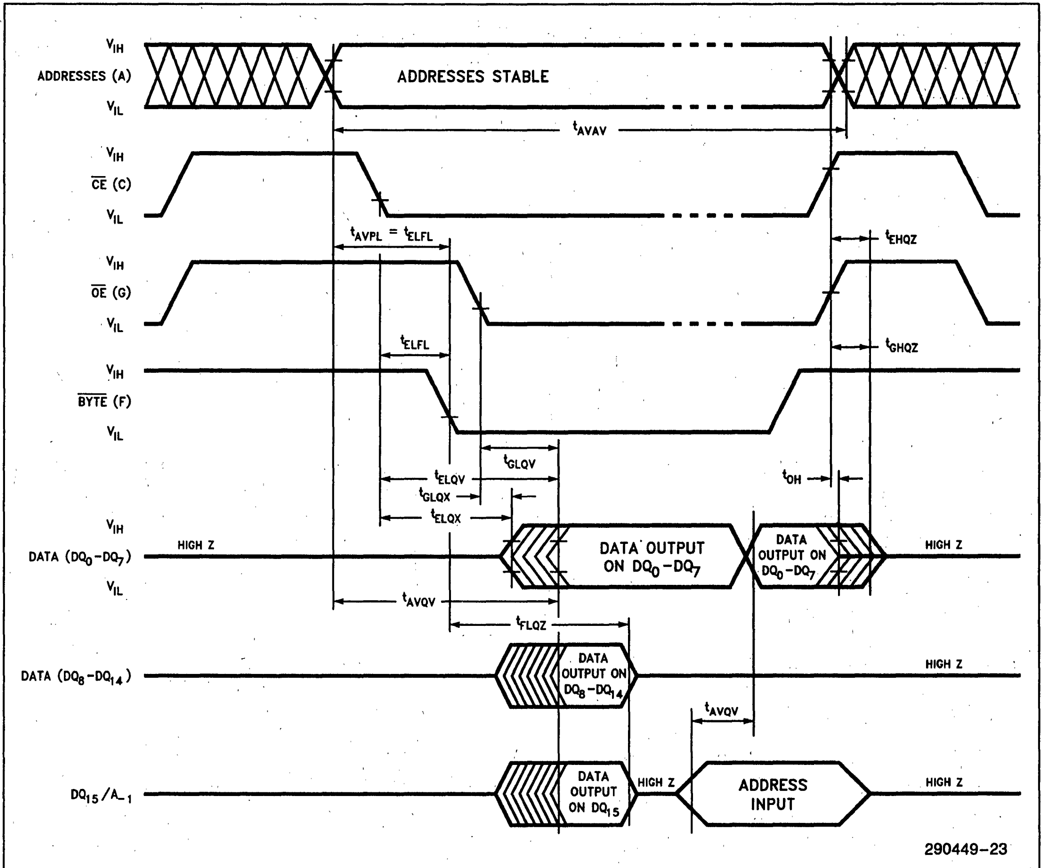


Figure 17. BYTE Timing Diagram for Both Read and Write Operations for 28F200BX-L

A.C. CHARACTERISTICS For $\overline{WE}$ Controlled Write Operations[1] V_{CC} = 3.3V ±0.3V, 5.0V ±10%

Versions[4]				28F200BX-L150 28F002BX-L150		Unit
Symbol		Parameter	Notes	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		150		ns
t_{PHWL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{WE}$ Going Low		1		μs
t_{ELWL}	t_{CS}	$\overline{CE}$ Setup to $\overline{WE}$ Going Low		0		ns
t_{PHHWH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{WE}$ Going High	6, 8	200		ns
t_{VPWH}	t_{VPS}	V_{PP} Setup to $\overline{WE}$ Going High	5, 8	200		ns
t_{AVWH}	t_{AS}	Address Setup to $\overline{WE}$ Going High	3	95		ns
t_{DVWH}	t_{DS}	Data Setup to $\overline{WE}$ Going High	4	100		ns
t_{WLWH}	t_{WP}	$\overline{WE}$ Pulse Width		100		ns
t_{WHDX}	t_{DH}	Data Hold from $\overline{WE}$ High	4	0		ns
t_{WHAX}	t_{AH}	Address Hold from $\overline{WE}$ High	3	10		ns
t_{WHEH}	t_{CH}	$\overline{CE}$ Hold from $\overline{WE}$ High		10		ns
t_{WHWL}	t_{WPH}	$\overline{WE}$ Pulse Width High		50		ns
t_{WHQV1}		Duration of Programming Operation (Boot)	2, 5, 6	6		μs
t_{WHQV2}		Duration of Word/Byte Programming Operation	2, 5, 6	0.3		s
t_{WHQV3}		Duration of Erase Operation (Parameter)	2, 5, 6	0.3		s
t_{WHQV4}		Duration of Erase Operation (Main)	2, 5, 6	0.6		s
t_{QWL}	t_{VPH}	V_{PP} Hold from Valid SRD	5, 8	0		ns
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	6, 8	0		ns
t_{PHBR}		Boot-Block Relock Delay	7, 8		200	ns

NOTES:
1. Read timing characteristics during write and erase operations are the same as during read-only operations. Refer to AC characteristics during Read Mode.
2. The on-chip WSM completely automates program/erase operations; program/erase algorithms are now controlled internally which includes verify and margining operations.
3. Refer to command definition table for valid A_{IN}.
4. Refer to command definition table for valid D_{IN}.
5. Program/Erase durations are measured to valid SRD data (successful operation, SR.7 = 1).
6. For Boot Block Program/Erase, $\overline{PWD}$ should be held at V_{HH} until operation completes successfully.
7. Time t_{PHBR} is required for successful relocking of the Boot Block.
8. Sampled but not 100% tested.

BLOCK ERASE AND BYTE/WORD WRITE PERFORMANCE V_{CC} = 3.3V ±0.3V, 5.0V ±10%

Parameter	Notes	28F200BX-L150 28F002BX-L150			Unit
		Min	Typ(1)	Max	
Boot/Parameter Block Erase Time	2		2.0	8.6	s
Main Block Erase Time	2		3.4	17.0	s
Main Block Byte Program Time	2		1.4	5.3	s
Main Block Word Program Time	2		0.7	2.7	s

NOTES:
1. 25°C, 12.0V V_{PP}.
2. Excludes System-Level Overhead.

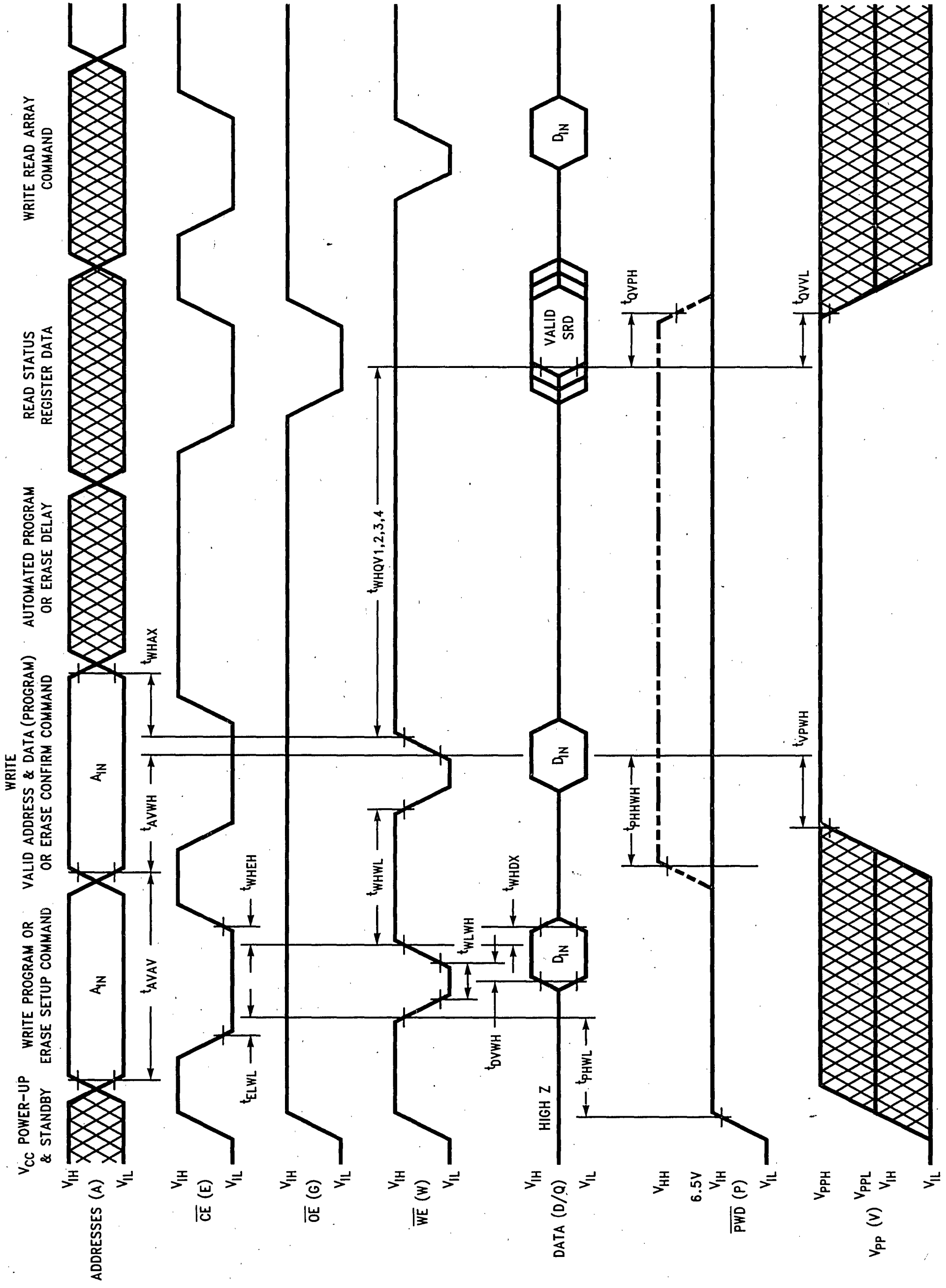


Figure 18. A.C. Waveforms for a Write and Erase Operations ($\overline{WE}$-Controlled Writes)

A.C. CHARACTERISTICS FOR $\overline{CE}$-CONTROLLED WRITE OPERATIONS

V_{CC} = 3.3V ±0.3V, 5.0V ±10%

Versions				28F200BX-L150 28F002BX-L150		Unit
Symbol		Parameter	Notes	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		150		ns
t_{PHEL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{CE}$ Going Low		1		µs
t_{WLEL}	t_{WS}	$\overline{WE}$ Setup to $\overline{CE}$ Going Low		0		ns
t_{PHHEH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{CE}$ Going High	6, 8	200		ns
t_{VPEH}	t_{VPS}	V_{PP} Setup to $\overline{CE}$ Going High	5, 8	200		ns
t_{AVEH}	t_{AS}	Address Setup to $\overline{CE}$ Going High	3	95		ns
t_{DVEH}	t_{DS}	Data Setup to $\overline{CE}$ Going High	4	100		ns
t_{ELEH}	t_{CP}	$\overline{CE}$ Pulse Width		100		ns
t_{EHDX}	t_{DH}	Data Hold from $\overline{CE}$ High	4	0		ns
t_{EHAX}	t_{AH}	Address Hold from $\overline{CE}$ High	3	10		ns
t_{EHWH}	t_{WH}	$\overline{WE}$ Hold from $\overline{CE}$ High		10		ns
t_{EHEL}	t_{CPH}	$\overline{CE}$ Pulse Width High		50		ns
t_{EHQV1}		Duration of Word/Byte Programming Operation (Boot)	2, 5, 6	6		µs
t_{EHQV2}		Duration of Erase Operation (Boot)	2, 5, 6	0.3		s
t_{EHQV3}		Duration of Erase Operation (Parameter)	2, 5, 6	0.3		s
t_{EHQV4}		Duration of Erase Operation (Main)	2, 5, 6	0.6		s
t_{QWL}	t_{VPH}	V_{PP} Hold from Valid SRD	5, 8	0		ns
t_{QVPH}	t_{PPH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	6, 8	0		ns
t_{PHBR}		Boot-Block Relock Delay	7		200	ns

NOTES:

1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of $\overline{CE}$ and $\overline{WE}$ in systems where $\overline{CE}$ defines the write pulse-width (within a longer $\overline{WE}$ timing waveform), all set-up, hold and inactive $\overline{WE}$ time should be measured relative to the $\overline{CE}$ waveforms.

2, 3, 4, 5, 6, 7, 8: Refer to AC characteristics for $\overline{WE}$-controlled write operations.

9. Read timing characteristics during write and erase operations are the same as during read-only operations. Refer to AC characteristics during read mode.

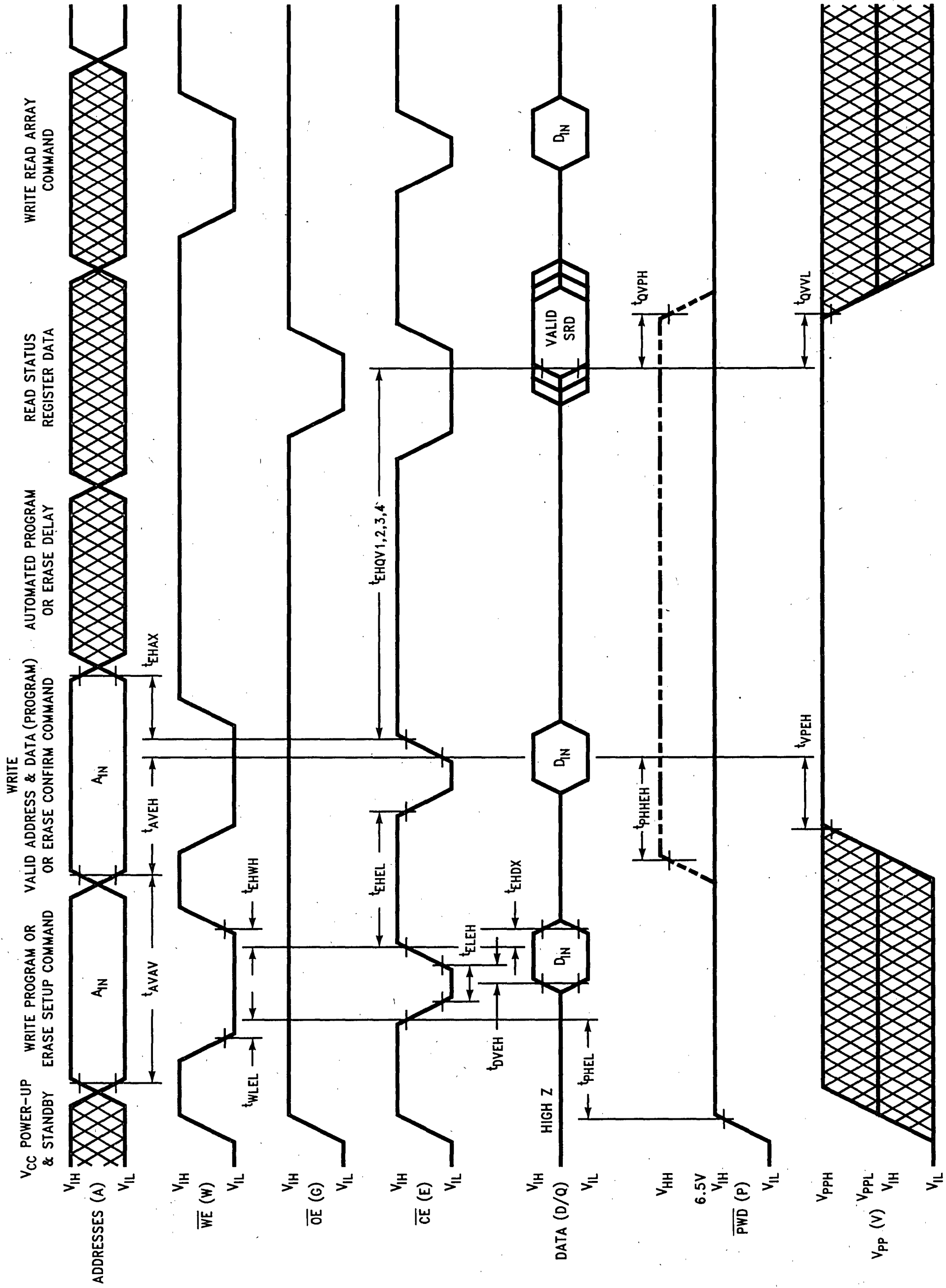


Figure 19. Alternate A.C. Waveforms for Write and Erase Operations ($\overline{CE}$-Controlled Writes)

ORDERING INFORMATION

E	2	8	F	2	0	0	B	X	-	L	1	5	0

PACKAGE
E = STANDARD 56 LEAD TSOP
PA = 44 LEAD PSOP

ACCESS SPEED (ns)
150 ns

290449-20

VALID COMBINATIONS:
E28F200BX-L150 PA28F200BX-L150

E	2	8	F	0	0	2	B	X	-	L	1	5	0

PACKAGE
E = STANDARD 40 LEAD TSOP

ACCESS SPEED (ns)
150 ns

290449-21

VALID COMBINATIONS:
E28F002BX-L150

Additional Information	Order Number
28F200BX/28F002BX Datasheet	290448
28F400BX/28F004BX Datasheet	290451
28F400BXL/28F004BXL Datasheet	290450
AP-363 "Extended Flash BIOS Design for Portable Computers"	292098
AP-357 "Power Supply Solutions for Flash Memory"	292092
ER-28 "ETOX-III Flash Memory Technology"	294012
ER-29 "The Intel 2/4 Mbit Boot Block Flash Memory Family"	294013

REVISION HISTORY

Number	Description
-001	Original Version
-002	Modified $\overline{BYTE}$ AC Timings Modified t_{DVWH} parameter for AC Characteristics for Write Operations

28F400BX-T/B, 28F004BX-T/B 4 MBIT (256K x16, 512K x8) BOOT BLOCK FLASH MEMORY FAMILY

- x8/x16 Input/Output Architecture
  - 28F400BX-T, 28F400BX-B
  - For High Performance and High Integration 16-bit and 32-bit CPUs
- x8-only Input/Output Architecture
  - 28F004BX-T, 28F004BX-B
  - For Space Constrained 8-bit Applications
- Optimized High Density Blocked Architecture
  - One 16 KB Protected Boot Block
  - Two 8 KB Parameter Blocks
  - One 96 KB Main Block
  - Three 128 KB Main Blocks
  - Top or Bottom Boot Locations
- Extended Cycling Capability
  - 100,000 Block Erase Cycles
- Automated Word/Byte Write and Block Erase
  - Command User Interface
  - Status Registers
  - Erase Suspend Capability
- SRAM-Compatible Write Interface
- Automatic Power Savings Feature
  - 1 mA Typical I_{CC} Active Current in Static Operation
- Very High-Performance Read
  - 60/80 ns Maximum Access Time
  - 30/40 ns Maximum Output Enable Time
- Low Power Consumption
  - 20 mA Typical x8 Active Read Current
  - 25 mA Typical x16 Active Read Current
- Deep Power-Down/Reset Input
  - 0.2 μA I_{CC} Typical
  - Acts as Reset for Boot Operations
- Extended Temperature Operation
  - $-40°C$ to $+85°C$
- Write Protection for Boot Block
- Hardware Data Protection Feature
  - Erase/Write Lockout During Power Transitions
- Industry Standard Surface Mount Packaging
  - 28F400BX: JEDEC ROM Compatible
    - 44-Lead PSOP
    - 56-Lead TSOP
  - 28F004BX: 40-Lead TSOP
- 12V Word/Byte Write and Block Erase
  - V_{PP} = 12V ±5% Standard
  - V_{PP} = 12V ±10% Option
- ETOX™ III Flash Technology
  - 5V Read

November 1992
Order Number: 290451-002

Intel's 4 Mbit Flash Memory Family is an extension of the Boot Block Architecture which includes block-selective erasure, automated write and erase operations and standard microprocessor interface. The 4 Mbit Flash Memory Family enhances the Boot Block Architecture by adding more density and blocks, x8/x16 input/output control, very high speed, low power, an industry standard ROM compatible pinout and surface mount packaging. The 4 Mbit flash family is an easy upgrade from Intel's 2 Mbit Boot Block Flash Memory Family.

The Intel 28F400BX-T/B are 16-bit wide flash memory offerings. These high density flash memories provide user selectable bus operation for either 8-bit or 16-bit applications. The 28F400BX-T and 28F400BX-B are 4,194,304-bit non-volatile memories organized as either 524,288 bytes or 262,144 words of information. They are offered in 44-Lead plastic SOP and 56-Lead TSOP packages. The x8/x16 pinout conforms to the industry standard ROM/EPROM pinout.

The Intel 28F004BX-T/B are 8-bit wide flash memories with 4,194,304 bits organized as 524,288 bytes of information. They are offered in a 40-Lead TSOP package, which is ideal for space-constrained portable systems.

These devices use an integrated Command User Interface (CUI) and Write State Machine (WSM) for simplified word/byte write and block erasure. The 28F400BX-T/28F004BX-T provide block locations compatible with Intel's MCS-186 family, 80286, i386™, i486™, i860™ and 80960CA microprocessors. The 28F400BX-B/28F004BX-B provide compatibility with Intel's 80960KX and 80960SX families as well as other embedded microprocessors.

The boot block includes a data protection feature to protect the boot code in critical applications. With a maximum access time of 60 ns, these 4 Mbit flash devices are very high performance memories which interface at zero-wait-state to a wide range of microprocessors and microcontrollers. A deep power-down mode lowers the total V_{CC} power consumption to 1 μW. This is critical in handheld battery powered systems. For very low power applications using a 3.3V supply, refer to the Intel 28F400BX-T/B, 28F004BX-T/B 4 Mbit Boot Block Flash Memory Family datasheet.

Manufactured on Intel's 0.8 micron ETOX™ III process, the 4 Mbit flash memory family provides world class quality, reliability and cost-effectiveness at the 4 Mbit density level.

1.0 PRODUCT FAMILY OVERVIEW

Throughout this datasheet the 28F400BX refers to both the 28F400BX-T and 28F400BX-B devices and 28F004BX refers to both the 28F004BX-T and 28F004BX-B devices. The 4 Mbit flash memory family refers to both the 28F400BX and 28F004BX products. This datasheet comprises the specifications for four separate products in the 4 Mbit flash memory family. Section 1 provides an overview of the 4 Mbit flash memory family including applications, pinouts and pin descriptions. Sections 2 and 3 describe in detail the specific memory organizations for the 28F400BX and 28F004BX products respectively. Section 4 combines a description of the family's principles of operations. Finally Section 5 describes the family's operating specifications.

Product Family

X8/X16 Products	X8-Only Products
28F400BX-T	28F004BX-T
28F400BX-B	28F004BX-B

1.1 Main Features

The 28F400BX/28F004BX boot block flash memory family is a very high performance 4 Mbit (4,194,304 bit) memory family organized as either 256 KWords (262,144 words) of 16 bits each or 512 KBytes (524,288 bytes) of 8 bits each.

Seven Separately Erasable Blocks including a **Hardware-Lockable boot block** (16,384 Bytes), **Two parameter blocks** (8,192 Bytes each) and **Four main blocks** (1 block of 98,304 Bytes and 3 blocks of 131,072 Bytes) are included on the 4 Mbit family. An erase operation erases one of the main blocks in typically 2.4 seconds and the boot or parameter blocks in typically 1.0 seconds independent of the remaining blocks. Each block can be independently erased and programmed 100,000 times.

The Boot Block is located at either the top (28F400BX-T, 28F004BX-T) or the bottom (28F400BX-B, 28F004BX-B) of the address map in order to accommodate different microprocessor protocols for boot code location. The **hardware lockable boot block** provides the most secure code storage. The boot block is intended to store the kernel code required for booting-up a system. When the $\overline{\text{PWD}}$ pin is between 11.4V and 12.6V the boot block is unlocked and program and erase operations can be performed. When the $\overline{\text{PWD}}$ pin is at or below 6.5V the boot block is locked and program and erase operations to the boot block are ignored.

The 28F400BX products are available in the ROM/EPROM compatible pinout and housed in the 44-Lead PSOP (Plastic Small Outline) package and the 56-Lead TSOP (Thin Small Outline, 1.2mm thick) package as shown in Figures 3 and 4. The 28F004BX products are available in the 40-Lead TSOP (1.2mm thick) package as shown in Figure 5.

The **Command User Interface (CUI)** serves as the interface between the microprocessor or microcontroller and the internal operation of the 28F400BX and 28F004BX flash products.

Program and Erase Automation allows program and erase operations to be executed using a two-write command sequence to the CUI. The internal Write State Machine (WSM) automatically executes the algorithms and timings necessary for program and erase operations, including verifications, thereby unburdening the microprocessor or microcontroller. Writing of memory data is performed in word or byte increments for the 28F400BX family and in byte increments for the 28F004BX family typically within 9 μs which is a 100% improvement over current flash memory products.

The **Status Register (SR)** indicates the status of the WSM and whether the WSM successfully completed the desired program or erase operation.

Maximum Access Time of **60 ns (TACC)** is achieved over the commercial temperature range (0°C to 70°C), 5% V_{CC} supply voltage range (4.75V to 5.25V) and 30 pF output load. Maximum Access Time of **70 ns (TACC)** is achieved over the commercial temperature range, 10% V_{CC} supply range (4.5V to 5.5V) and 100 pF output load.

I_{PP} maximum Program current is 40 mA for x16 operation and 30 mA for x8 operation. I_{PP} Erase current is 30 mA maximum. V_{PP} erase and programming voltage is 11.4V to 12.6V (V_{PP} = 12V ± 5%) under all operating conditions. As an option, V_{PP} can also vary between 10.8V to 13.2V (V_{PP} = 12V ± 10%) with a guaranteed number of 100 block erase cycles.

Typical I_{CC} Active Current of 25 mA is achieved for the X16 products (28F400BX). **Typical I_{CC} Active Current of 20 mA** is achieved for the X8 products (28F400BX, 28F004BX). Refer to the I_{CC} active current derating curves in this datasheet.

The 4 Mbit boot block flash family is also designed with an Automatic Power Savings (APS) feature to minimize system battery current drain and allows for very low power designs. Once the device is accessed to read array data, APS mode will immediately put the memory in static mode of operation where I_{CC} active current is typically 1 mA until the next read is initiated.

When the $\overline{CE}$ and $\overline{PWD}$ pins are at V_{CC} and the $\overline{BYTE}$ pin (28F400BX-only) is at either V_{CC} or GND the **CMOS Standby** mode is enabled where I_{CC} is typically **50 μA.**

A **Deep Power-Down Mode** is enabled when the $\overline{PWD}$ pin is at ground minimizing power consumption and providing write protection during power-up conditions. **I_{CC} current** during deep power-down mode is **0.20 μA typical.** An initial maximum access time or Reset Time of 300 ns is required from $\overline{PWD}$ switching until outputs are valid. Equivalently, the device has a maximum wake-up time of 215 ns until writes to the Command User Interface are recognized. When $\overline{PWD}$ is at ground the WSM is reset, the Status Register is cleared and the entire device is protected from being written to. This feature prevents data corruption and protects the code stored in the device during system reset. The system Reset pin can be tied to $\overline{PWD}$ to reset the memory to normal read mode upon activation of the Reset pin. With on-chip program/erase automation in the 4 Mbit family and the $\overline{PWD}$ functionality for data protection, when the CPU is reset and even if a program or erase command is issued, the device will not recognize any operation until $\overline{PWD}$ returns to its normal state.

For the 28F400BX, Byte-wide or Word-wide Input/Output Control is possible by controlling the $\overline{BYTE}$ pin. When the $\overline{BYTE}$ pin is at a logic low the device is in the byte-wide mode (x8) and data is read and written through DQ[0:7]. During the byte-wide mode, DQ[8:14] are tri-stated and DQ15/A-1 becomes the lowest order address pin. When the $\overline{BYTE}$ pin is at a logic high the device is in the word-wide mode (x16) and data is read and written through DQ[0:15].

1.2 Applications

The 4 Mbit boot block flash family combines high density, high performance, cost-effective flash memories with blocking and hardware protection capabilities. Its flexibility and versatility will reduce costs throughout the product life cycle. Flash memory is ideal for Just-In-Time production flow, reducing system inventory and costs, and eliminating component handling during the production phase.

During the product life cycle, when code updates or feature enhancements become necessary, flash memory will reduce the update costs by allowing either a user-performed code change via floppy disk or a remote code change via a serial link. The 4 Mbit boot block flash family provides full function, blocked flash memories suitable for a wide range of applications. These applications include **Extended PC BIOS and ROM-able** applications storage, Digital Cellular Phone program and data storage, **Telecommunication** boot/firmware, **Printer** firmware/font storage and various other embedded applications where both program and data storage are required.

Reprogrammable systems such as personal computers, are ideal applications for the 4 Mbit flash products. Portable and handheld personal computer applications are becoming more complex with the addition of power management software to take advantage of the latest microprocessor technology, the availability of ROM-based application software, pen tablet code for electronic hand writing, and diagnostic code. Figure 1 shows an example of a 28F400BX-T application.

This increase in software sophistication augments the probability that a code update will be required after the PC is shipped. The 4 Mbit flash products provide an inexpensive update solution for the notebook and handheld personal computers while extending their product lifetime. Furthermore, the 4 Mbit flash products' power-down mode provides added flexibility for these battery-operated portable designs which require operation at very low power levels.

The 4 Mbit flash products also provide excellent design solutions for Digital Cellular Phone and Telecommunication switching applications requiring high performance, high density storage capability coupled with modular software designs, and a small form factor package (X8-only bus). The 4 Mbit's blocking scheme allows for an easy segmentation of the embedded code with; 16 Kbytes of Hardware-Protected Boot code, 4 Main Blocks of program code and 2 Parameter Blocks of 8 Kbytes each for frequently updatable data storage and diagnostic messages (e.g., phone numbers, authorization codes). Figure 2 is an example of such an application with the 28F004BX-T.

These are a few actual examples of the wide range of applications for the 4 Mbit Boot Block flash memory family which enable system designers achieve the best possible product design. Only your imagination limits the applicability of such a versatile product family.

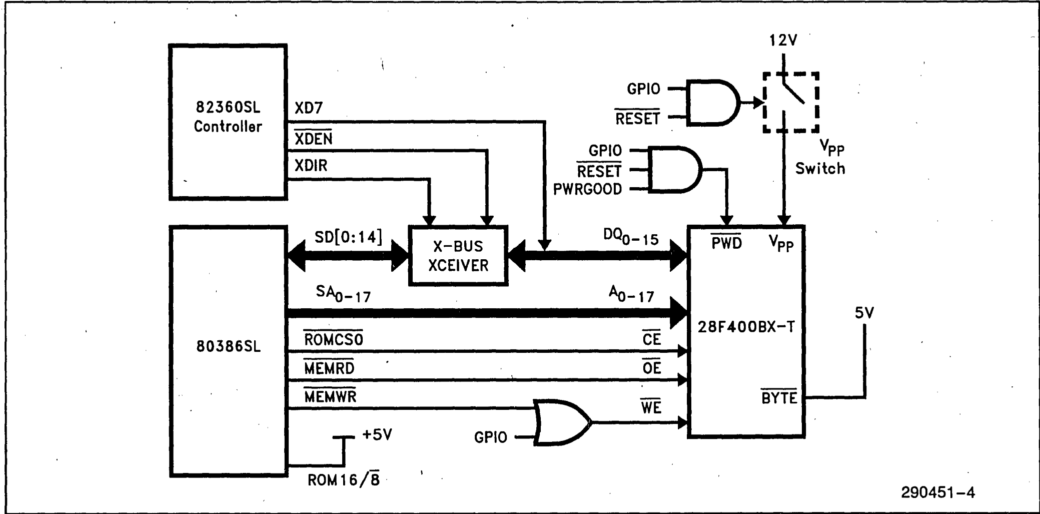


Figure 1. 28F400BX Interface to INTEL386SL™ Microprocessor Superset

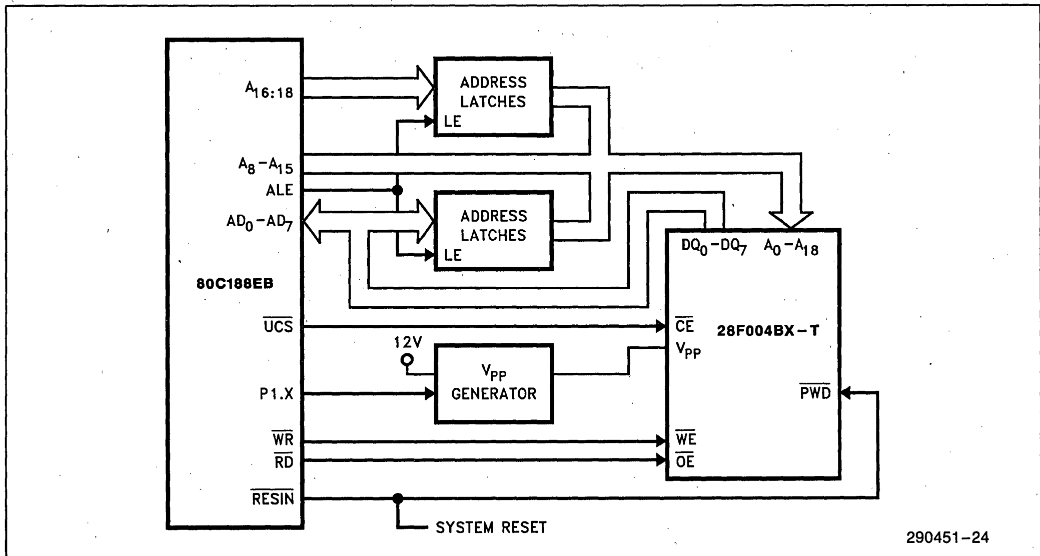


Figure 2. 28F004BX Interface to INTEL 80C188EB 8-Bit Embedded Microprocessor

1.3 Pinouts

The 28F400BX 44-Lead PSOP pinout follows the industry standard ROM/EPROM pinout as shown in Figure 3. Furthermore, the 28F400BX 56-Lead TSOP pinout shown in Figure 4 provides density upgrades to future higher density boot block memories.

The 28F004BX 40-Lead TSOP pinout shown in Figure 5 is 100% compatible and provides a density upgrade for the 2 Mbit Boot Block flash memory or the 28F002BX.

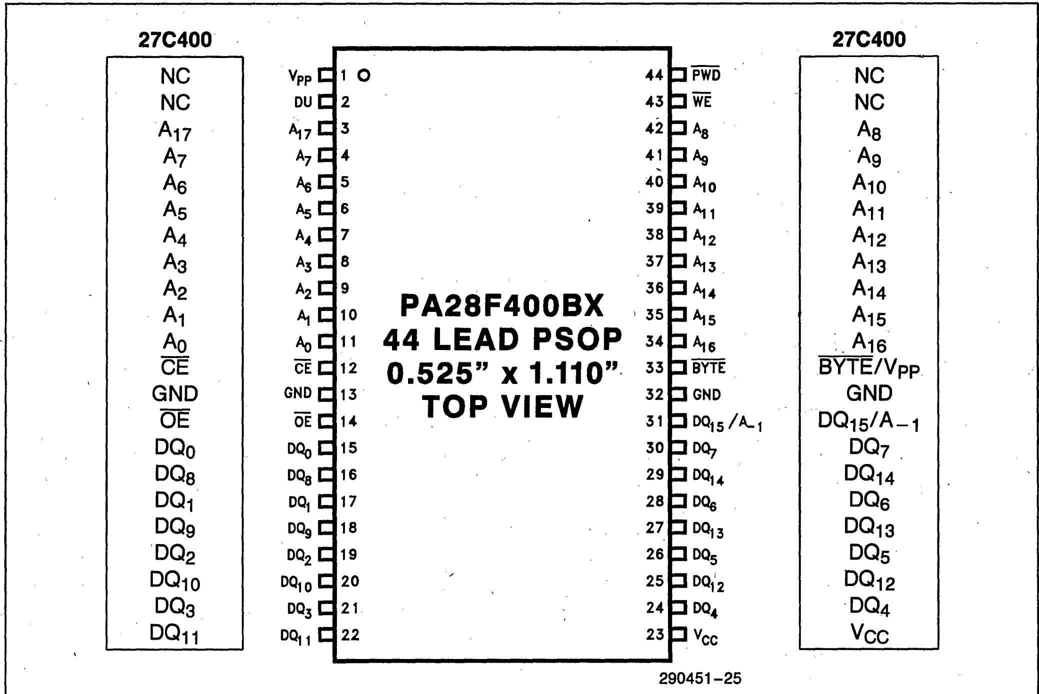


Figure 3. PSOP Lead Configuration for x8/x16 28F400BX

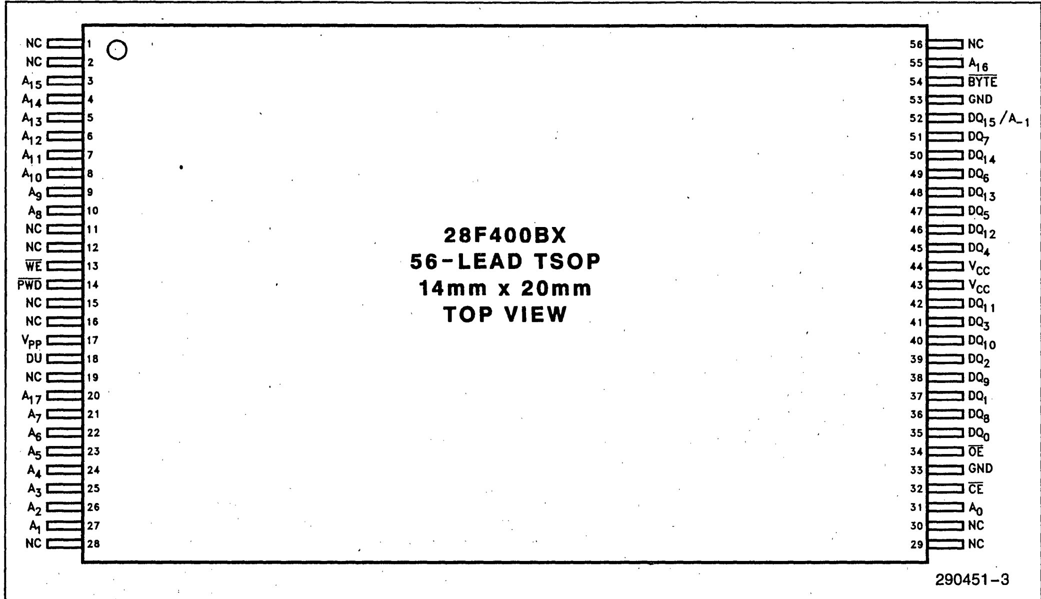


Figure 4. TSOP Lead Configuration for x8/x16 28F400BX

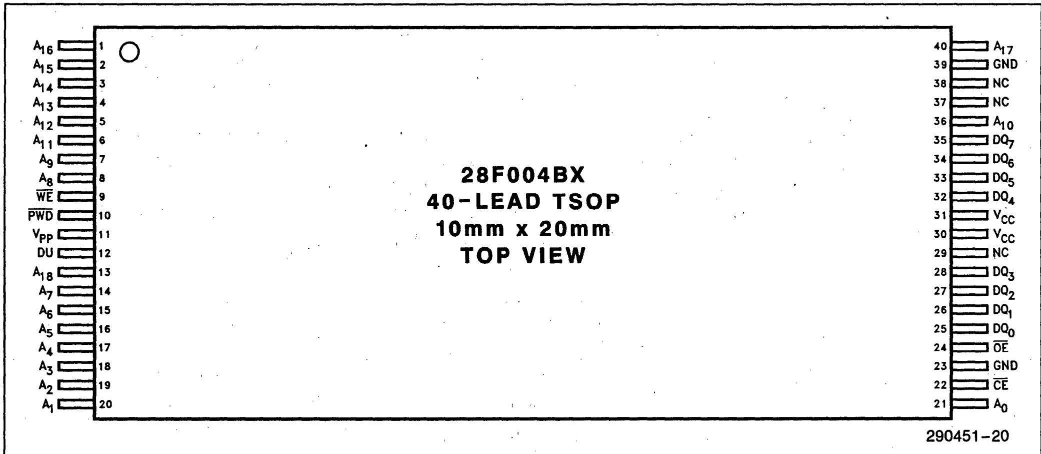


Figure 5. TSOP Lead Configuration for x8 28F004BX

1.4 28F400BX Pin Descriptions

Symbol	Type	Name and Function
A_0–A_{17}	I	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
A_9	I	**ADDRESS INPUT:** When A_9 is at 12V the signature mode is accessed. During this mode A_0 decodes between the manufacturer and device ID's. When $\overline{BYTE}$ is at a logic low only the lower byte of the signatures are read. DQ_{15}/A_{-1} is a don't care in the signature mode when $\overline{BYTE}$ is low.
DQ_0–DQ_7	I/O	**DATA INPUTS/OUTPUTS:** Inputs array data on the second $\overline{CE}$ and $\overline{WE}$ cycle during a program command. Inputs commands to the command user interface when $\overline{CE}$ and $\overline{WE}$ are active. Data is internally latched during the write and program cycles. Outputs array, intelligent identifier and Status Register data. The data pins float to tri-state when the chip is deselected or the outputs are disabled.
DQ_8–DQ_{15}	I/O	**DATA INPUTS/OUTPUTS:** Inputs array data on the second $\overline{CE}$ and $\overline{WE}$ cycle during a program command. Data is internally latched during the write and program cycles. Outputs array data. The data pins float to tri-state when the chip is deselected or the outputs are disabled as in the byte-wide mode ($\overline{BYTE}$ = "0"). In the byte-wide mode DQ_{15}/A_{-1} becomes the lowest order address for data output on DQ_0-DQ_7.
$\overline{CE}$	I	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels. If $\overline{CE}$ and $\overline{PWD}$ are high, but not at a CMOS high level, the standby current will increase due to current flow through the $\overline{CE}$ and $\overline{PWD}$ input stages.
$\overline{PWD}$	I	**POWER-DOWN:** Provides three-state control. Puts the device in deep power-down mode. Locks the boot block from program/erase. When $\overline{PWD}$ is at logic high level and equals 6.5V maximum the boot block is locked and cannot be programmed or erased. When $\overline{PWD}$ = 11.4V minimum the boot block is unlocked and can be programmed or erased. When $\overline{PWD}$ is at a logic low level the boot block is locked, the deep power-down mode is enabled and the WSM is reset preventing any blocks from being programmed or erased, therefore providing data protection during power transitions. $\overline{PWD}$ terminates any internally timed erase or program activities when it is taken to a logic low. PSD activates the $\overline{CE}$ input stage and requires 300 ns recovery time to transition from deep powerdown to valid data on the outputs or 215 ns delay before the device can recognize valid inputs.
$\overline{OE}$	I	**OUTPUT ENABLE:** Gates the device's outputs through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	I	**WRITE ENABLE:** Controls writes to the Command Register and array blocks. $\overline{WE}$ is active low. Addresses and data are latched on the rising edge of the $\overline{WE}$ pulse.
$\overline{BYTE}$	I	**$\overline{BYTE}$ ENABLE:** Controls whether the device operates in the byte-wide mode (x8) or the word-wide mode (x16). $\overline{BYTE}$ pin must be controlled at CMOS levels to meet 100A CMOS current in the standby mode. $\overline{BYTE}$ = "0" enables the byte-wide mode, where data is read and programmed on DQ_0–DQ_7 and DQ_{15}/A_{-1} becomes the lowest order address that decodes between the upper and lower byte. DQ_8–DQ_{14} are tri-stated during the byte-wide mode. $\overline{BYTE}$ = "1" enables the word-wide mode where data is read and programmed on DQ_0–DQ_{15}.
V_{PP}		**PROGRAM/ERASE POWER SUPPLY:** For erasing memory array blocks or programming data in each block. **Note:** $V_{PP} < V_{PPLMAX}$ memory contents cannot be altered.
V_{CC}		**DEVICE POWER SUPPLY (5V ±10%, 5V ±5%)**
GND		**GROUND:** For all internal circuitry.
NC		**NO CONNECT:** Pin may be driven or left floating.
DU		**DON'T USE PIN:** Pin should not be connected to anything.

1.5 28F004BX Pin Descriptions

Symbol	Type	Name and Function
A_0–A_{18}	I	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
A_9	I	**ADDRESS INPUT:** When A_9 is at 12V the signature mode is accessed. During this mode A_0 decodes between the manufacturer and device ID's.
DQ_0–DQ_7	I/O	**DATA INPUTS/OUTPUTS:** Inputs array data on the second $\overline{CE}$ and $\overline{WE}$ cycle during a program command. Inputs commands to the command user interface when $\overline{CE}$ and $\overline{WE}$ are active. Data is internally latched during the write and program cycles. Outputs array, intelligent identifier and status register data. The data pins float to tri-state when the chip is deselected or the outputs are disabled.
$\overline{CE}$	I	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels. If $\overline{CE}$ and $\overline{PWD}$ are high, but not at a CMOS high level, the standby current will increase due to current flow through the $\overline{CE}$ and $\overline{PWD}$ input stages.
$\overline{PWD}$	I	**POWERDOWN:** Provides Three-State control. Puts the device in deep power-down mode. Locks the Boot Block from program/erase. When $\overline{PWD}$ is at logic high level and equals 6.5V maximum the Boot Block is locked and cannot be programmed or erased. When $\overline{PWD}$ = 11.4V minimum the Boot Block is unlocked and can be programmed or erased. When $\overline{PWD}$ is at a logic low level the Boot Block is locked, the deep power-down mode is enabled and the WSM is reset preventing any blocks from being programmed or erased, therefore providing data protection during power transitions. $\overline{PWD}$ terminates any internally timed erase or program activities when it is taken to a logic low. $\overline{PWD}$ activates the $\overline{CE}$ input stage and requires 300 ns recovery time to transition from deep power-down to valid data on the outputs or 215 ns delay before the device can recognize valid inputs.
$\overline{OE}$	I	**OUTPUT ENABLE:** Gates the device's outputs through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	I	**WRITE ENABLE:** Controls writes to the Command Register and array blocks. $\overline{WE}$ is active low. Addresses and data are latched on the rising edge of the $\overline{WE}$ pulse.
V_{PP}		**PROGRAM/ERASE POWER SUPPLY:** For erasing memory array blocks or programming data in each block. **NOTE:** $V_{PP} < V_{PPLMAX}$ memory contents cannot be altered.
V_{CC}		**DEVICE POWER SUPPLY (5V ±10%, 5V ±5%)**
GND		**GROUND:** For all internal circuitry.
NC		**NO CONNECT:** Pin may be driven or left floating.
DU		**DON'T USE PIN:** Pin should not be connected to anything.

2.0 28F400BX WORD/BYTE-WIDE PRODUCTS DESCRIPTION

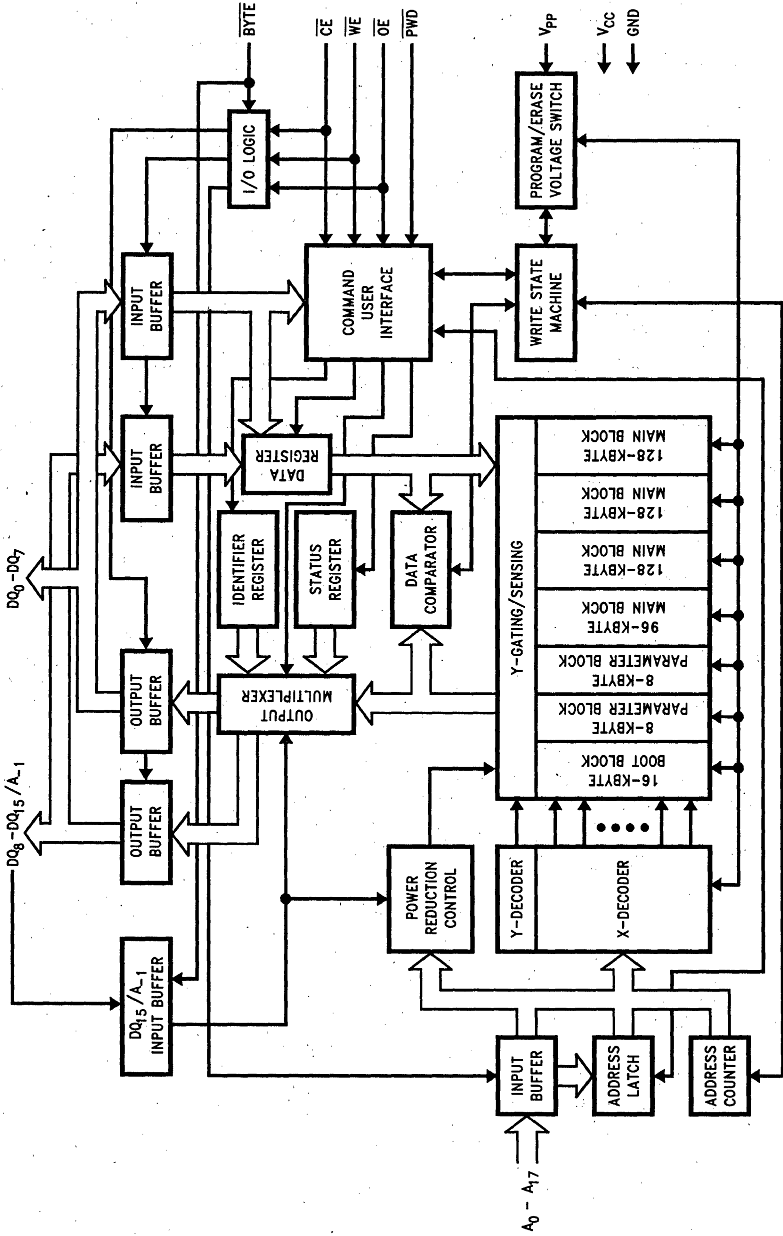


Figure 6. 28F400BX Word/Byte Block Diagram

2.1 28F400BX Memory Organization

2.1.1 BLOCKING

The 28F400BX uses a blocked array architecture to provide independent erasure of memory blocks. A block is erased independently of other blocks in the array when an address is given within the block address range and the Erase Setup and Erase Confirm commands are written to the CUI. The 28F400BX is a random read/write memory, only erasure is performed by block.

2.1.1.1 Boot Block Operation and Data Protection

The 16 Kbyte boot block provides a lock feature for secure code storage. The intent of the boot block is to provide a secure storage area for the kernel code that is required to boot a system in the event of power failure or other disruption during code update. This lock feature ensures absolute data integrity by preventing the boot block from being written or erased when $\overline{PWD}$ is not at 12V. The boot block can be erased and written when $\overline{PWD}$ is held at 12V for the duration of the erase or program operation. This allows customers to change the boot code when necessary while providing security when needed. See the Block Memory Map section for address locations of the boot block for the 28F400BX-T and 28F400BX-B.

2.1.1.2 Parameter Block Operation

The 28F400BX has 2 parameter blocks (8 Kbytes each). The parameter blocks are intended to provide storage for frequently updated system parameters and configuration or diagnostic information. The parameter blocks can also be used to store additional boot or main code. The parameter blocks however, do not have the hardware write protection feature that the boot block has. The parameter blocks provide for more efficient memory utilization when dealing with parameter changes versus regularly blocked devices. See the Block Memory Map section for address locations of the parameter blocks for the 28F400BX-T and 28F400BX-B.

2.1.1.3 Main Block Operation

Four main blocks of memory exist on the 28F400BX (3 x 128 Kbyte blocks and 1 x 96 Kbyte blocks). See the following section on Block Memory Map for the address location of these blocks for the 28F400BX-T and 28F400BX-B products.

2.1.2 BLOCK MEMORY MAP

Two versions of the 28F400BX product exist to support two different memory maps of the array blocks in order to accommodate different microprocessor protocols for boot code location. The 28F400BX-T memory map is inverted from the 28F400BX-B memory map.

2.1.2.1. 28F400BX-B Memory Map

The 28F400BX-B device has the 16 Kbyte boot block located from 00000H to 01FFFH to accommodate those microprocessors that boot from the bottom of the address map at 00000H. In the 28F400BX-B the first 8 Kbyte parameter block resides in memory space from 02000H to 02FFFH. The second 8 Kbyte parameter block resides in memory space from 03000H to 03FFFH. The 96 Kbyte main block resides in memory space from 04000H to 0FFFFH. The three 128 Kbyte main block resides in memory space from 10000H to 1FFFFH, 20000H to 2FFFFH and 30000H to 3FFFFH (word locations). See Figure 7.

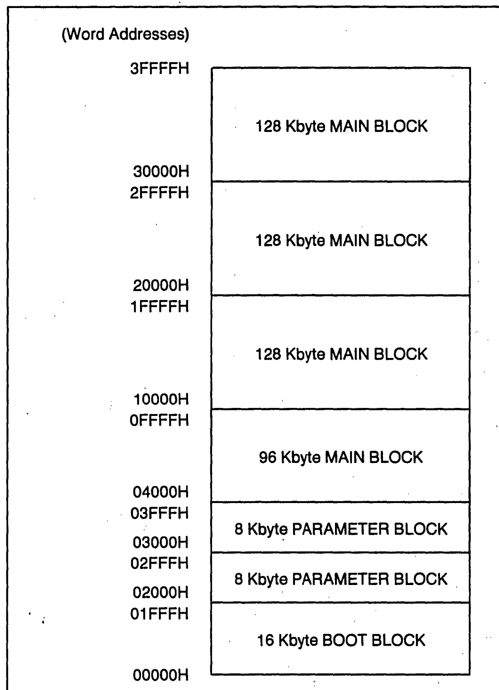


Figure 7. 28F400BX-B Memory Map

2.1.2.2 28F400BX-T Memory Map

The 28F400BX-T device has the 16 Kbyte boot block located from 3E000H to 3FFFFH to accommodate those microprocessors that boot from the top of the address map. In the 28F400BX-T the first 8 Kbyte parameter block resides in memory space from 3D000H to 3DFFFH. The second 8 Kbyte parameter block resides in memory space from 3C000H to 3CFFFH. The 96 Kbyte main block resides in memory space from 30000H to 3BFFFH. The three 128 Kbyte main blocks reside in memory space from 20000H to 2FFFFH, 10000H to 1FFFFH and 00000H to 0FFFFH as shown below in Figure 8.

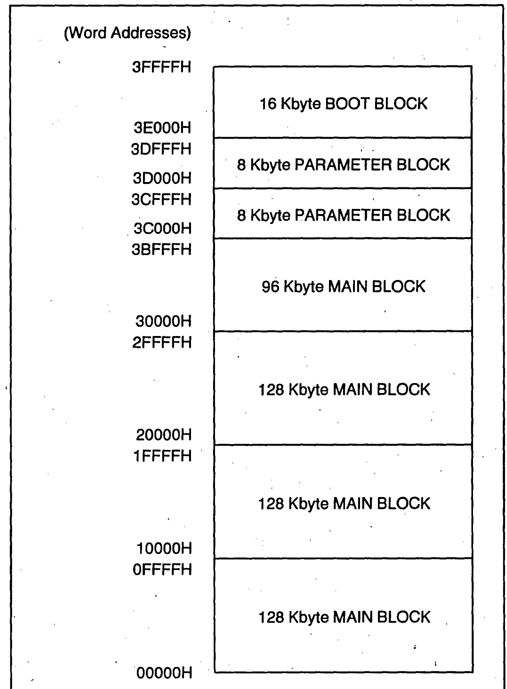


Figure 8. 28F400BX-T Memory Map

3.0 28F004BX PRODUCT DESCRIPTION

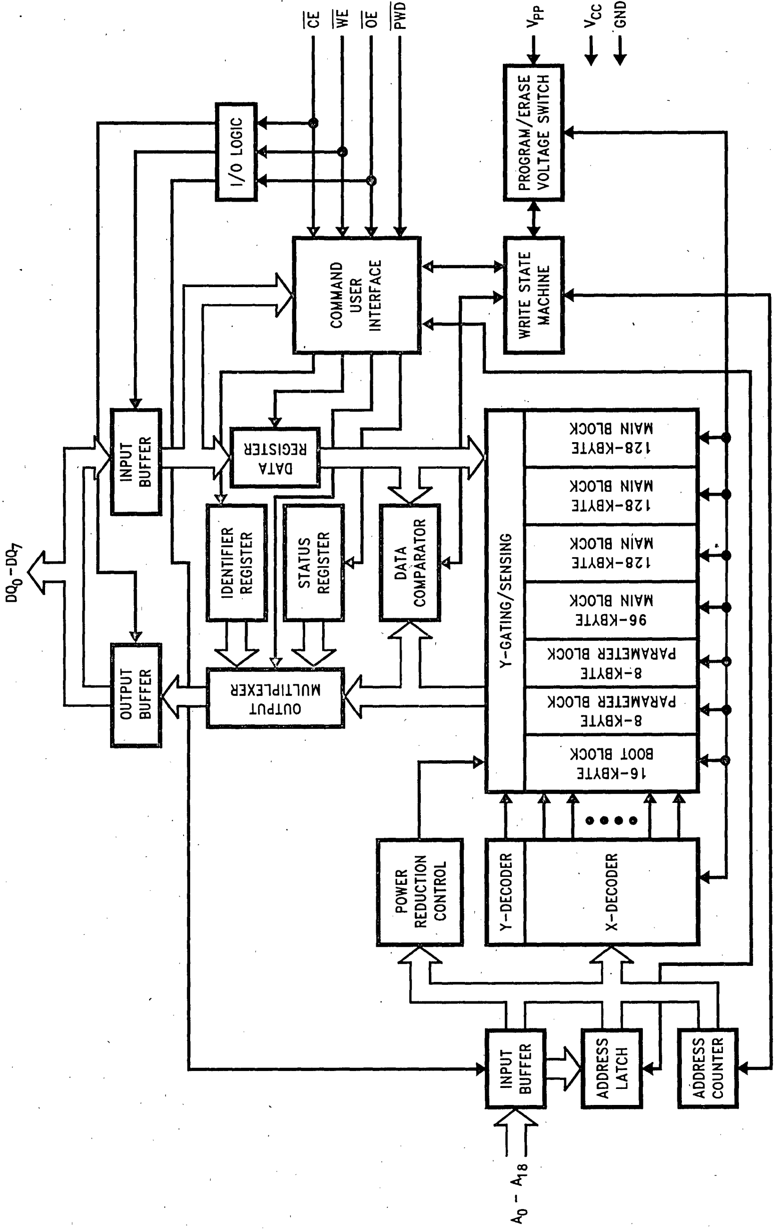


Figure 9. 28F004BX Byte-Wide Block Diagram

3.1 28F004BX Memory Organization

3.1.1 BLOCKING

The 28F004BX uses a blocked array architecture to provide independent erasure of memory blocks. A block is erased independently of other blocks in the array when an address is given within the block address range and the Erase Setup and Erase Confirm commands are written to the CUI. The 28F004BX is a random read/write memory, only erasure is performed by block.

3.1.1.1 Boot Block Operation and Data Protection

The 16 Kbyte boot block provides a lock feature for secure code storage. The intent of the boot block is to provide a secure storage area for the kernel code that is required to boot a system in the event of power failure or other disruption during code update. This lock feature ensures absolute data integrity by preventing the boot block from being programmed or erased when $\overline{PWD}$ is not at 12V. The boot block can be erased and programmed when $\overline{PWD}$ is held at 12V for the duration of the erase or program operation. This allows customers to change the boot code when necessary while still providing security when needed. See the Block Memory Map section for address locations of the boot block for the 28F004BX-T and 28F004BX-B.

3.1.1.2 Parameter Block Operation

The 28F004BX has 2 parameter blocks (8 KBytes each). The parameter blocks are intended to provide storage for frequently updated system parameters and configuration or diagnostic information. The parameter blocks can also be used to store additional boot or main code. The parameter blocks however, do not have the hardware write protection feature that the boot block has. Parameter blocks provide for more efficient memory utilization when dealing with small parameter changes versus regularly blocked devices. See the Block Memory Map section for address locations of the parameter blocks for the 28F004BX-T and 28F004BX-B.

3.1.1.3 Main Block Operation

Four main blocks of memory exist on the 28F004BX (3 × 128 KByte blocks and 1 × 96 KByte blocks). See the following section on Block Memory Map for the address location of these blocks for the 28F004BX-T and 28F004BX-B.

3.1.2 BLOCK MEMORY MAP

Two versions of the 28F004BX product exist to support two different memory maps of the array blocks in order to accommodate different microprocessor protocols for boot code location. The 28F004BX-T memory map is inverted from the 28F004BX-B memory map.

3.1.2.1 28F004BX-B Memory Map

The 28F004BX-B device has the 16 Kbyte boot block located from 00000H to 03FFFH to accommodate those microprocessors that boot from the bottom of the address map at 00000H. In the 28F004BX-B the first 8 Kbyte parameter block resides in memory from 04000H to 05FFFH. The second 8 Kbyte parameter block resides in memory space from 06000H to 07FFFH. The 96 Kbyte main block resides in memory space from 08000H to 1FFFFH. The three 128 Kbyte main block reside in memory space from 20000H to 3FFFFH, 40000H to 5FFFFH and 60000H to 7FFFFH. See Figure 10.

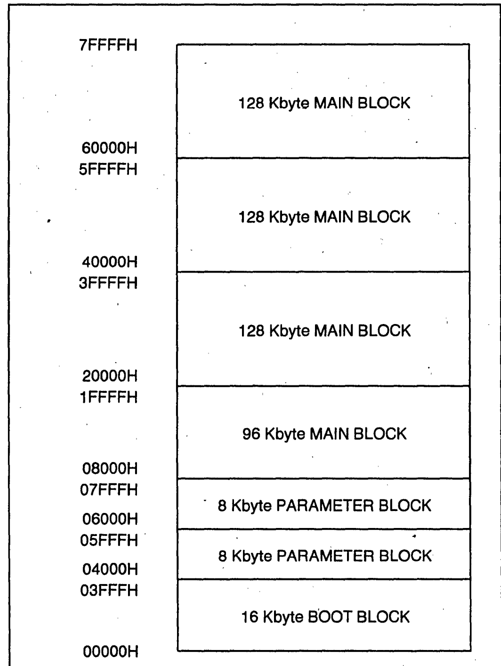


Figure 10. 28F004BX-B Memory Map

3.1.2.2 28F004BX-T Memory Map

The 28F004BX-T device has the 16 Kbyte boot block located from 7C000H to 7FFFFH to accommodate those microprocessors that boot from the top of the address map. In the 28F004BX-T the first 8 Kbyte parameter block resides in memory space from 7A000H to 7BFFFH. The second 8 Kbyte parameter block resides in memory space from 78000H to 79FFFH. The 96 Kbyte main block resides in memory space from 60000H to 77FFFH. The three 128 Kbyte main blocks reside in memory space from 40000H to 5FFFFH, 20000H to 3FFFFH and 00000H to 1FFFFH.

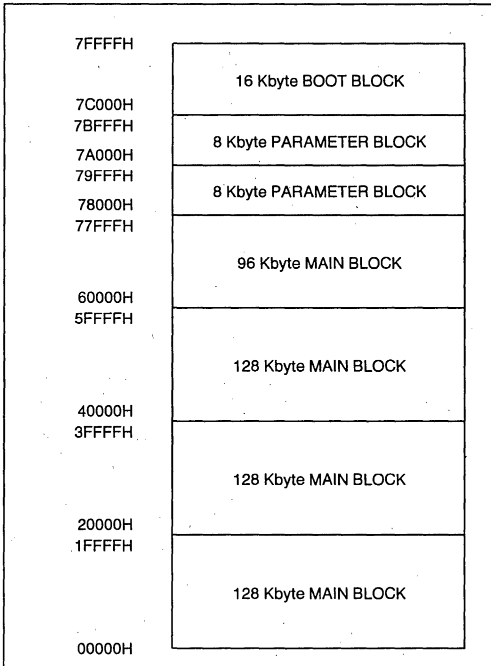


Figure 11. 28F004BX-T Memory Map

4.0 PRODUCT FAMILY PRINCIPLES OF OPERATION

Flash memory augments EPROM functionality with in-circuit electrical write and erase. The 4 Mbit flash family utilizes a Command User Interface (CUI) and internally generated and timed algorithms to simplify write and erase operations.

The CUI allows for 100% TTL-level control inputs, fixed power supplies during erasure and programming, and maximum EPROM compatibility.

In the absence of high voltage on the V_{PP} pin, the 4 Mbit boot block flash family will only successfully execute the following commands: Read Array, Read Status Register, Clear Status Register and Intelligent Identifier mode. The device provides standard EPROM read, standby and output disable operations. Manufacturer Identification and Device Identification data can be accessed through the CUI or through the standard EPROM A_9 high voltage access (V_{ID}) for PROM programming equipment.

The same EPROM read, standby and output disable functions are available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} allows write and erase of the device. All functions associated with altering memory contents: write and erase, Intelligent Identifier read and Read Status are accessed via the CUI.

The purpose of the Write State Machine (WSM) is to completely automate the write and erasure of the device. The WSM will begin operation upon receipt of a signal from the CUI and will report status back through a Status Register. The CUI will handle the $\overline{WE}$ interface to the data and address latches, as well as system software requests for status while the WSM is in operation.

4.1 28F400BX Bus Operations

Flash memory reads, erases and writes in-system via the local CPU. All bus cycles to or from the flash memory conform to standard microprocessor bus cycles.

Table 1. Bus Operations for WORD-WIDE Mode ($\overline{BYTE}$ = V_{IH})

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_9	A_0	V_{PP}	DQ_{0-15}
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	X	D_{OUT}
Output Disable		V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	High Z
Standby		V_{IH}	V_{IH}	X	X	X	X	X	High Z
Deep Power-Down	9	V_{IL}	X	X	X	X	X	X	High Z
Intelligent Identifier (Mfr)	4	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IL}	X	0089H
Intelligent Identifier (Device)	4, 5	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IH}	X	4470H 4471H
Write	6, 7, 8	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	D_{IN}

Table 2. Bus Operations for BYTE-WIDE Mode (BYTE = V_{IL})

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_9	A_0	A_{-1}	V_{PP}	DQ_{0-7}	DQ_{8-14}
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	X	X	D_{OUT}	High Z
Output Disable		V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	X	High Z	High Z
Standby		V_{IH}	V_{IH}	X	X	X	X	X	X	High Z	High Z
Deep Power-Down	9	V_{IL}	X	X	X	X	X	X	X	High Z	High Z
Intelligent Identifier (Mfr)	4	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IL}	X	X	89H	High Z
Intelligent Identifier (Device)	4, 5	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IH}	X	X	70H 71H	High Z
Write	6, 7, 8	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	X	D_{IN}	High Z

NOTES:
1. Refer to DC Characteristics.
2. X can be V_L, V_{IH} for control pins and addresses, V_{PPL} or V_{PPH} for V_{PP}.
3. See DC Characteristics for V_{PPL}, V_{PPH}, V_{HH}, V_{ID} voltages.
4. Manufacturer and Device codes may also be accessed via a CUI write sequence. A_1–A_{17} = X.
5. Device ID = 4470H for 28F400BX-T and 4471H for 28F400BX-B.
6. Refer to Table 4 for valid D_{IN} during a write operation.
7. Command writes for Block Erase or Word/Byte Write are only executed when V_{PP} = V_{PPH}.
8. To write or erase the boot block, hold $\overline{PWD}$ at V_{HH}.
9. $\overline{PWD}$ must be at GND ±0.2V to meet the 1.2 μA maximum deep power-down current.

4.2 28F004BX Bus Operations

Table 3. Bus Operations

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_9	A_0	V_{PP}	DQ_{0-7}
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	X	D_{OUT}
Output Disable		V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	High Z
Standby		V_{IH}	V_{IH}	X	X	X	X	X	High Z
Deep Power-Down	9	V_{IL}	X	X	X	X	X	X	High Z
Intelligent Identifier (Mfr)	4	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IL}	X	89H
Intelligent Identifier (Device)	4, 5	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{IH}	X	78H 79H
Write	6, 7, 8	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	D_{IN}

NOTES:
1. Refer to DC Characteristics.
2. X can be V_{IL} or V_{IH} for control pins and addresses, V_{PPL} or V_{PPH} for V_{PP}.
3. See DC Characteristics for V_{PPL}, V_{PPH}, V_{HH}, V_{ID} voltages.
4. Manufacturer and Device codes may also be accessed via a CUI write sequence. A_1-A_{18} = X.
5. Device ID = 78H for 28F004BX-T and 79H for 28F004BX-B.
6. Refer to Table 4 for valid D_{IN} during a write operation.
7. Command writes for Block erase or byte program are only executed when $V_{PP} = V_{PPH}$.
6. Program or erase the Boot block by holding $\overline{PWD}$ at V_{HH}.
9. $\overline{PWD}$ must be at GND ±0.2V to meet the 1.2 μA maximum deep power-down current.

4.3 Read Operations

The 4 Mbit boot block flash family has three user read modes; Array, Intelligent Identifier, and Status Register. Status Register read mode will be discussed in detail in the "Write Operations" section.

During power-up conditions (V_{CC} supply ramping), it takes a maximum of 600 ns from when V_{CC} is at 4.5V minimum to valid data on the outputs.

4.3.1 READ ARRAY

If the memory is not in the Read Array mode, it is necessary to write the appropriate read mode command to the CUI. The 4 Mbit boot block flash family has three control functions, all of which must be logically active, to obtain data at the outputs. Chip-Enable $\overline{CE}$ is the device selection control. Power-Down $\overline{PWD}$ is the device power control. Output-Enable $\overline{OE}$ is the DATA INPUT/OUTPUT (DQ[0:15] or DQ[0:7]) direction control and when active is used to drive data from the selected memory on to the I/O bus.

4.3.1.1 Output Control

With $\overline{OE}$ at logic-high level (V_{IH}), the output from the device is disabled and data input/output pins (DQ[0:15] or DQ[0:7] are tri-stated. Data input is then controlled by $\overline{WE}$.

4.3.1.2 Input Control

With $\overline{WE}$ at logic-high level (V_{IH}), input to the device is disabled. Data Input/Output pins (DQ[0:15] or DQ[0:7]) are controlled by $\overline{OE}$.

4.3.2 INTELLIGENT IDENTIFIERS

28F400BX PRODUCTS

The manufacturer and device codes are read via the CUI or by taking the A_9 pin to 12V. Writing 90H to the CUI places the device into Intelligent Identifier read mode. A read of location 00000H outputs the manufacturer's identification code, 0089H, and location 00001H outputs the device code; 4470H for 28F400BX-T, 4471H for 28F400BX-B. When $\overline{BYTE}$ is at a logic low only the lower byte of the above signatures is read and DQ_{15}/A_{-1} is a "don't care" during Intelligent Identifier mode. A read array command must be written to the memory to return to the read array mode.

28F004BX PRODUCTS

The manufacturer and device codes are also read via the CUI or by taking the A_9 pin to 12V. Writing 90H to the CUI places the device into Intelligent Identifier read mode. A read of location 00000H outputs the manufacturer's identification code, 89H, and location 00001H outputs the device code; 78H for 28F004BX-T, 79H for 28F004BX-B.

4.4 Write Operations

Commands are written to the CUI using standard microprocessor write timings. The CUI serves as the interface between the microprocessor and the internal chip operation. The CUI can decipher Read Array, Read Intelligent Identifier, Read Status Register, Clear Status Register, Erase and Program commands. In the event of a read command, the CUI simply points the read path at either the array, the Intelligent Identifier, or the status register depending on the specific read command given. For a program or erase cycle, the CUI informs the write state machine that a write or erase has been requested. During a program cycle, the Write State Machine will control the program sequences and the CUI will only respond to status reads. Durlng an erase cycle, the CUI will respond to status reads and erase suspend. After the Write State Machine has completed its task, it will allow the CUI to respond to its full command set. The CUI will stay in the current command state until the microprocessor issues another command.

The CUI will successfully initiate an erase or write operation only when V_{PP} is within its voltage range. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable, available only when memory updates are desired. The system designer can also choose to "hard-wire" V_{PP} to 12V. The 4 Mbit boot block flash family is designed to accommodate—either design practice. It is recommended that $\overline{PWD}$ be tied to logical Reset for data protection during unstable CPU reset function as described in the "Product Family Overview" section.

4.4.1 BOOT BLOCK WRITE OPERATIONS

In the case of Boot Block modifications (write and erase), $\overline{PWD}$ is set to V_{HH} = 12V typically, in addition to V_{PP} at high voltage.

However, if $\overline{PWD}$ is not at V_{HH} when a program or erase operation of the boot block is attempted, the corresponding status register bit (Bit 4 for Program and Bit 5 for Erase, refer to Table 5 for Status Register Definitions) is set to indicate the failure to complete the operation.

4.4.2 COMMAND USER INTERFACE (CUI)

The Command User Interface (CUI) serves as the interface to the microprocessor. The CUI points the read/write path to the appropriate circuit block as described in the previous section. After the WSM has completed its task, it will set the WSM Status bit to a "1", which will also allow the CUI to respond to its full command set. Note that after the WSM has returned control to the CUI, the CUI will remain in its current state.

4.4.2.1 Command Set

Command Codes	Device Mode
00	Invalid/Reserved
10	Alternate Program Setup
20	Erase Setup
40	Program Setup
50	Clear Status Register
70	Read Status Register
90	Intelligent Identifier
B0	Erase Suspend
D0	Erase Resume/Erase Confirm
FF	Read Array

4.4.2.2 Command Function Descriptions

Device operations are selected by writing specific commands into the CUI. Table 4 defines the 4 Mbit boot block flash family commands.

Table 4. Command Definitions

Command	Bus Cycles Req'd	Notes	First Bus Cycle			Second Bus Cycle		
		8	Operation	Address	Data	Operation	Address	Data
Read Array	1	1	Write	X	FFH			
Intelligent Identifier	3	2, 4	Write	X	90H	Read	IA	IID
Read Status Register	2	3	Write	X	70H	Read	X	SRD
Clear Status Register	1		Write	X	50H			
Erase Setup/Erase Confirm	2	5	Write	BA	20H	Write	BA	D0H
Word/Byte Write Setup/Write	2	6, 7	Write	WA	40H	Write	WA	WD
Erase Suspend/Erase Resume	2		Write	X	B0H	Write	X	D0H
Alternate Word/Byte Write Setup/Write	2	6, 7	Write	WA	10H	Write	WA	WD

NOTES:
1. Bus operations are defined in Tables 1, 2, 3.
2. IA = Identifier Address: 00H for manufacturer code, 01H for device code.
3. SRD = Data read from Status Register.
4. IID = Intelligent Identifier Data.
Following the Intelligent Identifier Command, two read operations access manufacturer and device codes.
5. BA = Address within the block being erased.
6. WA = Address to be written.
WD = Data to be written at location WD.
7. Either 40H or 10H commands is valid.
8. When writing commands to the device, the upper data bus [DQ_8–DQ_{15}] = X (28F400BX-only) which is either V_{CC} or V_{SS} to avoid burning additional current.

Invalid/Reserved

These are unassigned commands. It is not recommended that the customer use any command other than the valid commands specified above. Intel reserves the right to redefine these codes for future functions.

Read Array (FFH)

This single write command points the read path at the array. If the host CPU performs a $\overline{CE}/\overline{OE}$ controlled read immediately following a two-write sequence that started the WSM, then the device will output status register contents. If the Read Array command is given after Erase Setup the device is reset to read the array. A two Read Array command sequence (FFH) is required to reset to Read Array after Program Setup.

Intelligent Identifier (90H)

After this command is executed, the CUI points the output path to the Intelligent Identifier circuits. Only Intelligent Identifier values at addresses 0 and 1 can be read (only address A_0 is used in this mode, all other address inputs are ignored).

Read Status Register (70H)

This is one of the two commands that is executable while the state machine is operating. After this command is written, a read of the device will output the contents of the status register, regardless of the address presented to the device.

The device automatically enters this mode after program or erase has completed.

Clear Status Register (50H)

The WSM can only set the Program Status and Erase Status bits in the status register, it can not clear them. Two reasons exist for operating the status register in this fashion. The first is a synchronization. The WSM does not know when the host CPU has read the status register, therefore it would not know when to clear the status bits. Secondly, if the CPU is programming a string of bytes, it may be more efficient to query the status register after programming the string. Thus, if any errors exist while programming the string, the status register will return the accumulated error status.

Program Setup (40H or 10H)

This command simply sets the CUI into a state such that the next write will load the address and data registers. Either 40H or 10H can be used for Program Setup. Both commands are included to accommodate efforts to achieve an industry standard command code set.

Program

The second write after the program setup command, will latch addresses and data. Also, the CUI initiates the WSM to begin execution of the program algorithm. While the WSM finishes the algorithm, the device will output Status Register contents. Note that the WSM cannot be suspended during programming.

Erase Setup (20H)

Prepares the CUI for the Erase Confirm command. No other action is taken. If the next command is not an Erase Confirm command then the CUI will set both the Program Status and Erase Status bits of the Status Register to a "1", place the device into the Read Array state, and wait for another command.

Erase Confirm (D0H)

If the previous command was an Erase Setup command, then the CUI will enable the WSM to erase, at the same time closing the address and data latches, and respond only to the Read Status Register and Erase Suspend commands. While the WSM is executing, the device will output Status Register data when $\overline{OE}$ is toggled low. Status Register data can only be updated by toggling either $\overline{OE}$ or $\overline{CE}$ low.

Erase Suspend (B0H)

This command only has meaning while the WSM is executing an Erase operation, and therefore will only be responded to during an erase operation. After this command has been executed, the CUI will set an output that directs the WSM to suspend Erase operations, and then return to responding to only Read Status Register or to the Erase Resume commands. Once the WSM has reached the Suspend state, it will set an output into the CUI which allows the CUI to respond to the Read Array, Read Status Register, and Erase Resume commands. In this mode, the CUI will not respond to any other commands. The WSM will also set the WSM Status bit to a "1". The WSM will continue to run, idling in the SUSPEND state, regardless of the state of all input control pins, with the exclusion of $\overline{PWD}$. $\overline{PWD}$ will immediately shut down the WSM and the remainder of the chip. During a suspend operation, the data and address latches will remain closed, but the address pads are able to drive the address into the read path.

Erase Resume (D0H)

This command will cause the CUI to clear the Suspend state and set the WSM Status bit to a "0", but only if an Erase Suspend command was previously issued. Erase Resume will not have any effect in all other conditions.

4.4.3 STATUS REGISTER

The 4 Mbit boot block flash family contains a status register which may be read to determine when a program or erase operation is complete, and whether that operation completed successfully. The status register may be read at any time by writing the Read Status command to the CUI. After writing this command, all subsequent Read operations output data from the status register until another command is written to the CUI. A Read Array command must be written to the CUI to return to the Read Array mode.

The status register bits are output on DQ[0:7] whether the device is in the byte-wide (x8) or word-wide (x16) mode for the 28F400BX. In the word-wide mode the upper byte, DQ[8:15] is set to 00H during a Read Status command. In the byte-wide mode, DQ[8:14] are tri-stated and DQ_{15}/A_{-1} retains the low order address function.

It should be noted that the contents of the status register are latched on the falling edge of $\overline{OE}$ or $\overline{CE}$ whichever occurs last in the read cycle. This prevents possible bus errors which might occur if the contents of the status register change while reading the status register. $\overline{CE}$ or $\overline{OE}$ must be toggled with each subsequent status read, or the completion of a program or erase operation will not be evident.

The Status Register is the interface between the microprocessor and the Write State Machine (WSM). When the WSM is active, this register will indicate the status of the WSM, and will also hold the bits indicating whether or not the WSM was successful in performing the desired operation. The WSM sets status bits "Three" through "Seven" and clears bits "Six" and "Seven", but cannot clear status bits "Three" through "Five". These bits can only be cleared by the controlling CPU through the use of the Clear Status Register command.

4.4.3.1 Status Register Bit Definition

Table 5. Status Register Definitions

WSMS	ESS	ES	PS	VPPS	R	R	R
7	6	5	4	3	2	1	0

	NOTES:
SR.7 = WRITE STATE MACHINE STATUS 1 = Ready 0 = Busy	Write State Machine Status bit must first be checked to determine byte/word program or block erase completion, before the Program or Erase Status bits are checked for success.
SR.6 = ERASE SUSPEND STATUS 1 = Erase Suspended 0 = Erase in Progress/Completed	When Erase Suspend is issued, WSM halts execution and sets both WSMS and ESS bits to "1". ESS bit remains set to "1" until an Erase Resume command is issued.
SR.5 = ERASE STATUS 1 = Error in Block Erasure 0 = Successful Block Erase	When this bit is set to "1". WSM has applied the maximum number of erase pulses to the block and is still unable to successfully perform an erase verify.
SR.4 = PROGRAM STATUS 1 = Error In Byte/Word Program 0 = Successful Byte/Word Program	When this bit is set to "1", WSM has attempted but failed to Program a byte or word.
SR.3 = V_{PP} STATUS 1 = V_{PP} Low Detect; Operation Abort 0 = V_{PP} OK	The V_{PP} Status bit unlike an A/D converter, does not provide continuous indication of V_{PP} level. The WSM interrogates the V_{PP} level only after the byte write or block erase command sequences have been entered and informs the system if V_{PP} has not been switched on. The V_{PP} Status bit is not guaranteed to report accurate feedback between V_{PPL} and V_{PPH}.
SR.2–SR.0 = RESERVED FOR FUTURE ENHANCEMENTS	These bits are reserved for future use and should be masked out when polling the Status Register.

4.4.3.2 Clearing the Status Register

Certain bits in the status register are set by the write state machine, and can only be reset by the system software. These bits can indicate various failure conditions. By allowing the system software to control the resetting of these bits, several operations may be performed (such as cumulatively programming several bytes or erasing multiple blocks in sequence). The status register may then be read to determine if an error occurred during that programming or erasure series. This adds flexibility to the way the device may be programmed or erased. To clear the status register, the Clear Status Register command is written to the CUI. Then, any other command may be issued to the CUI. Note again that before a read cycle can be initiated, a Read Array command must be written to the CUI to specify whether the read data is to come from the array, status register, or Intelligent Identifier.

4.4.4 PROGRAM MODE

Program is executed by a two-write sequence. The Program Setup command is written to the CUI followed by a second write which specifies the address and data to be programmed. The write state machine will execute a sequence of internally timed events to:

1. Program the desired bits of the addressed memory word (byte), and
2. Verify that the desired bits are sufficiently programmed.

Programming of the memory results in specific bits within a byte or word being changed to a "0".

If the user attempts to program "1"s, there will be no change of the memory cell content and no error occurs.

Similar to erasure, the status register indicates whether programming is complete. While the program sequence is executing, bit 7 of the status register is a "0". The status register can be polled by toggling either $\overline{CE}$ or $\overline{OE}$ to determine when the program sequence is complete. Only the Read Status Register command is valid while programming is active.

When programming is complete, the status bits, which indicate whether the program operation was successful, should be checked. If the programming operation was unsuccessful, Bit 4 of the status register is set to a "1" to indicate a Program Failure. If Bit 3 is set then V_{PP} was not within acceptable limits, and the WSM will not execute the programming sequence.

The status register should be cleared before attempting the next operation. Any CUI instruction can follow after programming is completed; however, it must be recognized that reads from the memory, status register, or Intelligent Identifier cannot be accomplished until the CUI is given the appropriate command. A Read Array command must first be given before memory contents can be read.

Figure 12 shows a system software flowchart for device byte programming operation. Figure 13 shows a similar flowchart for device word programming operation (28F400BX-only).

4.4.5 ERASE MODE

Erasure of a single block is initiated by writing the Erase Setup and Erase Confirm commands to the CUI, along with the addresses, A[12:17] for the 28F400BX or A[12:18] for the 28F004BX, identifying the block to be erased. These addresses are latched internally when the Erase Confirm command is issued. Block erasure results in all bits within the block being set to "1".

The WSM will execute a sequence of internally timed events to:

1. Program all bits within the block
2. Verify that all bits within the block are sufficiently programmed
3. Erase all bits within the block and
4. Verify that all bits within the block are sufficiently erased

While the erase sequence is executing, Bit 7 of the status register is a "0".

When the status register indicates that erasure is complete, the status bits, which indicate whether the erase operation was successful, should be checked. If the erasure operation was unsuccessful, Bit 5 of the status register is set to a "1" to indicate an Erase Failure. If V_{PP} was not within acceptable limits after the Erase Confirm command is issued, the WSM will not execute an erase sequence; instead, Bits of the status register is set to a "1" to indicate an Erase Failure, and Bit 3 is set to a "1" to identify that V_{PP} supply voltage was not within acceptable limits.

The status register should be cleared before attempting the next operation. Any CUI instruction can follow after erasure is completed; however, it must be recognized that reads from the memory array, status register, or Intelligent Identifier can not be accomplished until the CUI is given the appropriate command. A Read Array command must first be given before memory contents can be read.

Figure 14 shows a system software flowchart for Block Erase operation.

4.4.5.1 Suspending and Resuming Erase

Since an erase operation typically requires 1 to 3 seconds to complete, an Erase Suspend command is provided. This allows erase-sequence interruption in order to read data from another block of the memory. Once the erase sequence is started, writing the Erase Suspend command to the CUI requests that the Write State Machine (WSM) pause the erase sequence at a predetermined point in the erase algorithm. The status register must be read to determine when the erase operation has been suspended.

At this point, a Read Array command can be written to the CUI in order to read data from blocks other than that which is being suspended. The only other valid command at this time is the Erase Resume command or Read Status Register operation.

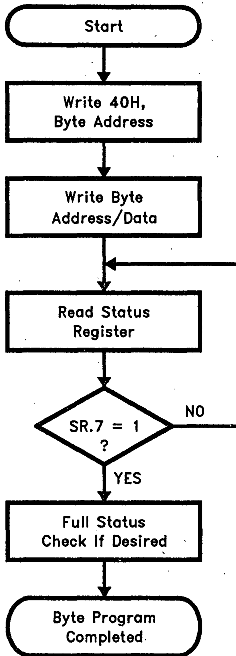
Figure 15 shows a system software flowchart detailing the operation.

During Erase Suspend mode, the chip can go into a pseudo-standby mode by taking $\overline{CE}$ to V_{IH} and the active current is now a maximum of 10 mA. If the chip is enabled while in this mode by taking $\overline{CE}$ to V_{IL}, the Erase Resume command can be issued to resume the erase operation.

Upon completion of reads from any block other than the block being erased, the Erase Resume command must be issued. When the Erase Resume command is given, the WSM will continue with the erase sequence and complete erasing the block. As with the end of erase, the status register must be read, cleared, and the next instruction issued in order to continue.

4.4.6 EXTENDED CYCLING

Intel has designed extended cycling capability into its ETOX III flash memory technology. The 4 Mbit boot block flash family is designed for 100,000 program/erase cycles on each of the seven blocks. The combination of low electric fields, clean oxide processing and minimized oxide area per memory cell subjected to the tunneling electric field, results in very high cycling capability.



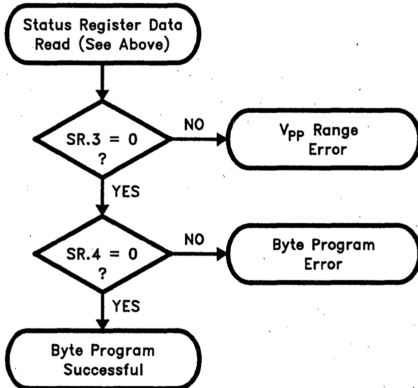
Bus Operation	Command	Comments
Write	Setup Program	Data = 40H Address = Byte to be programmed
Write	Program	Data to be programmed Address = Byte to be programmed
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent bytes.

Full status check can be done after each byte or after a sequence of bytes.

Write FFH after the last byte programming operation to reset the device to Read Array Mode.

Full Status Check Procedure



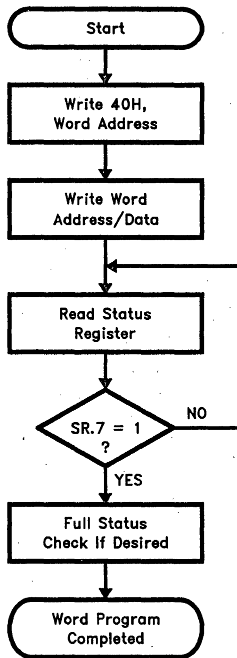
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Byte Program Error

SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple bytes are programmed before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 12. Automated Byte Programming Flowchart



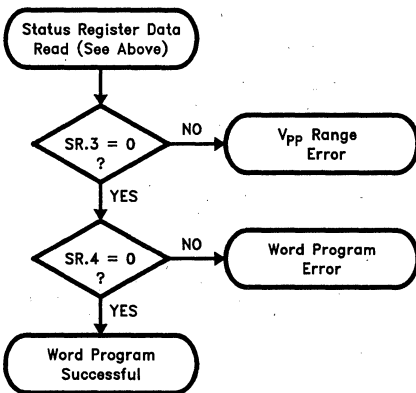
Bus Operation	Command	Comments
Write	Setup Program	Data = 40H Address = Word to be programmed
Write	Program	Data to be programmed Address = Word to be programmed
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent words.

Full status check can be done after each word or after a sequence of words.

Write FFH after the last word programming operation to reset the device to Read Array Mode.

Full Status Check Procedure



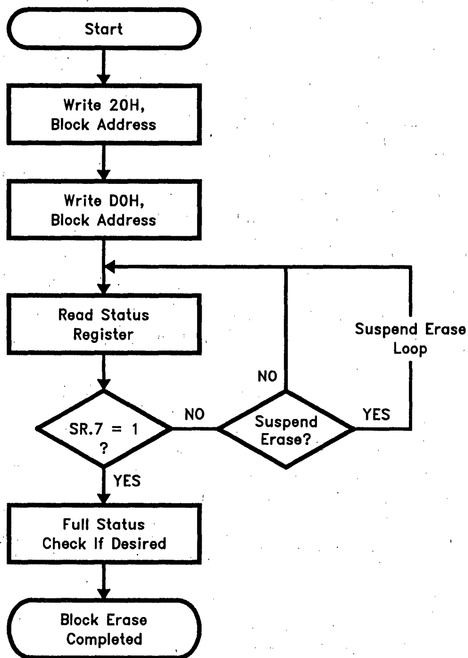
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Word Program Error

SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple words are programmed before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 13. Automated Word Programming Flowchart



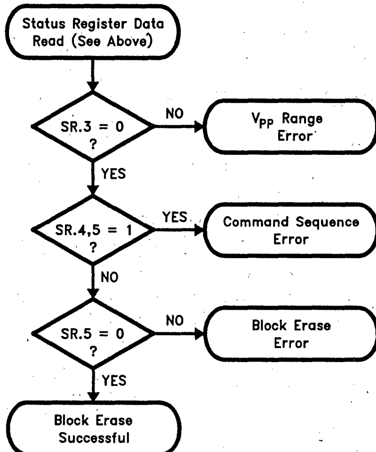
Bus Operation	Command	Comments
Write	Setup Erase	Data = 20H Address = Within block to be erased
Write	Erase	Data = D0H Address = Within block to be erased
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent blocks.

Full status check can be done after each block or after a sequence of blocks.

Write FFH after the last block erase operation to reset the device to Read Array Mode.

Full Status Check Procedure



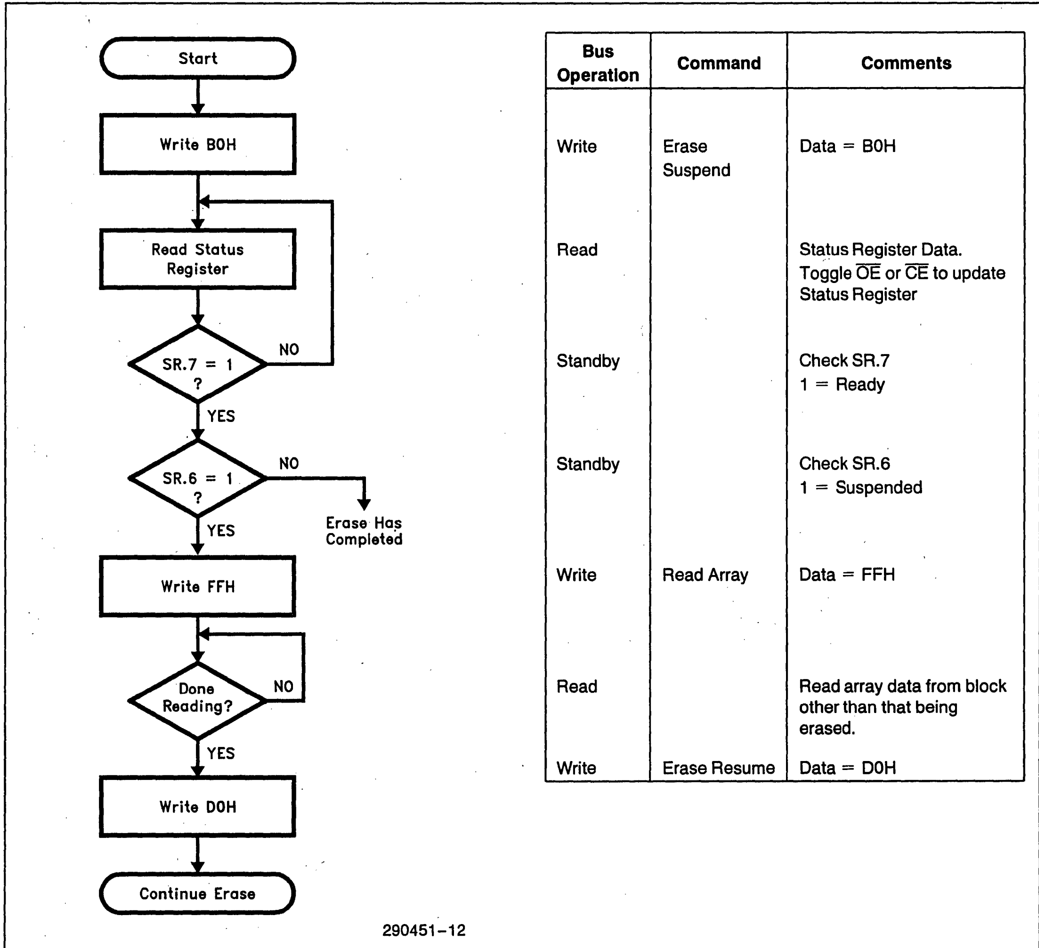
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4,5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error

SR.3 MUST be cleared, if set during an erase attempt, before further attempts are allowed by the Write State Machine.

SR.5 is only cleared by the Clear Status Register Command, in cases where multiple blocks are erased before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 14. Automated Block Erase Flowchart



Bus Operation	Command	Comments
Write	Erase Suspend	Data = B0H
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready
Standby		Check SR.6 1 = Suspended
Write	Read Array	Data = FFH
Read		Read array data from block other than that being erased.
Write	Erase Resume	Data = D0H

Figure 15. Erase Suspend/Resume Flowchart

4.5 Power Consumption

4.5.1 ACTIVE POWER

With $\overline{CE}$ at a logic-low level and $\overline{PWD}$ at a logic-high level, the device is placed in the active mode. The device I_{CC} current is a maximum 60 mA at 10 MHz with TTL input signals.

4.5.2 AUTOMATIC POWER SAVINGS

Automatic Power Savings (APS) is a low pwer feature during active mode of operation. The 4 Mbit family of products incorporate Power Reduction Control (PRC) circuitry which basically allows the device to put itself into a low current state when it is not being accessed. After data is read from the memory array, PRC logic controls the device's power consumption by entering the APS mode where maximum I_{CC} current is 3 mA and typical I_{CC} current is 1 mA. The device stays in this static state with outputs valid until a new location is read.

4.5.3 STANDBY POWER

With $\overline{CE}$ at a logic-high level (V_{IH}), and the CUI in read mode, the memory is placed in standby mode where the maximum I_{CC} standby current is 100 μA with CMOS input signals. The standby operation disables much of the device's circuitry and substantially reduces device power consumption. The outputs (DQ[0:15] or DQ[0:7]) are placed in a high-impedance state independent of the status of the $\overline{OE}$ signal. When the 4 Mbit boot block flash family is deselected during erase or program functions, the devices will continue to perform the erase or program function and consume program or erase active power until program or erase is completed.

4.5.4 DEEP POWERDOWN

The 4 Mbit boot block flash family supports a typical I_{CC} of 0.2 μA in deep power-down mode. One of the target markets for these devices is in portable equipment where the power consumption of the machine is of prime importance. The 4 Mbit boot block flash family has a $\overline{PWD}$ pin which places the device in the deep powerdown mode. When $\overline{PWD}$ is at a logic-low (GND ±0.2V), all circuits are turned off and the device typically draws 0.2 μA of V_{CC} current.

During read modes, the $\overline{PWD}$ pin going low deselects the memory and places the output drivers in a high impedance state. Recovery from the deep power-down state, requires a minimum of 400 ns to access valid data (t_{PHQV}).

During erase or program modes, $\overline{PWD}$ low will abort either erase or program operation. The contents of the memory are no longer valid as the data has been corrupted by the $\overline{PWD}$ function. As in the read mode above, all internal circuitry is turned off to achieve the 0.2 μA current level.

$\overline{PWD}$ transitions to V_{IL} or turning power off to the device will clear the status register.

4.6 Power-up Operation

The 4 Mbit boot block flash family is designed to offer protection against accidental block erasure or programming during power transitions. Upon power-up the 4 Mbit boot block flash family is indifferent as to which power supply, V_{PP} or V_{CC}, powers-up first. Power supply sequencing is not required.

The 4 Mbit boot block flash family ensures the CUI is reset to the read mode on power-up.

In addition, on power-up the user must either drop $\overline{CE}$ low or present a new address to ensure valid data at the outputs.

A system designer must guard against spurious writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either signal to V_{IH} will inhibit writes to the device. The CUI architecture provides an added level of protection since alteration of memory contents can only occur after successful completion of the two-step command sequences. Finally the device is disabled until $\overline{PWD}$ is brought to V_{IH}, regardless of the state of its control inputs. This feature provides yet another level of memory protection.

4.7 Power Supply Decoupling

Flash memory's power switching characteristics require careful device decoupling methods. System designers are interested in 3 supply current issues:

- Standby current levels (I_{CCS})
- Active current levels (I_{CCR})
- Transient peaks produced by falling and rising edges of $\overline{CE}$.

Transient current magnitudes depend on the device outputs' capacitive and inductive loading. Two-line control and proper decoupling capacitor selection will suppress these transient voltage peaks. Each flash device should have a 0.1 μF ceramic capacitor connected between each V_{CC} and GND, and between its V_{PP} and GND. These high frequency, low-inherent inductance capacitors should be placed as close as possible to the package leads.

4.7.1 V_{PP} TRACE ON PRINTED CIRCUIT BOARDS

Writing to flash memories while they reside in the target system, requires special consideration of the V_{PP} power supply trace by the printed circuit board designer. The V_{PP} pin supplies the flash memory cells current for programming and erasing. One should use similar trace widths and layout considerations given to the V_{CC} power supply trace. Adequate V_{PP} supply traces and decoupling will decrease spikes and overshoots.

4.7.2 V_{CC}, V_{PP} AND $\overline{PWD}$ TRANSITIONS

The CUI latches commands as issued by system software and is not altered by V_{PP} or $\overline{CE}$ transitions or WSM actions. Its state upon power-up, after exit from deep power-down mode or after V_{CC} transitions below V_{LKO} (Lockout voltage), is Read Array mode.

After any word/byte write or block erase operation is complete and even after V_{PP} transitions down to V_{PPL}, the CUI must be reset to Read Array mode via the Read Array command when accesses to the flash memory are desired.

ABSOLUTE MAXIMUM RATINGS*

Commercial Operating Temperature
During Read .... 0°C to 70°C[1]
During Block Erase
and Word/Byte Write .... 0°C to 70°C
Temperature Under Bias .... −10°C to +80°C

Extended Operating Temperature
During Read .... −40°C to +85°C
During Block Erase
and Word/Byte Write .... −40°C to +85°C
Temperature Under Bias .... −40°C to +85°C

Storage Temperature .... −65°C to +125°C

Voltage on Any Pin
(except V_{CC} and V_{PP})
with Respect to GND .... −2.0V to +7.0V[2]

Voltage on Pin $\overline{PWD}$ or Pin A_9
with Respect to GND ..... −2.0V to +13.5V[2, 3]

V_{PP} Program Voltage with Respect
to GND during Block Erase
and Word/Byte Write ..... −2.0V to +14.0V[2, 3]

V_{CC} Supply Voltage
with Respect to GND .... −2.0V to +7.0V[2]

Output Short Circuit Current............. 100 mA[4]

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

***WARNING:** *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC voltage is −0.5V on input/output pins. During transitions, this level may undershoot to −2.0V for periods <20 ns.
Maximum DC voltage on input/output pins is V_{CC} + 0.5V which, during transitions, may overshoot to V_{CC} + 2.0V for periods <20 ns.
3. Maximum DC voltage on A_9 or V_{PP} may overshoot to +14.0V for periods <20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.
5. 10% V_{CC} specifications reference the 28F400BX-60/28F004BX-60 in their standard test configuration, and the 28F400BX-80/28F004BX-80.
6. 5% V_{CC} specifications reference the 28F400BX-60/28F004BX-60 in their high speed test configuration.

OPERATING CONDITIONS

Symbol	Parameter	Notes	Min	Max	Units
T_A	Operating Temperature		0	70	°C
V_{CC}	V_{CC} Supply Voltage (10%)	5	4.50	5.50	V
V_{CC}	V_{CC} Supply Voltage (5%)	6	4.75	5.25	V

DC CHARACTERISTICS

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{LI}	Input Load Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND

3

DC CHARACTERISTICS (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{CCS}	V_{CC} Standby Current	1, 3			1.5	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{IH}$
					100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{CC} \pm 0.2V$ 28F200BX: $\overline{BYTE} = V_{CC} \pm 0.2V$ or GND
I_{CCD}	V_{CC} Deep Powerdown Current	1		0.20	1.2	μA	$\overline{PWD}$ = GND $\pm 0.2V$
I_{CCR}	V_{CC} Read Current for 28F200BX Word-Wide Mode	1, 5, 6			60	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 10 MHz, I_{OUT} = 0 mA CMOS Inputs
					65	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 10 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCR}	V_{CC} Read Current for 28F200BX Byte-Wide Mode and 28F004BX	1, 5, 6			55	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 10 MHz, I_{OUT} = 0 mA CMOS Inputs
					60	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 10 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCW}	V_{CC} Word Write Current	1			70	mA	Word Write in Progress
I_{CCW}	V_{CC} Byte Write Current	1			60	mA	Byte Write in Progress
I_{CCE}	V_{CC} Block Erase Current	1			30	mA	Block Erase in Progress
I_{CCES}	V_{CC} Erase Suspend Current	1, 2		5	10	mA	Block Erase Suspended, $\overline{CE} = V_{IH}$
I_{PPS}	V_{PP} Standby Current	1			± 10	μA	$V_{PP} \leq V_{CC}$
I_{PPD}	V_{PP} Deep PowerDown Current	1			5.0	μA	$\overline{PWD}$ = GND $\pm 0.2V$
I_{PPR}	V_{PP} Read Current	1			200	μA	$V_{PP} > V_{CC}$
I_{PPW}	V_{PP} Word Write Current	1			40	mA	$V_{PP} = V_{PPH}$ Word Write in Progress
I_{PPW}	V_{PP} Byte Write Current	1			30	mA	$V_{PP} = V_{PPH}$ Byte Write in Progress
I_{PPE}	V_{PP} Block Erase Current	1			30	mA	$V_{PP} = V_{PPH}$ Block Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1			200	μA	$V_{PP} = V_{PPH}$ Block Erase Suspended
I_{ID}	A_9 Intelligent Identifier Current	1			500	μA	$A_9 = V_{ID}$
V_{ID}	A_9 Intelligent Identifier Voltage		11.5		13.0	V	
V_{IL}	Input Low Voltage		-0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage				0.45	V	$V_{CC} = V_{CC}$ Min I_{OL} = 5.8 mA

DC CHARACTERISTICS (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
V_{OH}	Output High Voltage		2.4			V	$V_{CC} = V_{CC}$ Min $I_{OH} = -2.5$ mA
V_{PPL}	V_{PP} during Normal Operations	3	0.0		6.5	V	
V_{PPH}	V_{PP} during Erase/Write Operations	7	11.4	12.0	12.6	V	
V_{PPH}	V_{PP} during Erase/Write Operations	8	10.8	12.0	13.2	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.0			V	
V_{HH}	$\overline{PWD}$ Unlock Voltage		11.5		13.0	V	Boot Block Write/Erase

EXTENDED TEMPERATURE OPERATING CONDITIONS

Symbol	Parameter	Notes	Min	Max	Unit
T_A	Operating Temperature		−40	85	°C
V_{CC}	V_{CC} Supply Voltage (10%)	5	4.50	5.50	V

3

DC CHARACTERISTICS: EXTENDED TEMPERATURE OPERATION

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{LI}	Input Load Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND
I_{CCS}	V_{CC} Standby Current	1, 3			1.5	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{IH}$
					100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{CC} \pm 0.2V$ 28F400BX: $\overline{BYTE} = V_{CC} \pm 0.2V$ or GND
I_{CCD}	V_{CC} Deep Power-Down Current	1		0.20	8	μA	$\overline{PWD}$ = GND ±2V

DC CHARACTERISTICS: EXTENDED TEMPERATURE OPERATION (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{CCR}	V_{CC} Read Current for 28F400BX Word-Wide Mode	1, 5, 6			70	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 10 MHz, I_{OUT} = 0 mA CMOS Inputs
					75	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 10 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCR}	V_{CC} Read Current for 28F400BX Byte-Wide Mode and 28F004BX	1, 5, 6			65	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 10 MHz, I_{OUT} = 0 mA CMOS Inputs
					70	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 10 MHz, I_{OUT} = 0 mA TTL Inputs
I_{CCW}	V_{CC} Word Write Current	1			75	mA	Word Write in Progress
I_{CCW}	V_{CC} Byte Write Current	1			65	mA	Byte Write in Progress
I_{CCE}	V_{CC} Block Erase Current	1			40	mA	Block Erase in Progress
I_{CCES}	V_{CC} Erase Suspend Current	1, 2		5	10	mA	Block Erase Suspended, $\overline{CE} = V_{IH}$
I_{PPS}	V_{PP} Standby Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PPD}	V_{CC} Deep Power-Down Current	1			5.0	μA	$\overline{PWD}$ = GND ±0.2V
I_{PPR}	V_{PP} Read Current	1			200	μA	$V_{PP} > V_{CC}$
I_{PPW}	V_{PP} Word Write Current	1			40	mA	$V_{PP} = V_{PPH}$ Word Write in Progress
I_{PPW}	V_{PP} Byte Write Current	1			40	mA	$V_{PP} = V_{PPH}$ Byte Write in Progress

DC CHARACTERISTICS: EXTENDED TEMPERATURE OPERATION (Continued)

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{PPE}	V_{PP} Block Erase Current	1			30	mA	$V_{PP} = V_{PPH}$ Block Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1			200	μA	$V_{PP} = V_{PPH}$ Block Erase Suspended
I_{ID}	A_9 Intelligent Identifier Current	1			500	μA	$A_9 = V_{ID}$
V_{ID}	A_9 Intelligent Identifier Current		11.5		13.0	V	
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	$V_{CC} = V_{CC}$ Min I_{OL} = 5.8 mA
V_{OH}	Output High Voltage		2.4			V	$V_{CC} = V_{CC}$ Min I_{OH} = −2.5 mA
V_{PPL}	V_{PP} during Normal Operations	3	0.0		6.5	V	
V_{PPH}	V_{PP} during Erase/Write Operations	7	11.4	12.0	12.6	V	
V_{PPH}	V_{PP} during Erase/Write Operations	8	10.8	12.0	13.2	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.0			V	
V_{HH}	$\overline{PWD}$ Unlock Voltage		11.5		13.0	V	Boot Block Write/Erase

3

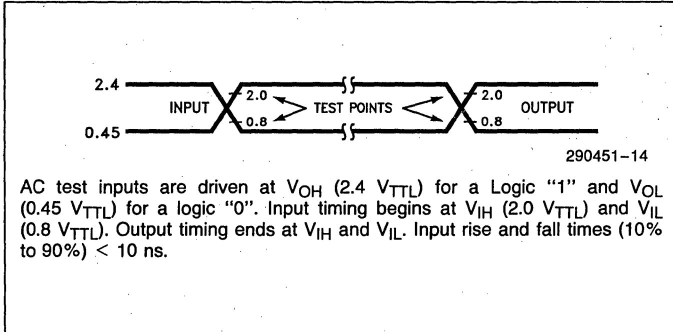
CAPACITANCE(4) T_A = 25°C, f = 1 MHz

Symbol	Parameter	Typ	Max	Unit	Condition
C_{IN}	Input Capacitance	6	8	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	10	12	pF	V_{OUT} = 0V

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C. These currents are valid for all product versions (packages and speeds).
2. I_{CCES} is specified with the device deselected. If the device is read while in Erase Suspend Mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Block Erases and Word/Byte Writes are inhibited when $V_{PP} = V_{PPL}$ and not guaranteed in the range between V_{PPH} and V_{PPL}.
4. Sampled, not 100% tested.
5. Automatic Power Savings (APS) reduces I_{CCR} to less than 1 mA typical in static operation.
6. CMOS Inputs are either V_{CC} ±0.2V or GND ±0.2V. TTL Inputs are either V_{IL} or V_{IH}.
7. V_{PP} = 12.0V ± 5% for applications requiring 100,000 block erase cycles.
8. V_{PP} = 12.0V ± 10% for applications requiring wider V_{PP} tolerances at 100 block erase cycles.
9. For the 28F004BX address pin A_{10} follows the C_{OUT} capacitance numbers.

STANDARD TEST CONFIGURATION(1)

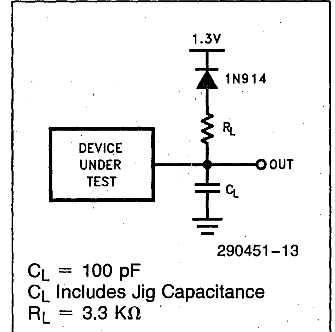
STANDARD AC INPUT/OUTPUT REFERENCE WAVEFORM



290451-14

AC test inputs are driven at V_{OH} (2.4 V_{TTL}) for a Logic "1" and V_{OL} (0.45 V_{TTL}) for a logic "0". Input timing begins at V_{IH} (2.0 V_{TTL}) and V_{IL} (0.8 V_{TTL}). Output timing ends at V_{IH} and V_{IL}. Input rise and fall times (10% to 90%) < 10 ns.

STANDARD AC TESTING LOAD CIRCUIT

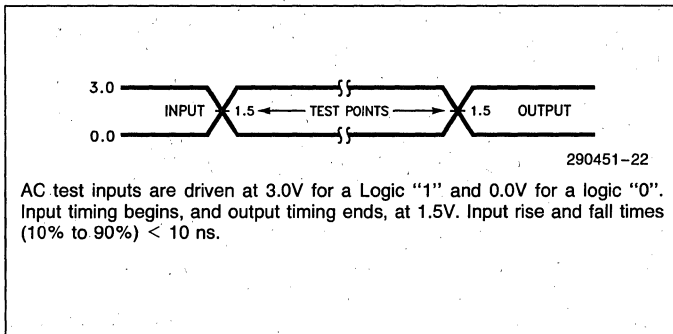


290451-13

C_L = 100 pF
C_L Includes Jig Capacitance
R_L = 3.3 KΩ

HIGH SPEED TEST CONFIGURATION(2)

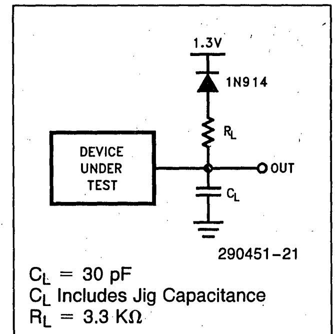
HIGH SPEED AC INPUT/OUTPUT REFERENCE WAVEFORM



290451-22

AC test inputs are driven at 3.0V for a Logic "1" and 0.0V for a logic "0". Input timing begins, and output timing ends, at 1.5V. Input rise and fall times (10% to 90%) < 10 ns.

HIGH SPEED AC TESTING LOAD CIRCUIT



290451-21

C_L = 30 pF
C_L Includes Jig Capacitance
R_L = 3.3 KΩ

NOTES:
1. Testing characteristics for 28F400BX-60/28F004BX-60 in standard test configuration and 28F400BX-80/28F004BX-80.
2. Testing characteristics for 28F400BX-60/28F004BX-60 in high speed test configuration.

AC CHARACTERISTICS—Read Only Operations(1)

Versions		$V_{CC} \pm 5\%$		28F400BX-60(4) 28F004BX-60(4)						Unit
		$V_{CC} \pm 10\%$				28F400BX-60(5) 28F004BX-60(5)		28F400BX-80(5) 28F004BX-80(5)		
Symbol		**Parameter**	**Notes**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{AVAV}	t_{RC}	Read Cycle Time		60		70		80		ns
t_{AVQV}	t_{ACC}	Address to Output Delay			60		70		80	ns
t_{ELQV}	t_{CE}	$\overline{CE}$ to Output Delay	2		60		70		80	ns
t_{PHQV}	t_{PWH}	$\overline{PWD}$ High to Output Delay			300		300		300	ns
t_{GLQV}	t_{OE}	$\overline{OE}$ to Output Delay	2		30		35		40	ns
t_{ELQX}	t_{LZ}	$\overline{CE}$ to Output Low Z	3	0		0		0		ns
t_{EHQZ}	t_{HZ}	$\overline{CE}$ High to Output High Z	3		20		25		30	ns
t_{GLQX}	t_{OLZ}	$\overline{OE}$ to Output Low Z	3	0		0		0		ns
t_{GHQZ}	t_{DF}	$\overline{OE}$ High to Output High Z	3		20		25		30	ns
	t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change, Whichever is First	3	0		0		0		ns
t_{ELFL} t_{ELFH}		$\overline{CE}$ to $\overline{BYTE}$ Switching Low or High	3		5		5		5	ns
t_{FHQV}		$\overline{BYTE}$ Switching High to Valid Output Delay	3, 6		60		70		80	ns
t_{FLQZ}		$\overline{BYTE}$ Switching Low to Output High Z	3		20		25		30	ns

NOTES:

1. See AC Input/Output Reference Waveform for timing measurements.
2. $\overline{OE}$ may be delayed up to $t_{CE}-t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.
3. Sampled, not 100% tested.
4. See High Speed Test Configuration.
5. See Standard Test Configuration.
6. t_{FLQV}, $\overline{BYTE}$ switching low to valid output delay, will be equal to t_{AVQV}, measured from the time DQ_{15}/A_{-1} becomes valid.

3

EXTENDED TEMPERATURE OPERATION
AC CHARACTERISTICS—Read Only Operations(1)

Versions				T28F400BX-90(4, 5) T28F004BX-80		Unit
Symbol		Parameter	Notes	Min	Max	
t_{AVAV}	t_{RC}	Read Cycle Time		90		ns
t_{AVQV}	t_{ACC}	Address to Output Delay			90	ns
t_{ELQV}	t_{CE}	$\overline{CE}$ to Output Delay			90	ns
t_{PHQV}	t_{PWH}	$\overline{PWD}$ High to Output Delay			300	ns
t_{GLQV}	t_{OE}	$\overline{OE}$ to Output Delay	2		45	ns
t_{ELQX}	t_{LZ}	$\overline{CE}$ to Output Low Z		0		ns
t_{EHQZ}	t_{HZ}	$\overline{CE}$ High to Output High Z			35	ns
t_{GLQX}	t_{OLZ}	$\overline{OE}$ to Output Low Z	3	0		ns
t_{GHQZ}	t_{DF}	$\overline{OE}$ High to Output High Z	3		35	ns
	t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change, Whichever is First	3	0		ns
t_{ELFL} t_{ELFH}		$\overline{CE}$ to $\overline{BYTE}$ Switching Low or High	3		5	ns
t_{FHQV}		$\overline{BYTE}$ Switching High to Valid Output Delay	3, 5		90	ns
t_{FLQZ}		$\overline{BYTE}$ Switching Low to Output High Z	3		35	ns

NOTES:
1. See AC Input/Output Reference Waveform for timing measurements.
2. $\overline{OE}$ may be delayed up to $t_{CE}-t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.
3. Sampled, not 100% tested.
4. See Standard Test Configuration.
5. t_{FLQV}, $\overline{BYTE}$ switching low to valid output delay, will be equal to t_{AVQV} from the time DQ_{15}/A_{-1} becomes valid.

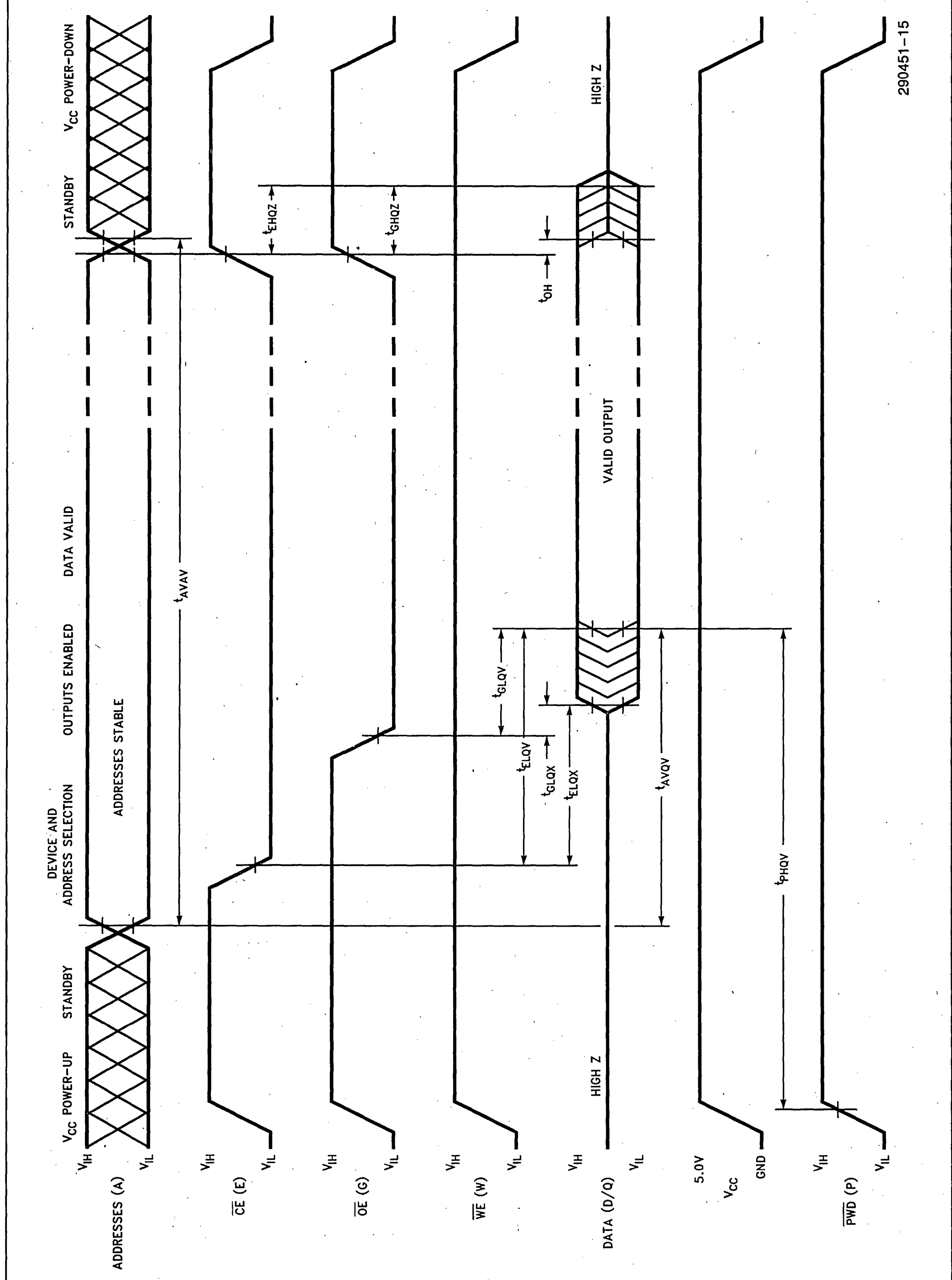


Figure 16. A.C. Waveforms for Read Operations

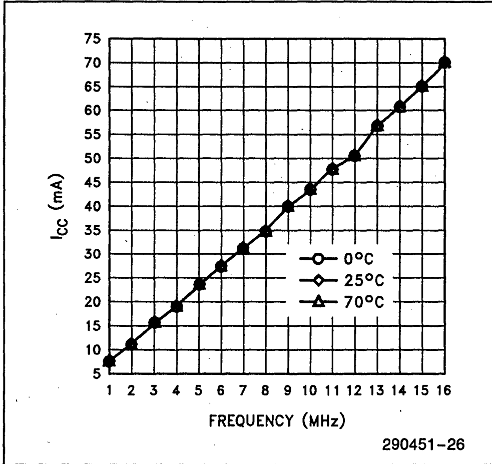


Figure 17. I_{CC} (RMS) vs Frequency (V_{CC} = 5.5V for x16 Operation

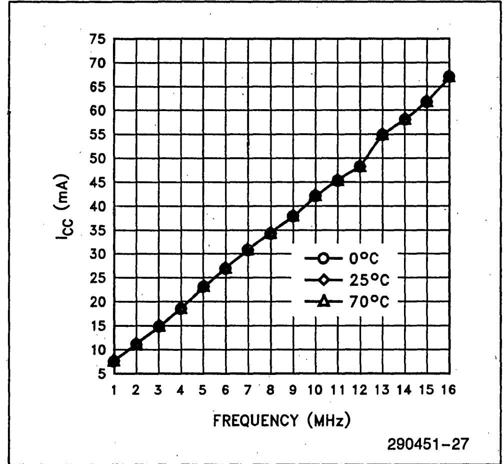


Figure 18. I_{CC} (RMS) vs Frequency (V_{CC} = 5.5V) for x8 Operation

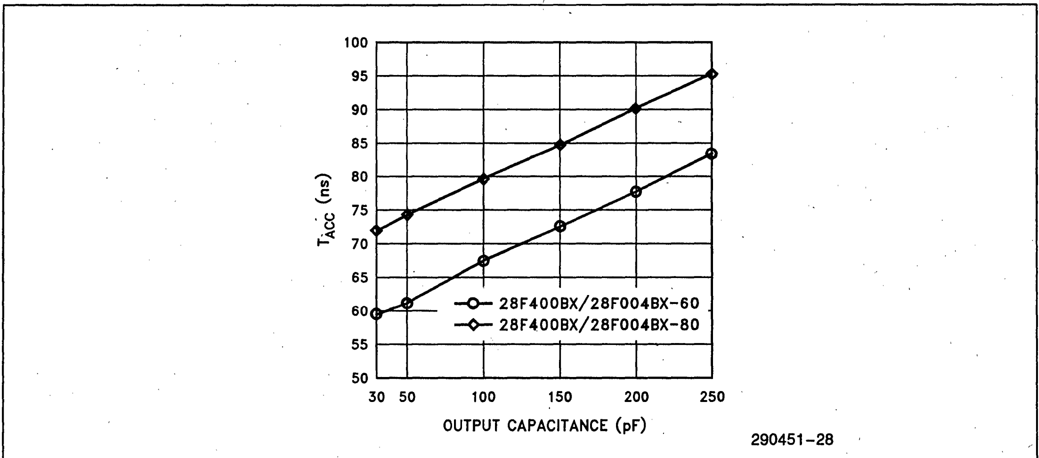


Figure 19. T_{ACC} vs Output Load Capacitance

STANDBY
DEVICE ADDRESS SELECTION
DATA VALID
STANDBY

V_{IH}
ADDRESSES (A)
DQ_8-DQ_{14}
ADDRESSES STABLE
t_{AVAV}

V_{IH}
$\overline{CE}$ (C)
V_{IL}
$t_{AVPL} = t_{ELFL}$
t_{EHQZ}

V_{IH}
$\overline{OE}$ (G)
V_{IL}
t_{ELFL}
t_{GHQZ}

V_{IH}
$\overline{BYTE}$ (F)
V_{IL}
t_{GLQV}
t_{ELQV}
t_{GLQX}
t_{ELQX}
t_{OH}

V_{IH}
DATA (DQ_0-DQ_7)
V_{IL}
HIGH Z
DATA OUTPUT ON DQ_0-DQ_7
DATA OUTPUT ON DQ_0-DQ_7
HIGH Z
t_{AVQV}

DATA (DQ_8-DQ_{14})
DATA OUTPUT ON DQ_8-DQ_{14}
HIGH Z
t_{FLQZ}
t_{AVQV}

DQ_{15}/A_{-1}
DATA OUTPUT ON DQ_{15}
HIGH Z
ADDRESS INPUT
HIGH Z

290451-29

Figure 20. $\overline{BYTE}$ Timing Diagram for Both Read and Write Operations for 28F400BX

AC CHARACTERISTICS—$\overline{WE}$ Controlled Write Operations(1)

Versions		$V_{CC} \pm 5\%$		28F400BX-60(9) 28F004BX-60(9)						Unit
		$V_{CC} \pm 10\%$				28F400BX-60(10) 28F004BX-60(10)		28F400BX-80(10) 28F004BX-80(10)		
Symbol		Parameter	Notes	Min	Max	Min	Max	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		60		70		80		ns
t_{PHWL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{WE}$ Going Low		215		215		215		ns
t_{ELWL}	t_{CS}	$\overline{CE}$ Setup to $\overline{WE}$ Going Low		0		0		0		ns
t_{PHHWH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{WE}$ Going High	6, 8	100		100		100		ns
t_{VPWH}	t_{VPS}	V_{PP} Setup to $\overline{WE}$ Going High	5, 8	100		100		100		ns
t_{AVWH}	t_{AS}	Address Setup to $\overline{WE}$ Going High	3	50		50		50		ns
t_{DVWH}	t_{DS}	Data Setup to $\overline{WE}$ Going High	4	50		50		50		ns
t_{WLWH}	t_{WP}	$\overline{WE}$ Pulse Width		50		50		50		ns
t_{WHDX}	t_{DH}	Data Hold from $\overline{WE}$ High	4	0		0		0		ns
t_{WHAX}	t_{AH}	Address Hold from $\overline{WE}$ High	3	10		10		10		ns
t_{WHEH}	t_{CH}	$\overline{CE}$ Hold from $\overline{WE}$ High		10		10		10		ns
t_{WHWL}	t_{WPH}	$\overline{WE}$ Pulse Width High		10		20		30		ns
t_{WHQV1}		Duration of Word/Byte Programming Operation	2, 5	6		6		6		µs
t_{WHQV2}		Duration of Erase Operation (Boot)	2, 5, 6	0.3		0.3		0.3		s
t_{WHQV3}		Duration of Erase Operation (Parameter)	2, 5	0.3		0.3		0.3		s
t_{WHQV4}		Duration of Erase Operation (Main)	2, 5	0.6		0.6		0.6		s
t_{QWL}	t_{VPH}	V_{PP} Hold from Valid SRD	5, 8	0		0		0		ns
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	6, 8	0		0		0		ns
t_{PHBR}		Boot-Block Relock Delay	7, 8		100		100		100	ns

AC CHARACTERISTICS—$\overline{WE}$ Controlled Write Operations(1) (Continued)

NOTES:
1. Read timing characteristics during write and erase operations are the same as during read-only operations. Refer to AC characteristics during Read Mode.
2. The on-chip WSM completely automates program/erase operations; program/erase algorithms are now controlled internally which includes verify and margining operations.
3. Refer to command definition table for valid A_{IN}.
4. Refer to command definition table for valid D_{IN}.
5. Program/Erase durations are measured to valid SRD data (successful operation, SR.7 = 1).
6. For Boot Block Program/Erase, $\overline{PWD}$ should be held at V_{HH} until operation completes successfully.
7. Time t_{PHBR} is required for successful relocking of the Boot Block.
8. Sampled but not 100% tested.
9. See High Speed Test Configuration.
10. See Standard Test Configuration.

BLOCK ERASE AND WORD/BYTE WRITE PERFORMANCE V_{PP} = 12.0V ±5%

Parameter	Notes	28F400BX-60 28F004BX-60			28F400BX-80 28F004BX-80			Unit
		Min	Typ(1)	Max	Min	Typ(1)	Max	
Boot/Parameter Block Erase Time	2		1.0	7		1.0	7	s
Main Block Erase Time	2		2.4	14		2.4	14	s
Main Block Byte Program Time	2		1.2	4.2		1.2	4.2	s
Main Block Word Program Time	2		0.6	2.1		0.6	2.1	s

NOTES:
1. 25°C
2. Excludes System-Level Overhead.

BLOCK ERASE AND WORD/BYTE WRITE PERFORMANCE V_{PP} = 12.0V ±10%

Parameter	Notes	28F400BX-60 28F004BX-60			28F400BX-80 28F004BX-80			Unit
		Min	Typ(1)	Max	Min	Typ(1)	Max	
Boot/Parameter Block Erase Time	2		5.8	40		5.8	40	s
Main Block Erase Time	2		14	60		14	60	s
Main Block Byte Program Time	2		6.0	20		6.0	20	s
Main Block Word Program Time	2		3.0	10		3.0	10	s

NOTES:
1. 25°C
2. Excludes System-Level Overhead.

EXTENDED TEMPERATURE OPERATION
AC CHARACTERISTICS—$\overline{WE}$ Controlled Write Operations(1)

Versions(4)				T28F400BX-90(9) T28F004BX-90(9)		Unit
Symbol		Parameter	Notes	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		90		ns
t_{PHWL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{WE}$ Going Low		210		ns
t_{ELWL}	t_{CS}	$\overline{CE}$ Setup to $\overline{WE}$ Going Low		0		ns
t_{PHHWH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{WE}$ Going High	6, 8	100		ns
t_{VPWH}	t_{VPS}	V_{PP} Setup to $\overline{WE}$ Going High	5, 8	100		ns
t_{AVWH}	t_{AS}	Address Setup to $\overline{WE}$ Going High	3	60		ns
t_{DVWH}	t_{DS}	Data Setup to $\overline{WE}$ Going High	4	60		ns
t_{WLWH}	t_{WP}	$\overline{WE}$ Pulse Width		60		ns
t_{WHDX}	t_{DH}	Data Hold from $\overline{WE}$ High	4	0		ns
t_{WHAX}	t_{AH}	Address Hold from $\overline{WE}$ High	3	10		ns
t_{WHEH}	t_{CH}	$\overline{CE}$ Hold from $\overline{WE}$ High		10		ns
t_{WHWL}	t_{WPH}	$\overline{WE}$ Pulse Width High		30		ns
t_{WHQV1}		Duration of Word/Byte Programming Operation	2, 5	7		µs
t_{WHQV2}		Duration of Erase Operation (Boot)	2, 5, 6	0.4		s
t_{WHQV3}		Duration of Erase Operation (Parameter)	2, 5	0.4		s
t_{WHQV4}		Duration of Erase Operation (Main)	2, 5	0.7		s
t_{QWL}	t_{VPH}	V_{PP} Hold from Valid SRD	5, 8	0		ns
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	6, 8	0		ns
t_{PHBR}		Boot-Block Relock Delay	7, 8		100	ns

EXTENDED TEMPERATURE OPERATION AC CHARACTERISTICS—$\overline{WE}$ Controlled Write Operations(1) (Continued)

NOTES:
1. Read timing characteristics during write and erase operations are the same as during read-only operations. Refer to AC characteristics during Read Mode.
2. The on-chip WSM completely automates program/erase operations; program/erase algorithms are now controlled internally which includes verify and margining operations.
3. Refer to command definition table for valid A_{IN}.
4. Refer to command definition table for valid D_{IN}.
5. Program/Erase durations are measured to valid SRD data (successful operation, SR.7 = 1).
6. For Boot Block Program/Erase, $\overline{PWD}$ should be held at V_{HH} until operation completes successfully.
7. Time t_{PHBR} is required for successful relocking of the Boot Block.
8. Sampled but not 100% tested.
9. See Standard Test Configuration.

EXTENDED TEMPERATURE OPERATION BLOCK ERASE AND WORD/BYTE WRITE PERFORMANCE V_{PP} = 12.0V ±5%

Parameter	Notes	T28F400BX-90 T28F004BX-90			Unit
		Min	Typ(1)	Max	
Boot/Parameter Block Erase Time	2		1.5	10.5	s
Main Block Erase Time	2		3.0	18	s
Main Block Byte Program Time	2		1.4	5.0	s
Main Block Word Program Time	2		0.7	2.5	s

NOTES:
1. 25°C
2. Excludes System-Level Overhead.

3

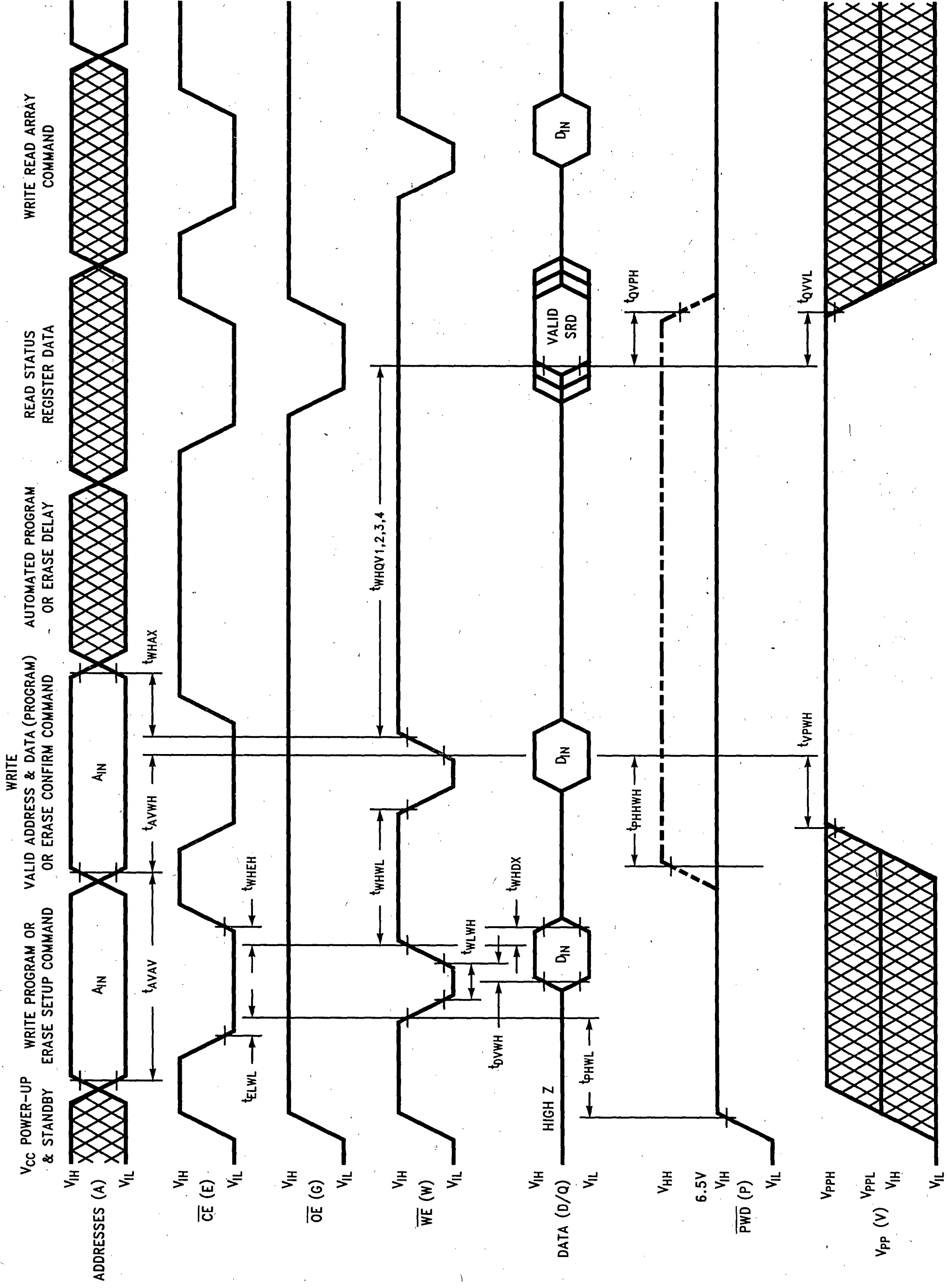


Figure 21. AC Waveforms for a Write and Erase Operations (WE-Controlled Writes)

AC CHARACTERISTICS—$\overline{CE}$-CONTROLLED WRITE OPERATIONS(1, 9)

Versions		V_{CC} ± 5%		28F400BX-60(10) 28F004BX-60(10)						Unit
		V_{CC} ±10%				28F400BX-60(11) 28F004BX-60(11)		28F400BX-80(11) 28F004BX-80(11)		
Symbol		Parameter	Notes	Min	Max	Min	Max	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		60		70		80		ns
t_{PHEL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{CE}$ Going Low		215		215		215		ns
t_{WLEL}	t_{WS}	$\overline{WE}$ Setup to $\overline{CE}$ Going Low		0		0		0		ns
t_{PHHEH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{CE}$ Going High	6, 8	100		100		100		ns
t_{VPEH}	t_{VPS}	V_{PP} Setup to $\overline{CE}$ Going High	5, 8	100		100		100		ns
t_{AVEH}	t_{AS}	Address Setup to $\overline{CE}$ Going High	3	50		50		50		ns
t_{DVEH}	t_{DS}	Data Setup to $\overline{CE}$ Going High	4	50		50		50		ns
t_{ELEH}	t_{CP}	$\overline{CE}$ Pulse Width		50		50		50		ns
t_{EHDX}	t_{DH}	Data Hold from $\overline{CE}$ High	4	0		0		0		ns
t_{EHAX}	t_{AH}	Address Hold from $\overline{CE}$ High	3	10		10		10		ns
t_{EHWH}	t_{WH}	$\overline{WE}$ Hold from $\overline{CE}$ High		10		10		10		ns
t_{EHEL}	t_{CPH}	$\overline{CE}$ Pulse Width High		10		20		30		ns
t_{EHQV1}		Duration of Word/Byte Programming Operation	2, 5	6		6		6		µs
t_{EHQV2}		Duration of Erase Operation (Boot)	2, 5, 6	0.3		0.3		0.3		s
t_{EHQV3}		Duration of Erase Operation (Parameter)	2, 5	0.3		0.3		0.3		s
t_{EHQV4}		Duration of Erase Operation (Main)	2, 5	0.6		0.6		0.6		s
t_{QWL}	t_{VPH}	V_{PP} Hold from Valid SRD	5, 8	0		0		0		ns
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	6, 8	0		0		0		ns
t_{PHBR}		Boot-Block Relock Delay	7		100		100		100	ns

AC CHARACTERISTICS—$\overline{CE}$-CONTROLLED WRITE OPERATIONS(1, 9) (Continued)

NOTES:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of $\overline{CE}$ and $\overline{WE}$ in systems where $\overline{CE}$ defines the write pulse-width (within a longer $\overline{WE}$ timing waveform), all set-up, hold and inactive $\overline{WE}$ times should be measured relative to the $\overline{CE}$ waveform.
2, 3, 4, 5, 6, 7, 8: Refer to AC Characteristics notes for $\overline{WE}$-Controlled Write Operations.
9. Read timing characteristics during write and erase operations are the same as during read-only operations. Refer to AC Characteristics during Read Mode.
10. See High Speed Test Configuration.
11. See Standard Test Configuration.

EXTENDED TEMPERATURE OPERATION
AC CHARACTERISTICS—$\overline{CE}$-CONTROLLED WRITE OPERATIONS(1, 9)

Versions				T28F400BX-90(10) T28F004BX-90(10)		Unit
Symbol		Parameter	Notes	Min	Max	
t_{AVAV}	t_{WC}	Write Cycle Time		90		ns
t_{PHEL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{CE}$ Going Low		210		ns
t_{WLEL}	t_{WS}	$\overline{WE}$ Setup to $\overline{CE}$ Going Low		0		ns
t_{PHHEH}	t_{PHS}	$\overline{PWD}$ V_{HH} Setup to $\overline{CE}$ Going High	6, 8	100		ns
t_{VPEH}	t_{VPS}	V_{PP} Setup to $\overline{CE}$ Going High	5, 8	100		ns
t_{AVEH}	t_{AS}	Address Setup to $\overline{CE}$ Going High	3	60		ns
t_{DVEH}	t_{DS}	Data Setup to $\overline{CE}$ Going High	4	60		ns
t_{ELEH}	t_{CP}	$\overline{CE}$ Pulse Width		60		ns
t_{EHDX}	t_{DH}	Data Hold from $\overline{CE}$ High	4	0		ns
t_{EHAX}	t_{AH}	Address Hold from $\overline{CE}$ High	3	10		ns
t_{EHWH}	t_{WH}	$\overline{WE}$ Hold from $\overline{CE}$ High		10		ns
t_{EHEL}	t_{CPH}	$\overline{CE}$ Pulse Width High		30		ns
t_{EHQV1}		Duration of Word/Byte Programming Operation	2, 5	7		μs
t_{EHQV2}		Duration of Erase Operation (Boot)	2, 5, 6	0.4		s
t_{EHQV3}		Duration of Erase Operation (Parameter)	2, 5	0.4		s
t_{EHQV4}		Duration of Erase Operation (Main)	2, 5	0.7		s
t_{QWL}	t_{VPH}	V_{PP} Hold from Valid SRD	5, 8	0		ns
t_{QVPH}	t_{PHH}	$\overline{PWD}$ V_{HH} Hold from Valid SRD	6, 8	0		ns
t_{PHBR}		Boot-Block Relock Delay	7		100	ns

NOTES:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of $\overline{CE}$ and $\overline{WE}$ in systems where $\overline{CE}$ defines the write pulse-width (within a longer $\overline{WE}$ timing waveform), all set-up, hold and inactive $\overline{WE}$ times should be measured relative to the $\overline{CE}$ waveform.
2, 3, 4, 5, 6, 7, 8: Refer to AC Characteristics for $\overline{WE}$-Controlled Write Operations.
9. Read timing characteristics during write and erase operations are the same as during read-only operations. Refer to AC Characteristics during Read Mode.
10. See Standard Test Configuration.

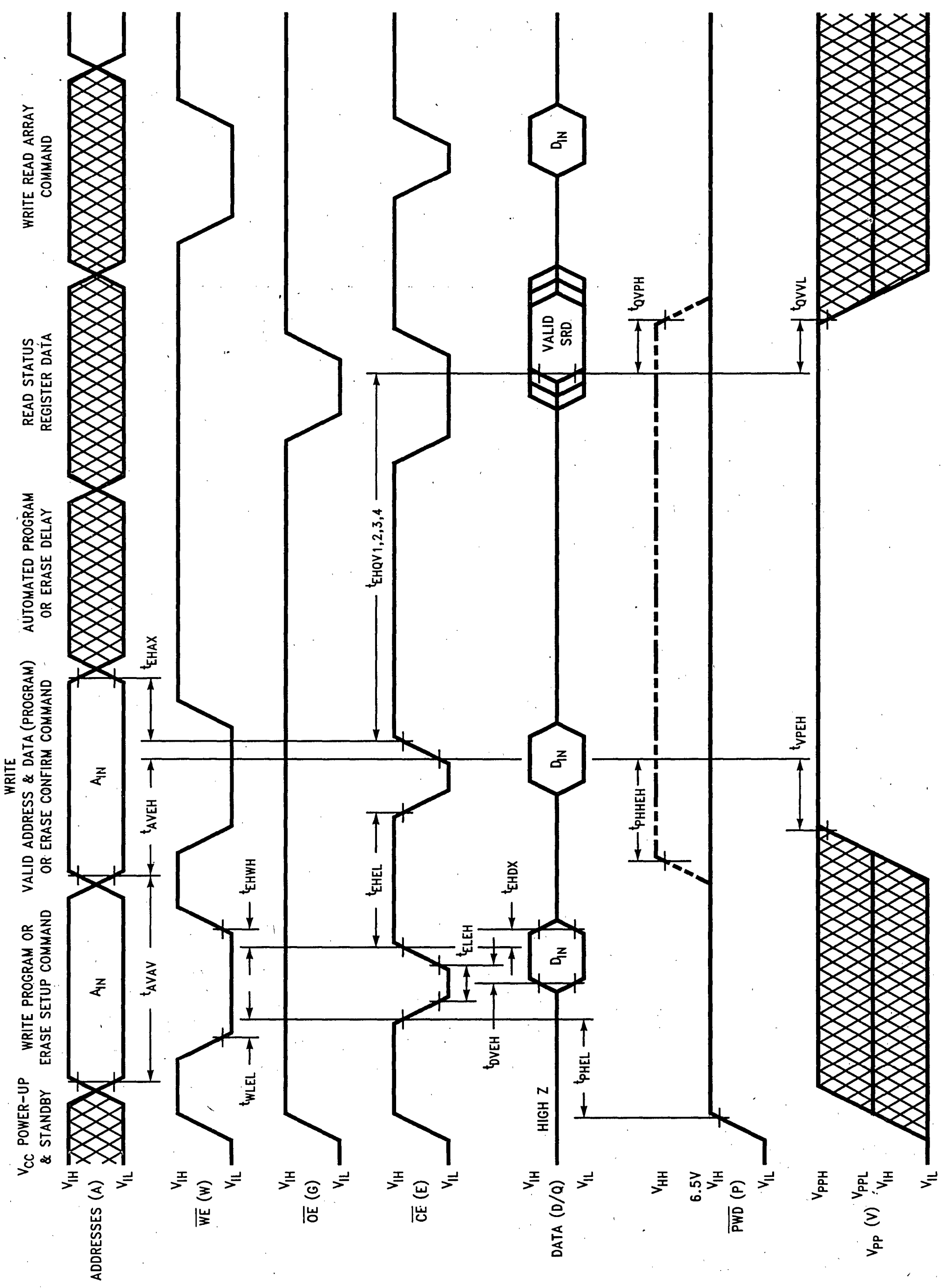


Figure 22. Alternate A.C. Waveforms for Write and Erase Operations (CE-Controlled Writes)

ORDERING INFORMATION

	E	2	8	F	4	0	0	B	X	-	6	0

OPERATING TEMPERATURE
T = EXTENDED TEMP
BLANK = COMMERCIAL TEMP

PACKAGE
E = STANDARD 56 LEAD TSOP
PA = 44 LEAD PSOP

ACCESS SPEED (ns)
60 ns
80 ns
90 ns

290451-18

VALID COMBINATIONS:

E28F400BX-60	PA28F400BX-60	TE28F400BX-T90	TPA28F400BX-T90
E28F400BX-80	PA28F400BX-80	TE28F400BX-B90	TPA28F400BX-B90

	E	2	8	F	0	0	4	B	X	-	6	0

OPERATING TEMPERATURE
T = EXTENDED TEMP
BLANK = COMMERCIAL TEMP

PACKAGE
E = STANDARD 56 LEAD TSOP
PA = 44 LEAD PSOP

ACCESS SPEED (ns)
60 ns
80 ns
90 ns

290451-30

VALID COMBINATIONS:

E28F004BX-60	TE28F004BX-T90
E28F004BX-80	TE28F004BX-B90

ADDITIONAL INFORMATION

	Order Number
28F200BX/28F002BX Datasheet	290448
28F200BXL/28F002BXL Datasheet	290449
28F400BXL/28F004BXL Datasheet	290450
AP-363 "Extended Flash BIOS Design for Portable Computers"	292098
ER-28 "ETOX-III Flash Memory Technology"	204012
ER-29 "The Intel 2/4-MBit Boot Block Flash Memory Family"	294013

REVISION HISTORY

Number	Description
-001	Original Version
-002	Removed -70 speed bin. Integrated -70 characteristics into -60 speed bin. Added Extended Temperature characteristics. Modified $\overline{BYTE}$ Timing Diagram.

28F008SA
8 MBIT (1 MBIT x 8) FLASH MEMORY

- **High-Density Symmetrically Blocked Architecture**
  - **Sixteen 64 KByte Blocks**
- **Extended Cycling Capability**
  - **100,000 Block Erase Cycles**
  - **1.6 Million Block Erase Cycles per Chip**
- **Automated Byte Write and Block Erase**
  - **Command User Interface**
  - **Status Register**
- **System Performance Enhancements**
  - **RY/$\overline{BY}$ Status Output**
  - **Erase Suspend Capability**
- **Deep-Powerdown Mode**
  - **0.20 μA I_{CC} Typical**
- **Very High-Performance Read**
  - **85 ns Maximum Access Time**
- **SRAM-Compatible Write Interface**
- **Hardware Data Protection Feature**
  - **Erase/Write Lockout during Power Transitions**
- **Industry Standard Packaging**
  - **40-Lead TSOP, 44-Lead PSOP**
- **ETOX™ III Nonvolatile Flash Technology**
  - **12V Byte Write/Block Erase**
- **Independent Software Vendor Support**
  - **Microsoft* Flash File System (FFS)**

Intel's 28F008SA 8 Mbit FlashFile™ Memory is the highest density nonvolatile read/write solution for solid state storage. The 28F008SA's extended cycling, symmetrically blocked architecture, fast access time, write automation and low power consumption provide a more reliable, lower power, lighter weight and higher performance alternative to traditional rotating disk technology. The 28F008SA brings new capabilities to portable computing. Application and operating system software stored in resident flash memory arrays provide instant-on, rapid execute-in-place and protection from obsolescence through in-system software updates. Resident software also extends system battery life and increases reliability by reducing disk drive accesses.

For high density data acquisition applications, the 28F008SA offers a more cost-effective and reliable alternative to SRAM and battery. Traditional high density embedded applications, such as telecommunications, can take advantage of the 28F008SA's nonvolatility, blocking and minimal system code requirements for flexible firmware and modular software designs.

The 28F008SA is offered in 40-lead TSOP (standard and reverse) and 44-lead PSOP packages. Pin assignments simplify board layout when integrating multiple devices in a flash memory array or subsystem. This device uses an integrated Command User Interface and state machine for simplified block erasure and byte write. The 28F008SA memory map consists of 16 separately erasable 64 Kbyte blocks.

Intel's 28F008SA employs advanced CMOS circuitry for systems requiring low power consumption and noise immunity. Its 85 ns access time provides superior performance when compared with magnetic storage media. A deep powerdown mode lowers power consumption to 1 μW typical thru V_{CC}, crucial in portable computing, handheld instrumentation and other low-power applications. The $\overline{PWD}$ power control input also provides absolute data protection during system powerup/down.

Manufactured on Intel's 0.8 micron ETOX process, the 28F008SA provides the highest levels of quality, reliability and cost-effectiveness.

*Microsoft is a trademark of Microsoft Corporation.

October 1992
Order Number: 290429-003

PRODUCT OVERVIEW

The 28F008SA is a high-performance **8 Mbit** (8,388,608 bit) memory organized as **1 Mbyte** (1,048,576 bytes) of 8 bits each. **Sixteen 64 KByte** (65,536 byte) **blocks** are included on the 28F008SA. A memory map is shown in Figure 6 of this specification. A block erase operation erases one of the sixteen blocks of memory in typically **1.6 seconds,** independent of the remaining blocks. Each block can be independently erased and written **100,000 cycles. Erase Suspend** mode allows system software to suspend block erase to read data or execute code from any other block of the 28F008SA.

The 28F008SA is available in the **40-lead TSOP** (Thin Small Outline Package, 1.2 mm thick) and **44-lead PSOP** (Plastic Small Outline) packages. Pinouts are shown in Figures 2 and 4 of this specification.

The **Command User Interface** serves as the interface between the microprocessor or microcontroller and the internal operation of the 28F008SA.

Byte Write and Block Erase Automation allow byte write and block erase operations to be executed using a two-write command sequence to the Command User Interface. The internal **Write State Machine** (WSM) automatically executes the algorithms and timings necessary for byte write and block erase operations, including verifications, thereby unburdening the microprocessor or microcontroller. Writing of memory data is performed in byte increments typically within **9 μs**, an 80% improvement over current flash memory products. **I_{PP} byte write and block erase currents** are **10 mA typical, 30 mA maximum. V_{PP} byte write and block erase voltage** is **11.4V to 12.6V.**

The **Status Register** indicates the status of the WSM and when the WSM successfully completes the desired byte write or block erase operation.

The **RY/$\overline{\text{BY}}$** output gives an additional indicator of WSM activity, providing capability for both hardware signal of status (versus software polling) and status masking (interrupt masking for background erase, for example). Status polling using RY/$\overline{\text{BY}}$ minimizes both CPU overhead and system power consumption. When low, RY/$\overline{\text{BY}}$ indicates that the WSM is performing a block erase or byte write operation. RY/$\overline{\text{BY}}$ high indicates that the WSM is ready for new commands, block erase is suspended or the device is in deep powerdown mode.

Maximum access time is **85 ns (t_{ACC})** over the commercial temperature range (0°C to +70°C) and over V_{CC} supply voltage range (4.5V to 5.5V and 4.75V to 5.25V). **I_{CC} active current** (CMOS Read) is **20 mA typical, 35 mA maximum at 8 MHz.**

When the $\overline{\text{CE}}$ and $\overline{\text{PWD}}$ pins are at V_{CC}, the **I_{CC} CMOS Standby** mode is enabled.

A **Deep Powerdown** mode is enabled when the $\overline{\text{PWD}}$ pin is at GND, minimizing power consumption and providing write protection. **I_{CC} current** in deep powerdown is **0.20 μA typical.** Reset time of 400 ns is required from $\overline{\text{PWD}}$ switching high until outputs are valid to read attempts. Equivalently, the device has a wake time of 1 μs from $\overline{\text{PWD}}$ high until writes to the Command User Interface are recognized by the 28F008SA. With $\overline{\text{PWD}}$ at GND, the WSM is reset and the Status Register is cleared.

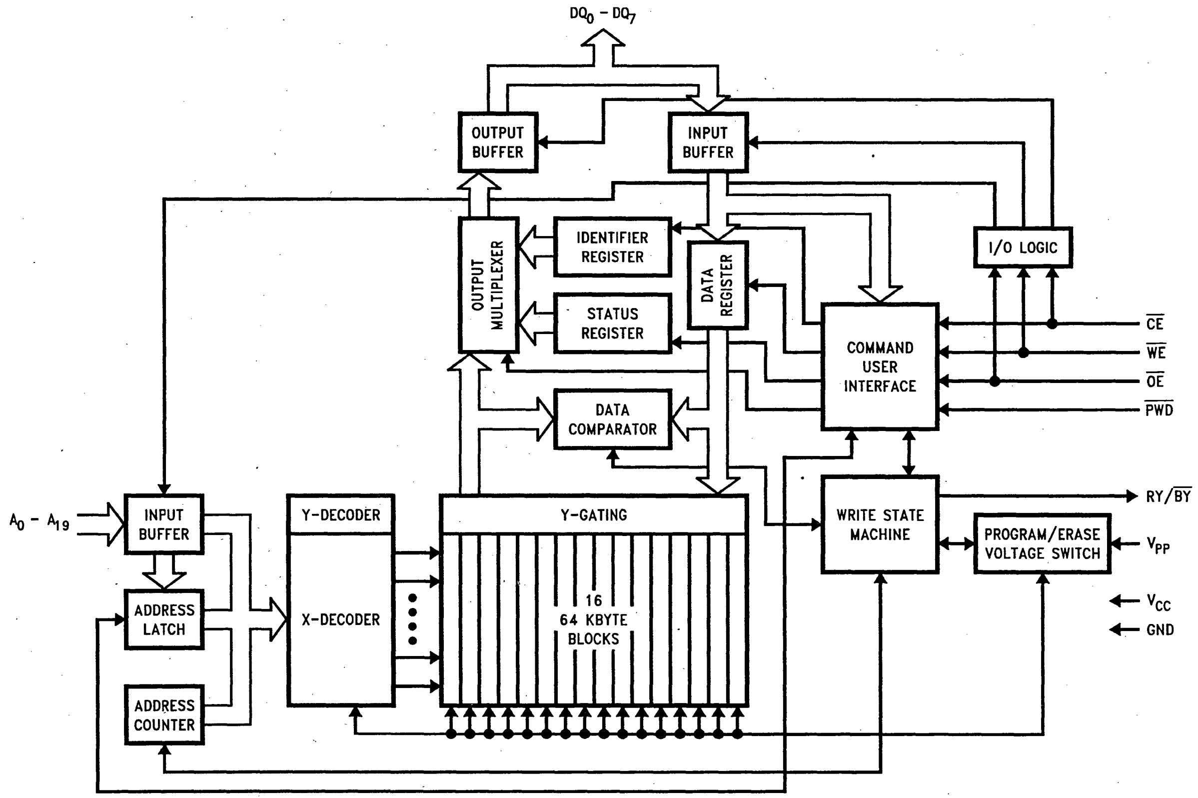


Figure 1. Block Diagram

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{19}	INPUT	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
DQ_0–DQ_7	INPUT/OUTPUT	**DATA INPUT/OUTPUTS:** Inputs data and commands during Command User Interface write cycles; outputs data during memory array, Status Register and Identifier read cycles. The data pins are active high and float to tri-state off when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}$	INPUT	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders, and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels.
$\overline{PWD}$	INPUT	**POWERDOWN:** Puts the device in deep powerdown mode. $\overline{PWD}$ is active low; $\overline{PWD}$ high gates normal operation. $\overline{PWD}$ also locks out block erase or byte write operations when active low, providing data protection during power transitions.
$\overline{OE}$	INPUT	**OUTPUT ENABLE:** Gates the device's outputs through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	INPUT	**WRITE ENABLE:** Controls writes to the Command User Interface and array blocks. $\overline{WE}$ is active low. Addresses and data are latched on the rising edge of the $\overline{WE}$ pulse.
RY/$\overline{BY}$	OUTPUT	**READY/$\overline{BUSY}$:** Indicates the status of the internal Write State Machine. When low, it indicates that the WSM is performing a block erase or byte write operation. RY/$\overline{BY}$ high indicates that the WSM is ready for new commands, block erase is suspended or the device is in deep powerdown mode. RY/$\overline{BY}$ is always active and does **NOT** float to tri-state off when the chip is deselected or data outputs are disabled.
V_{PP}		**BLOCK ERASE/BYTE WRITE POWER SUPPLY** for erasing blocks of the array or writing bytes of each block. **NOTE:** With $V_{PP} < V_{PPLMAX}$, memory contents cannot be altered.
V_{CC}		**DEVICE POWER SUPPLY (5V ±10%, 5V ±5%)**
GND		**GROUND**

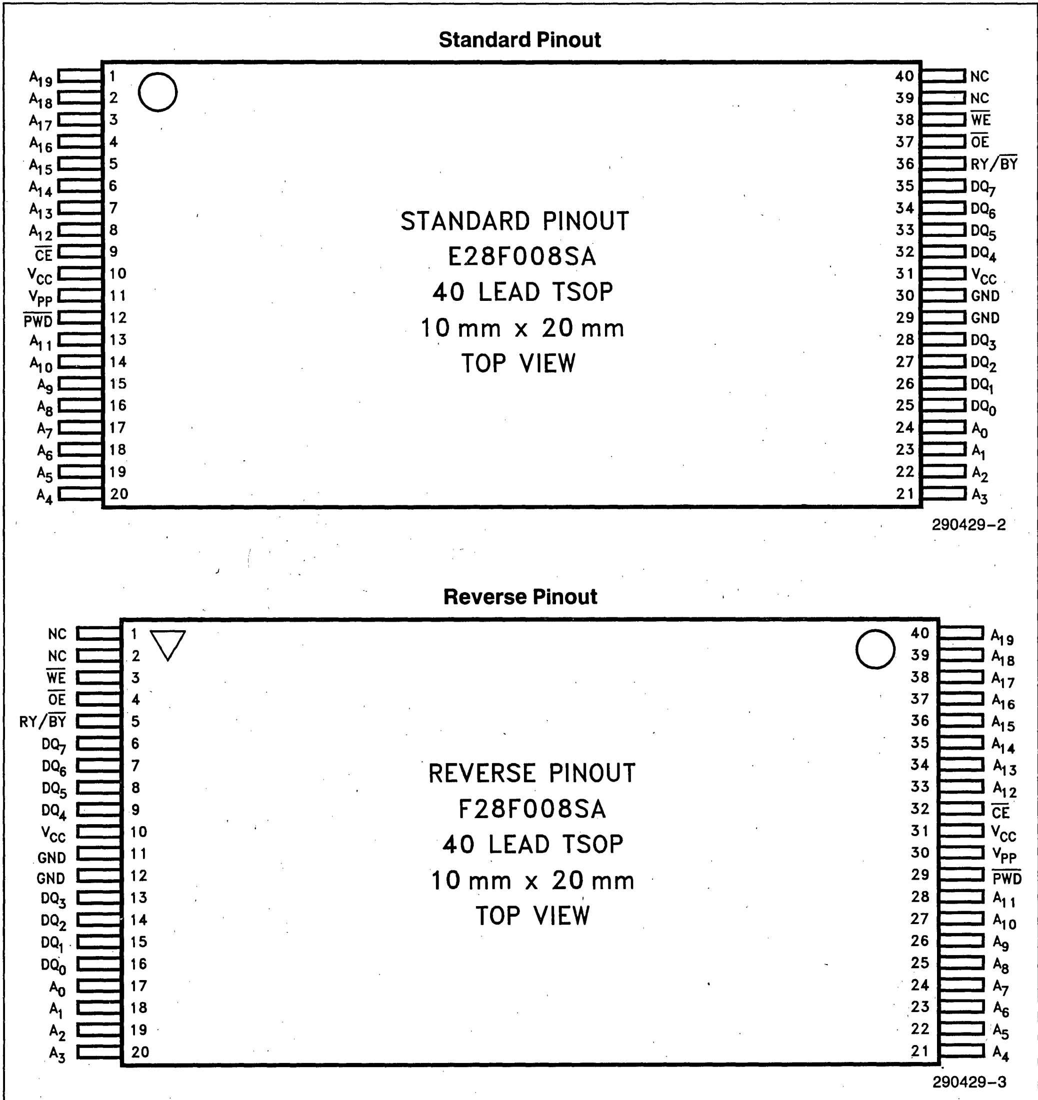


Figure 2. TSOP Lead Configurations

3

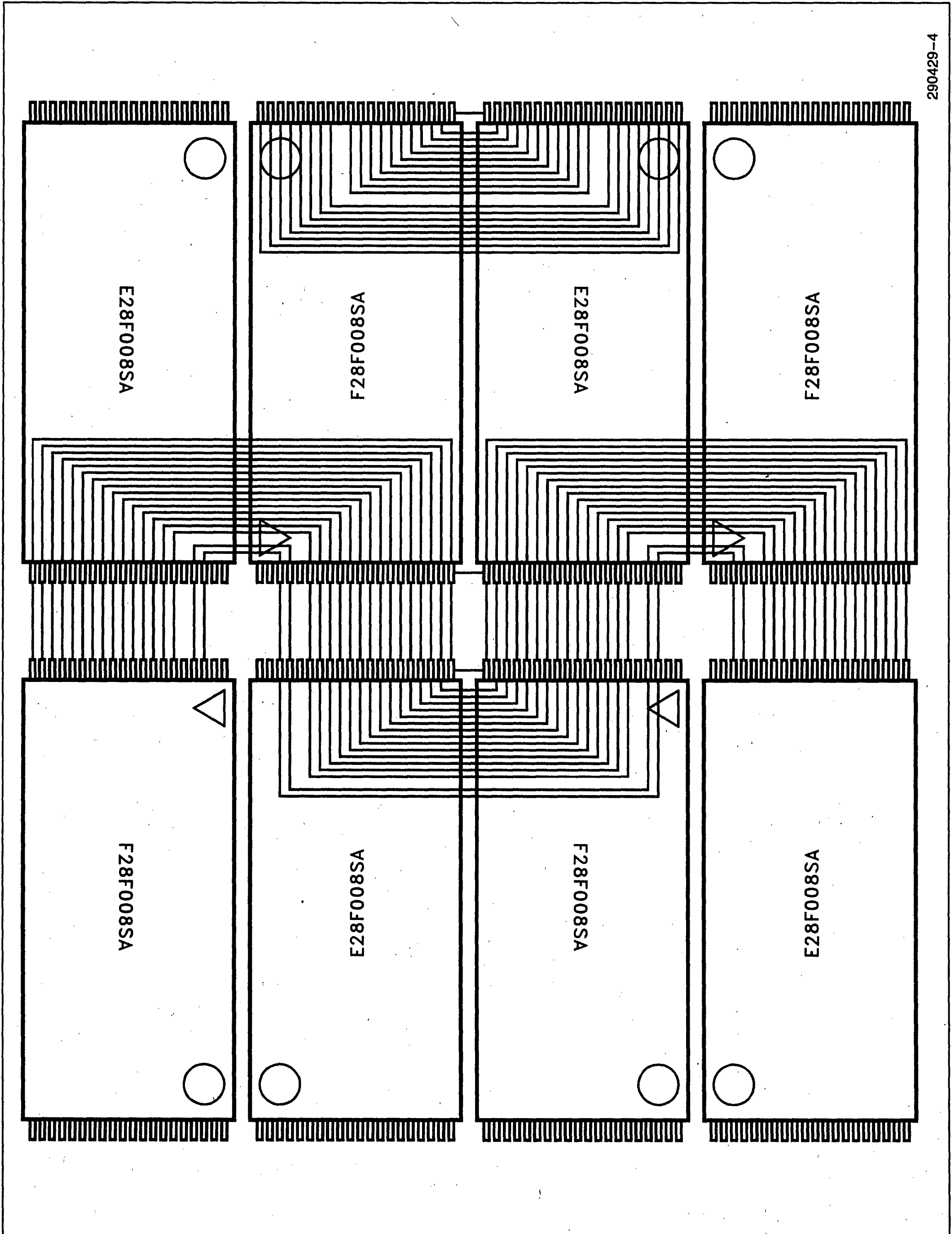


Figure 3. TSOP Serpentine Layout

NOTE:
1. Connect all V_{CC} and GND pins of each device to common power supply outputs. DO NOT leave V_{CC} or GND inputs disconnected.

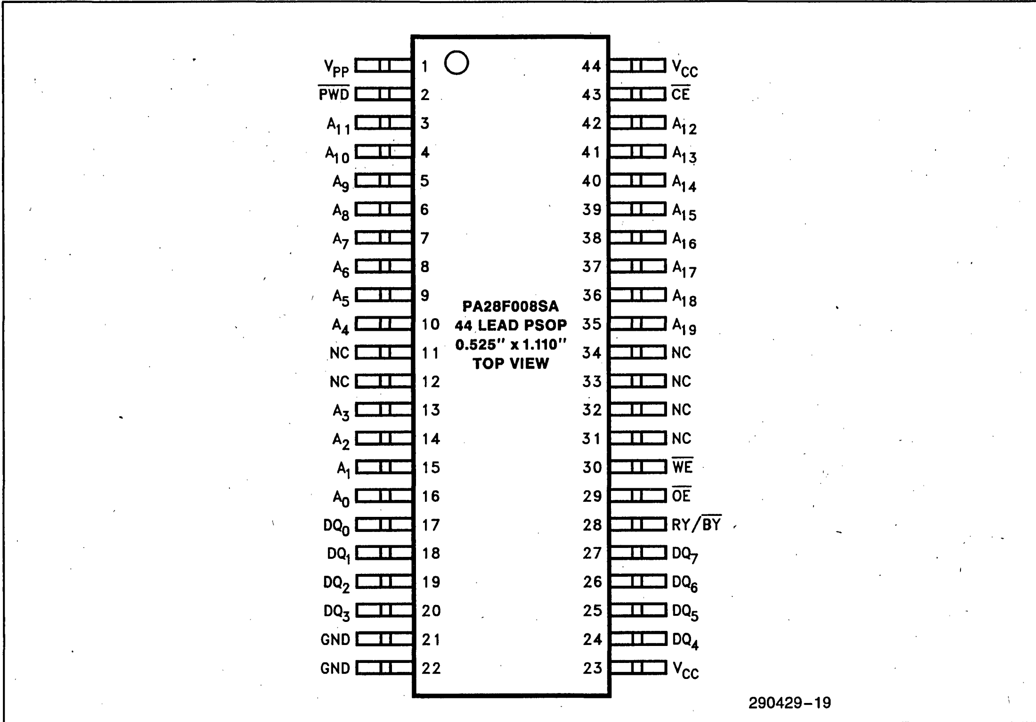


Figure 4. PSOP Lead Configuration

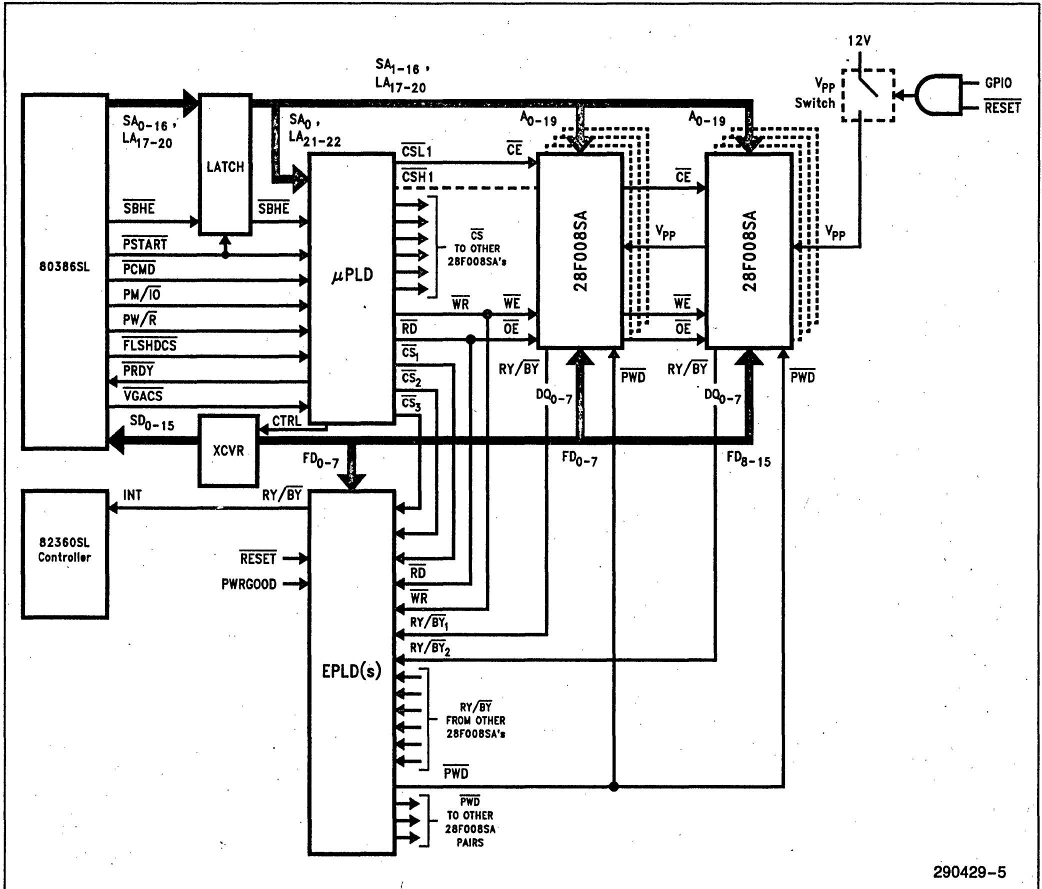


Figure 5. 28F008SA Array Interface to Intel386SL Microprocessor Superset through PI Bus (Including RY/BY Masking and Selective Powerdown), for DRAM Backup during System SUSPEND, Resident O/S and Applications and Motherboard Solid-State Disk.

PRINCIPLES OF OPERATION

The 28F008SA includes on-chip write automation to manage write and erase functions. The Write State Machine allows for: 100% TTL-level control inputs; fixed power supplies during block erasure and byte write; and minimal processor overhead with RAM-like interface timings.

After initial device powerup, or after return from deep powerdown mode (see Bus Operations), the 28F008SA functions as a read-only memory. Manipulation of external memory-control pins allow array read, standby and output disable operations. Both Status Register and intelligent identifiers can also be accessed through the Command User Interface when $V_{PP} = V_{PPL}$.

This same subset of operations is also available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} enables successful block erasure and byte writing of the device. All functions associated with altering memory contents—byte write, block erase, status and intelligent identifier—are accessed via the Command User Interface and verified thru the Status Register.

Commands are written using standard microprocessor write timings. Command User Interface contents serve as input to the WSM, which controls the block erase and byte write circuitry. Write cycles also internally latch addresses and data needed for byte write or block erase operations. With the appropriate command written to the register, standard microprocessor read timings output array data, access the intelligent identifier codes, or output byte write and block erase status for verification.

Interface software to initiate and poll progress of internal byte write and block erase can be stored in any of the 28F008SA blocks. This code is copied to, and executed from, system RAM during actual flash memory update. After successful completion of byte write and/or block erase, code/data reads from the 28F008SA are again possible via the Read Array command. Erase suspend/resume capability allows system software to suspend block erase to read data and execute code from any other block.

Address	Block
FFFFF–F0000	64 Kbyte Block
EFFFF–E0000	64 Kbyte Block
DFFFF–D0000	64 Kbyte Block
CFFFF–C0000	64 Kbyte Block
BFFFF–B0000	64 Kbyte Block
AFFFF–A0000	64 Kbyte Block
9FFFF–90000	64 Kbyte Block
8FFFF–80000	64 Kbyte Block
7FFFF–70000	64 Kbyte Block
6FFFF–60000	64 Kbyte Block
5FFFF–50000	64 Kbyte Block
4FFFF–40000	64 Kbyte Block
3FFFF–30000	64 Kbyte Block
2FFFF–20000	64 Kbyte Block
1FFFF–10000	64 Kbyte Block
0FFFF–00000	64 Kbyte Block

Figure 6. Memory Map

Command User Interface and Write Automation

An on-chip state machine controls block erase and byte write, freeing the system processor for other tasks. After receiving the Erase Setup and Erase Confirm commands, the state machine controls block pre-conditioning and erase, returning progress via the Status Register and RY/$\overline{BY}$ output. Byte write is similarly controlled, after destination address and expected data are supplied. The program and erase algorithms of past Intel Flash memories are now regulated by the state machine, including pulse repetition where required and internal verification and margining of data.

Data Protection

Depending on the application, the system designer may choose to make the V_{PP} power supply switchable (available only when memory byte writes/block erases are required) or hardwired to V_{PPH}. When $V_{PP} = V_{PPL}$, memory contents cannot be altered. The 28F008SA Command User Interface architecture provides protection from unwanted byte write or block erase operations even when high voltage is applied to V_{PP}. Additionally, all functions are disabled whenever V_{CC} is below the write lockout voltage V_{LKO}, or when $\overline{PWD}$ is at V_{IL}. The 28F008SA accommodates either design practice and encourages optimization of the processor-memory interface.

The two-step byte write/block erase Command User Interface write sequence provides additional software write protection.

BUS OPERATION

Flash memory reads, erases and writes in-system via the local CPU. All bus cycles to or from the flash memory conform to standard microprocessor bus cycles.

Read

The 28F008SA has three read modes. The memory can be read from any of its blocks, and information can be read from the intelligent identifier or Status Register. V_{PP} can be at either V_{PPL} or V_{PPH}.

The first task is to write the appropriate read mode command to the Command User Interface (array, intelligent identifier, or Status Register). The 28F008SA automatically resets to Read Array mode upon initial device powerup or after exit from deep powerdown. The 28F008SA has four control pins, two of which must be logically active to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the device selection control, and when active enables the selected memory device. Output Enable ($\overline{OE}$) is the data input/output (DQ_0–DQ_7) direction control, and when active drives data from the selected memory onto the I/O bus. $\overline{PWD}$ and $\overline{WE}$ must also be at V_{IH}. Figure 10 illustrates read bus cycle waveforms.

Output Disable

With $\overline{OE}$ at a logic-high level (V_{IH}), the device outputs are disabled. Output pins (DQ_0–DQ_7) are placed in a high-impedance state.

Standby

$\overline{CE}$ at a logic-high level (V_{IH}) places the 28F008SA in standby mode. Standby operation disables much of the 28F008SA's circuitry and substantially reduces device power consumption. The outputs (DQ_0–DQ_7) are placed in a high-impedence state independent of the status of $\overline{OE}$. If the 28F008SA is deselected during block erase or byte write, the device will continue functioning and consuming normal active power until the operation completes.

Table 2. Bus Operations

Mode	Notes	$\overline{PWD}$	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A_0	V_{PP}	DQ_{0-7}	RY/$\overline{BY}$
Read	1, 2, 3	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	D_{OUT}	X
Output Disable	3	V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	High Z	X
Standby	3	V_{IH}	V_{IH}	X	X	X	X	High Z	X
Deep PowerDown		V_{IL}	X	X	X	X	X	High Z	V_{OH}
Intelligent Identifier (Mfr)		V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{IL}	X	89H	V_{OH}
Intelligent Identifier (Device)		V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{IH}	X	A2H	V_{OH}
Write	3, 4, 5	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	D_{IN}	X

NOTES:
1. Refer to DC Characteristics. When $V_{PP} = V_{PPL}$, memory contents can be read but not written or erased.
2. X can be V_{IL} or V_{IH} for control pins and addresses, and V_{PPL} or V_{PPH} for V_{PP}. See DC Characteristics for V_{PPL} and V_{PPH} voltages.
3. RY/$\overline{BY}$ is V_{OL} when the Write State Machine is executing internal block erase or byte write algorithms. It is V_{OH} when the WSM is not busy, in Erase Suspend mode or deep powerdown mode.
4. Command writes involving block erase or byte write are only successfully executed when $V_{PP} = V_{PPH}$.
5. Refer to Table 3 for valid D_{IN} during a write operation.

Deep Power-Down

The 28F008SA offers a deep powerdown feature, entered when $\overline{PWD}$ is at V_{IL}. Current draw thru V_{CC} is 0.20 μA typical in deep powerdown mode, with current draw through V_{PP} typically 0.1 μA. During read modes, $\overline{PWD}$-low deselects the memory, places output drivers in a high-impedence state and turns off all internal circuits. The 28F008SA requires time t_{PHQV} (see AC Characteristics-Read-Only Operations) after return from powerdown until initial memory access outputs are valid. After this wakeup interval, normal operation is restored. The Command User Interface is reset to Read Array, and the upper 5 bits of the Status Register are cleared to value 10000, upon return to normal operation.

During block erase or byte write modes, $\overline{PWD}$ low will abort either operation. Memory contents of the block being altered are no longer valid as the data will be partially written or erased. Time t_{PHWL} after $\overline{PWD}$ goes to logic-high (V_{IH}) is required before another command can be written.

Intelligent Identifier Operation

The intelligent identifier operation outputs the manufacturer code, 89H; and the device code, A2H for the 28F008SA. The system CPU can then automatically match the device with its proper block erase and byte write algorithms.

The manufacturer- and device-codes are read via the Command User Interface. Following a write of 90H to the Command User Interface, a read from address location 00000H outputs the manufacturer code (89H). A read from address 00001H outputs the device code (A2H). It is not necessary to have high voltage applied to V_{PP} to read the intelligent identifiers from the Command User Interface.

Write

Writes to the Command User Interface enable reading of device data and intelligent identifiers. They also control inspection and clearing of the Status Register. Additionally, when $V_{PP} = V_{PPH}$, the Command User Interface controls block erasure and byte write. The contents of the interface register serve as input to the internal state machine.

Table 3. Command Definitions

Command	Bus Cycles Req'd	Notes	First Bus Cycle			Second Bus Cycle		
			Operation	Address	Data	Operation	Address	Data
Read Array/Reset	1	1	Write	X	FFH			
Intelligent Identifier	3	2, 3, 4	Write	X	90H	Read	IA	IID
Read Status Register	2	3	Write	X	70H	Read	X	SRD
Clear Status Register	1		Write	X	50H			
Erase Setup/Erase Confirm	2	2	Write	BA	20H	Write	BA	D0H
Erase Suspend/Erase Resume	2		Write	X	B0H	Write	X	D0H
Byte Write Setup/Write	2	2, 3, 5	Write	WA	40H	Write	WA	WD
Alternate Byte Write Setup/Write	2	2, 3, 5	Write	WA	10H	Write	WA	WD

NOTES:
1. Bus operations are defined in Table 2.
2. IA = Identifier Address: 00H for manufacturer code, 01H for device code.
   BA = Address within the block being erased.
   WA = Address of memory location to be written.
3. SRD = Data read from Status Register. See Table 4 for a description of the Status Register bits.
   WD = Data to be written at location WA. Data is latched on the rising edge of $\overline{WE}$.
   IID = Data read from intelligent identifiers.
4. Following the intelligent identifier command, two read operations access manufacture and device codes.
5. Either 40H or 10H are recognized by the WSM as the Byte Write Setup command.
6. Commands other than those shown above are reserved by Intel for future device implementations and should not be used.

The Command User Interface itself does not occupy an addressable memory location. The interface register is a latch used to store the command and address and data information needed to execute the command. Erase Setup and Erase Confirm commands require both appropriate command data and an address within the block to be erased. The Byte Write Setup command requires both appropriate command data and the address of the location to be written, while the Byte Write command consists of the data to be written and the address of the location to be written.

The Command User Interface is written by bringing $\overline{WE}$ to a logic-low level (V_{IL}) while $\overline{CE}$ is low. Addresses and data are latched on the rising edge of $\overline{WE}$. Standard microprocessor write timings are used.

Refer to AC Write Characteristics and the AC Waveforms for Write Operations, Figure 11, for specific timing parameters.

COMMAND DEFINITIONS

When V_{PPL} is applied to the V_{PP} pin, read operations from the Status Register, intelligent identifiers, or array blocks are enabled. Placing V_{PPH} on V_{PP} enables successful byte write and block erase operations as well.

Device operations are selected by writing specific commands into the Command User Interface. Table 3 defines the 28F008SA commands.

Read Array Command

Upon initial device powerup and after exit from deep powerdown mode, the 28F008SA defaults to Read Array mode. This operation is also initiated by writing FFH into the Command User Interface. Microprocessor read cycles retrieve array data. The device remains enabled for reads until the Command User Interface contents are altered. Once the internal Write State Machine has started a block erase or byte write operation, the device will not recognize the Read Array command, until the WSM has completed its operation. The Read Array command is functional when $V_{PP} = V_{PPL}$ or V_{PPH}.

Intelligent Identifier Command

The 28F008SA contains an intelligent identifier operation, initiated by writing 90H into the Command User Interface. Following the command write, a read cycle from address 00000H retrieves the manufacturer code of 89H. A read cycle from address 00001H returns the device code of A2H. To terminate the operation, it is necessary to write another valid command into the register. Like the Read Array command, the intelligent identifier command is functional when $V_{PP} = V_{PPL}$ or V_{PPH}.

Table 4. Status Register Definitions

WSMS	ESS	ES	BWS	VPPS	R	R	R
7	6	5	4	3	2	1	0

SR.7 = WRITE STATE MACHINE STATUS
1 = Ready
0 = Busy

SR.6 = ERASE SUSPEND STATUS
1 = Erase Suspended
0 = Erase in Progress/Completed

SR.5 = ERASE STATUS
1 = Error in Block Erasure
0 = Successful Block Erase

SR.4 = BYTE WRITE STATUS
1 = Error in Byte Write
0 = Successful Byte Write

SR.3 = V_{PP} STATUS
1 = V_{PP} Low Detect; Operation Abort
0 = V_{PP} OK

SR.2–SR.0 = RESERVED FOR FUTURE ENHANCEMENTS
These bits are reserved for future use and should be masked out when polling the Status Register.

NOTES:

RY/$\overline{BY}$ or the Write State Machine Status bit must first be checked to determine byte write or block erase completion, before the Byte Write or Erase Status bit are checked for success.

If the Byte Write AND Erase Status bits are set to "1"s during a block erase attempt, an improper command sequence was entered. Attempt the operation again.

If V_{PP} low status is detected, the Status Register must be cleared before another byte write or block erase operation is attempted.

The V_{PP} Status bit, unlike an A/D converter, does not provide continuous indication of V_{PP} level. The WSM interrogates the V_{PP} level only after the byte write or block erase command sequences have been entered and informs the system if V_{PP} has not been switched on. The V_{PP} Status bit is not guaranteed to report accurate feedback between V_{PPL} and V_{PPH}.

Read Status Register Command

The 28F008SA contains a Status Register which may be read to determine when a byte write or block erase operation is complete, and whether that operation completed successfully. The Status Register may be read at any time by writing the Read Status Register command (70H) to the Command User Interface. After writing this command, all subsequent read operations output data from the Status Register, until another valid command is written to the Command User Interface. The contents of the Status Register are latched on the falling edge of $\overline{OE}$ or $\overline{CE}$, whichever occurs last in the read cycle. $\overline{OE}$ or $\overline{CE}$ must be toggled to V_{IH} before further reads to update the Status Register latch. The Read Status Register command functions when $V_{PP} = V_{PPL}$ or V_{PPH}.

Clear Status Register Command

The Erase Status and Byte Write Status bits are set to "1"s by the Write State Machine and can only be reset by the Clear Status Register Command. These bits indicate various failure conditions (see Table 4). By allowing system software to control the resetting of these bits, several operations may be performed (such as cumulatively writing several bytes or erasing multiple blocks in sequence). The Status Register may then be polled to determine if an error occurred during that sequence. This adds flexibility to the way the device may be used.

Additionally, the V_{PP} Status bit (SR.3) MUST be reset by system software before further byte writes or block erases are attempted. To clear the Status Register, the Clear Status Register command (50H) is written to the Command User Interface. The Clear Status Register command is functional when $V_{PP} = V_{PPL}$ or V_{PPH}.

Erase Setup/Erase Confirm Commands

Erase is executed one block at a time, initiated by a two-cycle command sequence. An Erase Setup command (20H) is first written to the Command User Interface, followed by the Erase Confirm command (D0H). These commands require both appropriate sequencing and an address within the block to be erased to FFH. Block preconditioning, erase and verify are all handled internally by the Write State Machine, invisible to the system. After the two-command erase sequence is written to it, the 28F008SA automatically outputs Status Register data when read (see Figure 8; Block Erase Flowchart). The CPU can detect the completion of the erase event by analyzing the output of the RY/$\overline{BY}$ pin, or the WSM Status bit of the Status Register.

When erase is completed, the Erase Status bit should be checked. If erase error is detected, the Status Register should be cleared. The Command User Interface remains in Read Status Register mode until further commands are issued to it.

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, reliable block erasure can only occur when $V_{PP} = V_{PPH}$. In the absence of this high voltage, memory contents are protected against erasure. If block erase is attempted while $V_{PP} = V_{PPL}$, the V_{PP} Status bit will be set to "1". Erase attempts while $V_{PPL} < V_{PP} < V_{PPH}$ produce spurious results and should not be attempted.

Erase Suspend/Erase Resume Commands

The Erase Suspend command allows block erase interruption in order to read data from another block of memory. Once the erase process starts, writing the Erase Suspend command (B0H) to the Command User Interface requests that the WSM suspend the erase sequence at a predetermined point in the erase algorithm. The 28F008SA continues to output Status Register data when read, after the Erase Suspend command is written to it. Polling the WSM Status and Erase Suspend Status bits will determine when the erase operation has been suspended (both will be set to "1"). RY/$\overline{BY}$ will also transition to V_{OH}.

At this point, a Read Array command can be written to the Command User Interface to read data from blocks other than that which is suspended. The only other valid commands at this time are Read Status Register (70H) and Erase Resume (D0H), at which time the WSM will continue with the erase process. The Erase Suspend Status and WSM Status bits of the Status Register will be automatically cleared and RY/$\overline{BY}$ will return to V_{OL}. After the Erase Resume command is written to it, the 28F008SA automatically outputs Status Register data when read (see Figure 9; Erase Suspend/Resume Flowchart). V_{PP} must remain at V_{PPH} while the 28F008SA is in Erase Suspend.

Byte Write Setup/Write Commands

Byte write is executed by a two-command sequence. The Byte Write Setup command (40H) is written to the Command User Interface, followed by a second write specifying the address and data (latched on the rising edge of $\overline{WE}$) to be written. The WSM then takes over, controlling the byte write and write verify algorithms internally. After the two-command byte write sequence is written to it, the 28F008SA automatically outputs Status Register data when read (see Figure 7; Byte Write Flowchart). The CPU can detect the completion of the byte write event by analyzing the output of the RY/$\overline{BY}$ pin, or the WSM Status bit of the Status Register. Only the Read Status Register command is valid while byte write is active.

When byte write is complete, the Byte Write Status bit should be checked. If byte write error is detected, the Status Register should be cleared. The internal WSM verify only detects errors for "1"s that do not successfully write to "0"s. The Command User Interface remains in Read Status Register mode until further commands are issued to it. If byte write is attempted while $V_{PP} = V_{PPL}$, the V_{PP} Status bit will be set to "1". Byte write attempts while $V_{PPL} < V_{PP} < V_{PPH}$ produce spurious results and should not be attempted.

EXTENDED BLOCK ERASE/BYTE WRITE CYCLING

Intel has designed extended cycling capability into its ETOX flash memory technologies. The 28F008SA is designed for 100,000 byte write/block erase cycles on each of the sixteen 64 Kbyte blocks. Low electric fields, advanced oxides and minimal oxide area per cell subjected to the tunneling electric field combine to greatly reduce oxide stress and the probability of failure. A 20 Mbyte solid-state drive using an array of 28F008SAs has a MTBF (Mean Time Between Failure) of 33.3 million hours[1], over 600 times more reliable than equivalent rotating disk technology.

AUTOMATED BYTE WRITE

The 28F008SA integrates the Quick-Pulse programming algorithm of prior Intel Flash devices on-chip, using the Command User Interface, Status Register and Write State Machine (WSM). On-chip integration dramatically simplifies system software and provides processor interface timings to the Command User Interface and Status Register. WSM operation, internal verify and V_{PP} high voltage presence are monitored and reported via the RY/$\overline{BY}$ output and appropriate Status Register bits. Figure 7 shows a system software flowchart for device byte write. The entire sequence is performed with V_{PP} at V_{PPH}. Byte write abort occurs when $\overline{PWD}$ transitions to V_{IL}, or V_{PP} drops to V_{PPL}. Although the WSM is halted, byte data is partially written at the location where byte write was aborted. Block erasure, or a repeat of byte write, is required to initialize this data to a known value.

AUTOMATED BLOCK ERASE

As above, the Quick-Erase algorithm of prior Intel Flash devices is now implemented internally, including all preconditioning of block data. WSM operation, erase success and V_{PP} high voltage presence are monitored and reported through RY/$\overline{BY}$ and the Status Register. Additionally, if a command other than Erase Confirm is written to the device following Erase Setup, both the Erase Status and Byte Write Status bits will be set to "1"s. When issuing the Erase Setup and Erase Confirm commands, they should be written to an address within the address range of the block to be erased. Figure 8 shows a system software flowchart for block erase.

Erase typically takes 1.6 seconds per block. The Erase Suspend/Erase Resume command sequence allows suspension of this erase operation to read data from a block other than that in which erase is being performed. A system software flowchart is shown in Figure 9.

The entire sequence is performed with V_{PP} at V_{PPH}. Abort occurs when $\overline{PWD}$ transitions to V_{IL} or V_{PP} falls to V_{PPL}, while erase is in progress. Block data is partially erased by this operation, and a repeat of erase is required to obtain a fully erased block.

DESIGN CONSIDERATIONS

Three-Line Output Control

The 28F008SA will often be used in large memory arrays. Intel provides three control inputs to accommodate multiple memory connections. Three-line control provides for:

a) lowest possible memory power dissipation

b) complete assurance that data bus contention will not occur

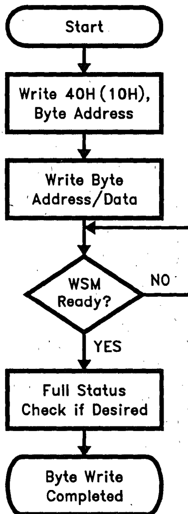
To efficiently use these control inputs, an address decoder should enable $\overline{CE}$, while $\overline{OE}$ should be connected to all memory devices and the system's $\overline{READ}$ control line. This assures that only selected memory devices have active outputs while deselected memory devices are in Standby Mode. Finally, $\overline{PWD}$ should either be tied to the system $\overline{RESET}$, or connected to V_{CC} if unused.

(1)Assumptions: 10 Kbyte file written every 10 minutes. (20 Mbyte array)/(10 Kbyte file) = 2,000 file writes before erase required.
(2000 files writes/erase) $\times$ (100,000 cycles per 28F008SA block) = 200 million file writes.
(200×10^6 file writes) $\times$ (10 min/write) $\times$ (1 hr/60 min) = **33.3×10^6 MTBF.**

RY/$\overline{BY}$ and Byte Write/Block Erase Polling

RY/$\overline{BY}$ is a full CMOS output that provides a hardware method of detecting byte write and block erase completion. It transitions low time t_{WHRL} after a write or erase command sequence is written to the 28F008SA, and returns to V_{OH} when the WSM has finished executing the internal algorithm.

RY/$\overline{BY}$ can be connected to the interrupt input of the system CPU or controller. It is active at all times, not tristated if the 28F008SA $\overline{CE}$ or $\overline{OE}$ inputs are brought to V_{IH}. RY/$\overline{BY}$ is also V_{OH} when the device is in Erase Suspend or deep powerdown modes.



290429-6

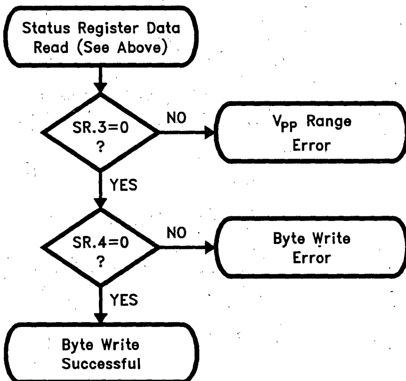
Bus Operation	Command	Comments
Write	Byte Write Setup	Data = 40H (10H) Address = Byte to be written
Write	Byte Write	Data to be written Address = Byte to be written
Standby/Read		Check RY/$\overline{BY}$ V_{OH} = Ready, V_{OL} = Busy or Read Status Register Check SR.7 1 = Ready, 0 = Busy Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register

Repeat for subsequent bytes

Full status check can be done after each byte or after a sequence of bytes

Write FFH after the last byte write operation to reset the device to Read Array Mode

FULL STATUS CHECK PROCEDURE



290429-7

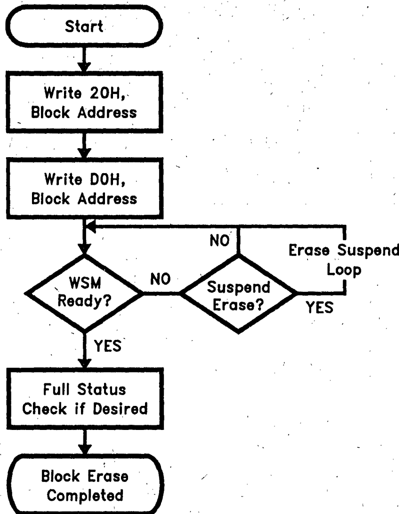
Bus Operation	Command	Comments
Optional Read		CPU may already have read Status Register data in WSM Ready polling above
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Byte Write Error

SR.3 MUST be cleared, if set during a byte write attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple bytes are written before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 7. Automated Byte Write Flowchart



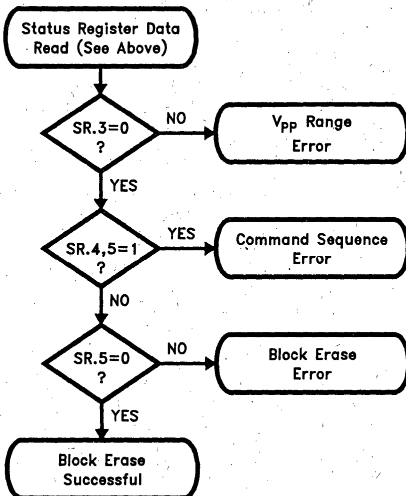
Bus Operation	Command	Comments
Write	Erase Setup	Data = 20H Address = Within block to be erased
Write	Erase	Data = D0H Address = Within block to be erased
Standby/Read		Check RY/$\overline{BY}$ V_{OH} = Ready, V_{OL} = Busy or Read Status Register Check SR.7 1 = Ready, 0 = Busy Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register

Repeat for subsequent bytes

Full status check can be done after each block or after a sequence of blocks

Write FFH after the last block erase operation to reset the device to Ready Array Mode

FULL STATUS CHECK PROCEDURE



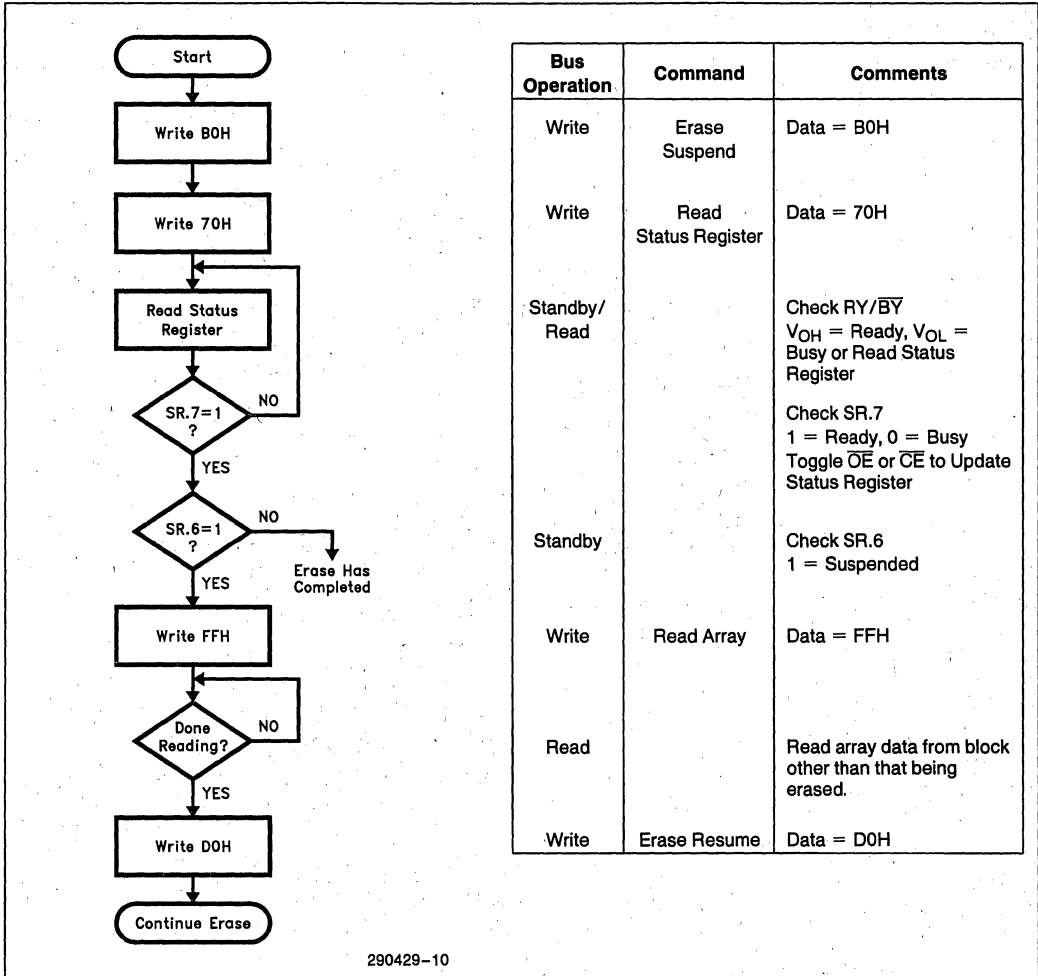
Bus Operation	Command	Comments
Optional Read		CPU may already have read Status Register data in WSM Ready polling above
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4,5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error

SR.3 MUST be cleared, if set during a block erase attempt, before further attempts are allowed by the Write State Machine

SR.5 is only cleared by the Clear Status Register Command, in cases where multiple blocks are erased before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 8. Automated Block Erase Flowchart



Bus Operation	Command	Comments
Write	Erase Suspend	Data = B0H
Write	Read Status Register	Data = 70H
Standby/ Read		Check RY/$\overline{BY}$ V_{OH} = Ready, V_{OL} = Busy or Read Status Register Check SR.7 1 = Ready, 0 = Busy Toggle $\overline{OE}$ or $\overline{CE}$ to Update Status Register
Standby		Check SR.6 1 = Suspended
Write	Read Array	Data = FFH
Read		Read array data from block other than that being erased.
Write	Erase Resume	Data = D0H

Figure 9. Erase Suspend/Resume Flowchart

Power Supply Decoupling

Flash memory power switching characteristics require careful device decoupling. System designers are interested in 3 supply current issues; standby current levels (I_{SB}), active current levels (I_{CC}) and transient peaks produced by falling and rising edges of $\overline{CE}$. Transient current magnitudes depend on the device outputs' capacitive and inductive loading. Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 µF ceramic capacitor connected between each V_{CC} and GND, and between its V_{PP} and GND. These high frequency, low inherent-inductance capacitors should be placed as close as possible to package leads. Additionally, for every 8 devices, a 4.7 µF electrolytic capacitor should be placed at the array's power supply connection between V_{CC} and GND. The bulk capacitor will overcome voltage slumps caused by PC board trace inductances.

V_{PP} Trace on Printed Circuit Boards

Writing flash memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the V_{PP} power supply trace. The V_{PP} pin supplies the memory cell current for writing and erasing. Use similar trace widths and layout considerations given to the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots.

V_{CC}, V_{PP}, $\overline{PWD}$ Transitions and the Command/Status Registers

Byte write and block erase completion are not guaranteed if V_{PP} drops below V_{PPH}. If the V_{PP} Status bit of the Status Register (SR.3) is set to "1", a Clear Status Register command MUST be issued before further byte write/block erase attempts are allowed by the WSM. Otherwise, the Byte Write (SR.4) or Erase (SR.5) Status bits of the Status Register will be set to "1"s if error is detected. $\overline{PWD}$ transitions to V_{IL} during byte write and block erase also abort the operations. Data is partially altered in either case, and the command sequence must be repeated after normal operation is restored. Device poweroff, or $\overline{PWD}$ transitions to V_{IL}, clear the Status Register to initial value 10000 for the upper 5 bits.

The Command User Interface latches commands as issued by system software and is not altered by V_{PP} or $\overline{CE}$ transitions or WSM actions. Its state upon powerup, after exit from deep powerdown or after V_{CC} transitions below V_{LKO}, is Read Array Mode.

After byte write or block erase is complete, even after V_{PP} transitions down to V_{PPL}, the Command User Interface must be reset to Read Array mode via the Read Array command if access to the memory array is desired.

Power Up/Down Protection

The 28F008SA is designed to offer protection against accidental block erasure or byte writing during power transitions. Upon power-up, the 28F008SA is indifferent as to which power supply, V_{PP} or V_{CC}, powers up first. Power supply sequencing is not required. Internal circuitry in the 28F008SA ensures that the Command User Interface is reset to the Read Array mode on power up.

A system designer must guard against spurious writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will inhibit writes. The Command User Interface architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

Finally, the device is disabled until $\overline{PWD}$ is brought to V_{IH}, regardless of the state of its control inputs. This provides an additional level of memory protection.

Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases usable battery life, because the 28F008SA does not consume any power to retain code or data when the system is off.

In addition, the 28F008SA's deep powerdown mode ensures extremely low power dissipation even when system power is applied. For example, portable PCs and other power sensitive applications, using an array of 28F008SAs for solid-state storage, can lower $\overline{PWD}$ to V_{IL} in standby or sleep modes, producing negligable power consumption. If access to the 28F008SA is again needed, the part can again be read, following the t_{PHQV} and t_{PHWL} wakeup cycles required after $\overline{PWD}$ is first raised back to V_{IH}. See AC Characteristics—Read-Only and Write Operations and Figures 10 and 11 for more information.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read ....0°C to +70°C(1)
During Block Erase/Byte Write ....0°C to +70°C

Temperature Under Bias ....−10°C to +80°C

Storage Temperature ....−65°C to +125°C

Voltage on Any Pin
(except V_{CC} and V_{PP})
with Respect to GND ....−2.0V to +7.0V(2)

V_{PP} Program Voltage with
Respect to GND during
Block Erase/Byte Write ...−2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage
with Respect to GND ....−2.0V to +7.0V(2)

Output Short Circuit Current.............100 mA(4)

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC voltage is −0.5V on input/output pins. During transitions, this level may undershoot to −2.0V for periods <20 ns. Maximum DC voltage on input/output pins is V_{CC} + 0.5V which, during transitions, may overshoot to V_{CC} + 2.0V for periods <20 ns.
3. Maximum DC voltage on V_{PP} may overshoot to +14.0V for periods <20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.
5. 5% V_{CC} specifications reference the 28F008SA-85 in its High Speed configuration. 10% V_{CC} specifications reference the 28F008SA-85 in its Standard configuration, and the 28F008SA-120.

OPERATING CONDITIONS

Symbol	Parameter	Notes	Min	Max	Unit
T_A	Operating Temperature		0	70	°C
V_{CC}	V_{CC} Supply Voltage (10%)	5	4.50	5.50	V
V_{CC}	V_{CC} Supply Voltage (5%)	5	4.75	5.25	V

DC CHARACTERISTICS

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{LI}	Input Load Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND
I_{CCS}	V_{CC} Standby Current	1, 3		1.0	2.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{IH}$
				30	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = \overline{PWD} = V_{CC} \pm 0.2V$
I_{CCD}	V_{CC} Deep PowerDown Current	1		0.20	1.2	μA	$\overline{PWD}$ = GND ±0.2V I_{OUT} (RY/$\overline{BY}$) = 0 mA
I_{CCR}	V_{CC} Read Current	1		20	35	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE}$ = GND f = 8 MHz, I_{OUT} = 0 mA CMOS Inputs
				25	50	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 8 MHz, I_{OUT} = 0 mA TTL Inputs

DC CHARACTERISTICS (Continued)

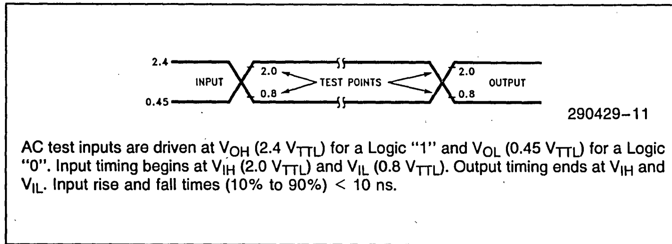
Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{CCW}	V_{CC} Byte Write Current	1		10	30	mA	Byte Write In Progress
I_{CCE}	V_{CC} Block Erase Current	1		10	30	mA	Block Erase In Progress
I_{CCES}	V_{CC} Erase Suspend Current	1, 2		5	10	mA	Block Erase Suspended $\overline{CE} = V_{IH}$
I_{PPS}	V_{PP} Standby Current	1		±1	±10	μA	$V_{PP} \leq V_{CC}$
				90	200	μA	$V_{PP} \leq V_{CC}$
I_{PPD}	V_{PP} Deep PowerDown Current	1		0.10	5.0	μA	$\overline{PWD}$ = GND ±0.2V
I_{PPW}	V_{PP} Byte Write Current	1		10	30	mA	$V_{PP} = V_{PPH}$ Byte Write in Progress
I_{PPE}	V_{PP} Block Erase Current	1		10	30	mA	$V_{PP} = V_{PPH}$ Block Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1		90	200	μA	$V_{PP} = V_{PPH}$ Block Erase Suspended
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage	3			0.45	V	$V_{CC} = V_{CC}$ Min I_{OL} = 5.8 mA
V_{OH}	Output High Voltage	3	2.4			V	$V_{CC} = V_{CC}$ Min I_{OH} = −2.5 mA
V_{PPL}	V_{PP} during Normal Operations	4	0.0		6.5	V	
V_{PPH}	V_{PP} during Erase/Write Operations		11.4	12.0	12.6	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.0			V	

CAPACITANCE(5) T_A = 25°C, f = 1 MHz

Symbol	Parameter	Typ	Max	Unit	Condition
C_{IN}	Input Capacitance	6	8	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C. These currents are valid for all product versions (packages and speeds).
2. I_{CCES} is specified with the device deselected. If the 28F008SA is read while in Erase Suspend Mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Includes RY/$\overline{BY}$.
4. Block Erases/Byte Writes are inhibited when $V_{PP} = V_{PPL}$ and not guaranteed in the range between V_{PPH} and V_{PPL}.
5. Sampled, not 100% tested.

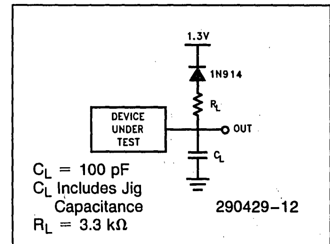
AC INPUT/OUTPUT REFERENCE WAVEFORM(1)



290429–11

AC test inputs are driven at V_{OH} (2.4 V_{TTL}) for a Logic "1" and V_{OL} (0.45 V_{TTL}) for a Logic "0". Input timing begins at V_{IH} (2.0 V_{TTL}) and V_{IL} (0.8 V_{TTL}). Output timing ends at V_{IH} and V_{IL}. Input rise and fall times (10% to 90%) < 10 ns.

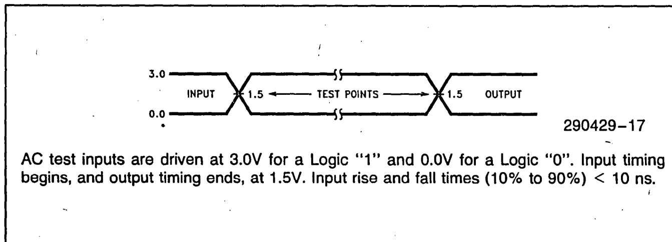
AC TESTING LOAD CIRCUIT(1)



C_L = 100 pF
C_L Includes Jig Capacitance
R_L = 3.3 kΩ

290429–12

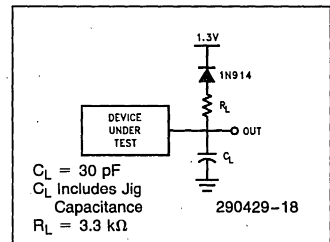
HIGH SPEED AC INPUT/OUTPUT REFERENCE WAVEFORM(2)



290429–17

AC test inputs are driven at 3.0V for a Logic "1" and 0.0V for a Logic "0". Input timing begins, and output timing ends, at 1.5V. Input rise and fall times (10% to 90%) < 10 ns.

HIGH SPEED AC TESTING LOAD CIRCUIT(2)



C_L = 30 pF
C_L Includes Jig Capacitance
R_L = 3.3 kΩ

290429–18

NOTES:
1. Testing characteristics for 28F008SA-85 in Standard configuration, and 28F008SA-120.
2. Testing characteristics for 28F008SA-85 in High Speed configuration.

AC CHARACTERISTICS—Read-Only Operations(1)

Versions			$V_{CC}\pm5\%$	28F008SA-85(4)						Unit
			$V_{CC}\pm10\%$			28F008SA-85(5)		28F008SA-120(5)		
Symbol		**Parameter**	**Notes**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{AVAV}	t_{RC}	Read Cycle Time		85		90		120		ns
t_{AVQV}	t_{ACC}	Address to Output Display			85		90		120	ns
t_{ELQV}	t_{CE}	$\overline{CE}$ to Output Delay	2		85		90		120	ns
t_{PHQV}	t_{PWH}	$\overline{PWD}$ High to Output Delay			400		400		400	ns
t_{GLQV}	t_{OE}	$\overline{OE}$ to Output Delay	2		40		45		50	ns
t_{ELQX}	t_{LZ}	$\overline{CE}$ to Output Low Z	3	0		0		0		ns
t_{EHQZ}	t_{HZ}	$\overline{CE}$ High to Output High Z	3		55		55		55	ns
t_{GLQX}	t_{OLZ}	$\overline{OE}$ to Output Low Z	3	0		0		0		ns
t_{GHQZ}	t_{DF}	$\overline{OE}$ High to Output High Z	3		30		30		30	ns
	t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change, Whichever is First	3	0		0		0		ns

NOTES:
1. See AC Input/Output Reference Waveform for timing measurements.
2. $\overline{OE}$ may be delayed up to $t_{CE}-t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.
3. Sampled, not 100% tested.
4. See High Speed AC Input/Output Reference Waveforms and High Speed AC Testing Load Circuits for testing characteristics.
5. See AC Input/Output Reference Waveforms and AC Testing Load Circuits for testing characteristics.

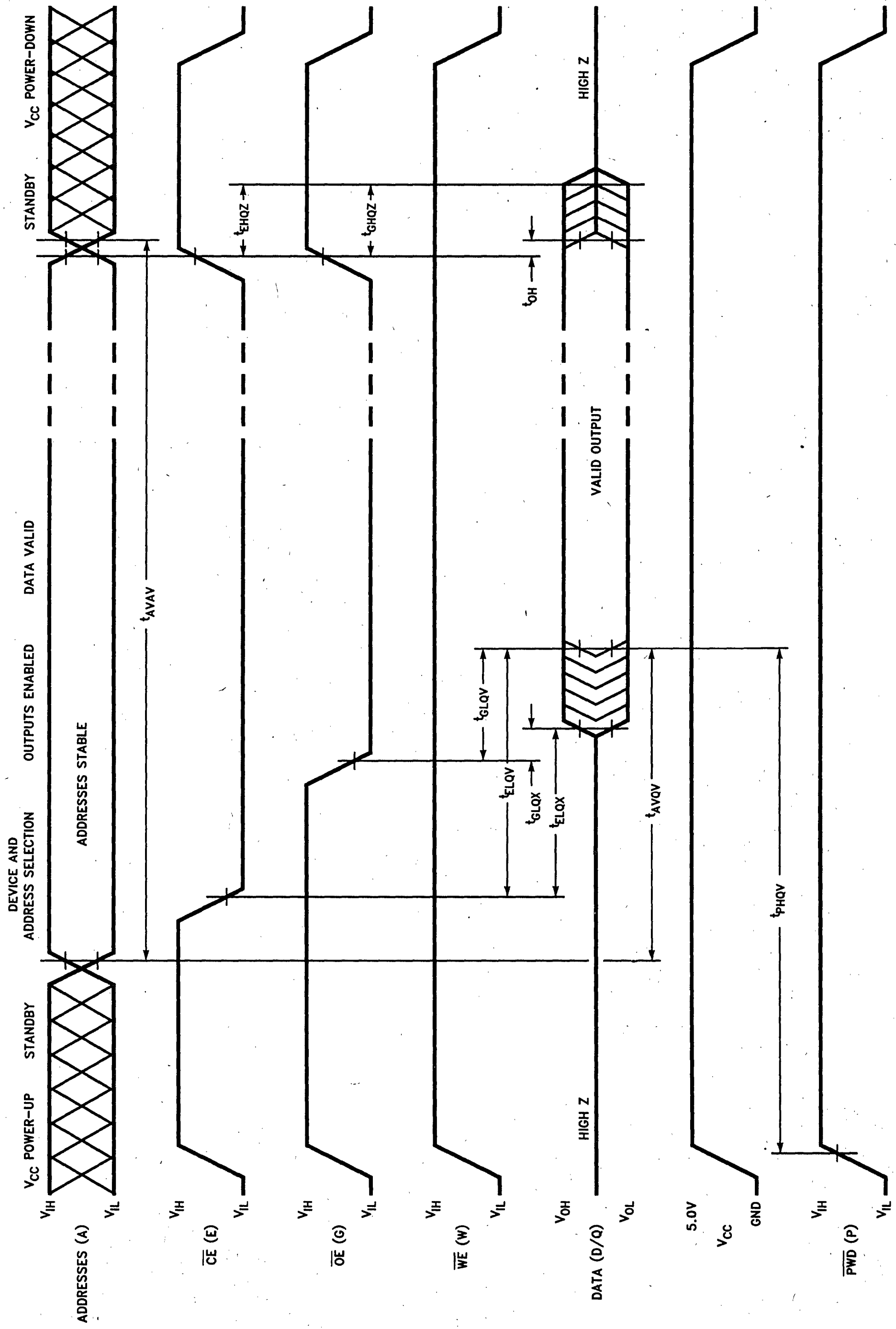


Figure 10. AC Waveform for Read Operations

AC CHARACTERISTICS—Write Operations[1]

Versions			$V_{CC} \pm 5\%$	28F008SA-85[7]						
			$V_{CC} \pm 10\%$			28F008SA-85[8]		28F008SA-120[8]		Unit
Symbol		**Parameter**	**Notes**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{AVAV}	t_{WC}	Write Cycle Time		85		90		120		ns
t_{PHWL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{WE}$ Going Low	2	1		1		1		µs
t_{ELWL}	t_{CS}	$\overline{CE}$ Setup to $\overline{WE}$ Going Low		10		10		10		ns
t_{WLWH}	t_{WP}	$\overline{WE}$ Pulse Width		40		40		40		ns
t_{VPWH}	t_{VPS}	V_{PP} Setup to $\overline{WE}$ Going High	2	100		100		100		ns
t_{AVWH}	t_{AS}	Address Setup to $\overline{WE}$ Going High	3	40		40		40		ns
t_{DVWH}	t_{DS}	Data Setup to $\overline{WE}$ Going High	4	40		40		40		ns
t_{WHDX}	t_{DH}	Data Hold from $\overline{WE}$ High		5		5		5		ns
t_{WHAX}	t_{AH}	Address Hold from $\overline{WE}$ High		5		5		5		ns
t_{WHEH}	t_{CH}	$\overline{CE}$ Hold from $\overline{WE}$ High		10		10		10		ns
t_{WHWL}	t_{WPH}	$\overline{WE}$ Pulse Width High		30		30		30		ns
t_{WHRL}		$\overline{WE}$ High to RY/$\overline{BY}$ Going Low			100		100		100	ns
t_{WHQV1}		Duration of Byte Write Operation	5, 6	6		6		6		µs
t_{WHQV2}		Duration of Block Erase Operation	5, 6	0.3		0.3		0.3		sec
t_{WHGL}		Write Recovery before Read		0		0		0		µs
t_{QVVL}	t_{VPH}	V_{PP} Hold from Valid SRD, RY/$\overline{BY}$ High	2, 6	0		0		0		ns

NOTES:

1. Read timing characteristics during erase and byte write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Sampled, not 100% tested.
3. Refer to Table 3 for valid A_{IN} for byte write or block erasure.
4. Refer to Table 3 for valid D_{IN} for byte write or block erasure.
5. The on-chip Write State Machine incorporates all byte write and block erase system functions and overhead of standard Intel flash memory, including byte program and verify (byte write) and block precondition, precondition verify, erase and erase verify (block erase).
6. Byte write and block erase durations are measured to completion (SR.7 = 1, RY/$\overline{BY}$ = V_{OH}). V_{PP} should be held at V_{PPH} until determination of byte write/block erase success (SR.3/4/5 = 0)
7. See High Speed AC Input/Output Reference Waveforms and High Speed AC Testing Load Circuits for testing characteristics.
8. See AC Input/Output Reference Waveforms and AC Testing Load Circuits for testing characteristics.

BLOCK ERASE AND BYTE WRITE PERFORMANCE

Parameter	Notes	28F008SA-85			28F008SA-120			Unit
		Min	Typ(1)	Max	Min	Typ(1)	Max	
Block Erase Time	2		1.6	10		1.6	10	sec
Block Write Time	2		0.6	2.1		0.6	2.1	sec

NOTES:
1. 25°C, 12.0 V_{PP}.
2. Excludes System-Level Overhead.

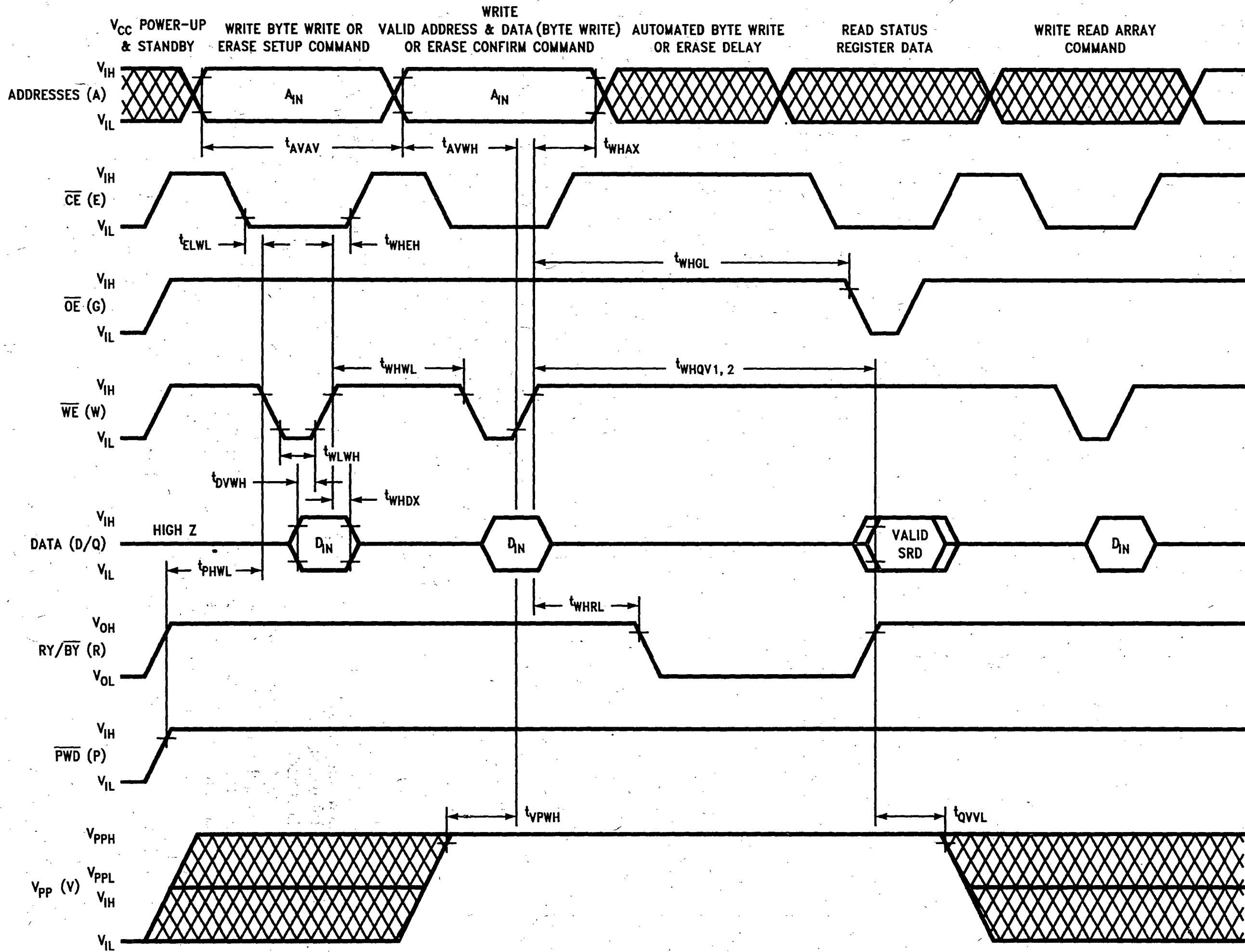


Figure 11. AC Waveform for Write Operations

ALTERNATIVE $\overline{CE}$-CONTROLLED WRITES

Versions			$V_{CC} \pm 5\%$	28F008SA-85(6)						Unit
			$V_{CC} \pm 10\%$			28F008SA-85(7)		28F008SA-120(7)		
Symbol		**Parameter**	**Notes**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{AVAV}	t_{WC}	Write Cycle Time		85		90		120		ns
t_{PHEL}	t_{PS}	$\overline{PWD}$ High Recovery to $\overline{CE}$ Going Low	2	1		1		1		μs
t_{WLEL}	t_{WS}	$\overline{WE}$ Setup to $\overline{CE}$ Going Low		0		0		0		ns
t_{ELEH}	t_{CP}	$\overline{CE}$ Pulse Width		50		50		50		ns
t_{VPEH}	t_{VPS}	V_{PP} Setup to $\overline{CE}$ Going High	2	100		100		100		ns
t_{AVEH}	t_{AS}	Address Setup to $\overline{CE}$ Going High	3	40		40		40		ns
t_{DVEH}	t_{DS}	Data Setup to $\overline{CE}$ Going High	4	40		40		40		ns
t_{EHDX}	t_{DH}	Data Hold from $\overline{CE}$ High		5		5		5		ns
t_{EHAX}	t_{AH}	Address Hold from $\overline{CE}$ High		5		5		5		ns
t_{EHWH}	t_{WH}	$\overline{WE}$ Hold from $\overline{CE}$ High		0		0		0		ns
t_{EHEL}	t_{EPH}	$\overline{CE}$ Pulse Width High		25		25		25		ns
t_{EHRL}		$\overline{CE}$ High to RY/$\overline{BY}$ Going Low			100		100		100	ns
t_{EHQV1}		Duration of Byte Write Operation	5	6		6		6		μs
t_{EHQV2}		Duration of Block Erase Operation	5	0.3		0.3		0.3		sec
t_{EHGL}		Write Recovery before Read		0		0		0		μs
t_{QVVL}	t_{VPH}	V_{PP} Hold from Valid SRD, RY/$\overline{BY}$ High	2, 5	0		0		0		ns

NOTES:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of $\overline{CE}$ and $\overline{WE}$. In systems where $\overline{CE}$ defines the write pulsewidth (within a longer $\overline{WE}$ timing waveform), all setup, hold and inactive $\overline{WE}$ times should be measured relative to the $\overline{CE}$ waveform.
2. Sampled, not 100% tested.
3. Refer to Table 3 for valid A_{IN} for byte write or block erasure.
4. Refer to Table 3 for valid D_{IN} for byte write or block erasure.
5. Byte write and block erase durations are measured to completion (SR.7 = 1, RY/$\overline{BY}$ = V_{OH}). V_{PP} should be held at V_{PPH} until determination of byte write/block erase success (SR.3/4/5 = 0)
6. See High Speed AC Input/Output Reference Waveforms and High Speed AC Testing Load Circuits for testing characteristics.
7. See AC Input/Output Reference Waveforms and AC Testing Load Circuits for testing characteristics.

VCC POWER-UP & STANDBY
WRITE BYTE WRITE OR ERASE SETUP COMMAND
WRITE VALID ADDRESS & DATA (BYTE WRITE) OR ERASE CONFIRM COMMAND
AUTOMATED BYTE WRITE OR ERASE DELAY
READ STATUS REGISTER DATA
WRITE READ ARRAY COMMAND

ADDRESSES (A)
WE (W)
OE (G)
CE (E)
DATA (D/Q)
RY/BY (R)
PWD (P)
VPP (V)

HIGH Z
VALID SRD

t_{AVAV} t_{AVEH} t_{EHAX} t_{WLEL} t_{EHWH} t_{EHGL} t_{EHEL} $t_{EHQV1,2}$ t_{ELEH} t_{DVEH} t_{EHDX} t_{PHEL} t_{EHRL} t_{VPEH} t_{QVVL}

290429–15

Figure 12. Alternate AC Waveform for Write Operations

ORDERING INFORMATION

VALID COMBINATIONS

E28F008SA-85	F28F008SA-85	PA28F0085A-85
E28F008SA-120	F28F008SA-120	PA28F0085A-120

ADDITIONAL INFORMATION

		Order Number
	28F008SA-L Datasheet	290435
AP-359	"28F008SA Hardware Interfacing"	292094
AP-360	"25F008SA Software Drivers"	292095
AP-364	"28F008SA Automation and Algorithms"	292099
ER-27	"The Intel 28F008SA Flash Memory"	294011
ER-28	"ETOX™ III Flash Memory Technology"	290412

REVISION HISTORY

Number	Description
002	Revised from **Advanced Information** to **Preliminary** Modified Erase Suspend Flowchart Removed -90 speed bin Integrated -90 characteristics into -85 speed bin Combined V_{PP} Standby current and V_{PP} Read current into one V_{PP} Standby current spec with two test conditions (DC Characteristics table) Lowered V_{LKO} from **2.2V** to **2.0V**.

October 1992

3

Using Flash Memory for In-System Reprogrammable Nonvolatile Storage

SAUL ZALES
DALE ELBERT
APPLICATIONS ENGINEERING
INTEL CORPORATION

Order Number: 292046-004

USING FLASH MEMORY FOR IN-SYSTEM REPROGRAMMABLE NONVOLATILE STORAGE

CONTENTS

PAGE

1.0 INTRODUCTION 3-312

1.1 PROM Programmer vs System-Processor Controlled Programming 3-312

1.2 Information Download and Upload 3-312

Version Updates (Download) 3-312

Data Acquisition (Upload) 3-312

2.0 DEVICE FEATURES AND ISW APPLICATION CONSIDERATIONS 3-313

2.1 Flash Memory Pinouts 3-313

2.2 Command Register Architecture 3-315

Simplified Processor Interface .. 3-315

Command Register Reset 3-316

Data Protection on Power Transitions 3-316

2.3 V_{PP} Specifications 3-317

3.0 HARDWARE DESIGN FOR ISW ... 3-317

3.1 V_{PP} Generation 3-317

3.1.1 Regulating Down from Higher Voltage 3-317

3.1.2 Pumping 5V up to 12V 3-317

3.1.3 Absolute Data Protection — V_{PP} On/Off Control 3-318

3.1.4 Writes and Reads during V_{PP} Transitions 3-318

3.1.5 Other V_{PP} Considerations 3-318

3.1.6 V_{PP} Circuitry and Trace Layout 3-319

3.2 Communications — Getting Data to and from the Flash Memory 3-319

4.0 SOFTWARE DESIGN FOR ISW 3-319

4.1 System Integration — Boot Code Requirements 3-319

4.1.1 ISW Flag Check 3-320

4.2 Communication Protocols and Flash Memory ISW 3-320

4.3 Data Accumulation Software Techniques 3-321

CONTENTS PAGE

4.4 Reprogramming Routines 3-321

4.4.1 Quick-Erase Algorithm 3-321

Algorithm Timing Delays .. 3-321

High Performance Parallel Device Erasure 3-322

4.4.2 Quick-Pulse Programming Algorithm 3-323

Algorithm Timing Delays .. 3-323

High Performance Parallel Device Programming ... 3-323

4.4.3 Pulse Width Timing Techniques 3-324

Software Methods and Examples 3-324

Hardware Methods 3-324

5.0 SYSTEM DESIGN EXAMPLE: AN 80C186 DESIGN 3-325

6.0 SUMMARY 3-326

CONTENTS PAGE

APPENDIX A: ON BOARD PROGRAMMING DESIGN CONSIDERATIONS 3-327

APPENDIX B: V_{PP} GENERATION CIRCUITS 3-337

APPENDIX C: LIST OF DC-DC CONVERTER COMPANIES 3-341

APPENDIX D: DETAILED PARALLEL ERASE FLOW CHART 3-342

APPENDIX E: DETAILED PARALLEL PROGRAMMING FLOW CHART 3-344

APPENDIX F: DETAILED SYSTEM SCHEMATICS ... 3-346

1.0 INTRODUCTION

Intel's ETOX™ II (EPROM tunnel oxide) flash memory technology uses a single-transistor cell to provide in-system reprogrammable nonvolatile storage. Reprogramming entails electrically erasing all bits in parallel and then randomly programming any byte in the array. This new technology offers designers alternatives for two of industry's needs: 1) a cost-effective means of updating program code; and 2) a solid-state approach for non-volatile data accumulation or storage.

This application note:

- introduces you to the concepts of in-system writing;
- discusses the hardware and software considerations for reprogramming flash memories in-system;
- offers a checklist for integrating Intel's flash memories into microprocessor- or microcontroller-based systems; and
- shows an example of an 80C186 design which incorporates flash memory.

1.1 PROM Programmer vs System-Processor Controlled Programming

While soldered to a printed circuit board, one of two sources controls flash memory reprogramming: 1) a PROM programmer connected to the board, or 2) the system's own central processing unit (CPU). These are called on-board programming (OBP), and in-system writing (ISW), respectively. With OBP, the PROM programmer supplies the programming voltage (V_{PP}) and the programming intelligence; with ISW, V_{PP} is generated locally and the system itself drives the reprogramming process. Both methods offer a variety of benefits. However this application note focuses on ISW.

NOTE:
See Appendix A for OPB design considerations.

1.2 Information Download and Upload

ETOX II flash memory technology programs extremely quick, permitting "on-the-fly" programming with unbuffered 19.2K baud data input. The remote ISW system handles the serial communication link for the host interface, as well as the flash memory reprogramming.

Version Updates (Download)

Flash memories enable code version updates using simple hardware designs. Beyond the basic system, a local V_{PP} supply is all that is needed for remote code download.

A central host computer can download program code to many remote systems. Flash memory offers this capability without the drawbacks of other technologies. It is solid-state and nonvolatile, thus eliminating mechanical component wear-out (common with disk drives) and the risk of losing updates (a concern with battery-backed RAM). These aspects of flash memory offer major advantages in automated factories, remote systems, portable equipment and other applications. Finally, flash memories provide this capability at a much lower cost than byte-alterable EEPROM and battery-backed SRAM.

Data Acquisition (Upload)

Intel's flash memories allow single-byte programming for data accumulation applications. A remote data-logger uploads its information to a central host via serial link. The flash memory device is then in-system erased

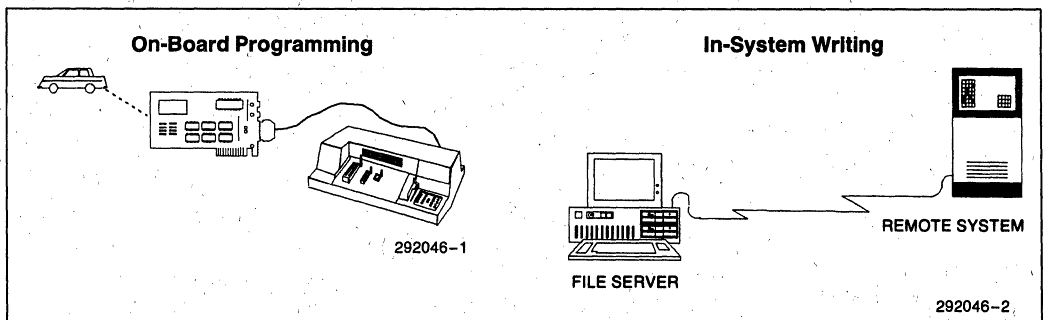


Figure 1. These diagrams illustrate OBP and ISW. In OBP, a PROM programmer updates a system's flash memory. The ISW diagram shows a host updating remote flash memory via serial link. The remote system performs the flash reprogramming with its own CPU.

Intel's ETOX flash memory process has patents pending.

for resumption of data acquisition. This is useful in an advanced electrical power meter, for example. It could be configured to track and monitor power usage and report the data to a central computer for billing and utility management. This reduces the cost of manual door-to-door meter reading.

2.0 DEVICE FEATURES AND ISW APPLICATION CONSIDERATIONS

This section gives a brief overview of Intel's flash memory features and explains how they facilitate ISW design.

2.1 Flash Memory Pinouts

The 32-pin DIP memory site is forward-compatible from the 256K bit to the 2 Mbit flash memory density. It fits into the 27C010 Mbit EPROM pinout and requires no multiplexed pins. Also, with just a single circuit-board jumper trace, a 28-pin EPROM can be placed in the lower pins of the 32-pin flash memory site. (See Figures 2A and 2B, Flash Memory Pinouts.) For more information on intertechnology pin compatibility see Ap Brief AB-25.

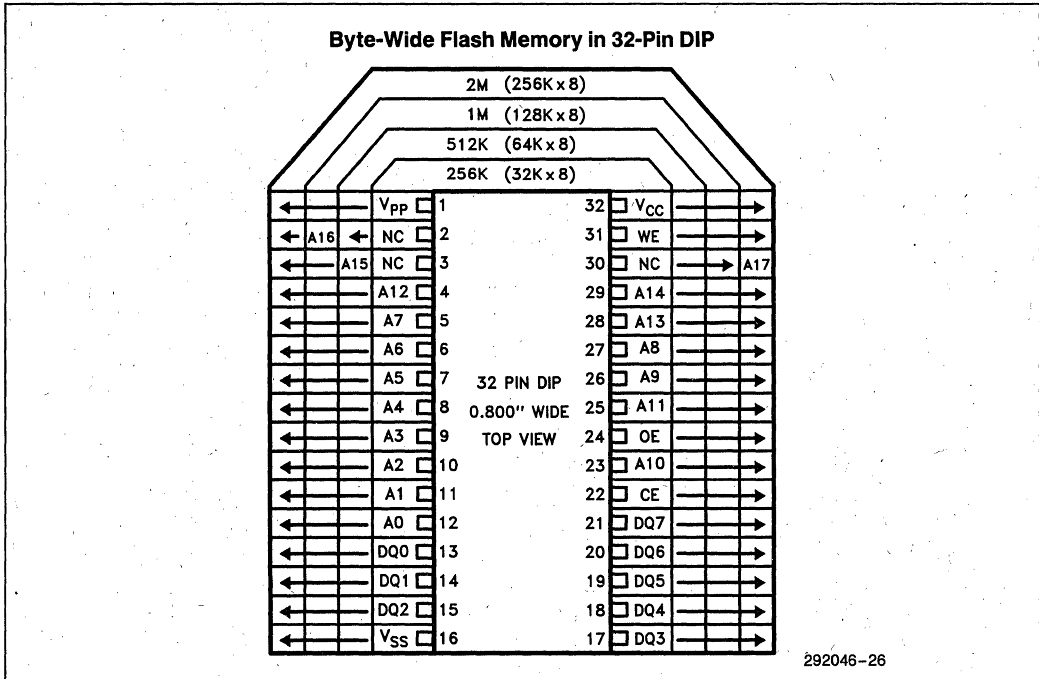


Figure 2A. Flash Memory Pinouts

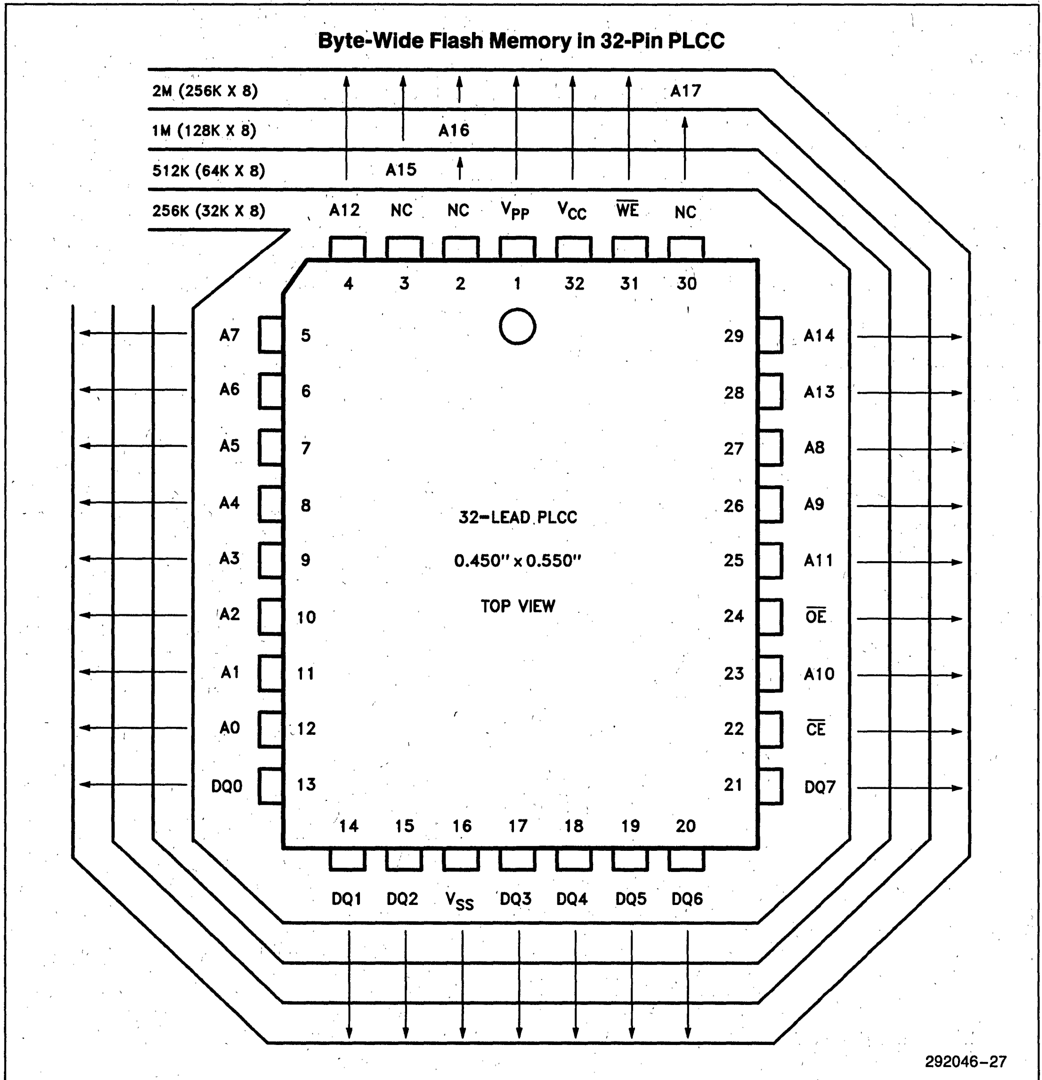


Figure 2B. Flash Memory Pinouts

Table 1. Command Register Instructions

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation	Addr(1)	Data(2)	Operation	Addr(1)	Data(2)
Read Memory(3)	1	Write	X	00H	Read	Valid	Valid
Read inteligent Identifier	1	Write	X	90H	Read	00/01H	ID
Set-Up Erase/Erase	2	Write	X	20H	Write	X	20H
Erase Verify	2	Write	EA	A0H	Read	X	EVD
Set-Up Program/Program	2	Write	X	40H	Write	PA	PD
Program Verify	2	Write	X	C0H	Read	X	PVD
Reset(3)	2	Write	X	FFH	Write	X	FFH

NOTES:
1. Addresses are latched on the falling edge of the Write-Enable pulse.
EA = Address of memory location to be read during erase verify.
PA = Address of memory location to be read during program verify.
2. EVD = Data read from location EA during erase verify.
PD = Data to be programmed at location PA. Data is latched on the rising edge of Write-Enable.
PVD = Data read from location PA during program verify. PA is latched on the Program command.
3. The second bus cycle must be followed by the next desired command register write, given the proper delay times.

2.2 Command Register Architecture

Simplified Processor Interface

Intel's command register architecture simplifies the processor interface. The command register allows CE\, WE\, and OE\ to have standard read/write functionality. All commands such as "Set-up Program" or "Program Verify" can be written with standard system timings. Raising V_{PP} to 12V enables the command register for memory read/write operation, while lowering V_{PP} below V_{CC} + 2V restores the device to a read only memory.

Writing to the register toggles an internal state-machine. The state-machine output controls device functionality. Some commands require one write cycle, while others require two. The command register itself does not occupy an addressable memory location. The register simply stores the command, along with address and data needed to execute the command. With this architecture, the device expects the first write cycle to be a command and does not corrupt data at the specified address. Table 1 contains a list of command register instructions.

The following sections describe the commands in relation to device operation. For more information on the command register see the appropriate flash memory data sheets, and Section 4.4 "Reprogramming Routines".

Read Memory Command—00H

This command allows for normal memory read operations with V_{PP} turned on. After writing the command and waiting 6 μs, the CPU can read from the memory at system speeds. Once placed in the read mode no further action is required on the command register for subsequent read operations.

Read inteligent Identifier Command—90H

Most PROM programmers read the device's inteligent Identifier to select the proper programming algorithm. On EPROMs, raising A9 to the V_{PP} level configures the device for this purpose. Since this is unacceptable in-system, you can read the flash memory inteligent Identifier by first writing command 90H. Follow this by reading address 0000 and 0001H for the manufacturer and device ID. Reset the device with the Read Memory command after you have read the identifier.

Set-Up Erase/Erase Commands—20H

Write this command (20H) twice in succession to initiate erasure. The first write cycle sets up the device for erasure. The device starts erasing itself on the second command's rising edge of Write-Enable. Erasure is stopped when the CPU issues the Erase Verify command or when the device's integrated stop timer times out. Integrated stop timers provide a safety net for complex system environments. In these environments, s/w timer accuracy may be difficult to achieve. Some method of timing is still required, however the timer need only meet a minimum specification (10 ms). This is far easier than calibrating a timer to meet both a minimum and maximum specification (10 ms $\pm$500 μs).

NOTE:
Prior to erasure, it is necessary to program all bytes to the same level (data = 00H). See the Quick-Erase algorithm for more details.

Erase Verify Command—A0H

The erase command erases all bytes of the array in parallel. After each erase operation, all bytes must be verified to see if they erased. Write the Erase Verify command (A0H) to stop erasure and setup verification.

Alternatively, you may allow the internal stop timer to halt erasure. You must still issue the Erase Verify command to set up verification.

The device latches the address to be verified on the falling edge of WE\ and the actual command on the rising edge. Wait 6 μs before reading the data at the address specified on the previous write cycle.

The flash memory applies an internally-generated reference voltage to the addressed byte. Reading 0FFH from the addressed byte in this mode indicates that all bits in the byte are erased with sufficient margin to V_{CC} and temperature fluctuations.

If the location is erased, then repeat the Erase Verify procedure for the next address location. Write the command prior to each byte verification to latch the byte's address. Continue this process for each byte in the array until a byte does not return 0FFH data, or the last address is accessed.

In the case where the data read is not 0FFH, perform another erase operation. (Refer to Set-up Erase/Erase). Continue verifying from the address of the last verified byte. Once you have accessed the last address, erasure is complete and you can proceed to program the device. Terminate the erase verify operation by writing another valid command (e.g., Program Set-up).

Set-up Program/Program Commands—40H

Write this command (40H) twice in succession to initiate programming. The first write cycle sets up the device for programming. The device latches address and data on the falling and rising edges of the second write cycle, respectively. It also begins programming on the rising edge. You stop the programming operation by issuing the Program Verify command, or by allowing the integrated program stop timer to time out. This timer works similiar to the erase stop timer. Again, a minimum specification replaces a tougher minimum/maximum combination (10 μs–25 μs).

Program Verify Command—C0H

Flash memory devices program on a byte-by-byte basis. After each programming operation, the byte just programmed must be verified. Write the Program Verify command (C0H) to stop programming and set-up verification. Should your software allow the integrated stop timer to halt programming, the software must resume the algorithm with the Program Verify command. The device executes this command on the rising edge of Write-Enable. The program Verify command stages the device for verification of the byte last programmed. No new address information is latched.

The flash memory applies an internally-generated reference voltage to the addressed byte. Wait 6 μs for the internal voltages to settle before reading the data at the address programmed. Reading valid data indicates that the byte programmed successfully.

Command Register Reset—FFH

Flash memories reset to the read mode during power-up, and remain in this mode as long as V_{PP} is less than V_{CC} + 2V. If your system leaves V_{PP} turned-on during a *system reset*, then incorporate a command register *device reset* into the hardware initialization routines. This is necessary because the CPU might be controlling programming or erasure when the system reset hits.

Write the reset command (0FFH) twice in succession to reset the device. The double write is necessary because of the state-machine reprogramming structure. For example, suppose the *system* is reset after a Set-up Program command. The flash memory state machine expects the next write cycle to contain valid address and data for programming, followed by another write cycle for program verification. The first Reset command will be mistaken for program data but will not corrupt the existing data. This is because the command (data = 0FFH) is a null condition for flash memory programming. Only data bits programmed to zero pull charge onto the memory cell and change the data. The second write cycle actually resets the device to the read function. Following the second reset cycle, you can write the next command (Read, Program Set-up, Erase Set-up, etc.).

If the V_{PP} supply is turned off upon system reset, the software reset is not required. The flash memory will reset itself automatically when V_{PP} powers down.

Data Protection on Power Transitions

The command register architecture offers another benefit in addition to simplified processor interface—during system power-up and power-down it protects data from corruption by unstable logic. Erasure or programming require V_{PP} to be greater than V_{CC} + 2V and the proper command sequence to be initiated. For example the CPU must write the erase command twice in succession. The odds of this occurring randomly are slim. Additionally, should V_{PP} ramp to 12V prior to V_{CC} ramping past 2.5V, the device will lock out all spurious writes and internally block 12V from the flash memory cells. For even greater security, you can switch V_{PP} as discussed in Section 3.13.

2.3 V_{PP} Specifications

Flash memories, like EPROMs, require a 12V externally-generated power supply for reprogramming. Intel's V_{PP} specifications 12.0V ±0.6V (5%) is compatible with most off-the-shelf (or available in-system) power supplies. (Note, Section 3.1 discusses V_{PP} generation techniques, and Appendix B shows different circuit alternatives.)

It is essential to use the specified V_{PP} when reprogramming the flash memory device. Once the command to erase, program, or verify is issued, the device internally derives the required voltages from the V_{PP} supply. The command register controls selection of internal reference circuitry tapped off of V_{PP}. An improper V_{PP} level causes the references to be wrong, degrading the performance of the part.

(When programming U.V. EPROMs, V_{CC} is raised to 6.5V. On flash memories, the V_{PP} reference circuitry and command register architecture provide the same function while keeping V_{CC} and V_{PP} at static levels. An incorrect V_{CC} level during U.V. EPROM programming poses similar hazards to improper V_{PP} levels on flash memories.)

The hardware design section discusses various methods for generating V_{PP}.

3.0 HARDWARE DESIGN FOR ISW

Covered in this section are the following:

- Description of ISW-specific functional system blocks including memory requirements
- V_{PP} generation techniques
- Communication Considerations

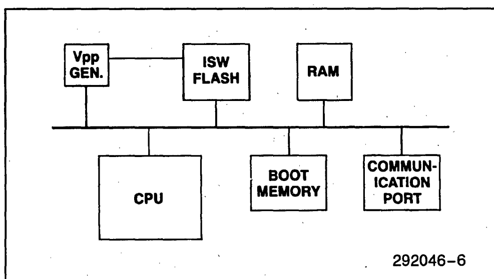


Figure 3. System Block Diagram

System Level Hardware Requirements for ISW:

- processor or controller
- limited amount EP/ROM or other flash memory devices for boot code, communications s/w, and reprogramming algorithms
- limited amount of RAM for variable storage (i.e., stacks, buffers, and other changing parameters)
- data import capability (i.e., serial line, LAN, floppy disk)
- flash memory for nonvolatile code or data storage needs
- V_{PP} generator or regulator

All of the functional blocks in Figure 3 are typical of any embedded or reprogrammable system with the exception of the V_{PP} generator. Some microcontrollers have on-chip EP/ROM, RAM and a serial port. With these devices, implementation of the ISW capability requires little additional hardware.

The next section discusses V_{PP} generation techniques and communications design considerations.

3.1 V_{PP} Generation

A static V_{PP} is needed to reprogram flash memories. The V_{PP} voltage can be generated by:

1) regulating it down from a higher voltage;
2) pumping it up from a lower voltage (i.e., charge pump, DC/DC converter, etc.); or
3) designing or specifying the system's 12V supply with the required ISW tolerances and specifications.

Sufficient current for reprogramming should be considered when selecting your V_{PP} generation option. Parallel reprogramming for flash memory in 16-bit or 32-bit systems will require, respectively, 2X or 4X additional current capability.

3.1.1 REGULATING DOWN FROM HIGHER VOLTAGE

V_{PP} is obtained from a higher voltage by using a linear regulator. Given the higher voltage, regulation offers the least expensive method of generating V_{PP}. Standard three terminal 12V ±1%, ±2%, ±4% non-adjustable regulators are available off-the-shelf. Some regulators have on/off control built-in. (See Appendix B, V_{PP} Circuit #1.) All regulators require a minimum input voltage greater than the output voltage. (See Appendix B, V_{PP} Circuit #2 and #3.)

3.1.2 PUMPING 5V UP TO 12V

V_{PP} can be obtained by pumping V_{CC} and regulating it to the proper voltage. A voltage charge-pump can be designed and built by using a charge-pump integrated circuit and some discrete components (see Appendix B, V_{PP} Circuit #4) or by using a monolithic DC/DC converter (see Appendix A, V_{PP} Circuit #5).

When using adjustable circuits containing discrete components, design the output voltage so it falls within the V_{PP} specifications for all corners of the components'

skew (i.e., V_{CC} ±10%; Rx ±1%, Ry ±1%, etc.). Include the resistors' temperature coefficients in the calculation matrix. Note that each of the various components can add error to the V_{PP} supply.

The monolithic DC/DC converter shown in Appendix B Circuit #5 fits into a 24-pin socket. It offers the advantages of close temperature tracking and ease of implementation. It has also been characterized at temperatures and meets all the V_{PP} specifications. Appendix C contains a partial list of vendors selling DC/DC converters.

Most DC/DC converters are only 50–60% efficient, so heat dissipation may be a concern. Some discrete boost circuits such as Appendix B, Circuit #4, offer much higher efficiency (70-85%). Circuit #4 as shown can supply 200 mA. Smaller inductor and capacitor component values and higher frequency boost convertors can be used where less power is required. For example, designs which reprogram one or two flash memories simultaneously might use the LT1172. (Contact Linear Technologies for more information.)

In all V_{PP} generation methods, a capacitor on the input voltage terminals reduces the output noise voltage. Some power supplies (Appendix B, Circuits #3 and #4) specify a large-valued capacitor to decrease the Effective Series Resistance (ESR). Place a 0.1 μF capacitor within 0.25 inches of each flash memory's V_{PP} input (in addition the one on the V_{PP} generator's input).

NOTE:
The ESR is inversely proportional to the capacitance value and the rated working voltage. To lower the ESR choose a capacitor with a large capacitance and a high working voltage (i.e., above 100V).

3.1.3 ABSOLUTE DATA PROTECTION— V_{PP} ON/OFF CONTROL

With V_{PP} below V_{CC} +2V or V_{CC} below 2.5V, internal circuitry disables the command register and eliminates the possibility of inadvertent erasure or programming. Switching the V_{PP} supply off provides the secondary benefits of improved power and thermal management.

There are two ways to switch V_{PP} on and off:

1) directly switch the V_{PP} generator's output, or
2) switch the input voltage supplying the regulation circuit.

Any switching circuit will cause a voltage drop, so choose a switch with this drop in mind. Some power supplies have asymmetrical tolerances on 12V (i.e. +5%, −4%). Flash memory allows the 12V supply to drop as low as −5%. The 1% difference between the supply and the device requirement allows the switch to have an ON resistance voltage drop of 0.12V. Continuing with this example, assume the system only reprograms one flash memory at a time. The current through the switch into the flash is I_{PP} = 30 mA. Solving for the allowable resistance across the switch: R = V/I = (0.12V)/(30 mA) = 4 Ohms. See Figure 4. Example Voltage Drop Across Switch. Note, one can reduce the effective R_{DS} (ON) by placing 2 or more FETs in parallel if necessary.

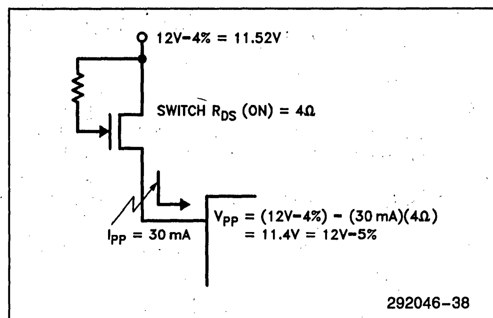


Figure 4

Controlling the input voltage of a DC/DC convertor with a MOSPOWER FET is another straightforward approach. (See Appendix B, Circuit #5.) Choose the FET switch carefully. It should have a very low on-resistance to minimize the voltage divider effect of the converter and FET switch. If the voltage across the FET switch is too high, the converter will not have the proper input voltage to meet its specifications. Always design the switching circuit with sufficient margin to maximum V_{PP} and V_{CC} load currents.

3.1.4 WRITES AND READS DURING V_{PP} TRANSITIONS

After switching V_{PP} off, the CPU can read from the flash memory without waiting for the capacitors on V_{PP} to bleed off. To do this, write the Read Memory command prior to issuing the V_{PP}_OFF instruction. Alternatively, the device resets automatically to read mode when V_{PP} drops below V_{CC} +2V.

Raising V_{PP} to 12V enables the command register. You must wait 100 ns after V_{PP} achieves its steady state value before writing to the command register. Remember that the steady state V_{PP} settling time depends on both the power supply slew rate and the capacitive load on the V_{PP} bus.

3.1.5 OTHER V_{PP} CONSIDERATIONS

The V_{PP} pin is an MOS input which can be damaged by electrostatic discharge (ESD). In OBP applications, an external power source supplies V_{PP} and then is removed. Electrostatic charge can build up on the floating V_{PP} pin. You can solve this problem by one of two means: 1) tie the pin to V_{CC} through a diode and pull-up resistor (Figure 5a) or through a resistor to ground (Figure 5b). With either approach use a 10 KΩ or larger resistor to minimize V_{PP} power consumption.

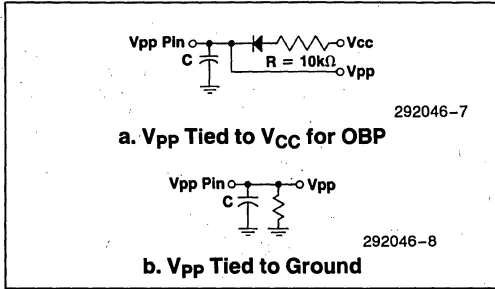


Figure 5

NOTE:
Typically EPROMs require V_{PP} to be within one diode drop of V_{CC} for optimal standby power consumption. Either approach can be used with the flash memory.

ISW applications do not require this ESD protection as most regulators and charge pumps contain a voltage divider on the output stage. A divider provides a resistive path to ground even with the supply turned off. (Note: check the schematics of the V_{PP} supply chosen.) However, if you directly switch the V_{PP} supply, add the resistor to ground; the switch isolates the V_{PP} pin and allows charge to build up.

3.1.6 V_{PP} CIRCUITRY AND TRACE LAYOUT

You should lay out V_{PP} circuitry and traces for high frequency operation since programming power characteristics exhibit an AC current component. Use the following standard power supply design rules:

- Keep leads as short as possible and use a single ground point or ground plane (a ground plane eliminates problems).
- Locate the resistor network (or a regulator) as close as possible to the adjustment pin to minimize noise pick-up in the feedback loop. The resistor divider network should also be as short as possible to minimize line loss.
- Keep all high current loops to a minimum length using copper connections that are as wide as possible. (This will decrease the inductive impedance which otherwise causes noise spikes.)
- Place the voltage regulator as close to the flash memory as practical to avoid an output ground loop. Excessive lead length results in an error voltage across the distributed line resistance.
- Separate the input capacitor return from the regulator load return line. This eliminates an input ground loop, which could result in excessive output ripple.

3.2 Communications—Getting Data to and from the Flash Memory

The flash memory does not care about the origin of the data to be programmed. The data could be downloaded from a serial link, parallel link, disk drive, or generated locally as in data accumulation applications.

While most systems communicate via serial link, sending a font to a printer's flash memory is an example of a parallel interface. In either format, designers must decide whether or not to buffer the incoming data. Error-free serial protocols will require buffering for reconstruction of information packets. With equal capacity of RAM and flash memory in a system, the download time would only be limited by the speed of the communication link.

Both worst case and typical analysis must be done for real time download and un-buffered programming. The maximum transmission rate is 19.2K baud assuming worst case programming times. The time between characters at 19.2K baud is 520 μs; the worst case byte programming time is approximately 0.5 ms (including software overhead). Typical byte programming takes 16 μs which allows for much higher unbuffered transmission rates. However, a single byte can take up to the full 400 μs specified time (plus software overhead), so you should not base transmission rate on typical programming times.

Partial buffering or FIFO schemes can also be implemented to increase transmission rates. An argument for buffering is reduction of interconnect time and costs.

4.0 SOFTWARE DESIGN FOR ISW

Covered in this section are the following software requirements:

- system integration of ISW
- reprogramming considerations for single- and multiple-flash memory based designs.

4.1 System Integration—Boot Code Requirements

Boot code in remote systems should contain various ISW-specific procedures in addition to standard initialization and diagnostic routines.

The most dependable boot code for remote version updates contains some basic communications capability and the ISW reprogramming algorithms. Thus, a datalink disruption while reprogramming would be recoverable. For manufacturing flexibility, this boot memory could be an OBP 256K flash memory.

1. Bootstrap, and reset flash memory;
2. Check "HOST_INT" and "VALID_AP" flags:

   If HOST_INT is inactive and VALID_AP = 4150H, jump to application start address;
3. If VALID_AP<>4150H, loop and wait for host (the link probably went down during update);
4. When "HOST_INT" is active, vector to host interaction code.

   (See next section.)

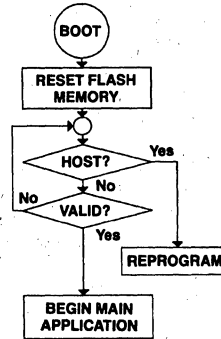


Figure 6. Example of ISW Integration to the Boot Sequence

An alternative to storing these routines in a separate boot device is storing them in the flash memory containing the program code. Prior to erasure, the CPU would transfer the ISW routines to system RAM and execute from there. This type of approach is suitable for battery-operated equipment or systems with back-up power supplies.

The communication link could be disrupted during reprogramming, leaving the device in an unknown configuration. Therefore, the boot code should reset the flash memory and check two ISW flags. The following section discusses the flag check concept.

4.1.1 ISW FLAG CHECK

After resetting the flash memories and initializing other system components, the CPU should check the communications link for a host interrupt. We will call this the HOST_INT flag. Had the communication link gone down prior to completion of downloading, then the host would have to re-establish contact to complete the task.

Assuming no HOST_INT request has been made, the boot protocol then checks a data sequence in the flash memory signifying a valid application (VALID_AP). You program this sequence into the memory array after confirmation of a successful download. If a download is interrupted midway through erasure or programming, then the VALID_AP flag locations will not contain the VALID_AP code. On the next system bootstrap the CPU recognizes this and holds up system boot until valid code is programmed. In Figure 6 an example flag protocol uses the VALID_AP sequence of 4150H (ASCII codes for "AP").

4.2 Communication Protocols and Flash Memory ISW

The remote download communications protocol must guarantee accurate transmission of flash memory instructions and program code. This protocol can be as simple as a read-back technique or as complex as an error-free transmission protocol. (See Figure 7 for possible system-level flash memory instructions.)

A simple read-back technique optimizes download for boot code memory needs and ease of implementation. The embedded CPU echoes the flash memory instruction (i.e., Erase or Program) to the host, and waits for a confirmation prior to execution. After programming the update, the remote system checks the update by transmitting it back to the host for confirmation. The remote system then programs the VALID_AP sequence. Note that programming and reading back 64 Kbytes at 19.2K baud takes about 0.57 minutes per direction:

(65,536 bytes) * (10 bits/byte) * (1 sec/19.2 Kbits) * (1 min/60 sec) = 0.57 minutes.

Implementing either software- or hardware-based error-free communications protocol improves transmission efficiency. It eliminates the possibility of errant data being programmed if not buffered and checked, and optimizes the download process for transmission time. Additionally, file compression and decompression routines can improve the transmission rate.

General ISW Instructions Include:
- STATUS CHECK
- INITIATE REPROGRAMMING
- MOVE ISW ROUTINES FROM FLASH MEMORY TO RAM (If not resident in separate boot memory)

Data accumulation-specific commands include:
- RETRIEVE DATA
- ERASE FLASH MEMORY

Figure 7. Sample System-Level ISW Instruction Set

Status Check

The host should request a status update from the remote system prior to sending a reprogramming instruction. Depending on the response, the host may break the link and reconnect later, or it may send an erasure or data-upload command. This type of handshaking is necessary when system downtime for reprogramming might not be acceptable. An example of this is an automated factory where robots handle caustic chemicals.

4.3 Data Accumulation Software Techniques

Data can be accumulated in a remote environment with flash memory and then uploaded to a host computer for manipulation. You can adapt various standard data-logging techniques for use with flash memory. With any technique, you determine the next available memory location by reading for erased data (0FFH). This address would only be located once on system bootstrap and then recalled from RAM and incremented as needed.

Given a repeating data string of known length and composition, program start and stop codes at either end of the string. Do not pick 00H or 0FFH data for these codes because they are used during erasure. The start and stop codes enable the CPU to differentiate between available memory for logging and logged data equal to 00H or 0FFH.

For non-regular data input, you can address this same issue by programming the logged data followed by the variable identifier. Again, do not pick 00H or 0FFH data for the variable identifiers.

With any technique, the host computer separates and manipulates the data after the uploading operation.

4.4 Reprogramming Routines

Intel's ETOX flash memories provide a cost-effective updatable, non-volatile code storage medium. The reliability and operation of the device is based on the use of specified erasure and programming algorithms.

Intel offers reprogramming software drivers to make it easy for you to design and implement flash memory applications. The software is designed around the CPU-family architectures and requires minimal modification to define your system parameters. For example, you supply the memory width (8-bit, 16-bit, or 32-bit), system timing, and a subroutine for control of V_{PP}.

NOTE:
Contact your nearest sales office for details.

If you prefer to implement the algorithms yourself, they are outlined in the device data sheets. Command register instructions required for the various operations are included in the data sheet flow charts.

The following sections describe both single-device and multiple-device parallel reprogramming implementations.

4.4.1 Quick-Erase Algorithm

Flash memories chip-erase all bits in the array in parallel. The erase time depends on the V_{PP} voltage level (11.4V–12.6V), temperature, and number of erase/write cycles on the part. See the device data sheets for specific parametric influences on reprogramming times.

Note that prior to erasing a flash memory device the processor must program all locations to 00H. This equalizes the charge on all memory cells insuring uniform and reliable erasure.

3

Algorithm Timing Delays

The Quick-Erase algorithm has three different time delays:

1) The first is an assumed delay when V_{PP} first turns on. The capacitors on the V_{PP} bus cause an RC ramp. After switching on V_{PP}, the delay required is proportional to the number of flash memory devices times 0.1 μF/device. V_{PP} must reach its final value 100 ns before the CPU writes to the command register. Systems that hardwire V_{PP} to the device can eliminate this delay.

2) The second delay is the "Time Out TEW" function, where TEW is the erase timing width. The function occurs after writing the erase command (the second time) and before writing the erase-verify command. The erase-verify command or the integrated stop timer internally stops erasure.

   TEW for ETOX II flash memories is a minimum of 10 ms. This delay can be either software or hardware controlled. Either way, the minimum nature of the timing specification allows for interrupt-driven timeout routines. Should the interrupt latency be longer than the minimum delay specification, the stop timer halts erasure.

3) The third delay in the erase algorithm is a 6 μs time out between writing the erase verify command and reading for 0FFH. During this delay, the internal voltages of the memory array are changing from the

erase levels to the verify levels. A read attempt prior to waiting 6 μs will give false data—it will appear that the chip does not erase. Repeatedly trying to erase verify the device without waiting 6 μs will cause over-erasure. This delay is short enough that it is best handled with software timing. Again, note that the delay specification is a minimum.

High Performance Parallel Device Erasure

In applications containing more than one flash memory, you can erase each device serially or you can reduce total erase time by implementing a parallel erase algorithm.[7] You save time by erasing all devices at the same time. However, since flash memories may erase at different rates, you must verify each device separately. This can be done in a word-wise fashion with the command register Reset command and a special masking algorithm.

Take for example the case of two-device (parallel) erasure. The CPU first writes the data word erase command 2020h twice in succession. This starts erasure. After 10 ms, the CPU writes the data word verify command A0A0h to stop erasure and setup erase verification. If both bytes are erased at the given address, then the CPU increments the address (by 2) and then writes the verify command A0A0h again. If neither byte is erased, then the CPU issues the erase sequence again without incrementing the address.

Suppose at the given address only the low byte verifies FFh data? Could the whole chip be erased? The answer is yes. Rather than check the rest of the low byte addresses independently of the high byte, simply use the reset command to mask the low byte from erasure and erase verification on the next erase loop. In this example the erase command would be 20FFh and the verify command would be A0FFh. Once the high byte verifies at that address, the CPU modifies the command back to the default 2020h and A0A0h, increments the address by 2, and writes the verify command to the next address.

See Figure 8 for a conceptual view of the parallel erase flow chart and Appendix D for the detailed version. These flow charts are for 16-bit systems and can be expanded for 32-bit designs.

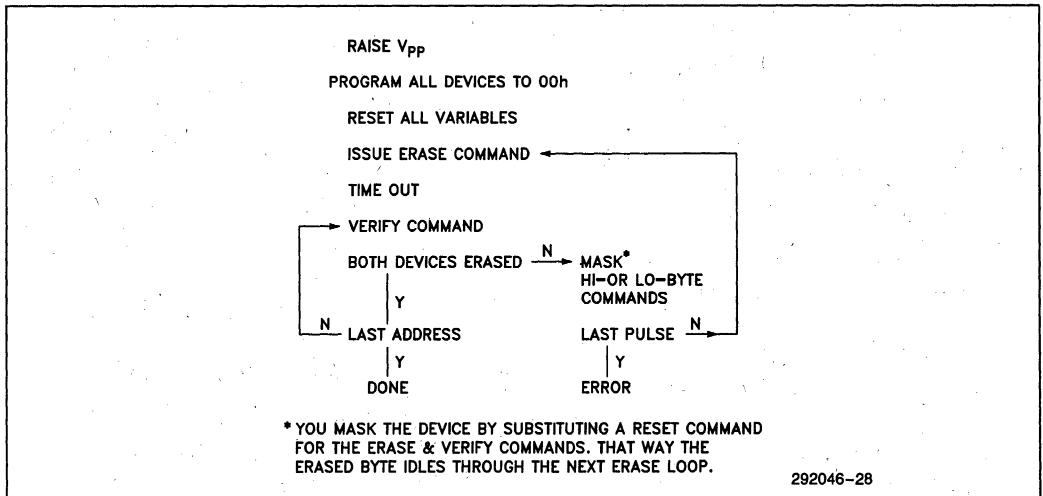


Figure 8. High Performance Parallel Erasure (Conceptual Overview)

7. Parallel Erasure and Programming require appropriate choice of V_{PP} supply to support the increased power consumption.

4.4.2 Quick-Pulse Programming Algorithm

Flash memories program with a modified version of the Quick-Pulse Programming algorithm used for U.V. EPROMs. It is an optimized closed-loop flow consisting of 10 μs program pulses followed by byte verification. Most bytes verify after the first pulse, although some may require more passes through the pulse/verify loop. As with U.V. EPROMs, this algorithm guarantees a minimum of ten years data retention. See the device data sheets for more details on the programming algorithm.

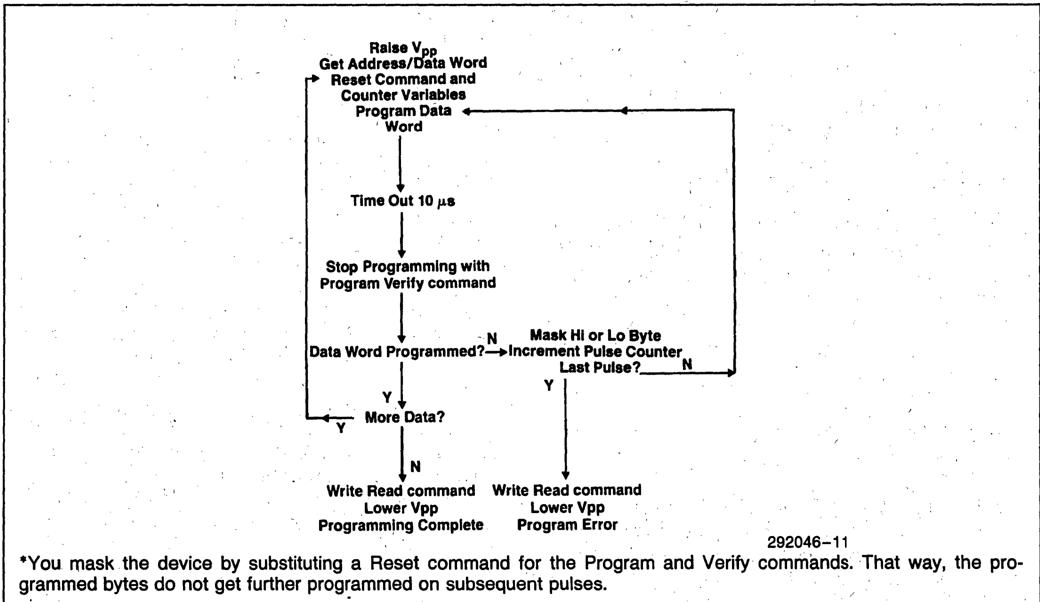
Algorithm Timing Delays

The Quick-Pulse Programming algorithm has three different time delays:

- The first and third—V_{PP} set-up and verify set-up delays—are the same as discussed in the erasure section. In this case the third delay is for the transition between writing the Program Verify command and reading for valid data.
- The second delay is the "Time Out 10 μs" function, which occurs after writing the data and before writing the program-verify command. This write command internally stops programming. The section entitled "Pulse Width Timing Techniques" gives 86-family assembly code for generating a 10 μs timer routine.

High Performance Parallel Device Programming

Software for word- or double-word programming can be written in two different manners. The first method offers simplicity of design and minimizes software overhead by using a byte programming routine on each device independently. Here you increment the address by 2 or 4 when addressing 1 of 2 or 4 devices, respectively. The second method offers higher performance by programming the word or double-word data in parallel. This method manipulates the command register instructions for independent byte control. See Figure 9 for conceptual 2-device parallel programming flow chart and Appendix E for the detailed version.



*You mask the device by substituting a Reset command for the Program and Verify commands. That way, the programmed bytes do not get further programmed on subsequent pulses.

Figure 9. Parallel Programming Flow Chart (Conceptual Overview)

NOTE:
Word or double-word programming assumes 2 or 4 8-bit flash memory devices.

Parallel Programming Algorithm Summary:

- Decreases programming time by programming 2 flash memories (16 bits) in parallel. The algorithm can be expanded for 32-bit systems.
- Eliminates tracking of high/low byte addresses and respective number of program pulses by directing the CPU to write data-words (16-bit) to the command register.
- Maintains word write and word read operations. Should a byte on one device program prior to a byte on the other, the CPU continues to write word-commands to both devices. However, it deselects the verified byte with software commands. An alternative is to independently program high and low bytes using hardware select capability.

4.4.3 Pulse Width Timing Techniques

Software or hardware methods can be used to generate the timing required for erasure and programming. With either method you should use an in-circuit emulator (ICE™) and an oscilloscope to verify proper timing. Also remove the flash memory device from the system during initial algorithm testing.

Software Methods and Examples

Software loops are easily constructed using a number of techniques. Timing loops need to be done in assembly language so that the number of clock cycles can be obtained from the instructions.

In order to calculate a delay loop three things are needed—

1) processor clock speed,
2) clock cycles per instruction, and
3) the duration of the delay loop.

As an example, the 80C186 divides the input clock by 2. With a 20 MHz input clock the processor's internal clock runs at 10 MHz. This translates to a 100 ns cycle time. Delays can be made by loading the CX register with a count and using the LOOP instruction. The LOOP instruction takes 16 clock cycles to execute per pass. It decrements the CX register on each pass and jumps to the specified operand until CX equals zero.

When writing a delay loop consider all instructions between the start and end of the delay. If a macro is written that delays 10 μs, add the clock cycles for all instructions in the macro.

Here is an example of a 10 μs delay and the calculation of the constant required for a 10 MHz 80C186.

```
WAIT_10 µs:
        push cx             ;10 clock cycles
        mov cx,DELAY        ;4 clock cycles
        loop $              ;see calculation
        pop cx              ;10 clock cycles

1. Start to End       = 10 µs/cycle time
                      = 10 µs/100 ns
                      = 100 cycles

2. Loop Instruction   = 100-24 cycles
                      = 76 cycles

3. Loop Cycles        = 76
                      = (15 × [DELAY - 1] + 5)

4. Solving for DELAY = 6
```

Hardware Methods

Using an Internal Timer—

Many microcontrollers and some microprocessors have on-chip timers. At higher input clock speeds these internal timers have a resolution of 1 μs or better. The timers are loaded with a count and then enabled. The timer starts counting and when it reaches the terminal count a bit is set. The CPU executes a polling algorithm that checks the timer status. Alternatively, a timer-controlled interrupt can be used. After the timer has been set and the interrupt enabled, the CPU can be programmed to wait in idle mode or it could continue executing until the timed interrupt.

One thing to take into account when using interrupts is the time required for the CPU to recognize and interrupt request (interrupt latency). This is important when figuring the timer value, because the time seen by the part will be the programmed delay plus the minmum interrupt latency time.

The 80C186 has three 16-bit timers on-chip. Timer #2 can be a prescaler for the other two timers, which extends timers #0 and #1 range out to 2^32. By using two timers, 10 μs pulses and 10 ms pulses can be easily achieved.

Using an External Timer—

External timers can take many forms. One popular example is the 82C54 (CHMOS Programmable Interval Timer) which has three 16-bit timers on-chip. One timer can be used as a prescaler for the others so that a count of 2^32 can be achieved as with the 80C186 internal timers.

5.0 SYSTEM DESIGN EXAMPLE: AN 80C186 DESIGN

A general purpose controller and/or data acquisition system was built to demonstrate 86-based ISW. The 80C186 CPU drives the system, which contains 16 Kbytes of EPROM (two 27C64's), 64 Kbytes of flash memory (two 28F256A's), 64 Kbytes of SRAM (two 32K x 8's) three 8-bit ports (82C55A), one serial port (82510), and a 5V to 12.0V DC/DC converter. Three 74HC573's demultiplex the address/data bus and latch the byte high enable line (BHE) and the status lines (if needed). Two data transceivers (74HC245) simulate the worst case data path for a system requiring added drive capability. If the transceivers are not needed they can be replaced with wired headers. See Appendix F for detailed schematics parts list, and changes for the 28F512 or 28F010.

The 80C186 reset (output) drives the reset input on the 82510, 82C55A, and the OE\ inputs on the address latches and data transceivers. The reset line goes inactive 5 clock cycles before the first code fetch. Also, the CPU's write signal is split into byte-write-high and byte-write-low to allow for byte or word writes.

The 80C186 has on-chip memory and peripheral chip selects. Two of the memory chip selects are dedicated. One is the Upper Chip Select (UCS, dedicated for the boot area) and the second is the Lower Chip Select (LCS, for the interrupt vector table area). See the memory map in Figure 10.

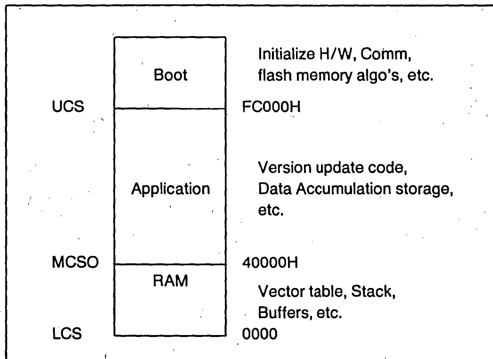


Figure 10. 80C186 Memory Map

The permanent code was placed in an EPROM in the UCS memory segment; this code includes routines for hardware initialization, communications, data uploading and downloading, erasure and programming algorithms, I/O drivers, ASCII to binary conversion tables, etc. This would be useful for systems reconfigured for different communication protocols as the last step prior to shipment.

Code and constants that might change are placed in the 64 Kbytes of flash memory. Application examples include operating systems, code for rapidly advancing biomedical technologies such as blood test software, engine-control code and parameters, character fonts for printers, postage rates, etc. The RAM is used for the interrupt table, stack, variable data storage, and buffers.

The three 8-bit ports on the 82C55A peripheral controller can be used for control and/or data acquisition. It powers-up with all port pins high. Similarly, all port pins go high after warm resets as well. Because the pins are high after a power-up/reset, an open collector invertor was used to control the MOSPOWER switch which in turn controls V_{PP}. You must drive the FET switch to one rail or the other to guarantee its low on-resistance. V_{PP} is turned off during power-up or reset as a hardware write protection solution. The DC/DC converter supplies V_{PP}.

The 82510 is a flexible single channel CHMOS UART offering high integration. The device off-loads the system and CPU of many tasks associated with asynchronous serial communications.

The part can be used as a basic serial port for the host serial link, or can be configured to support high speed modem applications. For more information on the 82510 see the 82510 data sheet and AP-401 "Designing with the 82510 Asynchronous Serial Controller".

Software was written to download code and data parameters (code updates) from a PC to the demo board through the PC's COM1 port (serial port). The system also can upload data (remote data acquisition) to the PC via the same link.

Once the download code and data has been programmed it can not be lost, even if power should fail. This is because Intel's ETOX II flash memory technology is based on EPROM technology and does not need power to retain data.

The end result: rugged, solid state, low power nonvolatile storage.

6.0 SUMMARY

Intel's flash memories offer designers cost-effective alternatives for remote version updates or for reliable data accumulation in the field or factory. Designers will also benefit from time savings in any kind of code development—no 15 minute waits for U.V. EPROM erasure.

This application note covers the basics of in-system writing to flash memories and can be used as a check list for systems other than the 80C186 design shown. The basic concepts remain the same: a CPU controls the reprogramming operations; a 12V supply must be applied to the flash memory for erasure and programming; and a communications link connects the host to the remote system and supplies the code to be programmed.

APPENDIX A
ON-BOARD PROGRAMMING DESIGN CONSIDERATIONS

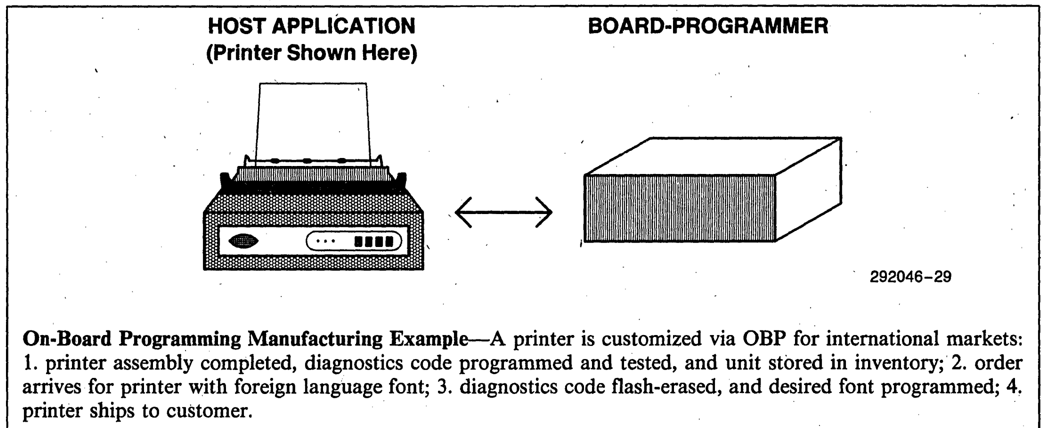
INTRODUCTION

On-board programming[1] (OBP) with Intel's flash memory provides designers with cost reduction capabilities for alterable code storage designs. When used in conjunction with on-board programming, flash memory presents opportunities for savings in two areas: greater testability in the factory, which translates to improved outgoing quality and reduced return rate; and quicker, more reliable field updates, which translates to decreased product support cost.

This appendix:

- outlines the design considerations associated with on-board programming, and the improvements afforded by Intel's flash memory;
- offers guidelines for converting current 64K EPROM OBP designs;
- designs an 8-bit system for on-board programming;
- suggests some 16-bit flash design considerations; and offers information on OBP equipment and vendors.

1. With on-board programming, non-volatile memory is programmed while socketed or soldered on the application board, rather than before hand as a discrete component. This programming method is also called in-module or in-circuit programming, and has been practiced by some major corporations since 1981. See sidebar on following pages for more information on U.V. EPROM OBP usage.



On-Board Programming Manufacturing Example—A printer is customized via OBP for international markets: 1. printer assembly completed, diagnostics code programmed and tested, and unit stored in inventory; 2. order arrives for printer with foreign language font; 3. diagnostics code flash-erased, and desired font programmed; 4. printer ships to customer.

INTEL'S FLASH MEMORY—DESIGNED TO MEET YOUR OBP NEEDS

Intel's flash memory simplifies OBP code updates by offering designers the command register architecture. As described in section 2.2, this architecture offers the full reliability of EPROM off-board programming without the hassles of elevating V_{CC}.

5 Volt Vcc Erasure and Programming Verification

Unlike EPROM OBP, flash memory enables Vcc to remain at 5.0V throughout all operations. Internal circuitry derives the erasure and programming verification levels from the voltage on Vpp rather than from Vcc. These verify modes enable use of a single Vcc bus for the entire board, as opposed to the two buses needed for U.V. EPROM OBP. (See sidebar entitled EPROM OBP).

EPROM OBP

EPROM OBP has been a proven manufacturing technique since 1981. Ingenuity and clever circuit design have enabled manufacturers to overcome the hurdles associated with OBP and enjoy the benefits.

In many cases, Intel's flash memory simplifies today's solutions and offers new capabilities to advance the state of OBP technology. The following paragraphs outline the hurdles and a few of the solutions in use today.

EPROMs require program verification at an elevated V_{CC} to insure long-term data retention. PROM programmers easily accommodate this requirement, and it is generally invisible to the end-user.

REPLACING CURRENT EPROM OBP DESIGNS WITH FLASH MEMORY

Hardware Considerations

A slight hardware modification is required to adapt most of the current EPROM OBP designs for use with Intel's flash memory. Simply convert the EPROM memory sites from 28 to 32 pins. All other board-design cirteria used for EPROM OBP apply to flash memory as well. (For discussions of these criteria see section entitled New OBP Designs).

Standard EPROM OBP requires the board designer to bus $\overline{PGM}$ to the edge connector. With flash memories' command register architecture, this same trace enables electrical erasure and programming, only now the line is called Write Enable ($\overline{WE}$). The timing for $\overline{WE}$ is similar to that of read accesses, although that is handled via software changes.

Another potential hardware change is on the board programmer side of the design–the V_{PP} supply. Many EPROMs program with 12.5-13.0V V_{PP} supplies. Intel's ETOX II flash memory requires 11.4-12.6V V_{PP}. This change should not be an issue since the V_{PP} supply on many board programmers is programmable.

Mixed memory systems containing both conventional U.V. EPROM and flash memories require special consideration. This type of memory design requires separation of the Chip Enable ($\overline{CE}$) control lines between the EPROM and flash devices to allow for independent reprogramming control and access. The $\overline{PGM}$ and $\overline{WE}$ lines can be common if the board programmer can give the appropriate timings to either type of device.

Software Considerations

Manufacturers who program EPROMs on-board today will need new board-programmer software to take advantage of flash memory's feature set, specifically software for the Quick-Erase and Quick-Pulse Programming algorithms.

Benefits of Converting Your EPROM OBP Design to Flash

The most pressing reason to convert from a standard EPROM to flash memory is the total cost savings. To appreciate this, you must consider your way of doing business at the board and system levels—from the factory to installation and repair in the field. In the factory, boards can be tested with a diagnostics program in the flash memory and then erased and reconfigured for shipment in the same step. Improved testing will decrease the probability of field failures and costly customer returns. Simplified test and rework methods will decrease your inventory holding costs. Also, if in the process of converting to flash memory you include the ability to OBP via a cable-connector, service calls for code updates will be quicker, more reliable, and cost less money. Your serviceman would simply connect the programming equipment to the system without dismantling it to remove the EPROMs. (See section entitled The System/Board-Programmer H/W Connection for details.)

EPROM OBP (cont'd)

With OBP, the EPROM board-programmer handles the elevated-V_{CC} requirement easily as well. However, when V_{CC} is greater than 5V, logic devices populating the same board may draw excessive current and not operate predictably.

One solution to this issue involves running separate V_{CC} traces to the board's edge connector—one for EPROM programming, and one for powering up the rest of the board.

A second consideration when designing for EPROM OBP has been accessing manufacturer and device codes.

The identifier mode requires forcing A9 to 12V. This translates to adding extra isolation, which implies the increased costs of buffers and extra board space.

NEW OBP DESIGNS

Design Considerations

As with EPROM in-circuit programming, flash memory board programming requires the use of a board-programmer. Unlike U.V. erasure for standard EPROM OBP, electrical erasure enables flash memory OBP without removing the board from the system.

We will look at designing a board that is to remain powered-up in the system during erasure and reprogramming. The key concept is to design the board in such a way that the programmer can take control of the system during code updates. The implementation of such a design is straightforward, easy, and suited to automated production assembly.

Taking Control

The board-programmer needs to take control of the system's address bus, data bus, control lines, etc. to update the code without damaging the system. (See Figure 2. System to Board-Programmer Interface.) Taking control simply means isolating the rest of the system from these lines.

Various methods of isolating the memory from the system include using tristate buffers, latches, or even the capabilities designed into microprocessors (μP) and microcontrollers (μC). For example, Intel's 86-based μP family has HLD/HLDA signals that were set-up for multiprocessor system designs where bus control is a major concern. The HLD signal, when acknowledged, tristates the address, data, and control lines. Although not designed for multiprocessor environments, Intel's MCS®-51 and MCS-96 microcontroller families have Reset capabilities to help simplify this same task.

One issue to be aware of when using a CPU's reset control function is that it may switch from the reset to active condition at a non-standard logic level. This only presents a problem if the address/data buffer takes longer to activate than the CPU, and the CPU attempts to fetch code from a memory device isolated from it.

One approach to insure successful programming takeover (i.e. without bus contention) is to have the board-programmer's lines in a high impedance state during connection to the system. Once connection to the system has been secured, the serviceman could hit a button on the board-programmer to start the system takeover procedure. Then when total control has been established, the programmer would commence with erasure and reprogramming.

Aside from the flash device's isolation from the system, various CPU control lines ($\overline{MEMRD}$, $\overline{WE}$, $\overline{PSEN}$, etc.) may need isolation as well. If active during Reset, these lines may put the CPU into an unspecified state. When designing a board for OBP, check the μC/μP data sheets carefully for any special reset conditions.

Printed Circuit Board Guidelines for V_{CC} and V_{PP}

Programming conventional EPROM and flash memories takes 30 mA of current on V_{CC} and V_{PP}, due to the nature of hot-electron injection. Most of the charge transfers to the memory cell's floating gate in a short current spike during the first pulse. You should design both the V_{CC} and V_{PP} traces with A.C. current spikes in mind. Wherever possible, limit the inductance by widening the two traces. Bypass capacitors (0.1 μF) should be placed as close as possible to the memory device's V_{CC} and GND pins, as well as the devices V_{PP} and GND pins. The capacitor on Vcc decreases the power supply droop. The capacitor on V_{PP} supplies added charge, and filters and protects the memory from high frequency over-voltage spikes[2].

2. For a complete discussion of electrical noise, grounds, power supply distribution and decoupling see Ap-74—High Speed Memory System Design Using the 2147H, and AP-125—Designing Microcontroller Systems for Electrically Noisy Environments.

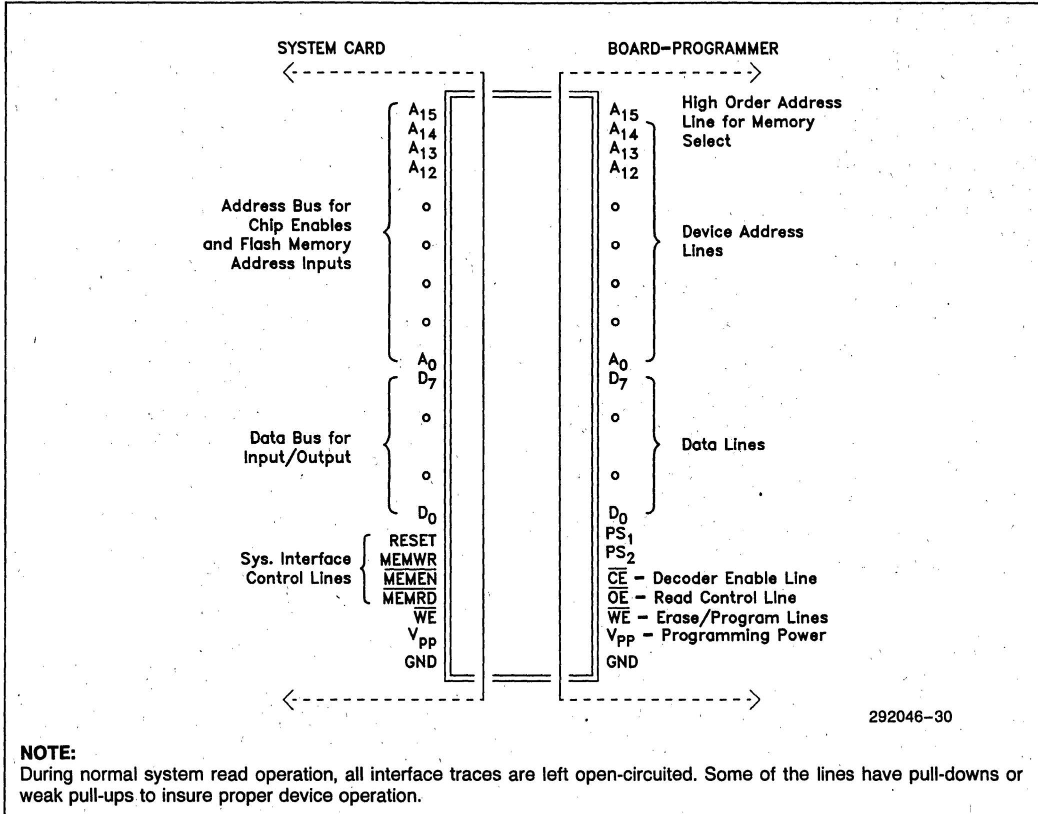
EPROM OBP (cont'd)

Some users of OBP get around this issue by programming all EPROMs with a common algorithm. However, this practice compromises the device's reliability, and should not be done.

A better solution than ignoring the identifier is to choose a qualified EPROM vendor and program with its algorithm only.

One subtle concern with EPROM OBP that designers often overlook is U.V. board erasure.

→ U.V. EPROM board erasure requires removal of the board from its host system. This incurs the hidden costs of labor, lower yields due to handling, and the reliability risks of dismantling a system. Flash memory decreases these costs by enabling a greater degree of factory automation, and increases the flexibility afforded by OBP.



NOTE:
During normal system read operation, all interface traces are left open-circuited. Some of the lines have pull-downs or weak pull-ups to insure proper device operation.

Figure 2. System to Board-Programmer Interface

EPROM OBP (cont'd)

→ Special U.V. board erasers must be purchased, at significant costs and with limited throughput. A low-end U.V. bulb costs $75-$100 each. A U.V. board eraser system could cost upwards of $10,000, with recurring costs of light bulbs and energy. Thus, the cost of U.V. erasure is often under-estimated.

→ Although portable board programmers are commercially available, U.V. lights by nature are not very rugged, and are not suited for out-of-factory code updates. This complicates field service.

→ Erasure can be easily controlled in a lab environment; however, it is not as clear on the manufacturing floor which label to remove for U.V. erasure, because parts other than EPROMs have windows (i.e. EPLD's, micro-controllers with embedded EPROM memory, etc.)

The System/Board-Programmer Hardware Connection

In most U.V. EPROM OBP applications, designers use the board's edge-connector as the programmer interface. This approach is the lowest cost solution for standard EPROM technology because U.V. erasable devices require system disassembly for erasure anyway. With flash memory, you can eliminate the system dismantling and capitalize on the erase feature by adding a cable connector to the board for reprogramming purposes. The connector should extend from the board through the system's chassis, and should be easy to reach by a serviceman.

Various types of cables exist on the market that could be used to connect programming equipment to the system. The key design consideration when choosing the type of cable is elimination of all transient noise that would interfere with the programming or erasure process.

Three types of noise interference and methods to diminish the noise are as follows:

1. line to line cross-talk (due to board-programmer's drivers that drive sharp step functions on adjacent address lines); solved with either ribbon cables, having alternate lines grounded, or with braided twisted-pairs that have a ground line for each active signal;
2. programmer line-driver-to-board impedance mismatches leading to transmission line effects of signal reflection, and interference; solved by limiting cable length, decreasing programmer switching speed (or allowing longer settling time between address switches) or by using matched line drivers on the programmer and high impedance buffers on the board end, or by using series termination resistors on the driving end of the cable (i.e.—board-programmer end, with the exception of the bi-directional data bus which needs series resistors at both ends);
3. rf pick-up in electrically noisy environments; use either shielded cable such as coax, ribbon cable with solid copper ground plane, or a new type that has recently become available called Flex cable.

Braided twisted-pair cables when kept under three feet in length generally reduce cross-talk to acceptable levels. This type of cable offers the most cost-effective solution which works well in most applications. Depending on the environment, the programmer and your design, you may need a combination of solutions, such as braided twisted-pairs with series termination.

At first all of these alternatives may seem expensive or superfluous, but keep in mind that the cost of a single cable and programmer gets amortized over the total number of systems programmed.

AN 8-BIT BUS DESIGN EXAMPLE

An example of an in-circuit reprogrammable controller board is an 80C31, two 28F256A's and some glue chips. (See Figure 3. for a system block diagram. See Appendix A. for a detailed system schematic.)[3] The important issues for erasure and reprogramming are as follows:

1. the board-programmer must have uncontested access and control of the flash memory array; and
2. the microcontroller must be reset (un-active) during the erasure and programming cycles.

SYSTEM DESIGN

Bus Control Circuitry

The 80C31 has an active-high reset pin, which tristates the address and data bases. Route this line (RESET) to the programming connector. Tie the $\overline{\text{OE}}$ pins on the low-order address latch (74HCT573), and the $\overline{\text{PSEN}}$ buffer-enable (74HCT125)[4] together, and route that line MEMWR[5] to another pin on the programmer-interface connector.

During normal system operations when the μC reads program code from the 28F256 devices, the pull-down on MEMWR keeps the address latches and $\overline{\text{PSEN}}$ buffer active. During flash memory OBP, the board-programmer drives MEMWR active-high, which disables these outputs, and isolates the address bus and $\overline{\text{PSEN}}$ from the programming signals.

The board-programmer must independently control the RESET and MEMWR traces because they disable at different V_{IL} values (2.5V for RESET vs 0.8V for MEMWR). If controlled by the same 5V supply, on power-up or after a reset condition the μC would try to execute code while still isolated from its code source—specifically before the address latches and $\overline{\text{PSEN}}$ buffer activate.

Address Decode Circuitry

This design shows two 28F256A flash memories. Systems with more than one memory device typically decode the CPU's high-order address to select a particular device.

3. Note that the flow-through latch on the data bus is not needed with the 80C31, but is drawn as an example for CPU's that can not tristate their data bus.
4. The isolation buffer is required on $\overline{\text{PSEN}}$ in this design because the 80C31 goes into unspecified states when the Reset and $\overline{\text{PSEN}}$ lines are active simultaneously. To avoid any possible problems, buffer $\overline{\text{PSEN}}$.
5. MEMWR = > bus isolation control of $\overline{\text{PSEN}}$ and the data bus.

Figure 3. System Block Diagram

This is accomplished as illustrated. When A15 is low, the lower 32K bytes are selected. The output of the inverter drives the other 28F256A's chip enable. This type of memory architecture promotes power savings by disabling all memories but the one being addressed.

To accomplish this two-line memory control architecture, route the inverter's input A15 to the 80C31 and to the programmer interface connector.[8] The board-programmer controls the inverter's output enable with $\overline{\text{MEMEN}}$.[9] The $\overline{\text{MEMEN}}$ line performs the function normally performed by $\overline{\text{CE}}$ in component programming. When driven to a logic "1" level $\overline{\text{MEMEN}}$ pulls the inverter's output high. This deselects all memory devices controlled by that I.C. During normal read and standby operations, the pull-down on $\overline{\text{MEMEN}}$ keeps the decoder enabled.

Erasure and Programming Control Circuitry

In this design, V_{PP} and $\overline{\text{WE}}$ are active only during reprogramming. At other times, the two inputs would be inactive. Simply tie the $\overline{\text{WE}}$ line to V_{CC} through a pull-up resistor. The pull-up limits the current to the board programmer during reprogramming. (Recall that $\overline{\text{WE}}$ is active low.) Flash memories allow V_{PP} to be at 12V, V_{CC} or ground for read operations. This design ties V_{PP} to V_{CC} through a diode and resistor to allow for EPROM OBP compatibility. If this option is not required, simply tie V_{PP} to ground through a current-limiting pull-down resistor.

Returning Control to the Host System

The board-programmer should return system-control to the host processor in an organized manner. First it should lower V_{PP} from 12V to 5V, or ground. Then the board programmer should place its address and data buses into a high impedance state. Next PS2, which controls MEMWR should be tristated thus disabling the $\overline{\text{PSEN}}$/Address latch isolation. Finally the board-programmer should switch PS1, which drives the RESET line to reactivate the μC. This sequence guarantees that the μC will begin operation at a known program code location.

16-BIT BUS DESIGN CONSIDERATIONS

An example of an On-Board programmable 16-bit system board would be an 80C186 microprocessor, two 28F010 flash memories, RAM, and some glue chips. The basic hardware design considerations would be the same as those in the previously discussed 8-bit bus example.

There are a few issues with 16-bit designs that do not arise in 8-bit designs. For the programmer to take control of the system, it must tristate and reset the μP as well as tristate the bus buffers and latches. The HOLD and RESET lines of Intel's 86-based family of microprocessors have been designed with bus isolation in mind for use in multiprocessor systems.

The designer has two options for erasing and programming the high and low bytes of the flash memory array independently.

1) The designer can route two $\overline{\text{WE}}$ lines to the programmer connector—BYTE HIGH $\overline{\text{WE}}$ and BYTE LOW $\overline{\text{WE}}$.

2) The reprogramming software can follow the masking procedure shown in section 4.4. This method allows a common $\overline{\text{WE}}$ line for the high and low bytes.

8. Note the lack of isolation buffers between the 80C31's high order addresses (Port 2) and the board-programmer interface, compared to the latch separating the low order addresses (Port 0) and the interface. In this design example, we make use of the 80C31's ability to tristate these ports, so no isolation is needed for any of the addresses. The latch on Port 0 is for the time-multiplexed address/data architecture of this microcontroller, and not specifically for isolation.
9. MEMEN = memory enable, active low.

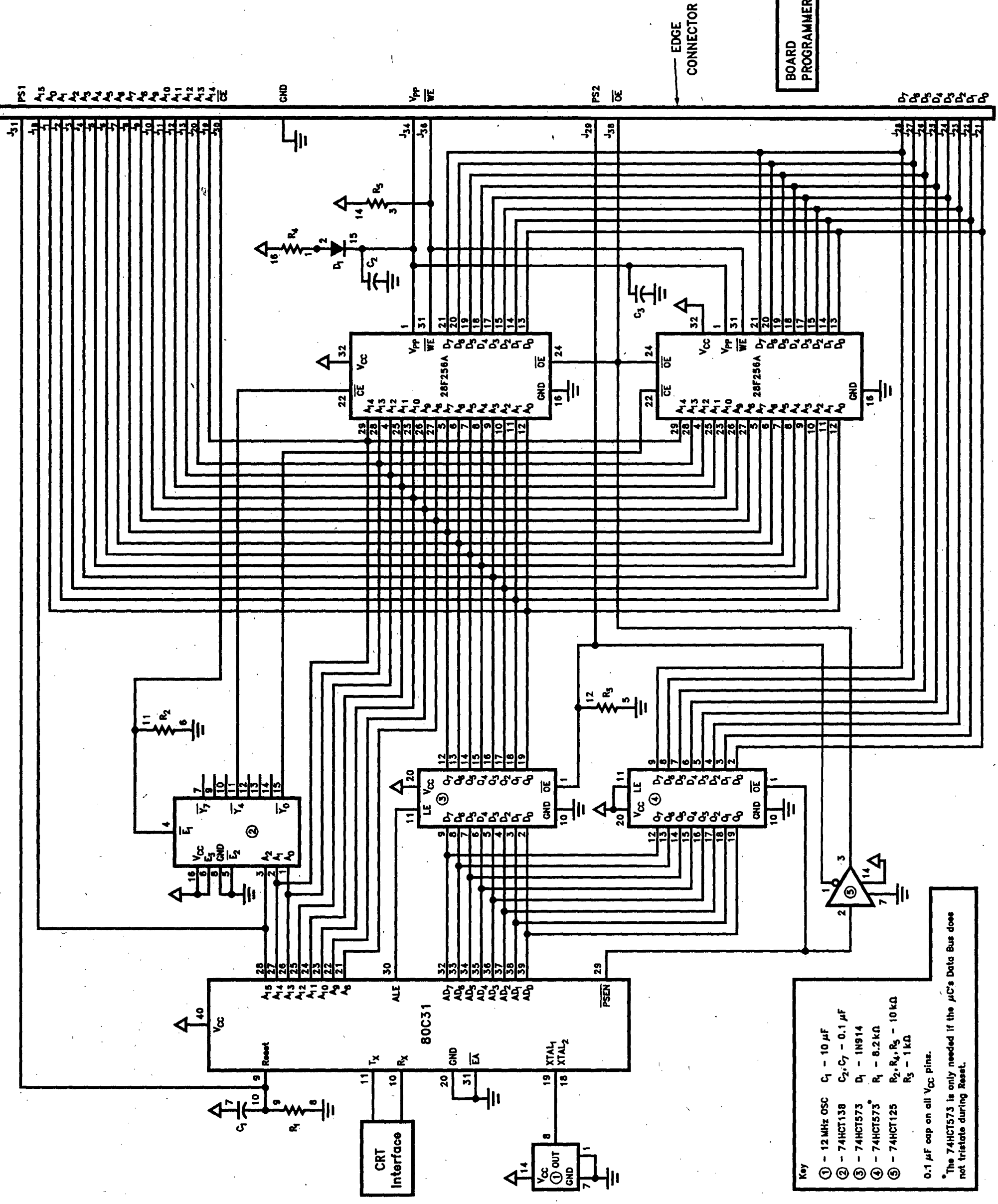


Figure 4. Detailed 8-Bit Bus Design Schematic

OBP EQUIPMENT AND VENDORS

If you are considering OBP for your next design, and have not used on-board programming before, you will need to choose a board-programmer vendor. Various suppliers offer OBP systems; therefore, it is well worth it to send out requests for programming support bids. If your production volume justifies the purchase of more than one board-programmer, you may want to negotiate a non-recurring engineering charge for development cost, followed by variable costs for additional units.

Most vendors offer a variety of basic systems, designed to easily adapt to your needs. Systems can be purchased that program either single boards serially, or a number of boards in parallel. Light-weight OBP equipment designed for field reprogramming can also be obtained from some of the vendors.

Most companies will work directly with you at the beginning of your design phase to ensure OBP compatibility. If your design is beyond the definition stage, the programmer manufacturer will request a copy of your schematics or block diagrams under non-disclosure. The vendor has an OBP design specialist that will check the design for OBP compatibility. Any potential problems will be located and corrected at this early stage.

Every board's architecture is different (i.e., based on different central processing units (CPU), decoding schemes, buffering methodologies, interface connectors, and types and densities of memories). Vendors write custom software modules for each application. Also, the vendor or the board designer typically builds an interface jig to connect the board's edge connector to the programmer. This choice is often left as a decision for the designer.

Partial List* of Companies Selling Board-Programmers

Following are a few of the companies who offer on-board programming solutions today:

Data I/O Corp.
Digelec
Elan Digital Systems
Oliver Advanced Engineering, Inc.
Stag Microsystems, Inc.

*This list is intended for example only, and in no way represents all companies that support on-board programming. Intel Corporation assumes no responsibility for circuity other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

SUMMARY

- On-board programming (OBP) has been around since 1981.
- Designing a board for OBP can be easily done by working with a board-programmer vendor's OBP-design-specialist during the initial design phase.
- In-circuit alterable code storage can be easily implemented by using flash memory and it's features.
- Time and money savings can be realized in a number of ways by taking advantage of flash memory OBP:

< > Decreased board costs and improved reliability from elimination of EPROM sockets;

< > Decreased manufacturing costs from elimination of board eraser depreciation costs, recurring U.V. light bulb and energy expenses;

< > Decreased inventory expense from simplified test and rework methods (one-step diagnostics, erasure, and board configuration);

< > Decreased product costs based on decreased board-handling loss;

< > Improved board diagnostics and testability leading to higher quality and decreased customer returns; and

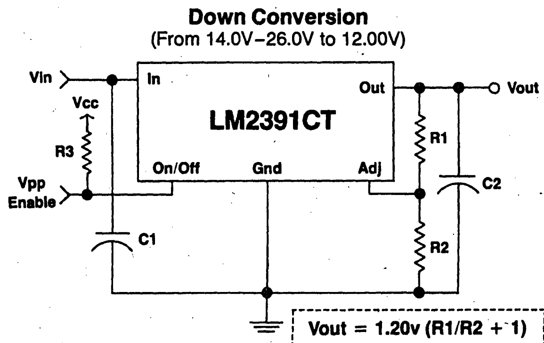
< > **Quicker, more reliable field code updates.**

APPENDIX B
V_{PP} GENERATION CIRCUITS

Circuit #1—Regulation from a higher voltage
Circuit #2—Regulation from a higher voltage
Circuit #3—Regulation from a higher voltage
Circuit #4—5V to 12V Boost
Circuit #5—5V to 12V Boost
Circuit #6—Monolithic DC/DC Convertor

For more detailed information on V_{PP} generation circuits, see AP-357 titled Power Supply Solutions for Flash Memory (Order Number 292092).

Circuit #1



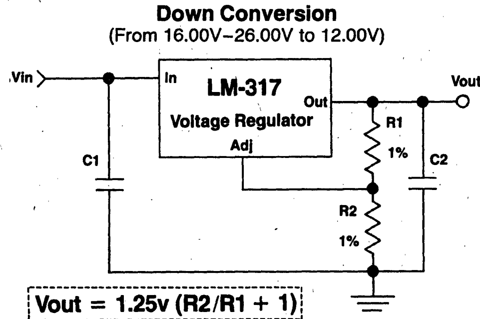
292046-12

COMPONENTS	COST*
LM2391CT	$0.75
R1 = 20 KΩ, 1%	0.045
R2 = 180 KΩ, 1%	0.045
R3 = 10 KΩ	0.02
C1 = 0.1 μF	0.02
C2 = 100 μF	0.15
	$1.03

NOTES:
—The LM2391 offers an enable pin for added data protection.
—The drop out voltage is 0.6V.
—R3 is NOT required if V_{PP} enable is driven by a CMOS device.
*Cost approximations assume 10,000 piece quantity.

Circuit #2

Down Conversion
(From 16.00V–26.00V to 12.00V)

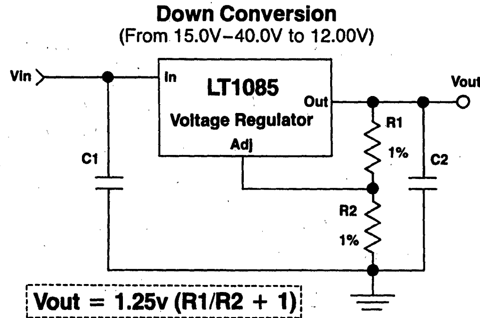


COMPONENTS	COST*
LM-317	0.40
R1 = 124Ω, 1%	0.045
R2 = 1070Ω, 1%	0.045
C1 = 0.1 μF	0.02
C2 = 100 μF	0.15
	$0.66

NOTES:
LM-317 requires a minimum $V_{IN}-V_{OUT}$ = 3.0V
*Cost approximations assume 10,000 piece quantity.

Circuit #3

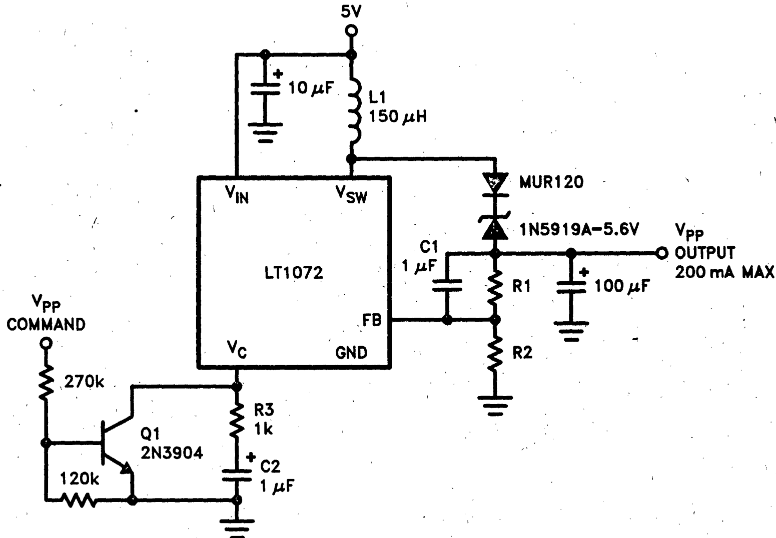
Down Conversion
(From 15.0V–40.0V to 12.00V)



COMPONENTS	COST*
LT-1085	2.50
R1 = 124Ω, 1%	0.045
R2 = 1070Ω, 1%	0.045
C1 = 10 μF	0.10
C2 = 10 μF	0.10
	$2.79

NOTES:
LT-1085 requires a minimum $V_{IN}-V_{OUT}$ = 1.5V
*Cost approximations assume 10,000 piece quantity.

Circuit #4



292046-33

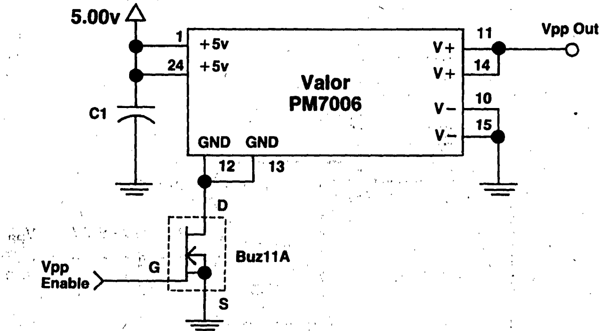
COMPONENTS	COST*
LT1072	1.82
R1 = 10.7k, 1%	0.045
R2 = 1.24k, 1%	0.045
R3 = 1k, 5%	0.02
R4 = 120k, 5%	0.02
R5 = 270k, 5%	0.02
C1 = 1 µF	0.10
C2 = 1 µF	0.10
C3 = 10 µF	0.15
L1 = 150 µH	1.00
Q1 = 2N3904	0.10
	$3.42

V_{PP}OUT	R1	R2	Resistor Tolerance
12.0V	10.7k	1.24k	1%

NOTES:
Drive V_{PP} COMMAND low to turn on the circuit.
*Cost approximations assume 10,000 piece quantity.

Circuit #5

Up Conversion Circuit
(From 5.0V to 12.0V)



COMPONENTS	COST*
PM7006	$6.25
C1 = 0.1 μF	0.05
Buz11A	2.59
	$8.89

NOTES:
1. The capacitor decreases output noise to 140 mV pk-pk.
2. We added the Buz11A Mospower nFET to enable/disable the converter. This control minimizes power consumption which under full load can reach 600 mA.
3. The voltage drop across the switch is 0.1V. Due to this drop the PM7006 will not maintain the V_{PP} spec with 10% fluctuations in V_{CC} supply.

*Cost approximations assume 10,000 piece quantity.

APPENDIX C
LIST* OF DC-DC CONVERTER COMPANIES

AT&T MICROELECTRONICS†
3000 Skyline Drive
Mesquite, TX 75149
Tel: (800) 526-7819
Fax: (214) 284-2317

BURR-BROWN CORP.†
P.O. Box 11400
Tucson, AZ 85734
Tel: (800) 548-6132
Fax: (602) 741-3895

LINEAR TECHNOLOGY CORP.Δ
1630 McCarthy Blvd.
Milpitas, CA 95035
Tel: (408) 432-1900
Fax: (408) 434-0507

MAXIM INTEGRATED PRODUCTSΔ
120 San Gabriel Drive
Sunnyvale, CA 94086
Tel: (408) 737-7600
Fax: (408) 737-7194

MOTOROLA INC.Δ
2100 E. Elliot Rd.
Tempe, AZ 85284
Tel: (800) 845-6686

NATIONAL SEMICONDUCTOR CORP.Δ
Mt. Prospect, IL 60056
Tel: (800) 628-7364
Fax: (800) 888-5113

SHINDENGEN AMERICA, INC.†
2649 Townsgate Rd., Suite 200
Westlake Village, CA 91361
Tel: (800) 634-3654
Fax: (805) 373-3710

SILICONIX INC.Δ
2201 Laurelwood Rd.
Santa Clara, CA 95056
Tel: (800) 554-5565
Fax: (408) 727-5414

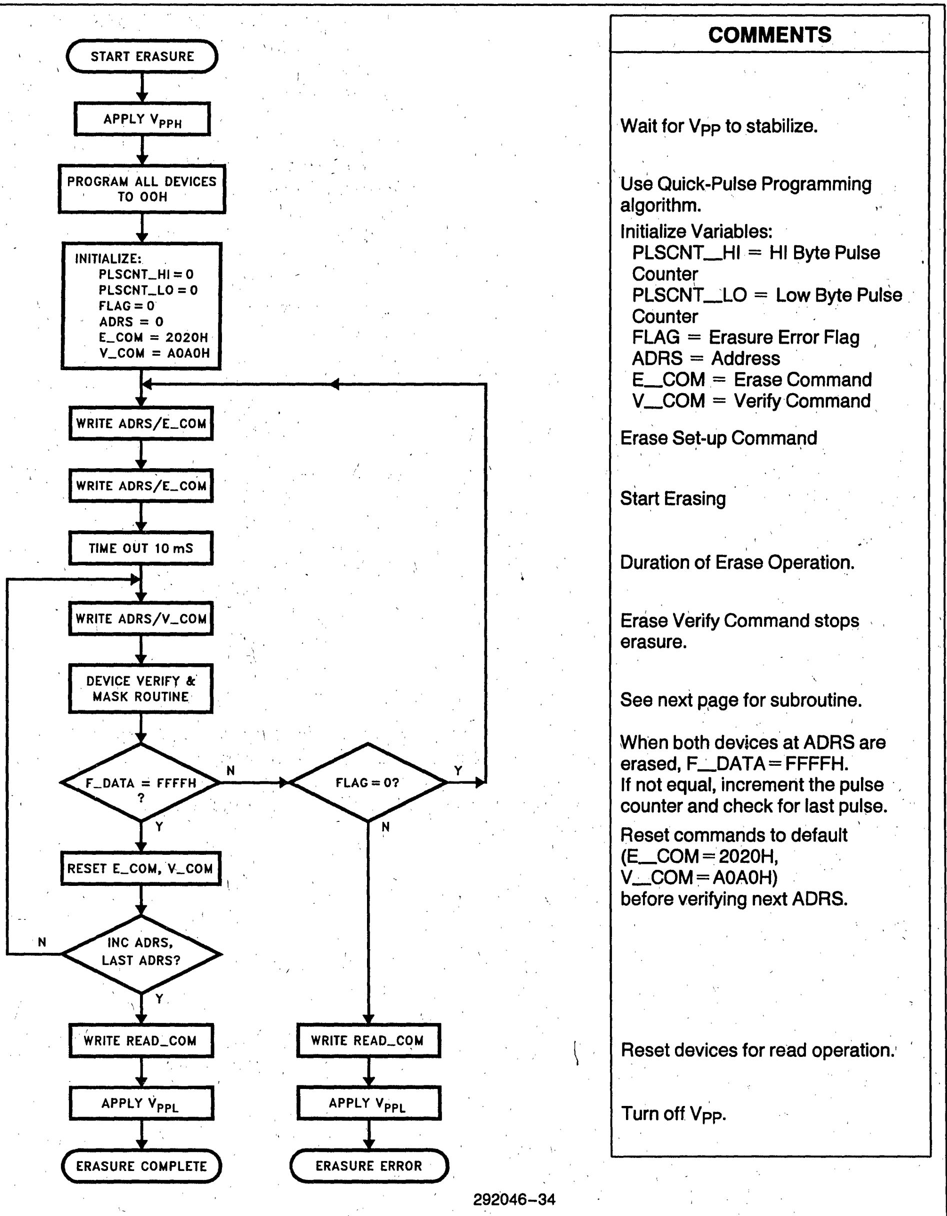
TOKO AMERICA, INC.†
1250 Feehanville Drive
Mount Prospect, IL 60056
Tel: (708) 297-0070
Fax: (708) 699-7864

VALOR ELECTRONICS†
6275 Nancy Ridge Dr.
San Diego, CA 92121
Tel: (619) 458-1471

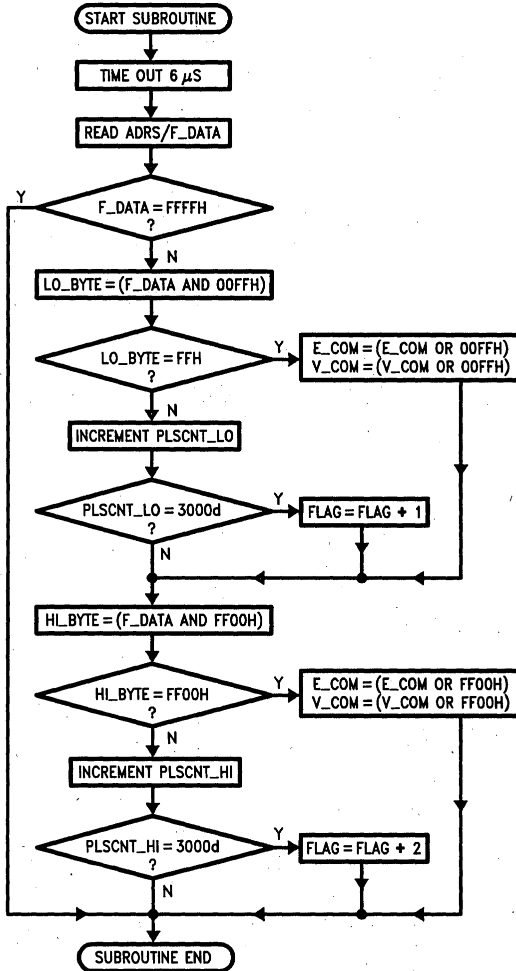
*This list is intended for reference only, and in no way represents all companies that support power conversion products. Since this industry develops many new solutions each year, Intel recommends that the designer contacts the vendors for the latest products. Intel will continue to work with the industry to develop optimum solutions for power conversion. Intel Corporation assumes no responsibilities for circuitry other than circuitry embodied in Intel products. No other circuit patent licenses are implied.

†Monolithic Solutions
ΔDiscrete DC to DC Converter Solutions

APPENDIX D
PARALLEL ERASE FLOW CHART



Device Erase Verify and Mask Subroutine



292046–35

COMMENTS

This subroutine reads the data word (F_DATA). It then masks the HI or LO Byte of the Erase and Verify commands from executing during the next operation.

If both HI and LO bytes verify, then return.

Mask* the HI Byte with 00H.

If the LO Byte verifies erasure, then mask* the next erase and verify commands with FFH (Reset).

If the LO Byte does not verify, increment its pulse counter and check for max count. FLAG = 1 denotes a LO Byte error.

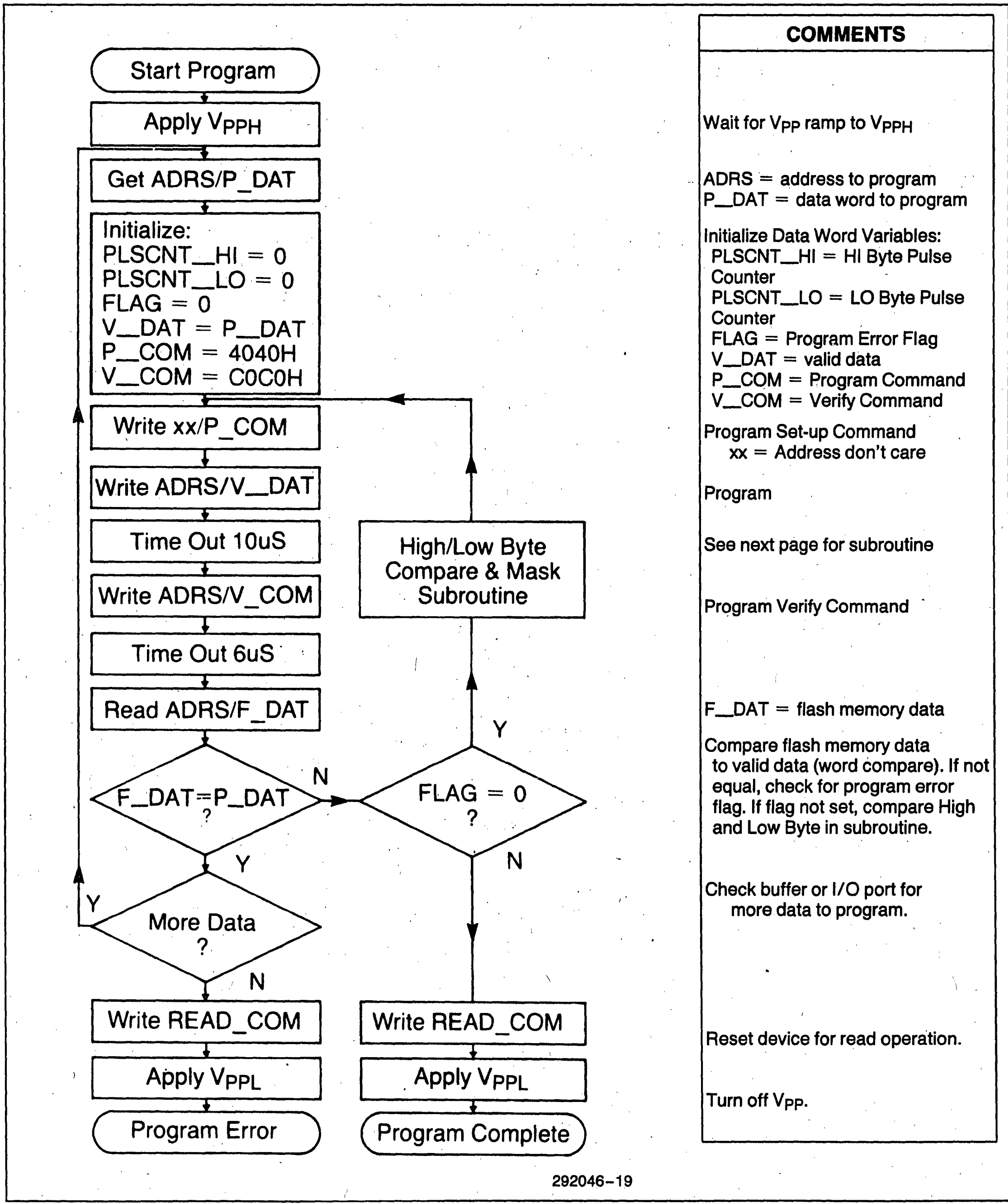
Repeat sequence for the HI Byte

FLAG = 2 denotes a HI Byte error. FLAG = 3 denotes both HI and LO Byte errors.

NOTE:
*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming flash data (F_DATA), the erase commands and the verify commands. Then manipulate the HI or LO register contents.

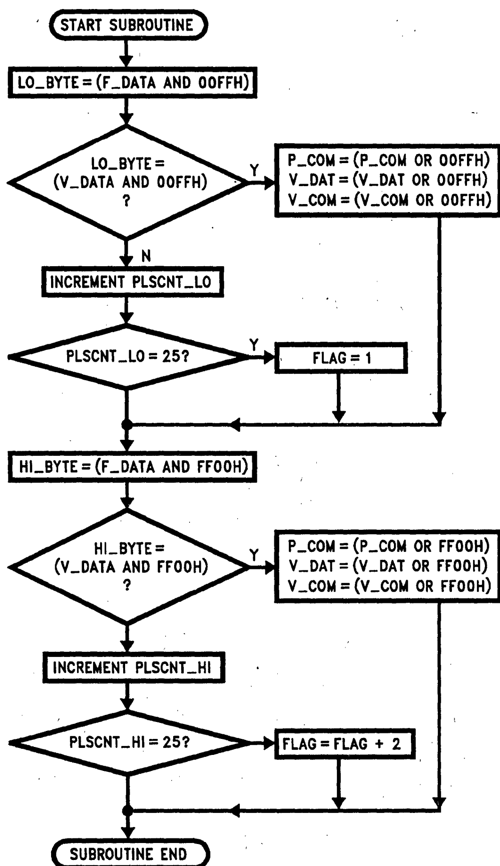
3

APPENDIX E
PARALLEL PROGRAMMING FLOW CHART



292046-19

Program Verify and Mask Subroutine



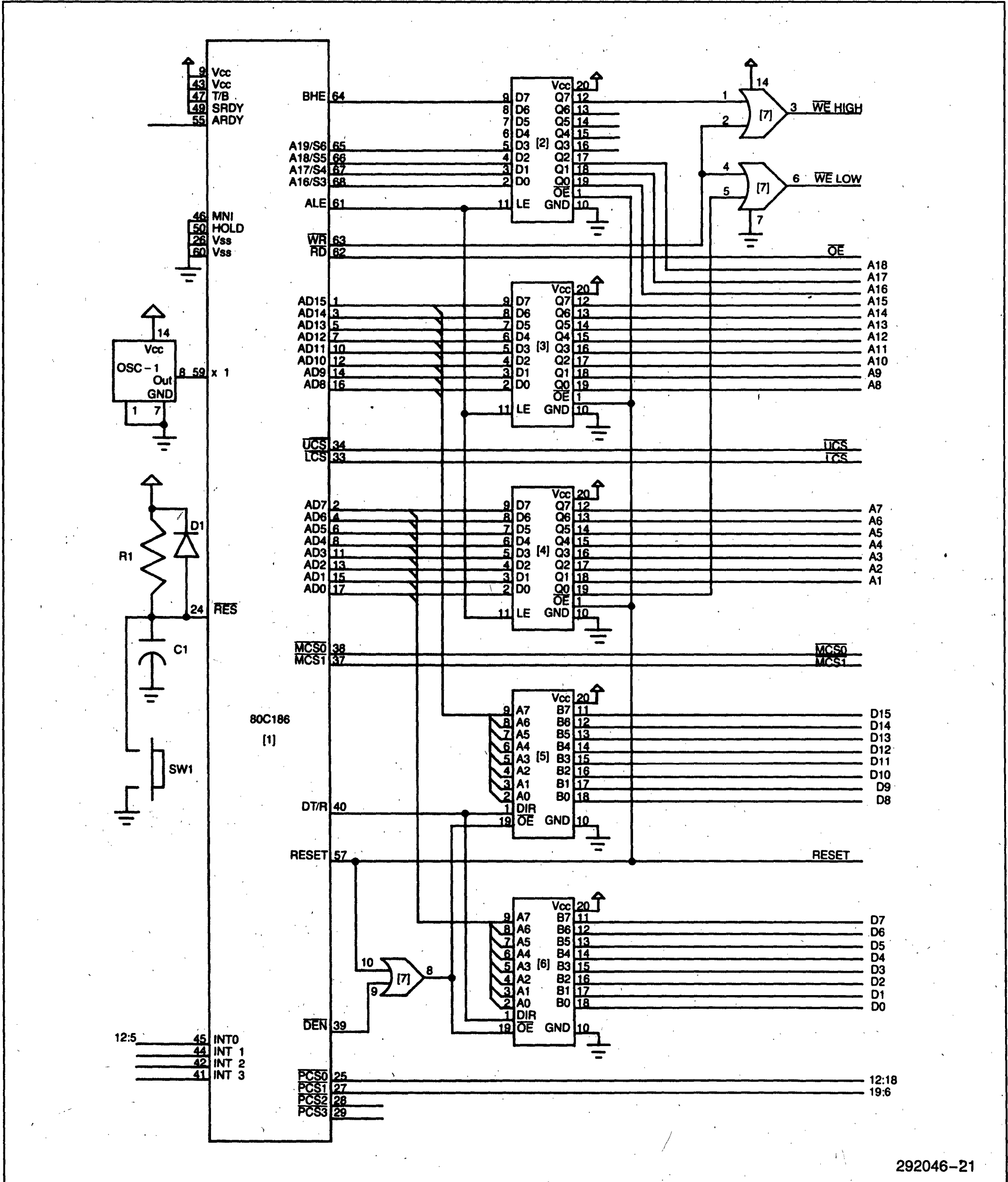
COMMENTS
To look at the LO Byte, mask* the HI Byte with 00.
If the LO Byte Verifies, mask the LO Byte commands with the reset command (FFH).
If the LO Byte does not verify, then increment its pulse counter and check for max count. FLAG = 1 denotes a LO byte error.
Repeat the sequence for the HI Byte.
FLAG = 2 denotes a HI Byte error. FLAG = 3 denotes both HI and LO Byte errors. FLAG = 0 denotes no max count errors; continue with algorithm.

292046–40

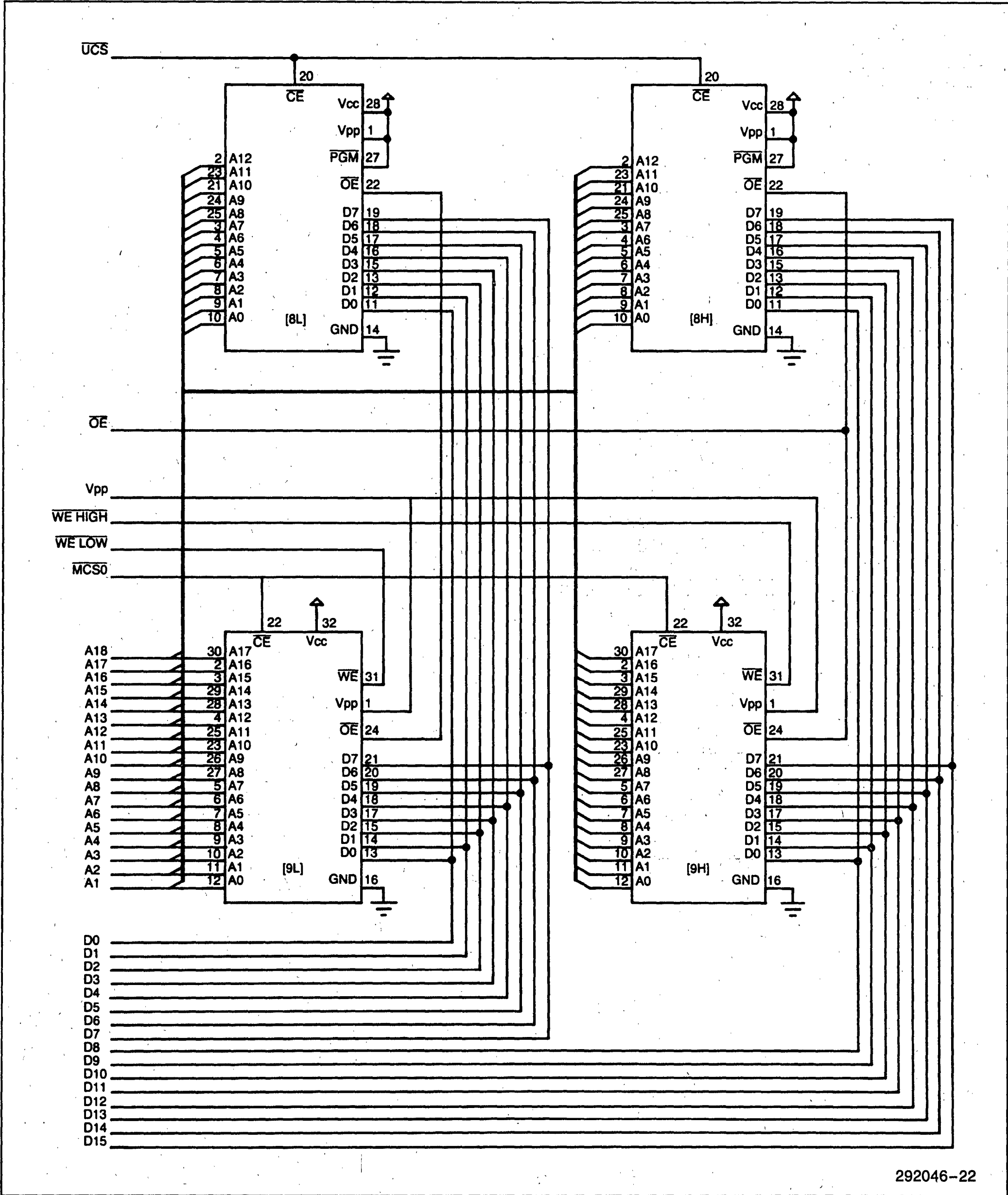
NOTE:
*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming data (F-DAT), the program commands and the verify commands. Then manipulate the HI or LO register contents.

3

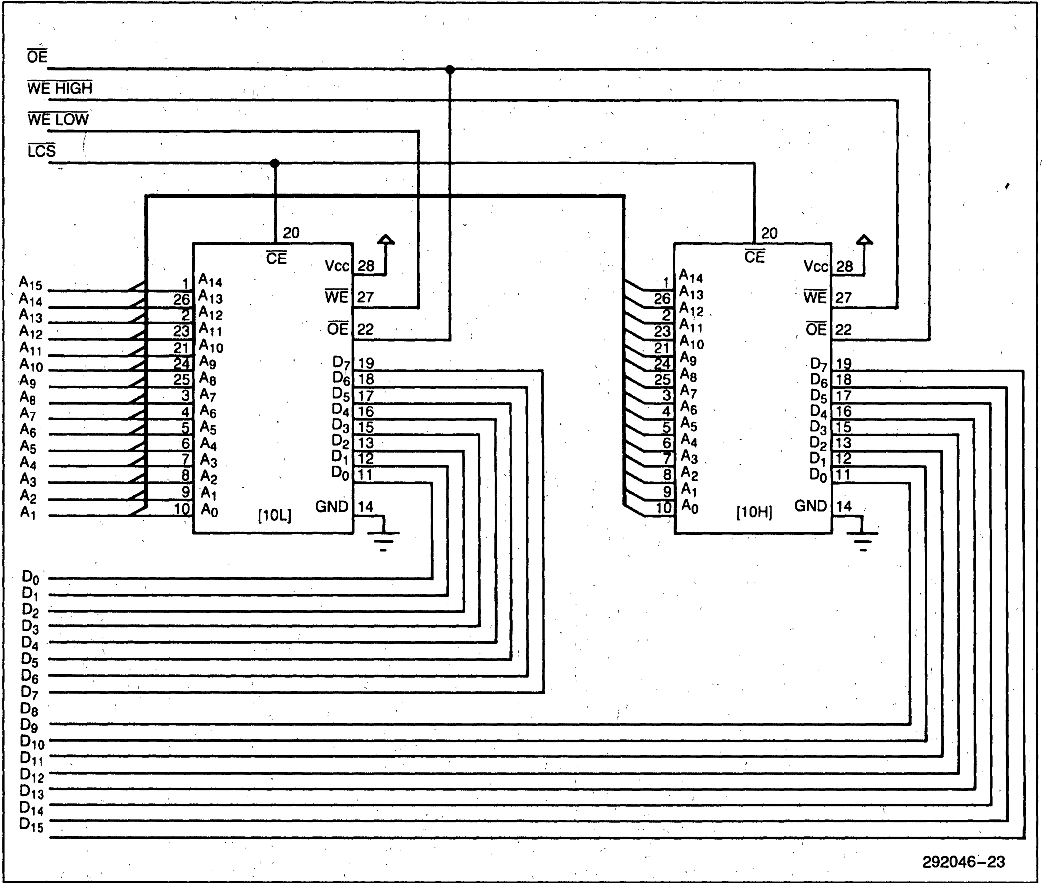
APPENDIX F
DETAILED SYSTEM SCHEMATICS



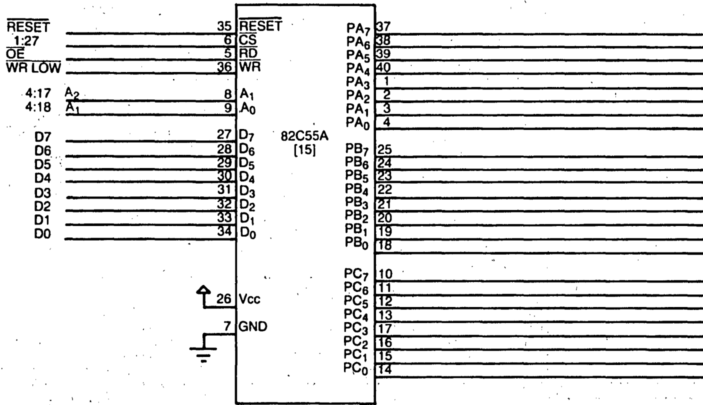
292046-21

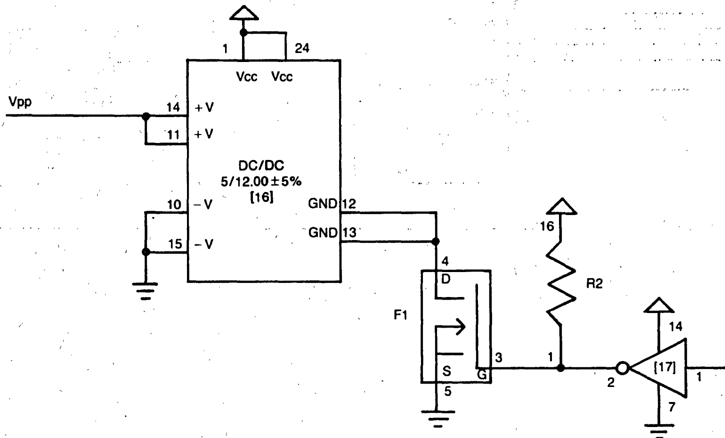


292046-22



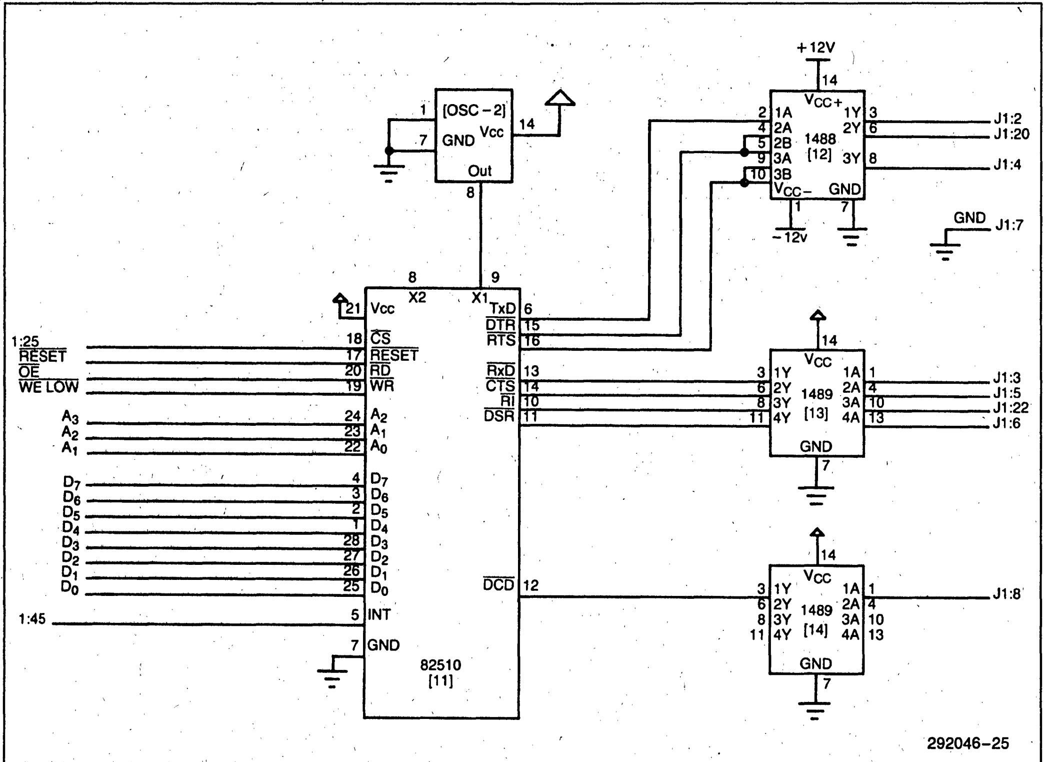
292046-23





292046–24

3



292046-25

256K FLASH MEMORY DEMO PARTS LIST

Device	Component	Pins	Description
[1]	80C186	68	16-bit high integration CPU
[2,3,4]	74HC573	20	Latch
[5,6]	74HC245	20	Transceiver
[7]	74HC32	14	OR gate
[8L,8H]	27C64	28	16 Kbyte EPROM
[9L,9H]	28F256A	32	64 Kbyte flash memory
[10L,10H]	32K x 8 SRAM	28	64 Kbyte SRAM
[11]	82510	28	Asynchronous Serial Controller
[12]	14C88	14	RS-232 Line Driver
[13,14]	14C89	14	RS-232 Line Receiver
[15]	82C55A	40	Programmable Peripheral Controller
[16]	PM7006	24	DC/DC Convertor (5V–12.00V)
[17]	7406	14	Invertor—Open Collector (O.C.)
C1	20 μF	2	Capacitor for CPU reset
D1	1N914	2	Diode for CPU reset
F1	BUZ11A	3	MOSPOWER nFET
J1	DB-25	25	Connector (male)
OSC-1	20 MHz	14	CPU Oscillator
OSC-2	18.432 MHz	14	Serial Controller Oscillator
R1	10 KΩ	2	¼W, 10% Resistor for CPU reset
R2	1 KΩ	2	¼W, 10% Resistor for O.C. pull-up
SW1		3	Momentary Push Button for CPU reset

NOTES:
1. Place a 0.1 μF bypass capacitor at the V_{CC} input of each IC.
2. Place a 0.1 μF bypass capacitor on the V_{PP} input of each 28F256 flash memory.

28F512 UPGRADE FOR THE 80C186/FLASH MEMORY DESIGN

To upgrade the 80C186/Flash memory design to handle 28F512's, the range of the $\overline{CE}$ signal has to be increased. There are a number of ways to generate a $\overline{CE}$ signal that will span the 128 Kbyte address range of two 28F512 devices.

1. AND two of the current MCS lines together (defined for 64 Kbytes each); or
2. Change the MCS individual block-select size from 64 Kbytes:
   MMCS_VALUE = 41F8H,
   MPCS_VALUE = 0A0B8H

   to 128 Kbytes:
   MMCS_VALUE = 01FEH,
   MPCS_VALUE = 0C0BEH

Also, cut the $\overline{CE}$ trace to the RAM sockets. Then wire $\overline{MCS0}$ to the RAM $\overline{CE}$. This eliminates the $\overline{MCS0}$ and $\overline{LMCS}$ range overlap caused by increasing the MCS range to 128 Kbytes. See 80C186 Data Sheet page 21 and 22 (Order # 270354).

28F010 UPGRADE TO THE 80C186/FLASH MEMORY DESIGN

To upgrade the 80C186/Flash memory design to handle 28F010's, a $\overline{CE}$ signal has to be generated. There are a number of ways to generate a $\overline{CE}$ signal that will span the 256 Kbyte address range of two 28F010 devices.

1. **AND two of the MCS lines together** (defined for 128 Kbytes each as noted in the 28F512 modifications):

   Cut the LMCS trace to the RAM sockets. Connect $\overline{MCS0}$ to $\overline{CE}$ on the RAM sockets (U10L,UH).

   Cut the $\overline{MCS2}$ trace to the flash memory. Add an AND gate. Connect $\overline{MCS2}$ (cut trace) and $\overline{MCS3}$ to the inputs of the AND gate. Then wire the AND gate output to the $\overline{CE}$ of the flash memories.

   Also, change the onboard memory MCS register to:
   MMCS__VALUE=01FEH, MPCS__VALUE=0C0BEH [128K blocks],

   and delete:
   LMCS__REG and LMCS__Value.

2. **Add a decoder;**

   Add a decoder (74HC138). Connect address lines A18 and A19 to the B and C inputs of the decoder. Tie the A input of the decoder low, and enable all the enables. By using outputs Y0, Y2, Y4, and Y6, you have four $\overline{CE}$ lines decoding 256 Kbyte blocks each.

   Cut the $\overline{MCS2}$ trace to the flash memories. Connect the Y2 output from the decoder to the $\overline{CE}$ input of the flash memory.

3. **Replace the address latch (U2) with a PLD that latches and decodes.**

   Program a 5C032 as an integrated latch and decoder. Replace the upper address latch [U2] with the Intel 5C032 EPLD. Cut the $\overline{CE}$ trace to the flash memories. Connect the flash memories' $\overline{CE}$ to the 5C032 pin 12. This maps the address space 40000H to 7FFFFH. See Figures 1 and 2 for a comparison of the 74HC573 (U2) and programmed 5C032 pin outs. Figure 3 is the source code for the EPLD.

   Also, change the value of the MMCS and MPCS registers to 64 Kbyte blocks so that the $\overline{MCS0}$ range does not overlap the LMCS range.
   MMCS__VALUE=41F8H, MPCS__VALUE=0A0B8H.

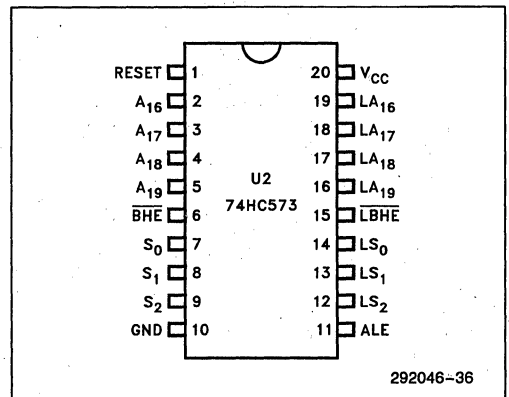


Figure 1. Latch Pinout

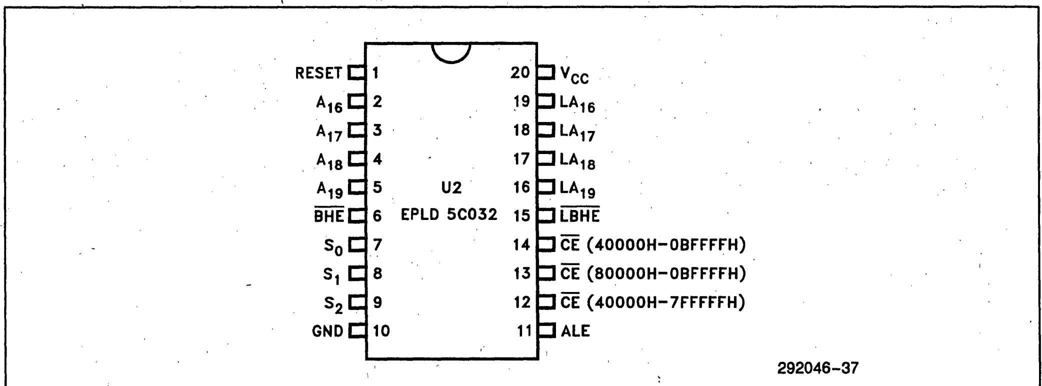


Figure 2. Integrated Latch and Decoder

```
Thom Bowns - PLFG Applications
Intel
January 13, 1989
EPLD HOTLINE: 1-800-323-EPLD
002
5C032
Custom Latched Decoder
OPTIONS: TURBO=ON
PART: 5C032

INPUTS: ALE@11, RESET@1, A19@5, A18@4, A17@3, A16@2, nBHE@6

OUTPUTS:  LA18@17, LA17@18, LA16@19, LnBHE@15, nCE3@14, LA19@16,
          nCE2@13, nCE1@12

NETWORK:
          ALE = IN (ALE)
          RESET = INP (RESET)
          nRESET = NOT (RESET)
          A19 = INP (A19)
          A18 = INP (A18)
          A17 = INP (A17)
          A16 = INP (A16)
          nBHE = INP (nBHE)
          LA19, LA19 = COIF (LA19d, nRESET)
          LA18, LA18 = COIF (LA18d, nRESET)
          LA17, LA17 = COIF (LA17d, nRESET)
          LA16, LA16 = COIF (LA16d, nRESET)
          LnBHE, LnBHE = COIF (LnBHE, nRESET)
          nCE3, nCE3 = COIF (nCE3, nRESET)
          nCE2, nCE2 = COIF (nCE2, nRESET)
          nCE1, nCE1 = COIF (nCE1, nRESET)

EQUATIONS:
          LA19d = A19 * ALE + LA19 * !ALE;
          LA18d = A18 * ALE + LA18 * !ALE;
          LA17d = A17 * ALE + LA17 * !ALE;
          LA16d = A16 * ALE + LA16 * !ALE;
          LnBHEd = nBHE * ALE + LnBHE * !ALE;
          nCE3d = nCE3EQN * ALE + nCE3 * !ALE;
          nCE2d = nCE2EQN * ALE + nCE2 * !ALE;
          nCE1d = nCE1EQN * ALE + nCE1 * !ALE;
          nCE2EQN = !(A19 * !A18);
          nCE1EQN = !( !A19 * A18);
          nCE3EQN = !( !A19 * A18 + A19 * !A18);

END$
```

Figure 3. Source Code for the Integrated Latch and Decoder

intel®

APPLICATION NOTE

AP-341

September 1992

Designing an Updatable BIOS Using FLASH Memory

BRIAN DIPERT
DON VERNER
MCD MARKETING APPLICATIONS

Order Number: 292077-004

Designing an Updatable BIOS Using Flash Memory

CONTENTS PAGE

1.0 INTRODUCTION 3-356

2.0 FLASH MEMORY 3-356

2.1 EPROM Roots; Review of Flash Process vs. EPROM & EEPROM .. 3-356

2.2 Program and Erase Automation .. 3-356

2.3 Blocked Architecture 3-357

2.4 Deep Powerdown Mode 3-358

2.5 28F001BX Pinouts, Physical Layout and Upgrade 3-359

Plastic Dual-In Line

Plastic Leaded Chip Carrier

Thin Small Outline Package

2.6 V_{PP} Specifications 3-362

Fixed V_{PP} and V_{CC}

$\overline{PWD}$, V_{CC} and V_{PP} Lockout Protection

3.0 HARDWARE DESIGN CONSIDERATIONS 3-363

3.1 BIOS Boot Code Requirements and System Configurations 3-364

Address Shift Configuration

Address Inversion Configuration

3.2 V_{PP} Generation 3-366

Using System 12V Directly

Pumping 5V to 12V

Security

Using a MOSFET Switch

3.3 Modifying an Existing Motherboard 3-367

EPROM/ROM Designs

28F010 Flash Memory Designs

3.4 In-System Write vs. On-Board Programming 3-368

3.5 Ideas for Using Extra Adaptor Space 3-369

CONTENTS PAGE

4.0 SOFTWARE DESIGN CONSIDERATIONS 3-369

4.1 Update Software for a Modified System 3-370

4.2 Pseudo-Code Overview 3-371

Pseudo-Code for Flash Update Routine

4.3 Initializing the System 3-371

Checking Power

4.4 Code Loader Routine 3-371

4.5 Flash Reprogramming Routines .. 3-372

On-Chip Erase Algorithm

Erase Suspend/Resume

On-Chip Programming Algorithm

Full Status Checks

4.6 Recovery Routine Overview 3-376

4.7 Power Management 3-376

5.0 SUMMARY 3-377

5.1 Traditional BIOS Storage and Disadvantages 3-377

5.2 Advantages of an Updatable BIOS 3-377

5.3 Advantages of Adding DOS in FLASH 3-377

5.4 Advantages of Adding 1 MB–4 MB of Resident Code Storage 3-377

APPENDIX A—Software Routines 3-378

APPENDIX B—MS-DOS ROM Version Overview 3-394

APPENDIX C—BIOS Vendor Information 3-395

APPENDIX D—Microprocessor/ Microcontroller Compatibility Chart 3-396

1.0 INTRODUCTION

As PC computing platforms increase in complexity, so does the associated BIOS code. Sophisticated hardware and BIOS software increase the potential for revisions. Time-to-market goals require faster completion of designs from conception to production, leaving less time for new-peripheral BIOS support. As an example, many 80286-based PC/ATs lack BIOS support for 3½" floppy drives! Once a computer is out the door, code revisions are far more difficult and costly. Code revisions with EPROM require either a service call or sending EPROMs to the end user, assuming nothing else goes wrong in the process. The alternative to BIOS update is a prematurely obsolete system, unable to support new industry standards and peripheral systems.

Flash memory offers the same nonvolatile storage as EPROM, but additionally offers in-system write capability. Using Intel's 28F001BX for BIOS storage, code updates are done quickly in the factory during test and debug, while allowing cost-effective field updates to end users via floppy disks or modem BBS.

This application note describes various methods of implementing a flash memory BIOS using the 28F001BX. Design targets are both laptop and desktop systems. The primary emphasis is on application of flash memory for BIOS and ROM executable software applications. Detailed 28F001BX information is covered in the datasheet, available through your local sales office.

2.0 FLASH MEMORY

This section provides a brief overview of Intel's Flash Memory and in particular, Intel's 28F001BX blocked flash memory family. It covers the following:

- Flash memory's EPROM roots
- Program and Erase Automation
- Blocked Architecture
- Deep Powerdown Mode
- Pinouts, physical layout and upgrade for different packages
- V_{PP} specifications

Major features of the 28F001BX are in-system write, selective block erase, program/erase automation, SRAM-like command interface, deep powerdown capability, fixed V_{CC} and V_{PP} supplies and hardware lockout protection.

2.1 EPROM Roots; Review of Flash Process vs EPROM & EEPROM

Intel's ETOX™ II (EPROM Tunnel OXide) flash memory is a single-transistor cell providing nonvolatile storage like EPROM, with electrical erase similar to EEPROM. Reprogramming flash memory entails electrically erasing all data bits in parallel, then randomly programming data into any byte in the array. The programming operation is achieved via channel hot electron injection (CHE), just like EPROMs. Flash electrical erasure, however, is accomplished through Fowler-Nordheim (FN) tunneling. Using separate program and erase methods (CHE vs. FN Tunneling), in different cell locations, drain vs. source, permits process optimization for high cycling endurance—the number of complete erase and re-writes. Traditional low-density EEPROMs tunnel through the same memory cell junction for both programming and erasure. Because EEPROMs erase before programming each byte, these processes must occur very fast. Therefore, internal voltages used to program or erase 5V-only EEPROM memory cells are high (e.g., 18V–30V). The combination of higher voltage with programming and erasing through the same junction contributes to EEPROM's oxide breakdown, poor data retention and reduced cycling capability.

Intel's flash memory erasure (tunneling) voltage is below the critical oxide breakdown voltage. By using block erasure instead of EEPROM's byte erasure, erase times are relaxed, reducing tunneling voltages. Programming Intel's Flash Memory is non-destructive to the floating gate oxide compared to EEPROM's use of tunneling for programming. These features for erase and programming provide Intel's Flash Memory with the highest endurance (typically over 100K cycles) compared to that of traditional EEPROM cycling. Furthermore, flash memory exhibits lower failure rates at any given cycle count.

2.2 Program and Erase Automation

The 28F001BX integrates a Write State Machine (WSM) on-chip to internally implement program and erase algorithms. Operations are initiated through command sequence writes to the Command Register, and progress is reported back to the user through Status Register bits. Software timers are no longer required, as timing is now regulated through the on-chip oscillator. System software requirements are decreased in comparison to past algorithms, minimizing overhead and development effort, and allowing code execution and interrupt servicing while simultaneously programming or erasing the device.

The Erase Suspend command halts block erase to execute code or read data from any other block. This feature gives the system the capability to service higher level interrupts requiring data from the 28F001BX during the erase interval. After issuing the Erase Resume command, the WSM continues erase where it left off when suspended.

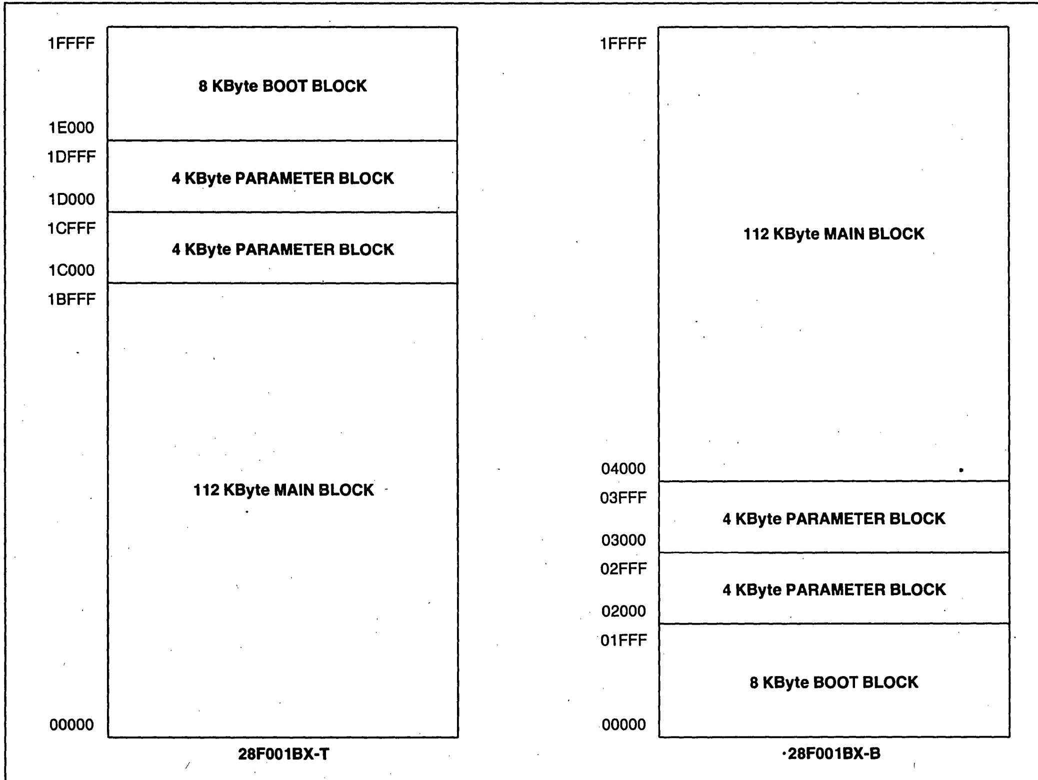


Figure 1. 28F001BX Memory Maps

2.3 Blocked Architecture

The 28F001BX family combines the safety of a hardware-protected 8 KByte "boot" block with the flexibility of two 4 KByte "parameter" blocks and one 112 KByte main block. Each block can be individually erased and programmed without affecting code stored in another block, ensuring data integrity. The boot block is intended to contain secure code which minimally will bring up the system and download code to the other blocks of the 28F001BX if required. Once programmed, it is hardware-locked from further alteration, guaranteeing true non-volatility.

The 28F001BX-T's lockable block location provides compatibility with microprocessors and microcontrollers that boot from the top of the memory map, such as Intel's x86 family and i860™ microprocessor. The segmentation of the 28F001BX-B is identical. Its lockable block location provides compatibility with microprocessors that boot from low memory, such as Motorola and AMD products. See Figure 1 for illustrations of the two memory maps available in the 28F001BX family.

The two 4 KByte parameter blocks have multiple uses in BIOS environments. They can be used to back up the CMOS setup parameters such as floppy and hard disk type, processor speed, system memory size, graphic display type and presence of a coprocessor. Today, should the system battery fail, the user loses information in battery-backed SRAM. The non-volatile parameter blocks provide the system capability to automatically back up and recover this information. Also, EISA systems can store software variable information such as add-in board addresses, DMA channels and interrupt values/levels.

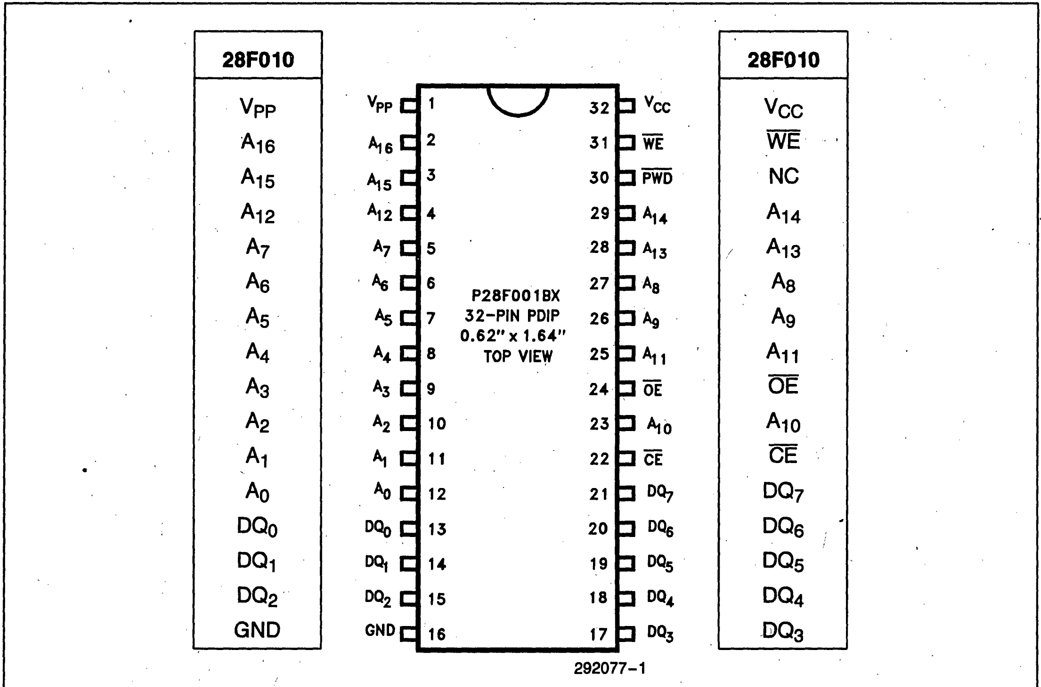


Figure 2. 28F001BX DIP Pin Configuration

2.4 Deep Powerdown Mode

Market analysts predict that the high-growth segments of the PC industry for the next several years will be in the laptop, palmtop and handheld product lines. Power management software is becoming an integral part of PC system BIOS as manufacturers attempt to squeeze the maximum amount of system operation time from their battery pack power supplies. A key indicator of this trend is Intel's i386™SL microprocessor superset, which adds hardware power management to the Intel386™ architecture and answers the needs of low-power applications.

The 28F001BX family features deep powerdown capability and is ideally suited for these same battery-operated systems. Powerdown is entered through low voltage on the $\overline{PWD}$ pin. Typical power consumption through V_{CC} in powerdown is 0.25 μW, regardless of power supply voltages and activity on the external bus. Once BIOS is shadowed to system DRAM for high-speed execution, the 28F001BX can be shut down for minimal current drain.

This same pin, with 12V present on it, gates program and erase of the boot block. Such a hardware lockout preserves the system boot code once initially programmed and protects it from inadvertent alteration or computer virus compromise.

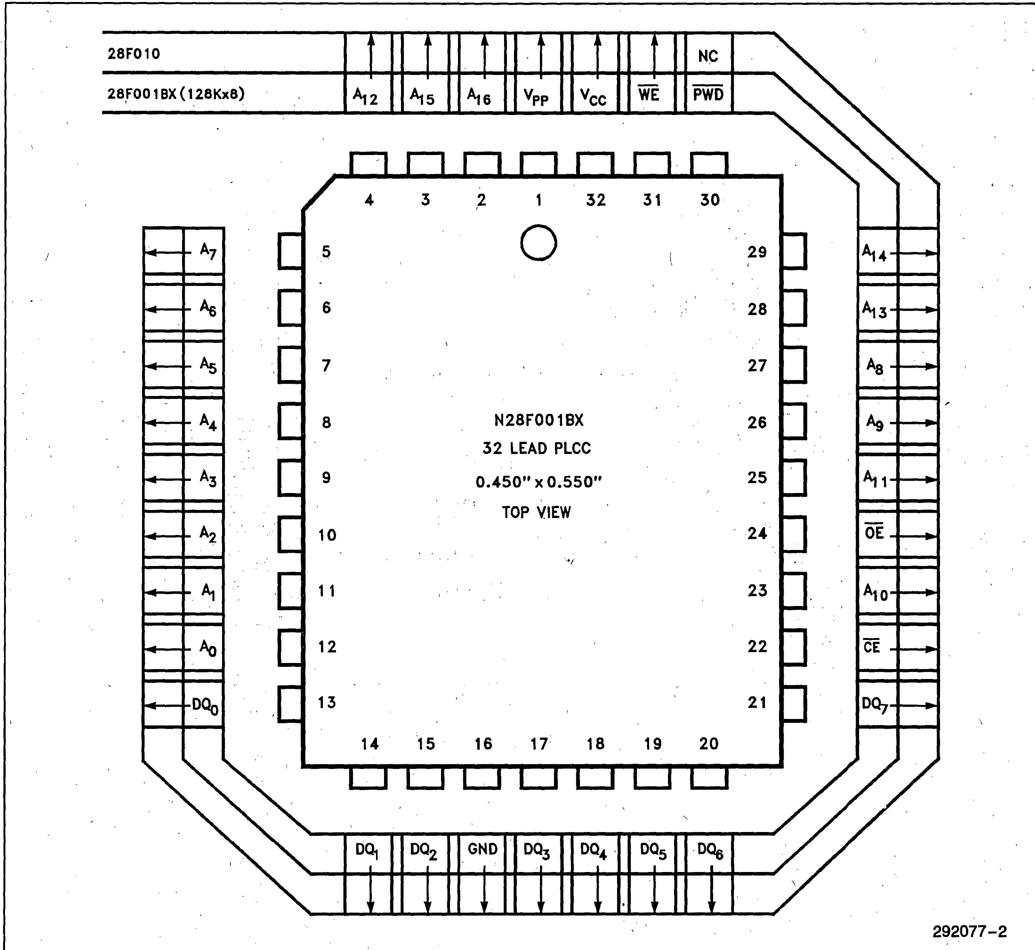


Figure 3. 28F001BX PLCC Lead Configuration

2.5 28F001BX Pinouts, Physical Layout and Upgrade

Intel's 28F001BX is offered in three standard 32-pin packages: Plastic Dual In-line Package (PDIP), Plastic Leaded Chip Carrier (PLCC), and Thin Small Outline Package (TSOP). All three pinouts provide backward compatibility with Intel's 28F010 bulk-erase flash. See Figures 2, 3, and 4 for pinout details.

Plastic Dual In-Line Package

PDIPs with sockets provide an excellent way to prototype and debug new designs. The 28F001BX is backward pin-compatible with 1 Mbit standard flash and EPROMs.

Plastic Leaded Chip Carrier

Most system designs today require surface mount technology (SMT) due to shrinking board real estate and portable form factors. PLCC is one SMT component that uses as little as 35% of the overall board space compared to PDIP. Its small size is attributed to the center-to-center lead spacing of 50 mils versus 100 mils, as well as its four-sided pinout. The J-lead design allows the PLCC to be directly soldered to the circuit board. Most SMT manufacturing equipment can easily handle the PLCC's 50-mil lead pitch. PLCC SMT sockets such as that offered by AMP (P/N 821977-1) have an identical foot-print for 32-pin devices. Such sockets can be used in place of directly soldering a PLCC for prototype build and code testing. Once the reprogramming code is tested and debugged, flash PLCCs can then be surface-mounted without socketing during production runs.

Thin Small Outline Package

TSOP is the package of choice for hand-held equipment or palmtop/laptop computers. These compact systems require minimal height and area for all components, for which TSOP excels. TSOP height measures 1.2mm versus 3.5mm for PLCC. TSOP area is 8mm x 20mm compared to PLCC's 11.43mm x 13.97mm. Therefore, TSOP has significantly less total volume: TSOP = 172.8mm^3, while PLCC = 656.3mm^3, and DIP = 1872.3mm^3. State-of-the-art center-to-center terminal spacing of 20 mils yields a smaller package with narrower conductor traces than PLCC or PDIP. Location of pins on both ends of the package allows traces for TSOP to be routed underneath the chip, reducing board layers. TSOP for the 28F001BX is available in the standard (E) pin configuration. For multiple chip flash systems, Intel's bulk-erase 28F010 flash memory is available in both standard (E) and reverse (F) pin configurations (see Figure 5) allowing components to be laid out end-to-end and side-to-side for highest board density (see Figure 6). Note how pins 32–17 on the standard pinout match pins 1–16 on the reverse pinout, and how pins 1–16 on the standard pinout match pins 32–17 on the reverse pinouts.

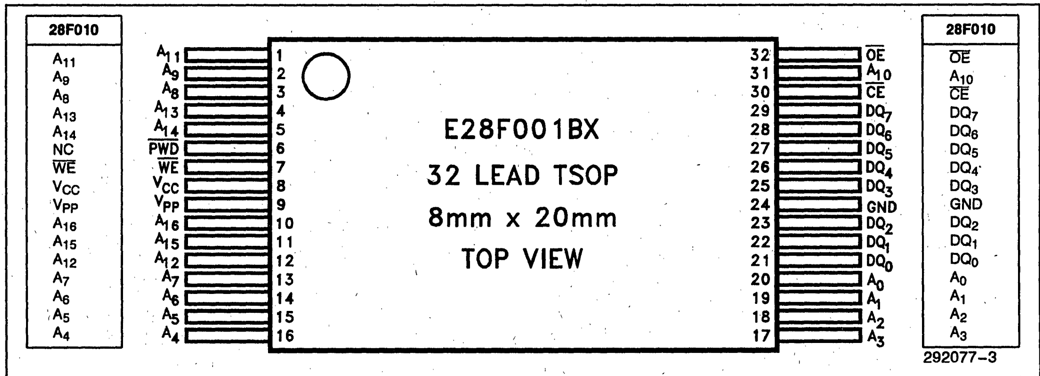


Figure 4. 28F001BX TSOP Lead Configuration

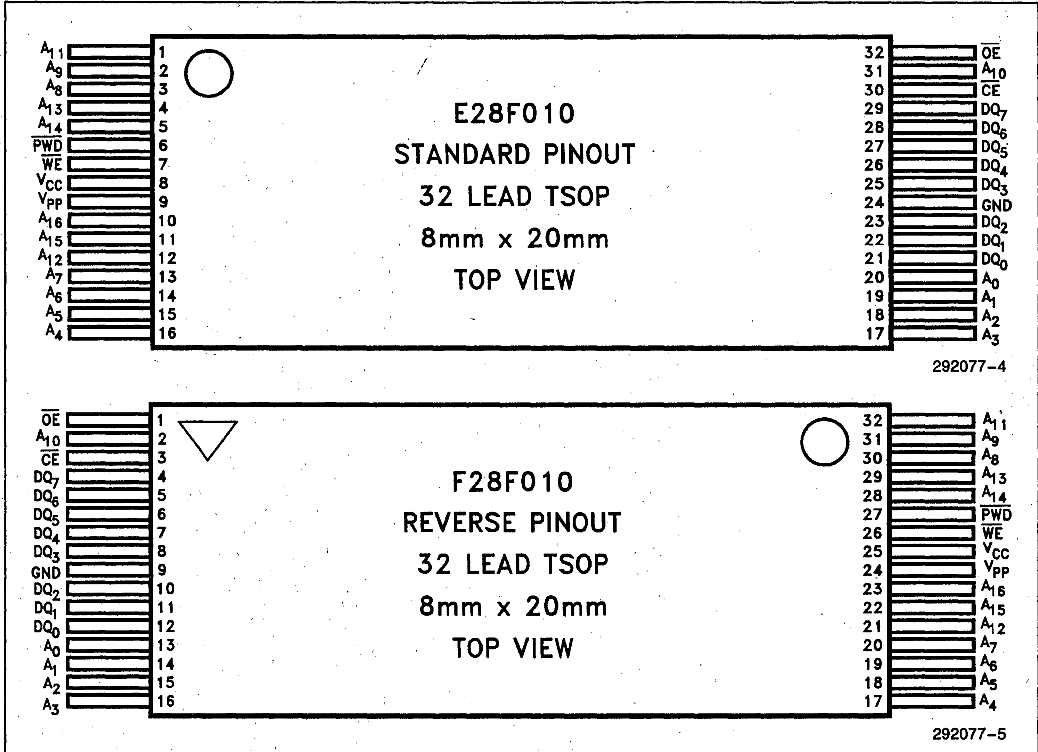


Figure 5. 28F010 Standard and Reverse TSOP Lead Configurations

3

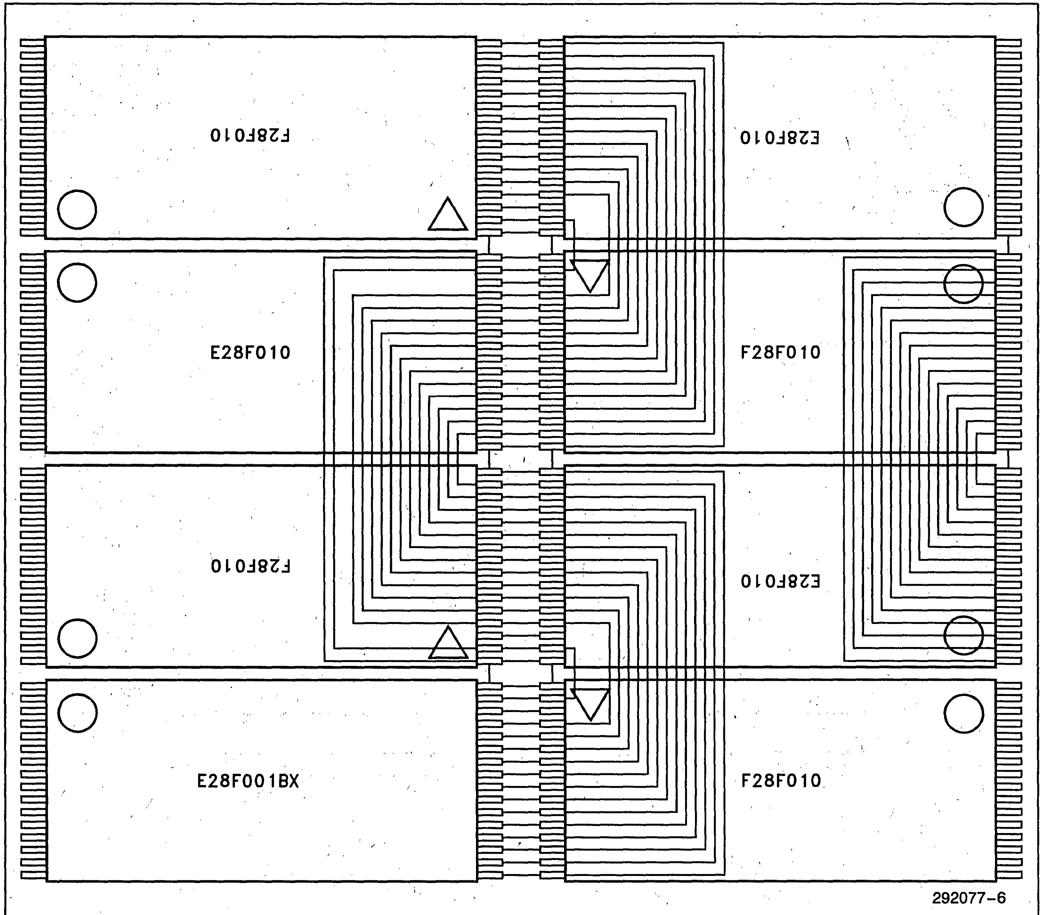


Figure 6. 28F001BX and 28F010's in Serpentine Layout Using TSOPs

2.6 V_{PP} Specifications

Fixed V_{PP} and V_{CC}

Flash memories, like EPROMs, require a 12V programming power supply. Unlike EPROMs, however, the V_{PP} for flash memory is a fixed, standard level. When combined with the Command Register erase/program control, Intel flash memories use a simple, SRAM-like hardware interface with standard microprocessor timings.

Intel's Flash Memory V_{PP} specification is 12.0V ±0.6V (5%), compatible with most off-the-shelf system power supplies. The IBM PC technical reference manual specifies the 12V power supply at 12.0V, +5% and −4%. Additionally, some hard and floppy drives require 12V ±5%. Therefore, most PC power supplies have 12V supplies with ±5% or better tolerance. Possible exceptions to this are laptop and/or palmtop PCs. Some of these require 5V-only designs, in which case 5V is charge-pumped to 12V. It is essential to use the specified V_{PP} when programming and erasing flash. Once the commands to program or erase are issued, the device internally derives the required voltage references from the V_{PP} supply. Therefore, an improper V_{PP} level degrades the performance of the part.

The Write State Machine automatically monitors the voltage present on the V_{PP} pin, beginning when program or erase setup commands are issued and continuing throughout the internal algorithm interval. If low V_{PP} is detected, the WSM automatically aborts the program or erase attempt and reports an improper voltage error to the user through the Status Register. The hardware design section discusses various methods of V_{PP} generator if your 12V power supply does not meet the proper tolerances or 12V is not available.

$\overline{PWD}$, V_{CC} and V_{PP} Lockout Protection

The deep powerdown $\overline{PWD}$ pin provides hardware write protection for 28F001BX flash memories. Until this pin transitions to TTL-level V_{IH}, write attempts to the device Command Register are ignored, regardless of power supply levels or activity on the system bus and control inputs. Typically, the system designer will gate this signal with a system POWER GOOD indicator output to ensure system stability before memory accesses begin.

The 28F001BX family provides additional protection for designs that tie 12V directly to the device. Since the 12V supply is less capacitively loaded than the 5V supply, the 12V power supply reaches full value faster during power-on. If Command Register lockout protection was not provided, a finite possibility exists that inadvertent writes may occur during power-on. For this case, Intel's 28F001BX flash memory supplies Command Register lockout protection when V_{CC} is below 2.5V, preventing writes to flash memory from occurring. Since CMOS logic is valid at 2.0V, a 0.5V margin of protection exists, providing extra time for control signals to settle before the Command Register is activated. Program/erase inhibit is guaranteed with V_{PP} below 6.5V, with corresponding V_{PP} low reported through the Status Register. Once V_{CC} reaches 2.5V, the Command Register begins processing valid commands, and program/erase attempts may initiate with V_{PP} greater than 6.5V. At this point, the system is responsible for write control.

When the 28F001BX V_{CC} powers up, or after the $\overline{PWD}$ pin transitions to V_{IL} and back to V_{IH}, the Command Register is automatically initialized to Read Array mode.

3.0 HARDWARE DESIGN CONSIDERATIONS

The system level hardware requirements for implementing BIOS and application storage in flash are:

- Write Enable available to all of the flash memory
- 12V routed to flash location or generated on-board
- CMOS control-signal interface, or $\overline{PWD}$ gated by a power-good signal
- Data buffer or transceiver that works in both write and read directions
- Space in memory map allocated for each application's size

Intel's i386SL microprocessor superset was chosen for the design example, shown in Figure 7. The Intel386SL microprocessor superset integrates all major components of a Intel386 based design on two chips, including bus memory, cache controllers and the ISA peripheral subsystem. Additionally, it consolidates hardware power management for battery-operated designs such as laptop, handheld and palmtop computers.

Note the clean interface between the superset and 28F001BX-T. Flash memory was comprehended early in the design of the Intel386SL microprocessor superset, to ensure a minimal-chip interface. Transceivers for the system bus, as well as a flash memory $\overline{CE}$ signal, are integrated on the Intel386SL.

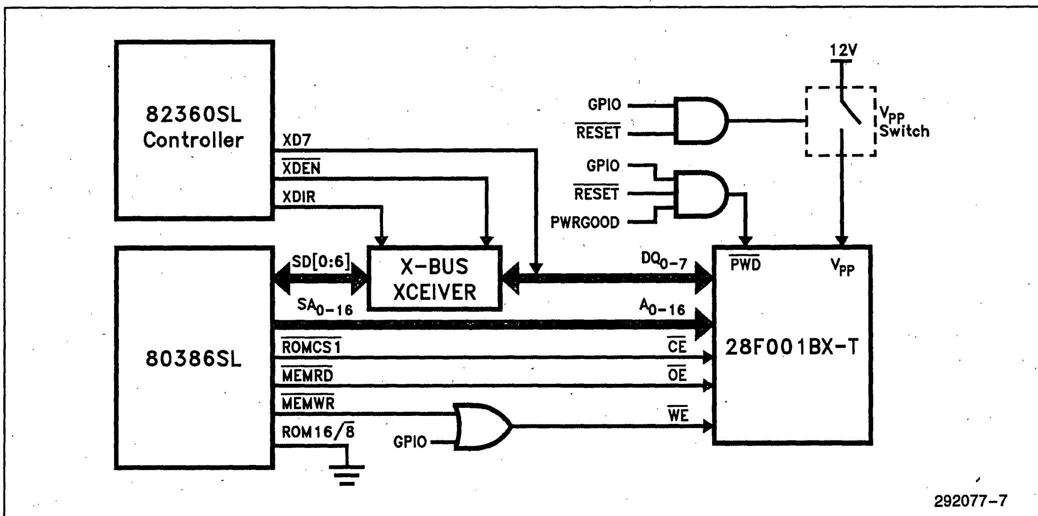


Figure 7. Intel386SL Microprocessor Superset with 28F001BX-T Flash BIOS

The $\overline{PWD}$ pin is gated by a power good signal to ensure control logic integrity before writes to the 28F001BX-T are allowed. It is also gated by System Reset, to abort program or erase if required, and by a separate General Purpose Input-Output (GPIO) line to power down the device once BIOS is shadowed to RAM. CMOS logic will guarantee lowest power dissipation.

Similarly, system 12V is gated both by System Reset and a GPIO line. Software can switch 12V to the 28F001BX-T only when programming or erasing it, minimizing system power consumption. V_{PP} generation and switching methods are discussed in Section 3.3.

Application code, assuming a ROM in the BIOS socket, is sometimes designed to write to BIOS locations to generate software timing delays (versus using NOPs). Gating $\overline{WR}$ to the flash memory with a GPIO line disables writes until desired by BIOS update software.

3.1 BIOS Boot Code Requirements and System Configurations

The previous design assumed that shadow RAM was available in the system. Referencing Figure 8, we see that the BIOS is actually stored in the main block of the 28F001BX, from system address E0000H–FBFFFH. In this scenario, the processor jump vectors, BIOS checksum and recovery code are stored in the 8 KByte boot block. This is the area the processor will jump to on powerup or after reset. The boot block code will execute a checksum check of the main block for a valid BIOS. If successful, the processor will check system RAM, copy the main block code to high memory DRAM and jump to this area for the remainder of Power On Self Test (POST), as well as further BIOS calls. Optionally, the 28F001BX can then be disabled for power savings.

If BIOS checksum determines an invalid BIOS, the system RAM and floppy drive (or possibly modem) are initialized using the boot block recovery code. The system requests (through screen display or speaker "beeps") that the user install the BIOS update floppy disk. A search of the floppy disk is made for a specific file name, and once found, update code is used to re-initialize the main BIOS block. System reboot restores normal operation. Alternatively, the BIOS recovery code can contain specific, non-DOS sector/track information pertaining to the location of the new BIOS update file. Thus, the file is protected and not readable to basic DOS users.

If ROM BIOS disable is overriden by system software or the user (through setup utility), the design must compensate for the altered BIOS location to prevent BIOS calls jumping to incorrect code locations. The following two methods provide alternative solutions for the system designer.

Address Shift Configuration

In this scenario, after BIOS initialization is complete, a write to a latch, register or flip-flop shifts addresses for future BIOS code fetches by 16 KBytes. This allows the system to correctly access the main block and bypass parameter and boot blocks. A system reset or loss of power clears this latch, allowing booting from the boot block once again. Figure 9 shows the input and output signals required for a μPLD address shifter. It shifts addresses in the range FFFFFH–E4000H (112 KBytes) to FBFFFH–E0000H.

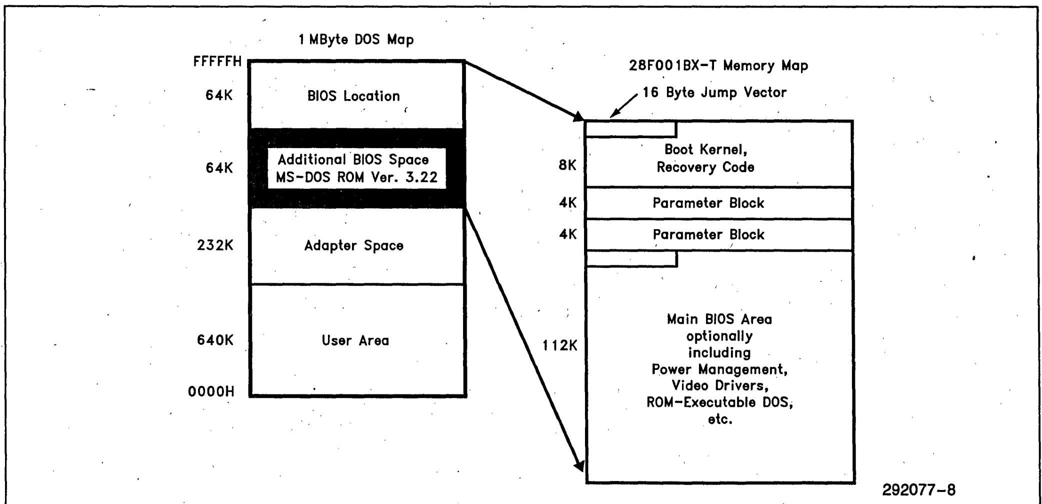


Figure 8. 28F001BX-T in 1 MByte DOS Memory Map

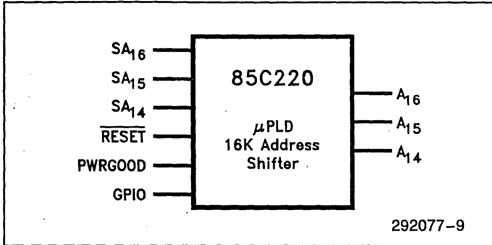


Figure 9. Address Shift Circuitry

Address Inversion Configuration

Figure 10 presents an alternative approach to configuring the 28F001BX in the system memory map. Simple inversion of address line A_{16} to the 28F001BX moves the boot block to the lower half of flash memory as seen by the system. In normal operation, the processor boots and executes from the main array areas, which store the system BIOS, video BIOS and/or DOS in ROM.

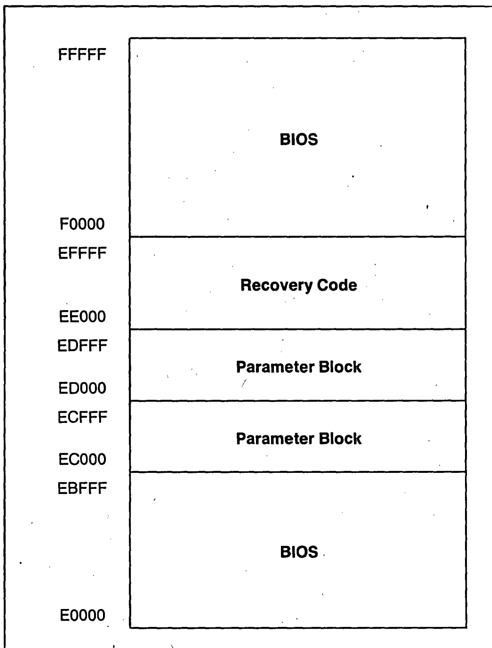


Figure 10. Inversion Configuration (Normal)

If power loss aborts a BIOS update, the main array block will be partially programmed/erased and the code in this block unusable. The system will "hang" or not boot at all. To boot from the boot recovery block, restore address A_{16} polarity, producing the memory map shown in Figure 11. A keyboard sequence, switch on the back of the PC or jumper on the motherboard can toggle A_{16} restore logic and "un-invert" it. After reconfiguration, the processor boots from the boot block and executes its recovery algorithm to restore main array block contents. Re-inverting A_{16} reinstates normal system bootup and operation.

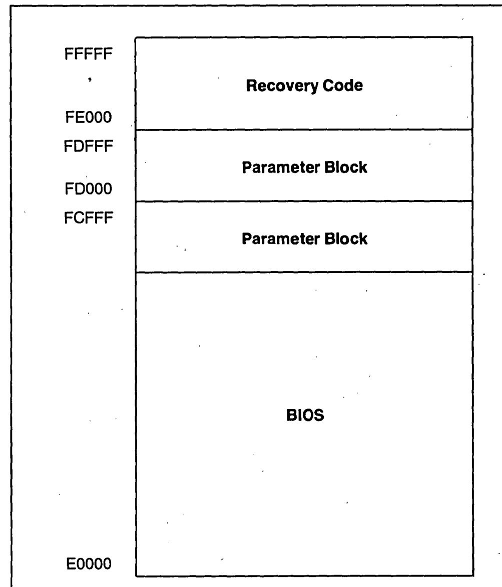


Figure 11. Inversion Configuration (Recovery)

Since standard BIOS code does not support boot block recovery, your BIOS software engineers must design the recovery code for the 8 KByte block. See Section 4.6 for a flowchart of an example recovery algorithm. Third-party BIOS vendors, working with Intel, have also developed recovery code for the 28F001BX (see Appendix C). With the exception of this recovery section, the rest of the BIOS remains the same.

3

3.2 V_{PP} Generation

For flash BIOS designs, the 12V V_{PP} can be provided by:

1. Using the existing 12V supply from PC Power Supply, or
2. Generating 12V using a charge pump or DC–DC converter from the 5V supply.

Flash typically requires only 10 mA for program or erase (30 mA max); otherwise only 10 μA is drawn in standby mode, and 0.8 μA in deep powerdown mode.

Using System 12V Directly

As stated earlier, the IBM PC technical reference manual specifies the 12V supply as +5% and −4%, which meets the Intel Flash Memory V_{PP} requirement. If your power supply meets this condition and has CMOS logic, 12V from the PC power supply can be tied directly to flash memory, eliminating the need to add extra circuitry for V_{PP} generation. This is possible due to the $\overline{PWD}$, V_{CC} and V_{PP} write lockout protection offered in the 28F001BX.

Pumping 5V to 12V

If your system does not provide 12V or does not meet flash memory specifications, several 5-to-12V converters are available, including surface-mount versions. Application Note AP-316, available from your local Intel Sales Office, lists several V_{PP} solutions which offer on/off control of V_{PP} and provide a steady V_{PP} rise and little overshoot. Figure 12 shows one example. On power-up, system reset or when V_{CC} is below 4.5V, V_{PP} is forced off. It is enabled (or disabled) by writing to the $\overline{V_{PP}EN}$ I/O port address. On/off capability is essential for battery-operated equipment and eliminates the need for $\overline{WE}$ filtering. The $\overline{V_{PP}EN}$ signal "OR'ed" with the system memory write ($\overline{MEMWR}$) functions as the clock signal for the 74FC74 D-flip-flop. The D-input is latched when $\overline{MEMWR}$ goes high. Writing a one or a zero turns V_{PP} on or off, respectively.

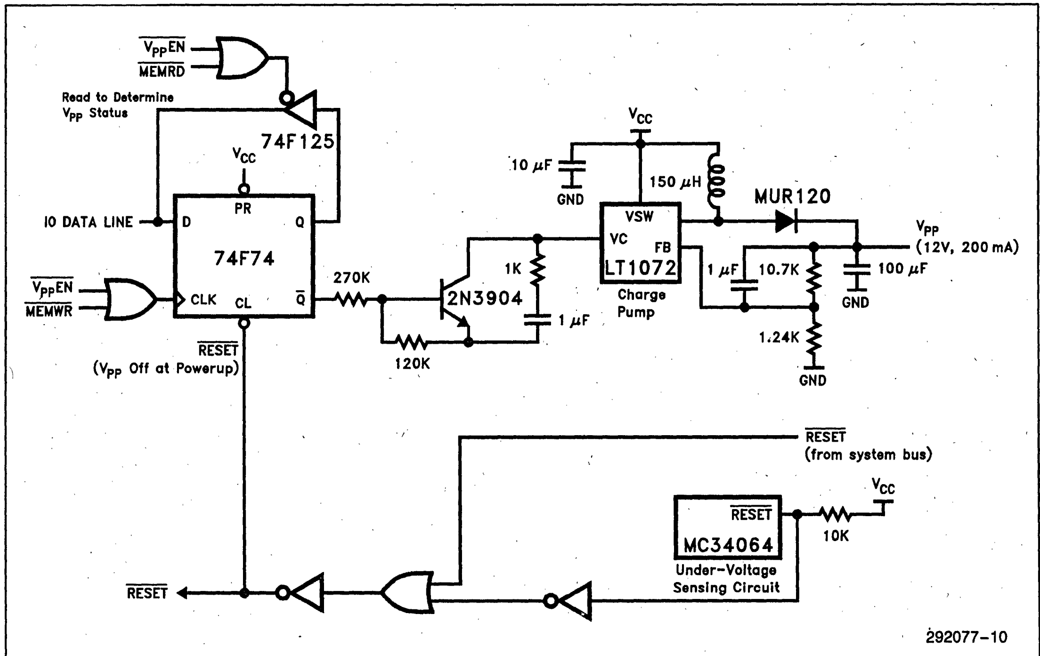


Figure 12. V_{PP} Generation with Write Protection

Linear Technology's LT1072, a switching regulator, is used as a 5V to 12V charge pump. The 10.7K and 1.24K resistors are used to establish the correct reference voltage to obtain 12V. The 100 μF capacitor at the output can handle up to 200 mA. For a single- or double-chip BIOS design, this capacitor value can be halved or even quartered to allow selection of a SMT capacitor value, since the maximum I_{PP} current per device is only 30 mA (10 mA typical). Allow sufficient time when switching V_{PP} on, letting the charge pump level out and enabling the Command Register to receive program or erase commands. The diode, MUR120, keeps the inductor from absorbing current from the charged output capacitor.

Security

Controlling V_{PP} provides the benefit of system hardware security. Beyond this, you can design for even higher security levels. The first level could be the design of a simple software password routine that would only turn on V_{PP} when a correct password is given. Alternatively, you can provide a jumper to allow 12V to the part for a BIOS update and then return it when reprogramming is finished. The system should check this pin to see if the jumper was left in the programming position and remind the user to move it. Unless V_{PP} is at 12V, the flash memory contents cannot be changed and acts just like ROM. Disabling V_{PP} until voltages have stabilized provides additional power-up protection.

The Motorola component, MC34064, is an under-voltage sensing circuit that begins functioning when V_{CC} is above 1V. Between 1V and 4.6V, the $\overline{RESET}$ output is active. This (or a system $\overline{RESET}$) clears the 74FC74, keeping V_{PP} off. Alternatively, if you use CMOS logic, you could make use of Intel's flash memory V_{CC} and V_{PP} lockout functions. While V_{CC} is below 2.5V, the Command Register is locked out. Since CMOS control logic is active at 2.0V, a 0.5V safety margin exists for control logic to settle down before the part becomes active. Program and erase attempts are inhibited with V_{PP} below 6.5V. For both CMOS and non-CMOS designs (i.e., control logic active at 2.0V), gate $\overline{PWD}$ with the power supply's "Power Good" signal or the MC34064's $\overline{RESET}$ output (Figure 13). Until $\overline{PWD}$ transitions to V_{IH}, the part ignores all write attempts, regardless of power supply voltages and bus activity.

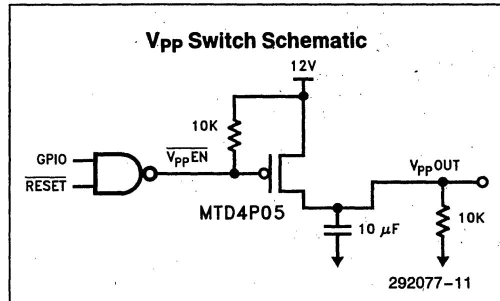


Figure 13. V_{PP} Switch Using MTD4P05

Using a MOSFET Switch

For laptops or palmtops, an always-active 12V may not provide acceptable power management. For these systems, a MOSFET switch will toggle 12V to the flash memory, minimizing current draw when not needed. Several DC switches exist, but there are a few issues to consider in your selection. Choose a switch with low "ON" resistance to keep the V_{PP} voltage within flash memory tolerances. The system 12V power supply must be specified to a tighter range to allow for any voltage drop through the switch. Allocate an I/O line (V_{PP} enable) to turn the switch on and off. To handle "warm RESETS", the V_{PP} enable must be gated with the system $\overline{RESET}$ line. The Motorola MTD4P05 is one example of a surface-mount switch with low drain-source resistance. Assuming a 12V + 5% and −4% supply:

R_{DS} = 0.6Ω

I_{PP} = 30 mA (Worst Case)

ΔV_{SWITCH} Drop = (30 mA x 0.6Ω) = 0.02V, $\ll$ (4% of V_{PP} =) 0.48V.

Figure 13 shows a schematic of a V_{PP} switch design.

3.3 Modifying an Existing Motherboard

EPROM/ROM Designs

If you are modifying an existing motherboard design for a flash memory BIOS, there are a few things you should consider. First, check the logic design to determine if $\overline{WR}$ is decoded and connected to the BIOS EPROM location. Typical motherboard logic designs do not allow writes to the EPROM locations and treat EPROM writes as invalid (e.g., $\overline{ROMCS}$ not generated with $\overline{MEMWR}$). This is overcome by generating the BIOS location's $\overline{WR}$ externally by either adding the necessary discrete logic or adding a 3-to-8 decoder (see Figure 14 for an example). In either case, tap into the M/$\overline{IO}$ and $\overline{WR}$ control lines and configure the decoder to provide a logic low for the M "AND" $\overline{WR}$ "AND" BIOS address condition.

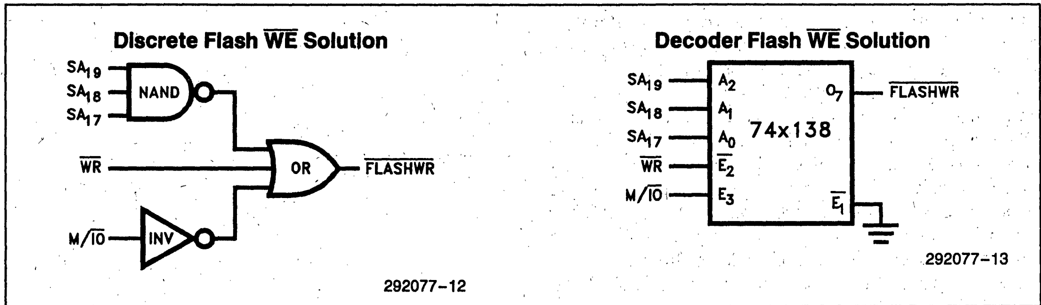


Figure 14. Discrete and Single-Chip Decoder $\overline{WE}$ Solutions

Secondly, check to see if the BIOS code transceiver or buffer for the EPROM location works in both directions. The transceiver may need a special BIOS call to unlock it in the "write" direction, or you may have to reprogram the logic for that portion of your board. If your chip set data buffer works only in one direction, a transceiver and direction logic must be added to the CPU bus to pass data to and from flash memory.

Your system must also be capable of routing 12V to the BIOS socket for program and erase. Optionally, provide capability for deep powerdown mode through $\overline{PWD}$ (see Section 3.0), or simply tie $\overline{PWD}$ to V_{CC} via a jumper, blue wire or trace modification to the motherboard. Finally, address inversion or shift mechanisms outline in Section 3.1 can optionally be added for recovery capability with the 28F001BX.

28F010 Flash Memory Designs

If your design currently incorporates Intel's 28F010 flash memory, hardware upgrade to the 28F001BX is simple. Transceiver, BIOS write and V_{PP} requirements will have already been considered in the original design. Gate the $\overline{PWD}$ input if powerdown capability is desired, or jumper this pin to V_{CC}. Finally, invert or shift the system addresses as in Section 3.1 if BIOS "ROM" access after shadowing to DRAM is anticipated.

3.4 In-System Write vs On-Board Programming

When devices are soldered directly to a printed circuit board, one of two sources control flash memory reprogramming:

1. the system's own processor, or
2. a PROM programmer connected to the board.

These options are called In-System Write (ISW) and On-Board Programming (OBP), respectively. Their respective benefits are discussed in detail in AP-316.

With ISW, the system drives the reprogramming process and generates V_{PP} locally. Under this scenario, the board manufacturer will initially program at least the boot block in a PROM programmer. This removes the need for circuitry on-board to unlock the boot block, guaranteeing boot code integrity throughout system life. A good design practice for ISW-type designs is to socket the first few flash BIOS prototypes. SMT-only designs can also socket using PLCC SMT sockets. Socketing enables the system designer to easily work out any bugs with in-system flash reprogramming by allowing the removal of a flash part for external reprogramming in a PROM programmer. Once ISW reprogramming is fully debugged, pre-programmed flash parts can be soldered directly to the circuit board without a socket. All flash memory components are exposed to a data-retention bake testing and checked for any data loss before shipping. It is extremely unlikely that data in a production flash device can be corrupted from heat by a production-run soldering application.

OBP uses an external board programmer to supply V_{PP} and V_{HH} and control the programming process. Certain design considerations must be evaluated prior to laying out the design. Some manufacturers using TSOP may also want to remove a handling step from the manufacturing process by providing the capability to program flash for the first time after being soldered directly onto the circuit board. OBP can accomplish this if the design is first laid out correctly to support OBP. External circuitry generates voltages needed to unlock and program/erase the boot block.

3.5 Ideas for Using Extra Adaptor Space

Laptop and palmtop systems may have adaptor space available in the system memory map since there typically isn't much room for add-in boards. Additionally, they may not use up the entire 128K of BIOS space due to their fixed feature set and limited upgrade capability. This extra memory space can hold ROM executable programs like Lotus 123, WordPerfect, Microsoft Works, etc. Using Intel's flash TSOPs, a small application cache can reduce a laptop's disk access and increase battery life.

Additionally, ROM-Executable DOS can be placed anywhere in adapter space. For example, MS-DOS ROM Version 3.22 requires 62 KB of adaptor space today (this may change on subsequent revisions). One location for MS-DOS ROM Version 3.22 is directly under the BIOS (again see Figure 8). Today's typical BIOS consumes 64 KB or less; consequently, both the BIOS and MS-DOS ROM Version 3.22 could reside in a single 28F001BX (128 KBytes), yielding reduced chipcount. However, if power management code is added to the BIOS, system BIOS code could grow to 80 KB or more. Therefore, designs that include both power management and MS-DOS ROM Version 3.22 should consider using both a 28F001BX and 28F010 flash device (or two 28F001BX's). This leaves extra space for BIOS and MS-DOS ROM to grow in the design, while providing additional storage for the video BIOS.

4.0 SOFTWARE DESIGN CONSIDERATIONS

Intel's Flash Memory provides a cost-effective, updatable, nonvolatile code storage medium. The 28F001BX integrates the Quick-Pulse Programming and Quick-Erase algorithms of prior Intel Flash Memories on-chip, using the Command Register, Status Register and Write State Machine (WSM). On-chip integration dramatically reduces system overhead, simplifies system software creation and debug and provides SRAM-like timings to the Command and Status Registers. WSM operation, internal program/erase verify and V_{PP} high voltage presence are monitored and reported via appropriate Status Register bits. Table 1 lists the 28F001BX command set, while Table 2 details the Status Register bits and their meanings.

Table 1. 28F001BX Command Definitions

Command	Bus Cycles Req'd	Notes	First Bus Cycle			Second Bus Cycle		
		1, 2	Operation	Address	Data	Operation	Address	Data
Read Array/Reset	1		Write	X	FFH			
Intelligent Identifier	3	1, 2, 3	Write	X	90H	Read	IA	IID
Read Status Register	2	2	Write	X	70H	Read	X	SRD
Clear Status Register	1		Write	X	50H			
Erase Setup/Erase Confirm	2	1	Write	BA	20H	Write	BA	D0H
Erase Suspend/Erase Resume	2		Write	X	B0H	Write	X	D0H
Program Setup/Program	2	1, 2	Write	PA	40H	Write	PA	PD

NOTES:
1. IA = Identifier Address; 00H for manufacturer code, 01H for device code.
   BA = Address within the block being erased.
   PA = Address of memory location to be programmed.
2. SRD = Data read from Status Register. See Table 2 for a description of Status Register bits.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of $\overline{WE}$.
   IID = Data read from intelligent identifier.
3. Following the intelligent identifier command, two read operations access the manufacturer and device codes.
4. Commands other than shown above are reserved by Intel for future device implementations and should not be used.

Table 2. 28F001BX Status Register Definitions

WSMS	ESS	ES	PS	VPPS	R	R	R
7	6	5	4	3	2	1	0

SR.7 = WRITE STATE MACHINE STATUS
1 = Ready
0 = Busy

SR.6 = ERASE SUSPEND STATUS
1 = Erase Suspended
0 = Erase In Progress/Completed

SR.5 = ERASE STATUS
1 = Error In Block Erase
0 = Successful Block Erase

SR.4 = PROGRAM STATUS
1 = Error In Byte Program
0 = Successful Byte Program

SR.3 = V_{PP} STATUS
1 = V_{PP} Low Detect; Operation Abort
0 = V_{PP} OK

NOTES:

The Write State Machines Status Bit must first be checked to determine program or erase completion, before the Program or Erase Status bits are checked for success.

If the Program AND Erase Status bits are set to 1's during an erase attempt, an improper command sequence was entered. Attempt the operation again.

If V_{PP} low status is detected, the Status Register must be cleared before another program or erase operation is attempted.

The V_{PP} Status bit, unlike an A/D converter, does not provide continuous indication of V_{PP} level. The WSM interrogates the V_{PP} level only after the program or erase command sequences have been entered and informs the system if V_{PP} has not been switched on. The V_{PP} Status bit is not guaranteed to report accurate feedback between V_{PPL} and V_{PPH}.

SR.2–SR.0 = RESERVED FOR FUTURE ENHANCEMENTS
These bits are reserved for future use and should be masked out when polling the Status Register.

The WSM on-chip oscillator internally times the program/erase algorithms, making software timers unnecessary. Block precondition is also controlled by the WSM as part of the erase algorithm. Block data programming to "0's" before erasing is no longer needed.

Intel's high quality design, manufacturing and testing result in outstanding reliability and performance throughout device life. Although Program Status and Erase Status bits are provided for Status Register completeness, errors will probably not be encountered, if proper V_{PP} levels and software sequences are implemented.

Intel offers standard software drivers, written in "C", to assist software engineers implementing 28F001BX reprogramming for update utilities. These high-level routines, found in Appendix A, are adaptable to a wide range of μP and μC platforms and system architectures.

Covered in this section are the major software steps for a flash BIOS update utility:

- Update software for a modified system
- Pseudo-Code overview
- Initializing the system
- Code loader routine
- Flash re-programming
- Recovery routines
- Power management

4.1 Update Software for a Modified System

The design example of Section 3.0 assumes BIOS shadowing for BIOS code execution while allowing BIOS writes to the flash socket. Many systems provide a register which enables BIOS writes and reads. Some systems may not allow BIOS reads from RAM while performing BIOS writes to the flash socket, or vice versa. The reasons may be simple; no shadow RAM exists in the system (8088 or 8086 systems), or system logic treats "ROM writes" as an invalid operation. In these cases, perform all your required BIOS calls before you erase and program the flash memory. But keep in mind, to update the user on the progress of flash programming and indicate when programming is finished, you should add some basic screen or speaker "beep" routines to your update utility.

4.2 Pseudo-Code Overview

The following pseudo-code for an update utility provides a brief description of the process of updating a BIOS in-situ. It is based on software developed by a customer for a PC platform with BIOS update capability. This Intel386-33 MHz system uses the 28F001BX for BIOS storage. Modify the flowchart below, if needed, for your particular chipset and hardware environment.

Pseudo-Code for Flash Update Routine

Initialize system (set up user screen, check battery power, check device ID)

Get BIOS file options (from floppy or modem)
    If no file present
        Send error message to insert BIOS update floppy, or press ESC to exit

Display BIOS update files, prompt user for choice and load to memory
    If file invalid,
        Prompt for file or exit

Inform user what is about to happen, with option to continue or exit
    If user continues, inform them to not turn off the power or soft-reboot (CNTL-ALT-DEL)

Erase 28F001BX main/parameter blocks
    If system interrupt occurs
        Suspend erase if flash memory access is required
            Resume Erase

Write file[s] into flash memory

Indicate to user that flash reprogramming is over

Reboot the system

4.3 Initializing the System

Checking Power

If your application is a laptop or palmtop computer, first check the battery to make sure there is enough power to do the update. If not, inform the user to recharge the system before continuing the update and exit the update program. This ensures that the system won't stop in the middle of an update. Next, initialize access to flash for reads and writes, then try reading the device ID through the Command Register. Checking the device ID before programming or erasing helps determine if reads and writes work correctly and that the flash memory in the system matches your code before starting to reprogram the part. The manufacturer ID for Intel flash memories is **89H** (10001001), located at device address 00000H. Device IDs are located at address 00001H; the ID for the 28F001BX-T is **94H** (10010100), and the 28F001BX-B device ID is **95H** (10010101). These device addresses, in the DOS memory map, correspond to system addresses E0000H (mfgr. ID) and E0001H (device ID). If A_{16} inversion is used as described in Section 3.1, system addresses for mfgr. ID and device ID under normal operation are F0000H and F0001H.

NOTE:
During the initialization, you can also perform a scan of the adaptor space to ascertain if there is more flash in the system. Other Intel Flash Memories share common manufacturer IDs but have unique device IDs, listed below:

Device	Device ID (Hex)	Device ID (Binary)
28F256A	B9H	10111001
28F512	B8H	10111000
28F010	B4H	10110100
28F020	BDH	10111101
28F001BX-T	94H	10010100
28F001BX-B	95H	10010101

4.4 Code Loader Routine

The update utility described in the previous section provides an optional mouse-driven color graphical user interface (GUI) and allows not only BIOS update to the main block but also update of the parameter blocks, and copy/compare of block data to a DOS file. These types of features convey to the end user the ease and simplicity of performing a BIOS update. For example, the main block update utility lists all possible BIOS files in the selected drive and directory, and prompts the user for the desired file. System OEMs may want to encode a specific BIOS file name into the generic loader utility ".COM" or ".EXE" file. This allows automatic reading of the new BIOS file into a program buffer, bypassing the user prompt.

Once the file is loaded into RAM, the routine informs the user of the impending BIOS update and provides the option to exit if desired. If continued, it warns the user to not turn off power or reboot during the BIOS update procedure. It then erases and reprograms the main block with new BIOS data, notifies the user of successful update and reboots.

4.5 Flash Reprogramming Routines

On-Chip Erase Algorithm

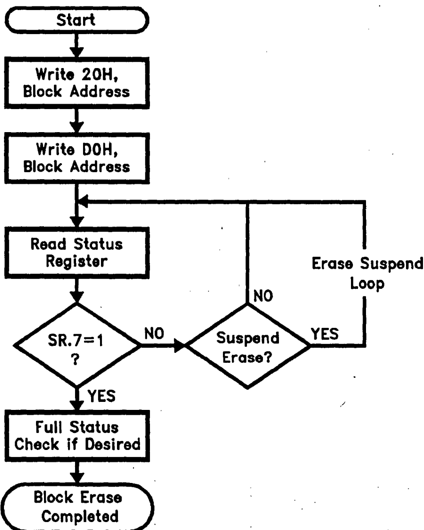
The 28F001BX system erase algorithm is shown in Figure 15. Note that the actual device erase algorithm (Quick-Erase) is controlled internally, including all timing and block preconditioning. This provides the same high level of reliability proven on Intel's ETOX II technology, while reducing system debug efforts. Erase progress is reported to system software thru specific Status Register bits. The 28F001BX erases all bits of a block in parallel. Minimum and typical erase times for each block are listed below:

Block	Minimum Time (Sec)	Typical Time (Sec)
Parameter (ea.)	1.3	2.1
Main	3.0	3.8
Boot	1.3	2.1

The actual erase time depends on the V_{PP} voltage level (11.4V–12.6V), temperature and the number of erase cycles already completed on the part. System software must comprehend adequate time for V_{PP}, after enabled, to ramp to 12V before erase is attempted. Capacitors on the V_{PP} bus, in addition to the intrinsic pump nature of many 12V solutions, cause an RC ramp. Systems that direct-wire 12V need not worry about this delay.

Erase Suspend/Resume

Erase suspend gives the user the ability, while erasing a block of the 28F001BX, to read data or execute code from another block. This capability, in conjunction with the minimal system overhead provided by the WSM, makes disabling of interrupts during block erase unnecessary. Once given the erase suspend command, the WSM halts, reports suspend status to the Status Register and allows array reads. When issued erase resume, it proceeds at the point where it was suspended. Figure 16 details the system code flowchart that suspends and resumes erase.



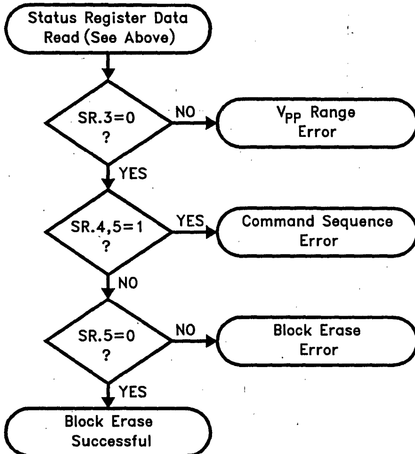
Bus Operation	Command	Comments
Write	Erase Setup	Data = 20H Address = Within Block to be erased
Write	Erase	Data = D0H Address = Within Block to be erased
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent blocks.

Full status check can be done after each block or after a sequence of blocks.

Write FFH after the last block erase operation to reset the device to Read Array Mode.

FULL STATUS CHECK PROCEDURE



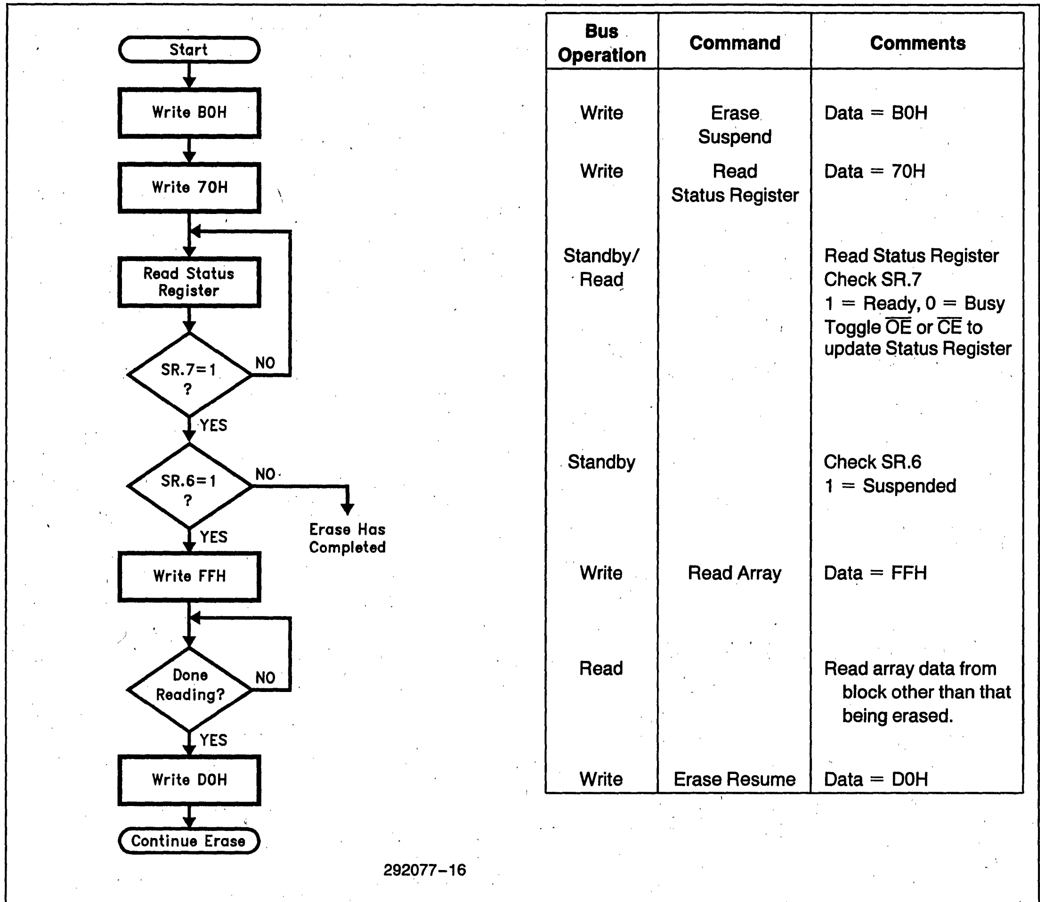
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4, 5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error

SR.3 MUST be cleared, if set during an erase attempt, before further attempts are allowed by the Write State Machine.

SR.5 is only cleared by the Clear Status Register Command, in cases where multiple blocks are erased before full status is checked.

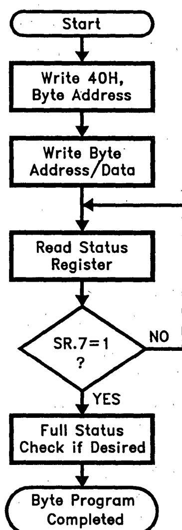
If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 15. 28F001BX Block Erase Algorithm



Bus Operation	Command	Comments
Write	Erase Suspend	Data = B0H
Write	Read Status Register	Data = 70H
Standby/ Read		Read Status Register Check SR.7 1 = Ready, 0 = Busy Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.6 1 = Suspended
Write	Read Array	Data = FFH
Read		Read array data from block other than that being erased.
Write	Erase Resume	Data = D0H

Figure 16. 28F001BX Erase Suspend/Resume Algorithm



292077-17

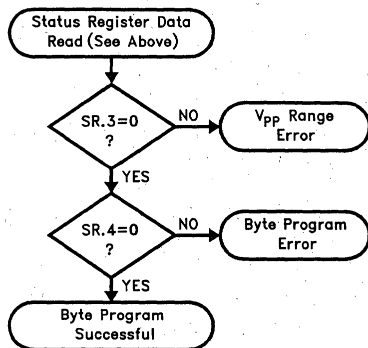
Bus Operation	Command	Comments
Write	Program Setup	Data = 40H Address = Byte to be programmed
Write	Program	Data to be Programmed Address = Byte to be programmed
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent bytes.

Full status check can be done after each byte or after a sequence of bytes.

Write FFH after the last byte programming operation to reset the device to Read Array Mode.

FULL STATUS CHECK PROCEDURE



292077-18

Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Byte Program Error

SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple bytes are programmed before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Figure 17. 28F001BX Byte Programming Algorithm

On-Chip Programming Algorithm

As with 28F001BX erase, the Intel flash Quick-Pulse algorithm is internally controlled by the WSM. Figure 17 shows a system software flowchart for the Command Register/Status Register interface. Minimum and typical programming times (per byte) are 15 μs and 18 μs, respectively. Actual time varies with V_{PP}, temperature and cumulative programming cycles on the device. Ensure that stable 12V is applied to the device before attempting byte programming.

Full Status Checks

After polling the Status Register and determining that the WSM is again READY, system software should further analyze the Status Register to ensure that program or erase has successfully completed. The WSM will return to READY status after program or erase command sequences under any of the following conditions:

- Program/erase completed successfully,
- V_{PP} transition below specification during the program/erase attempt,
- Improper sequence of erase setup/confirm commands to the WSM, or
- Inability to erase the specified block, or program the desired byte.

Figures 15 and 17 detail the additional Status Register data analysis to ensure that program or erase have successfully occurred.

4.6 Recovery Routine Overview

Unsuccessful BIOS update can occur for any of the reasons listed below:

1. V_{PP} transitions out of specified tolerance during program or erase.
2. Incorrect code in the update BIOS disk file, or damaged BIOS disk.
3. Loss of system power during program or erase.
4. System reset (such as reboot) during program or erase.

The Status Register, through bit 3, reports V_{PPH} loss to system software. The BIOS update utility can detect scenario 1 and recover by simply re-attempting block update.

A checksum of update BIOS code after copy from disk to RAM, before flash erase and reprogram, will eliminate error caused by scenario 2.

PC motherboard logic should gate the 28F001BX $\overline{PWD}$ pin with both POWER GOOD and $\overline{RESET}$ signals, to abort program or erase attempts if either scenario 3 or 4 were to occur. This allows the processor to execute code out of the 8 KByte boot block upon system recovery. System reset or loss of system power will clear the Status Register to value 80H and leave the block being updated partially programmed or erased. As detailed previously in Section 3.1, a checksum of the main block will alert the system to an incomplete BIOS. Recovery is achieved by the following or similar steps:

- Initialize CPU and system logic.
- Initialize the system floppy disk.
- Prompt the user to insert a BIOS diskette, through speaker "beep".
- Erase and reprogram the main and/or parameter blocks with file data.
- Reboot

4.7 Power Management

Battery-powered PCs incorporate a variety of techniques to prolong system life between recharges. Typically, power management software senses user inactivity and shuts off power-intensive sections of the system. Options include:

- Display powerdown
- Disk/hard drive powerdown
- System clock slowdown or suspend, and
- Powerdown of non-volatile circuitry in the system.

The 28F001BX fits the latter description. When the $\overline{PWD}$ pin transitions to GND, the device enters an ultra-low power mode, typically consuming 0.25 μW thru V_{CC}. This technique can also be used to power down the BIOS memory after BIOS code has been shadowed to DRAM, if available in the system. When not programming or erasing the 28F001BX, the system should shut off 12V V_{PP} to the part to minimize current draw through this supply.

User inactivity is typically detected if the keyboard has not been used, or the disk drive has not been accessed, for a predetermined interval (this is often user-programmable). Power management software must ensure that a BIOS update is not occurring, before powering down the 28F001BX, to prevent incomplete update.

For more information on power management techniques, consult datasheets and application notes on the Intel386SL microprocessor superset.

5.0 SUMMARY

5.1 Traditional BIOS Storage and Disadvantages

Traditional BIOS storage has been in EPROM, which offers nonvolatility and factory programming capability. In earlier PCs, the BIOS code was fairly simple (relative to today's software) and updates were infrequent, so EPROMs or ROMs were an acceptable BIOS storage medium. Today's systems are much more sophisticated, with many designs supporting the Intel i386/i486™ microprocessors and new bus architectures like MCA and EISA for the first time. These new buses allow peripherals to take control of the system bus . . . it is difficult to guess what new system configurations might emerge. Therefore, the potential for a change in the BIOS code is much greater and the frequency of change is likely to increase.

A system designer may use EPROMs for BIOS storage to reduce initial system (component) costs, but the long-term update cost is much more than the difference between EPROM and flash memory components. A major manufacturer of PCs has estimated that a service call for a BIOS update with EPROMs can cost upwards of $300.00 for **ONE** update at **ONE** site. EPROMs are also susceptible to bent leads during insertion by the technician, or more likely, the end user. Service is becoming a key differentiator between the multitudes of PC makers. Reducing the number of times a PC has to be opened for any reason and providing improved service increases customer confidence and promotes a reliable image.

5.2 Advantages of an Updatable BIOS

Using flash memory for BIOS storage provides a flexible code medium that allows the BIOS code to adapt to changing hardware and software conditions. BIOS updates in flash are inexpensive, via a floppy disk or modem. They remove EPROM inventories, reduce packaging requirements, reduce total postage costs and eliminate service cost for BIOS code updates by removing the need for a technician to do the update. A company that supports mutiple OEMs can improve version management control by using a flash BIOS and floppies or a BBS for updates. An additional benefit is that not only the BIOS, but DOS itself can be stored in the same flash memory device.

5.3 Advantages of Adding DOS in FLASH

Once the requirements for flash memory BIOS are met, the capability is also in place for adding DOS in FLASH. Why put DOS in FLASH? For laptop and palmtop PCs, battery longevity is of paramount concern, followed closely by weight and increasing user RAM (640 KB) space. Extra user RAM is needed for applications that require more than the typical 570 KBytes (640 KB–70 KB) available with disk-based DOS. Digital Research Incorporated and Microsoft both make "DOS-in-ROM" products that address these needs. MS-DOS ROM version 3.22 is an example.

Microsoft's MS-DOS ROM Version 3.22 is a full-function version of MS-DOS 3.2. It features instant-on and employs only 15 KB of the 640 KB DOS RAM user space, leaving the rest for applications. Since MS-DOS ROM Version 3.22 loads from adaptor space, both disk access and DOS loadtime are reduced. For laptops, anything that can reduce disk access equates to battery longevity. Laptops can reduce weight by using MS-DOS ROM Version 3.22 and replacing the floppy drive with an IC card. Adding MS-DOS ROM Version to desktops also liberates additional user RAM for the same above reasons, but may not be optimal for high speed 32-bit systems.

All future versions of MS-DOS will be supported with equivalent versions of MS-DOS ROM. See Appendix B for more information.

5.4 Advantages of Adding 1 MB–4 MB of Resident Code Storage

There is a growing need for systems to be able to provide a small suite of bundled applications. Benefits to the user are faster application execution thru reduced hard or floppy disk access, no power used to store the resident code, and instant-on. No time is wasted transferring data over a disk I/O interface. The code is instead loaded to RAM with a simple memory copy function or procedure. In some cases, code is directly executed by the processor. Tandy's Deskmate is an example of such a system. Future versions of Deskmate-like user interfaces could easily be made flash-updatable. SRAM is too expensive and requires power to just store files. Furthermore, battery backup is not a reliable means of achieving nonvolatility. Intel's Flash Memory can provide user configurability for 1 MB–4 MB of code storage for just 2x–3x the cost of EPROMs and less than half the cost of SRAM. Applications such as Lotus 123, WordPerfect and Microsoft Works also come in either a direct-execute "ROM" version or a load-from-ROM format. Many other ROM application software packages are in development, servicing the successful and growing needs of the laptop/palmtop computers. Therefore, if an application can be stored or runs from ROM, it can be stored and run from flash. As software packages are periodically updated, flash memory provides the capability of updating these "ROM" applications at little cost to the software vendor and with no system disassembly required.

APPENDIX A
SOFTWARE ROUTINES

```
/*********************************************************************************/
/*  Copyright Intel Corporation, 1991                                            */
/*  Brian Dipert, Intel Corporation, July 14, 1991, Revision 1.4                 */
/*  The following drivers control the Command and Status Registers of            */
/*    the 28F001BX Flash Memory to drive byte program, block erase, Status       */
/*    Register read and clear and array read algorithms.                         */
/*    Sample Vpp and /PWD control blocks are also included.                      */
/*  The functions listed below are included:                                     */
/*    erasbgn():  Begins block erasure                                           */
/*    erassusp(): Suspends erase to allow reading data from a block of the       */
/*                28F001BX other than that being erased                          */
/*    erasres():  Resumes erase if suspended                                     */
/*    end():      Polls the Write State Machine to determine if block erase or   */
/*                byte program have completed                                    */
/*    eraschk():  Executes full status check after erase completion              */
/*    progbgn():  Begins byte programming                                        */
/*    progchk():  Executes full status check after byte program completion       */
/*    idread():   Reads and returns the manufacturer and device IDs of the       */
/*                target 28F001BX                                                */
/*    statrd():   Reads and returns the contents of the Status Register          */
/*    statclr():  Clears the Status Register                                     */
/*    rdmode():   Puts the 28F001BX in Read Array mode                           */
/*    rdbyte():   Reads and returns a specified byte from the target 28F001BX    */
/*    vppup():    Enables high voltage Vpph                                      */
/*    vppdown():  Disables Vpph                                                  */
/*    pwdon():    Ramps the /PWD pin to high voltage Vhh, enabling boot block    */
/*                program/erase                                                  */
/*    pwdoff():   Disables high voltage Vhh on /PWD, disabling program           */
/*                and erase of boot block                                        */
/*                                                                               */
/*  Addresses are transferred to functions as pointers to far bytes (ie long     */
/*    integers). An alternate approach is to create a global array the size of the */
/*    28F001BX and locate "over" the 28F001BX in the system memory map. Accessing  */
/*    specific locations of the 28F001BX is then accomplished by passing the chosen */
/*    function an offset from the array base versus a specific address. Different */
/*    microprocessor architectures will require different array definitions; ie for */
/*    the x86 architecture, define it as "byte boot [2][10000]" and pass each    */
/*    function TWO offsets to access a specific location. MCS-51 architectures   */
/*    are limited to "byte boot[10000]"; alternate approaches such as writing to */
/*    control bits will be required to access the full flash array               */
/*                                                                               */
/*  To create a far pointer, a function such as MK_FP() can be used, given       */
/*    a segment and offset in the x86 architecture. I use Turbo-C; see your      */
/*    compiler reference manual for additional information.                      */
/*********************************************************************************/
```

```
/****************************************************************************************/
/*  Revision History: Rev 1.4                                                           */
/*                                                                                      */
/*  Changes from 1.0 to 1.1: Added typedef for "byte" to accurately reflect            */
/*    this x8 device. Altered variable definitions accordingly. Combined                */
/*    functions progend() and erasend() into function end().                           */
/*                                                                                      */
/*  Changes from 1.1 to 1.2: Added this revision history block. Added above            */
/*    comments on alternate addressing methods.                                         */
/*                                                                                      */
/*  Changes from 1.2 to 1.3: Added pass/fail error return from idread(),               */
/*    idread() at beginning of progbgn() and erasbgn(), pass/fail error                 */
/*    return from progbgn() and erasbgn().                                              */
/*                                                                                      */
/*  Changes from 1.3 to 1.4: Revised code to reflect simplified program and            */
/*    erase algorithms. 28F001BX automatically transitions to Read Status Register     */
/*    mode after program command sequence, erase command sequence and remains in       */
/*    Read Status Register mode after Erase Suspend is issued. Address 0000H is no     */
/*    longer required to read or clear the Status Register.                            */
/****************************************************************************************/

typedef unsigned char byte;

/****************************************************************************************/
/* Function: Main                                                                       */
/* Description: Included only to omit errors when attempting to compile code.          */
/*  The end customer would insert their main program here.                             */
/****************************************************************************************/

main()
{
}

/****************************************************************************************/
/*  Function: Erasgbn                                                                   */
/*    Description: Begins erase of a block.                                             */
/*    Inputs: blckaddr: System address within the block to be erased                   */
/*    Outputs: None                                                                     */
/*    Returns: 0 = Erase successfully initiated                                         */
/*             1 = Erase not initiated (ID check error)                                 */
/*    Device Read Mode on Return: Status Register (ID if returns 1)                    */
/****************************************************************************************/

#define ERASETUP    0X20    /* Erase Setup command                                      */
#define ERASCONF    0XD0    /* Erase Confirm command                                    */

int erasbgn(blckaddr)

byte far *blckaddr;         /* blckaddr is an address within the block to be           */
                            /*    erased                                                */

{
 if (idread()==1)           /* ID read error; device not powered up?                    */
     return (1);
 *blckaddr = ERASETUP;      /* Write Erase Setup command to block address               */
 *blckaddr = ERASCONF;      /* Write Erase Confirm command to block address             */
 return (0);
}
```

```
/*************************************************************************************/
/*  Function: Erassusp                                                              */
/*    Description: Suspends block erase to read from another block                  */
/*    Inputs: None                                                                  */
/*    Outputs: None                                                                 */
/*    Returns: 0 = Erase suspended                                                  */
/*             1 = Error; Write State Machine not busy (erase suspend not possible) */
/*    Device Read Mode on Return: Read Status Register                              */
/*************************************************************************************/

#define RDYMASK      0X80    /* Mask to isolate the WSM Status bit of the          */
                             /*      Status Register                               */
#define WSMRDY       0X80    /* Status Register value after masking, signifying    */
                             /*      that the WSM is no longer busy                */
#define SUSPMASK     0X40    /*Mask to isolate the Erase Suspend Status bit of the */
                             /*      Status Register                               */
#define ESUSPYES     0X40    /* Status Register value after masking, signifying    */
                             /*      that erase has been suspended                 */
#define STATREAD     0X70    /* Read Status Register command                       */
#define SYSADDR      0       /* This constant can be initialized to any address    */
                             /*      within the memory map of the target 28F001BX  */
                             /*      and is alterable depending on the system      */
                             /*      architecture                                  */
#define SUSPCMD      0XB0    /* Erase Suspend command                              */

int erassusp()

{
 byte far *stataddr;         /* Pointer variable used to write commands to device   */

 stataddr  = (byte far *)SYSADDR;
 *stataddr = SUSPCMD;        /* Write Erase Suspend command to the device          */
 *stataddr = STATREAD;       /* Write Read Status Register command to 28F001BX     */
 while ((*stataddr & RDYMASK) != WSMRDY)
     ;                       /* Will remain in while loop until bit 7 of the       */
                             /*       Status Register goes to 1, signifying that the */
                             /*       WSM is no longer busy                         */
 if ((*stataddr & SUSPMASK) == ESUSPYES)
            return(0);       /* Erase is suspended ... return code "0"             */
 return(1);                  /* Erase has already completed; suspend not possible.  */
                             /*       Error code "1"                               */
}
```

```
/*************************************************************************************/
/*  Function: Erasres                                                               */
/*    Description: Resumes block erase previously suspended                         */
/*    Inputs: None                                                                  */
/*    Outputs: None                                                                 */
/*    Returns: 0 = Erase resumed                                                    */
/*             1 = Error; Erase not suspended when function called                  */
/*    Device Read Mode on Return: Status Register                                   */
/*************************************************************************************/

#define RDYMASK     0X80    /* Mask to isolate the WSM Status bit of the           */
                            /*      Status Register                                */
#define WSMRDY      0X80    /* Status Register value after masking, signifying     */
                            /*      that the WSM is no longer busy                 */
#define SUSPMASK    0X40    /* Mask to isolate the Erase Suspend Status bit        */
                            /*      of the Status Register                         */
#define ESUSPYES    0X40    /* Status Register value after masking, signifying     */
                            /*      that erase has been suspended                  */
#define STATREAD    0X70    /* Read Status Register Command                        */
#define SYSADDR     0       /* This constant can be initialized to any             */
                            /*      address within the memory map of the target    */
                            /*      28F001BX and is alterable depending on the     */
                            /*      system architecture                            */
#define RESUMCMD    0XD0    /* Erase Resume Command                                */

int erasres()

{
byte far *stataddr;         /* Pointer variable used to write commands to device  */

stataddr  = (byte far *)SYSADDR,
*stataddr = STATREAD;       /* Write Read Status Register command to 28F001BX      */
if ((*stataddr & SUSPMASK) != ESUSPYES)
    return (1);             /* Erase not suspended. Error code "1"                 */
*stataddr = RESUMCMD;       /* Write Erase Resume command to the device            */
while ((*stataddr & SUSPMASK) == ESUSPYES)
    ;                       /* Will remain in while loop until bit 6 of the        */
                            /*      Status Register goes to 0, signifying          */
                            /*      erase resumption                               */
while ((*stataddr & RDYMASK) == WSMRDY)
    ;                       /* Will remain in while loop until bit 7 of the        */
                            /*      Status Register goes to 0, signifying          */
                            /*      that the WSM is once again busy                */
return (0);
}
```

```
/*****************************************************************************/
/*  Function: End                                                            */
/*    Description: Checks to see if the WSM is busy                          */
/*                 (is program/erase completed?)                             */
/*    Inputs: None                                                           */
/*    Outputs: statdata: Status Register data read from device               */
/*    Returns: 0 = Program/Erase completed                                   */
/*             1 = Program/Erase still in progress                           */
/*    Device Read Mode on Return: Status Register                            */
/*****************************************************************************/

#define RDYMASK     0X80    /* Mask to isolate the WSM Status bit of the      */
                            /*      Status Register                          */
#define WSMRDY      0X80    /* Status Register value after masking, signifying */
                            /*      that the WSM is no longer busy           */
#define STATREAD    0X70    /* Read Status Register command                  */
#define SYSADDR     0       /* This constant can be initialized to any       */
                            /*      address within the memory map of the target */
                            /*      28F001BX and is alterable depending on the */
                            /*      system architecture                      */

int end (statdata)

byte *statdata;             /* Allows Status Register data to be passed back */
                            /*      to the main program for further analysis */
{
 byte far *stataddr;        /* Pointer variable used to write commands to    */
                            /*      device                                   */
 stataddr  = (byte far*)SYSADDR;
 *stataddr = STATREAD;      /* Write Read Status Register command to 28F001BX */
 if (((*statdata = *stataddr) & RDYMASK) != WSMRDY)
     return (1);            /* Program/erasure still in progress...code "1"  */
 return (0);                /* Program/erase attempt completed...code "0"    */
}
```

```
/*****************************************************************************/
/*  Function: Eraschk                                                        */
/*    Description: Completes full Status Register check for erase (proper    */
/*      command sequence, VPP low detect, erase success). This routine assumes */
/*      that erase completion has already been checked in function end() and */
/*      therefore does not check the WSM Status bit of the Status Register   */
/*    Inputs: statdata: Status Register data read in function end            */
/*    Outputs: None                                                          */
/*    Returns: 0 = Erase completed successfully                              */
/*             1 = Error; Vpp low detect                                     */
/*             2 = Error; Block erase error                                  */
/*             3 = Error; Improper command sequencing                        */
/*    Device Read Mode on Return: Same as when entered                       */
/*****************************************************************************/

#define ESEQMASK     0X30      /* Mask to isolate the Erase and Program            */
                               /*         Status bits of the Status Register       */
#define ESEQFAIL     0X30      /* Status Register value after masking if erase     */
                               /*         command sequence error has been detected */
#define ERRMSK       0X20      /* Mask to isolate the Erase Status bit of the      */
                               /*         Status Register                          */
#define ERASERR      0X20      /* Status Register value after masking if erase error */
                               /*         has been detected                        */
#define VLOWMASK     0X08      /* Mask to isolate the Vpp Status bit of the Status */
                               /*         Register                                 */
#define VPPLOW       0X08      /* Status Register value after masking if Vpp low   */
                               /*         has been detected                        */

int eraschk(statdata)

 byte statdata;                /* Status Register data that has been already read  */
                               /*         from the 28F001BX in function end()      */

{
 if ((statdata & VLOWMASK) == VPPLOW)
     return (1);               /* Vpp low detect error, return code "1"            */
 if ((statdata & ERRMSK) == ERASERR)
     return (2);               /* Block erase error detect, return code "2"        */
 if ((statdata & ESEQMASK) == ESEQFAIL)
     return (3);               /* Erase command sequence error, return code "3"    */
 return (0);                   /* Block erase success, return code "0"             */
}
```

3

```
/**************************************************************************************/
/*  Function: Progbgn                                                                 */
/*    Description: Begins byte program sequence                                       */
/*    Inputs: pdata: Data to be programmed into the device                            */
/*            paddr: Target address to be programmed                                  */
/*    Outputs: None                                                                   */
/*    Returns: 0 = Program successfully initiated                                     */
/*             1 = Program not initiated (ID check error)                             */
/*    Device Read Mode on Return: Status Register (ID if returns 1)                   */
/**************************************************************************************/

#define SETUPCMD    0X40    /*Program Setup command                                   */

int progbgn (pdata,paddr)

byte pdata;                 /* Data to be programmed into the 28F001BX                */
byte far *paddr;            /* paddr is the destination address for the data          */
                            /*         to be programmed                               */

{
 if (idread() == 1)         /* Device ID read error...powered up?                     */
     return (1);
*paddr = SETUPCMD;          /* Write Program Setup command and                        */
                            /*         destination address                            */
*paddr = pdata;             /* Write program data                                     */
                            /*         and destination address                        */
 return (0);
}
```

```
/**************************************************************************************/
/*  Function: Progchk                                                                */
/*   Description: Completes full Status Register check for byte program (VPP low     */
/*     detect, programming success). This routine assumes that byte program          */
/*     completion has already been checked in function end() and                     */
/*     therefore does not check the WSM Status bit of the Status Register            */
/*   Inputs: statdata: Status Register data read in function end                     */
/*   Outputs: None                                                                   */
/*   Returns: 0 = Byte programming completed successfully                            */
/*            1 = Error; VPP low detect                                              */
/*            2 = Error; Byte program error                                          */
/*   Device Read Mode on Return: Status Register                                     */
/**************************************************************************************/

#define PERRMSK   0X10  /* Mask to isolate the Program Status bit of the              */
                        /*              Status Register                                */
#define PROGERR   0X10  /* Status Register value after masking if program             */
                        /*              error has been detected                        */
#define VLOWMASK  0X08  /* Mask to isolate the VPP Status bit of the Status           */
                        /*              Register                                       */
#define VPPLOW    0X08  /* Status Register value after masking if VPP low             */
                        /*              has been detected                              */

int progchk (statdata)

byte statdata;          /* Status Register data that has been already read            */
                        /*              from the 28F001BX in function end()            */

{
 if ((statdata & VLOWMASK) == VPPLOW)
      return (1);       /* VPP low detect error, return code "1"                      */
 if ((statdata & PERRMSK) == PROGERR)
      return (2);       /* Byte program error detect, return code "2"                 */
 return (0);            /* Byte/string program success, return code "0"               */
}
```

3

```
/*****************************************************************************************/
/*  Function: Idread                                                                     */
/*    Description: Reads the manufacturer and device IDs from the target 28F001BX        */
/*    Inputs: None                                                                       */
/*    Outputs: mfgrid: Returned manufacturer ID                                          */
/*             deviceid: Returned device ID                                              */
/*    Returns: 0 = ID read correct                                                       */
/*             1 = Wrong or no ID                                                        */
/*    Device Read Mode on Return: inteligent Identifier                                  */
/*****************************************************************************************/

#define MFGRADDR      0         /* Address "0" for the target 28F001BX...              */
                                /*          alterable depending on the system          */
                                /*          architecture                               */
#define DEVICADD      1         /* Address "1" for the target 28F001BX...              */
                                /*          alterable depending on the system          */
                                /*          architecture                               */
#define IDRDCOMM      0X90      /* inteligent Identifier command                       */
#define INTELID       0X89      /* Manufacturer ID for Intel devices                   */
#define DVCIDBT       0X94      /* Device ID for 28F001BX-T; change to 95H if          */
                                /*          using 28F001BX-B!!!                        */

int idread(mfgrid,deviceid)

byte *mfgrid;                   /* The manufacturer ID read by this function, to      */
                                /*          be transferred back to the calling         */
                                /*          program                                    */
byte *deviceid;                 /* The device ID read by this function, to be         */
                                /*          transferred back to the calling function */

{
 byte far *tempaddr;            /* Pointer address variable used to read IDs          */

 tempaddr = (byte far*)MFGRADDR;
 *tempaddr= IDRDCOMM;           /* Write inteligent Identifier command to an          */
                                /*          address within the 28F001BX memory map     */
                                /*          (in this case, 00H)                        */
 *mfgrid  = *tempaddr;          /* Read mfgr ID, tempaddr still points at address "0" */
 tempaddr = (byte far*)DEVICADD;/* Point to address "1" for the device specific ID  */
 *deviceid= *tempaddr;          /* Read device ID                                      */
 if ((*mfgrid != INTELID)||(*deviceid != DVCIDBT))
     return (1);                /* ID read error; device powered up?                  */
 return (0);
}
```

```
/*****************************************************************************/
/*  Function: Statrd                                                         */
/*     Description: Returns contents of the target 28F001BX Status Register  */
/*     Inputs: None                                                          */
/*     Outputs: statdata: Returned Status Register data                      */
/*     Returns: Nothing                                                      */
/*     Device Read Mode on Return: Status Register                           */
/*****************************************************************************/

#define STATREAD    0X70     /* Read Status Register command                 */
#define SYSADDR     0        /* This constant can be initialized             */
                             /*        to any address within the             */
                             /*        memory map of the target 28F001BX     */
                             /*        and is alterable depending on         */
                             /*        the system architecture               */
int statrd(statdata)

byte *statdata;              /* Allows Status Register data to               */
                             /*        be passed back to the calling program */
                             /*        for further analysis                  */

{
 byte far *stataddr;         /* Pointer variable used to write               */
                             /*        commands to device                    */
 stataddr  = (byte far*)SYSADDR;
 *stataddr = STATREAD;       /* Write Read Status Register                   */
                             /*        command to 28F001BX                   */
 *statdata = *stataddr;
 return;
}
```

3

```
/**********************************************************************************/
/*  Function: Statclr                                                            */
/*    Description: Clears the 28F001BX Status Register                           */
/*    Inputs: None                                                               */
/*    Outputs: None                                                              */
/*    Returns: Nothing                                                           */
/*    Device Read Mode on Return: Status Register                                */
/**********************************************************************************/

#define STATCLER     0X50     /* Clear Status Register command                   */
#define SYSADDR      0        /* This constant can be initialized to any         */
                              /*         address within the memory map of the target*/
                              /*         28F001BX and is alterable depending on  */
                              /*         the system architecture                 */

int statclr()

{
 byte far *stataddr;          /* Pointer variable used to write commands to      */
                              /*         device                                  */

 stataddr  = (byte far*)SYSADDR;
 *stataddr = STATCLER;        /* Write Clear Status Register command to          */
                              /*         28F001BX                                */
 return;
}


/**********************************************************************************/
/*  Function: Rdmode                                                             */
/*    Description: Puts the target 28F001BX in Read Array Mode. This function    */
/*      might be used, for example, to prepare the system for return to code     */
/*      execution out of the Flash memory after program or erase algorithms      */
/*      have been executed off-chip                                              */
/*    Inputs: None                                                               */
/*    Outputs: None                                                              */
/*    Returns: Nothing                                                           */
/*    Device Read Mode on Return: Array                                          */
/**********************************************************************************/

#define RDARRAY      0XFF     /* Read Array command                              */
#define SYSADDR      0        /* This constant can be initialized to any         */
                              /*       address within the memory map of the target */
                              /*       28F001BX and is alterable depending on    */
                              /*       the system architecture                   */

int rdmode()

{
 byte far *tempaddr;          /* Pointer variable used to write commands to      */
                              /*         device                                  */
 tempaddr  = (byte far*)SYSADDR;
 *tempaddr = RDARRAY;         /* Write Read Array command to 28F001BX            */
 return;
}
```

```
/************************************************************************************/
/*  Function: Rdbyte                                                                */
/*    Description: Reads a byte of data from a specified address and
/*                 returns it to the calling program                                */
/*    Inputs: raddr: Target addres to be read from                                  */
/*    Outputs: rdata: Data at the specified address                                 */
/*    Returns: Nothing                                                              */
/*    Device Read Mode on Return: Array                                             */
/************************************************************************************/

#define RDARRAY     OXFF    /* Read array command                                   */

int rdbyte (rdata,raddr)

byte *rdata;                /* Returns data read from the device at                 */
                            /*         specified address                            */
byte far *raddr;            /* Raddr is the target address to be read from          */

{
 *raddr = RDARRAY;          /* Write read array command to an address within        */
                            /*         the 28F001BX memory map (in this case the    */
                            /*         target address)                              */
 *rdata = *raddr;           /* Read from the specified address and store            */
 return;
}
```

3

```
/*************************************************************************************/
/*  Function: Vppup                                                                  */
/*    Description: Ramps the Vpp supply to the target 28F001BX to enable             */
/*     programming or erase. This routine can be tailored to the individual          */
/*     system architecture. For purposes of this example, I assumed that a           */
/*     system Control Register existed at system address 20000 hex.                  */
/*     with the following definitions:                                               */
/*                                                                                   */
/*      Bit 7: Vpph Control:      1 = Enabled                                        */
/*                                0 = Disabled                                       */
/*      Bit 6: PWD Control:       1 = PowerDown Enabled                              */
/*                                0 = PowerDown Disabled                             */
/*      Bits 5-0: Undefined                                                          */
/*    Inputs: None                                                                   */
/*    Outputs: None                                                                  */
/*    Returns: Nothing                                                               */
/*    Device Read Mode on Return: As existed before entering the function.           */
/*     Part is now ready for program or erase command sequence                       */
/*************************************************************************************/

#define VPPHIGH              0X80     /* Bit 7 = 1, Vpp elevated to Vpph         */
#define SYSCADDR             0X20000  /* Assumed system Control Register Address */

int vppup()

{
 byte far *contaddr;                   /* Pointer variable used to write data     */
                                       /*          to the System Control Register  */
 contaddr  = (byte far*)SYSCADDR;
 *contaddr = *contaddr | VPPHIGH;      /* Read current Control Register data,     */
                                       /*          "OR" with constant to ramp Vpp  */
 return;
}
```

```
/*************************************************************************************/
/*  Function: Vppdown                                                                */
/*    Description: Ramps down the Vpp supply to the target 28F001BX to              */
/*     disable programming/erase. See above for a description of the               */
/*     assumed system Control Register.                                             */
/*    Inputs: None                                                                  */
/*    Outputs: None                                                                 */
/*    Returns: Nothing                                                              */
/*    Device Read Mode on Return: As existed before entering the function. Part    */
/*     now has high Vpp disabled. If program or erase was in progress when         */
/*     this function was called, it will complete unsuccessfully with Vpp low error */
/*     in the Status Register.                                                      */
/*************************************************************************************/

#define VPPDWN      0X7F    /* Bit 7 = 0, Vpp lowered to Vppl                          */

#define SYSCADDR    0X20000 /* Assumed system Control Register Address                 */

int vppdown()

{
 byte far *contaddr;        /* Pointer variable used to write data to the             */
                            /*         system Control Register                        */
 contaddr  = (byte far*)SYSCADDR;

 *contaddr = *contaddr & VPPDWN;
                            /* Read current Control Register data, "AND" with         */
                            /*         constant to lower Vpp                          */
 return;
}
```

3

```
/***********************************************************************************/
/*  Function: Pwdon                                                                */
/*    Description: Toggles the 28F001BX /PWD pin low to put the device in Deep     */
/*     PowerDown mode. See above for a description of the assumed                  */
/*     system Control Register.                                                    */
/*    Inputs: None                                                                 */
/*    Outputs: None                                                                */
/*    Returns: Nothing                                                             */
/*    Device Read Mode on Return: The part is powered down. If program or erase    */
/*     was in progress when this function was called, it will abort with           */
/*     resulting partially programmed or erased data. Recovery in the form of      */
/*     repeat of program or erase will be required once the part                   */
/*     transitions out of powerdown, to initialize data to a known state.          */
/***********************************************************************************/

#define PWD        0X40      /* Bit 6 = 1, /PWD enable                             */
#define SYSCADDR   0X20000   /* Assumed system Control Register Address            */

int pwdon()

{
 byte far *contaddr;          /* Pointer variable used to write data to the        */
                              /*           system Control Register                  */
 contaddr  = (byte far*)SYSCADDR;
 *contaddr = *contaddr | PWD; /* Read current Control Register data, "OR" with      */
                              /*           constant to enable Deep PowerDown        */
 return;
}
```

```
/*****************************************************************************************/
/*  Function: Pwdoff                                                                     */
/*    Description: Toggles the 28F001BX /PWD pin high to transition the part            */
/*     out of Deep PowerDown. See above for a description of the assumed system         */
/*     Control Register.                                                                 */
/*    Inputs: None                                                                       */
/*    Outputs: None                                                                      */
/*    Returns: Nothing                                                                   */
/*    Device Read Mode on Return: Read Array mode. Low voltage is removed                */
/*     from /PWD. 28F001BX output pins will output valid data time tPHQV                 */
/*     after the /PWD pin transitions high (reference the datasheet AC                   */
/*     Read Characteristics) assuming valid states on all other control                  */
/*     and power supply pins.                                                            */
/*****************************************************************************************/

#define PWDOFF        0XBF      /* Bit 6 = 0, /PWD disabled                              */
#define SYSCADDR      0X20000   /* Assumed system Control Register Address               */

int pwdoff()

{
 byte far *contaddr;            /* Pointer variable used to write data to the            */
                                /*        system Control Register                        */
 contaddr  = (byte far*)SYSCADDR;
 *contaddr = *contaddr & PWDOFF; /* Read current Control Register data, "AND" with       */
                                /*        constant to disable Deep PowerDown             */
 return;
}
```

3

APPENDIX B
MS-DOS ROM VERSION OVERVIEW

Technical Highlights

(Taken from Microsoft Product Overview)

RAM Economy

Because MS-DOS ROM Version executes from ROM, only 15 KB of system RAM space is required for MS-DOS. For a typical user, this will result in a savings of about 40 KB of RAM over disk-based MS-DOS. As a result of this savings, the user is able to run more programs and work with larger data files with the ROM Version than with disk-based MS-DOS. Instant-On MS-DOS ROM Version provides a significant reduction in "boot time", or the amount of time it takes from the completion of the power-on self test until a DOS prompt appears. With the ROM Version, this typically takes one second.

No End-User Installation

MS-DOS ROM Version is pre-installed by the OEM (original equipment manufacturer) in the system, thus freeing end users from the task of installing MS-DOS.

Adaptable to OEM Hardware Platforms

MS-DOS ROM Version is structured such that it allows the OEM to include a specific routine to determine which drive to boot from and any specific parameters if booting from the ROM drive. This makes it possible to easily port the ROM Version to a wide variety of hardware environments. MS-DOS ROM Version is also positioned independent, in that it can reside anywhere in the "reserved" space (the area between 640 KB and 1 MB). This provides an additional Version to the specific requirements of the OEM's hardware platform.

ROM Economy

MS-DOS ROM Version occupies only 62 KB of ROM space, thus minimizing the amount of ROM that an OEM must include in the system. Three modules reside in the reserved space: COMMAND.COM, IO.SYS and the DOS Kernel. All three are position independent, so an OEM can decide where to place these modules in the reserved area.

National Language Support

Microsoft offers a full compliment of localized version of MS-DOS ROM Version, including Kanji and Chinese translations.

Ease of Development

As PCs become the engines for many embedded applications, manufacturers would like to develop new applications utilizing existing PC software tools. MS-DOS ROM allows manufacturers to take full advantage of these tools. For instance, a programmer can develop and debug an application onto a PC subsystem which may be embedded into a larger system. This benefit translates into a cost savings when developing a solution for vertical markets.

APPENDIX C
BIOS VENDOR INFORMATION

American Megatrends Inc. (AMI)
1346 Oakbrook Drive, Suite 120
Norcross, GA 30093
(404) 263-8181

Award Software Inc.
130 Knowles Drive
Los Gatos, CA 95030
(408) 370-7979

Phoenix Technologies, LTD.
40 Airport Parkway
San Jose, CA 95110
(408) 452-6500

Systemsoft Corporation
313 Speen Street
Natick, MA 01760
(508) 651-0088

This list is intended for example only, and in no way represents all companies that support BIOS software. Since this industry developes many new solutions each year, Intel recommends that the designer contact the vendors for their latest products. Intel will continue to work with BIOS vendors to develop optimum solutions. Intel Corporation assumes no responsibility for circuitry or software other than circuitry embodied in Intel products. No software patent licenses are implied.

APPENDIX D MICROPROCESSOR/MICROCONTROLLER COMPATIBILITY CHART

28F001BX-T	28F001BX-B
x86 Family	i960™ KA/KB Microprocessor
i860™ Family	i960™ SA/SB Microprocessor
i960™ CA Microprocessor	MCS®-51 Family
	MCS®-96 Family

REVISION HISTORY

Number	Description
-004	Added 10K resistor to FET output, Figure 13. Updated Erase Suspend Flowchart, Figure 16 Updated Erase Suspend "C" Code, page A-3

intel®

APPLICATION NOTE

AP-357

July 1992

3

Power Supply Solutions for Flash Memory

ANIL SAMA
BRIAN DIPERT
APPLICATIONS ENGINEERING
INTEL CORPORATION

Order Number: 292092-001

Power Supply Solutions for Flash Memory

CONTENTS PAGE

1.0 INTRODUCTION 3-399

2.0 FLASH MEMORY POWER REQUIREMENTS 3-399

V_{CC} Characteristics 3-399

V_{PP} Characteristics 3-399

2.1 Supplies for Battery Powered Applications 3-400

2.2 Choice of a DC-DC Converter 3-400

The Modular Solution 3-400

The Discrete Switching Regulator Solution 3-400

Attributes of a DC-DC Converter 3-401

3.0 12V V_{PP} SOLUTIONS: CONVERTING UP FROM 5V 3-401

3.1 Maxim MAX732 3-401

3.2 Linear Technology LT1110-12 3-403

3.3 Linear Technology LT1109-12 3-404

3.4 Motorola MC34063A 3-406

4.0 12V V_{PP} SOLUTIONS: CONVERTING UP FROM 3V 3-407

4.1 Linear Technology LT1110-12 3-408

4.2 Maxim MAX732 @ 30 mA 3-409

4.3 Maxim MAX732 @ 60 mA 3-411

5.0 5V V_{CC} SOLUTIONS: CONVERTING UP FROM 3V 3-412

5.1 Maxim MAX658 @ 250 mA 3-413

5.2 Linear Tech LT1110-5 @ 150 mA 3-414

6.0 12V V_{PP} SOLUTIONS: DOWN-CONVERTING FROM A HIGHER VOLTAGE 3-415

6.1 Maxim MAX667 3-416

6.2 Linear Technology LT1111-5 3-417

6.3 National Semiconductor LM2940CT-12 3-418

7.0 12V V_{PP} FROM 12V UNREGULATED 3-419

8.0 SUMMARY 3-419

APPENDIX A: Modular Solutions 3-420

APPENDIX B: Survey of Solutions Presented 3-421

APPENDIX C: Sources for DC-DC Converters 3-422

APPENDIX D: Sources for Discrete Components 3-424

APPENDIX E: Other Design Considerations 3-426

APPENDIX F: PC Layouts 3-429

1.0 INTRODUCTION

Intel flash memory is rapidly being incorporated into a wide range of applications, adding enhanced capability to existing "traditional" memory markets, and creating new markets that exploit its benefits. Sometimes the design platforms may not possess the low powered 12V supply for writing flash memory. The system design engineer then needs to identify a power conversion solution with features and capabilities matching the needs of the application. For example, portable equipment requires a power supply converter that minimizes size and weight, maximizes efficiency to extend battery life, and can be switched into a standby mode to conserve power.

The following pages present some state of the art DC-DC converter solutions. These new solutions are smaller and more efficient than those typically seen in the past. Each of these solutions optimizes a subset of all possible power converter features. The choice of an optimal solution for a given application will be a tradeoff between several attributes. The solutions shown should meet the conversion needs of the majority of applications involving flash memory. Specifically, the solutions that follow encompass the following five categories:

- 5V to 12V conversion
- 3V (2 alkaline/NiCd cells) to 12V conversion
- 3V (2 Alkaline/NiCd cells) to 5V conversion
- Downconverting to 12V from a higher voltage
- Converting 12V unregulated to 12V regulated

More than one solution is presented within each of these categories. These different solutions have distinct optimal features/advantages. The optimal attributes of each solution are outlined. In addition, the appendix contains a survey of all solutions presented here, and provides a basis for comparing their features. The reader should reference it to choose an optimal solution for his/her application.

NOTE:
Solutions were selected from products offered by over thirty DC-DC converter vendors. Since this industry develops many new solutions each year, Intel recommends that designers contact vendors for latest products. Intel will continue to work with the industry to develop optimum solutions for power conversion. Intel Corporation assumes no responsibility for circuitry other than circuitry embodied in Intel products. No other circuit patent licenses are implied.

2.0 INTEL FLASH MEMORY POWER REQUIREMENTS

Intel flash memory is powered by two sources; a 5V V_{CC} line and a 12V V_{PP} line. V_{CC} is the primary power source and the only power source needed to read the memory. V_{PP} is required when writing or erasing the memory.

V_{CC} Characteristics

V_{CC} supplies power to the flash device during all operational modes. Maximum V_{CC} current is demanded by the device during the read operation. The data sheets for all Intel flash memory devices at the time this application note was written specify a maximum read current (I_{CC}) of 30 mA at 5V ±10%. This is the guaranteed worst case DC V_{CC} current that may be required by a flash device for reading one byte of data. If multiple components are read simultaneously, the V_{CC} current requirement increases proportionately. V_{CC} tolerance must be maintained to within specification limits at all times for proper functioning of the device.

V_{PP} Characteristics

The supplemental V_{PP} source provides the higher voltages needed to carry out the erase, erase verify, program, and program verify operations. Maximum V_{PP} current is typically demanded during the program and erase modes. Data sheets for all Intel flash memory devices at the time this application note was written specify a maximum I_{PP} current of 30 mA at 12V ±5% for both program and erase operations. This is the guaranteed worst case V_{PP} supply current that will be required by a flash device for writing one byte of data or erasing one block/component. If multiple components are programmed/erased simultaneously, the current requirement increases proportionately. V_{PP} must be maintained to within specification limits at all times during device program, and erase. The tolerance specification on V_{PP} must be strictly maintained. Over-voltage can damage the device, and under-voltage can decrease specified device reliability. Although the 12V supply must meet these worst case specifications, power usage will typically be much lower. The lower typical values seen in the data sheets should be used in calculating typical battery life.

2.1 Supplies for Battery Powered Applications

In applications where batteries are the primary source of power, the power supplies providing V_{CC} and V_{PP} need to be selected very carefully. Optimized operating efficiency of these supplies is important to extend battery life. Another attractive feature is the capability of these supplies to be switched into a very low power shutdown mode. It is important to minimize this shutdown current consumption as well since flash memory V_{PP} generators will often be in this state for extended periods of time. Moreover, since these supplies are used in equipment that is physically small and space-constrained, size and height of the supply need to be minimized.

Where two alkaline/NiCd batteries are used as the primary source of power, the primary voltage varies depending on the type and the state of discharge of the batteries. For example, alkaline batteries start life off at 1.5V, but may still retain a significant amount of energy when the voltage falls to 1.0V with use, and will work all the way down to 0.8V. On the other hand, NiCd cells maintain a near constant voltage of 1.25V throughout most of their discharge cycle, and work down to 1.0V. A solution that derives V_{CC} or V_{PP} from 2 AA batteries must hence be capable of doing so from an input voltage that lies in the range of 1.6V to 3.0V.

It is best to directly convert the primary battery voltage into the various voltages needed throughout the system. A step conversion (e.g. a 3V to 5V converter for V_{CC}, followed by a 5V to 12V converter for V_{PP}) is not recommended, since the inefficiency involved in each conversion step combines into one large inefficiency for the sum 3V to 12V conversion. Section 4 presents appropriate 3V battery to 12V converter solutions. Most of the solutions presented in this application note, while specifically designed for battery powered applications, are also viewed as ideal for other applications that incorporate flash memory.

2.2 Choice of a DC-DC Converter

The solution to finding the right power supply for flash memory lies in picking the right DC-DC converter for the job. Two broad categories of DC-DC converters available in the market today can be applied towards this purpose. These are the low power hybrid DC-DC converter module (or modular solution), and the low power discrete switching regulator IC solution.

The Modular Solution

The modular solution generally consists of a push-pull (Royer type) oscillator built around an isolation transformer, and in some cases followed by a linear regulator; all of which is encapsulated within a module. This hybrid module includes all components that are required by the DC-DC converter, and so no additional design effort is needed. The input and output voltages are fixed, and the input and output are almost always isolated via the isolation transformer. The main advantage of these solutions is that no design effort and/or external components are involved. They simply plug into a socket on a PC board. Disadvantages include lower efficiency (generally 60%), larger size/height (in most cases), and higher cost (generally 3x to 10x the cost of discrete solutions).

It would seem that the integration inherent in these solutions contributes towards system reliability, however the type and quality of the discrete components used internal to these hybrid devices is open to question. The isolation offered between the input and output is viewed as overkill for flash applications, since the total power required is typically less than 1W. Note also that the isolation transformer is often the main reason for the lower efficiencies.

The Discrete Switching Regulator Solution

The discrete switching regulator IC solution consists of a DC-DC converter IC (containing a switching regulator controller and an output power switch), along with a few discrete external components (inductor, diode, capacitors, resistors, etc.). The layout of the power supply system in this case is mostly left up to the user. However, application notes and data sheets explain the design process, and provide recommended circuits for commonly used solutions. The design can be tailored to deliver different output voltages and current levels depending on the characteristics of the input voltage and the external components.

Some vendors offer fixed output voltage versions, further simplifying the design process. The newer generation of high frequency low power switching regulator ICs are specifically targeted at battery powered operation, and most can be switched into a low quiescent current shutdown mode to extend battery life. These have typical efficiencies in the 75% to 90% range. Furthermore, the higher switching frequencies of these new parts (typically 100 KHz to 200 KHz) allow the use of smaller external components, which are available in surface mount varieties. As a consequence, these newer solutions are overall much smaller than what was typically seen just a year ago.

Attributes of a DC-DC Converter

Several attributes of a power supply converter must be evaluated and prioritized when choosing the best solution for a given application. These attributes include:

- Input Voltage Range
- Output Voltage and Tolerance
- Output Current Capability
- Efficiency of Conversion
- Printed Circuit Area
- Height
- Total Cost
- Shutdown Capability
- Quiescent Current Consumed in Shutdown Mode
- Rise Time from Shutdown
- Surface Mountability

The reader is referred to Appendix B, which provides a survey of all the solutions that are presented in this application note, in order to compare their attributes.

This application note primarily presents state of the art discrete switching regulator IC solutions which have been carefully designed for operation with flash memory. Included along with schematics are component values and sources/contacts for obtaining all the components. Actual layouts have also been included where possible. These are provided in Appendix F.

NOTE:
External components recommended in the designs should be used. These components (inductors, capacitors, resistors) were chosen based on recommendations by the converter IC vendors and provide the necessary quality for a clean design. Alternate "equivalent" parts should be chosen with care as their resistive and inductive elements can affect the operation of the solution. Please contact the respective converter IC companies for assistance if you select an alternate value/source for passive components.

3.0 V_{PP} SOLUTIONS: CONVERTING UP FROM 5V

Most computer systems have available a 5V V_{CC} line that is used for the majority of system power. Frequently, this 5V supply is used to generate 12V for flash memory. This section presents some of the new state of the art solutions that can achieve this function. These are all discrete switching regulators that optimize different attributes, mentioned along with the main features section of each example. Refer to Appendix B for a more detailed comparison of the attributes of these solutions.

3

3.1 Maxim Integrated Products—MAX732: V_{PP} @ 30 mA, 60 mA, 120 mA

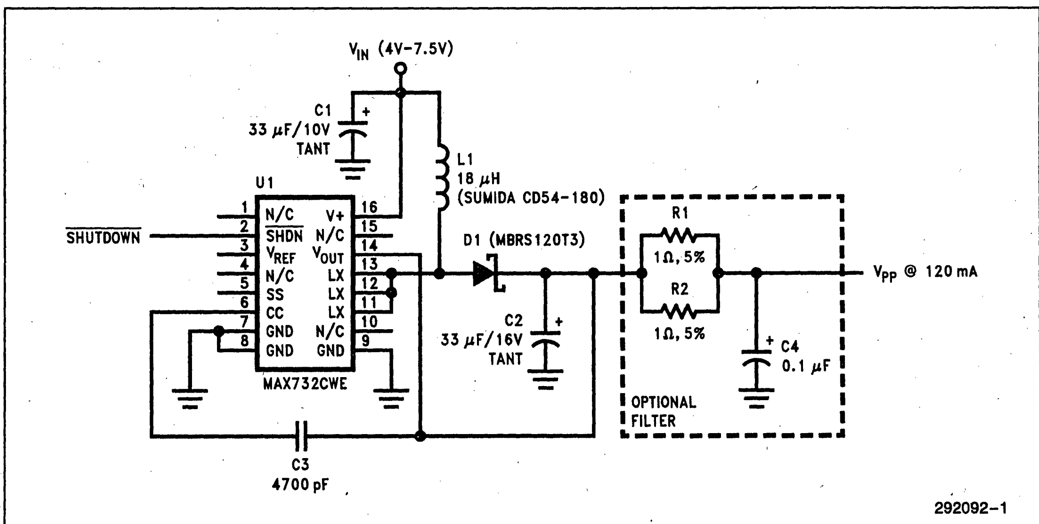


Figure 3-1. Maxim MAX732 5V to 12V Converter

Optimal Attributes

- Highest Efficiency
- Low Shutdown Current
- Wide Input Voltage Range
- All Surface Mount

Main Features

- Input Voltage Range: 4V to 7.5V
- Output Voltage: 12V ±4%
- Output Current Capability: Up to 120 mA
- Typical Efficiency: 90% at I_{LOAD} = 60 mA
- 170 KHz Operation
- Shutdown Feature On Chip
- Low Quiescent Current at Shutdown: 70 μA typical
- Low Operating Quiescent Current: 1.6 mA typical
- Rise Time from Shutdown: 1 ms Typical
- Will Work off Existing 5V Supply or a 6 NiCd Battery Pack

The MAX732 design as shown is capable of providing up to 120 mA of V_{PP} current at an efficiency of 90%. The 5V input should be able to source the peak currents and start-up currents required by the circuit. This converter circuit can also run directly off a 6 cell NiCd pack present on many notebook/laptop computers. It is available in a 16-pin wide SOIC package, and uses small external surface mount components (5 in all). Voltage spikes may be present in the output due to incorrect layout, excessive output filter capacitor ESL (Equivalent Series Inductance) and diode switching transients. The optional filter circuit is recommended to eliminate any sharp transients. The supply can be switched into a shutdown mode where the output voltage falls to approximately V_{CC} − 550 mV. A layout is presented in Appendix F. Applications assistance and a surface mount evaluation kit is also available from Maxim.

Table 3-1. Parts List for the MAX732 5V to 12V Converter

Ref	Part #	Value/Type	Source	Cost*
U1	MAX732CWE	SMPS IC	Maxim (408) 737-7600	$2.50
C1	267M1002-336-MR-720	33 μF/10V Tantalum	Matsuo (714) 969-2491	$0.31
C2	267M1602-336-MR-720	33 μF/16V Tantalum	Matsuo (714) 969-2491	$0.31
C3	C4532C0G1H472K	4700 pF	TDK (708) 803-6100	$0.04
C4	GRM40Z5U104M050AD	0.1 μF	Murata Erie (404) 436-1300	$0.06
R1, R2	9C08052A1R00JLR	1.0Ω, 5%	Philips (817) 325-7871	$0.04
D1	MBRS120T3	Schottky Diode	Motorola (800) 521-6274	$0.30
L1	CD54-180	18 μH	Sumida (708) 956-0666	$0.38
			Total Cost	$3.93

*Cost estimates based on published 10K unit pricing at the time this application note was written.

3.2 Linear Technology LT1110-12: V_{PP} @ 30 mA, 60 mA, 120 mA

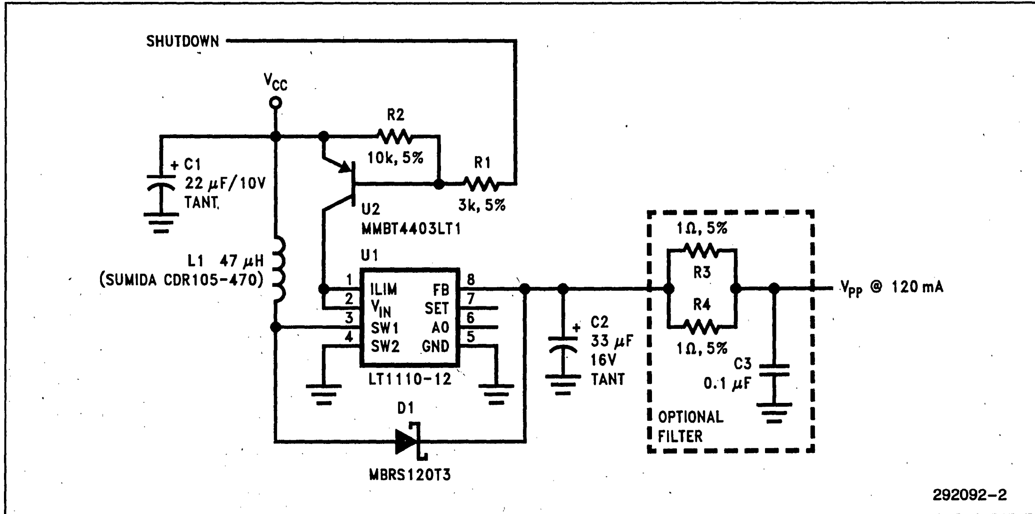


Figure 3-2. Linear Technology LT1110-12 5V to 12V Converter

Optimal Attributes

- Small Size: 0.45 sq. in. Total Board Area (Single Sided)
- Very Low Shutdown Current: 16 µA
- All Surface Mount

Main Features

- Input Voltage Range: 4.5V to 5.5V
- Output Voltage: 12V ±5%
- Output Current Capability: Up to 120 mA
- Typical Efficiency: 76%
- 60 KHz Operation
- Shutdown Possible Using External Components as Shown
- Low Quiescent Current at Shutdown: 16 µA typical
- Rise Time from shutdown: 800 µs typical

The Linear Technology LT1110-12 is a fixed 12V output part which is well suited to flash memory applications. The part is available in a small 8-pin surface mount (SO8) package. The part needs 7 external components to implement a small size 5V to 12V converter solution that can be shutdown to a very low quiescent current state—16 µA typical. The 5V source must be capable of supplying the instantaneous start-up and peak currents required during operation. Voltage spikes may be present in the output due to incorrect layout, excessive output filter capacitor ESL (Equivalent Series Inductance) and diode switching transients. The optional RC filter circuit is recommended in order to eliminate these sharp transients. The output voltage during shutdown falls to approximately V_{IN} − 550 mV. A recommended board layout appears in Appendix F. Applications assistance is available from Linear Technology Corporation.

Table 3-2. Part List for the LT1110-12 5V to 12V Converter

Ref	Part #	Value/Type	Source	Cost*
U1	LT1110-12	SMPS IC	Linear Tech (408) 954-8400	$2.60
C1	267M1002-226-MR-720	22 μF/10 V Tantalum	Matsuo (714) 969-2491	$0.23
C2	267M1602-336-MR-720	33 μF/16 V Tantalum	Matsuo (714) 969-2491	$0.31
C3	GRM40Z5U104M050AD	0.1 μF	Murata Erie (404) 436-1300	$0.06
D1	MBRS120T3	Schottky Diode	Motorola (800) 521-6274	$0.30
L1	CDR105-470	47 μH Shielded	Sumida (708) 956-0666	$0.40
R1	9C08052A3001JLR	3 KΩ, 5%	Philips (817) 325-7871	$0.02
R2	9C08052A1002JLR	10 KΩ, 5%	Philips (817) 325-7871	$0.02
R3, R4	9C08052A1R00JLR	1Ω, 5%	Philips (817) 325-7871	$0.04
U2	MMBT4403LT1	2N4403 PNP Transistor	Motorola (800) 521-6274	$0.09
			Total Cost	$4.07

*Cost estimates based on published 10K unit pricing at the time this application note was written.

3.3 Linear Technology LT1109-12: V_{PP} @ 30 mA, 60 mA

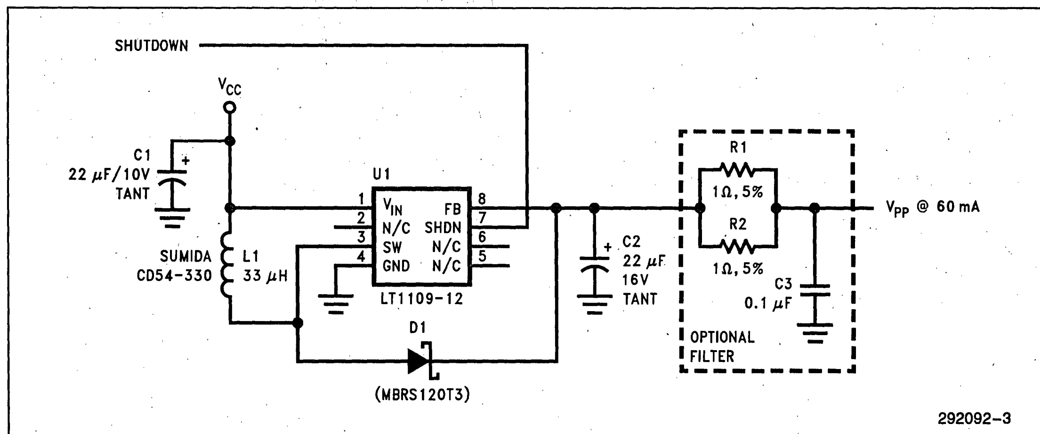


Figure 3-3. Linear Technology LT1109-12 5V to 12V Converter

Optimal Attributes

- Smallest Size
- Low Shutdown Current
- All Surface Mount

Main Features

- Input Voltage Range: 4.5V to 5.5V
- Output Voltage: 12V ±5%
- Output Current Capability: Up to 60 mA
- Typical Efficiency: 84%
- 130 KHz Operation
- Shutdown Feature On Chip
- Low Quiescent Current at Shutdown: 375 μA typical
- Rise Time from shutdown: 800 μs typical
- Small Size: SO8 plus 4 small external components

The Linear Technology LT1109-12 is a fixed 12V output part which is very well suited to flash memory applications. The part is available in a very small 8-pin surface mount (SO8) package. The part needs just 4 small external components to implement an extremely small size 5V to 12V converter solution that can be shutdown to a low quiescent current state—375 μA typical. The 5V source must be capable of supplying the instantaneous start-up and peak currents required by the operation. Voltage spikes may be present in the output due to incorrect layout, excessive output filter capacitor ESL (Equivalent Series Inductance) and diode switching transients. The optional RC filter circuit is recommended in order to eliminate these sharp transients. The output during shutdown falls to approximately V_{IN} − 550 mV. A typical board layout is presented in Appendix F. Applications assistance is available from Linear Technology Corporation.

Table 3-3. Parts List for the LT1109-12 5V to 12V Converter

Ref	Part #	Value/Type	Source	Cost*
U1	LT1109-12	SMPS IC	Linear Tech (408) 432-1900	$2.37
C1	267M1002-226-MR-720	22 μF/10V Tant Chip Capacitor	Matsuo (714) 969-2491	$0.23
C2	267M2502-106-MR-720	10 μF/25V Tant Chip Capacitor	Matsuo (714) 969-2491	$0.29
C3	GRM40Z5U104M050AD	0.1 μF	Murata Erie (404) 436-1300	$0.06
D1	MBRS120T3	Schottky Diode	Motorola (800) 521-6274	$0.30
R1, R2	9C08052A1R00JLR	1Ω, 5%	Philips (817) 325-7871	$0.04
L1	CD54-330	33 μH	Sumida (708) 956-0666	$0.32
			Total Cost	$3.61

*Cost estimates based on published 10K unit pricing at the time this application note was written.

3.4 Motorola MC34063A: V_{PP} @ 30 mA, 60 mA, 120 mA

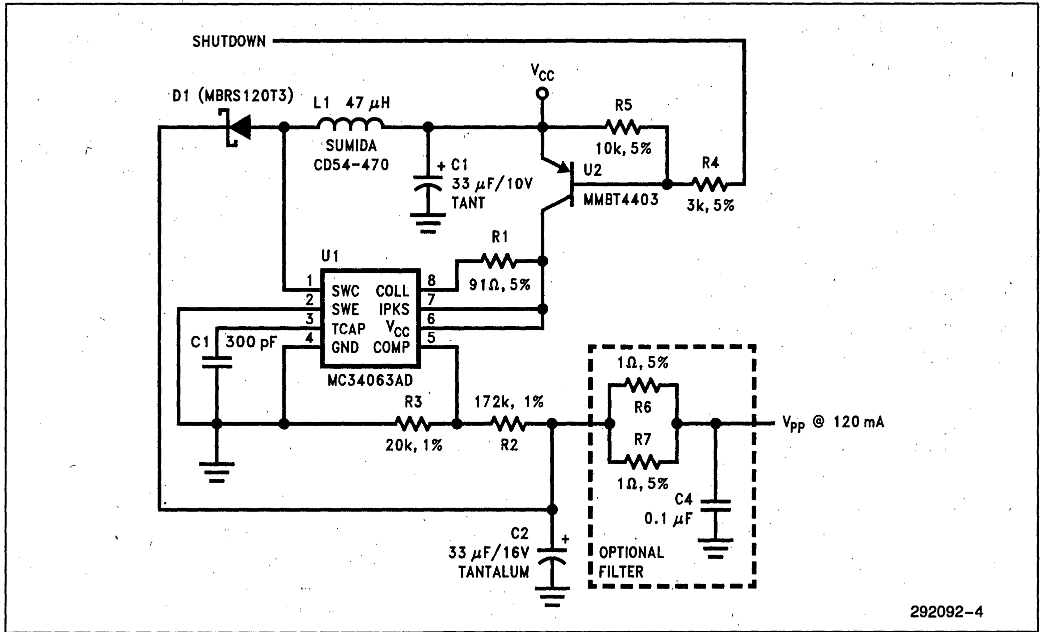


Figure 3-4. Motorola MC34063A 5V to 12V Converter

Optimal Attributes

- Lowest Cost
- Very Low Shutdown Current
- All Surface Mount

Main Features

- Input Voltage Range: 4.5V to 5.5V
- Output Voltage: 12V ±5%
- Output Current Capability: Up to 120 mA
- Typical Efficiency: 80%
- 100 KHz Operation
- Shutdown Feature Using External Components
- Low Quiescent Current at Shutdown: 25 μA typical
- Rise Time From Shutdown: 2 ms typical
- SO8 Plus 11 Small External Components—All SMD

The Motorola MC34063A solution presented uses 11 small sized external components to implement a low cost surface mount 5V to 12V converter solution. Three external components (U2, R4, R5) are used to shut down supply to the part when V_{PP} is not needed. These could be eliminated to further lower the cost if power consumption is not important. The quiescent current in shutdown state is a low 25 μA. The output voltage in shutdown is V_{CC} − 550 mV. Voltage spikes may be present in the output due to incorrect layout, excessive output filter capacitor ESL (Equivalent Series Inductance) and diode switching transients. The optional RC filter circuit is recommended in order to eliminate these sharp transients. Applications assistance is available from Motorola.

Table 3-4. Parts List for the MC34063A 5V to 12V Converter

Ref	Part #	Value/Type	Source	Cost*
U1	MC34063AD	SMPS IC (SO8)	Motorola (800) 521-6274	$0.63
R1	9C08052A9100JLR	91Ω, 5%	(Philips (817) 325-7871	$0.02
R2	9B08053A1723FCB	172 KΩ, 1%	(Philips (817) 325-7871	$0.04
R3	9B08053A2002FCB	20 KΩ, 1%	Philips (817) 325-7871	$0.04
R4	9C08052A3001JLR	3 KΩ, 5%	Philips (817) 325-7871	$0.02
R5	9C08052A1002JLR	10 KΩ, 5%	Philips (817) 325-7871	$0.02
R6, R7	9C08052A1R00JLR	1Ω, 5%	Philips (817) 325-7871	$0.04
C1	267M1002-336-MR-720	33 μF/16V Tantalum	Matsuo (714) 969-2491	$0.28
C2	267M1602-336-MR-720	33 μF/16V Tantalum	Matsuo (714) 969-2491	$0.31
C3	GRM40X7R301M050AD	300 pF	Murata Erie (404) 436-1300	$0.03
C4	GRM40Z5U104M050AD	0.1 μF	Murata Erie (404) 436-1300	$0.06
D1	MBRS120T3	Schottky Diode	Motorola (800) 521-6274	$0.30
L1	CD54-470	47 μH	Sumida (708) 956-0666	$0.37
U2	MMBT4403LT1	PNP Transistor	Motorola (800) 521-6274	$0.09
			Total Cost	$2.25

* Cost estimates based on published 10K unit pricing at the time this application note was written.

4.0 V_{PP} SOLUTIONS: CONVERTING UP FROM 2 NiCd/ALKALINE CELLS

Palmtop computers that use 2 alkaline/NiCd batteries require that the system work even when the battery voltage is down near 1.8V. Currently there exist two good solutions that achieve a 12V output with inputs as low as 1.8V, and yet supply at least 30 mA of current. These are the LT1110-12 from Linear Technology Corporation, and the MAX732 from Maxim Integrated Products.

4.1 Linear Technology LT1110-12: V_{PP} @ 30 mA from 2 AA Cells

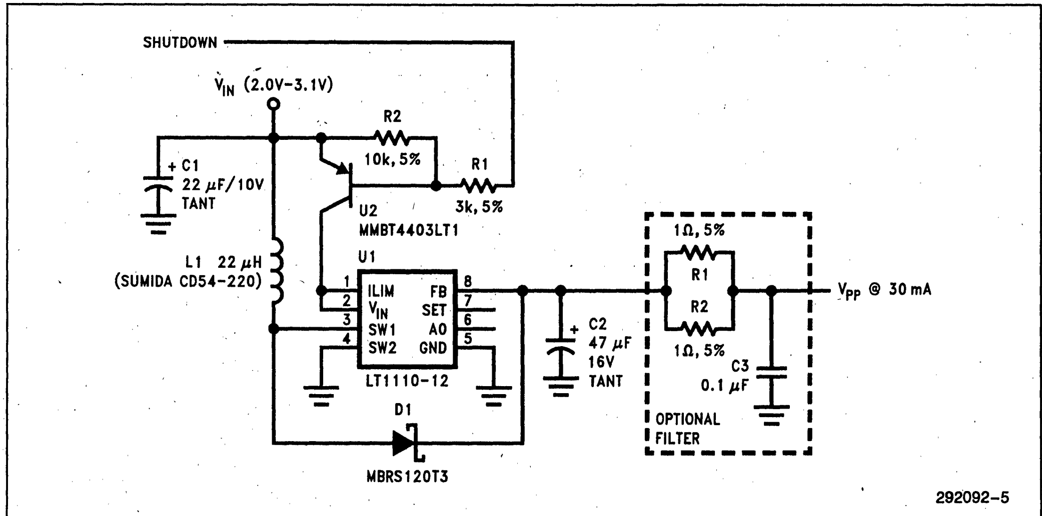


Figure 4-1. Linear Technology LT1110-12 3V to 12V Converter

Optimal Attributes

- Smallest Size
- Low Shutdown Current
- All Surface Mount

Main Features

- Input Voltage Range: 2.0V to 3.1V
- Output Voltage: 12V ±5%
- Output Current Capability: Up to 30 mA
- Typical Efficiency: 70%
- 60 KHz Operation
- Shutdown Mode Using External Components
- Low Quiescent Current at Shutdown: 16 µA typical
- Rise Time from Shutdown: 4 ms typical

The LT1110-12 from Linear Technology Corporation, as shown, can be used to generate V_{PP} from an input voltage between 2.0V and 3.1V (most of the usable life of 2 alkaline/NiCd cells in series). This design is similar to the 5V to 12V converter design presented in Section 3.2. Replacing L1 and C2 with a lower inductance and a higher capacitance, respectively, allows the part to work down to 2.0V, while reducing the output current capability to 30 mA. The external PNP transistor is used to shut off the input supply to the converter IC, and puts the part in shutdown state. Note that a disadvantage of this scheme of shutdown is that the control signal source sinks approximately 5 mA (V_{CC}/1K) when the part is not in shutdown. However, the quiescent current in shutdown state is a low 16 µA. See Appendix E for an alternate shutdown solution. The output voltage in shutdown falls to approximately V_{IN} − 550 mV. Voltage spikes may be present in the output due to incorrect layout, excessive output filter capacitor ESL (Equivalent Series Inductance) and diode switching transients. The optional RC filter circuit is recommended in order to eliminate any sharp transients. A surface mount layout appears in Appendix F.

Table 4-1. Parts List for the LT1110-12 3V to 12V Converter

Ref	Part #	Value/Type	Source	Cost*
U1	LT1110-12	SMPS IC	Linear Tech (408) 954-8400	$2.60
C1	267M1002-220-MR-720	22 μF/10V Tantalum	Matsuo (714) 969-2491	$0.23
C2	267M1602-470-MR-720	47 μF/16V Tantalum	Matsuo (714) 969-2491	$0.47
C3	GRM40Z5U104M050AD	0.1 μF	Murata Erie (404) 436-1300	$0.06
D1	MBRS120T3	Schottky Diode	Motorola (800) 521-6274	$0.30
L1	CD54-220	22 μH	Sumida (708) 956-0666	$0.37
R1	9C08052A3001JLR	3 KΩ, 5%	Philips (817) 325-7871	$0.02
R2	9C08052A1002JLR	10 KΩ, 5%	Philips (817) 325-7871	$0.02
R3, R4	9C08052A1R00JLR	1Ω, 5%	Philips (817) 325-7871	$0.04
U2	MMBT4403LT1	2N4403 PNP Transistor	Motorola (800) 521-6274	$0.09
			Total Cost	$4.20

*Cost estimates based on published 10K unit pricing at the time this application note was written.

4.2 Maxim Integrated Products—MAX732: V_{PP} @ 30 mA

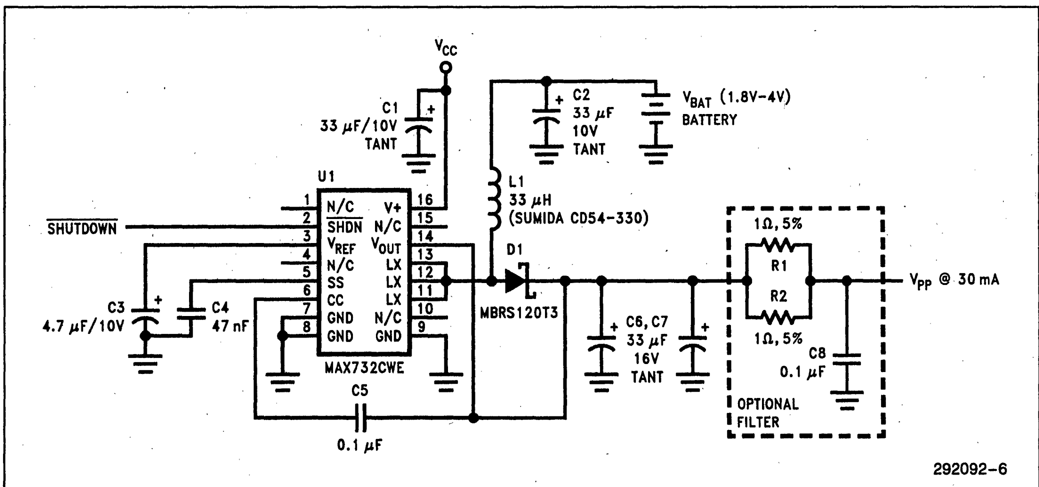


Figure 4-2. Maxim MAX732 3V to 12V Converter (30 mA)

Optimal Attributes

- Highest Efficiency
- All Surface Mount

Main Features

- Input Voltage Range: 1.8V to 5.0V
- Output Voltage: 12V ±4%
- Output Current Capability: Up to 30 mA
- Typical Efficiency: 87%
- 170 KHz Operation
- Shutdown Mode On Chip
- Low Quiescent Current at Shutdown: 45 μA typical
- Rise Time from shutdown: 25 ms typical

The MAX732 circuit as shown here can provide up to 30 mA at 12V from an input voltage as low as 1.8V. Note that the chip itself is powered from the 5V V_{CC} line required to use present day flash memory devices, whereas the inductor is connected to the primary battery supply. Voltage spikes may be present in the output due to incorrect layout, excessive output filter capacitor ESL and diode switching transients. The optional RC filter circuit is recommended in order to eliminate these sharp transients. Applications assistance and an evaluation kit is available from Maxim.

Table 4-2. Parts List for the MAX732 3V to 12V Converter (30 mA)

Ref	Part #	Value/Type	Source	Cost*
U1	MAX732CWE	SMPS IC	Maxim (408) 737-7600	$2.50
C1, C2	267M1002-336-MR-720	33 μF/10V Tantalum	Matsuo (714) 969-2491	$0.56
C3	267M1002-475-MR-720	4.7 μF/10V Tantalum	Matsuo (714) 969-2491	$0.20
C4	GRM40X7R473M050AD	47 nF	Murata Erie (404) 436-1300	$0.08
C5, C8	GRM40Z5U104M050AD	0.1 μF	Murata Erie (404) 436-1300	$0.12
C6, C7	267M1602-336-MR-720	33 μF/16V Tantalum	Matsuo (714) 969-2491	$0.62
R1, R2	9C08052A1R00JLR	1Ω, 5%	Philips (817) 325-7871	$0.04
D1	MBRS120T3	Schottky Diode	Motorola (800) 521-6274	$0.30
L1	CD54-330	33 μH	Sumida (708) 956-0666	$0.38
			Total Cost	$4.80

*Cost estimates based on published 10K unit pricing at the time this application note was written.

4.3 Maxim Integrated Products—MAX732: V_{PP} @ 60 mA

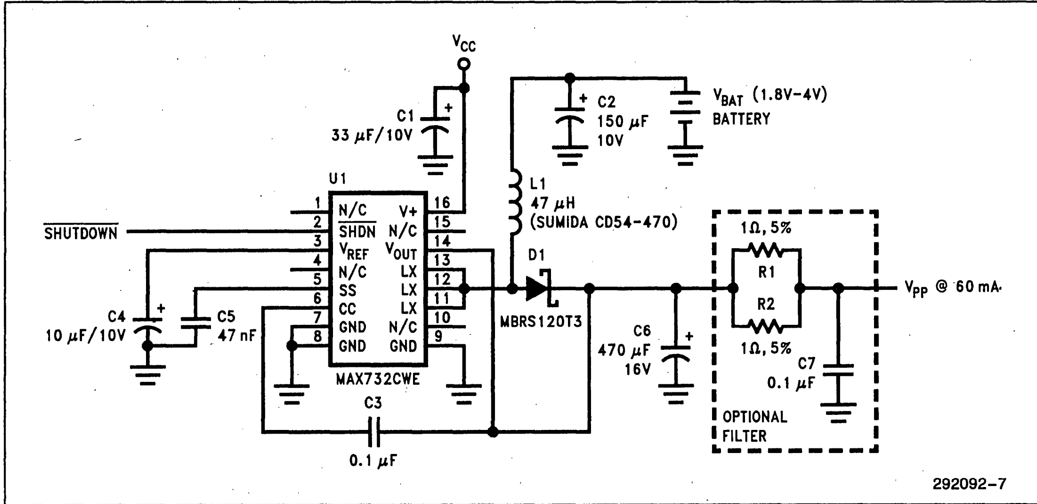


Figure 4-3. Maxim MAX732 3V to 12V Converter (60 mA)

Optimal Attributes

- Highest Efficiency
- 60 mA Output Current Capability

Main Features

- Input Voltage Range: 1.8V to 5.0V
- Output Voltage: 12V ±4%
- Output Current Capability: Up to 60 mA
- Typical Efficiency: 87%
- 170 KHz Operation
- Shutdown Mode On Chip
- Low Quiescent Current at Shutdown: 45 µA typical
- Rise Time from shutdown: 75 ms typical

The MAX732 circuit as shown here can provide up to 60 mA at 12V from an input voltage as low as 1.8V. This solution is similar to the previous one presented but is not entirely surface mountable, because of the larger output and input filter capacitors. Currently, it is the only solution employing a single IC that can provide 60 mA at 12V from a 1.8V input. The 470 µF/16V filter capacitor must be a low-ESR (Equivalent Series Resistance) type. Voltage spikes may be present in the output due to incorrect layout, excessive output filter capacitor ESL (Equivalent Series Inductance) and diode switching transients. The optional RC filter circuit is recommended in order to eliminate these sharp transients. Applications assistance and an evaluation kit is available from Maxim.

Table 4-3. Parts List for the MAX732 3V to 12V Converter (60 mA)

Ref	Part #	Value/Type	Source	Cost*
U1	MAX732CWE	SMPS IC	Maxim (408) 737-7600	$2.50
C1	267M1002-336-MR-720	33 μF/10V Tantalum	Matsuo (714) 969-2491	$0.31
C2	UPR1A151MPH	150 μF/10V	Nichicon (708) 843-7500	$0.10
C3, C7	GRM40Z5U104M050AD	0.1 μF	Murata Erie (404) 436-1300	$0.12
C4	267M1002-106-MR-720	10 μF/10V Tantalum	Matsuo (714) 969-2491	$0.21
C5	GRM40X7R473M050AD	47 nF	Murata Erie (404) 436-1300	$0.08
C6	UPR1C471MPH	470 μF/16V	Nichicon (708) 843-7500	$0.14
R1, R2	9C08052A1R00JLR	1Ω, 5%	Philips (817) 325-7871	$0.04
D1	MBRS120T3	Schottky Diode	Motorola (800) 521-6274	$0.30
L1	CD75-470	47 μH	Sumida (708) 956-0666	$0.38
			Total Cost	$4.15

*Cost estimates based on published 10K unit pricing at the time this application note was written.

5.0 V_{CC} SOLUTIONS: CONVERTING UP FROM TWO NiCd/ALKALINE CELLS

Palmtop and hand-held computers that use two AA size NiCd or alkaline batteries need a converter solution to provide the V_{CC} supply for the system as well as flash memory. Two good solutions are offered currently for this purpose, the MAX658 from Maxim Integrated Products and the LT1110-5 from Linear Technology Corporation.

5.1 Maxim Integrated Products—MAX658: V_{CC} @ 250 mA

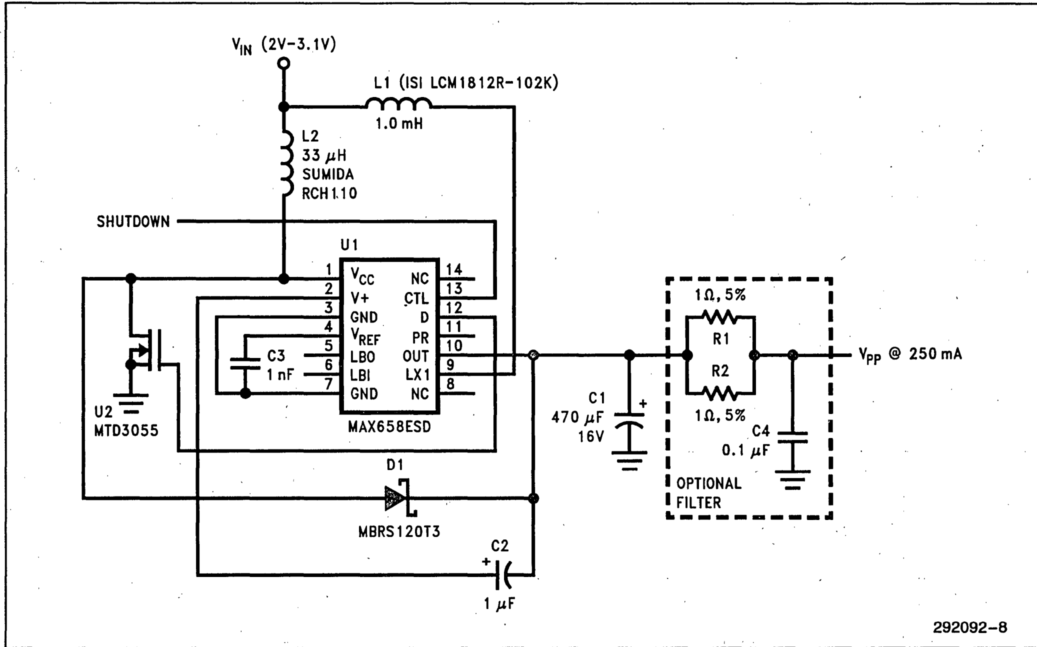


Figure 5-1. Maxim MAX658 3V to 5V Converter (250 mA)

Optimal Attributes

- Highest Efficiency
- 250 mA Output Current Capability
- Low Shutdown Current

Main Features

- Input Voltage Range: 2.0V to 3.1V
- Output Voltage: 5V ±10%
- Output Current Capability: Up to 250 mA
- Typical Efficiency: 85%
- 18 KHz Operation
- Shutdown Mode On Chip
- Low Quiescent Current at Shutdown: 80 µA typical
- Rise Time from shutdown: 25 ms typical

The MAX658, available from Maxim Integrated Products in a 14-pin surface mount package, is a good high current solution for obtaining V_{CC} from a pair of NiCd/alkaline cells. The entire solution, however, is not 100% surface mountable. It uses a high current through-hole inductor and a large through-hole filter capacitor at the output. Voltage spikes may be present in the output due to incorrect layout, excessive output filter capacitor ESR (Equivalent Series Resistance) and diode switching transients. The optional RC filter circuit is recommended in order to eliminate any sharp transients. Applications assistance and an evaluation kit are available from Maxim.

Table 5-1. Parts List for the MAX658 3V to 5V Converter

Ref	Part #	Value/Type	Source	Cost*
U1	MAX658	SMPS IC	Maxim (408) 737-7600	$2.45
C1	UPR1A471MPH	470 μF/10V Low Z	Nichicon (708) 843-7500	$0.12
C2	267M1602-105-MR-720	1 μF/16V Tantalum	Matsuo (714) 969-2491	$0.15
C3	GRM40X7R102M050AD	1 nF	Murata Erie (404) 436-1300	$0.03
C4	GRM40Z5U104M050AD	0.1 μF	Murata Erie (404) 436-1300	$0.06
R1, R2	9C08052A1R00JLR	1Ω, 5%	Philips (817) 325-7871	$0.04
D1	MBRS120T3	Schottky Diode	Motorola (800) 521-6274	$0.30
L1	LCM1812R-102K	1.0 mH Chip Inductor	Inductor Supply Inc. (800) 854-1881	$0.22
L2	RCH110-330	33 μH	Sumida (708) 956-0666	$0.40
U2	MTD3055E	NFET	Motorola (800) 521-6274	$0.70
			Total Cost	$4.47

*Cost estimates based on published 10K unit pricing at the time this application note was written.

5.2 Linear Technology LT1110-5: V_{CC} @ 150 mA

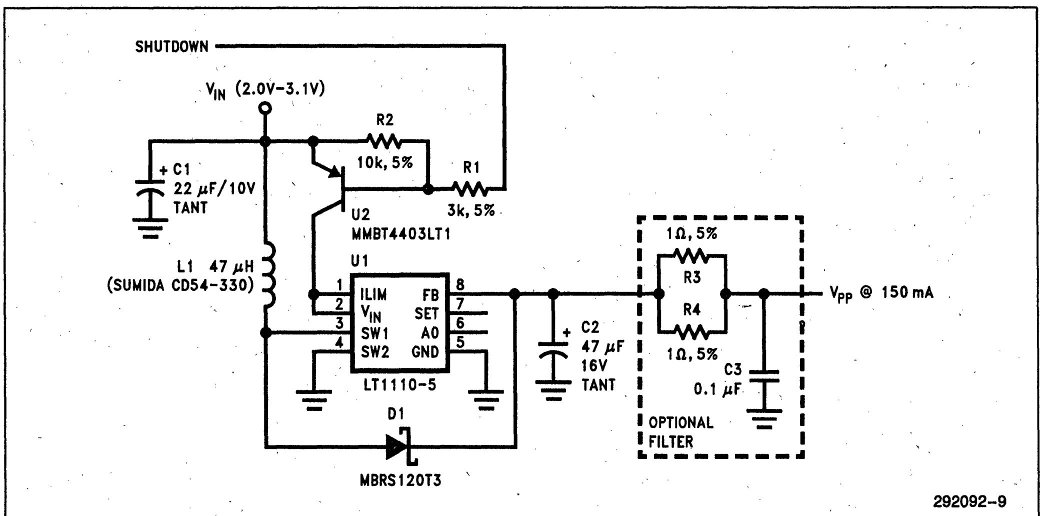


Figure 5-2. Linear Technology LT1110-5 3V to 5V Converter (150 mA)

Optimal Attributes

- Smallest Size
- Low Shutdown Current
- All Surface Mount

Main Features

- Input Voltage Range: 2.0V to 3.1V
- Output Voltage: 5V ±5%
- Output Current Capability: Up to 150 mA
- Typical Efficiency: 76%
- 60 KHz Operation
- Shutdown Mode Using External Components
- Low Quiescent Current at Shutdown: 16 μA typical
- Rise Time from Shutdown: 4 ms typical

The LT1110-5 from Linear Technology is a fixed 5V version of the converter shown for the 12V design in Section 4.1.

Table 5-2. Parts List for the LT1110-5 3V to 5V Converter

Ref	Part #	Value/Type	Source	Cost*
U1	LT1110-5CS8	SMPS IC	Linear Tech (408) 954-8400	$2.60
C1	267M1002-226-MR-720	22 μF/10V Tantalum Chip	Matsuo (714) 969-2491	$0.23
C2	267M1602-476-MR-720	47 μF/16V Tantalum Chip	Matsuo (714) 969-2491	$0.47
C3	GRM40Z5U104M050AD	0.1 μF	Murata Erie (404) 436-1300	$0.06
D1	MBRS120T3	Schottky Diode	Motorola (800) 521-6274	$0.30
L1	CD75-330	33 μH	Sumida (708) 956-0666	$0.38
R1	9C08052A3001JLR	3 KΩ, 5%	Philips (817) 325-7871	$0.02
R2	9C08052A1002JLR	10 KΩ, 5%	Philips (817) 325-7871	$0.02
R3, R4	9C08052A1R00JLR	1Ω, 5%	Philips (817) 325-7871	$0.04
U2	MMBT4403LT1	PNP Transistor	Motorola (800) 521-6274	$0.09
			Total Cost	$4.21

*Cost estimates based on published 10K unit pricing at the time this application note was written.

6.0 DOWN-CONVERTING TO 12V

The ability to down-convert to 12V from a higher voltage is often needed (as in the telecommunications environment). This section presents some good solutions for obtaining V_{PP} from a higher voltage.

6.1 Maxim Integrated Products MAX667

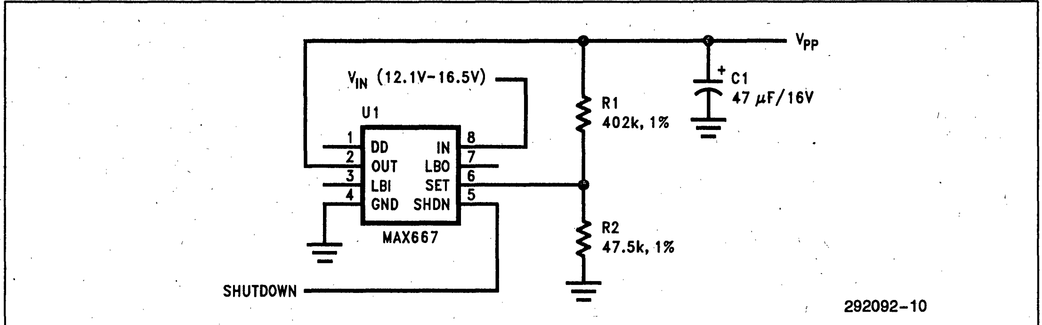


Figure 6-1. Maxim MAX667 12V Linear Voltage Regulator

Optimal Attributes

- Small Size
- Ultra Low Shutdown Current
- All Surface Mount
- Very Low Dropout
- Output Current Capability: Up to 120 mA
- Typical Efficiency: 70%
- Shutdown Mode On Chip
- Low Quiescent Current at Shutdown: 0.2 μA Typical
- Rise Time from Shutdown: Less than 0.1 ms Typical

Main Features

- Input Voltage Range: 12.1V to 16.5V
- Output Voltage: 12V ±5%

Table 6-1. Parts List for the MAX667 12V Step Down Converter

Ref	Part #	Value/Type	Source	Cost*
U1	MAX667CSA	SMPS IC-SO8 Package	Maxim (408) 737-7600	$2.10
C1	267M1602-476-MR-720	7 μF/16V Tantalum	Matsuo (714) 969-2491	$0.47
R1	9C08053A4023JLR	402 KΩ, 1%	Philips (817) 325-7871	$0.03
R2	9C08053A4752JLR	47.5 KΩ,1%	Philips	$0.03
			Total Cost	$2.63

*Cost estimates based on published 10K unit pricing at the time this application note was written.

6.2 Linear Technology Corporation LT1111-12

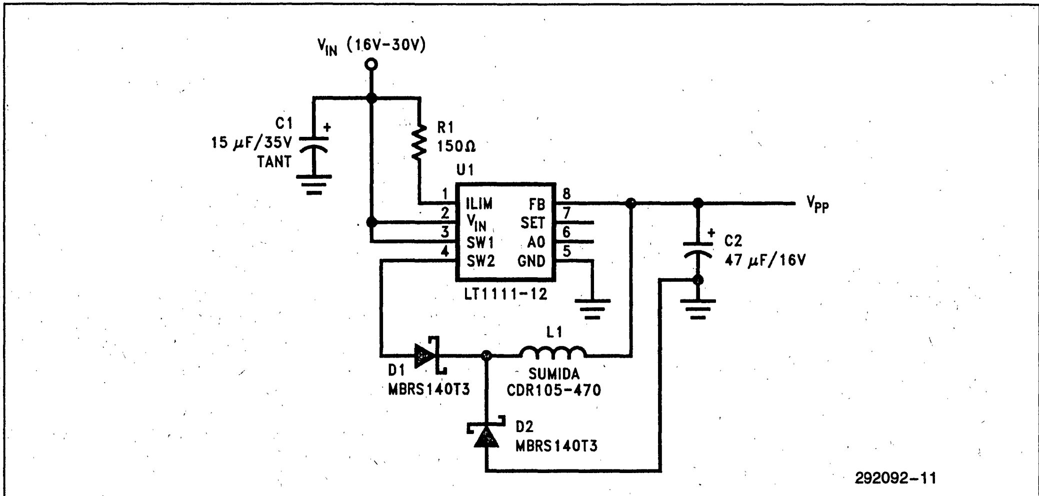


Figure 6-2. Linear Technology LT1111-12 Step Down Switcher

Optimal Attributes

- High Efficiency
- All Surface Mount

Main Features

- Input Voltage Range: 16V to 30V
- Output Voltage: 12V ±5%
- Output Current Capability: Up to 120 mA
- Typical Efficiency: 80%

Table 6-2. Parts List for the LT1111-12 12V Step Down Converter

Ref	Part #	Value/Type	Source	Cost*
U1	LT1111-12	SMPS IC- SO8 Package	Linear Tech (408) 432-1900	$2.20
C1	267M3502-225-MR-720	2.2 µF/35V Tantalum	Matsuo (714) 969-2491	$0.28
C2	267M1602-476-MR-720	47 µF/16V Tantalum	Matsuo (714) 969-2491	$0.47
R1	9C08052A1500JLR	150Ω, 5%	Philips (817) 325-7871	$0.02
L1	CDR105-470	47 µH	Sumida (708) 956-0666	$0.38
D1, D2	MBRS140T3	Schottky Diode	Motorola (800) 521-6274	$0.60
			Total Cost	$3.95

*Cost estimates based on published 10K unit pricing at the time this application note was written.

6.3 National Semiconductor LM2940CT-12

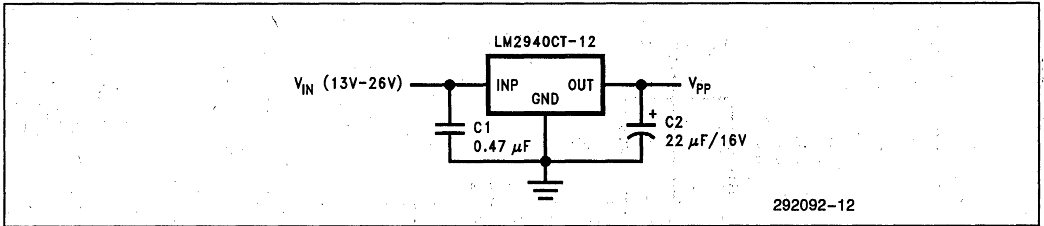


Figure 6-3. National LM2940CT-12 12V Linear Regulator

Optimal Attributes

- Lowest Cost

Main Features

- Input Voltage Range: 13V to 26V
- Output Voltage: 12V ±3%
- Output Current Capability: 1A

The LM2940CT-12 is a low drop-out linear regulator from National Semiconductor. This is a good low cost fixed 12V output solution. The part is offered in a standard TO-220 plastic package. The input capacitor is required only if the regulator is located far away from the input power supply filter, and the output capacitor must be at least 22 μF in order to maintain stability.

Table 6-3. Parts List for the LM2940CT-12 Step Down Converter

Ref	Part #	Value/Type	Source	Cost*
U1	LM2940CT-12	Voltage Reg TO-220	National (408) 721-5000	$0.95
C1	GRM43-2Z5U474M050AD	0.47 μF/50V	Murata Erie (404) 436-1300	$0.07
C2	267M1602-226-MR-720	22 μF/16V Tantalum	Matsuo (714) 969-2491	$0.28
			Total Cost	$1.30

*Cost estimates based on published 10K unit pricing at the time this application note was written.

7.0 OBTAINING V_{PP} FROM 12V UNREGULATED

In systems like the desktop computer, a 12V supply exists but may not be regulated to ±5%. If this voltage is used as the V_{PP} source for flash memory, it may well degrade the write/erase performance of the memory, or adversely affect its reliability. Fortunately, in most of the situations where a 12V unregulated (or not regulated to within 5%) supply exists, a 5V supply also exists in the system (the desktop computer is a good example). It is recommended in such cases that the existing 5V supply be used to obtain the 12V ±5% rail. This approach is more economical, more efficient, and provides space savings over a buck-boost topology that takes unregulated 12V and regulates it to ±5%.

In the rare case where a 5V supply is not present, modular solutions exist that will regulate the unregulated 12V supply to ±5%. However, these are bulky and expensive. Moreover, many of them require that a minimum load be maintained in order to stay in regulation. One such solution in presented in Appendix A.

8.0 SUMMARY

For battery powered applications, the author views the discrete switching regulator IC solution as a better choice than the modular solution. The lower cost, higher efficiency, and smaller size/height associated with discrete solutions justify the small additional design effort required to incorporate them in flash memory applications. In applications where the primary source of power is a wall power outlet, or in applications where the flash memory will be written to infrequently, efficiency and quiescent current take on secondary importance. In such cases, it may be acceptable to use a 12V regulated (to within ±5%) tap from the system supply. Alternatively, the ability to easily design-in modular solutions may outweigh the disadvantages of lower efficiency and increased cost. For those users wishing to incorporate modular solutions, Appendix A provides some of the lower cost solutions from this industry segment.

APPENDIX A MODULAR SOLUTIONS

Modular solutions may work well in non-battery powered situations where the efficiency of the power supply converter is not critical. These are also advantageous in that they usually do not need any external components and there is no converter design involved. However, the type and quality of the discrete components used in these hybrid solutions is open to question. This is not true in the case of the discrete converter designs presented in the earlier sections, where the quality of the components used are under the control of the system design engineer. Hence, even though modular solutions offer the convenience of a single package and ease of testability, the quality/reliability of comparably priced modular solutions may be questionable.

Some modular solutions suited to flash memory applications are presented below, with a brief description of each. Sources for obtaining these are listed in Appendix B.

A.1 International Power/Newport Components NMF0512S

The NMF0512S is a 5V to 12V hybrid power module that has an output current capability of 80 mA. Output tolerance is ±5%. It is equipped with a shutdown pin which can be used to switch V_{PP} off. However, power dissipated in the shutdown mode is relatively high (about 100 mW). The part is small in size and measures 0.76 in. (19.5 mm) x 0.4 in. (9.8 mm) x 0.4 in. (9.8 mm), and costs about $7.90 in 10K quantities (at the time this application note was written). Typical efficiency of conversion is 62%.

A.2 Xentek NPSC-0512S

The Xentek NPSC-0512S is a 1W power module that converts 5V to V_{PP} and will source up to 80 mA of continuous current. However, it uses two external filter capacitors—one at the input and one at the output. The input filter capacitor is 47 μF/10V, and the output filter capacitor is 100 μF/16V. Size of the solution (converter alone) is 0.87 in. (22 mm) x 0.39 in. (10 mm) x 0.79 in. (20 mm). The NPSC-0512S does not have a shutdown mode. The part costs around $5.00 in 10K quantities (at the time this application note was written). Typical efficiency of conversion is 60%.

A.3 Shindengen America Inc. HDF-0512D

The HDF-0512D module from Shindengen will convert unregulated 12V to 12V ±5%. This part is a dual output part (±12V), but only the +12V line is used. The conversion efficiency is high (75% typical), and the part will provide a regulated V_{PP} voltage from input voltages as low as 8V, and as high as 16.5V. A minimum load of 5 mA needs to be maintained to guarantee regulation. Size of the solution is 1.75 in. (44 mm) x 0.43 in. (11 mm) x 0.8 in. (20 mm). Cost is approximately $10.00 in quantities of 10K (at the time this application note was written).

APPENDIX B
SURVEY OF SOLUTIONS PRESENTED

Ref #	Vendor Name	Part #	Input C (Volts)	Output V (Volts)	Output C (mA)	Effic (%)	# Ext Comp (Note 1)	100% SMD ?	Cost (Note 2)	PC Area (Note 3)	Height (In)	SHDN ?	ISHDN (Note 4)	R Time (Note 5)	Temp
3.1	Maxim	MAX732	4V–7V	12V, 4%	120	90	5; D, L, 3C	Yes	$3.93	0.56	0.18	Yes	70 µA	1 ms	0°C, +70°C
3.2	Linear Tech	LT1110-12	5V, 10%	12V, 5%	120	76	7; D, L, T, 2R, 2C	Yes	$4.58	0.45	0.20	Yes	16 µA	1 ms	0°C, +70°C
3.3	Linear Tech	LT1109-12	5V, 10%	12V, 5%	60	84	4; D, L, 2C	Yes	$3.61	0.38	0.18	Yes	375 µA	1 ms	0°C, +70°C
3.4	Motorola	MC34063A	5V, 10%	12V, 5%	120	75	11; D, L, T, 3C, 5R	Yes	$2.25	0.49	0.18	Yes	25 µA	2 ms	0°C, +70°C
4.1	Linear Tech	LT1110-12	2V–3.1V	12V, 5%	30	70	7; D, L, T, 2R, 2C	Yes	$4.71	0.45	0.18	Yes	16 µA	4 ms	0°C, +70°C
4.2	Maxim	MAX732	1.8V–4V	12V, 4%	30	87	9; D, L, 7C,	Yes	$4.80	0.7	0.18	Yes	45 µA	25 ms	0°C, +70°C
4.3	Maxim	MAX732	1.8V–4V	12V, 4%	60	85	8; D, L, 6C,	No	$4.15	1.11	0.49	Yes	45 µA	75 ms	0°C, +70°C
5.1	Maxim	MAX658	2V–3.1V	5V, 5%	250	85	7; D, 2L, T, 3C	No	$4.47	0.92	0.39	Yes	80 µA	25 ms	0°C, +70°C
5.2	Linear Tech	LT1110-5	2V–3.1V	5V, 5%	150	76	7; D, L, T, 2R, 2C	Yes	$4.72	0.45	0.20	Yes	16 µA	1 ms	0°C, +70°C
6.1	Maxim	MAX667	12.1V–16V	12V, 5%	250	75	3; 2R, C	Yes	$2.73	0.25	0.15	Yes	0.2 µA	0.1 ms	0°C, +70°C
6.2	Linear Tech	LT1111-12	16V–30V	12V, 5%	120	80	6; 2D, L, 2C, R	Yes	$3.95	0.78	0.2	No	N/A	N/A	0°C, +70°C
6.3	National	LM294OCT-12	13V–26V	12V, 3%	1A	$12/V_{IN}$	2; 2C	No	$1.30	0.5	0.18	No	N/A	N/A	0°C, +70°C
A.1	International Power	NMF0512S	5V, 10%	12V, 5%	80	62	0	No	$7.90	0.3	0.40	Yes	20 mA	10 µs	−40°C, +70°C
A.2	Shindengen	HDF1212D	8V–16.5V	12V, 5%	120	77	0	No	$10.00	0.76	0.80	No	N/A	N/A	−10°C, +70°C
A.3	Xentek	NPSC-0512S	5V, 10%	12V, 5%	80	60	2; 2C	No	$5.50	0.34	0.79	No	N/A	N/A	−10°C, +70°C

NOTES:
1. # External components. D: Diode, L: Inductor, C: Capacitor, R: Resistor, T: Transistor.
2. Cost. Cost estimates assume 10K quantities, based on published pricing at the time this application note was written.
3. PC Area. PC Aarea is conservatively estimated as 2.0x (area of all components). Where actural layouts are presented, the lower value is given. Note that this estimate is for a single sided board, and area can be reduced considerably if both sides of the board are utilized.
4. I Shdn. Current consumed by supply at shutdown. Output settles to V_{CC} at shutdown, so some additional flash V_{PP} leakage/standby will exist.
5. R Time. Rise time from shutdown state. Erase/Writes should not be attempted till V_{PP} level has risen to valid level after shutdown is disabled.

APPENDIX C
SOURCES/CONTACTS FOR RECOMMENDED DC-DC CONVERTERS

Linear Technology Corporation

Recommended Products:

- LT1110-12 (DC-DC Converter IC)
- LT1110-5 (DC-DC Converter IC)
- LT1109-12 (DC-DC Converter IC)
- LT1111-12 (DC-DC Converter IC)

In U.S.A.:
1630 McCarthy Blvd.
Milpitas, CA 95035-7487
Tel: (408) 432-1900
Fax: (408) 432-0507

In Europe (U.K.):
111 Windmill Road
Sunbury
Middlesex TW16 7EF
U.K.
Tel (44)(932) 765688
Fax (44)(932) 781936

In Asia (Japan):
4F Ichihashi Bldg
1-8-4 Kudankita Chiyoda-ku
Tokyo 102 Japan
Tel (81) (03) 3237-7891
Fax (81) (03) 3237-8010

Maxim Integrated Products

Recommended Products:

- MAX732 (DC-DC Converter IC)
- MAX658 (DC-DC Converter IC)
- MAX667 (DC-DC Converter IC)

In U.S.A.:
120 San Gabriel Drive
Sunnyvale, CA 94086
Tel (408) 737-7600
Fax (408) 737-7194

In Europe (U.K.):
Maxim Integrated Products (UK), Ltd.
Tel: (44) (734) 845255

In Asia (Japan):
Maxim Japan Co., Ltd.
Tel: 81 (03) 3232-6141

Motorola Semiconductor Inc.

Recommended Product:

- MC34063AD (DC-DC Converter IC)

In U.S.A.:
616 West 24th Street
Tempe, AZ 85282
Tel: (800) 521-6274

In Europe (U.K.):
Tel: (44) (296) 395-252

In Asia (Japan):
Tel: (81) (3) 440-3311

National Semiconductor

Recommended Product:

- LM2940CT-12 (Voltage Regulator IC)

In the U.S.:
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052
Tel: (408) 721-5000

In Europe:
National Semiconductor (UK) Ltd.
The Maple, Kembrey Park
Swindon, Wiltshire SN26UT
U.K.
Tel: (07-93) 614141
Fax: (07-93) 697522

In Asia:
National Semiconductor Japan Ltd.
Sanseido Bldg. 5F
4-15 Nishi Shinjuku
Shinjuku-ku
Tokyo 160 Japan
Tel: (81) (3) 299-7001
Fax: (81) (3) 299-7000

Newport Components/ International Power

Recommended Product:

— NMF0512S (5V–12V Converter Module)

In U.S.A.:
International Power Sources
200 Butterfield Drive
Ashland, MA 01721
Tel: (508) 881-7434
Fax: (508) 879-8669

In Europe:
Newport Components
4 Tanners Drive
Blakelands North
Milton Keynes MK14 5NA
Tel: (0908) 615232
Fax: (0908) 617545

Shindengen Electric Co. Ltd.

Recommended Product:

— HDF0512D (12V unreg. to 12V reg. converter module)

In the U.S.:
2649 Townsgate Road #200
Westlake Village, CA 91361
Tel: (800) 634-3654
Fax: (805) 373-3710

In Europe:
Shindengen Magnaquest U.K. Ltd.
Unit 13, River Road,
Barking Business Park,
33 River Road, Barking,
Essex 1G11 ODA
Tel: (44) (81) 591-8703
Fax: (44) (81) 591-8792

In Asia:
2-1,2-Chome Ohtemachi
Chiyoda-ku
Tokyo 100
Japan
Tel: (81) (3) 279-4431
Fax: (81) (3) 279-6478

Xentek Inc.

Recommended Product:

— NPSC0512S (5V–12V Converter Module)

In U.S.A.:
760 Shadowridge Drive
Vista, CA 92083
Tel: (619) 727-0940
Fax: (619) 727-8926

In Europe (Germany):
Xentek, Inc.
c/o Taiyo Yuden GMBH.
Obermaierstrasse 10,
D-8500 Nurnberg 10
Federal Republic of Germany
Tel: (49) (911) 350-8400
Fax: (49) (911) 350-8460

In Asia (Japan):
Xentek, Inc,
c/o Taiyo Yuden., Ltd.
6-16-20, Ueno, Taito-ku
Tokyo 110
Japan
Tel: (81) (3) 3837-6547
Fax: (81) (3) 3835-4752

3

APPENDIX D
CONTACTS FOR DISCRETE COMPONENTS

Matsuo Electric Co., Ltd.

Matsuo's 267 series surface mount tantalum chip capacitors are recommended by Maxim and Linear Technology for input and output filter capacitors on their DC-DC converters. Part #s are included on the parts list that accompanies most solutions. If alternate "equivalents" are required, choose high reliability, low ESR (Equivalent Series Resistance) and low ESL (Equivalent Series Inductance) type tantalums, which help in keeping output ripple and switching noise to a minimum.

In U.S.A.:
2134 Main St., Ste. 200
Huntington Beach, CA 92648
Tel: (714) 969-2491
Fax: (714) 960-6492

In Europe:
Steucon - Center II Mergenthalleralle 77
D-6236 Eschben/Ts.
Federal Republic of Germany
Tel: 6196-470-361
Fax: 6196-470-360

In Asia:
Oak Esaka Bldg.
10-28 Hiroshiba-Cho
Suita-shi
Osaka 564
Tel: (06) 337-6450
Fax: (06) 337-6456

Sumida Electric Co. Ltd.

Sumida CD series surface mount inductors are recommended by Maxim, Linear Technology for their miniature size and relatively low cost. These are well suited to low power DC-DC converter applications. Contact Sumida Electric directly for procuring these. The part #s are included in the parts list that accompanies most solutions. In applications where noise (EMI) is a concern, shielded varieties are also offered by Sumida.

In U.S.A.:
637 East Golf Road
Suite 209
Arlington Heights, IL 60005
Tel: (708) 956-0666
Fax: (708) 956-0702

In Asia:
4-8 Kanamachi 2-chome,
Katsushika-ku,
Tokyo 125
Japan
Tel: (81) (03) 3607-5111
Fax: (81) (03) 3607-5428

Coiltronix Inc.

Coiltronix is recommended as a good alternate source for surface mount inductors. The CTX series offered by Coiltronix is well suited to DC-DC converter applications. These are shielded, and have a toroidal core. However, they are bigger in size and currently much more expensive (7X to 8X) than the Sumida varieties recommended in the solutions herein. The equivalent part numbers are:

Sumida CD54-470 → Coiltronix CTX50-1
Sumida CD54-180 → Coiltronix CTX20-1
Sumida CD54-220 → Coiltronix CTX20-1
Sumida CD75-470 → Coiltronix CTX50-2
Sumida CDR105-470 → Coiltronix CTX50-2

In U.S.A.:
Coiltronix Inc.
984 S.W. 13th Court
Pompano Beach, FL 33069
Tel: (305) 781-8900
Fax: (305) 782-4163

In U.K.:
Microelectronics Technology Ltd.
Great Haseley Trading Estate
Great Haseley
Oxfordshire OX9 7PF
U.K.
Tel: (08) 44 278781
Fax: (08) 44 278746

In Asia:
Serial System Mktg.
Poh Leng Bldg., #02-01
21 Moonstone Lane
Singapore 1232
Tel: 2938830
Fax: 2912673

Coilcraft

Coilcraft is also recommended as a good alternate source for surface mount inductors. The N2724-A shielded series is well suited to DC-DC converter applications. These are bigger and currently more expensive (2x to 3x) than the Sumida inductors recommended in the solutions. Contact Coilcraft directly for any applications assistance or for procurement of these parts. The equivalent part numbers are:

Sumida CD54-470 → Coilcraft N2724-A 47 μH
Sumida CD54-180 → Coilcraft N2724-A 18 μH
Sumida CDR105-470 → Coilcraft N2724-A 47 μH

In the US:
1102 Silver Lake Road
Cary , IL 60013
Tel: (708) 639-6400
Fax: (708) 639-1469

In Europe:
21 Napier Place
Wardpark North
Cumbernauld
Scotland G68 0LL
Tel: 0236 730595
Fax: 0236 730627

In Asia:
Block 101, Boon Keng Road
#06-13/20
Kallang Basin Industrial Estate
Singapore 1233
Tel: 2966933
Fax: 2964463

Philips Components

Philips Components is recommended as a good source for surface mount (SMD) resistors (standard 9C series, and 9B (MELF) series). Part #s are included in the parts list that accompanies most of the solutions in the application note. Many alternate sources exist.

In the US:
2001 W. Blue Heron Blvd.
P.O. Box 10330
Riviera Beach, FL 33404
Tel: (407) 881-3200
Fax: (407) 881-3304

In Europe:
Philips Components Ltd.
Mullard House
Torrington Place
London WC1E 7HD
Tel: (44) 71 580 6633
Fax: (44) 71 636 0394

In Asia:
Philips K.K.
Philips Bldg. 13-37
Kohnan 2-chome
Minato-Ku Tokyo 108
Tel: (81) 3 740-5028
Fax: (81) 3 740-5035

Siliconix-Logic Level PFETs

Siliconix offers low-"on" resistance logic level PFETs (Si9400, and Si9405) that can be used for switching a DC-DC converter into a shutdown state by using these switches on the high side of the input to the converter (see Appendix E).

In the US:
2201 Laurelwood Road
P.O. Box 54951
Santa Clara, CA 95056-9951
Tel: (408) 988-8000
Fax: (408) 727-5414

In Europe:
Weir House
Overbridge Square, Hambridge Lane
Newbury, Berks RG14 5UX
Tel: (0635) 30905
Fax: (0635) 34805

In Asia:
Room 709, Chinachem Golden Plaza
77 Mody Road
TST East Kowloon
Tel: (852) 724-3377
Fax: (852) 311-7909

APPENDIX E
OTHER DESIGN CONSIDERATIONS

E.1 V_{PP} Valid Handshake Logic

It is often desirable to have, along with the V_{PP} solution, a handshake signal (using extra hardware) that is asserted as long as the voltage level on V_{PP} is valid. The following schematic illustrates a good way of achieving this. This handshake signal could be used to determine when it is suitable to perform writes/erases on the flash device. The circuit shown uses a precision zener voltage reference and a comparator, along with bias resistors, to monitor the voltage level on V_{PP}. The point at which the comparator trips must be set after careful consideration of the variation in the reference voltage and the tolerances on the bias resistors. The worst case conditions on these variations must guarantee that the handshake signal is asserted when V_{PP} is at its worst case lower-end level (11.4V). Care must be taken to use the exact same components as specified in order to maintain the tight tolerance on the trip level of the output signal.

E.2 Obtaining Shutdown Using Logic Level PFETs

Low "on" resistance logic level PFETs can be used on the high side of the input to the DC-DC converters to obtain shutdown. One such part is the Si9405 from Siliconix Inc. The device is part of the "little foot" series, and is available in an SO8 (8-pin surface mount) package. The Si9405 is a logic level PFET with an "on resistance" of 0.2Ω (at a gate drive of 4.5V). It is important to have as low an "on" resistance as possible, since the peak currents and start-up currents into the supply are high. Care must be taken to ensure that the DC-DC conversion process is not affected after accounting for the drop in input voltage across the PFET.

E.3 Working of the Discrete Step Up Switching Regulator

This section presents a brief overview of the operation of discrete step up switching regulators, and presents issues that the user needs to be concerned with while designing these solutions into the system.

The four most basic elements of a discrete switching regulator power supply are:

1. The SMPS IC (which includes the switch control element and logic, along with the power switch itself),
2. An inductor for storage and transfer of energy between the input and output,
3. A switching diode to direct the inductor energy to "catch", or channel, the inductor energy to the output, and
4. An output filter capacitor.

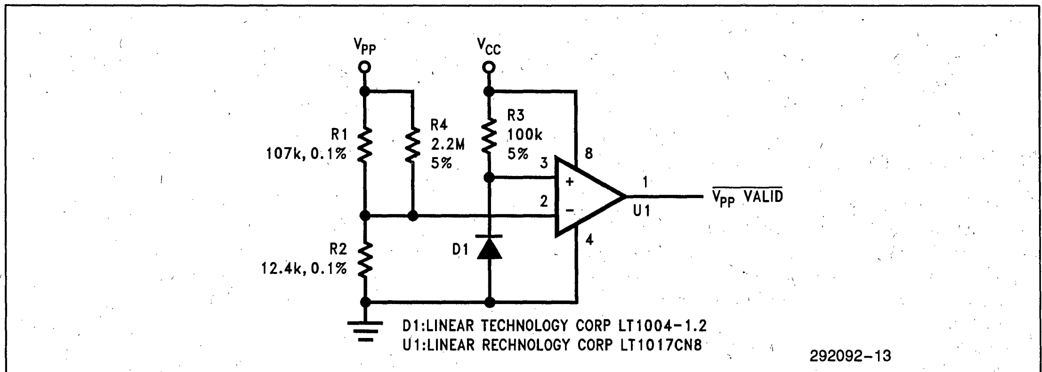


Figure E-1. V_{PP} Valid Handshake Circuit

In the boost configuration where the output voltage is greater than the input voltage, the basic switching power supply configuration is as shown in Figure E.2:

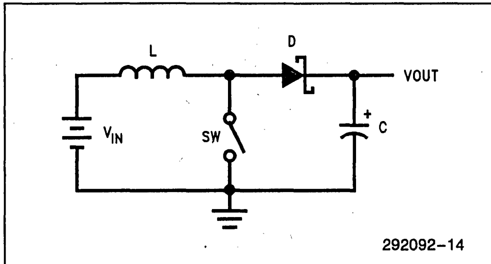


Figure E-2. Working of the Step-Up Switching Regulator

The power switch SW can be turned on and off; the control for it is derived from a feedback mechanism that senses the output voltage. While the switch is turned on, the inductor stores energy as the current flows through it from the input supply. The peak current through the inductor I_L can be approximated as $(V_{IN}/L * t_{ON})$; where t_{ON} is the on time of the switch. During this time, the energy is supplied by the input voltage, $V_L = V_{IN}$. The output is isolated from the inductor via the reverse-biased diode, and the load current is supplied by the output filter capacitor. When the switch turns off, the energy stored in the inductor appears as a rapidly increasing voltage across the inductor. As soon as this voltage reaches a value equal to the output voltage plus the voltage drop across the diode, the diode switches on and current starts to flow through the diode. This diode current supplies the load current while also at the same time charging up the output filter capacitor to the output voltage.

The switch is controlled by sensing the output voltage via a feedback mechanism—usually a pair of resistors. This sense voltage is gated via a comparator whose output acts as a control signal to an oscillator. The oscillator output controls the switch.

The power into the inductor P_L can be approximated as:

$$P_L = 0.5 * L * I_{PK}^2 * f_{OSC}$$

and the power into the load P_{LOAD} (out of the inductor) can be approximated as

$$P_{LOAD} = (V_{OUT} + V_D - V_{IN}) * I_{OUT}$$

The peak currents through the inductor is usually several times higher than the load current, is mostly of the value of the load current and builds up during time t_{ON}. On most of the solutions presented here, peak operating currents lie in the range of 500 mA to 1.2A. Though this may seem high, most of this in-rush of energy is transferred to the output, and little is lost to heat due to the efficient energy storage characteristic of inductors. Note that since the peak currents are high, the input voltage source must be capable of providing this current, and the current capability of the input source must not be calculated simply as $(V_{OUT} * I_{OUT})/(V_{IN} * Eff)$. A large bypass capacitor at the input pin of the converter is hence also necessary on all designs.

Some of the solutions presented in this application note are of the fixed duty cycle or fixed on time type (e.g. LT1110-12, LT1109-12, MC34063A), whereas some of them vary the duty cycle depending on the load current (e.g. MAX732, MAX658). These latter ones provide higher efficiencies.

Inductor Selection

The choice of an inductor is crucial to the design of the power supply system. To begin with, the inductor value must be low enough to supply the peak currents needed when the input voltage V_{IN}, as well as the on time t_{on}, are at their worst case low value. On the other hand, the inductor value must be high enough so that the peak currents at the worst case high values do not exceed the maximum peak currents that can be handled by the switch. Furthermore, once the value has been picked, the physical inductor that is chosen for the job must be able to handle these peak currents, and must not saturate. This is done by picking an inductor whose DC current rating is more than the worst case peak current that will be required by the operation of the device. The other characteristic to consider is the resistance of the inductor. In order to keep losses to a minimum, it is essential that the resistance of the coil is a minimum. Thus, it is important to use the inductors specified in the parts list that accompanies the solutions. These have been carefully chosen after reviewing the requirements. Alternate inductors may be used, as long as they are "equivalent".

EMI Concerns

Since the switching regulators presented in this application note switch at frequencies between 60 KHz and 200 KHz, there exists a potential for EMI. In cases where EMI may be a problem, shielded inductors can be used. This will reduce EMI significantly. Shielded versions of the inductors specified are readily available. Contact the vendor directly for these.

Output Switching Noise

Output switching noise has several sources. The most significant one is the IR drop through the ESR (Equivalent Series Resistance) of the output filter capacitor. This is caused by switching current pulses from the inductor. There is also noise in the form of switching spikes riding on the DC output. This is due to the output filter capacitor's ESL (Equivalent Series Inductance), current spikes in the ground trace and rectifier turn-on transients.

It is important to use low ESR and low ESL output and input filter capacitors. Proper layout is also essential in order to avoid spikes in the output. The safest solution is to use a filter circuit at the output. LC filters are not recommended, because of the transient nature of the load currents on flash devices. An RC filter is recommended on most solutions as an option. Two 1Ω resistors are used in parallel to avoid causing a significant drop across the resistance. This method is inexpensive and assures that the spikes riding on the output waveform are contained to within the 5% tolerance requirement on V_{PP}.

In addition, care must be taken to keep the leads from the output of the solution to all flash devices as short as possible. Use of a 0.1 μF capacitor at the V_{PP} pin of each flash device is highly recommended.

APPENDIX F
PC LAYOUTS FOR SOME RECOMMENDED SOLUTIONS

F.1 Maxim Integrated Products MAX732

The double-sided layout presented below (Figure F-1) has been designed for the MAX732 5V–12V converter solution (Section 3.1). It is a double sided layout and has been designed for the parts specified in the parts list that accompanies the solution. Contact Maxim for any additional layout assistance.

F.2 Linear Technology Corporation LT1110-12

The single-sided layout presented below (Figure F-2) can be used to implement the LT1110-12 5V to 12V converter (Section 3.2), the LT1110-12 3V–12V converter (Section 4.1), or the LT1110-5 3V to 5V converter (Section 5.2). The layout has been designed for the parts that are specified in the parts list that accompanies these solutions. Contact Linear Technology for any additional layout assistance.

F.3 Linear Technology Corporation LT1109-12

The single-sided layout presented below (Figure F-3) can be used to implement the LT1109-12 5V–12V converter solution (Section 3.3). The layout has been designed for the parts that are specified in the parts list that accompanies the solution. Contact Linear Technology for any additional layout assistance.

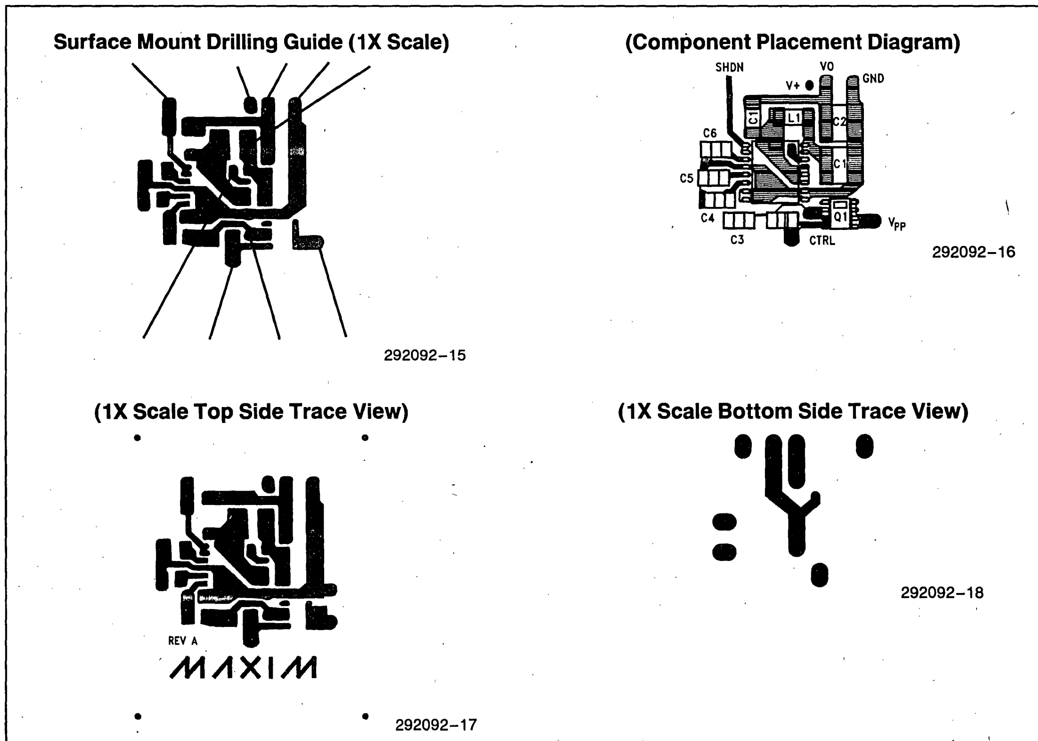


Figure F-1

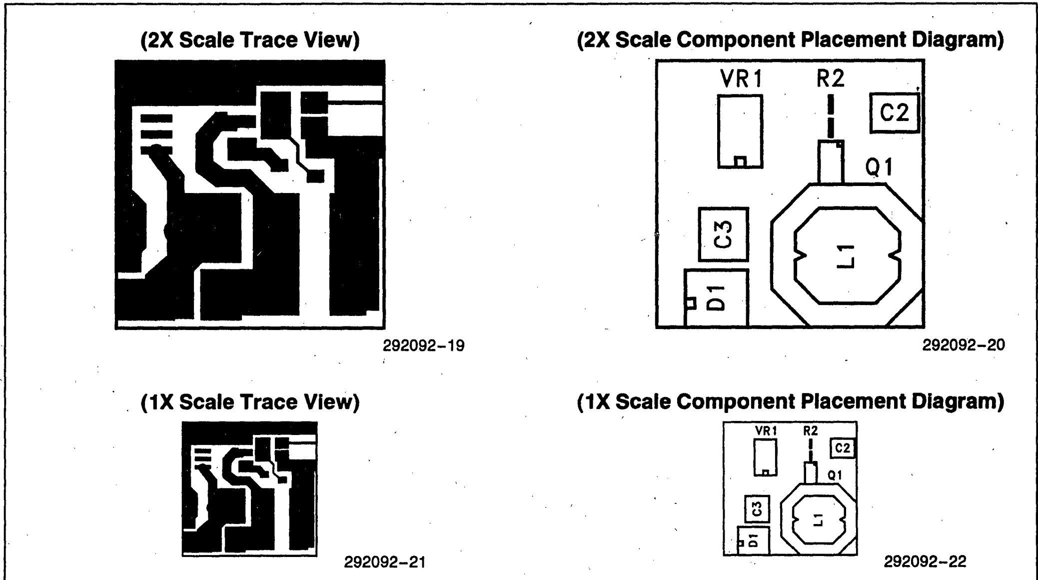


Figure F-2

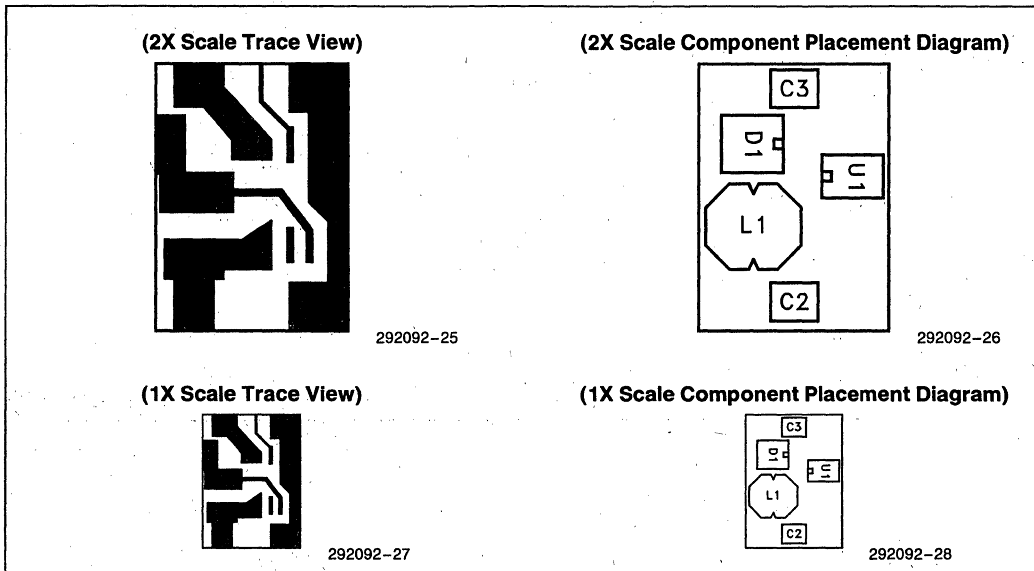


Figure F-3

APPLICATION NOTE

AP-359

August 1992

28F008SA Hardware Interfacing

BRIAN DIPERT
MCD MARKETING APPLICATIONS

Order Number: 292094-002

28F008SA Hardware Interfacing

CONTENTS PAGE

1.0 INTRODUCTION 3-434

2.0 HARDWARE INTERFACING 3-435

2.1 V_{PP} (Byte Write/Block Erase Voltage) 3-435

V_{PP} Generation Circuits 3-436

Controlling V_{PP} to 28F008SA Component(s) 3-436

2.2 RY/$\overline{BY}$ (Ready/Busy) Output 3-437

2.3 $\overline{PWD}$ (Powerdown) Input 3-437

Deep Powerdown Mode 3-437

Write Protection 3-438

2.4 $\overline{WE}$ (Write Enable) Input 3-438

2.5 High Density/In^2 Layout 3-438

2.6 Power Supply Decoupling 3-439

2.7 High Speed Design Techniques .. 3-440

2.8 Example Bus Interfaces 3-440

CONTENTS PAGE

ADDITIONAL INFORMATION 3-440

APPENDIX A: Intel386™ SL PI Bus Interface 3-441

APPENDIX B: Intel486™ SX Local CPU Bus Interface 3-442

3

1.0 INTRODUCTION

The 28F008SA FlashFile™ Memory is a very high performance 8 Mbit (8,388,608 bit) memory, organized as 1 Mbyte (1,048,576 bytes) of 8 bits each. The 28F008SA contains sixteen 64 Kbyte (65,536 byte) blocks, each block separately eraseable and capable of 100,000 byte write-block erase cycles. On-chip automation dramatically simplifies software algorithms, and frees the system microprocessor to service higher priority tasks during component data update. An enhanced system interface allows switching the 28F008SA into a deep powerdown mode during periods of inactivity, and gives a hardware indication of the status of the internal Write State Machine. High-speed access time allows minimal wait-state interfacing to microprocessor buses, and advanced packaging provides optimum density/in^2.

Features of the 28F008SA include:

- High-Density Symmetrically Blocked Architecture:
  - Sixteen 64 Kbyte Blocks
- Extended Cycling Capability
  - 100,000 Block Erase Cycles
  - 1.6 Million Block Erase Cycles per Chip
- Automated Byte Write and Block Erase
  - Command User Interface
  - Status Register
- System Performance Enhancements
  - RY/$\overline{\text{BY}}$ Status Output
  - Erase Suspend Capability
- Deep Powerdown Mode
  - 0.20 μA I_{CC} Typical
- Very High Performance Read
  - 85 ns Maximum Access Time
- SRAM-Compatible Write Interface
- Hardware Data Protection Features
  - Erase/Write Lockout during Power Transitions
- Industry Standard Packaging
  - 40 Lead TSOP, 44 Lead PSOP
- ETOX™ III Nonvolatile Flash Memory Technology
  - 12V Byte Write/Block Erase

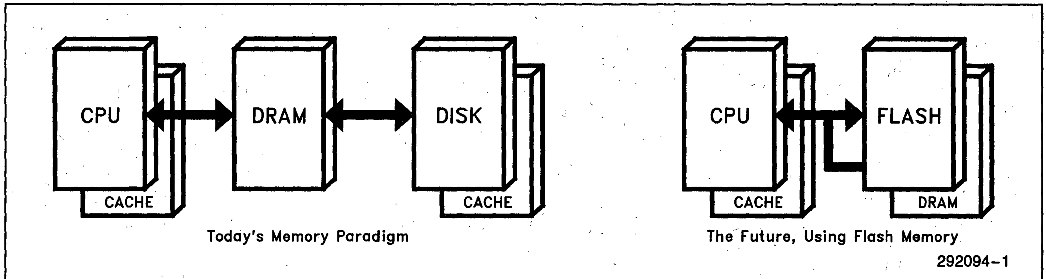


Figure 1. The 28F008SA Revolutionizes the Architecture of Computing

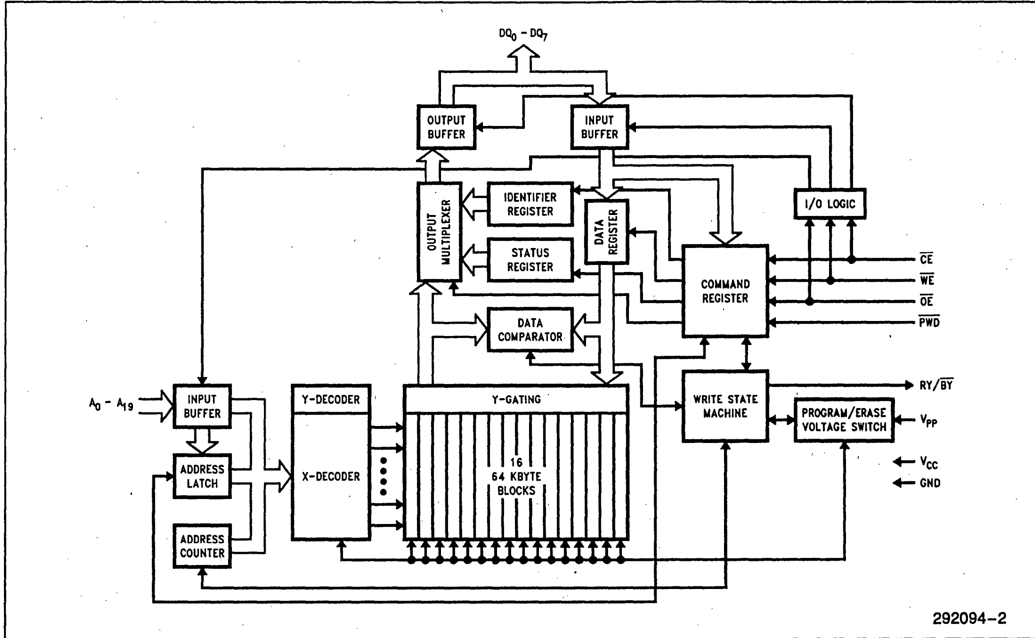


Figure 2. 28F008SA Block Diagram

Traditional system architectures combine slow, high density nonvolatile mass storage (such as a disk drive) and fast, volatile memory (such as DRAM) to fully address system requirements. As Figure 1 illustrates, flash memory combines the best features of both the above memory technologies, making a "disk/DRAM" approach to system architecture unnecessary and ultimately wasteful. Flash memory is rapidly approaching DRAM in both cost and performance (especially in cached systems), while adding capabilities (such as nonvolatility), that DRAM cannot claim. The 28F008SA will be the building block memory of choice for emerging computing markets, whether integrated in a memory card or disk drive form factor, or resident on the system motherboard.

This application note discusses hardware interfacing of the 28F008SA flash memory to system designs. The 28F008SA datasheet (order number 290429) is a valuable reference document, providing in-depth device technical specifications, package pinouts and timing waveforms. Additionally, companion application note AP-360, "28F008SA Software Drivers" (order number 292095) provides example ASM-86 and "C" routines for controlling the 28F008SA. AP-364 "28F008SA Automation and Algorithms" discusses in-depth operation of the 28F008SA Write State Machine and internal algorithms, emphasizing how they interface to system software and hardware. AP-360 and AP-364 should be reviewed in conjunction with this application note and the 28F008SA datasheet for a complete understanding of this device.

2.0 HARDWARE INTERFACING

Figure 2 shows a block diagram of the 28F008SA and its internal contents. The $\overline{CE}$ (chip enable) and $\overline{OE}$ (output enable) inputs have comparable enable and read functions to those of other memory technologies such as SRAM. Similarly, V_{CC} is the component power supply (5V ±10%), while GND should be connected to system ground. Address inputs allow the system to select a specific byte for reading or writing/erasing, and the 8-bit data bus transfers information to and from the 28F008SA. The other control lines ($\overline{WE}$, $\overline{PWD}$, RY/$\overline{BY}$ and V_{PP}) are discussed below.

2.1 V_{PP} (Byte Write/Block Erase Voltage)

The V_{PP} input supplies high voltage to the 28F008SA to enable byte write and block erase. V_{PP} is specified at 12V ±5% (11.4V–12.6V). Attempting to byte write or block erase the 28F008SA beyond the 5% 12V tolerance is not recommended. V_{PP} above 12.6V can potentially result in device damage, and V_{PP} below 11.4V dramatically lengthens write/erase time and compromises data reliability. The 28F008SA is guaranteed to prevent byte write and block erase attempts with V_{PP} below 6.5V, and in this situation it reports a "low V_{PP} error" through the component Status Register (see AP-360, AP-364 or the 28F008SA datasheet).

V_{PP} Generation Circuits

12V is often already present in systems, used to power the hard drive, display, RS-232 circuitry, flash BIOS update, etc. If it meets the tolerance and current capability requirements of the 28F008SA, such a power supply could be used directly as the 28F008SA update voltage source. However, 12V is sometimes not present or otherwise required, and in such cases, the 28F008SA V_{PP} must be derived from existing voltages and supplies.

Fortunately, flash memory's rapidly increasing popularity has driven ever-improving 12V converter availability in the market. These solutions derive a regulated 12V from a wide range of input voltages, and offer varied levels of integration and current delivery capability. In general, the input for 12V converters should come from the unregulated system power source, particularly in battery-powered systems.

Table 1 lists and briefly describes several 12V generation solutions available at the time this document was published. This is by no means an exhaustive list, and does not reflect any specific recommendation by Intel Corporation. For in-depth information on power supply solutions for flash memory, reference Intel application note AP-357 (order number 292092), available through your local Intel sales office or distributor.

Controlling V_{PP} to 28F008SA Component(s)

Once 12V is available in the system, how is it controlled? One approach is to hard-wire 12V from the supply directly to the V_{PP} inputs of each 28F008SA in the system. The advantage here is in design simplicity and board space savings. The 28F008SA Command User Interface architecture and two-step byte write/block erase command sequences provide protection from unwanted data alteration even with high voltage present on V_{PP}. All 28F008SA functions are disabled with V_{CC} below lockout voltage V_{LKO} (2.2V), or when $\overline{PWD}$ is at V_{IL} (see section 2.3). This provides data protection during system powerup, when the minimally-loaded V_{PP} supply often ramps to 12V before V_{CC} (and therefore control inputs to the device) are stable.

For additional data protection, the system designer can choose to make the V_{PP} supply switchable via a GPIO (General Purpose Input/Output) line, enabling 12V to the 28F008SA only during byte write or block erase attempts. A switchable V_{PP} also minimizes power consumption by both the flash memory components and the 12V supply or converter (due to efficiency losses). Many 12V converters integrate an ENABLE input, eliminating external circuitry. If such an input is not available, a low drain-source resistance MOSFET switch such as the Motorola MTD4P05 can be used at the 12V supply output. An example schematic for this switch is shown in Figure 3. The calculations below show that the low drain-source resistance of the MTD4P05 will keep a 12V input within the 5% tolerance required by the 28F008SA.

$R_{DS} = 0.6\Omega$

$I_{PP} = 60$ mA

(worst case, two components being byte written or block erased)

ΔV_{SWITCH} DROP = (60 mA $\times$ 0.6Ω) = 0.04V

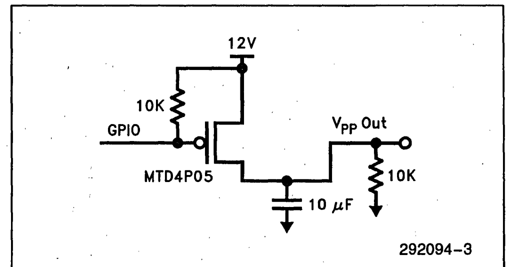


Figure 3. V_{PP} Switch Schematic

Table 1. 12V Conversion Solutions for V_{PP}

Manufacturer	Part Number	Input (V)	Package	Current Output	Total Components Needed	Est. Cost (10K)
Maxim	MAX732	4 to 7.5	16 SOIC	120 mA	9	$3.93
Linear Technology	LT1110-12	4.5 to 5.5	SO8	120 mA	11	$4.58
Linear Technology	LT1109-12	4.5 to 5.5	SO8	60 mA	8	$3.61
Motorola	MC34063A	4.5 to 5.5	SO8	120 mA	15	$2.25
Maxim	MAX667	12.1 to 16.5	SO8	120 mA	4	$2.63
Linear Technology	LT1111-12	16 to 30	SO8	120 mA	7	$3.95
National Semiconductor	LM2940CT-12	13 to 26	TO-220	1A	3	$1.30

2.2 RY/$\overline{BY}$ (Ready/Busy) Output

The 28F008SA offers similar automated byte write/ block erase capabilities to those first seen in the 28F001BX Bootblock flash memory family, introduced by Intel in May of 1991. It enhances these capabilities via the RY/$\overline{BY}$ output, which provides hardware indication of internal Write State Machine (WSM) operation. RY/$\overline{BY}$ is a full CMOS output, constantly driven by the 28F008SA and not tristated if the device $\overline{CE}$ or $\overline{OE}$ inputs are brought to V_{IH}. RY/$\overline{BY}$'s default state after device powerup is V_{OH}. It transitions low to V_{OL} when a byte write or block erase sequence is initiated by system software, and RY/$\overline{BY}$'s rising edge (return to V_{OH}) alerts the system to byte write or block erase completion. RY/$\overline{BY}$ also goes to V_{OH} after the 28F008SA is put in Erase Suspend or Deep Powerdown modes.

RY/$\overline{BY}$ is intended to interface the 28F008SA to a system microprocessor rising-edge-triggered interrupt input. In a multiple-chip memory array, external EPLD logic or an interrupt controller can be used to combine and prioritize RY/$\overline{BY}$s into one system interrupt (see Figure 4). The system can then, using a flash memory "activity table" set up in RAM, poll the individual 28F008SA Status Registers to determine which device has returned "ready", or read the RY/$\overline{BY}$ inputs directly at the EPLD, as shown.

Figure 5 provides an alternative method for connecting multiple RY/$\overline{BY}$s to one interrupt input. The diode/resistor combination converts the 28F008SA full CMOS output into an open-drain "wired-OR" equivalent. Any RY/$\overline{BY}$ at V_{OL} will drive the interrupt input low, and this input is pulled high by the resistors when all RY/$\overline{BY}$s are at V_{OH}. It is important in a design like this to use diodes with low forward voltage drops, so that the 28F008SA V_{OL} (0.45V) plus the diode voltage drop is still less than or equal to the destination input V_{IH} (0.8V). For the schematic shown in Figure 5, the equation is:

$$V_{OL} + V_{DIODE} = 0.45\ V_{MAX} + 0.3V = 0.75V \leq 0.8V$$

Note that should the system connect RY/$\overline{BY}$ to an interrupt, disable that interrupt prior to suspending erase, as RY/$\overline{BY}$ will transition to V_{OH} when the device is suspended.

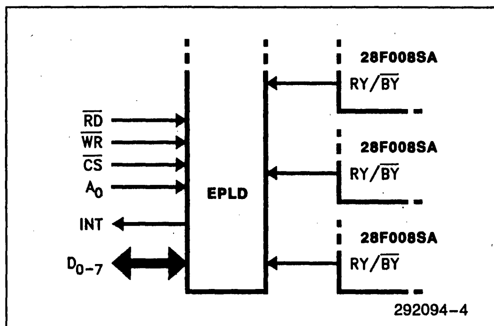


Figure 4. EPLD-Based RY/$\overline{BY}$ Implementation

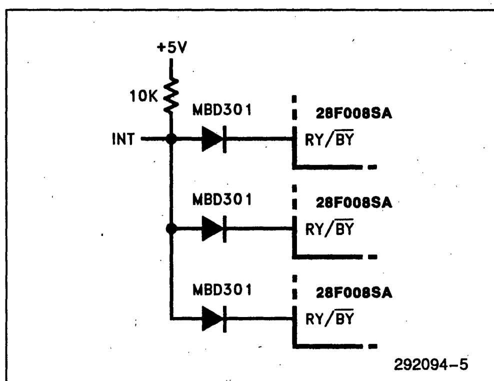


Figure 5. "Wired-OR" RY/$\overline{BY}$ Implementation

2.3 $\overline{PWD}$ (Powerdown) Input

Deep Powerdown Mode

The $\overline{PWD}$ input, when driven to V_{IL} by the system, switches the 28F008SA into a deep powerdown mode with negligable power consumption. This feature integrates the V_{CC} power FET often used with low power designs. Power consumption thru V_{CC} is typically 1 μW in deep powerdown mode. $\overline{PWD}$-low deselects the memory, places output drivers for D_{0-7} in a high-impedence state and turns off a majority of internal circuits. RY/$\overline{BY}$ is driven to V_{OH} while in deep powerdown mode. Depending on the flexibility desired, system designers can choose to put either the entire flash device array into deep powerdown mode, or any individual components via selective input control. The 28F008SA requires a "wakeup" time after $\overline{PWD}$ returns to V_{IH} before it can be successfully written (t_{PHWL}) or outputs are valid to read attempts (t_{PHQV}).

Write Protection

Since $\overline{PWD} = V_{IL}$ deselects the 28F008SA, this input can be used not only as a means of entering deep powerdown mode but also as an active-high "chip enable" to block spurious writes during system power transitions. Figure 6 shows one possible $\overline{PWD}$ implementation, controlled by a GPIO line for power management and by a system POWER GOOD for power sequencing protection. In this design, the 5V monitoring circuit begins functioning at $V_{CC} = 1V$, and will enable the device only after V_{CC} transitions above 4.6V (and system control signals are therefore stable). As V_{CC} drops below 4.6V during system powerdown, $\overline{PWD}$ protection is again activated.

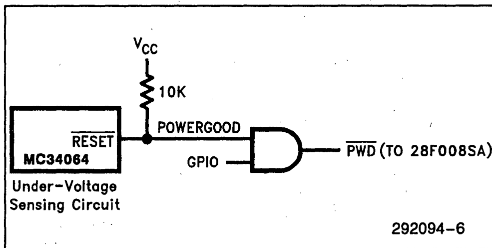


Figure 6. $\overline{PWD}$ Gating

2.4 $\overline{WE}$ (Write Enable) Input

When flash memory is written, the result can range from a 28F008SA that is placed in "read intelligent identifier" or "read Status Register" modes to alteration of nonvolatile flash memory contents. System hardware can prevent spurious writes to flash memory by application software or an operating system by gating the system $\overline{WE}$ to flash memory components to enable writes only when desired.

Figure 7 shows a simple design that gates $\overline{WE}$ with a GPIO line, enabling writes to the 28F008SA only when the GPIO is a "0". The GPIO is initialized to "1" on system powerup and the BIOS, a dedicated update software routine, a special keyboard sequence, switch on the back of the system or jumper on the system motherboard can then control the GPIO. This circuit ensures that flash memory contents are as permanent as "ROM" unless alteration is specifically desired.

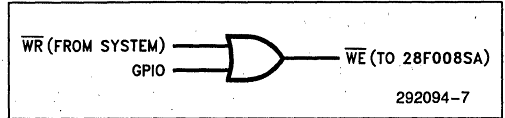


Figure 7. $\overline{WE}$ Gating

2.5 High Density/In² Layout

Figure 8 shows an 8 Mbyte flash memory array using TSOP (Thin Small Outline)-packaged 28F008SAs in standard (E) and reverse (F) configurations. A layout like this is used in Intel's Series II Flash Memory Cards (in densities to 20 Mbytes) and provides optimum array density for available board space.

Address and data lines are connected to all components in parallel. $\overline{OE}$ and $\overline{WE}$ are similarly connected. Section 2.7 of this document discusses alternate methods of implementing these signals for highest speed reads and writes in large memory arrays.

Component RY/$\overline{BY}$s are shown as not connected in Figure 8. They can be left unused, in which case the system software will substitute polling of component Status Registers for hardware interrupt, or RY/$\overline{BY}$s can be implemented as described in section 2.2.

$\overline{CE}$s are also not connected, intended to be individually driven by system chip enable decoding logic. This provides capability to read from and write to the array on a byte-by-byte basis. In a x16-only system, upper and lower byte 28F008SAs can have their $\overline{CE}$s bused together if desired.

Finally, V_{CC}, V_{PP} and $\overline{PWD}$ are connected in parallel to all components. Section 2.6 discusses bypass capacitor filtering of supply voltage inputs, while section 2.3 provides uses for $\overline{PWD}$. If desired, individual component, component pair, etc. selective powerdown control can be substituted for the global control shown in Figure 8.

In space-constrained designs, a multiple-layer partial "serpentine" trace layout at the edges of the 28F008SA array may be implemented, with a full serpentine layout within the array as in Figure 8.

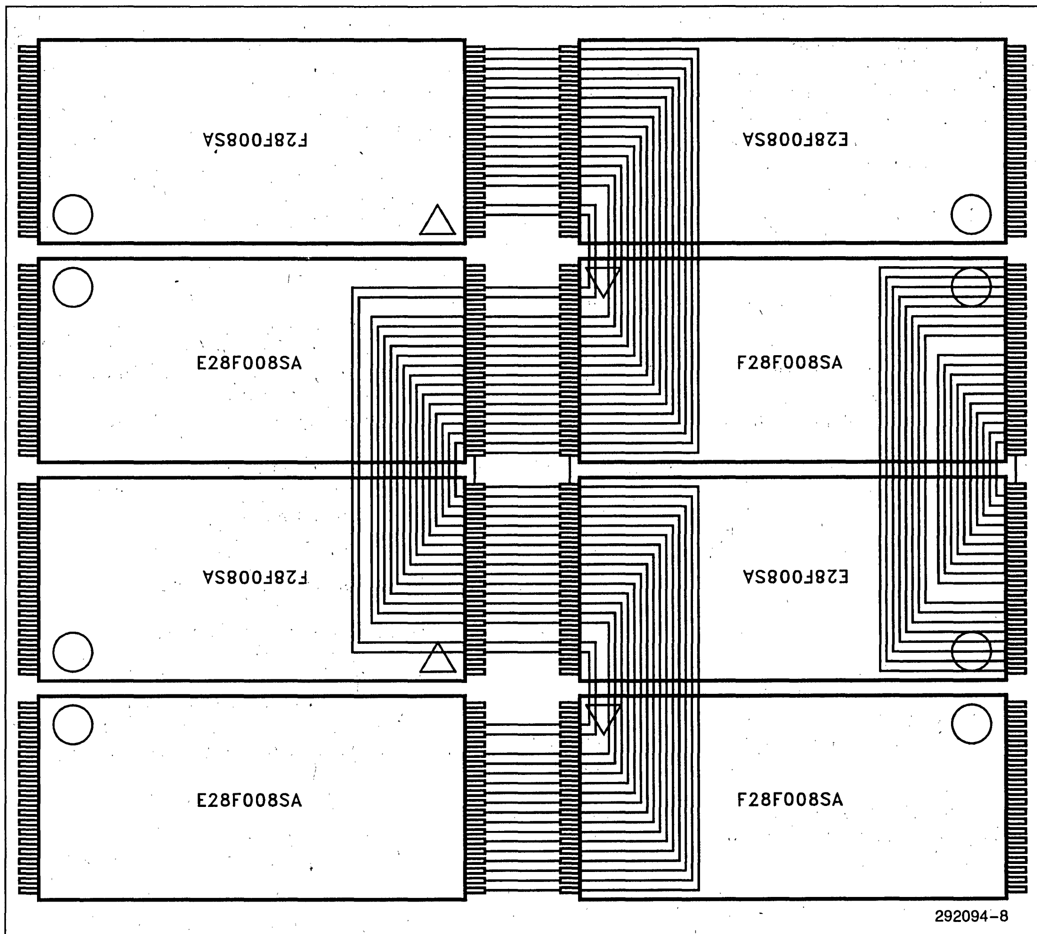


Figure 8. TSOP Serpentine Layout

2.6 Power Supply Decoupling

Both the V_{CC} and V_{PP} inputs to each 28F008SA should be decoupled at the package leads to provide noise immunity and supply current for transient current spikes during read, byte write and block erase. Additional bulk capacitance for groups of flash memories overcomes voltage slump caused by PC board trace inductances. Calculations for individual component and bulk capacitors (one per 8 devices) are shown below.

Basic Equation:

$I = C\ dv/dt$

Assumptions:

$I = 35$ mA per device (V_{CC}), therefore
$I = 17.5$ mA per device input (V_{CC})
$I = 30$ mA per device (V_{PP})
$dv = 0.1V$ (0.2V peak-peak)
$dt = 20$ ns

Per-Component-Input Decoupling Capacitor (V_{CC}):

$C = I\ dt/dv = (17.5\ mA \times 20\ ns)/0.1V = 3.5\ nF$

4x margin $= 4 \times 3.5\ nF = 14\ nF$

Standard Equivalent = **0.01 μF**

NOTE:
Calculations above assume that each 28F008SA is driving CMOS inputs (with corresponding high impedance and negligible input current requirements). If 28F008SA outputs are driving non-CMOS inputs, larger per-component capacitance may be needed to supply current while outputs are switching.

Bulk Capacitor (V_{CC}):

$$C = 10 \times (\text{Total of Decoupling Capacitors})$$

$$\text{Bulk Capacitor (4 Mbyte array)} = 10 \times (8 \times 0.01\ \mu F) = 0.8\ \mu F$$

$$\text{Standard Equivalent} = \mathbf{1\ \mu F}$$

Per-Component Decoupling Capacitor (V_{PP}):

$$C = I\ dt/dv = (30\ mA \times 20\ ns)/0.1V = 6\ nF$$

$$4x\ \text{margin} = 4 \times 6\ nF = 24\ nF$$

$$\text{Standard Equivalent} = \mathbf{0.033\ \mu F}$$

2.7 High Speed Design Techniques

The 28F008SA's fast read access and command write specifications make it a natural choice for high performance memory arrays. The following tips will optimize the memory interface for optimum read/write speed. The common recommendation in all instances centers around minimizing fanout and capacitive bus loading to allow highest switching speed, lowest rise and fall times, and therefore greatest performance.

- Minimize address bus loading from the microprocessor to the memory array. Multiple address latches feeding subsets of the array speed address input to each 28F008SA and $\overline{CE}$ decoding by external logic.
- Similarly, drive the memory array with multiple $\overline{OE}$s and $\overline{WE}$s. Most EPLD and discrete logic timing is specified at a 30 pF load, which equates to driving 4 28F008SA inputs at maximum input capacitance. Anything more than this may severely impact the logic's propagation delay.
- Finally, remember that each 28F008SA, when read, drives not only the system microprocessor or transceiver but also any other flash memory components connected to the common data bus. Each 28F008SA data output is specified at 12 pF, and the 28F008SA read timings are tested at either 30 pF or 100 pF of loading, depending on the chosen speed bin.

For large flash arrays where sequential data can be distributed on many devices, hardware interleaving provides additional performance.

2.8 Example Bus Interfaces

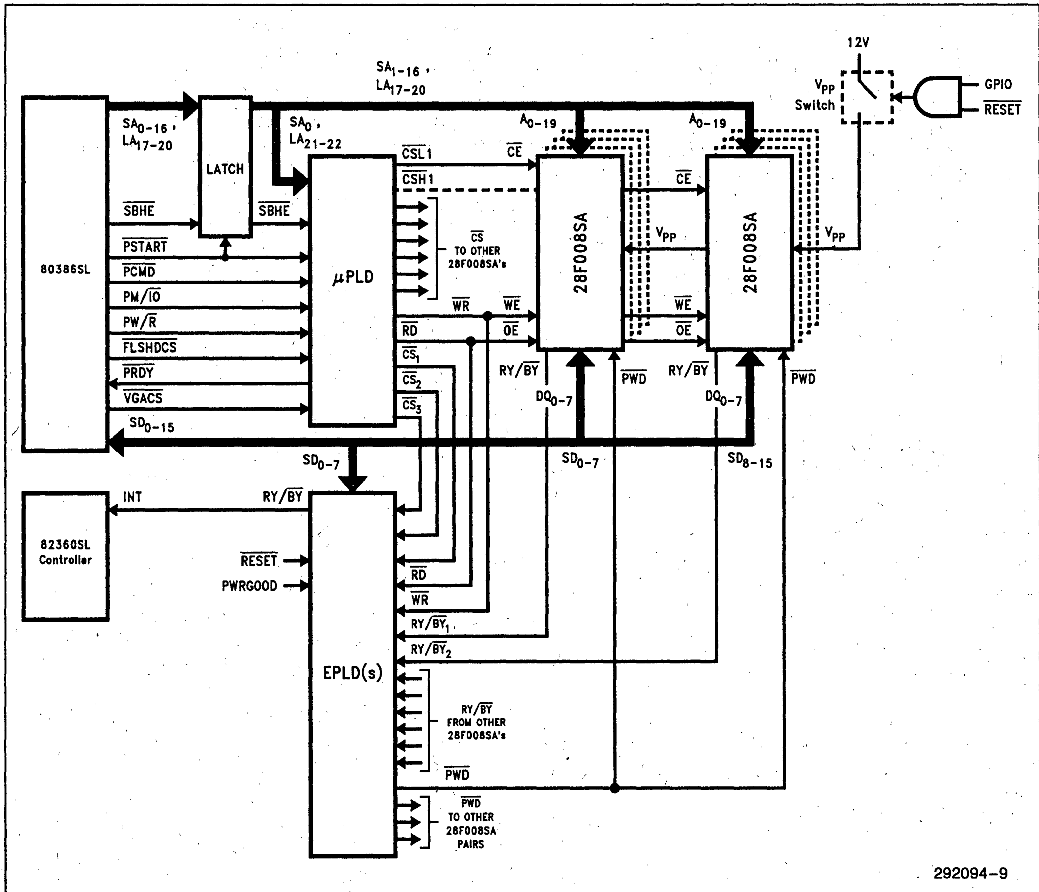
Appendix A shows hardware interface to the Intel386™SL PI bus, and Appendix B shows interface to the Intel486™SX local CPU bus. Both interfaces incorporate techniques described in sections 2.1–2.7 of this document. These designs are intended to be examples which can be modified to suit requirements of the end system.

ADDITIONAL INFORMATION

		Order Number
	28F008SA Datasheet	290429
	28F008SA-L Datasheet	290435
AP-357	"Power Supply Solutions for Flash Memory"	292092
AP-360	"28F008SA Software Drivers"	292095
AP-364	"28F0083A Automation and Algorithms"	292099
ER-27	"The Intel 28F008SA Flash Memory"	294011
ER-28	"ETOX™-III Flash Memory Technology"	294012

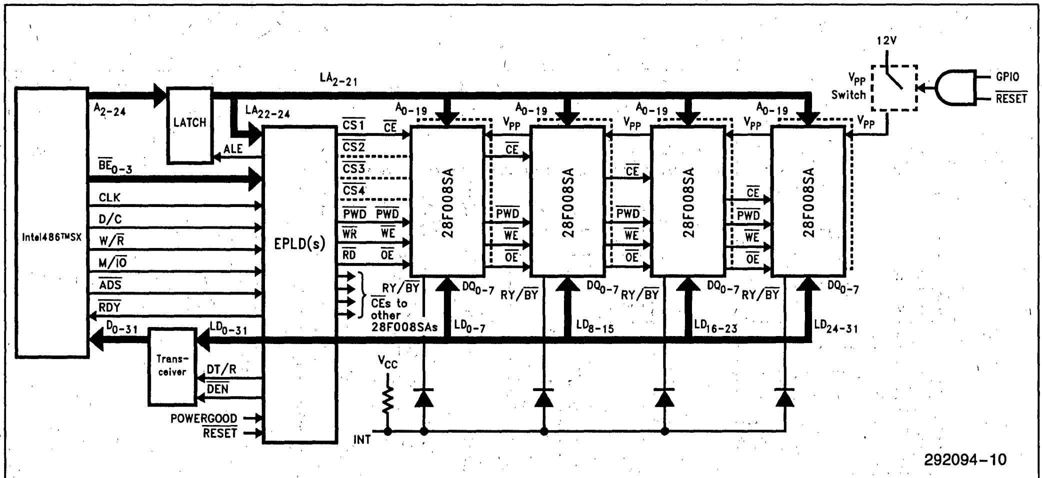
APPENDIX A

Intel386™SL PI BUS INTERFACE



NOTE:
The DRAM interface is not shown, for graphic simplicity.

APPENDIX B
Intel486™ SX LOCAL CPU BUS INTERFACE



NOTE:
The DRAM interface is not shown, for graphic simplicity.

REVISION HISTORY

Number	Description
-002	Added 10K resistor to FET output, Figure 3. Added AP-364 references.

intel®

APPLICATION NOTE

AP-360

November 1992

28F008SA Software Drivers

BRIAN DIPERT
MCD MARKETING APPLICATIONS

Order Number: 292095-002

28F008SA Software Drivers

CONTENTS PAGE

1.0 INTRODUCTION 3-445

2.0 ASM86 ASSEMBLY DRIVERS 3-446

3.0 "C" DRIVERS 3-450

ADDITIONAL INFORMATION 3-464

1.0 INTRODUCTION

This application note provides example software code for byte writing, block erasing and otherwise controlling Intel's 28F008SA 8 Mbit symmetrically blocked FlashFile™ Memory family. Two programming languages are provided; high-level "C" for multi-platform support, and ASM-86 assembly. In many cases, the driver routines can be inserted "as is" into the main body of code being developed by the system software engineer. The text accompanying each routine describes the existing code and suggests area for possible alteration to fit specific applications. These explanations, along with in-line commenting, minimize driver modification efforts.

Companion product datasheets for the 28F008SA and 28F008SA-L are valuable reference documents. Datasheets should be reviewed in conjunction with this application note for a complete understanding of the devices. AP-359, "28F008SA Hardware Interfacing" is the hardware-oriented application note equivalent for these devices and can also be referenced.

The internal automation of the 28F008SA makes software timing loops unnecessary and results in platform-independent code. This software is designed to be executed in any type of memory and with all processor clock rates. "C" code can be used with many microprocessors and microcontrollers, while ASM-86 assembly code provides the smallest code "kernal" for Intel microprocessors and embedded processors.

2.0 ASM-86 DRIVERS

```
;        Copyright Intel Corporation, 1992
;        Brian Dipert, Intel Corporation, February 8, 1992, Revision 1.0

;        Revision History: Rev 1.0

;        The following code controls byte write of data to a single 28F008SA (x8 write)
;        DS:[SI] points to the data to be written, ES:[DI] is the location to be written
;        In protected mode operation, DS and ES reference a descriptor
;        Register AX is modified by this procedure
WRITE_SETUP          EQU      40H
READ_ID              EQU      90H
INTEL_ID             EQU      89H
DEVICE_ID            EQU      0A2H
DEVICE_ID2           EQU      0A1H
READY                EQU      80H
W_ERR_FLAG           EQU      10H
VPP_FLAG             EQU      08H
;        Insert code here to ramp Vpp and disable component /PWD input. If a string of bytes is
;                  to be written at one time, Vpp ramp to 12V and ID check need only occur once,
;                  before the first byte is written
         MOV       AX,        "Address 0 for target 28F008SA-segment"
                                                ; Initialize pointer to 28F0008SA address 0
         MOV       ES,        AX
         MOV       DI,        "Address 0 for target 28F008SA-offset"
         MOV       BYTE PTR ES:[DI],   READ_ID      ; Write Inteligent Identifier command
         CMP       BYTE PTR ES:[DI],   INTEL_ID     ; Does manufacturer ID read correctly?
         JNZ       W_BYT_ID_ERR
         MOV       DI,        "Address 1 for target 28F008SA-offset"
                                                ; Initialize pointer to 28F008SA address 1
         CMP       BYTE PTR ES:[DI],   DEVICE_ID    ; Does device ID read correctly?
         JZ        W_BYT_ID_PASS
         CMP       BYTE PTR ES:[DI],   DEVICE_ID2
         JNZ       W_BYT_ID_ERR

W_BYT_ID_PASS:
         MOV       AX,        "Byte write destination address-segment"
                                                ; Initialize pointer to byte write dest. address
         MOV       ES,        AX
         MOV       DI,        "Byte write destination address-offset"
         MOV       BYTE PTR ES:[DI],   WRITE_SETUP  ; Write byte write setup command
         MOV       AL,        DS:[SI]               ; Load AL with data to write
         MOV       ES:[DI],   AL                    ; Write to device
W_BYT_LOOP:
         TEST      BYTE PTR ES:[DI],   READY        ; Read 28F008SA Status Register
         JZ        W_BYT_LOOP                       ; Loop until bit 7 = 1

         TEST      BYTE PTR ES:[DI],   (W_ERR_FLAG OR VPP_FLAG)
         JZ        W_BYT_CONT                       ; Success!

         TEST      BYTE PTR ES:[DI],   W_ERR_FLAG   ; Check Status Register bit 4
         JNZ       W_BYT_ERR                        ; Jump if = 1, Byte Write Error

         TEST      ES:[DI],   VPP_FLAG              ; Check Status Register bit 3
         JNZ       W_BYT_VPP                        ; Jump if = 1, Vpp Low Error

W_BYT_ID_ERR:
;                  Insert code to service improper device ID read error here.
;                  Is 28F008SA /PWD input disabled? Is Vcc applied to the 28F008SA?
W_BYT_ERR:
;                  Insert code to service byte write error here
W_BYT_VPP:
;                  Insert code to service byte write Vpp low error here
W_BYT_CONT:
;                  Code continues from this point.....
```

This routine writes a byte of data to a single 28F008SA. Note the use of BYTE PTR notation to force x8 accesses. If a string of bytes is to be written at one time, the V_{PP} ramp up, $\overline{PWD}$ disable and device ID checks need only be done before the first byte write attempt. Additionally, when writing multiple bytes at once, examination of bits other than bit 7 (WSM Status) need only occur after the last byte write has completed. The Status Register retains any error bits until the Clear Status Register command is written.

```
;         The following code controls byte write of data to a pair of 28F008SAs (x16 write)
;         DS:[SI] points to the data to be written, ES:[DI] is the location to be written
;         In protected mode operation, DS and ES reference a descriptor
;         Register AX is modified by this procedure
WRITE_SETUP          EQU      40H
READ_ID              EQU      90H
INTEL_ID             EQU      89H
DEVICE_ID            EQU      0A2H
DEVICE_ID2           EQU      0A1H
READY                EQU      80H
W_ERR_FLAG           EQU      10H
VPP_FLAG             EQU      08H
;         Insert code here to ramp Vpp and disable component /PWD inputs. If a string of words is
;                    to be written at one time, Vpp ramp to 12V and ID check need only occur once,
;                    before the first word is written
          MOV        AX,        "Address 0 for target 28F008SA-segment"
                                                   ; Initialize pointer to 28F008SA address 0
          MOV        ES,        AX
          MOV        DI,        "Address 0 for target 28F008SA-offset"
          MOV        ES:[DI],   ((READ_ID SHL 8) OR READ_ID)
                                                   ; Write Inteligent Identifier command
          CMP        ES:[DI],   ((INTEL_ID SHL 8) OR INTEL_ID)
                                                   ; Does manufacturer ID read correctly?
          JNZ        W_WRD_ID_ERR
          MOV        DI,        "Address 1 for target 28F008SA-offset"
                                                   ; Initialize pointer to 28F008SA address 1
          CMP        ES:[DI],   ((DEVICE_ID SHL 8) OR DEVICE_ID)
                                                   ; Does device ID read correctly?
          JZ         W_WRD_ID_PASS
          CMP        ES:[DI],   ((DEVICE_ID2 SHL 8) OR DEVICE_ID2)
          JNZ        W_WRD_ID_ERR

W_WRD_ID_PASS:
          MOV        AX,        "Byte write destination address-segment"
                                                   ; Initialize pointer to byte write dest. address
          MOV        ES,        AX
          MOV        DI,        "Byte write destination address-offset"
          MOV        ES:[DI],   ((WRITE_SETUP SHL 8) OR WRITE_SETUP)     ; Write byte write setup command
          MOV        AX,        DS:[SI]            ; Load AX with data to write
          MOV        ES:[DI],   AX                 ; Write to devices
W_WRD_LOOP:
          TEST       ES:[DI],   ((READY SHL 8) OR READY)  ; Read 28F008SA Status Registers
          JZ         W_WRD_LOOP                    ; Loop until bit 7 = 1

          TEST       ES:[DI],   (((W_ERR_FLAG OR VPP_FLAG) SHL 8) OR (W_ERR_FLAG OR VPP_FLAG))
          JZ         W_WRD_CONT                    ; Success!

          MOV        AX,        ES:[DI]            ; Load Status Register data into AX
          TEST       AL,        W_ERR_FLAG         ; Check Status Register bit 4 (low byte)
          JNZ        W_WRD_ERR                     ; Jump if = 1
          TEST       AH,        W_ERR_FLAG         ; Check Status Register bit 4 (high byte)
          JNZ        W_WRD_ERR                     ; Jump if = 1

          TEST       AL,        VPP_FLAG           ; Check Status Register bit 3 (low byte)
          JNZ        W_WRD_VPP                     ; Jump if = 1
          TEST       AH,        VPP_FLAG           ; Check Status Register bit 3 (high byte)
          JNZ        W_WRD_VPP                     ; Jump if = 1

W_WRD_ID_ERR:
;         Insert code to service improper device ID read error here.
;         Are 28F008SA /PWD inputs disabled? Is Vcc applied to the 28F008SAs?

W_WRD_ERR:
;         Insert code to service byte write error here
W_WRD_VPP:
;         Insert code to service byte write Vpp low error here
W_WRD_CONT:
;         Code continues from this point.....
```

This routine writes a word of data to a pair of 28F008SAs. Note that all constants have been "OR'd" for parallel read/write of two devices at once. If a string of words is to be written at one time, the V_{PP} ramp up, $\overline{PWD}$ disable and device ID checks need only be done before the first word write attempt. Additionally, when writing multiple words at once, examination of bits other than bit 7 (WSM Status) need only occur after the last word write has completed. The Status Register retains any error bits until the Clear Status Register command is written.

```
;         The following code controls block erase of a single 28F008SA (x8 block erase)
;         ES:[DI] points to the block to be erased
;         In protected mode operation, ES references a descriptor
;         Register AX is modified by this procedure
ERASE_SETUP           EQU       20H
ERASE_CONFIRM         EQU       0D0H
READ_ID               EQU       90H
INTEL_ID              EQU       89H
DEVICE_ID             EQU       0A2H
DEVICE_ID2            EQU       0A1H
READY                 EQU       80H
E_ERR_FLAG            EQU       20H
E_CMD_FLAG            EQU       30H
VPP_FLAG              EQU       08H
;         Insert code here to ramp Vpp and disable component /PWD input. If a string of blocks is
;                   to be erased at one time, Vpp ramp to 12V and ID check need only occur once,
;                   before the first block is erased
          MOV       AX,       "Address 0 for target 28F008SA-segment"
                                                  ; Initialize pointer to 28F008SA address 0
          MOV       ES,       AX
          MOV       DI,       "Address 0 for target 28F008SA-offset"
          MOV       BYTE PTR ES:[DI],   READ_ID     ; Write Inteligent Identifier command
          CMP       BYTE PTR ES:[DI],   INTEL_ID    ; Does manufacturer ID read correctly?
          JNZ       E_BYT_ID_ERR
          MOV       DI,       "Address 1 for target 28F008SA-offset"
                                                  ; Initialize pointer to 28F008SA address 1
          CMP       BYTE PTR ES:[DI],   DEVICE_ID   ; Does device ID read correctly?
          JZ        E_BYT_ID_PASS
          CMP       BYTE PTR ES:[DI],   DEVICE_ID2
          JNZ       E_BYT_ID_ERR

E_BYT_ID_PASS:
          MOV       AX,       "Block erase destination address-segment"
                                                  ; Initialize pointer to block erase dest.address
          MOV       ES,       AX
          MOV       DI,       "Block erase destination address-offset"
          MOV       BYTE PTR ES:[DI],   ERASE_SETUP    ; Write block erase setup command
          MOV       BYTE PTR ES:[DI],   ERASE_CONFIRM  ; Write block erase confirm command
E_BYT_LOOP:
          TEST      BYTE PTR ES:[DI],   READY       ; Read 28F0008SA Status Register
          JZ        E_BYT_LOOP                      ; Loop until bit 7 = 1

          TEST      BYTE PTR ES:[DI],   (E_CMD_FLAG OR VPP_FLAG)
          JZ        E_BYT_CONT                      ; Success!

          TEST      BYTE PTR ES:[DI],   E_CMD_FLAG  ; Check Status Register bits 4 and 5
          JNZ       E_BYT_CMD_ERR                   ; Jump if = 1

          TEST      BYTE PTR ES:[DI],   E_ERR_FLAG  ; Check Status Register bit 5
          JNZ       E_BYT_ERR                       ; Jump if = 1

          TEST      BYTE PTR ES:[DI],   VPP_FLAG    ; Check Status Register bit 3
          JNZ       E_BYT_VPP                       ; Jump if = 1

E_BYT_ID_ERR:
;         Insert code to service improper device ID read error here.
;         Is 28F008SA /PWD input disabled? Is Vcc applied to the 28F008SA?
E_BYT_CMD_ERR:
;         Insert code to service block erase command sequence error here
;         (setup followed by a command other than confirm)
E_BYT_ERR:
;         Insert code to service block erase error here
E_BYT_VPP:
;         Insert code to service block erase Vpp low error here
E_BYT_CONT:
;         Code continues from this point.....
```

This routine erases a block of a single 28F00SA. Note the use of BYTE PTR notation to force x8 accesses. If a string of blocks is to be erased at one time, the V_{PP} ramp up, $\overline{PWD}$ disable and device ID checks need only be down before the first block erase attempt. Additionally, when erasing multiple blocks at once, examination of bits other than bit 7 (WSM Status) need only occur after the last block erase has completed. The Status Register retains any error bits until the Clear Status Register command is written.

```
;         The following code controls block erase of a pair of 28F008SAs (x16 block erase)
;         ES:[DI] points to the blocks to be erased
;         In protected mode operation, ES references a descriptor
;         Register AX is modified by this procedure
ERASE_SETUP         EQU       20H
ERASE_CONFIRM       EQU       0D0H
READ_ID             EQU       90H
INTEL_ID            EQU       89H
DEVICE_ID           EQU       0A2H
DEVICE_ID2          EQU       0A1H
READY               EQU       80H
E_ERR_FLAG          EQU       20H
E_CMD_FLAG          EQU       30H
VPP_FLAG            EQU       08H
;         Insert code here to ramp Vpp and disable component /PWD inputs. If a string of blocks is
;                   to be erased at one time, Vpp ramp to 12V and ID check need only occur once,
;                   before the first block pair is erased
          MOV       AX,       "Address 0 for target 28F008SA-segment"
                                                  ; Initialize pointer to 28F008SA address 0
          MOV       ES,       AX
          MOV       DI,       "Address 0 for target 28F008SA-offset"
          MOV       ES:[DI],  ((READ_ID SHL 8) OR READ_ID)
                                                  ; Write Inteligent Identifier command
          CMP       ES:[DI],  ((INTEL_ID SHL 8) OR INTEL_ID)
                                                  ; Does manufacturer ID read correctly?
          JNZ       E_WRD_ID_ERR
          MOV       DI,       "Address 1 for target 28F008SA-offset"
                                                  ; Initialize pointer to 28F008SA address 1
          CMP       ES:[DI],  ((DEVICE_ID SHL 8) OR DEVICE_ID)
                                                  ; Does device ID read correctly?
          JZ        E_WRD_ID_PASS
          CMP       ES:[DI],  ((DEVICE_ID2 SHL 8) OR DEVICE_ID2)
          JNZ       E_WRD_ID_ERR

E_WRD_ID_PASS:
          MOV       AX,       "Block erase destination address-segment"
                                                  ; Initialize pointer to block erase dest. address
          MOV       ES,       AX
          MOV       DI,       "Block erase destination address-offset"
          MOV       ES:[DI],  ((ERASE_SETUP SHL 8) OR ERASE_SETUP)
                                                  ; Write block erase setup command
          MOV       ES:[DI],  ((ERASE_CONFIRM SHL 8) OR ERASE_CONFIRM)
                                                  ; Write block erase confirm command
E_WRD_LOOP:
          TEST      ES:[DI],  ((READY SHL 8) OR READY)  ; Read 28F008SA Status Registers
          JZ        E_WRD_LOOP                    ; Loop until bit 7 = 1

          TEST      ES:[DI],  (((E_CMD_FLAG OR VPP_FLAG) SHL 8) OR (E_CMD_FLAG OR VPP_FLAG))
          JZ        E_WRD_CONT                    ; Success!

          MOV       AX,       ES:[DI]             ; Load Status Register data into AX
          TEST      AL,       E_CMD_FLAG          ; Check Status Reg bits 4 and 5 (low byte)
          JNZ       E_WRD_CMD_ERR                 ; Jump if = 1
          TEST      AH,       E_CMD_FLAG          ; Check Status Register bits 4 and 5 (high byte)
          JNZ       E_WRD_CMD_ERR                 ; Jump if = 1

          TEST      AL,       E_ERR_FLAG          ; Check Status Register bit 5 (low byte)
          JNZ       E_WRD_ERR                     ; Jump if = 1
          TEST      AH,       E_ERR_FLAG          ; Check Status Register bit 5 (high byte)
          JNZ       E_WRD_ERR                     ; Jump if = 1

          TEST      AL,       VPP_FLAG            ; Check Status Register bit 3 (low byte)
          JNZ       E_WRD_VPP                     ; Jump if = 1
          TEST      AH,       VPP_FLAG            ; Check Status Register bit 3 (high byte)
          JNZ       E_WRD_VPP                     ; Jump if = 1

E_WRD_ID_ERR:
;         Insert code to service improper device ID read error here.
;         Are 28F008SA /PWD inputs disabled? Is Vcc applied to the 28F008SAs?
E_WRD_CMD_ERR:
;         Insert code to service block erase command sequence error here
;         (setup followed by a command other than confirm)
E_WRD_ERR:
;         Insert code to service block erase error here
E_WRD_VPP:
;         Insert code to service block erase Vpp low error here
E_WRD_CONT:
;         Code continues from this point.....
```

This routine erases a block pair of two 28F008SAs. Note that all constants have been "OR'd" for parallel read/write of two devices at once. If a string of block pairs is to be erased at one time, the V_{PP} ramp up, $\overline{PWD}$ disable and device ID checks need only be done before the first block pair erase attempt. Additionally, when erasing multiple block pairs at once, examination of bits other than bit 7 (WSM Status) need only occur after the last block pair erase has completed. The Status Register retains any error bits until the Clear Status Register command is written.

3.0 'C' DRIVERS

```
/*************************************************************************************************/
/*        Copyright Intel Corporation, 1992                                                     */
/*        Brian Dipert, Intel Corporation, May 7, 1992, Revision 2.1                             */
/*        The following drivers control the Command and Status Registers of the 28F008SA Flash   */
/*                 Memory to drive byte write, block erase, Status Register read and clear and    */
/*                 array read algorithms. Sample Vpp and /PWD control blocks are also included,   */
/*                 as are example programs combining drivers into full algorithms                 */
/*        The functions listed below are included:                                               */
/*                 erasbgn(): Begins block erasure                                                */
/*                 erassusp(): Suspends block erase to allow reading data from a block of the     */
/*                             28F008SA other than that being erased                              */
/*                 erasres(): Resumes block erase if suspended                                    */
/*                 end(): Polls the Write State Machine to determine if block erase or byte write */
/*                             have completed                                                     */
/*                 eraschk(): Executes full status check after block erase completion             */
/*                 writebgn(): Begins byte write                                                  */
/*                 writechk(): Executes full status check after byte write completion             */
/*                 idread(): Reads and returns the manufacturer and device IDs of the target      */
/*                             28F008SA                                                           */
/*                 statrd(): Reads and returns the contents of the Status Register                */
/*                 statclr(): Clears the Status Register                                          */
/*                 rdmode (): Puts the 28F008SA in Read Array mode                                */
/*                 rdbyte (): Reads and returns a specified byte from the target 28F008SA         */
/*                 vppup(): Enables high voltage Vpph                                             */
/*                 vppdown(): Disables Vpph                                                       */
/*                 pwden(): Enables active low signal /PWD                                        */
/*                 pwddis(): Disables active low signal /PWD                                      */
/*                                                                                                */
/*        Addresses are transferred to functions as pointers to far bytes (ie long integers). An  */
/*                 alternate approach is to create a global array the size of the 28F008SA and    */
/*                 located "over" the 28F008SA in the system memory map. Accessing specific       */
/*                 locations of the 28F008SA is then accomplished by passing the chosen function  */
/*                 an offset from the array base versus a specific address. Different             */
/*                 microprocessor architectures will require different array definitions; ie for  */
/*                 the x86 architecture, define it as "byte eightmeg[16][10000]" and pass each    */
/*                 function TWO offsets to access a specific location. MCS-96 architectures are   */
/*                 limited to "byte eightmeg[10000]"; alternate approaches such as using port pins */
/*                 for paging will be required to access the full flash array                     */
/*                                                                                                */
/*        To create a far pointer, a function such as MK_FP() can be used, given a segment and    */
/*                 offset in the x86 architecture. I use Turbo-C; see your compiler reference     */
/*                 manual for additional information.                                             */
/*************************************************************************************************/

/*************************************************************************************************/
/*        Revision History: Rev 2.1                                                               */
/*                                                                                                */
/*        Changes From Revision 1.0 to Revision 2.0:                                              */
/*                 Added alternate 28F008SA device ID to routine idread()                         */
/*                                                                                                */
/*        Changes from 2.0 to 2.1: Revised the Erase Suspend algorithm to remove potential        */
/*                 "infinite loop" caused by the WSM going "ready" after the system reads the     */
/*                 Status Register, and before the system issues the Erase Suspend command        */
/*************************************************************************************************/

typedef  unsigned char byte;
```

```
/*******************************************************************************************/
/*      Function: Main                                                                     */
/*      Description: The following code shows examples of byte write and block            */
/*               erase algorithms that can be modified to fit the specific application and */
/*               hardware design                                                           */
/*******************************************************************************************/
main()
{
        byte far *address;
        byte data,status;

/*      The following code gives an example of a possible byte write algorithm.            */
/*      Note that Vpp does not need to be cycled between byte writes when a string of byte */
/*      writes occurs. Ramp Vpp to 12V before the first byte write and leave at 12V until after */
/*      completion of the last byte write. Doing so minimizes Vpp ramp up-down delay and   */
/*      maximizes byte write throughput                                                    */
        vppup();
/*      "INSERT SOFTWARE DELAY FOR VPP RAMP IF REQUIRED"                                   */
        pwddis();
        address    = 0XxxxxxL;
        data       = 0Xyy;
        if (writebgn(data,address) == 1)
/*               "RECOVERY CODE-POWER NOT APPLIED (ID CHECK FAIL)"                         */
        else
        {
                 while (end(&status) )
                 switch (writechk(status))
                 {
                           case 0:
                                    break;
                           case 1:
/*                                  "RECOVERY CODE-VPP LOW DETECT ERROR"                   */
                                    break;
                           case 2:
/*                                  "RECOVERY CODE-BYTE WRITE ERROR"                       */
                                    break;
                 }
                 statclr();
        }
        vppdown();
```

This "C" routine gives an example of combining lower-level functions (found in following pages) to complete a byte write. Routines vppup( ) and pwddis( ) enable the 28F008SA for byte write. Function writebgn( ) issues a byte write sequence to the device, end( ) detects byte write completion via Status Register bit 7, and writechk( ) analyzes Status Register bits 3–6 to determine byte write success. If a string of bytes is to be written at one time, V_{PP} ramp up and $\overline{PWD}$ disable need only be done before the first byte write attempt. Additionally, when writing multiple bytes at once, examination of bits other than bit 7 (WSM Ready) need only occur after the last byte write has completed. The Status Register retains any error bits until the Clear Status Register command is written.

```
/*      The following code gives an example of a possible block erase algorithm.              */
/*      Note that Vpp does not need to be cycled between block erases when a string of block   */
/*      erases occurs. Ramp Vpp to 12V before the first block erase and leave at 12V until after */
/*      completion of the last block erase. Doing so minimizes Vpp ramp up-down delay and      */
/*      maximizes block erase throughput                                                       */
        vppup();
/*      "INSERT SOFTWARE DELAY FOR VPP RAMP IF REQUIRED"                                       */
        pwddis();
        address  = 0XxxxxxL;
        if (erasbgn(address) == 1)
/*              "RECOVERY CODE-POWER NOT APPLIED (ID CHECK FAIL)"                              */
        else
        {
                while (end(&status) )
                        ;
                switch (eraschk(status))
                {
                        case 0:
                                break;
                        case 1:
/*                              "RECOVERY CODE-VPP LOW DETECT ERROR"                           */
                                break;
                        case 2:
/*                              "RECOVERY CODE-BLOCK ERASE ERROR"                              */
                                break;
                        case 3:
/*                              "RECOVERY CODE-ERASE SEQUENCE ERROR"                           */
                                break;
                }
                statclr();
        }
        vppdown();
}
```

This "C" routine gives an example of combining lower-level functions (found in following pages) to complete a block erase. Routines vppup( ) and pwddis( ) enable the 28F008SA for block erase. Function erasbgn( ) issues a block erase sequence to the device, end( ) detects block erase completion via Status Register bit 7, and eraschk( ) analyzes Status Register bits 3–6 to determine block erase success. If a string of blocks is to be erased at one time, V_{PP} ramp up and $\overline{PWD}$ disable need only be done before the first block erase attempt. Additionally, when erasing multiple blocks at once, examination of bits other than bit 7 (WSM Ready) need only occur after the last block erase has completed. The Status Register retains any error bits until the Clear Status Register command is written.

```
/****************************************************************************************************/
/*      Function: Erasgbn                                                                           */
/*      Description: Begins erase of a block.                                                       */
/*      Inputs:    blckaddr: System address within the block to be erased                          */
/*      Outputs:   None                                                                             */
/*      Returns:   0 = Block erase successfully initiated                                           */
/*                 1 = Block erase not initiated (ID check error)                                   */
/*      Device Read Mode on Return: Status Register (ID if returns 1)                               */
/****************************************************************************************************/

#define    ERASETUP  0X20                  /* Erase Setup command                                    */
#define    ERASCONF  0XD0                  /* Erase Confirm command                                  */

int erasbgn(blckaddr)

byte far *blckaddr;                        /* blckaddr is an address within the block to be erased   */

{
        byte mfgrid,deviceid;

        if (idread(&mfgrid,&deviceid)==1)  /* ID read error; device not powered up?                  */
                return (1);
        *blckaddr = ERASETUP;              /* Write Erase Setup command to block address             */
        *blckaddr = ERASCONF;              /* Write Erase Confirm command to block address           */
        return (0);
}
```

Routine erasbgn( ) issues a block erase command sequence to a 28F008SA. It is passed the desired system address for the block to be erased. After calling idread( ), it writes the erase command sequence at the specified address. It returns "0" if block erase initiation was successful, and "1" if the ID read fails (device not powered up or $\overline{\text{PWD}}$ not disabled).

```
/*************************************************************************************************/
/*      Function: Erassusp                                                                      */
/*      Description: Suspends block erase to read from another block                            */
/*      Inputs:   None                                                                          */
/*      Outputs:  None                                                                          */
/*      Returns:  0 = Block erase suspended                                                     */
/*                1 = Error; Write State Machine not busy (block erase suspend not possible)    */
/*      Device Read Mode on Return: Status Register                                             */
/*************************************************************************************************/

#define   RDYMASK   0X80              /* Mask to isolate the WSM Status bit of the Status Register */
#define   WSMRDY    0X80              /* Status Register value after masking, signifying that      */
                                      /*          the WSM is no longer busy                        */
#define   SUSPMASK  0X40              /* Mask to isolate the erase suspend status bit of the       */
                                      /*          Status Register                                  */
#define   ESUSPYES  0X40              /* Status Register value after masking, signifying that      */
                                      /*          block erase has been suspended                   */
#define   STATREAD  0X70              /* Read Status Register command                              */
#define   SYSADDR   0                 /* This constant can be initialized to any address within    */
                                      /*          the memory map of the target 28F008SA and is     */
                                      /*          alterable depending on the system architecture   */
#define   SUSPCMD   0XB0              /* Erase Suspend command                                     */

int erassusp()

{
          byte far *stataddr;         /* Pointer variable used to write commands to device         */

          stataddr  = (byte far *)SYSADDR;
          *stataddr = SUSPCMD;        /* Write Erase Suspend command to the device                 */
          *stataddr = STATREAD;       /* Write Read Status Register command..necessary in case     */
                                      /*          erase is already completed                       */
          while ((*stataddr & RDYMASK) != WSMRDY)
                    ;                 /* Will remain in while loop until bit 7 of the Status       */
                                      /*          Register goes to 1, signifying that              */
                                      /*          the WSM is no longer busy                        */
          if ((*stataddr & SUSPMASK) = ESUSPYES)
                    return (0);       /* Erase is suspended..return code "0"                       */
          return (1);                 /*Erase has already completed; suspend not possible.         */
                                      /*          Error code "1"                                   */
}
```

Routine erassusp( ) issues the erase suspend command to a 28F008SA. It first makes sure the WSM is truly busy, then issues the erase suspend command and polls Status Register bits 7 and 6 until they indicate erase suspension. It returns "0" if block erase was successful, and "1" if the WSM was not busy when suspend was attempted.

```
/*****************************************************************************************************/
/*      Function: Erasres                                                                            */
/*      Description: Resumes block erase previously suspended                                        */
/*      Inputs:   None                                                                               */
/*      Outputs:  None                                                                               */
/*      Returns:  0 = Block erase resumed                                                            */
/*                1 = Error; Block erase not suspended when function called                          */
/*      Device Read Mode on Return: Status Register                                                  */
/*****************************************************************************************************/

#define   RDYMASK   0X80             /* Mask to isolate the WSM Status bit of the Status Register */
#define   WSMRDY    0X80             /* Status Register value after masking, signifying that the  */
                                     /*       WSM is no longer busy                               */
#define   SUSPMASK  0X40             /* Mask to isolate the erase suspend status bit of the       */
                                     /*       Status Register                                     */
#define   ESUSPYES  0X40             /* Status Register value after masking, signifying that      */
                                     /*       block erase has been suspended                      */
#define   STATREAD  0X70             /* Read Status Register command                              */
#define   SYSADDR   0                /* This constant can be initialized to any address within    */
                                     /*       the memory map of the target 28F008SA and is        */
                                     /*       alterable depending on the system architecture      */
#define   RESUMCMD  0XD0             /* Erase Resume command                                      */

int erasres()

{
          byte far *stataddr;        /* Pointer variable used to write commands to device         */

          stataddr  = (byte far *)SYSADDR;
          *stataddr = STATREAD;      /* Write Read Status Register command to 28F008SA            */
          if ((*stataddr & SUSPMASK) != ESUSPYES)
                    return (1);      /* Block erase not suspended. Error code "1"                 */
          *stataddr = RESUMCMD;      /* Write Erase Resume command to the device                  */
          while ((*stataddr & SUSPMASK) == ESUSPYES)
                    ;                /* Will remain in while loop until bit 6 of the Status       */
                                     /*       Register goes to 0, signifying block                */
                                     /*       erase resumption                                    */
          while ((*stataddr & RDYMASK) == WSMRDY)
                    ;                /* Will remain in while loop until bit 7 of the Status       */
                                     /*       Register goes to 0, signifying that the WSM is      */
                                     /*       once again busy                                     */
          return (0);
}
```

Routine erasres( ) issues the erase resume command to a 28F008SA. It first makes sure the WSM is truly suspended, then issues the erase resume command and polls Status Register bits 7 and 6 until the indicate WSM resumption. It returns "0" if block erase resume was successful, and "1" if the WSM was not suspended when resumption was attempted.

3

```
/*****************************************************************************************************/
/*          Function: End                                                                            */
/*          Description: Checks to see if the WSM is busy (is byte write/block erase completed?)     */
/*          Inputs:               None                                                               */
/*          Outputs:              statdata: Status Register data read from device                    */
/*          Returns:              0 = Byte Write/Block Erase completed                               */
/*                                1 = Byte Write/Block Erase still in progress                       */
/*          Device Read Mode on Return: Status Register                                              */
/*****************************************************************************************************/
#define     RDYMASK        0X80                 /* Mask to isolate the WSM Status bit of the Status   */
#define     WSMRDY         0X80                 /* Register value after masking, signifying that the  */
                                                /*          WSM is no longer busy                     */
#define     STATREAD       0X70                 /* Read Status Register command                       */
#define     SYSADDR        0                    /* This constant can be initialized to any address within*/
                                                /*          the memory map of the target 28F008SA and is  */
                                                /*          alterable depending on the system architecture */

int end(statdata)

byte *statdata;                                 /* Allows Status Register data to be passed back to the  */
                                                /*          main program for further analysis             */

{
            byte far *stataddr;                 /* Pointer variable used to write commands to device      */

            stataddr       = (byte far *)SYSADDR;
            *stataddr      = STATREAD;          /* Write Read Status Register command to 28F008SA         */
            if (((*statdata = *stataddr) & RDYMASK) != WSMRDY)
                           return (1);          /* Byte write/block erasure still in progress...code "1" */
            return (0);                         /* Byte write/block erase attempt completed...code "0"   */
}
```

Routine end( ) detects completion of byte write or block erase operations of a 28F008SA. It passes back the Status Register data it reads from the device. It also returns "0" if Status Register bit 7 indicates WSM "Ready", and "1" if indication is that the WSM is still "Busy".

```
/****************************************************************************************************/
/*      Function: Eraschk                                                                          */
/*      Description: Completes full Status Register check for block erase (proper command          */
/*              sequence, Vpp low detect, block erase success). This routine assumes that block    */
/*              erase completion has already been checked in function end(), and therefore does    */
/*              not check the WSM Status bit of the Status Register                                */
/*      Inputs:   statdata: Status Register data read in function end                              */
/*      Outputs:  None                                                                             */
/*      Returns:  0 = Block erase completed successfully                                           */
/*                1 = Error; Vpp low detect                                                        */
/*                2 = Error; Block erase error                                                     */
/*                3 = Error; Improper command sequencing                                           */
/*      Device Read Mode on Return: Same as when entered                                           */
/****************************************************************************************************/

#define  ESEQMASK  0X30           /* Mask to isolate the erase and byte write status bits of  */
                                  /*       the Status Register                                */
#define  ESEQFAIL  0X30           /* Status Register value after masking if block erase       */
                                  /*       command sequence error has been detected           */
#define  EERRMSK   0X20           /* Mask to isolate the erase status bit of the              */
                                  /*       Status Register                                    */
#define  ERASERR   0X20           /* Status Register value after masking if block erase error */
                                  /*       has been detected                                  */
#define  VLOWMASK  0X08           /* Mask to isolate the Vpp status bit of the Status Register */
#define  VPPLOW    0X08           /* Status Register value after masking if Vpp low           */
                                  /*       has been detected                                  */

int eraschk(statdata)

byte statdata;                    /* Status Register data that has been already read from the */
                                  /*       28F008SA in function end()                         */

{
        if ((statdata & VLOWMASK) == VPPLOW)
                return (1);     /* Vpp low detect error, return code "1"                    */
        if ((statdata & EERRMSK) == ERASERR)
                return (2);     /* Block erase error detect, return code "2"                */
        if ((statdata & ESEQMASK) == ESEQFAIL)
                return (3);     /* Block erase command sequence error, return code "3"      */
        return (0);             /* Block erase success, return code "0"                     */
}
```

Routine eraschk( ) takes the Status Register data read in end( ) and further analyzes it. It returns "0" if block erase was successful, "1" if V_{PP} low error was detected, "2" if block erase error was reported and "3" if an erase command sequence error was found (erase setup followed by a command other than erase confirm). This is useful after a block or string of blocks has been erased, to check for successful completion.

```
/*****************************************************************************************************/
/*        Function: Writebgn                                                                         */
/*        Description: Begins byte write sequence                                                    */
/*        Inputs:   wdata: Data to be written into the device                                        */
/*                  waddr: Target address to be written                                              */
/*        Outputs:  None                                                                             */
/*        Returns:  0 = Byte write successfully initiated                                            */
/*                  1 = Byte write not initiated (ID check error)                                    */
/*        Device Read Mode on Return: Status Register (ID if returns 1)                              */
/*****************************************************************************************************/

#define   SETUPCMD  0X40              /* Byte Write Setup command                                    */

int writebgn(wdata,waddr)

byte wdata;                           /* Data to be written into the 28F008SA                        */
byte far *waddr;                      /* waddr is the destination address for the data               */
                                      /*        to be written                                        */

{
          byte mfgrid,deviceid;

          if (idread(&mfgrid,&deviceid)==1)  /* Device ID read error...powered up?                   */
                    return (1);
          *waddr    = SETUPCMD;       /* Write Byte Write Setup command and destination address      */
          *waddr    = wdata;          /* Write byte write data and destination address               */
          return (0);
}

/*****************************************************************************************************/
/*        Function: Writechk                                                                         */
/*        Description: Completes full Status Register check for byte write (Vpp low detect, byte     */
/*                  write success). This routine assumes that byte write completion has already      */
/*                  been checked in function end() and therefore does not check the WSM Status       */
/*                  bit of the Status Register                                                       */
/*        Inputs:   statdata: Status Register data read in function end()                            */
/*        Outputs:  None                                                                             */
/*        Returns:  0 = Byte write completed successfully                                            */
/*                  1 = Error; Vpp low detect                                                        */
/*                  2 = Error; Byte write error                                                      */
/*        Device Read Mode on Return: Status Register                                                */
/*****************************************************************************************************/

#define   WERRMSK   0X10              /* Mask to isolate the byte write error bit of the             */
                                      /*        Status Register                                      */
#define   WRITERR   0X10              /* Status Register value after masking if byte write error     */
                                      /*        has been detected                                    */
#define   VLOWMASK  0X08              /* Mask to isolate the Vpp status bit of the                   */
                                      /*        Status Register                                      */
#define   VPPLOW    0X08              /* Status Register value after masking if Vpp low              */
                                      /*        has been detected                                    */

int writechk(statdata)

byte statdata;                        /* Status Register data that has been already read from the    */
                                      /*        28F008SA in function end()                           */
{
          if ((statdata & VLOWMASK) == VPPLOW)
                    return (1);       /* Vpp low detect error, return code "1"                       */
          if ((statdata & WERRMSK) == WRITERR)
                    return (2);       /* Byte write error detect, return code "2"                    */
          return (0);                 /* Byte/string write success, return code "0"                  */
}
```

Routine writebgn( ) issues a byte write command sequence to a 28F008SA. It is passed the desired system address for the byte to be written, as well as the data to be written there. After calling idread( ), it writes the byte write command sequence at the specified address. It returns "0" if byte write initiation was successful, and "1" if the ID read fails (device not powered up or $\overline{\text{PWD}}$ not disabled).

Routine writechk( ) takes the Status Register data read in end( ) and further analyzes it. It returns "0" if byte write was successful, "1" if V_{PP} low error was detected, and "2" if byte write error was reported. This is useful after a byte or string of bytes has been written, to check for successful completion.

```
/*****************************************************************************************************/
/*        Function: Idread                                                                          */
/*        Description: Reads the manufacturer and device IDs from the target 28F008SA               */
/*        Inputs:   None                                                                            */
/*        Outputs:  mfgrid: Returned manufacturer ID                                                */
/*                  deviceid: Returned device ID                                                    */
/*        Returns:  0 = ID read correct                                                             */
/*                  1 = Wrong or no ID                                                              */
/*        Device Read Mode on Return: inteligent Identifier                                         */
/*****************************************************************************************************/

#define   MFGRADDR  0                 /* Address "0" for the target 28F008SA...alterable depending */
                                      /*        on the system architecture                         */
#define   DEVICADD  1                 /* Address "1" for the target 28F008SA...alterable depending */
                                      /*        on the system architecture                         */
#define   IDRDCOMM  0X90              /* Inteligent Identifier command                             */
#define   INTELID   0X89              /* Manufacturer ID for Intel devices                         */
#define   DVCID     0X0A2             /* Device IDs for 28F008SA                                   */
#define   DVCID2    0X0A1

int idread(mfgrid,deviceid)

byte *mfgrid;                         /* The manufacturer ID read by this function, to be          */
                                      /*        transferred back to the calling program            */
byte *deviceid;                       /* The device ID read by this function, to be transferred    */
                                      /*        back to the calling function                       */
{
          byte far *tempaddr;         /* Pointer address variable used to read IDs                 */
          tempaddr  = (byte far *)MFGRADDR;
          *tempaddr = IDRDCOMM;       /* Write inteligent Identifier command to an address within  */
                                      /*        the 28F008SA memory map (in this case 00 hex)      */
          *mfgrid   = *tempaddr;      /* Read mfgr ID, tempaddr still points at address "0"        */
          tempaddr  = (byte far *)DEVICADD;  /* Point to address "1" for the device specific ID    */
          *deviceid = *tempaddr;      /* Read device ID                                            */
          if ((*mfgrid != INTELID) || ((*deviceid != DVCID) && (*deviceid != DVCID2)))
                    return (1);       /* ID read error; device powered up?                         */
          return (0);
}
```

Routine idread( ) issues the Intelligent Identifier command to a 28F008SA. It passes back the manufacturer and device IDs it reads. In addition, it returns "0" if the IDs read matched those expected for the 28F008SA or 28F008SA-L, and "1" if either the manufacturer or device IDs did not match.

```
/*********************************************************************************************************/
/*        Function: Statrd                                                                               */
/*        Description: Returns contents of the target 28F008SA Status Register                           */
/*        Inputs:   None                                                                                 */
/*        Outputs:  statdata: Returned Status Register data                                              */
/*        Returns:  Nothing                                                                              */
/*        Device Read Mode on Return: Status Register                                                    */
/*********************************************************************************************************/
#define   STATREAD  0X70                  /* Read Status Register command                              */
#define   SYSADDR   0                     /* This constant can be initialized to any address within   */
                                          /*        the memory map of the target 28F008SA and is       */
                                          /*        alterable depending on the system architecture     */

int statrd(statdata)

byte *statdata;                           /* Allows Status Register data to be passed back to the     */
                                          /*        calling program for further analysis               */
{
        byte far *stataddr;               /* Pointer variable used to write commands to device         */

        stataddr  = (byte far *)SYSADDR;
        *stataddr = STATREAD;             /* Write Read Status Register command to 28F008SA            */
        *statdata = *stataddr;
        return;
}

/*********************************************************************************************************/
/*        Function: Statclr                                                                              */
/*        Description: Clears the 28F008SA Status Register                                               */
/*        Inputs:   None                                                                                 */
/*        Outputs:  None                                                                                 */
/*        Returns:  Nothing                                                                              */
/*        Device Read Mode on Return: Array                                                              */
/*********************************************************************************************************/

#define   STATCLER  0X50                  /* Clear Status Register command                             */
#define   SYSADDR   0                     /* This constant can be initialized to any address within   */
                                          /*        the memory map of the target 28F008SA and is       */
                                          /*        alterable depending on the system architecture     */

int statclr()

{
        byte far *stataddr;               /* Pointer variable used to write commands to device         */

        stataddr  = (byte far *)SYSADDR;
        *stataddr = STATCLER;             /* Write Clear Status Register command to 28F008SA           */
        return;
}
```

Routine statrd( ) reads a 28F008SA Status Register. It issues the Read Status Register command and passes back the data it obtains.

Routine statclr( ) issues the Clear Status Register command to a 28F008SA. This routine is required after analyzing Status Register contents in routines like eraschk( ) and writechk( ). The 28F008SA Status Register retains state of bits 3–6 until they are cleared by the Clear Status Register command.

```
/***********************************************************************************************/
/*       Function: Rdmode                                                                      */
/*       Description: Puts the target 28F008SA in Read Array Mode. This function might be used, for */
/*                example, to prepare the system for return to code execution out of the Flash  */
/*                memory after byte write or block erase algorithms have been executed off-chip */
/*       Inputs:  None                                                                         */
/*       Outputs: None                                                                         */
/*       Returns: Nothing                                                                      */
/*       Device Read Mode on Return: Array                                                     */
/***********************************************************************************************/

#define  RDARRAY  OXFF             /* Read Array command                                     */
#define  SYSADDR  0                /* This constant can be initialized to any address within */
                                   /*       the memory map of the target 28F008SA and is     */
                                   /*       alterable depending on the system architecture   */

int rdmode()

{
        byte far *tempaddr;        /* Pointer variable used to write commands to the device  */

        tempaddr  = (byte far *)SYSADDR;
        *tempaddr = RDARRAY;       /* Write Read Array command to 28F008SA                   */
        return;
}

/***********************************************************************************************/
/*       Function: Rdbyte                                                                      */
/*       Description: Reads a byte of data from a specified address and returns it to the      */
/*                  calling program                                                            */
/*       Inputs:  raddr: Target address to be read from                                       */
/*       Outputs: rdata: Data at the specified address                                         */
/*       Returns: Nothing                                                                      */
/*       Device Read Mode on Return: Array                                                     */
/***********************************************************************************************/

#define  RDARRAY  OXFF             /* Read array command                                     */

int rdbyte(rdata,raddr)
byte *rdata;                       /* Returns data read from the device at specified address */
byte far *raddr;                   /* Raddr is the target address to be read from            */

{
        *raddr   = RDARRAY;        /* Write read array command to an address within the      */
                                   /*       28F008SA (in this case the target address)       */
        *rdata   = *raddr;         /* Read from the specified address and store              */
        return;
}
```

Routine rdmode( ) simply puts a 28F008SA in Read Array mode. This is useful after byte write and block erase operations, to return the 28F008SA to its "normal" mode of operation. After block erase or byte write, the 28F008SA will continue to output Status Register data until the Read Array command is issued to it, for example.

Routine rdbyte( ) not only puts the 28F008SA in Read Array mode, it also reads a byte of data. It is passed the desired system byte address, and passes back the data at that address.

```
/*********************************************************************************************/
/*       Function: Vppup                                                                     */
/*       Description: Ramps the Vpp supply to the target 28F008SA to enable byte write or block     */
/*                erase. This routine can be tailored to the individual system architecture. For */
/*                purposes of this example, I assumed that a system Control Register existed at  */
/*                system address 20000 hex, with the following definitions:                      */
/*                                                                                               */
/*                          Bit 7: Vpph Control: 1 = Enabled                                     */
/*                                               0 = Disabled                                    */
/*                          Bit 6: PWD Control:  1 = PowerDown Enabled                           */
/*                                               0 = PowerDown Disabled                          */
/*                          Bits 5-0: Undefined                                                  */
/*                                                                                               */
/*       Inputs:   None                                                                          */
/*       Outputs:  None                                                                          */
/*       Returns:  Nothing                                                                       */
/*       Device Read Mode on Return: As existed before entering the function. Part is now ready for */
/*                program or erase                                                               */
/*********************************************************************************************/

#define  VPPHIGH   0X80             /* Bit 7 = 1, Vpp elevated to Vpph                          */
#define  SYSCADDR  0X20000          /* Assumed system Control Register Address                  */

int vppup()

{
         byte far *contaddr;        /* Pointer variable used to write data to the system        */
                                    /*        Control Register                                   */

         contaddr  = (byte far *)SYSCADDR;
         *contaddr = *contaddr | VPPHIGH;  /* Read current Control Register data, "OR" with     */
                                    /*        constant to ramp Vpp                               */
         return;
}

/*********************************************************************************************/
/*       Function: Vppdown                                                                   */
/*       Description: Ramps down the Vpp supply to the target 28F008SA to disable byte write/block  */
/*                erase. See above for a description of the assumed system Control Register.     */
/*       Inputs:   None                                                                          */
/*       Outputs:  None                                                                          */
/*       Returns:  Nothing                                                                       */
/*       Device Read Mode on Return: As existed before entering the function. Part now has high Vpp */
/*                disabled. If byte write or block erase was in progress when this function was  */
/*                called, it will complete unsuccessfully with Vpp low error in the              */
/*                Status Register.                                                               */
/*********************************************************************************************/

#define  VPPDWN    0X7F             /* Bit 7 = 0, Vpp lowered to Vppl                           */
#define  SYSCADDR  0X20000          /* Assumed system Control Register Address                  */

int vppdown()

{
         byte far *contaddr;        /* Pointer variable used to write data to the system        */
                                    /*        Control Register                                   */

         contaddr  = (byte far *)SYSCADDR;
         *contaddr = *contaddr & VPPDWN;  /* Read current Control Register data, "AND" with     */
                                    /*        constant to lower Vpp                              */
         return;
}
```

Functions vppup( ) and vppdown( ) give examples of how to control via software the hardware that enables or disables 12V V_{PP} to a 28F008SA. The actual hardware implementation chosen will drive any modification of these routines.

```
/*****************************************************************************************************/
/*        Function: Pwden                                                                            */
/*        Description: Toggles the 28F008SA /PWD pin low to put the device in Deep PowerDown mode.   */
/*                  See above for a description of the assumed system Control Register.              */
/*        Inputs:   None                                                                             */
/*        Outputs:  None                                                                             */
/*        Returns:  Nothing                                                                          */
/*        Device Read Mode on Return: The part is powered down. If byte write or block erase was in  */
/*                  progress when this function was called, it will abort with resulting partially   */
/*                  written or erased data. Recovery in the form of repeat of byte write or block    */
/*                  erase will be required once the part transitions out of powerdown, to            */
/*                  initialize data to a known state.                                                */
/*****************************************************************************************************/

#define    PWD       0X40               /* Bit 6 = 1, /PWD enabled                                   */
#define    SYSCADDR  0X20000            /* Assumed system Control Register Address                   */

int pwden()

{
        byte far *contaddr;             /* Pointer variable used to write data to the system
                                        /*        Control Register                                   */

        contaddr  = (byte far *)SYSCADDR;
        *contaddr = *contaddr | PWD;    /* Read current Control Register data, "OR" with constant    */
                                        /*        to enable Deep PowerDown                           */
        return;
}

/*****************************************************************************************************/
/*        Function: Pwddis                                                                           */
/*        Description: Toggles the 28F008SA /PWD pin high to transition the part out of Deep         */
/*                  PowerDown. See above for a description of the assumed system Control Register.   */
/*        Inputs:   None                                                                             */
/*        Outputs:  None                                                                             */
/*        Returns:  Nothing                                                                          */
/*        Device Read Mode on Return: Read Array mode. Low voltage is removed from the /PWD pin.     */
/*                  28F008SA output pins will output valid data time tPHQV after the /PWD pin        */
/*                  transitions high (reference the datasheet AC Read Characteristics) assuming      */
/*                  valid states on all other control and power supply pins.                         */
/*****************************************************************************************************/

#define    PWDOFF    0XBF               /* Bit 6 = 0, /PWD disabled                                  */
#define    SYSCADDR  0X20000            /* Assumed system Control Register Address                   */

int pwddis()

{
        byte far *contaddr;             /* Pointer variable used to write data to the system         */
                                        /*        Control Register                                   */

        contaddr  = (byte far *)SYSCADDR;
        *contaddr = *contaddr & PWDOFF;    /* Read current Control Register data, "AND" with         */
                                           /*        constant to disable Deep PowerDown              */
        return;
}
```

Functions pwden( ) and pwddis( ) give examples of how to control via software the hardware that enables or disables a 28F008SA $\overline{\text{PWD}}$ input. The actual hardware implementation chosen will drive any modification of these routines.

ADDITIONAL INFORMATION

		Order Number
	28F008SA Datasheet	290429
	28F008SA-L Datasheet	290435
AP-359	"28F008SA Hardware Interfacing"	292094
AP-364	"28F008SA Automation and Algorithms"	292099
ER-27	"The Intel 28F008SA Flash Memory"	294011
ER-28	"ETOX™-III Flash Memory Technology"	294012

REVISION HISTORY

Number	Description
002	Revised Erase Suspend Algorithm in "C" Drivers.

APPLICATION NOTE

AP-363

September 1992

3

Extended Flash Bios Design For Portable Computers

SALIM FEDEL
SENIOR APPLICATIONS ENGINEER

Order Number: 292098-002

EXTENDED FLASH BIOS DESIGN FOR PORTABLE COMPUTERS

CONTENTS PAGE

1.0 INTRODUCTION 3-467

2.0 PC BIOS TODAY AT THE 128 KBYTE CODE SIZE LIMIT 3-467

3.0 WHY BIOS CODE WILL GROW BEYOND 128 KBYTES 3-467

3.1 Advanced Power Management .. 3-469

3.2 Optimize New Portable Applications 3-469

3.3 Putting Microsoft* MS-DOS 5.0 Operating System ROM Version into the BIOS Chip 3-469

3.4 Relocated Resident VGA Code: VIDEO BIOS 3-469

4.0 HARDWARE DESIGN FOR A 256 KBYTE BIOS 3-470

4.1 Intel 28F200BX/002BX Boot Block Flash Memory Family 3-470

4.2 Intel386SL™ Microprocessor Superset Platform Overview 3-471

4.3 Programming the Intel386SL™ Registers 3-476

4.4 The ISA Sliding Window 3-477

4.5 The 28F200BX-T/28F002BX-T in the 1 MByte DOS Memory Map 3-478

CONTENTS PAGE

5.0 SOFTWARE DESIGN CONSIDERATIONS 3-478

5.1 16 KBytes Recovery Code 3-480

5.2 28F200BX Reprogramming 3-480

5.3 Power Management 3-480

6.0 DESIGNING A 3.3V SYSTEM 3-481

6.1 Low Voltage Chips 3-481

6.2 Power Savings and Improved Battery Life 3-481

7.0 CONCLUSIONS 3-481

7.1 Benefits of Extended Flash BIOS 3-481

REFERENCES 3-482

APPENDIX A 3-483

APPENDIX B 3-484

APPENDIX C 3-489

APPENDIX D 3-493

*Microsoft is a registered trademark of Microsoft Corporation.

1.0 INTRODUCTION

PC BIOS has been migrating to Flash-based designs with the introduction of highly optimized Flash memory architectures. The first phase of this shift in paradigm was from ROM/EPROM-based BIOS to Bulk Erase Flash memory-based BIOS to provide for in-system updatable BIOS and hence an easy update capability when BIOS changes are required.

The second phase improved the basic flash design to migrate towards boot block flash memory architecture with the Intel 28F001BX Flash memory. This improvement enabled the implementation of additional features and provided a true design capability for portable PC BIOS.

The third phase of this paradigm shift now starting to evolve, deals with the need to grow beyond the traditional BIOS space limit of 128 KBytes imposed by the original PC architecture to accommodate the advanced features of today's portable and desktop systems.

This application note describes in detail this third phase in BIOS hardware and software implementation. Specifically it will investigate why BIOS needs to grow beyond the 128 KByte code size. Then, a design example using the Intel 28F200BX Boot Block Flash memory in an Intel386SL™-based portable system will be explained in terms of both hardware and software viewpoints. Finally, low voltage PC BIOS designs incorporating 3.3V components are described.

2.0 PC BIOS TODAY AT THE 128 KBYTE CODE SIZE LIMIT

The Basic Input/Output System (BIOS) code is the lowest system level software which manages the interaction between all hardware components (CPU, Chip-Sets and I/O) with all software modules (Operating Systems and Applications Code). BIOS manages many functions in a PC, such as Power-On Self Test (POST), input vector creation, I/O services and system initialization. Therefore BIOS is the essential interface layer for full system functionality and compatibility.

The original PC architecture, developed in 1981, put restrictions on the size and mapping of the BIOS code which was then a very simple piece of software (on the order of 32 KBytes for the original BIOS code). It was located at the top of the PC's (8088) memory map which at the time was a maximum of 1 MByte.

Then in 1984, the PC AT (80286) BIOS was expanded another 32 KBytes for a total of 64 KBytes. Subsequently, towards the late 1980s, more elaborate BIOS set-up utilities started to be an integral part of the BIOS code. In addition, personal computer manufacturers designed custom features into their BIOS code to offer more system flexibility. This increase in code complexity expanded the AT BIOS code another 64 KBytes (for a total of 128 KBytes) to occupy the total BIOS reserved space in the 1 MByte memory map.

In the DOS memory map, BIOS is mapped down from the top of the 1 MByte address space (F0000H to FFFFFH). Additional BIOS code space is available for future enhancements from E0000H to EFFFFH. The next 256 KBytes in the DOS memory map are reserved for adapter space to accommodate add-in boards (for enhanced graphics cards for instance). Finally the remaining 640 KBytes are reserved for the user to load his/her applications for execution. See Figure 1 for a graphical description.

3.0 WHY BIOS CODE WILL GROW BEYOND 128 KBYTES

As advances in computer design affect both desktop systems (with the addition of EISA and SCSI capabilities) and portable systems (with the addition of advanced power management capability and I/O cards), the need for larger amounts of non-volatile memory space becomes evident.

A Notebook or a Palmtop computer design, for instance, may put the operating system, the system management code, set-up or utility programs into the non-volatile memory area to conserve precious RAM space for applications.

Additionally, Video BIOS can also be mapped into the Flash BIOS area as in the case of the Intel386SL Microprocessor Superset.

Therefore, to implement advanced capabilities and provide new features (as described above) into powerful mobile computers, the 128 KByte BIOS code size limit had to be removed as it shall be explored in the next section.

The BIOS of today and the future must adapt to the new requirements of portable PC designs and take advantage of the new capabilities of low power PC chipsets and I/O devices to achieve the highest performance and longest battery life at the lowest system cost.

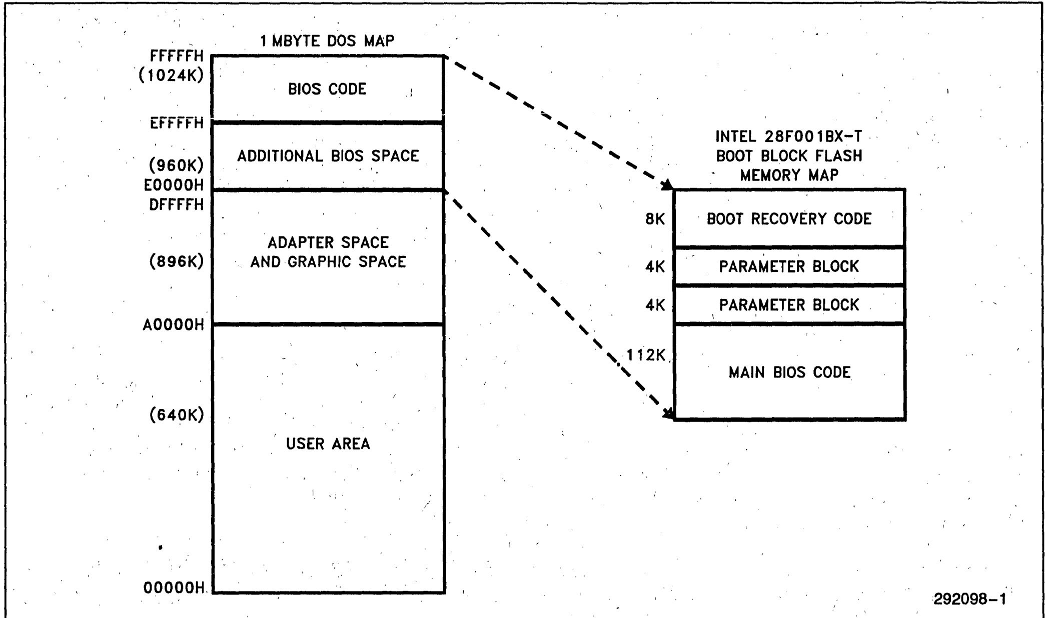


Figure 1. 128 KByte BIOS Code Segmentation in 1 MByte DOS Memory Map

3.1 Advanced Power Management

High integration CPUs and chip-sets such as the Intel386SL Microprocessor SuperSet allow for the design of light, small form factor portable computers with long battery life.

BIOS is the ideal place for implementing the new power management techniques available with the Intel386SL Microprocessor Superset. BIOS software vendors have implemented APM code for the latest generation of Notebook PCs.

This added level of functionality imposed on the BIOS code increases the need for larger code space beyond the traditional 128 KByte BIOS implementations seen in today's portable systems.

APM code typically requires an additional 32 KByte of code space beyond the basic 64 KByte standard BIOS. Therefore, with the addition of APM, BIOS code grows to 96 KBytes.

3.2 Optimize New Portable Applications

With the implementation of I/O card capabilities for non-volatile file storage (with flash memory cards) and the ability to communicate over a telephone line (with modem cards), mobile computers have truly become powerful tools on the road.

The establishment of a common PC card standard for full system compatibility is described in the PCMCIA Standard release 2.0 (Personal Computer Memory Card International Association). Intel has developed a similar set of specifications fully compatible with the PCMCIA release 2.0 standard called the Exchangeable Card Architecture (ExCA). In order to implement ExCA capability in a portable computer, additional BIOS code called Socket Services is required to manage the system I/O card functionality. To implement the ExCA specifications, socket services needs an additional 16 KByte of code space.

Furthermore, in the pen-based PC applications, there are even greater BIOS requirements to design-in unique features such as: pen extensions, touchscreen capability and character recognition interface code. These new features require the implementation of additional BIOS code (may be 16 KBytes to 32 KBytes).

To take full advantage of Desktop system capabilities and performance while still having a portable computer to take on the road, docking station designs were conceived. This added level of complexity for the portable computer increases the code required in a basic system BIOS.

3.3 Putting Microsoft MS-DOS 5.0 Operating System ROM Version into the BIOS Chip

Additionally, MS-DOS 5, ROM version is now becoming a standard in virtually all diskless sub-notebook, notebook and pen PC implementations. Many factors contribute to this approach. Chief among them is the reduced disk access and the resulting longer battery life. Another factor is the instant boot capability which is essential in certain applications.

Today's MS-DOS 5, ROM version occupies 64 KBytes of code space as specified from the Microsoft Product Description.

3.4 Relocated Resident VGA Code: VIDEO BIOS

As described above in section 3.0, video BIOS can also be mapped into the system Flash BIOS memory allowing the entire system non-volatile storage requirements to be satisfied with one Flash device. Resident VGA BIOS code takes approximately another 32 KBytes of memory space.

In summary, adding all the above code size requirements, the resulting BIOS storage area increases from the initial 128 KByte requirement to somewhere between 208 KBytes (without pen extensions) to 240 KBytes (including a pen input capability).

There are already portable designs today which have filled a 256 KByte code space to accommodate some of the above mentioned needs.

4.0 HARDWARE DESIGN FOR A 256 KBYTE BIOS

4.1 Intel 28F200BX/002BX Boot Block Flash Memory Family

Building upon the wide acceptance of the Intel28F001BX 1 Mbit flash memory for BIOS designs, a new family of higher density flash components is now available to solve the PC designer's need of implementing extended BIOS code beyond 128 KBytes.

These new flash memories at the 2 Megabit density levels, are structured around the same boot block architecture as the Intel 28F001BX and are therefore compatible. They provide block erasure capability, boot code hardware protection and very low power consumption as in the case of the 28F001BX.

In addition, they incorporate new features to simplify the device interface and allow the system designer to optimize platform designs.

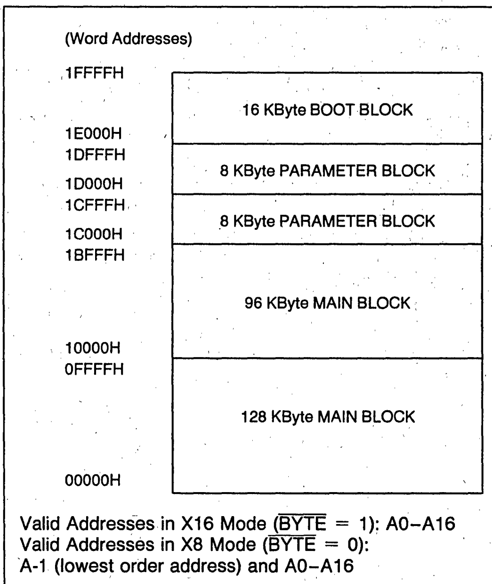
These new features are summarized as follows:

- User selectable 8-bit or 16-bit read/write operation (28F200BX)
- 60 ns access time performance
- 16 KBytes Boot Block space which is hardware protected
- Two, 8 KBytes Parameter Blocks
- One, 96 KByte Main Block
  One, 128 KByte Main Block
- 8-bit only operation and packaging for space sensitive applications (28F002BX)

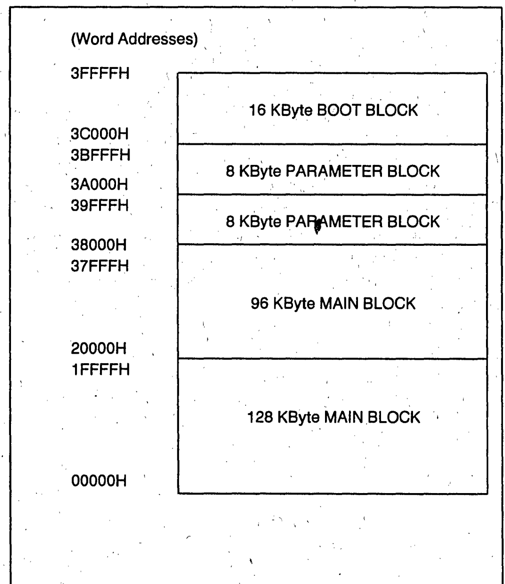
In this section, we will describe these new features in more detail and discuss their system applicability.

The blocking scheme, while still of the boot block type, is expanded by defining additional blocks (2 main blocks) to allow for software modularization and a self-contained design.

As the size of the BIOS code stored in any one device grows due to the complexity and high integration of chip-sets, so does the "kernel" code stored in the boot section. The boot and parameter blocks were accordingly doubled in size in comparison to the 28F001BX device. The two parameter blocks of 8 KBytes each allow the PC designer to store BIOS extensions or Battery-Backed SRAM configuration data (CMOS RAM, EISA configuration parameters). The two main blocks are used to store the main BIOS code in modular fashion if so desired for future easy updates. These main blocks can also be used to store ROM-executable Operating System software such as MS-DOS 5, ROM version or drivers and utilities. Refer to Figure 2 for the block locations for both the 28F200BX-T and 28F002BX-T.



28F200BX-T Top Boot Map



28F002BX-T Top Boot Map

Figure 2. 28F200BX-T/002BX-T Memory Maps

In addition, the 28F200BX/002BX devices incorporate new capabilities desired by today's sophisticated PC designer.

The byte-wide or word-wide feature available as a designer-selectable option gives the ability to interface to an 8-bit or 16-bit wide bus. The performance and hardware goals of some systems may require 16-bit BIOS data bus. A system with a small amount of RAM and a large amount of flash memory-based operating system code for example, may require a 16-bit BIOS data bus to maintain a high performance level of operation.

The high access speed of these new devices, which for the first time break the 60 ns barrier, is another big advantage the PC designer can fully exploit to increase system performance and acceptance in the marketplace. For example, a PC designer may choose to execute BIOS directly out of flash memory instead of shadowing to system RAM as well as execute MS-DOS 5, ROM version out of flash for instant-on capability and to achieve better overall system performance.

Block erasure allows independent modification of code and data and maximum flexibilty in production as well as after the system is shipped. The boot block, which is hardware protected, insures that minimal BIOS code is present to always boot up the system successfully. The 16 KByte boot block is protected from alteration during system power excursions by a high voltage pin. This write protection pin called $\overline{PWD}$ has to transition to 12V with the normal Vpp voltage pin to allow for boot block write and erase operations.

If systems are designed with the ability to switch $\overline{PWD}$ to high voltage (12V), guaranteed full non-volatility of the boot code is achieved. This feature always guarantees system recovery from power failure and provides the security needed for the end user when performing BIOS code updates.

To meet the crucial needs of lower power consumption, the 28F200BX/002BX devices incorporate a deep-power down current mode activated through the $\overline{PWD}$ pin under the TTL/CMOS level control. When this pin transitions to ground the device typically consumes 1 microwatt through the V_{CC} supply pin.

In addition, the 28F200BX/002BX devices include an Automatic Power Savings feature during active mode of operation. This feature allows the memory chip to put itself in a very low current state when it is enabled but not accessing a new memory location.

The 28F200BX/002BX devices incorporate an internal Write State Machine, Command User Interface and a Status Register to fully control the program and erase operations and greatly simplify the user write and erase algorithms and hence the update code procedure. They also include an erase suspend feature which allow the system to service interrupts and access the device during BIOS code updates (refer to appendix B).

Finally, the 28F200BX/002BX, 2 Megabit devices have an equivalent 4 Megabit boot block flash memory family of devices called the 28F400BX/004BX, allowing for easy density upgrade and total compatibility between systems using both types of memories. Refer to the documentation mentioned in the reference section of this application note.

Figure 3 is a block diagram description of the 28F200BX/002BX products.

4.2 Intel386SL™ Microprocessor Superset Platform Overview

This design example focuses primarily on how to interface the Intel 28F200BX-T or 28F002BX-T Boot Block Flash memories to the Intel386SL Microprocessor Superset in a 16-bit wide or an 8-bit wide configuration respectively. This is an extended BIOS design example which demonstrates how the barrier of the 128 KBytes BIOS size memory is eliminated. Figure 4 shows an interface diagram of the Intel 28F200BX-T Boot Block Flash memory (128 K x 16) to the Intel386SL Microprocessor superset. Figure 5 shows an equivalent interface diagram of the Intel 28F002BX-T Boot Block flash memory (256 K x 8) to the Intel386SL Microprocessor Superset.

The Intel386SL Microprocessor Superset Flash BIOS interface supports up to 256 KBytes of Flash BIOS (a 2 MEGABIT Flash memory device) to enable the system designer to meet specific design goals as described in section 3 above.

The Intel386SL Microprocessor Superset supports the following features:

- Up to 256 KBytes Flash BIOS
- VGA BIOS mapping into system BIOS
- 8-bit or 16-bit BIOS interface
- Programmable number of flash wait states for read access (from zero to fifteen Wait-States to optimize the system performance)
- BIOS shadowing mechanism

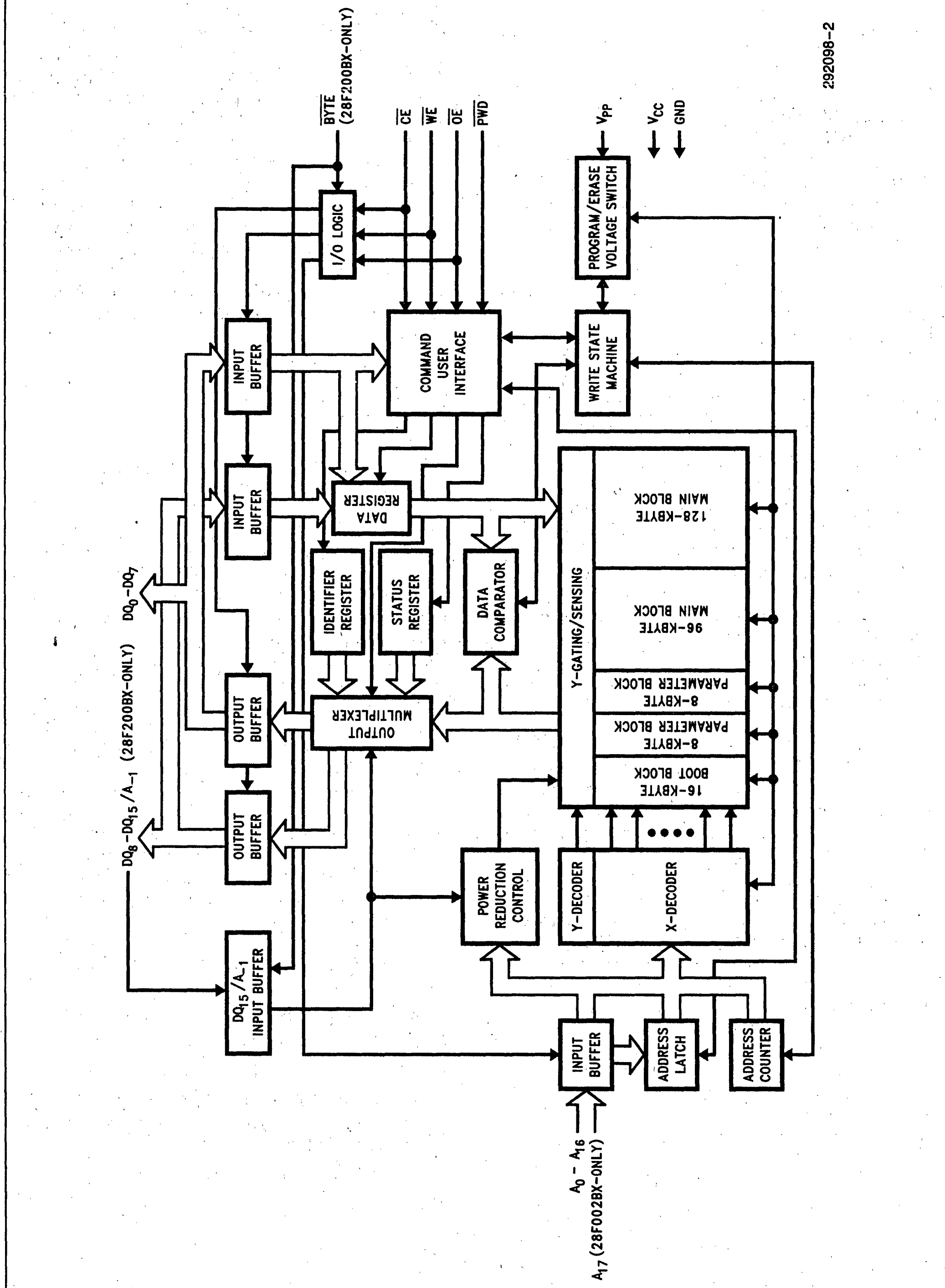


Figure 3. Block Diagram of the 28F200BX/28F002BX

Flash BIOS size configurations in the Intel386SL Microprocessor Superset system are controlled by programming certain registers located in the normal I/O address space, namely:

- EBC1CR: External Bus Unit Configuration Register 1 located at 300H
- ROMCS_DEC: ROM Chip Select Decode register located at Index 2FH
- ISAWINDOW: ISA Window control register located at B00H

When a 256 KByte Flash BIOS configuration is programmed into the Intel386SL Microprocessor Superset, 128 KBytes are directly accessible in the E0000H–FFFFFH address range. The other 128 KBytes are decoded at the top of the 16th or 32nd Megabyte of the Intel386SL superset address space, i.e., either:

FC0000H–FDFFFFH or 1FC0000H–1FDFFFFH. This extra ROM space is accessed by programming the ISA sliding control register to point to one of these two areas and then accessing the ISA sliding window in the D0000H–DFFFFH address range (64 KBytes). This mechanism allows complete access of the 256 KBytes of BIOS code without having to enter the Intel386SL protected mode.

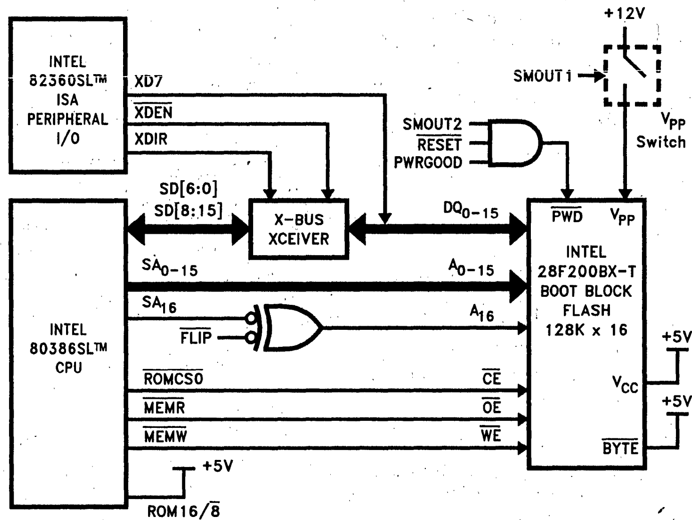
The BIOS code size is used to internally decode the ROM address space and generate two chip select signals: $\overline{\text{ROMCS0}}$ and $\overline{\text{ROMCS1}}$, to control the Flash BIOS device. In the case of a single Flash BIOS device, only $\overline{\text{ROMCS0}}$ is needed to drive the $\overline{\text{CE}}$ chip select signal of the flash memory.

In the diagram of Figure 4 note the following:

- $\overline{\text{BYTE}}$ signal is set high to enable 16-bit operation of the flash device.
- The highest order system address line SA_{16} is inverted when $\overline{\text{FLIP}}$ signal becomes active at boot-up time to relocate the boot kernel code at the top of the 1 MByte memory map for the system to boot from it. (See section 4.5).
- ROM16/$\overline{8}$ is set high to enable 16-bit bus operations.
- $\overline{\text{PWD}}$ signal is gated by the PWRGOOD signal (for reset when power fails) and by system RESET signal.
- V_{PP} supply voltage is switched to the flash device only when BIOS updates are required.

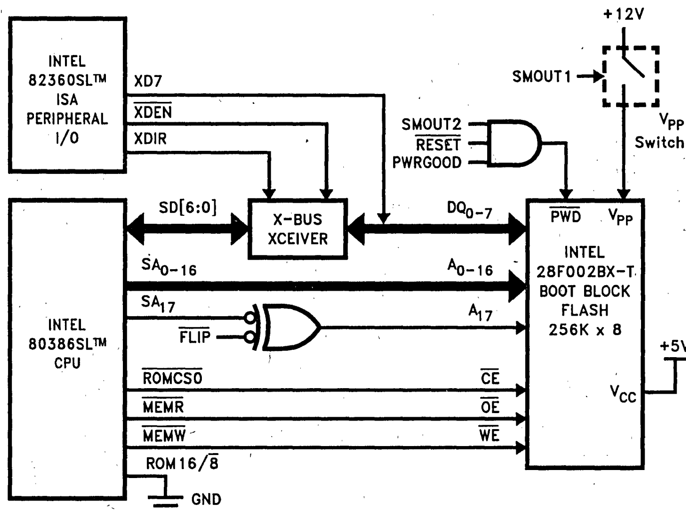
In addition to the above considerations, note the following in Figure 5:

- The highest order system address line SA_{17} is also inverted when $\overline{\text{FLIP}}$ signal becomes active at system boot-up time.
- ROM16/$\overline{8}$ is set low to enable 8-bit only bus operations.



LEGEND:
$\overline{XDEN}$ = X-BUS DATA ENABLE
XDIR = X-BUS DATA DIRECTION
SMOUT1,2 = SYSTEM MANAGEMENT OUTPUT CONTROLS
$\overline{ROMCS0}$ = ROM CHIP SELECT FOR SYSTEM FLASH BIOS
$\overline{MEMR}$ = MEMORY READ
$\overline{MEMW}$ = MEMORY WRITE
$\overline{FLIP}$ = BOOT BLOCK MEMORY MAPPING SIGNAL
ROM 16/$\overline{8}$ = ROM 16 BITS OR 8 BITS
SD[6:0], SD[15:8] = SYSTEM DATA BUS
SA0–16 = SYSTEM ADDRESS BUS
XD7 = X-BUS DATA BIT 7
PWRGOOD = POWER SUPPLY POWER GOOD SIGNAL

Figure 4. Intel386SL™ Microprocessor Superset with the 28F200BX-T Flash BIOS Chip



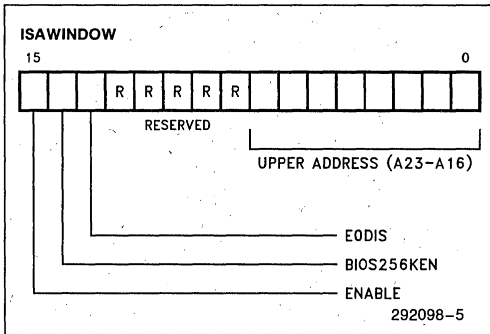
LEGEND:
XDEN = X-BUS DATA ENABLE
XDIR = X-BUS DATA DIRECTION
SMOUT1,2 = SYSTEM MANAGEMENT OUTPUT CONTROLS
ROMCS0 = ROM CHIP SELECT FOR SYSTEM FLASH BIOS
MEMR = MEMORY READ
MEMW = MEMORY WRITE
FLIP = BOOT BLOCK MEMORY MAPPING SIGNAL
ROM 16/8 = ROM 16 BITS OR 8 BITS
SD[6:0] = SYSTEM DATA BUS
SA0–17 = SYSTEM ADDRESS BUS
XD7 = X-BUS DATA BIT 7
PWRGOOD = POWER SUPPLY POWER GOOD SIGNAL

Figure 5. Intel386SL™ Microprocessor Superset with the 28F002BX-T Flash BIOS Chip

4.3 Programming the Intel386SL™ Registers

ISA SLIDING WINDOW PROGRAMMING: ISAWINDOW

When the ISA window is enabled, any access to the area of memory from D0000H to DFFFFH is remapped according to the upper address field specified in the ISAWINDOW control register. To use the ISA sliding window mechanism, the ISAWINDOW enable bit Bit 15 of the ISAWINDOW control register must be set. See the diagram below for a definition of the ISAWINDOW register:



292098-5

Bits 0–7: ISA Window upper Address bits

Corresponds to the upper address bits of the remapped address, A23–A16

Bit 7 = A23, Bit 0 = A16

Bit 8–12: Reserved. Should be reset to zero for proper system operation

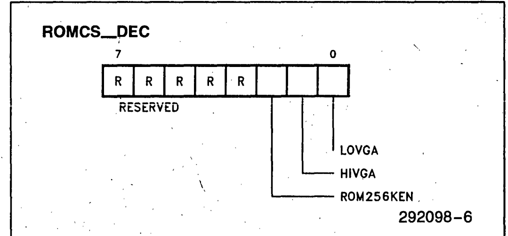
Bit 13: E0000H segment disable.
Normal ISA-bus cycles are generated for access to address range E0000H–EFFFFH.

Bit 14: BIOS256KEN, 256 KByte Flash enable. Enables support for a 256 KByte Flash device.

Bit 15: ISA-Window Enable.

ROM CHIP-SELECT DECODE DEFINITION: ROMCS_DEC

The ROM chip-select decode register is used to indicate to the Intel 82360SL I/O chip how to control the X-bus buffers and XD7 signal with regard to BIOS accesses. The Intel386SL Superset allows systems to have VGA BIOS to physically reside in system BIOS code space. See the diagram below for a definition of the ROM chip-select register:



292098-6

Bit 0: LOVGA
Low VGA BIOS mapped into system BIOS.

Bit 1: HIVGA
High VGA BIOS mapped into system BIOS.

Bit 2: ROM256KEN
256 KByte ROM enable. When this bit is set, the Intel 82360SL will properly assert $\overline{XDEN}$ and route XD7 through SD7 properly for accesses to the following ranges:

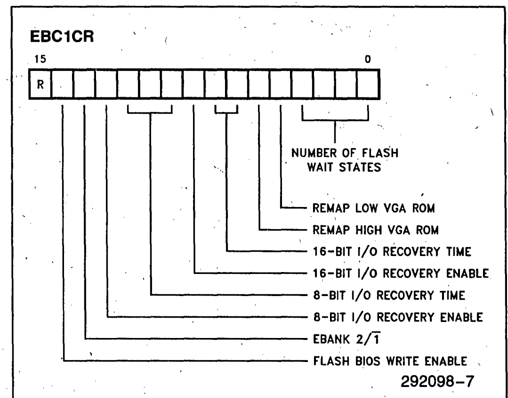
E0000H–FFFFFH

FC0000H–FFFFFFH

Bits 2–7: Reserved

EXTERNAL BUS UNIT CONFIGURATION REGISTER 1:EBC1CR

This register has many functions. It controls the number of flash memory wait states, the mapping of flash memory chip-selects, the I/O recovery time for the ISA-bus and write cycles to the Flash device. The following diagram depicts the definition of this register.



292098-7

Bit 0–3: Number of Flash wait states.
Flash access cycles with zero to fifteen wait states can be programmed i.e.

[0,0,0,0] represents 0 wait states.

[1,1,1,1] represents 15 wait states.

Bit 4: Low VGA ROM remap
Allows VGA BIOS to be remapped into system BIOS Flash from E0000H to E3FFFH.

Bit 5: High VGA ROM remap
Allows VGA BIOS to be remapped into system BIOS Flash from E4000H to E7FFFH.

Bits 6–7: 16-bit I/O recovery time

Bit 8: 16-bit I/O recovery enable

Bits 9–11: 8-bit I/O recovery time

Bit 12: 8-bit I/O recovery enable

Bit 13: EBANK2/1̄, Number of BIOS Flash Banks, 2 or 1
Used to determine whether one (low) or two (high) banks of BIOS Flash are used in the system.

Bit 14: Flash BIOS write enable
Used to enable writes to the Flash device. When this bit is set, write accesses can occur and BIOS updates are possible.

Bit 15: Reserved

4.4 The ISA Sliding Window

The ISA sliding window mechanism is similar to the Expanded Memory System or EMS mechanism. It allows the access of extended memory on the ISA bus in real mode. Hence, there is no conflict in memory allocation when the BIOS chip is accessed during system power-up. This is a straightforward method of implementing a 256 KByte extended Flash BIOS in an Intel386SL-based Personal Computer. Figure 6 shows how extended memory is accessed with the ISA sliding window.

Therefore to use a 256 KByte Flash device in an Intel386SL Microprocessor Superset design, the following steps are specified:

Example:

Enable support of 256 KByte Flash BIOS

Enable Flash BIOS write access

One BIOS Flash Bank used

Program zero Flash wait states (for 16 MHz Intel386SL with a 60 ns Intel 28F200BX-T or 28F002BX-T).

Program the ISA Sliding Window to access memory between FC0000H and FDFFFFH.

if external bus unit configuration space is not enabled
  enable external bus unit configuration space
    set bit 13 to one
    set bit 14 to one
    set bits 0–3 to [0,0,0,0]

if ROM Chip-select decode register is not enabled
  enable ROM chip-select decode register
    set bit 2 to one

enable ISAWINDOW control register
    set bit 15 to one
    set bit 14 to one

set upper address field of the ISAWINDOW register to FCH

3

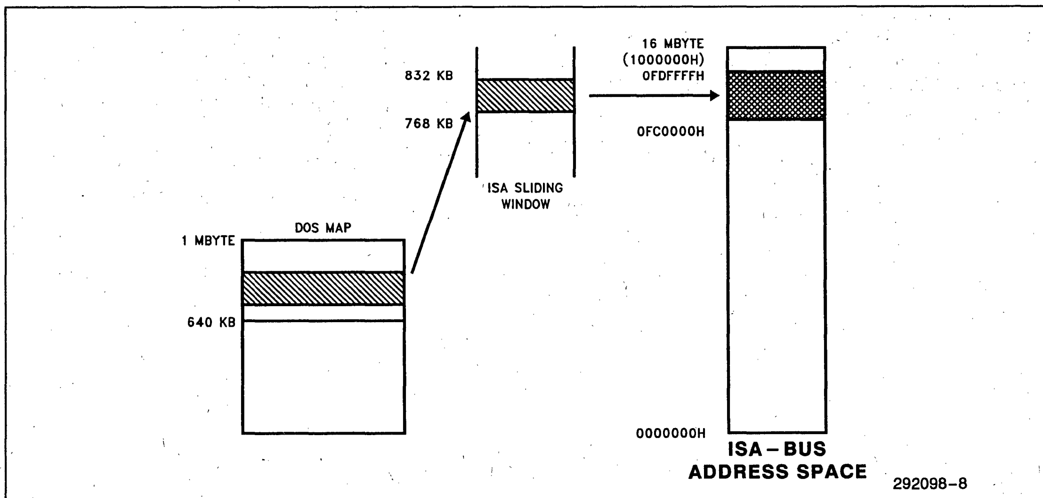


Figure 6. ISA Sliding Window and Extended Memory Maps

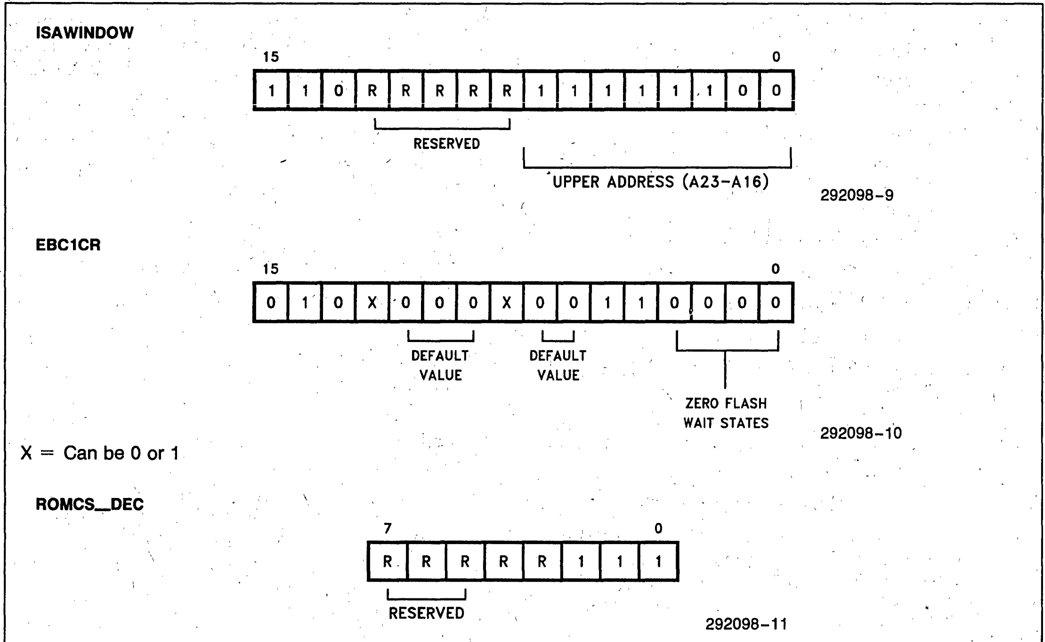


Figure 7. Programming of the Intel386SL™ Microprocessor Superset Registers

4.5 The 28F200BX-T/28F002BX-T in the 1 MByte DOS Memory Map

In addition, due to the above mapping considerations when interfacing a 256 KByte Flash BIOS chip to the Intel386SL™ Microprocessor Superset, it is necessary to flip the 256 KByte Flash device in half at boot time to relocate the Boot Block in the correct physical address at FFFFFFH for the Intel386SL™ CPU to boot from it. This is due to the fact that the Intel386SL™ Microprocessor Superset uses the bottom 128 KBytes of the 256 KByte Flash chip to map down to the real mode 128 KByte allocated BIOS space.

Figure 8 shows how the Intel 28F200BX-T flash device fits in the DOS 1 MByte memory map. The same memory map applies to the 28F002BX-T device when designing an 8-bit only system.

5.0 SOFTWARE DESIGN CONSIDERATIONS

The subject of BIOS code update is already discussed extensively in references [1] and [3]. Appendix A of this application note is an example of a flash BIOS update routine. In this section, we mainly discuss considerations of code segmentation and describe how the system handles the different pieces of code under various conditions.

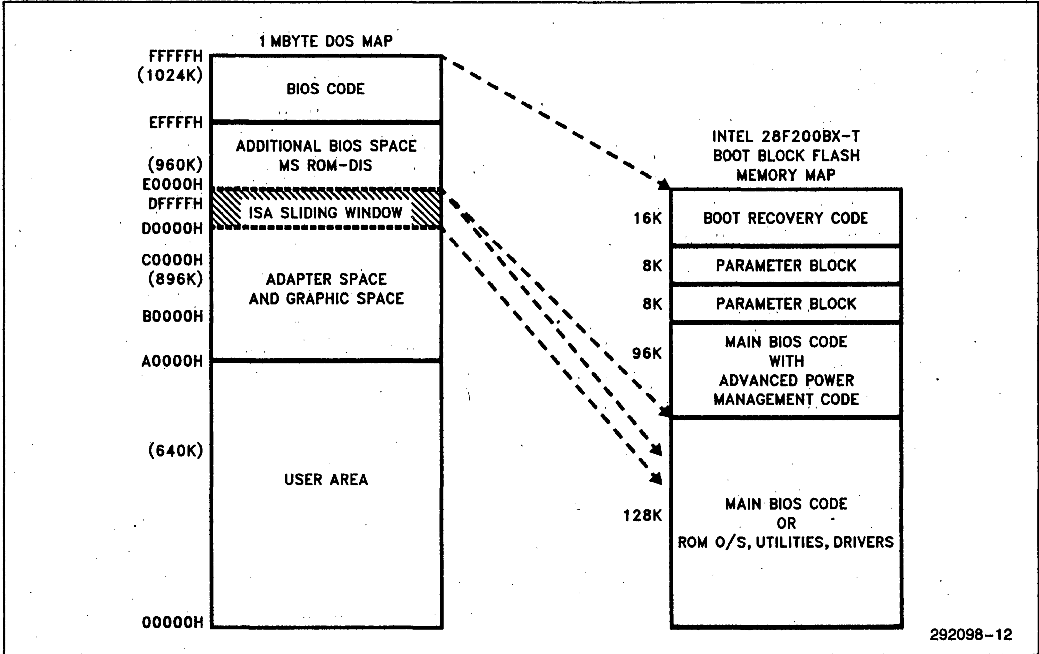


Figure 8. 28F200BX-T/28F002BX-T in the 1 MByte DOS Memory Map

As explained in section 3, advanced notebook designs require a large ROM space to conserve valuable RAM space for applications. In addition to the boot kernel code and standard BIOS code, one may put the following code modules:

- APM code
- VGA BIOS
- ExCA socket services
- Operating Systems such as Microsoft MS-DOS 5.0 ROM version

When designing a 256 KByte BIOS system as described in this application note using Intel's 28F200BX/002BX, proper code segmentation is essential to achieve optimum system performance.

We can divide the system's operating environment into either:

BOOT-TIME CONDITIONS (when system powers-up or is rebooted)

or RUN-TIME CONDITIONS (after boot-up is complete)

The code modules described above are used at different times during normal system operation. Hence, a segregation of the code stored in the flash memory is necessary for proper system operation.

BOOT-TIME EXECUTION

The system requires the boot kernel to be present at the FFFFFH segment during this portion of the cycle.

When using the 28F200BX/002BX flash memory, the first page present will be the top 128 KByte. The rest of the first page is used to store APM code, VGA BIOS code or ExCA socket services. These code modules are then copied from their location below the boot block to system RAM for later initialization after the standard BIOS has started.

Once all code modules are copied into RAM, this first 128 KByte page of flash can be swapped out or exchanged with the 2nd 128 KByte page which is required for run-time execution.

RUN-TIME EXECUTION

Standard BIOS code is required to be present from E0000H to FFFFFH segment during this time period to handle any BIOS calls and maintain proper system operation.

Microsoft MS-DOS 5.0 ROM version code is also required to be present so the BIOS can "SCAN" it in as an adapter.

5.1 16 KByte Recovery Code

The 16 KByte recovery code is critical when a BIOS update does not successfully complete. The reasons for this failure can be divided into two categories:

1. System power failure during BIOS update
2. System reset (soft boot) during BIOS update

The processor will execute recovery code out of the 16 KB boot block when either power is restored or when the system boots up after the reset function.

The 16 KB size recovery code allows the software designer to incorporate as many BIOS system checks as possible to implement basic system functionality.

For instance, these checks may include:

Cursor Positioning
Keyboard services
Time of day service
Basic System services

5.2 28F200BX Reprogramming

Three basic algorithms allow the system designer to reprogram the Flash BIOS chip and perform the necessary tasks for a BIOS update. These algorithms are:

Automated Byte/Word-wide programming
Automated Block erase
Erase Suspend and Resume

For a description of these algorithms, the reader is encouraged to study reference [8]. Appendix B in this application note includes the four flowcharts associated with the algorithms.

To obtain the software drivers necessary to control the device reprogramming operations, consult your local Intel sales office.

5.3 Power Management

Power management is an essential part of any true portable PC design. The design of sophisticated power management techniques is becoming a key differentiator between different machines, and hence is a competitive advantage for the system integrator.

To help the system designer with this often difficult task of optimizing system performance and battery life, the 28F200BX family of products includes three distinct low power modes of operation. These are:

- Standby Current Mode, where the device typically consumes 50 μA

- Automatic Power Savings Feature, where the device typically consumes 1 mA
- Deep powerdown Mode, where the device typically consumes 0.2 μA

For a detailed description of these modes of operation, consult reference [8].

6.0 DESIGNING A 3.3V SYSTEM

The ability to design 3.3V systems used to be a future consideration. Longer battery life and lighter weight portable computers are some of the key objectives for any portable design. Now, thanks to the increasing availability of low voltage components, true low power machines are possible to realize in practice.

6.1 Low Voltage Chips

The list of 3.3V components necessary to build the essential parts of a portable computer is becoming longer every day. Semiconductor manufacturers have recognized the urgent need to supply low voltage chips to the portable marketplace.

The Intel 28F200BX/002BX Boot Block Flash Memories are now sampling in 3.3V versions. These low voltage versions of the 28F200BX/002BX are functionally equivalent to their 5V counterparts and are 100% pinout compatible. So a system converting to 3.3V operation can substitute low voltage 2 Mbit chips (28F200BX-L/002BX-L) when desired without any circuit board modification.

6.2 Power Savings and Improved Battery Life

The 28F200BX/002BX 3.3V chips reduce the total power drawn during normal read operation to less than 25% of the total power in 5V mode. This is a substantial savings in current which translates to 25% less battery drain and hence longer battery life.

Similarly, low voltage version of the most popular microprocessors reduce the total power dissipated by a substantial amount.

The combination of these current savings plus the other system components current reductions improve battery life dramatically and allow the mobile PC user to benefit from weight reduction, longer operating time, lower system cost and higher performance.

7.0 CONCLUSIONS

7.1 Benefits of Extended Flash BIOS

This application note deals with the concepts of extended BIOS implementations in portable PC designs, but it can also be easily adapted to desktop PC systems for which BIOS code requirements can easily exceed 128 KBytes.

We have attempted to explain the requirements and the needs of today's advanced portable BIOS designs which have to meet many difficult and often conflicting requirements.

Boot Block Flash memory is the ideal storage solution to implement the above mentioned features. Furthermore, as the cost of solid state non-volatile flash memory keeps decreasing, the need to switch to these types of media for storing extended BIOS, operating system software, utilities, and in the future application code, becomes more evident and perhaps the only way a PC manufacturer can effectively compete by producing the best engineered, most optimally designed notebooks, palmtop PCs and pen-based computers.

REFERENCES

For more information on the concepts presented in this application note, the reader is encouraged to reference the following documents.

[1] Technical Paper: "Flash: The Optimum BIOS Storage Device" by Brian Dipert, 1991 SVPC — Order Number 297003

[2] "ROM BIOS: The best place for portable PC Power-Management features" by Lance Hansche, 1991 SVPC

[3] AP-341: "Designing an Updatable BIOS Using Flash Memory"—Order Number 292077

[4] AP-357: "Power Supply Solutions for Flash Memory,"—Order Number 292092

[5] Intel386SL Microprocessor Superset System Design Guide—Order Number 240816

[6] Intel386SL Microprocessor Superset Programmer's Reference Manual—Order Number 240815

[7] Intel386SL Microprocessor Superset Data Book—Order Number 240814

[8] Intel 28F200BX/002BX Datasheet—Order Number 290448

[9] Intel 28F400BX/004BX Datasheet—Order Number 290451

[10] Intel 28F200BX-L/002BX-L Datasheet—290449

[11] Intel 28F400BX-L/004BX-L Datasheet—290450

APPENDIX A

Example of a Flash Update Utility Pseudo-Code

This example is for a standard BIOS code and APM code update using the 28F200BX/002BX flash device in an Intel 386SL-based portable computer. Modify this utility, if required, to suit your particular system needs.

Initialize system (set-up user screen, check battery power, check device ID)

Get BIOS/APM file Options (from floppy or through modem)

   If no file present
   Send error message to insert BIOS update floppy, or press ESC to exit

Display BIOS/APM update files, prompt user for choice and load to memory

   If file is invalid
   Prompt for correct file or exit

Inform user of upcoming event, provide option to continue or exit

   If user continues, inform user not to turn off the power or soft-reboot system (CNTL-ALT-DEL)

Erase 28F200BX/002BX device 96 KByte main block

   If system interrupt occurs
   Suspend erase operation if flash memory access is required

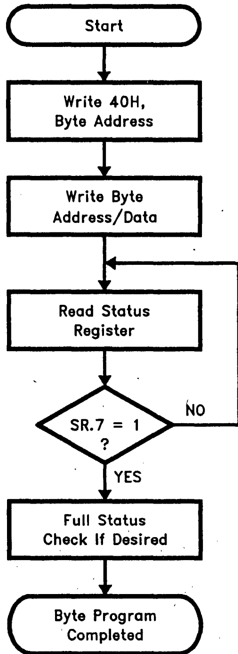
Resume erase operation of 96 KByte main block

Write new file(s) into flash memory 96 KByte main block

Indicate to user that flash reprogramming is complete

Prompt user to reboot the system to continue normal operation

APPENDIX B
28F200BX-T/28F002BX-T Programming Flowcharts



292098-13

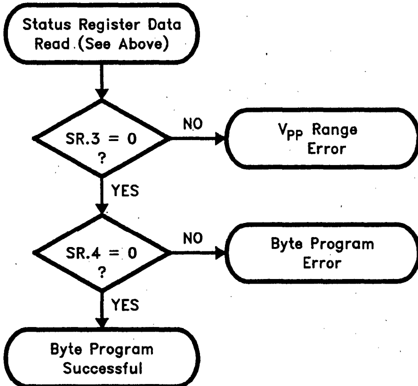
Bus Operation	Command	Comments
Write	Setup Program	Data = 40H Address = Byte to be programmed
Write	Program	Data to be programmed Address = Byte to be programmed
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent bytes.

Full status check can be done after each byte or after a sequence of bytes.

Write FFH after the last byte programming operation to reset the device to Read Array Mode.

Full Status Check Procedure



292098-14

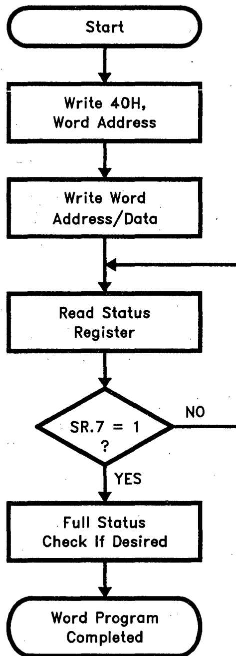
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Byte Program Error

SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple bytes are programmed before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Automated Byte Programming Flowchart



292098-15

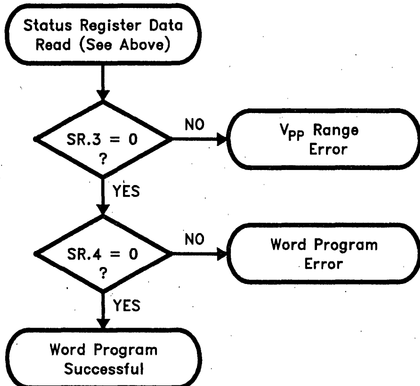
Bus Operation	Command	Comments
Write	Setup Program	Data = 40H Address = Word to be programmed
Write	Program	Data to be programmed Address = Word to be programmed
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent words.

Full status check can be done after each word or after a sequence of words.

Write FFH after the last word programming operation to reset the device to Read Array Mode.

Full Status Check Procedure



292098-16

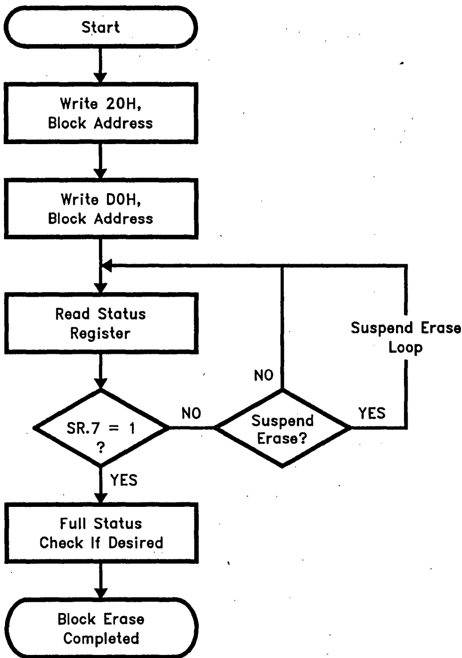
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Word Program Error

SR.3 MUST be cleared, if set during a program attempt, before further attempts are allowed by the Write State Machine.

SR.4 is only cleared by the Clear Status Register Command, in cases where multiple words are programmed before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Automated Word Programming Flowchart



292098–17

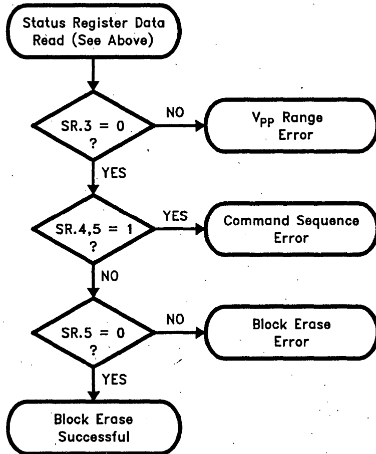
Bus Operation	Command	Comments
Write	Setup Erase	Data = 20H Address = Within block to be erased
Write	Erase	Data = D0H Address = Within block to be erased
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready, 0 = Busy

Repeat for subsequent blocks.

Full status check can be done after each block or after a sequence of blocks.

Write FFH after the last block erase operation to reset the device to Read Array Mode.

Full Status Check Procedure



292098–18

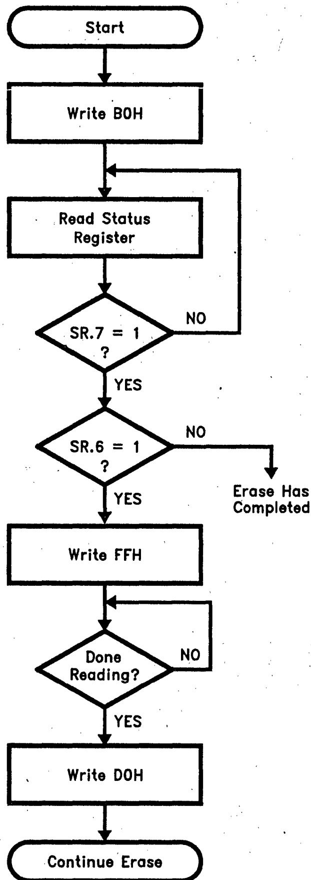
Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4,5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error

SR.3 MUST be cleared, if set during an erase attempt, before further attempts are allowed by the Write State Machine.

SR.5 is only cleared by the Clear Status Register Command, in cases where multiple blocks are erased before full status is checked.

If error is detected, clear the Status Register before attempting retry or other error recovery.

Automated Block Erase Flowchart



Bus Operation	Command	Comments
Write	Erase Suspend	Data = B0H
Read		Status Register Data. Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Standby		Check SR.7 1 = Ready
Standby		Check SR.6 1 = Suspended
Write	Read Array	Data = FFH
Read		Read array data from block other than that being erased.
Write	Erase Resume	Data = D0H

292098–19

Erase Suspend/Resume Flowchart

APPENDIX C

ROM Chip-Select Decoding Table

COMPLETE ROM DECODINGS

The following table illustrates the decoding of $\overline{ROMCS0}$ and $\overline{ROMCS1}$ (not including the FLASH Disk interface decodings):

ROM Chip-Select Decode Table

EBANK2/$\overline{1}$	256KEN	VGAHI	VGALOW	E0DIS	$\overline{ROMCS0}$	$\overline{ROMCS1}$
0	0	0	0	0	00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	0	0	0	1	00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	0	0	1	0	00C0000H–00C3FFFH 00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	0	0	1	1	00C0000H–00C3FFFH 00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	0	1	0	0	00C4000H–00C7FFFH 00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	0	1	0	1	00C4000H–00C7FFFH 00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	0	1	1	0	00C0000H–00C7FFFH 00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	0	1	1	1	00C0000H–00C7FFFH 00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	1	0	0	0	0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH 00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	

ROM Chip-Select Decode Table (Continued)

EBANK2/$\overline{1}$	256KEN	VGAHI	VGALOW	E0DIS	$\overline{\text{ROMCS0}}$	$\overline{\text{ROMCS1}}$
0	1	0	0	1	0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH 00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	1	0	1	0	00C0000H–00C3FFFH 0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH 00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	1	0	1	1	00C0000H–00C3FFFH 0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH 00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	1	1	0	0	00C4000H–00C7FFFH 0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH 00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	1	1	0	1	00C4000H–00C7FFFH 0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH 00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	1	1	1	0	00C0000H–00C7FFFH 0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH 00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
0	1	1	1	1	00C0000H–00C7FFFH 0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH 00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	
1	0	0	0	0	00F0000H–00FFFFFH 0FF0000H–0FFFFFFH 1FF0000H–1FFFFFFH	00E0000H–00EFFFFH 0FE0000H–0FEFFFFH 1FE0000H–1FEFFFFH

ROM Chip-Select Decode Table (Continued)

EBANK2/$\overline{1}$	256KEN	VGAHI	VGALOW	E0DIS	$\overline{ROMCS0}$	$\overline{ROMCS1}$
1	0	0	0	1	00F0000H–00FFFFFH 0FF0000H–0FFFFFFH 1FF0000H–1FFFFFFH	0FE0000H–0FEFFFFH 1FE0000H–1FEFFFFH
1	0	0	1	0	00F0000H–00FFFFFH 0FF0000H–0FFFFFFH 1FF0000H–1FFFFFFH	00C0000H–00C3FFFH 00E0000H–00EFFFFH 0FE0000H–0FEFFFFH 1FE0000H–1FEFFFFH
1	0	0	1	1	00F0000H–00FFFFFH 0FF0000H–0FFFFFFH 1FF0000H–1FFFFFFH	00C0000H–00C3FFFH 0FE0000H–0FEFFFFH 1FE0000H–1FEFFFFH
1	0	1	0	0	00F0000H–00FFFFFH 0FF0000H–0FFFFFFH 1FF0000H–1FFFFFFH	00C4000H–00C7FFFH 00E0000H–00EFFFFH 0FE0000H–0FEFFFFH 1FE0000H–1FEFFFFH
1	0	1	0	1	00F0000H–00FFFFFH 0FF0000H–0FFFFFFH 1FF0000H–1FFFFFFH	00C4000H–00C7FFFH 0FE0000H–0FEFFFFH 1FE0000H–1FEFFFFH
1	0	1	1	0	00F0000H–00FFFFFH 0FF0000H–0FFFFFFH 1FF0000H–1FFFFFFH	00C0000H–00C7FFFH 00E0000H–00EFFFFH 0FE0000H–0FEFFFFH 1FE0000H–1FEFFFFH
1	0	1	1	1	00F0000H–00FFFFFH 0FF0000H–0FFFFFFH 1FF0000H–1FFFFFFH	00C0000H–00C7FFFH 0FE0000H–0FEFFFFH 1FE0000H–1FEFFFFH
1	1	0	0	0	00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH
1	1	0	0	1	00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH
1	1	0	1	0	00C0000H–00C3FFFH 00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH
1	1	0	1	1	00C0000H–00C3FFFH 00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH
1	1	1	0	0	00C4000H–00C7FFFH 00E0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH
1	1	1	0	1	00C4000H–00C7FFFH 00F0000H–00FFFFFH 0FE0000H–0FFFFFFH 1FE0000H–1FFFFFFH	0FC0000H–0FDFFFFH 1FC0000H–1FDFFFFH

ROM Chip-Select Decode Table (Continued)

EBANK2/$\overline{1}$	256KEN	VGAHI	VGALOW	E0DIS	$\overline{\text{ROMCS0}}$	$\overline{\text{ROMCS1}}$
1	1	1	1	0	00C0000H-00C7FFFH 00E0000H-00FFFFFH 0FE0000H-0FFFFFFH 1FE0000H-1FFFFFFH	0FC0000H-0FDFFFFH 1FC0000H-1FDFFFFH
1	1	1	1	1	00C0000H-00C7FFFH 00F0000H-00FFFFFH 0FE0000H-0FFFFFFH 1FE0000H-1FFFFFFH	0FC0000H-0FDFFFFH 1FC0000H-1FDFFFFH

APPENDIX D

List of BIOS software vendors already supporting or announcing their future support for the 28F200BX/002BX flash BIOS chips:

SystemSoft Corporation
Contact: Cliff Sharin
508-651-0088
313 Speen Street
Natick, MA 01760

Phoenix Technologies Ltd.
Contact: Howard Cohen
408-452-6529
40 Airport Parkway
San Jose, CA 95110

Award Software, Inc.
Contact: Jeffry Flink
408-370-7979 (ext. 214)
130 Knowles Drive
Los Gatos, CA 95030-1832

American Megatrends, Incorporated
Contact: Tom Rau
404-246-8612
1346 Oakbrook Drive, Suite 120
Norcross, GA 30093

Quadtel
Contact: Dale Buscaino
714-754-4422 (ext. 250)
3190-J Airport Loop
Costa Mesa, CA 92626

APPLICATION NOTE

AP-364

November 1992

28F008SA Automation and Algorithms

BRIAN DIPERT
MCD MARKETING APPLICATIONS

Order Number: 292099-002

28F008SA Automation and Algorithms

CONTENTS

	PAGE
1.0 INTRODUCTION	3-496
2.0 AUTOMATION AND ALGORITHMS	3-496
2.1 28F008SA Automation and the Write State Machine	3-496
Command User Interface	
Status Register	
2.2 Byte Write Algorithm	3-499
2.3 Block Erase Algorithm	3-501
2.4 Erase Suspend/Resume Algorithm	3-501
2.5 Write State Machine Current/Next State Overview	3-504
Read Array	
Byte Write Setup	
Byte Write	
Erase Setup	
Erase	
Erase Command Error	
Erase Suspend to Status/Array	
Read Status	
Read Identifier	
2.6 Block Erase As a Background Task	3-506
ADDITIONAL INFORMATION	3-507

3

1.0 INTRODUCTION

The Intel 28F008SA FlashFile™ Memory is today's optimum solution for high density solid state storage. Flash memory, exemplified by the 28F008SA, is an enabling technology for today's powerful system designs that are higher performance, more compact, lighter, more rugged and have longer battery life.

Features of the 28F008SA include:

- High-Density Symmetrically Blocked Architecture:
  — Sixteen 64-Kbyte Blocks
- Extended Cycling Capability
  — 100,000 Block Erase Cycles
  — 1.6 Million Block Erase Cycles per Chip
- Automated Byte Write and Block Erase
  — Command User Interface
  — Status Register
- System Performance Enhancements
  — RY/$\overline{BY}$ Status Output
  — Erase Suspend Capability
- Deep Powerdown Mode
  — 0.20 μA I_{CC} Typical
- Very High Performance Read
  — 85 ns Maximum Access Time
- SRAM-Compatible Write Interface
- Hardware Data Protection Features
  — Erase/Write Lockout during Power Transitions
- Industry Standard Packaging
  — 40 Lead TSOP, 44 Lead PSOP
- ETOX™ III Nonvolatile Flash Memory Technology
  — 12V Byte Write/Block Erase

The 28F008SA's automation is a significant enhancement to the manual algorithms of first-generation flash memory devices. System software and hardware designs that fully understand and exploit this automation will greatly benefit from its versatility and capabilities. The concepts presented in this document are applicable to such designs.

This application note discusses in-depth operation of the 28F008SA FlashFile memory Write State Machine and internal algorithms, emphasizing how they interface to system hardware and software. The 28F008SA datasheet (order number 290429) is a valuable reference document, providing in-depth device technical specifications, package pinouts and timing waveforms. Companion application note AP-359, "28F008SA Hardware Interfacing" (order number 292094) describes supply voltage derivation and filtering, control input/output implementation, high density layout and high speed design techniques, as well as providing example system interfaces to common microprocessor buses. AP-360, "28F008SA Software Drivers" (order number 292095) provides example ASM-86 and "C" routines for controlling the 28F008SA. AP-359 and AP-360 should be reviewed in conjunction with this application note and the 28F008SA datasheet for a complete understanding of this device.

2.0 AUTOMATION AND ALGORITHMS

Figure 1 shows a block diagram of the 28F008SA and its internal contents. Although a main subject of this application note is software interface to read and alter memory contents, it is useful to begin with an overview of the 28F008SA hardware subsections that are directly manipulated by the system. In particular, this application note will first discuss the Write State Machine (WSM) and Command User Interface/Status Register, and then explain the software routines that control this hardware.

2.1 28F008SA Automation and the Write State Machine

When the system microprocessor reads flash memory data from the 28F008SA, it uses control lines $\overline{CE}$ and $\overline{OE}$, along with address inputs, to select a byte of data directly from the memory cell array. However, the system does not directly access the array when it writes to the 28F008SA; instead it writes to the Command User Interface, whose register contents are interpreted and translated into WSM actions. The WSM can be thought of as a dedicated "processor", along with companion clock-generation circuitry, integrated into the flash memory. After receiving proper commands or command sequences, it controls byte write and block erase algorithms internally. The status of the WSM is not invisible to the system; the WSM interfaces to the outside world through a full-featured Status Register and dedicated RY/$\overline{BY}$ (Ready/$\overline{Busy}$) output. Automation has significant benefits, some of which are more obvious than others.

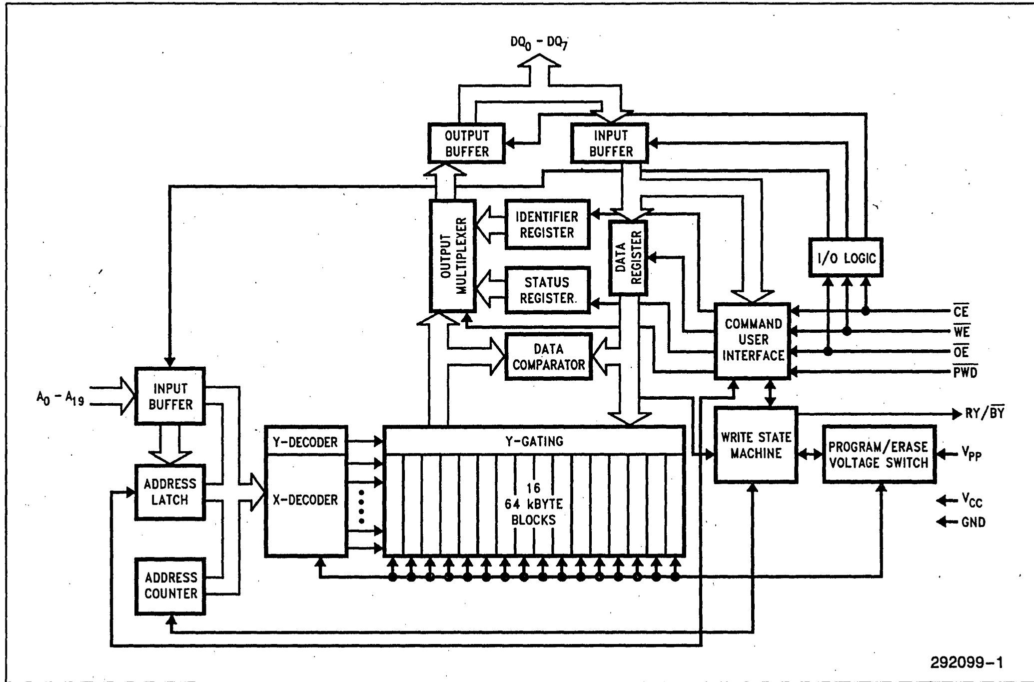


Figure 1. 28F008SA Block Diagram

The WSM architecture dramatically simplifies the program and erase algorithms of first-generation flash memory devices. Hardware/software timers, erase pre-programming, byte-by-byte verification and margining, pulse repetition and limited microprocessor multitasking capability throughout data update have been eliminated, replaced by a simple two-command write for both block erase and byte write. The 28F008SA WSM halts itself when its internal algorithms are complete, and can alert the system to this completion by a hardware interrupt (using RY/$\overline{\text{BY}}$) or via software polling of the 28F008SA Status Register.

Internal automation frees the system to execute higher-priority tasks while a 28F008SA is being block erased or byte written, and inherent in this capability is the most powerful advantage of the WSM. Operating systems prioritize file operations in the following order:

- Read
- Write
- Erase

When an array of 28F008SA components is used as solid-state storage (in a memory card, integrated in a flash-based "hard drive" form factor or resident on the system motherboard), system software can initiate slower block erase (0.3 sec minimum) of one or several 28F008SAs and, by not being "tied" to the erase algorithm, execute higher priority reads (85 ns minimum) or writes (6 µs minimum) of other 28F008SAs as operating system requests dictate. Additionally, erase suspend/resume capability allows data retrieval from a 28F008SA currently being block erased, again enabling "read" as the highest priority task. Block erase as a background task is discussed in Section 2.6 of this document.

Command User Interface

Table 2 shows the various command sequences that are accepted and interpreted by the 28F008SA Command User Interface and WSM. Writes to the CUI enable reading of device data and intelligent identifiers, reading and clearing of the Status Register, and commencement of internal byte write, block erase and erase suspend/resume algorithms. The CUI itself does not occupy a specifically addressable memory location, and contains a latch used to store the command and address/data information needed to execute the command. Erase Setup and Erase Confirm commands require both appropriate command data and an address within the block to be erased. The Byte Write Setup command requires appropriate command data and the address of the location to be written, while the Byte Write command consists of the data to be written and its address location.

Table 1. Status Register Definitions

WSMS	ESS	ES	BWS	VPPS	R	R	R
7	6	5	4	3	2	1	0

SR.7 = WRITE STATE MACHINE STATUS
1 = Ready
0 = Busy

SR.6 = ERASE SUSPEND STATUS
1 = Erase Suspended
0 = Erase in Progress/Completed

SR.5 = ERASE STATUS
1 = Error in Block Erasure
0 = Successful Block Write

SR.4 = BYTE WRITE STATUS
1 = Error in Byte Write
0 = Successful Byte Write

SR.3 = V_{PP} STATUS
1 = V_{PP} Low Detect; Operation Abort
0 = V_{PP} OK

NOTES:
RY/$\overline{BY}$ or the Write State Machine Status bit must first be checked to determine byte write or block erase completion, before the Byte Write or Erase Status bits are checked for success.
If the Byte Write AND Erase Status bits are set to "1"s during a block erase attempt, an improper command sequence was entered. Attempt the operation again.
If V_{PP} low status is detected, the Status Register must be cleared before another byte write or block erase operation is attempted.
The V_{PP} Status bit, unlike an A/D converter, does not provide continuous indication of V_{PP} level. The WSM interrogates the V_{PP} level only after the byte write or block erase command sequences have been entered and informs the system if V_{PP} has not been switched on. The V_{PP} Status bit is not guaranteed to report accurate feedback between V_{PPL} and V_{PPH}.

SR.2–SR.0 = RESERVED FOR FUTURE ENHANCEMENTS
These bits are reserved for future use and should be masked out when polling the Status Register.

Table 2. Command Definitions

Command	Bus Cycles Req'd	Notes	First Bus Cycle			Second Bus Cycle		
			Operation	Address	Data	Operation	Address	Data
Read Array/Reset	1		Write	X	FFH			
Intelligent Identifier	3	1, 2, 3	Write	X	90H	Read	IA	IID
Read Status Register	2	2	Write	X	70H	Read	X	SRD
Clear Status Register	1		Write	X	50H			
Erase Setup/Erase Confirm	2	1	Write	BA	20H	Write	BA	D0H
Erase Suspend/Erase Resume	2		Write	X	B0H	Write	X	D0H
Byte Write Setup/Write	2	1, 2, 4	Write	WA	40H	Write	WA	WD
Alternate Byte Write Setup/Write	2	1, 2, 4	Write	WA	10H	Write	WA	WD

NOTES:
1. IA = Identifier Address: 00H for manufacturer code, 01H for device code.
BA = Address within the block being erased.
WA = Address of memory location to be written.
2. SRD = Data read from Status Register. See Table 4 for a description of the Status Register bits.
WD = Data to be written at location WA. Data is latched on the rising edge of WE.
IID = Data read from intelligent identifiers.
3. Following the intelligent identifier command, two read operations access manufacturer and device codes.
4. Either 40H or 10H are recognized by the WSM as the Byte Write Setup command.
5. Commands other than those shown above are reserved by Intel for future device implementations and should not be used.

Status Register

Table 1 shows the 28F008SA Status Register and defines its various bits. Like the Command User Interface, it does not occupy a specific memory location within the device. It functions as an output of the WSM, informing the system when internal byte write or block erase algorithms have completed, if these algorithms completed successfully, and whether the 28F008SA is currently in Erase Suspend mode. Bit 7 (Write State Machine Status) is replicated in the device RY/$\overline{BY}$ hardware output. The default state of the upper 5 bits of the Status Register after powerup and return from deep powerdown mode is 10000 (binary).

A separate Clear Status Register command allows re-initialization of Status Register data after analysis. The Status Register is not cleared until this command is written to the 28F008SA.

Bits 5 and 4 of the Status Register, if set by the WSM via a byte write or block erase attempt, do not block subsequent attempts (they need not be cleared before another byte write/block erase command sequence is written to the device). However, if the WSM detects a "low V_{PP}" condition and subsequently sets bit 3 of the Status Register, the Status Register MUST be cleared before another algorithm command sequence will be recognized by the 28F008SA.

It is important to note that the V_{PP} Status bit of the Status Register DOES NOT act like an always-functional A/D converter; its normal state, even with V_{PP} below 6.5V, is "0". The WSM only analyzes the V_{PP} level after a byte write or block erase command sequence has been written to the device, and if it detects that V_{PP} is "low" it will cancel the impending byte write or block erase operation and set the V_{PP} Status bit to "1". Therefore, the V_{PP} Status bit cannot be used by the system as an indication of proper V_{PP} level, before a byte write or block erase sequence is initiated. The system should instead insert an appropriate software delay between turning on V_{PP} and writing an initial command sequence, or use external hardware as a V_{PP} feedback mechanism.

2.2 Byte Write Algorithm

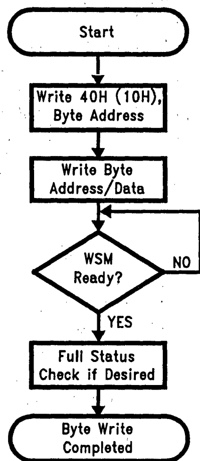
Figure 2 provides a graphical representation of the 28F008SA byte write algorithm. As can be seen, this consists solely of a two-command write sequence, followed by a periodic poll of the device RY/$\overline{BY}$ output or Status Register. The 28F008SA automatically outputs Status Register data when read after the two-command byte write sequence (see Section 2.5). Byte write typically completes in 9 μs.

The byte write algorithm requires high voltage V_{PPH} (12V ±5%) on the device V_{PP} input until internal algorithm completion is reported by the WSM. If byte write is attempted while $V_{PP} = V_{PPL}$ (≤6.5V), the V_{PP} Status bit of the Status Register will be set to "1", and array data will not be altered. Byte write attempts while $V_{PPL} < V_{PP} < V_{PPH}$ produce spurious results and should not be attempted.

The Status Register will only report errors for "1"s that do not write to "0"s during a byte write attempt. Erasure (see Section 2.3) is the method used to change data "0"s to "1"s using flash technology. If the system software attempts to write "1"s to a byte at bit locations already at value "0", no Status Register error will be reported for those specific bits.

It is often desired to write multiple bytes of data at one time to memory. Since the Status Register is only cleared after the Clear Status Register command is written to the 28F008SA, a string of bytes can be sequentially written to the device before the "full status check procedure" examines Status Register bits other than SR.7.

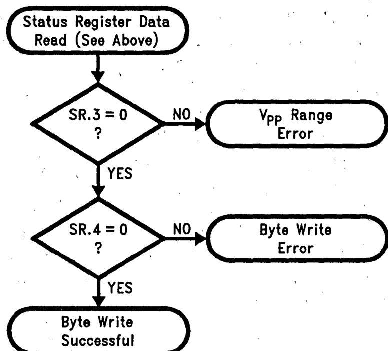
Byte write abort occurs when the 28F008SA $\overline{PWD}$ input drops to V_{IL} (deep powerdown mode is entered), or V_{PP} drops to V_{PPL}. Although the WSM is halted in either case, byte data is partially written at the location where aborted. A repeat byte write sequence after system integrity is restored will complete the desired operation, or data can be initialized to a known value of "FF" thru block erasure.



292099-2

Bus Operation	Command	Comments
Write	Byte Write Setup	Data = 40H (10H) Address = Byte to be written
Write	Byte Write	Data to be written Address = Byte to be written
Standby/ Read		Check RY/$\overline{BY}$ V_{OH} = Ready, V_{OL} = Busy or Read Status Register Check SR.7 1 = Ready, 0 = Busy Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register
Repeat for subsequent bytes Full status check can be done after each byte or after a sequence of bytes Write FFH after the last byte write operation to reset the device to Read Array Mode		

FULL STATUS CHECK PROCEDURE



292099-3

Bus Operation	Command	Comments
Optional Read		CPU may already have read Status Register data in WSM Ready polling above
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Byte Write Error
SR.3 MUST be cleared, if set during a byte write attempt, before further attempts are allowed by the Write State Machine SR.4 is only cleared by the Clear Status Register Command, in cases where multiple bytes are written before full status is checked If error is detected, clear the Status Register before attempting retry or other error recovery.		

Figure 2. Automated Byte Write Flowchart

2.3 Block Erase Algorithm

Figure 3 provides a graphical representation of the 28F008SA block erase algorithm, similar in its two-command write sequence to the byte write algorithm discussed earlier. Both the Erase Setup and Erase Confirm commands must be accompanied by an address within the desired block to be erased to FFH. The 28F008SA automatically outputs Status Register data when read after the two-command block erase sequence (see Section 2.5). Block erase typically completes in 1.6 sec.

Again similar to byte write, the block erase algorithm requires high voltage V_{PPH} (12V ±5%) on the device V_{PP} input until internal algorithm completion is reported by the WSM. If block erase is attempted while $V_{PP} = V_{PPL}$ (≤6.5V), the V_{PP} Status bit of the Status Register will be set to "1", and array data will not be altered. Block erase attempts while $V_{PPL} < V_{PP} < V_{PPH}$ produce spurious results and should not be attempted.

If write of the Erase Setup command is followed by write of any other command but Erase Confirm, the WSM will decode this as an illegal sequence. It will not attempt to erase the specified block, and will report error back to the system by setting both the Erase Status and Byte Write Status bits of the Status Register to "1". Since the Status Register is only cleared after the Clear Status Register command is written to the 28F008SA, a string of blocks within a 28F008SA can be sequentially erased before the "full status check procedure" examines Status Register bits other than SR.7.

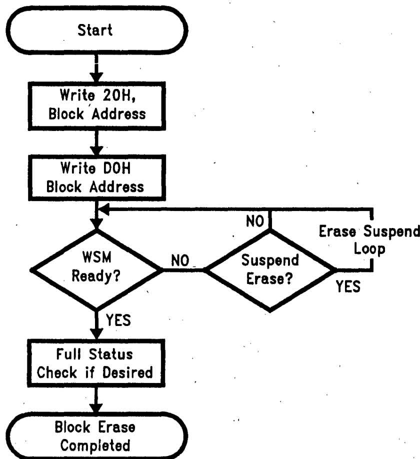
Block erase abort occurs when the 28F008SA $\overline{PWD}$ input drops to V_{IL} (deep powerdown mode is entered) or V_{PP} drops to V_{PPL}. A repeat block erase sequence after system integrity is restored will complete the desired operation.

2.4 Erase Suspend/Resume Algorithm

Figure 4 gives a software flowchart for implementing erase suspend/resume using the 28F008SA. As mentioned in Section 2.1, operating systems prioritize data reads highest, and consequently the 28F008SA has been designed with read as its highest performance function. Erase suspend allows system software to postpone WSM-controlled block erase if the system requests read of data from **a different block** of the same device. Although any block of the 28F008SA can be read, the block being erased when suspended will contain unknown data.

The 28F008SA is suspended by writing the Erase Suspend command (B0H) to it while the WSM is executing an erase algorithm. The WSM will halt block erase, set bits 7 and 6 of the Status Register to "1" and transition RY/$\overline{BY}$ to V_{OH}, after which time system software can read data from either the array or Status Register. Issuing the Erase Resume command (D0H) signals the WSM to resume block erase.

V_{PP} must remain at V_{PPH} throughout the erase suspend interval, even when reading from the flash memory array. The 28F008SA will detect a V_{PP} transition to V_{PPL} while suspended, and report this error via Status Register bit 3 (set to "1") after the Erase Resume command is written to it.

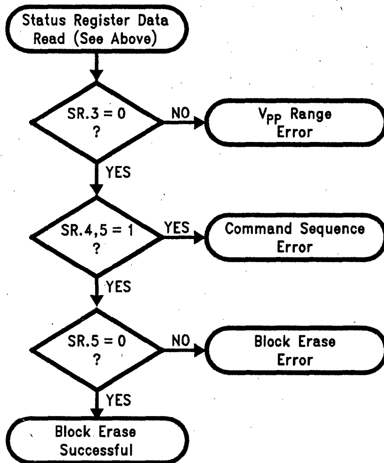


Bus Operation	Command	Comments
Write	Erase Setup	Data = 20H Address = Within block to be erased
Write	Erase Confirm	Data = D0H Address = Within block to be erased
Standby/ Read		Check RY/$\overline{BY}$ V_{OH} = Ready, V_{OL} = Busy or Read Status Register Check SR.7 1 = Ready, 0 = Busy Toggle $\overline{OE}$ or $\overline{CE}$ to update Status Register

Repeat for subsequent blocks

Full status check can be done after each block or after a sequence of blocks

Write FFH after the last block erase operation to reset the device to Read Array Mode



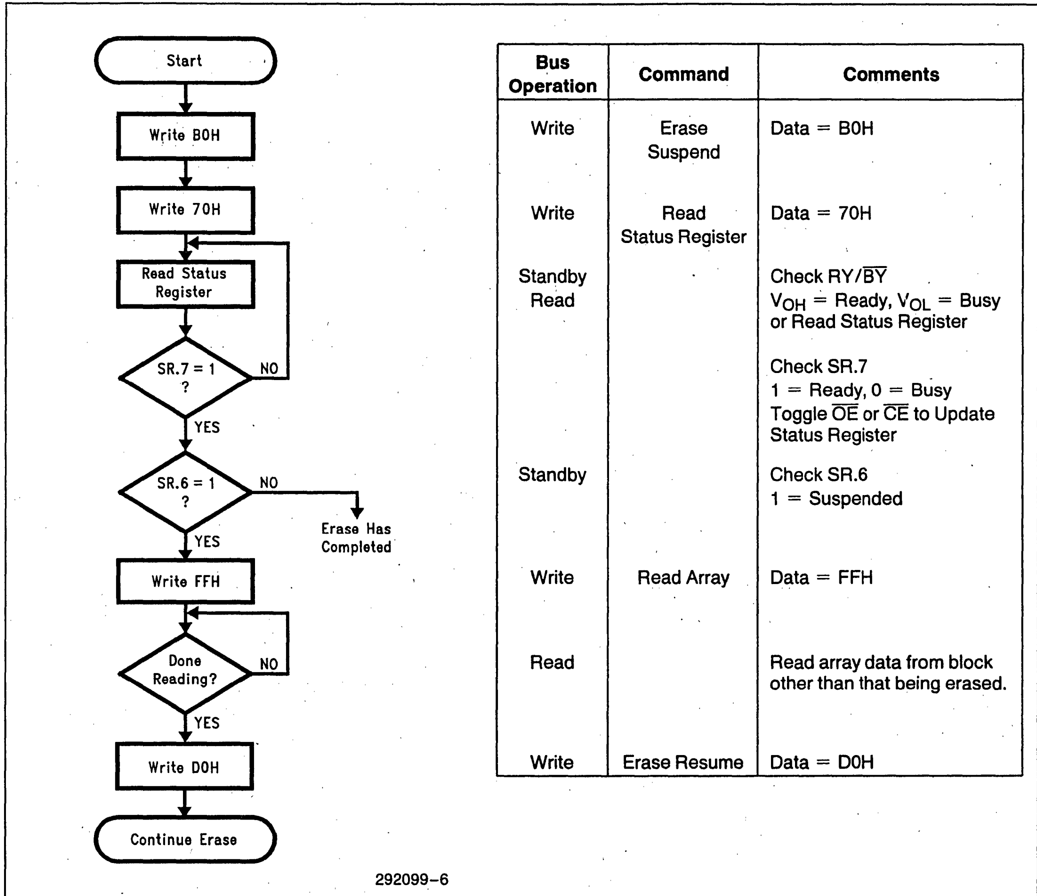
Bus Operation	Command	Comments
Optional Read		CPU may already have read Status Register data in WSM Ready polling above
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4,5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error

SR.3 MUST be cleared, if set during a block erase attempt, before further attempts are allowed by the Write State Machine

SR.5 is only cleared by the Clear Status Register Command, in cases where multiple blocks are erased before full status is checked

If error is detected, clear the Status Register before attempting retry or other error recovery

Figure 3. Automated Block Erase Flowchart



Bus Operation	Command	Comments
Write	Erase Suspend	Data = B0H
Write	Read Status Register	Data = 70H
Standby Read		Check RY/$\overline{BY}$ V_{OH} = Ready, V_{OL} = Busy or Read Status Register Check SR.7 1 = Ready, 0 = Busy Toggle $\overline{OE}$ or $\overline{CE}$ to Update Status Register
Standby		Check SR.6 1 = Suspended
Write	Read Array	Data = FFH
Read		Read array data from block other than that being erased.
Write	Erase Resume	Data = D0H

Figure 4. Erase Suspend/Resume Flowchart

Since the WSM is driven by its own oscillator, internal to the 28F008SA, it operates asynchronously to the system CPU and its clock. Therefore, the possibility exists that the WSM could complete erase, returning to "ready", **between** when the system reads "busy" from the Status Register and writes the Erase Suspend command. Analyzing both the WSM Status and Erase Suspend Status bits of the Status Register, as shown in the flowchart, will alert the system to such an occurrence.

2.5 Write State Machine Current/Next State Overview

Byte write and block erase automation equate to tremendous power and capability in system implementations of the 28F008SA, if fully exploited. An in-depth understanding of the WSM, its states and its responses to inputs, will assist the software engineer in developing optimized driver routines for flash memory-based file storage and other high-performance applications. Table 3 lists all possible WSM "current states", command inputs and resultant "next states".

Non-shaded boxes highlight those state transitions which will most commonly occur when reading from and modifying 28F008SA contents, and these transitions should be understood in most depth. Shaded boxes, on the other hand, represent lesser-used or nonsensical transitions, such as improper erase command sequences.

Before reading the 28F008SA, if the current WSM mode is not known (if, for example, an interrupt service routine has potentially interacted with the device), first write the desired output command (i.e. Read Status Register, Read Array or Intelligent Identifier). This ensures that the 28F008SA will be in a known state when read and will output expected data.

Read Array

The 28F008SA automatically defaults to Read Array mode when powered up, or when it returns from Deep Powerdown mode. As the name implies, the 28F008SA outputs array data when read in Read Array mode. Read Array is also the default mode after the Clear Status Register command is written in most other modes.

Byte Write Setup

The 28F008SA transitions to Byte Write Setup mode after it receives the Byte Write Setup command. If the 28F008SA is read in Byte Write Setup mode, it outputs Status Register data.

Byte Write

After the 28F008SA is placed in Byte Write Setup mode, the next address/data combination written to it transitions the WSM to Byte Write mode, where the "Byte Write Command" is latched as desired data to write to the array at the specified address location. Immediately, the WSM examines V_{PP}, and if it detects an invalid level, it halts with V_{PP} error indication in the Status Register (bit 3 = "1"). Bit 7 of the Status Register is "0", and the RY/$\overline{BY}$ output is driven to V_{OL}, while the WSM is executing the internal byte write algorithm in Byte Write mode. The 28F008SA automatically outputs Status Register data when in Byte Write mode.

Erase Setup

The 28F008SA transitions to Erase Setup mode after it receives the Erase Setup command. If the 28F008SA is read in Erase Setup mode, it outputs Status Register data.

Table 3. Write State Machine Current/Next States

Current State	RY/$\overline{BY}$ Status	Data When Read	Command Input (and Next State)								
			Read Array (FFH)	Byte Write Setup (10/40H)	Erase Setup (20H)	Erase Confirm (D0H)	Erase Suspend (B0H)	Erase Resume (D0H)	Read Status (70H)	Clear Status (50H)	Read ID (90H)
Read Array	V_{OH}	Array	Read Array	Byte Write Setup	Erase Setup	Read Array	Read Array	Read Array	Read Status	Read Array	Read ID
Byte Write Setup	V_{OH}	Status	Byte Write (Command Input = Data to be Byte Written)								
Byte Write (Not Complete)	V_{OL}	Status	Byte Write								
Byte Write (Complete)	V_{OH}	Status	Read Array	Byte Write Setup	Erase Setup	Read Array			Read Status	Read Array	Read ID
Erase Setup	V_{OH}	Status	Erase Command Error			Erase	Erase Command Error	Erase	Erase Command Error		
Erase Command Error	V_{OH}	Status	Read Array	Byte Write Setup	Erase Setup	Read Array			Read Status	Read Array	Read ID
Erase (Not Complete)	V_{OL}	Status	Erase				Erase Suspend to Status	Erase			
Erase (Complete)	V_{OH}	Status	Read Array	Byte Write Setup	Erase Setup	Read Array			Read Status	Read Array	Read ID
Erase Suspend to Status	V_{OH}	Status	Erase Suspend to Array	Reserved	Erase Suspend to Array	Erase	Erase Suspend to Array	Erase	Erase Suspend to Status	Erase Suspend to Array	Reserved
Erase Suspend to Array	V_{OH}	Array	Erase Suspend to Array	Reserved	Erase Suspend to Array	Erase	Erase Suspend to Array	Erase	Erase Suspend to Status	Erase Suspend to Array	Reserved
Read Status	V_{OH}	Status	Read Array	Byte Write Setup	Erase Setup	Read Array			Read Status	Read Array	Read ID
Read Identifier	V_{OH}	ID	Read Array	Byte Write Setup	Erase Setup	Read Array			Read Status	Read Array	Read ID

NOTE:
1. State transitions labeled "Reserved" are set aside by Intel Corporation for potential future device implementations. Command sequences to access these states should not be attempted.

3

Erase

After the 28F008SA is placed in Erase Setup mode, write of the Erase Confirm command transitions the WSM to Erase mode, where the specified address is decoded into one of 16 blocks to be erased. Immediately, the WSM examines V_{PP}, and if it detects an invalid level, it halts with V_{PP} error indication in the Status Register (bit 3 = "1"). Bit 7 of the Status Register is "0", and the RY/$\overline{BY}$ output is driven to V_{OL}, while the WSM is executing the internal block erase algorithm in Erase mode. The 28F008SA automatically outputs Status Register data when in Erase mode.

Erase Command Error

This is the other possible transition mode after Erase Setup, and occurs when an invalid command (anything but Erase Confirm/Resume) is written to the 28F008SA as the second in the two-command block erase sequence. In this mode, the WSM does not attempt a block erase, and it returns an error indication to the system by setting both bits 4 and 5 of the Status Register to "1". The 28F008SA automatically outputs Status Register data when in Erase Command Error mode.

Erase Suspend to Status/Array

While the WSM is busy executing an internal block erase algorithm, it can be placed in erase suspend by writing the Erase Suspend command. After receiving and decoding this command, the WSM suspends block erase, drives the RY/$\overline{BY}$ output to V_{OH}, sets bits 6 and 7 of the Status Register to "1" and transitions to "Erase Suspend to Status" mode. The 28F008SA automatically outputs Status Register data when in "Erase Suspend to Status" mode.

The only valid command other than Read Status and Erase Resume at this time is Read Array, which transitions the WSM to "Erase Suspend to Array" mode. As the name implies, the 28F008SA outputs array data, not Status Register contents, in this mode. While in both Erase Suspend modes, V_{PP} must remain at V_{PPH} for erase to complete successfully when resumed.

Writing the Erase Resume (same as Erase Confirm) command to the 28F008SA transitions the WSM out of Erase Suspend and back to Erase. In conjunction with this, the WSM returns RY/$\overline{BY}$ to V_{OL} and resets bits 6 and 7 of the WSM to "0".

Read Status

As the name implies, the 28F008SA automatically outputs Status Register contents when read in Read Status mode. If system software writes the Clear Status command at this point, the WSM resets the Status Register to its default value and transitions to Read Array mode.

Read Identifier

The 28F008SA outputs its manufacturer identifier of 89H when read from address 00000H when in Read Identifier mode. Similarly, a read from address 00001H returns the device identifier A2H. Using this information, the system can automatically match the device with its proper block erase and byte write algorithms. Reads from addresses other than 00000H and 00001H are not supported by Intel, and consistent results of such reads are not documented, guaranteed or recommended.

2.6 Block Erase as a Background Task

As mentioned earlier, the internal WSM block erase algorithm typically takes 1.6 seconds to complete. Proper implementation of block erase from a hardware and software standpoint, however, can mask this delay, by taking advantage of the 28F008SA's internal automation and full-featured system interface. Execution of block erase as a background task, with higher priority read and write functions in the foreground, is the key.

The recommended scenario includes an "intelligent" operating system routine which can keep track of "busy" devices in the 28F008SA array. After initiating block erase on these components, the operating system is free to concentrate on reads and writes, or any other pending requests that demand its attention. The 28F008SA RY/$\overline{BY}$ output alerts the system when block erase completes, and the operating system acts on this completion in the resulting interrupt service routine.

Hardware interrupt via the RY/$\overline{BY}$ output is a recommended technique for block erase. However, this method should be evaluated closely for alerting the system to byte write completion. The WSM typically completes a byte write attempt in 9 μs, a much shorter time than that consumed in many CPU interrupt latencies. In such cases, software polling of the 28F008SA Status Register to detect WSM "ready", versus hardware interrupt, will result in highest byte write performance. Reference AP-359, "28F008SA Hardware Interfacing", for circuit implementations that not only combine RY/$\overline{BY}$s into a common INT, but also allow RY/$\overline{BY}$ masking if desired.

ADDITIONAL INFORMATION

		Order Number
28F008SA Datasheet		290429
28F008SA-L Datasheet		290435
AP-359	"28F008SA Hardware Interfacing"	292094
AP-360	"28F008SA Software Drivers"	292095
ER-27	"The Intel 28F008SA Flash Memory"	294011
ER-28	"ETOX™-III Flash Memory Technology"	294012

REVISION HISTORY

Number	Description
002	Page 3-502: Erase Command Error: In this mode, the WSM does not attempt a block erase, and it returns an error indication to the system by setting both bits **4** and **5** of the Status Register to "1".

intel®

APPLICATION NOTE

AP-371

October 1992

Designing Flash Card Readiness into Today's Systems

GARY FORNI
FLASH CARD APPLICATIONS GROUP

Order Number: 292109-001

Designing Flash Card Readiness into Today's Systems

CONTENTS PAGE

1.0 STANDARDS 3-510

2.0 HARDWARE 3-510

3.0 SOFTWARE 3-511

4.0 UPGRADABILITY 3-513

4.1 Example 1 3-513

4.2 Example 2 3-513

5.0 CONCLUSION 3-514

APPENDIX A: WHERE TO GO FOR MORE INFORMATION 3-515

This application note was created to inform the OEM how to make his system Flash card ready.

A number of requests have come in from customers who are adding PCMCIA slots to their systems and want to include the capability to upgrade to flash cards and Microsoft's Flash File System (FFS). This summary discusses how to include flash card upgradability in systems being designed today.

This guide is divided into four areas: standards, hardware, software, and upgrades.

1.0 STANDARDS

PCMCIA revision 2.0 sets the standard for the physical and electrical characteristics of memory and I/O PC cards. Unfortunately, what it doesn't do is set the standard for the host system's architecture. That's where the ExCA™ (Exchangable Card Architecture) specification comes in. ExCA specifies the minimum requirements for what the host system must provide. The ExCA specification covers both the hardware and software requirements of the system. The recommendations in this design guide follow the ExCA requirements.

2.0 HARDWARE

While software can be upgraded at a later date, either in the factory, or in the field, hardware typically cannot. That's why it's so important that all of the hardware necessary to support flash cards be designed into a system up front.

Fortunately, ExCA compliant hardware is available today. The Intel 82365SL PC Card Interface Controller (PCIC) is ExCA compliant and has rapidly become the PC Card controller standard.

The PCIC interfaces the ISA bus to the PCMCIA socket, and handles interrupt steering, memory and I/O window mapping, and power management. A schematic for creating two fully buffered sockets is found in the 82365SL™ datasheet and is included here as Figure 1.(1)

Writing and erasing Flash cards requires $V_{PP} = 12V$, with $I_{PP} = 60$ mA minimum. Remember, when choosing a voltage converter, other devices (video, etc) may also require 12V, so take care to pick one that will supply all of your simultaneous needs. For more information and numerous examples on power conversion read Application Note 357 Power Supply Solutions for Flash Memory.

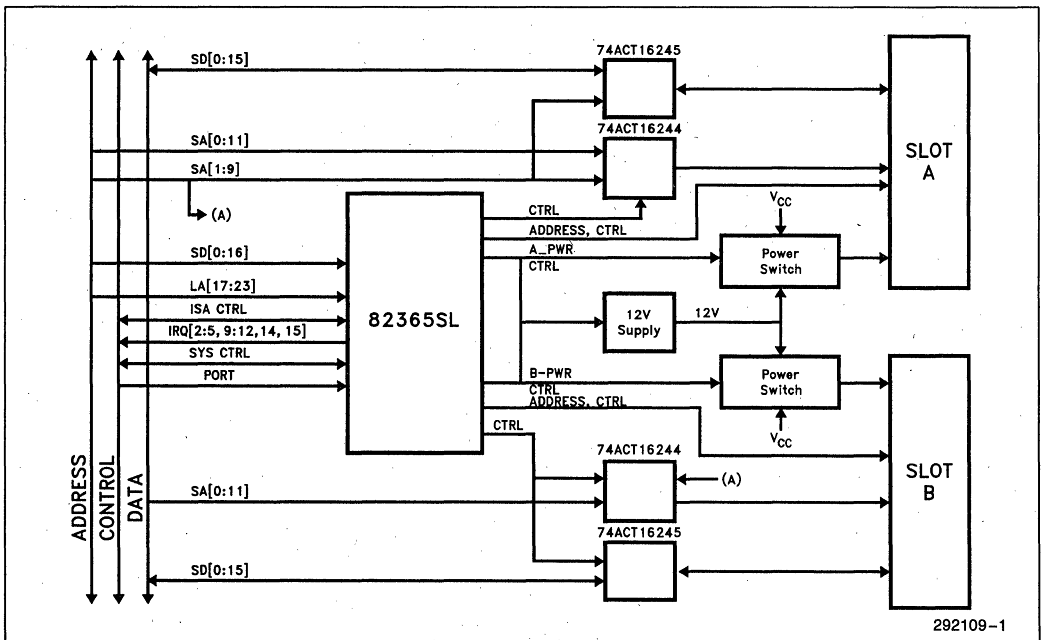


Figure 1. A Buffered, Two-Socket Implementation of the 82365SL

NOTE:
1. Some designers may want to simplify this interface. Note that the buffers allow a user to remove either card without glitching any signals on the system bus.

Many of the future pieces of software that you will want to upgrade your system with are expected to reside in BIOS. Starting with the power-managed notebook and the complex high end desktop class of systems, upgradable flash BIOS has become a common design practice. Leaving sufficient headroom to handle additional code has also become prevalent. Using a 2 MB flash chip as the BIOS storage media should provide sufficient room for most software upgrades.

3.0 SOFTWARE

While ExCA hardware is available today, the full ExCA software hierarchy is still in development. Systems that exist today are not fully ExCA compliant. The problem is much of the software is still being created or is in alpha and beta testing. Figure 2 shows the ExCA software hierarchy.

A brief introduction to the software layers:

Socket Services: The only hardware specific (with the exception of the MTD) piece of software. Socket Services can be loaded as a device driver but eventually will be integrated into the BIOS. All other pieces of software work through Socket Services to communicate with the hardware. Socket Services is available from your favorite BIOS vendor. Socket Services recently was revised from version 1.01 to 2.0. When dealing with your BIOS vendor make sure to request version 2.0. See the appendix for the names of BIOS vendors working on socket services.

Card Services: This is the resource manager of the PC Card system. All drivers notify Card Services upon being loaded. Card Services allocates the system resources as required by both the PC Cards and their respective software drivers. Initially this will be a device driver but is expected to also migrate to the BIOS. The first Card Services should be available from BIOS vendors starting in early Q4'92. When discussing Card Services with your BIOS vendor make a point of letting them know which version of Socket Services you are planning to use.

Memory Technology Driver (MTD): While Carddrv has the ability to program SRAM and bulk erase (Intel Series I) cards, it does not have the ability to program Series 2 flash cards or to take advantage of the special performance and power management registers on the newer cards. Memory card specific programming algorithms and information on the use of special card features reside inside the MTD. When future cards are released different programming algorithms may be required. By maintaining the memory card specific information as a separate device driver, a system can be upgraded on the fly, with an MTD that is carried on the new card or with an accompanying floppy disk. These new MTD's can be appended to the host's MTD. The first MTD should be available near the end of 1992. Since MTD's are card specific, Intel will provide the MTDs for it's own cards. Contact Intel for a copy of the Series 2 MTD.

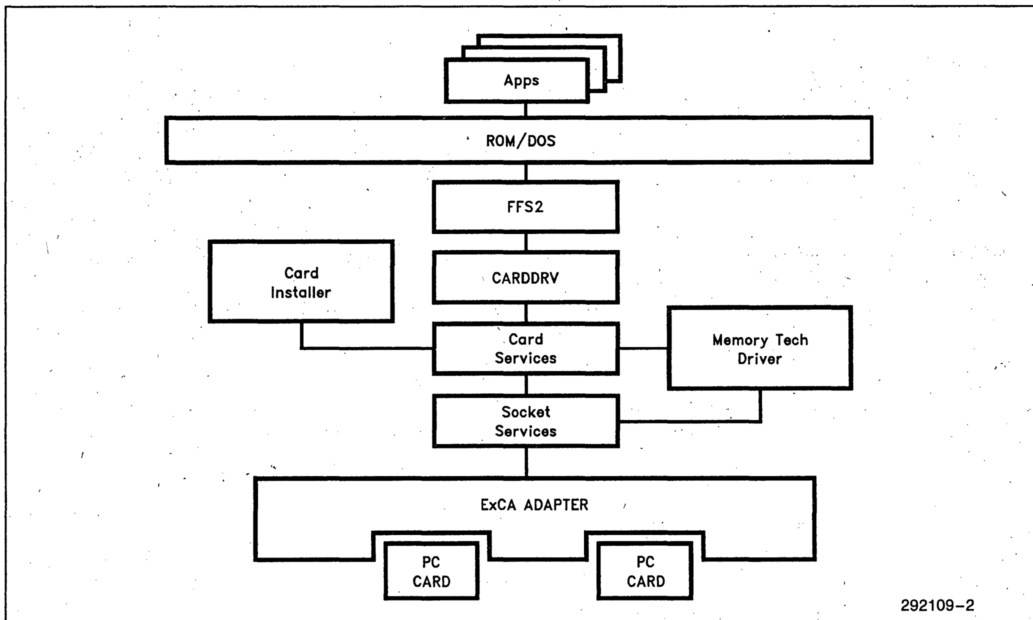


Figure 2. ExCA Software Hierarchy

Card Install: Card Install is a device driver that does for I/O cards much of what MTD does for memory cards. The Card Install software reads a PC Card's attribute memory and if it is an I/O card, configures the card and system accordingly. Flash memory does not use the Card Install software.

FFS2: Microsoft's Flash File System (also called MS-Flash.Sys): FFS2 acts as a redirector by capturing the DOS Int 21h and routing it to the Carddrv software. FFS2 provides the wear-leveling, cleanup algorithms and linked list file structure that maximize the efficiency of flash as a mass storage media.

Carddrv: The interface between FFS2 and the actual ExCA software. Carddrv handles read and write accesses, queuing and erasing blocks in the flash card. Carddrv has the ability to partition cards into more than one type of format. This multi-partitioning gives FFS cards the ability to add a BPB/FAT partition (sometimes called a disk image) and allows bootability. Carddrv has general abilities to write to flash memory, but passes off more card specific requests to the Memory Technology Driver.

Figure 3 shows the main transitional phases of the ExCA software hierarchy. Today, systems that use FFS with Flash cards are running phase 1 software with a customized carddrv that encompasses all of the functions of Socket Services, Card Services, Carddrv, and the MTD. Phase 2 pulls the Socket Services functions out of Carddrv and makes the system software hardware independent. Phase 2 software is just becoming available today. Phase 3 is the first time all of the ExCA software becomes available. Phase 3 is expected to occur late in Q4'92.

Altogether a system will need 50 KB–70 KB of space free for these upgrades. If the OEM stores these features in BIOS he will need at least a 2 MB component.

Due to the recent decision to add compression to their filing system, Microsoft's Flash File System (FFS) production version is not available as of the print deadline of this application note. Though many companies have Beta copies of FFS, it is not expected to go into production until the middle of Q4'92. To get a Beta copy of FFS call Microsoft at the number listed in the Appendix.

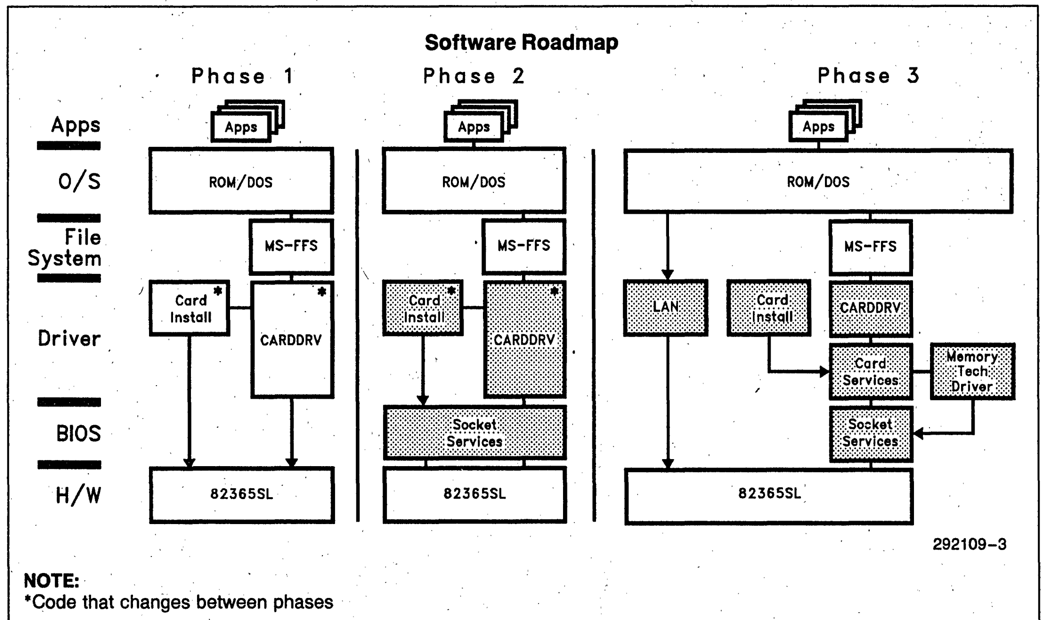


Figure 3. Roadmap of ExCA Software Development

4.0 UPGRADABILITY

By having the hardware designed in up front, a system in the field can be upgraded by simply sending out an upgrade disk. The second half of 1992 will see Card Services, MTD's and Microsoft's Flash File System released. How will existing systems make use of all this new software? Indeed how will they even load it? How do endusers get a hold of this new software?

If the OEM chooses to upgrade the BIOS in purchased systems he will need to include some kind of BIOS upgrade utility (typically available from the BIOS vendors) to either append the new code to the existing BIOS or completely replace the existing BIOS. If the OEM chooses to upgrade systems in the field with device drivers, he can simply send out the code on floppy disks with installation instructions.

Case 1: A system with a floppy disk. The system can be upgraded directly from the diskette.

Case 2: A system without a floppy disk. But with a functional card slot, the software can be copied onto an SRAM or any BPB/FAT based memory card in an external reader/writer. The system could then be upgraded to use flash cards directly from the memory card.

Case 3: A system without a floppy disk or functional memory card slot. The system can be upgraded through the serial port or parallel port with any of the after market products like Laplink or Brooklyn Bridge.

Note that when the system is upgraded it's important to delete all older versions of the software. Different rev versions of the software don't necessarily work together (i.e., Socket Services 2.0 won't necessarily work with Card Services 1.0 etc.).

If the OEM has a technical hotline, customers can call in and request or purchase an update package that would include all the software necessary to upgrade the existing systems to use Flashcards.

4.1 Example 1: An OEM Has a Box Today and Wants to Add Flash Card Functionality

If the design follows the layout shown in Figure 1. the OEM can upgrade his system to flash cards by adding two pieces of software and two lines to his config.sys file.

The first piece of software is Microsoft's Flash File System (MS-FLASH.SYS). The second piece of software is a low level driver called iCARDRV1.EXE and is provided by Intel. iCARDRV1 is a modification of the Microsoft Carddrv program which has been converted to work with the recommended ExCA hardware design.

Load both of these files in a directory and call them from the config.sys file with the following lines:

```
Device=C:\exca\iCardrvl.exe /port=3E0
/base=D0 /slots=2 /part=1
Device = C:\exca\MS-FLASH.SYS
```

The iCarddrv command line options are:

"Port" refers to the I/O address of the 82365SL.

"Base" refers to the starting address of the memory window in system memory. "Base" is a 4K value, so "/base=D0" represents a base address of D0000h.

"Slots" refers to the number of PCMCIA sockets installed on the system.

"Part" refers to the number of partitions in the card (the default is 1).

If the design does not utilize the recommended hardware implementation shown in Figure 1, but does use the 82365SL PC card controller, than the OEM can take the Intel iCarddrv source code and modify it to their specific hardware.

If the OEM does not use the 82365SL, then he can use the Microsoft version of Carddrv source code (which is included with MS FFS) and modify it for his specific hardware needs.

4.2 Example 2: An OEM Designing a Box Today that will be in Production Early Next Year

An OEM who is still in the process of designing a PC card based system can plan for the evolving software in the PC card market. A PC still in design should incorporate the following:

1) The recommended ExCA hardware shown in Figure 1.
2) 12V V_{PP} at the PCMCIA socket.
3) A 2 MB Flash BIOS storage device (to allow for upgrades)
4) Microsoft's Flash File System
5) Both Socket and Card Services in the BIOS—Contact your favorite BIOS vendor
6) iCardrvl object code from Intel (available in the PC Card Kit, or the ExCA Hardware Developer's Kit)

5.0 CONCLUSION

Software is easily upgradable, hardware is not. While some of the software pieces are still in development, it is still important to design in the hardware pieces today. Having a V_{PP} capable of supplying 12V, using the 82365SL, and having enough room in the BIOS storage device for the upcoming software will allow a manufacturer to take advantage of today's hardware and tomorrow's software.

Intel's PC Card Kit (order code "PC Card Kit") is the perfect vehicle to assist a card based PC designer. The kit includes the ExCA Developer's Board (block diagram shown in Figure 1), iCardrvl.exe, a 4 MB Series 2 flash memory card, 82365SL diagnostic software, ExCA specification revision 1.1, application notes, datasheets and a User's Guide showing how to install and run the kit in a desktop. For OEM's who already have Series 2 flash memory cards but still want the ExCA Developer's board and associated documentation there is the ExCA Hardware Evaluation Board (order code "ExCAHEBD"), the contents are the same as the PC Card Kit without the Series 2 card.

If an OEM implements the ExCA hardware design shown in Figure 1, they can *check out the flash functionality of their box today* by using Microsoft's Beta version of FFS (available from Microsoft) in conjunction with a Carddrv that has been specifically modified by Intel to handle all of the Carddrv, Card Services, MTD, and Socket Services functions. This Carddrv is available from Intel by contacting your local sales office.

APPENDIX A
WHERE TO GO FOR MORE INFORMATION:

	Order Number
imCF004FLSA,010,020 Flash Memory Card Datasheet	#290434
82365SL PC Card Interface Controller Datasheet	#290423
AP-341 Designing an Updatable BIOS	#292077
Implementing the Integrated Regs of Series 2	#292096
28F002BX Datasheet	#290448
AP-357 Power Supply Solutions	#292092
ExCA Specification	Revision 1.1
PCMCIA Specification	Revision 2.0
PC Card Kit*	PCCARDKIT
ExCA Hardware Developer's Kit**	EXCAHWEBD

Microsoft Flash File System	Call Microsoft at 206-936-9100 (USA)
Connectors:	Foxconn, AMP, DuPont, Molex, ITTCannon, Fujitsu, JAE, DDK, etc.

BIOS vendors working on ExCA Socket Services and Card Services:

AMI	404-246-8612
Award	408-370-7979
Phoenix	408-452-6540
SystemSoft	508-651-0088
Quadtel	714-754-4422

Voltage Converters: Maxim, Linear Tech, etc.

Important phone numbers for card based system designers

PCMCIA	408-720-0107
ExCA	916-351-2562
Microsoft FFS	206-936-9100

NOTES:

*PC CARD KIT—The PC Card Kit contains: 4 MB Series 2 flash card, Carddrv executable code, PCIC diagnostic software, ExCA Developers board, application notes, datasheets, product briefs, ExCA specification, and a kit User's Guide.

**ExCA Hardware Developer's Platform contains: Carddrv executable code, PCIC diagnostic software, ExCA Developers board, application notes, datasheets, product briefs, ExCA specification, and a kit User's Guide.

3

intel®

APPLICATION BRIEF

AB-29

September 1992

Flash Memory Applications in Laser Printers

BRIAN DIPERT
MCD MARKETING APPLICATIONS

COPYRIGHT INTEL CORPORATION, 1992

Order Number 292110-001

Recently, several laser printer and printer peripheral companies have introduced products that incorporate flash memory. Their advertisements validate the unique capabilities and benefits that flash memory features provide. OEM interest, and predictions by market analysts like BIS Strategic Decisions, point to increasing future flash memory usage as laser printer manufacturers continue to differentiate their product lines to meet user needs. This application brief discusses the uses and benefits of flash memory in laser printer designs. Specifically, flash memory usage for system code storage, and for font and "font-like" data storage, will be highlighted.

1.0 INTRODUCTION

Within the computer industry, the laser printer market is one of the most rapidly growing business sectors. The graph of figure 1 shows growth rate of various market segments since 1988, as well as predicted growth through 1996. Laser printer proliferations are expanding to capture the needs of more and more user groups. Simultaneously, more and more computer users are turning to laser printers versus traditional "impact printer" alternatives, as features proliferate, capabilities expand, offices become more automated, and unit prices fall. This combination grows the total **number** of laser printer market segments, as well as the **size** of each segment.

When a large and steadily increasing supply of potential laser printer users exists, the "invisible hands" of economics unequivocally dictate that a large numbers of suppliers will appear to service this demand. What does this all mean to you, the laser printer manufacturer/designer? In a word, **competition**! How can any one company expand (or at a minimum maintain) their market share over the efforts of all others?

3

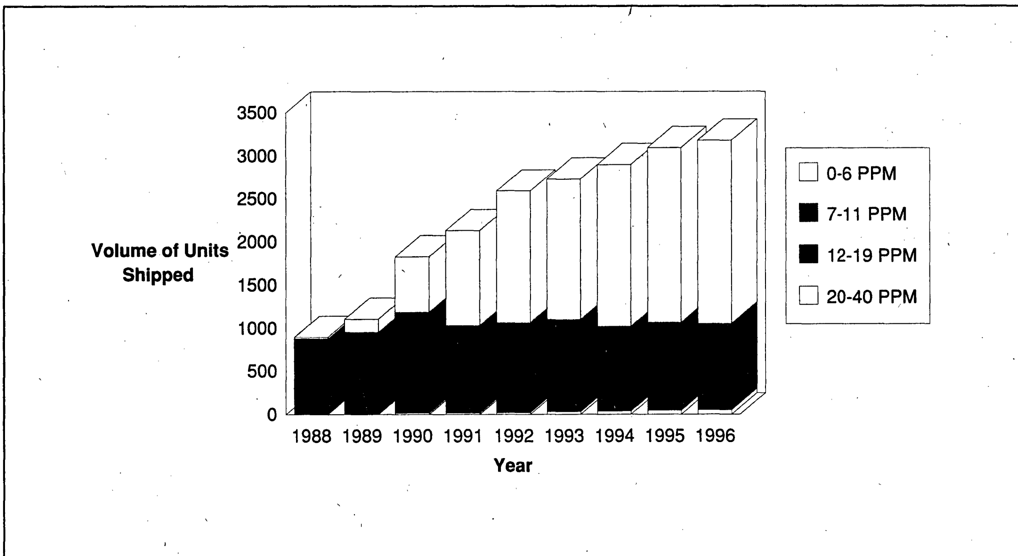


Figure 1. Laser Printer Market Growth 1988-1996 (BIS Strategic Decisions, 1992)

One way to do this is by providing equivalent product features as all other suppliers, at a lower unit price. This is referred to as **pricing-driven demand**. Unfortunately, as in the example of the "clone" market in today's PC industry, pricing-driven demand does not often translate to long-term financial health for your company.

The other method of establishing a leadership image (and the preferable means) is by stimulating **differentiation-based demand** through key product features that uniquely answer market needs. The advantages to this approach are many:

- Product differentiation makes good business **market share** sense. Uniquely meeting customer needs through product features establishes a short-term leadership image and the potential for a long-term protectable market position.
- Value-based pricing also makes good business **profitability** sense. If product features are chosen carefully, the resultant customer demand allows the supplier to price comfortably above per-unit cost, guaranteeing profitability and long term business health!

Flash memory is a key technology whose capabilities trendsetting OEMs are exploiting to differentiate their current and future laser printer designs. It enables clear benefits in system expandability, flexibility, performance and ease-of-use; benefits that translate directly to customer satisfaction and long-term loyalty. Flash memory combines the attributes of nonvolatility (like ROM or EPROM) and in-system updateability (like RAM or a hard disk drive), while simultaneously providing high density and compatible read performance to DRAM. This unique combination of features allows flash memory to not only replace more "traditional" laser printer memory approaches, but also to enhance systems beyond the limitations of these "conservative" memory solutions. The remainder of this application brief will review the major memory subsystems in today's laser printer designs, and how flash memory capabilities can be applied in each case.

2.0 MEMORY USAGE IN LASER PRINTERS

A high-level laser printer block diagram is shown in figure 2. Memory uses in laser printers can be grouped in the following three areas:

- System and PDL (printer description language) emulation code storage
- Font (and font-like data type) storage
- Temporary bitmap image storage and manipulation

Memory is used differently in each case; therefore optimum memory features are similarly specialized, even within a common memory technology. Semiconductor vendors, for example, have optimized various types of DRAMs for the applications in which they'll be used. Similarly, Intel has optimized unique product "families" within its flash memory line to match the requirements of applications like those found in laser printers. These product "families" will be referred to in the discussions below.

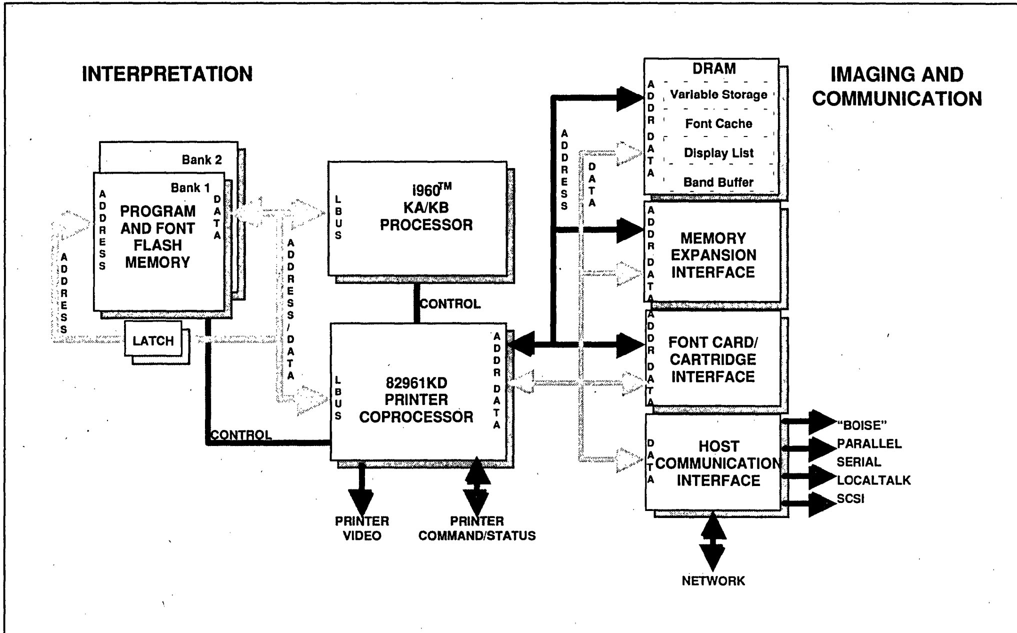


Figure 2. Flash Memory, A Key Element In Leading Edge Laser Printers

2.1 System Code Memory

The system code memory stores the software that is executed by the embedded processor to run and control the laser printer. This includes code to interface the processor with the input/output, coprocessor, print drum and motor subsystems. System code memory also includes a large amount of software devoted to emulating various printer description languages. The most commonly known PDLs are PostScript* (pioneered by Adobe and currently at version 2.0) and PCL* (Printer Control Language, pioneered by Hewlett Packard and currently at version 5). Beyond these two "industry standards", dozens of additional proprietary "languages" have been developed by various hardware and software vendors.

Depending on the complexity and capabilities of the laser printer and the number of supported emulations, system code size varies. It ranges from as little as 256 KBytes for an entry-level personal printer to several MBytes for powerful high-end network laser printers. The traditional memory technology used to store system code is high density ROM (for nonvolatility) shadowed to DRAM on system powerup (for fast read access time).

Flash memory, when used for system code storage, combines the nonvolatility and high density of ROM with the fast read performance of DRAM, making the ROM/DRAM redundancy unnecessary. Additionally, the in-system upgradeability of flash memory makes laser printers flexible and updateable both in the manufacturing line at your factory, and once the system is in the hands of the end customer. Although the per-device cost of flash memory is higher than that of ROM, flash memory's upgradeability can result in a *lower system cost* through a laser printer's operating lifetime.

Flash memory eliminates costly inventory of ROMs, ROMs that must be scrapped if an enhanced software version is released or a software "bug" is discovered. Using flash memory, one hardware design can service multiple markets via simple "end-of-line" programming as the system leaves the factory. Additionally, diagnostic code can be programmed on the assembly line for full system debug, and replaced with the final software version later in the manufacturing flow.

Today, upgradeability once a system reaches the customer's hands is achieved via proprietary, costly add-in ROM cards. These emulation modules connect to the host system through low-performance interface buses. Updating the local code memory in the printer requires a technician visit, is very costly to the customer, and is therefore rarely done. The result can be a less-than-optimized system with subpar performance, and a dissatisfied customer that will not consider your company for his/her next printer purchase!

Flash memory's in-system reprogramming makes system code upgrade as simple as running an "UPDATE" utility on the host computer, and can be done **by the customer** at his/her PC using a diskette sent by the printer manufacturer, or a file downloaded from a OEM computer bulletin board service. Configuring the printer with the exact emulations needed is equally straightforward. Customer service is perhaps the supreme differentiator in multiple-source markets. As companies focus more and more on the customer and the service aspects of their business, they will turn to flash memory as a means of readily and economically achieving their goals.

Intel's BootBlock flash memory product line has been specifically defined to meet the specific requirements of high-density embedded code storage and execution. These devices are also available in ROM-compatible pinouts. This allows printer OEMs to achieve quick time-to-market with rev. 0 software (updateable once initial systems are in customer hands) and later convert their designs to ROM if desired, once final production code is stable. For further information on these products, reference the Additional Information section at the conclusion of this application brief.

2.2 Font Storage

Today's laser printers ship from the factory with a number of "resident" fonts stored in nonvolatile ROM. The density of this memory varies with the end market for the printer. A "Roman" set of resident typefaces requires anywhere from 1-2 MBytes of storage. Japanese "Kanjii" fonts, on the other hand, require upwards of 5 MBytes **per typeface**. A minimum-configured laser printer for the Japanese market therefore needs 10-20 MBytes of resident font memory. Additional permanent font storage is often available through ROM font cartridges, similar to the "emulation" fonts mentioned earlier. Finally, software such as Adobe Type Manager* and Microsoft* TrueType* downloads font information to the printer, storing it in volatile DRAM. This latter temporary font data is lost when the printer is turned off or reset. Resets can occur, for example, each time the printer output jams, or when the paper supply is depleted.

Computer users are more and more outgrowing the capabilities of the resident ROM fonts stored in their laser printers, or available through the ROM expansion font cartridges. ATM and TrueType are enabling these users to customize their documents by varying not only font size and attributes, but also the font typefaces themselves. Many corporations have developed custom fonts for use by their employees for a consistent documentation "look and feel". These unique typefaces are not a part of the resident standard typeface set. Finally, not only fonts, but also graphic bitmaps (corporate logos, bitmapped signatures, etc) and page layout templates are being integrated into desktop-published documents. All of this non-resident information must be repeatedly downloaded from the host computer to the printer DRAM after each printer poweroff or reset. Since this download is accomplished via the serial or parallel connection, print performance is dramatically and negatively impacted, especially noticeable in a networked printer arrangement.

Flash memory, with its unique set of attributes, combines the best qualities of today's font and template storage solutions while incorporating none of their weaknesses. Like DRAM, it is in-system updateable and has comparable per-device cost at higher densities. Like ROM, it is nonvolatile. Like both of these technologies, it is a very dense storage medium, available in sizes up to 1MByte per component, and 20MBytes per card, at the time this application note was written. Where *extremely* high density memory is needed, as in the case of Kanjii font storage, flash memory components have an over 200,000x first read access advantage and an over 14x data transfer performance advantage over hard disk drives. The performance of a printer computing subsystem is significantly hindered by the slow access time of a HDD. Flash memory, with its sub-100ns read speed, is the superior solution.

A resident high-density array of flash memory is coupled directly to the CPU local bus for highest performance. It allows the customer to exactly configure the printer font, bitmap graphic and page template information for his/her specific applications. This data is downloaded to the printer **once**, and from that point on is always available for use, even after the printer is turned off or reset. If expanded printer usage (as in a network environment) requires additional resident "font" storage in the future, easy density upgrade is enabled by designing in a PCMCIA/ExCA™ memory-I/O card socket, again interfacing directly to the embedded processor bus. Plugging in a flash memory card means no printer disassembly is required!

Intel's FlashFile™ flash memory component and Series 2 flash memory card lines combine the high density and high performance required for resident "font" storage. For further information on these products, reference the Additional Information section at the conclusion of this application brief.

2.3 Image Storage and Manipulation

The temporary graphic memory subsystem stores the image to be printed as it is "constructed" by the processor from data provided by the host computer. Optimum characteristics of this memory include full "real-time" bit-level alteration, infinite rewrite capability and fast read/write performance. Nonvolatility is not required in this area of the memory subsystem. Therefore, DRAM will continue to be the memory of choice for temporary image storage.

3.0 SUMMARY

This application brief has discussed the various memory subsystems in today's laser printers, and their operating characteristics. Flash memory is an exciting new approach that offers the very real potential to significantly improve your next-generation laser printer designs. Its capabilities are superior to traditional solutions in the system code and font memory areas, and enable laser printers that are more expandable, more flexible, higher performance and easier to use than ever before. The end result is a satisfied customer, a customer that will choose your product over a competitor's, and a customer that will remain loyal to your company far into the future.

ADDITIONAL INFORMATION

For additional information on the Intel flash memory products mentioned in this article, please reference the following documents, available through your local Intel sales representative.

BootBlock Components	Order Number
28F001BX Datasheet	290406
28F200BX/28F002BX Datasheet	290448
28F400BX/28F004BX Datasheet	290451
ER-26 "The Intel 28F001BX-T and 28F001BX-B Flash Memories"	294010
ER-29 "The Intel 2/4 Mbit BootBlock Flash Memory Family"	294013

FlashFile Components	Order Number
28F008SA Datasheet	290429
AP-359 "28F008SA Hardware Interfacing"	292094
AP-360 "28F008SA Software Drivers"	292095
AP-364 "28F008SA Automation and Algorithms"	292099
ER-27 "The Intel 28F008SA Flash Memory"	294011

FlashFile Series 2 Cards	Order Number
Series 2 Flash Memory Card Datasheet	290434
AP-361 "Implementing the Integrated Registers of the Series 2 Flash Memory Card"	292096

General Flash Information	Order Number
AP-357 "Power Supply Solutions for Flash Memory"	292092
ER-20 "ETOX™ II Flash Memory Technology"	294005
ER-28 "ETOX™ III Flash Memory Technology"	294012

Microsoft and TrueType are trademarks of Microsoft Corporation. Adobe Type Manager and PostScript are trademarks of Adobe Systems Incorporated. PCL is a trademark of Hewlett Packard Corporation. ETOX, FlashFile and ExCA are trademarks of Intel Corporation.

intel®

ENGINEERING REPORT

ER-20

September 1991

ETOX™II Flash Memory Technology

JASON ZILLER
PRODUCT ENGINEERING

Order Number: 294005-006

3

ETOX™ II FLASH MEMORY TECHNOLOGY

CONTENTS

PAGE

INTRODUCTION 3-525

ETOX II FLASH MEMORY CELL 3-525

MEMORY ARRAY CONSIDERATIONS 3-525

ETOX II FLASH MEMORY RELIABILITY 3-525

SUMMARY 3-526

INTRODUCTION

Intel's ETOX™ II (EPROM tunnel oxide) flash memory technology is derived from the CHMOS** III-E EPROM technology. It replaces ultraviolet erasability with a non-volatile memory cell that is electrically erasable in bulk array form. Intel flash memory combines the EPROM programming mechanism with EEPROM erase, producing a versatile memory device that is highly reliable and cost effective. This report describes the fundamentals of the ETOX II flash memory cell in comparison to the standard EPROM, and gives insight into its operation in a system environment.

The ETOX II flash memory cell is nearly identical in size to CHMOS III-E EPROM. This allows comparable densities. The primary difference between ETOX II flash memory and EPROM cells is the flash memory cell's thinner gate oxide, which permits the electrical erase capability. (See Photo 1.)

ETOX II FLASH MEMORY CELL

Intel's ETOX II flash memory cell is composed of a single transistor with a floating gate for charge storage, like the traditional EPROM. (See Figure 1.) In contrast, conventional two-transistor EEPROM cells are typically much larger. Intel produces ETOX II flash memory devices on 1.0μ photolithography.

The ETOX II cell's programming mechanism is identical to the EPROM; that is, hot channel electron injection. The device programming mode forces the cell's control gate and drain to a high voltage while leaving the source grounded. The high drain voltage generates "hot" electrons that are swept across the channel. These hot electrons collide with other atoms along the way, creating even more free electrons. Meanwhile, the high voltage on the control gate attracts these free electrons across the lower gate oxide into the floating gate, where they are trapped. (See Figure 2.) Typically, this process takes less than 10 μs.

Flash memory's advantage over EPROM is electrical erasure, discharging the floating gate without ultraviolet light exposure. The erase mechanism is an EEPROM adaptation which uses "Fowler-Nordheim"[1] tunneling. A high electric field across the lower gate oxide pulls electrons off the floating gate. The erase mode routes the same external voltage used for programming to the source of the memory cell, while the gate is grounded and the drain is left disconnected. (Figure 3.)

MEMORY ARRAY CONSIDERATIONS

The ETOX II flash memory cells have the same array configuration as standard EPROM, thereby matching EPROM in density. Also, identical peripheral circuitry for normal access achieves the same read performance as the Intel CHMOS III-E EPROMs.

Intel flash memory's programming circuitry is also identical to Intel's EPROM designs. Row decoders drive the selected wordline to high voltage, while input data combined with column decoders determine the number of bitlines that are gated to high voltage. This provides the same byte programmability as an EPROM. Intel Flash Memories offer the efficient Quick-Pulse Programming algorithm that is featured on advanced EPROMs.

Array erase is unique to flash memory technology. Unlike conventional EEPROMs, which use a select transistor for individual byte erase control, flash memories achieve much higher density with single transistor cells. Therefore, the erase mode supplies high voltage to the sources of every cell simultaneously, performing a full array erasure. A programming operation must be performed before every erase to equalize the amount of charge on each cell. Then Intel's Quick-Erase™ algorithm intelligently erases the array down to the appropriate minimum threshold level required to read all "ones" data. This procedure ensures a tight distribution of erased cell thresholds throughout the array.

ETOX™ II FLASH MEMORY RELIABILITY

The reliability of Intel's CHMOS ETOX II flash memory process is equivalent to its sister EPROM technology. The ETOX II and EPROM processes share the same data retention characteristics. Preliminary qualification data shows that 1 Megabit flash memories produced on the ETOX II process provide at least 10,000 program and erase cycles (typical 100,000) with no cycling failures due to oxide stress or breakdown. This extended cycling capability is attributed to improvements in tunnel oxide processing and advantages inherent in the ETOX II cell approach.

[1]M. Lenzlinger, E.H. Snow, "Fowler-Nordheim Tunneling into Thermally Grown SiO2," Journal of Applied Physics, Vol. 40 (1969), p. 278.
*Intel's ETOX II flash memory process has patents pending.
**CHMOS is a patented process of Intel Corporation.

SUMMARY

ETOX II flash memory technology is the optimal combination of EPROM and EEPROM technologies. Intel's new ETOX II flash memory process offers extended cycling capability with the density and manufacturability of EPROMs. From an application standpoint, flash memory technology provides the capability to improve overall system quality throughout the product development and manufacturing stages. Also, flash memory density is ideally suited for applications requiring version updates of entire programs which, in turn, suit the "flash" characteristics of erasing the entire array at once. In addition, individual byte programming allows for data acquisition. Flash memory devices produce on the ETOX II process provide a high density, low cost solution to many system memory storage requirements which were previously unavailable.

Table I

	EPROM	ETOX II Flash Memory	EEPROM
Normalized Cell Size	1.0	1.2–1.3	3.0
Programming:			
Mechanism	Hot Electron Injection	Hot Electron Injection	Tunneling
Resolution	Byte	Byte	Byte
Typ. Time	< 100 μs	< 10 μs	5 ms
Erase:			
Mechanism	UV Light	Tunneling	Tunneling
Resolution	Bulk Array	Bulk Array	Byte
Typ. Time	20 Min.	< 1 Sec.	5 ms

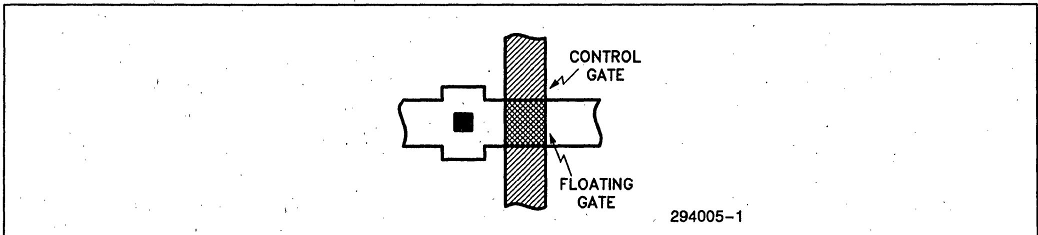


Figure 1. ETOX II Flash Memory Cell Layout (Top View)

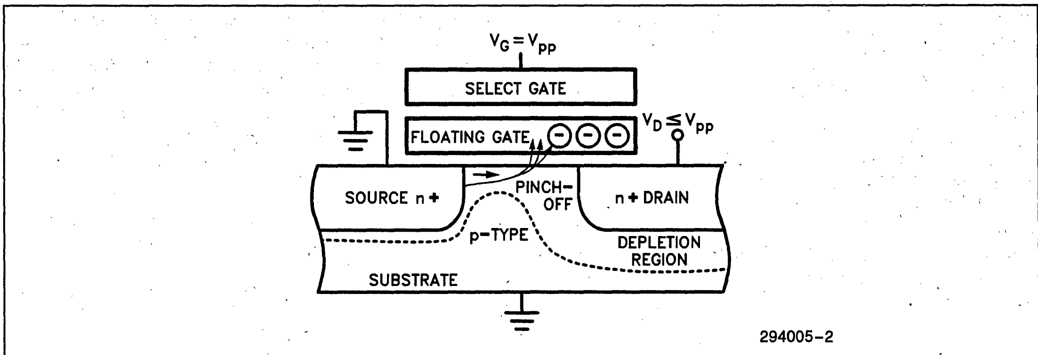


Figure 2. ETOX II Flash Memory Cell during Programming (Side View)

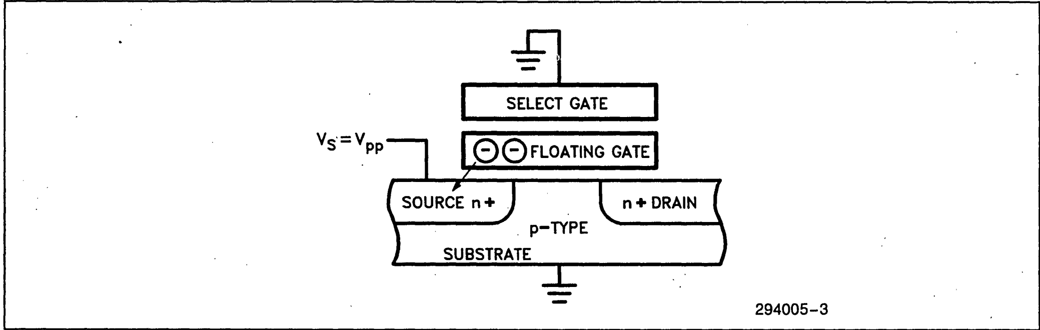
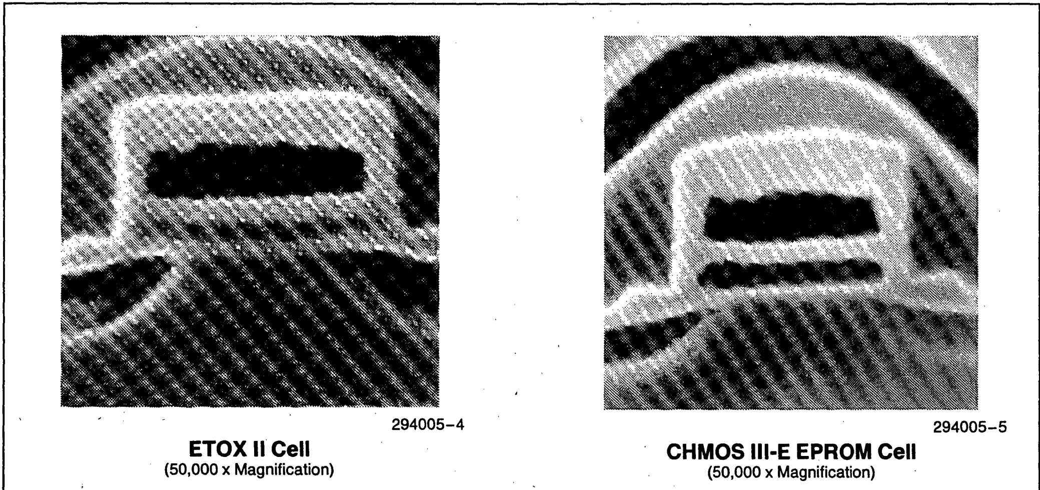


Figure 3. ETOX II Flash Memory Cell during Erase (Side View)

PHOTO 1



ETOX II Cell
(50,000 x Magnification)

CHMOS III-E EPROM Cell
(50,000 x Magnification)

intel®

ENGINEERING REPORT

ER-24

August 1992

Intel Flash Memory 28F256A 28F512 28F010 28F020

Order Number: 294008-006

Intel Flash Memory
28F256A, 28F512, 28F010, 28F020

CONTENTS PAGE

INTRODUCTION 3-530

TECHNOLOGY OVERVIEW 3-530

DEVICE ARCHITECTURE 3-530

DEVICE RELIABILITY 3-531

SUMMARY 3-532

3

INTRODUCTION

Intel's ETOX™ II (EPROM tunnel oxide) Flash Memory adds electrical chip erasure and reprogramming to EPROM non-volatility and ease of use. Advances in tunnel oxides and photolithography have made it possible to develop a double-polysilicon single-transistor read/write random access nonvolatile memory, capable of greater than 10,000 reprogramming cycles (typical 100,000). Intel Flash Memory electrically erases all bits in the array matrix via electron tunneling. The EPROM programming mechanism of hot electron injection is employed for electrical byte programming.

A command port interface, internal margin voltage generation, power up/down protection and address and data latches augment standard EPROM circuitry to optimize Intel's Flash Memory for microprocessor-controlled reprogramming.

Read timing parameters on Intel's 28F256A, 28F512, 28F010 and 28F020 are equivalent to those of CMOS EPROMs, EEPROMs, and SRAMs. The 90 ns, 120 ns and 150 ns access times result from a memory cell current of approximately 50 μA, low resistance poly-silicide wordlines, advanced scaled periphery transistors, and an optimized data-out buffer.

TECHNOLOGY OVERVIEW

Intel's ETOX II flash memory technology is derived from its standard CMOS EPROM process base. Using advanced 1.0 μm double-polysilicon n-well CMOS technology, Intel Flash Memory employs a 3.8 μm x 4.0 μm single transistor cell, affording equivalent array density as comparable EPROM technology. The flash memory cell structure is identical to the EPROM structure, except for the thinner gate (tunnel) oxide. Figure 1 compares the flash memory cell to the EPROM cell.

High quality tunnel oxide under the single floating polysilicon gate promotes electrical erasure. All cells in the array are simultaneously erased via Fowler-Nordheim tunneling. Applying 12V on the source junctions and grounding the select gates erases the entire array in one second (typical). Programming is accomplished with the standard EPROM mechanism of hot electron injection from the cell drain junction to the floating gate. Programming is initiated by bringing both the select gate and the cell drain to high voltage. Programming occurs at a rate of 10 μs pulses per byte.

DEVICE ARCHITECTURE

Command Port

One feature which differentiates Intel's Flash Memory is the command port architecture, illustrated in Figure 2.

The command port simplifies microprocessor control of the erase, erase verify, program, program verify, and read operations, without the need for additional control pins or the multiplexing of high voltage with control functions. On-chip address and data latches minimize system interface logic and free the system bus during erase and program operations. High voltage (12V) on the V_{PP} pin enables the command port. In the absence of this high voltage, the command port defaults to the read operation, inhibiting erasure or programming of the device.

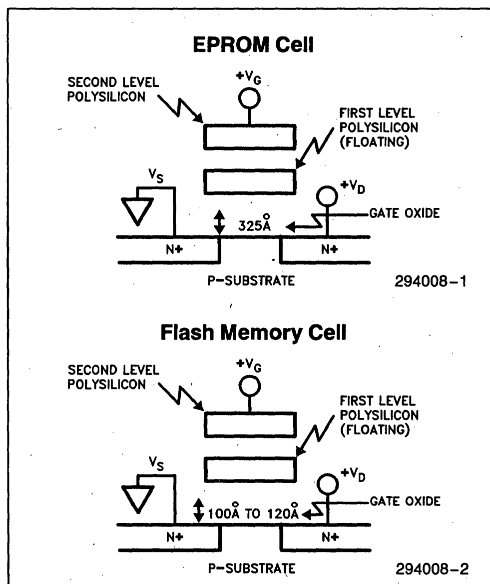


Figure 1. EPROM Cell vs. Flash Memory Cell

The command port consists of a command register, command decoder and state latch, the data-in latch, and the address latch. The command decoder output directs the operation of the high voltage flash-erase switch, program voltage generator, and the erase/program verify voltage generator.

Functions are selected via the command port in a microprocessor write cycle controlled by the Chip-Enable and Write-Enable pins. Contents of the address latch are updated on the falling edge of Write-Enable. The rising edge of Write-Enable latches the command and data registers, and initiates operations.

Erasure

Erasure is achieved through a two-step write sequence. The erase set-up code is written to the command register in the first cycle. The erase confirmation code is written in the second cycle. The rising edge of this second Write-Enable pulse initiates the erase operation. The command decoder triggers the high voltage flash-erase switch, connecting the 12V supply to the source of all bits in the array, while all wordlines are grounded. Fowler-Nordheim tunneling results in the simultaneous erasure of all bits.

The array source switch, shown in Figure 3, switches high voltage onto the source junctions. During erasure, the high voltage latch formed by M_5 through M_8 enables transistor M_{15}. Transistor M_{15} pulls the array source up to 12V. Transistor M_{16} pulls the source to ground during read and program operations.

To obtain fast erase times, the device must supply the grounded gate breakdown current which occurs on the sources of the memory array. The upper boundary for current sourcing capability of M_{15} is set by the maximum allowable substrate current. If V_{PP} is raised to 12V before V_{CC} is above approximately 1.8V, the low V_{CC} detect circuit formed by transistors M_1 to M_4 drives the node LOW V_{CC} to 9V. Transistors M_9 to M_{11} then force the erase circuit into a non-erase state with M_{15} off and M_{16} on. When V_{CC} rises above 1.8V, the chip will be reset into the read state.

Writing the erase verify code into the command register terminates erasure, latches the address of the byte to verify, and sets the internally-generated erase margin voltage. The microprocessor then accesses the output from the addressed byte using standard read timings. The verify procedure repeats for all addresses. Should a byte require more time to reach the erased state, another erase operation is applied. The erase and verify operations continue until the entire array is erased.

Programming

Programming follows a similar flow. The program set-up command is written to the command register on the first cycle. The second cycle loads the address and data latches. The rising edge of the second Write-Enable pulse initiates programming by applying high voltage to the gates and drains of the bits to be programmed.

Writing the program verify command to the register terminates the programming operation and applies the program verify voltage to the newly programmed byte. Again, the addressed byte can be read using standard microprocessor read timings. Should the addressed byte require more time to reach the programmed state, the programming operation and verification are repeated until the byte is programmed.

DEVICE RELIABILITY

Cell Margining

Erase and program verification ensure the data retention of the newly altered memory bits. The cell margining performed in the Quick-Pulse Programming and Quick-Erase algorithms is more reliable than historical overpulsing schemes as margining tests the amount of charge stored on the floating gate.

Intel's 28F256A through 28F020 Flash Memories employ a unique circuit to internally generate the erase and program verify voltages. Figure 4 shows a simplified version of the circuit. The circuit consists of a high voltage switch and the verify voltage generator. Transistors M_1 through M_4 constitute the high voltage switch which disconnects V_{PP} from the resistor when the device is not in the verify mode. The verify voltage generator includes a resistor divider and a buffer. Internal margin voltage generation maintains microprocessor compatibility by eliminating the need for external reference voltages.

Erase/Program Cycling

One of the most significant aspects of Intel Flash Memory is its capability for a minimum of 10,000 erase/program cycles (typical 100,000). Destructive oxide breakdown has been a limiting factor in extended cycling of thin oxide EEPROMs. Intel's ETOX II flash memory technology extends cycling performance through: improved tunnel oxide processing that increases charge carrying capability ten-fold; reduced oxide area under stress minimizing probability of oxide defects in the region; and reduced oxide stress due to a lower peak electric field (lower erase voltage than EEPROM).

A typical cell erase/program margin (Vt) is shown as a function of reprogramming cycles in Figure 5. After 10,000 reprogramming cycles, a 2.5V program read margin exists, ensuring reliable data retention. Accelerated retention bake experiments, for devices cycled 10,000 times, show minimal program Vt shift.

Reliable erase/program cycling also requires proper selection of the erase Vt maximum and maintenance of a tight Vt distribution. The maximum erased Vt is set to 3.2V via the erase algorithm and the internal erase verify circuits. Superior oxide quality gives an erased Vt distribution width that improves slightly with cycling (Figure 6). The tight erase Vt distribution gives an order of magnitude of erase time margin to the fastest erasing cell (Figure 7).

SUMMARY

Intel's ETOX II flash memory technology is a breakthrough in adding electrical chip-erasure to high-density EPROM technology. Intel's CMOS Flash Memory offers the most cost-effective and reliable alternative for read/write random access non-volatile memory. Microprocessor-compatible specifications, straightforward interfacing, and in circuit alterability allow designers to easily augment memory flexibility and satisfy the need for nonvolatile storage in today's designs.

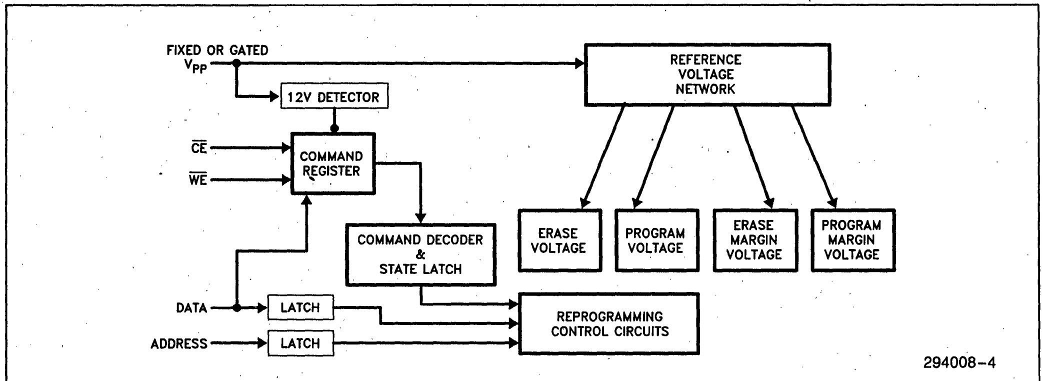


Figure 2. Command Port Block Diagram

294008-5

Figure 3. Array Source Switch

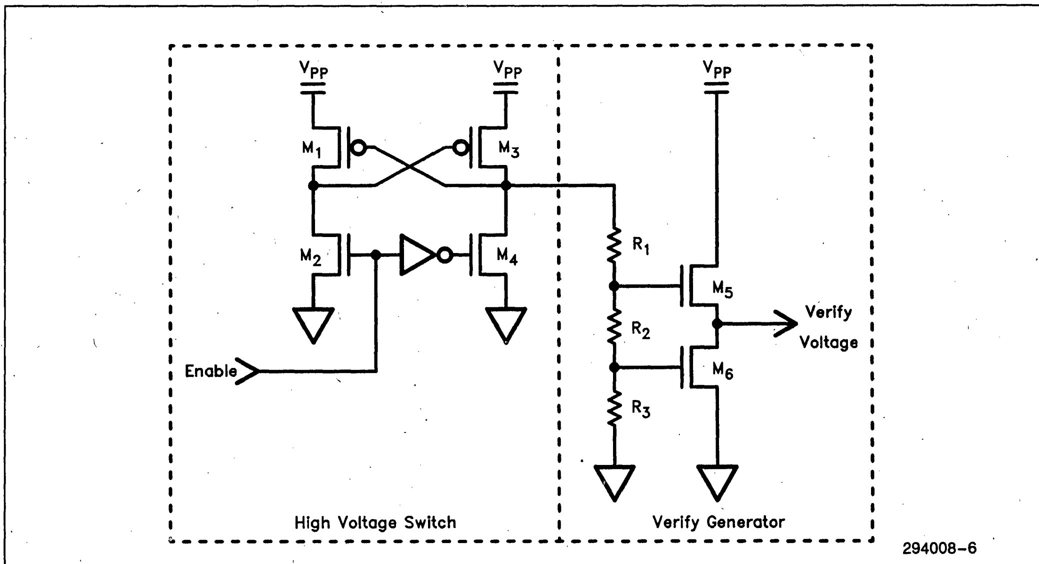


Figure 4. Erase/Program Verify Generator

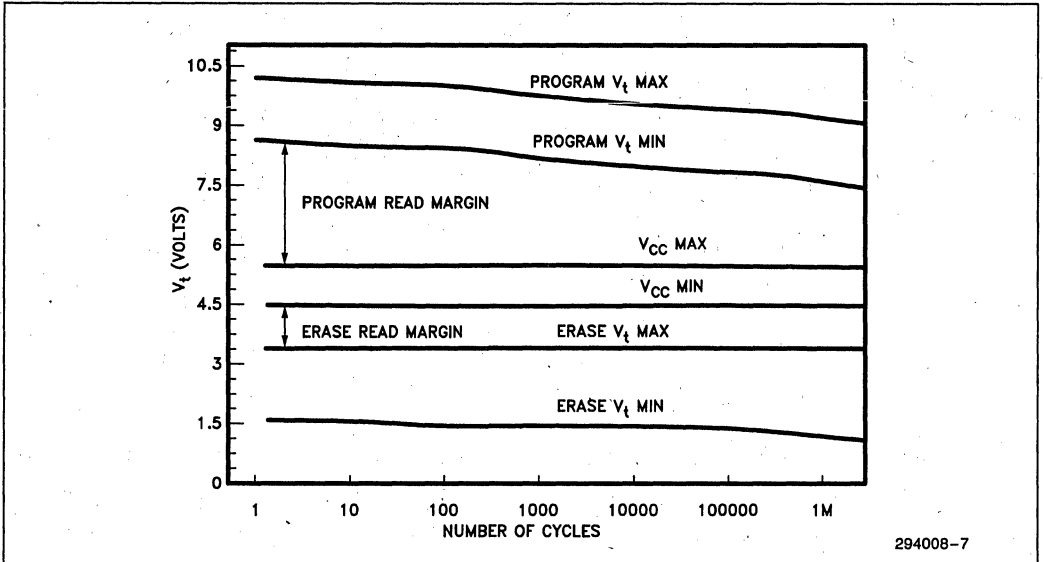


Figure 5. 1M Array V_t vs Cycles

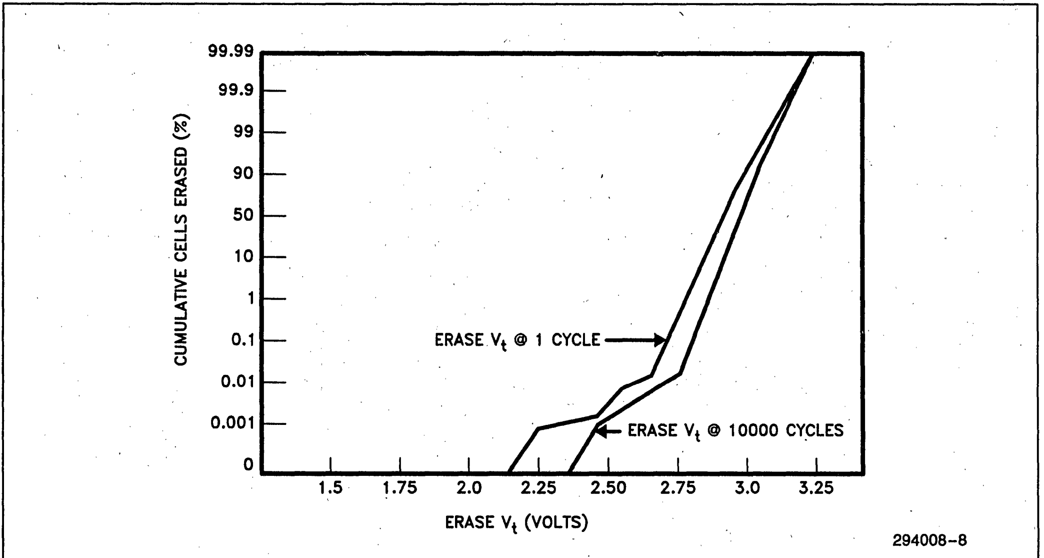


Figure 6. Erase V_t Distribution vs Cycling

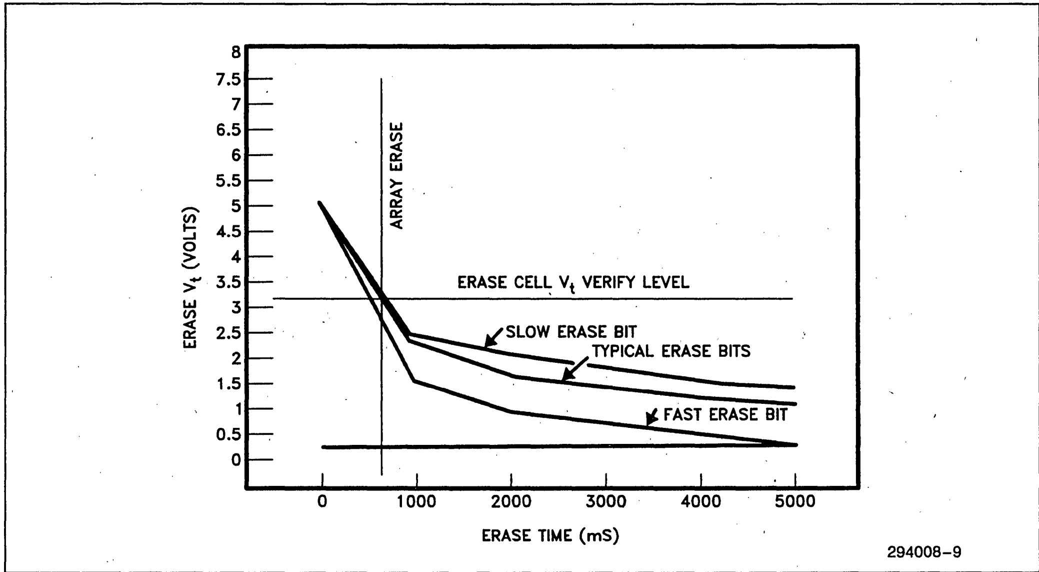


Figure 7. Array Erase V_t vs Erase Time

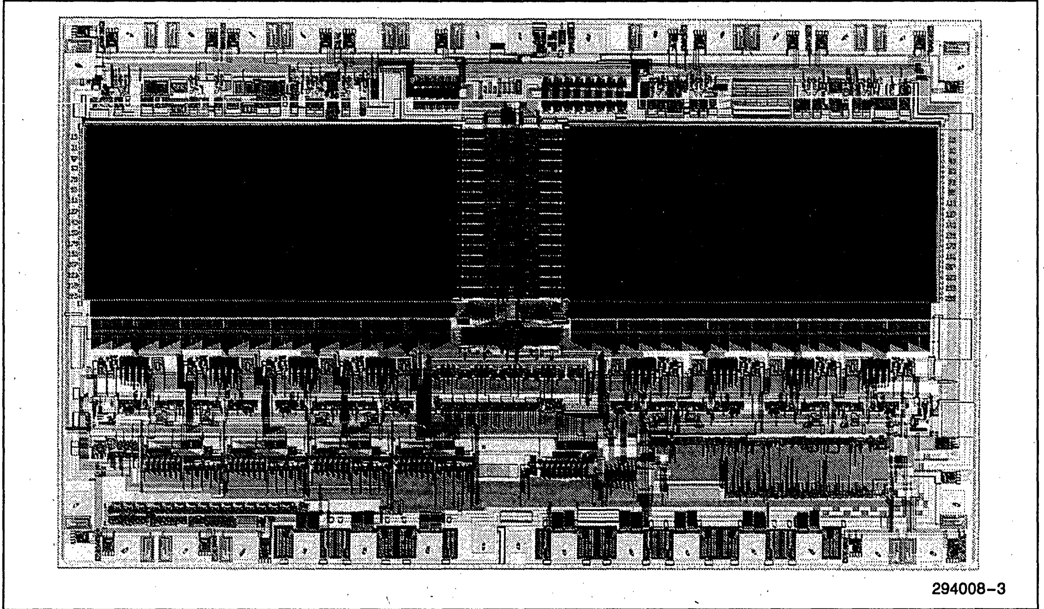
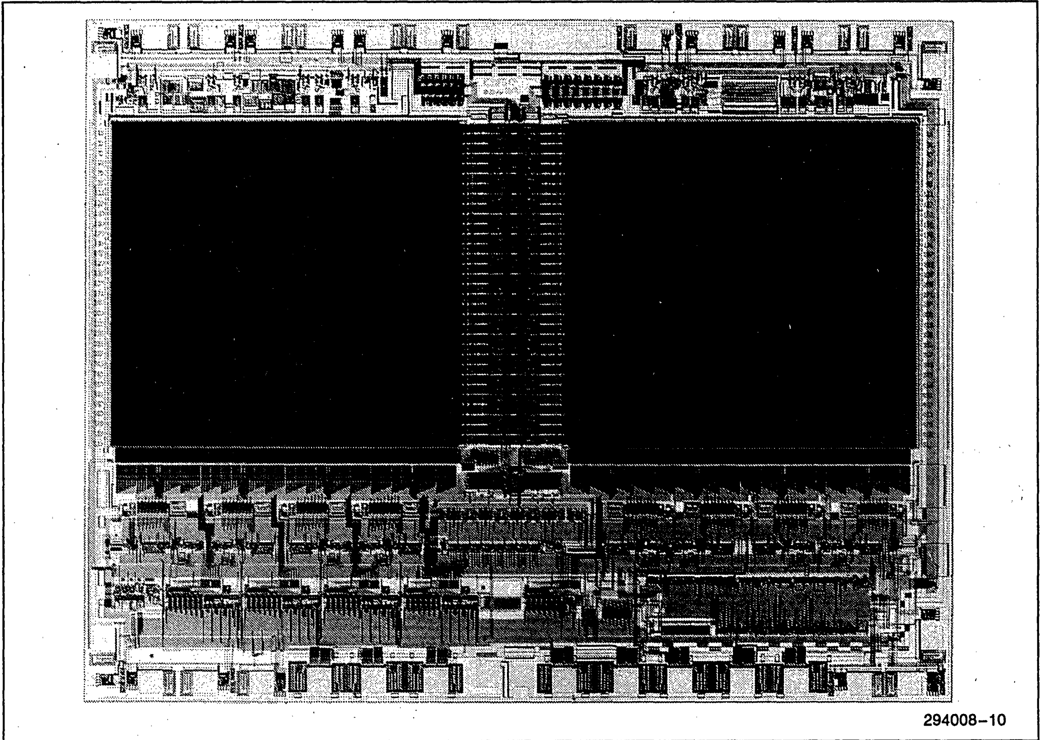
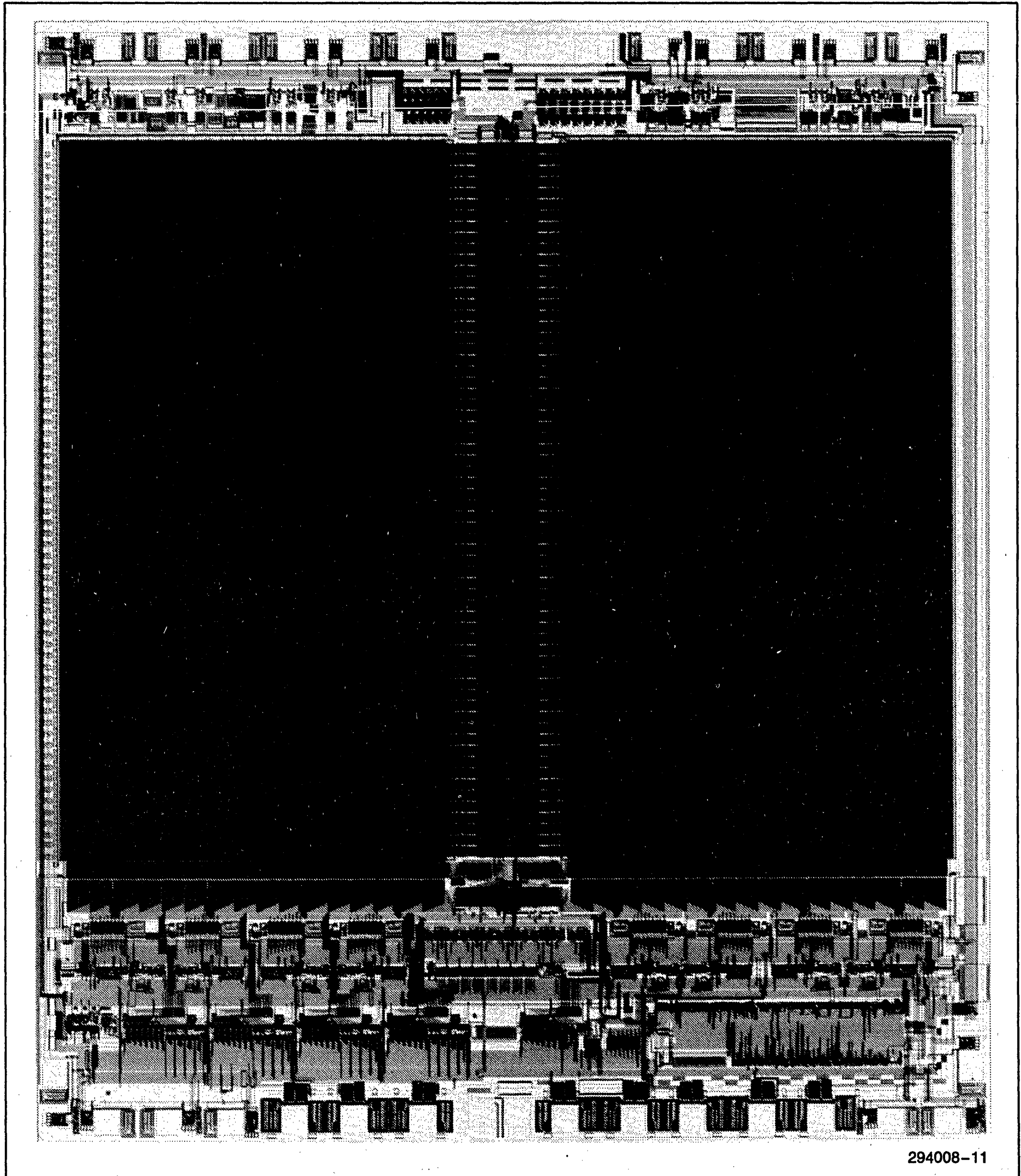


Figure 8. 28F256A Die Photograph



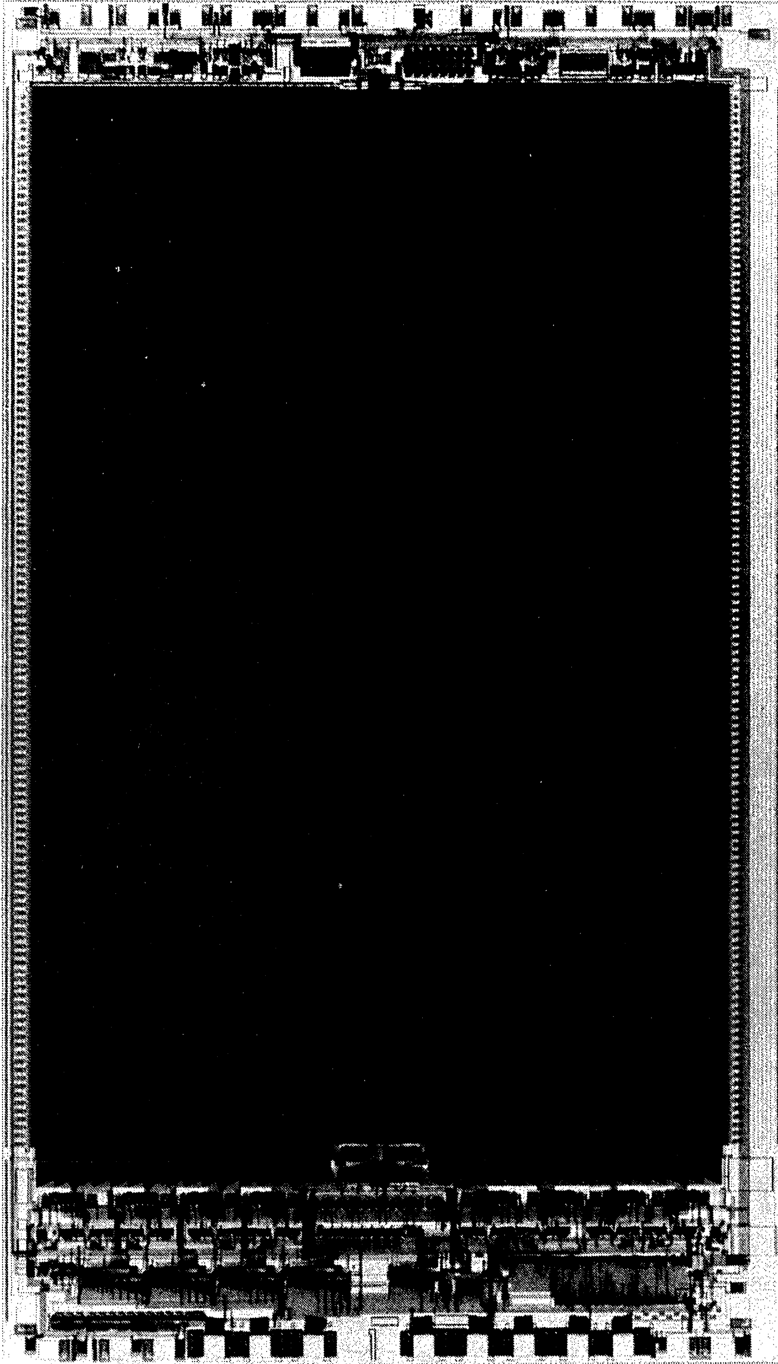
294008-10

Figure 9. 28F512 Die Photograph



294008-11

Figure 10. 28F010 Die Photograph



294008-12

Figure 11. 28F020 Die Photograph

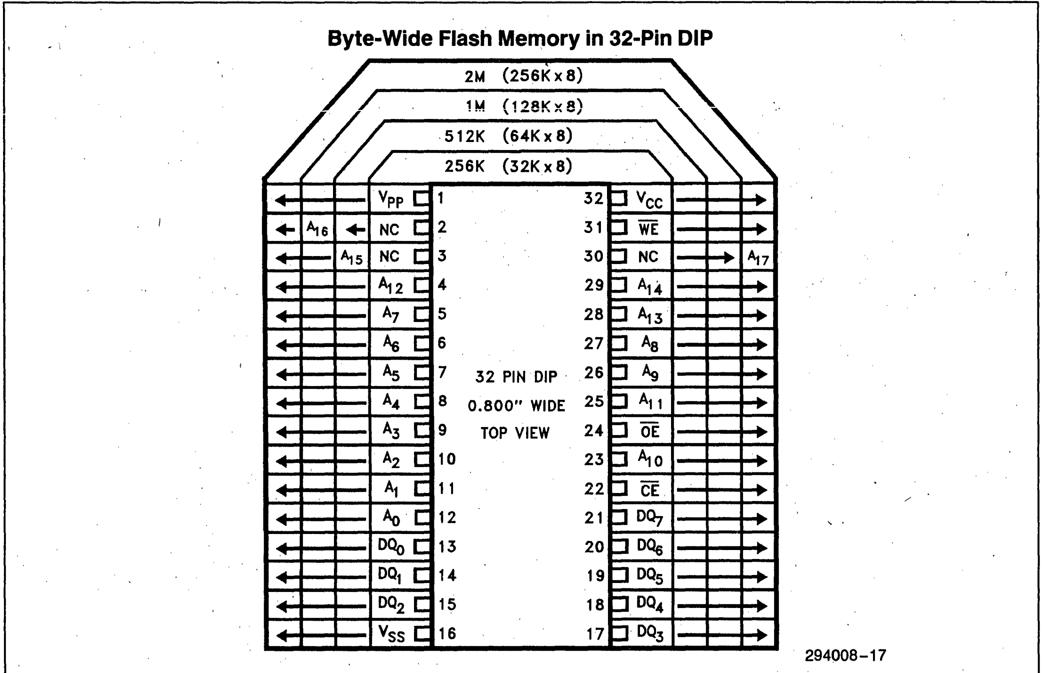


Figure 12. Flash Memory Pinouts

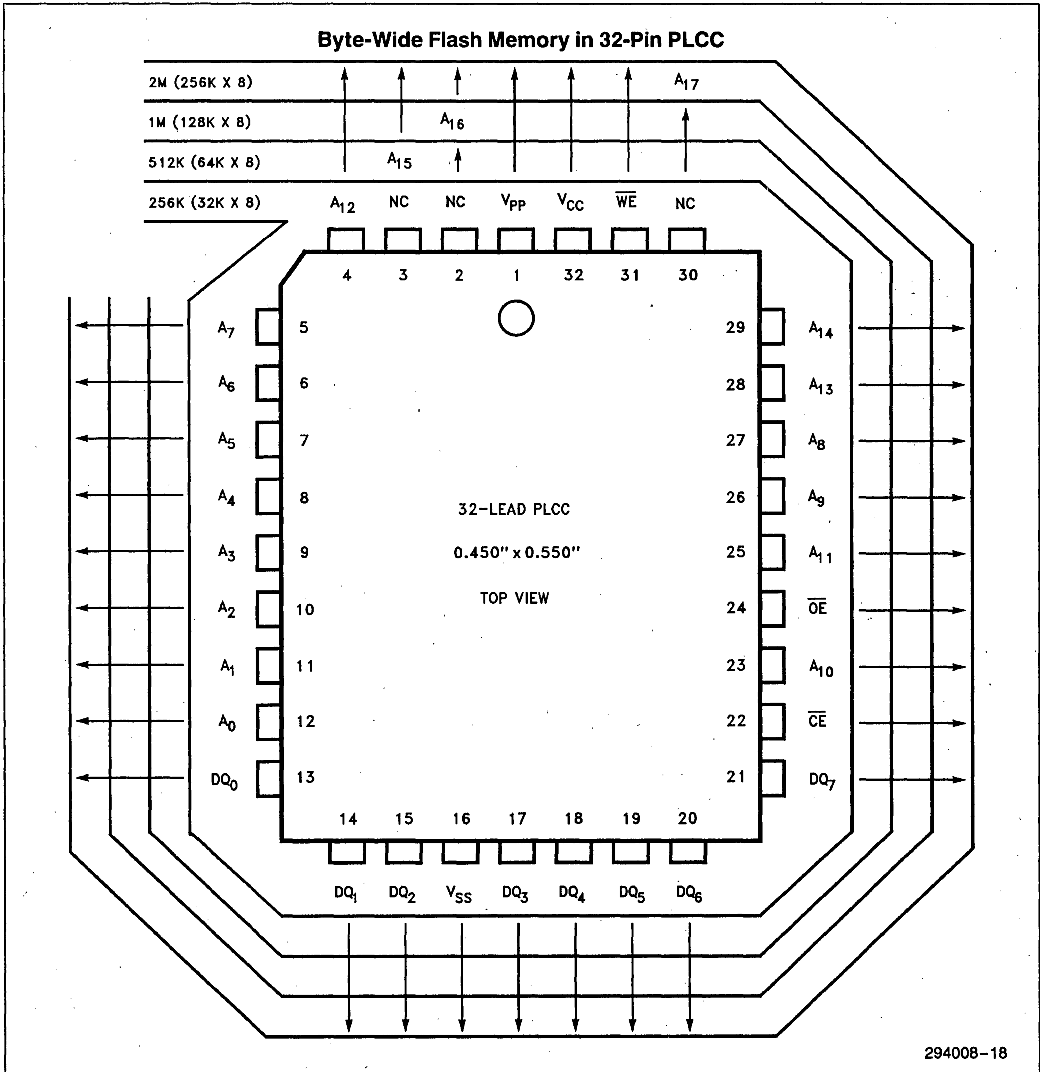


Figure 13. Flash Memory Pinouts

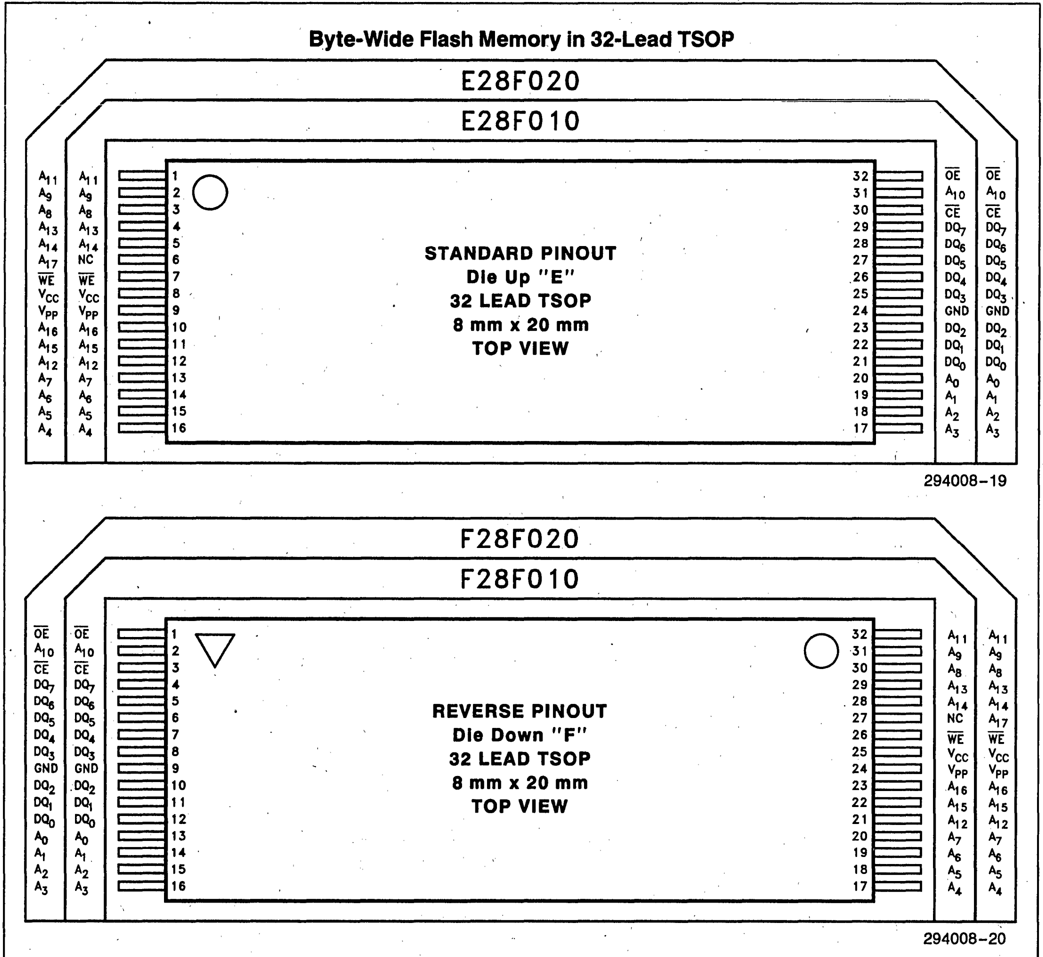


Figure 14. Flash Memory Pinouts

Columns are number 0 through 511 beginning with the column nearest the X-decoder.
Outputs are grouped as follows:

Array Organization:							Left Half Array IO_0 IO_1 IO_2 IO_3 $BL_{384} \leftarrow BL_0$		Right Half Array IO_4 IO_5 IO_6 IO_7 $BL_0 \rightarrow BL_{384}$	
Address							Bitlines			
A_{14}	A_{12}	A_{10}	A_2	A_1	A_0	A_3	IO_0 & IO_7	IO_1 & IO_6	IO_2 & IO_5	IO_3 & IO_4
0	0	0	0	0	0	0	BL_{384}	BL_{256}	BL_{128}	BL_0
0	0	0	0	0	0	1	BL_{385}	BL_{257}	BL_{129}	BL_1
0	0	0	0	0	1	0	BL_{386}	BL_{258}	BL_{130}	BL_2
0	0	0	0	0	1	1	BL_{387}	BL_{259}	BL_{131}	BL_3
0	0	0	0	1	0	0	BL_{388}	BL_{260}	BL_{132}	BL_4
0	0	0	0	1	0	1	BL_{389}	BL_{261}	BL_{133}	BL_5
0	0	0	0	1	1	0	BL_{390}	BL_{262}	BL_{134}	BL_6
0	0	0	0	1	1	1	BL_{391}	BL_{263}	BL_{135}	BL_7
•	•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	0	0	BL_{508}	BL_{380}	BL_{252}	BL_{124}
1	1	1	1	1	0	1	BL_{509}	BL_{381}	BL_{253}	BL_{125}
1	1	1	1	1	1	0	BL_{510}	BL_{382}	BL_{254}	BL_{126}
1	1	1	1	1	1	1	BL_{511}	BL_{383}	BL_{255}	BL_{127}

Figure 15. 28F256A Bitline Decoding

X Address								Row
A_7	A_6	A_5	A_4	A_{13}	A_{11}	A_9	A_8	WL
0	0	0	0	0	0	0	0	XL_0
0	0	0	0	0	0	0	1	XL_1
0	0	0	0	0	0	1	0	XL_2
0	0	0	0	0	0	1	1	XL_3
0	0	0	0	0	1	0	0	XL_4
0	0	0	0	0	1	0	1	XL_5
0	0	0	0	0	1	1	0	XL_6
0	0	0	0	0	1	1	1	XL_7
0	0	0	0	1	0	0	0	XL_8
0	0	0	0	1	0	0	1	XL_9
0	0	0	0	1	0	1	0	XL_{10}
0	0	0	0	1	0	1	1	XL_{11}
0	0	0	0	1	1	0	0	XL_{12}
0	0	0	0	1	1	0	1	XL_{13}
0	0	0	0	1	1	1	0	XL_{14}
0	0	0	0	1	1	1	1	XL_{15}
0	0	0	1	1	1	1	1	XL_{16}
0	0	0	1	1	1	1	0	XL_{17}
0	0	0	1	1	1	0	1	XL_{18}
0	0	0	1	1	1	0	0	XL_{19}
0	0	0	1	1	0	1	1	XL_{20}
0	0	0	1	1	0	1	0	XL_{21}
0	0	0	1	1	0	0	1	XL_{22}
0	0	0	1	1	0	0	0	XL_{23}
0	0	0	1	0	1	1	1	XL_{24}
0	0	0	1	0	1	1	0	XL_{25}
0	0	0	1	0	1	0	1	XL_{26}
0	0	0	1	0	1	0	0	XL_{27}
0	0	0	1	0	0	1	1	XL_{28}
0	0	0	1	0	0	1	0	XL_{29}
0	0	0	1	0	0	0	1	XL_{30}
0	0	0	1	0	0	0	0	XL_{31}

Figure 16. 28F256A Wordline Decoding

X Address								Row
A_7	A_6	A_5	A_4	A_{13}	A_{11}	A_9	A_8	WL
0 • 0	0 • 0	1 • 1	0 • 0	0 • 1	0 • 1	0 • 1	0 • 1	XL32 ••• XL47
0 • 0	0 • 0	1 • 1	1 • 1	1 • 0	1 • 0	1 • 0	1 • 0	XL48 ••• XL63
0 • 0	1 • 1	0 • 0	0 • 0	0 • 1	0 • 1	0 • 1	0 • 1	XL64 ••• XL79
0 • 0	1 • 1	0 • 0	1 • 1	1 • 0	1 • 0	1 • 0	1 • 0	XL80 ••• XL95
1 • 1	1 • 1	1 • 1	0 • 0	0 • 1	0 • 1	0 • 1	0 • 1	XL234 ••• XL249
1 • 1	1 • 1	1 • 1	1 • 1	1 • 0	1 • 0	1 • 0	1 • 0	XL250 ••• XL255

Figure 16. 28F256A Wordline Decoding (Continued)

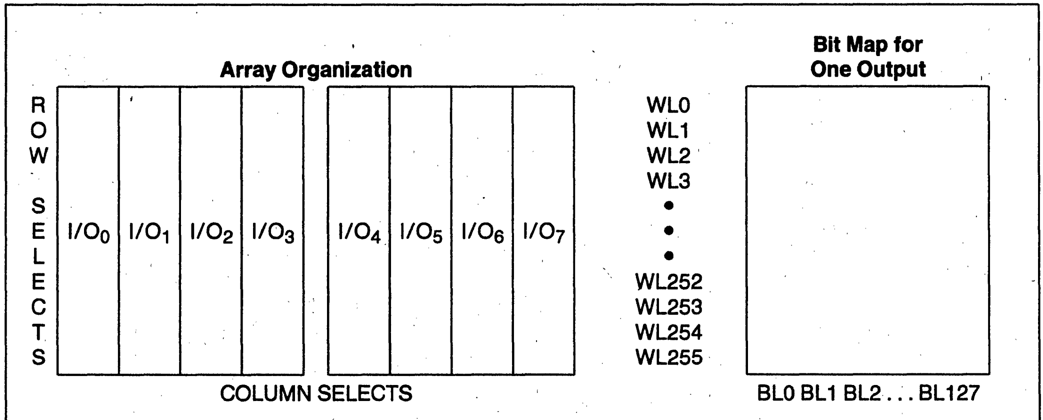


Figure 17. 28F256A Bit Map

Columns are numbered 0 through 511 beginning with the column nearest the X-decoder.

Outputs are grouped as follows:

							Left Half Array O0 O1	O2 O3	Right Half Array O4 O5	O6 O7
Address							Bitlines			
A14	A15	A3	A10	A2	A1	A0	IO0/7	IO1/06	IO2/05	IO3/04
0	0	0	0	0	0	0	BL384	BL256	BL128	BL0
0	0	1	0	0	0	0	BL385	BL257	BL129	BL1
0	0	0	0	0	0	1	BL386	BL258	BL130	BL2
0	0	1	0	0	0	1	BL387	BL259	BL131	BL3
0	0	0	0	0	1	0	BL388	BL260	BL132	BL4
0	0	1	0	0	1	0	BL389	BL261	BL133	BL5
0	0	0	0	0	1	1	BL390	BL262	BL134	BL6
0	0	1	0	0	1	1	BL391	BL263	BL135	BL7
•	•	•	•	•	•	•	•	•	•	•
1	1	0	1	1	1	0	BL508	BL380	BL252	BL124
1	1	1	1	1	1	0	BL509	BL381	BL253	BL125
1	1	0	1	1	1	1	BL510	BL382	BL254	BL126
1	1	1	1	1	1	1	BL511	BL383	BL255	BL127

Figure 18. 28F512 Bitline Decoding

X-DECODING: Wordlines are numbered 0 through 511 beginning at the top of the array.

X Address									Row
A12	A7	A6	A5	A4	A13	A11	A9	A8	WL
0	0	0	0	0	0	0	0	0	XL0
0	0	0	0	0	0	0	0	1	XL1
0	0	0	0	0	0	0	1	0	XL2
0	0	0	0	0	0	0	1	1	XL3
0	0	0	0	0	0	1	0	0	XL4
0	0	0	0	0	0	1	0	1	XL5
0	0	0	0	0	0	1	1	0	XL6
0	0	0	0	0	0	1	1	1	XL7
0	0	0	0	0	1	0	0	0	XL8
0	0	0	0	0	1	0	0	1	XL9
0	0	0	0	0	1	0	1	0	XL10
0	0	0	0	0	1	0	1	1	XL11
0	0	0	0	0	1	1	0	0	XL12
0	0	0	0	0	1	1	0	1	XL13
0	0	0	0	0	1	1	1	0	XL14
0	0	0	0	0	1	1	1	1	XL15
0	0	0	0	1	1	1	1	1	XL16
0	0	0	0	1	1	1	1	0	XL17
0	0	0	0	1	1	1	0	1	XL18
0	0	0	0	1	1	1	0	0	XL19
0	0	0	0	1	1	0	1	1	XL20
0	0	0	0	1	1	0	1	0	XL21
0	0	0	0	1	1	0	0	1	XL22
0	0	0	0	1	1	0	0	0	XL23
0	0	0	0	1	0	1	1	1	XL24
0	0	0	0	1	0	1	1	0	XL25
0	0	0	0	1	0	1	0	1	XL26
0	0	0	0	1	0	1	0	0	XL27
0	0	0	0	1	0	0	1	1	XL28
0	0	0	0	1	0	0	1	0	XL29
0	0	0	0	1	0	0	0	1	XL30
0	0	0	0	1	0	0	0	0	XL31

Figure 19. 28F512 Wordline Decoding

X-DECODING: Wordlines are number 0 through 511 beginning at the top of the array.

X Address									Row
A12	A7	A6	A5	A4	A13	A11	A9	A8	WL
0 • 0	0 • 0	0 • 0	1 • 1	0 • 0	0 • 1	0 • 1	0 • 1	0 • 1	XL32 • XL47
0 • 0	0 • 0	0 • 0	1 • 1	1 • 1	1 • 0	1 • 0	1 • 0	1 • 0	XL48 • XL63
0 • 0	0 • 0	1 • 1	0 • 0	0 • 0	0 • 1	0 • 1	0 • 1	0 • 1	XL64 • XL79
0 • 0	0 • 0	1 • 1	0 • 0	1 • 1	1 • 0	1 • 0	1 • 0	1 • 0	XL80 • XL95
1 • 1	1 • 1	1 • 1	1 • 1	0 • 0	0 • 1	0 • 1	0 • 1	0 • 1	XL480 • XL495
1 • 1	1 • 1	1 • 1	1 • 1	1 • 1	1 • 0	1 • 0	1 • 0	1 • 0	XL496 • XL511

Figure 19. 28F512 Wordline Decoding (Continued)

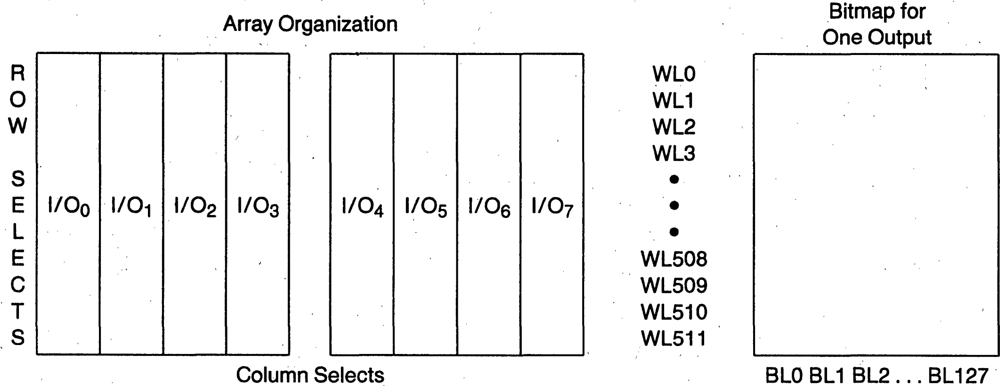


Figure 20. 28F512 Bit Map

Columns are number 0 through 511 beginning with the column nearest the X-decoder.
Outputs are grouped as follows:

Array Organization:							Left Half Array IO_0 IO_1 IO_2 IO_3 $BL_{384} \leftarrow BL_0$		Right Half Array IO_4 IO_5 IO_6 IO_7 $BL_0 \rightarrow BL_{384}$	
Address							Bitlines			
A_{16}	A_{15}	A_{10}	A_2	A_1	A_0	A_3	IO_0 & IO_7	IO_1 & IO_6	IO_2 & IO_5	IO_3 & IO_4
0	0	0	0	0	0	0	BL_{384}	BL_{256}	BL_{128}	BL_0
0	0	0	0	0	0	1	BL_{385}	BL_{257}	BL_{129}	BL_1
0	0	0	0	0	1	0	BL_{386}	BL_{258}	BL_{130}	BL_2
0	0	0	0	0	1	1	BL_{387}	BL_{259}	BL_{131}	BL_3
0	0	0	0	1	0	0	BL_{388}	BL_{260}	BL_{132}	BL_4
0	0	0	0	1	0	1	BL_{389}	BL_{261}	BL_{133}	BL_5
0	0	0	0	1	1	0	BL_{390}	BL_{262}	BL_{134}	BL_6
0	0	0	0	1	1	1	BL_{391}	BL_{263}	BL_{135}	BL_7
•	•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	0	0	BL_{508}	BL_{380}	BL_{252}	BL_{124}
1	1	1	1	1	0	1	BL_{509}	BL_{381}	BL_{253}	BL_{125}
1	1	1	1	1	1	0	BL_{510}	BL_{382}	BL_{254}	BL_{126}
1	1	1	1	1	1	1	BL_{511}	BL_{383}	BL_{255}	BL_{127}

Figure 21. 28F010 Bitline Decoding

X Address										Row
A_{14}	A_{12}	A_7	A_6	A_5	A_4	A_{13}	A_{11}	A_9	A_8	WL
0	0	0	0	0	0	0	0	0	0	XL_0
0	0	0	0	0	0	0	0	0	1	XL_1
0	0	0	0	0	0	0	0	1	0	XL_2
0	0	0	0	0	0	0	0	1	1	XL_3
0	0	0	0	0	0	0	1	0	0	XL_4
0	0	0	0	0	0	0	1	0	1	XL_5
0	0	0	0	0	0	0	1	1	0	XL_6
0	0	0	0	0	0	0	1	1	1	XL_7
0	0	0	0	0	0	1	0	0	0	XL_8
0	0	0	0	0	0	1	0	0	1	XL_9
0	0	0	0	0	0	1	0	1	0	XL_{10}
0	0	0	0	0	0	1	0	1	1	XL_{11}
0	0	0	0	0	0	1	1	0	0	XL_{12}
0	0	0	0	0	0	1	1	0	1	XL_{13}
0	0	0	0	0	0	1	1	1	0	XL_{14}
0	0	0	0	0	0	1	1	1	1	XL_{15}
0	0	0	0	0	1	1	1	1	1	XL_{16}
0	0	0	0	0	1	1	1	1	0	XL_{17}
0	0	0	0	0	1	1	1	0	1	XL_{18}
0	0	0	0	0	1	1	1	0	0	XL_{19}
0	0	0	0	0	1	1	0	1	1	XL_{20}
0	0	0	0	0	1	1	0	1	0	XL_{21}
0	0	0	0	0	1	1	0	0	1	XL_{22}
0	0	0	0	0	1	1	0	0	0	XL_{23}
0	0	0	0	0	1	0	1	1	1	XL_{24}
0	0	0	0	0	1	0	1	1	0	XL_{25}
0	0	0	0	0	1	0	1	0	1	XL_{26}
0	0	0	0	0	1	0	1	0	0	XL_{27}
0	0	0	0	0	1	0	0	1	1	XL_{28}
0	0	0	0	0	1	0	0	1	0	XL_{29}
0	0	0	0	0	1	0	0	0	1	XL_{30}
0	0	0	0	0	1	0	0	0	0	XL_{31}

Figure 22. 28F010 Wordline Decoding

X Address										Row
A_{14}	A_{12}	A_7	A_6	A_5	A_4	A_{13}	A_{11}	A_9	A_8	WL
0 • 0	0 • 0	0 • 0	0 • 0	1 • 1	0 • 0	0 • 1	0 • 1	0 • 1	0 • 1	XL32 ••• XL47
0 • 0	0 • 0	0 • 0	0 • 0	1 • 1	1 • 1	1 • 0	1 • 0	1 • 0	1 • 0	XL48 ••• XL63
0 • 0	0 • 0	0 • 0	1 • 1	0 • 0	0 • 0	0 • 1	0 • 1	0 • 1	0 • 1	XL64 ••• XL79
0 • 0	0 • 0	0 • 0	1 • 1	0 • 0	1 • 1	1 • 0	1 • 0	1 • 0	1 • 0	XL80 ••• XL95
1 • 1	1 • 1	1 • 1	1 • 1	1 • 1	0 • 0	0 • 1	0 • 1	0 • 1	0 • 1	XL992 ••• XL1007
1 • 1	1 • 1	1 • 1	1 • 1	1 • 1	1 • 1	1 • 0	1 • 0	1 • 0	1 • 0	XL1008 ••• XL1023

Figure 22. 28F010 Wordline Decoding (Continued)

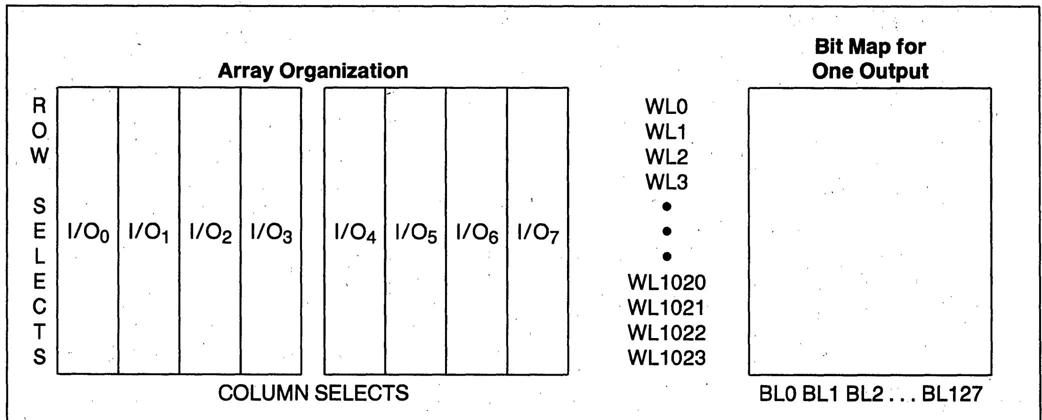


Figure 23. 28F010 Bit Map

Columns are number 0 through 511 beginning with the column nearest the X-decoder.
Outputs are grouped as follows:

Array Organization:							Left Half Array IO_0 IO_1 IO_2 IO_3 $BL_{384} \leftarrow BL_0$		Right Half Array IO_4 IO_5 IO_6 IO_7 $BL_0 \rightarrow BL_{384}$	
Address							Bitlines			
A_{16}	A_{15}	A_{10}	A_2	A_1	A_0	A_3	IO_0 & IO_7	IO_1 & IO_6	IO_2 & IO_5	IO_3 & IO_4
0	0	0	0	0	0	0	BL_{384}	BL_{256}	BL_{128}	BL_0
0	0	0	0	0	0	1	BL_{385}	BL_{257}	BL_{129}	BL_1
0	0	0	0	0	1	0	BL_{386}	BL_{258}	BL_{130}	BL_2
0	0	0	0	0	1	1	BL_{387}	BL_{259}	BL_{131}	BL_3
0	0	0	0	1	0	0	BL_{388}	BL_{260}	BL_{132}	BL_4
0	0	0	0	1	0	1	BL_{389}	BL_{261}	BL_{133}	BL_5
0	0	0	0	1	1	0	BL_{390}	BL_{262}	BL_{134}	BL_6
0	0	0	0	1	1	1	BL_{391}	BL_{263}	BL_{135}	BL_7
•	•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	0	0	BL_{508}	BL_{380}	BL_{252}	BL_{124}
1	1	1	1	1	0	1	BL_{509}	BL_{381}	BL_{253}	BL_{125}
1	1	1	1	1	1	0	BL_{510}	BL_{382}	BL_{254}	BL_{126}
1	1	1	1	1	1	1	BL_{511}	BL_{383}	BL_{255}	BL_{127}

Figure 24. 28F020 Bitline Decoding

X Address											Row
A_{17}	A_{14}	A_{12}	A_7	A_6	A_5	A_4	A_{13}	A_{11}	A_9	A_8	WL
0	0	0	0	0	0	0	0	0	0	0	XL_0
0	0	0	0	0	0	0	0	0	0	1	XL_1
0	0	0	0	0	0	0	0	0	1	0	XL_2
0	0	0	0	0	0	0	0	0	1	1	XL_3
0	0	0	0	0	0	0	0	1	0	0	XL_4
0	0	0	0	0	0	0	0	1	0	1	XL_5
0	0	0	0	0	0	0	0	1	1	0	XL_6
0	0	0	0	0	0	0	0	1	1	1	XL_7
0	0	0	0	0	0	0	1	0	0	0	XL_8
0	0	0	0	0	0	0	1	0	0	1	XL_9
0	0	0	0	0	0	0	1	0	1	0	XL_{10}
0	0	0	0	0	0	0	1	0	1	1	XL_{11}
0	0	0	0	0	0	0	1	1	0	0	XL_{12}
0	0	0	0	0	0	0	1	1	0	1	XL_{13}
0	0	0	0	0	0	0	1	1	1	0	XL_{14}
0	0	0	0	0	0	0	1	1	1	1	XL_{15}
0	0	0	0	0	0	1	1	1	1	1	XL_{16}
0	0	0	0	0	0	1	1	1	1	0	XL_{17}
0	0	0	0	0	0	1	1	1	0	1	XL_{18}
0	0	0	0	0	0	1	1	1	0	0	XL_{19}
0	0	0	0	0	0	1	1	0	1	1	XL_{20}
0	0	0	0	0	0	1	1	0	1	0	XL_{21}
0	0	0	0	0	0	1	1	0	0	1	XL_{22}
0	0	0	0	0	0	1	1	0	0	0	XL_{23}
0	0	0	0	0	0	1	0	1	1	1	XL_{24}
0	0	0	0	0	0	1	0	1	1	0	XL_{25}
0	0	0	0	0	0	1	0	1	0	1	XL_{26}
0	0	0	0	0	0	1	0	1	0	0	XL_{27}
0	0	0	0	0	0	1	0	0	1	1	XL_{28}
0	0	0	0	0	0	1	0	0	1	0	XL_{29}
0	0	0	0	0	0	1	0	0	0	1	XL_{30}
0	0	0	0	0	0	1	0	0	0	0	XL_{31}

Figure 25. 28F020 Wordline Decoding

X Address											Row
A_{17}	A_{14}	A_{12}	A_7	A_6	A_5	A_4	A_{13}	A_{11}	A_9	A_8	WL
0	0	0	0	0	1	0	0	0	0	0	XL32
•	•	•	•	•	•	•	•	•	•	•	•••
0	0	0	0	0	1	0	1	1	1	1	XL47
0	0	0	0	0	1	1	1	1	1	1	XL48
•	•	•	•	•	•	•	•	•	•	•	•••
0	0	0	0	0	1	1	0	0	0	0	XL63
0	0	0	0	1	0	0	0	0	0	0	XL64
•	•	•	•	•	•	•	•	•	•	•	•••
0	0	0	0	1	0	0	1	1	1	1	XL79
0	0	0	0	1	0	1	1	1	1	1	XL80
•	•	•	•	•	•	•	•	•	•	•	•••
0	0	0	0	1	0	1	0	0	0	0	XL95
0	1	1	1	1	1	0	0	0	0	0	XL992
•	•	•	•	•	•	•	•	•	•	•	•••
0	1	1	1	1	1	0	1	1	1	1	XL1007
0	1	1	1	1	1	1	1	1	1	1	XL1008
•	•	•	•	•	•	•	•	•	•	•	•••
0	1	1	1	1	1	1	0	0	0	0	XL1023
1	1	1	1	1	1	1	0	0	0	0	XL2016
•	•	•	•	•	•	•	•	•	•	•	•••
1	1	1	1	1	1	0	1	1	1	1	XL2031
1	1	1	1	1	1	1	1	1	1	1	XL2032
•	•	•	•	•	•	•	•	•	•	•	•••
1	1	1	1	1	1	1	0	0	0	0	XL2047

Figure 25. 28F020 Wordline Decoding (Continued)

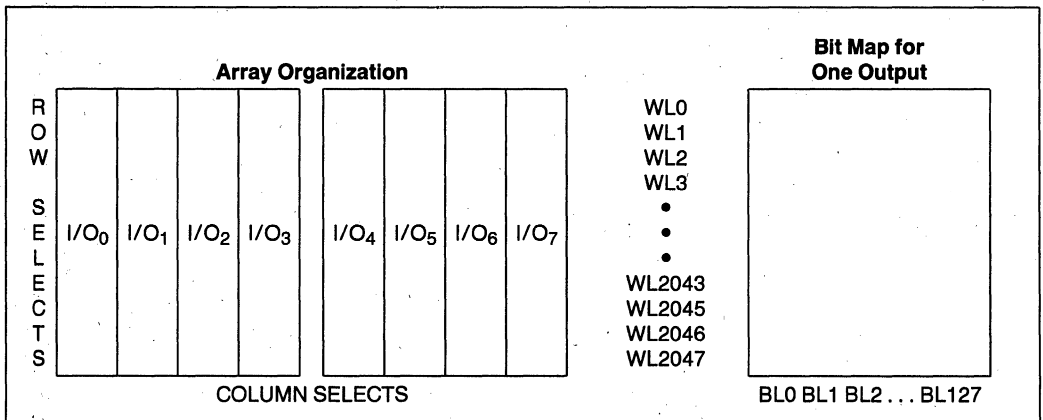


Figure 26. 28F020 Bit Map

intel®

ENGINEERING REPORT

ER-26

August 1992

3

The Intel 28F001BX-T and 28F001BX-B Flash Memories

BRIAN DIPERT
OWEN JUNGROTH
MEMORY COMPONENTS DIVISION

Order Number: 294010-002

The Intel 28F001BX-T and 28F001BX-B Flash Memories

CONTENTS PAGE

INTRODUCTION 3-553

TECHNOLOGY OVERVIEW 3-553

DEVICE ARCHITECTURE 3-554

Write State Machine and Command/Status Registers 3-554

Internal Oscillator 3-555

Supply Voltage Sensing 3-555

Erasure 3-556

Programming 3-557

Power Down 3-557

DEVICE RELIABILITY 3-558

Cell Margining 3-558

Erase/Program Cycling 3-559

SUMMARY 3-559

INTRODUCTION

Intel's 28F001BX ETOX™II (EPROM tunnel oxide) flash memories add selective block erasure, an integrated Write State Machine and powerdown capability to Intel's standard flash memory product line. Flash memory enhances EPROM non-volatility and ease of use through electrical erasure and reprogramming. Advances in tunnel oxides and photolithography have made it possible to develop a double-polysilicon single-transistor read/write random access nonvolatile memory, capable of greater than 10,000 reprogramming cycles (typically 100,000). The 28F001BX flash memories electrically erase all bits in a block matrix via electron tunneling. The EPROM programming mechanism of hot electron injection is employed for electrical byte programming.

A Command Register/Status Register interface to a Write State Machine, internal margin voltage generation, power up/down protection and address/data latches augment standard EPROM circuitry to optimize Intel's 28F001BX family for microprocessor-controlled reprogramming.

Read timing parameters are equivalent to those of CMOS EPROMs, EEPROMs and SRAMs. The 120 ns access time results from a memory cell current of approximately 50 μA, low resistance poly-silicide wordlines, advanced scaled periphery transistors and an optimized data-out buffer.

The dense one-transistor cell structure, coupled with high array efficiency, yields a one megabit die measuring 235 by 268 mils.

TECHNOLOGY OVERVIEW

Intel's ETOX II flash memory technology is derived from its standard CMOS EPROM process base. Using advanced 1.0 μm double-polysilicon n-well CMOS technology, the 131,072 x 8 bit flash memories employ a 3.8 μm x 4.0 μm single transistor cell, affording equivalent array density as comparable EPROM technology. The flash memory cell structure is identical to the EPROM structure, except for the thinner gate (tunnel) oxide. Figure 1 compares the flash memory cell to the EPROM cell.

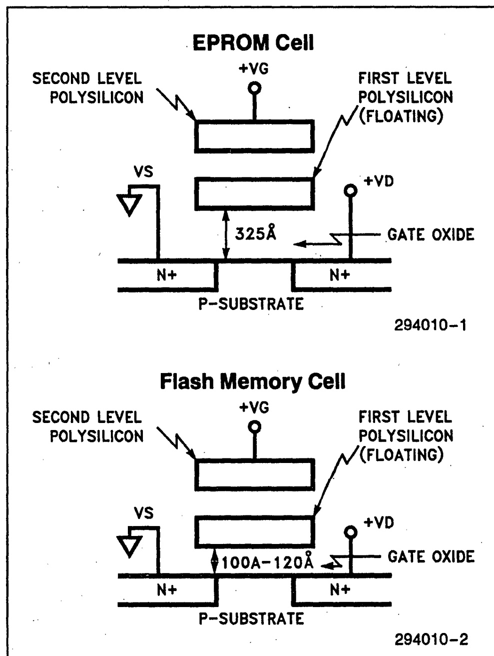


Figure 1. EPROM Cell vs. Flash Memory Cell

High quality tunnel oxide under the single floating polysilicon gate promotes electrical erasure. All cells of a given block are simultaneously erased via Fowler-Nordheim tunneling. Applying 12V on the block source junctions and grounding the select gates erases a given block. The internal Write State Machine (WSM) controls the erase algorithm, including block pre-programming before erasure. WSM-controlled erasure, including internal pre-programming, takes 2.1 seconds typical for each parameter block and the boot block, and 3.8 sec. typical for the main block.

Programming is accomplished with the standard EPROM mechanism of hot electron injection from the cell drain junction to the floating gate. Programming is initiated by bringing both the select gate and the cell drain to high voltage. The internal WSM regulates the internal program algorithm after the correct command sequence is written to the 28F001BX. Typical program time is 18 μs per byte.

DEVICE ARCHITECTURE

Write State Machine and Command/Status Registers

Intel's 28F001BX flash memories contain an on-chip Write State Machine that automatically controls erase and program algorithms, dramatically simplifying user interface. Figure 2 shows the 28F001BX block diagram.

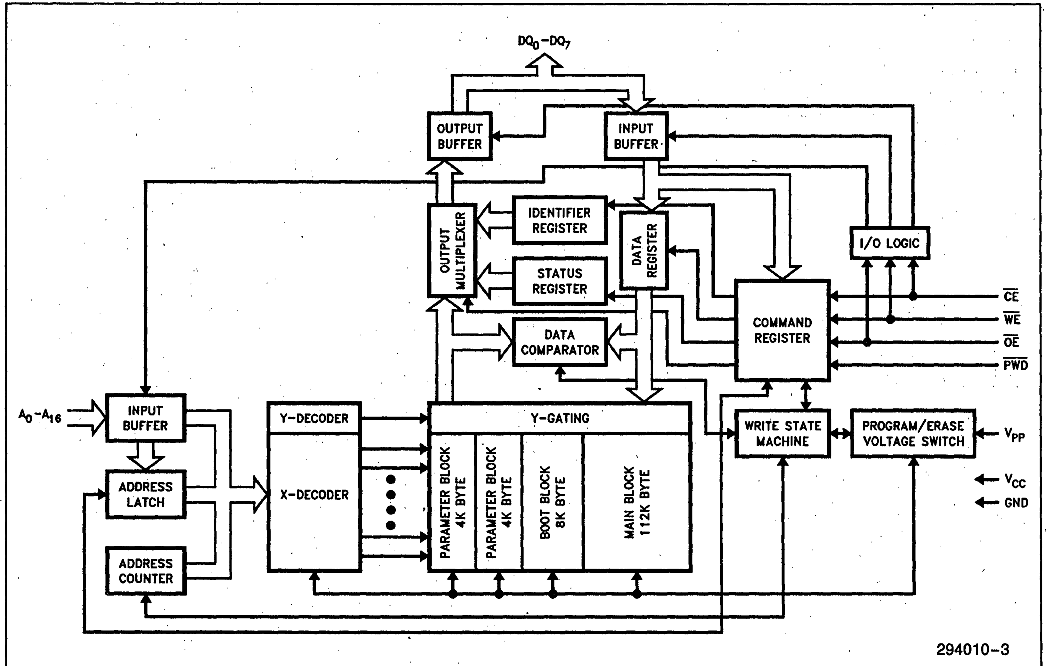


Figure 2. 28F001BX Block Diagram

The WSM simpifies microprocessor control of the erase, program, Status Register read/clear, ID read and array read operations, without the need for additional control pins or the multiplexing of high voltage with control functions. The WSM, with its integrated oscillator, performs a majority of the standard flash memory program and erase algorithms automatically. This makes system timers no longer necessary and frees the system to service interrupts or perform other functions during device erase or program. On-chip address and data latches minimize system interface logic and free the system bus. The Write State Machine accepts array read, ID read and Status Register read and clear commands whenever power is applied to the 28F001BX. High voltage (12V) on V_{PP} additionally enables successful program and erase.

The WSM consists of a Command Register, Status Register, State Machine, oscillator, command decoder, data latch and address latch. The command decoder output feeds the State Machine, enabling the high voltage flash-erase switch, program voltage generator and erase/program verify voltage generator.

Functions are selected via the Command Register in a microprocessor write cycle controlled by the Chip Enable ($\overline{CE}$) and Write Enable ($\overline{WE}$) pins. The rising edge of $\overline{WE}$ latches the address and data-in registers, and initiates an operation. Status Register contents are driven to the outputs on the falling edge of $\overline{CE}$ or Output Enable ($\overline{OE}$), whichever occurs last in the read cycle.

Internal Oscillator

The Write State Machine is designed using clocked logic circuits. An on chip ring oscillator generates the clock signals. The frequency of a standard ring oscillator varies with processing, temperature and supply voltage. The improved design used on the 28F001BX minimizes these variations.

The switching current of each stage in the ring oscillator is set by a current reference. This reference current varies linearly with V_{CC}. The trip point of each ring oscillator inverter also varies linearly with V_{CC}. These two effects essentially cancel each other out and the resulting oscillator period is proportional to RC, with only a small dependence on V_{CC}.

The value of R is set by an on chip resistor. The value of C is set by the gate capacitance of the inverters in the ring oscillator. Process variations in the values are reduced by trimming the period of each oscillator during manufacturing. The resistor is the only source of temperature variation.

Figure 3 shows how the oscillator period varies with temperature and supply voltage. The circuit works for supply voltages outside the normal operating conditions and for military temperatures.

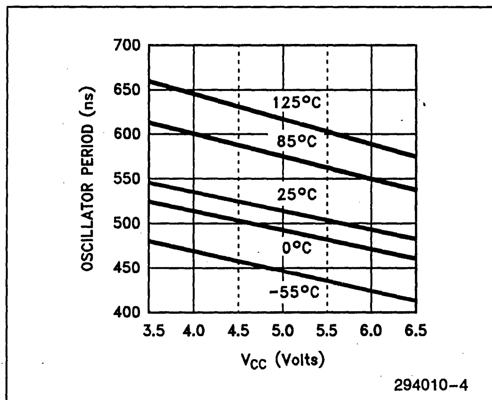


Figure 3. Internal Oscillator Frequency vs Supply Voltage and Temperature

Supply Voltage Sensing

The circuit that generates $LOWV_{CC}$ and $LOWV_{PP}$ is shown in Figure 4. Power supply voltages V_{CC} and V_{PP} are divided down and compared to a reference voltage. If the reference voltage is greater than the divided power supply voltage, the $LOWV_{CC}$ or $LOWV_{PP}$ signal will be pulled high. The V_{REF} level generated by the voltage reference is independent of the supply voltage to the first order.

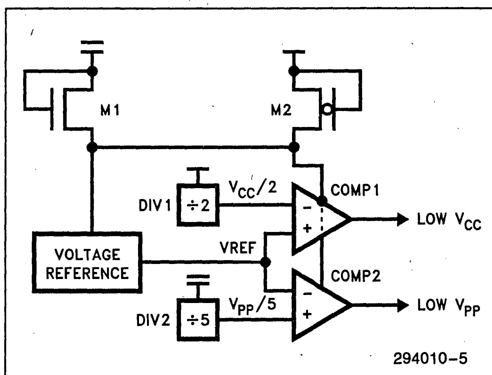


Figure 4. Low Power Detector Circuit

The positive power supply to the circuit is provided by M1 and M2. The source of M1 and M2 will be pulled up to the maximum of (V_{PP}-V_{TN}) and (V_{CC}-V_{TW}). V_{TN} is the threshold of an implanted N channel device, about 0.9V. V_{TW} is the threshold of a native N channel device, about 0V. This scheme ensures that the circuit will work regardless of the applied supply voltages.

The LOWV_{CC} signal not only goes to the erase circuits, but also to the programming circuits and to the control logic to prevent any accidental writes to the array. The LOWV_{PP} signal goes to the Write State Machine. If V_{PP} is detected as being low during a write, the low V_{PP} bit will be set in the Status Register.

Erasure

Erasure is achieved through a two-step write sequence. The erase setup code is written to the Command Register in the first cycle. The erase confirm code is written in the second cycle. The block to be erased is specified by writing both commands to any address within the block. The address is latched and decoded internally by the 28F001BX, and erase of the desired block is subsequently enabled. The rising edge of this second $\overline{WE}$ pulse initiates the erase operation. The boot block will not erase unless the $\overline{PWD}$ or $\overline{OE}$ signal is brought to high voltage V_{HH}.

The State Machine triggers the high voltage flash-erase switch, connecting the 12V supply to the source of all bits in the specified block, while all wordlines are grounded. The organization of the block source switches is shown in Figure 5. Fowler-Nordheim tunneling results in the simultaneous erasure of all bits in the addressed block.

The block source switch controls the source voltage of the bits in a particular block. This circuit is shown in Figure 6. During erase, M2 is off and M1 pulls the source to V_{PP}. When not in erase, M1 is off and M2 pulls the source to ground. The high voltage latch formed by M4–M7 converts the low voltage ERASE signal to a high voltage signal that turns M1 off or on.

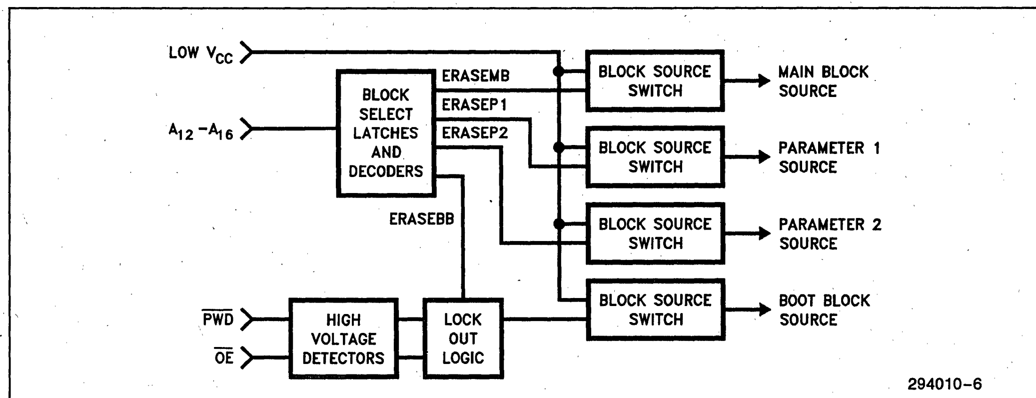


Figure 5. Array Erase Blocking

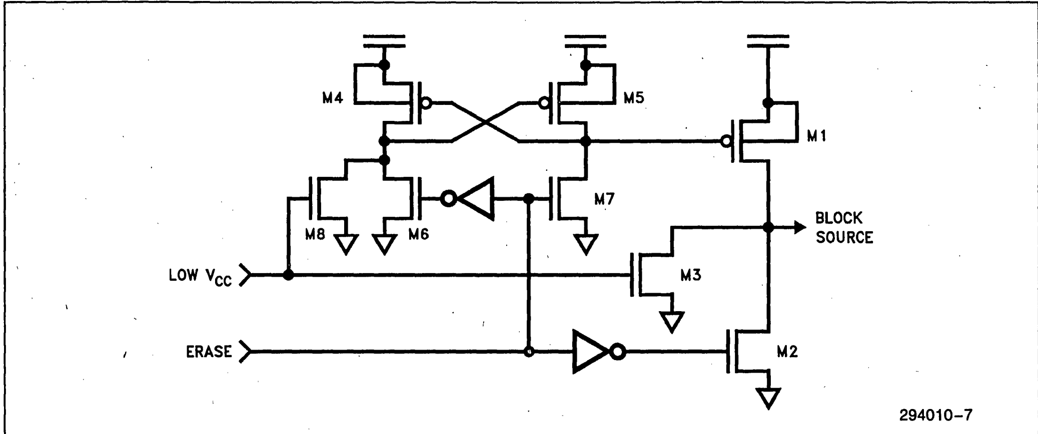


Figure 6. Block Source Switch

The tunneling that occurs during erase requires only a small amount of current. However, the grounded gate initial erase current that occurs on the source of every bit in the array is large. M1 is made large enough to supply this current and still keep the voltage on the source high enough for fast erase time.

The LOWV_{CC} signal protects the array from being erased when V_{PP} is at a high voltage but V_{CC} is a low voltage. When this occurs, M3 will pull the block source to ground. The high voltage latch will be forced into the state that turns M1 off by M8.

After receiving the erase command sequence, the WSM automatically controls block precondition (programming of all bytes to 00H within the chosen block), erase pulses and pulse repetition, timeout delays and byte-by-byte verification of all block addresses using the internally-generated erase margin voltage. The internal erase and verify operations continue until the entire block is erased. System software need only poll the Status Register to determine when the WSM has successfully completed the erase algorithm.

Programming

Programming follows a similar flow. The program set-up command is written to the Command Register on the first cycle. The second cycle loads the address and data latches. The rising edge of the second $\overline{WE}$ pulse initiates programming by applying high voltage to the gates and drains of the bits to be programmed.

As with erasure, the WSM controls program pulses and pulse repetition, timeout delays and byte verification. Program and program verify (at the internally-generated verify voltage) continue until the byte is programmed. System software, polling the Status Register, is informed of programming state thru specific status bits.

Power Down

The 28F001BX has a deep power down mode that reduces I_{CC} and I_{PP} to typically 0.05 μA and 0.8 μA, respectively. When $\overline{PWD}$ is low, the part is in deep power down mode. When $\overline{PWD}$ is high, the part can be placed in an active or standby mode by state of the $\overline{CE}$ pin.

The deep power down mode is similar to the standby mode except that more circuits are turned off. This means that much less power is consumed; it also means that it takes longer for the part to transition into the active mode.

A diagram of the power down circuit is shown in Figure 7. The TTL buffer formed by M1–M3 enables the low power detect circuits, the redundancy address flash bits and the $\overline{CE}$ TTL buffer formed by M4–M6. In previous Intel flash chips these circuits were always enabled. The time for these circuits to turn on determines the $\overline{PWD}$ access time and write specifications.

$\overline{PWD}$ will function properly with TTL level inputs. However, to get the lowest possible power consumption, full CMOS levels should be used. If voltage on the gate of M3 raises above its threshold voltage of 0.9V, it will turn on and draw current. Input voltages in the 0.7–0.9 range could cause enough subthreshold conduction in M3 to exceed the deep power down current specification. This is why the input voltage for $\overline{PWD}$ is specified as GND ±0.2V.

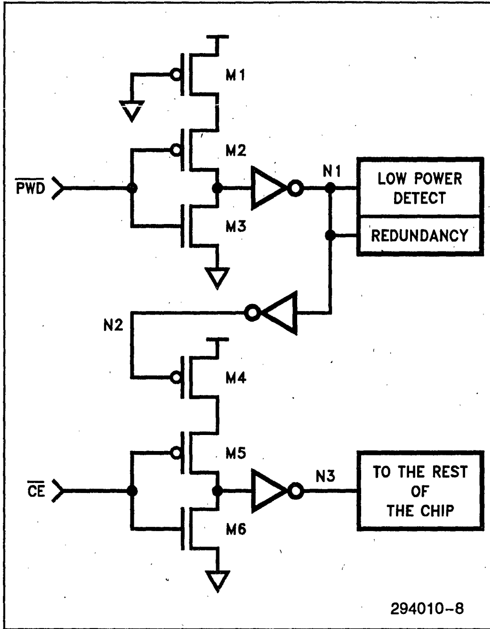


Figure 7. Power Down Circuits

DEVICE RELIABILITY

Cell Margining

Erase and program verification ensure the data retention of the newly altered memory bits. The cell margining performed by the WSM during the verify phase of the automated algorithms is more reliable than historical EEPROM schemes, as margining tests the amount of charge stored on the floating gate.

Intel's 28F001BX flash memories employ a unique circuit to internally generate the erase and program verify voltages. Figure 8 shows a simplified version of the circuit. The circuit consists of a high voltage switch and the verify voltage generator. Transistors M1 through M4 constitute the high voltage switch which disconnects V_{PP} from the resistor when the device is not in the verify mode. The verify voltage generator includes a resistor divider and a buffer. Internal margin voltage generation maintains microprocessor compatibility by eliminating the need for external reference voltages.

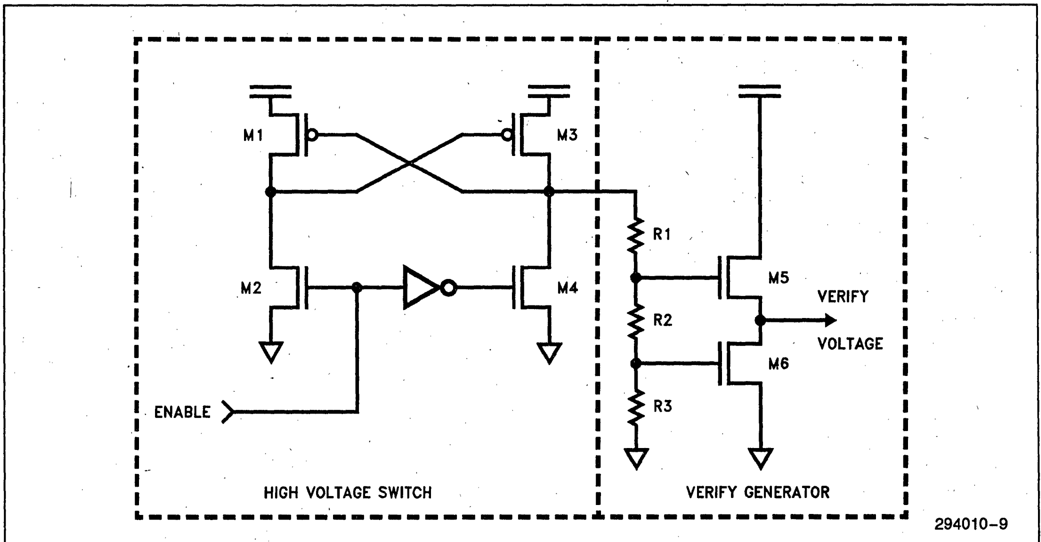


Figure 8. Erase/Program Verify Generator

Erase/Program Cycling

One of the most significant aspects of 28F001BX flash memories is their capability for a minimum of 10,000 erase/program cycles (typically 100,000 per block). Destructive oxide breakdown has been a limiting factor in extended cycling of thin oxide EEPROMs. Intel's ETOX II flash memory technology extends cycling performance through:

- Improved tunnel oxide processing that increases charge carrying capability tenfold;
- Reduced oxide area under stress minimizing probability of oxide defects in the region; and
- Reduced oxide stress due to a lower peak electric field (lower erase voltage than EEPROM).

A typical cell erase/program margin (V_t) is shown as a function of reprogramming cycles in Figure 9. After 10,000 reprograming cycles, a 2.5V program read margin exists, ensuring reliable data retention. Accelerated retention bake experiments, for devices cycled 10,000 times, show minimal program V_t shift.

Reliable erase/program cycling also requires proper selection of the erase V_t maximum and maintenance of a tight V_t distribution. The maximum erased V_t is set to 3.2V via the internal erase algorithm and erase verify circuits. Superior oxide quality gives an erased V_t distribution width that improves slightly with cycling (Figure 10). The tight erase V_t distribution gives an order of magnitude of erase time margin to the fastest erasing cell.

SUMMARY

Intel's ETOX II flash memory technology is a breakthrough in adding electrical chip-erasure to high-density EPROM technology. Intel's 28F001BX family enhances Intel's standard flash memory line by adding block erase capability, Write State Machine-controlled program and erase and deep powerdown mode. Microprocessor-compatible specifications, straightforward interfacing and in-circuit selective alterability using simple software command sequences allow designers to easily augment memory flexibility and satisfy the need for nonvolatile storage in today's designs.

3

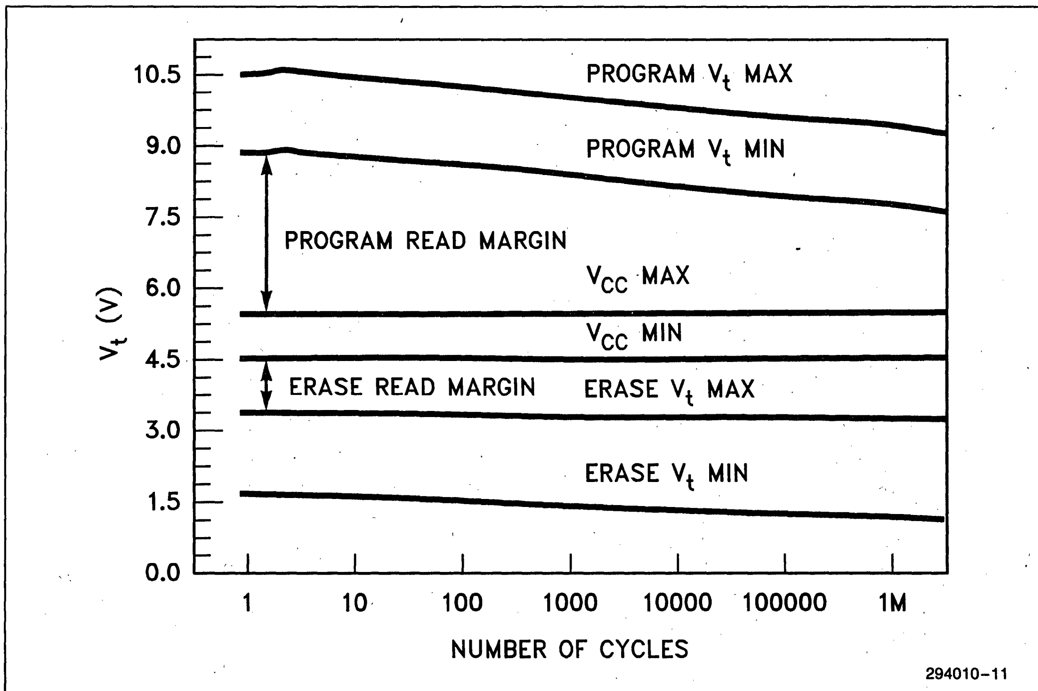


Figure 9. Block V_t vs Cycles

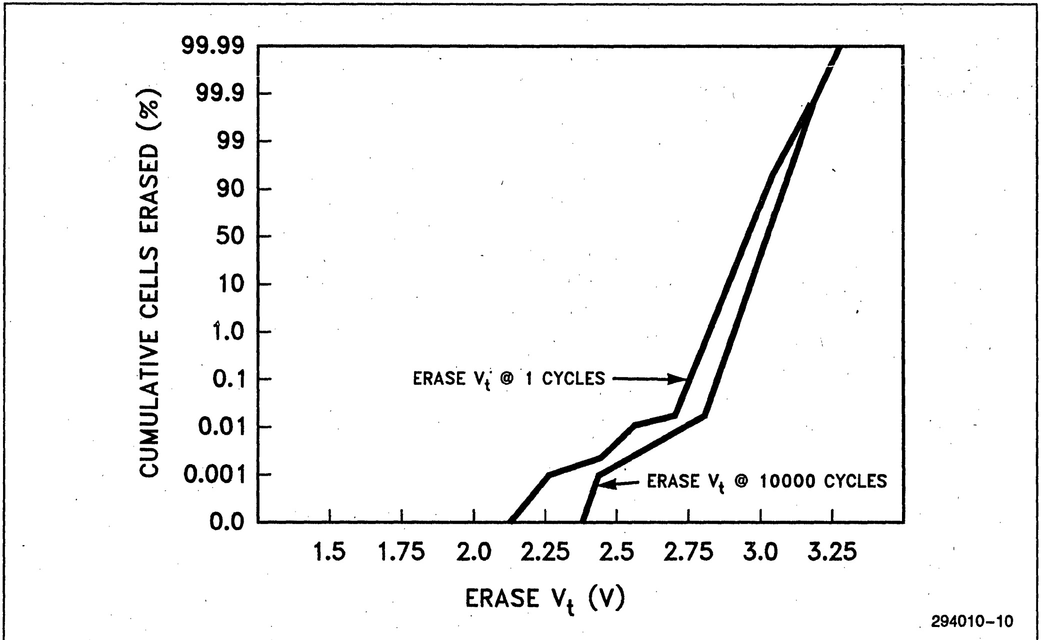


Figure 10. Erase V_t Distribution vs Cycling

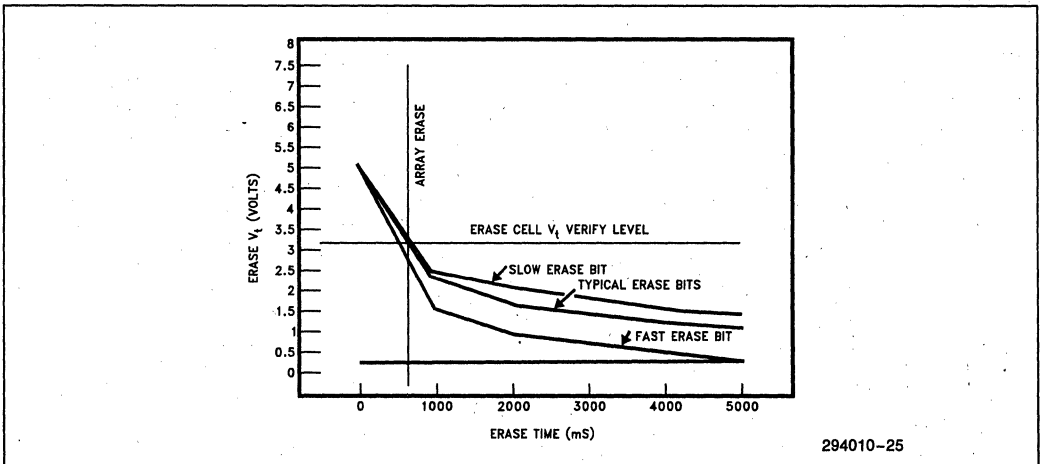
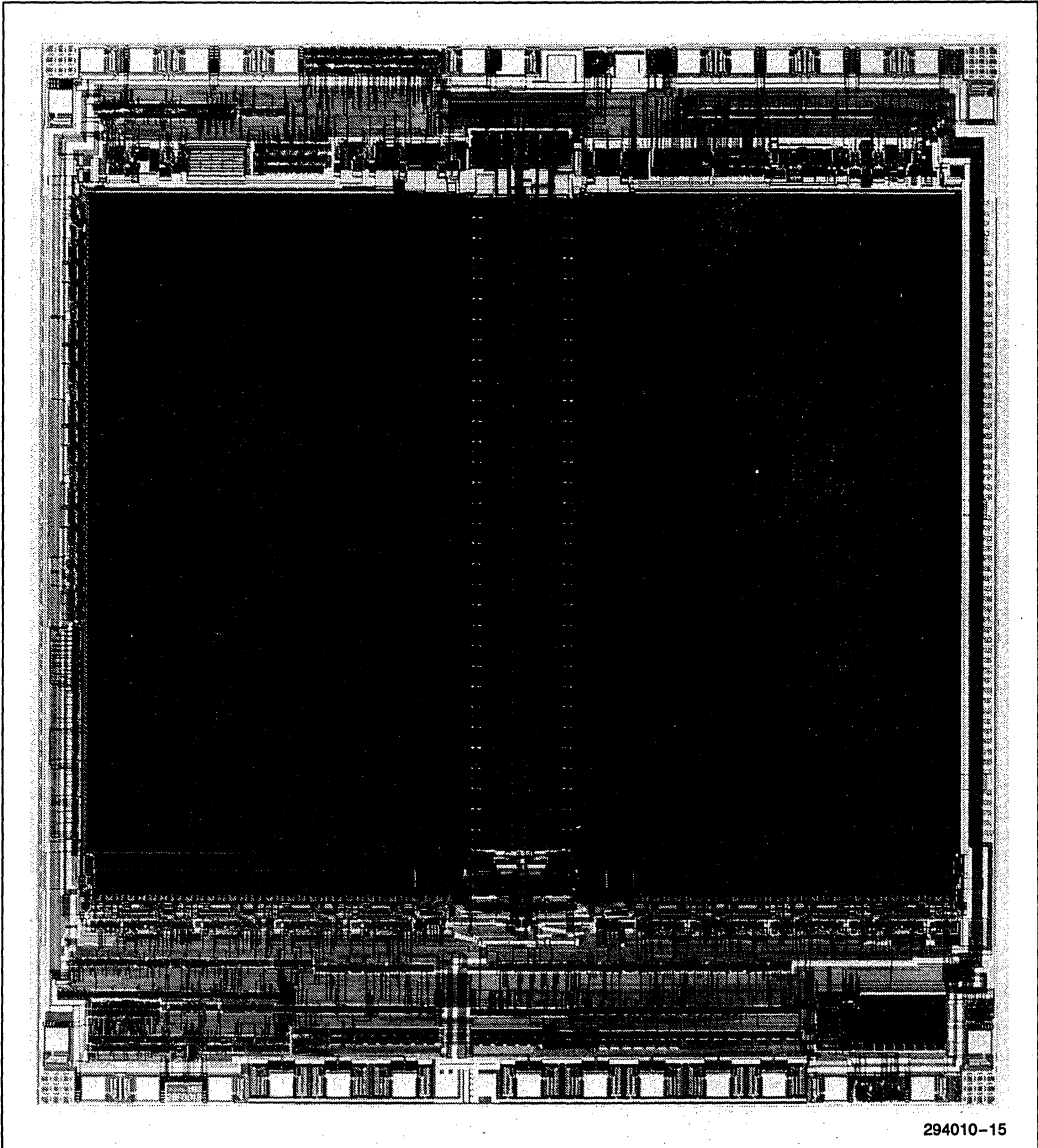


Figure 11. Block Erase V_t vs Erase Time



294010-15

Figure 12. 28F001BX Die Photograph

P28F001BX
32-PIN
PDIP
0.62" x 1.64"
TOP VIEW

Pin	Name	Pin	Name
1	V_{PP}	32	V_{CC}
2	A_{16}	31	$\overline{WE}$
3	A_{15}	30	$\overline{PWD}$
4	A_{12}	29	A_{14}
5	A_7	28	A_{13}
6	A_6	27	A_8
7	A_5	26	A_9
8	A_4	25	A_{11}
9	A_3	24	$\overline{OE}$
10	A_2	23	A_{10}
11	A_1	22	$\overline{CE}$
12	A_0	21	DQ_7
13	DQ_0	20	DQ_6
14	DQ_1	19	DQ_5
15	DQ_2	18	DQ_4
16	GND	17	DQ_3

294010-12

Pin Names

A_0–A_{16}	Address Inputs
DQ_0–DQ_7	Data Inputs/Outputs
$\overline{CE}$	Chip Enable
$\overline{PWD}$	Power Down
$\overline{OE}$	Output Enable
$\overline{WE}$	Write Enable
V_{PP}	Program/Erase Power Supply
V_{CC}	Device Power Supply
GND	Ground

E28F001BX
32 LEAD TSOP
8mm x 20mm
TOP VIEW

Pin	Name	Pin	Name
1	A_{11}	32	$\overline{OE}$
2	A_9	31	A_{10}
3	A_8	30	$\overline{CE}$
4	A_{13}	29	DQ_7
5	A_{14}	28	DQ_6
6	$\overline{PWD}$	27	DQ_5
7	$\overline{WE}$	26	DQ_4
8	V_{CC}	25	DQ_3
9	V_{PP}	24	GND
10	A_{16}	23	DQ_2
11	A_{15}	22	DQ_1
12	A_{12}	21	DQ_0
13	A_7	20	A_0
14	A_6	19	A_1
15	A_5	18	A_2
16	A_4	17	A_3

294010-13

Figure 13. 28F001BX Pinout Configurations

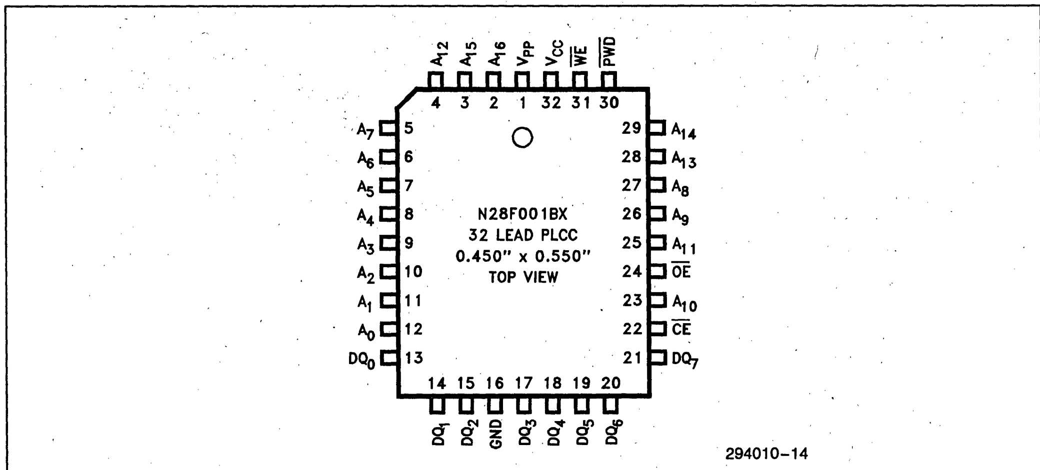


Figure 14. 28F001BX Pinout Configurations

Columns are number 0 through 511 beginning with the column nearest the X-decoder. Outputs are grouped as follows:

Array Organization (Main Block):							Left Half Array IO_0 IO_1 IO_2 IO_3 $BL_{511} \leftarrow BL_{64}$		Right Half Array IO_4 IO_5 IO_6 IO_7 $BL_{64} \rightarrow BL_{511}$	
Address							Bitlines			
A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_1	A_0	IO_0 & IO_7	IO_1 & IO_6	IO_2 & IO_5	IO_3 & IO_4
0	0	0	0	0	0	0	BL_{400}	BL_{288}	BL_{176}	BL_{64}
0	0	0	0	0	0	1	BL_{401}	BL_{289}	BL_{177}	BL_{65}
0	0	0	0	0	1	0	BL_{402}	BL_{290}	BL_{178}	BL_{66}
0	0	0	0	0	1	1	BL_{403}	BL_{291}	BL_{179}	BL_{67}
0	0	0	0	1	0	0	BL_{404}	BL_{292}	BL_{180}	BL_{68}
0	0	0	0	1	0	1	BL_{405}	BL_{293}	BL_{181}	BL_{69}
0	0	0	0	1	1	0	BL_{406}	BL_{294}	BL_{182}	BL_{70}
0	0	0	0	1	1	1	BL_{407}	BL_{295}	BL_{183}	BL_{71}
*	*	*	*	*	*	*	*	*	*	*
1	1	0	1	1	0	0	BL_{508}	BL_{396}	BL_{284}	BL_{172}
1	1	0	1	1	0	1	BL_{509}	BL_{397}	BL_{285}	BL_{173}
1	1	0	1	1	1	0	BL_{510}	BL_{398}	BL_{286}	BL_{174}
1	1	0	1	1	1	1	BL_{511}	BL_{399}	BL_{287}	BL_{175}

Figure 15. Bitline Decoding (Main Block, 28F001BX-T)

Array Organization (Parameter Block 1):							Right Half Array IO_0–IO_7 $BL_0 \rightarrow BL_{31}$							
Address							Bitlines							
A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_1	A_0	IO_0	IO_1	IO_2	IO_3	IO_4	IO_5	IO_6	IO_7
1	1	1	0	0	0	0	BL_0	BL_4	BL_8	BL_{12}	BL_{16}	BL_{20}	BL_{24}	BL_{28}
1	1	1	0	0	0	1	BL_1	BL_5	BL_9	BL_{13}	BL_{17}	BL_{21}	BL_{25}	BL_{29}
1	1	1	0	0	1	0	BL_2	BL_6	BL_{10}	BL_{14}	BL_{18}	BL_{22}	BL_{26}	BL_{30}
1	1	1	0	0	1	1	BL_3	BL_7	BL_{11}	BL_{15}	BL_{19}	BL_{23}	BL_{27}	BL_{31}

Figure 16. Bitline Decoding (Parameter Block 1, 28F001BX-T)

Array Organization (Parameter Block 2):							Right Half Array IO_0–IO_7 $BL_{32} \rightarrow BL_{63}$							
Address							Bitlines							
A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_1	A_0	IO_0	IO_1	IO_2	IO_3	IO_4	IO_5	IO_6	IO_7
1	1	1	0	1	0	0	BL_{32}	BL_{36}	BL_{40}	BL_{44}	BL_{48}	BL_{52}	BL_{56}	BL_{60}
1	1	1	0	1	0	1	BL_{33}	BL_{37}	BL_{41}	BL_{45}	BL_{49}	BL_{53}	BL_{57}	BL_{61}
1	1	1	0	1	1	0	BL_{34}	BL_{38}	BL_{42}	BL_{46}	BL_{50}	BL_{54}	BL_{58}	BL_{62}
1	1	1	0	1	1	1	BL_{35}	BL_{39}	BL_{43}	BL_{47}	BL_{51}	BL_{55}	BL_{59}	BL_{63}

Figure 17. Bitline Decoding (Parameter Block 2, 28F001BX-T)

NOTES:

1. Bitline decoding listed is for 28F001BX-T. To convert to 28F001BX-B, invert polarity of addresses A_{16}–A_{12} (i.e. 0000000 becomes 1111100).

Array Organization (Boot Block):							Left Half Array IO_7–IO_0 $BL_0 \rightarrow BL_{63}$								
Address							Bitlines								
A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_1	A_0	IO_7	IO_6	IO_5	IO_4	IO_3	IO_2	IO_1	IO_0
1	1	1	1	0	0	0	BL_0	BL_8	BL_{16}	BL_{24}	BL_{32}	BL_{40}	BL_{48}	BL_{56}
1	1	1	1	0	0	1	BL_1	BL_9	BL_{17}	BL_{25}	BL_{33}	BL_{41}	BL_{49}	BL_{57}
1	1	1	1	0	1	0	BL_2	BL_{10}	BL_{18}	BL_{26}	BL_{34}	BL_{42}	BL_{50}	BL_{58}
1	1	1	1	0	1	1	BL_3	BL_{11}	BL_{19}	BL_{27}	BL_{35}	BL_{43}	BL_{51}	BL_{59}
1	1	1	1	1	0	0	BL_4	BL_{12}	BL_{20}	BL_{28}	BL_{36}	BL_{44}	BL_{52}	BL_{60}
1	1	1	1	1	0	1	BL_5	BL_{13}	BL_{21}	BL_{29}	BL_{37}	BL_{45}	BL_{53}	BL_{61}
1	1	1	1	1	1	0	BL_6	BL_{14}	BL_{22}	BL_{30}	BL_{38}	BL_{46}	BL_{54}	BL_{62}
1	1	1	1	1	1	1	BL_7	BL_{15}	BL_{23}	BL_{31}	BL_{39}	BL_{47}	BL_{55}	BL_{63}

Figure 18. Bitline Decoding (Boot Block, 28F001BX-T)

NOTE:

1. Bitline decoding listed is for 28F001BX-T. To convert to 28F001BX-B, invert polarity of addresses A_{16}–A_{12} (i.e. 0000000 becomes 1111100).

X Address										Row
A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	WL
0	0	0	0	0	0	0	0	0	0	XL_0
0	0	0	0	0	0	0	0	0	1	XL_1
0	0	0	0	0	0	0	0	1	0	XL_2
0	0	0	0	0	0	0	0	1	1	XL_3
0	0	0	0	0	0	0	1	0	0	XL_4
0	0	0	0	0	0	0	1	0	1	XL_5
0	0	0	0	0	0	0	1	1	0	XL_6
0	0	0	0	0	0	0	1	1	1	XL_7
0	0	0	0	0	0	1	0	0	0	XL_8
0	0	0	0	0	0	1	0	0	1	XL_9
0	0	0	0	0	0	1	0	1	0	XL_{10}
0	0	0	0	0	0	1	0	1	1	XL_{11}
0	0	0	0	0	0	1	1	0	0	XL_{12}
0	0	0	0	0	0	1	1	0	1	XL_{13}
0	0	0	0	0	0	1	1	1	0	XL_{14}
0	0	0	0	0	0	1	1	1	1	XL_{15}
0	0	0	0	0	1	1	1	1	1	XL_{16}
0	0	0	0	0	1	1	1	1	0	XL_{17}
0	0	0	0	0	1	1	1	0	1	XL_{18}
0	0	0	0	0	1	1	1	0	0	XL_{19}
0	0	0	0	0	1	1	0	1	1	XL_{20}
0	0	0	0	0	1	1	0	1	0	XL_{21}
0	0	0	0	0	1	1	0	0	1	XL_{22}
0	0	0	0	0	1	1	0	0	0	XL_{23}
0	0	0	0	0	1	0	1	1	1	XL_{24}
0	0	0	0	0	1	0	1	1	0	XL_{25}
0	0	0	0	0	1	0	1	0	1	XL_{26}
0	0	0	0	0	1	0	1	0	0	XL_{27}
0	0	0	0	0	1	0	0	1	1	XL_{28}
0	0	0	0	0	1	0	0	1	0	XL_{29}
0	0	0	0	0	1	0	0	0	1	XL_{30}
0	0	0	0	0	1	0	0	0	0	XL_{31}

Figure 19. Wordline Decoding (28F001BX-T and 28F001BX-B)

X Address										Row
A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	WL
0	0	0	0	1	0	0	0	0	0	XL_{32}
*	*	*	*	*	*	*	*	*	*	***
0	0	0	0	1	0	1	1	1	1	XL_{47}
0	0	0	0	1	1	1	1	1	1	XL_{48}
*	*	*	*	*	*	*	*	*	*	***
0	0	0	0	1	1	0	0	0	0	XL_{63}
0	0	0	1	0	0	0	0	0	0	XL_{64}
*	*	*	*	*	*	*	*	*	*	***
0	0	0	1	0	0	1	1	1	1	XL_{79}
0	0	0	1	0	1	1	1	1	1	XL_{80}
*	*	*	*	*	*	*	*	*	*	***
0	0	0	1	0	1	0	0	0	0	XL_{95}
1	1	1	1	1	0	0	0	0	0	XL_{992}
*	*	*	*	*	*	*	*	*	*	***
1	1	1	1	1	0	1	1	1	1	XL_{1007}
1	1	1	1	1	1	1	1	1	1	XL_{1008}
*	*	*	*	*	*	*	*	*	*	***
1	1	1	1	1	1	0	0	0	0	XL_{1023}

Figure 19. Wordline Decoding (28F001BX-T and 28F001BX-B) (Continued)

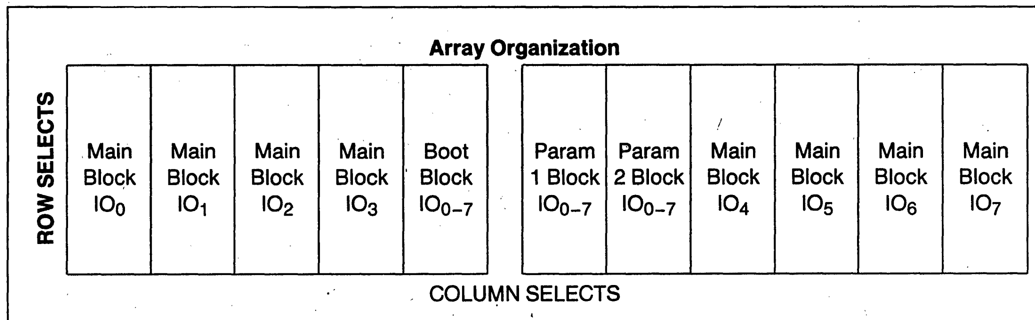


Figure 20. Array Organization

REVISION HISTORY

Number	Description
002	Swapped Figures 9 and 10 Added Figure 11

ENGINEERING REPORT

ER-27

November 1992

The Intel 28F008SA Flash Memory

ALAN BUCHECKER
JERRY KREIFELS
MEMORY COMPONENTS DIVISION

Order Number: 294011-001

The Intel 28F008SA Flash Memory

CONTENTS PAGE

INTRODUCTION 3-568

TECHNOLOGY OVERVIEW 3-568

DEVICE ARCHITECTURE 3-569

Array Organization 3-569

Write/Erase Automation 3-569

Command User Interface (CUI) 3-570

Write State Machine (WSM) 3-570

Status Register 3-571

Ready/Busy Indication (RY/$\overline{BY}$) 3-571

Internal Oscillator 3-571

Supply Voltage Sensing 3-571

Power Down 3-572

Block Erase 3-573

Erase Suspend/Resume 3-574

Byte Write 3-574

CONTENTS PAGE

DEVICE CHARACTERIZATION 3-575

AC and DC Parameters 3-575

Energy/Power Consumption 3-575

Byte-Write/Block-Erase Times 3-575

DEVICE RELIABILITY 3-575

Byte-Write/Block-Erase Cycling 3-575

Data Protection 3-576

SUMMARY 3-576

OTHER REFERENCES 3-576

SUPPLEMENTARY INFORMATION 3-577

INTRODUCTION

The ETOX™-III (EPROM tunnel oxide) 28F008SA is a high-density product offering from Intel's second generation of flash memory devices. This 1,048,576 x 8 memory with its symmetrical blocking (16 blocks x 64 Kbytes), very high cycling endurance, on-chip write/erase automation, and erase-suspend/resume capability can be termed a block-alterable non-volatile RAM. In addition to selective block erasure, integrated Command User Interface (CUI), Write State Machine (WSM), Status Register, and deep power-down capability, the 28F008SA adds a dedicated READY/BUSY output (RY/$\overline{BY}$). This new feature provides immediate hardware signaling of byte-write/block-erase completion and erase-suspend/resume actuation.

Flash memories combine inherent non-volatility with in-system alterability of device contents. Advances in process control have allowed development of a double-polysilicon single-transistor flash memory capable of 100,000 write/erase cycles per block. The 28F008SA electrically erases all bits in a block via electron tunneling. The EPROM programming mechanism of hot-electron injection is employed for high-performance electrical byte write as required for file and data storage applications.

The Command User Interface and Status Register interface to power-up/down protection, address/data latches, and the Write State Machine (which in turn controls internal byte write, block erase, cell-margin circuits, and the dedicated READY/BUSY status output). These features augment prior flash memory circuitry to optimize Intel's 28F008SA for microprocessor-controlled byte write and block erase.

Read timing parameters are comparable to those of CMOS DRAMs, SRAMs, EPROMs, and EEPROMs. The 85 ns access time results from a memory cell-current of approximately 70 μA, low-resistance polysilicide wordlines strapped with metal, advanced scaled periphery transistors, and an optimized data-out buffer.

The dense one-transistor cell structure, coupled with high array efficiency, yields a one-megabyte die measuring 539 by 286 mils.

TECHNOLOGY OVERVIEW

Intel's ETOX-III flash memory technology incorporates advances from ETOX-I and ETOX-II processes and leverages over two decades of EPROM manufacturing experience. Using advanced 0.8 μm double-polysilicon N-well/P-well CMOS technology, the 1,048,576 x 8-bit flash memory employs a 2.5 μm x 2.9 μm single-transistor cell affording array density equivalent to comparable EPROM technology, and twice that of Intel's ETOX-II process. The ETOX-III flash memory cell is identical to EPROM, with an additional source implant which optimizes erase performance. Figure 1 shows a cross-section of the flash memory cell.

High-quality tunnel oxide under the single floating polysilicon gate promotes electrical erasure. All cells within the selected block are simultaneously erased via Fowler-Nordheim tunneling. Applying 12V to block source junctions and grounding the select gates erases all cells within that block. The internal WSM controls the automated block-erase algorithm, including pre-erase conditioning (i.e., pre-programming all block bits) and margin verification, in response to user requests relayed by the CUI. WSM-controlled block erasure, including pre-programming, typically requires 1.6 seconds.

Byte write is accomplished with the standard EPROM mechanism of channel hot-electron injection from the cell drain junction to the floating gate. Bringing both the select gate and the cell drain to high voltage initiates programming. The WSM regulates the internal byte-write algorithm, including margin verification, after the correct command sequence is written and decoded. Byte write typically requires 9 μs.

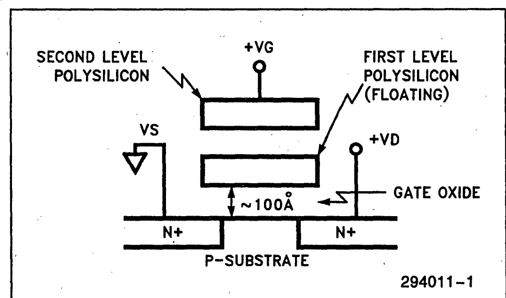


Figure 1. Flash Memory Cell

DEVICE ARCHITECTURE

Array Organization

The 28F008SA is a 1,048,576 x 8 memory comprised of 2048 rows by 8192 columns. Array layout is segmented as four quadrants, each 1024 rows by 2048 columns. Access time is reduced by limiting column length to 1024 cells. The polysilicon row is strapped in metal every 512 columns to reduce wordline delay. Two row decoders run vertically between quadrants, and column decoders run horizontally between quadrants. Figure 2 shows block placement and array organization. A die photo of the chip is shown in Figure 30.

Each quadrant is subdivided into four 64-Kbyte blocks. Each block source is electrically isolated from the source of other blocks. This allows individual block erase without altering data in the remaining 15 blocks.

Each block is further subdivided into eight Input/Outputs. Data for I/O_0 is stored in the left-most 64 columns, with the next 64 storing data for I/O_1, etc.

Rows in the upper quadrants are numbered 0–1023 from top to bottom; lower quadrant rows similarly 1024–2047.

Addresses A_9–A_0 select one of 1024 rows. while A_{19} selects upper or lower decoder. Row address lines are decoded sequentially for selection. Row address bitmaps are listed in Table 3.

Columns are numbered 0–8191 from left to right, top to bottom. Addresses A_{19}–A_{16} select one of 16 blocks, while A_{15}–A_{10} select eight of the 512 columns within that block. These ten address lines are also decoded sequentially to access all 8192 columns. Block address bitmaps are listed in Table 4; column address bitmaps are listed in Table 5.

Write/Erase Automation

Intel's 28F008SA contains an on-chip Command User Interface, Write State Machine, Status Register, and address/data latches to dramatically simplify user interface. This combination of functional units reduces microprocessor control complexity of byte-write, block-erase, erase-suspend/resume, Status Register read/clear, ID read, and array read operations. Figure 3 shows the 28F008SA block diagram.

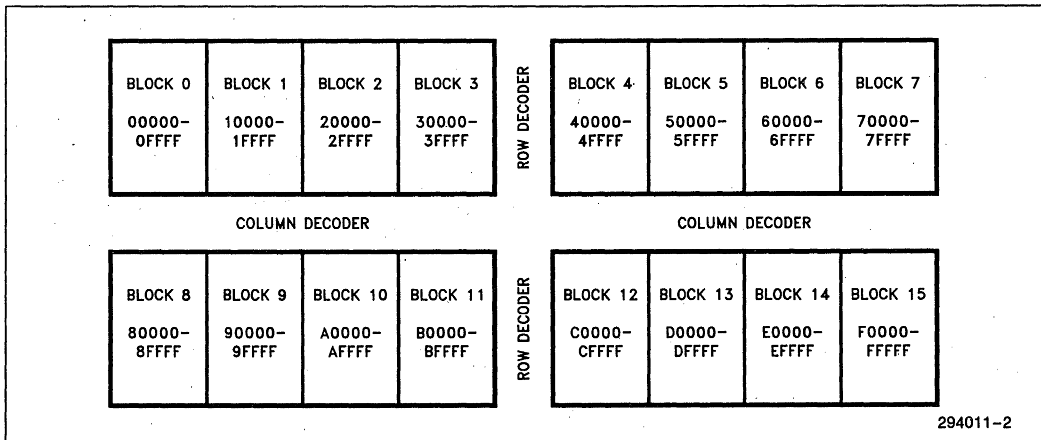


Figure 2. 28F008SA Block Placement and Array Organization

Command User Interface (CUI)

The CUI consists of a command decoder and command register. User requests are decoded and latched in a microprocessor write cycle controlled by Chip Enable ($\overline{CE}$) and Write Enable ($\overline{WE}$). Status Register read/clear, ID read, and array read commands are directly handled by the CUI. The CUI also accepts byte-write, block-erase, and erase-suspend/resume commands. $\overline{WE}$'s rising edge latches address, command and data-in registers, and requests WSM initiation of the selected operation. These on-chip address, command, and data latches controlled by the CUI minimize system interface logic, and free the system bus.

Write State Machine (WSM)

The WSM processes byte-write, block-erase, and erase-suspend/resume requests received from the CUI. The WSM rejects byte-write and block-erase requests if the WSM is currently busy, if V_{PP} is not at high voltage (12V), or if the Low V_{PP} Status Register flag is set (i.e., not cleared from a previous low-voltage condition).

The WSM consists of an integrated oscillator and control circuitry. It generates signals which control the byte-write, block-erase, erase-suspend/resume, and verify circuits. It also receives feedback from these circuits allowing Status Register update. The WSM and associated circuits perform the equivalent of first-generation flash memory program and bulk-erase algorithms automatically. This eliminates the need for system timers, and frees the microprocessor to service interrupts or perform other functions during device byte-write or block-erase operations.

The WSM provides feedback to the CUI to determine when a given command is valid. Although nearly all commands are available when the WSM is inactive, only status read is valid while the WSM performs a byte-write operation. During block erase, only the read-status and erase-suspend commands are available. Read-array, read-status, and erase-resume commands are valid with the WSM in an erase-suspended state. Invalid operations are interpreted as the read-array command when the WSM is inactive or erase-suspended, and as the read-status command when the WSM is active in byte write or block erase.

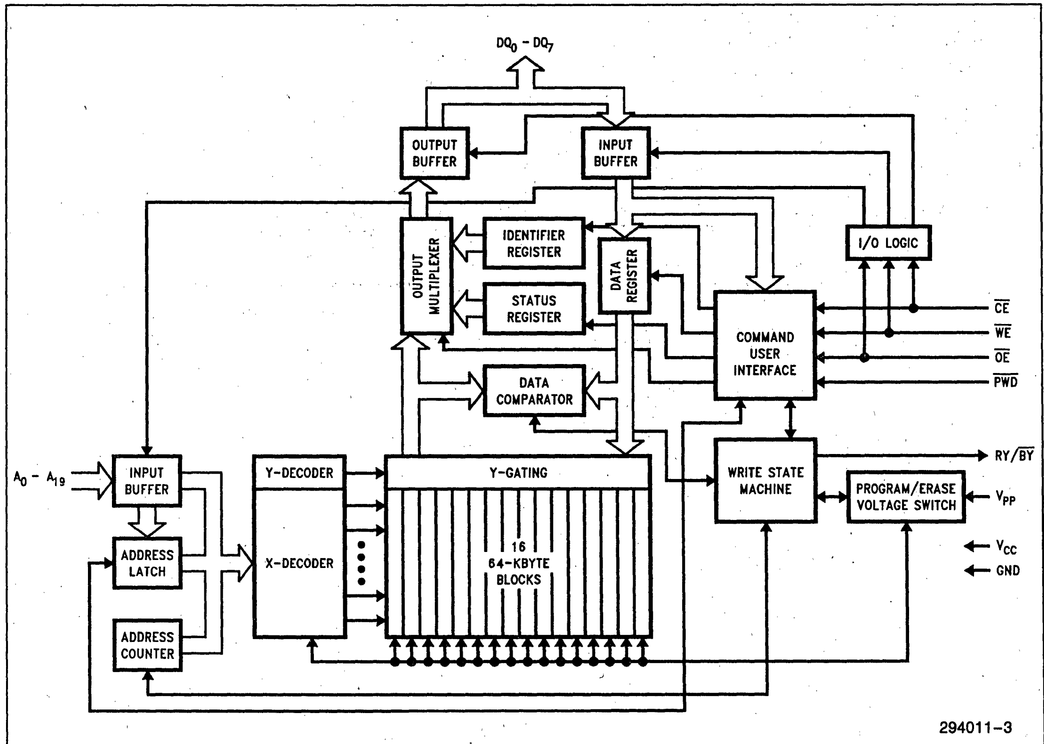


Figure 3. 28F008SA Block Diagram

Status Register

The internal Status Register contains a full complement of activity status bits. These bits and their meaning (1,0) are:

SR.7: WSM status (READY/BUSY)

SR.6: Erase-suspend status (ERASE SUSPENDED/ ERASE IN PROGRESS OR COMPLETED)

SR.5: Block-erase status (ERROR/SUCCESS)

SR.4: Byte-write status (ERROR/SUCCESS)

SR.3: V_{PP} status (LOW/OK)

All bits are set by the WSM, and read via the CUI. The WSM can only set SR.3, SR.4, and SR.5; it cannot clear them. They remain set until the CUI processes a clear Status Register command. There are two reasons for operating in this fashion.

First is synchronization; the WSM does not know when the host CPU has read the Status Register, therefore does not know when to clear it.

Secondly, allowing system software to control reset adds flexibility to the way this device may be used. The CPU may write several bytes or erase several blocks back-to-back while monitoring RY/$\overline{BY}$ or polling SR.7 to determine when the next byte-write or block-erase command can be given. When all bytes are written, or all blocks erased, the system polls the other status flags to determine if all operations were successful or if an error occurred. While other approaches require the controlling microprocessor to watch for non-completion of write or erase within a specified time to indicate an error, this implementation requires no external system timers or software timing loops. As such, the system can reduce its polling overhead while still identifying any potential error conditions.

Status Register contents are driven to device outputs on the falling edge of $\overline{CE}$ or Output Enable ($\overline{OE}$), whichever occurs last in the read cycle. $\overline{CE}$ or $\overline{OE}$ must be toggled to update Status Register contents.

Ready/Busy Indication (RY/$\overline{BY}$)

A dedicated output pin, RY/$\overline{BY}$, provides additional indication of WSM activity. This capability allows both hardware signal of status and/or software polling to determine activation, completion, or suspension of internal byte-write/block-erase operations. Hardware signaling minimizes both CPU overhead and system power consumption.

Internal Oscillator

The WSM is designed using clocked logic circuits. An on-chip ring oscillator generates the clock signals. The frequency of a standard ring oscillator varies with processing, temperature and supply voltage. An improved design, used on the 28F001BX and 28F008SA, minimizes these variations.

The switching current of each stage in the ring oscillator is controlled by a current reference which varies linearly with V_{CC}. The trip point of each ring oscillator inverter also varies linearly with V_{CC}. These two effects offset each other, and the resulting oscillator period is proportional to RC with only a small dependence on V_{CC}.

An on-chip resistor sets the value of R. The gate capacitance of the inverters in the ring oscillator sets the value of C. Process variations in these values are reduced by trimming the period of each oscillator during manufacturing. The resistor is the only source of temperature variation.

Supply Voltage Sensing

The $LOWV_{CC}$ and $LOWV_{PP}$ generation circuit is shown in Figure 4. Power supply voltages (V_{CC} and V_{PP}) are divided down and compared to a reference voltage. If V_{REF} is greater than the divided power supply voltage, the $LOWV_{CC}$ or $LOWV_{PP}$ signal is driven high. The generated V_{REF} level is supply-voltage independent to the first order.

Positive power to the circuit is supplied by M1 and M2. M1 and M2 sources are pulled up to the higher of (V_{PP} − V_{tn}) or (V_{CC} − V_{tw}). V_{tn} is the threshold of an implanted N-channel device, about 0.9V. V_{tw} is the threshold of a native N-channel device, about 0V. This scheme ensures that the circuit works regardless of the applied supply voltages.

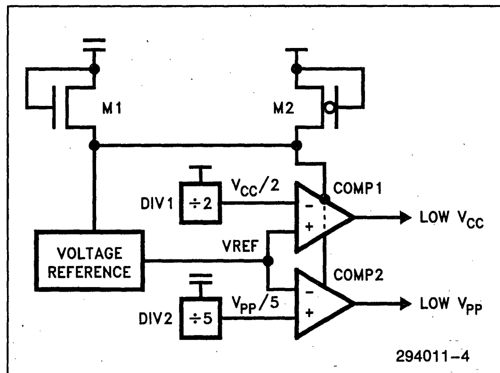


Figure 4. Low Power Detector Circuit

The $LOWV_{CC}$ signal is used by the byte-write and block-erase circuits, as well as the CUI and WSM. If $LOWV_{CC}$ is active, the CUI will not accept user writes and resets to an array read condition. The WSM is similarly reset by $LOWV_{CC}$. The $LOWV_{PP}$ signal is used by the WSM; if V_{PP} drops below the high-voltage detector trip point during byte write or block erase, the Status Register's low V_{PP} bit is set and WSM operation halts. The system must clear the Status Register before any subsequent byte-write or block-erase operations can succeed.

Power-Down

The 28F008SA incorporates a deep power-down mode that reduces I_{CC} and I_{PP} to typically 0.20 μA and 0.10 μA respectively. $\overline{PWD}$ low selects deep power-down mode. When $\overline{PWD}$ is high, the device can be placed in an active or standby mode depending on $\overline{CE}$'s state.

Deep power-down is similar to standby except that all circuits excluding the $\overline{PWD}$ buffer are turned off. This mode greatly reduces power consumption, but requires more time to transition the device into an active mode. A read wake-up time (t_{PHQV}) is required from $\overline{PWD}$ switching high until output and sense circuitry become fully functional and data can be read from the part. Similarly, a write wake-up time (t_{PHWL}) is needed before the CUI recognizes writes. After this interval, normal operation is restored; the CUI is reset to read-array mode and the Status Register is cleared to 80H.

A diagram of the power-down circuit is shown in Figure 5. The TTL buffer formed by M1–M3 disables the low-power detect circuits, the redundancy-address flash bits, and the $\overline{CE}$ TTL buffer formed by M4–M6. In previous Intel flash devices, these circuits were always enabled. Turn-on delays of these circuits determine $\overline{PWD}$ access time and write specifications.

$\overline{PWD}$ functions properly with TTL-level inputs. However, to attain lowest possible power consumption, full CMOS levels must be used. If the voltage on the gate of M3 rises above its 0.9V threshold M3 will turn on and draw current. Input voltages in the 0.7V–0.9V range could cause enough subthreshold conduction in M3 to exceed the I_{CC} deep power-down current (I_{CCD}) specification. This is why $\overline{PWD}$'s input voltage is specified as GND ±0.2V.

$\overline{PWD}$ also functions as a hardware reset to the WSM and CUI. If $\overline{PWD}$ is driven active during byte-write, block-erase, or erase-suspend operation, that operation is aborted leaving the addressed memory locations in an unknown state. The Status Register is cleared, and CUI is set to array read. The aborted operation (byte write or block erase) must be repeated with $\overline{PWD}$ inactive to obtain a valid condition in the memory array.

Reset using $\overline{PWD}$ should be restricted to system reset only (as in the case of power supply failure), and should not be used as a software means to terminate byte-write or block-erase operations.

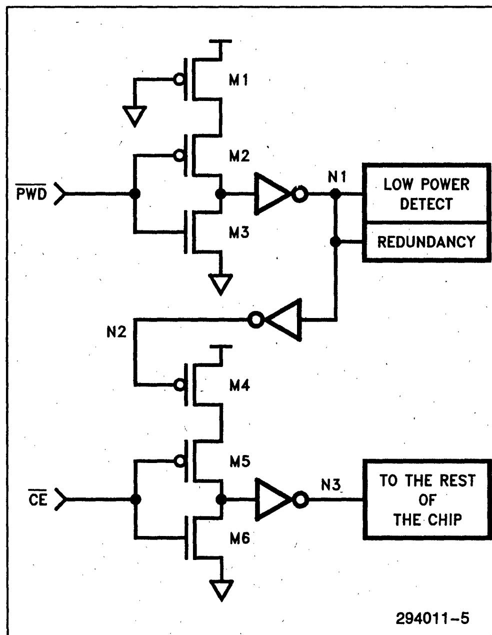


Figure 5. Power-Down Circuits

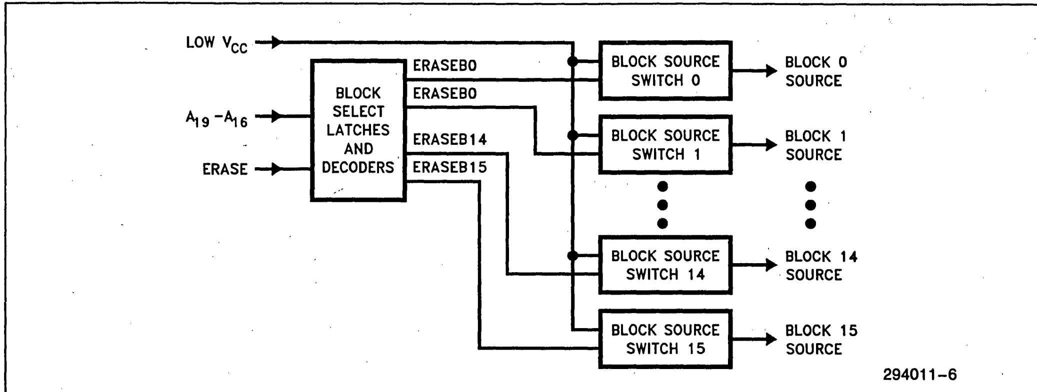


Figure 6. Array Erase Blocking

Block Erase

Block erasure is achieved by a two-step write sequence. The erase-setup code is written to the CUI in the first cycle. Erase confirm is written in the second cycle. The address supplied with the erase-confirm command is latched and decoded internally by the 28F008SA; erase is subsequently enabled in that block. The second $\overline{WE}$ rising edge initiates the operation ($\overline{WE}$-controlled write).

The WSM triggers the high-voltage flash-erase switch connecting the 12V supply to the source of all bits in the specified block, while all wordlines are grounded. Figure 6 shows organization of the block source switches. Fowler-Nordheim tunneling results in simultaneous erasure of all bits in the selected block.

The block source switch controls the source voltage of all bits in a particular block. This circuit is shown in Figure 7. During block erase, M2 is off and M1 pulls the source to V_{PP}. When not in erase, M1 is off and M2 pulls the source to ground. The high-voltage latch formed by M4–M7 converts the low-voltage ERASE signal to a high-voltage signal that controls M1.

The tunneling that occurs during block erase requires only a small amount of current. However, the initial current required to charge the block's large source capacitance to the erase voltage is significant. M1 is sized to limit this current yet still apply sufficient source voltage to achieve fast block-erase time.

The LOWV_{CC} signal protects the array from erasure when V_{PP} is at a high voltage but V_{CC} is below the write/erase lockout voltage (V_{LKO}). When this occurs, M3 pulls the block source to ground. The high-voltage latch is forced by M8 into the state that turns M1 off.

V_{PP} is continually monitored during all phases of the block-erase operation. If V_{PP} falls below the trip point of its high-voltage detect circuitry, erasure will not occur (or halts) and Status Register V_{PP} status (SR.3), block-erase status (SR.5) and WSM status (SR.7) bits are set to "1".

3

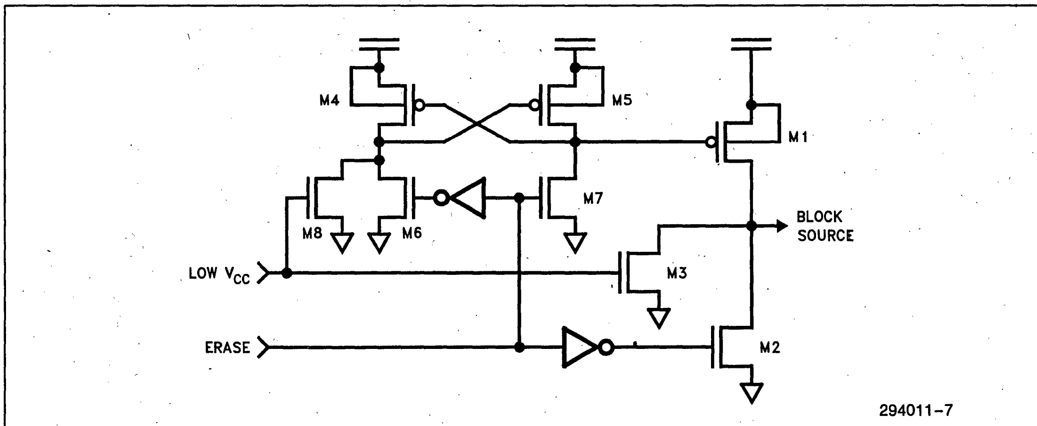


Figure 7. Block Source Switch

If SR.3 (Low V_{PP}) is set, WSM operation is inhibited. The WSM will not execute further byte-write or block-erasure sequences until the Status Register has been reset by system software. Byte-write or block-erase requests with error flags SR.4 or SR.5 set are not inhibited, but the system loses the ability to determine success. The clear Status Register command resets these bits.

After receiving the block-erase command sequence, the WSM automatically controls block pre-condition (programming all bytes to 00H within the chosen block), erase pulses and pulse repetition, timeout delays, and byte-by-byte verification of all block addresses (sequentially checked via the address counter) using an alternative sensing reference to verify margin. The internal erase and verify operations continue until the entire block is erased. A read cycle applied to the part following the block-erase command sequence outputs Status Register contents; system software can poll the Status Register to determine when block erase is complete, and if it was successful. Alternately, the system can monitor $RY/\overline{BY}$ until that output is driven high, and then poll the Status Register to determine success. Following block erasure, the device remains in Status Register read mode; a read-array command must be written to the device to access array data.

If the erase-setup command is followed with a command other than erase confirm, the device will not erase. The WSM sets both byte-write status and block-erase status bits in the Status Register to indicate an invalid sequence.

Erase Suspend/Resume

Erase suspend allows the system to interrupt block erase to read data from another array block. The ability to suspend erase and read data from another block offers the flexibility required for file system applications. Upon receiving the erase-suspend command, the CUI requests that the WSM pause at one of several predetermined points in the algorithm. Upon reaching a suspend point, the WSM sets SR.6 (erase-suspend status) and SR.7 to "1", and drives the $RY/\overline{BY}$ pin high. The system must poll the Status Register to determine if the suspend has been processed or the block erase has actually completed. Block-erase completion is indicated by SR.6 cleared to "0" and SR.7 set to "1". Read bus cycles default to Status Register read after issuing the erase-suspend command.

Once suspended, the WSM asserts a signal to the CUI which allows response to the read-array, read-status, and erase-resume commands. The system can write the read-array command allowing read access to blocks other than that which is suspended. The WSM continues to run idling in a suspended state, regardless of all control inputs except $\overline{PWD}$. $\overline{PWD}$ driven low immediately shuts down the WSM, aborting the suspended erase operation.

The erase-resume command must be issued upon completion of reads from other array blocks to continue block-erase operation. The WSM then clears SR.6 and SR.7, drives $RY/\overline{BY}$ low, and resumes erase operation from the suspension point. Read cycles following the erase-resume command output Status Register data.

Byte Write

Byte write follows a flow similar to block erase. The byte-write setup command is first written to the CUI. A second write cycle loads address and data latches. The rising edge of the second $\overline{WE}$ pulse requests that the WSM initiate activity, applying high voltage to the gates and drains of all bits to be written. Unlike block erase, byte write will proceed regardless of what data is applied on the second CUI write cycle; however, applying data FFH does not modify memory contents.

Like block erase, the WSM controls program pulses and pulse repetition, timeout delays and byte verification. Byte write and verify (with alternate sensing reference and internally-generated verify voltage) continue until the byte is written. Internal byte-write verify checks that all bits written to zero have been correctly modified; it does not check bits specified as one. Byte write cannot change existing zeros to ones; this can only be accomplished by erase.

Read bus cycles following byte write operations output Status Register data. System software, polling the Status Register, is informed of status through bits SR.3, SR.4, and SR.7. The $RY/\overline{BY}$ output can also be monitored to determine completion. The read-array command must be written to the CUI following byte write to access array data.

In a scenario similar to that described under block erasure, byte write does not occur (or halts) if V_{PP} is detected low. In such a case SR.3, SR.4, and SR.7 are set high, and no further writes can take place until the Status Register is cleared by the clear Status Register command.

DEVICE CHARACTERIZATION

AC and DC Parameters

Figures 9 through 24 show graphs of several device parameters as a function of temperature and supply voltage. The graphs illustrate that the 28F008SA has significant margin to data sheet specifications.

In particular, note Figure 9 which shows typical read performance t_{AVQV} (t_{ACC}) of the 28F008SA as a function of V_{CC} and ambient temperature. t_{ELQV} (t_{CE}) in Figure 10 and t_{GLQV} (t_{OE}) in Figure 11 are also of particular interest. Access times t_{AVQV}, t_{ELQV}, and t_{GLQV} are specified and tested with an output load of 100 pF; additional output load capacitance slows device operation.

Table 1 shows typical supply currents at room temperature for several operating modes.

Table 1. RMS Current Values

Mode	I_{CC} (V_{CC} = 5.0V, CMOS Inputs)	I_{PP} (V_{PP} = 12V)
Read	20 mA	100 μA
Byte Write	10 mA	12 mA
Block Erase	10 mA	12 mA
Standby	40 μA	100 μA
Deep Power-Down	0.20 μA	0.07 μA

Energy/Power Consumption

The system designer is primarily concerned with power consumption during block erase and byte write. Typical curves for I_{CC} and I_{PP} during block erase are shown in Figure 25. I_{CC} and I_{PP} for byte write are illustrated in Figure 26.

Byte-Write/Block-Erase Times

The 28F008SA advances byte-write and block-erase performance compared to previous flash memories. The on-chip algorithm is improved over the 28F001BX to take advantage of process enhancements. This improvement is most apparent when compared to first-generation flash parts with externally controlled algorithms. First-generation device times shown in Table 2 assume optimal system overhead, and as such are absolute best case.

Table 2. Byte-Write and Block-Erase Performance vs Previous Devices

Device	Byte-Write Time	Block-Erase Time/# Bytes	Erase Time per Kbyte
Second-Generation Flash Memory Devices(1)			
28F008SA	9 μs	1.5s/64K	23 ms
28F001BX	18 μs	3.8s/112K	34 ms
		2.1s/8K	256 ms
		2.1s/4K	513 ms
First-Generation Flash Memory Devices(2)			
28F020	16.5 μs	6.8s/256K	27 ms
28F010	16.5 μs	3.9s/128K	30 ms
28F512	16.5 μs	2.4s/64K	37 ms
28F256A	16.5 μs	1.6s/32K	51 ms

NOTES:
1. Typical measured time.
2. Times calculated based on typical erase and precondition pulse requirements, with minimum write timings. Calculations are described in Figure 8.

Figure 27 shows block-write and block-erase times at 0°C and 70°C over cycling.

DEVICE RELIABILITY

Byte-Write/Block-Erase Cycling

One of the most important reliability aspects of the 28F008SA is its capability of 100,000 write/erase cycles per block. Destructive oxide breakdown has been a limiting factor in extended cycling of thin-oxide EEPROMs. Intel's ETOX-III flash memory technology extends cycling performance through:

- Improved tunnel-oxide processing that increases charge-carrying capability tenfold.
- Significantly reduced oxide area under stress that minimizes probability of oxide defects in the region.
- Reduced oxide stress due to a lower peak electric field (lower erase voltage than EEPROM).

Reliable byte-write/block-erase cycling requires proper selection of the maximum erase threshold voltage (V_t), and maintenance of a tight distribution. Maximum erase V_t is set to 3.4V via the internal block-erase algorithm and verify circuits. Tight erase V_t distribution gives an order of magnitude of erase-time margin to the fastest erasing cell, with virtually identical erase V_t distributions at 1 and 10,000 cycles (Figure 28). Program V_t distribution is similarly consistent over cycling (Figure 29).

28F008SA array architecture enhances cycling capability by reducing gate disturb conditions on cells in unrelated blocks during byte write and erase pre-conditioning. First, only one of the two row decoders is active at any time (selected by A_{19}). Rows in the other two quadrants are grounded. Secondly, two separate internally-switched voltages supply the left and right quadrants. Only one supply (selected by block address A_{18}) is switched to programming voltages while the other remains at read voltages. This $A_{19}-A_{18}$ row decoding ensures that during byte write, 12 of the 16 blocks have a gate voltage below that required for programming.

Data Protection

The 28F008SA offers protection against accidental block erasure or byte write during power transitions. Internal circuitry creates a device insensitive to V_{PP}/V_{CC} supply power-up sequencing. $V_{PP} \leq V_{PPL}$ locks out byte-write and block-erase circuits. $V_{CC} \leq V_{LKO}$ disables CUI command writes, resets the CUI to array read mode, and holds the WSM inactive. The system designer must still guard against spurious command writes for $V_{CC} > V_{LKO}$ when $V_{PP} > V_{PPL}$.

Several strategies are available to prevent data modification in the 28F008SA. The CUI provides a degree of software write protection since memory alteration occurs only after successful completion of a two-step write sequence. $\overline{WE}$ and $\overline{CE}$ must both go active to perform this sequence; driving either high inhibits command/data writes. Secondly, the system can place the device in deep power-down mode ($\overline{PWD} = V_{IL}$) to disable command writes, reset the CUI to array read mode, and hold the WSM inactive, effectively protecting array data. Finally, the system designer may hardwire V_{PP} to V_{PPH}, or switch it to V_{PPH} only when memory updates are required. Since byte-write and block-erase circuits are disabled by $V_{PP} \leq V_{PPL}$, V_{PP} switching adds another level of data security.

SUMMARY

The 28F008SA is the first flash memory with features optimized for solid-state systems and file storage. These features include symmetrical block-erase, automation of byte write and block erase, erase suspend for data read, a deep power-down mode, a write/erase Status Register, and a dedicated RY/$\overline{BY}$ status pin. With simple microprocessor interfacing and software command sequences, the 28F008SA is the non-volatile storage solution of choice for today's designs.

OTHER REFERENCES

Related documents of interest to readers of this engineering report:

28F008SA Data Sheet (order #290429)

28F008SA-L Data Sheet (order #290435)

AP-359 "28F008SA Hardware Interfacing" (order #292094)

AP-360 "28F008SA Software Drivers" (order #292095)

AP-364 "28F008SA Automation and Algorithms" (order #292099)

ER-28 "ETOX™ III Flash Memory Technology Engineering Report" (order #294012)

SUPPLEMENTARY INFORMATION

FORMULA:

b = # bytes in a block (256K, 128K, 64K, 32K)

n = # of erase pulses required (90 pulses)

w = time for a write cycle (150 ns, t_{AVAV})

v = time to verify (6055 ns, $t_{WHGL} + t_{GLQV}$)

p = program pulse width (10 µs, t_{WHWH1}) one pulse programming assumed

e = erase pulse width (10 ms, t_{WHWH2})

Precondition and precondition verify time is:

$$b\,(2w + p + v)$$

Erase/verify, each loop where some byte does not pass verify:

$$(n - 1)\,(2w + e + v)$$

Last erase pulse:

$$(1)\,(2w + e)$$

Passing erase-verify, all bytes:

$$b\,(w + v)$$

Total time can be summarized as:

$$b\,(3w + p + 2v) + n\,(2w + e + v) - (v)$$

or substituting in times for write, verify, program and erase pulse widths:

$$b\,(22.56 \times 10^{-6}) + n\,(10.006355 \times 10^{-3}) - (6.055 \times 10^{-6})\ \text{Seconds}$$

Figure 8. Erase Time Calculations for First-Generation Flash Memories

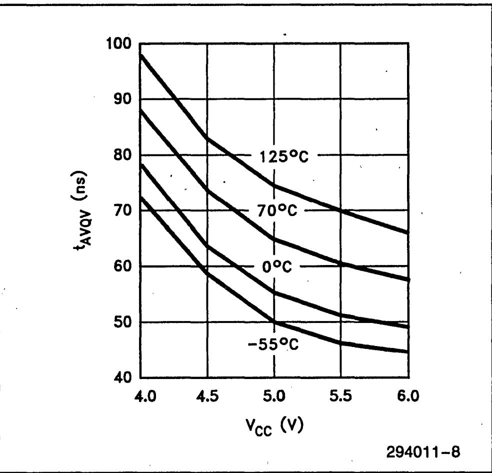


Figure 9. t_{AVQV} (t_{ACC}) vs V_{CC} and Temperature

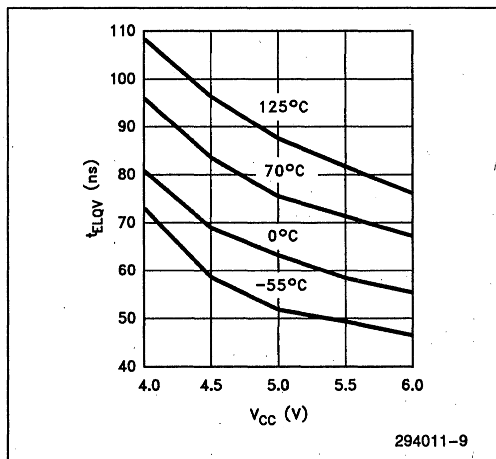


Figure 10. t_{ELQV} (t_{CE}) vs V_{CC} and Temperature

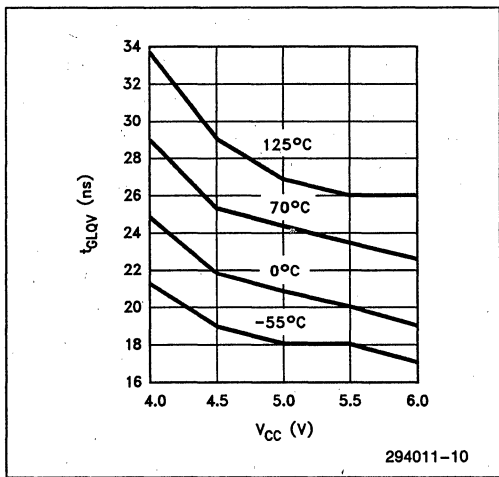


Figure 11. t_{GLQV} (t_{OE}) vs V_{CC} and Temperature

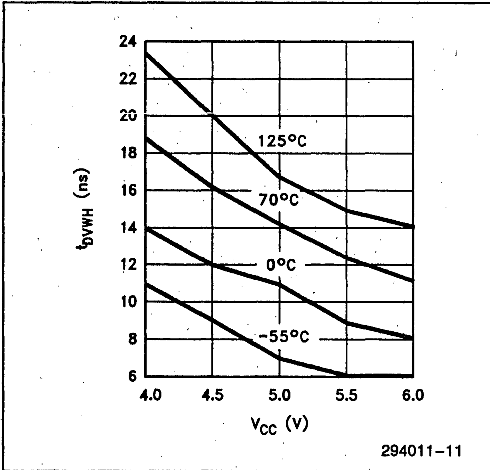


Figure 12. t_{DVWH} (t_{DS} $\overline{WE}$) vs V_{CC} and Temperature

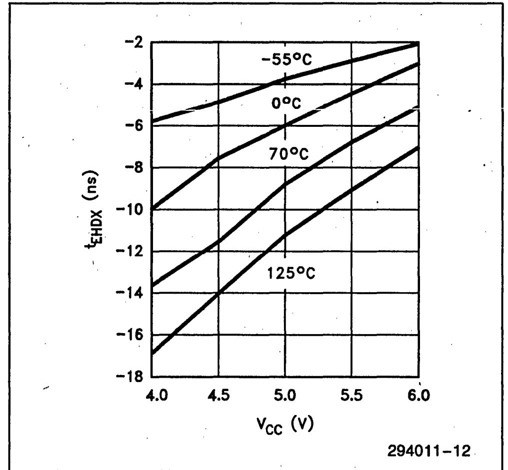


Figure 13. t_{EHDX} (t_{DH} $\overline{CE}$) vs V_{CC} and Temperature

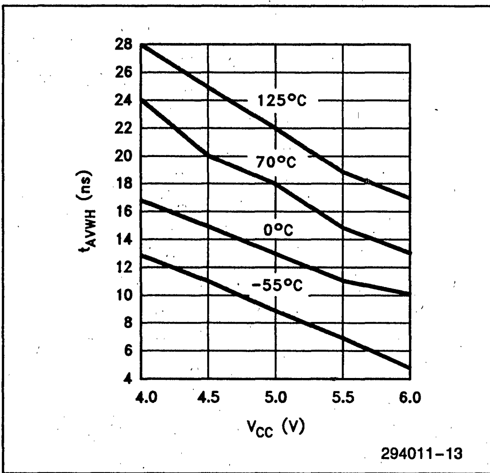


Figure 14. t_{AVWH} (t_{AS} $\overline{WE}$) vs V_{CC} and Temperature

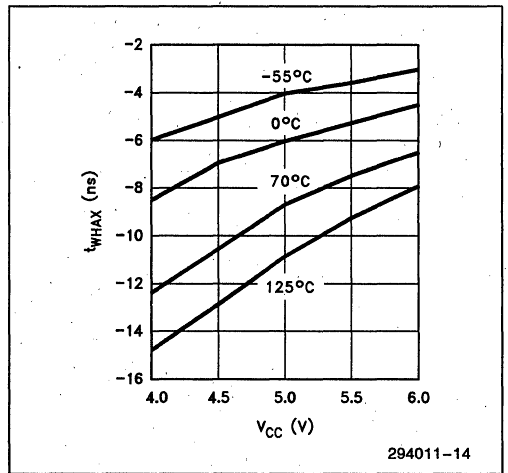


Figure 15. t_{WHAX} (t_{AH} $\overline{WE}$) vs V_{CC} and Temperature

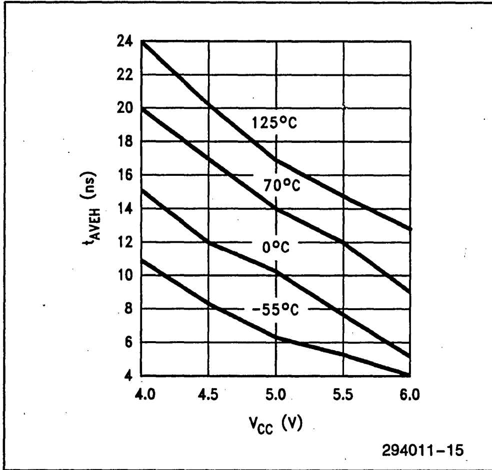


Figure 16. t_{AVEH} (t_{AS} $\overline{CE}$) vs V_{CC} and Temperature

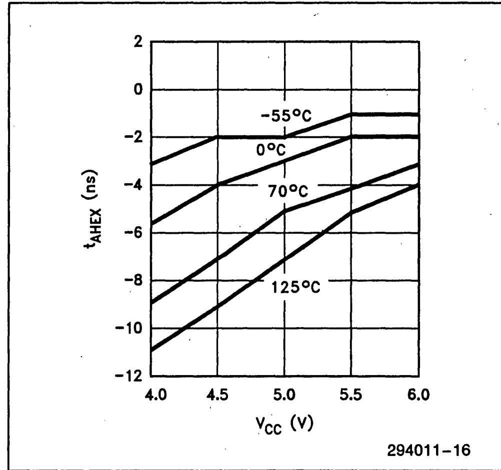


Figure 17. t_{AHEX} (t_{AH} $\overline{CE}$) vs V_{CC} and Temperature

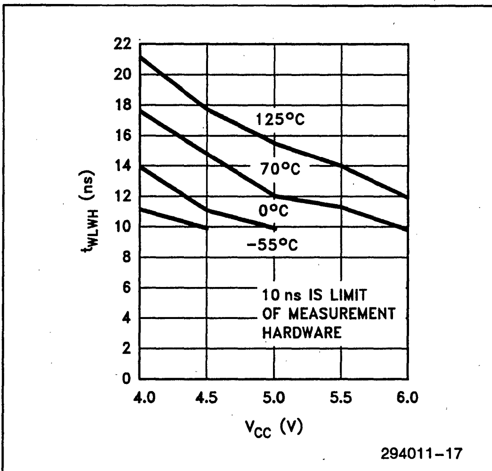


Figure 18. t_{WLWH} (t_{WP}) vs V_{CC} and Temperature

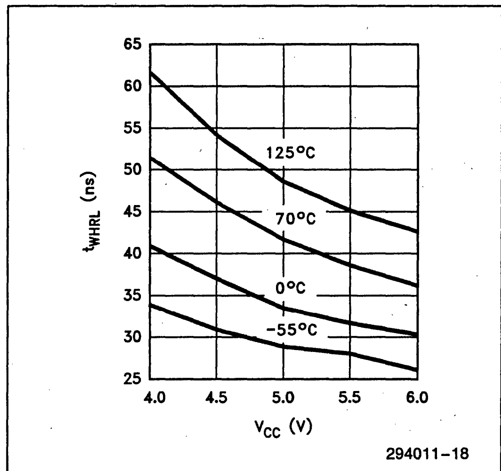


Figure 19. t_{WHRL} vs V_{CC} and Temperature

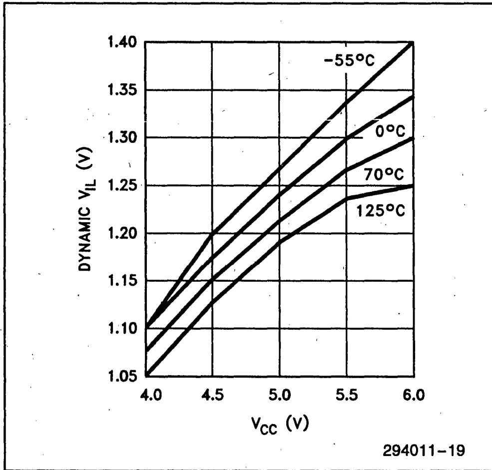


Figure 20. Dynamic V_{IL} vs V_{CC} and Temperature

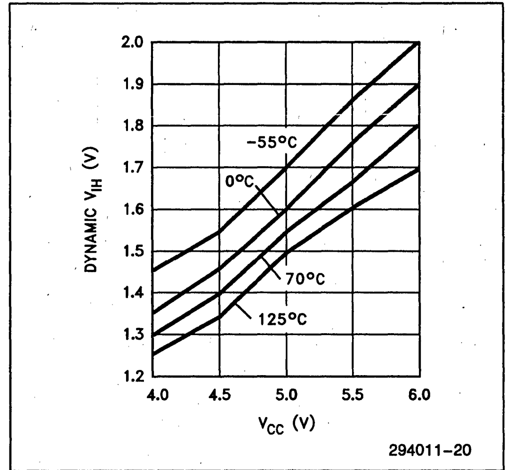


Figure 21. Dynamic V_{IH} vs V_{CC} and Temperature

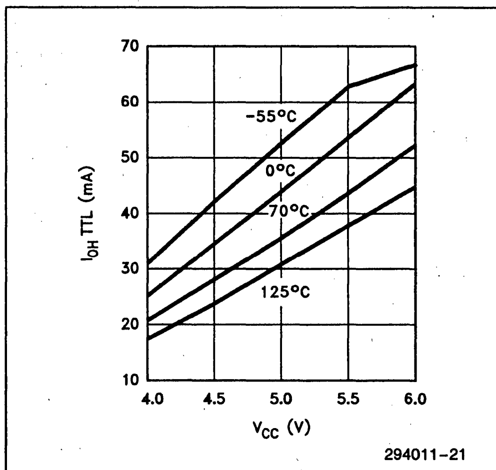


Figure 22. I_{OH} TTL vs V_{CC} and Temperature (V_{OH} = 2.4V)

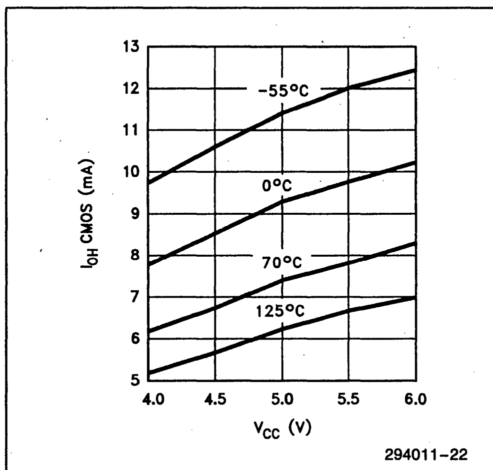


Figure 23. I_{OH} CMOS vs V_{CC} and Temperature (V_{OH} = V_{CC} − 0.4V)

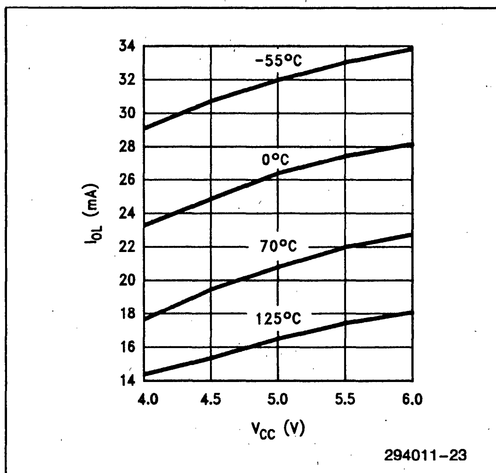


Figure 24. I_{OL} vs V_{CC} and Temperature (V_{OL} = 0.45V)

3

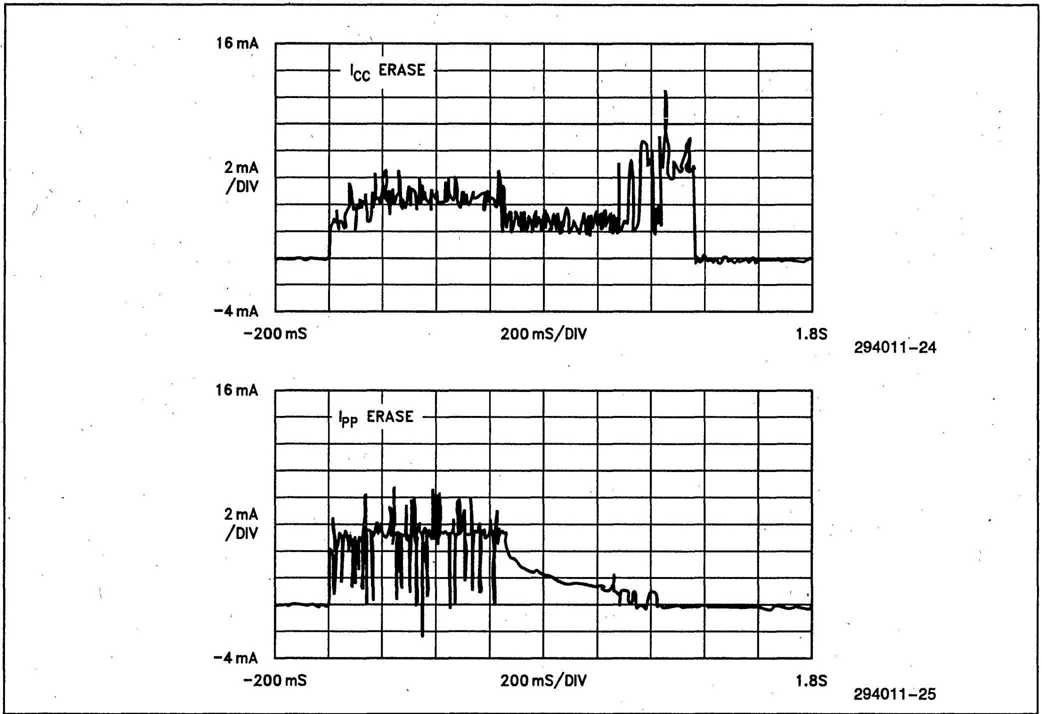
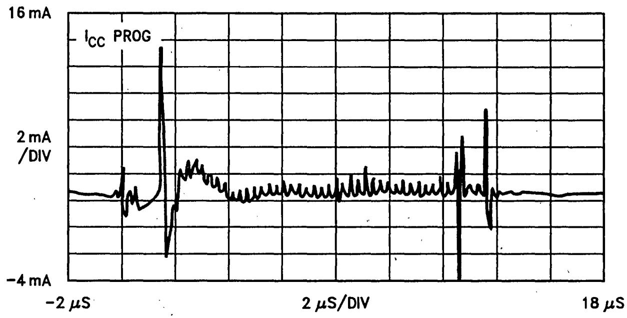


Figure 25. I_{CC} and I_{PP} under Block-Erase Operation



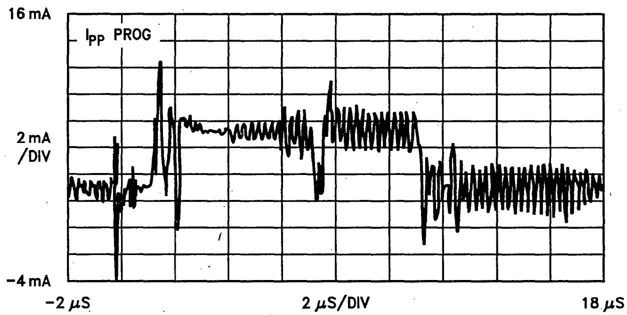


Figure 26. I_{CC} and I_{PP} under Byte-Write Operation

3

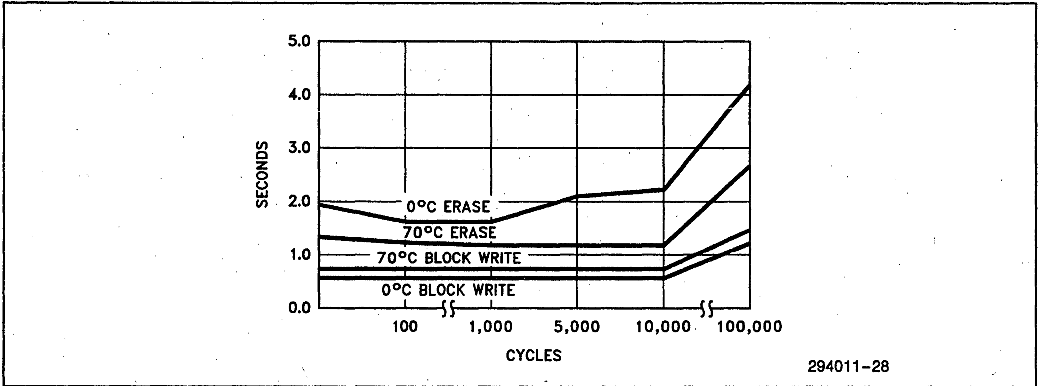


Figure 27. Write and Erase Times vs Cycling

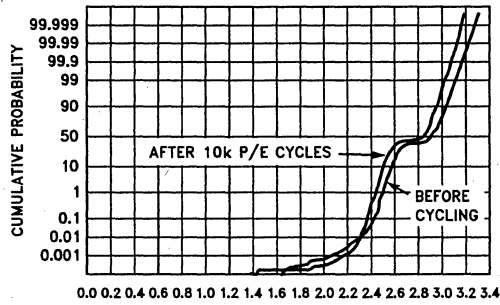


Figure 28. Erase V_t vs Cycles

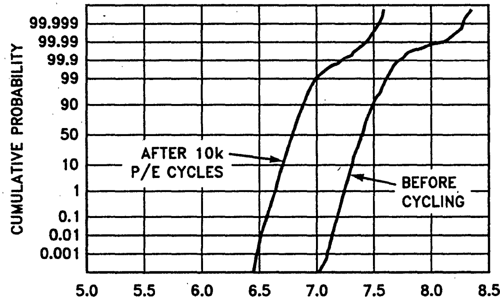


Figure 29. Program V_t vs Cycles

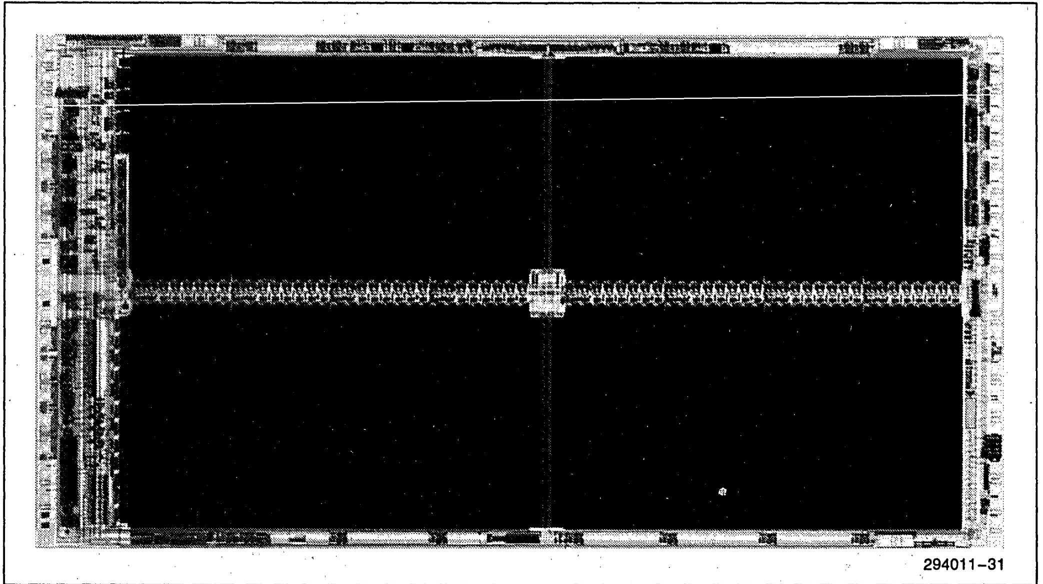
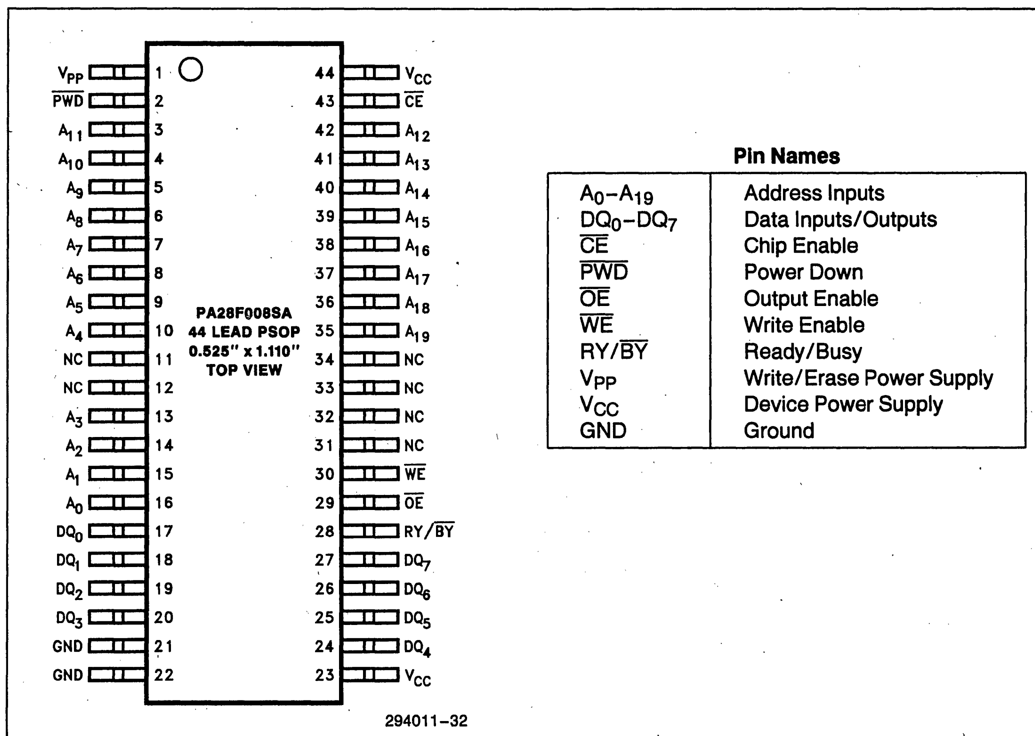

294011-31

Figure 30. 278F008SA Die Photo



Pin Names

A_0-A_{19}	Address Inputs
DQ_0-DQ_7	Data Inputs/Outputs
$\overline{CE}$	Chip Enable
$\overline{PWD}$	Power Down
$\overline{OE}$	Output Enable
$\overline{WE}$	Write Enable
$RY/\overline{BY}$	Ready/Busy
V_{PP}	Write/Erase Power Supply
V_{CC}	Device Power Supply
GND	Ground

Figure 31. PSOP Lead Configuration

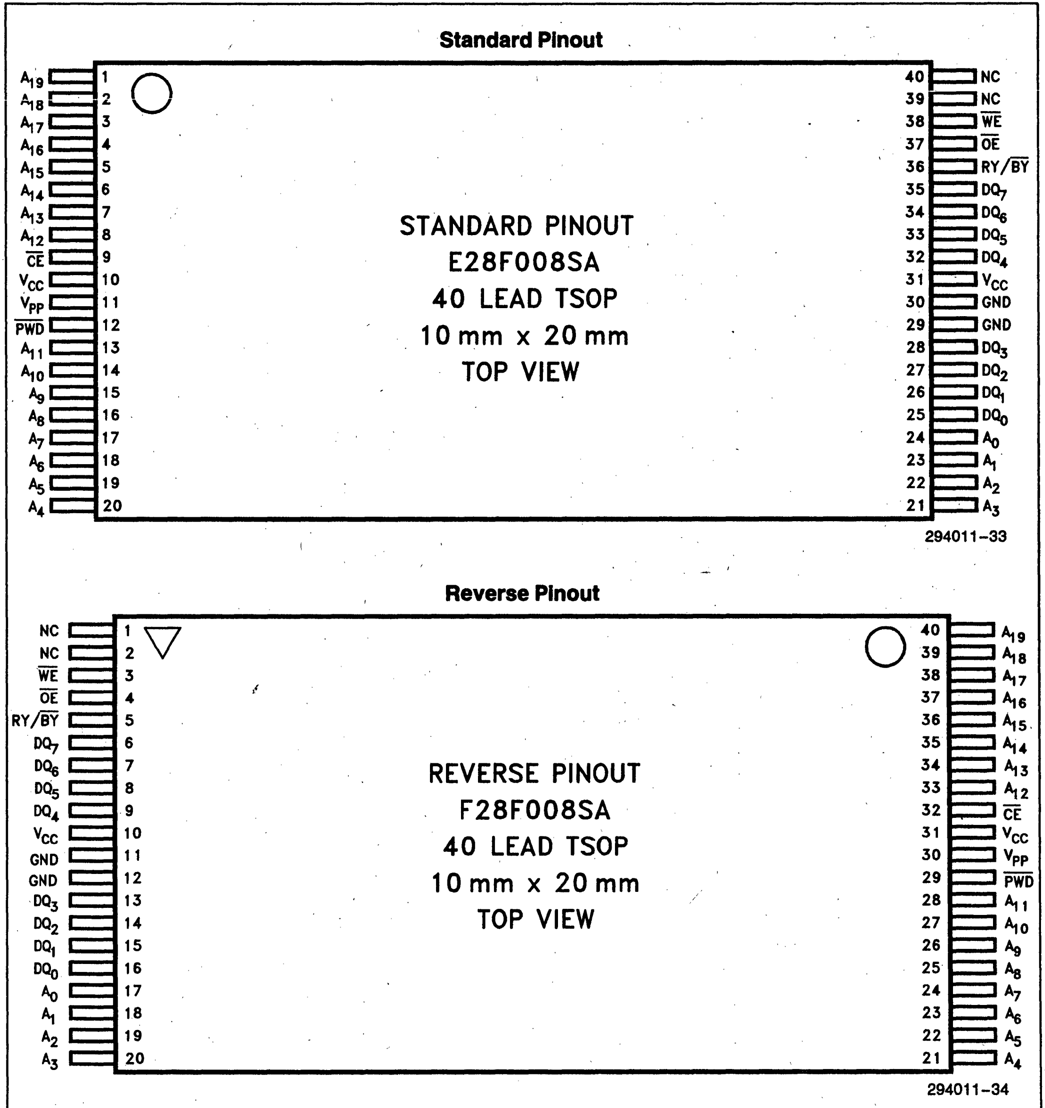


Figure 32. TSOP Lead Configuration

Wordlines are numbered sequentially from top to bottom. Addresses A_9-A_0 sequentially decode wordlines: block address A_{19} selects between upper and lower row-decoder. Wordlines 0–1023 serve the left and right quadrants at top of device; 1024–2047 serve the lower quadrants.

Table 3. Row Address Bitmap

A_{19}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Wordline
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	0	0	0	1	1	3
0	0	0	0	0	0	0	0	1	0	0	4
0	0	0	0	0	0	0	0	1	0	1	5
0	0	0	0	0	0	0	0	1	1	0	6
0	0	0	0	0	0	0	0	1	1	1	7
0	0	0	0	0	0	0	1	0	0	0	8
0	0	0	0	0	0	0	1	0	0	1	9
0	0	0	0	0	0	0	1	0	1	0	10
0	0	0	0	0	0	0	1	0	1	1	11
0	0	0	0	0	0	0	1	1	0	0	12
0	0	0	0	0	0	0	1	1	0	1	13
0	0	0	0	0	0	0	1	1	1	0	14
0	0	0	0	0	0	0	1	1	1	1	15
0	0	0	0	0	0	1	0	0	0	0	16
0	0	0	0	0	0	1	•	•	•	•	•
0	0	0	0	0	0	1	1	1	1	1	31
0	•	•	•	•	•	•	•	•	•	•	•
0	1	1	1	1	1	1	0	0	0	0	1008
0	1	1	1	1	1	1	•	•	•	•	•
0	1	1	1	1	1	1	1	1	1	1	1023
1	0	0	0	0	0	0	0	0	0	0	1024
1	0	0	0	0	0	0	•	•	•	•	•
1	0	0	0	0	0	0	1	1	1	1	1039
1	0	0	0	0	0	1	0	0	0	0	1040
1	•	•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	1	1	1	1	1	2047

Blocks are numbered sequentially right to left, top to bottom. Columns within a block are numbered left to right. I/Os within a block are numbered from 0–7 left to right.

Table 4. Block Address Bitmap

A_{19}	A_{18}	A_{17}	A_{16}	Block	Columns
0	0	0	0	0	0–511
0	0	0	1	1	512–1023
0	0	1	0	2	1024–1535
0	0	1	1	3	1536–2047
0	1	0	0	4	2048–2559
0	1	0	1	5	2560–3071
0	1	1	0	6	3072–3583
0	1	1	1	7	3584095
1	0	0	0	8	4096–4607
1	0	0	1	9	4608–5119
1	0	1	0	10	5120–5631
1	0	1	1	11	5632–6143
1	1	0	0	12	6144–6655
1	1	0	1	13	6656–7167
1	1	1	0	14	7168–7679
1	1	1	1	15	7680–8191

Columns are numbered from left to right across the top quadrants, and left to right across the bottom quadrants. Addresses A_{15}–A_{10} sequentially count columns. Columns are listed for block 0; other blocks are counted similarly.

Table 5. Column Address Bitmap

						Column In							
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	I/O_0	I/O_1	I/O_2	I/O_3	I/O_4	I/O_5	I/O_6	I/O_7
0	0	0	0	0	0	0	64	128	192	256	320	384	448
0	0	0	0	0	1	1	65	129	193	257	321	385	449
0	0	0	0	1	0	2	66	130	194	258	322	386	450
0	0	0	0	1	1	3	67	131	195	259	323	387	451
0	0	0	1	0	0	4	68	132	196	260	324	388	452
•	•	•	•	•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	0	62	126	190	254	318	382	446	510
1	1	1	1	1	1	63	127	191	255	319	383	447	511

intel®

ENGINEERING REPORT

ER-28

October 1992

3

ETOX™ III Flash Memory Technology

ALAN BUCHECKER
MARK NEWMAN
MEMORY COMPONENTS DIVISION

Order Number: 294012-002

ETOX™ III Flash Memory Technology

CONTENTS PAGE

INTRODUCTION 3-593

ETOX™ III FLASH MEMORY CELL ... 3-593

Cell Processing 3-594

Byte Write 3-594

Block Erase 3-595

ETOX™ III PROCESS CHARACTERISTICS 3-595

Write/Erase Performance with Voltage and Temperature 3-595

Write/Erase Cycling 3-596

Process Variation 3-597

Electrical Testing 3-597

CONTENTS PAGE

ETOX™ III MEMORY ARRAY CONSIDERATIONS 3-597

28F008SA Array Architecture 3-597

28F008SA Byte-Write Operation 3-597

28F008SA Block-Erase Operation 3-597

28F008SA Cell Voltage Threshold 3-597

FLASH VS. OTHER SEMICONDUCTOR MEMORY TECHNOLOGIES 3-598

SRAM and DRAM 3-600

EPROM 3-601

EEPROM 3-601

NAND 3-602

SUMMARY 3-602

OTHER REFERENCES 3-602

APPENDIX A 3-603

INTRODUCTION

Intel's ETOX™ III (EPROM tunnel oxide) Flash Memory technology builds on previous flash and EPROM processes spanning over two decades of manufacturing experience. This third-generation process produces devices based on 0.8 μm photolithography. Intel's Flash Memories combine EPROM programming with EEPROM-like in-system electrical erasure. This functionality, experience and technology yield a versatile non-volatile memory that is highly reliable and cost effective.

ETOX III cell integrity enables applications requiring 100,000 write/erase cycles. New designs incorporate array blocking schemes and on-chip automation of write and erase to simplify customer designs and software interface. These features combine with existing Intel Flash Memory technology to produce a device that can be termed a block-alterable non-volatile RAM. Access time (t_{ACC}) and die size decrease via this smaller photolithography, making new Intel Flash Memories competitive with DRAM in read speed and cost.

This report describes the fundamentals of Intel's ETOX III Flash Memory cell. It provides insight into device reliability and performance enhancements based on ETOX III advances, and compares other semiconductor memory technologies.

The Intel 28F008SA 8-Mbit Flash Memory is the first ETOX III product offering. This report references the 28F008SA to explain device-level concepts, and ends by highlighting important flash memory application trends.

ETOX™ III FLASH MEMORY CELL

ETOX III is a 0.8 μm double-polysilicon N-well and P-well CMOS process. This lithographic advance improves memory cell/array compaction more than two-fold over its predecessor, the 1.0 μm ETOX II process. ETOX III-aided compaction allows for a 4x product density growth given current packaging constraints.

Second-generation device architecture (see Appendix A) and 0.8 μm geometries increase byte-write and read access performance by 2x over ETOX II products. Double-metal technology enhances these improvements by aiding die size compaction, and wordline strapping. EPI wafer processing, which reduces latch-up, also factors into this performance boost by shrinking transistor layout.

Cell Processing

Intel's single-transistor Flash Memory cell stores charge on a floating polysilicon gate. Dimensions of 2.5 μm by 2.9 μm make an ETOX III cell measuring 7.25 μm^2. Cell layout locates the polysilicon select gate above the floating gate (Figure 1). Tungsten silicide, deposited on the select gate, reduces wordline resistance. Two dielectrics isolate metal-1 from the select gate.

Inter-poly dielectrics of oxide and nitride isolate the floating gate from the select gate. A very thin tunnel oxide ($\sim$ 100Å) separates the floating gate from its silicon interface. Both the floating and select gates have additional isolation between them and their respective source/drain regions. A deeper source diffusion prevents breakdown during erase operations. In the array metal-2 straps the wordline to enhance access times. As with ETOX I and ETOX II, metal-1 carries bitline data to the sense amps and routes voltages to cell sources.

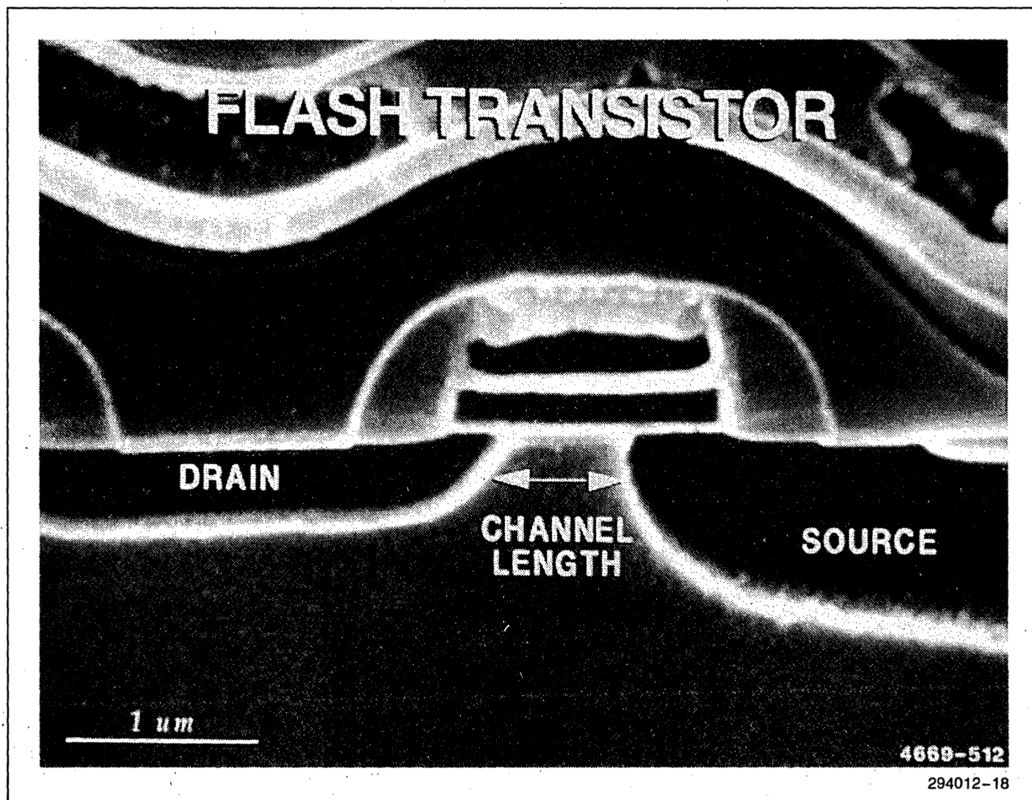


Figure 1. ETOX™ III Flash Memory Cell (Side View)

Byte Write

Writing data to an addressed byte transitions selected cells from the "1" (erased or no charge) state to the "0" (charged) state. This involves a programming mechanism called channel hot-electron injection. When programming (Figure 2), a cell's select gate (wordline) connects to the external programming supply voltage (V_{PP} at 12V). The drain (bitline) sees an intermediate level ($\sim V_{PP}/2$), while the source is at ground. V_{PP} on the select gate capacitively couples to the floating gate through the intervening dielectric. This coupling raises the floating gate to a programming voltage, inverting the channel underneath.

The channel electrons now have a higher drift velocity, with resulting increased kinetic energy. Collisions between these electrons and substrate atoms heat the silicon lattice. At the programming bias voltage, the electrons cannot transfer their kinetic energy to the atoms fast enough to maintain a thermal balance. They become "hotter," and many scatter toward the tunnel oxide. These electrons overcome the 3.1eV (electron voltage) tunnel oxide barrier and accumulate onto the floating gate.

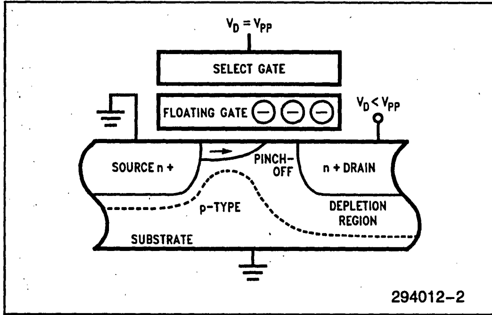


Figure 2. ETOX™ III Flash Memory Cell during Programming (Side View)

The electrons stored on the floating gate raise the turn-on voltage threshold (V_t) of that cell. During device read operations this transistor remains in the off state. A "0" results at the output because the "off" cell does not pass current, causing the bitline to electrically stay at/pull-up to the V_{CC} read voltage.

The internal programming pulse on ETOX III is 4 μs (excluding WSM overhead), reduced from 10 μs on ETOX II devices. This optimized pulse width yields faster byte-write times and greater cycling reliability. Like previous ETOX products, the automated WSM allows for the occasional byte requiring more than one pulse.

Block Erase

During erasure, electrons are pulled off selected memory cells simultaneously. The erase process ("Fowler-Nordheim" tunneling) starts by routing V_{PP} to the source, ground to the select gate, and floats the drain (Figure 3).

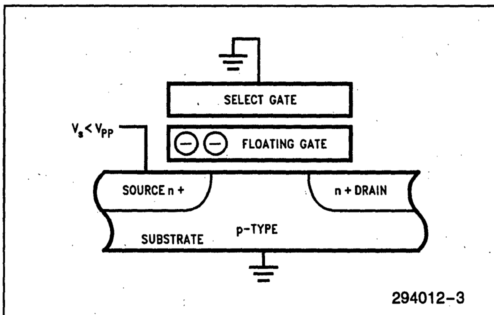


Figure 3. ETOX™ III Flash Memory Cell during Erase (Side View)

While biased in this fashion, electrons tunnel off the floating gate. They pass through the thin oxide to the source, lowering that cell's V_t. During a read operation, the resulting "1" at the output corresponds to an "on" cell discharging its bitline through the grounded source.

Erase automation sets the internal V_{PP} pulse to 10 ms. The WSM provides sufficient 10 ms pulses, and automatically verifies all memory cells in a given block. This optimized pulse width enhances block-erase time and cycling endurance.

ETOX™ III PROCESS CHARACTERISTICS

Intel leverages over two decades of EPROM/flash technology and manufacturing experience to produce reliable memory products. Refined processing techniques inherent to new Intel memory technologies and continuous improvements in process control ensure tunnel oxide quality. A scaled substrate EPI thickness reduces product latch-up. Double-metal technology requires improved planarization processing, which in turn enhances moisture performance. Additionally, decreases in defect density show lasting cell integrity in cycling and data retention.

Write/Erase Performance with Voltage and Temperature

Voltage affects byte-write and block-erase performance. Maximum V_{PP} improves byte-write and block-erase times. Figure 4 shows little difference in block-write time versus V_{PP}, but visible differences in block-erase performance. A secondary and negligible effect results from the operating supply voltage (V_{CC}). Byte-write and block-erase times are guaranteed to specification across minimum and maximum voltage levels.

Temperature also affects byte-write and block-erase performance. Low temperatures cause block-erase times to increase and byte-write times to improve (Figure 4). When cold, the breakdown voltage at the source lowers, clamping the external voltage applied for block erase. This nets a lower potential between the source and gate, slowing the tunneling process. Increased erase time results from the WSM providing extra pulses. Although electron mobility decreases at hotter temperatures, typical cells still require only one programming pulse.

Write/Erase Cycling

Intel designs extended cycling capability into its ETOX III products. For example, the 28F008SA is designed for 100,000 write/erase cycles on each of its sixteen 64-Kbyte blocks. Low electric fields, advanced low-defect oxides, and minimal oxide area per cell combine to greatly reduce oxide stress and the probability of failure.

From a performance perspective, an intrinsic mechanism occurs in long-term cycling that cause byte-write and block-erase times to increase (Figure 5), but still conform to specification. Specifically, hot electrons from programming trap in the tunnel oxide near the drain junction. This creates a negatively-charged barrier, slowing hot-electron injection. Similarly, erase times increase due to charge trapping near the source junction, making tunneling less efficient. The robustness of ETOX III minimizes these effects. Write and erase times remain consistent over the first 10,000 cycles, and typically double as the device nears 100,000 cycles.

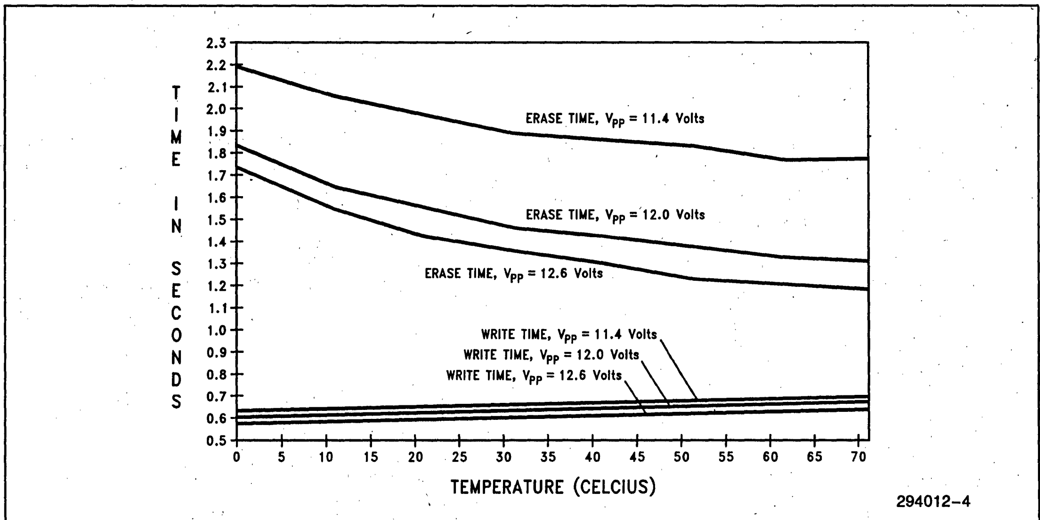


Figure 4. 28F008SA Block Write and Erase Times vs Temperature and V_{PP}

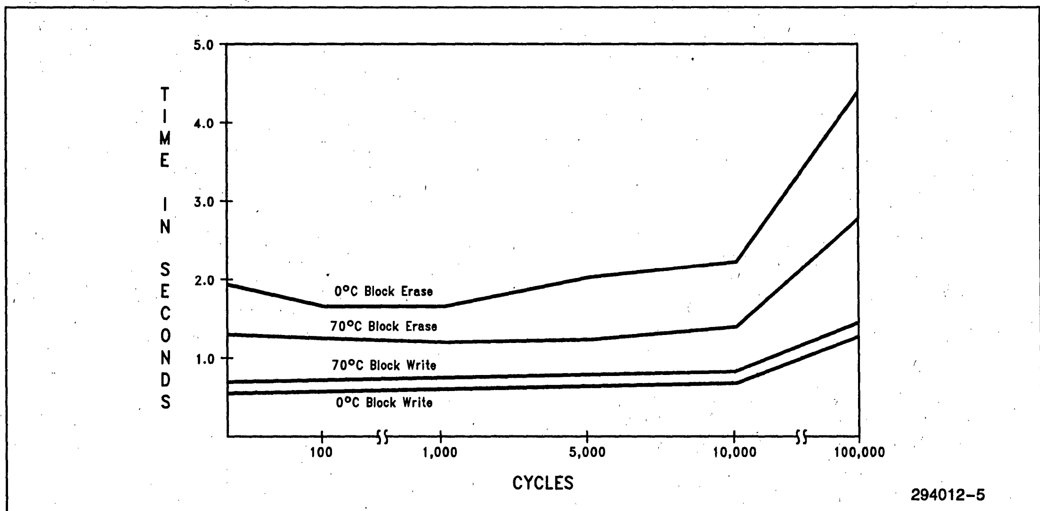


Figure 5. Write and Erase Times vs Cycling

Process Variation

Intel's process control of ETOX III critical cell dimensions keep write and erase electrical characteristics on target with little variance. This capability produces devices with consistent byte-write/block-erase times making product performance predictable.

Over the processes lifetime, internal cell dimensions may exhibit some small variance. The primary process variable affecting erase is tunnel oxide thickness. Channel length (L_{eff}) has the largest impact on programming. Outgoing product testing ensures performance to specification regardless of these minor variances.

Electrical Testing

Electrical testing provides added value to Intel Flash Memory products. This elaborate testing gives insight to device characteristics and ensures product longevity. Moreover, flash reliability qualifications assure product performance and long-term durability.

Electrical erase at wafer and package test allow high confidence of detecting oxide defects. This electrical testing also ensures that outgoing products perform to specified temperature extremes. Optimization of ETOX III process and designs, developed from previous ETOX generations and continuous data collection, yield a very manufacturable and cost-effective technology.

ETOX™ III FLASH MEMORY ARRAY CONSIDERATIONS

Intel Flash Memory architecture has evolved from bulk arrays (full-chip electrical erase), to array segmentation referred to as blocking. Blocking divides the device memory array into smaller sections that function as individually-erasable units.

28F008SA Array Architecture

Figure 6 illustrates the 28F008SA. Sixteen equal 64-Kbyte blocks make up this 8,338,608-bit memory array. Each block consists of 512 columns by 1024 rows. Columns in each block are further subdivided into eight input/outputs, each containing 64 columns. Typical block-erase and block-write times for this device are 1.6 and 0.6 seconds. Typical byte-write time, including WSM overhead, is 9 μs.

28F008SA Byte-Write Operation

During byte write, column address decoding determines which eight bitlines of a target block connect to the intermediate programming voltage ($\sim V_{PP}/2$). Row decoding determines wordline drive to V_{PP}. For example, writing a byte of data in block 0 sets that one wordline to V_{PP}. Address selection sets all other wordlines in the array to ground. Array decoding/layout and cell durability assure device performance and reliability, and long-term data retention.

28F008SA Block-Erase Operation

Erasing a block involves simultaneous erasure of all bits in that block. For example, erasing block 0 sets all block 0 sources to V_{PP} and all block 0 wordlines to ground. Address decoding drives all other wordlines, bitlines and sources in the array to ground. This eliminates the possibility of corruption to data stored in non-selected blocks.

28F008SA Cell Voltage Threshold

Efficient blocking layout and optimized decoding result in smaller die size. Blocking tightens program and erase V_ts by dividing process variation into smaller regions. The internal WSM algorithms and their associated program/erase and verify circuits also keep V_t variations to a minimum. This design for manufacturing approach increases product stability.

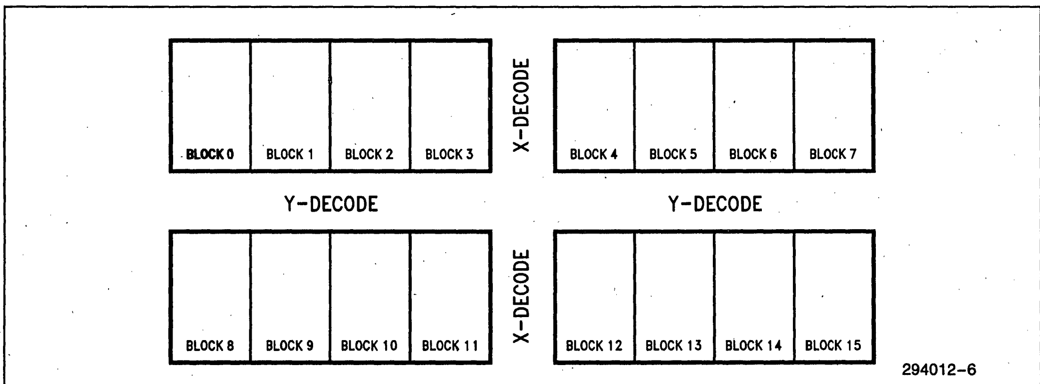


Figure 6. Intel 28F008SA Memory Array Layout

FLASH VS. OTHER SEMICONDUCTOR MEMORY TECHNOLOGIES

Intel's scaling advances in flash memory manufacturing and design provide optimal cell/array compaction. In roughly twenty years, Intel non-volatile memory density has gone from 2,048 bits to 8,388,608 bits, a 4096x increase. Figure 7 compares other memory types to show relative density progression. The fast ramp in ETOX flash memory density results from its similarity to EPROM.

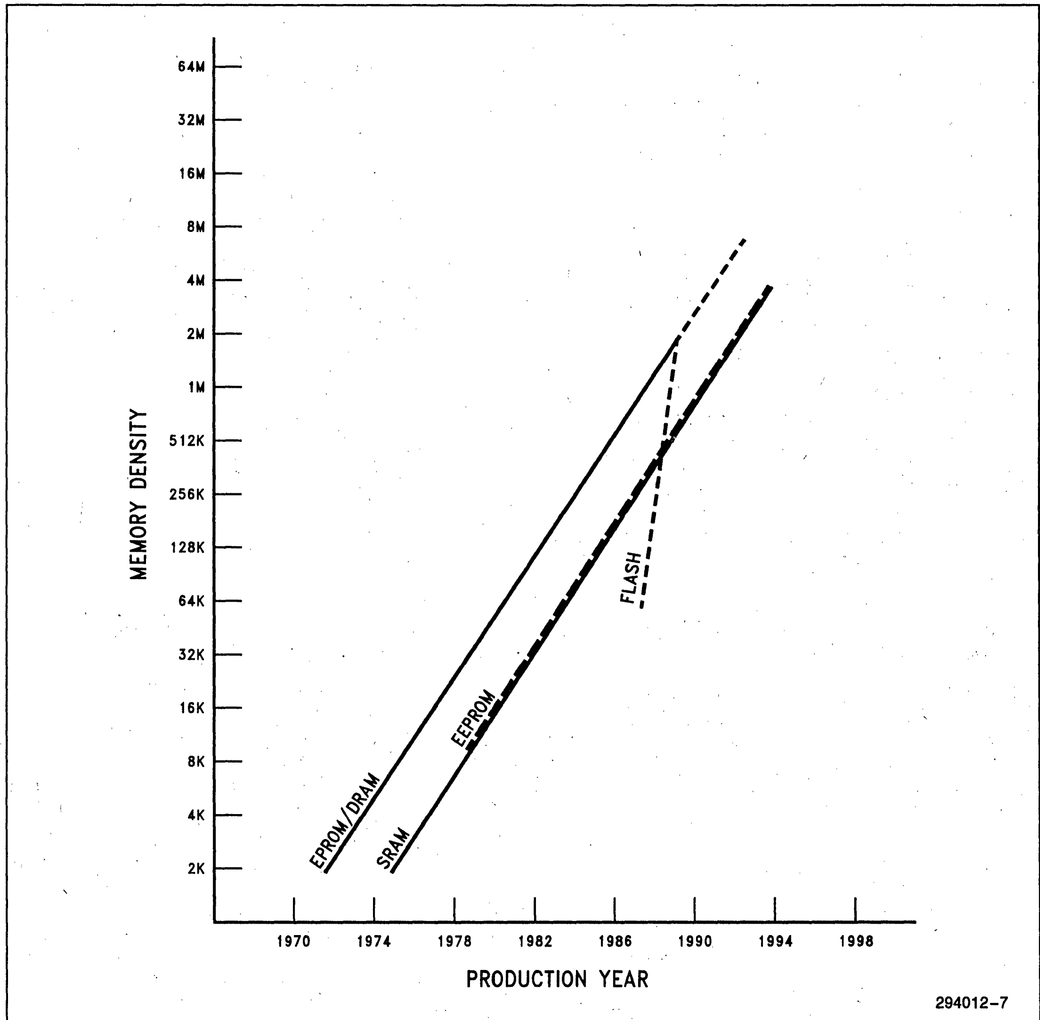


Figure 7. Memory Evolution

Figure 8 illustrates the relationship between cell sizes of different memory types and minimum geometries. As dimensions scale, certain memory types become cellsize limited (i.e. some components cannot shrink proportionally). The memory cost per bit learning curve shows flash in a strong position. This curve, shown in Figure 9, reflects how Intel's experience reduces cost for increased memory density.

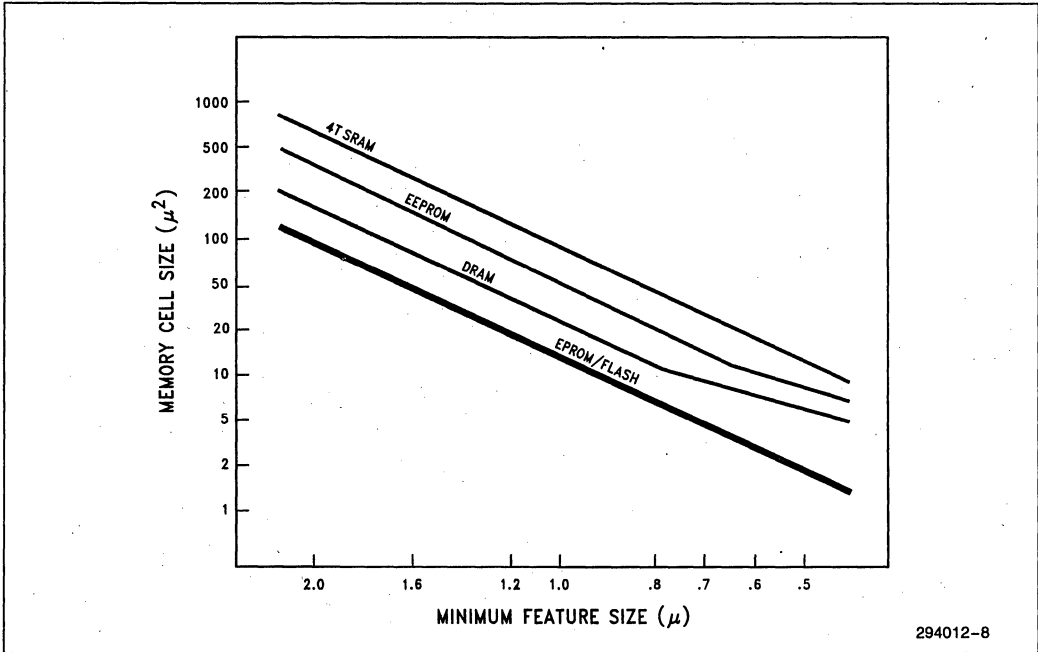


Figure 8. Memory Cell Size Trends

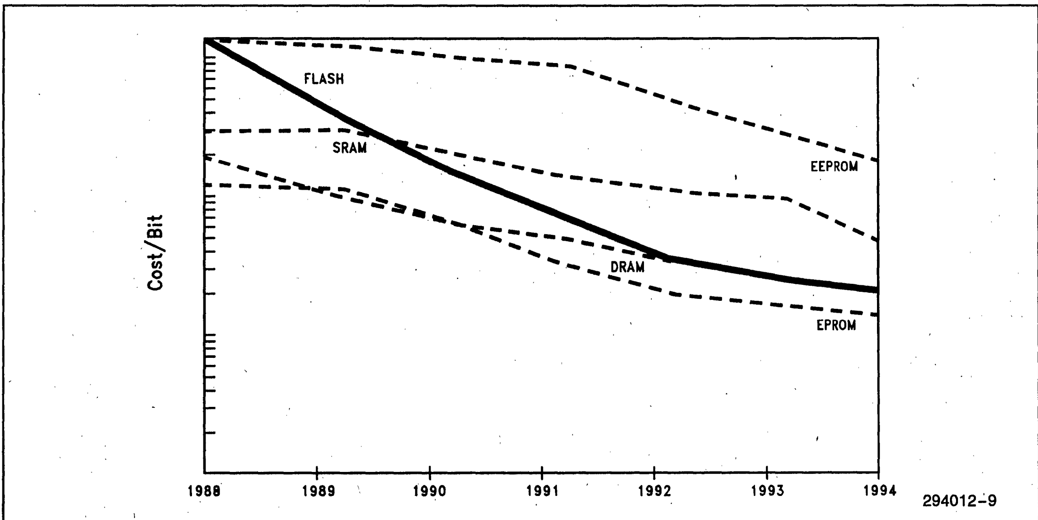


Figure 9. Memory Cost/Bit Learning Curves

Since the late 1980s, a new memory sub-system has arrived on the market, offering an alternative to high-density file system media. Intel's Series 2 Flash Memory Cards take advantage of the 28F008SA and its second-generation architecture to provide card densities of up to 20 Mbytes and new functionality. This relatively new technology offers a solid-state file system (Figure 10) that will double in density with new ETOX generations.

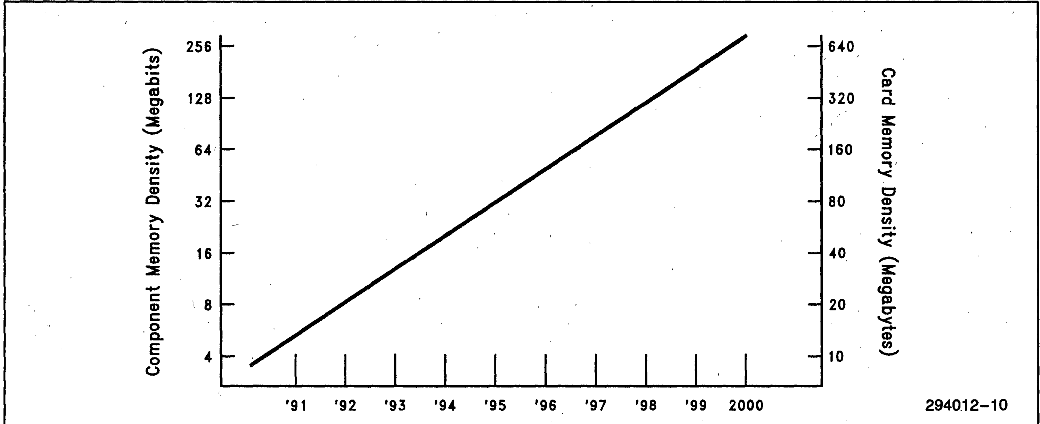


Figure 10. ETOX Component and Memory Card Density Over Time

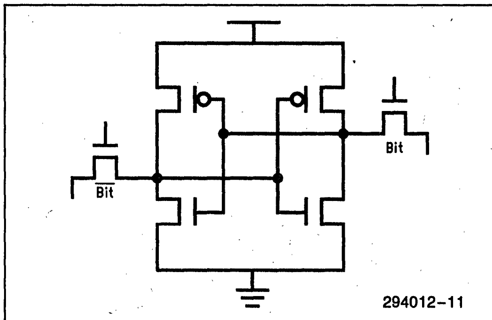


Figure 11. Six-Transistor SRAM Cell Schematic

SRAM and DRAM

SRAM and DRAM have fast read/write speeds. Both are volatile memories requiring continuous power to retain data. Standard SRAMs (Figure 11) require four to six transistors for each flip-flop cell. This greatly reduces memory capacity per unit area, raising product cost for a given density.

DRAM requires constant refresh of the capacitor-like storage mechanism (Figures 12 and 13) due to leakage currents and read operations. Charge storage requirements limit size reduction of the capacitor, which in turn limits memory array compaction. With smaller geometries, DRAM cell structures are more complex, requiring more process manufacturing steps.

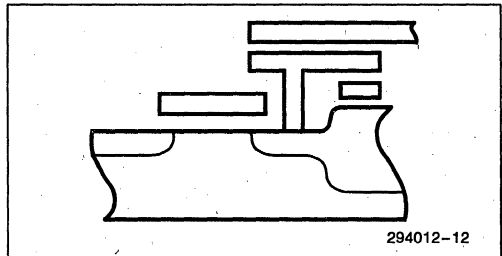


Figure 12. Stacked DRAM Cell (Side View)

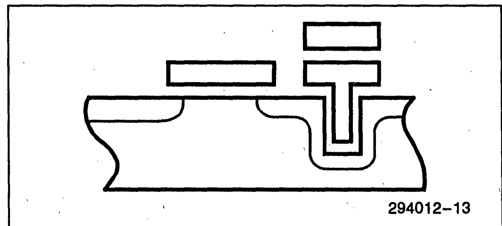


Figure 13. Trenched DRAM Cell (Side View)

Most DRAMs require a read parity bit for two reasons. First, alpha particle strikes can disturb cells by ionizing radiation, resulting in lost data. Second, when reading DRAM, the cell's storage mechanism capacitively shares its charge with the bitline through a select transistor. This creates a small voltage differential to be sensed during read access. This low voltage differential can also be influenced by nearby bitline voltages and device noise.

ETOX III floating-gate technology electrically isolates the substrate from the charge storage mechanism. Unlike DRAM, floating-gate charge determines cell V_ts, which in turn controls bitline voltages. This allows flash memory read sensing to easily detect cell V_ts. A current swing of 70 μA, from strong cell V_ts, make bitline voltage transitions a key factor to fast read speeds.

Most DRAMs require high active power consumption. Charge storage requirements and read signal strength constrain DRAM cell compaction. Low-power ETOX III flash memory has a simple single-transistor cell with only minor scaling limitations through the year 2000. This results in a mainstream memory that does not need power to retain data.

EPROM

Intel's EPROM and Flash Memory cells share a common stacked-gate profile (Figure 14), with two basic differences relating to their respective erase mechanisms. EPROM requires ultraviolet light to erase; flash erases electrically. For this reason, flash has a thinner cell oxide to allow Fowler-Nordheim tunneling, and a deeper source junction to prevent breakdown during erase.

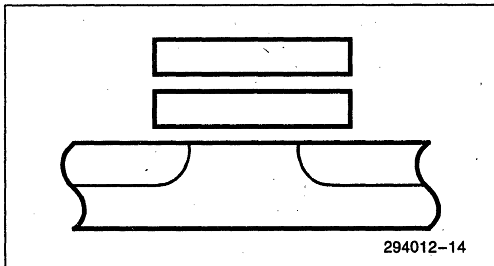


Figure 14. EPROM Cell (Side View)

EPROM technologies that migrate toward smaller geometries make floating-gate discharge (erase) via UV exposure increasingly difficult. One problem is that the width of metal bitlines cannot reduce proportionally with advancing process technologies. EPROM metal width requirements limit bitline spacing, thus reducing the amount of high-energy photons that reach charged cells. EPROM products built on submicron technologies will face longer UV exposure times.

Intel's ETOX III technology employs double-metal processing to strap wordlines in metal for improved read performance. This advance is not likely to appear on EPROM because it would block even more UV light. Since flash memory electrically erases, it eliminates these concerns. Moreover, flash electrical erasure eliminates the UV window and its associated cost, and allows for the most advanced and innovative plastic surface-mount packaging solutions.

EEPROM

Conventional two-transistor EEPROM cells (Figures 15 and 16) limit layout density. This is primarily due to the second transistor (bit select) and associated decoding required for single byte program and erase capability. Technology design requirements make EEPROM cells, like triple poly, significantly larger than flash. Typical EEPROM technologies are more complex, making wafer manufacturing difficult and expensive.

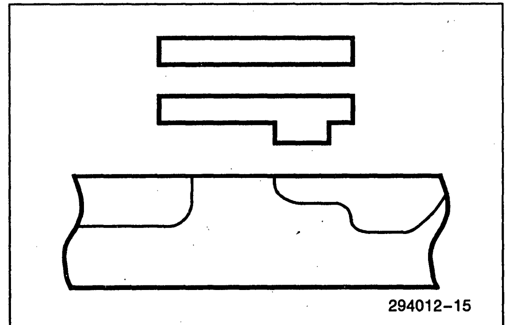


Figure 15. Flotox EEPROM Cell (Side View)

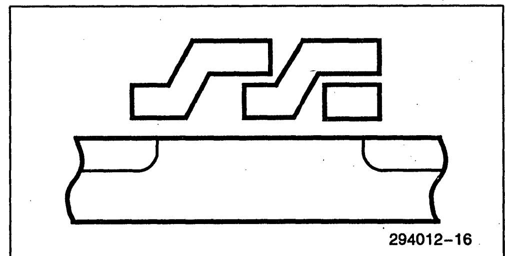


Figure 16. Triple Poly EEPROM Cell (Side View)

Because of their traditional application, EEPROMs use a very high internal voltage (17V to 30V) to achieve fast program and erase times. These high voltages and resulting electric fields cause cell oxides to breakdown, shortening cycling life and degrading cell thresholds. Additionally, this high voltage stresses periphery transistors. Intel's Flash Memories are more akin to EPROM; both use a significantly lower voltage around 12V.

NAND

Futuristic types of EEPROM (Figure 17) that have shifted from highly-manufacturable NOR-gate architectures, appear to provide ETOX-like density on a per cell basis and potential use in similar applications. A closer examination reveals internal positive and negative charge pumping. When applied across a memory cell, the dual charge pumps net a high voltage that cause oxide stress. These stacked-gate cells program and erase via tunneling. They program from the substrate to the floating gate, and erase in the reverse bias.

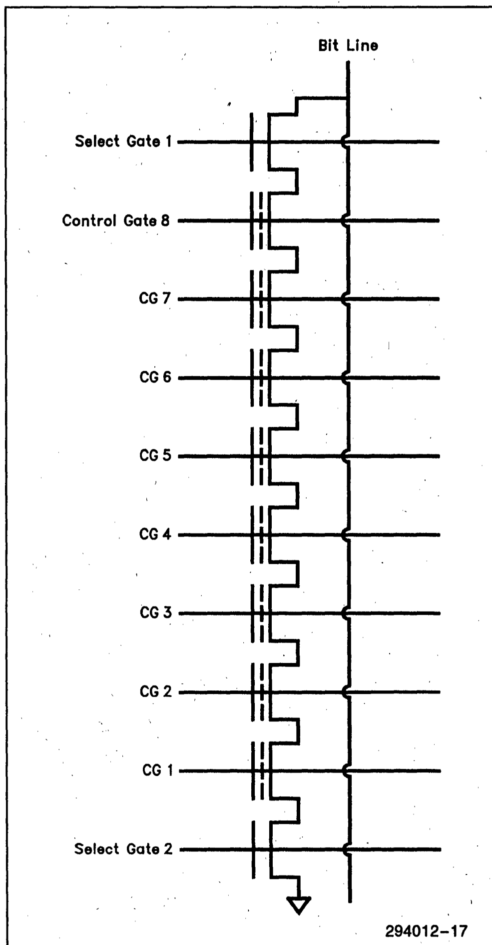


Figure 17. Eight Cell NAND Configuration

Layout of the NAND array groups eight cells as a set, each set requires two select transistors to control bitline access. Due to this NAND configuration, the read path goes through other cells making access slow. Increased read speed requires an internal SRAM page buffer, however the first read always remains slow. Fast write times require a page buffer for full wordline programming. Several wordlines erase at once, setting the block size. Product scaling becomes limited from high-voltage requirements on periphery transistors and isolation technology. Compared to EEPROM, the NAND approach improves array compaction at the expense of more complex decoding and periphery circuitry.

SUMMARY

Intel's technology advances result in products that are more efficient, more reliable, less expensive and higher performance. Submicron geometries and double-metal technology allow considerable memory array compaction, providing increased read and byte-write performance. Design compaction also improves with EPI wafers that reduce latch-up, allowing closer transistor layout. Strong cycling endurance results from the quality of the thin low-defect tunnel oxide, and the electrical characteristics of internal program and erase operations. Cycling, voltage and temperature exhibit only a small influence on byte-write and block-erase speeds. Products built on Intel's CMOS ETOX III Flash Memory technology require minimal power consumption during writes, erasure, read, and low-power sleep or standby modes.

Intel Flash Memory products designed on ETOX III will satisfy many different applications.The Thin Small-Outline Package (TSOP) provides customers with high memory density in the smallest footprint. Some applications for ETOX III flash products include memory cards, solid-state drives, non-volatile operating systems, high-performance system storage, data acquisition, and application and embedded code storage. The solid-state nature of flash results in improved ruggedness over mechanical rotating media. With blocking, applications can perform background erase to optimize system performance. Today, Intel's technological advances in flash memory are driving cost to parity with DRAM. This steep decline in the price learning curve enables new classes of systems and system architectures.

OTHER REFERENCES

Related documents of interest to readers of this engineering report:

28F008SA Data Sheet (Order No. 290429)

Series 2 Memory Card Data Sheet (Order No. 290434)

ER-27: The Intel 28F008SA Flash Memory (Order No. 294011)

APPENDIX A

First-generation flash devices, like the 28F020, use externally-controlled algorithms for byte write and bulk erase. These algorithms require that customer software control:

- Cell pulse widths, and pulse repetition where required.
- Erase preconditioning (i.e. pre-programming all cells before erase).
- Timeout delays to allow cell voltages to transition from program or erase levels to read verify levels.
- Read compare operations to determine success.

Second-generation architectures, like that on the 28F008SA, contain an internal Write State Machine (WSM) to simplify software development. This WSM provides internal control of all of these first-generation requirements, as well as reporting on activity progress/success through the internal Status Register. A dedicated output on the 28F008SA allows immediate hardware signaling of WSM activity status. The Command User Interface (CUI) provides customer control.

The other major architectural feature of second-generation devices is array segmentation, also referred to as "blocking". First-generation products erase in bulk. This means that all cells in the array erase simultaneously. Second-generation "sectored" architectures divide the array into separately-erasable block segments. This provides logical segmentation of customer code, and allows reads of other device blocks while one is erasing (i.e. via the erase-suspend/resume commands).

intel®

ENGINEERING REPORT

ER-29

November 1992

The Intel 2/4 Mbit Boot Block Flash Memory Family

ALAN BUCHECKER
PETER HAZEN
MEMORY COMPONENTS DIVISION

Order Number: 294013-001

The Intel 2/4 Mbit Boot Block Flash Memory Family

CONTENTS PAGE

INTRODUCTION 3-606
Word/Byte-Wide Versions 3-606
Byte-Wide Versions 3-606

TECHNOLOGY OVERVIEW 3-606

DEVICE ARCHITECTURE 3-607
Array Organization 3-607
Block Memory Maps 3-608
Write/Erase Automation 3-608
Command User Interface (CUI) 3-608
Write State Machine (WSM) 3-611
Status Register 3-612
Internal Oscillator 3-613
Supply Voltage Sensing 3-613
Deep Power-Down and Device Reset ... 3-613
Automatic Power Savings 3-614
Block Erase 3-614
Erase Suspend/Resume 3-616
Word/Byte Write 3-616

DEVICE CHARACTERIZATION 3-616
A.C. and D.C. Parameters 3-616
Energy/Power Consumption 3-617
Word/Byte-Write and Block-Erase Times 3-617

DEVICE RELIABILITY 3-617
Word/Byte-Write and Block-Erase Cycling 3-617
Data Protection 3-618

SUMMARY 3-618

OTHER REFERENCES 3-618

3

INTRODUCTION

The ETOX™ III (EPROM tunnel oxide) 2/4 Mbit family of Intel boot block Flash Memories are a continuation of boot-top and boot-bottom architectures first introduced in the 1 Mbit 28F001BX-T and 28F001BX-B devices. Top (-T) and bottom (-B) boot block offerings provide compatibility for microprocessors that boot from high or low memory addresses.

All versions of this new family provide a 16 Kbyte hardware-protected boot block, two 8 Kbyte parameter blocks and a 96 Kbyte main block. The 2 Mbit products have an additional 128 Kbyte main block, while 4 Mbit offerings contain three additional 128 Kbyte main blocks.

Flash memories combine inherent non-volatility with in-system alterability of device contents. Selective block erasure allows manipulation of data contents within one of the seven (or five) mini-array segments without affecting data stored in the other six (or four). Advances in process control have allowed development of a double-polysilicon single-transistor flash memory capable of 100,000 write/erase cycles per block. The 2/4 Mbit boot block family electrically erases all bits in a block via electron tunneling. The EPROM programming mechanism of hot-electron injection is employed for high-performance electrical word/byte write as required for computing and embedded applications.

An integrated Command User Interface (CUI) simplifies microprocessor control of device operations (word/byte write, block erase, erase suspend/resume, Status Register read/clear, array read and device identifier read). The internal Write State Machine (WSM) controls all functions and circuits associated with word/byte-write and block-erase operations, including pulse widths and repetition, timeout delays, erase preconditioning and margined verifications. The WSM continually updates the internal Status Register during these functions. The Status Register is read via outputs DQ_0-DQ_7 providing feedback of WSM activities.

The CUI and Status Register interface to power-up/down-protection circuitry, address/data latches and the WSM. This interface augments first-generation flash memory circuitry to optimize Intel's 2/4 Mbit boot block family for microprocessor-controlled word/byte write and block erase.

A deep-power-down mode enables extremely low power consumption to augment reduced-power standby operation. An automatic power savings feature reduces I_{CC} during read mode.

A 60 ns access time (t_{ACC}) results from a memory cell current of approximately 70 μA, low-resistance polysilicide wordlines strapped with metal, advanced scaled periphery transistors and improved circuit techniques.

The dense one-transistor cell structure, coupled with high array efficiency, yields a 4 Mbit die measuring 295 by 331 mils and a 2 Mbit die measuring 295 by 221 mils.

Word/Byte-Wide Versions

Intel's 28F400BX is a 4,194,304-bit non-volatile memory organized as either 262,144 words (256K x 16) or 524,288 bytes (512K x 8). A dedicated $\overline{\text{BYTE}}$ control input provides selection of the desired input/output (I/O) configuration (either x16 or x8) for read operations and data writes.

Intel's 28F200BX is a 2,097,152-bit non-volatile memory organized as either 131,072 words (128K x 16) or 262,144 bytes (256K x 8). This device also provides $\overline{\text{BYTE}}$ control of I/O configuration.

Initial offerings of the 28F400BX and 28F200BX are in the 44-lead Plastic Small-Outline Package (PSOP). These products will also be offered in 56-lead Thin Small-Outline Package (TSOP).

Byte-Wide Versions

Intel's 28F004BX is a 4,194,304-bit non-volatile memory arranged as 524,288 bytes (512K x 8). Intel's 28F002BX is a 2,097,152-bit non-volatile memory arranged as 262,144 bytes (256K x 8).

With lower pin counts compared to their x8/x16 equivalents, the 28F004BX and 28F002BX allow housing in the smaller 40-lead TSOP.

TECHNOLOGY OVERVIEW

Intel's ETOX III Flash Memory technology incorporates advances from ETOX I and ETOX II processes, and leverages over two decades of EPROM manufacturing experience. Using advanced 0.8 μm double-polysilicon N-well/P-well CMOS technology, the 2/4 Mbit boot block flash memories employ a 2.5 μm x 2.9 μm single-transistor cell affording array density equivalent to comparable EPROM technology, and twice that of Intel's ETOX II process. The ETOX III flash memory cell is identical to 0.8 μm EPROM, except for an additional source implant which optimizes erase performance. Figure 1 shows a cross-section of the flash memory cell.

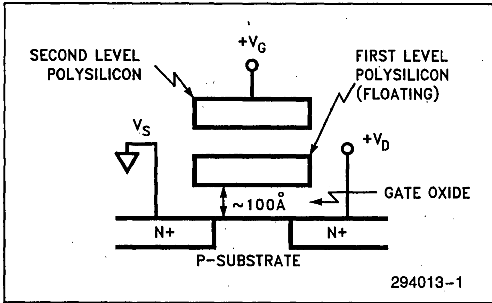


Figure 1. Flash Memory Cell

High-quality tunnel oxide under the single floating polysilicon gate promotes electrical erasure. All cells within the selected block are simultaneously erased via Fowler-Nordheim tunneling. Applying 12V to block source junctions and grounding the select gates erases all cells within that block. The internal WSM controls the automated block-erase algorithm, including pre-erase conditioning (i.e. pre-programming all block bits) and margin verification, in response to user requests relayed by the CUI. WSM-controlled block erasure, including pre-programming, typically ranges from 1.0 to 2.4 seconds depending on the size of the block selected.

Word/byte write is accomplished with the standard EPROM mechanism of channel hot-electron injection from the cell drain junction to the floating gate. Bringing both the select gate and the cell drain to high voltage initiates programming. The WSM regulates the internal word/byte-write algorithm, including margin verification, after the correct command sequence is written and decoded. Word/byte write typically requires 9 μs.

DEVICE ARCHITECTURE

Array Organization

Layout of the 2/4 Mbit boot block family is segmented as two array planes (see Figure 2). This organization allows improved access times via minimal internal busing, thus balancing the need for high speed with the requirement of small die size for cost-effective solutions. In the 4 Mbit family, each array plane is 1024 rows by 2048 columns. For the 2-Mbit family, each array plane is 1024 rows by 1024 columns. Access time is reduced by limiting column length to 1024 cells. The polysilicon row is strapped in metal every 512 columns to reduce wordline delay. Two row decoders run vertically along the array plane sides, and column decoders run horizontally between array planes. Figure 36 shows a die photo of the 4 Mbit family.

Each array plane is divided into eight I/Os for the 28F400BX/200BX. The upper plane consists of the high-byte I/Os (DQ_8–DQ_{15}), while the lower plane consists of the low-byte I/Os (DQ_0–DQ_7). During byte-wide operation ($\overline{BYTE}$ = "0") the high-byte I/Os are multiplexed through the low-byte I/Os via A_{-1} decoding. Data for I/O_0 is stored in the left-most 256 columns (or 128 columns for the 28F200BX) of the lower plane, with the next 256 columns (or 128) storing data for I/O_1, etc. Data for I/O_8 is stored in the left-most 256 (or 128) columns of the upper plane, with the next 256 (or 128) columns storing data for I/O_9, etc.

For the dedicated byte-wide products (28F004BX/002BX), data for a given I/O is divided between the upper and lower array planes as decoded by A_{10}. Data for I/O_0 is stored in the left-most 256 columns (or 128 columns for the 28F002BX) of both the upper and lower planes, with the next 256 columns (or 128) in each plane storing data for I/O_1, etc.

Since each I/O is a grouping of adjacent columns (256 or 128), the independently-erasable blocks (seven or five) are segmented within each I/O. Each I/O in the 4 Mbit family is divided into seven blocks; including a 16 Kbyte boot block, two 8 Kbyte parameter blocks, one 96 Kbyte main block and three 128 Kbyte main blocks. Each I/O in the 2 Mbit family is divided into five blocks; including a 16 Kbyte boot block, two 8 Kbyte parameter blocks, one 96 Kbyte main block and one 128 Kbyte main block. Each block source is electrically isolated from the sources of the other six (or four) blocks. This allows individual block erase without altering data in the other blocks.

Addresses A_9–A_0 select one of 1024 rows. Row address lines are decoded sequentially for selection. Table 3 lists row address bitmaps.

Columns are decoded by A_{18}–A_{10} for the 28F004BX, A_{17}–A_{10} for the 28F400BX and 28F002BX, and A_{16}–A_{10} for the 28F200BX. 4 Mbit family columns are numbered 0–255 for each I/O, and 2 Mbit columns are numbered 0–127 for each I/O. Table 4 lists column address bitmaps.

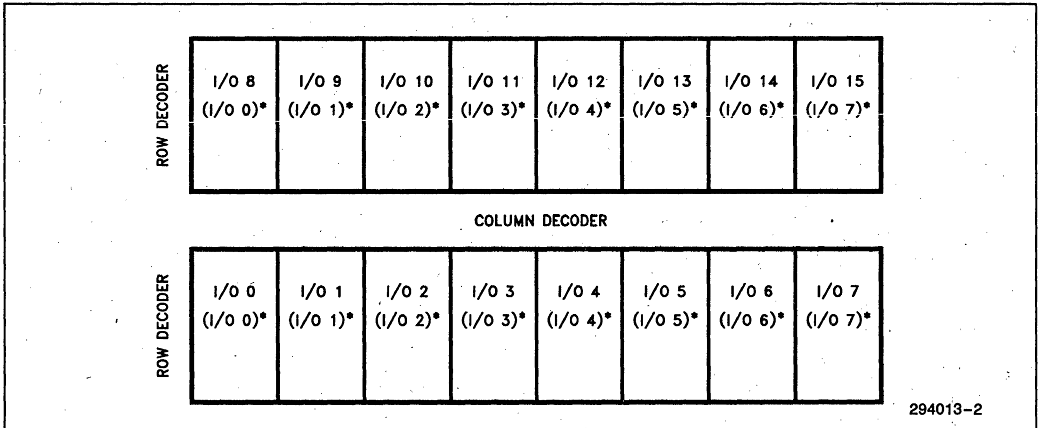


Figure 2. 2/4 Mbit Boot Block Family Array Organization ((I/O#)* Denotes 28F004BX/002BX I/Os)

Block Memory Map

Top (-T) and bottom (-B) boot block offerings have block address maps which are inverted from one another to provide compatibility for microprocessors that boot from high or low memory addresses. Figures 3, 4, 5 and 6 show 28F400BX-T/B, 28F200BX-T/B, 28F004BX-T/B and 28F002BX-T/B memory maps. The addresses shown in Figures 3 and 4 for the 28F400BX-T/B and 28F200BX-T/B, decode two bytes of data. A_{-1} decodes between the two bytes when $\overline{BYTE}$ is low.

Write/Erase Automation

Intel's 2/4 Mbit boot block Flash Memories contain an on-chip Command User Interface. Write State Machine, Status Register and address/data latches to dramatically simplify user interface. This combination of functional units reduces microprocessor control complexity of word/byte-write block-erase, erase-suspend/resume, Status Register-read/clear, ID-read and array-read operations. Figures 7 and 8 show 28F400BX/200BX and 28F004BX/002BX block diagrams.

Command User Interface (CUI)

The CUI consists of a command decoder and command register. User requests are decoded and latched in a microprocessor write cycle controlled by Chip Enable ($\overline{CE}$) and Write Enable ($\overline{WE}$). Status Register-read/clear, ID-read and array-read commands are directly handled by the CUI. The CUI also accepts word/byte-write, block-erase and erase-suspend/resume commands. $\overline{WE}$'s rising edge latches address, command and data-in registers, and requests WSM initiation of the selected operation. These on-chip address, command and data latches, controlled by the CUI, minimize system interface logic and free the system bus.

28F400BX-T

3FFFFH
16 Kbyte BOOT BLOCK
3E000H
3DFFFH
8 Kbyte PARAMETER BLOCK
3D000H
3CFFFH
8 Kbyte PARAMETER BLOCK
3C000H
3BFFFH
96 Kbyte MAIN BLOCK
30000H
2FFFFH
128 Kbyte MAIN BLOCK
20000H
1FFFFH
128 Kbyte MAIN BLOCK
10000H
0FFFFH
128 Kbyte MAIN BLOCK
00000H

28F400BX-B

3FFFFH
128 Kbyte MAIN BLOCK
30000H
2FFFFH
128 Kbyte MAIN BLOCK
20000H
1FFFFH
128 Kbyte MAIN BLOCK
10000H
0FFFFH
96 Kbyte MAIN BLOCK
04000H
03FFFH
8 Kbyte PARAMETER BLOCK
03000H
02FFFH
8 Kbyte PARAMETER BLOCK
02000H
01FFFH
16 Kbyte BOOT BLOCK
00000H

Figure 3. 28F400BX-T/B Memory Maps

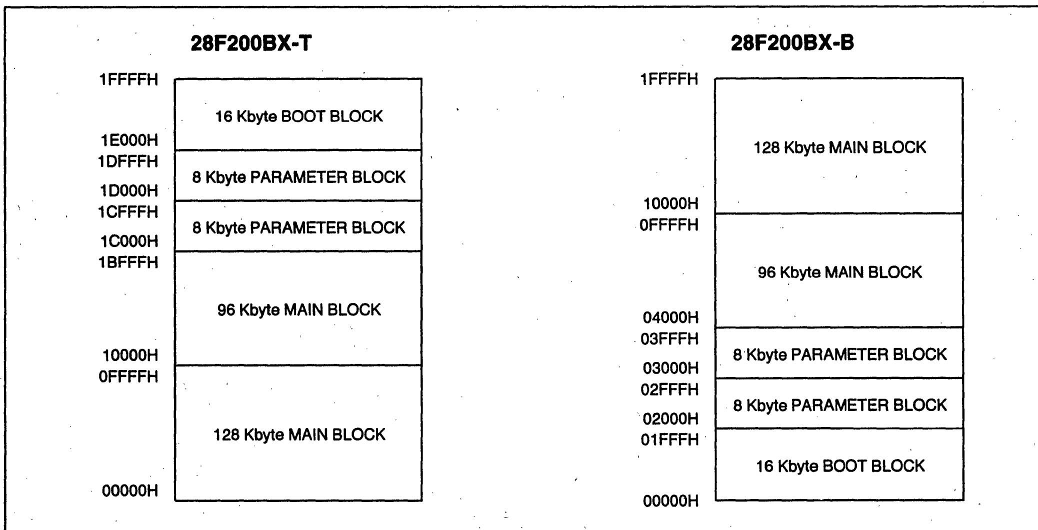


Figure 4. 28F200BX-T/B Memory Maps

28F004BX-T

7FFFFH
16 Kbyte BOOT BLOCK
7C000H
7BFFFH
8 Kbyte PARAMETER BLOCK
7A000H
79FFFH
8 Kbyte PARAMETER BLOCK
78000H
77FFFH
96 Kbyte MAIN BLOCK
60000H
5FFFFH
128 Kbyte MAIN BLOCK
40000H
3FFFFH
128 Kbyte MAIN BLOCK
20000H
1FFFFH
128 Kbyte MAIN BLOCK
00000H

28F004BX-B

7FFFFH
128 Kbyte MAIN BLOCK
60000H
5FFFFH
128 Kbyte MAIN BLOCK
40000H
3FFFFH
128 Kbyte MAIN BLOCK
20000H
1FFFFH
96 Kbyte MAIN BLOCK
08000H
07FFFH
8 Kbyte PARAMETER BLOCK
06000H
05FFFH
8 Kbyte PARAMETER BLOCK
04000H
03FFFH
16 Kbyte BOOT BLOCK
00000H

Figure 5. 28F004BX-T/B Memory Maps

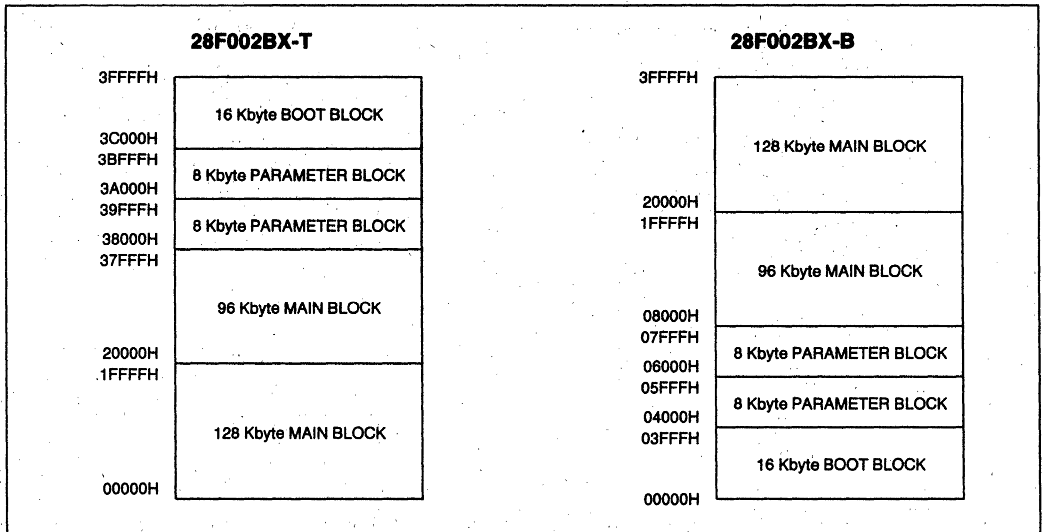


Figure 6. 28F002BX-T/B Memory Maps

Write State Machine (WSM)

The WSM processes word/byte-write, block-erase and erase-suspend/resume requests received from the CUI. The WSM rejects word/byte-write and block-erase requests if the WSM is currently busy, if V_{PP} is not at high voltage (12V) or if the Low V_{PP} Status Register flag is set (i.e. not cleared from a previous low-voltage condition).

The WSM consists of an integrated oscillator and control circuitry. It generates signals which control the word/byte-write, block-erase, erase-suspend/resume and verify circuits. It also receives feedback from these circuits, allowing Status Register update. The WSM and associated circuits perform the equivalent of first-generation flash memory program and bulk-erase algorithms automatically.

This eliminates the need for system timers, and frees the microprocessor to service interrupts or perfonn other functions during device word/byte-write or block-erase operations.

The WSM provides feedback to the CUI to determine when a given command is valid. Although nearly all commands are available when the WSM is inactive, only status read is valid while the WSM performs a word/byte-write operation. During block erase, only the read-status and erase-suspend commands are available. Read-array, read-status, and erase-resume commands are valid with the WSM in an erase-suspended state. Invalid operations are interpreted as the read-array command when the WSM is inactive or erase suspended, and as the read-status command when the WSM is active in word/byte write or block erase.

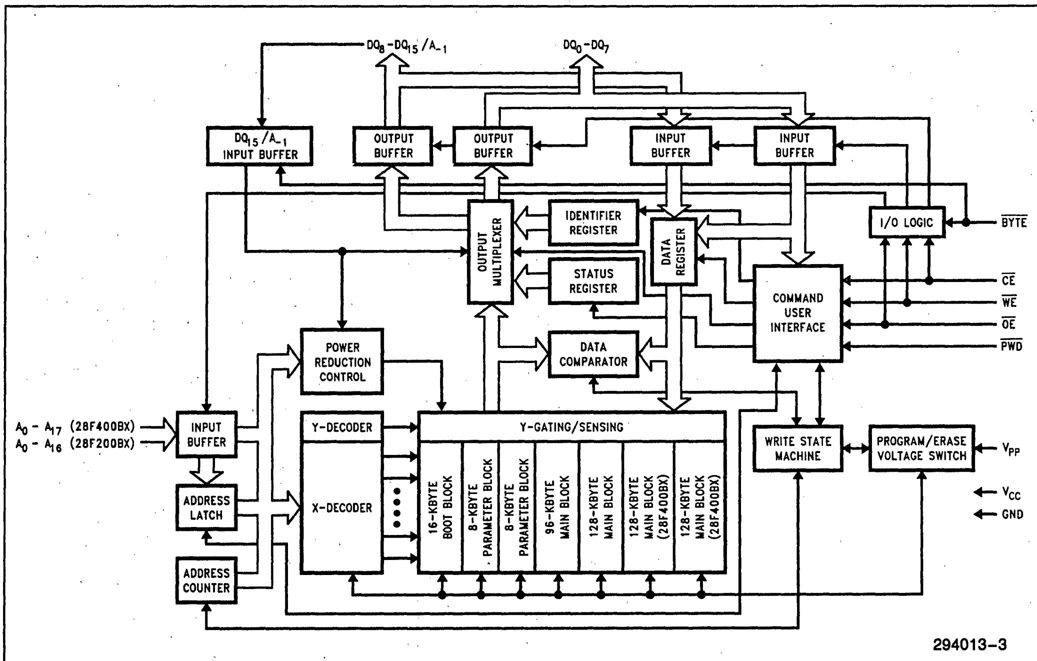


Figure 7. 28F400BX/200BX Block Diagrams

Status Register

The internal Status Register contains a full complement of activity status bits. These bits and their meaning "1/0" are:

SR.7: WSM status (READY/$\overline{\text{BUSY}}$)

SR.6: Erase-suspend status (ERASE SUSPENDED/ $\overline{\text{ERASE IN PROGRESS or COMPLETED}}$)

SR.5: Block-erase status (ERROR/$\overline{\text{SUCCESS}}$)

SR.4: Word/byte-write status (ERROR/$\overline{\text{SUCCESS}}$)

SR.3: V_{PP} status (LOW/$\overline{\text{OK}}$)

All bits are set by the WSM, and read via the CUI. The WSM can only set SR.3, SR.4 and SR.5; it cannot clear them. They remain set until the CUI processes a clear Status Register command. There are two reasons for operating in this fashion. First is synchronization; the WSM does not know when the host CPU has read the Status Register, therefore does not know when to clear it.

Secondly, allowing system software to control reset adds flexibility to the way this device may be used. The CPU may write several words/bytes, or erase several blocks back-to-back, while polling SR.7 to determine when the next word/byte-write or block-erase command can be given. When all words/bytes are written, or all blocks erased, the system polls the other status flags to determine if all operations were successful, or if an error occurred. While other approaches require the controlling microprocessor to watch for non-completion of write or erase within a specified time to indicate an error, this implementation requires no external system timers or software timing loops. As such, the system can reduce its polling overhead while still identifying any potential error conditions.

Status Register contents are driven to device outputs on the falling edge of $\overline{\text{CE}}$ or Output Enable ($\overline{\text{OE}}$), whichever occurs last in the read cycle. $\overline{\text{CE}}$ or $\overline{\text{OE}}$ must be toggled to update Status Register contents.

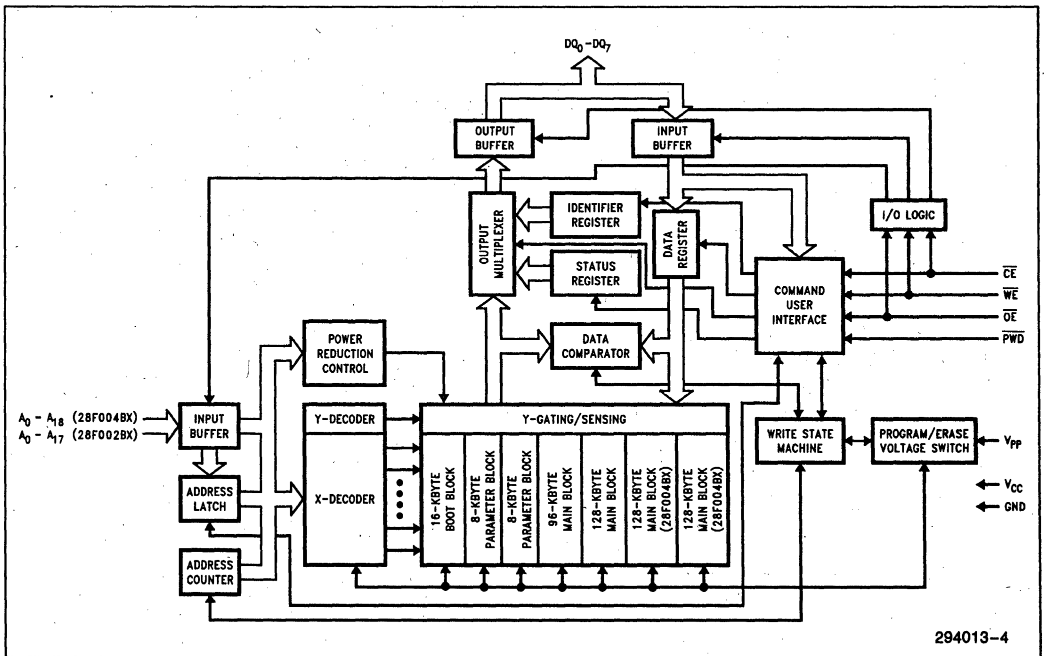


Figure 8. 28F004BX/002BX Block Diagrams

Internal Oscillator

The WSM is designed using clocked logic circuits. An on-chip ring oscillator generates the clock signals. The frequency of a standard ring oscillator varies with processing, temperature and supply voltage. A proven design used on the 28F001BX-T/B, 28F008SA and 2/4 Mbit boot block family minimizes these variations.

The switching current of each stage in the ring oscillator is controlled by a current reference which varies linearly with V_{CC}. The trip point of each ring oscillator inverter also varies linearly with V_{CC}. These two effects offset each other, and the resulting oscillator period is proportional to RC with only a small dependence on V_{CC}.

An on-chip resistor sets the value of R. The gate capacitance of the inverters in the ring oscillator sets the value of C. Process variations in these values are reduced by trimming the period of each oscillator during manufacturing. The resistor is the only source of temperature variation.

Supply Voltage Sensing

Figure 9 shows the $LOWV_{CC}$ and $LOWV_{PP}$ generation circuits. Power supply voltages (V_{CC} and V_{PP}) are divided down and compared to a reference voltage. If V_{REF} is greater than the divided power supply voltage, the $LOWV_{CC}$ or $LOWV_{PP}$ signal is driven high. The generated V_{REF} level is supply-voltage independent to the first order.

Positive power to the circuit is supplied by M1 and M2. M1 and M2 sources are pulled up to the higher of ($V_{PP} - V_{tn}$) or ($V_{CC} - V_{tw}$). V_{tn} is the threshold of an implanted N-channel device, about 0.9V. V_{tw} is the threshold of a native N-channel device, about 0V. This scheme ensures that the circuit works regardless of the applied supply voltages.

The $LOWV_{CC}$ signal is used by the word/byte-write and block-erase circuits, as well as the CUI and WSM.

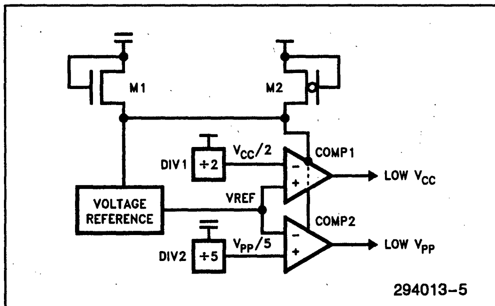


Figure 9. Low-Power Detector Circuit

If $LOWV_{CC}$ is active, the CUI will not accept user writes and resets to an array-read condition. The WSM is similarly reset by $LOWV_{CC}$. The $LOWV_{PP}$ signal is used by the WSM; if V_{PP} drops below the high-voltage detector trip point during word/byte write or block erase, the Status Register's low V_{PP} bit is set and WSM operation halts. The system must clear the Status Register before any subsequent word/byte-write or block-erase operations can succeed.

Deep Power-Down and Device Reset

The 2/4 Mbit boot block family incorporates a deep-power-down mode that reduces I_{CC} and I_{PP} to typically 0.20 μA and 0.07 μA respectively. $\overline{PWD}$ low selects deep power-down. When $\overline{PWD}$ is high, the device can be placed in an active or standby mode depending on $\overline{CE}$'s state.

Deep power-down is similar to standby except that all circuits excluding the $\overline{PWD}$ buffer are turned off. This mode greatly reduces power consumption, but requires more time to transition the device into an active mode. A read wake-up time (t_{PHQV}) is required from $\overline{PWD}$ switching high until output and sense circuitry become fully functional and data can be read from the part. Similarly, a write wake-up time (t_{PHWL}) is needed before the CUI recognizes writes. After this interval normal operation is restored; the CUI is reset to read-array mode and the Status Register is cleared to 80H.

Figure 10 shows a diagram of the power-down circuit. The TTL buffer formed by M1–M3 disables the low-power detect circuits, the redundancy-address flash bits and the $\overline{CE}$ TTL buffer formed by M4–M6. These circuits were always enabled in first-generation Intel flash architectures. Turn-on delays of these circuits determine $\overline{PWD}$ access time and write specifications.

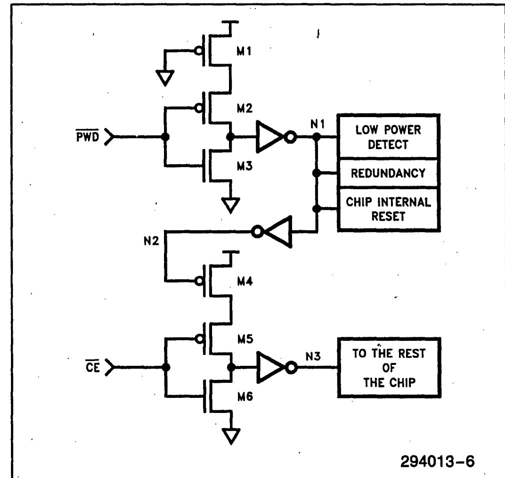


Figure 10. Power-Down and Reset Functions

3

$\overline{PWD}$ functions properly with TTL-level inputs. However, to attain lowest possible power consumption, full CMOS levels must be used. If the voltage on the gate of M3 rises above its 0.9V threshold, M3 will turn on and draw current. Input voltages in the 0.7V–0.9V range could cause enough current conduction in M3 to exceed the I_{CC} deep-power-down current (I_{CCD}) specification. This is why $\overline{PWD}$'s input voltage is specified as GND ±0.2V.

$\overline{PWD}$ also functions as a hardware reset to the WSM and CUI. If $\overline{PWD}$ is driven low ("0") during word/byte write, block erase, or erase suspend, that operation is aborted leaving the addressed memory locations in an unknown state. The Status Register is cleared, and CUI is set to array read. The aborted operation (word/byte write or block erase) must be repeated with $\overline{PWD}$ inactive to obtain a valid condition in the memory array.

$\overline{PWD}$ reset should be restricted to system reset only (as in the case of power supply failure), and should not be used as a standard means to terminate word/byte-write or block-erase operations. NOTE: Use the erase-suspend command to read another block (see the "Erase Suspend/Resume" section).

This use of $\overline{PWD}$ during system reset is important with automated write/erase devices. When the system comes out of reset it expects to read from the flash memory. Automated flash memories provide status information when accessed during write/erase modes. If a CPU reset occurs with no flash memory reset, proper CPU initialization would not occur because the flash memory would be providing the status information instead of array data. Intel's Flash Memories allow proper CPU initialization following a system reset through the use of the $\overline{PWD}$ input.

Automatic Power Savings

The 2/4 Mbit boot block flash memories include an automatic power-savings feature that reduces I_{CC} during read mode. Within one access time (t_{ACC} or t_{CE}) after an address or $\overline{CE}$ switches, I_{CC} automatically powers down from the 60 mA I_{CC} CMOS specification (or 65 mA TTL specification) to less than 1 mA. I_{CC} remains below 1 mA until either $\overline{CE}$ or an address is switched, or until the device is taken out of read mode. This feature provides significant power savings for applications that access the device slower than their specified read access times.

Block Erase

Block erasure is achieved by a two-step write sequence. The erase-setup code is written to the CUI in the first cycle. Erase confirm is written in the second cycle. The address supplied with the erase-confirm command is latched and decoded internally by the device; erase is subsequently enabled in that block. The second $\overline{WE}$ rising edge initiates the operation ($\overline{WE}$-controlled write).

The WSM triggers the high-voltage flash-erase switch connecting the 12V supply to the source of all bits in the specified block, while all wordlines are grounded. Figure 12 shows organization of the block source switches. Fowler-Nordheim tunneling results in simultaneous erasure of all bits in the selected block.

The block source switch (shown in Figure 11) controls the source voltage of all bits in a particular block. During block erase, M2 is off and M1 pulls the source to V_{PP}. When not in erase, M1 is off and M2 pulls the source to ground. The high-voltage latch formed by M4–M7 converts the low-voltage ERASE signal to a high-voltage signal that controls M1.

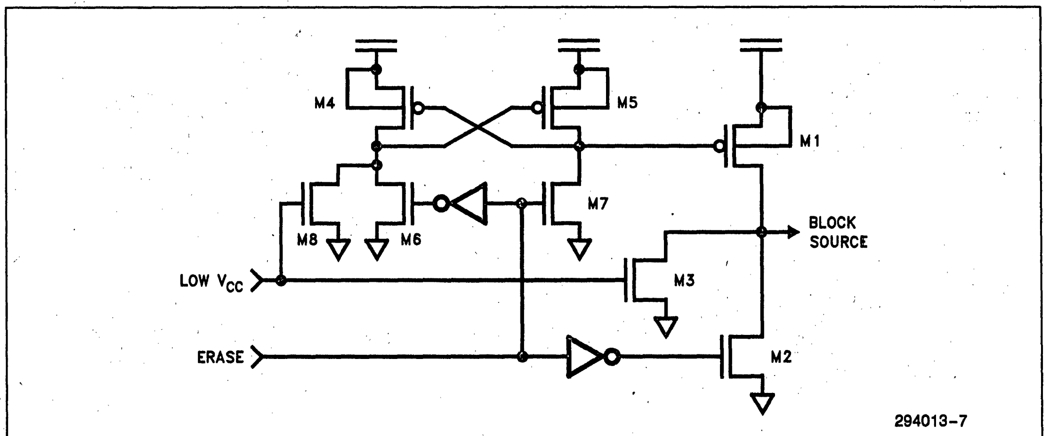


Figure 11. Block Source Switch

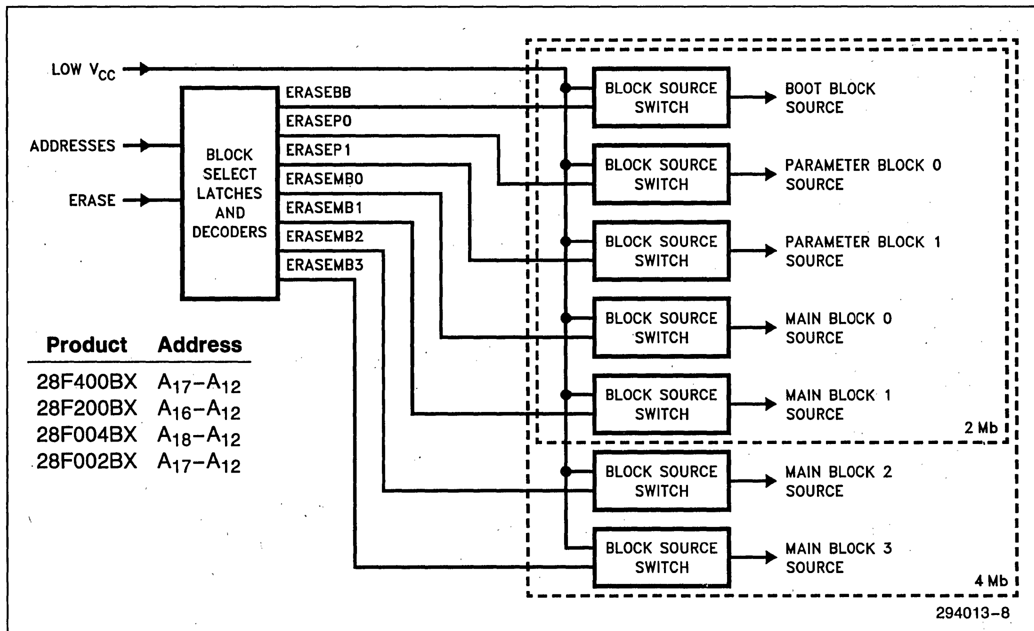


Figure 12. Array Erase Blocking

The tunneling that occurs during block erase requires only a small amount of current. However, the initial current required to charge the block's cumulative source capacitance to the erase voltage is large. Supply decoupling design practices minimize the system impact of the source charging.

The LOWV$_{CC}$ signal protects the array from erasure when V_{PP} is at a high voltage, but V_{CC} is below the write/erase lockout voltage (V_{LKO}). When this occurs, M3 pulls the block source to ground. The high-voltage latch is forced by M8 into the state that turns Ml off.

V_{PP} is continually monitored during all phases of the block-erase operation. If V_{PP} falls below the trip point of its high-voltage detect circuitry, erasure will not occur (or halts) and Status Register V_{PP} status (SR.3), block-erase status (SR.5) and WSM status (SR.7) bits are set to "1".

If SR.3 is set (Low V_{PP}), WSM operation is inhibited. The WSM will not execute further word/byte-write or block-erasure sequences until the Status Register has been reset by system software. Word/byte-write or block-erase requests with error flags SR.4 or SR.5 set are not inhibited, but the system loses the ability to determine success. The clear Status-Register command resets these bits.

After receiving the block-erase command sequence, the WSM automatically controls block pre-condition (programming all words to 0000H within the chosen block), erase pulses and pulse repetition, timeout delays, and word-by-word verification of all block addresses (sequentially checked via the address counter) using an alternative sensing reference to verify margin. The internal erase and verify operations continue until the entire block is erased. A read cycle applied to the part following the block-erase command sequence outputs Status Register contents; system software can poll the Status Register to determine when block erase has successfully completed. Following block erasure, the device remains in Status Register read block erasure, the device remains in Status Register read mode; a read-array command must be written to the device to access array data.

If the erase-setup command is followed with a command other than erase confirm, the device will not erase. The WSM sets both word/byte-write status and block-erase status bits in the Status Register to indicate an invalid sequence.

Erase Suspend/Resume

Erase suspend allows the system to interrupt block erase to read data from another array block. The ability to suspend erase and read data from another block offers the flexibility required for embedded applications. Upon receiving the erase-suspend command. the CUI requests that the WSM pause at one of several predetermined points in the algorithm. Upon reaching a suspend point, the WSM sets SR.6 (erase-suspend status) and SR.7 to "1". The system must poll the Status Register to determine if the suspend has been processed or the block erase has actually completed. Block-erase completion is indicated by SR.6 cleared to "0" and SR.7 set to "1". Read bus cycles default to Status Register read after issuing the erase-suspend command.

Once suspended, the WSM asserts a signal to the CUI which allows response to the read-array, read-status, and erase-resume commands. The system can write the read-array command allowing read access to blocks other than that which is suspended. The WSM continues to run, idling in a suspended state, regardless of all control inputs except $\overline{PWD}$. $\overline{PWD}$ driven low immediately shuts down the WSM, aborting the suspended erase operation.

The erase-resume command must be issued upon completion of reads from other array blocks to continue block-erase operation. The WSM then clears SR.6 and SR.7, and resumes erase operation from the suspension point. Read cycles following the erase-resume command output Status Register data.

See the appropriate data sheet (as listed under the "Other References" heading of this document) for a description of eratta relating to Erase Suspend/Resume.

Word/Byte Write

Word/byte write follows a flow similar to block erase. The word/byte-write-setup command is first written to the CUI. A second write cycle loads address and data latches. The rising edge of the second $\overline{WE}$ pulse requests that the WSM initiate activity, applying high voltage to the gates and drains of all bits to be written. Unlike block erase, word/byte write will proceed regardless of what data is applied on the second CUI write cycle; however writing data FFFFH (or FFH) does not modify memory contents.

Like block erase, the WSM controls program pulses and pulse repetition, timeout delays and word/byte verification. Word/byte write and verify (with alternate sensing reference and internally-generated verify voltage) continue until the word/byte is written. Internal word/byte-write verify checks that all bits written to zero have been correctly modified; it does not check bits specified as one. Word/byte write cannot change existing zeros to ones; this can only be accomplished by erase.

Read bus cycles following word/byte-write operations output Status Register data. System software, polling the Status Register, is informed of status through bits SR.3, SR.4, and SR.7. The read-array command must be written to the CUI following word/byte write to access array data.

In a scenario similar to that described under block erasure, word/byte write does not occur (or halts) if V_{PP} is detected low. In such a case SR.3, SR.4, and SR.7 are set high, and no further writes can take place until the Status Register is cleared by the clear Status Register command.

DEVICE CHARACTERIZATION

AC and DC Parameters

Figures 14 through 35 show graphs of several device parameters as a function of temperature and supply voltage.

In particular, note Figure 14 which shows typical read performance t_{AVQV} (t_{ACC}) of the 28F400BX, which is representative of the 2/4 MBit boot block flash family, as a function of V_{CC} and ambient temperature. t_{ELQV} (t_{CE}) in Figure 15 and t_{GLQV} (t_{OE}) in Figure 16 are also of particular interest. Access times t_{AVQV}, t_{ELQV}, and t_{GLQV} are specified and tested with an output load of 100 pF; decreased output load capacitance improves device operation.

Table 1 shows typical supply currents for several operating modes.

Table 1. RMS Current Values

Mode	I_{CC} (V_{CC} = 5.0V, CMOS Inputs)		I_{PP} (V_{PP} = 12V)
	x8	x16	
Read (6 MHz)	23 mA	25 mA	100 μA
Write	20 mA	22 mA	10 mA
Block Erase	15 mA	15 mA	12 mA
Standby	50 μA	50 μA	100 μA
Deep Power-Down	0.20 μA	0.20 μA	0.07 μA

Energy/Power Consumption

The system designer may be concerned with power consumption during block erase and word/byte write. Figure 32 shows I_{CC} and I_{PP} during block erase. Figure 33 shows I_{CC} and I_{PP} during word/byte write.

Word/Byte-Write and Block-Erase Times

The 2/4 Mbit boot block family and 28F008SA advance word/byte-write and block-erase performance compared to previous flash memories. The on-chip algorithm is improved over the 28F001BX to take advantage of process enhancements. This improvement is most apparent when compared to first-generation flash parts with externally controlled algorithms. First-generation device times shown in Table 2 assume optimal system overhead, and as such are absolute best case.

DEVICE RELIABILITY

Word/Byte-Write and Block-Erase Cycling

One of the most important reliability aspects of the 2/4 Mbit boot block family is its capability of 100,000 write/erase cycles per block. Destructive oxide breakdown has been a limiting factor in extended cycling of thin-oxide EEPROMs. Intel's ETOX Flash Memory technology extends cycling performance through:

- Improved tunnel-oxide processing that increases charge-carrying capability tenfold.
- Significantly reduced oxide area under stress that minimizes probability of oxide defects in the region.
- Reduced oxide stress due to a lower peak electric field (lower erase voltage than EEPROM).

Table 2. Word/Byte-Write and Block-Erase Performance vs Previous Devices

Device	Word/Byte-Write Time	Block-Erase Time/# Bytes	Erase Time per Kbyte
SECOND-GENERATION FLASH MEMORY DEVICES(1)			
28F400BX	9 μs	2.4s/128K	19 ms
28F004BX	9 μs	2.2s/96K	23 ms
28F200BX	9 μs	1.0s/16K	63 ms
28F002BX	9 μs	1.0s/8K	125 ms
28F008SA	9 μs	1.5s/64K	23 ms
28F001BX	18 μs	3.8s/112K 2.1s/8K 2.1s/4K	34 ms 256 ms 513 ms
FIRST-GENERATION FLASH MEMORY DEVICES(2)			
28F020	16.5 μs	6.8s/256K	27 ms
28F010	16.5 μs	3.9s/128K	30 ms
28F512	16.5 μs	2.4s/64K	37 ms
28F256A	16.5 μs	1.6s/32K	51 ms

NOTES:
1. Typical measured time.
2. Times calculated based on typical erase and precondition pulse requirements, with minimum write timings. Calculations are described in Figure 13.

Reliable word/byte-write and block-erase cycling requires proper selection of the maximum erase threshold voltage (V_t), and maintenance of a tight distribution. Maximum erase V_t is set to 3.4V via the internal block-erase algorithm and verify circuits. Tight erase V_t distribution gives an order of magnitude of erase-time margin to the fastest erasing cell, with virtually identical erase V_t distributions at 1 and 10,000 cycles (Figure 34). Program V_t distribution is similarly consistent over cycling (Figure 35).

Data Protection

The 2/4 Mbit boot block family offers protection against accidental block erasure or word/byte write during power transitions. Internal circuitry creates a device insensitive to V_{PP}/V_{CC} supply power-up/down sequencing.

$V_{PP} \leq V_{PPLMAX}$ locks out word/byte-write and block-erase circuits. $V_{CC} \leq V_{LKO}$ disables CUI command writes, resets the CUI to array-read mode, and holds the WSM inactive. The system designer must still guard against spurious command writes for $V_{CC} > V_{LKO}$ when $V_{PP} < V_{PPLMAX}$.

Several strategies are available to prevent data modification in the 2/4 Mbit family. The CUI provides a degree of software write protection since memory alteration occurs only after successful completion of a two-step write sequence. $\overline{WE}$ and $\overline{CE}$ must both go active to perform this sequence; driving either high inhibits command/data writes. Secondly, the system can place the device in deep-power-down mode ($\overline{PWD} = V_{IL}$) to disable command writes, reset the CUI to array-read mode, and hold the WSM inactive, effectively protecting array data and providing a way to reset the flash memory during system reset conditions. Finally, the system designer may switch V_{PP} to V_{PPH} when memory updates are required.

SUMMARY

Intel's 2/4 Mbit boot block Flash Memory family contains features that optimize this product group for computing and embedded applications. These features include hardware-implemented write/erase protection of the boot block, specialized array blocking, dedicated x8 and x8/16 user-configurable versions, automation of word/byte write and block erase, erase suspend for data read, deep-power-down/automatic-power-savings modes and a write/erase Status Register. With simple microprocessor interfacing and software command sequences, this family is the non-volatile computing and embedded solution of choice for today's designs.

OTHER REFERENCES

Related documents of interest to readers of this engineering report:

28F400BX-T/B, 28F004BX-T/B; "4 Mbit Flash Memory Family" Data Sheet (Order #290450)

28F400BX-TL/BL, 28F004BX-TL/BL; "4 Mbit Flash Memory Family" Data Sheet (Order #290451)

28F200BX-T/B, 28F002BX-T/B; "2 Mbit Flash Memory Family" Data Sheet (Order #290448)

28F200BX-TL/BL, 28F002BX-TL/BL; "2 Mbit Flash Memory Family" Data Sheet (Order #290449)

ER-28 "ETOX™ III Flash Memory Technology" Engineering Report (Order #294012)

AP-341 "Designing an Updatable BIOS Using Flash Memory" (Order #292077)

AP-363 "Extended Flash BIOS for Portable Computers" (Order #292098)

SUPPLEMENTARY INFORMATION

FORMULA:

b = # bytes in a block (256K, 128K, 64K, 32k)

n = # of erase pulses required (90 pulses)

w = Time for a write cycle (150 ns, t_{AVAV})

v = Time to verify (6055 ns, $t_{WHGL} + t_{GLQV}$)

p = Program pulse width (10 μs, t_{WHWH1}) one pulse programming assumed

e = Erase pulse width (10 ms, t_{WHWH2})

Precondition and precondition verify time is:

$$b(2w + p + v)$$

Erase/verify, each loop where some byte does not pass verify:

$$(n - 1)(2w + e + v)$$

Last erase pulse:

$$(1)(2w + e)$$

Passing erase-verify, all bytes:

$$b(w + v)$$

Total time can be summarized as:

$$b(3w + p + 2v) + n(2w + e + v) - (v)$$

or substituting in times for write, verify, program and erase pulse widths:

$$b(22.56 \times 10^{-6}) + n(10.006355 \times 10^{-3}) - (6.055 \times 10^{-6}) \text{ Seconds}$$

Figure 13. Erase Time Calculations for First-Generation Flash Memories

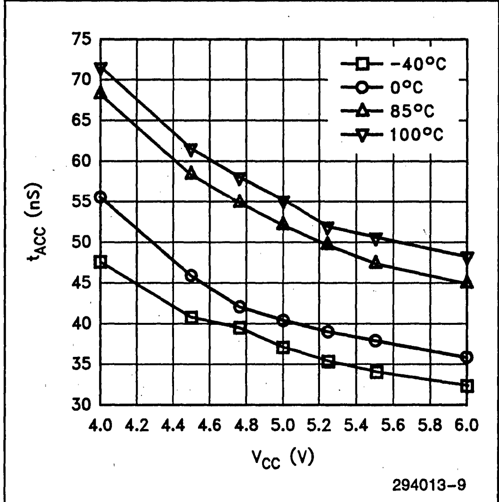


Figure 14. t_{AVQV} (t_{ACC}) vs V_{CC} and Temperature

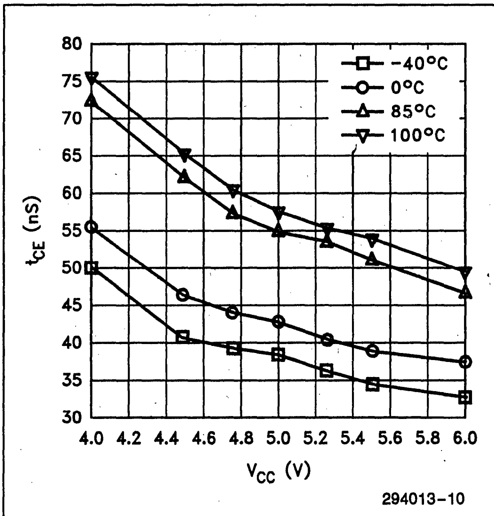


Figure 15. t_{ELQV} (t_{CE}) vs V_{CC} and Temperature

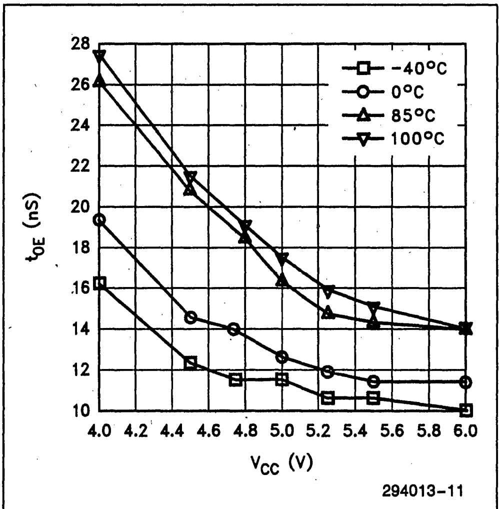


Figure 16. t_{GLQV} (t_{OE}) vs V_{CC} and Temperature

3

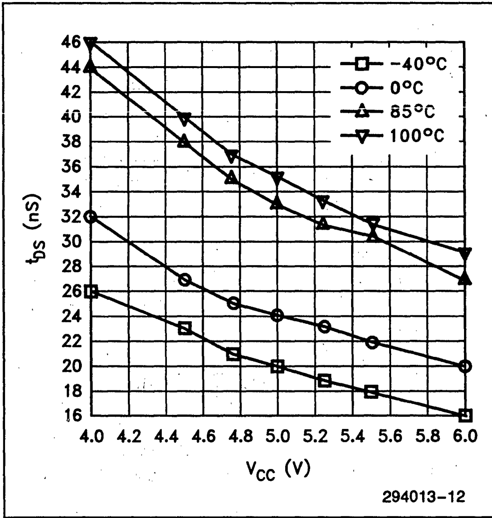


Figure 17. t_{DVWH} (t_{DS} $\overline{WE}$) vs V_{CC} and Temperature

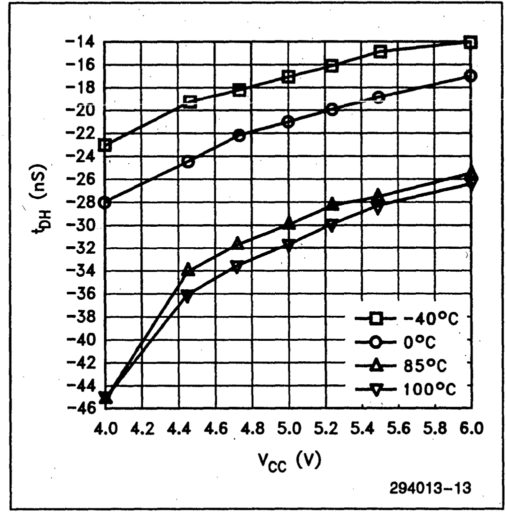


Figure 18. t_{WHDX} (t_{DH} $\overline{WE}$) vs V_{CC} and Temperature

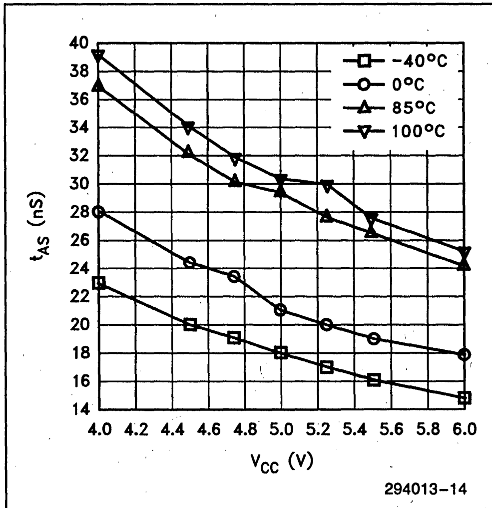


Figure 19. t_{AVWH} (t_{AS} $\overline{WE}$) vs V_{CC} and Temperature

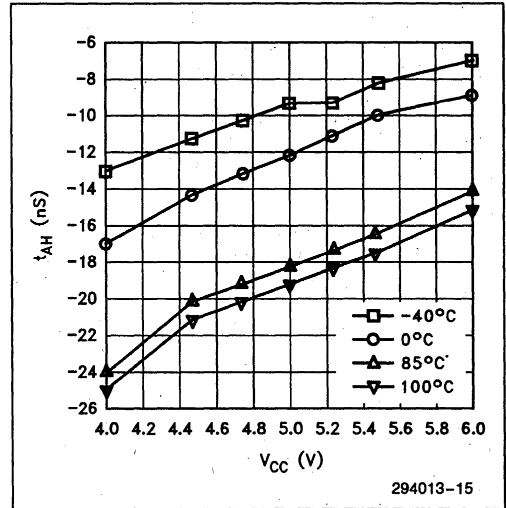


Figure 20. t_{WHAX} (t_{AH} $\overline{WE}$) vs V_{CC} and Temperature

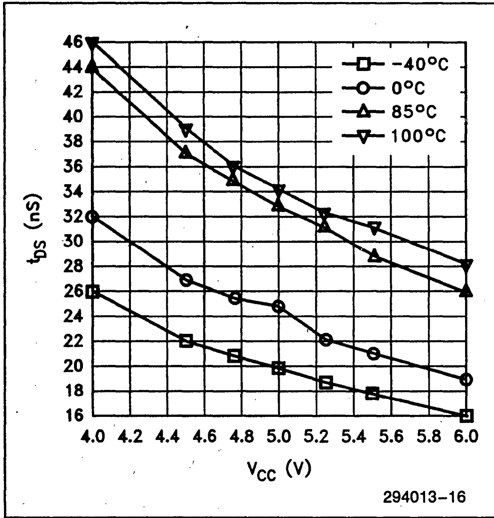


Figure 21. t_{DVEH} (t_{DS} $\overline{CE}$) vs V_{CC} and Temperature

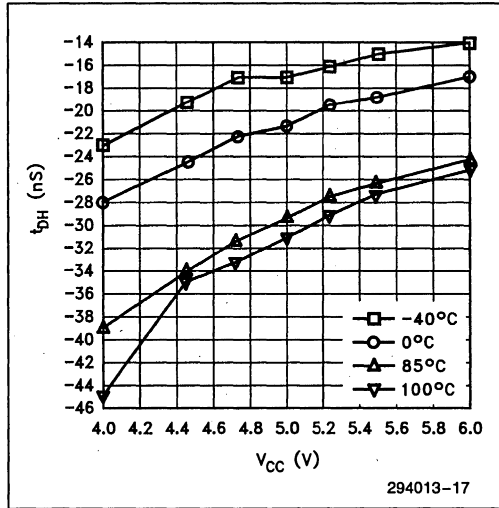


Figure 22. t_{EHDX} (t_{DH} $\overline{CE}$) vs V_{CC} and Temperature

3

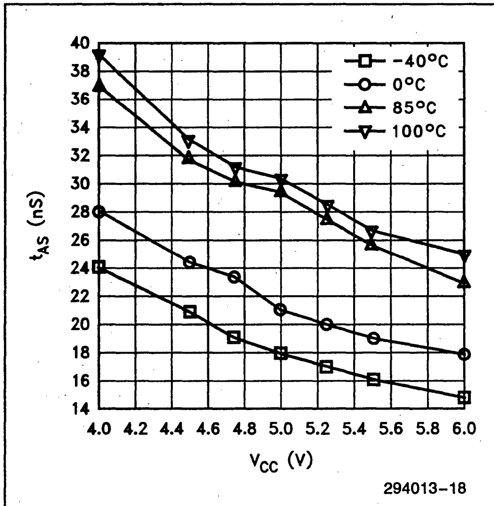


Figure 23. t_{AVEH} (t_{AS} $\overline{CE}$) vs V_{CC} and Temperature

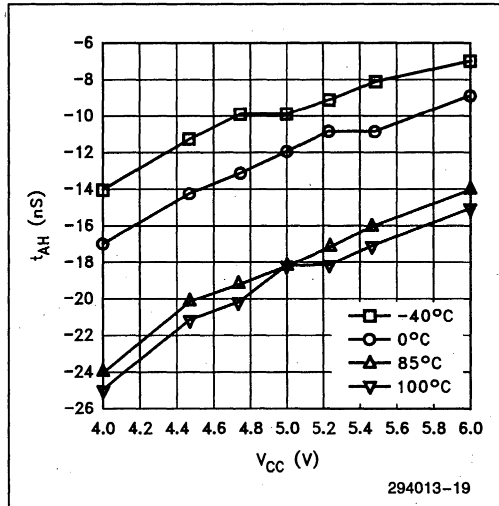


Figure 24. t_{EHAX} (t_{AH} $\overline{CE}$) vs V_{CC} and Temperature

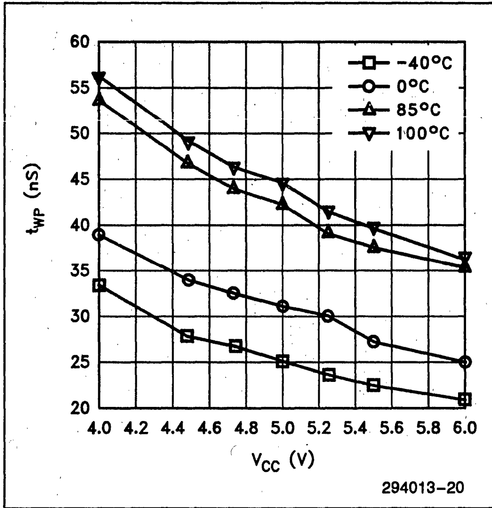


Figure 25. t_{WLWH} (t_{WP}) vs V_{CC} and Temperature

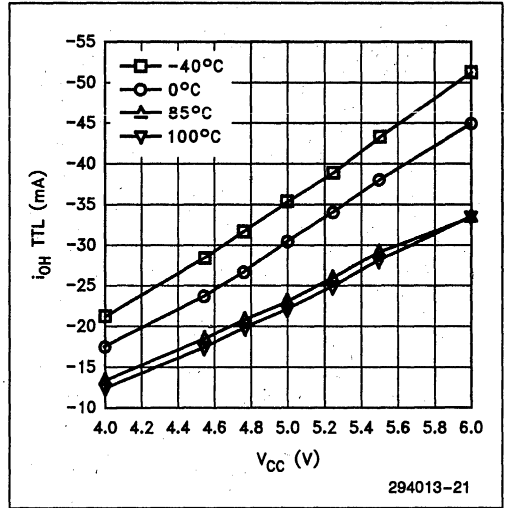


Figure 26. I_{OH} TTL vs V_{CC} and Temperature

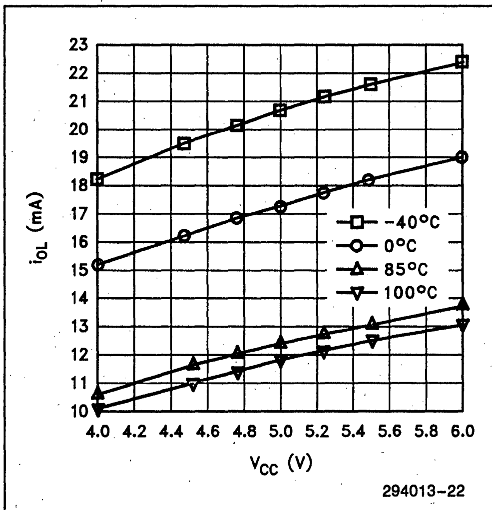


Figure 27. I_{OL} vs V_{CC} and Temperature (V_{OL} = 0.45V)

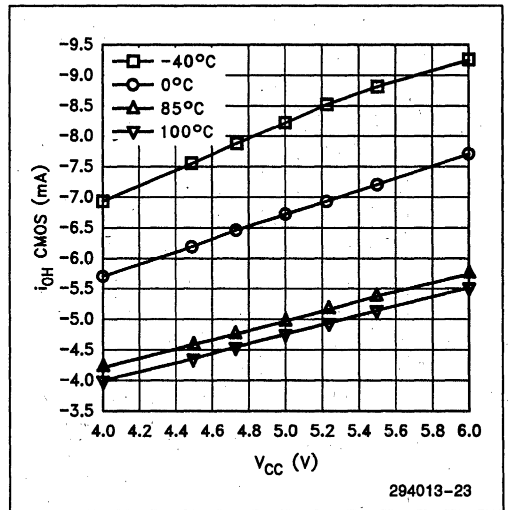


Figure 28. I_{OH} CMOS vs V_{CC} and Temperature

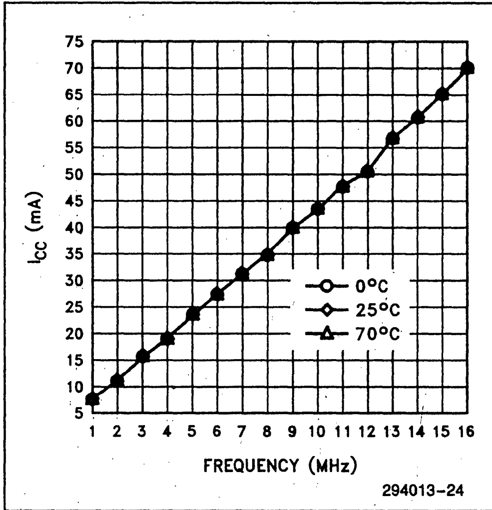


Figure 29. I_{CC} (RMS) vs Frequency x16 Operation

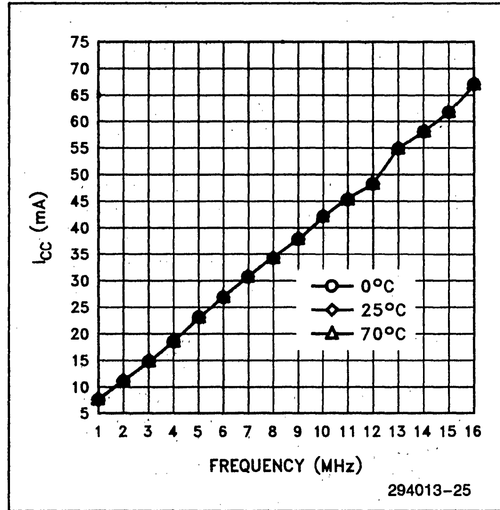


Figure 30. I_{CC} (RMS) vs Frequency x8 Operation

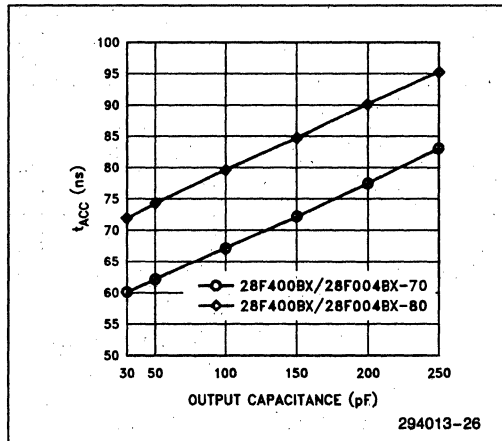


Figure 31. Access Time vs Output Loading

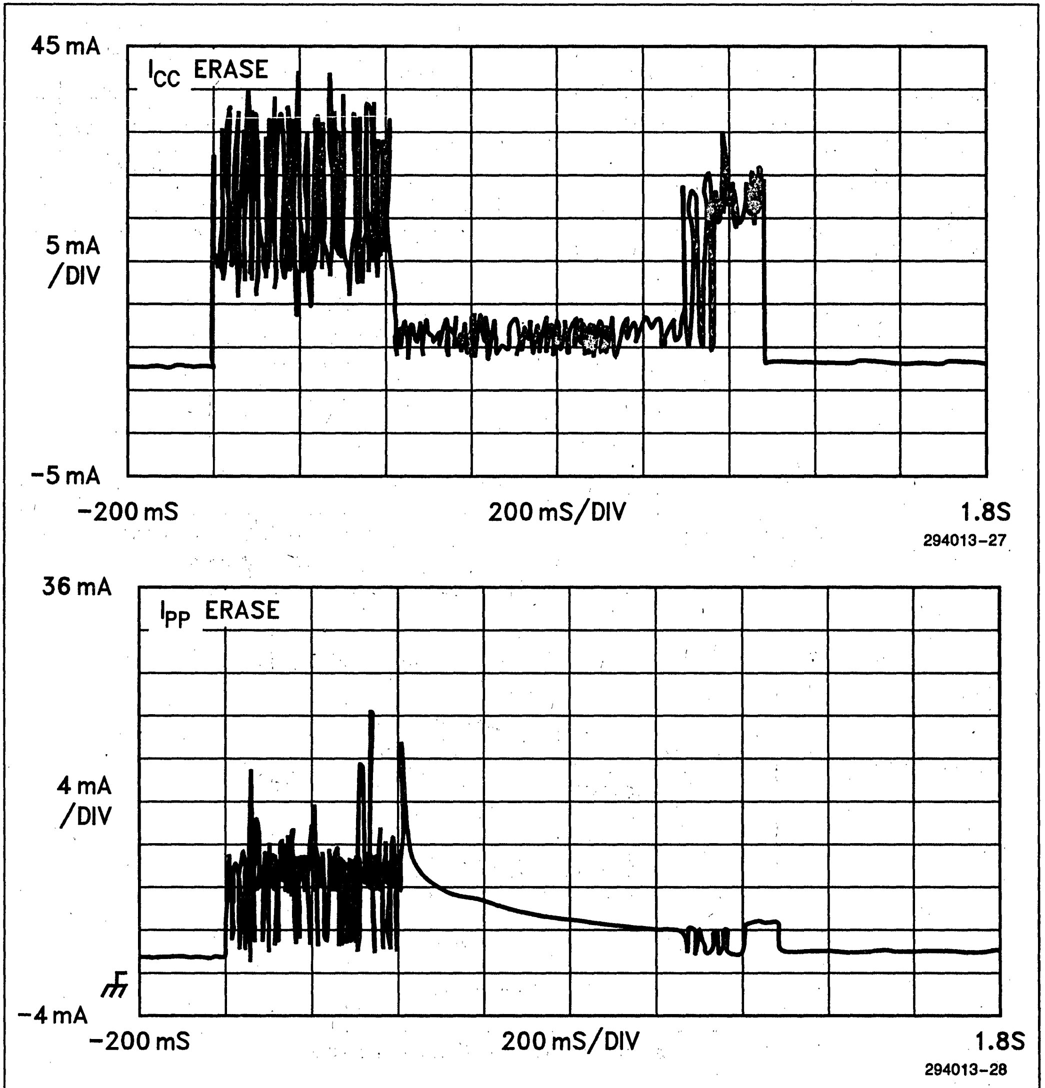


Figure 32. I_{CC} and I_{PP} during Block-Erase Operation

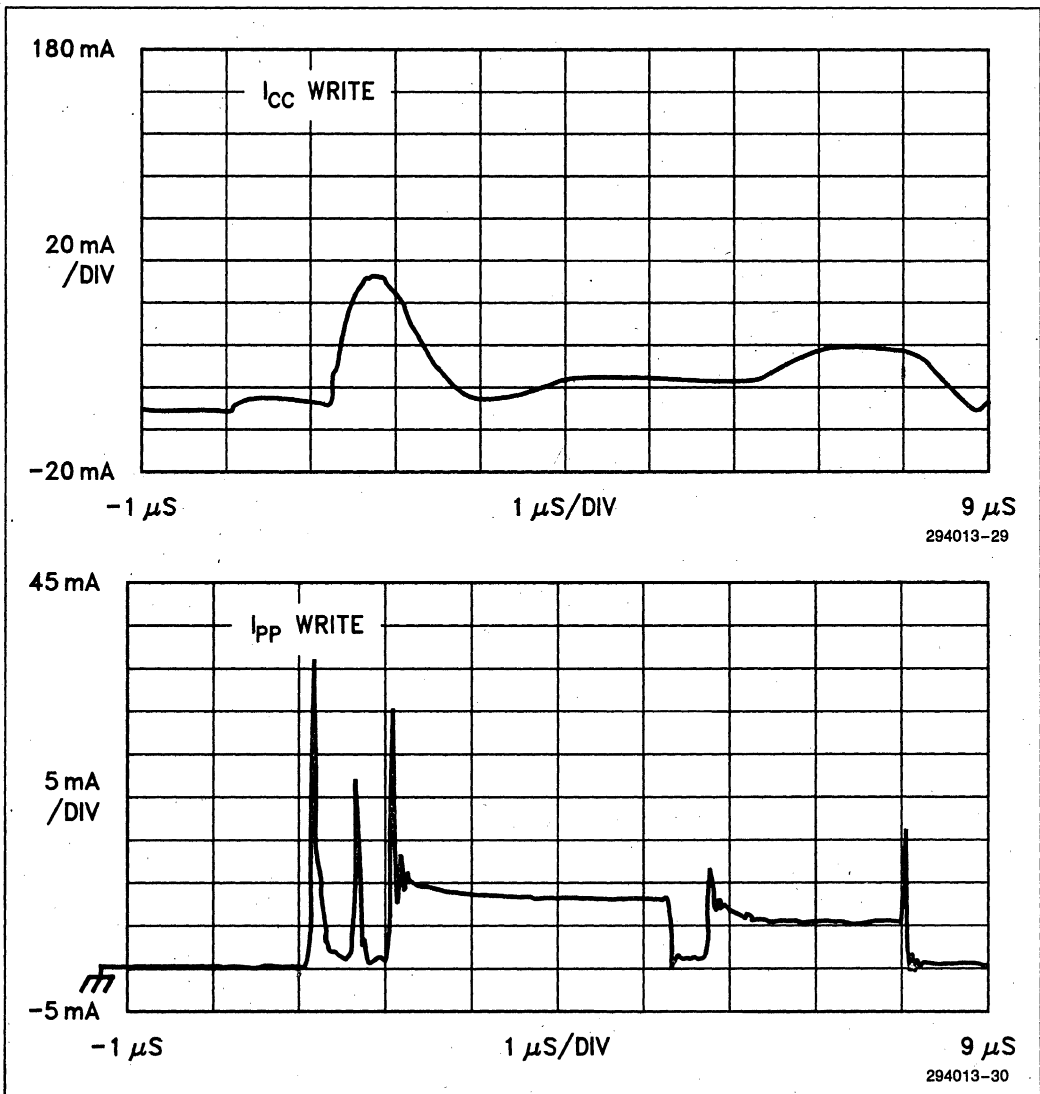


Figure 33. I_{CC} and I_{PP} during Word-Write Operation

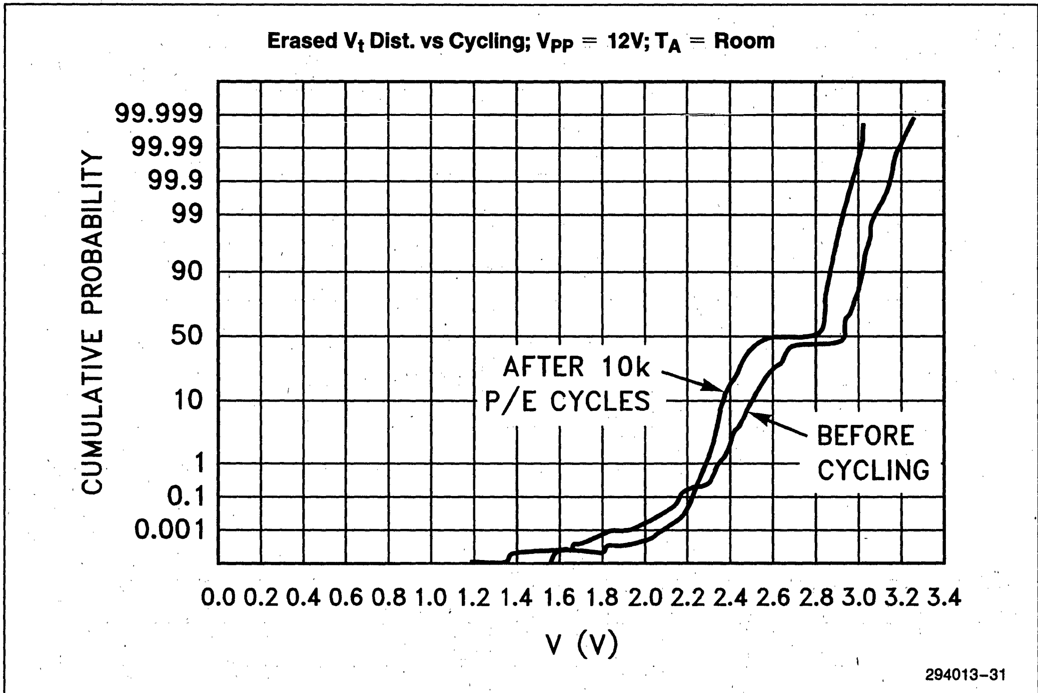


Figure 34. Typical Process Data for Erase V_t vs Cycles

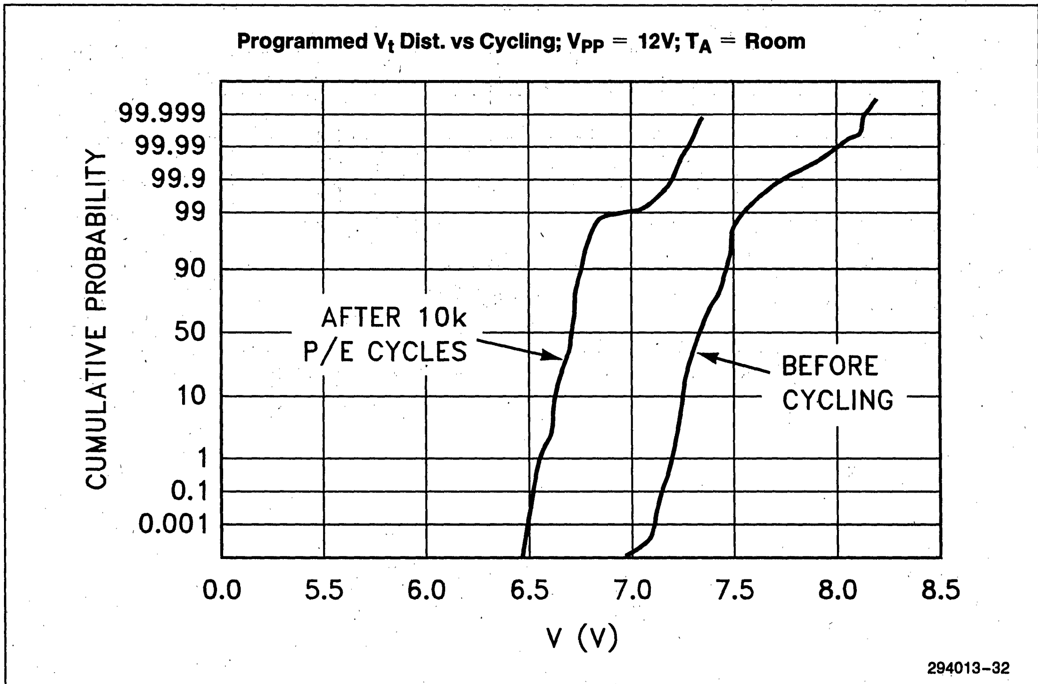


Figure 35. Typical Process Data for Program V_t vs Cycles

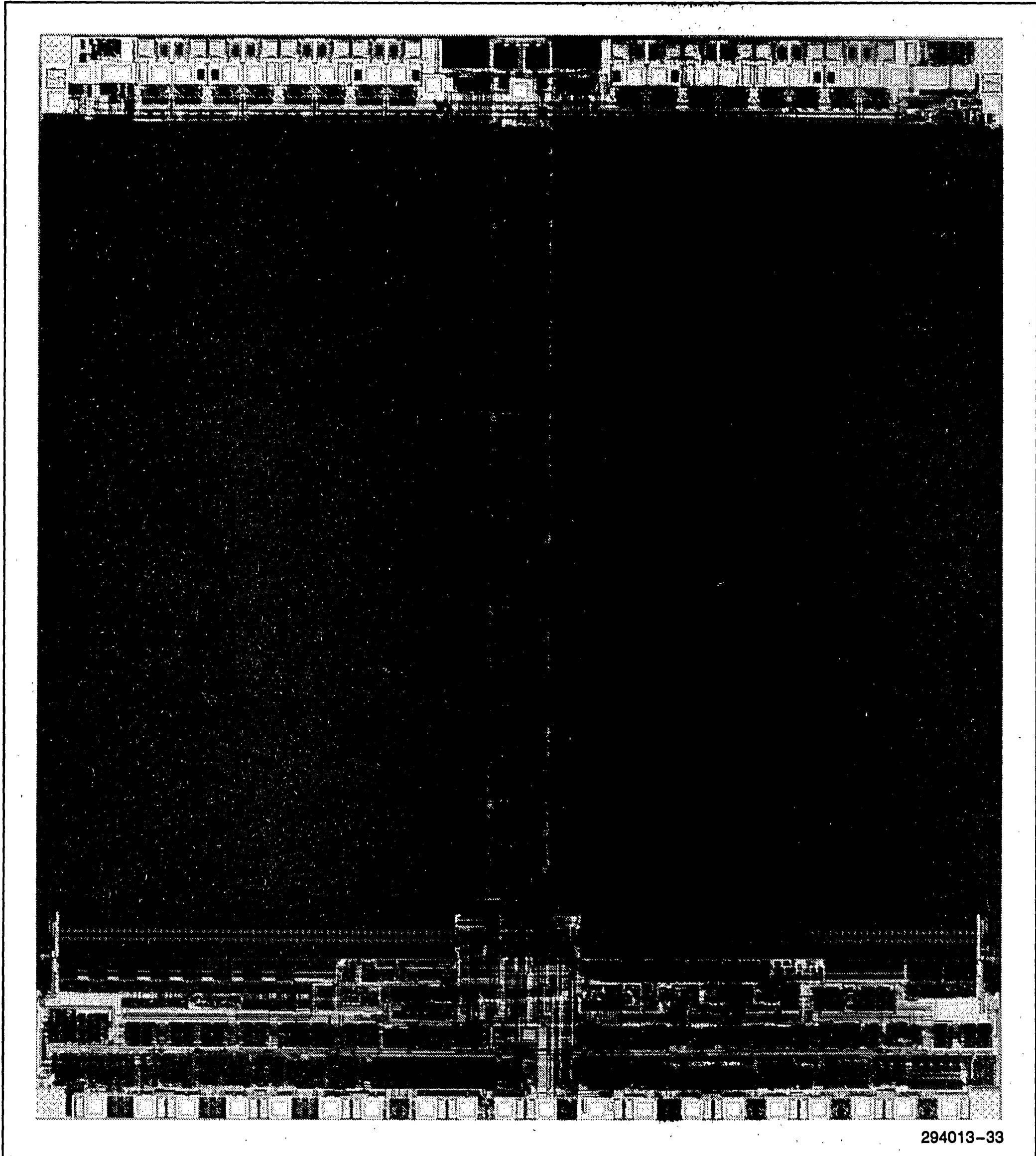

294013-33

Figure 36. 28F400BX/28F004BX Die Photograph

Pin Names

A_0–A_{17}	Address Inputs
DQ_0–DQ_{15}	Data Inputs/Outputs
$\overline{BYTE}$	Byte Enable
$\overline{CE}$	Chip Enable
$\overline{PWD}$	Power-Down/Reset
$\overline{OE}$	Output Enable
$\overline{WE}$	Write Enable
V_{PP}	Write/Erase Power Supply
V_{CC}	Device Power Supply
GND	Ground
DU	Don't Use
NC	No Internal Connection

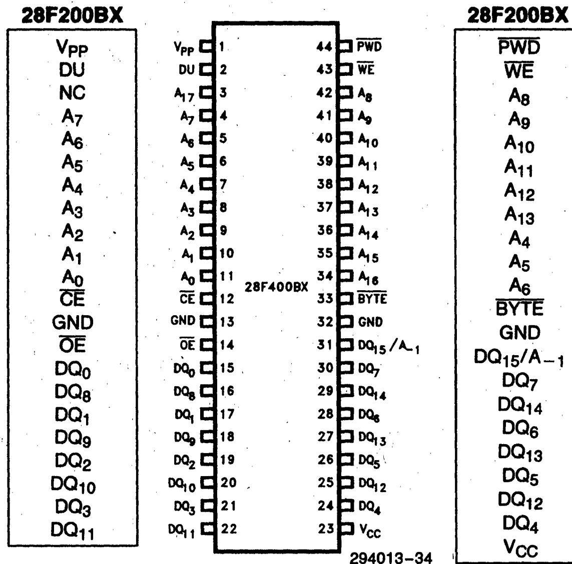
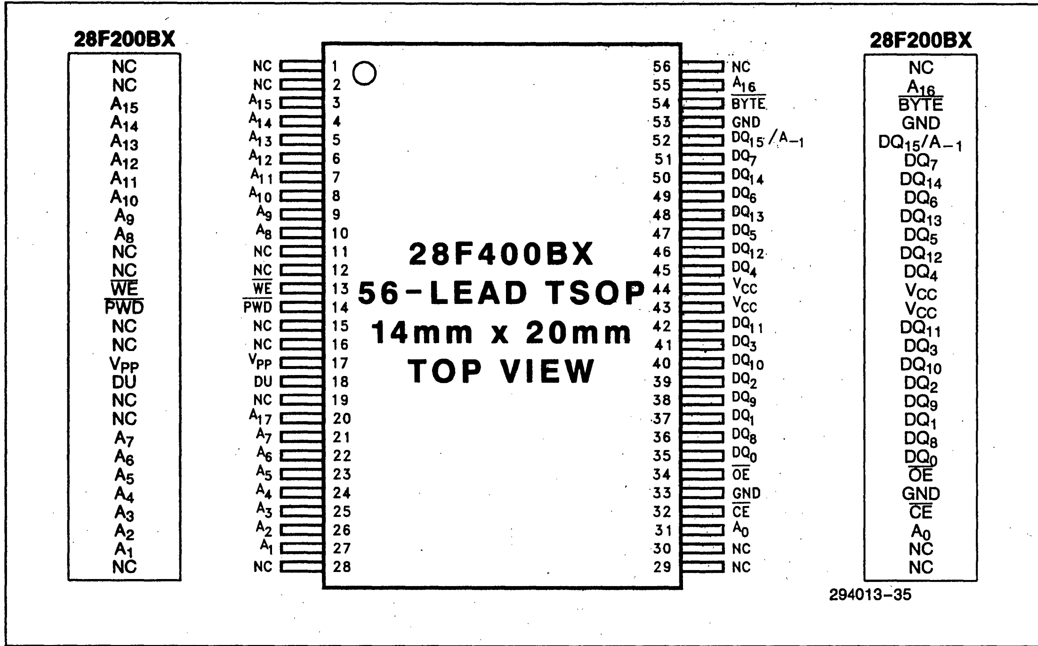


Figure 37. PSOP Lead Configuration



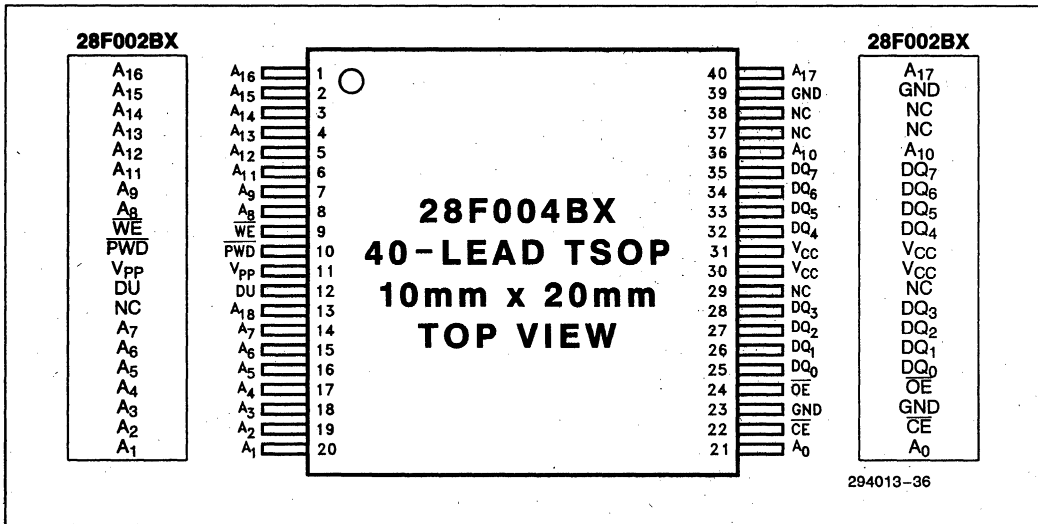


Figure 38. TSOP Lead Configuration

Addresses A_9–A_0 sequentially decode wordlines. Wordlines 0–1023 serve the upper and lower array planes.

Table 3. Row Address Bitmap

A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Wordline
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	0	0	1	1	3
0	0	0	0	0	0	0	1	0	0	4
0	0	0	0	0	0	0	1	0	1	5
0	0	0	0	0	0	0	1	1	0	6
0	0	0	0	0	0	0	1	1	1	7
0	0	0	0	0	0	1	0	0	0	8
0	0	0	0	0	0	1	0	0	1	9
0	0	0	0	0	0	1	0	1	0	10
0	0	0	0	0	0	1	0	1	1	11
0	0	0	0	0	0	1	1	0	0	12
0	0	0	0	0	0	1	1	0	1	13
0	0	0	0	0	0	1	1	1	0	14
0	0	0	0	0	0	1	1	1	1	15
0	0	0	0	0	1	0	0	0	0	16
0	0	0	0	0	1	•	•	•	•	•
0	0	0	0	0	1	1	1	1	1	31
•	•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	0	0	1008
1	1	1	1	1	1	•	•	•	•	•
1	1	1	1	1	1	1	1	1	1	1023

Columns for each block are distributed throughout the array in each I/O. For the 28F400BX/200BX, each I/O contains 8 columns for the boot block, 4 columns each for parameter block 0 and parameter block 1, 48 columns for main block 0 and 64 columns for each of the remaining main blocks (main block 1 for the 28F200BX and main block 1–3 for the 28F400BX) for a total of 256 columns per output.

Table 4. 28F400BX/200BX Column Decoding

A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	Block	Column
0	0	0	0	0	0	0	0	Boot	0
0	0	0	0	0	•	•	•	Boot	•
0	0	0	0	0	1	1	1	Boot	7
0	0	0	0	1	0	0	0	Parameter 0	8
0	0	0	0	1	0	•	•	Parameter 0	•
0	0	0	0	1	0	1	1	Parameter 0	11
0	0	0	0	1	1	0	0	Parameter 1	12
0	0	0	0	1	1	•	•	Parameter 1	•
0	0	0	0	1	1	1	1	Parameter 1	15
0	0	0	1	0	0	0	0	Main 0	16
0	0	•	•	•	•	•	•	Main 0	•
0	0	1	1	1	1	1	1	Main 0	63
0	1	0	0	0	0	0	0	Main 1	64
0	1	•	•	•	•	•	•	Main 1	•
0	1	1	1	1	1	1	1	Main 1	127
1	0	0	0	0	0	0	0	Main 2	128
1	0	•	•	•	•	•	•	Main 2	•
1	0	1	1	1	1	1	1	Main 2	191
1	1	0	0	0	0	0	0	Main 3	192
1	1	•	•	•	•	•	•	Main 3	•
1	1	1	1	1	1	1	1	Main 3	255

NOTE: 28F400BX = Complete Table, 28F200BX = Shaded Area Only

3

Columns for each block are distributed throughout the array in each I/O. For the 28F004BX/002BX, each I/O contains 16 columns for the boot block, 8 columns each for parameter block 0 and parameter block 1, 96 columns for main block 0 and 128 columns for each of the remaining main blocks (main block 1 for the 28F002BX) and main block 1–3 for the 28F004BX for a total of 256 columns per output.

Table 5. 28F004BX/002BX Column Decoding

A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	Block	Column
0	0	0	0	0	0	0	0	0	Boot	0
0	0	0	0	0	•	•	•	•	Boot	•
0	0	0	0	0	1	1	1	1	Boot	15
0	0	0	0	1	0	0	0	0	Parameter 0	16
0	0	0	0	1	0	•	•	•	Parameter 0	•
0	0	0	0	1	0	1	1	1	Parameter 0	23
0	0	0	0	1	1	0	0	0	Parameter 1	24
0	0	0	0	1	1	•	•	•	Parameter 1	•
0	0	0	0	1	1	1	1	1	Parameter 1	31
0	0	0	1	0	0	0	0	0	Main 0	32
0	0	•	•	•	•	•	•	•	Main 0	•
0	0	1	1	1	1	1	1	1	Main 0	127
0	1	0	0	0	0	0	0	0	Main 1	128
0	1	•	•	•	•	•	•	•	Main 1	•
0	1	1	1	1	1	1	1	1	Main 1	255
1	0	0	0	0	0	0	0	0	Main 2	256
1	0	•	•	•	•	•	•	•	Main 2	•
1	0	1	1	1	1	1	1	1	Main 2	383
1	1	0	0	0	0	0	0	0	Main 3	384
1	1	•	•	•	•	•	•	•	Main 3	•
1	1	1	1	1	1	1	1	1	Main 3	511

NOTE: 28F004BX = Complete Table, 28F002BX = Shaded Area Only

November 1992

3

ETOX™II Flash Memory Reliability Data Summary

Order Number: 293002-008

ETOX™II Flash Memory Reliability Data Summary

CONTENTS — PAGE

THE IMPORTANCE OF RELIABILITY 3-635

Quality ≠ Reliability 3-635

Monitor Program 3-635

ETOX™II FLASH MEMORY TECHNOLOGY OVERVIEW 3-635

Similarities with EPROM 3-636

Differences from EPROM 3-636

Erase/Write Cycling 3-637

ETOXII FLASH MEMORY RELIABILITY TESTING 3-637

Failure Rate Calculations 3-638

CONTENTS — PAGE

RELIABILITY DATA SUMMARY 3-640

P/N28F256A 3-641

P/N28F512 3-643

P/N/E 28F010 3-646

P/N/E 28F020 3-649

APPENDIX A. FAILURE RATE CALCULATIONS FOR 60% UPPER CONFIDENCE LEVEL 3-652

APPENDIX B. FLASH MEMORY BIT MAPS AND DIE PHOTOGRAPHS 3-657

THE IMPORTANCE OF RELIABILITY

Reliability of the non-volatile memories in your end product is critical to your total system reliability. The use of Intel flash memories can make a difference. Reliability is not just tested, but designed into each component Intel manufactures.

Quality ≠ Reliability

A quality component is one that meets your specification when received and tested. A reliable component continues to meet your specification even years after you have shipped your product.

CONSIDER QUALITY VS. RELIABILITY

The true cost of any component involves more than just the purchase price. The true component cost encompasses the initial purchase price, cost of rework during system production, and the cost of field repairs due to component failures. "Rework" costs during system production are incurred prior to shipment of your end product, and are a function of the quality of the component you purchase.

Repair costs incurred in the field after end product shipments, are a function of the reliability of the components. In addition to the increasing real cost of a system field service call, there is the intangible cost of a poor reliability reputation to the end use of your product. These costs depend upon the reliability of the components you purchase. Thus, reliability may impact costs during the system lifetime more than the initial quality of the components!

In-circuit reprogrammability of flash memories enables the addition of production line testing and system level screening. This capability, along with the inherent reliability of Intel flash components, provides your systems with significant reliability enhancements. Soldering the flash memory directly to the board enhances contact integrity. Since flash memories do not have to be removed for reprogramming, reliability risk due to handling is eliminated upon device installation. In addition, single socket testing reduces component handling during incoming inspection.

Monitor Program

Reliability is designed into each component Intel manufactures. From the moment the design is put to paper, stringent reliability standards must be met at each step for a product to bear the Intel name.

Designing-in reliability, however, is only the beginning. Ongoing tests must be conducted to ensure that the original reliability specifications remain as valid in volume production as they were when the device was first qualified.

Intel's Reliability Monitor Program, devised to measure and control device reliability in production, is available to our customers. The Monitor Program subjects all of Intel's technologies to a 48 hour dynamic burn-in at 125°C (with a portion of these devices continued for a 1000 hour lifetest) and provides answers about device reliability that are not generally available from limited testing programs. When test rejects are encountered, failure analysis is performed on each failed part. Isolating the fault and determining the failure mechanism is a critical part of the Monitor Program.

The primary objective is to deliver reliable, quality devices. Actions that Intel takes to meet this objective may include a process or design change, or added reliability screen. Each decision is made with our customers in mind so that they receive the parts—and the performance—that they ordered by specifying Intel. Reliability qualification assures that all new production material meets Intel's reliability standards. The Reliability Monitor Program ensures that these high standards are continually maintained over the duration of a device's life. This reliability improves the lifetime reputation of your product, reducing the required number of field service calls.

ETOX™II FLASH MEMORY TECHNOLOGY OVERVIEW

Intel's ETOX™II (EPROM tunnel oxide) flash memory technologies* consist of a non-volatile memory cell that electrically erases in bulk array form. Derived from Intel's CHMOS** II-E EPROM technology, ETOXII flash memory technology combines the EPROM program mechanism with the EEPROM erase mechanism. The memory cell is composed of a single transistor with a floating gate for charge storage, like the conventional EPROM. The primary difference between flash memory and EPROM cells is the flash memory cell's thinner gate oxide, which enables the electrical erase capability. This report compares and contrasts ETOXII technology and EPROM reliability, describes Intel's flash reliability testing methodology, and summarizes the reliability data of Intel's flash memories.

*Intel's ETOXII flash memory process has patents pending.
**CHMOS is a patented process of Intel Corporation.

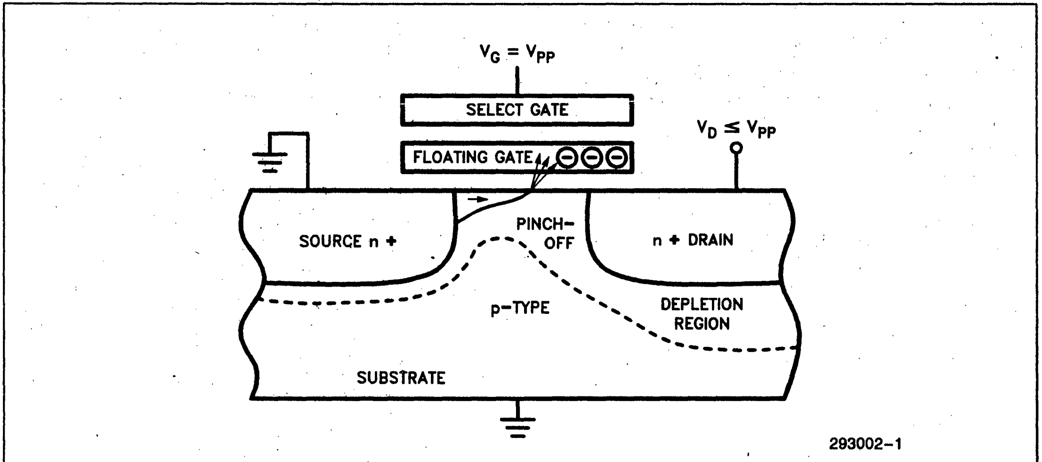


Figure 1. ETOX™II Flash Memory Cell during Programming (Side View)

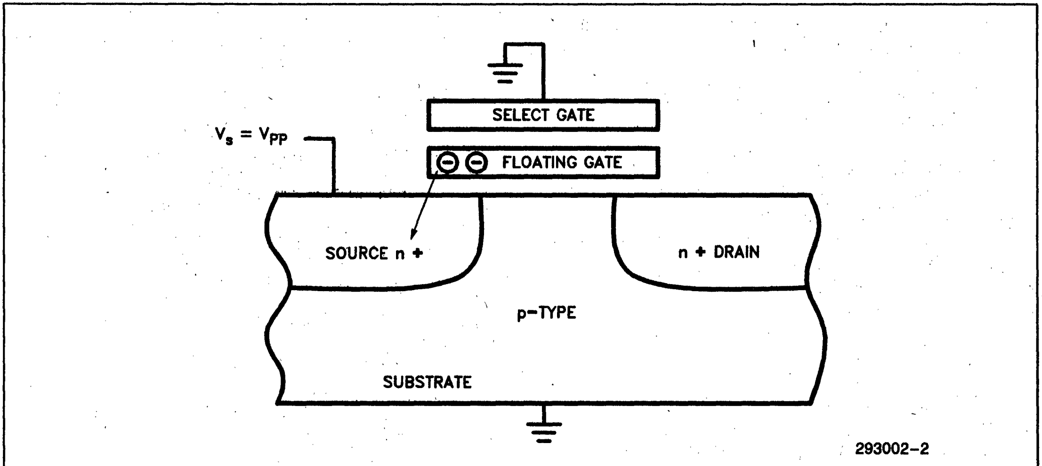


Figure 2. ETOX™II Flash Memory Cell during Erase (Side View)

Similarities with EPROM

When in program mode, a flash memory behaves exactly like a conventional EPROM. A high drain voltage generates "HOT" electrons that are swept across the channel. High voltage on the control gate attracts these free electrons across the lower gate oxide into the floating gate, where they are trapped. See Figure 1. Thus, ETOXII flash memory cells exhibit the same reliability characteristics as conventional EPROMs during program mode even with a thinner oxide. When in read mode, a flash memory behaves just like an EPROM.

Differences from EPROM

With respect to functionality, the major difference between flash memory technology and EPROM technology lies with the erase mechanism. For EPROM cells, ultraviolet light neutralizes the charge on the floating gate, thus erasing the cell. For ETOXII flash memory cells, an electric field across the lower gate oxide pulls electrons off the floating gate to the source region, thus erasing the cell. See Figure 2. This erase mechanism is an EEPROM adaptation using "Fowler-Nordheim"(1) tunneling. The electric field during erase is the only new stress compared to EPROM that may impact overall reliability.

Erase/Write Cycling

Failure mechanisms traditionally associated with cycling electrically erasable memories include charge loss due to defective bits, destructive oxide breakdown, and electron trapup. ETOXII flash memory technology minimizes these failure mechanisms by improvements in process technology, reducing the electric field stressing the gate oxide, and using efficient erase/write algorithms to control programming and erasure.

OXIDE QUALITY

Thin oxides used in tunnelling have been a reliability concern for electrically erasable memories. The quality of the ETOXII tunnel oxide is approximately 10 times better than that of other tunnel oxide approaches. This breakthrough in tunnel oxide quality results from explicit process improvements and through the implicit advantages of the ETOXII flash cell approach.

OXIDE BREAKDOWN

Oxide breakdown, due to erase/write cycling, has also been a major reliability concern for thin oxide tunnelling. ETOXII technology addresses this concern by reducing the amount of stress placed on the tunnel oxide during programming and erasure. First, erasing the flash cell involves tunnelling only through the gate/source overlap, thus reducing the area under stress. This, coupled with the improvement in oxide quality, lowers the probability of an oxide defect. Secondly, the flash cell is erased using a lower-voltage erase pulse, resulting in lower stress on the tunnel oxide. This lower electric field across the tunnel oxide (10MV/cm versus 12MV/cm) yields a theoretical wear out time 10^8 times longer than other EEPROM approaches.

ELECTRON TRAPUP

The phenomenon of electron trapup, the gradual reduction of electron mobility through the tunnel oxide, results in increasing program and erase times as cycling occurs. The program and erase algorithms must apply more pulses to add charge to or bleed charge off the floating gate to ensure data retention and integrity. This is seen as a failure to program or erase within the algorithm's allowed time and not as a hard failure. The Quick-Pulse Programming™ and Quick-Erase™ algorithms maintain an efficient program and erase time for the specified number of cycles listed in the flash memory data sheets.

ETOXII FLASH MEMORY RELIABILITY TESTING

Intel flash memories undergo comprehensive testing to insure electrical reliability. This testing is done at qualification and during ongoing monitor checks.

Information on flash memory reliability testing procedures follows.

High Temperature 5.25V Dynamic Lifetest—This test is used to accelerate failure mechanisms by operating the devices at an elevated temperature of 125°C. During the test, the memory is sequentially addressed and the outputs are exercised, but not monitored or loaded. A checkerboard data pattern is used to simulate random patterns expected during actual use. Results of lifetesting have been summarized along with the failure analysis.

In order to best determine long-term failure rate, all devices used for lifetesting are subjected to standard INTEL testing. The 48 hour burn-in results are an indication of infant mortality and are not included in the failure rate calculation. (See Figure 3 for typical burn-in bias and timing diagrams.)

High Temperature High Voltage Dynamic Lifetest—This test is used to accelerate oxide breakdown failures. The test setup is identical to the one used for the dynamic lifetest except V_{CC} is increased. The acceleration factor due to this test can be found in the failure rate prediction tables. This data plus the standard dynamic lifetest data are used to calculate the 0.3 eV failure rate (See Figure 4 for typical bias and timing diagram).

Data Retention Bake—This test is used to accelerate charge loss from the floating gate. The test is performed by subjecting devices containing a 98% programmed pattern to a 250°C bake with no applied bias. In addition to data retention, this test is used to detect mechanical reliability problems (e.g., bond integrity) and process instability.

Temperature Cycle—This test consists of cycling the temperature of the chamber housing the subject devices from −65°C to +150°C and back. One thousand cycles are performed with a complete cycle taking 20 minutes. This test is to detect mechanical reliability problems and microcracks.

ESD Testing—This test is performed to validate the product's tolerance to Electro Static Discharge damage. All products incorporate ESD protection networks on appropriate pins.

Two types of tests are performed. First, all devices are tested using Mil STD 883 test criteria. In addition, a charged device test is performed to further validate protection occurring during mechanical handling.

Erase/Write Cycling (ETOXII Flash Memories)—This test consists of repeatedly programming the device to an all 00H pattern and then erasing to all 0FFH data. Worst case voltage levels are used to maximize charge transfer to and from the floating gates. Cycling is used to ensure devices meet reprogrammability requirements as well as precondition for other reliability stresses.

85/85 Test

During the 85°C/85% relative humidity test, the devices are subjected to a high temperature, high humidity environment. The object of the test is to accelerate failure mechanisms through an electrolytic process. This is accomplished through a combination of moisture penetration of the plastic, voltage potentials and contamination which, if present, would combine with the moisture to act as an electrolyte.

Steam

Steam stressing performed at 121°C, 2 atm. accelerates moisture penetration through the plastic package material to the surface of the die. The objective of this test is to accelerate failures of the device as a result of moisture on the die surface. Corrosion, as typically seen in plastic encapsulated devices, is a very minor contributor to the Flash failure mechanisms. Due to the floating gate storage cell composition, Flash memories have a distinctive failure mode which requires special considerations and solutions.

The floating gate itself is a highly phosphorous doped structure on which electrons are stored, thus creating the non-volatile memory cell. Passivation defects or marginalities can allow moisture penetration to a single Flash cell causing oxide deterioration, thus showing up as a charge loss failure. This becomes the predominant failure mode for Flash product, opposed to corrosion which historically has been the dominant plastic mode of failure. Intel has developed a proprietary, multi-layer passivation which has successfully solved this problem.

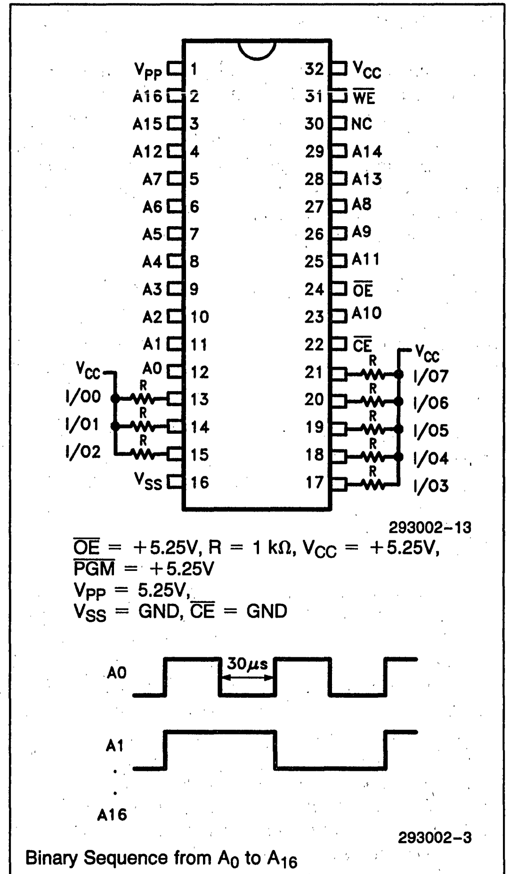


Figure 3. 28F010 Burn-In Bias and Timing Diagrams

Failure Rate Calculations

Failure rate calculations are given for each relevant activation energy. Failure rate calculations are made using the appropriate energy (2,3,4,5) and the Arrhenius Plot as shown in Figure 5*. The total equivalent device hours at a given temperature can be determined. The failure rate is then calculated by dividing the number of failures by the equivalent device hours and is expressed as a %/1000 hours. To arrive at a confidence level associated failure rate, the failure rate is adjusted by a factor related to the number of device hours using a chi-square distribution. A conservative estimate of the failure rate is obtained by including zero failures at 0.3 eV.

NOTE:

*The activation energies for various failure mechanisms are listed in Table 1. The methodology for calculating failure rates is detailed in Appendix A.

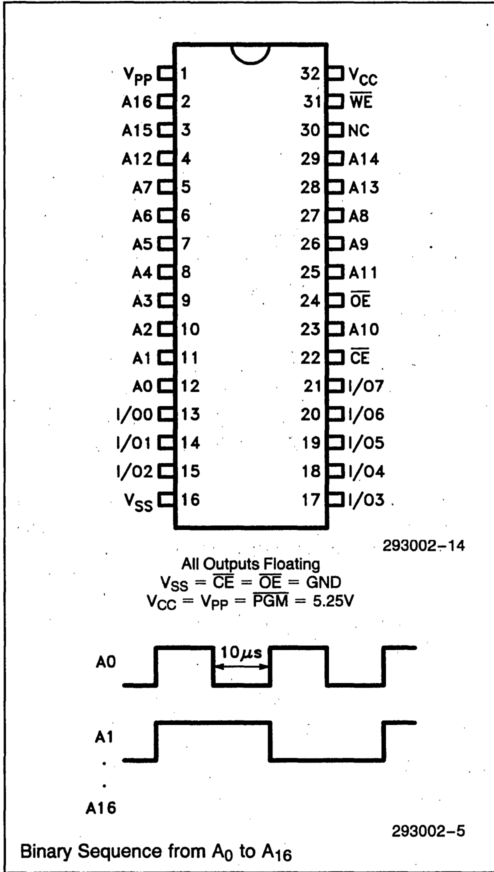


Figure 4. 28F010 Lifetest Bias and Timing Diagram

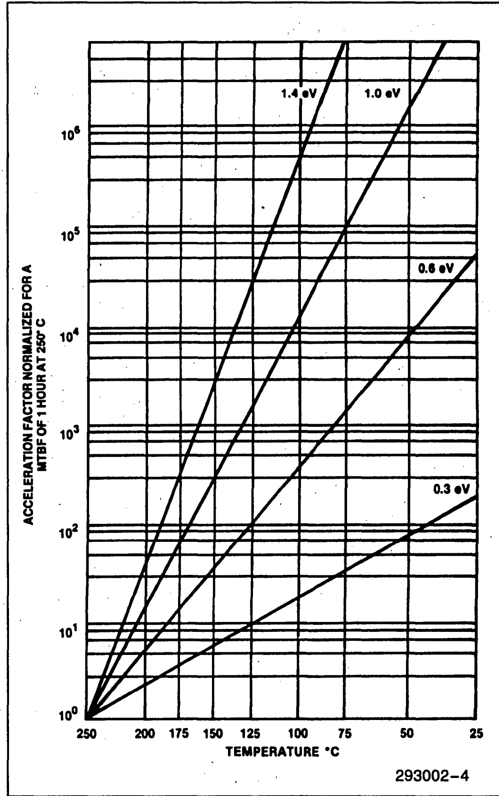


Figure 5. Arrhenius Plot

Table 1. Failure Mechanism Activation Energies Relevant to ETOXII Flash Memories

Failure Mode	eV
Oxide	0.3
SBCL/SBCG/MBCL/MBCG	0.6
Contamination	1.0
Speed Degradation	0.3-1.0
Intrinsic Charge Loss	1.4
Contact Spiking	0.8

A typical lifetest bias and timing diagram is shown in Figure 4.

RELIABILITY DATA SUMMARY

The following data is an accumulation of recent qualification and monitor program results. Failure rate calculation methods listed in Appendix A were used to arrive at the tabularized failure rates.

In reviewing the reliability data as presented, questions may arise as to why lot sizes often decrease from one test to another without a corresponding number of identified failures. This is due to a variety of factors. Many tests require smaller sample sizes and as a result all parts from a previous test do not necessarily flow through to a succeeding test.

In addition, various parts are pulled from a sample lot when mechanical or handler failures occur. These "failures" are not a result of the specific test just completed but are nonetheless removed from the sample lot size and are not included in any failure rate calculation. It can also happen that a particular test is done incorrectly through human error or faulty test equipment and "invalid" failures are put aside for retesting at a later date, decreasing the lot size for a succeeding test. If these parts are found to be truly defective, they are treated as failures and listed. If they test out properly, they are removed from any calculation data base.

References

1. M. Lenzlinger.E. H. Snow, "Fowler-Nordheim tunneling into thermally grown SiO_2", Journal of Applied Physics, Vol. 40 (1969), p. 278.
2. S. Rosenberg, D. Crook, B. Euzent, "16th Annual Proceedings of the International Reliability Physics Symposium", pp. 19–25, 1978.
3. S. Rosenberg, B. Euzent, "HMOS Reliability" Reliability Report RR-18, Intel Corporation, 1979.
4. R. M. Alexander, "Calculating Failure Rates from Stress Data, April 1984 International Reliability Physics Symposium.
5. "EPROM Reliability DATA Summary" Reliability Report RR-35, Intel Corporation, 1985.
6. "E^2PROM and NVRAM Reliability DATA Summary" Reliability Report RR-59B, Intel Corporation, 1986.
7. E.S. Anolick, G.R. Nelson, "Low Field Time Dependent Dielectric Integrity", 1979 International Reliability Physics Symposium, pp. 8–12.

NOTE:
The methodology for calculating failure rates is detailed in Appendix A.

28F256A

The Intel 28F256A is a 256-Kbit bulk-erasable flash memory.

Number of Bits:	262,144	Process:	ETOXII Flash Memory
Organization:	32,768 × 8	Technology:	CMOS
Pin Out:	32-Lead PDIP (P)/32-Lead PLCC (N)	Cell Size:	3.8 × 4.0 μM
Die Size:	228 × 141 mils	Programming Voltage Options:	12.0V ±5%

Table 1. Reliability Data Summary

Year	Burn-In	125°C Dynamic Lifetest				7.0V Dynamic Lifetest		Program/Erase
	48 Hrs	168 Hrs	500 Hrs	1K Hrs	2K Hrs	48 Hrs	168 Hrs	10K
1990	0/300	1/300	0/298	1/295	0/294	3/3004	1/2996	1/408
1991	0/374	0/374	1/374	0/372		0/2040	0/2039	0/300
1992	0/725	0/724	0/722	0/689		1/8955	0/8942	
Total	0/1399	1/1398	1/1394	1/1356	0/294	4/13999	1/13977	1/708
		A	F	B		C	D	E

Table 2. Additional Qualification Tests—PLCC (N)

Year	140°C Data Retention Bake				Temperature Cycling		
	48 Hrs	168 Hrs	500 Hrs	1K Hrs	200 Cycles	500 Cycles	1k Cycles
1990							
1991							
1992	0/525	0/525	0/525	0/523	0/638	0/633	0/633
Total	0/525	0/525	0/525	0/523	0/638	0/633	0/633

Year	85°C/85% RH			Steam
	168 Hrs	500 Hrs	1K Hrs	168 Hrs
1990				
1991				
1992	0/450	0/449	0/449	0/640
Total	0/450	0/449	0/449	0/640

Table 3. Additional Qualification Tests—PDIP (P)

Year	Temperature Cycling			Thermal Shock		
	200 Cycles	500 Cycles	1K Cycles	50 Cycles	200 Cycles	500 Cycles
1990	0/160	0/160	0/158	0/160	0/160	0/160
1991	0/239	0/239	0/239	—	—	—
Total	0/399	0/399	0/397	0/160	0/160	0/160

P/N28F256A Failure Rate Prediction

125°C Actual Device Hours	Activation Energy (eV)	Equivalent Hours		# Fail	Failure Rate %/1K Hours (60% U.C.L.)	
		55°C	70°C		55°C	70°C
1.61E + 06 1.68E + 06	0.3 eV ELT 0.3 eV HVLT × VAF	1.04E + 07 2.82E + 08	6.55E + 06 1.77E + 08	2 1		
		TOTAL 0.3 eV Failures =		3	0.0015	0.0023
1.61E + 06 1.68E + 06	0.6 eV ELT 0.6 eV HVLT	6.72E + 07 7.01E + 07	2.65E + 07 2.77E + 07	0 0		
		TOTAL 0.6 eV Failures =		0	0.0007	0.0017
1.61E + 06 1.68E + 06	1.0 eV ELT 1.0 eV HVLT	8.11E + 08 8.45E + 08	1.73E + 08 1.80E + 08	0 0		
		TOTAL 1.0 eV Failures =		0	0.0001	0.0003
				Combined Failure Rate: FITs:	0.0023 23	0.0043 43

VAF = Voltage Acceleration Factor of 26

Failure Analysis:

A. 1-Output shorted—destroyed in analysis
B. 1-lsb
C. 3-lsb 1-Single Bit Charge Loss
D. V_{CC} max CAM failure
E. 1-Decoder failure; carbon defect.
F. 1-Input leakage; bondpad to substrate short

28F512

The Intel 28F512 is a 512-Kbit bulk-erasable flash memory.

Number of Bits:	524,288	Process:	ETOXII Flash Memory
Organization:	65,536 × 8	Technology:	CMOS
Pin Out:	32-Pin PDIP (P)/32-Pin PLCC (N)	Cell Size:	3.8 × 4.0 μM
Die Size:	227 × 181 mils	Programming Voltage Options:	12.0V ±5%

Table 1. Reliability Data Summary

Year	Burn-In	125°C Dynamic Lifetest				6.5V Dynamic Lifetest			Program/Erase
	48 Hrs	168 Hrs	500 Hrs	1K Hrs	2K Hrs	48 Hrs	168 Hrs	500 Hrs	10K
1990	0/99	0/99	0/98	0/98	1/97	0/299	0/299	1/299	2/770
1991						1/4576	0/4575	1/391	0/300
1992	0/1000	0/999	0/999	0/999		1/10988	0/10929		
Total	0/1099	0/1098	0/1097	0/1097	1/97	2/15863	0/15803	2/690	2/1070
					A	D		B	E

Table 2. Additional Qualification Tests—PLCC(N)

Year	140°C Data Retention Bake		
	48 Hours	168 Hours	500 Hours
1990	0/59	0/59	0/59
1991			
1992			
Total	0/59	0/59	0/59

Year	Temperature Cycling			Thermal Shock		
	200 Cycles	500 Cycles	1K Cycles	50 Cycles	200 Cycles	500 Cycles
1990	0/234	0/234	1/234	0/160	0/160	0/160
1991	0/199	0/199	0/199			
1992						
Total	0/433	0/433	1/433	0/160	0/160	0/160
			C			

Year	85°C/85% RH				Steam	
	168 Hours	500 Hrs	1K Hrs	2K Hrs	168 Hrs	336 Hrs
1990	0/250	0/250	0/250	0/227	0/160	0/160
1991						
1992						
Total	0/250	0/250	0/250	0/227	0/160	0/160

Table 3. Additional Qualification Tests—PDIP(P)

Year	140°C Data Retention Bake				Temperature Cycling		
	48 Hrs	168 Hrs	500 Hrs	1K Hrs	200 Cycles	500 Cycles	1K Cycles
1990							
1991					0/163	0/163	0/163
1992	0/825	0/825	0/825	0/823	0/867	0/863	0/863
Total	0/825	0/825	0/825	0/823	0/1030	0/1026	0/1026

Year	85°C/85% RH			Steam
	168 Hrs	500 Hrs	1K Hrs	168 Hrs
1990				
1991	0/100	0/100	0/100	0/98
1992	0/650	0/650	0/650	0/815
Total	0/750	0/750	0/750	0/913

P/N28F512 Failure Rate Prediction

125°C Actual Device Hours	Ea (eV)	Equivalent Hours		# Fail	Fail Rate %/1K Hours	
		55°C	70°C		55°C	70°C
1.15E + 06 2.04E + 06	0.3 BI 0.3 × VAF	7.44E + 06 3.43E + 08	4.68E + 06 2.15E + 08	0 1		
		TOTAL 0.3 eV Failures =		1	0.0006	0.0010
1.15E + 06 2.04E + 06	0.6 BI 0.6 HVELT	4.81E + 07 8.53E + 07	1.90E + 07 3.37E + 07	1 1		
		TOTAL 0.6 eV Failures =		2	0.0024	0.0059
1.15E + 06 2.04E + 06	1.0 BI 1.0 HVELT	5.80E + 08 1.03E + 09	1.23E + 08 2.19E + 08	0 0		
		TOTAL 1.0 eV Failures =		0	0.0001	0.0003
			Combined Failure Rate: FITs:		0.0031 31	0.0072 72

Voltage Accel. Factor (VAF)
for HVELT on this process is = 26.0

3

Failure Analysis:
A. 1-Single bit charge loss
B. 1-Defect not found
   1-lsb
C. 1-Shorts failure due to metal stringer.
D. 1-Single bit change loss
   1-Passivation Crack
E. 1-Dual Column
   1-Erase pushout

28F010

The Intel 28F010 is a 1024-Kbit bulk-erasable flash memory.

Number of Bits:	1,048,576	Process:	ETOXII Flash Memory
Organization:	2 (512 × 1024)	Technology:	CMOS
Pin Out:	32-Pin PDIP (P) 32-Pin PLCC (N) 32-Pin TSOP (E/F)	Cell Size:	3.8 × 4.0 μM
		Programming Voltage:	12.0V ±5%
Die Size:	225 × 265 mils		

Table 1. Reliability Data Summary

Year	Burn-In	125°C Dynamic Lifetest				6.5V Dynamic Lifetest			Program/Erase
	48 Hrs	168 Hrs	500 Hrs	1K Hrs	2K Hrs	48 Hrs	168 Hrs	500 Hrs	10K
1990	0/1396	0/1389	0/1385	1/1385	3/602	2/3009	1/3002	1/1803	14/4429
1991	0/2553	0/2552	0/2550	0/1250	2/854	2/6021	1/6018	1/2553	5/5282
1992	2/6902	0/6895	2/4692	0/2693		1/12940	0/12914		1/450
Total	2/10851	0/10836	2/8627	1/5328	5/1456	5/21970	2/21934	2/4356	20/10161
	H		I	A	B	C	D	E	O

Table 2. Additional Qualification Tests—PLCC (N)

Year	140°C Data Retention Bake		
	48 Hours	168 Hours	500 Hours
1990	0/147	0/147	0/146
1991			
1992	0/969	0/969	0/969
Total	0/1116	0/1116	0/1116

Year	Temperature Cycling			Thermal Shock		
	200 Cycles	500 Cycles	1K Cycles	50 Cycles	200 Cycles	500 Cycles
1990	0/555	0/555	0/553	0/397	0/397	0/397
1991	2/567	0/564	0/564	0/318	0/318	2/318
1992	0/957	0/946	1/937			
Total	2/2079	0/2065	1/2054	0/715	0/715	2/715
	L		P			M

Year	85°C/85% RH				Steam
	168 Hours	500 Hrs	1K Hrs	2K Hrs	168 Hrs
1990	1/695	0/692	0/689	0/686	0/546
1991	0/485	0/485	0/485		2/842
1992	0/592	0/582	0/465		0/1119
Total	1/1772	0/1759	0/1639	0/686	2/2507
	F				N

Table 3. Additional Qualification Tests—PDIP (P)

Year	Temperature Cycling			Thermal Shock		
	200 Cycles	500 Cycles	1K Cycles	50 Cycles	200 Cycles	500 Cycles
1990	1/428	0/427	0/426	0/351	0/351	0/207
1991	0/120	0/120	0/120	0/127	0/127	0/127
1992						
Total	1/548	0/547	0/546	0/478	0/478	0/334
	G					

Year	85°C/85% RH			140°C Data Retention Bake				Steam
	168 Hrs	500 Hrs	1K Hrs	48 Hrs	168 Hrs	500 Hrs	1K Hrs	168 Hrs
1990	0/453	0/453	0/453	0/675	0/675	0/675		
1991	0/100	0/100	0/100	0/695	0/695	0/695		0/98
1992	0/650	0/650	0/650	0/210	0/210	0/210	0/210	0/815
Total	0/750	0/750	0/750	0/1580	0/1580	0/1580	0/210	0/913

Table 4. Additional Qualification Tests—TSOP (E)

Year	Temperature Cycling			Thermal Shock		
	200 Cycles	500 Cycles	1K Cycles	50 Cycles	200 Cycles	500 Cycles
1990	0/234	0/232	0/232	0/233	0/233	0/233
Total	0/234	0/232	0/232	0/233	0/233	0/233

Year	85°C/85% RH				Steam
	168 Hours	500 Hrs	1K Hrs	2K Hrs	168 Hrs
1990	1/385	0/383	1/383	0/382	0/446
Total	1/385	0/383	1/383	0/382	0/446
	J		K		

3

P/N/E28F010 Failure Rate Prediction

125°C Actual Device Hours	Ea (eV)	Equivalent Hours		# Fail	Fail Rate %/1K Hours	
		55°C	70°C		55°C	70°C
8.28E + 06 4.08E + 06	0.3 BI 0.3 + VAF	5.36E + 07 6.86E + 08	3.37E + 07 4.32E + 08	8 2		
		TOTAL 0.3 eV Failures =		10	0.0016	0.0025
8.28E + 06 4.08E + 06	0.6 BI 0.6 HVELT	3.46E + 08 1.70E + 08	1.37E + 08 6.73E + 07	0 2		
		TOTAL 0.6 eV Failures =		2	0.0007	0.0016
8.28E + 06 4.08E + 06	1.0 BI 1.0 HVELT	4.18E + 09 2.06E + 09	8.89E + 08 4.38E + 08	0 0		
		TOTAL 0.8 eV Failures =		0	0.0001	0.0001
				Combined Failure Rate: FITs:	0.0024 24	0.0042 42

Voltage Accel. Factor (VAF)
for HVELT on this process is = 26.0

Failure Analysis:

A. 1-Bond pad leakage

B. 3-Bond pad leakage
   2-Speed failure

C. 3-lsb
   2-Single bit charge loss

D. 2-Single bit charge loss

E. 2-lsb

F. 1-Cluster bit charge loss (passivation damage)

G. 1-lsb

H. 1-Single bit charge loss
   1-Output leakage

I. 1-lsb
   1-Output leakage

J. 1-lsb

K. 1-Input leakage—pass after bake

L. 1-Single bit charge loss
   1-lsb

M. 1-Input leakage
   1-Opens

N. 2-Single bit charge loss

O. 9-Dual column (metal filaments)
   2-Erase pushout
   1-lsb
   3-Spurious program
   2-Oxide breakdown
   1-Poly defect
   2-Slow to program

P. 1-Delamination

P/N/E28F020

The Intel 28F020 is a 2048-Kbit bulk-erasable flash memory.

Number of Bits:	2,097,152	Process:	ETOX™II Flash Memory
Organization:	2 (512 × 2048)	Technology:	CMOS
Pin Out:	32-pin PLCC/PDIP/TSOP	Cell Size:	3.8 × 4.0 μM
Die Size:	258 × 423 mils	Programming Voltage:	12.0V ±5%

Table 1. Reliability Data Summary

Year	Burn-In	125°C Dynamic Lifetest				6.5V Dynamic Lifetest				
	48 Hours	168 Hrs	500 Hrs	1K Hours	2K Hrs	48 Hrs	168 Hrs	500 Hrs	1K Hrs	2K Hrs
1990	0/588	0/588	0/587	0/585	2/485	0/883	1/878	0/871	0/869	0/713
1991	0/288	0/288	0/286	0/286	2/186	0/433	0/432	0/431	0/431	0/281
1992						0/374	0/374	0/374	0/374	0/298
Total	0/876	0/876	0/873	0/871	4/671	0/1690	1/1684	0/1676	0/1674	0/1292
					A		B			

Table 1A. Reliability Data Summary

Year	Program/Erase	7.0V Dynamic Lifetest	
	10K	48 Hrs	168 Hrs
1990	6/2244	1/2859	0/2858
1991	2/1029	1/2859	0/2854
1992	0/533	0/2997	1/2993
Total	8/3806	2/8715	1/8705
	C	D	H

Table 2. Additional Qualification Tests—PLCC (N)

Year	Temperature Cycling			Thermal Shock		
	200 Cycles	500 Cycles	1K Cycles	50 Cycles	200 Cycles	500 Cycles
1990	0/382	0/380	0/380	0/79	0/79	0/78
1992	0/79	0/77	0/77			
Total	0/461	0/457	0/457	0/79	0/79	0/78

Year	85°C/85% RH				Steam	
	168 Hours	500 Hrs	1K Hrs	2K Hrs	168 Hrs	336 Hrs
1990	1/325	0/324	0/324	0/250	1/229	0/228
1992	0/50	0/50	0/42		0/80	
Total	1/375	0/374	0/366	0/250	1/309	0/228
	E				F	

Table 3. Additional Qualification Tests—PDIP (P)

Year	140°C Data Retention Bake		
	48 Hours	168 Hours	500 Hours
1990	0/226	0/226	0/226
1991	0/226	0/226	0/226
Total	0/452	0/452	0/452

Year	Temperature Cycling			Thermal Shock		
	200 Cycles	500 Cycles	1K Cycles	50 Cycles	200 Cycles	500 Cycles
1990	0/474	0/474	16/474	0/474	0/473	0/399
1991	0/234	0/234	0/234	0/233	0/233	—
Total	0/708	0/708	16/708	0/707	0/706	0/399
			G			

Year	85°C/85% RH				Steam
	168 Hours	500 Hrs	1K Hrs	2K Hrs	168 Hrs
1990	0/395	0/389	0/389	0/144	0/415
1991	0/245	0/245	0/245	0/245	0/189
Total	0/640	0/634	0/634	0/389	0/604

Table 4. Additional Qualification Tests—TSOP (T)

Year	Temperature Cycling			Thermal Shock		
	200 Cycles	500 Cycles	1K Cycles	50 Cycles	200 Cycles	500 Cycles
1990	0/233	0/231	0/231	0/234	0/231	0/231
Total	0/233	0/231	0/231	0/234	0/231	0/231

Year	85°C/85% RH			Steam
	168 Hours	500 Hrs	1K Hrs	168 Hrs
1990	0/382	0/382	0/382	0/387
Total	0/382	0/382	0/382	0/387

P/N/E 28F020 Failure Rate Prediction

125°C Actual Device Hours	Activation Energy (eV)	Equivalent Hours		# Fail	Failure Rate %/1K Hours (60% U.C.L.)	
		55°C	70°C		55°C	70°C
1.04E + 06 1.05E + 06 2.89E + 06	0.3 eV HVLT + IME 0.3 eV ELT 0.3 eV HVLT + VAF	6.26E + 08 6.79E + 06 4.86E + 08	3.94E + 08 4.27E + 06 3.06E + 08	0 0 1		
TOTAL 0.3 eV Failures =				1	0.0002	0.0003
1.05E + 06 2.89E + 06	0.6 eV ELT 0.6 eV HVLT	4.39E + 07 1.21E + 08	1.73E + 07 4.76E + 07	4 1		
TOTAL 0.6 eV Failures =				5	0.0039	0.0097
1.05E + 06 2.89E + 06	1.0 eV ELT 1.0 eV HVLT	5.29E + 08 1.46E + 09	1.13E + 08 3.10E + 08	0 0		
TOTAL 1.0 eV Failures =				0	0.0001	0.0003
Combined Failure Rate: FITs:					0.0042 42	0.0103 103

VAF = Voltage Acceleration Factor of 26 (@6.5V)
IME = Voltage Acceleration Factor of 93 (@7.0V)

Failure Analysis:
A. 4-Single bit charge loss (0.6 eV)
B. 1-lsb (0.3 eV)
C. 2-Dual column
   6-Higher erase time
D. 2-lsb
E. 1-Open output
F. 1-lsb
G. 16-Leakage due to thin film cracking—design fix in place for 1991
H. 1-Single bit charge gain

APPENDIX A
FAILURE RATE CALCULATIONS FOR 60% UPPER CONFIDENCE LEVEL

Step 1. Accumulate data from 48 hours of burn-in through lifetest of each lot. (Note: 48-hour burn-in results measure infant mortality and are not included in the failure rate calculation.)

Step 2. Determine the failure rate mechanism for each failure and assign an activation energy (E_A) corresponding to each failure mechanism. (See Table 1 below.)

Table 1. Failure Mechanisms Activation Energies Relevant to ETOX™II Flash Memories

Failure Mode	Activation Energy
Defective Big Charge Gain/Loss	0.6 eV
Oxide Breakdown	0.3 eV
Silicon Defects	0.3 eV
Contamination	1.0 eV–1.2 eV
Intrinsic Charge Loss	1.4 eV

Step 3. Calculate the total number of device hours from 48 hours of burn-in through lifetest.

Example: 125°C Burn-In/Lifetest and a 2 lot sample

$$\frac{\#\ \text{failures}}{\text{total}\ \#\ \text{devices}}$$

	48 Hours	168 Hours	500 Hours	1K Hours	2K Hours
Lot #1	0/1000	1/1000	0/999	0/998	0/994
Lot #2	0/221	0/201	1/201	1/100	0/99
Totals	0/1221	1/1201	1/1200	1/1098	0/1093

Device Hours = (Number of Devices) (Number of Hours)

Total Device Hours = 1201 (168 hrs − 48 hrs) + 1200 (500 hrs − 168 hrs)

+ 1098 (1000 hrs − 500 hrs) + 1093 (2000 hrs − 1000 hrs)

= 1201 (120 hrs) + 1200 (332) + 1098 (500 hrs)

+ 1093 (1000 hrs)

= 2.185×10^6 Device Hours

Step 4. Use E_A tables to find the equivalent device hours at a desired temperature for each activation energy (failure mechanism), or use the Arrhenius relation.

$$R = A \exp\left[\frac{-E_A}{KT}\right]$$

K = 8.617×10^{-5} eV/°K (Boltzman's Constant)
A = proportionality constant
R = mean rate to failure
E_A = activation energy
T = Temperature in Kelvin

$$\frac{R_1}{R_2} = \frac{A_1 \exp\left[\frac{-E_A}{KT_1}\right]}{A_2 \exp\left[\frac{-E_A}{KT_2}\right]} = \exp\left[\frac{E_A}{K}\left(\frac{1}{T_2} - \frac{1}{T_1}\right)\right]$$

Where $A_1 = A_2 = A$ for the same failure mechanism (i.e., same E_A)

Where R_1 and R_2 are rates for a normal operating temperature and an elevated temperature respectively

$$R_1 = R_2 \times \exp\left[\frac{E_A}{K}\left(\frac{1}{T_2} - \frac{1}{T_1}\right)\right]$$

However, since rate (R) has the units $\left(\frac{1}{\text{time}}\right)$, we can think in terms of time to one failure or MTBF.

Thus,

$$R_1 = \frac{1}{t_1} \quad \text{where } t_1 = \text{MTBF at same temperature } T_1$$

and

$$R_2 = \frac{1}{t_2} \quad \text{where } t_2 = \text{MTBF at same temperature } T_2$$

Thus the Arrhenius Relation becomes:

$$\frac{1}{t_1} = \frac{1}{t_2} \times \exp\left[\frac{E_A}{K}\left(\frac{1}{T_2} - \frac{1}{T_1}\right)\right]$$

or

$$t_1 = \exp\left[\frac{E_A}{K}\left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right] \times t_2$$

We then define the Acceleration Factor as:

$$\text{A.F.} = \frac{t_1}{t_2} = \exp\left[\frac{E_A}{K}\left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right]$$

For example: For $E_A = 0.6$ eV, $T_2 = 398$°K, $T_1 = 328$°K

$$t_1 = 41.7\ t_2$$

Therefore, one hour at 125°C is the equivalent to 41.7 hours at 55°C for a failure mechanism of activation energy $E_A = 0.6$ eV. Then 41.7 is the thermal acceleration factor for time.

3

NOTE:
The Arrhenius Plot is simply ln (Acceleration Factor) vs. 1/Temperature normalized for an MTBF (t_2) of one hour at 250°C (T_2). This plot can also be used to determine the acceleration factor between two temperatures (other than 250°C).

For example: For a 0.3 eV failure at 125°C, the acceleration factor is 8.1 relative to a 0.3 eV failure at 250°C. For a 0.3 eV failure at 25°C, the acceleration factor is 152 relative to 250°C. Therefore, the acceleration factor between 125°C and 25°C is:

$$\text{A.F.} = \frac{t_1}{t_2} = \frac{152}{8.1} = 18.7$$

Step 5. Organize the burn-in/lifetest data by E_A, Total Device Hours at the burn-in/lifetest temperature T_2, Thermal Acceleration Factors for each failure mechanism (E_A), Number of Failures for each failure mechanism, and the calculated equivalent device hours at the desired operating temperature T_1.

NOTE:
The rise in junction temperature due to the thermal resistivity of the package (θ_{JA}) must be added to the desired and actual burn-in/lifetest temperatures.

$$T_{test} = T_J + T_A = \theta_{JA} (IV @ T_A) + T_A$$

E_A (eV)	Total Device Hrs @ T_2	Acceleration Factors	# Fail	Equivalent Hours @ T_1
0.3	T.D.H.	X	N_1	X (T.D.H.)
0.6	T.D.H.	Y	N_2	Y (T.D.H.)
1.0	T.D.H.	Z	N_3	Z (T.D.H.)

The failure rates for individual failure mechanisms and the total combined failure rate can be predicted using the data table and the following formula:

$$\text{\% fail/1K hrs.} = \frac{\chi^2 (n, \alpha)}{2T} (10^5)$$

Where χ^2 (n, α) is the value of the chi squared distribution for n degrees of freedom and confidence level of α. T is the total equivalent device hours at T_1. The total combined rate is just the sum of the individual failure rates for each failure mechanism.

For a 60% UCL, the above formula converts to the following:

# Failures	% fail/1K hours 60% UCL
0	$0.915 \times 10^5/T$
1	$2.02 \times 10^5/T$
2	$3.105 \times 10^5/T$
3	$4.17 \times 10^5/T$
3 < # < 15	$\frac{1.049 (\text{\# failures for a particular } E_A) + 1.0305}{\text{Equivalent hours @ } T_1} \left[10^5\right]$
> 15	$\frac{(0.2533 + \sqrt{(4 \times \text{\# failed}) + 3})^2}{4T} \left[10^5\right]$

Example 1:

Assume for this example, that I_{CC} active is 57 mA at T_A = 125°C and I_{CC} active is 60 mA at T_A = 55°C.

Also assume that θ_{JA} = 35°C/W.

Then,

T_2 = (35°C/W) (57 mA) (5V) + 125°C

≈ 135°C = 408°K

T_1 = (35°C/W) (60 mA) (5V) + 55°C

≈ 65°C = 338°K

E_A (eV)	Actual Device Hours @ 125°C	Acceleration for 135°C to 65°C	Equivalent Hours at 55°C	# Fail	55°C % Fail/ 1K Hrs
0.3	2.185×10^6	5.85	1.278×10^7	0	0.0081
0.6	2.185×10^6	34.18	7.468×10^7	2	0.0042
1.0	2.185×10^6	359.93	7.864×10^8	1	0.0003
			Total Combined Failure Rate =		0.0126
			=		126 FITs

Example 2:

Assume that an additional lot of 800 flash devices is burned in using a 6.5V lifetest. Using Table 2 below, a voltage acceleration factor of 55 results from a 20% overstress (5.5V to 6.5V).

	48 Hours	168 Hours	500 Hours
Lot #3	0/800	1/800	0/799

Device Hours = 800 (48 hrs − 0 hrs) + 800 (168 hrs − 48 hrs) + 799 (500 hrs − 168 hrs)

= 3.997×10^5

Table 2. Time-Dependent Oxide Failure Accelerations

Type	Supply Voltage (Volts)	Oxide Thickness (Å)	Operating Stress (MV/cm)	Acceleration Factor at __% Over Stress			
				10%	20%	50%	100%
HMOS* E	5	700	0.714	3.2	10	320	99,500
HMOS* IIE	5	400	1.25	7.5	55	23,700	5.6×10^8
ETOX™II	5	235	2.15	3.9	26	910	8.3×10^5

ASSUMES:
1. No Bias Generators
2. Depletion Loads

Since this voltage accelerated stress is used to predict an oxide breakdown failure rate, the 5.5V burn-in/lifetest 55°C hours for E_A = 0.3 eV are added to the 6.5V burn-in/lifetest 55°C equivalent hours as follows:

125°C Burn-In/Lifetest	E_A (eV)	Actual Device Hours @ 125°C	Acceleration Factors for 135°C to 65°C	Equivalent Hours @ 55°C
5.5V	0.3	2.185×10^6	5.85	1.278×10^7
6.5V	0.3	3.997×10^5	(5.85 × 55)	1.286×10^8
		Total Equivalent E_A = 0.3 eV Device Hours =		1.414×10^8

The following failure rate predictions include the total equivalent 55°C, E_A = 0.3 eV device hours found above:

E_A (eV)	Actual Device Hours @ 125°C	Acceleration Factors for 135°C to 65°C	Equivalent Hours @ 55°C	# Fail	55°C % Fail/ 1K Hrs
0.3	2.185×10^6	5.85	—	—	—
0.3 + 55(1)	3.997×10^5	(5.85 × 55)	1.414×10^8	1	0.0015
0.6	2.185×10^6	34.18	7.468×10^7	2	0.0042
1.0	2.185×10^6	359.93	7.864×10^8	1	0.0003
			Total Combined Failure Rate =		0.0060
			=		60 FITs

NOTE:
1. The notation 0.3 + 55 is used to show that 6.5V and 5.5V burn-in lifetest equivalent hours have been combined.

APPENDIX B
FLASH MEMORY BIT MAPS AND DIE PHOTOGRAPHS

Columns are numbered 0 through 255 beginning with the column nearest the X-decoder.

Outputs are grouped as follows:

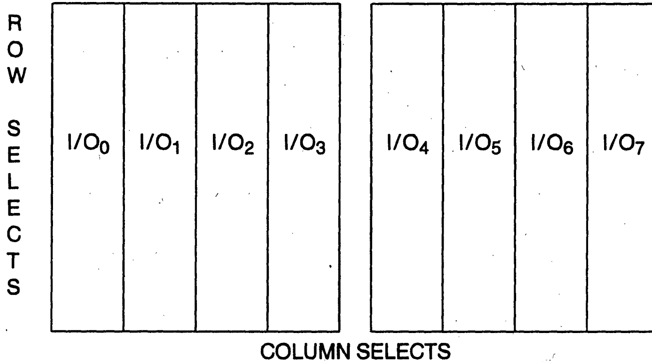
Quadrant Decoding		Column Decoding				LEFT HALF ARRAY O0 O1 O2 O3		RIGHT HALF ARRAY O4, O5 O6 O7	
A4	A3	A2	A1	A0	A10	O0/O7	O1/O6	O2/O5	O3/O4
0	0	0	0	0	0	BL192	BL128	BL64	BL0
0	1	0	0	0	0	BL193	BL129	BL65	BL1
0	1	0	0	0	1	BL194	BL130	BL66	BL2
0	0	0	0	0	1	BL195	BL131	BL67	BL3
0	0	0	0	1	0	BL196	BL132	BL68	BL4
0	1	0	0	1	0	BL197	BL133	BL69	BL5
0	1	0	0	1	1	BL198	BL134	BL70	BL6
0	0	0	0	1	1	BL199	BL135	BL71	BL7
*	.	.	.	.	.	.	.	.	.
*	.	.	.	.	.	.	.	.	.
*	.	.	.	.	.	.	.	.	.
1	0	1	1	0	0	BL248	BL184	BL120	BL56
1	1	1	1	0	0	BL249	BL185	BL121	BL57
1	1	1	1	0	1	BL250	BL186	BL122	BL58
1	0	1	1	0	1	BL251	BL187	BL123	BL59
1	0	1	1	1	0	BL252	BL188	BL124	BL60
1	1	1	1	1	0	BL253	BL189	BL125	BL61
1	1	1	1	1	1	BL254	BL190	BL126	BL62
1	0	1	1	1	1	BL255	BL191	BL127	BL63

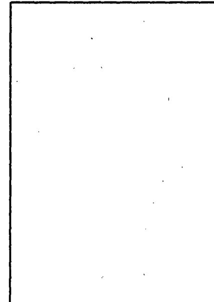
28F256A Bitline Decoding

X-DECODING: Wordlines are numbered 0 through 511 beginning at the top of the array

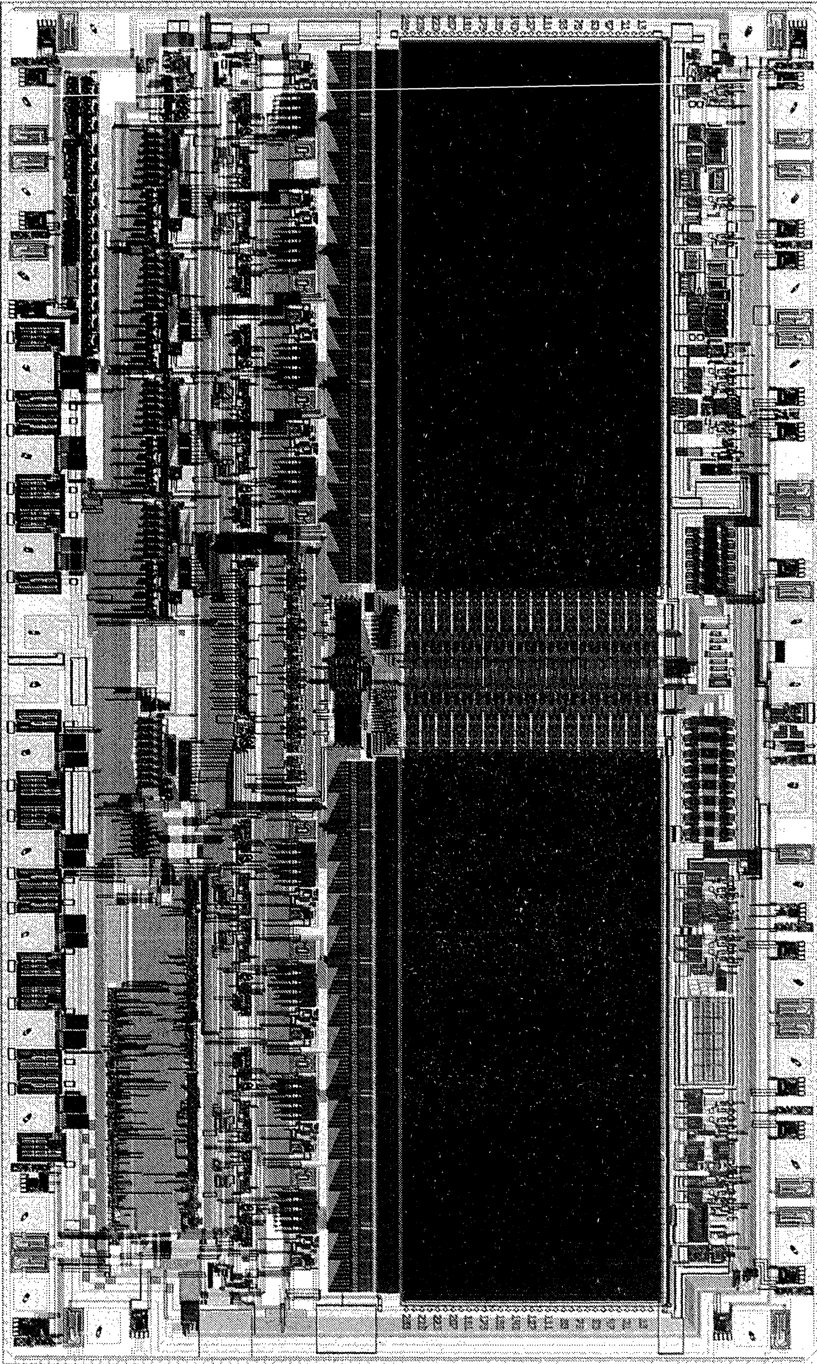
WL	A14	A13	A12	A7	A6	A5	A11	A9	A8
WL0	0	0	0	0	0	0	0	0	0
WL1	0	0	0	0	0	0	0	0	1
WL2	0	0	0	0	0	0	0	1	0
WL3	0	0	0	0	0	0	0	1	1
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
WL508	1	1	1	1	1	1	1	0	0
WL509	1	1	1	1	1	1	1	0	1
WL510	1	1	1	1	1	1	1	1	0
WL511	1	1	1	1	1	1	1	1	1

28F256A Wordline Decoding





28F256A Bit Map



293002-16

28F256A Die Photograph

28F512 (C) Bitline Decoding

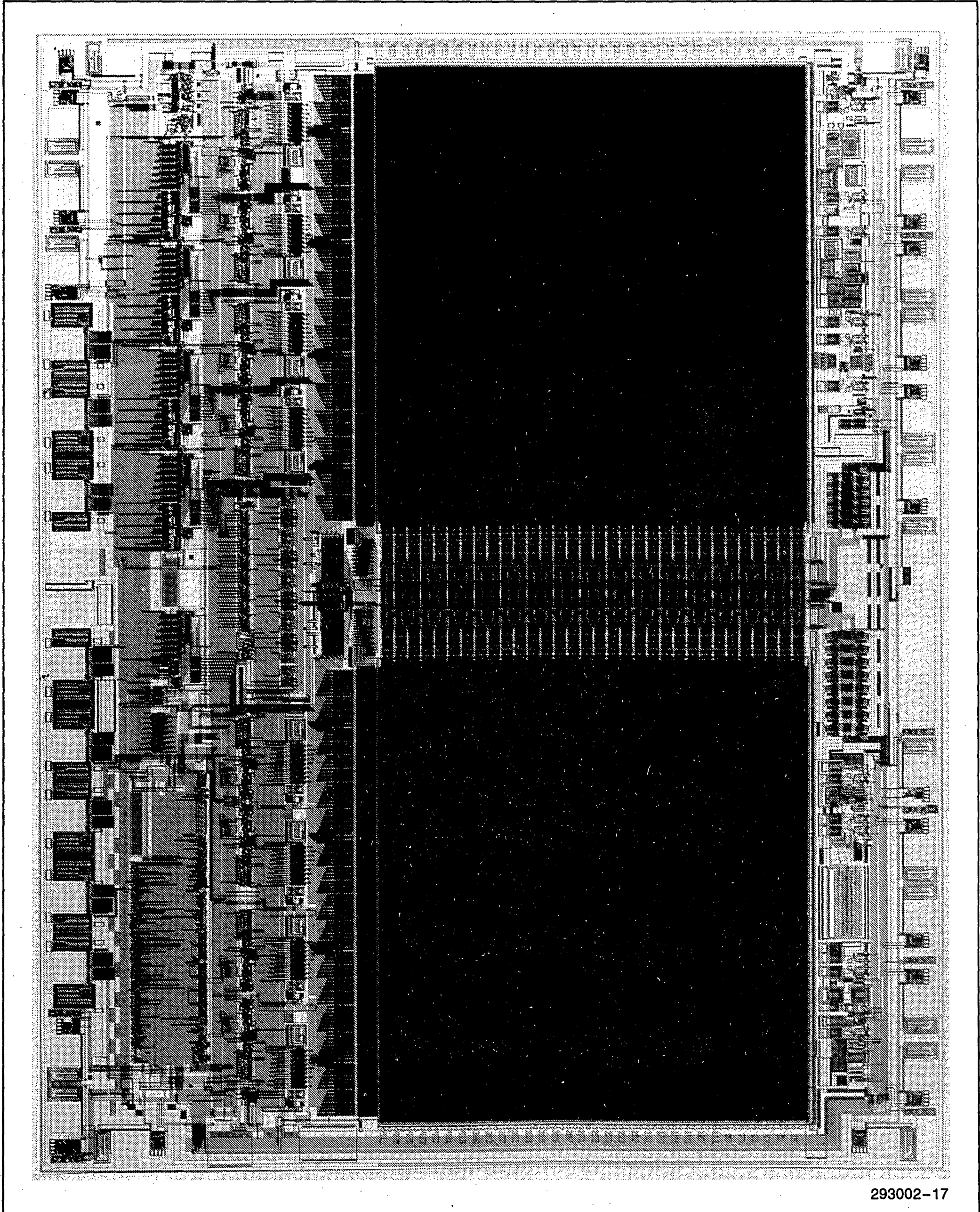
Address							Bitlines			
A14	A15	A10	A2	A1	A0	A3	IO0/7	IO1/6	IO2/5	IO3/4
0	0	0	0	0	0	0	BL384	BL256	BL128	BL0
0	0	0	0	0	0	1	BL385	BL257	BL129	BL1
0	0	0	0	0	1	0	BL386	BL258	BL130	BL2
0	0	0	0	0	1	1	BL387	BL259	BL131	BL3
0	0	0	0	1	0	0	BL388	BL260	BL132	BL4
0	0	0	0	1	0	1	BL389	BL261	BL133	BL5
0	0	0	0	1	1	0	BL390	BL262	BL134	BL6
0	0	0	0	1	1	1	BL391	BL263	BL135	BL7
...	...	...	...	...	...	...	...	...	...	...
1	1	1	1	1	0	0	BL508	BL380	BL252	BL124
1	1	1	1	1	0	1	BL509	BL381	BL253	BL125
1	1	1	1	1	1	0	BL510	BL382	BL254	BL126
1	1	1	1	1	1	1	BL511	BL383	BL255	BL127

28F512 (C) Wordline Decoding

X Address									Row
A12	A7	A6	A5	A4	A13	A11	A9	A8	WL
0	0	0	0	0	0	0	0	0	XL0
0	0	0	0	0	0	0	0	1	XL1
0	0	0	0	0	0	0	1	0	XL2
0	0	0	0	0	0	0	1	1	XL3
0	0	0	0	0	0	1	0	0	XL4
0	0	0	0	0	0	1	0	1	XL5
0	0	0	0	0	0	1	1	0	XL6
0	0	0	0	0	0	1	1	1	XL7
0	0	0	0	0	1	0	0	0	XL8
0	0	0	0	0	1	0	0	1	XL9
0	0	0	0	0	1	0	1	0	XL10
0	0	0	0	0	1	0	1	1	XL11
0	0	0	0	0	1	1	0	0	XL12
0	0	0	0	0	1	1	0	1	XL13
0	0	0	0	0	1	1	1	0	XL14
0	0	0	0	0	1	1	1	1	XL15
0	0	0	0	1	1	1	1	1	XL16
0	0	0	0	1	1	1	1	0	XL17
0	0	0	0	1	1	1	0	1	XL18
0	0	0	0	1	1	1	0	0	XL19
0	0	0	0	1	1	0	1	1	XL20
0	0	0	0	1	1	0	1	0	XL21
0	0	0	0	1	1	0	0	1	XL22
0	0	0	0	1	1	0	0	0	XL23
0	0	0	0	1	0	1	1	1	XL24
0	0	0	0	1	0	1	1	0	XL25
0	0	0	0	1	0	1	0	1	XL26
0	0	0	0	1	0	1	0	0	XL27
0	0	0	0	1	0	0	1	1	XL28
0	0	0	0	1	0	0	1	0	XL29
0	0	0	0	1	0	0	0	1	XL30
0	0	0	0	1	0	0	0	0	XL31

28F512 (C) Wordline Decoding (Continued)

X Address									Row
A12	**A7**	**A6**	**A5**	**A4**	**A13**	**A11**	**A9**	**A8**	**WL**
0	0	0	1	0	0	0	0	0	XL32
.	.	.	.	.	.	.	.	.	. . .
0	0	0	1	0	1	1	1	1	XL47
0	0	0	1	1	1	1	1	1	XL48
.	.	.	.	.	.	.	.	.	. . .
0	0	0	1	1	0	0	0	0	XL63
0	0	1	0	0	0	0	0	0	XL64
.	.	.	.	.	.	.	.	.	. . .
0	0	1	0	0	1	1	1	1	XL79
0	0	1	0	1	1	1	1	1	XL80
.	.	.	.	.	.	.	.	.	. . .
0	0	1	0	1	0	0	0	0	XL95
1	1	1	1	0	0	0	0	0	XL480
.	.	.	.	.	.	.	.	.	. . .
1	1	1	1	0	1	1	1	1	XL495
1	1	1	1	1	1	1	1	1	XL496
.	.	.	.	.	.	.	.	.	. . .
1	1	1	1	1	0	0	0	0	XL511



28F512 Die Photo

28F010 (A, B, C) Bitline Decoding

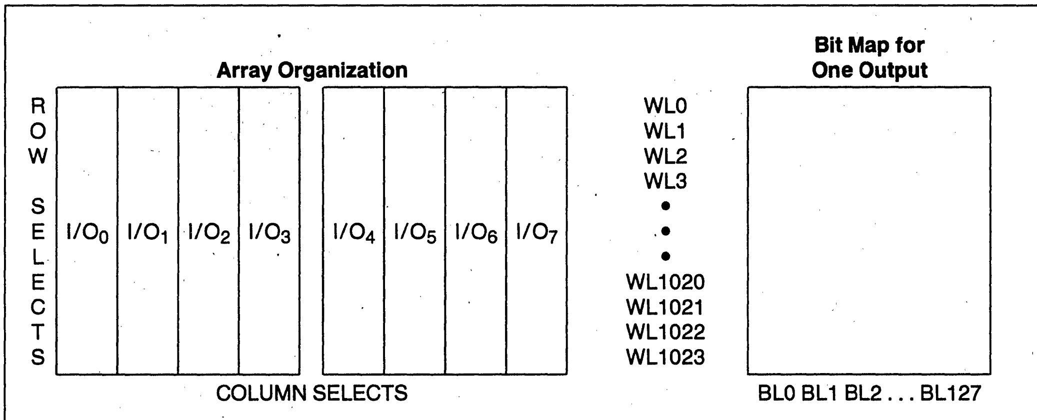
Address							Bitlines			
A16	A15	A10	A2	A1	A0	A3	IO0/7	IO1/6	IO2/5	IO3/4
0	0	0	0	0	0	0	BL384	BL256	BL128	BL0
0	0	0	0	0	0	1	BL385	BL257	BL129	BL1
0	0	0	0	0	1	0	BL386	BL258	BL130	BL2
0	0	0	0	0	1	1	BL387	BL259	BL131	BL3
0	0	0	0	1	0	0	BL388	BL260	BL132	BL4
0	0	0	0	1	0	1	BL389	BL261	BL133	BL5
0	0	0	0	1	1	0	BL390	BL262	BL134	BL6
0	0	0	0	1	1	1	BL391	BL263	BL135	BL7
...	...	...	...	...	...	...	...	...	...	...
1	1	1	1	1	0	0	BL508	BL380	BL252	BL124
1	1	1	1	1	0	1	BL509	BL381	BL253	BL125
1	1	1	1	1	1	0	BL510	BL382	BL254	BL126
1	1	1	1	1	1	1	BL511	BL383	BL255	BL127

28F010 (A, B, C) Wordline Decoding

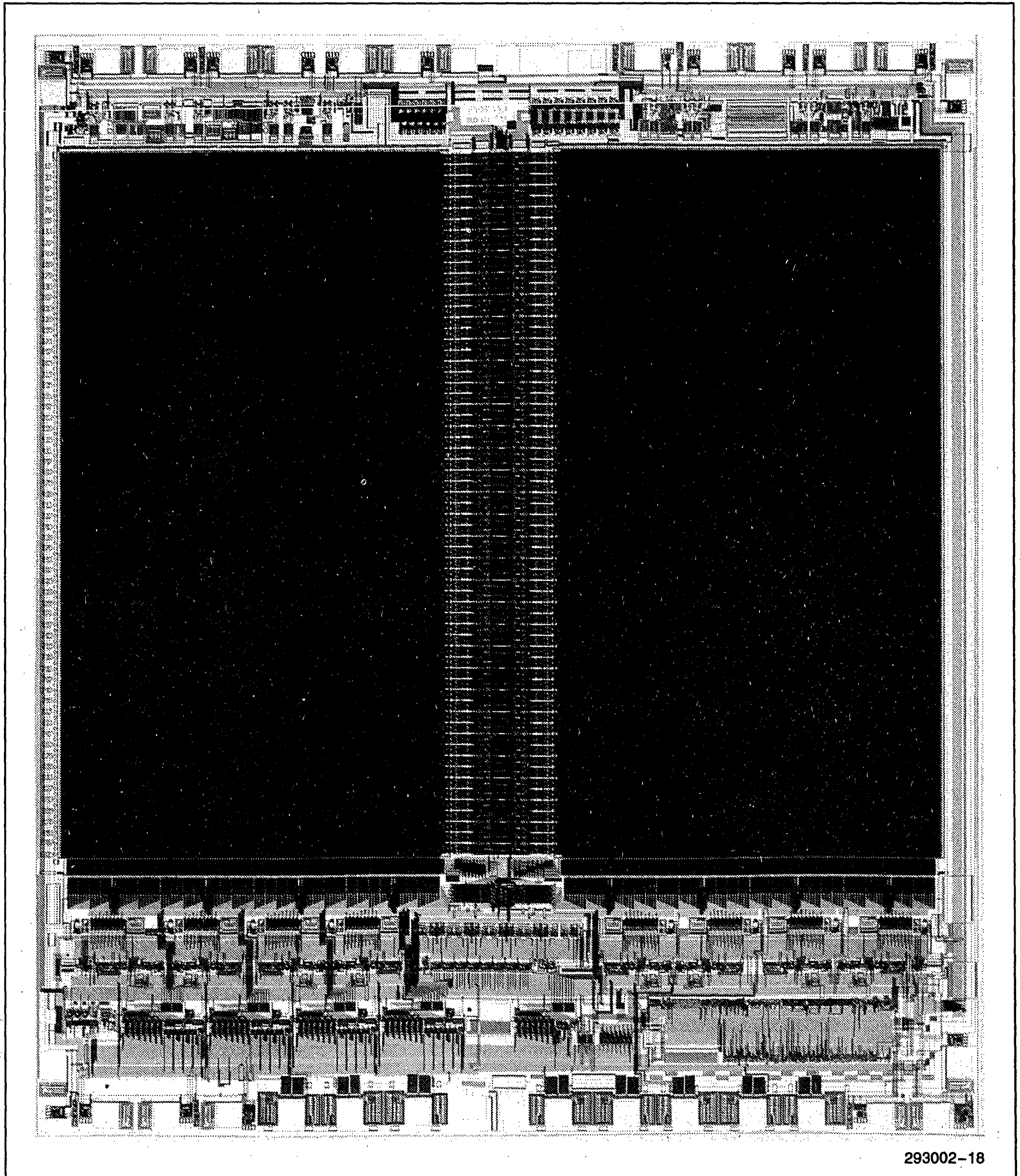
X Address										Row
A14	A12	A7	A6	A5	A4	A13	A11	A9	A8	WL
0	0	0	0	0	0	0	0	0	0	XL0
0	0	0	0	0	0	0	0	0	1	XL1
0	0	0	0	0	0	0	0	1	0	XL2
0	0	0	0	0	0	0	0	1	1	XL3
0	0	0	0	0	0	0	1	0	0	XL4
0	0	0	0	0	0	0	1	0	1	XL5
0	0	0	0	0	0	0	1	1	0	XL6
0	0	0	0	0	0	0	1	1	1	XL7
0	0	0	0	0	0	1	0	0	0	XL8
0	0	0	0	0	0	1	0	0	1	XL9
0	0	0	0	0	0	1	0	1	0	XL10
0	0	0	0	0	0	1	0	1	1	XL11
0	0	0	0	0	0	1	1	0	0	XL12
0	0	0	0	0	0	1	1	0	1	XL13
0	0	0	0	0	0	1	1	1	0	XL14
0	0	0	0	0	0	1	1	1	1	XL15
0	0	0	0	0	1	1	1	1	1	XL16
0	0	0	0	0	1	1	1	1	0	XL17
0	0	0	0	0	1	1	1	0	1	XL18
0	0	0	0	0	1	1	1	0	0	XL19
0	0	0	0	0	1	1	0	1	1	XL20
0	0	0	0	0	1	1	0	1	0	XL21
0	0	0	0	0	1	1	0	0	1	XL22
0	0	0	0	0	1	1	0	0	0	XL23
0	0	0	0	0	1	0	1	1	1	XL24
0	0	0	0	0	1	0	1	1	0	XL25
0	0	0	0	0	1	0	1	0	1	XL26
0	0	0	0	0	1	0	1	0	0	XL27
0	0	0	0	0	1	0	0	1	1	XL28
0	0	0	0	0	1	0	0	1	0	XL29
0	0	0	0	0	1	0	0	0	1	XL30
0	0	0	0	0	1	0	0	0	0	XL31

28F010 (A, B, C) Wordline Decoding (Continued)

X Address										Row
A14	**A12**	**A7**	**A6**	**A5**	**A4**	**A13**	**A11**	**A9**	**A8**	**WL**
0	0	0	0	1	0	0	0	0	0	XL32
.	.	.	.	.	.	.	.	.	.	...
0	0	0	0	1	0	1	1	1	1	XL47
0	0	0	0	1	1	1	1	1	1	XL48
.	.	.	.	.	.	.	.	.	.	...
0	0	0	0	1	1	0	0	0	0	XL63
0	0	0	1	0	0	0	0	0	0	XL64
.	.	.	.	.	.	.	.	.	.	...
0	0	0	1	0	0	1	1	1	1	XL79
0	0	0	1	0	1	1	1	1	1	XL80
.	.	.	.	.	.	.	.	.	.	...
0	0	0	1	0	1	0	0	0	0	XL95
1	1	1	1	1	0	0	0	0	0	XL992
.	.	.	.	.	.	.	.	.	.	...
1	1	1	1	1	0	1	1	1	1	XL1007
1	1	1	1	1	1	1	1	1	1	XL1008
.	.	.	.	.	.	.	.	.	.	...
1	1	1	1	1	1	0	0	0	0	XL1023



28F010 Bit Map



293002-18

28F010 Die Photo

28F020 Bitline Decoding

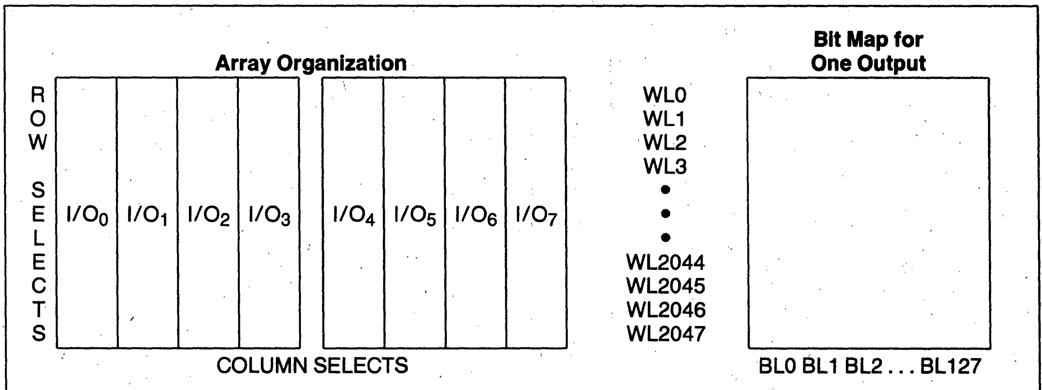
Address							Bitlines			
A16	A15	A10	A2	A1	A0	A3	IO0/7	IO1/6	IO2/5	IO3/4
0	0	0	0	0	0	0	BL384	BL256	BL128	BL0
0	0	0	0	0	0	1	BL385	BL257	BL129	BL1
0	0	0	0	0	1	0	BL386	BL258	BL130	BL2
0	0	0	0	0	1	1	BL387	BL259	BL131	BL3
0	0	0	0	1	0	0	BL388	BL260	BL132	BL4
0	0	0	0	1	0	1	BL389	BL261	BL133	BL5
0	0	0	0	1	1	0	BL390	BL262	BL134	BL6
0	0	0	0	1	1	1	BL391	BL263	BL135	BL7
...	...	...	...	...	...	...	...	...	...	...
1	1	1	1	1	0	0	BL508	BL380	BL252	BL124
1	1	1	1	1	0	1	BL509	BL381	BL253	BL125
1	1	1	1	1	1	0	BL510	BL382	BL254	BL126
1	1	1	1	1	1	1	BL511	BL383	BL255	BL127

28F020 Wordline Decoding

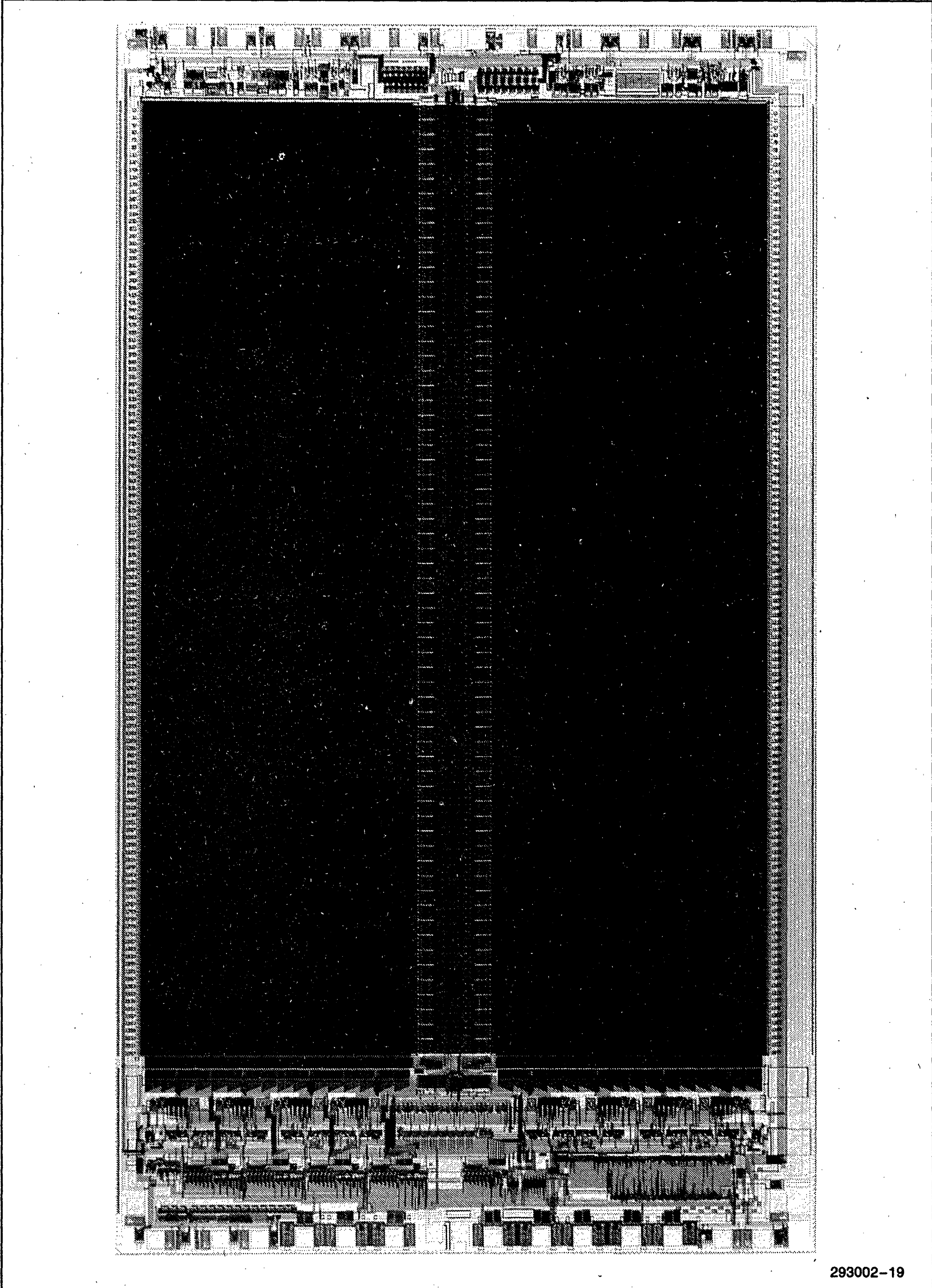
X Address											Row
A17	A14	A12	A7	A6	A5	A4	A13	A11	A9	A8	WL
0	0	0	0	0	0	0	0	0	0	0	XL0
0	0	0	0	0	0	0	0	0	0	1	XL1
0	0	0	0	0	0	0	0	0	1	0	XL2
0	0	0	0	0	0	0	0	0	1	1	XL3
0	0	0	0	0	0	0	0	1	0	0	XL4
0	0	0	0	0	0	0	0	1	0	1	XL5
0	0	0	0	0	0	0	0	1	1	0	XL6
0	0	0	0	0	0	0	0	1	1	1	XL7
0	0	0	0	0	0	0	1	0	0	0	XL8
0	0	0	0	0	0	0	1	0	0	1	XL9
0	0	0	0	0	0	0	1	0	1	0	XL10
0	0	0	0	0	0	0	1	0	1	1	XL11
0	0	0	0	0	0	0	1	1	0	0	XL12
0	0	0	0	0	0	0	1	1	0	1	XL13
0	0	0	0	0	0	0	1	1	1	0	XL14
0	0	0	0	0	0	0	1	1	1	1	XL15
0	0	0	0	0	0	1	1	1	1	1	XL16
0	0	0	0	0	0	1	1	1	1	0	XL17
0	0	0	0	0	0	1	1	1	0	1	XL18
0	0	0	0	0	0	1	1	1	0	0	XL19
0	0	0	0	0	0	1	1	0	1	1	XL20
0	0	0	0	0	0	1	1	0	1	0	XL21
0	0	0	0	0	0	1	1	0	0	1	XL22
0	0	0	0	0	0	1	1	0	0	0	XL23
0	0	0	0	0	0	1	0	1	1	1	XL24
0	0	0	0	0	0	1	0	1	1	0	XL25
0	0	0	0	0	0	1	0	1	0	1	XL26
0	0	0	0	0	0	1	0	1	0	0	XL27
0	0	0	0	0	0	1	0	0	1	1	XL28
0	0	0	0	0	0	1	0	0	1	0	XL29
0	0	0	0	0	0	1	0	0	0	1	XL30
0	0	0	0	0	0	1	0	0	0	0	XL31

28F020 Wordline Decoding (Continued)

X Address											Row
A17	**A14**	**A12**	**A7**	**A6**	**A5**	**A4**	**A13**	**A11**	**A9**	**A8**	**WL**
0	0	0	0	0	1	0	0	0	0	0	XL32
.	.	.	.	.	.	.	.	.	.	.	. . .
0	0	0	0	0	1	0	1	1	1	1	XL47
0	0	0	0	0	1	1	1	1	1	1	XL48
.	.	.	.	.	.	.	.	.	.	.	. . .
0	0	0	0	0	1	1	0	0	0	0	XL63
0	0	0	0	1	0	0	0	0	0	0	XL64
.	.	.	.	.	.	.	.	.	.	.	. . .
0	0	0	0	1	0	0	1	1	1	1	XL79
0	0	0	0	1	0	1	1	1	1	1	XL80
.	.	.	.	.	.	.	.	.	.	.	. . .
0	0	0	0	1	0	1	0	0	0	0	XL95
0	1	1	1	1	1	0	0	0	0	0	XL992
.	.	.	.	.	.	.	.	.	.	.	. . .
0	1	1	1	1	1	0	1	1	1	1	XL1007
0	1	1	1	1	1	1	1	1	1	1	XL1008
.	.	.	.	.	.	.	.	.	.	.	. . .
0	1	1	1	1	1	1	0	0	0	0	XL1023
1	0	0	0	0	1	0	0	0	0	0	XL1056
.	.	.	.	.	.	.	.	.	.	.	. . .
1	0	0	0	0	1	0	1	1	1	1	XL1071
1	0	0	0	0	1	1	1	1	1	1	XL1072
.	.	.	.	.	.	.	.	.	.	.	. . .
1	0	0	0	0	1	1	0	0	0	0	XL1087
1	0	0	0	1	0	0	0	0	0	0	XL1088
.	.	.	.	.	.	.	.	.	.	.	. . .
1	0	0	0	1	0	0	1	1	1	1	XL1103
1	0	0	0	1	0	1	1	1	1	1	XL1104
.	.	.	.	.	.	.	.	.	.	.	. . .
1	0	0	0	1	0	1	0	0	0	0	XL1119
1	1	1	1	1	1	0	0	0	0	0	XL2016
.	.	.	.	.	.	.	.	.	.	.	. . .
1	1	1	1	1	1	0	1	1	1	1	XL2031
1	1	1	1	1	1	1	1	1	1	1	XL2032
.	.	.	.	.	.	.	.	.	.	.	. . .
1	1	1	1	1	1	1	0	0	0	0	XL2047



28F020 Bit Map



293002-19

28F020 Die Photo

RELIABILITY REPORT

RR-69

November 1992

28F008SA Reliability Summary

Order Number: 293008-002

28F008SA Reliability Summary

CONTENTS — PAGE

INTRODUCTION 3-672

QUALITY/RELIABILITY VERIFICATION 3-672

SUMMARY OF RESULTS 3-672

10 KBYTE-WRITE/BLOCK-ERASE CYCLING 3-673

QUALITY VALIDATION SUMMARY ... 3-673

INFANT MORTALITY EVALUATION SUMMARY 3-674

HIGH TEMPERATURE OPERATING LIFETEST 3-674

ROOM TEMPERATURE HIGH VOLTAGE OPERATING LIFETEST 3-675

HIGH TEMPERATURE STORAGE (BAKE) 3-675

STEAM (AUTOCLAVE) 3-675

TEMPERATURE CYCLE 3-676

THERMAL SHOCK 3-676

ELECTROSTATIC DISCHARGE 3-676

LATCH-UP 3-677

DATA REPORTING FORMAT 3-677

3

INTRODUCTION

The 28F008SA is manufactured at D2 in Santa Clara, California on ETOX™ III process 620, and assembled at Mitsui, Japan. The 28F008SA is packaged in 40-lead TSOP die up (E) and die down (F) packages and 44-lead PSOP. The typical I_{CC} is 10 mA at a nominal V_{CC} of 5.0V and an ambient temperature of +70°C. All readouts in this qualification were done on a Genesis II production tester.

This report contains the preliminary qualification data for the 28F008SA and is intended to give a status of the certification.

QUALITY/RELIABILITY VERIFICATION

This report is designed to provide a detailed description of the methods used to verify that the 28F008SA meets or exceeds Intel's stringent quality and reliability requirements. Each test is described and the results are presented. The reliability of a device is generally defined as the probability that the device will perform its intended function under the specified operating conditions throughout its life. To determine a device's reliability, Intel subjects sample lots of the device to a variety of stress tests.

The evaluations performed for the TSOP (E/F28F008SA) included the following:

- 10K Byte-Write/Block-Erase Cycling(1) Quality Validation
- Infant Mortality Evaluation (IME)
- High Voltage Temperature Operating Lifetest (HVTOL)
- Room Temperature High Voltage Operating Lifetest (RTHVOL)
- High Temperature Storage (Bake)
- Steam (autoclave)
- Temperature Cycle (T/C)
- Thermal Shock (T/S)
- Electrostatic Discharge (ESD)
- Latch-Up

NOTE:
1. Data collection in progress for 100K cycle specification.

Stress lots containing die up material are designated with an "E", lots with die down material with an "F". Cycled devices are denoted with a "C", the uncycled units with a "U".

All of the stress tests were conducted on units that completed the full manufacturing flow. In all cases, a failure is defined as failure to meet any data sheet parameter. Stress test readouts for electrical test endpoints are done on the same equipment used for electrical test in commercial product manufacturing. Stresses with endpoints other than electrical test have explanations of the endpoints in the stress descriptions.

SUMMARY OF RESULTS

TEST/ STRESS	RESULTS	RESULT	DATA SET
1. 10K Write/Erase Cycling:(1)			
	0/1099	0 DPM	E28F008SA
2. Quality Validation:			
	2/4424	452 DPM	E28F008SA
			F28F008SA
3. Infant Mortality Evaluation:			
24 hrs, 6.5V	0/2265	0 DPM	E28F008SA
after 24 hr PBI*			F28F008SA
4. High Temperature Operating Lifetest:			
168 hrs	1/2247	445 DPM	E28F008SA
			F28F008SA
5. Room Temperature High Voltage Operating Lifetest:			
168 hrs	0/143		E28F008SA
6. High Temperature Storage (140°C Bake):			
1000 hrs.	4/438	0.91%	E28F008SA
7. Steam (autoclave):			
168 hrs	0/806	0.00%	F28F008SA
8. Temperature Cycle (T/C):			
1000 Cycles	0/248	0.00%	E28F008SA
Condition "C"			F28F008SA
9. Thermal Shock (T/S):			
200 Cycles	0/249	0.00%	E28F008SA
Condition "C"			F28F008SA
10. Electrostatic Discharge ESD:			
No Failures			E28F008SA
			F28F0085A
11. Latch-Up:			
No Failures			E28F008SA
			F28F008SA

*PBI = Production Burn In

NOTE:
1. Data Collection in progress for 100K cycle specification.

10 KBYTE-WRITE/BLOCK-ERASE CYCLING(1)

This test checks for any writing or erasure problems over 10,000 write/erase cycles at ambient temperatures of both +70°C and 0°C. A sample of cycled units are then submitted for further testing in reliability stresses.

Table 1a

10,000 W/E Cycling +70°C	
Lot #	**10K Cycles**
1EC	0/165
2EC	0/172
3EC	0/ 38
4EC	0/172
Totals	0/547

Table 1b

10,000 W/E Cycling 0°C	
Lot #	**10K Cycles**
1EC	0/172
2EC	0/169
3EC	0/ 39
4EC	0/172
Totals	0/552

NOTE:
1. Data Collection in progress from 100K cycle specification.

QUALITY VALIDATION SUMMARY

Quality Validation is done after the production flow to verify the Production Test/Inspection Test guardbands, temperature range and test repeatability. This evaluation is performed on units which have seen standard production testing. Subsequent testing is done utilizing a special test tape which exercises the part at the worst case voltage and timing conditions. This testing is done at three temperatures to guarantee full functionality at all datasheet conditions. This evaluation best represents a customer's worst case incoming EDPM for the product.

Table 2

Lot	Quality Validation		
	80°C	**Room**	**0°C**
1EC	1/ 337 B	0/ 336	0/ 336
1EU	0/ 77	0/ 77	0/ 77
2EC	1/ 340 A	0/ 339	0/ 339
2EU	0/ 472	0/ 472	0/ 472
3EC	0/ 77	0/ 77	0/ 77
4EC	0/ 344	0/ 344	0/ 344
4EU	0/1006	0/1006	0/1006
5EU	0/ 724	0/ 724	0/ 724
6EU	0/ 587	0/ 587	0/ 587
1FU	0/ 230	0/ 230	0/ 230
2FU	0/ 230	0/ 230	0/ 230
Totals	2/4423	0/4422	0/4422

Failure Analysis Summary:

ID	Qty	Description
A	1	V_{CC} min failure
B	1	Isb CMOS failure. Leakage = 152 μA, Spec = 100 μA

INFANT MORTALITY EVALUATION SUMMARY

Infant mortality evaluation (IME) data is used to predict the product early life failure rate. The data is also used to determine the required burn-in time. Burn-in is used in production as needed to ensure that the early life failure rate goals are met.

The units are functionally exercised at 6.5V at an elevated ambient temperature of 125°C for 48 hours. During the test the memory is sequentially addressed and the outputs are exercised but not monitored or loaded. A checkerboard data pattern is used to simulate random patterns expected during actual use. Then all units used for infant mortality evaluation are subjected to standard Intel testing.

Table 3. Infant Mortality Evaluation (V_{CC} = 6.5V, T = 125°C)

Lot #	12 Hrs	18 Hrs	24 Hrs	48 Hrs
1EU	0/ 63	0/ 63	0/ 63	0/ 63
1EC	1/ 215 A	0/ 211	0/ 211 E	0/ 210
2EU	0/ 193	0/ 193	0/ 193 F	0/ 191
2EC	0/ 216 B	0/ 215	0/ 215	0/ 215
3EC	0/ 77	0/ 77	0/ 77 G	0/ 76
4EU	0/ 481	1/ 481 C	0/ 480	0/ 480
4EC	0/ 217	0/ 217	0/ 217	0/ 217
5EU	0/ 208	0/ 206 D	0/ 206	0/ 206
6EU	0/ 497	0/ 497	0/ 497	0/ 497
1FU	0/ 55	0/ 55	0/ 55	0/ 55
2FU	0/ 55	0/ 55	0/ 55	0/ 55
Totals	1/2277	1/2270	0/2269	0/2265

Invalid Test Fallout Summary:

ID	Qty	Description
A	3	EOS due to mis-socketing
B	1	EOS due to mis-socketing
D	2	Lost in handler
E	1	Passed retest
F	2	EOS due to mis-socketing
G	1	EOS due to mis-socketing

Failure Analysis Summary:

ID	Qty	Description
A	1	Isb > 100 μA spec limit.
C	1	Fails opens/shorts testing

HIGH TEMPERATURE OPERATING LIFETEST

High temperature lifetest is a dynamic life test performed to accelerate failure mechanisms in the infant mortality (early life) and random failure/wear-out portion of a product's life curve. This is done to predict field reliability. In this test, failure mechanisms are accelerated by functionally exercising the device at an elevated ambient temperature of +125°C and a V_{CC} of 6.5V. During the test the memory is sequentially addressed and the outputs arc exercised but not monitored or loaded. A checkerboard data pattern is used to simulate random patterns expected during actual use. Then all devices used for lifetesting are subjected to standard Intel testing.

Table 4. High Voltage Electrical Life Test (V_{CC} = 6.5V, T = 125°C, Dynamic)

Lot #	120 Hr	*500 Hr	1K Hr
1EU	0/ 63	0/ 63	0/ 63
1EC	0/ 210	0/ 55	0/ 55
2EU	0/ 191	0/ 110	0/ 108
2EC	0/ 200	0/ 55	0/ 55
3EC	0/ 76	0/ 76	1/ 76 B
4EU	0/ 479	0/ 110	0/ 110
4EC	1/ 217 A	0/ 55	0/ 55
5EU	0/ 206	0/ 205	0/ 204
6EU	0/ 497	0/ 250	0/ 250
1FU	0/ 55	0/ 55	0/ 55
2FU	0/ 53	0/ 53	0/ 53
Totals	1/2247	0/1087	1/1084

Failure Analysis Summary:

ID	Qty	Ae	Description
A	1	0.3	Isb over spec limit
B	1	0.6	Single bit change loss

NOTE:
*Sample size decreased due to capacity constraints.

ROOM TEMPERATURE HIGH VOLTAGE OPERATING LIFETEST

In this test, failure mechanisms are accelerated by functionally exercising the device at the ambient temperature (25°C) while maintaining a V_{CC} of 6.5V. During this test the memory is sequentially addressed and the outputs are exercised, but not monitored or loaded. A checkerboard data pattern is used to simulate random patterns expected during actual use. This data with the voltage acceleration factors and the standard dynamic lifetest data is used to calculate the failure rate.

Table 5. Electrical Life Test (V_{CC} = 6.5V, T = 25°C)

Lot #	48 Hr	168 Hr	500 Hr	1000 Hr
1EC	0/ 17	0/ 17	0/ 17	0/ 17
2EU	0/ 27	0/ 27	0/ 27	0/ 27
2EC	0/ 15	0/ 15	0/ 15	0/ 15
4EU	0/ 28	0/ 28	0/ 28	0/ 28
4EC	0/ 28	0/ 28	1/ 28 A	0/ 27
3EU	0/ 28	0/ 28	0/ 28	0/ 28
Totals	0/143	0/143	1/143	0/142

Failure Analysis Summary:

ID	Qty	Ae	Description
A	1	0.3	Power failure

HIGH TEMPERATURE STORAGE (BAKE)

A 140°C bake, with no applied voltage, is performed to determine the effect of high temperature storage without electrical bias. The bake evaluation accelerates failure mechanisms such as bond degradation and process wear-out mechanisms such as ionic contamination. This stress also checks contact integrity. The test is conducted as per the specification Mil-Std-883C method 1008.2.

Table 6. Data Retention Bake (Plastic, T = 140°C)

Lot #	72 Hr	168 Hr	500 Hr	1000 Hr
1EC	0/ 71	0/ 71	0/ 71	1/ 71 C
2EU	0/ 75	0/ 75	0/ 75	0/ 75
2EC	0/ 71	0/ 71	1/ 71 B	0/ 71
4EU	0/ 75	0/ 75	0/ 75	0/ 75
4EC	1/ 71 A	0/ 70	0/ 70	1/ 70 D
5EU	0/ 75	0/ 75	0/ 75	0/ 75
Totals	1/438	0/437	1/437	2/436

Failure Analysis Summary:

ID	Qty	Description
A	1	Single bit charge loss V_{CC} max = 5.5V
B	1	Single bit charge loss V_{CC} max = 5.0V
C	1	Single bit charge loss V_{CC} max = 4.9V
D	1	Single bit charge loss V_{CC} max = 5.3V

STEAM (AUTOCLAVE)

The Steam stress accelerates moisture penetration through the plastic packaging material to the surface of the die. The objective of the test is to accelerate the problems found in very moist environments. Failure mechanisms typically seen from this stress include corrosion, passivation defects, leakage and contamination. Passivation defects or marginalities can allow moisture penetration to a single FLASH cell causing oxide deterioration resulting in a charge loss failure. The test chamber is maintained at a temperature of 121°C and an absolute pressure of two atmospheres. The devices contain 98% + programmed pattern. After the stress the units are subjected to standard Intel testing.

Table 7. Steam

Lot #	96 Hr	168 Hr
4EU	0/ 80	0/ 73
5EU	0/ 81	0/ 76
6EU	0/ 83	0/ 76
1FU	0/ 66	0/ 66
2FU	0/ 74	0/ 74
7EU	0/397	0/396
8EU	0/ 45	0/ 45
Totals	0/826	0/806

TEMPERATURE CYCLE

Temperature cycling is performed to evaluate the mechanical integrity of the device when exposed to temperature extremes. Mechanical failure mechanisms such as package cracking, die cracking, thin film cracking, bond wire lifting and die attach problems are accelerated by this stress. Temperature cycling also checks for changes in electrical characteristics due to mechanical displacement or rupture of conductors and insulating materials. Other effects can include delamination of finishes and degradation of package integrity. In this stress, the devices are alternately exposed to −65°C to +150°C (condition C). The units must be transferred between temperatures within one minute and must reach the specified temperature within fifteen minutes. The units must be at that temperature for a minimum of ten minutes. Heating and cooling are done by convection. This test is conducted in conformance with specification Mil-Std-833C method 1010.5. The endpoint for this stress was 1000 cycles.

Table 8. Temperature Cycle (−65°C to +150°C)

Lot #	200 Cycle	500 Cycle	1K Cycle
4EU	0/ 80	0/ 80	0/ 80
5EU	0/ 80	0/ 79	0/ 79
1FU	0/ 44	0/ 44	0/ 44
2FU	0/ 45	0/ 45	0/ 45
Totals	0/248	0/248	0/248

THERMAL SHOCK

Thermal shock is a liquid to liquid stress used to evaluate device resistance to sudden extreme changes in temperature. Thermal shock can result in die, package or thin film cracking, and delamination. Bond wire lifting and die attach problems are also accelerated by this stress. Thermal shock uses the same temperature ranges as temperature cycling, −65°C to 150°C (condition C), but the units must be transferred between temperatures within ten seconds and are immersed in the specified temperature for five minutes. The test is conducted in conformance with specification Mil-Std-883C method 1011.5. The endpoint for this stress was 200 cycles.

Table 9. Thermal Shock (−65°C to +150°C)

Lot #	50 Cycle	200 Cycle	500 Cycle
4EU	0/ 80	0/ 80	0/ 80
5EU	0/ 80	0/ 80	0/ 80
1FU	0/ 45	0/ 45	0/ 45
2FU	0/ 44	0/ 44	0/ 44
Totals	0/249	0/249	0/249

ELECTROSTATIC DISCHARGE

Electrostatic discharge (ESD) testing is done to measure a device's sensitivity to damage caused by ESD due to mechanical or human handling.

Human handling is modeled in specification Mil-Std-883 method 3015 testing. This type of event occurs when a person transfers a charge from their body into a device.

Military specification testing is done with an automated STAG ESD tester. A 100 pF capacitor is discharged into the device through a 1.5 KΩ resistor. The units are stressed to both polarities and are zapped a minimum of three times for each polarity. The following pin combinations are used:

- Each pill with respect to electrically discrete groups of V_{CC} pins. All other pins are left floating.
- Each pin with respect to electrically discrete groups of GND pins. All other pins are left floating.
- All supply pins with respect to each electrically discrete supply. All other pins left floating.

A failure is defined as any failure to meet data sheet specifications. Electrostatic discharge is a function of die design. Package differences have secondary impact on ESD performance.

Table 10. ESD Military Test

Lot #	+2000V	−2000V
4EU	0/ 10	0/ 10
1FU	0/ 10	0/ 10
2FU	0/ 10	0/ 10
Totals	0/ 30	0/ 30

LATCH-UP

Latch-up is caused by parasitic transistor turn-on, creating a current path from power to ground. This condition is characterized by a sudden increase in supply current to the device. During Latch-up testing, the V_{CC} voltage is monitored for a sudden drop due to loading. The following latch-up tests were performed on this device at +125°C:

- V_{CC} Latch-Up
  In this test the V_{CC} voltage is increased up to 10V. A 50Ω series resistor is used to prevent excessive current flow in the case of latch-up.
- V_{PP} Latch-Up
  In this test the V_{PP} voltage is increased up to 14V.
- I/O Latch-Up
  In this test a high input voltage is applied to the I/O pins. This voltage is slowly increased up to +8V for the positive polarity, and down to −2V for the negative polarity.

Latch-up is a function of process and die design. Package differerces have secondary impact on latch-up performance.

Table 11. CMOS Latch-Up Test

Lot #	Input +/−	Output +/−	V_{PP} +	V_{CC} +
4EU	0/ 15	0/ 15	0/ 15	0/ 15
1FU	0/ 15	0/ 15	0/ 15	0/ 15
2FU	0/ 15	0/ 15	0/ 15	0/ 15
Totals	0/ 45	0/ 45	0/ 45	0/ 45

DATA REPORTING FORMAT

Rejects/Sample Size

Rejects = Valid failures

Sample size = Total units minus valid and invalid rejects

RULE #1

Invalid failures are not included in the sample size.

RULE #2

Only valid failures are reported as rejects.

EXAMPLE:

Raw data looks like the following:

0 Hr	48 Hr	168 Hr	500 Hr
100 units put on stress	1/100	2/99	1/97
	(A)	(B)	(C)

NOTES:

A. 1-Blown unit (Electrical Overstress) [INVALID]
B. 1-Tacc failure [VALID]
   1-V_{CC} open (reverse socketing) [INVALID]
C. 1-Unit dropped during board unload [INVALID]

Data reported:

48 Hr	168 Hr	500 Hr
0/99	1/98	0/96
(A)	(B)	(C)

A. 1-Blown unit Invalid due to electrical overstress
B. 1-Tacc failure
   1-V_{CC} open Invalid due to reverse socketing
C. 1-Unit dropped during board unload

intel®

A 90ns 100K Erase/ Program Cycle Megabit Flash Memory

by Virgil Niles Kynett, Jim Anderson, Greg Atwood, Pat Dix, Mick Fandrich, Owen Jungroth, Susan Kao, Jerry A. Kreifels, Stefan Lai, Ho-Chun Liou, Benedict Liu, Richard Lodenquai, Weh-Juei Lu, Roy Pavloff, Daniel Tang, J.C. Tzeng, George Tsau, Branislav Vajdic, Gautam Verma, Simon Wang, Steven Wells, Mark Winston, and Lisa Yang

ABSTRACT

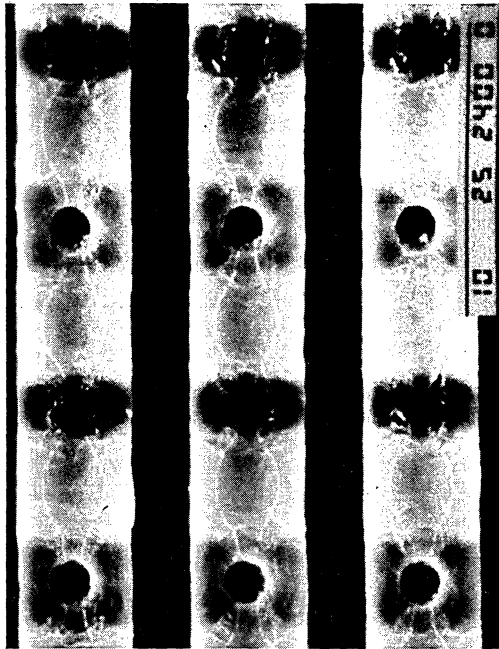
Using advanced 1.0μm CMOS technology, a 245 mil square 131072 × 8 device has been fabricated with a 3.8μm × 4.0μm cell. The memory exhibits a 90ns read access time with a 900ms electrical array erase and 10μs/byte program time. The device has been optimized for in-system microprocessor-controlled reprogramming with endurance performance greater than 100,000 erase/program cycles. Column redundancy is implemented with the utilization of flash memory cells to store repaired addresses.

ADVANCES in photolithography have made it possible to develop an electrically erasable reprogrammable 90ns 1Mb flash memory which is capable of greater than 100,000 erase/program cycles. This 1Mb memory implements a command port and an internal reference voltage generator, allowing microprocessor-controlled reprogramming [1].

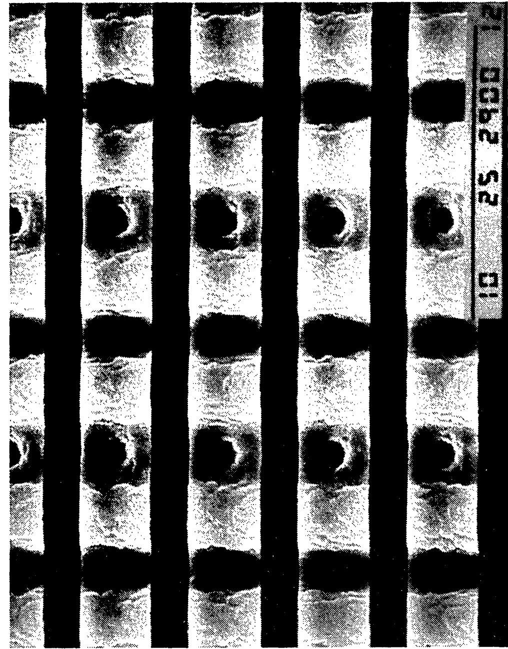
The 90ns access time results from a high memory cell current (95μA), low resistance poly-silicide wordlines, advanced scaled periphery transistors, and a di/dt optimized data-out buffer. Using CMOS inputs, power dissipation is 40mW in the active state and 20μW in the standby mode. The memory electrically erases in 900ms and programs at the rate of 10μs/byte. The device contains thirty-two columns of redundant elements and utilizes flash memory cells to store the address of repaired columns. The use of the flash memory cell reduces the required silicon area significantly over the commonly found large metal-shielded EPROM cells [2].

The 1Mb flash memory was fabricated on a 1.0μm double poly n-well CMOS process. Silicide was utilized on the wordlines to help achieve the 90ns access time performance. The CMOS periphery circuits were constructed with 0.9μm L_{eff}, 250 Å gate oxide LDD transistors. The density of this 1μm flash technology is demonstrated on the 1.0μm and 1.5μm memory cell comparison shown in Figure 1. The 1.0μm memory cell has a 15.2μm² area, which is over twice as small as the 1.5μm memory cell. A microphotograph of the 245 mil², 128K × 8 flash memory is shown in Figure 2. The process/device characteristics are summarized in Table 1.

© 1989 IEEE. Reprinted, with permission, from
1989 IEEE International Solid-State Circuits Conference Digest of Technical Papers,
New York, NY, February 15-17, 1989.



a) 1.5μ Lithography
(5,000 x magnification)



b) 1.0μ Lithography
(5,000 x magnification)

Figure 1. Array SEM microphotograph: (a) 1.5μm memory cell (6μ × 6μ) (b) 1.0μm memory cell (3.8μ × 4μ)

One of the most significant aspects of this device is its 100,000-cycle capability. A typical cell erase/program Vt margin is shown as a function of reprogramming cycles in Figure 3. After 100,000 cycles there still exists a 2.5V program read margin to insure reliable data retention. Accelerated retention bake experiments done at 250°C for 168 hours indicate that after 10,000 cycles the memory will exhibit only 0.7V program Vt shift. Program and erase time degrade slightly due to normal charge trap-up in the tunnel oxide (Figure 4). In addition, endurance reliability has been excellent with no tunnel oxide breakdown.

Table 1. Device Parameters

Technology	Cell	Periphery	Device
1.0-μm Lithography	Area = 3.8μm × 4μm	Tox = 250 Å	Die Size: 60116 mils²
1-Ploly, 1-Silicide	Gate Oxide > 100 Å	Leff N+P = 0.9μm	Organized: 128K × 8
N-Well CMOS	Read Current = 95μA	Xjn = 0.3μm	Access Time: 90ns
Epi on P+	Terase = 900ms	Xjp = 0.6μm	Active Power: 8mA
	Tprog = 10μs/byte		Standby Power: 4μA
			Package: 32-pin Cerdip

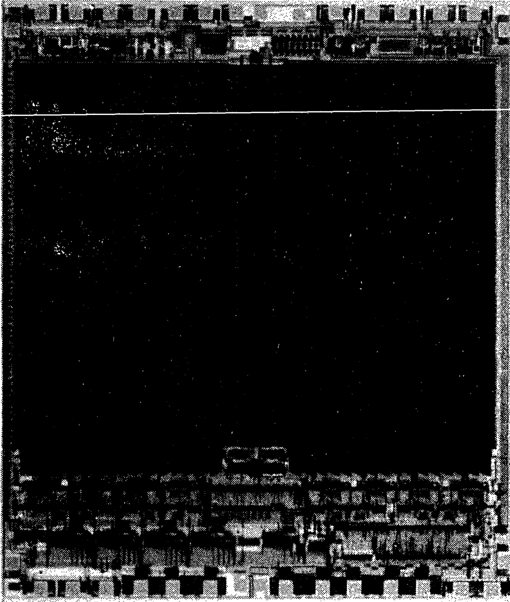


Figure 2. 1Mb die photograph

However, to build a manufacturable 1Mb flash memory, it is essential to be able to control the memory array erase Vt. The key is the proper selection of the erase Vt maximum and maintenance of a tight Vt distribution. The maximum erased Vt is set to 3.2V via the erase algorithm and the internal erase verify circuits [3]. Good oxide quality gives an erased Vt distribution width that does not change appreciably with cycling (Figure 5). The tight erase Vt distribution gives an order of magnitude of erase time margin to the fastest erasing cell (Figure 6).

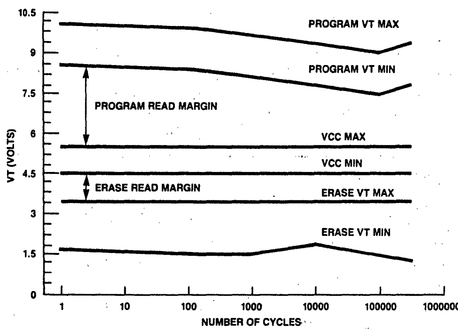


Figure 3. Array Vt vs. cyles

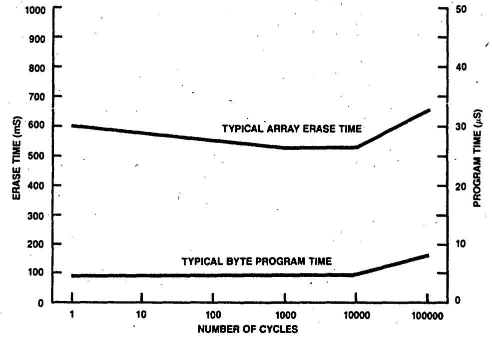


Figure 4. Erase/program time vs. cycling

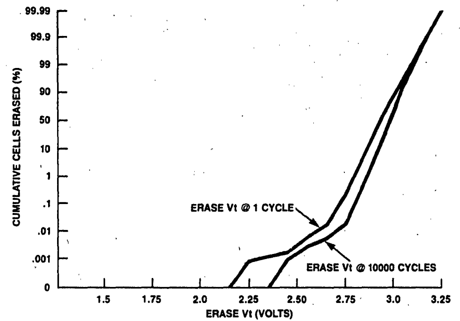


Figure 5. Erase Vt distribution vs. cycling

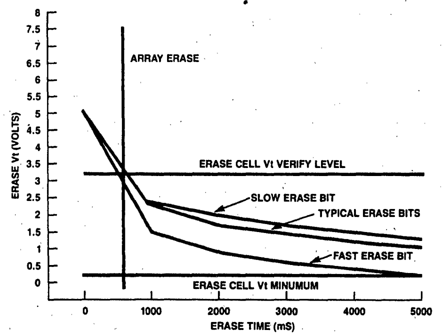


Figure 6. Array erase Vt profile vs. erase time

Array erase is executed by switching high voltage onto the source junction of all cells and grounding all select lines. The array source switch, shown in Figure 7, switches high voltage onto the source junctions. Transistor M16 is a very large device which pulls the source to ground during read and program modes. During erase mode, the high voltage latch formed by M5-M8 enables transistor M15, which then pulls the array source up to 12V. To obtain fast array erase times, this device must be made large enough to supply the grounded gate breakdown current which occurs on the sources of the memory array. The upper boundary on M15 current sourcing capability is set by the maximum allowable substrate current. If VPP is raised to 12V before VCC is above approximately 1.8V, the low VCC detect circuit formed by M1-M4 drives the node LOWVCC to 9V. Transistors M9-M11 then force the erase circuit into a non-erase state with M15 off and M16 on. When VCC rises above 1.8V, the chip will be reset into a read state.

Redundancy circuits consist of two flash memory cells combined with a cross-coupled bias and sense circuit ensuring low power consumption (Figure 8). When either M7 or M8 is programmed, the latch no longer draws power. By setting the levels of CLAMP and BIAS to Vt and 2Vt respectively, the B and BB levels are held to approximately one Vt. The signals F and FB along with the address signal drive the inputs to the XNOR circuits. The MATCH signals for all column addresses are combined to create the full match signal which enables a redundant column.

In summary, a 90ns 1Mb flash memory has been developed through the ability to scale the flash memory cell onto a standard CMOS 1.0μm technology. This memory has been optimized for in-system microprocessor-controlled reprogramming for more than 100,000 erase/program cycles.

REFERENCES

[1] V. Kynett et al., "An In-System Reprogrammable 256K CMOS Flash Memory," in ISSCC Dig. Tech. Papers, Feb. 1988, pp. 132-133, 345.

[2] G. Canepa et al., "A 90ns 4Mb CMOS EPROM," in ISSCC Dig. Tech. Papers, Feb. 1988, pp. 120-121, 323.

[3] V. Kynett et al., "An In-System Reprogrammable 32K × 8 CMOS Flash Memory," IEEE J. Solid-State Circuits, vol. 23, pp. 1157-1163, Oct. 1988.

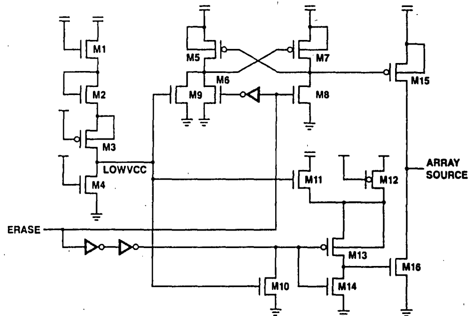


Figure 7. Array source switch

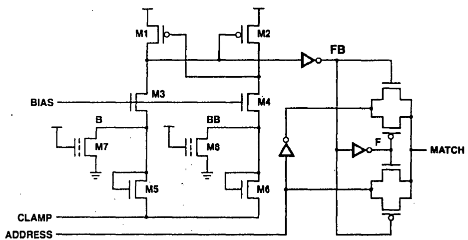


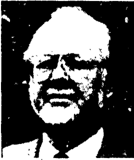
Figure 8. Redundancy circuits

3

SILICON BITS

Stan Baker

The Memory Driver



The primary driving force behind the personal computer revolution has been memory, not microprocessors. While one cannot give all the credit in one place, and microprocessors and software have their essential roles, the architectures and viability of these small computers has been due mostly to memory technologies—both semiconductor and magnetic.

That situation continues and more memory trends are afoot that will force computer systems in new directions in the near future. DRAMs are running out of the economic gas that has propelled the memory costs downward, not only leaving the door open for other memory technologies but demanding they enter.

Today's memory technologies are bubbling with new possibilities that will further revolutionize systems. At the heart of the changes will be nonvolatile devices. And the major player there will be flash technology.

The initial personal computers could have been made with CPUs that were not fully integrated, using gate arrays, LSI discrete logic or 2901 bit-slice architectures. But they could not have been made without low-cost, dense DRAM chips and low-cost floppy disk drives. The success of PCs then gave the economic stimulus to miniaturize hard disks which stimulated the PC business further.

The center of the computing universe is the data, not the processing engine. And the data is in the memory. And the ideal memory is nonvolatile.

Besides changing systems, the nonvolatile technologies will also alter the architecture of the semiconductor business internationally, with large scale impact on trade, political, and macro-economic issues. The leaders in the nonvolatile technologies are American companies. And they will not license their technology so readily as in the past.

There is a host of possibilities from flash, EPROM, EEPROM, battery backing, magnetic, optical, and the more remote ferroelectric technologies. Ferroelectric comes the closest to being the ideal nonvolatile RAM, but it is the furthest from reality. However, flash is here and, for the first time, promises to bring nonvolatile devices into the processing heart of computing systems in a big way.

Flash memories can have smaller cells than DRAMS and will be able to get more benefit from the latest lithographic and other processing equipment than DRAMs will. With only a year on the market the bit-count of flash devices has caught up with EPROMs and DRAMs, all now at 1 megabit per chip.

The 1-Mbit flash device just introduced by Intel has a die of 60,000 square mils. Current 1-Mbit DRAMs are larger, at about 70,000 square mils, and 256 kilobit SRAMs use about 75,000 square mils. 1-Mbit EEPROMs are about double, on the order of 130,000 square mils.

Flash will continue to track EPROM densities and soon outstrip even DRAMs, according to Richard Pashley, general manager of Intel's nonvolatile memory business. The only memory technologies that continue to track lithography in their cell size are EPROMs and flash devices.

Flash devices can be read as rapidly as EPROMs or DRAMs. But writing into them takes tens of microseconds per byte. And they are bulk erased in tens to hundreds of milliseconds.

Such long erase and write times may seem extremely limiting at first thought. But actually, the bulk of program and data storage does not need fast erase/write. That is why magnetic storage is so important. And that is what has some flash memory marketeers so excited—especially at Intel, which is nowhere in the DRAM and SRAM businesses, but the world leader in EPROMs. For flash devices are very similar to EPROMs.

Consider this example. If a computer were constructed with megabytes of fast volatile RAM directly serving the CPU, that can be erased and rewritten rapidly, massive blocks of nonvolatile flash RAM can take the place of magnetic storage backing that volatile memory. A few 4-Mbit flash chips will carry more data than most floppy disks.

That backup storage will significantly speed-up system performance and eliminate electro-mechanical reliability problems, as well as lots of weight and power drain. The flash devices can also be used to reduce the amount of volatile RAM, because some is used to store programs and data that seldom needs to be erased and changed. Such write-seldom sections of memory can be updated in a second or so, which is less than would irritate a human operator.

Fitting in this scenario, future microprocessors will have more and more memory on their die. That will be a good place for the fast RAM, made even faster by eliminating the inter-package wiring. And these internal RAMs will be organized to match the processing characteristics of the CPU which is not the case now with discrete RAMs. The flash and EPROM devices can then connect directly to the microprocessor package, eliminating discrete DRAMs and SRAMs.

Memory companies everywhere are working on flash devices. But Seeq Technology and Intel were the first to market. Since then, Texas Instruments and Toshiba have introduced versions. But Intel seems to be the only one supplying in significant volume, and it's Intel that has put the most corporate commitment—money and talent—behind flash. At Intel, flash technology plays directly off its EPROM technology in which Intel is still the world leader.

A passion for flash

According to Pashley, "flash is the way Intel will get back in the read-write memory business." In Pashley, Intel and perhaps the industry has its flash champion, and the success of any new technology depends on having the capable individuals that have the faith and lead the charge.

Pashley was the pioneer of scaling, the technique of shrinking MOS devices that is fundamental to the evolution of more and more dense MOS ICs. His process at Intel was termed "HMOS."

At the recent ISSCC in New York Intel described its 1-Mbit flash memory chip. Seeq Technology and National Semiconductor, who are jointly working on flash devices, described a 1-Mbit as well. And Texas Instruments described its latest flash, a 256k device that uses only a single 5-V supply.

December 1989

3

Flash Memories: The Best Of Two Worlds

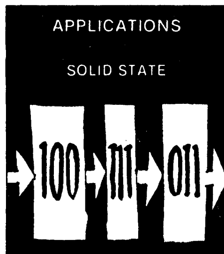
By
RICHARD D. PASHLEY
STEFAN K. LAI
INTEL CORPORATION

1989 Intel Corp.
Reprinted from IEEE Spectrum with permission.

Order number 295040-001

Flash memories: the best of two worlds

Filling a niche between conventional EPROMs and EEPROMs, these dense memories offer the latter's reprogramming convenience at relative cost advantages



In evolving from a concept paper in 1984 to megabit devices only five years later, flash memories have moved up the transistor-density curve faster than any previous semiconductor memory IC. They are based on the technology of either erasable programmable ROMs (EPROMs) or electrically erasable programmable ROMs (EEPROMs), and in price and functionality fall somewhere between the two, suiting any applications that require the former's denser storage plus the latter's ability to be reprogrammed without removal from a system. They also share these memories' nonvolatility and fast read access.

Those flash devices more akin to EPROMs cost less and promise rapid device and price scaling; their suppliers include Intel Corp., Santa Clara, Calif.; Seeq Technology Inc., San Jose, Calif.; and Tokyo's Toshiba Corp. The others, which utilize the more complex EEPROM technology, are slightly more expensive but provide more flexible reprogramming. They are made by Texas Instruments Inc., Houston, Texas, and sampled by Atmel Corp., San Jose, Calif., among others.

Of these three types of reprogrammable memory, EPROMs remain the best choice for applications where data almost never needs changing. Otherwise, flash memory devices should be considered. Although the average EPROM may sell for about $7, and a flash memory for about $25, the differential is wiped out by the expense of a single reprogramming. The in-system reprogramming of a flash device may cost as little as $1, whereas pulling an EPROM out of a system to erase it by exposure to 20 minutes of ultraviolet (UV) light may cost over $80 when equipment, downtime, and labor are factored in.

Meanwhile, EEPROMs should remain popular wherever it is necessary to erase bytes selectively. But flash products, which are erased in their entirety or section by large section, might do better for updating stored logic, when this must be done more than once but less often than in main memory, cache memory, or registers. Reprogramming costs are similar, but flash memories are less than half the price of EEPROMs.

Flash memories may even oust some battery-backed static RAMs (SRAMs), a bulky combination. In laptop computers, the flash device would occupy less space, at a lower cost per bit, and eliminate the dependency of memory on battery power.

Most flash memories have the same pinout as EPROMs, so that substituting them in a system requires primarily software alterations. (The densest flash ICs may have a few extra pins, however.) Furthermore, whereas a flash memory will nearly always be surface-mountable, that is not the case for some applications of EPROMs, whose packages include a window that must be accessible to UV reprogramming equipment. A flash-for-EEPROM swap, though, will almost certainly require rerouting the printed circuit, because flash memories may be up to four times denser.

Also, flash ICs closer to EEPROM technology need only a 5-volt supply, whereas those closer to EPROMs require a 12-V one as well, to drive the high-energy electrons that write data into them. This kind also requires a multistep algorithm that verifies erasure. Its makers say that the 12-V supply helps protect the IC against accidental erasure; those that supply 5-V-only versions counter that there are software techniques that may be employed to render such an accident too rare to be worth consideration.

The choice between the two flash memory types is sometimes determined by the application. In small embedded controller systems, missiles, or remote battery-powered systems, a 5-V-only flash memory is preferable. On the other hand, a 12-V supply is already available in some systems, such as desktop personal computers and laser printers.

With software of all kinds becoming more complex, the likelihood of changes to it, to update it or eliminate bugs, increases proportionately. That, in turn, argues for efficiently reprogrammable nonvolatile memories, and bodes well for the popularity of flash memory.

Consider the basic I/O system of a PC. It is typically stored in ROMs or PROM. Flash memory would allow the changing of I/O system code over a network or modem within minutes.

Also, portable computer systems' hard-disk drives may be replaced by flash memory modules offering lower cost, size, and weight, plus the greater reliability of solid state.

The operating system for an IBM PC AT or an Apple Macintosh is big enough as a rule to need storing on magnetic hard disk. With each year, however, flash ICs become more economical for greater amounts of storage; up to 2M bytes of flash memory are available with a new laptop from Psion Inc., Watertown, Conn., for example [see photo]. As this trend continues, flash ICs may supplant hard-disk drives of small capacity (up to 10 megabytes) in systems that could use a small, reliable memory with low power

Richard D. Pashley and Stefan K. Lai Intel Corp.

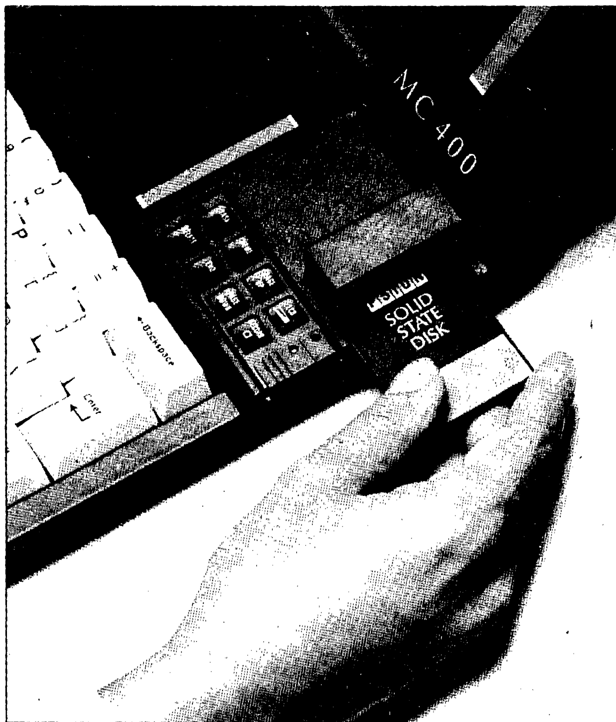
Defining terms

Electron trapping: the accumulation of electrons in imperfections in silicon dioxide, so that negative charge builds up and delays erasure of programmable memory devices.

Fowler-Nordheim tunnelling: a quantum mechanical process in which electrons tunnel through a thin dielectric from (or to) a floating gate to (or from) a conducting channel—the erase mechanism in flash memories and the program and erase mechanism in electrically erasable programmable ROMs (EEPROMs).

Hot-electron injection: in this context, the injection into the memory cell's floating gate by a vertical electric field of electrons with excess energy acquired from a high source-to-drain channel electric field.

Nonvolatile memory: memory that does not lose stored bits after power is switched off (includes ROMs, PROMs, EPROMs, EEPROMs, and flash memories).



Psion Inc.

The MC400 laptop computer from Psion Inc., Watertown, Conn., uses flash memory modules to replace disk drives. The computer has four module slots, and each module contains four Intel 128K-bit flash ICs in plastic leaded chip-carriers for a total of 2M bits of memory. The primary requirement for switching from disk to solid-state memory is rewriting the operating system's memory management code. Psion also makes an accessory that lets the user plug the MC400's modules into any IBM Corp. or compatible personal computer.

consumption. In such cases, though, another software modification becomes necessary. Data is stored in serial form on hard disk, and must be reformatted into bytes before it can be sent to RAM. Data stored on flash ICs is already in byte format, and operating systems are being rewritten to accommodate this.

At present, the programs for embedded controllers, such as those that operate automobiles and production machinery, are kept in other forms of nonvolatile memory. But flash ICs could serve here, too. And they could also speed up laser printers. Much of the formatting font and size information, along with the data to be printed, must now be loaded for each page from the central processing unit to the laser printer. Flash memory in a laser printer could store the font information for an entire print session just the once, so that pages would print out one after the other with less delay. A change in print session parameters would simply invoke an erase/reprogram cycle.

Flash memories are also candidates for use in flight data recorders and in communication equipment where parameters change often to accommodate different data communication formats.

Flash technology

Even as semiconductor memory began displacing magnetic core memory in the early 1970s, the inability of RAMs to retain data after power was turned off remained a problem. The invention of the nonvolatile EPROM deserved to be successful, despite the clumsiness and low repeatability of UV erasure. EEPROMs subsequently offered speedy in-system erasure with a strong electric field, which, however, at 12 megavolts per centimeter, so stressed the device's tunnel oxide as to limit the number of erase and write cycles possible. Also, the EEPROM memory cell, larger than the EPROM's, meant less favorable economics.

Accordingly, the challenge for semiconductor engineers was to fabricate a memory with the EEPROM's electric erasability but priced more like an EPROM, of comparable memory retention and cell size, and higher read/write cycling capability. Like both devices, the new one was to have high-speed read access. The flash memory is the response to this challenge.

Most flash memories are programmed with the EPROM's hot-electron injection technique. Each memory cell's field-effect transistor (FET) is turned on or off by the absence or presence of charge on a floating gate sitting above the conducting channel. Electrons accumulate on the floating gate because of the field produced by a large positive voltage on the select gate above the floating gate, and a similar voltage on the drain while the source is grounded. Once on the floating gate, the electrons are trapped there by the surrounding nonconducting oxide. The electric field they produce will then turn off the FET, storing a logic 0 in that bit location. Where no excess electrons are trapped on the floating gate, the FET's channel can conduct current and the cell has a logic value of 1. Thus, both in cell structure and in programming technique, the flash memory is very similar to the EPROM.

All flash memories are erased electrically in the system and in 1 or 2 seconds, like EEPROMs, but in bulk, like EPROMs. Electrons tunnel back into the source region in response to an electric field between gate and source. Some devices have been designed to make programming and erasing consistent with microprocessor control. The EEPROM's individual byte erasure is made possible by equipping each memory cell with a second, select transistor, and by forgoing the select transistor to obtain bulk erasure, flash memories can be built with much higher densities than EEPROMs.

Structurally, the flash memory cell is like the EPROM cell, being only slightly larger and with a thinner gate-oxide layer, usually 10–20 nanometers deep. But each supplier of flash memories has taken a slightly different approach to the device. Intel uses its ETOX (EPROM tunnel oxide) technology. Seeq employs a "phantom transistor" approach, which has a stepped-gate structure. Toshiba employs a triple-polysilicon, three-gate design. TI's is also a stepped, two-gate structure, with a thin dielectric to ease electron tunneling to the floating gate. The first three designs are programmed by hot-electron injection and erased via Fowler-Nordheim tunneling. TI's design depends on tunneling for both the write and erase mechanisms.

The Seeq cell puts a second transistor in series with the first to control erasure and also enable the erasure of small subsections. The approach in effect lengthens the channel, however, and limits programming performance. Programming and erasure occur through the same junction, stressing the gate oxide and guaranteeing fewer than 10 000 cycles, although up to 100 000 is typical.

Toshiba's flash cell is the most complex and largest of these three. It has a phantom transistor like Seeq's, also to control erasure. It has another layer of polysilicon for erasure through a polysilicon-to-polysilicon oxide. The poly-to-poly-oxide erase path requires higher voltage than either the Intel or the Seeq flash cell. Tunneling through poly-to-poly oxide tends to trap more electrons, so that the cell threshold increases with cycling. As a consequence, Toshiba's memory cells are specified for 100 cycles, and to 1000 cycles with special electrical screening tests.

TI calls its flash memory a merged-transistor advanced contactless EEPROM. The cell transistor and pass transistor are

merged so that its flash memory cell is structurally similar to Seeq's phantom cell, except that it has a thin (10-nanometer) oxide tunnel window for programming erasure.

Intel's cell is programmed through the drain, and erased through the source, which results in the reduction of stress on the gate oxide and a typical cycling capability in the 100 000 to 1 million range. The cell is smaller than the other three, which means not just smaller chips and more of them per wafer but a faster-to-program device with a shorter channel length.

The Intel cell uses a single FET with a floating gate for storage. To program a row of eight cells (a byte), the row decoder selects them and drives them to 12 V. The bits within the byte that are to be programmed as logic 0 are selected by the column decoders, which takes them to about 7 V.

Typically, hot-electron programming takes less than 10 microseconds per byte while tunneling takes between 5 and 20 milliseconds per page for programming, per chip for erasing. For bulk erasure, a second is more than adequate, and the reduced stress on the flash memory's gate oxide (compared with the EEPROM's) markedly ameliorates oxide-related problems that affect memory retention, time to program, and time to erase. All involve mechanisms affecting device quality and reliability.

Quality vs. reliability

IC quality is not to be confused with IC reliability. Quality describes how closely a chip conforms to its specifications upon delivery and is a manufacturing concern, dependent on the thoroughness of testing processes. Reliability describes how closely an IC continues to conform to its specifications over years of use, measured by failure rates of components that have been qualified and installed in fielded systems. Reliability will bear on the overall cost of ownership of a system because repair in the field costs far more than repair during production.

The failure mechanisms usually associated with program/erase cycling of electrically erasable memories (EEPROMs and flash memories) are charge loss due to latent oxide breakdown and electron trapup. Both manifest themselves when a device cannot be reliably programmed or erased within the maximum time specified.

To reduce oxide breakdown, a chip manufacturer can both improve oxide quality and attempt to reduce the stress on the tunnel oxide during programming and erasing. With Intel's flash IC, for example, the area of oxide involved (and the area that is stressed by the electric field) is confined to an overlapping area between source and gate. In addition, the reduction in electric field intensity across the tunnel oxide in flash memories to 10 MV/cm from the 12 MV/cm typical for an EEPROM theoretically should increase its reprogramming durability. In experiments where over 2000 Intel 1M-bit flash memories were cycled more than 20 000 times, none of the devices failed because of oxide breakdown, and several devices survived 1 million program/erase cycles.

The smaller the memory cell, the less capacitance it has, and the less charge need be added or removed for programming or erasing. Trapup is directly related to the amount of charge moving through the oxide, so the smaller charge requirement will tend to postpone its occurrence, stretching it out over many more program/erase cycles than for larger cells, with higher capacitance. Thus, the larger the cell, the more susceptible it is to trapup, and the more programming or erase pulses it will take to push sufficient electrons onto or off the floating gate.

Beyond the cell itself, peripheral chip functions and oxides are affected by repeated program/erase cycles. In general, the higher those voltages, the more vulnerable the peripheral circuitry is to functional failures. Intel's flash memories need no more than 11.4 V to meet their program/erase specifications. Other flash memories that require the same nominal 12-V external supply have on-chip charge pumps and internal voltage levels of 20 V and higher. These voltages put more stress on peripheral circuit oxides, to the possible detriment of reliability.

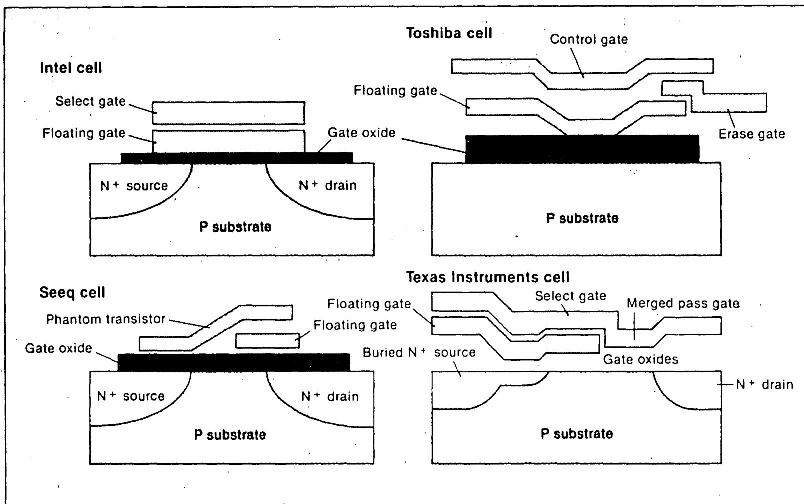
Looking down the road

CMOS is today's mainstream technology for both logic and memories. Within that technology, EPROMs and flash memories should be more scalable than dynamic RAMs (DRAMs) or SRAMs and EEPROMs. At a first level, there are differences in memory cell complexity. SRAM cells have four or six transistors. Turning transistors on or off to store logic levels (writing) is faster than storing charge in a capacitive well (DRAMs) or on a floating gate (EPROMs, flash memories, and EEPROMs). Sensing logic levels (reading) on SRAMs is also faster than on the other devices. But the price of that speed is increased cell complexity. Absolute SRAM cell sizes are over 10 times larger than for single-transistor devices (EPROMs and some flash memories) and scaling is more complicated because all dimensions cannot be reduced proportionately without upsetting some minimum spacing rules between lines and active devices. For example, distances between devices may not be allowed to shrink proportionately with line widths.

An EEPROM cell, employing both a bit-storage and a select

A comparison of flash ICs, EPROMs, and EEPROMs

Company, location	Density	Access time, nanoseconds	Power consumption, milliamperes	Erase/write cycles	Minimum erase area	Erase/write times	Voltage requirements, volts
Flash memories							
Intel Corp., Santa Clara, Calif.	1M bit	120	30	10^5	Bulk	1 second/chip, 10 μs/byte	5/12
Seeq Technology Inc., San Jose, Calif.	1M bit	200/250	30	10^3	Sector, bulk	12 seconds/chip, 525 μs/byte	5/12
Texas Instruments Inc., Houston, Texas	256K bits	170	15	10^3	Bulk	15 ms/chip, 15 ms/page	5
Toshiba America Electronic Components Inc., Irvine, Calif.	256K bits	170	30	10^2	Bulk	100 ms/chip, 200 μs/byte	5/12.75
Electrically erasable programmable ROMs (EEPROMs)							
Simtek Corp., Colorado Springs, Colo.	256K bits	120	80	10^5	Byte	10 ms/chip, 160 μs/byte	5
Xicor Inc., Milpitas, Calif.	1M bit	200	50	10^5	Byte	5 ms/page	5
Erasable programmable ROMs (EPROMs)							
Microchip Technology Inc., Chandler, Ariz.	256K bits	55	65	Up to 100	Blanket, ultraviolet	20 minutes max.	5/12
Texas Instruments Inc., Houston, Texas	1M bit	170	50	Up to 100	Blanket, UV	20 minutes max.	5/12



Of the four approaches to flash memory technology, Intel Corp., Toshiba Corp., and Seeq Technology Inc. have chosen designs closer to EPROM technology, Texas Instruments Inc. one closer to EEPROM technology. Flash memory cells are similar to an EPROM cell, except they have shallower gate oxides, usually 10–20 nanometers deep, to allow Fowler-Nordheim electron tunneling. Intel's cell employs a single field-effect transistor with a floating gate for storage, and is programmed and erased through different areas of its gate oxide. Seeq employs a stepped-gate, two-transistor structure; the second transistor helps control erasure and also enables the erasure of small subsections. Toshiba's cell has a triple-polysilicon, three-gate design; it also uses an additional transistor to control erasure. The source and drain of the Toshiba cell are perpendicular to the plane of the page. These three designs are programmed by hot-electron injection and erased by tunneling. TI's cell is also a stepped-gate, two-transistor design, but its oxide layer is constructed to ease tunneling, upon which it depends for both its write and erase mechanisms. Though several of these designs have more than one transistor, only the memory cell is depicted. Intel and Seeq are currently the only two companies producing 1M-bit flash memory products.

transistor, is over 2.5 times the size of a flash cell. Here, too, the added complexity and high voltages required may make proportionate scaling (equal reduction of dimensions and spacings) somewhat elusive. Surprisingly, even the DRAM cell, composed of a select transistor and a storage capacitor, is over 1.5 times as large as the flash memory cell. But on today's submicrometer scale, planar DRAM capacitors hold too little charge for reliable bit sensing, so that designers have gone to three-dimensional structures, such as stacked or trench capacitors. These constructions complicate manufacturing, reducing reliability and raising costs.

Having an active memory-cell transistor sense current like an SRAM and lacking the soft-error sensitivity of DRAMs, flash and EPROM technology may well be the most scalable memory technologies by the year 2000.

In geological terms, 10 years is insignificant, but in solid-state technology, 10 years is one-fourth the age of the transistor. However, by the year 2000, a 256M-byte flash memory using 0.25-micrometer geometry on a die 0.7 inch on a side is projected to sell for $1 per megabyte. Alternatively, less flash memory could be combined on the same die with application interfaces, such as high-speed data transfer interfaces, similar to today's burst mode, page mode, and "nibl" mode transfer schemes.

Several 256M-bit flash devices, without today's on-chip control features, may form a multichip memory subsystem run by a single controller IC, akin to the DRAM and DRAM controller subsystems of today. In embedded control systems, the flash memory device may be combined with other application-oriented logic to simplify and shrink the design and lower its cost. Such chips will be similar to EPROM-resident microcontrollers (such as Intel's 8748, 8749, and 8751) but have far more memory and can be much more functional than microcontroller processors.

The tendency is to view nonvolatile memories as vehicles for software modification; but in the future they may also be used to change the functions of hardware. Just as some of today's programmable logic devices (PLDs) use SRAM to control logic programming, tomorrow's PLDs may use flash memories for the same function. One other area in which flash memories promise to contribute is neural network systems, or processors that mimic the way the human brain works. It is likely that computers based upon neural networking concepts will use flash memories or EEPROM memories as key elements of their processing units.

To probe further

The first concept paper on flash memories was presented by Toshiba Corp., Tokyo, at the 1984 International Electron Devices Meeting. The 1984 *IEDM Technical Digest* can be ordered from the IEEE New Jersey Service Center, at 445 Hoes Lane, Piscataway, N.J. 08855; or call 201-562-5493.

The nonvolatile memory section of the 1989 International Solid-State Circuits Conference (ISSCC) *Digest of Technical Papers* contains papers on flash technology from Intel Corp., Santa Clara, Calif.; Seeq Technology Inc., San Jose, Calif.; and Texas Instruments Inc., Houston, Texas. It is also available from the service center.

The paper, "A 90ns One-Million Erase/Program Cycle Megabit Flash Memory," from Intel Corp., was published in the October 1989 *Journal of Solid State Circuits* special issue on logic and memory.

About the authors

Richard D. Pashley (SM) has served as general manager of Intel's Flash Memory Operation, Folsom, Calif., since April 1986. The previous five years he was director of Intel's Technology Development group in Santa Clara, and before that, he managed Intel's static RAM, static logic, bipolar memory, EPROM and EEPROM technology development activities. In 1976, Pashley developed Intel's HMOS (high-performance metal oxide semiconductor) process technology. He holds a doctorate in electrical engineering from the California Institute of Technology in Pasadena.

Stefan K. Lai (SM) has been engineering manager for Intel flash memories since 1986. As program manager for flash memory technology development during the two years prior to that, he co-invented Intel's ETOX (EPROM tunneling oxide) flash memory process technology. From 1979 to 1982, he worked at IBM Corp.'s Thomas J. Watson Research Center, Yorktown Heights, N.Y., where he was involved in tunnel oxide dielectric research. He holds a B.S. in applied physics from the California Institute of Technology and a Ph.D. in applied quantum physics from Yale University, New Haven, Conn. ◆

intel®

ARTICLE
REPRINT

AR-486

June 1990

Flash Memory Outshines ROM and EPROM

SAUL ZALES

INTEL CORPORATION

© 1990 Intel Corporation, 1990
Reprinted with permission from E E Times, May 1990

Order Number 295050-001

Flash Memory Outshines ROM and EPROM

SAUL ZALES, INTEL CORP., FOLSOM, CALIF.

As competitive pressures continue to mount, project and design leaders find themselves taking on more responsibility for overall system costs. In many cases, systems designers are expected to design with an eye toward controlling costs in all phases of the product life cycle, from conception through design and manufacturing—even up to post-sales service.

Flash memories may improve product costs and reliability in many markets

Beyond paying strict attention to total life-cycle costs, today's systems designers face other significant challenges. These include choosing the most efficient CPU architecture, designing for cross-vendor system connectivity, maintaining component and system quality, and building systems that can be serviced easily. Software issues facing project and design leaders include planning for past and future compatibility, minimizing code size, and balancing system performance and stability with time-to-market concerns.

Memory designs have coalesced around basic choices: disk and DRAM architectures, and EPROM-based designs. Although these technologies have been improved over the years, they still require designers to make some trade-offs for certain applications. One such application that has taken on a growing importance for many manufacturers is embedded control.

In the past decade, designers of electromechanical systems have come to rely heavily on the use of embedded microcontrollers. These parts have vastly improved control functionality and performance. Consider, for instance, the growing number of consumer-electronics items that are microprocessor controlled: microwave ovens, washing machines, VCRs, audio systems, and exercise equipment, to name but a few. These products ship with operating code stored in ROM or EPROM; the manufacturer assumes that the code stored in these memories will never change.

Consumer products are not the only items equipped with embedded microcontrollers, of course. Industrial machines, office-automation equipment, medical equip-

ment, communications equipment, avionics systems, and data loggers all include code stored in ROM or EPROM. In nonconsumer products, code changes are more likely to occur, requiring frequent memory fixes. Reasons for such changes include evolving customer needs, frequent demands for new features, improved algorithms, changing connectivity protocols, and eliminating software bugs.

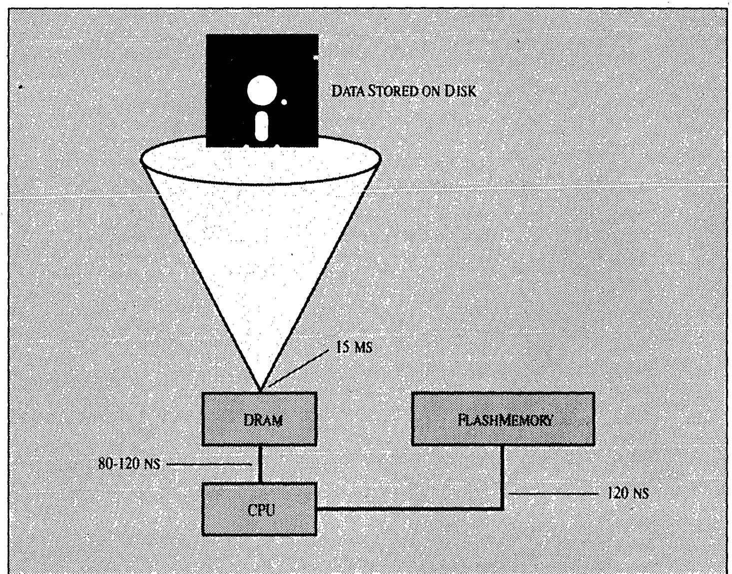
Code updates are impossible with some memory technologies; with others, they incur high costs and threaten product reliability (see box, "Memory Alternatives: Major Trade-Offs"). Designers of products that are likely to require code updating in the course of their life cycles should look beyond conventional memory options and investigate a technology that is well suited to such applications: reprogrammable flash memory.

Reprogrammable flash memories have the potential to improve product costs and reliability in many market segments. For example, ROM and EPROM parts are currently used in electronic engine controllers for automobiles. These parts cannot be replaced; they are sealed under a moisture-resistant coating. To change the code enclosed in one EPROM, the service center must replace the entire controller module, which costs about $200.

Starting in 1993, Delco and a few automakers will begin using flash memory in these modules. With flash memory in place, if a code change is required—for instance, if the Environmental Protection Agency mandates a code change to reduce exhaust emissions—the service center can simply reprogram the memory using a serial link from the service bay's diagnostic computer. No parts need to be replaced, eliminating the risk of damage to other components in the module.

Although flash-memory parts are more expensive than EPROM components, the additional up-front costs eliminate expenses that occur later in the product's life cycle—that is, when code has to be updated. Although customers generally assume the burden of paying for code upgrades, there are certain hidden costs to manufacturers that, taken together, can be significant. For example, consider the cost difference between transferring an upgrade electronically—as can be done with flash memory—and sending a highly paid technician to make a service call.

For whatever reason, most manufacturers do not include upgrade costs in their overall system cost estimates. If a year or two after a product is sold a code update is required, that expense usually is unaccounted for in the system's overall cost. But just because the expense is unaccounted for doesn't mean money isn't spent. Costly updates to existing products will show up on the organization's bottom line somehow. For this reason, even if only one code change is anticipated in the lifetime of a product, designers should opt for flash memory.



In a typical computer, the CPU's performance is slowed by the data bottleneck created by relatively sluggish disk access times. Embedding code in flash memory can eliminate this bottleneck for key operations.

■ PC APPLICATIONS

Flash memory fits extremely well in the embedded-control world, but its usefulness is not limited to such applications. It is becoming more apparent that flash memory will play a critical role in the reprogrammable environment as well. For example, today's personal computers are based on DRAM and disk drives. Obviously, these systems provide acceptable performance to users. Their performance can be improved more, however, by employing firmware based on flash memory.

ONE AREA IN WHICH FLASH MEMORY COULD BOOST PC PERFORMANCE IS IN THE COMPUTER'S BIOS

One area in which flash-memory firmware could greatly boost PC performance is in the computer's basic input/output system (BIOS). A PC's BIOS contains the system initialization, a power-on self-test, and basic component-level drivers. Currently, a computer's BIOS typically is stored in ROM or EPROM, which means that, in essence, it is an embedded-control memory. Without the nonvolatility these memories offer, the system could not initialize itself sufficiently to load software from the disk. This is because the CPU cannot read directly from disk—the disk's access time is much too slow.

Although the average PC user does not think about changing a machine's BIOS, a number of major PC vendors are planning to use flash memory for storing BIOS for several reasons. One is the rapidly changing nature of microcomputer technology. Consider the increase in system complexity from the PCs of 10 years ago to today's top-of-the-line machines. Today's 32-bit PCs offer the computing power of minicomputers. Computer makers need technology that allows for rapid adaptation to ever-changing situations.

The open nature of microcomputer systems adds to the need for flexible BIOS. Multiple vendors develop products that rely on the system's BIOS. In other words, the BIOS drivers hold the key to compatibility with both older hardware and software and newer products. The average stand-alone user who buys a computer, adds in a couple of boards, and runs a half dozen or so popular software packages usually doesn't need to worry about BIOS compatibility. But what about MIS or DP managers at Fortune 500 organizations? They may be responsible for hundreds or

even thousands of PCs fitted with any number of different add-in boards and running hundreds of different software packages. Incompatibilities are likely to abound in this type of environment.

With BIOS stored in EPROM or ROM devices, revisions to BIOS code are impractical. The end result is that many users must make do with systems that do not act predictably with certain software or hardware. If BIOS is stored in flash memory, however, vendors can provide a disk with new code and a simple upgrade utility.

Designing a flash-memory-based BIOS poses similar considerations as designing for the embedded-control environment (see box, "Designing with Flash Memory"). The system must contain a 12-V power supply regulated to ±5 percent. In addition, designers must deal with issues regarding the boot code. If power goes down midway through the BIOS upgrade, from where will the system boot? A boot PROM with the basic hardware initialization code could be included for safety's sake. It could be shadowed out once the flash memory has been properly initialized. Still, including a boot PROM incurs added costs and uses board space.

Designers at Ing. C. Olivetti have built a flash-memory BIOS without the PROM safety net. The Italian company's new 80486-based microcomputer includes a flash-memory BIOS without a boot PROM. Designers decided that the risks involved in eliminating the PROM were minimal. A typical flash-memory upgrade takes 7 to 10 seconds. The chances of power going down during those 7 to 10 seconds are not great enough to merit inclusion of the PROM, Olivetti decided.

■ Beyond BIOS

BIOS is only one PC component that can benefit from the flexible-firmware concept. The setup and diagnostics programs

WITH FLASH MEMORY, IF OPERATING SYSTEMS ARE UPGRADED, USERS CAN SIMPLY REPROGRAM EMBEDDED PARTS

that ship with every system can be stored in flash memory. Another design improvement is to put the operating system into flash memory. Consider the PC's boot sequence. When the system is powered up, the CPU executes the BIOS hardware initialization, performs a power-on self-test, and spins the disk up to speed. Only after the disk stabilizes at its operating velocity can the CPU begin to read the operating system.

Most operating-system code is read-only, which means that, theoretically, it could be included in the BIOS ROM to allow for faster system power up. The reason that this has not been done on a wide scale is simple: Systems designers recognize that operating systems evolve and improve over time, and that a flexible environment is required. With flash memory, however, concerns about upgrades are eliminated. If operating systems are upgraded, users can simply reprogram flash memories, as with flash-memory BIOS. Two major operating-systems houses, Microsoft Corp. and Digital Research Inc., now offer their operating systems in a form suitable for storage on ROM or flash memory.

In addition to storing the operating system in desktop or laptop systems, flash memory can increase network performance for intelligent terminals and diskless engineering workstations. Large networks, such as those used in airline-reservation systems or retail point-of-sale systems, often bog down during peak transaction periods. The load on the network can be reduced by storing the operating system, LAN protocols, or even scheduling or pricing tables in flash memory. Updates can occur during off hours through the network itself. Curtis Inc. (St. Paul, Minn.), which has offered EPROM or SRAM solutions to this problem for a number of years, now also offers flash-memory products to improve network performance.

■ Office Applications

Firmware based on flash memory can lead to significant efficiency improvements in several office applications. For instance, most offices now deploy laser printers for high-quality output. Laser printers allow users to produce typeset-quality memos, presentations, and the like. In many cases, the type fonts used with laser printers are stored in software on a PC's hard-disk drive. Users download these fonts as they are needed to the printer's RAM. Every time a font is changed, the downloading process must take place. Alternatively, many laser printers are equipped to take fonts from ROM- or EPROM-based cartridges that plug directly into the printer. The problem with ROM cartridges is that fonts stored in them cannot be changed. Any given cartridge may contain only a couple of desired fonts, which severely limits the flexibility of laser technology.

Flash memory could greatly enhance the speed and flexibility of laser printers. Using a flash-memory-based cartridge, users or work groups could develop their own font libraries. Since flash memory is nonvolatile, these fonts will stay resident in the printer without the use of batteries or uninterruptible power supplies.

Flash technology is well suited to application storage as well. Software stored on disk greatly reduces system performance, since disk access times are very slow compared with memory and CPU speeds (see figure). Of course, this speed bottleneck is not critical to most PC users; if a program takes a few seconds to load, so be it. For high-end systems, however, flash memory can boost system speed significantly by functioning as a code accelerator by storing programs or code that are accessed most often.

One high-end application that could benefit greatly from flash memory is CAD. Some CAD programs minimize the RAM used for program storage in order to maximize the data-memory space. To do this, the complex software swaps submodules into memory from disk as needed. Accessing the disk for a new module or library ties up the system and degrades performance.

Many CAD users install large add-in memory boards set up as RAM disks to avoid the transfer bottleneck. Whenever the system resets or powers down, however, this RAM disk loses its memory. A system containing a flash-memory disk emulator would not have this problem. Such a system for CAD and engineering applications is available from Digipro (Huntsville, Ala.).

■ Portable PCs

Laptop and notebook-sized portable computers stand to benefit greatly from flash-memory technology. Although small, relatively efficient 2.5-inch disk

FOR HIGH-END SYSTEMS, FLASH MEMORY CAN BOOST SPEED BY FUNCTIONING AS A CODE ACCELERATOR

drives have come onto the market for laptop computers, their efficiency rating is derived primarily from power-management schemes. If no access has been made to the disk for a certain period of time, the

3

drive goes into a low-power standby mode. Unfortunately, the delay caused by switching from standby back to operational mode further impedes the already-slow disk-to-memory transfer.

From a power perspective, these small drives use relatively less power than desktop disk drives. Active power specifications in the 1-W range are attainable, as opposed to the multiple watts consumed by desktop-system drives. Compared with flash memories, which use about 50 mW during active reads, the 2.5-inch drives are power hungry. In addition, in standby mode flash memories use only 150 μW of power.

Notebook-sized computers do not have space for even the smallest of drives. Additionally, users of notebook-sized systems demand much longer battery operation than is typical of laptops. Typical battery operation for notebooks range from 50 to 100 hours, compared with two to five hours for laptops.

The typical notebook computer contains ROM cards for applications and SRAM for storage of data files. This architecture has a few flaws. Users who purchase software for a desktop system may not want to spend another few hundred dollars for ROM-card versions. If flash-memory storage is provided, users can download software at a much lower cost.

Some notebook-sized computers let users load applications into SRAM or low-refresh DRAM (pseudo-SRAM). This procedure, however, is highly dependent on the system's battery; when the battery dies, memory is erased. With flash memory, no power is consumed when the system is off. Additionally, flash memory is less expensive for bulk storage than is SRAM.

Psion, a company based in the United Kingdom, recently introduced a notebook computer based on flash memory. Psion compared the price differences between SRAM and flash memory and found flash memory to be much more cost-effective. Based on the power savings of flash memory and very tight system design criteria, Psion built a 60-hour operational system.

■ File Storage Under DOS

Given the compelling reasons to adopt flash memory in the various PC environments, designers can then ponder the question: How can a bulk-erasable memory be used for file storage under DOS? The DOS directory and file-allocation tables (FATs) require the ability to erase and rewrite single bytes and files. The answer to this has many facets and depends on the degree of flexibility needed.

The most inflexible, but easiest to implement, approach involves creating a fixed disk image. Before programming the flash memory, the vendor combines the operating system, utilities, and specific applications onto a disk. It then runs this suite of programs through a utility that adapts the software for ROM or flash-memory storage. Finally, another utility creates the DOS directory and FATs and assembles the disk image. Digital Research offers both a ROM-format DOS version and disk-image utilities. Microsoft offers an optimized ROM-format DOS for the portable market as well.

Software embedded in flash memory can be changed without compromising system performance

Makers of notebook-sized computers, intelligent terminals, and dedicated hand-held computers might consider this strategy. A laptop vendor that offers a low-power disk might consider this approach as well. The most-used software, such as the operating system, calendar, alarm, and communications software, will load more quickly and not burn as much power on each access. Upgrade-software disks could be sold to registered system owners already formatted with a new disk image.

Another approach currently used by Digipro involves adapting RAM-disk emulator drivers to flash memory. This entails trapping disk writes and programming the information algorithmically.

From the user perspective, the flash-memory disk operates as a hard disk, but with 100 times the typical read performance of a hard disk. To load the flash-memory disk emulator, a user simply issues the DOS Copy command.

A drawback to this approach is that loading any application or its libraries forces a rewrite of the entire directory and FATs. For example, a file that contains only 2 Kbytes and should program in 50 ms may take several seconds to store because of file-system overhead. Note that this drawback may not be important as far as code acceleration is concerned. Files are loaded to the flash-memory disk as an off-line task, so the write-performance impact is minimal to read-mostly performance.

A third solution to the DOS compatibility problem comes from Microsoft, which has developed a flash-memory file system that loads under MS-DOS and other compatible operating systems. When a user adds a directory or a file to the flash-memory disk, the file system adds the information in a linked list. Should the user delete the directory or file, the file system marks an attribute field as being inactive. When the disk fills up, the user copies the active files and directories to a hard disk on the desktop system or to another disk on a portable system. Since the DOS Copy command only grabs active files, the newly transferred version contains a clean, unfragmented linked list. The user then erases the original flash-memory disk.

As these examples illustrate, flash memory offers alternatives to current problems in system architecture. To offer true solutions, a memory technology must also satisfy three objectives: It must provide acceptable density, incur a reasonable cost, and offer EPROM-level reliability.

Intel's ETOX flash-memory technology is the first new technology in nearly 20 years to satisfy all three objectives. It is dense—the 1-Mbit 28F010 has been shipping in production units since April 1989. Costs have decreased dramatically as volume has ramped. And because the ETOX process and memory cell so closely parallel mainstream EPROM technology, flash memory has reliably doubled in density three times in the last two years. Additionally, a flash-memory device can be reprogrammed about 100,000 times—a sufficient number for most firmware and disk applications.

Current Intel flash-memory product offerings include the 28F256, 28F512, and 28F010. These devices are 32-Kbit × 8, 64-Kbit × 8, and 128-Kbit × 8, respectively. All products ship in either surface-mounted ceramic DIP or PLCC. Higher densities, plastic DIPs, and smaller packages will be available shortly. Additionally, for those designs requiring a bulk-memory solution similar to that offered by DRAM vendors, Intel offers eight 28F010 PLCC units mounted on a single in-line memory module (SIMM).

With flash-memory technology, software can be changed without compromising the performance of disk access. The task can be accomplished in a reliable manner and without the high costs associated with service calls by field technicians. Finally, flash memory is nonvolatile—data stay resident even when the power supply is cut off. Together, these attributes enable flash-memory technology to provide users with flexible firmware and improved system design. ■

About the Author

Saul Zales *is a senior applications engineer at Intel Corp. in Folsom, Calif. He holds a BSEE from the University of Pennsylvania.*

Designing with Flash Memory

For many embedded-control applications, whether the control is of the sequential, closed-loop, or data-control variety, flash memory offers dense, reliable, rewritable, nonvolatile storage. From a systems perspective, the memory reads the same way as an EPROM, EEPROM, or SRAM, with access speeds of up to 120 ns. Intel fabricates flash-memory devices using its EPROM Tunnel Oxide (ETOX) process. ETOX is based on Intel's high-volume CMOS EPROM process. Because of this, flash memory reprograms in the same way as an EPROM—via CPU-controlled algorithms.

With flash memory, the designer implements the reprogramming algorithms, which are simple closed-loop algorithms that require 500 to 1,000 bytes of code. (Intel offers sample code generated for different base processors to minimize the software effort.) Because flash memory is bulk-erasable, this code must be stored and executed from another memory while the main code is updated. For this process, many designs rely on internal ROM space on microcontrollers or small EPROM boot loaders on Intel 80X86 systems. The boot loader contains sufficient code to initialize the system and reprogram the flash memory.

As with EPROM, flash memory requires a programming power supply of 12 V ±5 percent. Some systems feature a 12-V supply, but others do not. Systems with analog circuitry, for instance, often contain rails of 15 V or higher. In these systems, the programming power supply can be generated by regulating the higher voltage using an LM317-type regulator.

For memories that have a power supply of 5 V, designers can opt for either monolithic or discrete charge pumps, such as those as offered by Valor Electronics and Linear Technologies. Since flash memory requires only 30 mA per device from the programming power supply during active programming and erasure, these boost circuits can be made fairly small.

Some flash-memory manufacturers claim to have simplified programming and erasure by developing automatic controls and 5-V-only programming. The 5-V technologies are based on EEPROM-programming rather than EPROM-programming techniques and generate high voltages internally. They require larger board spaces and are less dense, more costly, and less reliable than 12-V designs. And although they are called flash memories, these parts tend to be nothing more than EEPROM technologies.

UPDATE

October 1990

Memory breakthrough drives miniaturization

You can use it like RAM and carry it around like a disk. Flash EPROM technology is poised to change the way portables are made.

Last year—a year in which assumptions about the world seemed to fall with bewildering regularity—a truism about computers suddenly became irrelevant. Intel Corp. in Santa Clara, Calif., unveiled a new type of memory chip, the flash EPROM, which combines the flexibility of RAM with the permanence of disks. The distinction between RAM and disk started crumbling like the Berlin Wall.

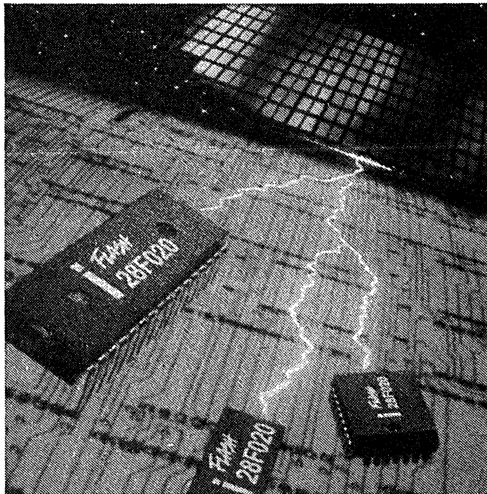
Now the implications of flash EPROMs for portable computing—particularly notebooks and handhelds—are becoming clearer. If supporters of the technology are correct, flash EPROMs could eventually replace bulky, power-sapping disk drives in computers and serve as a universal storage medium for nearly every electronic device that uses memory.

Based on older EPROM (erasable programmable read-only memory) technology, flash EPROMs do not need a backup power supply to retain data. Like regular EPROMs and dynamic RAMs (DRAMs), they can be packaged in plastic cases and plugged into computer motherboards. And like static RAMs (SRAMs), they can be integrated on credit-card-size memory cards, providing a removable storage medium for software. Intel guarantees 10 years of data life.

Flash EPROMs are likely to rearrange the mix of chip- and disk-based storage in all computers, especially the smallest portables. When used on removable cards, they can replace floppy drives, eventually for one-quarter the cost of currently available SRAM cards, according to Intel. At this writing, however, computer makers were waiting for lower prices before making the jump from SRAMs to the new chips.

Flash EPROMs could also be used to store programs (such as Tandy's Deskmate interface or MS-DOS) which now take several seconds to load from hard disk to RAM while sucking up precious electricity. Programs stored in flash EPROMs load nearly instantaneously. One company, Cardinal Technologies Inc. (Lancaster, Pa.) plans to offer an expansion board containing Digital Research's DR DOS in regular ROM paired with 2MB of flash EPROM for storing applications.

Engineers also envision flash EPROMs bringing major changes to the embedded computers and controllers that are becoming mainstays of modern life. Computerized engines could be reprogrammed to reflect evolving fuel mixtures as a car ages (currently, such programmable memory is dependent on battery power). Digital electronic cameras will use flash EPROMs instead of digital tape or film to store photographs with sharpness comparable to that of 35-millimeter cameras.



Intel's flash EPROM chips can store data without a constant supply of electricity.

Although London-based Psion PLC is first out of the gate with two flash-based notebook computers, the MC200 and MC400 (available this summer) you can expect to see other innovative

© INTEL CORPORATION, 1990.
Reprinted with permission from Portable Computing.

Order Number: 295055-001

machines using the new technology by early next year, according to Intel.

Hardware and software vendors are already using flash EPROMs to store vital code normally residing on ROM chips that can't be conveniently reprogrammed. When encoded in flash EPROMs, the computer's ROM BIOS, which manages hardware's interaction with software, can be updated via modem or floppy disk. (Phoenix Technologies Ltd., the largest BIOS maker, began offering flash EPROM versions of its products last spring.) BIOS upgradability will let you take advantage of new power-management breakthroughs and laptop peripherals without buying a new machine.

Flash EPROMs have one disadvantage that slightly limits their use: writing data to them takes nearly as much time as writing to a floppy. For this reason, they aren't as efficient as regular RAM at running applications like word processors and databases, which involve opening and closing files regularly. Flash EPROM proponents admit that computers based on the technology will sometimes use traditional DRAM storage for data manipulation, with flash EPROM cards taking the place of floppy and hard disks.

All the major laptop vendors are considering flash EPROMs, according to Kurt Robinson, Intel's product line architect for flash EPROMs. "Just about everybody is updating the BIOS portion of their machines from ROM to flash," Robinson says. In addition, Microsoft has thrown its support behind the technology by releasing file-management software that lets MS-DOS treat flash EPROMs like disk drives.

Intel expects steady increases in flash EPROM storage density at least through 1996. One-megabit chips are selling now, two-megabit versions should be available later this year, and four-megabit chips should follow in 1991. When 16-megabit chips arrive by 1994, vendors could introduce 32MB and 48MB "hard drives" on a card roughly two by three inches.

Texas Instruments (TI) is offering a similar technology that uses less power than Intel's flash EPROMs during data writing. In the typical flash-EPROM computer, all logic and memory operations require five volts of electricity, except for writing to the flash EPROM, which takes 12 volts. TI's new chips eliminate the need for a 12-volt power supply anywhere in the machine, saving space and weight. 256-kilobit chips are already available, with one-megabit versions expected by the end of this year.

Production efficiencies and price competition are likely to drive flash EPROM prices down to the level of dynamic RAM chips by 1994, Robinson asserts. By then, the whir of disk drives could be little more than a fast-fading memory.

intel®

ARTICLE REPRINT

AR-493

November 1990

Look Out EPROMs, Here Comes Flash

© 1990 INTEL CORPORATION

Reprinted with permission from Electronics Magazine, November 1990

Order Number: 295057-001

LOOK OUT EPROMs, HERE COMES FLASH

AS THE LAPTOP BOOM SPARKS MEGAGROWTH IN NONVOLATILE MEMORY, THE PLAYERS ARE LINING UP AT BAT

BY SAMUEL WEBER

MEMORIES

3

IN THE WORLD OF NONVOLatile memory, flash is where the action is. As technology advances push densities higher and costs lower, memory vendors are anticipating an explosion of applications for this versatile device. They are casting covetous eyes at the current $3 billion market for the venerable ultraviolet-erasable EPROM and the high-density segments of the more sophisticated full-featured electrically erasable PROM (EEPROM). Some even speculate that as speeds get better than 100 ns, some RAM applications may fall to flash as well.

The potential market size and growth rate are so attractive that old-line semiconductor vendors like Advanced Micro Devices, Hitachi, Mitsubishi, Signetics, and Texas Instruments are mounting big efforts with first-time flash products to join battle with companies already in the market. These include market leader Intel Corp., along with smaller but experienced flash suppliers like Atmel, Catalyst, and Seeq. Exel, NEC, and Waferscale are also expected to launch products soon.

One factor driving the growth of flash is the boom in portable, laptop, and palm-size computers, expected to grow fivefold to 11 million units by 1994. They offer a big opportunity to suppliers of an all-solid-state substitute for floppy and hard disks (see p. 52). In this market, flash offers distinct gains in size, power dissipation, reliability, and speed.

"At the 4-meg level," says Robert Tabone, Hitachi Ltd.'s product marketing manager for static RAMs in Brisbane, Calif., "suddenly we will have memory cards with the density to rival a hard disk. While the cost won't be at parity then, by 1994 we expect that to occur. With the explosion of laptop and notebook computers and many applications requiring high density, this technology is absolutely going through the roof."

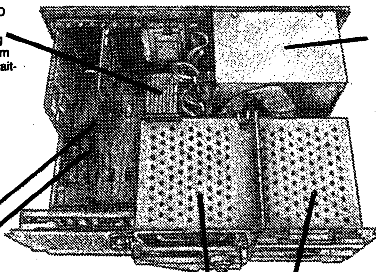
There are literally thousands of applications for low-cost, high-density electrically reprogrammable memory in automotive, telecom, point of sale, computer peripheral, industrial control, instrumentation, military, medical, and many other areas where large numbers of sockets await. All this means that flash sales will swell from a small base of $37 million this year to $134 million in 1991 and $1 billion by 1994, says Mary Olsson, industry analyst for Dataquest Inc., the San Jose, Calif., market research firm.

This growth largely stems from the industry's need for an in-system programming solution, says Krish Panu, vice president of marketing and sales for Catalyst Semiconductor Inc. of Santa Clara, Calif. "Time to market is getting critical and system life cycles are getting shorter. Having in-system reprogrammability makes it easy to upgrade and modify products quickly and reliably," he says.

And flash is quick: a device can be erased and reprogrammed in less than 5 s, about half the time it takes for a full-featured EEPROM. And erasure can be done without removing the device from its socket, unlike EPROMs, which erase in about 20 min. Thus, code changes in prototypes can be made in seconds, and board updates can be done without disassembly. These are just two simple examples of the possibilities flash offers to system designers.

At this incipient stage, there are several

How Flash Enhances Systems

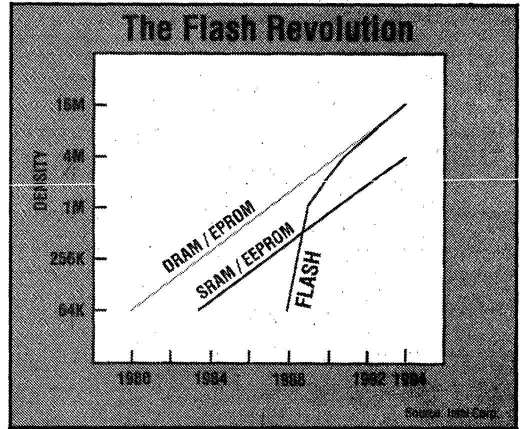


approaches to flash technology and a lack of standards, although one may be emerging in at least de facto form. Right now, chip makers follow different roads in cell size and design, method of writing and erasing, power-supply requirements, and endurance (the number of write/erase cycles that can be performed before deterioration of the gate oxide).

Each variation has advantages and disadvantages, a situation that can be confusing to potential users. Also confusing is just where flash fits in the hierarchy of nonvolatile memory options available (EPROMs, full-featured EEPROMS, battery-backed RAMs, and nonvolatile RAMS of different types). Invariably, the choice boils down to cost and system requirements. The system designer must consider such questions as frequency of writing or erasing, whether byte, page, sector, or bulk alterability is required, the density needed, power-supply availability, and price.

The multifarious designs sort themselves into two basic approaches, distinguished by whether they require one or two voltage supplies. Both can trace their lineage to EPROM technology, using a floating-gate structure but with a thinner gate oxide. But they differ in their cell structure—whether they require one or several transistors per cell. In general, the one-transistor cell requires a 12-V supply for programming and a 5-V supply for read, but yields a small cell size. This results in higher density, smaller chip size, and lower cost than the 5-V-only approach.

Intel's self-aligned stacked-gate cell, based on its proprietary ETOX (EPROM tunnel-oxide) technology, is the paradigm for the one-transistor school. At midyear, the Santa Clara firm announced the highest-density flash part currently available, the 2-Mbit 28F020. The competition either has or will shortly introduce 1-Mbit parts, and most have 4-Mbit flash devices in development. These should start to appear in late 1991.



The Intel approach received validation in the September announcement by Advanced Micro Devices Inc. of a 1-Mbit flash memory compatible with Intel's 12-V memory pinout and software-programming routines. AMD is committed to "establishing a de facto standard" for flash, says Steve Grossman, director of marketing for memory products at the Sunnyvale, Calif., company. Catalyst also follows the Intel pinout and algorithm lead in its 1-Mbit, 120-ns part, the CAT28F010, now being offered as samples. Exel, Hitachi, Mitsubishi, and Toshiba also opt for Intel compatibility.

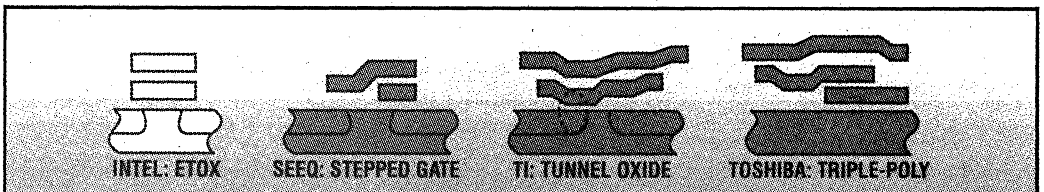
One problem with the single-transistor cell is the possibility of overerasure and consequent current leakage, resulting in false data readings. This occurs when a cell in the zero state receives an erase pulse, whereby it can be driven into the depletion mode. The column-sense amplifier can read this leakage current falsely as an erased cell. Intel and its emulators overcome this with a programming algorithm that first programs up all the cells on a chip to 1 before erasing.

Seeq Technology Inc., a major competitor of Intel's, overcomes this problem by means of a different cell structure. Its split-gate cell employs what amounts to a two-transistor architecture, but exacts only a small premium in cell area. Through a diffusion process, the split gate creates a "phantom transistor" that looks like a series transistor, says Richard Norris, marketing manager for Seeq's Memory Division in San Jose. "This allows us to isolate the cell from others in a column. The series transistor acts like a valve, and even if you overerase the cell and it gets leaky, if you don't select the transistor the leakage doesn't reach the column."

Another difference from Intel is the ability of the Seeq parts to erase a sector or small portion of the memory. With the Intel device, the entire chip must be erased before writing in new data. "We have sector-erase," says Norris. "There are 128 columns and you can erase and reprogram any one of those without altering any of the others."

Norris insists that the 15% premium on cell size and 10% on overall chip size is worth it for the advantages Seeq offers. "Furthermore, in our next-generation part, cell sizes will be within 5 mil^2 of [Intel's], because Intel adds a lot more external circuitry to prevent that overerase from happening." Seeq now produces two parts, the 512-Kbit 48F512 and the 1-Mbit 48F010.

The dual-power-supply requirements of most of today's flash EEPROMS add cost and space penalties for system design, and for this reason, some vendors are developing single-supply flash technology. Both Texas Instruments Inc. in Dallas and Atmel Corp. of San Jose have introduced 256-Kbit products of this



VARIATIONS ON A THEME

Chip makers are taking many routes to flash, including these, but all the designs trace their lineage to EPROM technology, using a floating-gate structure but with a thinner gate oxide.

EMBEDDED GROWTH

THE POTENTIAL POWER of embedded flash in microcontroller applications is spurring development of new devices. The embedded market could grow to $1 billion by 1992 or '93, says Bruce McCormick, vice president of marketing at Intel Corp.'s flash memory operation. A big chunk of that will be automotive, says Greg Armstrong, manager of application-specific memories at Texas Instruments Inc.'s Semiconductor Group in Dallas. Here, Armstrong says, programmable memories that now employ EPROMs for storage codes, lookup tables, and engine and transmission parameters will go flash by the end of the 1990s. This trend will be accompanied by the use of larger amounts of flash memory on automotive microcontrollers for greater system-level integration.

A case in point: at the recent Convergence Conference in Detroit, Chrysler Corp. demonstrated its Ultradrive trans-axle electronic control system. With TI flash devices and microcontrollers, it keeps the shift quality consistent throughout the life of the transmission, monitors system performance, and provides diagnostics at the assembly line and dealership.

TI is also merging its flash technology with the emerging boundary-scan JTAG (IEEE 1149.1) test standard to provide a unique device for automatically and permanently maintaining system or board history. Dubbed a testability-ported diary memory, the TMS29F816 can store diagnostic information in the on-board 5-V, 16-K flash memory.

Also recognizing the potential of this embedded market, Intel is about to introduce the 88F51FC, a CHMOS single-chip 8-bit controller with 32 Kbytes of on-chip user-programmable flash.

The Inmos Division of SGS-Thomson Microelectronics in Phoenix is taking the module approach to embedded flash by combining a 16-bit IMS T222 Transputer with 256 Kbytes of flash in a credit-card-sized module, the IMS B418 ROM TRAM. Unlike Intel's entry, the device can be block-erased in 4 Kbytes.—*S. W.*

genre, using very similar technologies. In fact, Atmel hints that the two companies are discussing mutual cooperation.

Atmel's device, which it calls a PEROM (for programmable erasable ROM) cell, uses two transistors. This eliminates the need for a high-voltage programming process because it isolates the cells being programmed from the other cells. The AT29C256 can be erased or programmed in 64- or 128-byte sectors (if desired, a bulk full-chip erase is available). The device has 160-ns read-access time and write time of 10 ms for 64 bytes (5 s full chip).

TI's TMS29F256 is based on a 1.5-transistor cell fabricated in the company's ACEE (advanced contactless EPROM) technology. For writing and erasing, it uses a tunnel diode in the channel region of the transistor cell rather than the conventional Fowler-Nordheim floating-gate structure. With access times of 170 ns, the device can be programmed 1 byte at a time or in the page mode from 2 to 64 bytes at a time. At the 256K level, TI is using a 1.5-μm ACEE process and will scale down to 1.0-μm for the 1-Mbit part. TI's 4-Mbit prototype uses 0.8-μm technology.

While flash device vendors refine device technology and tussle over the right approach, they are also taking hard aim at those beckoning sockets. Last month, Intel introduced the first flash-memory-based IC card in 1- and 4-Mbyte sizes aimed at laptop, notebook, and palm-top computers. Card densities should converge rapidly toward those of hard disks and greatly exceed those of floppy disks in the late 1990s, the company says.

The Intel cards are intended for applications in updatable application code, application-code and data-file storage, and data acquisition. They will be competing with existing memory-card technology, which includes expensive battery-backed RAM cards, unalterable ROM cards, and one-time-programmable EPROM cards. They are unsurpassed as a disk replacement in portable PCs, says Jim Weisenstein, Intel's flash-memory-card manager in Folsom, Calif. Reduced power consumption, resistance to shock, a doubling in write speed, and 3.5 times faster read time are among the benefits he cites.

Meanwhile, Microsoft Corp. of Redmond, Wash., has issued a Flash File System that runs as a software driver under MS-DOS. It effectively makes the flash-card memory behave like a disk, reacting to familiar DOS commands and storing data files sequentially.

On another front, Western Digital Corp. of Irvine, Calif., is working on a solid-state disk using flash technology with partners SunDisk Corp. of Santa Clara and AT&T Co. in Allentown, Pa. Details were scant at press time, but Ilene Graney, Western Digital's director of marketing for storage products, says the company has been involved in the project for two years. The device is not intended for general EPROM replacement but can be used for this application. It will be a 4-Mbit chip, Graney says, capable of assembly in 10-, 20-, and 40-Mbyte disk replacements potentially equivalent to a 1.6-in. disk. Access time will be under 2 ms.

Packaging is an issue in flash, and increasingly the 1-Mbit-and-higher chips are being produced in the new TSOP (thin small-outline) packages. Its small form factor of 20 by 8 by 1.2 mm makes TSOP ideal for the flash-card market as well as other embedded applications. It also is desirable for surface-mounted boards. Another possibility comes from White Technology Inc. The Phoenix company's WF1024KB-150 is an 8-Mbit flash-memory module packaged in a 34-pin, hermetically sealed metal package. It is built with eight 1-Mbit flash chips, organized as 1 Mbyte by 8, and assembled on a thick-film substrate. Each of its eight pages can be erased a page at a time. The device is guaranteed for 10,000 erase/program cycles.

Meanwhile, cell sizes for flash memories are shrinking rapidly under the assault of new approaches to cell-structure design. Intel radically trims size with a new contactless single-transistor cell. At the upcoming International Electron Devices Meeting, company researchers will describe their Flash Array Contactless EPROM (FACE) technology, which reduces the area of the ETOX cell by 55% to 8.4 μm^2. That reduction is based on 1.0-μm design rules. At 0.8-μm, the cell can be almost halved again to 2.48 μm^2.

Also at IEDM, Toshiba Corp. will show a NAND-structured memory cell of only 2.3 μm^2 (0.6-μm design rules). The cell can achieve 16-Mbit and larger flash memories, Toshiba says. Mitsubishi Corp. has achieved a single-transistor, stacked-gate cell of only 3.6 μm^2 in a 16-Mbit flash. It uses 0.6-μm design rules and achieves 5-V-only programming and erasure by a unique negative-gate-biasing erasing condition. ■

3

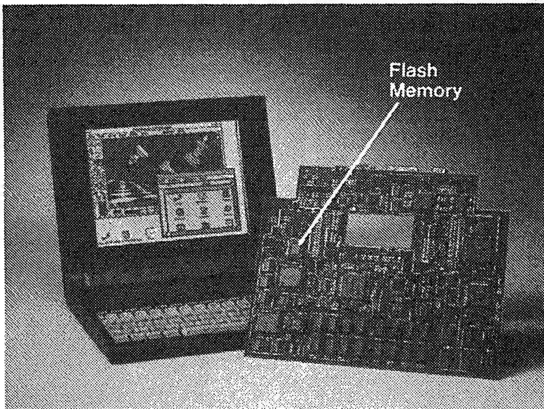
LAPTOP VENDORS JOIN THE FLASH BANDWAGON

AIRIS'S VH-286 USES FLASH TO STORE BIOS, BUT THAT'S JUST THE BEGINNING OF THE APPLICATION POSSIBILITIES

BY JACK SHANDLE

MEMORIES

WHEN THE founders of Airis Computer Corp. left Zenith Data Systems in 1988 to start a new notebook-computer firm, they knew, as all entrepreneurs must, that their product strategy had to stand out in a crowd. Flash memory technology (see p. 44) will play a major role in Airis's strategy when the first VH-286 computers start rolling off the production line this month. But Airis is unlikely to be alone for long.



INSTANT UPDATES

The BIOS in Airis's laptops is stored in a bank of flash memory. With TeleROM, BIOS can be updated by modem.

Flash is versatile—it lets clever systems houses such as Airis play with innovative marketing techniques as well as advanced technology. For Chicago-based Airis, the bright idea is TeleROM, says Steve Valentor, engineering vice president. All Airis computers have built-in modems, and by combining that capability with a bank of flash memory that stores system BIOS, Airis can offer users a highly desired feature: instantly updatable BIOS. Simply by dialing into Airis's bulletin board, users will be able to update their BIOS for a nominal charge.

Software-updatable BIOS ensures compatibility with the latest features and software. There are, for example, undocumented features in IBM Corp.'s VGA graphics specification, says Valentor. As these are revealed and utilized in new applications software, Airis users will stay compatible with a phone call.

Airis dedicated 128 Kbytes of flash to BIOS updating: 32 for system BIOS, 32 for video BIOS, and 64 to a program to update the BIOS. Patents have been requested for the updating scheme. In particular, provisions must be made for the possibility that system power may be lost during the BIOS update. "You have to be sure you have enough BIOS available at all times to boot the system," Valentor says. Airis purchases its flash chips from Seeq Technology Inc., San Jose, Calif., because they offer a sector-erase feature that helps implement the fail-safe updating procedure.

Storing BIOS is just the beginning for flash applications. John Wharton, a contributing editor to the *Microprocessor Report*, a Sebastapol, Calif.-based newsletter, says there is more to come. Conventional PC-memory systems are organized on three-levels: rotating mass storage, dynamic random-access memory, and static RAM cache, he says. Each level adds expense in the form of control logic, interconnects, access time, and reliability. "If executable programs and data are all already on-line in moderately fast memory," he says, "why copy them to an intermediate DRAM first? As larger caches migrate into the central processing unit, the performance characteristics of external memory become less critical." Several companies are hard at work leveling the conventional three-tier memory architecture by means of flash-based "silicon disks."

Among them is Psion Inc. The Watertown, Conn., company employs small flash-based modules as replacements for floppy disks. And at least one company—SunDisk Corp. of Santa Clara, Calif.—is building a flash-based storage system to replace Winchester drives. Flash could even be used to store applications software, but cost and reliability in massive read/erase/write environments continue to be inhibitors to widespread acceptance.

Besides the advantages to laptop and notebook end-users, flash offers considerable advantages in manufacturing, says John Wagner, manager of Zenith Data Systems' Portable Products Development Group, Mt. Prospect, Ill. Although Zenith has not yet implemented flash, it is studying the technology closely in part because of manufacturing issues. "Producing a machine requires several stages of firmware development," he says, "and using flash memory would let us implement the latest version in the final stages of manufacturing. You can also include the latest BIOS and system configuration on a floppy."

Psion is already using flash as a floppy-drive stand-in. Its Flash Packs use 1

© 1990 INTEL CORPORATION
Reprinted with permission from Electronics Magazine, November 1990

Order Number: 295058-001

Mbyte of Intel Corp. flash chips and measure about 1 by 2 in., says Brian James, marketing support manager. Flash Packs can be used as rewritable storage or as a medium for applications programs. Right now, users must download applications programs from a desktop PC to the Psion notebook computer, but licensing agreements with major software houses should be in place by 1991 that will make memory-card versions of popular MS-DOS software available. Price is high: $650 for a 1-Mbyte card.

The most controversial application for flash is mass storage. Whether it will one day supplant Winchesters depends on the balance of the technology's strengths and weaknesses. For notebooks, laptops, and portables, flash will save valuable real estate, says Zenith's Wagner. But just as important is its form-factor flexibility.

"Flash devices can fit into unusual space configurations within the cabinet. You don't have to lock up space for a drive early in the design cycle," he says. They are also immune to the read/write-head failures of rotating media, he says, and are at least 10 times faster than rotating media.

FLASH'S BIG DRAWBACK IS cost: a 20-Mbyte flash-based storage device would cost an outrageous $4,800 at today's prices, says Airis's Valentor. That compares with an OEM price of $300 for a 2.5-in Winchester. But Valentor points out that the access-time differential between silicon and rotating media must be traded off against the cost differential.

"Using data-compression techniques, you can reduce the number of chips needed to store a given amount of data and still deliver performance better than or equal to hard-disk storage," he says. "In the next two years, we could start to see flash drives at about twice the cost of rotating memory, and that will make flash's form-factor and performance advantages more attractive."

Flash's advantage in power consumption speaks directly to the concerns of portable PCs. Somewhat surprisingly, power consumption is "about a wash in access mode," says Wagner, but in nonaccess mode, the disk continues to spin while flash goes to near zero. "You have to look at the peak voltage [12- or 5-V erase, depending on vendor] and how many times you use it," Wagner says. "Changing BIOS does not happen often, so it is not an issue there, but in mass storage, erases happen much more frequently."

Though the biggest market inhibitor is cost, flash also has a reliability issue to deal with, and Airis is taking a wait-and-see posture on mass storage. "At this point in time," says Valentor, "I do not believe the parts have the number of reprogramming cycles needed for a hard-disk replacement, but there is not a fundamental inhibitor to longer life, and I expect to see their longevity improve." Intel's chips lead the pack with 100,000 erase and reprogram cycles, but Valentor points out that a portable or laptop running a spreadsheet program, for example, reads and writes to the same portion of the disk. This means that in a flash-based device some chips would be used much more than others. ■

3

intel®

ARTICLE REPRINT

AR-495

November 1990

Store Data in a Flash

© 1990 INTEL CORPORATION
Reprinted with permission from Byte Magazine, November 1990

Order Number: 295059-001

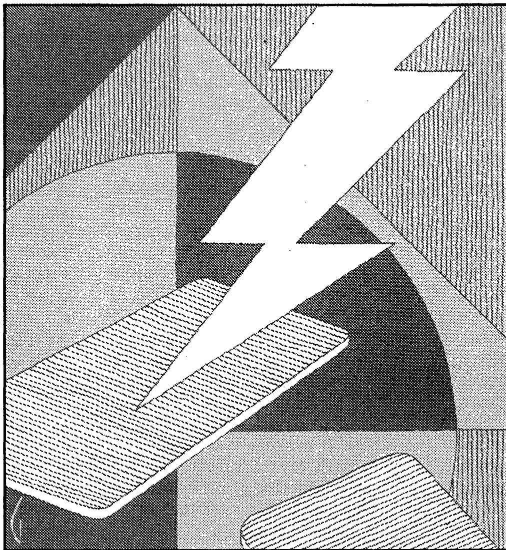
Store Data in a Flash

Flash-memory ICs offer new options for personal computer storage

Walter Lahti and Dean McCarron

Normally, you'd think of a flash flood as a natural disaster, something that could pick you up and carry you away. But the flood of flash memory that is about to reach the personal computer world will be a positive event. It will carry the power to expand the reaches of personal computing.

Flash memory is a nonvolatile memory IC. Born of the blending of EPROM and EEPROM, the flash IC is functionally and technologically the offspring of these parents (see the text box "Do You Remember?" on page 312). It is reportedly named for the speed with which it can be reprogrammed.



While flash and EPROM memory cells usually contain a single transistor, a DRAM cell typically contains a transistor and a capacitor, an EEPROM cell two transistors, and a static RAM (SRAM) cell four or six transistors. Obviously, the more cells, the more real estate (silicon) a memory requires. And real estate is always expensive.

Advantages of Flash

Flash's two significant attributes, nonvolatility and DRAM-like speed, are ideal for solid-state "disk" drives. Flash-based disks are very fast compared to most available disk drives (see figure 1). In 120 nanoseconds, you can access data stored in flash memory, while it takes 15 to 30 milliseconds to access data stored on today's typical hard disk. In some implementations, such as in portable computers, the speed advantage of flash over disk drives is even greater.

Today, a personal computer's hard disk drive is one of its most power-hungry components. When you use a desktop machine, you may not notice this power consumption. But the power a battery-operated portable can supply is limited—and hard disk drives use up that power quickly. Most portables today require fairly sophisticated power management facilities to extend the amount of time the machine can be used.

A portable's power management facility often turns off the hard disk drive if it isn't being used. While this is great for extending a portable's limited battery life, it is terrible for performance. When the power comes back on, the disk drive's motor can take several seconds to bring it up to speed before disk I/O can begin. A flash-based disk needs no warm-up. When you turn on the power, the data is immediately available. With no waiting, you experience no loss in performance.

In addition to achieving power savings from an "instant-on" flash disk, you also realize savings from not having to operate power-hungry motors and servos. A 1-megabyte flash disk requires a maximum of only 1.2 watts while operating.

ILLUSTRATION: JOE GAST © 1990

Do You Remember?

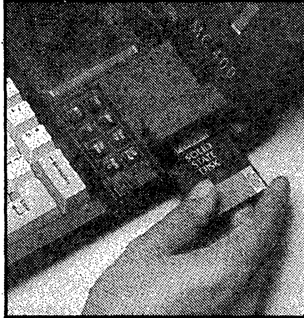
There are two kinds of memory: volatile and nonvolatile. Memory such as DRAM is called volatile if it forgets what it had stored when you turn off your computer's power. Memory such as ROM is called nonvolatile if it retains its data whether or not your computer's power is on. As all users who have ever turned off their computers before saving files to disk can tell you, the DRAM used in your personal computer to store programs and data cannot retain information without power.

DRAM, however, is reprogrammable; the information it contains can be changed. When you load a new file, the new information replaces the old. ROM, though, is not reprogrammable—the programs and data in ROM are permanent, and you can't change them.

In the early 1970s, the only semiconductor memory available was DRAM, its cousin static RAM—which is also volatile—and ROM. The choices open to computer designers were using memory that was reprogrammable but lost information without power, and using memory that always retained information but could never be changed. What designers really needed was memory that could be reprogrammed in the system and that also retained its contents when the power was off.

A few years after DRAM became available, a new kind of memory known as electrically programmable read-only memory, or EPROM, was introduced. EPROM is reprogrammable and nonvolatile. But it has one drawback. In order to reprogram EPROM chips, you have to remove them, expose them to high-intensity ultraviolet light for as long as 20 minutes, reprogram them, and then replace them in your computer. Thus, EPROM fell short of being the ideal memory. Today, because vendors find them easier to program, EPROM chips are largely used as replacements for your personal computer's ROM.

Electrically erasable programmable read-only memory, or EEPROM, was introduced in the late 1970s. EEPROM



Psion uses four Intel 1-Mb flash-memory ICs in its credit-card-size solid-state disk.

(like EPROM) is reprogrammable and nonvolatile, and it can also be easily reprogrammed within the computer.

Still, there are drawbacks. EEPROM is slow and expensive and doesn't hold very much data. Today, you can store 1 megabit of data in an ordinary DRAM chip. You can access the data in 80 nanoseconds, and it costs $5. In contrast, it takes 150 ns to access a 1-Mb EEPROM, which costs $265.

In the mid-1980s, Toshiba Semiconductor invented flash memory. About the same time, Intel and Seeq Semiconductor were also working on flash memory. While each manufacturer built its flash memory differently, they operate similarly.

Like both EPROMs and EEPROMs, flash memory is nonvolatile and reprogrammable. But it has none of the faults of these other types of memory. Unlike EEPROM, it is inexpensive: Today, a 1-Mb flash memory costs about $15. Unlike EPROM, flash memory can be reprogrammed electrically while it is embedded in the system—either by you or via system software.

Still, one drawback remains. With DRAM, you can change a single bit at a time, but with flash memory, you can change only a sector (consisting of multiple bytes) at a time. While constraints of sector-level reprogrammability prevent it from replacing your computer's DRAM, flash memory is well suited to other applications.

The type of storage that hard and floppy disk drives provide resembles that of flash memory. Disks are nonvolatile—they hold onto data with or without power. And disks are reprogrammable—you can change the files whenever you want to. The similarities between flash memory and disk storage led to the building of "disks" based on the concept of flash memory.

A flash disk isn't a disk drive at all; there are no disks or moving parts. A flash disk is a set of flash-memory parts mounted in a credit-card-size package that acts as a hard disk. This same set of parts could be mounted on a board inside a machine. The difference between the two is that one is removable storage and one is fixed storage. A flash disk emulates a disk drive.

A flash disk is built from one or more flash-memory ICs and some controlling logic devices. For example, to build a 512K-byte flash disk, you could connect four 1-Mb flash-memory ICs and place them on a small card. Psion has used this principle with its flash disk (see the photo).

Flash disks operate fairly simply. At the hardware level, the computer simply sends digital read or write signals to the disk with the address of the information. If it is a read signal, the disk responds with the requested information. If it is a write signal, the disk takes information from the computer and stores it.

In addition to flash-disk hardware, you also need software to manage the files on a flash disk. This file-system software handles creating and deleting files, changing the file sizes, and formatting the flash disk. Microsoft has worked with Intel to create the Microsoft flash file system, a standard MS-DOS-compatible flash-disk interface that makes it much easier for vendors to use flash disks in their computers.

The lowest-power hard disk drives today require about 3 W.

The fact that flash-based disks have no moving parts carries with it yet another advantage—reliability. While hard disk drives have become remarkably tough, on occasion they still do crash.

Flash-based storage is very reliable because a flash disk is as tough as the rest of the electronic hardware in a personal computer. It takes a lot for a flash disk to fail: The flash memory must be damaged physically, through destruction of the device package, or electrically, by an extreme electric shock or a power spike.

Disadvantages of Flash

Flash memory's extremely high speed, low power, and high reliability would seem to make it the ideal storage technology. Unfortunately, there are two significant drawbacks to flash disks. The most severe limitation is its cost. A conventional 40-MB hard disk drive costs about $320, or $8 per megabyte. Today, a 1-megabit flash IC costs $15. Eight flash ICs are needed per megabyte of flash disk, making a flash disk cost about $120 per megabyte.

Thus, you would have to pay about $4800 for a 40-MB flash disk, or about 15 times what an ordinary hard disk drive would cost. Because of this present inequality, the first mass-produced flash-based disks probably will store less than 40 MB. In the future, flash-based disk prices will certainly decline, making large amounts of flash-disk storage more affordable. In a few years, you should only have to pay about $600 for a 40-MB flash disk.

The other problem with flash disks is that they can't compare with hard disks in density. The highest-density flash memory available today stores 2 Mb per IC—you would need 160 of these ICs to produce a 40-MB disk. Like all memories, flash memory is expected to grow in density, so eventually far fewer ICs will be needed.

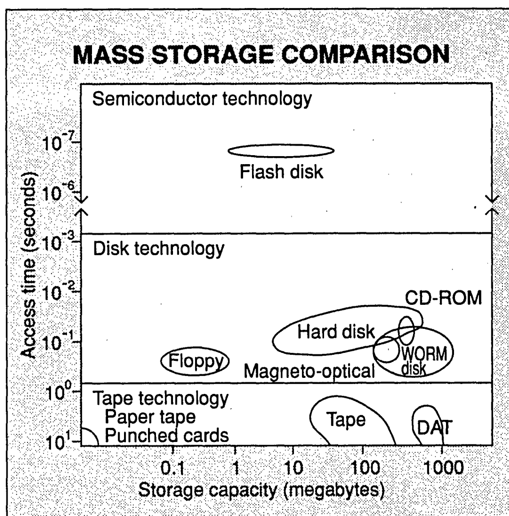


Figure 1: *Flash disks are 125,000 to 250,000 times faster than today's hard disk drives. However, they are limited to up to 40 MB in capacity, whereas hard disk drives can store from 5 MB to 1 gigabyte.*

Two Flavors

Manufacturers currently offer flash devices in two programming flavors: those that require a 5-volt power supply, and those that require a 12-V supply. With both erasure and programmability possible at 5 V, only one power supply is required at the system level. The benefits of this feature are reduced system-component cost and space savings. Thus, flash is ideal for portable-computing applications.

The 5-V flash cell is generally a modified two-transistor (or split-gate) derivative of EEPROM and is packaged with a different pin-out than the 12-V varieties. Five-volt programming lets a system interface with the device in much the same way it would with SRAM. Therefore, for some applications, a flash device can replace SRAM, particularly in systems that use SRAM with battery backup.

While both 12-V and 5-V flash memory can be used as an SRAM replacement, the 5-V feature becomes more desirable for portable equipment where no external 12-V power is available and the addition of a 12-V power supply is not feasible.

Ideal for Laptops and Palmtops

Laptop and notebook computers are the ideal applications for flash disks. With current hard disk drives, you must carry around heavy batteries, deal with short amounts of work time, or suffer from hard disk drives operating at floppy disk drive speeds. Flash disks will answer all your critical needs for laptop and notebook computers by providing speed, rugged construction, and low power consumption.

You can also benefit from flash memory in other implementations. Flash will let you update your laptop's ROM with the latest versions of DOS, or any other operating system, whenever you want to. Laptops save space on disks and in RAM by placing the operating system in ROM. The problem with this is that you can't update the operating system without replacing the entire ROM—an expensive proposition. Thus, laptops often use old but reliable versions of DOS. Using an old version of DOS may mean that your computer won't need a ROM replacement in the near future, but it may not run recently written programs, either.

One thing lacking in palmtop computers, such as the Poqet PC and Atari Portfolio, is small, convenient mass storage. Without any optional peripherals, their storage is limited to programs on ROM cards and memory-expansion cards that lose their contents when they are removed. With flash-based memory cards, you can put your own programs and data onto the card, modify them at will, and not worry about losing the information when you remove the card. These features make flash-memory cards the logical choice for the palmtop's missing "floppy disk drive."

Laser Printers

If you use a laser printer, you can benefit significantly by using flash memory instead of ROM. In laser printers, ROM stores programs and fonts. ROM replacements are expensive because printer-control language programs have become large and are subject to frequent upgrades and improvements. Using a laser printer equipped with flash memory instead of ROM to store control-language programs, you can reprogram your printer's control language yourself at no cost and without replacing any ROM.

Currently, laser-printer font-storage options leave much to be desired. You have three choices. You can download a font to the printer each time it is needed, wasting your time and the laser printer's memory. You can place a font in a ROM cartridge and plug it into the printer, but you are limited to a selection of only a few fonts out of the hundreds available. Or you can store a font on a dedicated hard disk connected to the printer.

But when you use flash memory inside your printer, you only have to download a font once and it remains in your printer until you choose to delete it from the printer's memory. Because you decide which fonts are stored in the printer's memory, you can really personalize them according to your preferences. You no longer have to buy cartridges that come with a half-dozen fonts just to get the one font you need.

Fabrication Techniques

Flash devices are manufactured using designs and processes similar to those used for EPROM and EEPROM, so the technology is evolutionary rather than revolutionary. Because manufacturers have dealt with similar products, they will be able to climb the learning curve much more rapidly than if the technology were completely new. Thus, vendors planning to produce flash memory should be able to attain manufacturing costs close to, but perhaps not equal to, those enjoyed by EPROM.

However, flash devices are a bit more complex and more silicon-hungry than EPROM devices. The most common flash chip is an array of single-transistor memory cells and looks much like an EPROM (see figure 2). It is slightly larger than an EPROM of equal density to allow for the command port and peripheral circuitry that supports the in-system rewrite function and provides an on-chip processor interface.

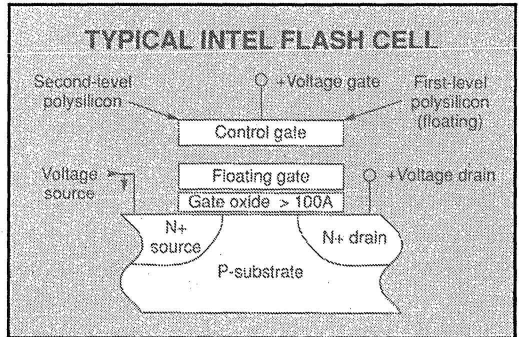


Figure 2: *A flash-memory cell is basically one memory bit (on or off). An array of up to 4 million flash-memory cells can be connected to form a flash IC.*

The typical EEPROM chip is made of an array of two-transistor cells to enable bit-level erase/reprogram. For any given density, it requires much more silicon than either the EPROM or flash cell. Because a major cost determinant in any IC is the silicon required, the EEPROM is a more expensive part.

Erasing and Reprogramming

In terms of reprogrammability, the flash IC falls somewhere between the traditional EPROM and EEPROM (see figure 3). A major difference between flash memory and EPROM is that flash does not require ultraviolet light for erasure, as does the traditional EPROM. While flash resides in your system, you can electrically erase it in much the same way as you would an EEPROM.

The energy needed to discharge or erase the gate in a typical EPROM is derived from UV light, a requirement that makes it difficult and time-consuming to erase an EPROM. In a typical flash IC or EEPROM, energy resident in the system can be used to erase a group of memory cells or the entire chip. This feature makes it easy and fast to erase a flash IC in the system.

You generally cannot erase a flash IC on a byte-level basis as you can with the EEPROM, but some flash ICs can be erased on a sector-level basis. Flash ICs are usually reprogrammable by hot electron injection, a solid-state physics process that uses the energy in the system. It is possible to program on a byte level, but because it is not possible to erase on a byte basis, reprogramming is limited to sector or the entire chip.

Because the flash device does not require UV light for erasure, the chip does not need to be housed in an expensive ceramic window package such as that required for an EPROM. Therefore, flash is also an excellent candidate for surface-mount technology.

The advantage of surface mount is that there is less distance between the device and the board. This reduction can lead to improved reliability, better system performance, and higher board density, as well as reduced cost. Also, the flash device can readily be packaged in memory-card configuration and handled as if it were a floppy disk, which is important to the portable computer world.

The total cost of using flash memory can be considerably lower than that for EEPROM and, with some applications, close to that for EPROM—about $6.50 for a 1-Mb EPROM versus over $250 for a similar-size EEPROM. On a comparable device-density basis, flash memory's $15 average selling price is much lower than the EEPROM's and greater than the EPROM's. With flash, application solutions are possible that would be impractical with either the UV-light erasure EPROM or the pricey EEPROM.

The law of the semiconductor jungle is that over time, all device types see improved performance and reach greater density levels. At the same time that silicon real estate is minimized, costs are significantly reduced. By 1994, the cost of a megabyte of flash memory is expected to move from its current level of about $120 to about $15.

Flash in the Pan?

Unless developers are able to overcome the current limitations of flash disks, you will probably continue to use hard disk drives on your desktop computer for mass storage. Hard disk drives are inexpensive and fairly reliable, and they can

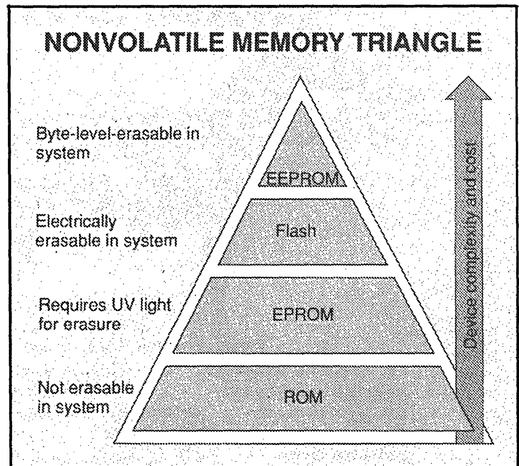


Figure 3: *Technology trade-offs for semiconductor nonvolatile memories. As programming flexibility increases, so do device complexity and cost.*

Laptop and notebook computers are the ideal applications for flash disks.

store plenty of data. Although single-user personal computers will probably continue to include hard disk drives, eventually network servers will probably offer both hard disk drives and flash disks on-board.

On many servers, you frequently access files, such as programs, that are rarely changed. Flash disks are ideally suited to perform this service. You can store seldom-changed program files on flash disks, relieving the burden on the server. By doing so, the server's response to program load requests will be far faster than if the files were stored on a hard disk.

Flash memory combines the advantages of an EPROM's low cost with an EEPROM's ease of reprogramming. These advantages will allow flash memory to make significant contributions to personal computers. Portable computers will be the first to benefit from this new technology, as flash-based disks increase their speed, operating time, and ruggedness. ■

Walter Lahti and Dean McCarron are vice presidents of In-Stat (Scottsdale, AZ), a company that provides market research for the electronics industry. They can be reached on BIX c/o "editors."

3

intel®

ARTICLE
REPRINT

AR-497

January, 1991

Designing With Flash Memory

By
MARKUS A. LEVY

© INTEL CORPORATION, 1991
Reprinted with permission from Circuit Cellar INK

Order Number: 295061-001

FEATURE ARTICLE

Markus A. Levy

Designing with Flash Memory

Is There a New Alternative to EEPROM and SRAM?

Flash memory (in general) is capturing market share from other memory technologies. It is replacing EPROMs that were traditionally used for code storage because, along with equivalent nonvolatility, it also allows in-system updates. Battery-backed SRAMs that once were used for data acquisition, parameter storage, and even solid-state disks are now targets for the inherently nonvolatile and lower-cost flash memory devices. Many notebook computer OEMs conclude that low power, light weight, and reliability are most easily obtained with a completely solid-state machine. Flash memory has achieved a density ramp from 256K bits to 2 megabits in two years. Combined with a special flash file system from Microsoft, flash memory can even replace the mechanical disk drive.

3

With the design described in this article, you have a platform demonstrating flash memory's functionality and flexibility. Applications range from data acquisition through an I/O port, to a DOS-compatible, solid-state disk. But first, a few essentials.

EPROM AND BEYOND

Derived from an EPROM process base, Intel's ETOX-II flash memory technology has similar nonvolatility, reliability, and array densities. In fact, the flash memory cell is identical to the EPROM structure, except for the thinner gate (tunnel) oxide. This is where the similarities end. The thinner gate oxide enables flash memory to be erased and reprogrammed in-circuit, typically 100,000 times. The name "flash" is derived from its one-second chip-level erase and microsecond-level byte-write times versus the slower, millisecond-level byte-write times for conventional EEPROMs.

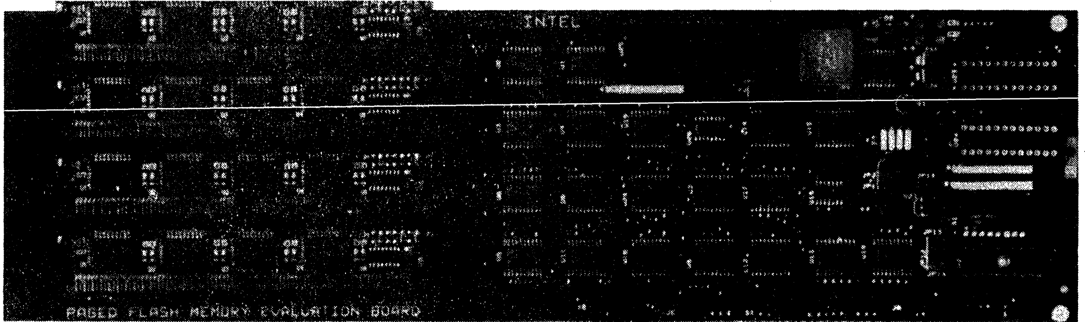
Flash memory devices have a command register architecture that provides a micronrocessor-compatible write interface. Erase, program, verifications, and other operations are initiated by issuing the proper command to the flash memory device. Twelve volts must be applied on V_{pp} for the command register to respond to writes and execute the operation. The 12V requirement doubles as an added security feature for data integrity. If you are familiar with other memory subsystems, designing with flash memory is as simple as any other technology.

In addition to discrete components, Intel offers flash memory in SIMM and IC memory card formats. This design will use these modules, so I've included some pertinent information. The 512K-byte x 16 Intel Flash SIMM (SM28F001AX) is based on an 80-pin JEDEC standard that accommodates density upgrades and presence detect (a hard-wired ID that indicates SIMM density and speed). The eight 1-megabit flash memory devices on this module are paired up as high and low bytes. They are selected using the SIMM's write enable high and low (*WEH and *WEL) signals.

Intel's IC memory card adheres to the Personal Computer Memory Card International Association (PCMCIA) standard. This standard specifies physical, electrical, information structure, and data format characteristics of the card. Most impressive is the size, measuring 85.6 mm x 54.0 mm x 3.3 mm. Its 68-pin interface includes 26 address lines used to directly address 64 megabytes. All buffering and chip-level decoding is contained within the card, greatly simplifying the board-level design. Intel's flash memory card is available with one and four megabytes. These cards will continue to grow in density, becoming more and more competitive as disk drive replacements.

MEMORY METHODS

Three fundamental addressing methods can be implemented when interfacing a flash memory array to a system bus: linear, I/O, and paged. Each method has its benefits and drawbacks. A linearly addressed memory array is mapped directly into the sys-



tem's memory space and allows the highest performance. However, the memory array would be insufficiently small in systems having limited memory space, as with the 8086. But this method is practical in an 80386 (or other 32-bit processor) family system with a large memory space available.

An I/O-mapped memory array uses one address—an I/O port—to transfer data. This method requires the least amount of system memory space but also yields the lowest performance.

A page-mapped memory system is a hybrid of these two approaches. It allows a very large memory array with a minimal system interface. A page is a moveable window into the total memory array. It selectively opens different portions of the array by writing a page number to the decoding circuitry. This page ranges in size from 8K to 64K bytes. Analogous to a cache, a larger page size requires less frequent switching. Although switching pages represents a performance degradation, this can provide the optimal balance between performance and memory space availability within the system.

Our design is based on this page-mapped technique. A 64K-byte page size reduces the decoding circuitry. The PC/AT has been chosen as the execution platform, but with minor modifications to the control signals, any microprocessor environment can be used. Before beginning this design, it would be helpful to reacquaint yourself with the basics of the AT I/O channel bus.

The subsystems within this design (Figure 1) are the memory decode circuitry, I/O and its associated decode logic, and a 12V generator for V_{pp}. The Intel flash memory resides in four SIMMs. The board handles an upgrade path to 16M bytes, based on 4M-byte SIMMs.

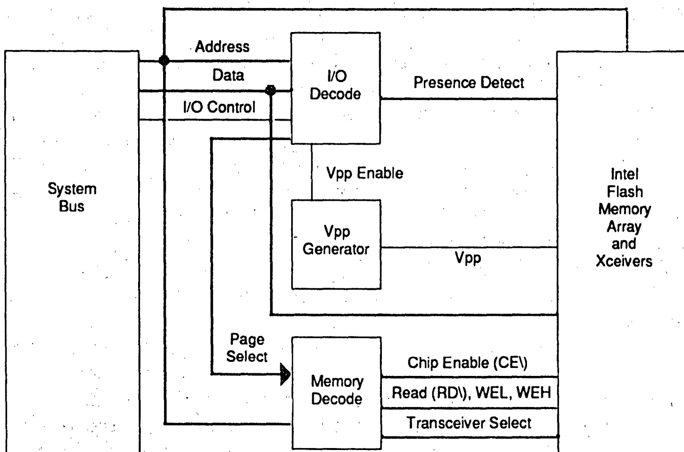


Figure 1—*The subsystems in a flash memory board design include memory decode, I/O, and a 12V generator.*

ADDRESS DECODING

Flash memory addresses can be decoded in one of two ways: row-column and conventional decoding using separate chip enables. The row-column approach of Figure 2 is appropriate if you are motivated to reduce board traces. In row-column decoding, rows are Output Enables (*OE), Write Lows (*WRL), and Write Highs (*WRH); and columns are Chip Enables (*CE). Although the SM28F001AX uses only four chip enables, eight are provided since four-megabyte SIMMs could consist of sixteen 2-megabit flash memory devices.

Page selection, discussed in more detail later, is accomplished by writing the page number through an 8-bit I/O port to a latch. This will allow access to 256 64K-byte pages. Page signals, P0–P2, are directly connected to A15–A17 on each SIMM. They decode pages on the device level. The row-column signals are derived by decoding page signals P3–P7. They enable components on the SIMMs.

The row-column approach, however, suffers from simultaneous selec-

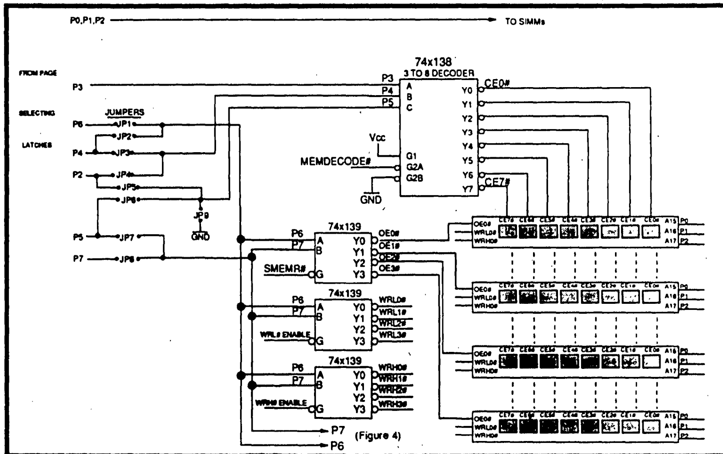


Figure 2—*Row-column addressing can be used to reduce board traces.*

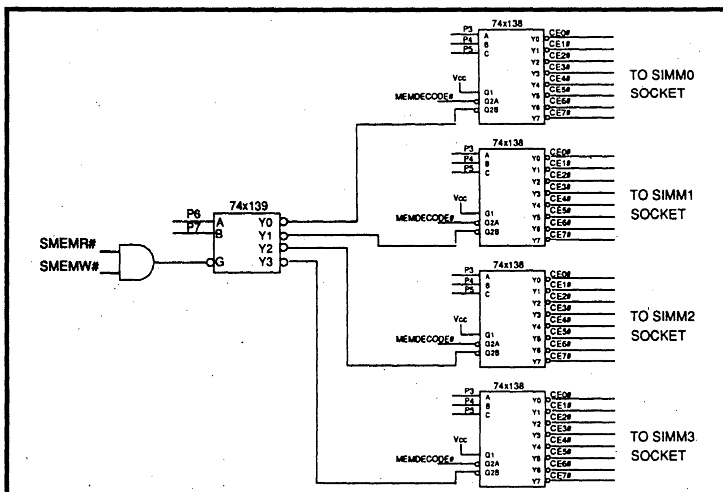


Figure 3—*Using separate chip enables trades off board complexity for low power usage.*

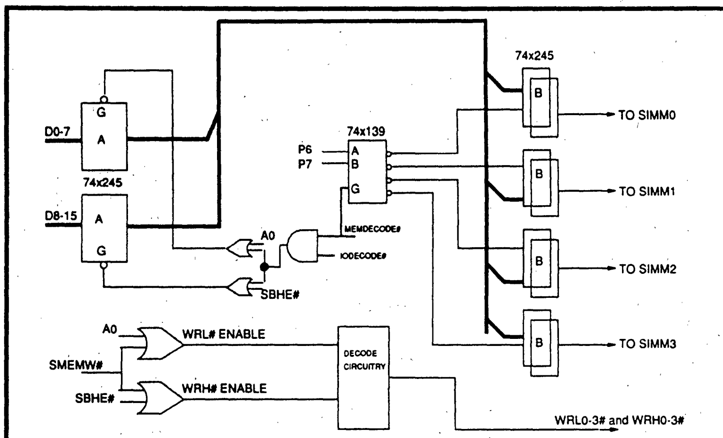


Figure 4—*A simple buffering scheme is used to connect the memory to the system bus.*

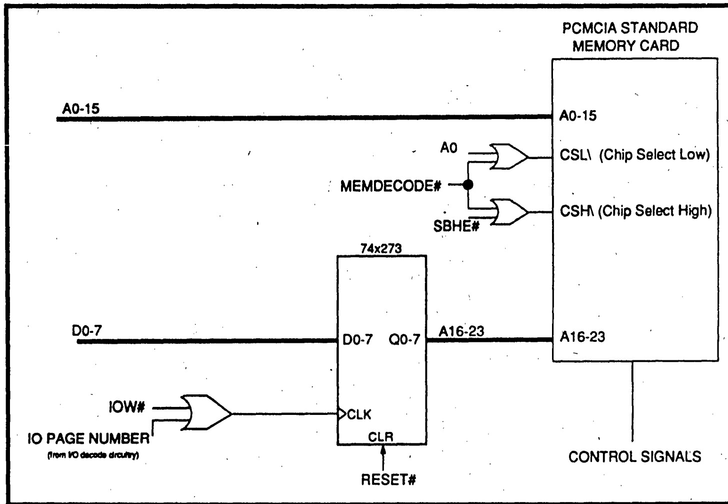


Figure 5—*Moving some buffering onto a separate memory card reduces the amount of necessary interface hardware.*

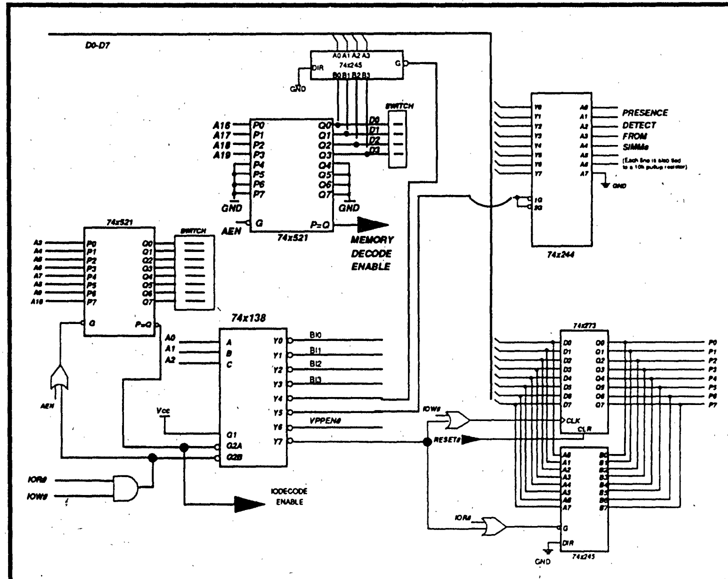


Figure 6—*DIP switches are used to set the board's address in the PC's I/O space.*

tion of eight parallel device chip enables. The conventional, separate chip enable decoding method has lower power consumption. Using this method (Figure 3), upper page signals, P6 and P7, decode which SIMM is selected.

Page selection becomes more complicated when accounting for density upgrades. To understand this there are two things to keep in mind: each SIMM socket handles a maximum of 64 pages, (i.e., 4-megabyte SIMMs), and A17 is used on a 2-megabit device but is a "No Connect" on the 1-megabit part. A "No Connect" pin implies that page selection will not be contiguous with SIMMs less than four megabytes in size. Accommodating noncontiguous pages increases software complexity. Regardless of the page decoding method, a jumper scheme rearranges the page signals and accommodates density upgrades.

According to Figure 2, the jumper settings are as follows:

1-MByte SIMM—J7, J2, J9, J4
2-MByte SIMM (16 x 28 F010)—J7, J1, J3, J5
2-MByte SIMM (8 x 28 F020)—J7, J1, J9, J3
4-MByte SIMM (16 x 28 F020)—J8, J1, J6, J3

Figure 4 shows the buffering required for system bus interfacing. The PC I/O channel bus is limited to two TTL loads on any one line. The "A" transceivers are connected directly to the I/O channel bus. Additionally, each SIMM has its own pair of "B" transceivers to reduce capacitive loading that results from tying more than eight flash memory devices together.

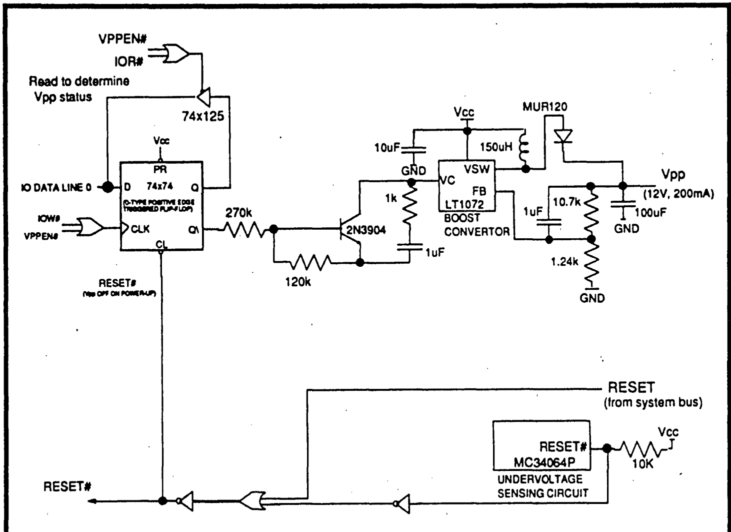


Figure 7—*A regulated boost converter made from Linear Technology's LT1072 supplies the 12 V necessary for programming.*

SIMPLIFIED DECODING HARDWARE

You are not limited to using SIMMs in your design. The same basic techniques will work for discrete components or other module types, such as memory cards. The IC memory card provides the simplest solution with its integrated decoding. Writing the page number through an I/O port to a latch accomplishes the same page selection as before, but now the "data" translates directly into memory card address inputs A16–A23. This eliminates the entire decoding structure shown in Figure 2 or 3. The IC card itself also contains the "B" transceiver buffering. This results in the hardware reduction shown in Figure 5. You will want to purchase a spring-loaded connector from Fujitsu, AMP,

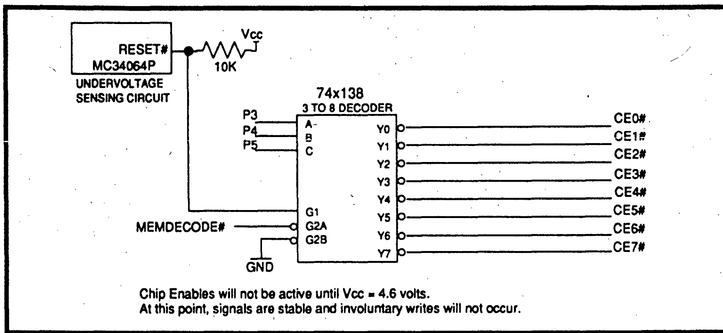


Figure 8—*An MC34064P undervoltage sensor is used to disable all writes to memory when the power starts to fail.*

or ITT Cannon to use with the memory card, laying out your board so that the memory card can be retrieved out the back of your PC. Pushing the button on the connector ejects the memory card for data security or transport.

I/O PORTS ON THE PAGE MEMORY BOARD

The page-selecting I/O port mentioned is one of eight ports in this page memory board design. The 74*x*521 comparator's inputs begin with A3 to simplify the decoding circuitry for the I/O port addresses. This places the base I/O port on an even 8-byte boundary. Use discretion when setting the switch to avoid conflict in the AT environment. I recommend using I/O addresses between 300H and 318H because this area is assigned for prototype cards.

Listed in the order in which they appear on the 74*x*138 decoder in Figure 6, the eight I/O ports in this design are: BI0–BI3 (accesses the board's identifiers); the window address within the system's memory space; the SIMM's presence detect; the V_{pp} enabling register; and the page number reading and writing ports.

I didn't diagram the board's identifier circuitry because the implementation is extremely simple. It consists of four identical units made up of a 74*x*244 and an 8-input DIP switch. The four enables, BI0–BI3, from the I/O decoder connect to the corresponding unit's enable. The I/O space is scanned for the identifiers to locate the board.

The user-selectable page window address can be placed on any 64K-byte boundary within the DOS 1-megabyte range, but only the adapter ROM area between C0000H and E0000H will alleviate compatibility problems. Using additional inputs on the "Memory Decode Enable" 74*x*521 and an 8-input DIP switch allows window placement above 1M byte. Simply connect address lines A20–A23 to input pins P4–P7, respectively. The Presence Detect (PD) pins from all four SIMMs are wire ORed together into the 74*x*244. This eliminates having to read each SIMM's PD pins separately, but SIMMs must be installed with equivalent density and speed.

The page number is written and read through the same I/O port. Minimal circuitry ensures that the system powers up at page zero. The page memory board's *RESET signal connected to the CLR input of the 74*x*273 latch accomplishes this task.

V_{pp} GENERATION

Why is V_{pp} generation necessary on this flash memory board when the AT I/O channel already has a 12-V supply? True IBM-compatible PCs specify the necessary ±5% 12-V supply tolerance. However, some systems have wider 12-V supply tolerances. Local 12V generation via the circuitry described below ensures fast, reliable flash memory operation.

Linear Technology's LT1072 switching regulator, a 5-V–to–12-V feedback regulated boost convertor, is the heart of the V_{pp} generator (Figure 7). A 100-µF capacitor at the output handles up to 200 mA, necessary for software that programs or erases

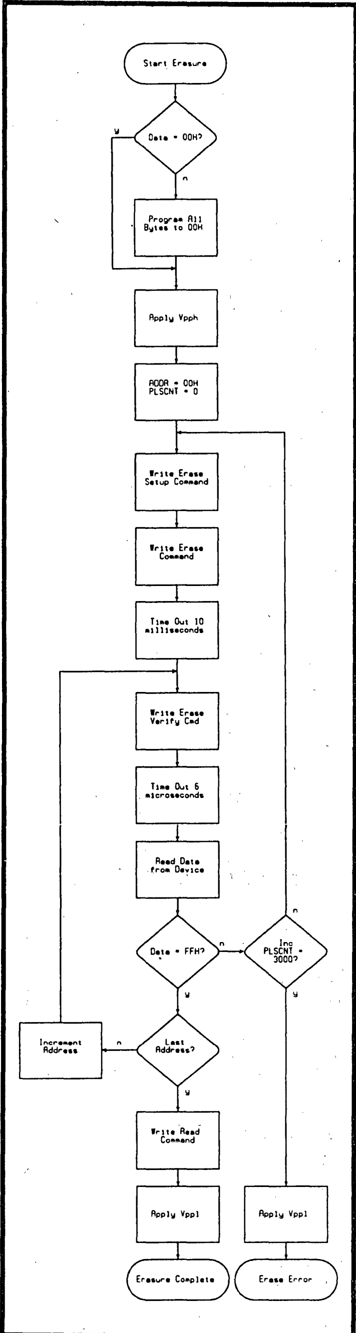


Figure 9—*The flash erase algorithm.*

eight flash devices simultaneously. Turning V_{pp} off when not in use conserves power, but this capacitance value requires approximately 100 ms to fully charge to 12 V. Reducing the output capacitance value and limiting the number of flash memory components accessed simultaneously decreases the ramp time. The diode, MUR120, prevents inductor current absorption from the charged output capacitor. During system power-up, spurious noise may generate writes which are actually the sequence of flash memory commands that initiate erasure or programming. Disabling V_{pp} until voltages stabilize provides power-up protection. The Motorola MC34064P is an undervoltage sensing circuit that begins functioning when V_{cc} is above 1 volt. Between 1 and 4.6 volts, the MC34064P's *RESET output or AT system RESET clears the 74*x*74. While the 74*x*74 remains cleared (or *Q = 1), the 2N3904 is on, the VC input of the LT1072 is 0 volts, and the VOLTAGE SWITCH (VSW) output is off. Writing a one to the V_{pp} enable latch forces *Q low, turning off the transistor. This puts the VC input at 5 V, and VSW output generates 12 V.

You do not need the circuitry just described if your system's 12-V supply meets the V_{pp} specifications. However, because software may accidentally (or coincidentally) generate a valid flash memory command to a flash memory address, install a switch to turn off V_{pp} when not in use. A low-resistance PFET (Motorola MTD4P05) performs this duty.

The possibility of spurious writes to the flash memory devices during power-up still exists. Again, the same undervoltage sensor (MC34064) solves this problem. The *RESET output becomes the 74*x*138 decoder's active-high enable. This controls the chip or write enables for the flash memory devices (Figure 8).

A FEW ADDITIONAL POINTERS

Ground *MEMCS16 so the PC/AT recognizes your board with a 16-bit bus width. The original design operated in both 8- and 16-bit systems. This flexibility is accomplished with additional decoding that multiplexes the high data bus onto the lower data bus. Again, the IC memory card automatically conforms to either bus width because the extra decoding is handled internally.

As with any circuit design, it is important to follow good design principles. For example: decouple power supplies with 0.1-µF capacitors between V_{cc} and V_{ss} of every device; and short board traces help minimize noise.

SOFTWARE

Hardware without software is like a computer without a processor. Therefore, understanding program and erase algorithms is the first step towards functional flash memory. Recall from our earlier discussion that operations on flash memory are software controlled using the internal command register architecture. I have included the complete algorithms (Figures 9 and 10). *[Editor's Note: Software for this article is available on the Circuit Cellar BBS and Software On Disk #18. See page 92 for downloading and ordering information.]* After working the Intel flash memory hotline, I would like to discuss the common mistakes.

The best piece of advice that I can give is *please* follow the algorithms. Many people unsuccessfully try their own custom versions. There are no cutting corners. Starting with the basics, let me elaborate a few points about the algorithms:

1) Flash is programmed to a binary zero by adding charge to the transistor's floating gate. Contrarily, charge removal erases the cell to a binary one. During the erase operation, the device is "flashed" as charge is simultaneously and equally pulled off the floating gate of every memory cell. The device must be preprogrammed to all zeros before erasure so an already erased cell is not further depleted of charge.

2) Prior to writing any command, switch on V_{pp} and allow ample ramp time for proper operation. Dropping V_{pp} below 7.5 volts from within any operation, places the device in the read mode. Similarly, abort an operation by issuing two consecutive reset commands (FFH) followed by the read command (00H).

3) Closely observe delay times to achieve the highest performance and

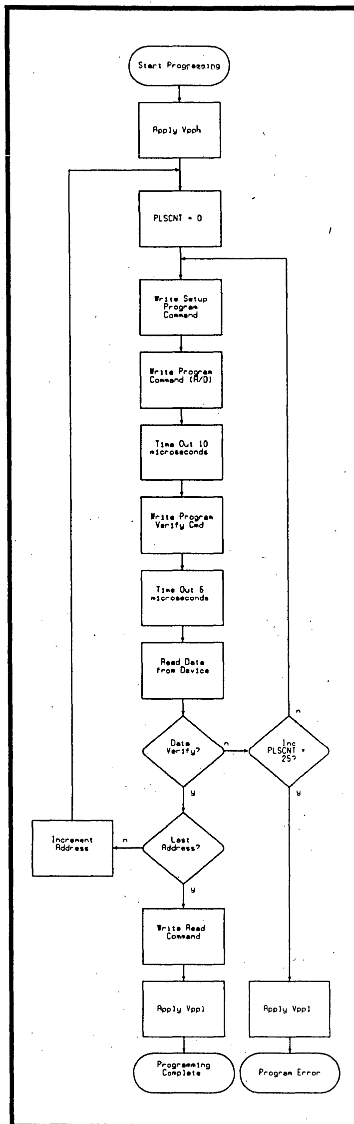


Figure 10—*The flash programming algorithm.*

reliability. Use the `STI` instruction in the software drivers to avoid system interrupts during these delays. Execute `CLI` once the corresponding verify command is issued.

4) The verify operation internally creates marginal conditions to ensure accurate and reliable results. The 6-µs slew time delay following the verify command allows the margin voltages to settle. In the verify mode, programmed data is guaranteed to be "permanent" when it matches the data being programmed.

USING YOUR PAGE MEMORY BOARD AS A DATA RECORDER

There is a nearly endless list of data recording applications, including digital imaging, digital photography, point-of-sale terminals, patient monitors, and flight recorders. The page memory board is appropriate for many data recording applications where an I/O port of the PC/AT accumulates data.

The programming algorithm demonstrates the byte programming capability of flash memory. In other words, once the device is erased, bytes reprogram randomly. In some recording applications, data is received in packets. A pointer determines where to begin programming the next free location within the flash memory.

Interleaving increases write performance by using the idle time during the 10-µs program delay. Addresses are offset such that each successive data byte gets written in different devices, looping back in time to issue the verify command. Reading the data back would be done in a similar fashion.

Word-wise, or parallel, programming of two devices provides an additional means of increasing write performance. Note that flash memory devices may program at different rates. Therefore, the original algorithm must be modified during the verify operation. If only one byte of the word has verified, the program command and data are sent again to the unverified byte. Mask the command sent to the device that verified to maintain word-wise programming. A mask is the substitution of a reset command for the program and verify commands. That way, the programmed bytes do not get further programmed on subsequent pulses.

The page memory board will perform these software techniques. However, first write the software that determines the location and capacity of the board. First, scan the I/O space for the board's identifier. The location of the first identifier byte is also the

base address for the eight I/O ports. Using the proper offset, read the I/O port that enables the base-memory address transceiver. For SIMMs, calculate the memory capacity by first reading the Presence Detect pins, followed by reading the individual flash memory device identifiers. Alternatively, read the Card Information Structure in the PCMCIA standard memory card for the capacity.

The preceding steps will confirm the basic functionality of your hardware. Practice programming the flash devices with data from a RAM-based array. For example:

```
; Software to read in ASCII test
; pattern to program into flash
DATA_ARRAY SEGMENT
STORE_IN_FLASH
    DB  'ASCII test pattern to
    DB  'be stored in flash'
DATA_ARRAY ENDS

CODE SEGMENT
    mov  ax,DATA_ARRAY
    mov  ds,ax
    mov  si,0
    mov  cx,size STORE_IN_FLASH

more_input:
    mov  al,[si]
    call FLASH_PROGRAM
    inc  si
    loop more_input
```

Now the fun begins. Once you've put your Flash Paged Memory Board together and tried it out, imagine converting it into a solid-state disk. It's been done using software drivers from Microsoft. In an upcoming article, we will discuss the structure of the Microsoft Flash File System and show you how to interface it to your board.

Finally, I would strongly suggest that before attempting this design, you obtain the appropriate flash memory literature from Intel (see the source box). ❖

SOURCE

Intel Literature: (800) 548-4725

Data sheets

SM28F001AX	1 M-byte SIMM	#290244
1MC001FL	1 M-byte IC memory card	#290399
4MC001FL	4 M-byte IC memory card	#290388
28F020	2 M-bit device	#290245

Application notes

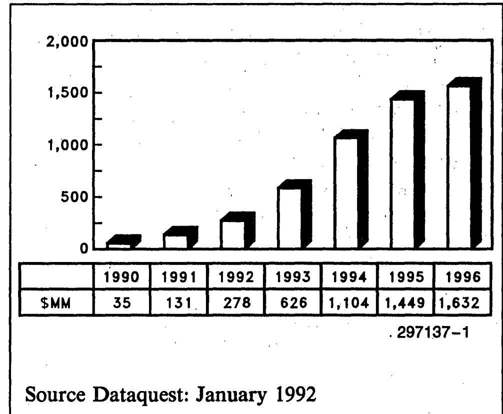
AP325, Guide to Flash Memory Reprogramming	#292059
AP343, High-Density Applications Using Intel Flash Memory	#292079

Markus Levy is an application engineer at Intel Corporation in Folsom, California, and holds a B.S.E.E from California State University. His specialties include software and hardware implementations of solid-state disks in portable computers. His favorite away-from-work pastimes include home remodeling, swimming, and parenting.

BOOT BLOCK FLASH: THE NEXT GENERATION

INTRODUCTION

Flash memory offers a whole new dimension to nonvolatile memory applications because of its high density and cost-effectiveness. Embedded systems, such as PC BIOS, hard disk drive controllers and laser printers, were among the first applications to utilize the easy update capability of Flash, allowing system differentiation and upgradeability. In 1992, Intel introduced a series of high-density Flash Memory Cards based on its 8-Mbit FlashFile™ component, products that are enabling truly portable computing capability with diskless, solid-state mass storage. Intel also added two flash memory products designed specifically to serve more traditional embedded and portable applications, offering both high performance and low power consumption options in an architecture specifically designed for these applications storage requirements.



	1990	1991	1992	1993	1994	1995	1996
$MM	35	131	278	626	1,104	1,449	1,632

Source Dataquest: January 1992

Figure 1

MARKET GROWTH DATA

In the four years since Intel first introduced its 256-Kbit flash devices, the worldwide demand has quadrupled annually, with 1992 market demand forecasted to exceed $270 million. Flash is expected to continue double-digit growth through the upcoming years while other memory technologies are relatively flat or increasing only slightly. By 1994, the flash market is expected to exceed $1 billion (Figure 1). The primary reasons for this growth are well acknowledged by customers: flexibility, nonvolatility, low power and cost-effectiveness. As a result of this seemingly insatiable demand for flash, the memory density treadmill that allowed DRAMs and EPROMs to double every 18 months, has accelerated faster than ever before. Intel's introduction of an 8-Mbit component two years after its 2-Mbit product is evidence of the flash memory's rapid advancement.

Intel's first generation of flash devices, ranging in density from 256-Kbit through 2-Mbit densities, were rapidly adopted into code storage applications which formerly used EPROM or EEPROM. In 1991, Intel introduced the first 1-Mbit Boot Block flash memory in response to direct customer requests. As a result of continuing to improve and develop an innovative and broad product line, Intel held an 85% share of the $130 million 1991 market.

INTEL SERVING TWO PRIMARY APPLICATIONS

Today, Intel flash is serving two major application segments: **updatable code storage** and **solid-state mass storage** (Figure 2). Code and data storage comprise the updatable non-volatile memory applications that require high performance, high density and easy update capability. These applications are infrequently updated when compared to solid-state mass storage applications. In this case, erase/write performance is not as critical as integration and performance requirements. This application segment is served effectively with full chip-erase or boot block products.

The second major application segment is **solid-state mass storage** that requires very high density, automated programming and high performance erase/write capability at a very low cost per bit. Erasing and writing portions of the code or data is much more frequent in solid-state mass storage than in updatable firmware applications. In April of 1992, Intel introduced the 8-Mbit FlashFile component whose architecture is optimized for data file storage. Its symmetrically blocked architecture and automated write/erase features gave programming flexibility for a high-performance, solid-state memory system. The compactness of an 8-Mbit device in a TSOP package allows for high-density flash arrays to be included on a system motherboard as well

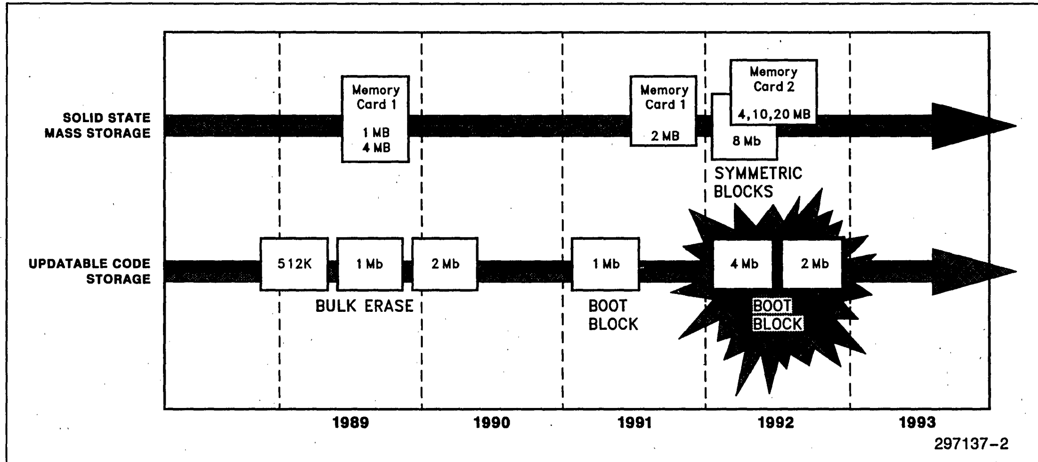


Figure 2

as in memory cards. Intel also offers a second generation of flash memory cards designed to serve the portable mass storage market that are based on the new 8-Mbit FlashFile component. Memory cards add the feature of removability and system upgradability in an industry-standard PCMCIA/ExCA™ format. While flash cards and FlashFile components serve solid-state mass storage applications directly, both can be used in updatable code storage applications as well.

By developing products to fit both of these application segments, Intel strives to serve the needs of a broader base of customer applications with the best nonvolatile memory solutions.

BOOT BLOCK UPDATE

The 28F001BX 1-Mbit Boot Block flash component introduced in June 1991 featured a sectored architecture that has been widely accepted in embedded code storage applications, particularly PC BIOS and cellular communications. Over 20 PC manufacturers use this device in their products, including Compaq, Dell and Zenith Data Systems. The Boot Block architecture gave the manufacturer added flexibility and the ability to differentiate. End users also benefit from the ability to upgrade BIOS software quickly and securely. It is possible for the manufacturer to upload BIOS updates to an electronic bulletin board where the end-user gets the upgrade for the price of the call. The blocked architecture allows the OEM customer to store critical system code securely in the boot block of the device that can minimally bring up the system and download to other locations of the device to initialize the system. The hardware boot locking feature guarantees that even if the power is disrupted during a BIOS update, the system will be able to recover immediately.

The success of the 1-Mbit Boot Block device has resulted in over 150 designs with annual shipments to exceed 1.5 million units worldwide in 1992.

EXTENDING THE BOOT BLOCK PRODUCT LINE: 2- AND 4-MEGABIT FLASH DEVICES

Once the Boot Block architecture became established in the marketplace, customers quickly began to ask for more features and enhancements: speed, density, low power, surface-mount options and an industry-standard upgrade path. These requests were based on the need to expand features in portable computing and communication products.

The 2-Mbit 28F200BX and 4-Mbit 28F400BX are Intel's newest additions to the flash Boot Block product line. These products offer 60 ns performance, two surface-mount packages, and a Boot Block architecture

similar to the 1-Mbit Boot Block device: one lockable Boot Block, two parameter blocks, and the balance of the device is divided into main blocks. The Boot Block securely stores the basic boot code required to initialize the system that can be protected by the hardware-locking feature, ensuring basic system operating code protection. The two parameter blocks can be used for a variety of purposes: manufacturing product code, setup parameters and storing frequently updated data such as system diagnostics. The 28F200BX contains one 16 Kbyte Boot Block, two 8 Kbyte parameter blocks, one 96 Kbyte and one 128 Kbyte main block (Figure 3a). The 28F400BX contains all of the blocks of the 28F200BX plus two additional 128 Kbyte main blocks (Figure 3b). Top and bottom Boot Block versions are available for both densities. Both devices are in the x16/x8 user-selectable organization of the industry-standard, ROM-compatible pin-out in 44-lead PSOP surface mount package. This pinout and package allow an easy upgrade from 2-Mbit to 4-Mbit since only one address is added at the top of the device package. Forty-lead TSOP x8-only versions are also available for both of these devices, providing high density in the smallest form factor. A 56-lead TSOP x16/x8 version will be available for applications requiring x16 organization in a TSOP package.

APPLICATION REQUIREMENTS—PCs

Notebook computer manufacturers today are competing to produce the lightest weight, slimmest and longest battery life products possible. Minimizing board size via reduced chip count is the first improvement the new flash devices provide. BIOS in portable systems has grown beyond the 1-Mbit density to accommodate more sophisticated power management and additional system features, such as PCMCIA card slots. For example, the i386™ SL architecture allows the expansion of BIOS beyond 128 Kbytes (1-Mbit) with internal registers for hardware paging. A new generation of BIOS support for low-power portable computers has been developed by SystemSoft and other vendors which supports the 28F200BX and the 28F400BX. In PC BIOS applications, the main blocks of the Boot Block devices are used for power management code, video drivers, and ROM-executable code, such as MS-DOS*. The arrangement of the blocks for PC BIOS applications based on Intel i386 and i486™ microprocessors is with the Boot Block on top. Other microprocessors and microcontrollers, such as the Intel 80960KX/SX and the Motorola 68000 series, use the bottom Boot Block version of the memory map. Many PC manufacturers have also moved to a x16 organization of their BIOS for

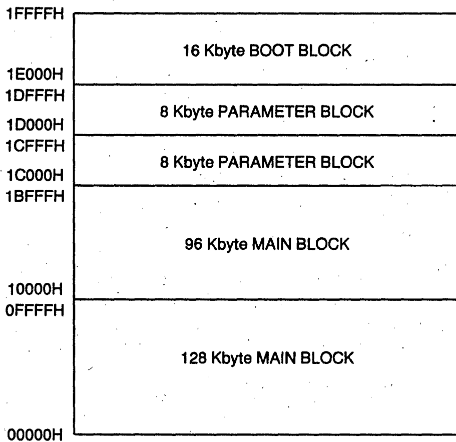


Figure 3a. 28F200BX-Top Boot

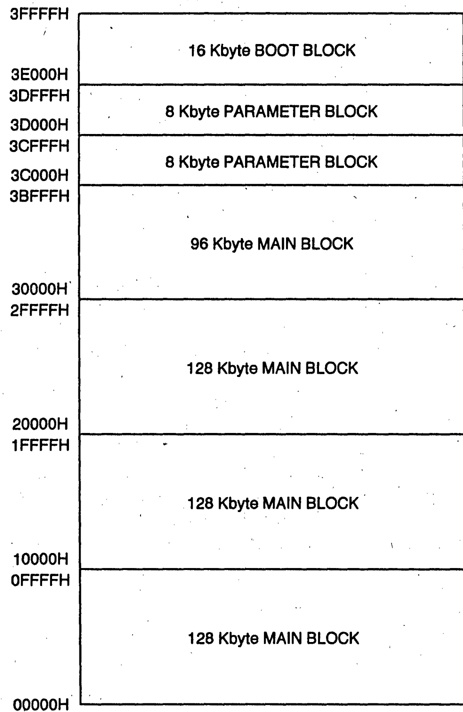


Figure 3b. 28F400BX-Top Boot

higher performance and are using 2-Mbit flash components in parallel to accomplish x16 performance. Other manufacturers have decided to forego x16 and use the 1-Mbit Boot Block and a 1-Mbit bulk-erase 28F010 flash memory to contain their power management code. The 28F200BX solves both of these problems in a compact, single-chip solution. The 28F400BX provides a higher density solution for high performance systems today, as well as a future upgrade for systems currently requiring only 2 megabits.

APPLICATION REQUIREMENTS—TELECOMMUNICATIONS/EMBEDDED

Similar to personal computers, cellular telephone applications use the hardware-lockable block to store basic initialization code to wake up the system and establish basic communication with the host base station. The small parameter blocks are ideal for frequently updated code such as the user's phone directory, re-dial and the activation code necessary to enable user-desired features. The main block contains the code for dynamically managing the cellular communications and executing the voice algorithms. The density requirement for the main code storage has increased significantly in recent years with the conversion to digital cellular transmission, particularly in Europe and Japan. Most cellular designs execute code directly from ROM rather than downloading to RAM. Typical microcontrollers used are: Intel 80C186, Zilog Z8080 and Siemens 80166, all operating in the 8 MHz–13 MHz range. The Motorola 68000 series 16-bit microcontrollers used in some digital cellular designs utilize the x16 and high speed capabilities of these new Boot Block devices. Flash access times ranging from 75 ns to 120 ns produce 0 to 1 wait-state performance. The high-performance 60 ns access time of the 2-Mbit and 4-Mbit Boot Block flash allows zero wait state performance in cellular telephones and other embedded control applications.

Low power has always been a strength of flash due to its inherent nonvolatility, a characteristic that eliminates battery drain required to maintain information stored in volatile RAM when the power is off. When flash is in use, its active power requirement is very low: typical I_{CC} = 35 mA, and standby current is 0.1 mA. The hallmark of this newest generation of flash is low power and high performance. Formerly these features were mutually exclusive, but through advanced circuit development, an automated power saving feature in the device reduces I_{CC} to a low DC level within one access time; 5 mA typical. This significant reduction of the active current time relates directly to the extension of battery life in systems which continuously execute code directly from flash, such as cellular telephones and portable instrumentation.

For optimum system power conservation, future systems are being designed today to operate at 3.3V rather than 5V. Intel will also offer 3.3V versions of the 2-Mbit and 4-Mbit Boot Block devices with I_{CC} active = 10 mA, and t_{ACC} performance = 150 ns.

An alternative design approach of a "5V-only" (programming voltage = operating voltage = 5V) flash memory has been proposed to eliminate the need for a 12V programming supply. While this proposal would eliminate the need for a 12V supply component in a 5V-only environment, it would not provide a low power solution as every flash component would carry the die size cost and power overhead of on-chip voltage pumping. Intel's approach is to first bring the operating voltage to 3.3V, in line with what system designers requested as their first priority for the next generation of portable computers and cellular phones. The focus of lowering the read voltage is also congruent with the usage model for embedded applications of "read often, write few" operation. The 12V programming voltage requirement is more efficiently met with a voltage pump from a vendor such as Maxim. (Inexpensive voltage pumps for 3.3V to 12V are currently available.) The use of a separate voltage pump is significantly more cost-efficient in systems where more than one flash device is used, as cost and space is amortized over multiple devices.

CONCLUSION

Intel continues to serve both updatable non-volatile memory applications as well as the rapidly emerging solid-state mass storage market with solutions tailored to meet their particular needs. The new 2-Mbit and 4-Mbit Boot Block devices offer high performance and low power options for updatable code applications. The Boot Block architecture is compatible with all major microprocessors and microcontrollers. As with all Intel flash products, their low power consumption and small surface mount packaging make the 2-Mbit and 4-Mbit Boot Block flash memories ideal for a wide variety of handheld portable applications.

NOTE:
FlashFile, ExCA, i386 and i486 SL are trademarks of Intel Corporation.

*MS-DOS is a registered trademark of Microsoft Corporation.

November 1992

Intel FlashFile™ Memory The Key to Diskless Mobile PCs

JANET WOODWORTH
MEMORY COMPONENTS DIVISION

Order Number: 297115-001

INTRODUCTION

As the PC evolves into what is truly a "personal" computer—one that can be held in your hand—a completely different system memory architecture will emerge. Step aside ROM, DRAM, floppy disk, and hard disk; Intel's FlashFile™ memory is here. FlashFile memory will finally make it possible to build a thin, 2-pound notebook computer that runs for many hours on a few AA batteries. But before these mobile PCs are built, designers must learn some new ways to configure system memory.

In April 1992, Intel introduced a new flash memory architecture with a combination of functionality and price that redefines mobile computing. This new architecture, when implemented in new system memory configurations, enables nonvolatile executable system memory and removable file and program storage. Intel's new flash architecture lets designers create a portfolio of products that will clearly differentiate them from their competition.

WHY A NEW MEMORY ARCHITECTURE?

The ideal memory system is:

- Dense (stores lots of code and data in a small amount of space and weighs very little)
- Fast (lets you read and write data quickly)
- Inexpensive (low cost per megabyte)
- Nonvolatile (data remains when power is removed)
- Power Conscious (prolongs battery life and reduces heat)
- Reliable (retains data when exposed to extreme temperature and mechanical shock)

Since PCs were introduced over 10 years ago, designers have grappled with how to construct memory systems that offer all these attributes. They have wisely elected to use to optimum combination of solid-state memory and magnetic storage, i.e., DRAMs plus magnetic hard disks. DRAMs are dense and inexpensive, yet slower than the processors they serve, and they are volatile. SRAMs are used in caching schemes to compensate for DRAM's slowness. While SRAMs keep pace with today's high-performance microprocessors, they are not as dense as DRAM, are inherently more expensive, and volatile. Magnetic hard disks are very dense, inexpensive on a cost-per-megabyte basis, and nonvolatile, but they are painfully slow, power hungry and subject to damage from physical shock.

ENTER FLASH MEMORY

Because Intel's ETOX™ III flash memory cell is 30-percent smaller than equivalent DRAM cells, the company expects it to track DRAM density closely. Intel's new 28F008SA FlashFile Memory can store 8 megabits, or one megabyte, of data today. Flash memory is more scalable than DRAM due to its simple cell structure, so as DRAM technology shrinks towards 0.25 microns and 64 megabits, flash will pace and ultimately overtake DRAM's technology treadmill. In fact, expect to see 256-Mbit flash memory by the end of the '90s.

FLASH MEMORY IS FAST

Don't be misled by technology-to-technology speed comparisons. Designing your system memory around flash will break the code/data bottleneck created by connecting a mechanical memory such as disks to a high-performance electronic system. For instance, data seek time for a 1.8" magnetic hard disk is 20 ms, plus an 8 ms average rotational delay, while flash is less than 0.1 ms. At the chip level, current read speeds for flash are about 90 ns. Thus, downloading from flash to system RAM or directly executing from flash will dramatically enhance system speed.

FLASH MEMORY IS INEXPENSIVE

At the 8-Mbit density, Intel flash pricing matches DRAM and Intel expects to continue decreasing price as both densities and volumes increase.

FLASH IS NONVOLATILE

Unlike SRAM or pseudo-SRAM (SRAM with built-in battery), flash needs no battery backup. Further, Intel's flash devices retain data typically for over 100 years, well beyond the useful lifetime of even the most advanced computer.

FLASH IS POWER CONSCIOUS

FlashFile Memory in a hard-disk drive configuration consumes less than one two-hundredth the average power of a comparable magnetic disk drive based on the typical user model. At the chip level, the 28F008SA has a DEEP POWERDOWN mode that reduces power consumption to less than 0.2 μA.

FLASH IS RUGGED AND RELIABLE

On average, today's hard-disk drives can withstand up to 10 Gs of operating shock; Intel FlashFile memory can withstand as much as 1000 Gs. FlashFile compo-

nents can operate at up to 70°C while magnetic drives are limited to 55°C. Intel FlashFile memory can be cycled 100,000 times per block or segment. By employing wear-leveling techniques, a 20-Mbyte flash array can provide over 30 million hours before failure.

WHY NOW?

Flash memory is not a new technology. Intel has been the flash technology and market leader since 1988. Then why hasn't flash taken the mobile PC market by storm yet? Why now?

One reason that 1992 is the pivotal year for flash-based systems is the sharply increased demand for highly mobile computers. The other reason is that a number of key capabilities, in development for some time, reached maturity together.

1. Intel Introduces FlashFile™ Memory

MS-DOS*, the ubiquitous operating system for PCs, was developed specifically to optimize the sectoring scheme inherent to disk technology. Intel's first generation "bulk-erase" flash required that all of the chip be erased before data could be re-written: a natural fit for updatable firmware and data acquisition, but not for data file storage or disk emulation. Intel FlashFile memory, based on a block-erase architecture, divides the flash memory space into segments that are somewhat analogous to the zones recognized by MS-DOS. For instance, the Intel 28F008SA contains sixteen identical, individually-erasable, 64-Kbyte blocks. This organization has been carefully optimized to maximize cycling capability while preserving the smallest granularity possible. The ability to segment block memory into individual segments allows disk-like data-file storage.

2. Standardization of Delivery System and Interface

Thanks to work by the Personal Computer Memory Card International Association (PCMCIA), and the Japanese Electronics Industry Development Association (JEIDA), there is now an internationally recognized standard for memory cards. PCMCIA cards are the size of a business card but about four times as thick. Intel is widely promulgating its Exchangeable Card Architecture (ExCA™), a hardware and software implementation of the PCMCIA system interface. When used with the proper BIOS, ExCA/PCMCIA-compatible cards will be completely interchangeable between systems and vendors, and can be equated to solid-state floppy disks, albeit with many advantages.

Flash-based solid-state disks, intended to replace magnetic hard disks in certain applications, with IDE interfaces will be "plug compatible" with existing systems that are already designed with IDE magnetic drives in mind.

3. Flash File System

Intel has worked very closely with Microsoft* to implement a DOS flash memory extension called Flash File System (FFS) that transparently handles swapping of data between flash blocks, much as DOS now handles swaps between disk sectors. With Flash File System, the user inputs a DOS comand and doesn't need to think about whether a magnetic disk or a flash memory is being used. Flash File System employs wear leveling algorithms that prevent any block from being cycled excessively, thus ensuring millions of hours of use across multiple chips.

4. Off-the-Shelf Hardware Interface

The introduction of the Intel 82365SL™ PC Card Interface Controller provides a ready-made interface between the PC's ISA bus and up to two PCMCIA sockets. It is a key component for memory and I/O card implementations since the designer is relieved from building the interface from scratch.

5. Cost Reductions

Magnetic drives do not scale well; that is, it becomes increasingly difficult to improve or even retain density as platter size shrinks. Thus, every reduction in drive size requires complete retooling and costly learning. Also, the complex controller circuitry provides a price floor under which magnetic drives cannot drop. Since flash is scalable, at some point in the near future, small magnetic drives are likely to become more expensive per megabyte than flash cards and are certain to have less capacity. But even today, the value of a particular memory technology is a result of more than just dollars per megabyte.

Notes market analysis expert Dataquest:

> "The question is, "Can you put a floppy disk drive in a palmtop PC to take advantage of the cost disparity (between disk and flash)?" The answer is, 'No.' There is not enough power (or space). The issue then, is not cost. Here, the removable storage medium dictates the product's capabilities and its success or failure in the marketplace. Without a memory card, a palmtop is nothing more than an electronic organizer. It is the memory card that transforms a palmtop into a full-fledged personal computer." . . . Nick Samaras, SAMS Newsletter.

All of the aforementioned features, Intel FlashFile memory's block-erase architecture, PCMCIA standards, ExCA, Flash File System, 82365SL ISA-PCMCIA interface controller and reduced costs, are deliverables ... now. And not a moment too soon based on the tremendous market opportunity created by the increasing demand for truly mobile computers. Dataquest predicts that the worldwide sale of portable PCs will increase from six million units in 1992 to nearly 30 million units in 1995. While laptop PCs are only expected to increase by about two million units, notebooks, pen-based, and handheld PCs will increase from three million units in 1992 to nearly 25 million in 1995, an eight-fold increase. This extraordinary growth will be greatly assisted by FlashFile memory.

ENABLING THE TRULY MOBILE COMPUTER

In the world of the desktop PC, DRAM is used for executable code storage and data manipulation. Since DRAM is volatile, if power is lost, both programs and data are lost, hence the need for a nonvolatile magnetic hard disk. With the addition of the hard disk, programs and data are stored on the hard disk and swapped in and out of DRAM as needed. Some part of the DRAM is reserved for use as a register to store temporary results during compute-intensive operations. Today's PCs are typically configured with 4 megabytes of DRAM and at least a 40-Mbyte disk.

FlashFile memory fully supports this system configuration when used simply as a magnetic drive replacement. Instructions and data are still swapped to DRAM but at a much faster rate. Plus, execution speed can be enhanced if the DRAM is replaced with SRAM.

In the solid-state computer, the "DRAM + magnetic hard drive" are replaced by a "flash memory + SRAM". The key to this architecture is the ability to eXecute-InPlace (XIP). Program instructions stored in the flash memory are read directly by the processor. Results are written directly to the flash memory. Compute-intensive operations that require the fastest memory and byte-alterability use high-speed SRAM or pseudo SRAM. Most of what we now think of as the "DRAM" is replaced by low-cost flash and only a relatively small part of the DRAM is replaced by SRAM.

The flash memory space is made even more storage-efficient through the use of compression techniques which offer at least 2:1 compression. For example, one 20-Mbyte flash card that uses 2:1 compression offers the same storage as a 40-Mbyte hard disk!

The advantages of a flash-based computer include:

- Blazingly fast speed
- Instant-on and instant-resume
- Ultra-light PC (2–4 lbs.)
- Very secure data retention
- Flexible firmware

As you can see, by changing the system memory architecture to a flash-based one, designers will be able to build a new generation of PCs that meets the needs of the computer user of both today and tomorrow.

Progress has been made toward implementing this approach with the introduction of Hewlett-Packard's successful HP95LX DOS-compatible palmtop. MS-DOS and Lotus 1-2-3* are stored in ROM. Internally, pseudo-static RAM is used, and a PCMCIA memory socket is provided. Lotus 1-2-3 was re-written to allow ROM-based storage so it could execute in place. Other ROM-executable versions of popular operating systems are expected to be available shortly.

DESIGNING YOUR SYSTEM WITH FLASHFILE MEMORY

Details of the three Intel flash applications and implementations—flash cards, silicon disks, and Resident Flash Array (RFA)—are presented below.

APPLICATION NUMBER 1: MEMORY CARDS

Memory cards are the most rugged and reliable of the removable memory media. A card can be slipped into a shirt pocket and moved from location to location. With high-density flash cards, you can download files from the desktop and use the card in your notebook or palmtop.

Memory cards have been around for some time. The first cards to be introduced were ROM-only cards used in video games and pocket organizers. These were produced in various formats prior to the formation of PCMCIA and JEIDA. Later cards included battery-backed SRAM and EEPROM. Neither became very popular due to their high cost of $500–$600 per megabyte and limited capacity. Flash cards overcome the cost barrier and they are certain to be multiply sourced, assuring availability and competitive pricing. A 20-Mbyte flash card has three times the real storage density of a 20-Mbyte 1.8" magnetic drive (0.95 Mbyte/cm^3 vs 0.34 Mbyte/cm^3) and it has 10 times the weight density (2 Mbyte/gm vs 0.21 Mbyte/gm). The PCMCIA has complete industry support, and enhanced versions, such as PCMCIA Version 2.0, are designed to be backward-compatible with earlier versions.

As part of its flash product family, Intel's new Series 2 memory cards are the first to utilize chips processed on its 0.8-micron flash technology. Storing up to 20 megabytes, these cards are designated Series 2 to differentiate them from the earlier bulk-erase flash cards. The cards consist of 4 to 20, 28F008SA TSOP FlashFile memory devices. Each 28F008SA contains 16 distinct, individually-erasable, 64-Kbyte blocks. Therefore, each card contains from 64 to 320 blocks.

With the release of PCMCIA Version 2.0 in September of 1991, the PCMCIA-compatible field grew somewhat larger. The PCMCIA interface grew from memory-only to one that supports many types of I/O devices. Intel's system-level implementation of PCMCIA 2.0, called ExCA, ensures that if there are two ExCA sockets available, one can be used for a flash memory card and one for a modem; and the cards may be interchanged.

How difficult is it to design-in an ExCA socket? Not very. Intel's open ExCA specification details the system implementation. Other than the physical incorporation of the socket and card, the only required hardware is an ISA-to-PCMCIA interface such as Intel's 82365SL chip, and an ExCA compliant BIOS from vendors like SystemSoft, Award and Phoenix. You'll also need a flash file management system like Microsoft's Flash File System. Intel's block-erase architecture, along with the DOS Filing System and ExCA BIOS, makes it easy to incorporate ExCA features. In addition, ExCA-compliant systems will allow system-to-system interoperability much like floppy disks.

APPLICATION NUMBER 2: FLASH-BASED SOLID STATE DISK

The implementation of block-erase flash as a "solid-state disk" (SSD) is something of a misnomer. It is not a disk at all, rather a flash module that has the same form, fit and function as a 2.5" or smaller magnetic drive.

A flash-based SSD implementation is the most direct route to adapting flash to an existing design. A built-in IDE interface would make it plug-compatible. But what a difference a silicon disk will make! A 1.8" drive typically uses one watt-hour/hour while a silicon disk uses as little as 0.035 watt-hour/hour. This kind of power savings makes it possible to reduce battery size and weight considerably. Or, consider reliability. We've already discussed differences in susceptibility to shock and temperature extremes. In addition, an SSD theoretically has a mean-time-between-failure (MBTF) of 250,000 hours, compared to 100,000 hours for the magnetic drive.

With all these advantages, when should you use memory cards and when is use of a flash-based SSD preferable?

First and foremost, the SSD is considered to be installable while memory cards are removable and transportable. In other words, the SSD is meant to be installed and then left alone, while memory cards are designed for constant removal and reinsertion. In operation, the only change a user would notice in a notebook computer equipped with a flash-based SSD is that access speed is unprecedented.

The flash-based solid-state drive is one very good way to get to market early with flash technology. In February 1992, Conner Peripherals, Inc., and Intel announced the signing of a joint product and technology development contract focused on designing and bringing to market proprietary FlashFile memory-based SSD storage products.

Incidentally, manufacturers of magnetic drives are starting to take notice. In a manner much like the tail wagging the dog, 1.8″ magnetic hard disks with PCMCIA interfaces are currently being developed.

APPLICATION NUMBER 3: RESIDENT FLASH ARRAY

The one approach that offers totally new capabilities is the Resident Flash Array (RFA). This is an arrangement of from 8 to 20, 8-Mbit FlashFile memories. In the long term, it replaces some of the motherboard's DRAM. This is the approach that is applicable to all levels of PC, from desktop to palmtop. For near-term applications, however, RFA is an ideal way of making code or ROM-executable operating systems such as DOS or Windows* updatable to protect the end-user's software investment. Also, when used as a resident application program and data-file storage medium on the local memory bus, RFA provides a high-performance, low-power solution.

The Resident Flash Array provides the highest possible performance of any option, especially since the processor can be closely coupled to it; and hence, would not be encumbered by IDE or PCMCIA interfaces, or even the ISA bus itself. The flash memory and the processor will sit side-by-side.

The proliferation of flash memory card-based systems will accelerate the process of converting disk-oriented applications to a flash-executable orientation. Those manufacturers who elect to be early adopters of Intel FlashFile memory will be able to develop a new generation of PC—the truly "personal" computer you can hold in your hand.

NOTE:
ETOX, ExCA and FlashFile are trademarks of Intel Corporation.
*Microsoft and MS-DOS are registered trademarks; Windows is a trademark of Microsoft Corporation.
*Lotus and 1-2-3 are registered trademarks of Lotus Development Corporation.

SOP PHYSICAL DIMENSIONS

1.0 SOP Case Outlines for Intel's 32-, 40-, 56-Lead TSOP and 44-Lead PSOP Packages

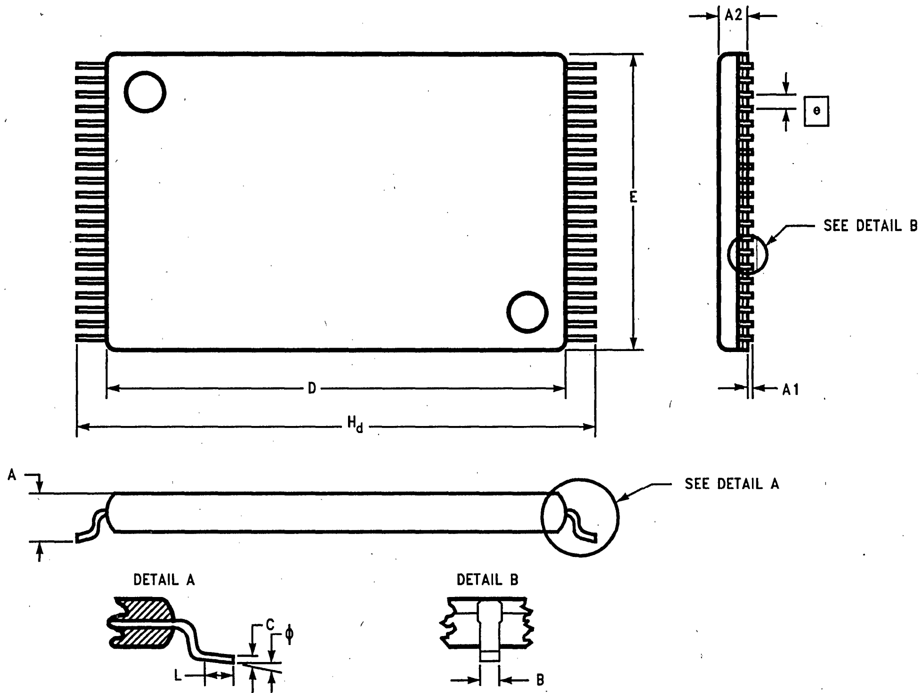
32-Lead Thin Small Outline Package (TSOP)

297186-1

Symbol		Millimeters			Inches		
		Min	Typ	Max	Min	Typ	Max
Package Height	A			1.20			0.0472
Standoff	A1	0.050			0.0019		
Package Body Thickness	A2	0.96	1.01	1.06	0.0378	0.0398	0.0417
Lead Width	B	0.150	0.200	0.250	0.0059	0.0079	0.0098
Lead Thickness	C	0.10	0.15	0.20	0.0039	0.0059	0.0079
Package Body Length	D	18.35	18.40	18.45	0.7224	0.7244	0.7264
Package Body Width	E	7.95	8.00	8.05	0.3130	0.3150	0.3169
Pitch	e		0.50			0.0196	
Terminal Dimension	H_d	19.90	20.00	20.10	0.7835	0.7874	0.7913
Lead Tip Angle	ϕ	1°	3°	5°	1°	3°	5°
Lead Tip Length	L	0.300	0.330	0.350	0.0118	0.0130	0.0138
Lead Count	N		32			32	
NOTES:							
Gull Wing Length		0.600	0.800	0.950	0.0236	0.0314	0.0374
Seating Plane Coplanarity				0.100			0.0039
Lead to Package Offset		0.20	0.25	0.30	0.0079	0.0098	0.0118

Figure 1-1. 32-Lead TSOP Package Drawing and Specifications

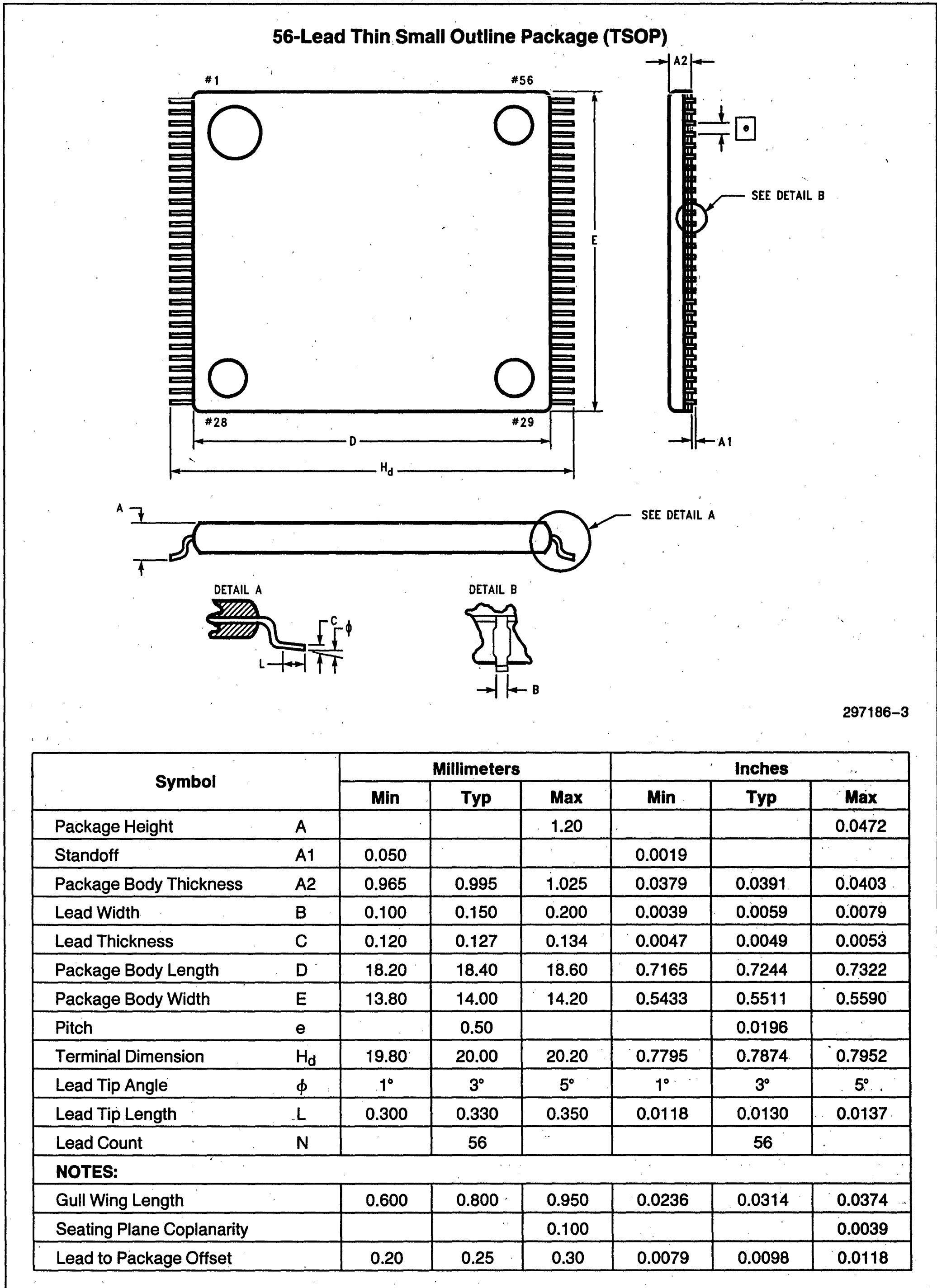
40-Lead Thin Small Outline Package (TSOP)



Symbol		Millimeters			Inches		
		Min	Typ	Max	Min	Typ	Max
Package Height	A			1.20			0.0472
Standoff	A1	0.050			0.019		
Package Body Thickness	A2	0.965	0.995	1.025	0.0379	0.0391	0.0403
Lead Width	B	0.150	0.20	0.250	0.0059	0.0078	0.0096
Lead Thickness	C	0.120	0.127	0.134	0.0046	0.0049	0.0052
Package Body Length	D	18.20	18.40	18.60	0.7165	0.7244	0.7322
Package Body Width	E	9.80	10.00	10.20	0.3858	0.3937	0.4015
Pitch	e		0.50			0.0196	
Terminal Dimension	H_d	19.80	20.00	20.20	0.7795	0.7874	0.7952
Lead Tip Angle	ϕ	1°	3°	5°	1°	3°	5°
Lead Tip Length	L	0.300	0.330	0.350	0.0118	0.0130	0.0137
Lead Count	N		40			40	
NOTES:							
Gull Wing Length		0.600	0.800	0.950	0.0236	0.0314	0.0374
Seating Plane Coplanarity				0.100			0.0039
Lead to Package Offset		0.20	0.25	0.30	0.0079	0.0098	0.0118

Figure 1-2. 40-Lead TSOP Physical Dimensions and Specifications

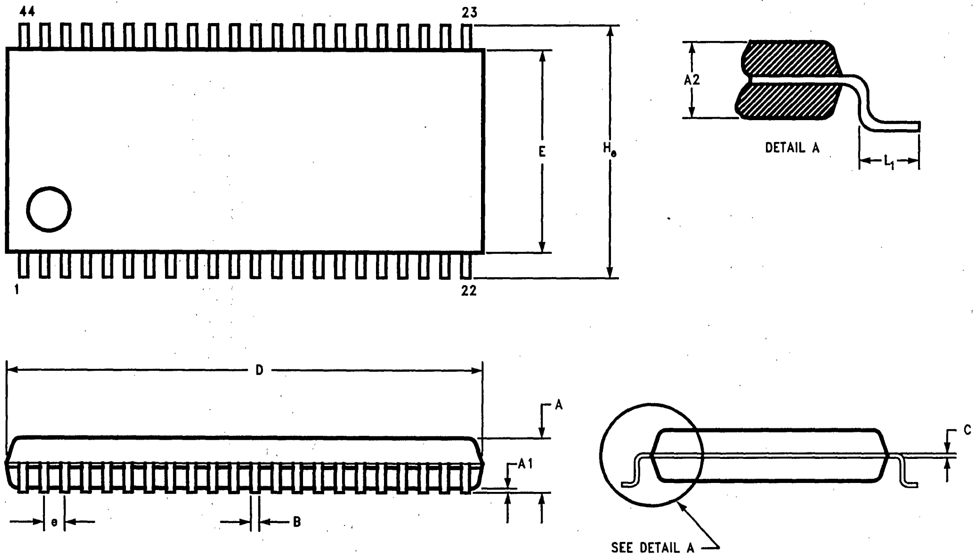
56-Lead Thin Small Outline Package (TSOP)



Symbol		Millimeters			Inches		
		Min	Typ	Max	Min	Typ	Max
Package Height	A			1.20			0.0472
Standoff	A1	0.050			0.0019		
Package Body Thickness	A2	0.965	0.995	1.025	0.0379	0.0391	0.0403
Lead Width	B	0.100	0.150	0.200	0.0039	0.0059	0.0079
Lead Thickness	C	0.120	0.127	0.134	0.0047	0.0049	0.0053
Package Body Length	D	18.20	18.40	18.60	0.7165	0.7244	0.7322
Package Body Width	E	13.80	14.00	14.20	0.5433	0.5511	0.5590
Pitch	e		0.50			0.0196	
Terminal Dimension	H_d	19.80	20.00	20.20	0.7795	0.7874	0.7952
Lead Tip Angle	ϕ	1°	3°	5°	1°	3°	5°
Lead Tip Length	L	0.300	0.330	0.350	0.0118	0.0130	0.0137
Lead Count	N		56			56	
NOTES:							
Gull Wing Length		0.600	0.800	0.950	0.0236	0.0314	0.0374
Seating Plane Coplanarity				0.100			0.0039
Lead to Package Offset		0.20	0.25	0.30	0.0079	0.0098	0.0118

Figure 1-3. 56-Lead TSOP Package Drawings and Specifications

44-Lead Plastic Small Outline Package (PSOP)



Symbol		Millimeters			Inches		
		Min	Typ	Max	Min	Typ	Max
Package Height	A			2.62			0.103
Standoff	A1	0.13	0.225	0.35	0.005	0.009	0.013
Package Body Thickness	A2	2.17	2.30	2.45	0.085	0.091	0.097
Lead Width	B	0.35	0.40	0.50	0.014	0.016	0.020
Lead Thickness	C	0.13	0.150	0.20	0.005	0.006	0.008
Package Body Length	D		28.20	28.70		1.100	1.130
Package Body Width	E	13.10	13.30	13.50	0.516	0.524	0.531
Pitch	e		1.27			0.050	
Terminal Dimension	H_e	15.70	16.00	16.30	0.618	0.630	0.642
Lead Tip Angle	ϕ			8°			8°
Lead Tip Length	L1	0.75	0.80	0.85	0.029	0.032	0.033
Lead Count	N		44			44	
NOTES:							
Gull Wing Length		1.30	1.35	1.40	0.051	0.053	0.055
Seating Plane Coplanarity				0.10			0.004

Figure 1-4. 44-Lead PSOP Physical Dimensions and Specifications

3

2.0 SOP Board Footprints

A typical land pad diagram for the 32-Lead TSOP package is shown in Figure 2-1.

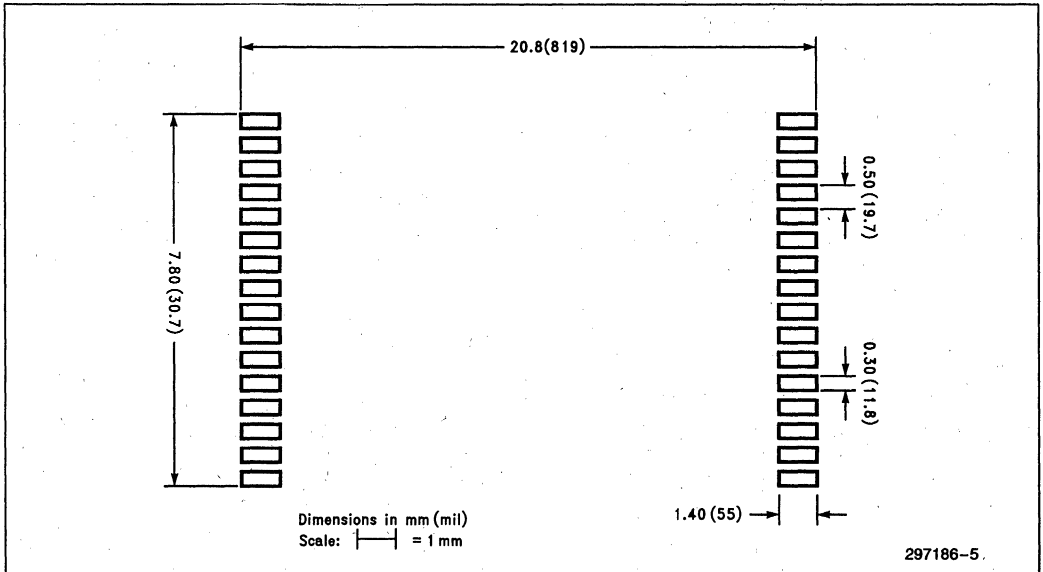


Figure 2-1. Typical TSOP Land Pad Diagram

Similar land pad diagrams can be constructed for the 40-lead and 56-lead TSOP packages. The 40-lead land pad diagram can be constructed by adding four leads to both sides of the 32-lead land pad diagram, while maintaining the spacing between the lead footprints. The total footprint width becomes 9.80mm instead of 7.80mm used with the 32-lead package. With the 56-lead package, 12 leads are added to both sides of the 32-lead land pad diagram, while maintaining the spacing between the lead footprints. The total footprint width becomes 13.80mm instead of the 7.80mm used with the 32-lead package.

A typical land pad diagram for the 44-Lead PSOP is shown in Figure 2-2.

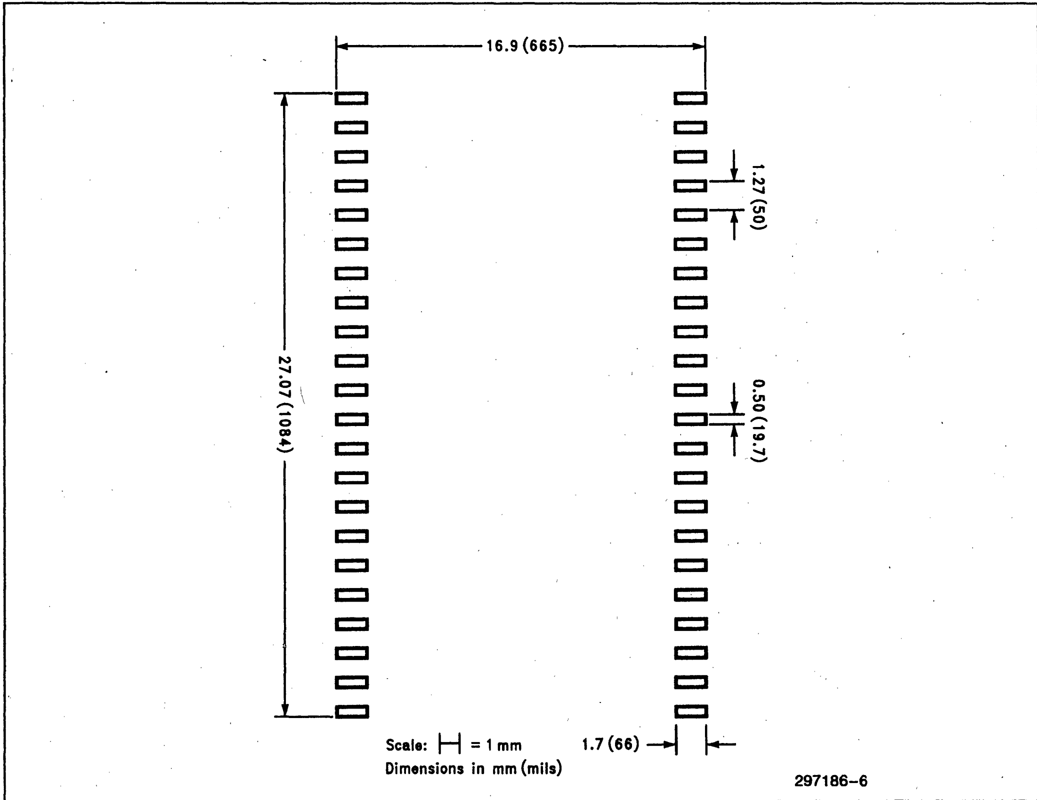


Figure 2-2. Typical PSOP Land Pad Diagram

3.0 SOP Component Volume and Weight

Table 3-1 shows the component volume and weight of the SOP package family.

Table 3-1. SOP Component Weight and Volume

Package	Max Height	Max Volume	Average Weight
32-Lead TSOP	1.20 mm	194.2 mm^3	0.37 gms
40-Lead TSOP	1.20 mm	247.2 mm^3	0.47 gms
56-Lead TSOP	1.20 mm	344.2 mm^3	0.65 gms
44-Lead TSOP	2.62 mm	1,225.7 mm^3	186 gms

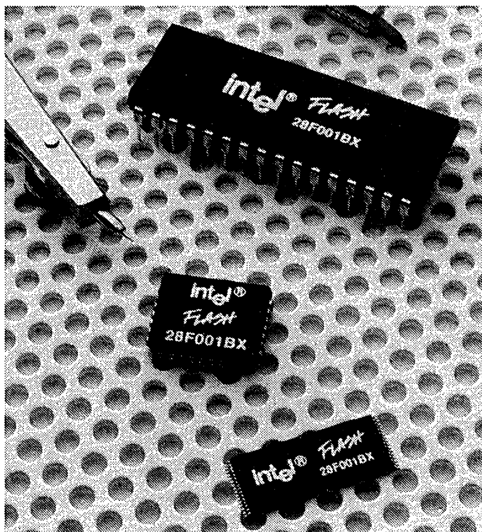
PRODUCT BRIEF

Intel Blocked Flash Memory 28F001BX-B/28F001BX-T

Intel's 28F001BX-B and 28F001BX-T combine the cost-effectiveness of Intel standard flash memory with features that simplify write and allow block erase. These devices aid the system designer by combining the functions of several components into one, making blocked flash memory an innovative alternative to EPROM and EEPROM or battery-backed static RAM. Many new and existing designs can take advantage of the 28F001BX's integration of blocked architecture, automated electrical reprogramming and standard processor interface.

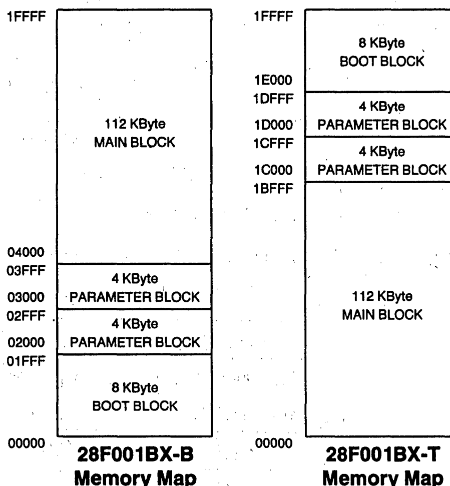
Product Highlights

- **High-Integration Blocked Architecture**
  - **One 8 Kb Boot Block w/Lock Out**
  - **Two 4 Kb Parameter Blocks**
  - **One 112 Kb Main Block**
  - **Min 10,000 Cycles/Block**
- **Simplified Program and Erase**
  - **Automated Algorithms Via On-Chip Write State Machine**
- **Deep-Powerdown Mode**
  - **0.05 μA I_{CC} Typical**
  - **0.8 μA I_{PP} Typical**
- **High-Performance Read**
  - **120 ns Maximum Access Time**
- **Hardware Data Protection Feature**
  - **Erase/Write Lockout During Power Transitions**
- **Advanced Packaging, JEDEC Pinouts**
  - **32-Pin PDIP**
  - **32-Lead PLCC, TSOP**
- **Extended Temperature Option**
  - **−40°C to +85°C**



296913-1

Intel 28F001BX Flash Memories



28F001BX-B Memory Map

28F001BX-T Memory Map

MCS®-186 is a registered trademark of Intel Corporation.
i386™, i486™ and i860™ are trademarks of Intel Corporation.

August 1992
Order Number: 296913-003

Product Description

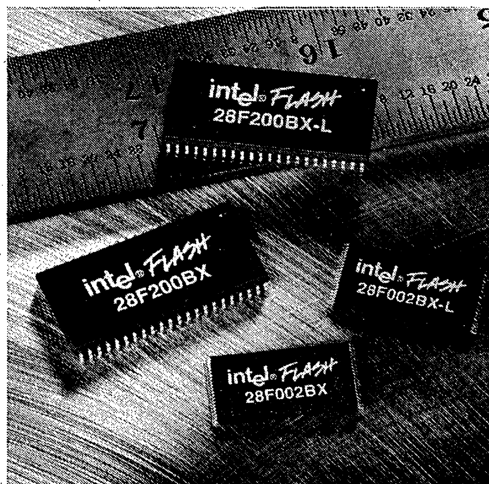
The 28F001BX-B and 28F001BX-T 1 Mbit (128K x 8) blocked flash memories augment the nonvolatility, in-system electrical erasure and reprogrammability of Intel's standard flash memory. They offer four separately erasable blocks and integrate a state machine to control erase and program functions. The tailored blocking architecture and automated programming of the 28F001BX family provide a full-function, nonvolatile flash memory ideal for a wide range of applications, including PC boot/BIOS memory, minimal-chip embedded program memory and parametric data storage.

The 28F001BX family combines the safety of a hardware-protected 8 Kbyte boot block with the flexibility of three separately reprogrammable blocks (two 4 Kbyte parameter blocks and one 112 Kbyte code block) into one versatile, cost-effective flash memory. Additionally, reprogramming one block does not affect code stored in another block, ensuring data integrity.

The 28F001BX-T's block locations provide compatibility with microprocessors and microcontrollers that boot from high memory, such as Intel's i386™, i486™, i860™, 80960CA and MCS®-186 families. The 28F001BX-B memory map is tailored for bottom-boot devices such as Intel's MCS-51, MCS-196, 80960KA/KB and 80960SA/SB microcontrollers.

Intel 2-Megabit Boot Block Flash Memory 28F200BX-T/B, 28F002BX-T/B

Intel's 28F200BX-T/B, 28F002BX-T/B product family is a new addition to the Boot Block Flash Memory product line. This product family is manufactured on Intel's third generation ETOX™ flash process. The boot block architecture in the 2 Megabit density is particularly well-suited for portable PC BIOS and updateable firmware applications. The 28F200BX/28F002BX family allows block selective erasure, automated write and erase operations and standard microprocessor interface. High-performance access times of 60ns and 80ns provide zero-wait-state operation with a wide range of microprocessors and microcontrollers.



297139–1

Product Highlights

- High-Integration Blocked Architecture
  - One 16 KB Boot Block w/Lock Out
  - Two 8 KB Parameter Blocks
  - One 96 KB Main Block
  - One 128 KB Main Block
- Hardware Data Protection Feature
- Top and Bottom Boot Versions Available
- High Performance 60ns and 80ns Read Access Times
- V_{PP} = 12V ±10% (Option)
- x8 or x16 User Configurable I/O Control
- I_{CC} Active = 20 mA (Typical)
- Deep-Powerdown Mode (0.2 µA Typical)
- Automatic Power Savings Feature
- 3.3V versions
- 100,000 Write-Erase Cycles/Block
- Write and Erase Automation
- Industry Standard Surface Mount Packages
  - 40-Ld TSOP (x8 only); — 44-Ld PSOP;
  - 56-Ld TSOP
- Easy upgrade path to 4 Megabits

Product Description

The 28F200BX-T/B and 28F002BX-T/B Flash Memories are components of a 2 Mbit (2,097,152 bit) family of flash memory devices, each organized as either 262,144 bytes or 131,072 words. These flash memory devices combine the safety of a hardware-lockable 16 Kbyte boot block with the flexibility of separately reprogrammable blocks (two 8 Kbyte parameter blocks, one 96 Kbyte main block and one 128 Kbyte main block). The specialized blocking architecture, high-speed performance, x16 organization, and small surface mount packaging provide a full-function, nonvolatile flash memory ideal for a wide range of applications, including PC BIOS, Telecommunications, Portable Instrumentation, Office/Factory Automation Equipment.

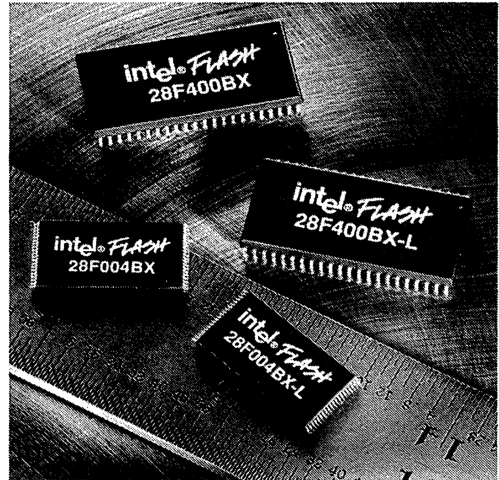
The 28F200BX-T and the 28F002BX-T (top boot) devices offer block locations that provide compatibility with microprocessors and microcontrollers that boot from high memory, such as Intel's i386™, i486™, i860™, 80960CA and MCS®-186 families. The 28F200BX-B and the 28F002BX-B (bottom boot) devices are tailored for microprocessors and microcontrollers that boot from low memory, such as Intel's MCS-51, MCS-196, 80960KX and 80960SX microcontrollers.

Order Number: 297139-001

FEATURES	BENEFITS
— High-Integration Blocked Architecture	— Modular code development/updates
— Top Boot and Bottom Boot Architecture	— Easy interface with wide range of processors
— x8 or x16 User Configurable Bus Operation	— Interfaces with high-end processors
— 60 ns Read Access Time	— DRAM-like access times
— Automated Byte-Write/Block Erase	— Simplifies design, frees processor for higher priority tasks
— Erase Suspend Capability	— Enables data/code access during erase
— Automatic Power Savings Feature	— Saves active power without reducing performance
— 0.2 μA Low Power Sleep Mode	— Extends battery life

Intel 4-Megabit Boot Block Flash Memory 28F400BX-T/B, 28F004BX-T/B

Intel's 28F400BX-T/B, 28F004BX-T/B product family is a new addition to the Boot Block Flash Memory product line. Manufactured on Intel's third generation ETOX™ flash process, the 28F400BX/28F004BX offers the most highly integrated and highest performance solution for updatable firmware. The 28F400BX/28F004BX family allows block selective erasure, automated write and erase operations and standard microprocessor interface. High-performance access times of 60 ns and 80 ns provide zero-wait-state operation with a wide range of microprocessors and microcontrollers. In-system update and upgrade flexibility eliminate the risk of software obsolescence.



297140-1

Product Highlights

- **High-Integration Blocked Architecture**
  - **One 16 KB Boot Block w/Lock Out**
  - **Two 8 KB Parameter Blocks**
  - **One 96 KB Main Block**
  - **Three 128 KB Main Blocks**
- **Hardware Data Protection Feature**
- **Top and Bottom Boot Versions Available**
- **High Performance 60 ns/80 ns Read Access Time**
- **V_{PP} = 12V ±10% (Option)**
- **x8 or x 16 User Configurable I/O Control**
- **I_{CC} Active = 20 mA (Typical)**
- **Deep-Powerdown Mode (0.2 μA Typical)**
- **Automatic Power Savings Feature**
- **100,000 Write-Erase Cycles/Block**
- **Write and Erase Automation**
- **Industry Standard Surface Mount Packages**
  - **40-Ld TSOP (x8 only); 44-Ld PSOP; 56-Ld TSOP**
- **Easy upgrade path from 2 Megabit**
- **Extended Temperature Operation**

Product Description

The 28F400BX-T/B, 28F004BX-T/B Flash Memory Family is a 4 Mbit (4,194,304 bit) family of flash memory devices, each organized as either 524,288 bytes or 262,144 words. These flash memory devices combine the safety of a hardware-lockable 16 Kbyte boot block with the flexibility of separately reprogrammable blocks (two 8 Kbyte parameter blocks, one 96 Kbyte main block and three 128 Kbyte main blocks). The tailored blocking architecture, high-speed performance, x16 organization and small surface mount packaging provide a full-function, nonvolatile flash memory ideal for a wide range of applications, including Telecommunications, Portable Instrumentation, Office/Factory Automation Equipment, and High-end PC boot/BIOS.

The 28F400BX-T and the 28F004BX-T (top boot) devices offer block locations that provide compatibility with microprocessors and microcontrollers that boot from high memory, such as Intel's i386™, i486™, i860™, 80960CA and MCS®-186 families. The 28F400BX-B and the 28F004BX-B (bottom boot) devices are tailored for microprocessors and microcontrollers that boot from low memory, such as Intel's MCS-51, MCS-196, 80960KX and 80960SX microcontrollers.

Order Number: 297140-001

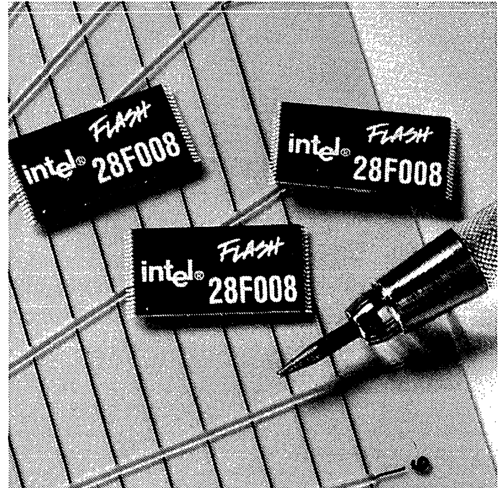
Features	Benefits
— High-Integration Blocked Architecture	— Modular code development/updates
— Top Boot and Bottom Boot Architecture	— Easy interface with wide range of processors
— x8 or x16 User Configurable Bus Operation	— Interfaces with high-end processors
— 60 ns Read Access Time	— DRAM-like access times
— Automated Byte-Write/Block Erase	— Simplifies design, frees processor for higher priority tasks
— Erase Suspend Capability	— Enables data/code access during erase
— Automatic Power Savings Feature	— Saves active power without reducing performance
— 0.2 μA Low Power Sleep Mode	— Extends battery life

intel®

PRODUCT BRIEF

Intel 28F008SA (8-Megabit) FlashFile™ Memory

Intel's symmetrically blocked 28F008SA is the first flash memory optimized for solid state and data file storage. Manufactured on Intel's third generation ETOX™ flash process, the 28F008SA offers the most cost-effective solution for read/write nonvolatile random access system memory. Combined with file management software, such as Microsoft's* Flash File System, the 28F008SA's symmetrically blocked architecture and extended cycling provide high-performance disk emulation capability. The 28F008SA provides new opportunities for system differentiation. High-performance access times of 85/90/120 ns provide instant-on performance for resident application and operating system software. In-system update and upgrade flexibility eliminate the risk of software obsolescence.



297107–1

Product Highlights

- **Symmetrically Blocked Architecture**
- **High Performance 85 ns Read Access Time**
- **100,000 Write-Erase Cycles/Block**
- **0.2 μA Sleep Mode Typical**
- **Byte-Write and Block-Erase Automation**
- **RY/$\overline{\text{BY}}$ Status Output**
- **Erase Suspend Capability**
- **Industry Standard 40-Ld TSOP and 44-Ld PSOP Packaging**

Product Description

The 28F008SA is a high performance 8 Mbit (8,388,608 bit) flash memory, organized as 1 Mbyte (1,048,576 bytes) of 8 bits each. The 28F008SA contains sixteen 64 Kbyte (65,536 byte) blocks. Each block is separately erasable and capable of 100,000 byte write-block erase cycles. On-chip automation dramatically simplifies software algorithms, and frees the system microprocessor to service higher priority tasks. An enhanced system interface allows switching the 28F008SA into a deep powerdown mode during periods of inactivity, and gives a hardware indication of the status of the internal Write State Machine. High-speed access times allow minimal wait-state interfacing to microprocessor buses. Advanced TSOP packaging (standard and reverse pinouts) provides optimum density/in^2.

ExCA and ETOX are trademarks of Intel Corporation.
*Microsoft is a trademark of Microsoft Corporation.

Order Number: 297107-001

Features	Benefits
— Solid State, Symmetrically Blocked Architecture	— High-performance, reliable, disk emulation capability, allows code partitioning
— 85 ns Read Access Time	— Instant-on performance for application and operating system software
— 0.2 μA Low Power Sleep Mode	— Extends battery life
— 100,000 Write-Erase Cycles/Block	— More reliable than disk
— Automated Byte-Write/Block-Erase	— Simplifies design, frees processor for higher priority tasks
— Erase Suspend Capability	— Enables data/code access during erase
— RY/$\overline{BY}$ Status Output	— Hardware indication of WSM, allows interrupt system notification for background erase completion

PRODUCT BRIEF

Intel Flash Memory Evaluation Kit II (D, FLASHEVAL2)

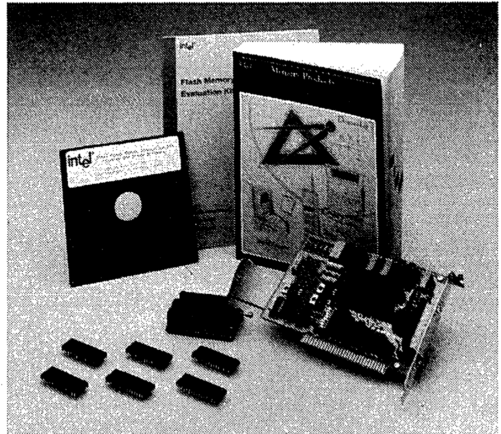
Intel's Flash Memory Evaluation Kit II provides the system designer with a cost-effective prototyping tool for programming and erasing Intel flash memory products. Its software upgrade capability enables Intel to provide programming support for new flash memories conincident with their introduction. The Flash Memory Evaluation Kit's modular design provides compatibility with future Intel flash memory packages and form factors thru hardware adapter upgrade modules. This kit is a significant enhancement to the Intel Flash Memory Evaluation Kit I, adding support for new devices and providing new software with easy user interface, additional capabilities and on-line help.

Upgrade Modules Available

D, FLASHEVAL 3 — Supports Series I Memory Card + 1 MB SIMM

D, FLASHEVAL 4 — Supports 28F008SA

D, FLASHEVAL 5 — Supports 28F400BX/28F004BX



296997-1

Intel Flash Memory Evaluation Kit II

Kit Contents

- (1) PC-AT*/PC-XT* Add-In Driver Board with DIP ZIF Connector
- (1) User's Manual
- Sample Flash Memory Devices, Including:
  - (1) 28F256A (1) 28F512
  - (1) 28F010 (1) 28F020
  - (1) 28F001BX-B (1) 28F001BX-T
- (1) 5.25" Floppy Disk with iFLASH2 Software
- Technical Documentation Describing Intel's Flash Memory Products
- Registration Card

Kit Description

The Intel Flash Memory Evaluation Kit II is a PC-driven flash memory programming solution. With this kit, a system designer can program and erase Intel's flash memory components directly, and flash memory cards, SIMMs and advance packaged components using separately available hardware adapters.

The kit's Users Guide provides software and hardware installation instructions and an overview of software features.

The provided iFLASH2 programming software incorporates a graphical, mouse-driven (optional) user interface. On-line help guides the user through the extensive kit capabilities without need to access instruction manuals. As new flash memories are introduced by Intel, software upgrades will update the installed customer base with the added code.

Technical documentation includes device datasheets, application notes and reliability information. Together, these documents provide a complete description of the technology and important design considerations.

*PC-XT and PC-AT are trademarks of International Business Machines, Inc.

Order Number: 296997-001

Intel 28F008SA (8-Megabit) FlashFile™ Memory Evaluation Module D, FLASHEVAL4

Intel's 28F008SA FlashFile Memory Evaluation Module provides system designers with a cost-effective prototyping tool to evaluate the functionality and benefits of Intel's 28F008SA. This evaluation module is a hardware adapter board upgrade to the Intel Flash Memory Evaluation Kit II (D, FLASHEVAL2). The module supports the 28F008SA in 40-lead TSOP (standard and reverse pinouts) and 44-lead PSOP packages.

297117–1

3

Kit Contents

- **28F008SA Adapter Board with:**
  40-ld TSOP Socket ("E", Standard Pinout)
  40-ld TSOP Socket ("F", Reverse Pinout)
  44-ld PSOP Socket
- **28F008SA Samples:**
  (1) E28F008SA
  (1) F28F008SA
  (1) PA28F008SA
- **5.25" Floppy Disk with iFlash2 Software (Version 2.1)**
- **Technical Documentation Describing Intel's 28F008SA FlashFile Memory**
- **Flash Memory Evaluation Kit II Installation Guide and User's Manual with 28F008SA Adapter Board Installation Instructions**
- **Manual Vacuum Wand**
- **Registration Card**

Kit Description

The 28F008SA Evaluation Module, used with Intel's Flash Memory Evaluation Kit, provides the hardware, software and system interface necessary to evaluate and integrate Intel's 8 Mbit Flash Memory into your next design.

The module provides instructions to install the 28F008SA adapter board. Technical documentation includes 28F008SA datasheets, engineering reports and application notes. Together, they provide a complete description of the technology and important design considerations.

*FlashFile is a trademark of Intel Corporation.

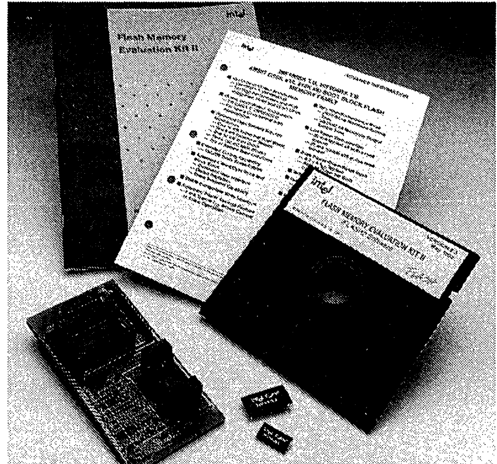
Order Number: 297117-001

intel®

PRODUCT BRIEF

Intel 28F400BX/28F004BX Flash Memory Evaluation Module D, FLASHEVAL5

Intel's 28F400BX/28F004BX Memory Evaluation Module provides system designers with a cost-effective prototyping tool for writing and erasing this flash memory device. This evaluation module is a hardware adapter board upgrade to the Intel Flash Memory Evaluation Kit II (D, FLASHEVAL2) which supports the 4 megabit flash memory in 40-lead TSOP and 44-lead PSOP packages.



297152-1

Kit Contents

- **28F400BX/28F004BX Adapter Board with:**
  40-ld TSOP Socket
  44-ld PSOP Socket
- **28F400BX and 28F004BX Samples:**
  (1) E28F004BX-T (40-ld TSOP, Top boot)
  (1) E28F004BX-B (40-ld TSOP, Bottom boot)
  (1) PA28F400BX-T (44-ld PSOP, Top boot)
  (1) PA28F400BX-B (44-ld PSOP, Bottom boot)
- **5.25" Floppy Disk with iFlash2 Software (Version 2.3)**
- **Technical Documentation Describing Intel's 28F400BX/28F004BX Flash Memory device**
- **Flash Memory Evaluation Kit II Installation Guide and User's Manual with 28F400BX/28F004BX Adapter Board Installation Instructions**
- **Manual Vacuum Wand**
- **Registration Card**

Kit Description

The 28F400BX/28F004BX Evaluation Module, used with Intel's Flash Memory Evaluation Kit, provides the hardware, software and system interface necessary to evaluate and integrate Intel's 4 Mbit Flash Memory into your next design.

The module provides instructions to install the 28F400BX/28F004BX adapter board. Technical documentation includes 28F400BX/28F004BX datasheets, engineering reports and application notes. Together, they provide a complete description of the technology and important design considerations.

Order Number: 297152-001

Flash Memory Systems

4

iMC001FLKA
1-MBYTE FLASH MEMORY CARD

- **Inherent Nonvolatility (Zero Retention Power)**
  — No Batteries Required for Back-Up
- **Over 1,000,000 Hours MTBF**
  — More Reliable than Disk
- **High-Performance Read**
  — 200 ns Maximum Access Time
- **CMOS Low Power Consumption**
  — 25 mA Typical Active Current (X8)
  — 400 μA Typical Standby Current
- **Flash Electrical Zone-Erase**
  — 1 Second Typical per 128 kByte Zone
  — Multiple Zone Erase > 128 kB/s
- **Random Writes to Erased Zones**
  — 10 μs Typical Byte Write
- **Write Protect Switch to Prevent Accidental Data Loss**
- **Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface**
- **ETOX™ II Flash Memory Technology**
  — 5V Read, 12V Erase/Write
  — High-Volume Manufacturing Experience
- **PCMCIA/JEIDA 68-Pin Standard**
  — Byte- or Word-Wide Selectable
- **Independent Software & Hardware Vendor Support**
  — Integrated System Solution Using Flash Filing Systems

Intel's iMC001FLKA Flash Memory Card is the removable solution for storing and transporting important user data and application code. The combination of rewritability and nonvolatility make the Intel Flash Memory Card ideal for data acquisition and updatable firmware applications. Designing with Intel's Flash Memory Card enables OEM system manufacturers to produce portable and dedicated function systems that are higher performance, more rugged, and consume less power.

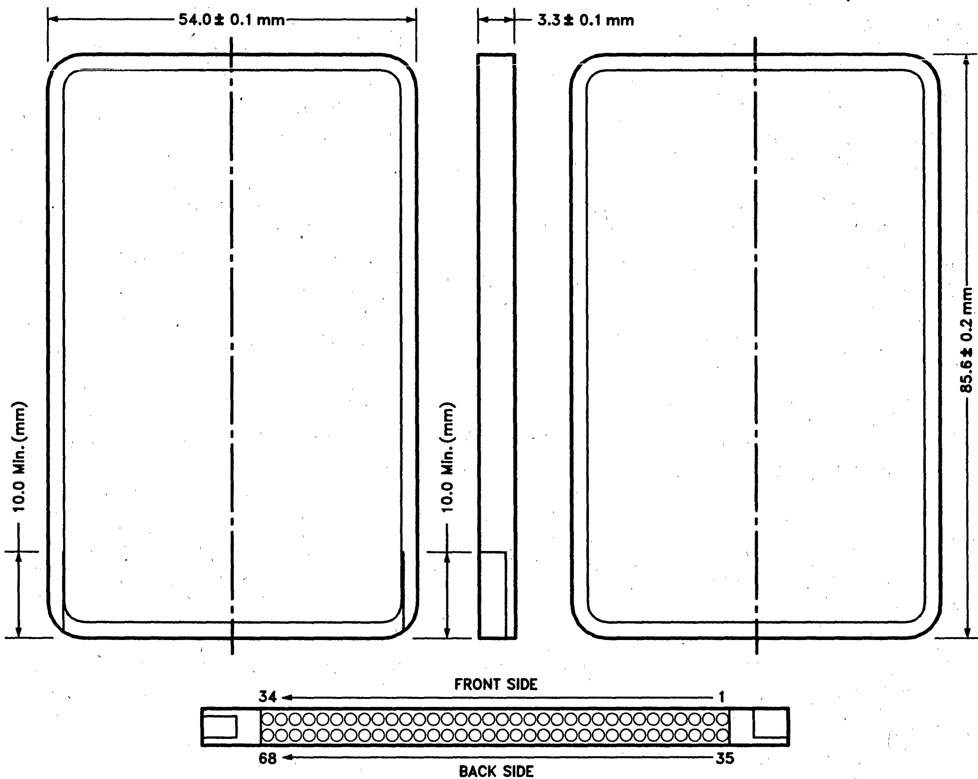
4

The iMC001FLKA conforms to the PCMCIA 1.0 international standard, providing compatibility at the hardware and data interchange level. OEMs may opt to write the Card Information Structure (CIS) at the memory card's address 00000H with a format utility. This information provides data interchange functional compatibility. The 200 ns access time allows for "execute-in-place" capability, for popular low-power microprocessors. Intel's 1-MByte Flash Memory Card operates in a byte-wide and word-wide configuration providing performance/power options for different systems.

Intel's Flash Memory card employs Intel's ETOX™ II Flash Memories. Filing systems, such as Microsoft's* Flash File System (FFS), facilitate data file storage and card erasure using a purely nonvolatile medium in the DOS environment. Flash filing systems, coupled with the Intel Flash Memory Card, effectively create an all-silicon nonvolatile read/write random access memory system that is more reliable and higher performance than disk-based memory systems.

*Microsoft is a trademark of Microsoft Corp.
ExCA is a trademark of Intel Corporation.

October 1992
Order Number: 290399-003



Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	GND	18	V_{PP1}	35	GND	52	V_{PP2}
2	D_3	19	A_{16}	36	$\overline{CD}_1$	53	NC
3	D_4	20	A_{15}	37	D_{11}	54	NC
4	D_5	21	A_{12}	38	D_{12}	55	NC
5	D_6	22	A_7	39	D_{13}	56	NC
6	D_7	23	A_6	40	D_{14}	57	NC
7	$\overline{CE}_1$	24	A_5	41	D_{15}	58	NC
8	A_{10}	25	A_4	42	$\overline{CE}_2$	59	NC
9	$\overline{OE}$	26	A_3	43	NC	60	NC
10	A_{11}	27	A_2	44	NC	61	$\overline{REG}$[1]
11	A_9	28	A_1	45	NC	62	$\overline{BVD}_2$[2]
12	A_8	29	A_0	46	A_{17}	63	$\overline{BVD}_1$[2]
13	A_{13}	30	D_0	47	A_{18}	64	D_8
14	A_{14}	31	D_1	48	A_{19}	65	D_9
15	$\overline{WE}$	32	D_2	49	NC	66	D_{10}
16	NC	33	WP	50	NC	67	$\overline{CD}_2$
17	V_{CC}	34	GND	51	V_{CC}	68	GND

NOTES:
1. $\overline{REG}$ = register memory select = No Connect (NC), unused. When $\overline{REG}$ is brought low, PCMCIA/JEIDA standard card information structure data is expected. This is accomplished by formatting the card with this data.
2. $\overline{BVD}$ = battery detect voltage = Pulled high through pull-up resistor.

Figure 1. iMC001FLKA Pin Configuration

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{19}	I	**ADDRESS INPUTS** for memory locations. Addresses are internally latched during a write cycle.
D_0–D_{15}	I/O	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the card is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}_1$, $\overline{CE}_2$	I	**CARD ENABLE:** Activates the card's high and low byte control logic, input buffers, zone decoders, and associated memory devices. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory card and reduces power consumption to standby levels.
$\overline{OE}$	I	**OUTPUT ENABLE:** Gates the cards output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	I	**WRITE ENABLE** controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **NOTE:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP1}, V_{PP2}		**ERASE/WRITE POWER SUPPLY** for writing the command register, erasing the entire array, or writing bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY** (5V ±5%).
GND		**GROUND**
$\overline{CD}_1$, $\overline{CD}_2$	O	**CARD DETECT:** The card is detected when $\overline{CD}_1$ and $\overline{CD}_2$ = ground.
WP	O	**WRITE PROTECT:** All write operations are disabled with WP = active high.
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating.
$\overline{BVD}_1$, $\overline{BVD}_2$	O	**BATTERY VOLTAGE DETECT:** Not Required.

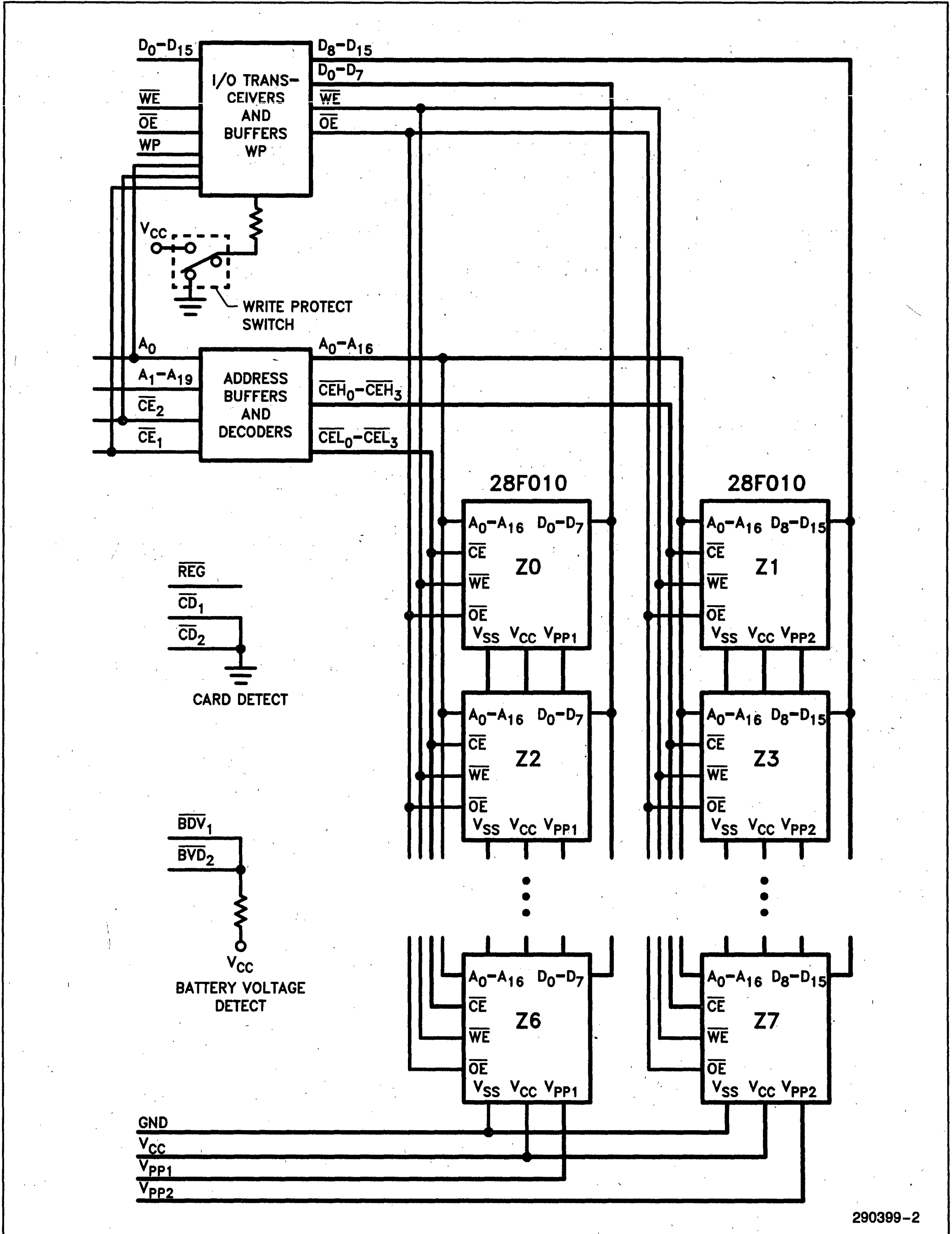


Figure 2. iMC001FLKA Block Diagram

APPLICATIONS

The iMC001FLKA Flash Memory Card allows for the storage of data and application programs on a purely solid-state removable medium. System resident flash filing systems, such as Microsoft's Flash File System, allow Intel's ETOX II highly reliable Flash Memory Card to effectively function as a physical disk drive.

User application software stored on the flash memory card substantially reduces the slow disk-DRAM download process. Replacing the disk results in a dramatic enhancement of read performance and substantial reduction of active power consumption, size, and weight—considerations particularly important in portables and dedicated systems. The iMC001FLKA's high performance read access time and command register microprocessor write interface allows for use of the flash memory system in an "execute-in-place" architecture. This configuration eliminates the need for the redundancy associated with DRAM and Disk memory system architectures. ROM based operating systems, such as Microsoft's MS-DOS ROM Version allow for "instant-on" capability. This enables the design of systems that boot, operate, store data files, and execute application code from/to purely nonvolatile memory.

The PCMCIA/JEIDA 68-pin interface enables the end-user to transport data and application code between portables and host systems. Intel Flash PC cards provide durable nonvolatile memory storage protecting valuable user code and data.

For systems currently using a static RAM/battery configuration for data acquisition, the iMC001FLKA's inherent nonvolatility eliminates the need for battery backup. The concern for battery failure no longer exists, an important consideration for portable computers and medical instruments, both requiring continuous operation. The iMC001FLKA consumes no power when the system is off. In addition, the iMC001FLKA offers a considerable cost and density advantage over memory cards based on static RAM with battery backup.

The flash memory card's electrical zone-erasure, byte writability, and complete nonvolatility fit well with data accumulation and recording needs. Electrical zone-erasure gives the designer the flexibility to selectively rewrite zones of data while saving other zones for infrequently updated look-up tables, for example.

PRINCIPLES OF OPERATION

Intel's Flash Memory Card combines the functionality of two mainstream memory technologies: the rewritability of RAM and the nonvolatility of EPROM. The flash memory card consists of an array of individual memory devices, each of which defines a physical zone. The iMC001FLKA's memory devices erase as individual blocks, equivalent in size to the 128 kByte zone. Multiple zones can be erased simultaneously provided sufficient current for the appropriate number of zones (memory devices). Note, multiple zone erasure requires higher current from both the V_{PP} and V_{CC} power supplies. Erased zones can then be written in bit- or byte-at-a-time fashion and read randomly like RAM. Bit level write capability also supports disk emulation.

In the absence of high voltage on the $V_{PP1/2}$ pins, the iMC001FLKA remains in the read-only mode. Manipulation of the external memory card-control pins yields the standard read, standby, and output disable operations.

The same read, standby, and output disable operations are available when high voltage is applied to the $V_{PP1/2}$ pins. In addition, high voltage on $V_{PP1/2}$

enables erasure and rewriting of the accessed zone(s). All functions associated with altering zone contents—erase, erase verify, write, and write verify—are accessed via the command register.

Commands are written to the internal memory register(s), decoded by zone size, using standard microprocessor write timings. Register contents for a given zone serve as input to that zone's internal state-machine which controls the erase and rewrite circuitry. Write cycles also internally latch addresses and data needed for write and erase operations. With the appropriate command written to the register(s), standard microprocessor read timings output zone data, or output data for erase and write verification.

Byte-Wide or Word-Wide Selection

The flash memory card can be read, erased, and written in a byte-wide or word-wide mode. In the word-wide configuration V_{PP1} and/or $\overline{CE}_1$ control the LO-Byte while V_{PP2} and $\overline{CE}_2$ control the HI-Byte (A_0 = don't care).

Read, Write, and Verify operations are byte- or word-oriented, thus zone independent. Erase Setup and Begin Erase Commands are zone dependent such that commands written to any address within a 128 kByte zone boundary initiate the erase operation in that zone (or two 128 kByte zones under word-wide operation).

Conventional x8 operation uses $\overline{CE}_1$ active-low, with $\overline{CE}_2$ high, to read or write data through the D_0–D_7 only. "Even bytes" are accessed when A_0 is low, corresponding to the low byte of the complete x16 word. When A_0 is high, the "odd byte" is accessed by transposing the high byte of the complete x16 word onto the D_0–D_7 outputs. This odd byte corresponds to data presented on D_8–D_{15} pins in x16 mode.

Note that two zones logically adjacent in x16 mode are multiplexed through D_0–D_7 in x8 mode and are toggled by the A_0 address. Thus, zone specific erase operations must be kept discrete in x8 mode by addressing even bytes only for one-half of the zone pair, then addressing odd bytes only for the other half.

Card Detection

The flash memory card features two card detect pins ($\overline{CD}_{1/2}$) that allow the host system to determine if the card is properly loaded. Note that the two pins are located at opposite ends of the card. Each $\overline{CD}$ output should be read through a port bit. Should only one of the two bits show the card to be present, then the system should instruct the user to re-insert the card squarely into the socket. Card detection can also tell the system whether or not to redirect drives in the case of system booting. $\overline{CD}_{1/2}$ is active low, internally tied to ground.

Write Protection

The flash memory card features three types of write protection. The first type features a mechanical Write Protect Switch that disables the circuitry that control Write Enable to the flash devices. When the switch is activated, $\overline{WE}$ is forced high, which disables any writes to the Command Register. The second type of write protection is based on the PCMCIA/JEIDA socket. Unique pin length assignments provide protective power supply sequencing during hot insertion and removal. The third type operates via software control through the Command Register when the card resides in its connector. The Command Register of each zone is only active when $V_{PP1/2}$ is at high voltage. Depending upon the application, the system designer may choose to make $V_{PP1/2}$ power supply switchable—available only when writes are desired. When $V_{PP1/2} = V_{PPL}$, the contents of the register default to the read command, making the iMC001FLKA a read-only memory card. In this mode, the memory contents cannot be altered.

The system designer may choose to leave $V_{PP1/2} = V_{PPH}$, making the high voltage supply constantly available. In this case, all Command Register functions are inhibited whenever V_{CC} is below the write lockout voltage, V_{LKO}. (See the section on Power Up/Down Protection.) The iMC001FLKA is designed to accommodate either design practice, and to encourage optimization of the processor-memory card interface.

BUS OPERATIONS

Read

The iMC001FLKA has two control functions, both of which must be logically active, to obtain data at the

outputs. Card Enable ($\overline{CE}$) is the power control and should be used for high and/or low zone(s) selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of accessed zone selection. In the byte-wide configuration, only one $\overline{CE}$ is required. The word-wide configuration requires both $\overline{CE}$s active low.

When $V_{PP1/2}$ is high (V_{PPH}), the read operations can be used to access zone data and to access data for write/erase verification. When $V_{PP1/2}$ is low (V_{PPL}), only read accesses to the zone data are allowed.

Output Disable

With Output Enable at a logic-high level (V_{IH}), output from the card is disabled. Output pins are placed in a high-impedance state.

Standby

With one Card Enable at a logic-high level, the standby operation disables one-half of the x16 output's read/write buffer. Further, only the zone corresponding to the selected address within the upper or lower $\overline{CE}_{1,2}$ bank is active at a time. (**NOTE:** A_0 must be low to select the low half of the x16 word when $\overline{CE}_2$ = 1 and $\overline{CE}_1$ = 0.) All other zones are deselected, substantially reducing card power consumption. For deselected banks, the outputs are placed in a high-impedance state, independent of the Output Enable signal. If the iMC001FLKA is deselected during erasure, writing, or write/erase verification, the accessed zone draws active current until the operation is terminated.

Inteligent Identifier Command

The manufacturer- and device-codes can be read via the Command Register, for instances where the iMC001FLKA is erased and rewritten in a universal reader/writer. Following a write of 90H to a zone's Command Register, a read from address location 00000H on any zone outputs the manufacturer code (89H). A read from address 0002H outputs the memory device code (B4H).

Write

Zone erasure and rewriting are accomplished via the Command Register, when high voltage is applied to $V_{PP1/2}$. The contents of the register serve as input to that zone's internal state-machine. The state-machine outputs dictate the function of the targeted zone.

The Command Register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

The Command Register is written by bringing Write Enable to a logic-low level (V_{IL}), while Card Enable(s) is/are low. Addresses are latched on the falling edge of Write Enable, while data is latched on the rising edge of the Write Enable pulse. Standard microprocessor write timings are used.

Refer to AC Write Characteristics and the Erase/ Write Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the V_{PP} pins(s), the contents of the zone Command Register(s) default to 00H, enabling read-only operations.

Placing high voltage on the V_{PP} pin(s) enable(s) read/write operations. Zone operations are selected by writing specific data patterns into the Command Register. Tables 3 and 4 define these iMC001FLKA register commands for both byte-wide and word-wide configurations.

All commands written to the Command Register require that the zone address be valid or the incorrect zone will receive the command. Any Command/ Data Write or Data Read requires the correct valid address.

Table 2. Bus Operations

	Pins / Operation	Notes	[1, 7] V_{PP2}	[1, 7] V_{PP1}	A_0	$\overline{CE_2}$	$\overline{CE_1}$	$\overline{OE}$	$\overline{WE}$	D_8–D_{15}	D_0–D_7
Read-Only	Read (x8)	8	V_{PPL}	V_{PPL}	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Even
	Read (x8)	9	V_{PPL}	V_{PPL}	V_{IH}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Odd
	Read (x8)	10	V_{PPL}	V_{PPL}	X	V_{IL}	V_{IH}	V_{IL}	V_{IH}	Data Out	Tri-state
	Read (x16)	11	V_{PPL}	V_{PPL}	X	V_{IL}	V_{IL}	V_{IL}	V_{IH}	Data Out	Data Out
	Output Disable		V_{PPL}	V_{PPL}	X	X	X	V_{IH}	V_{IH}	Tri-state	Tri-state
	Standby		V_{PPL}	V_{PPL}	X	V_{IH}	V_{IH}	X	X	Tri-state	Tri-state
Read/Write	Read (x8)	3, 8	V_{PPX}	V_{PPH}	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Even
	Read (x8)	3, 9	V_{PPH}	V_{PPX}	V_{IH}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Odd
	Read (x8)	10	V_{PPH}	V_{PPX}	X	V_{IL}	V_{IH}	V_{IL}	V_{IH}	Data Out	Tri-state
	Read (x16)	3, 11	V_{PPH}	V_{PPH}	X	V_{IL}	V_{IL}	V_{IL}	V_{IH}	Data Out	Data Out
	Write (x8)	5, 8	V_{PPX}	V_{PPH}	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{IL}	Tri-state	Data In-Even
	Write (x8)	9	V_{PPH}	V_{PPX}	V_{IH}	V_{IH}	V_{IL}	V_{IH}	V_{IL}	Tri-state	Data In-Odd
	Write (x8)	10	V_{PPH}	V_{PPX}	X	V_{IL}	V_{IH}	V_{IH}	V_{IL}	Data In	Tri-state
	Write (x16)	11	V_{PPH}	V_{PPH}	X	V_{IL}	V_{IL}	V_{IH}	V_{IL}	Data In	Data In
	Standby	4	V_{PPH}	V_{PPH}	X	V_{IH}	V_{IH}	X	X	Tri-state	Tri-state
	Output Disable		V_{PPH}	V_{PPH}	X	X	X	V_{IH}	V_{IL}	Tri-state	Tri-state

NOTES:

1. Refer to DC Characteristics. When $V_{PP1/2} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes may be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
3. Read operations with $V_{PP1/2} = V_{PPH}$ may access array data or the inteligent Identifier codes.
4. With $V_{PP1/2}$ at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
5. Refer to Table 3 for valid Data-In during a write operation.
6. X can be V_{IL} or V_{IH}.
7. $V_{PPX} = V_{PPH}$ or V_{PPL}.
8. This x8 operation reads or writes the low byte of the x16 word on DQ_{0-7}, i.e., A_0 low reads "even" byte in x8 mode.
9. This x8 operation reads or writes the high byte of the x16 word on DQ_{0-7} (transposed from DQ_{8-15}), i.e., A_0 high reads "odd" byte in x8 mode.
10. This x8 operation reads or writes the high byte of the x16 on DQ_{8-15}. A_0 is "don't care".
11. A_0 is "don't care", unused in x16 mode. High and low bytes are presented simultaneously.

Table 3. Command Definitions Byte-Wide Mode

Command	Notes	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
			Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory		1	Write	RA	00H			
Read inteligent Identifier Codes	4	3	Write	IA	90H	Read		
Set-Up Erase/Erase	5	2	Write	ZA	20H	Write	ZA	20H
Erase Verify	5	2	Write	EA	A0H	Read	EA	EVD
Set-Up Write/Write	6	2	Write	WA	40H	Write	WA	WD
Write Verify	6	2	Write	WA	C0H	Read	WA	WVD
Reset	7	2	Write	ZA	FFH	Write	ZA	FFH

Table 4. Command Definitions Word-Wide Mode

Command	Notes	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
			Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory		1	Write	RA	0000H			
Read inteligent Identifier Codes	4	3	Write	IA	9090H	Read		
Set-Up Erase/Erase	5	2	Write	ZA	2020H	Write	ZA	2020H
Erase Verify	5	2	Write	EA	A0A0H	Read	EA	EVD
Set-Up Write/Write	6	2	Write	WA	4040H	Write	WA	WD
Write Verify	6	2	Write	WA	C0C0H	Read	WA	WVD
Reset	7	2	Write	ZA	FFFFH	Write	ZA	FFFFH

NOTES:

1. Bus operations are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   RA = Read Address
   WA = Address of memory location to be written.
   ZA = Address of 128 kByte zones involved in erase operation.
   Addresses are latched on the falling edge of the Write Enable pulse.
3. ID = Data read from location IA during device indentification. (Mfr = 89H, Device = B4H).
   EVD = Data read from location EA during erase verify.
   WD = Data to be written at location WA. Data is latched on the rsing edge of Write Enable.
   WVD = Data read from location WA during write verify. WA is latched on the Write command.
4. Following the Read Inteligent ID command, two read operations access manufacturer and device codes.
5. Figure 5 illustrates the Erase Algorithm.
6. Figure 6 illustrates the Write Algorithm.
7. The second bus cycle must be followed by the desired command register write.
8. The Reset command operation on Zone Basic to Reset entire Card, requires reset Write cycles to each zone.

Read Command

While $V_{PP1/2}$ is high, for erasure and writing, zone memory contents can be accessed via the read command. The read operation is initiated by writing 00H (0000H for the word-wide configuration) into the zone Command Register(s). Microprocessor read cycles retrieve zone data. The accessed zone remains enabled for reads until the Command Register(s) contents are altered.

The default contents of each zone's register(s) upon $V_{PP1/2}$ power-up is 00H (00000H for word-wide). This default value ensures that no spurious alteration of memory card contents occurs during the $V_{PP1/2}$ power transition. Where the $V_{PP1/2}$ supply is left at V_{PPH}, the memory card powers up and remains enabled for reads until the command Register contents of targeted zones are changed. Refer to the AC Read Characteristics and Waveforms for specific timing parameters.

inteligent Identifier Command

Each zone of the iMC001FLKA contains an inteligent Identifier to identify memory card device characteristics. The operation is initiated by writing 90H (9090H for word-wide) into the Command Register(s) with zone address. Following the command write, a read cycle from address 00000H retrieves the manufacturer code 89H (8989H for word-wide). A read cycle from address 0002H returns the device code B4H (B4B4H for word-wide). To terminate the operation, it is necessary to write another valid command into the register(s).

Set-Up Erase/Erase Commands

Set-Up Erase stages the targeted zone(s) for electrical erasure of all bytes in the zone. The set-up erase operation is performed by writing 20H to the Command Register (2020H for word-wide) with zone address.

To commence zone-erasure, the erase command (20H or 2020H) must again be written to the register(s) with zone address. The erase operation begins with the rising edge of the Write-Enable pulse and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that zone memory contents are not accidentally erased. Also, zone-erasure can only occur when high voltage is applied to the $V_{PP1/2}$ pins. In the absence of this high voltage, zone memory contents are protected against erasure. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all of the bytes of the zone in parallel. After each erase operation, all bytes in the zone must be individually verified. In byte-mode operations, zones are segregated by A_0 in odd and even banks; erase and erase verify operations must be done in complete passes of even-bytes-only then odd-bytes-only. See the Erase Algorithm for byte-wide mode. The erase verify operation is initiated by writing A0H (A0A0H for word-wide) into the Command Register(s). The address for the byte(s) to be verified must be supplied as it is latched on the falling edge of the Write Enable pulse. The register write terminates the erase operation with the rising edge of its Write Enable pulse.

The enabled zone applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased. Similarly, reading FFFFH from the addressed word indicates that all bits in the word are erased.

The erase-verify command must be written to the Command Register prior to each byte (word) verification to latch its address. The process continues for each byte (word) in the zone(s) until a byte (word) does not return FFH (FFFFH) data, or the last address is accessed.

In the case where the data read is not FFH (FFFFH), another erase operation is performed. (Refer to Set-Up Erase/Erase.) Verification then resumes from the address of the last-verified byte (word). Once all bytes (words) in the zone(s) have been verified, the erase step is complete. The accessed zone can now be written. At this point, the verify operation is terminated by writing a valid command (e.g., Write Set-Up) to the Command Register. The Erase algorithms for byte-wide and word-wide configurations illustrate how commands and bus operations are combined to perform electrical erasure of the iMC001FLKA. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Set-Up Write/Write Commands

Set-Up write is a command-only operation that stages the targeted zone for byte writing. Writing 40H (4040H) into the Command Register(s) performs the set-up operation.

Once the write set-up operation is performed, the next Write Enable pulse causes a transition to an active write operation. Addresses are internally latched on the falling edge of the Write Enable pulse. Data is internally latched on the rising edge of the Write Enable pulse. The rising edge of Write Enable also begins the write operation. The write operation terminates with the next rising edge of Write Enable, which is used to write the verify command. Refer to AC Write Characteristics and Waveforms for specific timing parameters.

Write Verify Command

The iMC001FLKA is written on a byte-by-byte or word-by-word basis. Byte or word writing may occur sequentially or at random. Following each write operation, the byte or word just written must be verified.

The write-verify operation is initiated by writing C0H (C0C0H) into the Command Register(s) with correct address. The register write(s) terminate(s) the write operation with the rising edge of its Write Enable pulse. The write-verify operation stages the accessed zone(s) for verification of the byte or word last written. The zone(s) apply(ies) an internally-generated margin voltage to the byte or word. A microprocessor read cycle outputs the data. A successful comparison between the written byte or word and true data means that the byte or word is successfully written. The write operation then proceeds to the next desired byte or word location. The Write algorithms for byte-wide and word-wide configurations illustrate how commands are combined with bus operations to perform byte and word writes. Refer to AC Write Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or write-command sequences. Following either set-up command (erase or write) with two consecutive writes of FFH (FFFFH or word-wide) will safely abort the operation. Zone memory contents will not be altered. A valid command must then be written to place the accessed zone in the desired state.

EXTENDED ERASE/WRITE CYCLING

Intel has designed extended cycling capability into its ETOX II flash memory technology enabling a flash memory card with a MTBF that is approximately 20 times more reliable than rotating disk technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field minimizes the probability of oxide defects in the region. The lower electric field greatly reduces oxide stress and the probability of failure.

WRITE ALGORITHMS

The write algorithm(s) use write operations of 10 μs duration. Each operation is followed by a byte or word verification to determine when the addressed byte or word has been successfully written. The algorithm(s) allows for up to 25 write operations per byte or word, although most bytes and words verify on the first or second operation. The entire sequence of writing and byte/word verification is performed with V_{PP} at high voltage.

ERASE ALGORITHM

The Erase algorithm(s) yield(s) fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the write algorithm, to simultaneously remove charge from all bits in the accessed zone(s).

Erasure begins with a read of memory zone contents. Reading FFH (FFFFH) data from the accessed zone(s) can be immediately followed by writing to the desired zone(s).

For zones being erased and rewritten, uniform and reliable erasure is ensured by first writing all bits in the accessed zone to their charged state (data = 00H byte-wide, 00000H word-wide). This is accomplished, using the write algorithm, in approximately two seconds per zone.

Erase execution then continues with an initial erase operation. Erase veification (data = FFH byte-wide, FFFFH word-wide) begins at address 00000H and continues through the zone to the last address, or until data other than FFH (FFFFH) is encountered. (**Note:** byte-wide erase operation requires separate even- and odd-address passes to handle the individual 128 kByte zones.) With each erase operation, an increasing number of bytes or words verify to the erased state. Erase efficiency may be improved by storing the address of the last byte or word verified in a register(s). Following the next erase operation, verification starts at that stored address location. Follow this procedure until all bytes in the zone are erased. Then, re-start the procedure for the next zone or word-wide zone pair. Erasure typically occurs in one second per zone.

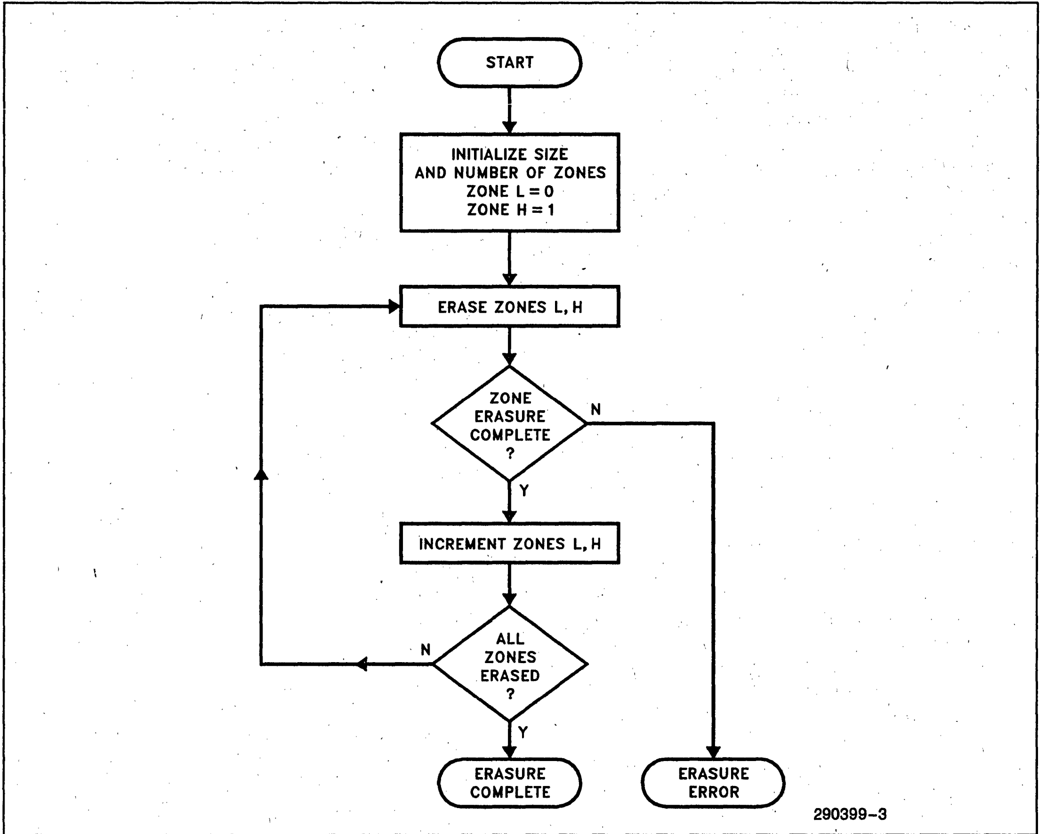
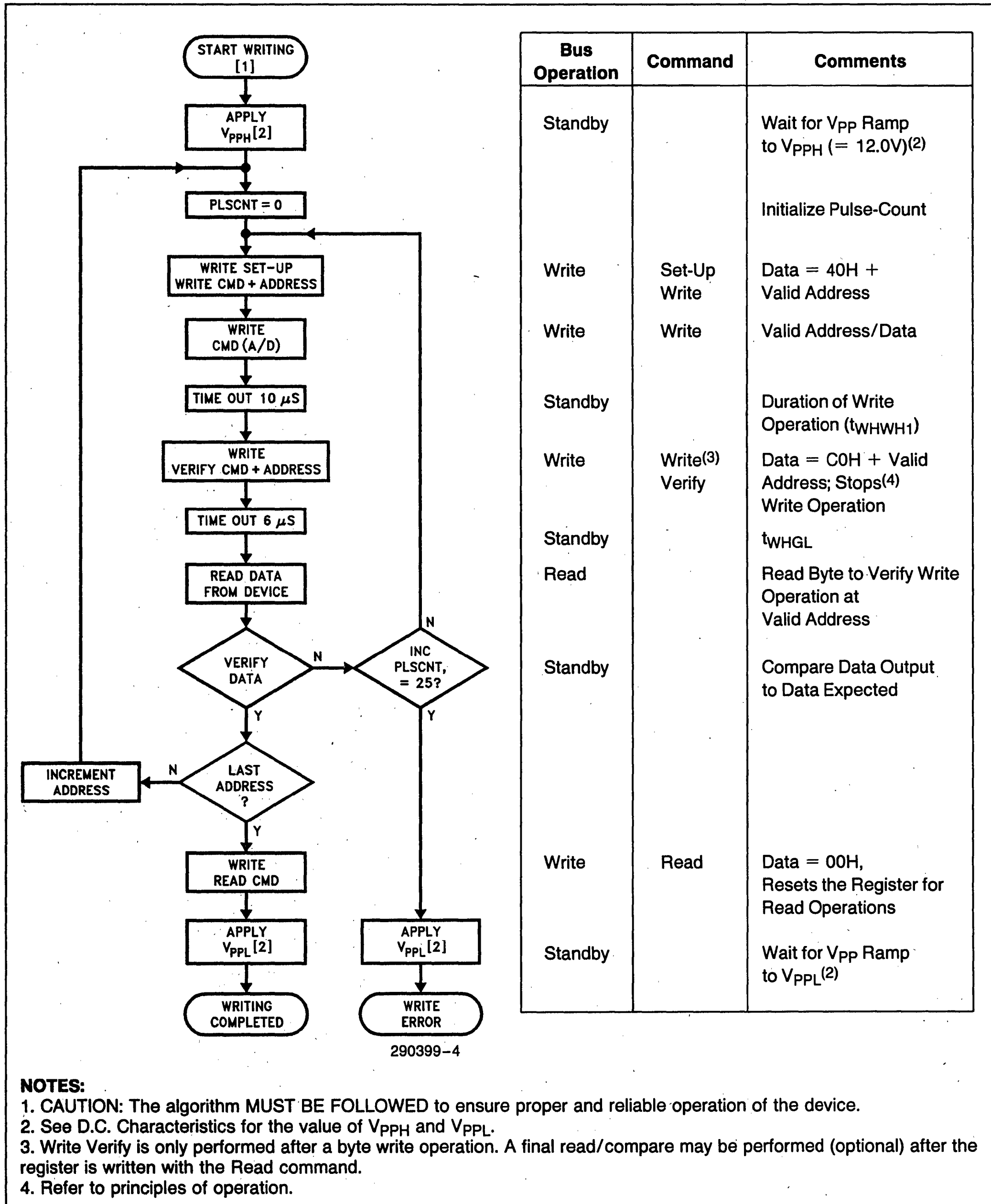


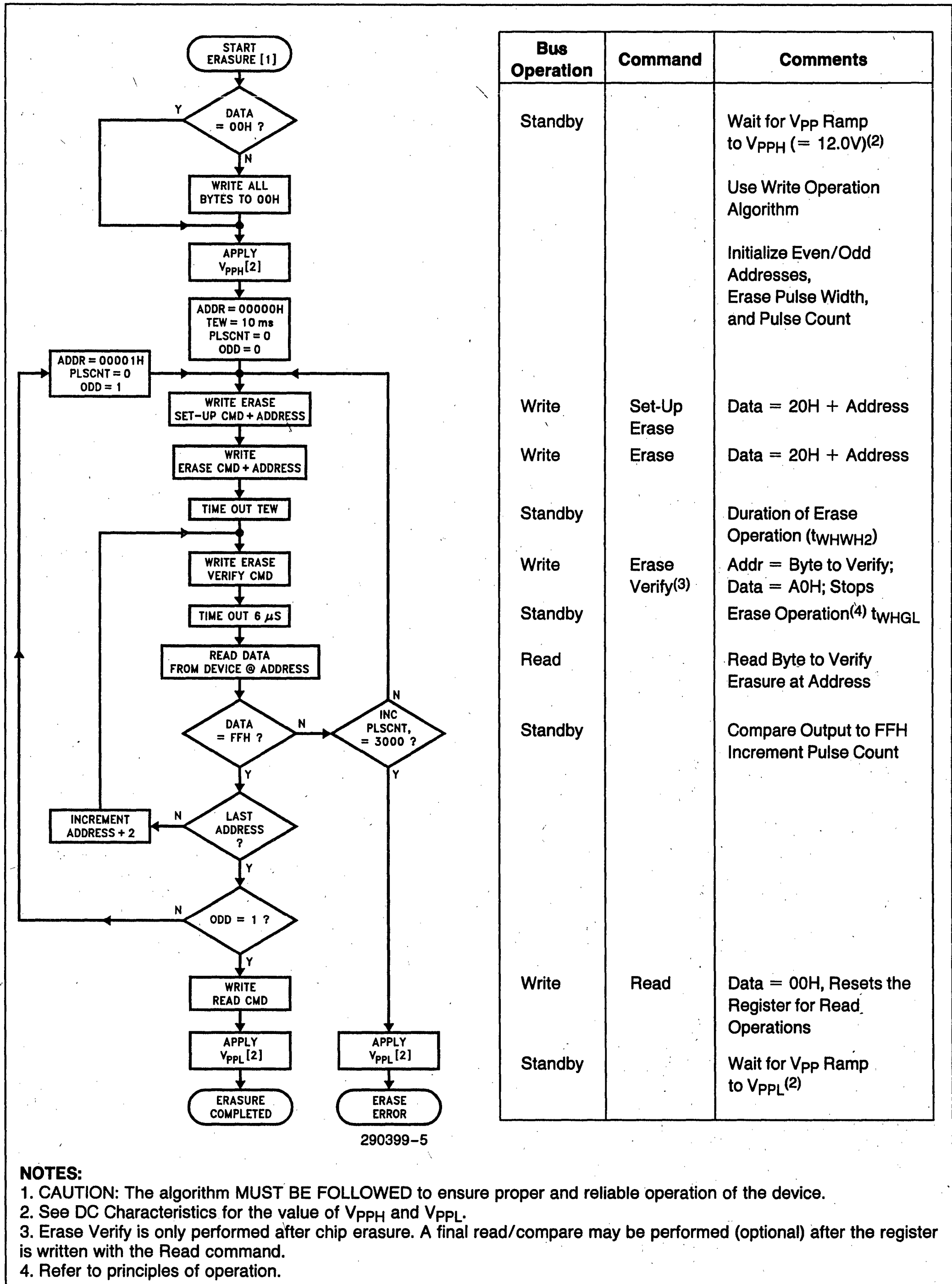
Figure 3. Full Card Erase Flow



Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH} (= 12.0V)(2)
		Initialize Pulse-Count
Write	Set-Up Write	Data = 40H + Valid Address
Write	Write	Valid Address/Data
Standby		Duration of Write Operation (t_{WHWH1})
Write	Write(3) Verify	Data = C0H + Valid Address; Stops(4) Write Operation
Standby		t_{WHGL}
Read		Read Byte to Verify Write Operation at Valid Address
Standby		Compare Data Output to Data Expected
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(2)

NOTES:
1. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.
2. See D.C. Characteristics for the value of V_{PPH} and V_{PPL}.
3. Write Verify is only performed after a byte write operation. A final read/compare may be performed (optional) after the register is written with the Read command.
4. Refer to principles of operation.

Figure 4. Write Algorithm for Byte-Wide Mode



Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH} (= 12.0V)(2)
		Use Write Operation Algorithm
		Initialize Even/Odd Addresses, Erase Pulse Width, and Pulse Count
Write	Set-Up Erase	Data = 20H + Address
Write	Erase	Data = 20H + Address
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase Verify(3)	Addr = Byte to Verify; Data = A0H; Stops
Standby		Erase Operation(4) t_{WHGL}
Read		Read Byte to Verify Erasure at Address
Standby		Compare Output to FFH Increment Pulse Count
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(2)

NOTES:
1. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.
2. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
3. Erase Verify is only performed after chip erasure. A final read/compare may be performed (optional) after the register is written with the Read command.
4. Refer to principles of operation.

Figure 5. Erase Algorithm for Byte-Wide Mode

Start Write Operation

Apply V_{PPH}

Get ADRS/W_DAT

Initialize:
V_DAT = W_DAT
W_COM = 4040H
V_COM = C0C0H
PLSCNT_HI = 0
PLSCNT_LO = 0
FLAG = 0

Write ADRS/W_COM + VALID ADDRESS

Write ADRS/V_DAT

Time Out 10 μS

Write ADRS/V_COM

Time Out 6 μS

Read ADRS/F_DAT

F_DAT = W_DAT ? (Y → More Data ?; N → Flag = 0 ?)

Flag = 0 ? (Y → High/Low Byte Compare & Mask Subroutine → Write ADRS/W_COM + VALID ADDRESS; N → Write READ_COM)

More Data ? (Y → Get ADRS/W_DAT; N → Write READ_COM)

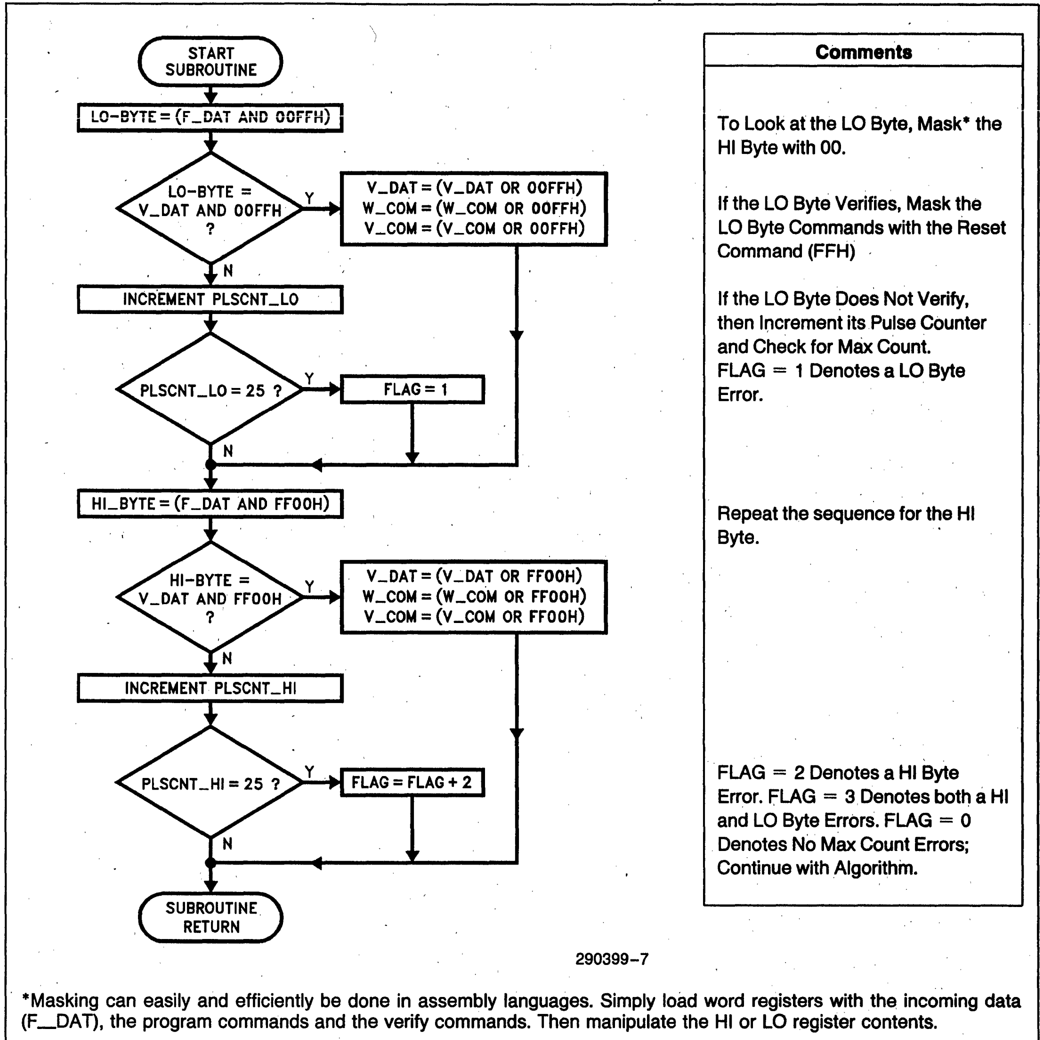
Write READ_COM → Apply V_{PPL} → Write Complete

Write READ_COM → Apply V_{PPL} → Write Error

Comments
Wait for V_{PP} ramp to V_{PPH}
ADRS = Address to Write W_DAT = Data Word to Write
Initialize Data Word Variables: V_DAT = Valid Data W_COM = Write Command V_COM = Write Verify Command PLSCNT_HI = HI Byte Pulse Counter PLSCNT_LO = LO Byte Pulse Counter FLAG = Write Error Flag
Write Set-Up Command Address needs to be Valid
Write
See Write Verify and Mask Subroutine
Write Verify Command
F_DAT = Flash Memory Data
Compare Flash Memory Data to Valid Data (Word Compare). If Not Equal, Check for Write Error Flag. If Flag Not Set, Compare High and Low Bytes in the Subroutine.
Check Buffer of I/O Port for More Data to Write.
Reset Device for Read Operation.
Turn off V_{PP}.

290399-6

Figure 6. Write Algorithm for Word-Wide Mode



*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming data (F_DAT), the program commands and the verify commands. Then manipulate the HI or LO register contents.

Figure 7. Write Verify and Mask Subroutine for Word-Wide Mode

Flowchart	Comments
START ERASURE	
APPLY V_{PPH}	Wait for V_{PP} to stabilize.
WRITE ALL DEVICES TO 00H	Use Write Operation Algorithm in x8 or x16 Configuration.
INITIALIZE: PLSCNT_HI = 0, PLSCNT_LO = 0, ADRS = 0, FLAG = 0, E_COM = 2020H, V_COM = A0A0H	Initialize Variables: PLSCNT_HI = HI Byte Pulse Counter; PLSCNT_LO = LO Byte Pulse Counter; FLAG = Erasure Error Flag; ADRS = Address; E_COM = Erase Command; V_COM = Verify Command
WRITE ADRS/E_COM	Erase Set-Up Command
WRITE ADRS/E_COM	Start Erasing
TIME OUT 10 MS	Duration of Erase Operation.
Write ADRS/V_COM	Erase Verify Command Stops Erasure.
BLOCK ERASE VERIFY & MASK SUBROUTINE	See Block Erase Verify and Mask Subroutine
F_DAT = FFFFH ? (N → FLAG = 0 ? ; Y → continue) FLAG = 0 ? (Y → return to WRITE ADRS/E_COM; N → WRITE READ_COM, APPLY V_{PPL}, ERASE ERROR)	When Both Devices at ADRS Are Erased, F_DATA = FFFFH. If Not Equal, Increment the Pulse Counter and Check for Last Pulse.
RESET V_COM, E_COM	Reset Commands Default to (E_COM = 2020H) (V_COM = A0A0H) before Verifying Next ADRS.
INCREMENT ADRS	
LAST ADDRESS ? (N → return to Write ADRS/V_COM; Y → continue)	
WRITE READ_COM	Reset Device for Read Operation.
APPLY V_{PPL}	Turn Off V_{PP}.
ERASURE COMPLETE	

290399–8

NOTES:
x16 Addressing uses A_1–A_{19} only. A_0 = 0 throughout word-wide operation.

Figure 8. Erase Algorithm for Word-Wide Mode

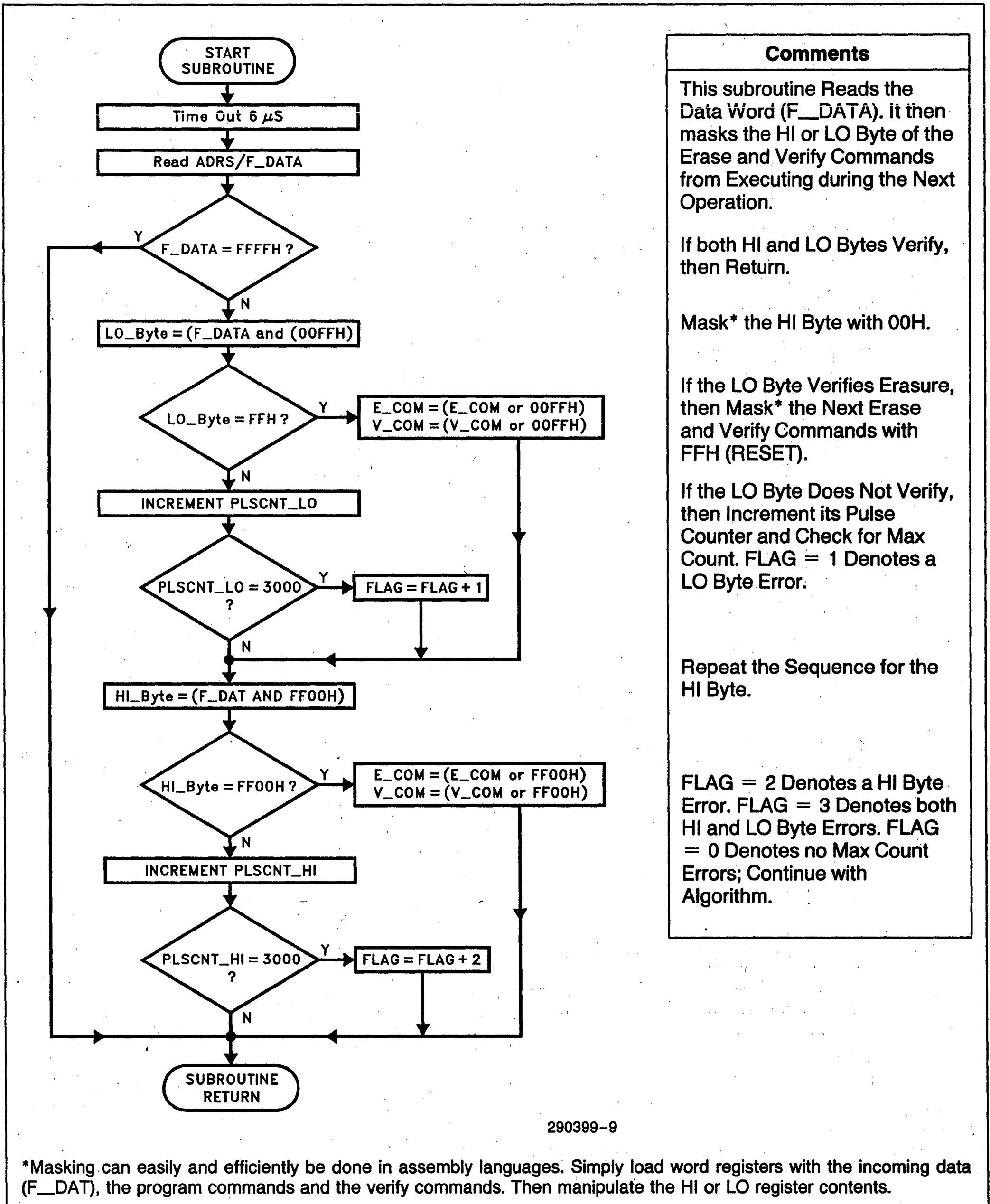


Figure 9. Erase Verify and Mask Subroutine for Word-Wide Mode

SYSTEM DESIGN CONSIDERATIONS

Three-Line Control

Three-line control provides for:

a. the lowest possible power dissipation and,

b. complete assurance that output bus contention will not occur.

To efficiently use these three control inputs, an address-decoder output should drive $\overline{CE}_{1,2}$, while the system's Read signal controls the card $\overline{OE}$ signal, and other parallel zones. This, coupled with the internal zone decoder, assures that only enabled memory zones have active outputs, while deselected zones maintain the low power standby condition.

Power-Supply Decoupling

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current issues—standby, active and transient current peaks, produced by falling and rising edges of $\overline{CE}_{1/2}$. The capacitive and inductive loads on the card and internal flash memory zones determine the magnitudes of these peaks.

Three-line control and proper decoupling capacitor selection will suppress transient voltage peaks. The iMC001FLKA features on-card ceramic decoupling capacitors connected between V_{CC} and V_{SS}, and between V_{PP1}/V_{PP2} and V_{SS}.

The card connector should also have a 4.7 µF electrolytic capacitor between V_{CC} and V_{SS}, as well as between V_{PP1}/V_{PP2} and V_{SS}. The bulk capacitors will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

Power Up/Down Protection

The PCMCIA/JEIDA socket is specified, via unique Pin lengths, to properly sequence the power supplies to the flash memory card. This assures that hot insertion and removal will not result in card damage or data loss.

Each zone in the iMC001FLKA is designed to offer protection against accidental erasure or writing, caused by spurious system-level signals that may exist during power transitions. The card will power-up into the read state.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}_{1,2}$ must be low for a command write, driving either to V_{IH} will inhibit writes. With its control register architecture, alteration of zone contents only occurs after successful completion of the two-step command sequences.

While these precautions are sufficient for most applications, it is recommended that V_{CC} reach its steady state value before raising $V_{PP1/2}$ above V_{CC} + 2.0V. In addition, upon powering-down, $V_{PP1/2}$ should be below V_{CC} + 2.0V, before lowering V_{CC}.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read .... 0°C to +60°C(1)
During Erase/Write .... 0°C to +60°C

Temperature under Bias .... −10°C to +70°C

Storage Temperature .... −30°C to +70°C

Voltage on Any Pin with
Respect to Ground .... −2.0V to +7.0V(2)

V_{PP1}/V_{PP2} Supply Voltage
with Respect to Ground
during Erase/Write .... −2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage with
Respect to Ground .... −2.0V to +7.0V(2)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

***** *WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
3. Maximum DC input voltage on V_{PP1}/V_{PP2} may overshoot to +14.0V for periods less than 20 ns.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature	0	60	°C	For Read-Only and Read/Write Operations
V_{CC}	V_{CC} Supply Voltage	4.75	5.25	V	
V_{PPH}	Active V_{PP1}, V_{PP2} Supply Voltages	11.40	12.60	V	
V_{PPL}	V_{PP} during Read Only Operations	0.00	6.50	V	

DC CHARACTERISTICS—Byte Wide Mode

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ	Max		
I_{LI}	Input Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.4	0.8	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC} = \pm 0.2V$
				4	7	mA	$\overline{CE} = V_{IH}$, $V_{CC} = V_{CC}$ Max
I_{CC1}	V_{CC} Active Read Current	1, 2		25	50	mA	$V_{CC} = V_{CC}$, Max $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Write Current	1, 2		5.0	15.0	mA	Writing in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		10.0	20.0	mA	Erasure in Progress
I_{CC4}	V_{CC} Write Verify Current	1, 2		10.0	20.0	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		10.0	20.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±80	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	1, 3		0.4	0.8	mA	$V_{PP} > V_{CC}$
					±0.08	mA	$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Write Current	1, 3		8.0	30	mA	$V_{PP} = V_{PPH}$ Write in Progress
I_{PP3}	V_{PP} Erase Current	1, 3		10	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Write Verify Current	1, 3		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 3		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.4		V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage				0.40	V	I_{OL} = 3.2 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		3.8			V	I_{OH} = −2.0 mA $V_{CC} = V_{CC}$ Min
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**Note:** Erase/Write are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C.
2. 1 chip active and 7 in standby for byte-wide mode.
3. Assumes 1 V_{PP} is active.

DC CHARACTERISTICS—Word Wide Mode

Symbol	Parameter	Notes	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
I_{LI}	Input Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.4	0.8	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{CC} \pm 0.2V$
				4	7	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1, 2		40	80	mA	$V_{CC} = V_{CC}$, Max $\overline{CE} = V_{IL}$ f = 6 MHz, $I_{OUT} = 0$ mA
I_{CC2}	V_{CC} Write Current	1, 2		5.0	25	mA	Writing in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		15.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Write Verify Current	1, 2		15.0	30	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		15.0	30	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±80	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	1, 3		0.7	1.6	mA	$V_{PP} \geq V_{CC}$
					±0.16		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Write Current	1, 3		16	60	mA	$V_{PP} = V_{PPH}$ Write in Progress
I_{PP3}	V_{PP} Erase Current	1, 3		20	60	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Write Verify Current	1, 3		5.0	12	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 3		5.0	12	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.4		$V_{CC} + 0.3$	V	
V_{OL}	Output Low Voltage				0.40	V	$I_{OL} = 3.2$ mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		3.8			V	$I_{OH} = -2.0$ mA $V_{CC} = V_{CC}$ Min
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**Note:** Erase/Write are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at $V_{CC} = 5.0V$, $V_{PP} = 12.0V$, T = 25°C.
2. 2 chips active and 6 in standby for word-wide mode.
3. Assumes 2 V_{PP}s are active.

CAPACITANCE T = 25°C, f = 1.0 MHz

Symbol	Parameter	Notes	Limits		Unit	Conditions
			Min	Max		
C_{IN1}	Address Capacitance			40	pF	$V_{IN} = 0V$
C_{IN2}	Control Capacitance			40	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance			40	pF	$V_{OUT} = 0V$
$C_{I/O}$	I/O Capacitance			40	pF	$V_{I/O} = 0V$

AC TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ...... 10 ns
Input Pulse Levels .... V_{OL} and V_{OH1}
Input Timing Reference Level .... V_{IL} and V_{IH}
Output Timing Reference Level .... V_{IL} and V_{IH}

AC CHARACTERISTICS—Read-Only Operations

Symbol	Parameter	Notes	Min	Max	Unit
t_{AVAV}/t_{RC}	Read Cycle Time	2	200		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time	2		200	ns
t_{AVQV}/t_{ACC}	Address Access Time	2		200	ns
t_{GLQV}/t_{OE}	Output Enable Access Time	2		100	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	2	5		ns
t_{EHQZ}	Chip Disable to Output in High Z	2		60	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	2	5		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	2		60	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	1, 2	5		ns
t_{WHGL}	Write Recovery Time before Read	2	5		µs

NOTES:
1. Whichever occurs first.
2. Rise/Fall time ≤ 10 ns.

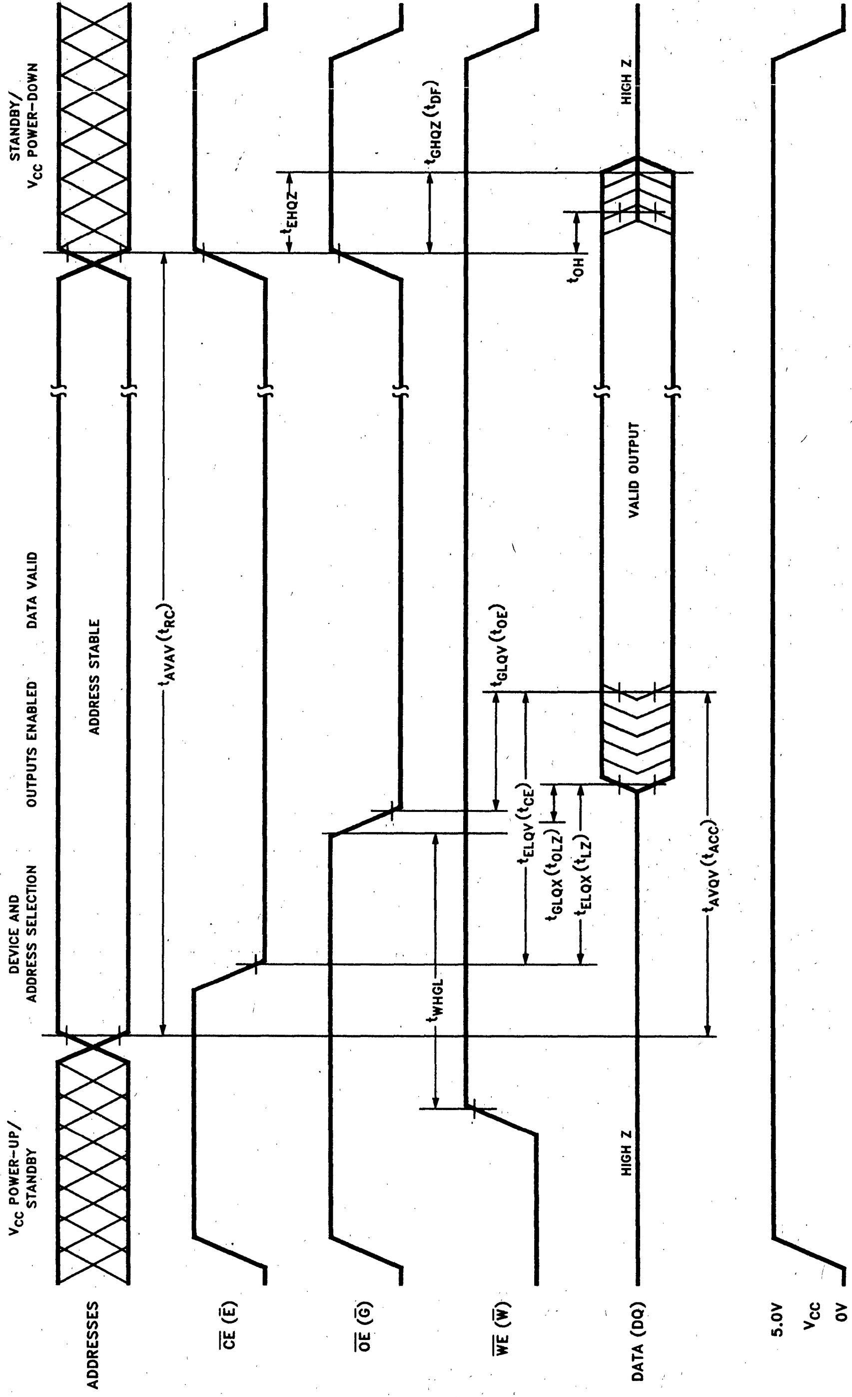


Figure 10. AC Waveforms for Read Operations

AC CHARACTERISTICS—For Write/Erase Operations

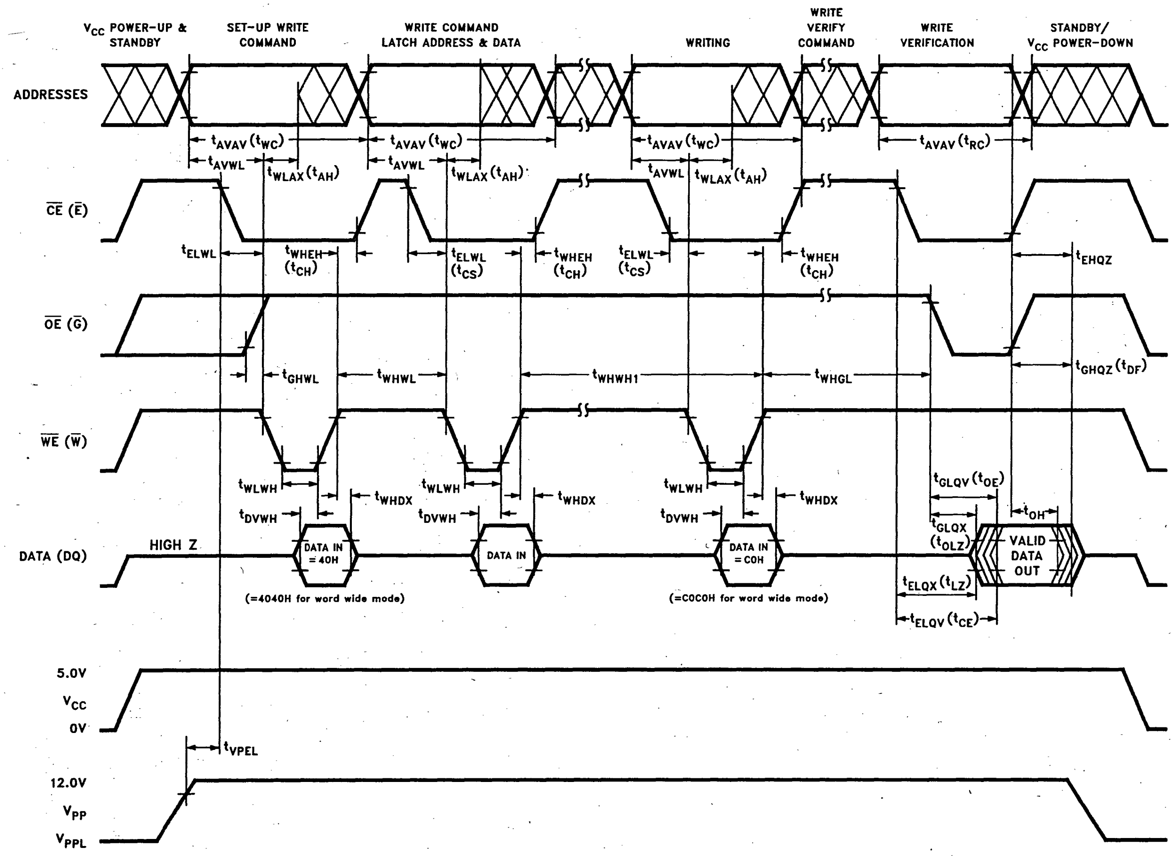
Symbol	Parameter	Notes	Min	Max	Unit
t_{AVAV}/t_{WC}	Write Cycle Time	1, 2	200		ns
t_{AVWL}/t_{AS}	Address Set-Up Time	1, 2	0		ns
t_{WLAX}/t_{AH}	Address Hold Time	1, 2	100		ns
t_{DVWH}/t_{DS}	Data Set-Up Time	1, 2	80		ns
t_{WHDX}/t_{DH}	Data Hold Time	1, 2	30		ns
t_{WHGL}	Write Recovery Time before Read	1, 2	6		µs
t_{GHWL}	Read Recovery Time before Write	1, 2	0		µs
t_{WLOZ}	Output High-Z from Write Enable	1, 2	5		ns
t_{WHOX}	Output Low-Z from Write Enable	1, 2		60	ns
t_{ELWL}/t_{CS}	Chip Enable Set-Up Time before Write	1, 2	40		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time	1, 2	0		ns
t_{WLWH}/t_{WP}	Write Pulse Width	1, 2	100		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High	1, 2	20		ns
t_{WHWH1}	Duration of Write Operation	1, 2, 3	10		µs
t_{WHWH2}	Duration of Erase Operation	1, 2, 3	9.5		ms
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low	1, 2	100		ns

NOTES:
1. Read timing parameters during read/write operations are the same as during read-only operations. Refer to A.C. Characteristics for Read-Only Operations.
2. Rise/Fall time ≤ 10 ns.
3. The integrated stop timer terminates the write/erase operations, thereby eliminating the need for a maximum specification.

ERASE/WRITE PERFORMANCE

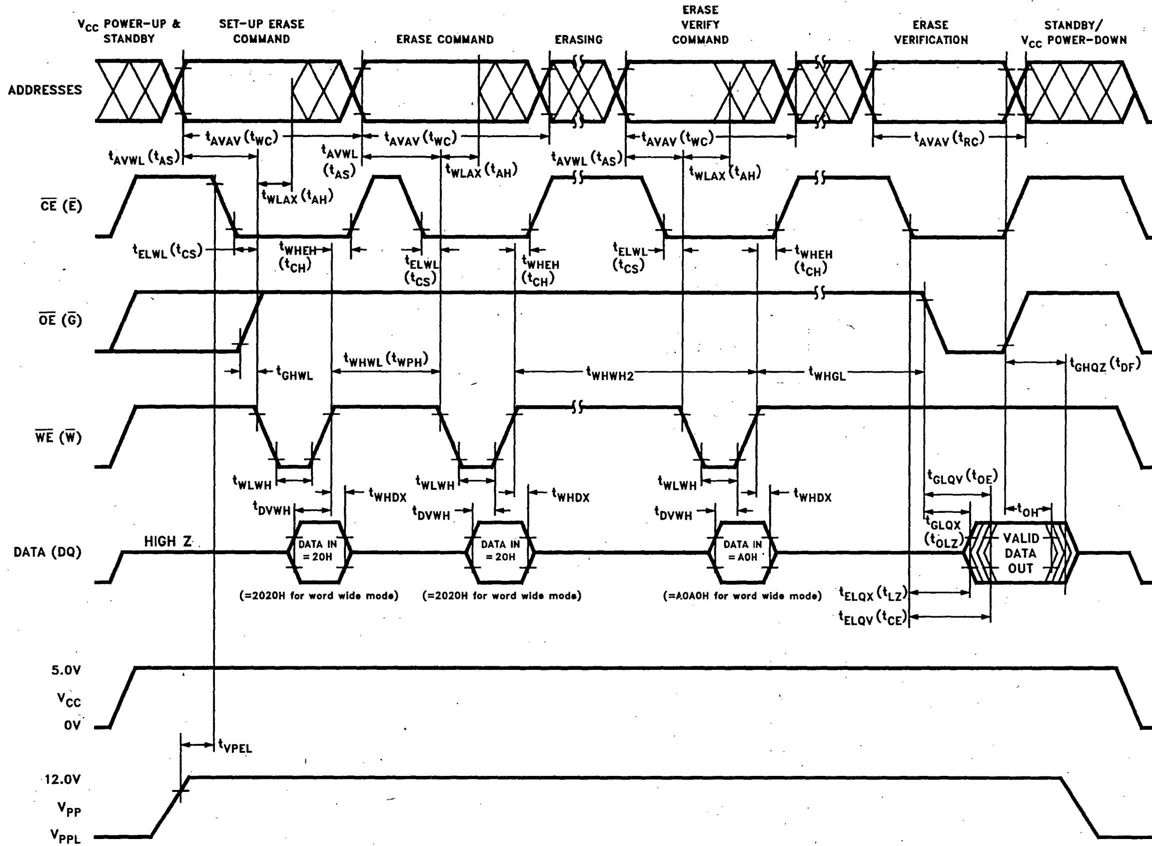
Parameter	Notes	Min	Typ	Max	Unit
Zone Erase Time	1, 3, 4		1.0	10	sec
Zone Write Time	1, 2, 4		2.0	12.5	sec
MTBF	5		10^6		Hrs

NOTES:
1. 25°C, 12.0V V_{PP}.
2. Minimum byte writing time excluding system overhead is 16 µs (10 µs program + 6 µs write recovery), while maximum is 400 µs/byte (16 µs × 25 loops allowed by algorithm). Max chip write time is specified lower than the worst case allowed by the write algorithm since most bytes write significantly faster than the worst case byte.
3. Excludes 00H writing Prior to Erasure.
4. One zone equals 128 kBytes.
5. MTBF = Mean Time Between Failure, 50% failure point for disk drives.



NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$

Figure 11. AC Waveforms for Write Operations



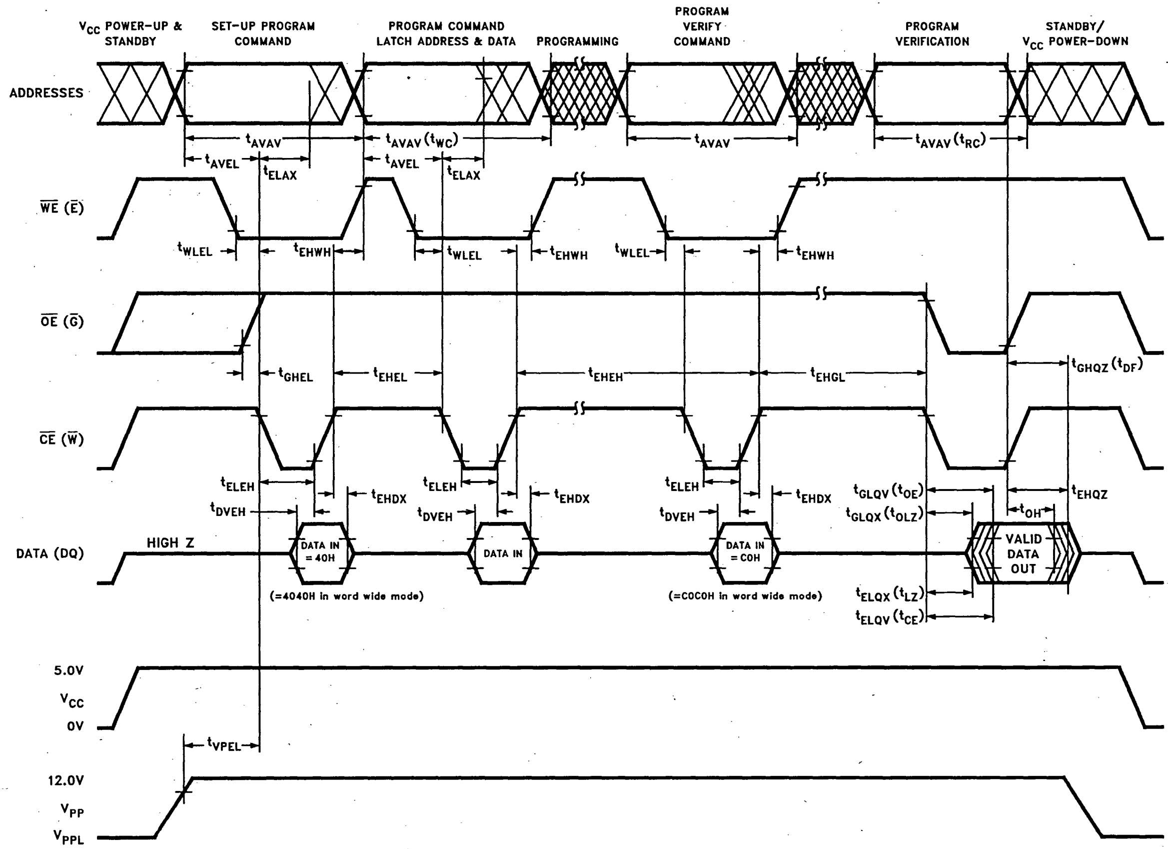
NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$

Figure 12. AC Waveforms for Erase Operations

ALTERNATIVE $\overline{CE}$-CONTROLLED WRITES

Symbol	Parameter	Notes	Min	Max	Unit
t_{AVAV}	Write Cycle Time		200		ns
t_{AVEL}	Address Set-Up Time		0		ns
t_{ELAX}	Address Hold Time		100		ns
t_{DVEH}	Data Set-Up Time		80		ns
t_{EHDX}	Data Hold Time		30		ns
t_{EHGL}	Write Recovery Time before Read		6		µs
t_{GHEL}	Read Recovery Time before Write		0		µs
t_{WLEL}	Write Enable Set-Up Time before Chip-Enable		0		ns
t_{EHWH}	Write Enable Hold Time		0		ns
t_{ELEH}	Write Pulse Width	1	100		ns
t_{EHEL}	Write Pulse Width High		20		ns
t_{PEL}	V_{PP} Set-Up Time to Chip-Enable Low		100		ns

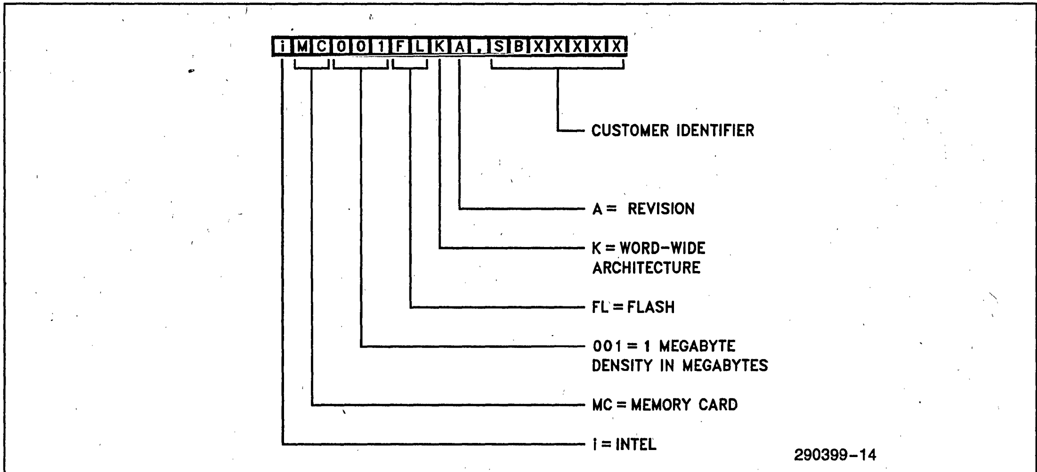
NOTES:
1. Chip Enable Controlled Writes: Write operations are driven by the valid combination of Chip Enable and Write Enable. In systems where Chip Enable defines the write pulse width (with a longer Write Enable timing waveforms) all set-up, hold and inactive Write Enable times should be measured relative to the Chip Enable waveform.



NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$

Figure 13. Alternative AC Waveforms for Write Operations

ORDERING INFORMATION



ADDITIONAL INFORMATION	Order Number
ER-20, "ETOX II Flash Memory Technology"	294005
RR-60, "ETOX II Flash Memory Reliability Data Summary"	293002
AP-343, "Solutions for High Density Applications using Flash Memory"	292079
RR70, "Flash Memory Card Reliability Data Summary"	293007

REVISION HISTORY

Number	Description
03	—Removed PRELIMINARY —Removed ExCA Compliance Section —Clarified need for Valid Address during commands —Corrected $V_{PP} = V_{PPH}$ in Erase Algorithm —Increased I_{CC2}–I_{CC5} D.C. current specs for both byte wide and word wide modes —Revised and updated Application Section discussion —Changed order number —Corrected Erase Algorithm Pulse count to 3000

iMC002FLKA
2-MBYTE FLASH MEMORY CARD

- **Inherent Nonvolatility (Zero Retention Power)**
  - **No Batteries Required for Back-up**
- **Over 1,000,000 Hours MTBF**
  - **More Reliable than Disk**
- **High-Performance Read**
  - **200 ns Maximum Access Time**
- **CMOS Low Power Consumption**
  - **25 mA Typical Active Current (X8)**
  - **400 μA Typical Standby Current**
- **Flash Electrical Zone-Erase**
  - **2 Seconds Typical per 256 kByte Zone**
  - **Multiple Zone-Erase**
- **Random Writes to Erased Zones**
  - **10 μs Typical Byte Write**
- **Write Protect Switch to Prevent Accidental Data Loss**
- **Command Register Architecture for Microprocesssor/Microcontroller Compatible Write Interface**
- **ETOX™ II Flash Memory Technology**
  - **5V Read, 12V Erase/Write**
  - **High-Volume Manufacturing Experience**
- **PCMCIA/JEIDA 68-Pin Standard**
  - **Byte- or Word-wide Selectable**
- **Independent Software & Hardware Vendor Support**
  - **Integrated System Solution Using Flash Filing Systems**

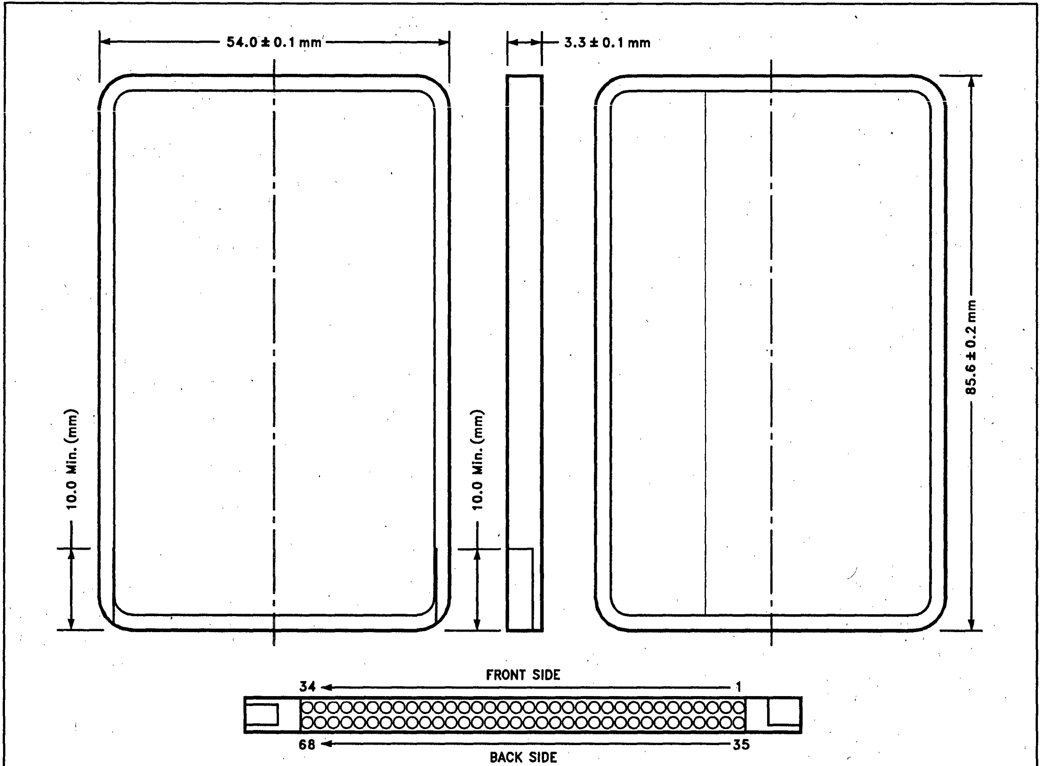
Intel's iMC002FLKA Flash Memory Card is the removable solution for storing and transporting important user data and application code. The combination of rewritability and nonvolatility make the Intel Flash Memory Card ideal for data acquisition and updatable firmware applications. Designing with Intel's Flash Memory Card enables OEM system manufacturers to produce portable and dedicated function systems that are higher performance, more rugged, and consume less power.

The iMC002FLKA conforms to the PCMCIA 1.0 international standard, providing standardization at the hardware and data interchange level. OEMs may opt to write the Card Information Structure (CIS) at the memory card's address 00000H with a format utility. This information provides data interchange functional capability. The 200 ns access time allows for "execute-in-place" capability, for popular low-power microprocessors. Intel's 2-MByte Flash Memory Card operates in a byte-wide and word-wide configuration providing performance/power options for different systems.

Intel's Flash Memory card employs Intel's ETOX™ II Flash Memories. Filing systems, such as Microsoft's* Flash File System (FFS), facilitate data file storage and card erasure using a purely nonvolatile medium in the DOS environment. Flash filing systems, coupled with the Intel Flash Memory Card, effectively create an all-silicon nonvolatile read/write random access memory system that is more reliable and higher performance than disk-based memory systems.

*Microsoft is a trademark of Microsoft Corp.
ExCA™ is a trademark of Intel Corporation.

October 1992
Order Number: 290412-002



1	GND	18	V_{PP1}	35	GND	52	V_{PP2}
2	D_3	19	A_{16}	36	$\overline{CD}_1$	53	NC
3	D_4	20	A_{15}	37	D_{11}	54	NC
4	D_5	21	A_{12}	38	D_{12}	55	NC
5	D_6	22	A_7	39	D_{13}	56	NC
6	D_7	23	A_6	40	D_{14}	57	NC
7	$\overline{CE}_1$	24	A_5	41	D_{15}	58	NC
8	A_{10}	25	A_4	42	$\overline{CE}_2$	59	NC
9	$\overline{OE}$	26	A_3	43	NC	60	NC
10	A_{11}	27	A_2	44	NC	61	$\overline{REG}$[1]
11	A_9	28	A_1	45	NC	62	$\overline{BVD}_2$[2]
12	A_8	29	A_0	46	A_{17}	63	$\overline{BVD}_1$[2]
13	A_{13}	30	D_0	47	A_{18}	64	D_8
14	A_{14}	31	D_1	48	A_{19}	65	D_9
15	$\overline{WE}$	32	D_2	49	A_{20}	66	D_{10}
16	NC	33	WP	50	NC	67	$\overline{CD}_2$
17	V_{CC}	34	GND	51	V_{CC}	68	GND

NOTES:

1. $\overline{REG}$ = register memory select = No Connect (NC), unused. When $\overline{REG}$ is brought low, PCMCIA/JEIDA standard card information structure data is expected. This is accomplished by formatting the card with this data.
2. $\overline{BVD}$ = battery detect voltage = Pulled High through Pull-Up Resistor.

Figure 1. iMC002FLKA Pin Configurations

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{20}	I	**ADDRESS INPUTS** for memory locations. Addresses are internally latched during a write cycle.
D_0–D_{15}	I/O	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the card is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}_1$, $\overline{CE}_2$	I	**CARD ENABLE:** Activates the card's high and low byte control logic, input buffers, zone decoders, and associated memory devices. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory card and reduces power consumption to standby levels.
$\overline{OE}$	I	**OUTPUT ENABLE:** Gates the cards output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	I	**WRITE ENABLE** controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **NOTE:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP1}, V_{PP2}		**ERASE/WRITE POWER SUPPLY** for writing the command register, erasing the entire array, or writing bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY** (5V ±5%).
GND		**GROUND**
$\overline{CD}_1$, $\overline{CD}_2$	O	**CARD DETECT.** The card is detected at $\overline{CD}_{1/2}$ = ground.
WP	O	**WRITE PROTECT.** All write operations are disabled with WP = active high.
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating.
$\overline{BVD}_1$, $\overline{BVD}_2$	O	**BATTERY VOLTAGE DETECT. NOT REQUIRED.**

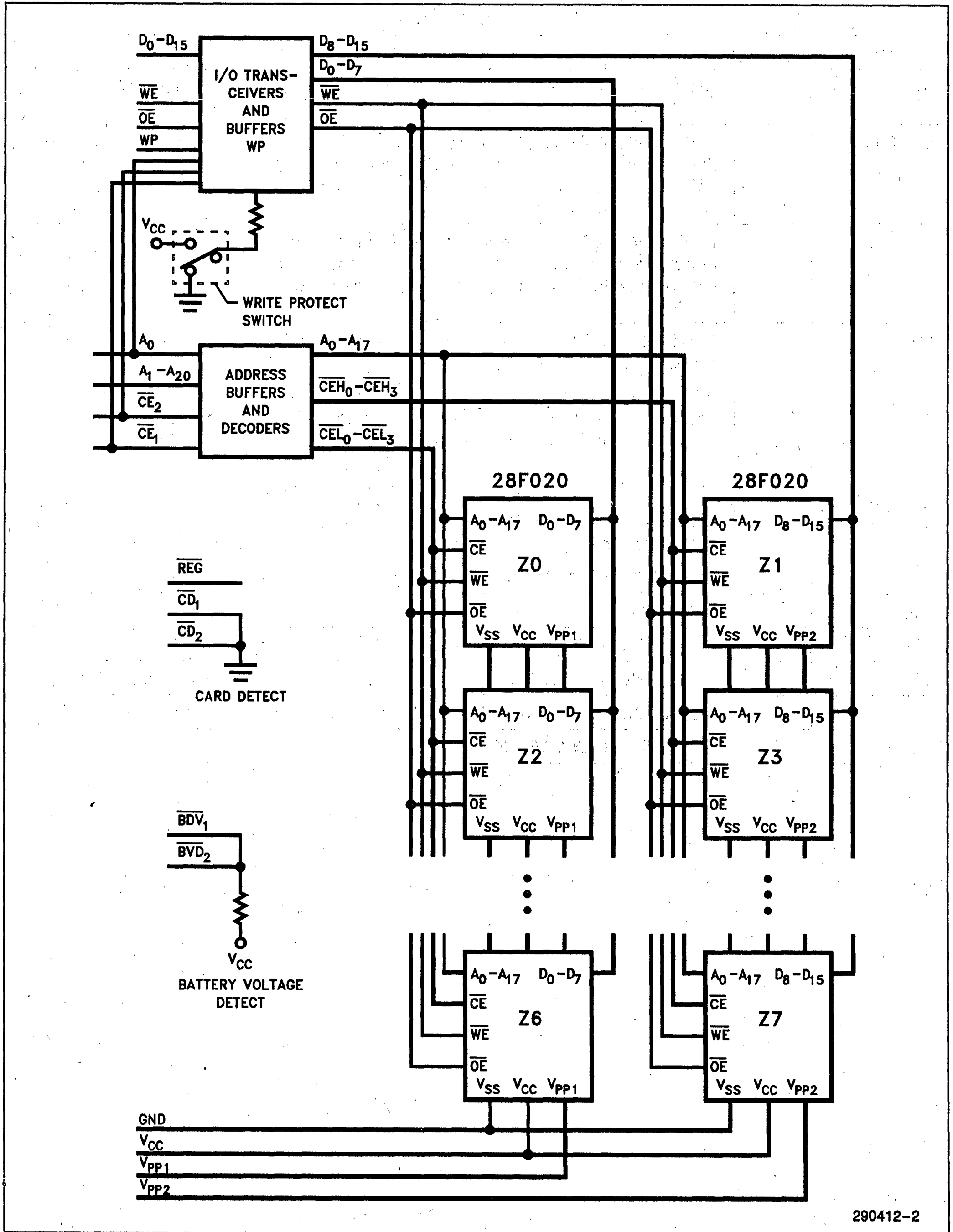


Figure 2. iMC002FLKA Block Diagram

APPLICATIONS

The iMC002FLKA Flash Memory Card allows for the storage of data and application programs on a purely solid-state removable medium. System resident flash filing systems, such as Microsoft's Flash File System, allow Intel's ETOX II highly reliable Flash Memory Card to effectively function as a physical disk drive.

User application software stored on the flash memory card substantially reduces the slow disk-DRAM download process. Replacing the disk results in a dramatic enhancement of read performance and substantial reduction of active power consumption, size, and weight—considerations particularly important in portables and dedicated systems. The iMC002FLKA's high performance read access time and command register microprocessor write interface allows for use of the flash memory system in an "execute-in-place" architecture. This configuration eliminates the need for the redundancy associated with DRAM and Disk memory system architectures. ROM based operating systems, such as Microsoft's MS-DOS ROM Version allow for "instant-on" capability. This enables the design of systems that boot, operate, store data files, and execute application code from/to purely nonvolatile memory.

The PCMCIA/JEIDA 68-pin interface with flash filing systems enables the end-user to transport data and application code between portables and host systems. Intel Flash PC cards provide durable nonvolatile memory storage protecting valuable user code and data.

For systems currently using a static RAM/battery configuration for data acquisition, the iMC002FLKA's inherent nonvolatility eliminates the need for battery backup. The concern of battery failure no longer exists, an important consideration for portable computers and medical instruments, both requiring continuous operation. The iMC002FLKA consumes no power when the system is off. In addition, the iMC002FLKA offers a considerable cost and density advantage over memory cards based on static RAM with battery backup.

The flash memory card's electrical zone-erasure, byte writability, and complete nonvolatility fit well with data accumulation and recording needs. Electrical zone-erasure gives the designer the flexibility to selectively rewrite zones of data while saving other zones for infrequently updated look-up tables, for example.

PRINCIPLES OF OPERATION

Intel's Flash Memory Card combines the functionality of two mainstream memory technologies: the rewritability of RAM and the nonvolatility of EPROM. The flash memory card consists of an array of individual memory devices, each of which defines a physical zone. The iMC002FLKA's memory devices erase as individual blocks, equivalent in size to the 256 kByte zone. Multiple zones can be erased simultaneously provided sufficient current for the appropriate number of zones (memory devices). Note, multiple zone erasure requires higher current from both the V_{PP} and V_{CC} power supplies. Erased zones can then be written in bit- or byte-at-a-time fashion and read randomly like RAM. Bit level write capability also supports disk emulation.

In the absence of high voltage on the $V_{PP1/2}$ pins, the iMC002FLKA remains in the read-only mode. Manipulation of the external memory card-control pin yields the standard read, standby, and output disable operations.

The same read, standby, and output disable operations are available when high voltage is applied to the $V_{PP1/2}$ pins. In addition, high voltage on $V_{PP1/2}$

enables erasure and rewriting of the accessed zone(s). All functions associated with altering zone contents—erase, erase verify, write, and write verify—are accessed via the command register.

Commands are written to the internal memory register(s), decoded by zone size, using standard microprocessor write timings. Register contents for a given zone serve as input to that zone's internal state-machine which controls the erase and rewrite circuitry. Write cycles also internally latch addresses and data needed for write and erase operations. With the appropriate command written to the register(s), standard microprocessor read timings output zone data, or output data for erase and write verification.

Byte-wide or Word-wide Selection

The flash memory card can be read, erased, and written in a byte-wide or word-wide mode. In the word-wide configuration V_{PP1} and/or $\overline{CE}_1$ control the LO-Byte while V_{PP2} and $\overline{CE}_2$ control the HI-Byte (A_0 = don't care).

Read, Write, and Verify operations are byte- or word-oriented, thus zone independent. Erase Setup and Begin Erase Commands are zone dependent such that commands written to any address within a 256 kByte zone boundary initiate the erase operation in that zone (or two 256 kByte zones under word-wide operation).

Conventional x8 operation uses $\overline{CE}_1$ active-low, with $\overline{CE}_2$ high, to read or write data through the D_0–D_7 only. "Even bytes" are accessed when A_0 is low, corresponding to the low byte of the complete x16 word. When A_0 is high, the "odd byte" is accessed by transposing the high byte of the complete x16 word onto the D_0–D_7 outputs. This odd byte corresponds to data presented on D_8–D_{15} pins in x16 mode.

Note that two zones logically adjacent in x16 mode are multiplexed through D_0–D_7 in x8 mode and are toggled by the A_0 address. Thus, zone specific erase operations must be kept discrete in x8 mode by addressing even bytes only for one-half of the zone pair, then addressing odd bytes only for the other half.

Card Detection

The flash memory card features two card detect pins ($\overline{CD}_{1/2}$) that allow the host system to determine if the card is properly loaded. Note that the two pins are located at opposite ends of the card. Each $\overline{CD}$ output should be read through a port bit. Should only one of the two bits show the card to be present, then the system should instruct the user to re-insert the card squarely into the socket. Card detection can also tell the system whether or not to redirect drives in the case of system booting. $\overline{CD}_{1/2}$ is active low, internally tied to ground.

Write Protection

The flash memory card features three types of write protection. The first type features a mechanical Write Protect Switch that disables the circuitry that control Write Enable to the flash devices. When the switch is activated, $\overline{WE}$ is forced high, which disables any writes to the Command Register. The second type of write protection is based on the PCMCIA/JEIDA socket. Unique pin length assignments provide protective power supply sequencing during hot insertion and removal. The third type operates via software control through the Command Register when the card resides in its connector. The Command Register of each zone is only active when $V_{PP1/2}$ is at high voltage. Depending upon the application, the system designer may choose to make $V_{PP1/2}$ power supply switchable—available only when writes are desired. When $V_{PP1/2} = V_{PPL}$, the contents of the register default to the read command, making the iMC002FLKA a read-only memory card. In this mode, the memory contents cannot be altered.

The system designer may choose to leave $V_{PP1/2} = V_{PPH}$, making the high voltage supply constantly available. In this case, all Command Register functions are inhibited whenever V_{CC} is below the write lockout voltage, V_{LKO}. (See the section on Power Up/Down Protection.) The iMC002FLKA is designed to accommodate either design practice, and to encourage optimization of the processor-memory card interface.

BUS OPERATIONS

Read

The iMC002FLKA has two control functions, both of which must be logically active, to obtain data at the outputs. Card Enable ($\overline{CE}$) is the power control and should be used for high and/or low zone(s) selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of accessed zone selection. In the byte-wide configuration, only one $\overline{CE}$ is required. The word-wide configuration requires both $\overline{CE}$s active low.

When $V_{PP1/2}$ is high (V_{PPH}), the read operations can be used to access zone data and to access data for write/erase verification. When $V_{PP1/2}$ is low (V_{PPL}), only read accesses to the zone data are allowed.

Output Disable

With Output Enable at a logic-high level (V_{IH}), output from the card is disabled. Output pins are placed in a high-impedance state.

Standby

With one Card Enable at a logic-high level, the standby operation disables one-half of the x16 output's read/write buffer. Further, only the zone corresponding to the selected address within the upper or lower $\overline{CE}_{1/2}$ bank is active at a time. (NOTE: A_0 must be low to select the low half of the x16 word when $\overline{CE}_2 = 1$ and $\overline{CE}_1 = 0$.) All other zones are deselected, substantially reducing card power consumption. For deselected banks, the outputs are placed in a high-impedance state, independent of the Output Enable signal. If the iMC002FLKA is deselected during erasure, writing, or write/erase verification, the accessed zone draws active current until the operation is terminated.

Intelligent Identifier Command

The manufacturer- and device-codes can be read via the Command Register, for instances where the iMC002FLKA is erased and rewritten in a universal reader/writer. Following a write of 90H to a zone's Command Register, a read from address location 00000H on any zone outputs the manufacturer code (89H). A read from address 0002H outputs the memory device code (BDH).

Write

Zone erasure and rewriting are accomplished via the Command Register, when high voltage is applied to $V_{PP1/2}$. The contents of the register serve as input to that zone's internal state-machine. The state-machine outputs dictate the function of the targeted zone.

The Command Register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

The Command Register is written by bringing Write Enable to a logic-low level (V_{IL}), while Card Enable(s) is/are low. Addresses are latched on the falling edge of Write Enable, while data is latched on the rising edge of the Write Enable pulse. Standard microprocessor write timings are used.

Refer to AC Write Charcteristics and the Erase/Write Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the V_{PP} pin(s), the contents of the zone Command Register(s) default to 00H, enabling read-only operations.

Placing high voltage on the V_{PP} pin(s) enable(s) read/write operations. Zone operations are selected by writing specific data patterns into the Command Register. Tables 3 and 4 define these iMC002FLKA register commands for both byte-wide and word-wide configurations.

All commands written to the Command Register require that the zone address be valid or the incorrect zone will receive the command. Any Command/Data Write or Data Read requires the correct valid address.

Table 2. Bus Operations

	Pins / Operation	Notes	[1, 7] V_{PP2}	[1, 7] V_{PP1}	A0	$\overline{CE_2}$	$\overline{CE_1}$	$\overline{OE}$	$\overline{WE}$	D_8-D_{15}	D_0-D_7
READ-ONLY	Read (x8)	8	V_{PPL}	V_{PPL}	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Even
	Read (x8)	9	V_{PPL}	V_{PPL}	V_{IH}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Odd
	Read (x8)	10	V_{PPL}	V_{PPL}	X	V_{IL}	V_{IH}	V_{IL}	V_{IH}	Data Out	Tri-state
	Read (x16)	11	V_{PPL}	V_{PPL}	X	V_{IL}	V_{IL}	V_{IL}	V_{IH}	Data Out	Data Out
	Output Disable		V_{PPL}	V_{PPL}	X	X	X	V_{IH}	V_{IH}	Tri-state	Tri-state
	Standby		V_{PPL}	V_{PPL}	X	V_{IH}	V_{IH}	X	X	Tri-state	Tri-state
READ/WRITE	Read (x8)	3, 8	V_{PPX}	V_{PPH}	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Even
	Read (x8)	3, 9	V_{PPH}	V_{PPX}	V_{IH}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Odd
	Read (x8)	10	V_{PPH}	V_{PPX}	X	V_{IL}	V_{IH}	V_{IL}	V_{IH}	Data Out	Tri-state
	Read (x16)	3, 11	V_{PPH}	V_{PPH}	X	V_{IL}	V_{IL}	V_{IL}	V_{IH}	Data Out	Data Out
	Write (x8)	5, 8	V_{PPX}	V_{PPH}	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{IL}	Tri-state	Data In-Even
	Write (x8)	9	V_{PPH}	V_{PPX}	V_{IH}	V_{IH}	V_{IL}	V_{IH}	V_{IL}	Tri-state	Data In-Odd
	Write (x8)	10	V_{PPH}	V_{PPX}	X	V_{IL}	V_{IH}	V_{IH}	V_{IL}	Data In	Tri-state
	Write (x16)	11	V_{PPH}	V_{PPH}	X	V_{IL}	V_{IL}	V_{IH}	V_{IL}	Data In	Data In
	Standby	4	V_{PPH}	V_{PPH}	X	V_{IH}	V_{IH}	X	X	Tri-state	Tri-state
	Output Disable		V_{PPH}	V_{PPH}	X	X	X	V_{IH}	V_{IL}	Tri-state	Tri-state

NOTES:
1. Refer to DC Characteristics. When $V_{PP1/2} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes may be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
3. Read operations with $V_{PP1/2} = V_{PPH}$ may access array data or the int_eligent Identifier codes.
4. With $V_{PP1/2}$ at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
5. Refer to Table 3 for valid Data-In during a write operation.
6. X can be V_{IL} or V_{IH}.
7. $V_{PPX} = V_{PPH}$ or V_{PPL}.
8. This x8 operation reads or writes the low byte of the x16 word on DQ_{0-7}, i.e., A_0 low reads "even" byte in x8 mode.
9. This x8 operation reads or writes the high byte of the x16 word on DQ_{0-7} (transposed from DQ_{8-15}), i.e., A_0 high reads "odd" byte in x8 mode.
10. This x8 operation reads or writes the high byte of the x16 on DQ_{8-15}. A_0 is "don't care."
11. A_0 is "don't care;" unused in x16 mode. High and low bytes are presented simultaneously.

Table 3. Command Definitions Byte-Wide Mode

Command	Notes	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
			Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	2	1	Write	RA	00H			
Read Intelligent ID Codes	4	3	Write	IA	90HT	Read		
Set-up Erase/Erase	5	2	Write	ZA	20H	Write	ZA	20H
Erase Verify	5	2	Write	EA	A0H	Read	EA	EVD
Set-up Write/Write	6	2	Write	WA	40H	Write	WA	WD
Write Verify	6	2	Write	WA	C0H	Read	WA	WVD
Reset	2, 7, 8	2	Write	ZA	FFH	Write	ZA	FFH

Table 4. Command Definitions Word-Wide Mode

Command	Notes	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
			Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	2	1	Write	RA	0000H			
Read Intelligent ID Codes	4	3	Write	IA	9090H	Read		
Set-up Erase/Erase	5	2	Write	ZA	2020H	Write	ZA	2020H
Erase Verify	5	2	Write	EA	A0A0H	Read	EA	EVD
Set-up Write/Write	6	2	Write	WA	4040H	Write	WA	WD
Write Verify	6	2	Write	WA	C0C0H	Read	WA	WVD
Reset	2, 7, 8	2	Write	ZA	FFFFH	Write	ZA	FFFFH

NOTES:

1. Bus operations are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   RA = Read Address
   WA = Address of memory location to be written.
   ZA = Address of 256 kByte zones involved in erase or Reset operations.
   Addresses are latched on the falling edge of the Write Enable pulse.
3. ID = Data read from location IA during device identification. (Mfr = 89H, Device = BDH).
   EVD = Data read from location EA during erase verify.
   WD = Data to be written at location WA. Data is latched on the rising edge of Write Enable.
   WVD = Data read from location WA during write verify. WA is latched on the Write command.
4. Following the Read Intelligent ID command, two read operations access manufacturer and device codes.
5. Figure 5 illustrates the Erase Algorithm.
6. Figure 6 illustrates the Write Algorithm.
7. The second bus cycle must be followed by the desired command register write.
8. The Reset command operates on a zone basis. To reset the entire card requires reset write cycles to each zone.

Read Command

While $V_{PP1/2}$ is high, for erasure and writing, zone memory contents can be accessed via the read command. The read operation is initiated by writing 00H (0000H for the word-wide configuration) into the zone Command Register(s). Microprocessor read cycles retrieve zone data. The accessed zone remains enabled for reads until the Command Register(s) contents are altered.

The default contents of each zone's register(s) upon $V_{PP1/2}$ power-up is 00H (00000H for word-wide). This default value ensures that no spurious alteration of memory card contents occurs during the $V_{PP1/2}$ power transition. Where the $V_{PP1/2}$ supply is left at V_{PPH}, the memory card powers-up and remains enabled for reads until the command Register contents of targeted zones are changed. Refer to the AC Read Characteristics and Waveforms for specific timing parameters.

Intelligent Identifier Command

Each zone of the iMC002FLKA contains an Intelligent Identifier to identify memory card device characteristics. The operation is initiated by writing 90H (9090H for word-wide) into the Command Register(s) with zone address. Following the command write, a read cycle from address 00000H retrieves the manufacturer code 89H (8989H for word-wide). A read cycle from address 0002H returns the device code BDH (BDBDH for word-wide). To terminate the operation, it is necessary to write another valid command into the register(s).

Set-up Erase/Erase Commands

Set-up Erase stages the targeted zone(s) for electrical erasure of all bytes in the zone. The set-up erase operation is performed by writing 20H to the Command Register (2020H for word-wide) with zone address.

To commence zone-erasure, the erase command (20H or 2020H) must again be written to the register(s) with zone address. The erase operation begins with the rising edge of the Write-Enable pulse and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command with zone address).

This two-step sequence of set-up followed by execution ensures that zone memory contents are not accidentally erased. Also, zone-erasure can only occur when high voltage is applied to the $V_{PP1/2}$ pins. In the absence of this high voltage, zone memory contents are protected against erasure. Refer to AC Erase Characterstics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all of the bytes of the zone in parallel. After each erase operation, all bytes in the zone must be individually verified. In byte-mode operations, zones are segregated by A_0 in odd and even banks; erase and erase verify operations must be done in complete passes of even-bytes-only then odd-bytes-only. See the Erase Algorithm for byte-wide mode. The erase verify operation is initiated by writing A0H (A0A0H for word-wide) into the Command Register(s). The address for the byte(s) to be verified must be supplied as it is latched on the falling edge of the Write Enable pulse. The register write terminates the erase operation with the rising edge of its Write Enable pulse.

The enabled zone applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased. Similarly, reading FFFFH from the addressed word indicates that all bits in the word are erased.

The erase-verify command must be written to the Command Register prior to each byte (word) verification to latch its address. The process continues for each byte (word) in the zone(s) until a byte (word) does not return FFH (FFFFH) data, or the last address is accessed.

In the case where the data read is not FFH (FFFFH), another erase operation is performed. (Refer to Set-up Erase/Erase.) Verification then resumes from the address of the last-verified byte (word). Once all bytes (words) in the zone(s) have been verified, the erase step is complete. The accessed zone can now be written. At this point, the verify operation is terminated by writing a valid command (e.g., Write Set-up) to the Command Register. The Erase algorithms for byte-wide and word-wide configurations illustrate how commands and bus operations are combined to perform electrical erasure of the iMCC002FLKA. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Set-up Write/Write Commands

Set-up write is a command-only operation that stages the targeted zone for byte writing. Writing 40H (4040H) into the Command Register(s) performs the set-up operation.

Once the write set-up operation is performed, the next Write Enable pulse causes a transition to an active write operation. Addresses are internally latched on the falling edge of the Write Enable pulse. Data is internally latched on the rising edge of the Write Enable pulse. The rising edge of Write Enable also begins the write operation. The write operation terminates with the next rising edge of Write Enable, which is used to write the verify command. Refer to AC Write Characteristics and Waveforms for specific timing parameters.

Write Verify Command

The iMC002FLKA is written on a byte-by-byte or word-by-word basis. Byte or word writing may occur sequentially or at random. Following each write operation, the byte or word just written must be verified.

The write-verify operation is initiated by writing C0H (C0C0H) into the Command Register(s) with correct address. The register write(s) terminate(s) the write operation with the rising edge of its Write Enable pulse. The write-verify operation stages the accessed zone(s) for verification of the byte or word last written. The zone(s) apply(ies) an internally-generated margin voltage to the byte or word. A microprocessor read cycle outputs the data. A successful comparison between the written byte or word and true data means that the byte or word is successfully written. The write operation then proceeds to the next desired byte or word location. The Write algorithms for byte-wide and word-wide configurations illustrate how commands are combined with bus operations to perform byte and word writes. Refer to AC Write Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase-, or write-command sequences. Following either set-up command (erase or write) with two consecutive writes of FFH (FFFFH for word-wide) will safely abort the operation. Zone memory contents will not be altered. A valid command must then be written to place the accessed zone in the desired state.

EXTENDED ERASE/WRITE CYCLING

Intel has designed extended cycling capability into its ETOX II flash memory technology enabling a flash memory card with a MTBF that is approximately 20 times more reliable than rotating disk technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field minimizes the probability of oxide defects in the region. The lower electric field greatly reduces oxide stress and the probability of failure.

WRITE ALGORITHMS

The write algorithm(s) use write operations of 10 µs duration. Each operation is followed by a byte or word verification to determine when the addressed byte or word has been successfully written. The algorithm(s) allows for up to 25 write operations per byte or word, although most bytes and words verify on the first or second operation. The entire sequence of writing and byte/word verification is performed with V_{PP} at high voltage.

ERASE ALGORITHM

The Erase algorithm(s) yield(s) fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the write algorithm, to simultaneously remove charge from all bits in the accessed zone(s).

Erasure begins with a read of memory zone contents. Reading FFH (FFFFH) data from the accessed zone(s) can be immediately followed by writing to the desired zone(s).

For zones being erased and rewritten, uniform and reliable erasure is ensured by first writing all bits in the accessed zone to their charged state (data = 00H byte-wide, 00000H word-wide). This is accomplished, using the write algorithm, in approximately four seconds per zone.

Erase execution then continues with an initial erase operation. Erase verification (data = FFH byte-wide, FFFFH word-wide) begins at address 00000H and continues through the zone to the last address, or until data other than FFH (FFFFH) is encountered. (Note: byte-wide erase operation requires separate even- and odd-address passes to handle the individual 256 kByte zones.) With each erase operation, an increasing number of bytes or words verify to the erased state. Erase efficiency may be improved by storing the address of the last byte or word verified in a register(s). Following the next erase operation, verification starts at the stored address location. Follow this procedure until all bytes in the zone are erased. Then, re-start the procedure for the next zone or word-wide zone pair. Erasure typically occurs in two seconds per zone.

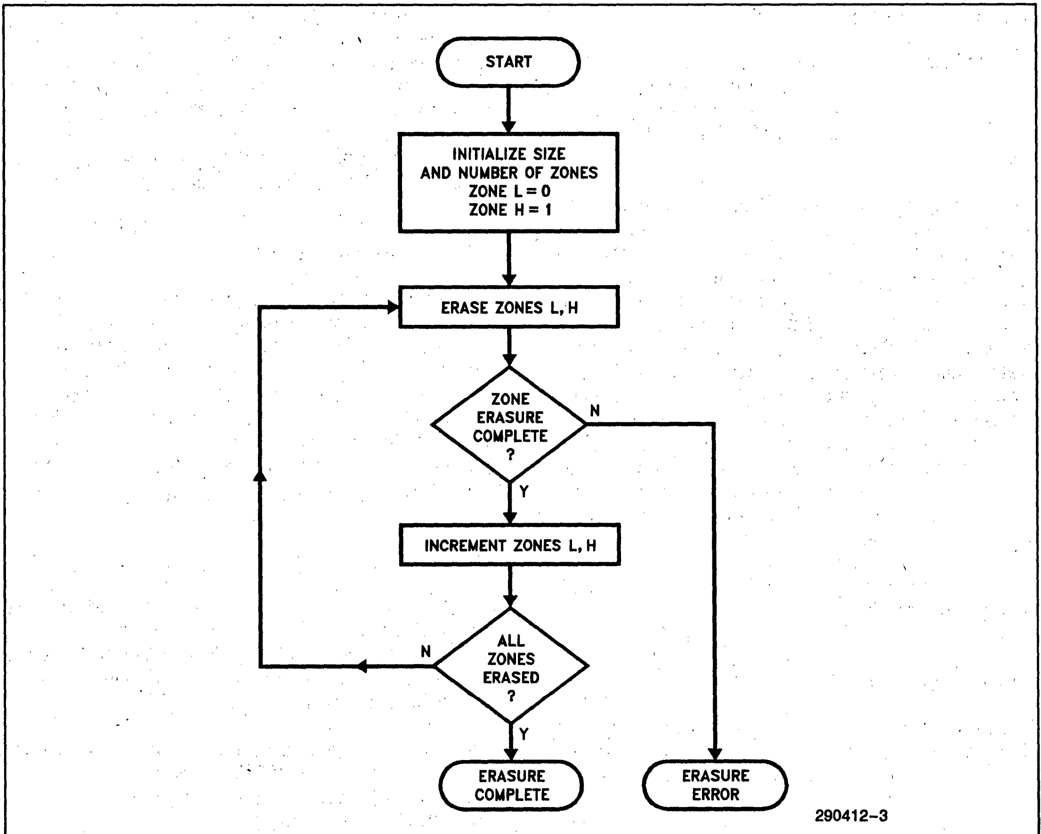
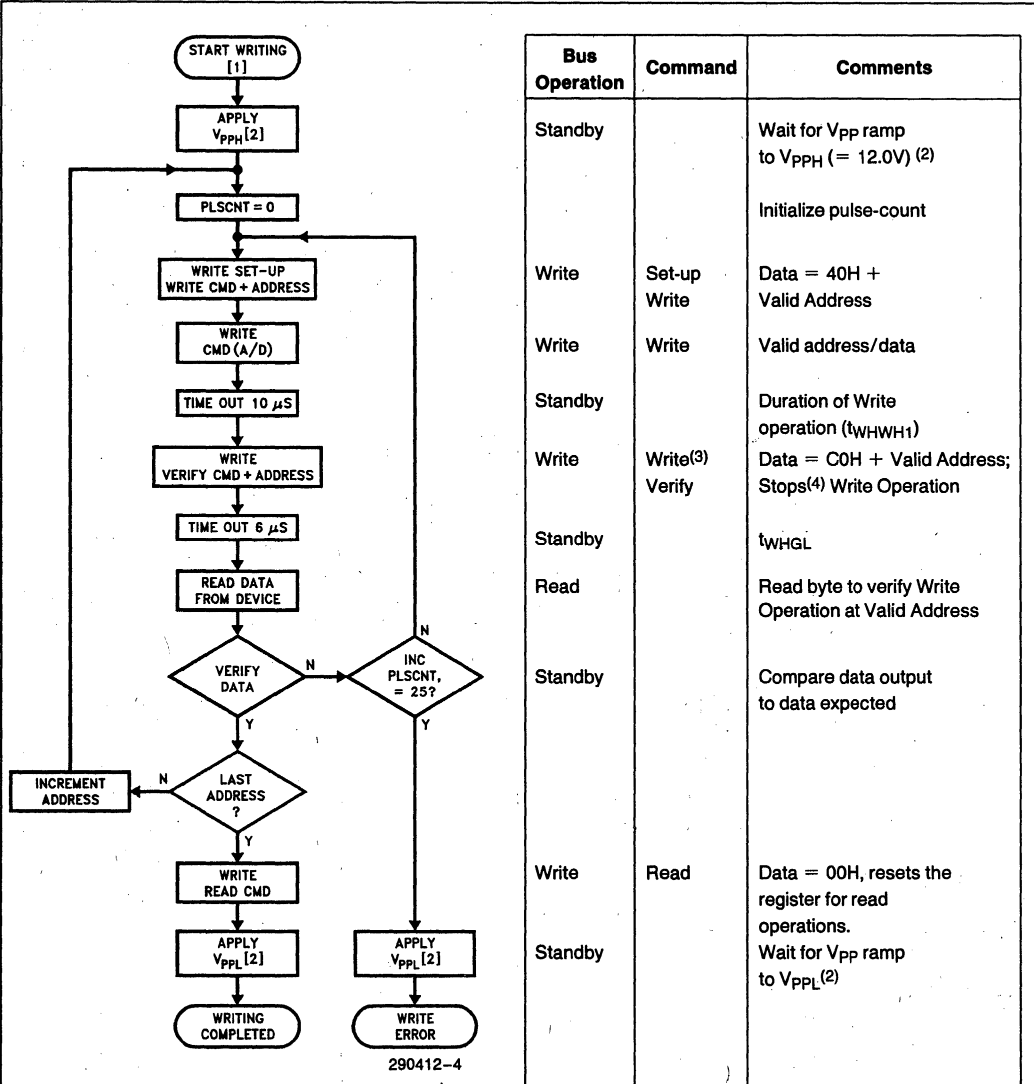


Figure 3. Full Card Erase Flow

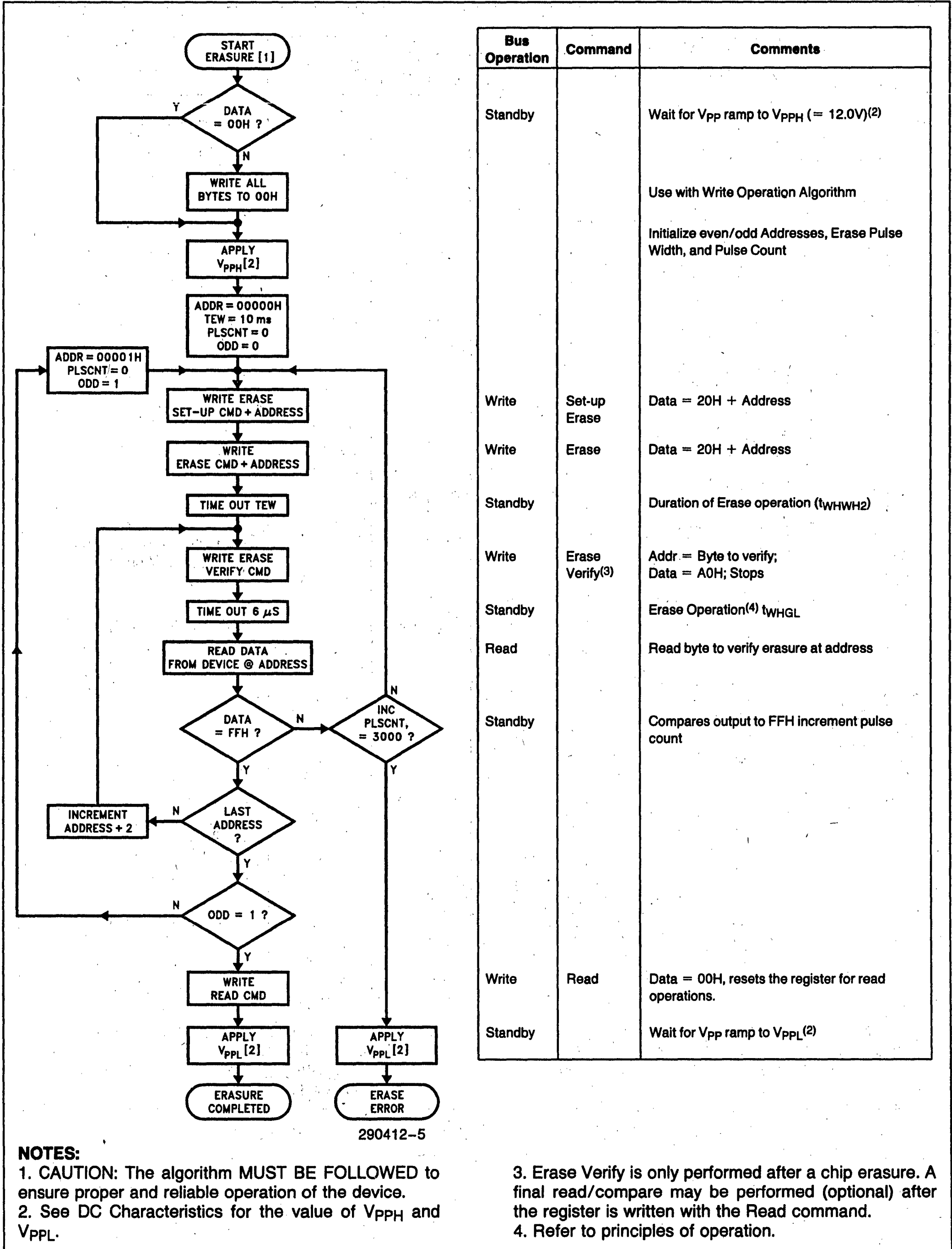


Bus Operation	Command	Comments
Standby		Wait for V_{PP} ramp to V_{PPH} (= 12.0V) [2]
		Initialize pulse-count
Write	Set-up Write	Data = 40H + Valid Address
Write	Write	Valid address/data
Standby		Duration of Write operation (t_{WHWH1})
Write	Write[3] Verify	Data = C0H + Valid Address; Stops[4] Write Operation
Standby		t_{WHGL}
Read		Read byte to verify Write Operation at Valid Address
Standby		Compare data output to data expected
Write	Read	Data = 00H, resets the register for read operations.
Standby		Wait for V_{PP} ramp to V_{PPL} [2]

NOTES:

1. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.
2. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
3. Write Verify is only performed after a byte write operation. A final read/compare may be performed (optional) after the register is written with the Read command.
4. Refer to principles of operation.

Figure 4. Write Algorithm for Byte-Wide Mode



Bus Operation	Command	Comments
Standby		Wait for V_{PP} ramp to V_{PPH} (= 12.0V)(2)
		Use with Write Operation Algorithm
		Initialize even/odd Addresses, Erase Pulse Width, and Pulse Count
Write	Set-up Erase	Data = 20H + Address
Write	Erase	Data = 20H + Address
Standby		Duration of Erase operation (t_{WHWH2})
Write	Erase Verify(3)	Addr = Byte to verify; Data = A0H; Stops
Standby		Erase Operation(4) t_{WHGL}
Read		Read byte to verify erasure at address
Standby		Compares output to FFH increment pulse count
Write	Read	Data = 00H, resets the register for read operations.
Standby		Wait for V_{PP} ramp to V_{PPL}(2)

NOTES:
1. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.
2. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
3. Erase Verify is only performed after a chip erasure. A final read/compare may be performed (optional) after the register is written with the Read command.
4. Refer to principles of operation.

Figure 5. Erase Algorithm for Byte-Wide Mode

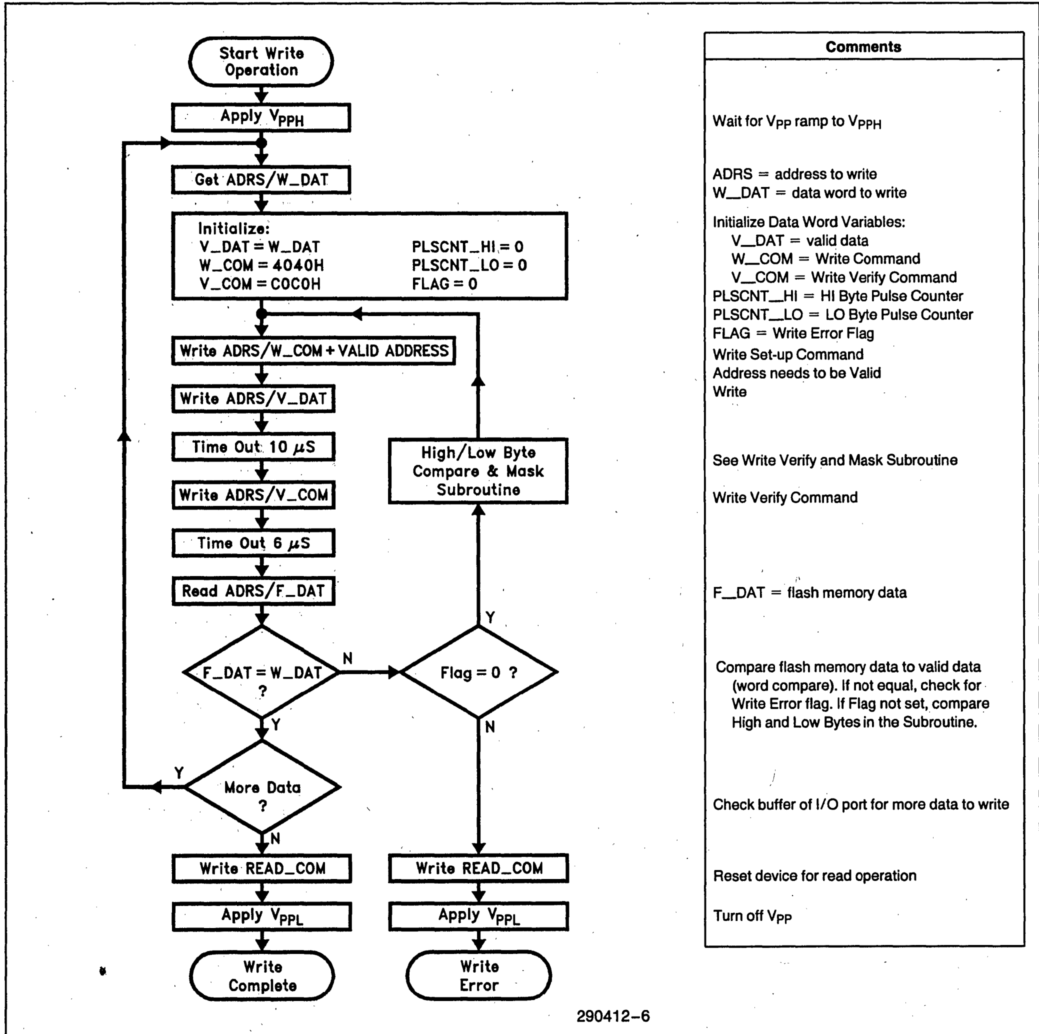
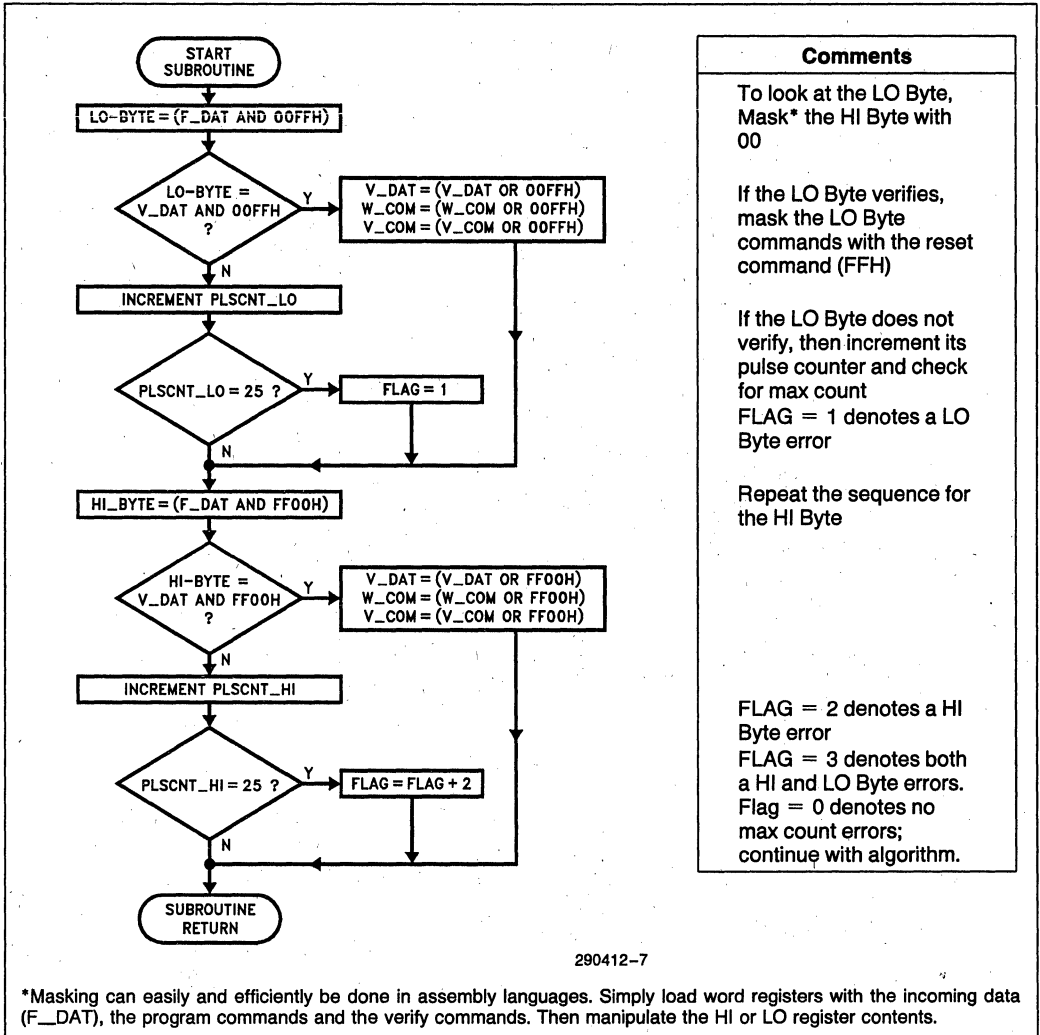
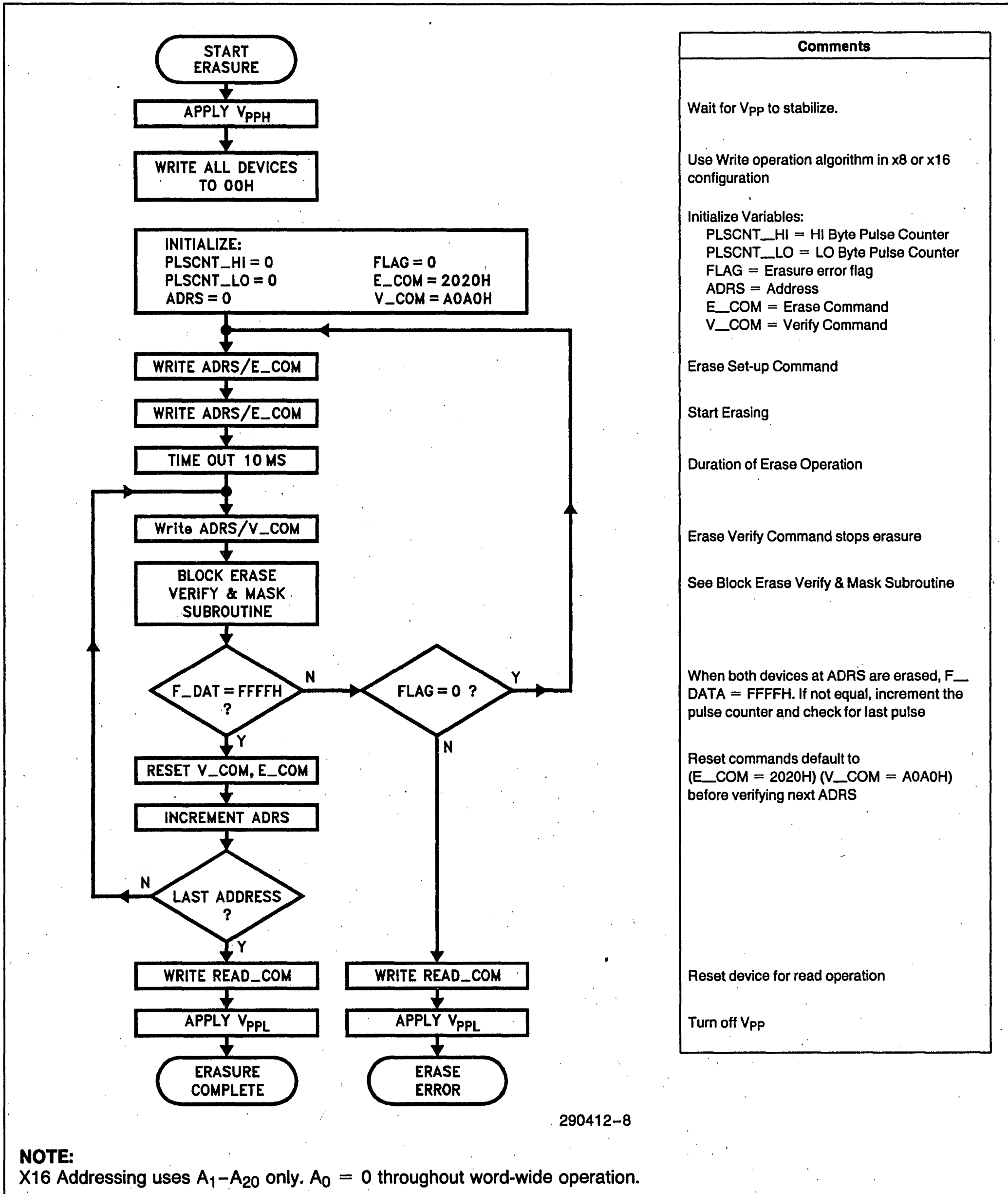


Figure 6. Write Algorithm for Word-Wide Mode



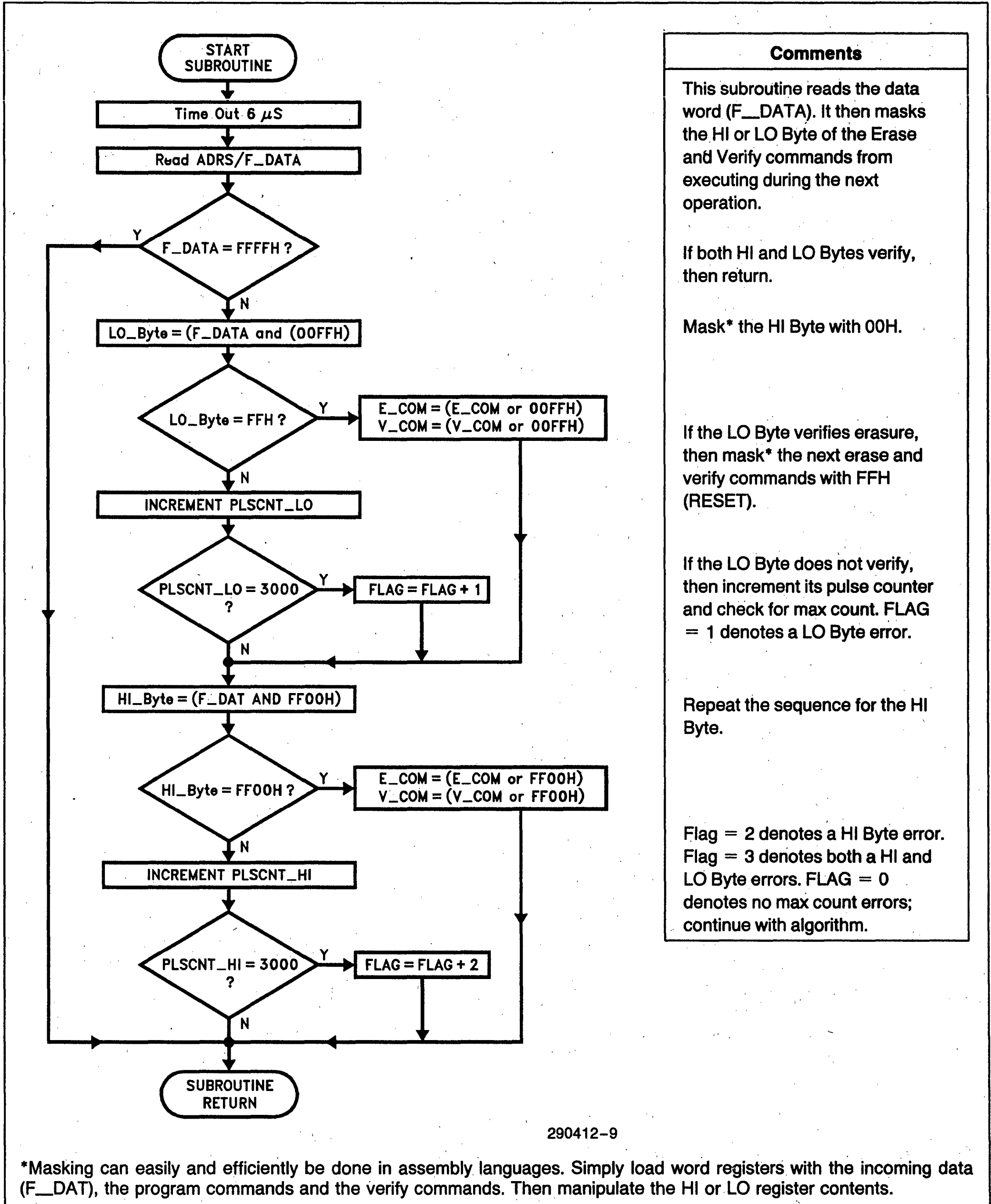
*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming data (F_DAT), the program commands and the verify commands. Then manipulate the HI or LO register contents.

Figure 7. Write Verify and Mask Subroutine for Word-Wide Mode



NOTE:
X16 Addressing uses A_1–A_{20} only. A_0 = 0 throughout word-wide operation.

Figure 8. Erase Algorithm for Word-Wide Mode



*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming data (F_DAT), the program commands and the verify commands. Then manipulate the HI or LO register contents.

Figure 9. Erase Verify and Mask Subroutine for Word-Wide Mode

SYSTEM DESIGN CONSIDERATIONS

Three-Line Control

Three-line control provides for:

a. the lowest possible power dissipation and.

b. complete assurance that output bus contention will not occur.

To efficiently use these three control inputs, an address-decoder output should drive $\overline{CE}_{1,2}$, while the system's Read signal controls the card $\overline{OE}$ signal, and other parallel zones. This, coupled with the internal zone decoder, assures that only enabled memory zones have active outputs, while deselected zones maintain the low power standby condition.

Power-Supply Decoupling

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current issues—standby, active and transient current peaks, produced by falling and rising edges of $\overline{CE}_{1/2}$. The capacitive and inductive loads on the card and internal flash memory zones determine the magnitudes of these peaks.

Three-line control and proper decoupling capacitor selection will suppress transient voltage peaks. The iMC002FLKA features on-card ceramic decoupling capacitors connected between V_{CC} and V_{SS}, and between V_{PP1}/V_{PP2} and V_{SS}.

The card connector should also have a 4.7 µF electrolytic capacitor between V_{CC} and V_{SS}, as well as between V_{PP1}/V_{PP2} and V_{SS}. The bulk capacitors will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

Power Up/Down Protection

The PCMCIA/JEIDA socket is specified, via unique Pin lengths, to properly sequence the power supplies to the flash memory card. This assures that hot insertion and removal will not result in card damage or data loss.

Each zone in the iMC002FLKA is designed to offer protection against accidental erasure or writing, caused by spurious system-level signals that may exist during power transitions. The card will power-up into the read state.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}_{1,2}$ must be low for a command write, driving either to V_{IH} will inhibit writes. With its control register architecture, alteration of zone contents only occurs after successful completion of the two-step command sequences.

While these precautions are sufficient for most applications, it is recommended that V_{CC} reach its steady state value before raising $V_{PP1/2}$ above V_{CC} + 2.0V. In addition, upon powering-down, $V_{PP1/2}$ should be below V_{CC} + 2.0V, before lowering V_{CC}.

Absolute Maximum Ratings*

Operating Temperature
During Read ....0°C to + 60°C(1)
During Erase/Write ....0°C to + 60°C

Temperature Under Bias ....−10°C to +70°C

Storage Temperature............−30°C to +70°C

Voltage on Any Pin with
Respect to Ground ....−2.0V to +7.0V(2)

V_{PP1}/V_{PP2} Supply Voltage with
Respect to Ground
During Erase/Write.......−2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage with
Respect to Ground ....−2.0V to +7.0V(2)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:

1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
3. Maximum DC input voltage on V_{PP1}/V_{PP2} may overshoot to +14.0V for periods less than 20 ns.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature	0	60	°C	For Read-Only and Read/Write Operations
V_{CC}	V_{CC} Supply Voltage	4.75	5.25	V	
V_{PPH}	Active V_{PP1}, V_{PP2} Supply Voltages	11.40	12.60	V	
V_{PPL}	V_{PP} During Read Only Operations	0.00	6.50	V	

DC CHARACTERISTICS—Byte Wide Mode

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{LI}	Input Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.4	0.8	mA	$V_{CC} = V_{CC}$ max $\overline{CE1} = \overline{CE2} = V_{CC} \pm 0.2V$
				4	7	mA	$V_{CC} = V_{CC}$ max $\overline{CE1} = \overline{CE2} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1, 2		25	50	mA	$V_{CC} = V_{CC}$ max $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Write Current	1, 2		5.0	15.0	mA	Writing in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		10.0	20.0	mA	Erasure in Progress
I_{CC4}	V_{CC} Write Verify Current	1, 2		10.0	20.0	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress

DC CHARACTERISTICS—Byte Wide Mode (Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{CC5}	V_{CC} Erase Verify Current	1, 2		10.0	20.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±80	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	1, 3		0.4	0.8	mA	$V_{PP} > V_{CC}$
					±0.08		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Write Current	1, 3		8.0	30	mA	$V_{PP} = V_{PPH}$ Write in Progress
I_{PP3}	V_{PP} Erase Current	1, 3		10	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Write Verify Current	1, 3		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 3		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.4		$V_{CC} \pm 0.3$	V	
V_{OL}	Output Low Voltage				0.40	V	$I_{OL} = 3.2$ mA $V_{CC} = V_{CC}$ min
V_{OH1}	Output High Voltage		3.8			V	$I_{OH} = -2.0$ mA $V_{CC} = V_{CC}$ min
V_{PPL}	V_{PP} During Read-Only Operations		0.00		6.5	V	Note: Erase/Write are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} During Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at $V_{CC} = 5.0V$, $V_{PP} = 12.0V$, $T = 25°C$.
2. 1 chip active and 7 in standby for byte-wide mode.
3. Assumes 1 V_{PP} is active.

DC CHARACTERISTICS—Word Wide Mode

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{LI}	Input Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.4	0.8	mA	$V_{CC} = V_{CC}$ max $\overline{CE1} = \overline{CE2} = V_{CC} \pm 0.2V$
				4	7	mA	$V_{CC} = V_{CC}$ max $\overline{CE1} = \overline{CE2} = V_{IH}$

DC CHARACTERISTICS—Word Wide Mode (Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{CC1}	V_{CC} Active Read Current	1, 2		40	80	mA	$V_{CC} = V_{CC}$ max $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Write Current	1, 2		7.0	25	mA	Writing in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		15	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Write Verify Current	1, 2		15	30	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		10	30	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±80	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current	1, 3		0.7	1.6	mA	$V_{PP} > V_{CC}$
	or Standby Current				±0.16		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Write Current	1, 3		16	60	mA	$V_{PP} = V_{PPH}$ Write in Progress
I_{PP3}	V_{PP} Erase Current	1, 3		20	60	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Write Verify Current	1, 3		5.0	12	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 3		5.0	12	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.4		V_{CC} ± 0.3	V	
V_{OL}	Output Low Voltage				0.40	V	I_{OL} = 3.2 mA $V_{CC} = V_{CC}$ min
V_{OH1}	Output High Voltage		3.8			V	I_{OH} = −2.0 mA $V_{CC} = V_{CC}$ min
V_{PPL}	V_{PP} During Read-Only Operations		0.00		6.5	V	Note: Erase/Write are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} During Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C.
2. 2 chips active and 6 in standby for word-wide mode.
3. Assumes 2 V_{PPS} are active.

CAPACITANCE $T = 25°C, f = 1.0\ MHz$

Symbol	Parameter	Notes	Limits		Unit	Conditions
			Min	Max		
C_{IN1}	Address Capacitance			40	pF	$V_{IN} = 0V$
C_{IN2}	Control Capacitance			40	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance			40	pF	$V_{OUT} = 0V$
$C_{I/O}$	I/O Capacitance			40	pF	$V_{I/O} = 0V$

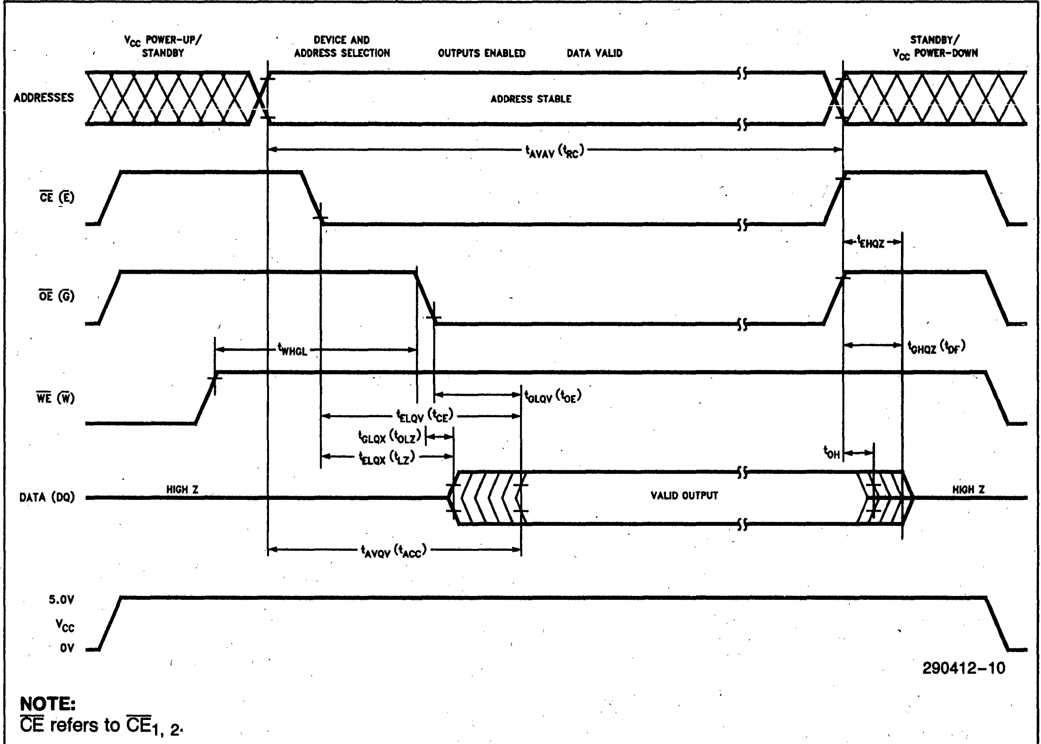
AC TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) . . . . . . 10 ns
Input Pulse Levels . . . . V_{OL} and V_{OH1}
Input Timing Reference Level . . . . V_{IL} and V_{IH}
Output Timing Reference Level . . . . V_{IL} and V_{IH}

AC CHARACTERISTICS—Read-Only Operations

Symbol	Characteristic	Notes	Min	Max	Unit
t_{AVAV}/t_{RC}	Read Cycle Time	2	200		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time	2		200	ns
t_{AVQV}/t_{ACC}	Address Access Time	2		200	ns
t_{GLQV}/t_{OE}	Output Enable Access Time	2		100	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	2	5		ns
t_{EHQZ}	Chip Disable to Output in High Z	2		60	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	2	5		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	2		60	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	1, 2	5		ns
t_{WHGL}	Write Recovery Time before Read	2	6		μs

NOTES:
1. Whichever occurs first.
2. Rise/Fall Time ≤ 10 ns.



NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$.

Figure 10. AC Waveforms for Read Operations

AC CHARACTERISTICS—For Write/Erase Operations

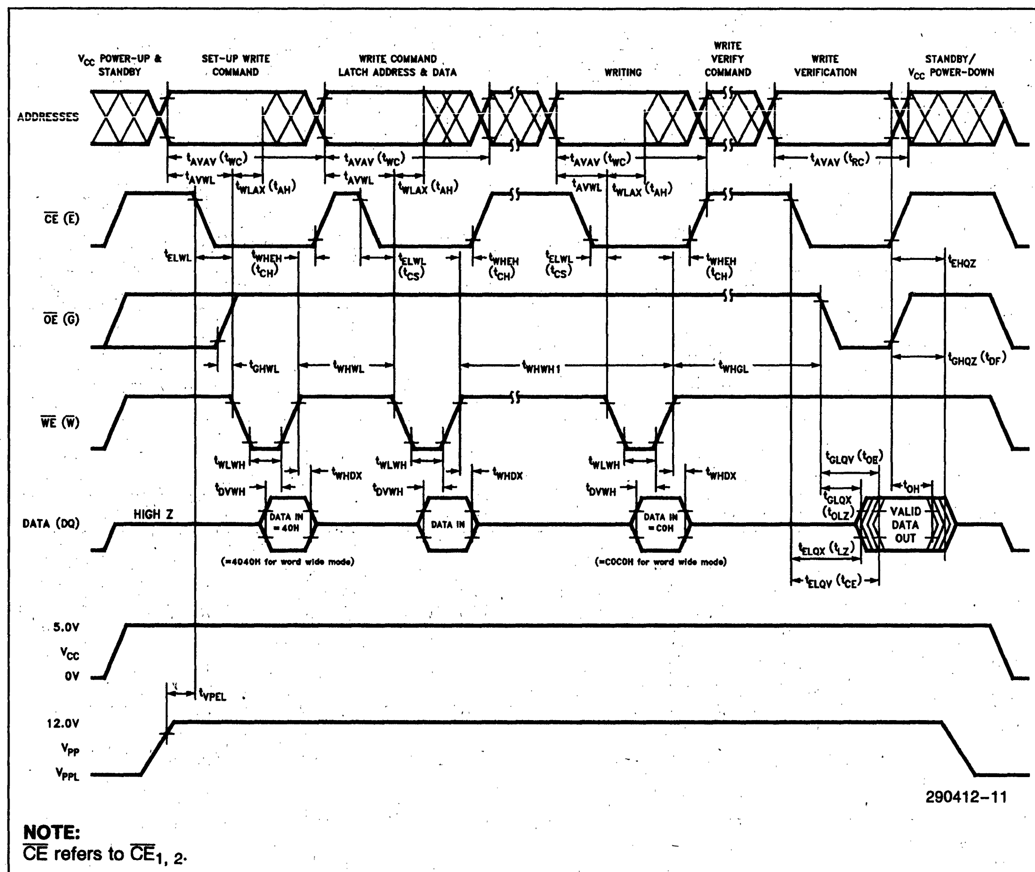
Symbol	Characteristic	Notes	Min	Max	Unit
t_{AVAV}/t_{WC}	Write Cycle Time	1, 2	200		ns
t_{AVWL}/t_{AS}	Address Set-up Time	1, 2	0		ns
t_{WLAX}/t_{AH}	Address Hold Time	1, 2	100		ns
t_{DVWH}/t_{DS}	Data Set-up Time	1, 2	80		ns
t_{WHDX}/t_{DH}	Data Hold Time	1, 2	30		ns
t_{WHGL}	Write Recovery Time before Read	1, 2	6		μs
t_{GHWL}	Read Recovery Time before Write	1, 2	0		μs
t_{WLOZ}	Output High-Z from Write Enable	1, 2	5		ns
t_{WHOZ}	Output Low-Z from Write Enable	1, 2		60	ns
t_{ELWL}/t_{CS}	Chip Enable Set-up Time before Write	1, 2	40		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time	1, 2	0		ns
t_{WLWH}/t_{WP}	Write Pulse Width	1, 2	100		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High	1, 2	20		ns
t_{WHWH1}	Duration of Write Operation	1, 2, 3	10		μs
t_{WHWH2}	Duration of Erase Operation	1, 2, 3	9.5		ms
t_{VPEL}	V_{PP} Set-up Time to Chip Enable Low	1, 2	100		ns

NOTES:
1. Read timing parameters during read/write operations are the same as during read-only operations. Refer to A.C. Characteristics for Read-Only Operations.
2. Rise/Fall time ≤ 10 ns.
3. The integrated stop timer terminates the write/erase operations, thereby eliminating the need for a maximum specifiction.

ERASE/WRITE PERFORMANCE

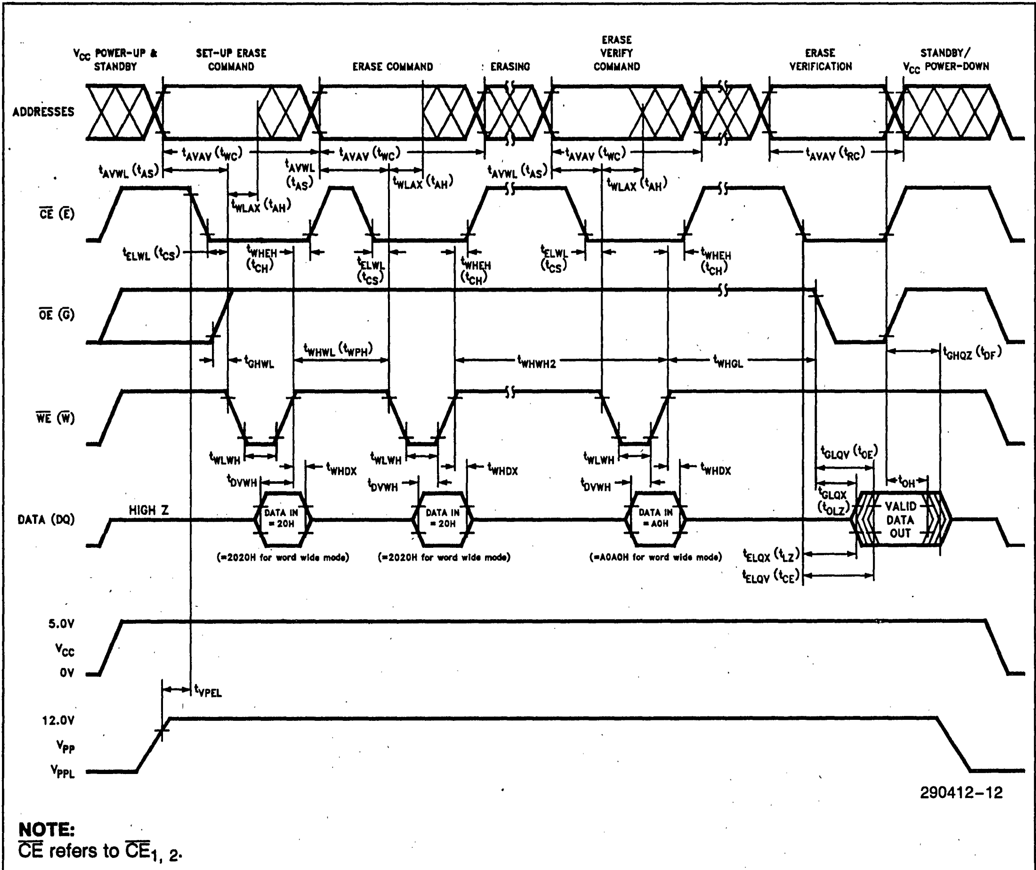
Parameter	Notes	Min	Typ	Max	Unit
Zone Erase Time	1, 3, 4		2.0	30	sec
Zone Write Time	1, 2, 4		4.0	25	sec
MTBF	5		10^6		Hrs

NOTES:
1. 25°C, 12.0V V_{PP}.
2. Minimum byte writing time excluding system overhead is 16 μs (10 μs program + 6 μs write recovery), while maximum is 400 μs/byte (16 μs x 25 loops allowed by algorithm). Max chip write time is specified lower than the worst case allowed by the write algorithm since most bytes write significantly faster than the worst case byte.
3. Excludes 00H writing Prior to Erasure.
4. One zone equals 256 kBytes.
5. MTBF — Mean Time between Failure, 50% failure point for disk drives.



NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$.

Figure 11. AC Waveforms for Write Operations



NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$.

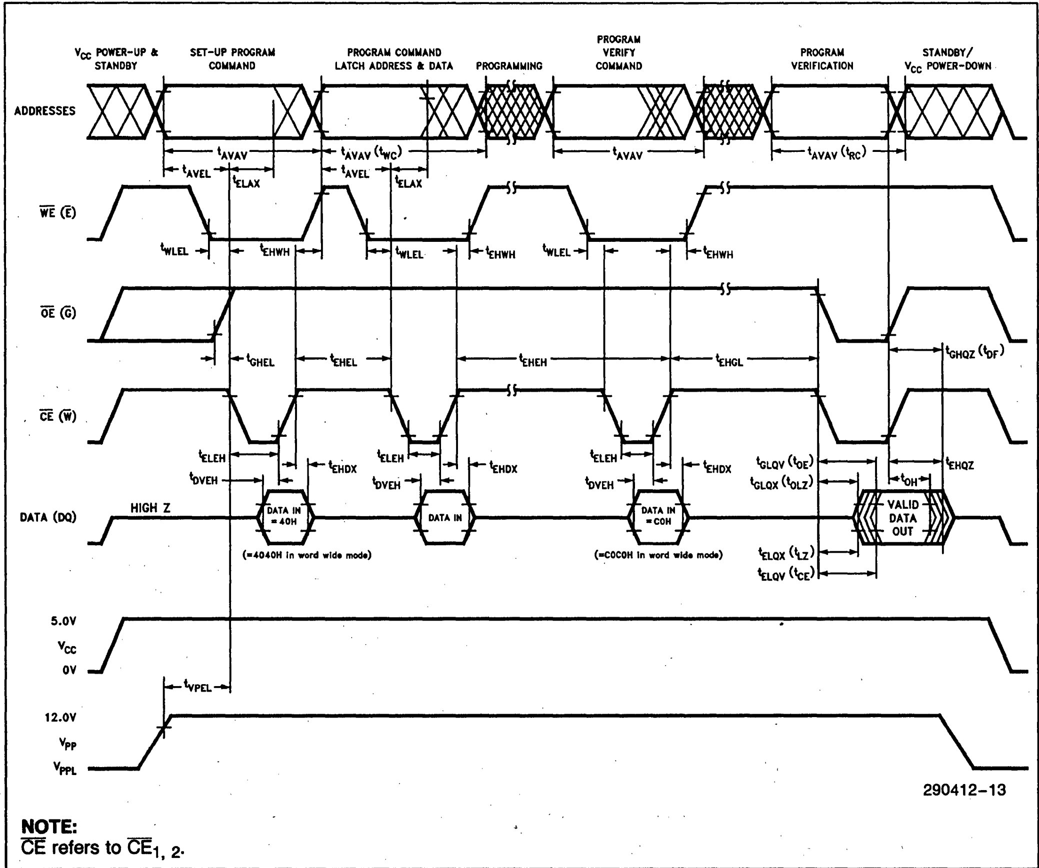
Figure 12. AC Waveforms for Erase Operations

ALTERNATIVE $\overline{CE}$-CONTROLLED WRITES

Symbol	Characteristic	Notes	Min	Max	Unit
t_{AVAV}	Write Cycle Time		200		ns
t_{AVEL}	Address Set-up Time		0		ns
t_{ELAX}	Address Hold Time		100		ns
t_{DVEH}	Data Set-up Time		80		ns
t_{EHDX}	Data Hold Time		30		ns
t_{EHGL}	Write Recovery Time before Read		6		µs
t_{GHEL}	Read Recovery Time before Write		0		µs
t_{WLEL}	Write Enable Set-Up Time before Chip-Enable		0		ns
t_{EHWH}	Write Enable Hold Time		0		ns
t_{ELEH}	Write Pulse Width	1	100		ns
t_{EHEL}	Write Pulse Width High		20		ns
t_{PEL}	V_{PP} Set-up Time to Chip Enable Low		100		ns

NOTES:

1. Chip Enable Controlled Writes: Write operations are driven by the valid combination of Chip Enable and Write Enable. In systems where Chip Enable defines the write pulse width (with a longer Write Enable timing waveform) all set-up, hold and inactive Write Enable times should be measured relative to the Chip Enable waveform.

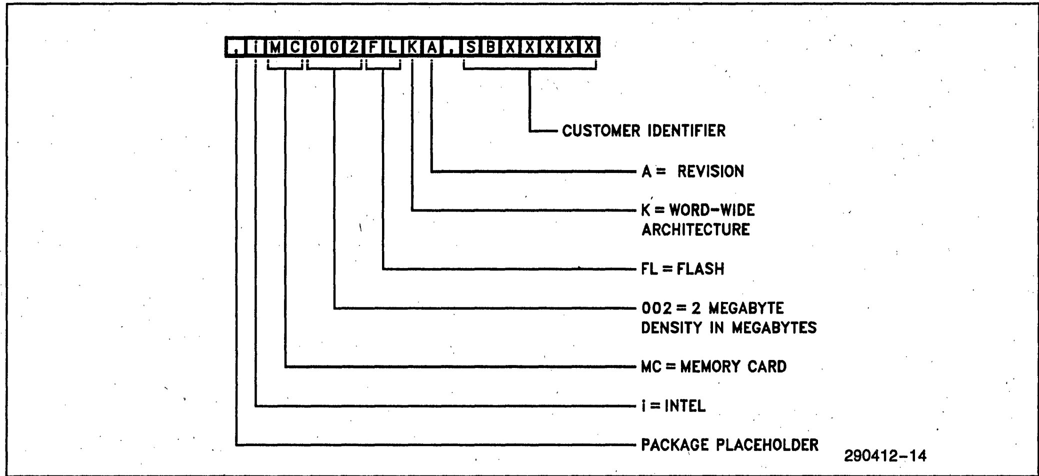


NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$.

Figure 13. Alternate AC Waveforms for Write Operations

4

ORDERING INFORMATION



ADDITIONAL INFORMATION	ORDER NUMBER
ER-20, "ETOX II Flash Memory Technology"	294005
RR-60, "ETOX II Flash Memory Reliability Data Summary"	293002
AP-343, "Solutions for High Density Applications using Flash Memory"	292079
RR-70, "Flash Memory Card Reliability Data Summary"	293007

REVISION HISTORY

Number	Description
-002	— Removed Preliminary — Removed ExCA Compliance Section — Clarified need for Valid Address during commands — Corrected $V_{PP} = V_{PPH}$ in Erase Algorithm — Increased I_{CC2}–I_{CC5} D.C. current specs for both Byte Wide and Word Wide modes — Revised and Updated Application Section discussion — Changed order number

iMC004FLKA
4-MBYTE FLASH MEMORY CARD

- **Inherent Nonvolatility (Zero Retention Power)**
  - **No Batteries Required for Back-up**
- **Over 1,000,000 Hours MTBF**
  - **More Reliable than Disk**
- **High-Performance Read**
  - **200 ns Maximum Access Time**
- **CMOS Low Power Consumption**
  - **40 mA Typical Active Current (X8)**
  - **800 μA Typical Standby Current**
- **Flash Electrical Zone-Erase**
  - **2 Seconds Typical per 256 kByte Zone**
  - **Multiple Zone-Erase**
- **Random Writes to Erased Zones**
  - **10 μs Typical Byte Write**
- **Write Protect Switch to Prevent Accidental Data Loss**
- **Command Register Architecture for Microprocesssor/Microcontroller Compatible Write Interface**
- **ETOX™ II Flash Memory Technology**
  - **5V Read, 12V Erase/Write**
  - **High-Volume Manufacturing Experience**
- **PCMCIA/JEIDA 68-Pin Standard**
  - **Byte- or Word-wide Selectable**
- **Independent Software & Hardware Vendor Support**
  - **Integrated System Solution Using Flash Filing Systems**

Intel's iMC004FLKA Flash Memory Card is the removable solution for storing and transporting important user data and application code. The combination of rewritability and nonvolatility make the Intel Flash Memory Card ideal for data acquisition and updatable firmware applications. Designing with Intel's Flash Memory Card enables OEM system manufacturers to produce portable and dedicated function systems that are higher performance, more rugged, and consume less power.

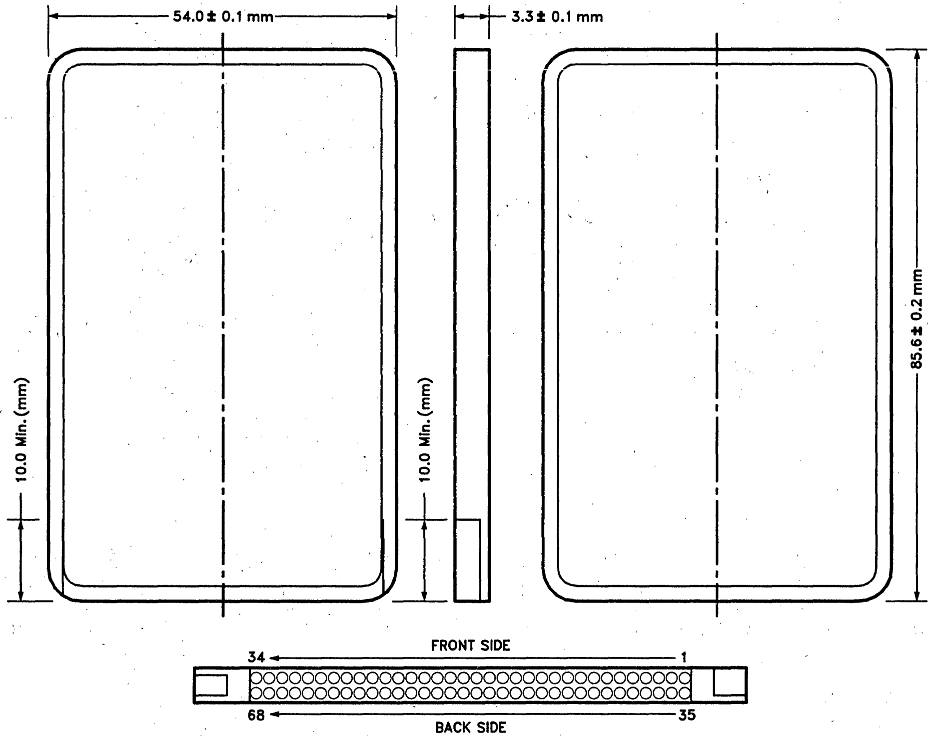
4

The iMC004FLKA conforms to the PCMCIA1.0 international standard, providing standardization at the hardware and data interchange level. OEMs may opt to write the Card Information Structure (CIS) at the memory card's address 00000H with a format utility. This information provides data interchange functional capability. The 200 ns access time allows for "execute-in-place" capability, for popular low-power microprocessors. Intel's 4-MByte Flash Memory Card operates in a byte-wide and word-wide configuration providing performance/power options for different systems.

Intel's Flash Memory card employs Intel's ETOX II Flash Memories. Filing systems, such as Microsoft's* Flash File System (FFS), facilitate data file storage and card erasure using a purely nonvolatile medium in the DOS environment. Flash filing systems, coupled with the Intel Flash Memory Card, effectively create an all-silicon nonvolatile read/write random access memory system that is more reliable and higher performance than disk-based memory systems.

*Microsoft is a trademark of Microsoft Corp.
ExCA™ is a trademark of Intel Corporporation.

October 1992
Order Number: 290388-003



1	GND	18	V_{PP1}	35	GND	52	V_{PP2}
2	D_3	19	A_{16}	36	$\overline{CD}_1$	53	NC
3	D_4	20	A_{15}	37	D_{11}	54	NC
4	D_5	21	A_{12}	38	D_{12}	55	NC
5	D_6	22	A_7	39	D_{13}	56	NC
6	D_7	23	A_6	40	D_{14}	57	NC
7	$\overline{CE}_1$	24	A_5	41	D_{15}	58	NC
8	A_{10}	25	A_4	42	$\overline{CE}_2$	59	NC
9	$\overline{OE}$	26	A_3	43	NC	60	NC
10	A_{11}	27	A_2	44	NC	61	$\overline{REG}$[1]
11	A_9	28	A_1	45	NC	62	$\overline{BVD}_2$[2]
12	A_8	29	A_0	46	A_{17}	63	$\overline{BVD}_1$[2]
13	A_{13}	30	D_0	47	A_{18}	64	D_8
14	A_{14}	31	D_1	48	A_{19}	65	D_9
15	$\overline{WE}$	32	D_2	49	A_{20}	66	D_{10}
16	NC	33	WP	50	A_{21}	67	$\overline{CD}_2$
17	V_{CC}	34	GND	51	V_{CC}	68	GND

NOTES:
1. $\overline{REG}$ = register memory select = No Connect (NC), unused. When $\overline{REG}$ is brought low, PCMCIA/JEIDA standard card information structure data is expected. This is accomplished by formatting the card with this data.
2. $\overline{BVD}$ = battery detect voltage = Pulled high through pull up resistor.

Figure 1. iMC004FLKA Pin Configurations

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{21}	I	**ADDRESS INPUTS** for memory locations. Addresses are internally latched during a write cycle.
D_0–D_{15}	I/O	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the card is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}_1$, $\overline{CE}_2$	I	**CARD ENABLE:** Activates the card's high and low byte control logic, input buffers, zone decoders, and associated memory devices. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory card and reduces power consumption to standby levels.
$\overline{OE}$	I	**OUTPUT ENABLE:** Gates the cards output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	I	**WRITE ENABLE** controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **NOTE:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP1}, V_{PP2}		**ERASE/WRITE POWER SUPPLY** for writing the command register, erasing the entire array, or writing bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY** (5V ±5%).
GND		**GROUND**
$\overline{CD}_1$, $\overline{CD}_2$	O	**CARD DETECT.** The card is detected when $\overline{CD}_1$ and $\overline{CD}_2$ = ground.
WP	O	**WRITE PROTECT.** All write operations are disabled with WP = active high.
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating.
$\overline{BVD}_1$, $\overline{BVD}_2$	O	**BATTERY VOLTAGE DETECT. NOT REQUIRED.**

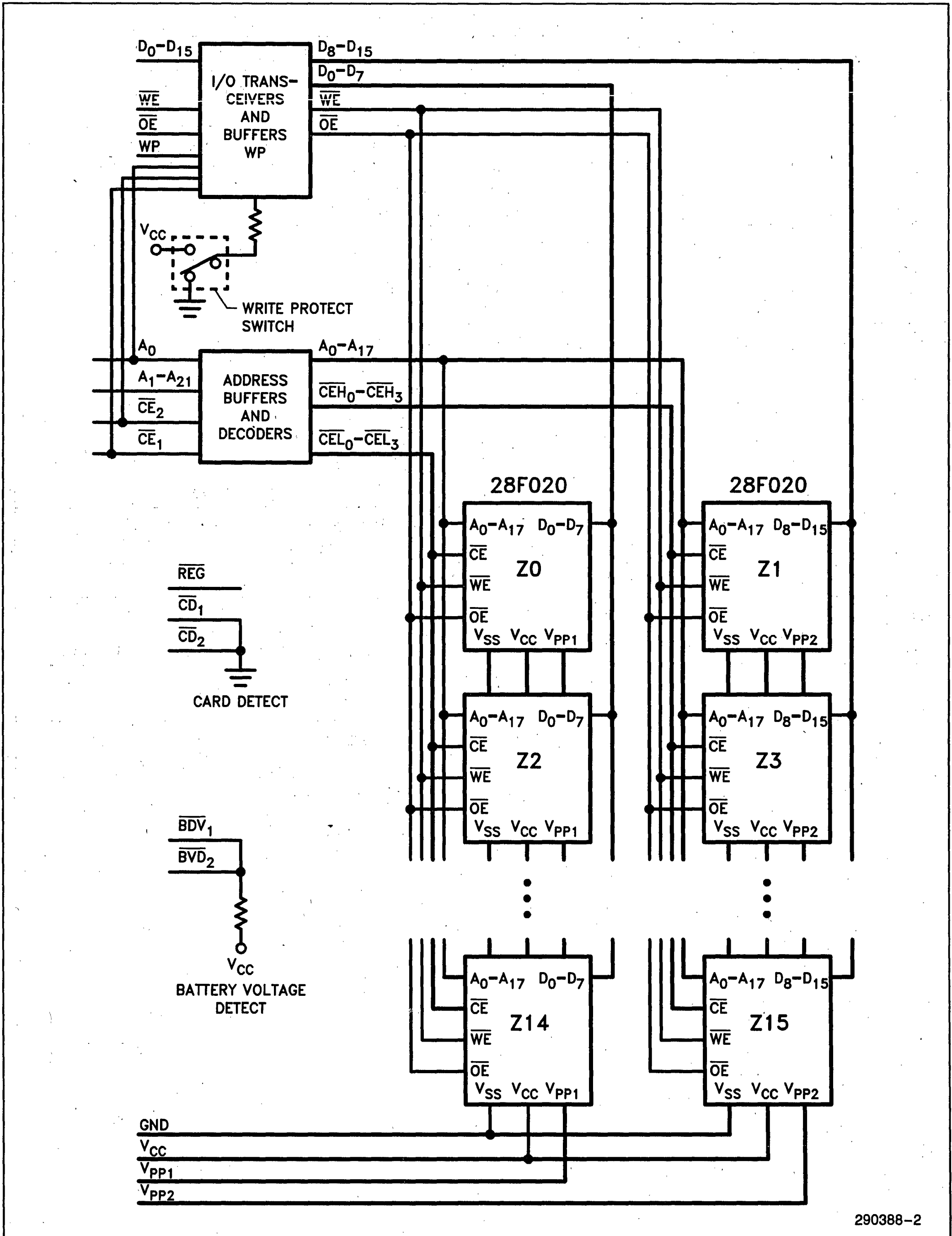


Figure 2. iMC004FLKA Block Diagram

APPLICATIONS

The iMC004FLKA Flash Memory Card allows for the storage of data files and application programs on a purely solid-state removable medium. System resident flash filing systems, such as Microsoft's Flash File System, allow Intel's ETOX II highly reliable Flash Memory Card to effectively function as a physical disk drive. The Intel Flash Memory Card in conjunction with flash filing systems provides an innovative alternative to both fixed hard disks and floppy disks in DOS-compatible portable PCs.

User application software stored on the flash memory card substantially reduces the slow disk-DRAM download process. Replacing the disk results in a dramatic enhancement of read performance and substantial reduction of active power consumption, size, and weight—considerations particularly important in portable PCs and equipment. The iMC004FLKA's high performance read access time and command register microprocessor write interface allows for use of the flash memory system in an "execute-in-place" architecture. This configuration eliminates the need for the redundancy associated with DRAM and Disk memory system architectures. ROM based operating systems, such as Microsoft's MS-DOS ROM Version allow for "instant-on" capability. This enables the design of PCs that boot, operate, store data files, and execute application code from/to purely nonvolatile memory.

Flash write performance is often 50% higher than hard disks for typical user file storage. This equates to ten times more performance when compared to "spun-down" disks, the common practice for portable machines.

Flash filing systems enable the storage and modification of data files by allocating flash memory space intelligently, thus minimizing the number of rewrite cycles. This management function allows the user to rewrite reliably to the flash memory card many more times than a fixed or floppy disk which concentrate rewrite operations into small fixed portions of the medium.

Flash filing systems implement Intel's Flash Memory Card as a redirected disk drive; similar to structures used in local area networks. This enables the end user to interact with the flash memory card in precisely the same way as a magnetic disk. Filing systems that run under popular operating systems, such as MS-DOS, can use the installed base of application software.

The Microsoft Flash File System enables the storage and modification of data files by utilizing a linked list directory structure that is evenly distributed along with the data across the memory card. The linked list approach minimizes file fragmentation losses by using variable-sized data structures rather than the standard sector/cluster method of disk-based systems.

The integration of the PCMCIA/JEIDA 68-pin interface with flash filing systems enables the end-user to transport user files and application code between portable PCs and desktop PCs with memory card Reader/Writers. Intel Flash PC cards provide durable nonvolatile memory storage for Notebook PCs on the road, facilitating simple transfer back into the desktop environment.

For systems currently using a static RAM/battery configuration for data acquisition, the iMC004FLKA's inherent nonvolatility eliminates the need for battery backup. The concern of battery failure no longer exists, an important consideration for portable computers and medical instruments, both requiring continuous operation. The iMC004FLKA consumes no power when the system is off. In addition, the iMC004FLKA offers a considerable cost and density advantage over memory cards based on static RAM with battery backup.

The flash memory card's electrical zone-erasure, byte writability, and complete nonvolatility fit well with data accumulation and recording needs. Electrical zone-erasure gives the designer the flexibility to selectively rewrite zones of data while saving other zones for infrequently updated look-up tables.

PRINCIPLES OF OPERATION

Intel's Flash Memory Card combines the functionality of two mainstream memory technologies: the rewritability of RAM and the nonvolatility of EPROM. The flash memory card consists of an array of individual memory devices, each of which defines a physical zone. The iMC004FLKA's memory devices erase as individual blocks, equivalent in size to the 256 kByte zone. Multiple zones can be erased simultaneously provided sufficient current for the appropriate number of zones (memory devices). Note, multiple zone erasure requires higher current from both the V_{PP} and V_{CC} power supplies. Erased zones can then be written in bit- or byte-at-a-time fashion and read randomly like RAM. Bit level write capability also supports disk emulation.

In the absence of high voltage on the $V_{PP1/2}$ pins, the iMC004FLKA remains in the read-only mode. Manipulation of the external memory card-control pin yields the standard read, standby, and output disable operations.

The same read, standby, and output disable operations are available when high voltage is applied to the $V_{PP1/2}$ pins. In addition, high voltage on $V_{PP1/2}$ enables erasure and rewriting of the accessed zone(s). All functions associated with altering zone contents—erase, erase verify, write, and write verify—are accessed via the command register.

Commands are written to the internal memory register(s), decoded by zone size, using standard microprocessor write timings. Register contents for a given zone serve as input to that zone's internal state-machine which controls the erase and rewrite circuitry. Write cycles also internally latch addresses and data needed for write and erase operations. With the appropriate command written to the register(s), standard microprocessor read timings output zone data, or output data for erase and write verification.

Byte-wide or Word-wide Selection

The flash memory card can be read, erased, and written in a byte-wide or word-wide mode. In the word-wide configuration V_{PP1} and/or $\overline{CE}_1$ control the LO-Byte while V_{PP2} and $\overline{CE}_2$ control the HI-Byte (A_0 = don't care).

Read, Write, and Verify operations are byte- or word-oriented, thus zone independent. Erase Setup and Begin Erase Commands are zone dependent such that commands written to any address within a 256 kByte zone boundary initiate the erase operation in that zone (or two 256 kByte zones under word-wide operation).

Conventional x8 operation uses $\overline{CE}_1$ active-low, with $\overline{CE}_2$ high, to read or write data through the D_0–D_7 only. "Even bytes" are accessed when A_0 is low, corresponding to the low byte of the complete x16 word. When A_0 is high, the "odd byte" is accessed by transposing the high byte of the complete x16 word onto the D_0–D_7 outputs. This odd byte corresponds to data presented on D_8–D_{15} pins in x16 mode.

Note that two zones logically adjacent in x16 mode are multiplexed through D_0–D_7 in x8 mode and are toggled by the A_0 address. Thus, zone specific erase operations must be kept discrete in x8 mode by addressing even bytes only for one-half of the zone pair, then addressing odd bytes only for the other half.

Card Detection

The flash memory card features two card detect pins ($\overline{CD}_{1/2}$) that allow the host system to determine if the card is properly loaded. Note that the two pins are located at opposite ends of the card. Each $\overline{CD}$ output should be read through a port bit. Should only one of the two bits show the card to be present, then the system should instruct the user to re-insert the card squarely into the socket. Card detection can also tell the system whether or not to redirect drives in the case of system booting. $\overline{CD}_{1/2}$ is active low, internally tied to ground.

Write Protection

The flash memory card features three types of write protection. The first type features a mechanical Write Protect Switch that disables the circuitry that control Write Enable to the flash devices. When the switch is activated, $\overline{WE}$ is forced high, which disables any writes to the Command Register. The second type of write protection is based on the PCMCIA/JEIDA socket. Unique pin length assignments provide protective power supply sequencing during hot insertion and removal. The third type operates via software control through the Command Register when the card resides in its connector. The Command Register of each zone is only active when $V_{PP1/2}$ is at high voltage. Depending upon the application, the system designer may choose to make $V_{PP1/2}$ power supply switchable—available only when writes are desired. When $V_{PP1/2} = V_{PPL}$, the contents of the register default to the read command, making the iMC004FLKA a read-only memory card. In this mode, the memory contents cannot be altered.

The system designer may choose to leave $V_{PP1/2} = V_{PPH}$, making the high voltage supply constantly available. In this case, all Command Register functions are inhibited whenever V_{CC} is below the write lockout voltage, V_{LKO}. (See the section on Power Up/Down Protection.) The iMC004FLKA is designed to accommodate either design practice, and to encourage optimization of the processor-memory card interface.

BUS OPERATIONS

Read

The iMC004FLKA has two control functions, both of which must be logically active, to obtain data at the outputs. Card Enable ($\overline{CE}$) is the power control and should be used for high and/or low zone(s) selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of accessed zone selection. In the byte-wide configuration, only one $\overline{CE}$ is required. The word-wide configuration requires both $\overline{CE}$s active low.

When $V_{PP1/2}$ is high (V_{PPH}), the read operations can be used to access zone data and to access data for write/erase verification. When $V_{PP1/2}$ is low (V_{PPL}), only read accesses to the zone data are allowed.

Output Disable

With Output Enable at a logic-high level (V_{IH}), output from the card is disabled. Output pins are placed in a high-impedance state.

Standby

With one Card Enable at a logic-high level, the standby operation disables one-half of the x16 output's read/write buffer. Further, only the zone corresponding to the selected address within the upper or lower $\overline{CE}_{1/2}$ bank is active at a time. (NOTE: A_0 must be low to select the low half of the x16 word when $\overline{CE}_2 = 1$ and $\overline{CE}_1 = 0$.) All other zones are deselected, substantially reducing card power consumption. For deselected banks, the outputs are placed in a high-impedance state, independent of the Output Enable signal. If the iMC004FLKA is deselected during erasure, writing, or write/erase verification, the accessed zone draws active current until the operation is terminated.

Intelligent Identifier Command

The manufacturer- and device-codes can be read via the Command Register, for instances where the iMC004FLKA is erased and rewritten in a universal reader/writer. Following a write of 90H to a zone's Command Register, a read from address location 00000H on any zone outputs the manufacturer code (89H). A read from address 0002H outputs the memory device code (BDH).

Write

Zone erasure and rewriting are accomplished via the Command Register, when high voltage is applied to $V_{PP1/2}$. The contents of the register serve as input to that zone's internal state-machine. The state-machine outputs dictate the function of the targeted zone.

The Command Register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

The Command Register is written by bringing Write Enable to a logic-low level (V_{IL}), while Card Enable(s) is/are low. Addresses are latched on the falling edge of Write Enable, while data is latched on the rising edge of the Write Enable pulse. Standard microprocessor write timings are used.

Refer to AC Write Charcteristics and the Erase/Write Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the V_{PP} pin(s), the contents of the zone Command Register(s) default to 00H, enabling read-only operations.

Placing high voltage on the V_{PP} pin(s) enable(s) read/write operations. Zone operations are selected by writing specific data patterns into the Command Register. Tables 3 and 4 define these iMC004FLKA register commands for both byte-wide and word-wide configurations.

All commands written to the Command Register require that the Zone Address be valid or the incorrect zone will receive the command. Any Command/Data Write or Data Read requires the correct Valid Address.

Table 2. Bus Operations

	Pins / Operation	Notes	[1, 7] V_{PP2}	[1, 7] V_{PP1}	A0	$\overline{CE}_2$	$\overline{CE}_1$	$\overline{OE}$	$\overline{WE}$	D_8-D_{15}	D_0-D_7
READ-ONLY	Read (x8)	8	V_{PPL}	V_{PPL}	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Even
	Read (x8)	9	V_{PPL}	V_{PPL}	V_{IH}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Odd
	Read (x8)	10	V_{PPL}	V_{PPL}	X	V_{IL}	V_{IH}	V_{IL}	V_{IH}	Data Out	Tri-state
	Read (x16)	11	V_{PPL}	V_{PPL}	X	V_{IL}	V_{IL}	V_{IL}	V_{IH}	Data Out	Data Out
	Output Disable		V_{PPL}	V_{PPL}	X	X	X	V_{IH}	V_{IH}	Tri-state	Tri-state
	Standby		V_{PPL}	V_{PPL}	X	V_{IH}	V_{IH}	X	X	Tri-state	Tri-state
READ/WRITE	Read (x8)	3, 8	V_{PPX}	V_{PPH}	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Even
	Read (x8)	3, 9	V_{PPH}	V_{PPX}	V_{IH}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	Tri-state	Data Out-Odd
	Read (x8)	10	V_{PPH}	V_{PPX}	X	V_{IL}	V_{IH}	V_{IL}	V_{IH}	Data Out	Tri-state
	Read (x16)	3, 11	V_{PPH}	V_{PPH}	X	V_{IL}	V_{IL}	V_{IL}	V_{IH}	Data Out	Data Out
	Write (x8)	5, 8	V_{PPX}	V_{PPH}	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{IL}	Tri-state	Data In-Even
	Write (x8)	9	V_{PPH}	V_{PPX}	V_{IH}	V_{IH}	V_{IL}	V_{IH}	V_{IL}	Tri-state	Data In-Odd
	Write (x8)	10	V_{PPH}	V_{PPX}	X	V_{IL}	V_{IH}	V_{IH}	V_{IL}	Data In	Tri-state
	Write (x16)	11	V_{PPH}	V_{PPH}	X	V_{IL}	V_{IL}	V_{IH}	V_{IL}	Data In	Data In
	Standby	4	V_{PPH}	V_{PPH}	X	V_{IH}	V_{IH}	X	X	Tri-state	Tri-state
	Output Disable		V_{PPH}	V_{PPH}	X	X	X	V_{IH}	V_{IL}	Tri-state	Tri-state

NOTES:

1. Refer to DC Characteristics. When $V_{PP1/2} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes may be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
3. Read operations with $V_{PP1/2} = V_{PPH}$ may access array data or the Intelligent Identifier codes.
4. With $V_{PP1/2}$ at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
5. Refer to Table 3 for valid Data-In during a write operation.
6. X can be V_{IL} or V_{IH}.
7. $V_{PPX} = V_{PPH}$ or V_{PPL}.
8. This x8 operation reads or writes the low byte of the x16 word on DQ_{0-7}, i.e., A_0 low reads "even" byte in x8 mode.
9. This x8 operation reads or writes the high byte of the x16 word on DQ_{0-7} (transposed from DQ_{8-15}), i.e., A_0 high reads "odd" byte in x8 mode.
10. This x8 operation reads or writes the high byte of the x16 on DQ_{8-15}. A_0 is "don't care."
11. A_0 is "don't care," unused in x16 mode. High and low bytes are presented simultaneously.

Table 3. Command Definitions Byte-Wide Mode

Command	Notes	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
			Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory		1	Write	RA	00H			
Read Intelligent ID Codes	4	3	Write	IA	90HT	Read		
Set-up Erase/Erase	5	2	Write	ZA	20H	Write	ZA	20H
Erase Verify	5	2	Write	EA	A0H	Read	EA	EVD
Set-up Write/Write	6	2	Write	WA	40H	Write	WA	WD
Write Verify	6	2	Write	WA	C0H	Read	WA	WVD
Reset	7	2	Write	ZA	FFH	Write	ZA	FFH

Table 4. Command Definitions Word-Wide Mode

Command	Notes	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
			Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory		1	Write	RA	0000H			
Read Intelligent ID Codes	4	3	Write	IA	9090H	Read		
Set-up Erase/Erase	5	2	Write	ZA	2020H	Write	ZA	2020H
Erase Verify	5	2	Write	EA	A0A0H	Read	EA	EVD
Set-up Write/Write	6	2	Write	WA	4040H	Write	WA	WD
Write Verify	6	2	Write	WA	C0C0H	Read	WA	WVD
Reset	7	2	Write	ZA	FFFFH	Write	ZA	FFFFH

NOTES:
1. Bus operations are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   RA = Read Address
   WA = Address of memory location to be written.
   ZA = Address of 256 kByte zones involved in erase operation.
   Addresses are latched on the falling edge of the Write Enable pulse.
3. ID = Data read from location IA during device identification. (Mfr = 89H, Device = BDH).
   EVD = Data read from location EA during erase verify.
   WD = Data to be written at location WA. Data is latched on the rising edge of Write Enable.
   WVD = Data read from location WA during write verify. WA is latched on the Write command.
4. Following the Read inteligent ID command, two read operations access manufacturer and device codes.
5. Figure 5 illustrates the Erase Algorithm.
6. Figure 6 illustrates the Write Algorithm.
7. The second bus cycle must be followed by the desired command register write.
8. The Reset command operation on a zone basic, To reset entire Card, requires reset write cycles to each zone.

Read Command

While $V_{PP1/2}$ is high, for erasure and writing, zone memory contents can be accessed via the read command. The read operation is initiated by writing 00H (0000H for the word-wide configuration) into the zone Command Register(s). Microprocessor read cycles retrieve zone data. The accessed zone remains enabled for reads until the Command Register(s) contents are altered.

The default contents of each zone's register(s) upon $V_{PP1/2}$ power-up is 00H (00000H for word-wide). This default value ensures that no spurious alteration of memory card contents occurs during the $V_{PP1/2}$ power transition. Where the $V_{PP1/2}$ supply is left at V_{PPH}, the memory card powers-up and remains enabled for reads until the command Register contents of targeted zones are changed. Refer to the AC Read Characteristics and Waveforms for specific timing parameters.

Intelligent Identifier Command

Each zone of the iMC004FLKA contains an Intelligent Identifier to identify memory card device characteristics. The operation is initiated by writing 90H (9090H for word-wide) into the Command Register(s) with Zone Address. Following the command write, a read cycle from address 00000H retrieves the manufacturer code 89H (8989H for word-wide). A read cycle from address 0002H returns the device code BDH (BDBDH for word-wide). To terminate the operation, it is necessary to write another valid command into the register(s).

Set-up Erase/Erase Commands

Set-up Erase stages the targeted zone(s) for electrical erasure of all bytes in the zone. The set-up erase operation is performed by writing 20H to the Command Register (2020H for word-wide) with Zone Address.

To commence zone-erasure, the erase command (20H or 2020H) must again be written to the register(s). The erase operation begins with the rising edge of the Write-Enable pulse and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that zone memory contents are not accidentally erased. Also, zone-erasure can only occur when high voltage is applied to the $V_{PP1/2}$ pins. In the absence of this high voltage, zone memory contents are protected against erasure. Refer to AC Erase Characterstics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all of the bytes of the zone in parallel. After each erase operation, all bytes in the zone must be individually verified. In byte-mode operations, zones are segregated by A_0 in odd and even banks; erase and erase verify operations must be done in complete passes of even-bytes-only then odd-bytes-only. See the Erase Algorithm for byte-wide mode. The erase verify operation is initiated by writing A0H (A0A0H for word-wide) into the Command Register(s). The address for the byte(s) to be verified must be supplied as it is latched on the falling edge of the Write Enable pulse. The register write terminates the erase operation with the rising edge of its Write Enable pulse.

The enabled zone applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased. Similarly, reading FFFFH from the addressed word indicates that all bits in the word are erased.

The erase-verify command must be written to the Command Register prior to each byte (word) verification to latch its address. The process continues for each byte (word) in the zone(s) until a byte (word) does not return FFH (FFFFH) data, or the last address is accessed.

In the case where the data read is not FFH (FFFFH), another erase operation is performed. (Refer to Set-up Erase/Erase.) Verification then resumes from the address of the last-verified byte (word). Once all bytes (words) in the zone(s) have been verified, the erase step is complete. The accessed zone can now be written. At this point, the verify operation is terminated by writing a valid command (e.g., Write Set-up) to the Command Register. The Erase algorithms for byte-wide and word-wide configurations illustrate how commands and bus operations are combined to perform electrical erasure of the iMC001FLKA. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Set-up Write/Write Commands

Set-up write is a command-only operation that stages the targeted zone for byte writing. Writing 40H (4040H) into the Command Register(s) performs the set-up operation.

Once the write set-up operation is performed, the next Write Enable pulse causes a transition to an active write operation. Addresses are internally latched on the falling edge of the Write Enable pulse. Data is internally latched on the rising edge of the Write Enable pulse. The rising edge of Write Enable also begins the write operation. The write operation terminates with the next rising edge of Write Enable, which is used to write the verify command. Refer to AC Write Characteristics and Waveforms for specific timing parameters.

Write Verify Command

The iMC004FLKA is written on a byte-by-byte or word-by-word basis. Byte or word writing may occur sequentially or at random. Following each write operation, the byte or word just written must be verified.

The write-verify operation is initiated by writing C0H (C0C0H) into the Command Register(s) with the correct address. The register write(s) terminate(s) the write operation with the rising edge of its Write Enable pulse. The write-verify operation stages the accessed zone(s) for verification of the byte or word last written. The zone(s) apply(ies) an internally-generated margin voltage to the byte or word. A microprocessor read cycle outputs the data. A successful comparison between the written byte or word and true data means that the byte or word is successfully written. The write operation then proceeds to the next desired byte or word location. The Write algorithms for byte-wide and word-wide configurations illustrate how commands are combined with bus operations to perform byte and word writes. Refer to AC Write Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or write-command sequences. Following either set-up command (erase or write) with two consecutive writes of FFH (FFFFH for word-wide) will safely abort the operation. Zone memory contents will not be altered. A valid command must then be written to place the accessed zone in the desired state.

EXTENDED ERASE/WRITE CYCLING

Intel has designed extended cycling capability into its ETOX II flash memory technology enabling a flash memory card with a MTBF that is approximately 20 times more reliable than rotating disk technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field minimizes the probability of oxide defects in the region. The lower electric field greatly reduces oxide stress and the probability of failure.

WRITE ALGORITHMS

The write algorithm(s) use write operations of 10 μs duration. Each operation is followed by a byte or word verification to determine when the addressed byte or word has been successfully written. The algorithm(s) allows for up to 25 write operations per byte or word, although most bytes and words verify on the first or second operation. The entire sequence of writing and byte/word verification is performed with V_{PP} at high voltage.

ERASE ALGORITHM

The Erase algorithm(s) yield(s) fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the write algorithm, to simultaneously remove charge from all bits in the accessed zone(s).

Erasure begins with a read of memory zone contents. Reading FFH (FFFFH) data from the accessed zone(s) can be immediately followed by writing to the desired zone(s).

For zones being erased and rewritten, uniform and reliable erasure is ensured by first writing all bits in the accessed zone to their charged state (data = 00H byte-wide, 0000OH word-wide). This is accomplished, using the write algorithm, in approximately four seconds per zone.

Erase execution then continues with an initial erase operation. Erase verification (data = FFH byte-wide, FFFFH word-wide) begins at address 00000H and continues through the zone to the last address, or until data other than FFH (FFFFH) is encountered. (Note: byte-wide erase operation requires separate even- and odd-address passes to handle the individual 256 kByte zones.) With each erase operation, an increasing number of bytes or words verify to the erased state. Erase efficiency may be improved by storing the address of the last byte or word verified in a register(s). Following the next erase operation, verification starts at the stored address location. Follow this procedure until all bytes in the zone are erased. Then, re-start the procedure for the next zone or word-wide zone pair. Erasure typically occurs in two seconds per zone.

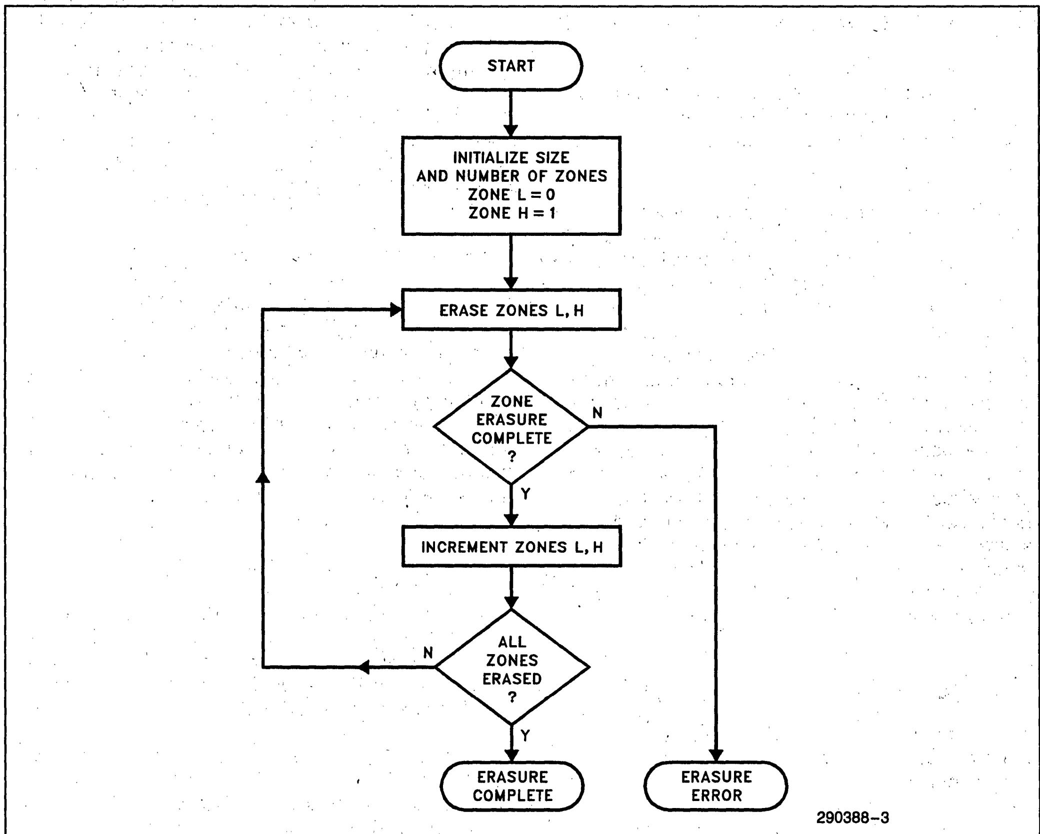
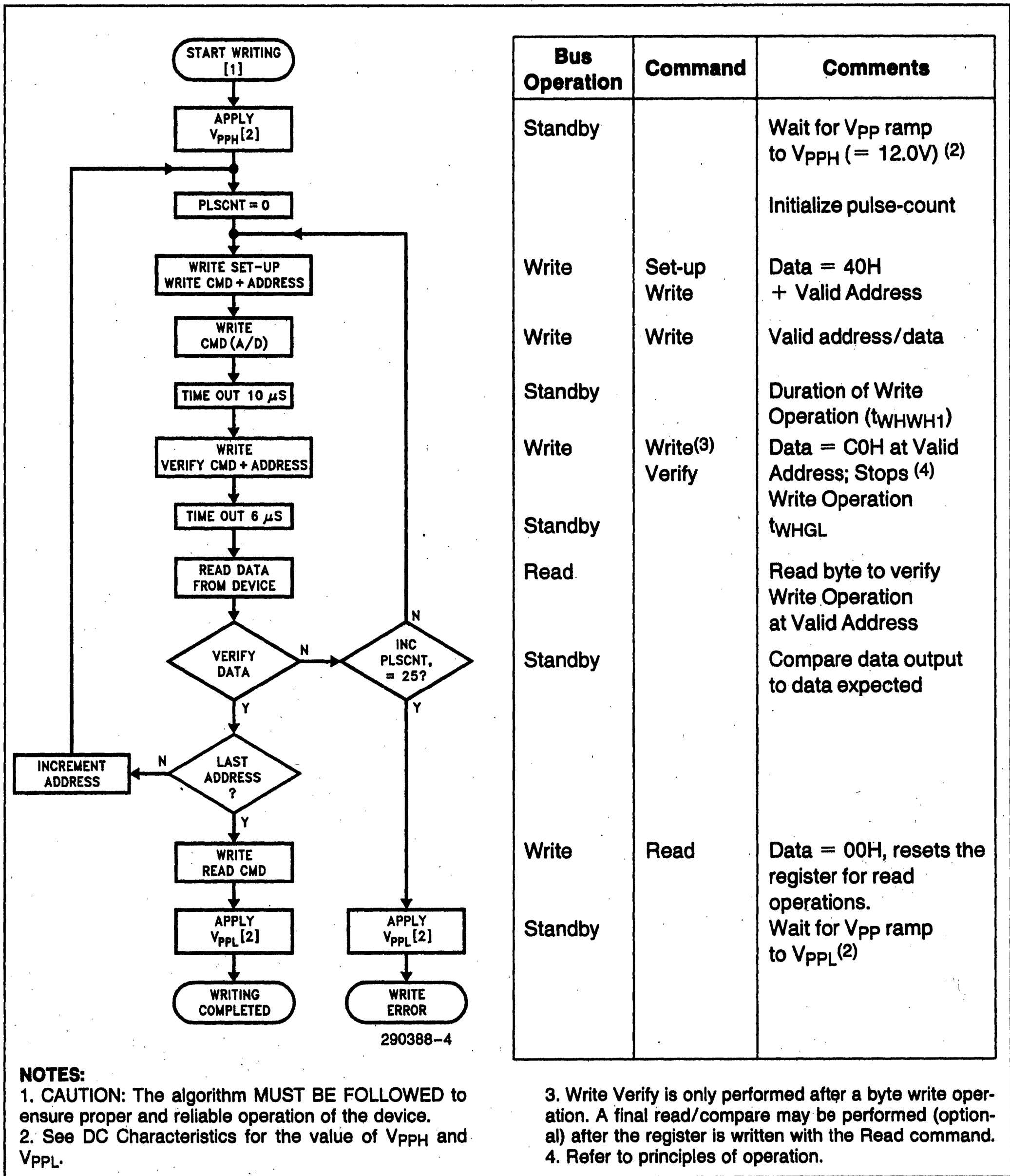


Figure 3. Full Card Erase Flow

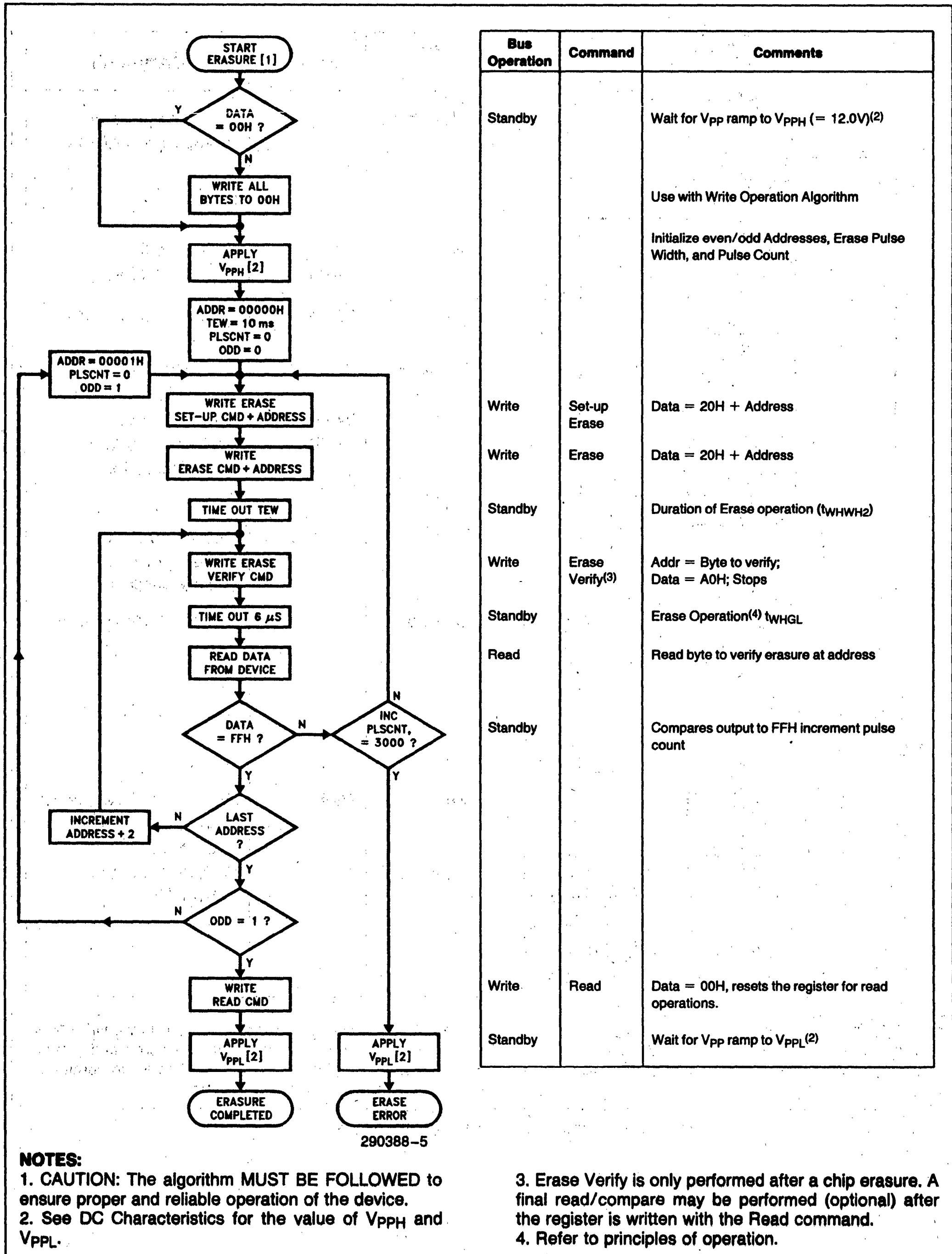


Bus Operation	Command	Comments
Standby		Wait for V_{PP} ramp to V_{PPH} (= 12.0V) (2)
		Initialize pulse-count
Write	Set-up Write	Data = 40H + Valid Address
Write	Write	Valid address/data
Standby		Duration of Write Operation (t_{WHWH1})
Write	Write(3) Verify	Data = C0H at Valid Address; Stops (4) Write Operation
Standby		t_{WHGL}
Read		Read byte to verify Write Operation at Valid Address
Standby		Compare data output to data expected
Write	Read	Data = 00H, resets the register for read operations.
Standby		Wait for V_{PP} ramp to V_{PPL}(2)

NOTES:

1. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.
2. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
3. Write Verify is only performed after a byte write operation. A final read/compare may be performed (optional) after the register is written with the Read command.
4. Refer to principles of operation.

Figure 4. Write Algorithm for Byte-Wide Mode



Bus Operation	Command	Comments
Standby		Wait for V_{PP} ramp to V_{PPH} (= 12.0V)(2)
		Use with Write Operation Algorithm
		Initialize even/odd Addresses, Erase Pulse Width, and Pulse Count
Write	Set-up Erase	Data = 20H + Address
Write	Erase	Data = 20H + Address
Standby		Duration of Erase operation (t_{WHWH2})
Write	Erase Verify(3)	Addr = Byte to verify; Data = A0H; Stops
Standby		Erase Operation(4) t_{WHGL}
Read		Read byte to verify erasure at address
Standby		Compares output to FFH increment pulse count
Write	Read	Data = 00H, resets the register for read operations.
Standby		Wait for V_{PP} ramp to V_{PPL}(2)

NOTES:

1. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.
2. See DC Characteristics for the value of V_{PPH} and V_{PPL}.
3. Erase Verify is only performed after a chip erasure. A final read/compare may be performed (optional) after the register is written with the Read command.
4. Refer to principles of operation.

Figure 5. Erase Algorithm for Byte-Wide Mode

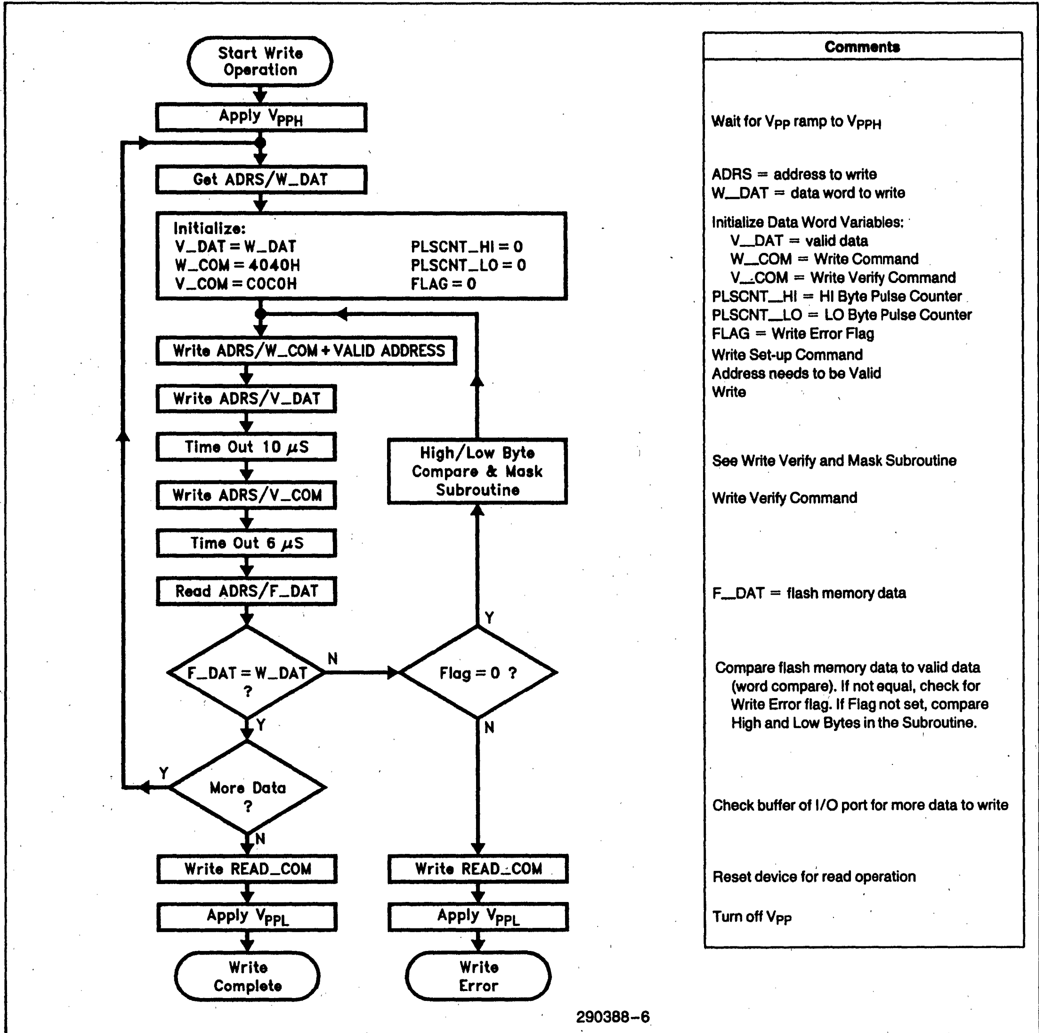
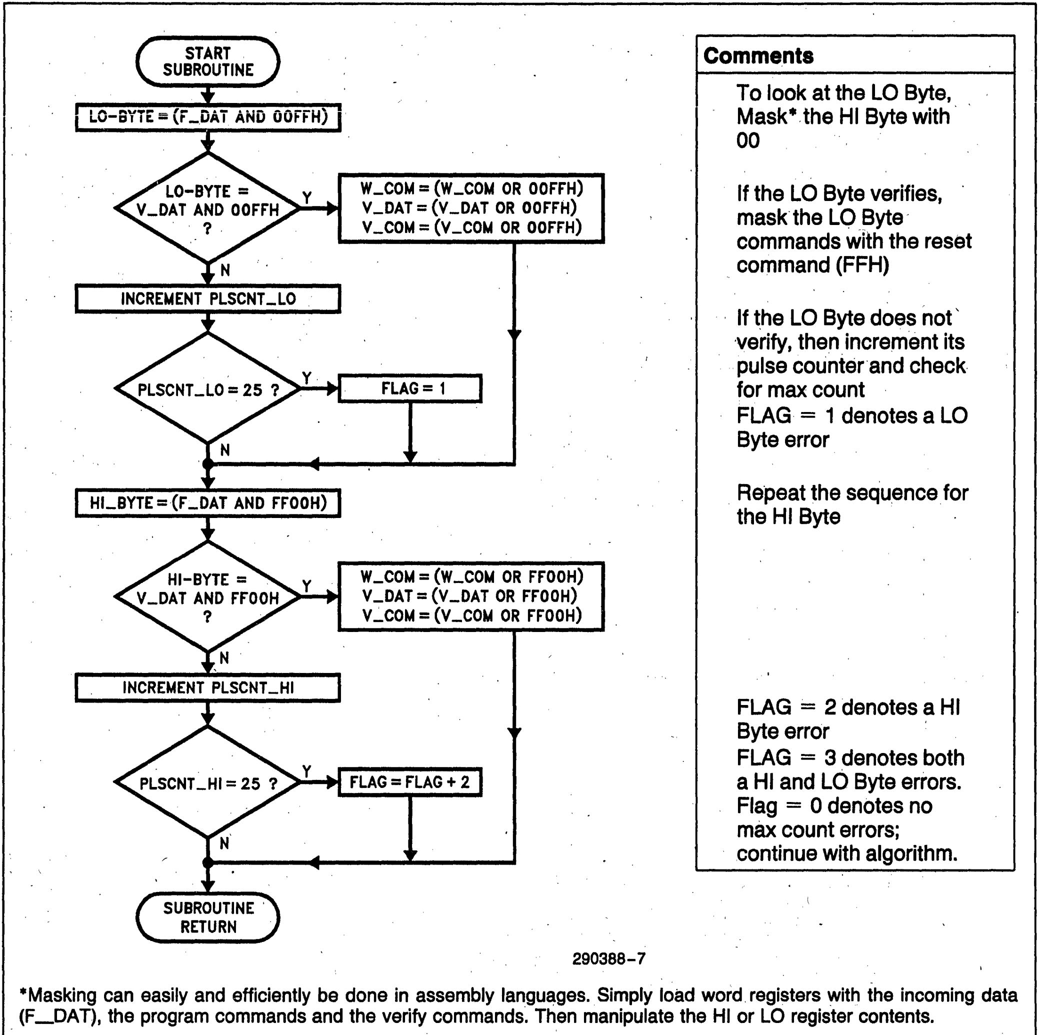


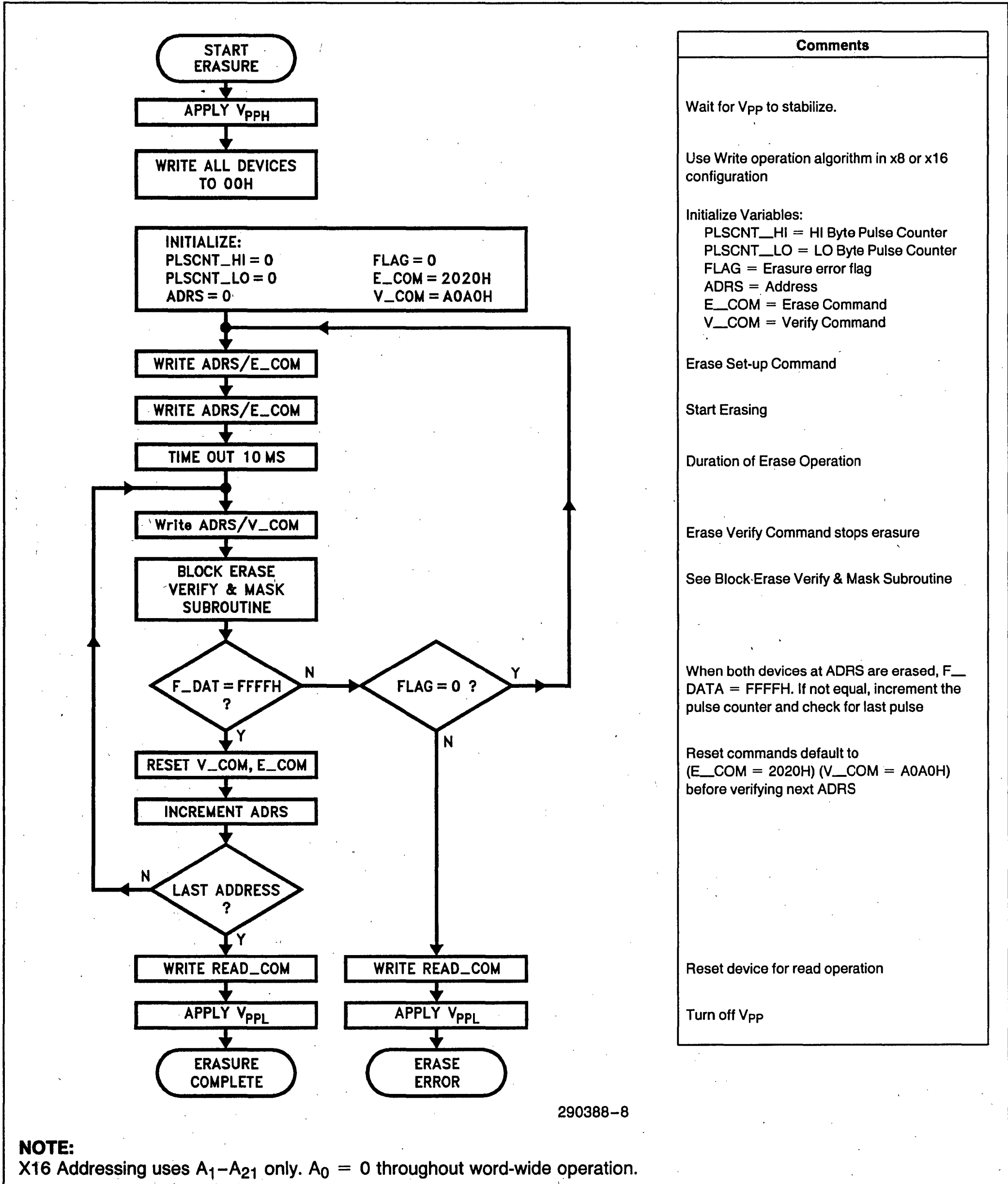
Figure 6. Write Algorithm for Word-Wide Mode

4



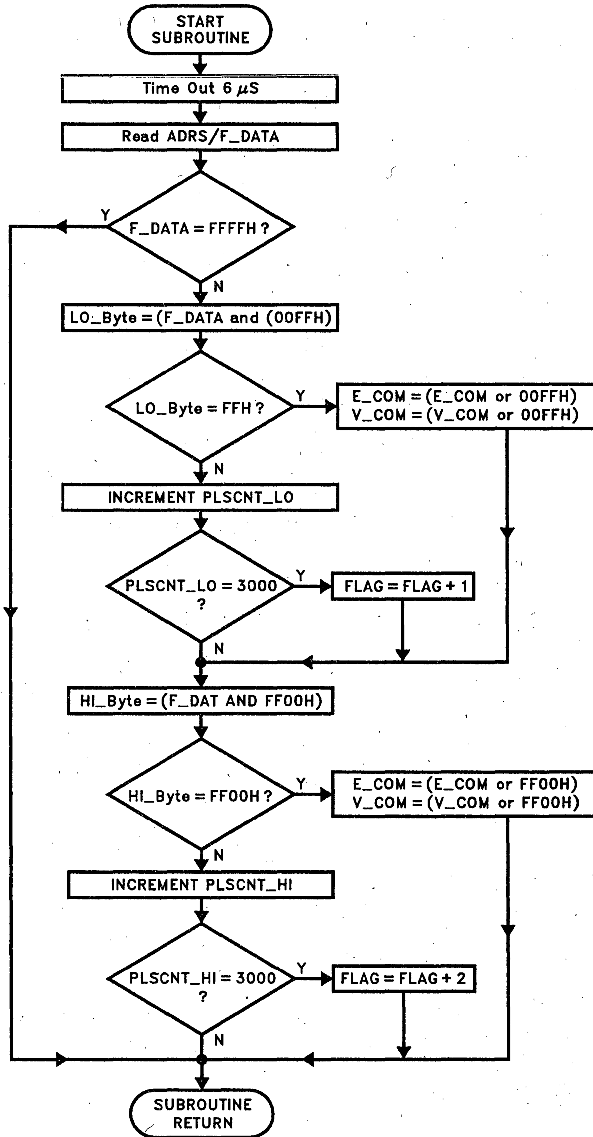
*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming data (F_DAT), the program commands and the verify commands. Then manipulate the HI or LO register contents.

Figure 7. Write Verify and Mask Subroutine for Word-Wide Mode



NOTE:
X16 Addressing uses A_1–A_{21} only. A_0 = 0 throughout word-wide operation.

Figure 8. Erase Algorithm for Word-Wide Mode



Comments
This subroutine reads the data word (F_DATA). It then masks the HI or LO Byte of the Erase and Verify commands from executing during the next operation.
If both HI and LO Bytes verify, then return.
Mask* the HI Byte with 00H.
If the LO Byte verifies erasure, then mask* the next erase and verify commands with FFH (RESET).
If the LO Byte does not verify, then increment its pulse counter and check for max count. FLAG = 1 denotes a LO Byte error.
Repeat the sequence for the HI Byte.
Flag = 2 denotes a HI Byte error. Flag = 3 denotes both a HI and LO Byte errors. FLAG = 0 denotes no max count errors; continue with algorithm.

290388-9

*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming data (F_DAT), the program commands and the verify commands. Then manipulate the HI or LO register contents.

Figure 9. Erase Verify and Mask Subroutine for Word-Wide Mode

SYSTEM DESIGN CONSIDERATIONS

Three-Line Control

Three-line control provides for:

a. the lowest possible power dissipation and.

b. complete assurance that output bus contention will not occur.

To efficiently use these three control inputs, an address-decoder output should drive $\overline{CE}_{1,2}$, while the system's Read signal controls the card $\overline{OE}$ signal, and other parallel zones. This, coupled with the internal zone decoder, assures that only enabled memory zones have active outputs, while deselected zones maintain the low power standby condition.

Power-Supply Decoupling

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current issues—standby, active and transient current peaks, produced by falling and rising edges of $\overline{CE}_{1/2}$. The capacitive and inductive loads on the card and internal flash memory zones determine the magnitudes of these peaks.

Three-line control and proper decoupling capacitor selection will suppress transient voltage peaks. The iMC004FLKA features on-card ceramic decoupling capacitors connected between V_{CC} and V_{SS}, and between V_{PP1}/V_{PP2} and V_{SS}.

The card connector should also have a 4.7 µF electrolytic capacitor between V_{CC} and V_{SS}, as well as between V_{PP1}/V_{PP2} and V_{SS}. The bulk capacitors will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

Power Up/Down Protection

The PCMCIA/JEIDA socket is specified, via unique Pin lengths, to properly sequence the power supplies to the flash memory card. This assures that hot insertion and removal will not result in card damage or data loss.

Each zone in the iMC004FLKA is designed to offer protection against accidental erasure or writing, caused by spurious system-level signals that may exist during power transitions. The card will power-up into the read state.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}_{1,2}$ must be low for a command write, driving either to V_{IH} will inhibit writes. With its control register architecture, alteration of zone contents only occurs after successful completion of the two-step command sequences.

While these precautions are sufficient for most applications, it is recommended that V_{CC} reach its steady state value before raising $V_{PP1/2}$ above V_{CC} + 2.0V. In addition, upon powering-down, $V_{PP1/2}$ should be below V_{CC} + 2.0V, before lowering V_{CC}.

Absolute Maximum Ratings*

Operating Temperature
During Read .... 0°C to + 60°C(1)
During Erase/Write .... 0°C to + 60°C

Temperature Under Bias .... −10°C to +70°C

Storage Temperature .... −30°C to +70°C

Voltage on Any Pin with
Respect to Ground .... −2.0V to +7.0V(2)

V_{PP1}/V_{PP2} Supply Voltage with
Respect to Ground
During Erase/Write .... −2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage with
Respect to Ground .... −2.0V to +7.0V(2)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:

1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
3. Maximum DC input voltage on V_{PP1}/V_{PP2} may overshoot to +14.0V for periods less than 20 ns.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature	0	60	°C	For Read-Only and Read/Write Operations
V_{CC}	V_{CC} Supply Voltage	4.75	5.25	V	
V_{PPH}	Active V_{PP1}, V_{PP2} Supply Voltages	11.40	12.60	V	
V_{PPL}	V_{PP} During Read Only Operations	0.00	6.50	V	

DC CHARACTERISTICS—Byte Wide Mode

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{LI}	Input Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.8	1.6	mA	$V_{CC} = V_{CC}$ max, CE = V_{CC} ±0.2V
				4	7	mA	$\overline{CE} = V_{IH}$, $V_{CC} = V_{CC}$ max
I_{CC1}	V_{CC} Active Read Current	1, 2		40	70	mA	$V_{CC} = V_{CC}$ max $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Write Current	1, 2		5.0	15	mA	Writing in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		10	20	mA	Erasure in Progress
I_{CC4}	V_{CC} Write Verify Current	1, 2		10	20	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress

DC CHARACTERISTICS—Byte Wide Mode (Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{CC5}	V_{CC} Erase Verify Current	1, 2		10	20	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±80	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	1, 3		0.7	1.6	mA	$V_{PP} > V_{CC}$
					±0.08		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Write Current	1, 3		8.0	30	mA	$V_{PP} = V_{PPH}$ Write in Progress
I_{PP3}	V_{PP} Erase Current	1, 3		10	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Write Verify Current	1, 3		3.0	6.0	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 3		3.0	6.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.4		$V_{CC} \pm 0.3$	V	
V_{OL}	Output Low Voltage				0.40	V	$I_{OL} = 3.2$ mA $V_{CC} = V_{CC}$ min
V_{OH1}	Output High Voltage		3.8			V	$I_{OH} = -2.0$ mA $V_{CC} = V_{CC}$ min
V_{PPL}	V_{PP} During Read-Only Operations		0.00		6.5	V	Note: Erase/Write are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} During Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at $V_{CC} = 5.0V$, $V_{PP} = 12.0V$, $T = 25°C$.
2. 1 chip active and 15 in standby for byte-wide mode.
3. Assumes 1 V_{PP} is active.

DC CHARACTERISTICS—Word Wide Mode

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{LI}	Input Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1		±1.0	±20	μA	$V_{CC} = V_{CC}$ max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		0.8	1.6	mA	$V_{CC} = V_{CC}$ max, $\overline{CE} = V_{CC} \pm 0.2V$
				4	7	mA	$\overline{CE} = V_{IH}$, $V_{CC} = V_{CC}$ max

4

DC CHARACTERISTICS—Word Wide Mode (Continued)

Symbol	Parameter	Notes	Limits Min	Limits Typical	Limits Max	Unit	Test Conditions
I_{CC1}	V_{CC} Active Read Current	1, 2		50	100	mA	$V_{CC} = V_{CC}$ max $\overline{CE} = V_{IL}$ f = 6 MHz, $I_{OUT} = 0$ mA
I_{CC2}	V_{CC} Write Current	1, 2		5.0	25	mA	Writing in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		15	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Write Verify Current	1, 2		15	30	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		15	30	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±160	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current	1, 3		1.5	3.0	mA	$V_{PP} > V_{CC}$
	or Standby Current				±.16		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Write Current	1, 3		17	63	mA	$V_{PP} = V_{PPH}$ Write in Progress
I_{PP3}	V_{PP} Erase Current	1, 3		20	60	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Write Verify Current	1, 3		5.0	12	mA	$V_{PP} = V_{PPH}$ Write Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 3		5.0	12	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.4		$V_{CC} \pm 0.3$	V	
V_{OL}	Output Low Voltage				0.40	V	$I_{OL} = 3.2$ mA $V_{CC} = V_{CC}$ min
V_{OH1}	Output High Voltage		3.8			V	$I_{OH} = -2.0$ mA $V_{CC} = V_{CC}$ min
V_{PPL}	V_{PP} During Read-Only Operations		0.00		6.5	V	Note: Erase/Write are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} During Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at $V_{CC} = 5.0V$, $V_{PP} = 12.0V$, T = 25°C.
2. 2 chips active and 14 in standby for word-wide mode.
3. Assumes 2 V_{PPS} are active.

CAPACITANCE $T = 25°C, f = 1.0 MHz$

Symbol	Parameter	Notes	Limits		Unit	Conditions
			Min	Max		
C_{IN1}	Address Capacitance			40	pF	$V_{IN} = 0V$
C_{IN2}	Control Capacitance			40	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance			40	pF	$V_{OUT} = 0V$
$C_{I/O}$	I/O Capacitance			40	pF	$V_{I/O} = 0V$

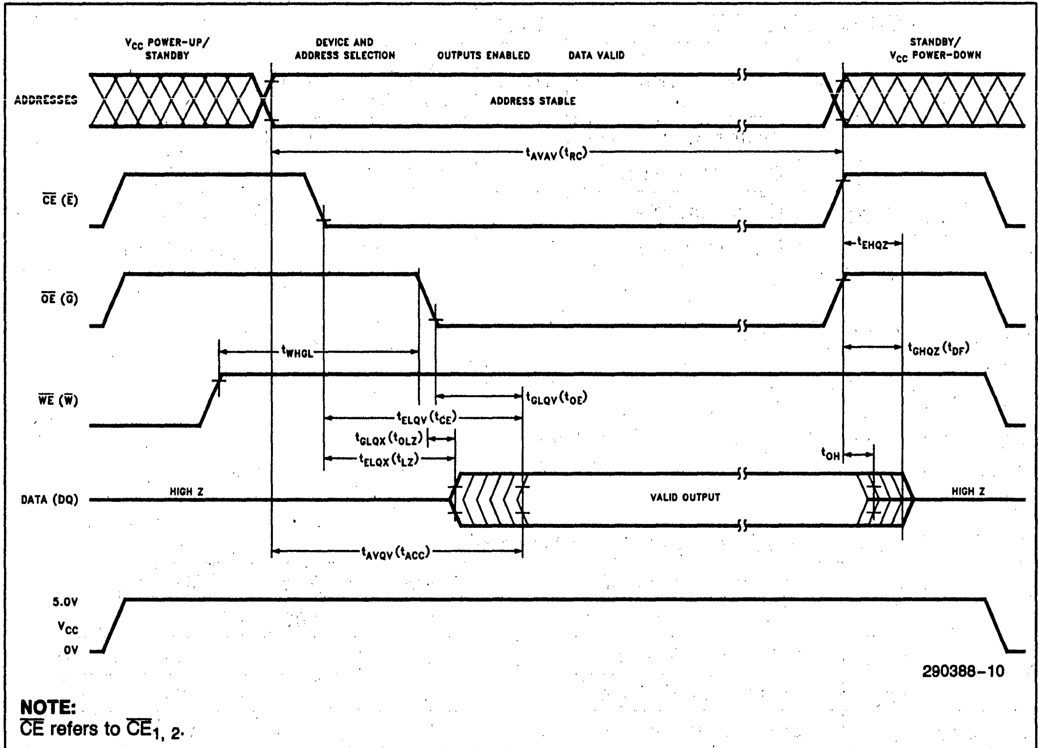
AC TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ...... 10 ns
Input Pulse Levels .... V_{OL} and V_{OH1}
Input Timing Reference Level .... V_{IL} and V_{IH}
Output Timing Reference Level .... V_{IL} and V_{IH}

AC CHARACTERISTICS—Read-Only Operations

Symbol	Characteristic	Notes	Min	Max	Unit
t_{AVAV}/t_{RC}	Read Cycle Time	2	200		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time	2		200	ns
t_{AVQV}/t_{ACC}	Address Access Time	2		200	ns
t_{GLQV}/t_{OE}	Output Enable Access Time	2		100	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	2	5		ns
t_{EHQZ}	Chip Disable to Output in High Z	2		60	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	2	5		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	2		60	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	1, 2	5		ns
t_{WHGL}	Write Recovery Time before Read	2	6		µs

NOTES:
1. Whichever occurs first.
2. Rise/Fall Time ≤ 10 ns.



NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$.

Figure 10. AC Waveforms for Read Operations

AC CHARACTERISTICS—For Write/Erase Operations

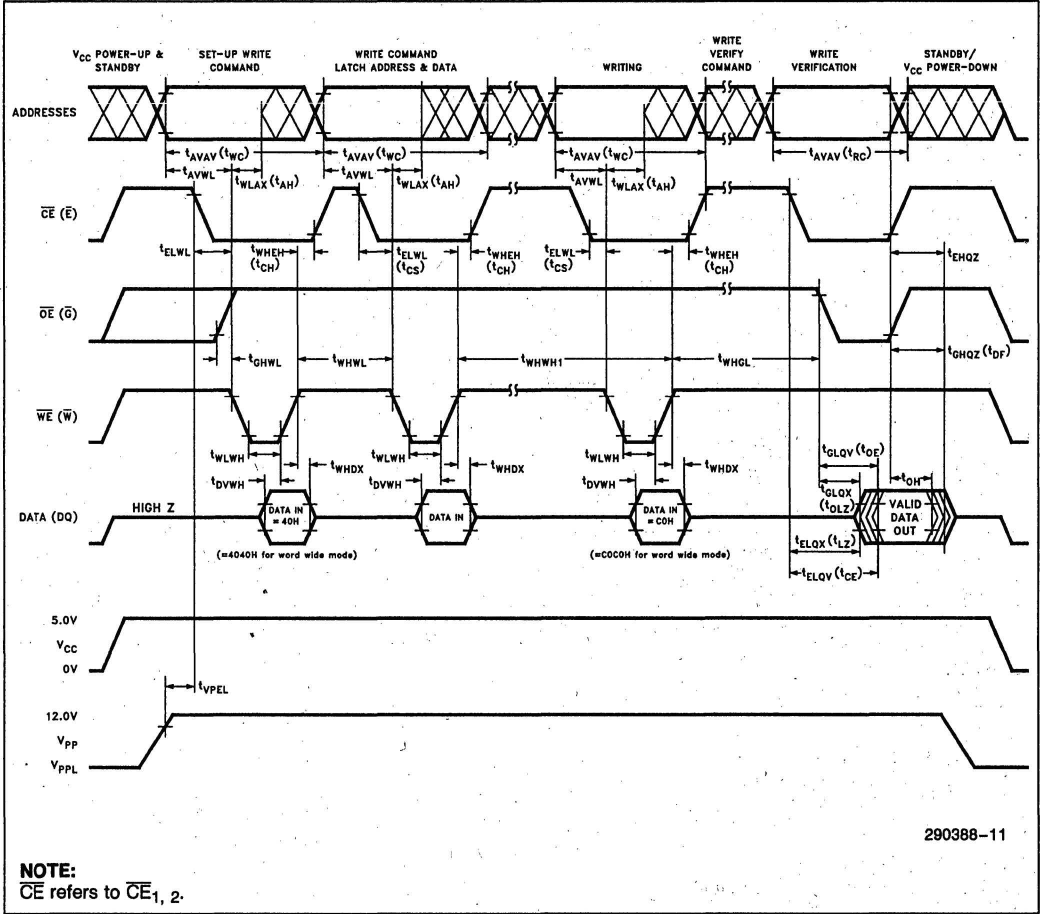
Symbol	Characteristic	Notes	Min	Max	Unit
t_{AVAV}/t_{WC}	Write Cycle Time	1, 2	200		ns
t_{AVWL}/t_{AS}	Address Set-up Time	1, 2	0		ns
t_{WLAX}/t_{AH}	Address Hold Time	1, 2	100		ns
t_{DVWH}/t_{DS}	Data Set-up Time	1, 2	80		ns
t_{WHDX}/t_{DH}	Data Hold Time	1, 2	30		ns
t_{WHGL}	Write Recovery Time before Read	1, 2	6		μs
t_{GHWL}	Read Recovery Time before Write	1, 2	0		μs
t_{WLOZ}	Output High-Z from Write Enable	1, 2	5		ns
t_{WHOZ}	Output Low-Z from Write Enable	1, 2		60	ns
t_{ELWL}/t_{CS}	Chip Enable Set-up Time before Write	1, 2	40		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time	1, 2	0		ns
t_{WLWH}/t_{WP}	Write Pulse Width	1, 2	100		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High	1, 2	20		ns
t_{WHWH1}	Duration of Write Operation	1, 2, 3	10		μs
t_{WHWH2}	Duration of Erase Operation	1, 2, 3	9.5		ms
t_{VPEL}	V_{PP} Set-up Time to Chip Enable Low	1, 2	100		ns

NOTES:
1. Read timing parameters during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Rise/Fall time ≤ 10 ns.
3. The integrated stop timer terminates the write/erase operations, thereby eliminating the need for a maximum specifiction.

ERASE/WRITE PERFORMANCE

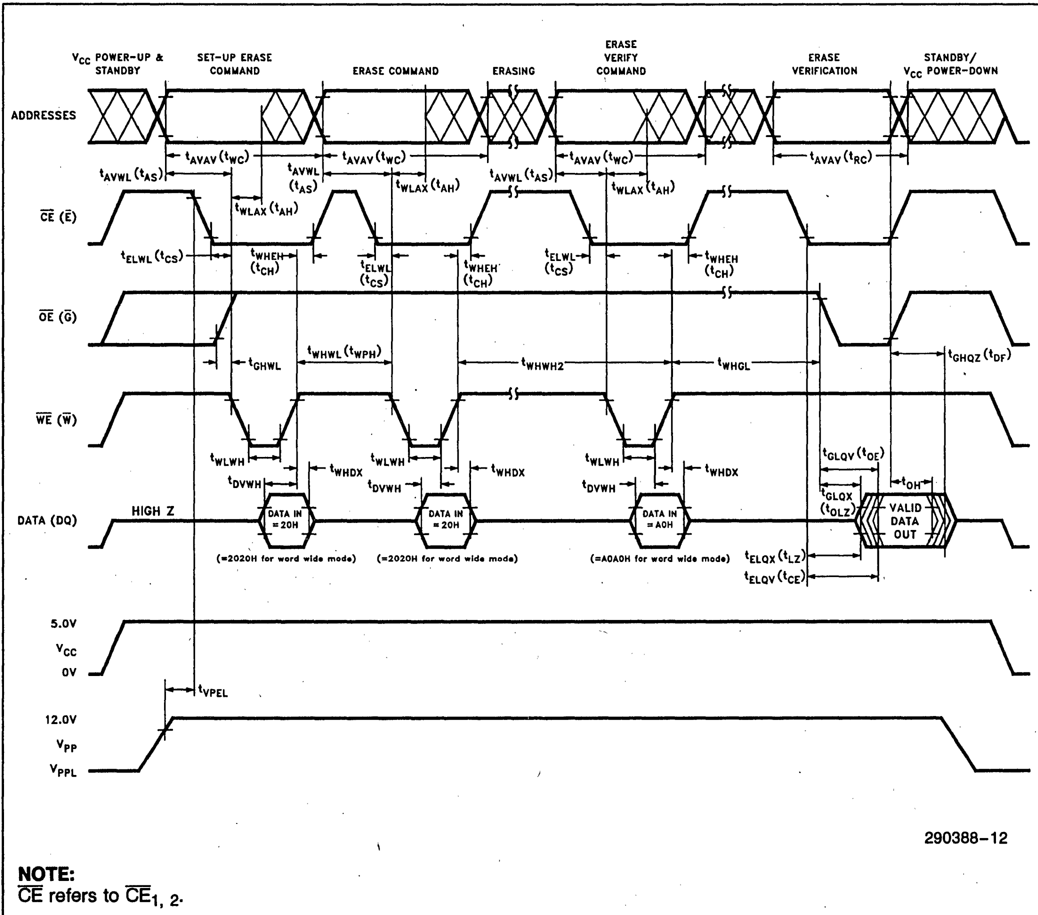
Parameter	Notes	Min	Typ	Max	Unit
Zone Erase Time	1, 3, 4		2.0	30	sec
Zone Write Time	1, 2, 4		4.0	25	sec
MTBF	5		$10^{(6)}$		Hrs

NOTES:
1. 25°C, 12.0V V_{PP}.
2. Minimum byte writing time excluding system overhead is 16 μs (10 μs program + 6 μs write recovery), while maximum is 400 μs/byte (16 μs x 25 loops allowed by algorithm). Max chip write time is specified lower than the worst case allowed by the write algorithm since most bytes write significantly faster than the worst case byte.
3. Excludes 00H writing Prior to Erasure.
4. One zone equals 256 kBytes.
5. MTBF – Mean Time between Failure, 50% failure point for disk drives.



NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$.

Figure 11. AC Waveforms for Write Operations



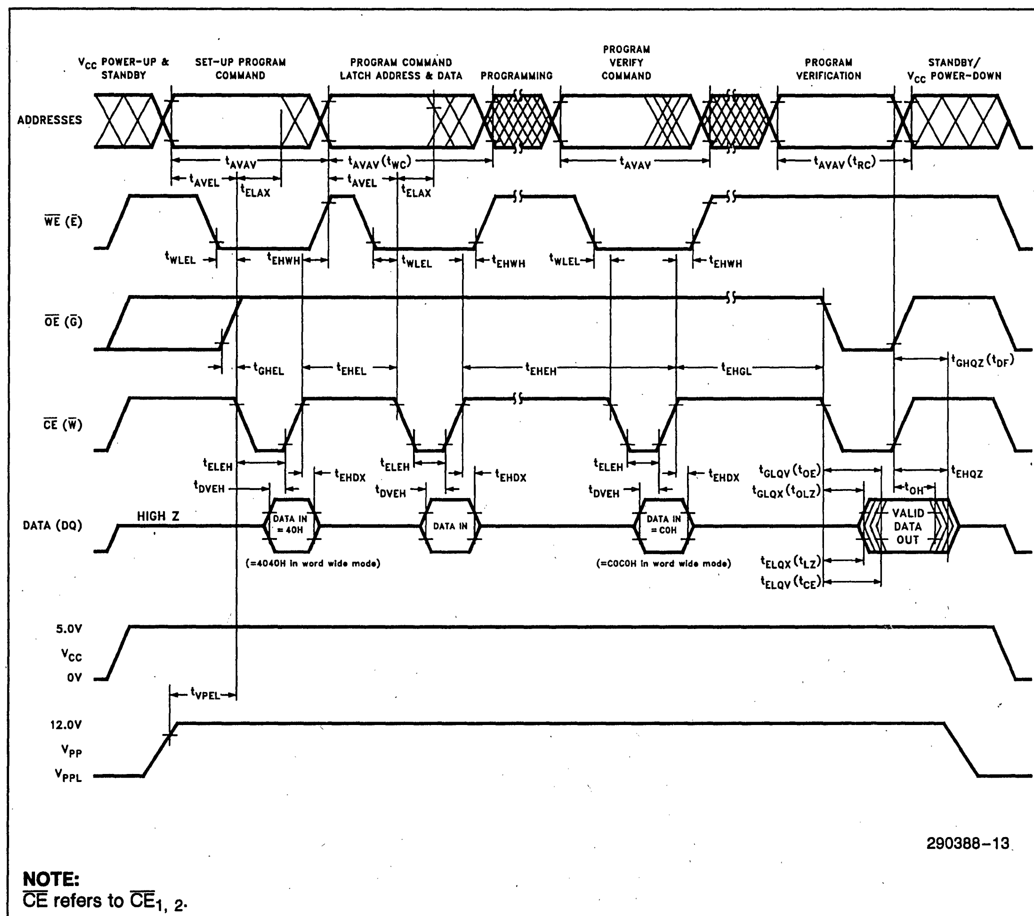
NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$.

Figure 12. AC Waveforms for Erase Operations

ALTERNATIVE $\overline{CE}$-CONTROLLED WRITES

Symbol	Characteristic	Notes	Min	Max	Unit
t_{AVAV}	Write Cycle Time		200		ns
t_{AVEL}	Address Set-up Time		0		ns
t_{ELAX}	Address Hold Time		100		ns
t_{DVEH}	Data Set-up Time		80		ns
t_{EHDX}	Data Hold Time		30		ns
t_{EHGL}	Write Recovery Time before Read		6		µs
t_{GHEL}	Read Recovery Time before Write		0		µs
t_{WLEL}	Write Enable Set-Up Time before Chip-Enable		0		ns
t_{EHWH}	Write Enable Hold Time		0		ns
t_{ELEH}	Write Pulse Width	1	100		ns
t_{EHEL}	Write Pulse Width High		20		ns
t_{PEL}	V_{PP} Set-up Time to Chip Enable Low		100		ns

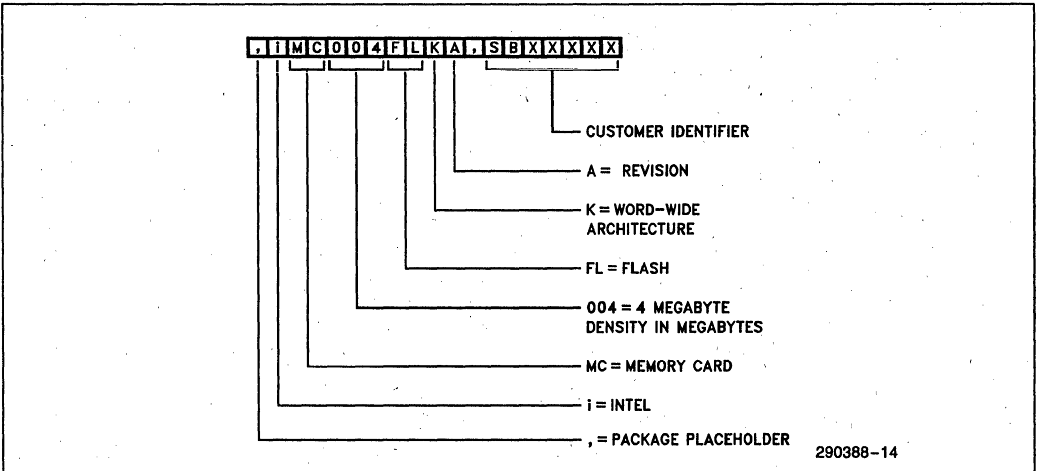
NOTES:
1. Chip Enable Controlled Writes: Write operations are driven by the valid combination of Chip Enable and Write Enable. In systems where Chip Enable defines the write pulse width (with a longer Write Enable timing waveform) all set-up, hold and inactive Write Enable times should be measured relative to the Chip Enable waveform.



NOTE:
$\overline{CE}$ refers to $\overline{CE}_{1,2}$.

Figure 13. Alternate AC Waveforms for Write Operations

ORDERING INFORMATION



ADDITIONAL INFORMATION	ORDER NUMBER
ER-20, "ETOX II Flash Memory Technology"	294005
RR-60, "ETOX II Flash Memory Reliability Data Summary"	293002
AP-343, "Solutions for High Density Applications using Flash Memory"	292079
RR-70, "Flash Memory Card Reliability Data Summary"	293007

REVISION HISTORY

Number	Description
03	Removed PRELIMINARY Removed ExCA Compliance Section Clarified need for Valid Address during Commands Corrected $V_{PP} = V_{PPH}$ in Erase Algorithm Increased I_{CC2}–I_{CC5} D.C. Current Specifications for both Byte-Wide and Word-Wide modes. Revised and updated Application Section discussion Changed order number

SERIES 2 FLASH MEMORY CARDS iMC004FLSA, iMC010FLSA, iMC020FLSA

- **4, 10 and 20 Megabyte Capacities**
- **PCMCIA 2.0/JEIDA 4.1 68-Pin Standard**
  - **Hardwired Card Information Structure**
  - **Byte- or Word-Wide Selectable**
- **ExCA™ Compatible for System-to-System Inter-Operability**
- **Component Management Registers for Card Status/Control and Flexible System Interface**
- **Automatic Erase/Write**
  - **Monitored with Ready/Busy Output**
- **Card Power-Down Modes**
  - **Deep-Sleep for Low Power Applications**
- **Mechanical Write Protect Switch**
- **Solid-State Reliability**
- **Intel FlashFile™ Architecture**
- **High-Performance Read Access**
  - **200 ns Maximum**
- **High-Performance Random Writes**
  - **10 μs Typical Word Write**
- **Erase Suspend to Read Command**
  - **Keeps Erase as Background Task**
- **Nonvolatility (Zero Retention Power)**
  - **No Batteries Required for Back-up**
- **ETOX™ III 0.8μ Flash Memory Technology**
  - **5V Read, 12V Erase/Write**
  - **High-Volume Manufacturing Experience**

Intel's Series 2 Flash Memory Card facilitates high-performance disk emulation in mobile PCs and dedicated equipment. Manufactured with Intel's ETOX III 0.8μ, FlashFile Memory devices, the Series 2 Card allows code and data retention while erasing and/or writing other blocks. Additionally, the Series 2 Flash Memory Card features low power modes, flexible system interfacing and a 200 ns read access time. When coupled with popular low-power microprocessors, like Intel's 386SL™, these cards enable high-performance implementations of mobile computers and systems.

Series 2 Cards conform to the Personal Computer Memory Card International Association (PCMCIA 2.0)/Japanese Electronics Industry Development Association (JEIDA 4.1) 68-pin standard, providing electrical and physical compatibility. The Series 2 Flash Memory Card is also compatible with Intel's Exchangeable Card Architecture (ExCA), an open hardware and software system implementation of PCMCIA Release 2.0 that allows inter-operability from system to system, independent of manufacturer.

Data file management software, such as Microsoft's* Flash File System (FFS), provide data file storage and memory management, much like a disk operating system. Intel's Series 2 Flash Memory Cards, coupled with flash file management software, effectively provide a removable, all-silicon mass storage solution with higher performance and reliability than disk-based memory architectures.

Designing with Intel's FlashFile Architecture enables OEM system manufacturers to design and manufacture a new generation of mobile PCs and dedicated equipment where high performance, ruggedness, long battery life and lighter weight are a requirement. For large user groups in workstation environments, the Series 2 Cards provide a means to securely store user data and backup system configuration/status information.

ExCA, ETOX, FlashFile, and i386SL are trademarks of Intel Corporation.
Microsoft is a trademark of Microsoft Corporation.

October 1992
Order Number: 290434-002

Table 1. Series 2 Flash Memory Card Pinout

Pin	Signal	I/O	Function	Active
1	GND		Ground	
2	DQ_3	I/O	Data Bit 3	
3	DQ_4	I/O	Data Bit 4	
4	DQ_5	I/O	Data Bit 5	
5	DQ_6	I/O	Data Bit 6	
6	DQ_7	I/O	Data Bit 7	
7	$\overline{CE}_1$	I	Card Enable 1	LO
8	A_{10}	I	Address Bit 10	
9	$\overline{OE}$	I	Output Enable	LO
10	A_{11}	I	Address Bit 11	
11	A_9	I	Address Bit 9	
12	A_8	I	Address Bit 8	
13	A_{13}	I	Address Bit 13	
14	A_{14}	I	Address Bit 14	
15	$\overline{WE}$	I	Write Enable	LO
16	$RDY/\overline{BSY}$		Ready-Busy	HI/LO
17	V_{CC}		Supply Voltage	
18	V_{PP1}		Supply Voltage	
19	A_{16}	I	Address Bit 16	
20	A_{15}	I	Address Bit 15	
21	A_{12}	I	Address Bit 12	
22	A_7	I	Address Bit 7	
23	A_6	I	Address Bit 6	
24	A_5	I	Address Bit 5	
25	A_4	I	Address Bit 4	
26	A_3	I	Address Bit 3	
27	A_2	I	Address Bit 2	
28	A_1	I	Address Bit 1	
29	A_0	I	Address Bit 0	
30	DQ_0	I/O	Data Bit 0	
31	DQ_1	I/O	Data Bit 1	
32	DQ_2	I/O	Data Bit 2	
33	WP	O	Write Protect	HI
34	GND		Ground	
35	GND		Ground	
36	$\overline{CD}_1$	O	Card Detect 1	LO
37	DQ_{11}	I/O	Data Bit 11	
38	DQ_{12}	I/O	Data Bit 12	
39	DQ_{13}	I/O	Data Bit 13	
40	DQ_{14}	I/O	Data Bit 14	
41	DQ_{15}	I/O	Data Bit 15	
42	$\overline{CE}_2$	I	Card Enable 2	LO
43	NC			
44	RFU		Reserved	
45	RFU		Reserved	
46	A_{17}	I	Address Bit 17	
47	A_{18}	I	Address Bit 18	
48	A_{19}	I	Address Bit 19	
49	A_{20}	I	Address Bit 20	
50	A_{21}	I	Address Bit 21	
51	V_{CC}		Supply Voltage	
52	V_{PP2}		Supply Voltage	
53	A_{22}	I	Address Bit 22	
54	A_{23}	I	Address Bit 23	
55	A_{24}	I	Address Bit 24	
56	A_{25}		No Connect	
57	RFU		Reserved	
58	RST	I	Reset	HI
59	WAIT	O	Extend Bus Cycle	LO
60	RFU		Reserved	
61	$\overline{REG}$	I	Register Select	LO
62	BVD_2	O	Batt. Volt Det 2	
63	BVD_1	O	Batt. Volt Det 1	
64	DQ_8	I/O	Data Bit 8	
65	DQ_9	I/O	Data Bit 9	
66	DQ_{10}	I/O	Data Bit 10	
67	$\overline{CD}_2$	O	Card Detect 2	LO
68	GND		Ground	

Table 2. Series 2 Flash Memory Card Pin Descriptions

Symbol	Type	Name and Function
A_0–A_{25}	I	**ADDRESS INPUTS:** A_0 through A_{25} are address bus lines which enable direct addressing of 64 megabytes of memory on a card. A_0 is not used in word access mode. A_{24} is the most significant address bit. Note: A_{25} is a no-connect but should be provided on host side.
DQ_0–DQ_{15}	I/O	**DATA INPUT/OUTPUT:** DQ_0 through DQ_{15} constitute the bidirectional data bus. DQ_{15} is the most significant bit.
$\overline{CE}_1$, $\overline{CE}_2$	I	**CARD ENABLE 1, 2:** $\overline{CE}_1$ enables even bytes, $\overline{CE}_2$ enables odd bytes. Multiplexing A_0, $\overline{CE}_1$ and $\overline{CE}_2$ allows 8-bit hosts to access all data on DQ_0 through DQ7. (See Table 3 for a more detailed description.)
$\overline{OE}$	I	**OUTPUT ENABLE:** Active low signal gating read data from the memory card.
$\overline{WE}$	I	**WRITE ENABLE:** Active low signal gating write data to the memory card.
RDY/$\overline{BSY}$	O	**READY/BUSY OUTPUT:** Indicates status of internally timed erase or write activities. A high output indicates the memory card is ready to accept accesses. A low output indicates that a device(s) in the memory card is(are) busy with internally timed activities. See text for an alternate function (READY-BUSY MODE REGISTER).
$\overline{CD}_1$ & $\overline{CD}_2$	O	**CARD DETECT 1, 2:** These signals provide for correct card insertion detection. They are positioned at opposite ends of the card to detect proper alignment. The signals are connected to ground internally on the memory card and will be forced low whenever a card is placed in the socket. The host socket interface circuitry shall supply 10K or larger pull-up resistors on these signal pins.
WP	O	**WRITE PROTECT:** Write Protect reflects the status of the Write-Protect switch on the memory card. WP set high = write protected, providing internal hardware write lockout to the flash array.
V_{PP1}, V_{PP2}		**WRITE/ERASE POWER SUPPLY:** (12V nominal) for erasing memory array blocks or writing data in the array. They must be 12V to perform an erase/write operation. V_{PP1} supplies even byte Erase/Write voltage and V_{PP2} supplies the odd byte Erase/Write voltage.
V_{CC}		**CARD POWER SUPPLY** (5V nominal) for all internal circuitry.
GND	I	**GROUND** for all internal circuitry.
$\overline{REG}$	I	**REGISTER SELECT** provides access to Series 2 Flash Memory Card registers and Card Information Structure in the Attribute Memory Plane.
RST	I	**RESET** from system, active high. Places card in Power-On Default State. RESET pulse width must be ≥ 200 ns.
$\overline{WAIT}$	O	**WAIT (Extend Bus Cycle)** is used by Intel's I/O cards and is driven high.
BVD_1, BVD_2	O	**BATTERY VOLTAGE DETECT:** Upon completion of the power on reset cycle, these signals are driven high to maintain SRAM-card compatibility.
RFU		**RESERVED FOR FUTURE USE**
NC		**NO INTERNAL CONNECTION.** Pin may be driven or left floating.

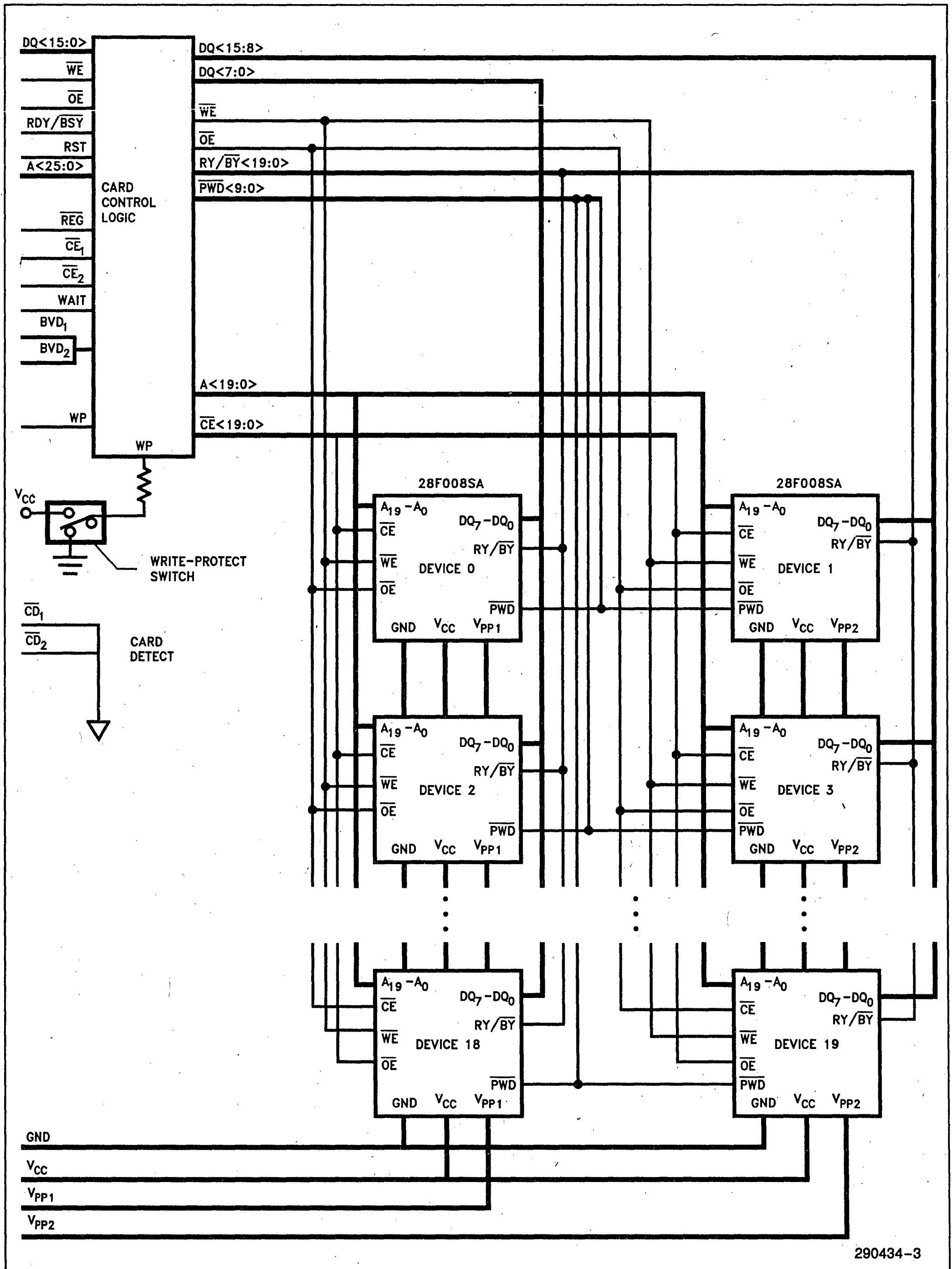


Figure 1. Detailed Block Diagram. The Card Control Logic Provides Decoding Buffering and Control Signals.

APPLICATIONS

Intel's second generation Series 2 Flash Memory Cards facilitate high performance disk emulation for the storage of data files and application programs on a purely solid-state removable medium. File management software, such as Microsoft's Flash File System, in conjunction with the Series 2 Flash Memory Cards enable the design of high-performance light-weight notebook, palmtop, and pen-based PCs that have the processing power of today's desktop computers.

Application software stored on the flash memory card substantially reduces the slow disk-to-DRAM download process. Replacing the mechanical disk results in a dramatic enhancement of read performance and substantial reduction of power consumption, size and weight—considerations particularly important in portable PCs and equipment. The Series 2 Card's high performance read access time allows the use of Series 2 Cards in an "execute-in-place" (XIP) architecture. XIP eliminates redundancy associated with DRAM/Disk memory system architectures. Operating systems stored in Flash Memory decreases system boot or program load times, enabling the design of PCs that boot, operate, store data files and execute application programs from/to nonvolatile memory without losing the ability to perform an update.

File management systems modify and store data files by allocating flash memory space intelligently. Wear leveling algorithms, employed to equally distribute the number of rewrite cycles, ensure that no particular block is cycled excessively relative to other blocks. This provides hundreds of thousands of hours of power on usage.

This file management software enables the user to interact with the flash memory card in precisely the same way as a magnetic disk.

For example, the Microsoft Flash File System enables the storage and modification of data files by utilizing a linked-list directory structure that is evenly distributed along with the data throughout the memory array. The linked-list approach minimizes file fragmentation losses by using variable-sized data structures rather than the standard sector/cluster method of disk-based systems.

Implementation of Intel's Exchangeable Card Architecture (ExCA) enables the user to transport files and application programs between portable and desktop PCs via memory card Reader/Writers. Series 2 Flash Memory Cards provide durable nonvolatile memory storage for mobile PCs on the road, facilitating simple transfer back into the desktop environment.

For systems currently using a static RAM/battery configuration for data acquisition, the Series 2 Flash Memory Card's nonvolatility eliminates the need for battery backup. The concern for battery failure no longer exists, an important consideration for portable computers and medical instruments, both requiring continuous operation. Series 2 Cards consume no power when the system is off, and only 5 μA in Deep-Sleep mode (20 Megabyte card). Furthermore, Flash Memory Cards offer a considerable cost and density advantage over memory cards based on static RAM with battery backup.

Besides disk emulation, the Series 2 Card's electrical block-erasure, data writability, and inherent nonvolatility fit well with data accumulation and recording needs. Electrical block-erasure provides design flexibility to selectively rewrite blocks of data, while saving other blocks for infrequently updated parameters and lookup tables. For example, networks and systems that utilize large banks of battery-backed DRAM to store configuration and status benefit from the Series 2 Flash Card's nonvolatility and reliability.

SERIES 2 ARCHITECTURE OVERVIEW

The Series 2 Flash Memory Card contains a 2 to 20 Megabyte Flash Memory array consisting of 2 to 20 28F008SA FlashFile Memory devices. Each 28F008SA contains sixteen individually-erasable, 64 Kbyte blocks; therefore, the Flash Memory Card contains from 32 to 320 device blocks. It also contains two Card Control Logic devices that manage the external interface, address decoding, and component management logic. (Refer to Figure 1 for a block diagram.)

To support PCMCIA-compatible word-wide access, devices are paired so that each accessible memory block is 64 KWords (see Figure 2). Card logic allows the system to write or read one word at a time, or one byte at a time by referencing the high or low byte. Erasure can be performed on the entire block pair (high and low device blocks simultaneously), or on the high or low byte portion separately.

Also in accordance with PCMCIA specifications this product supports byte-wide operation, in which the flash array is divided into 128K x 8 bit device blocks. In this configuration, odd bytes are multiplexed onto the low byte data bus.

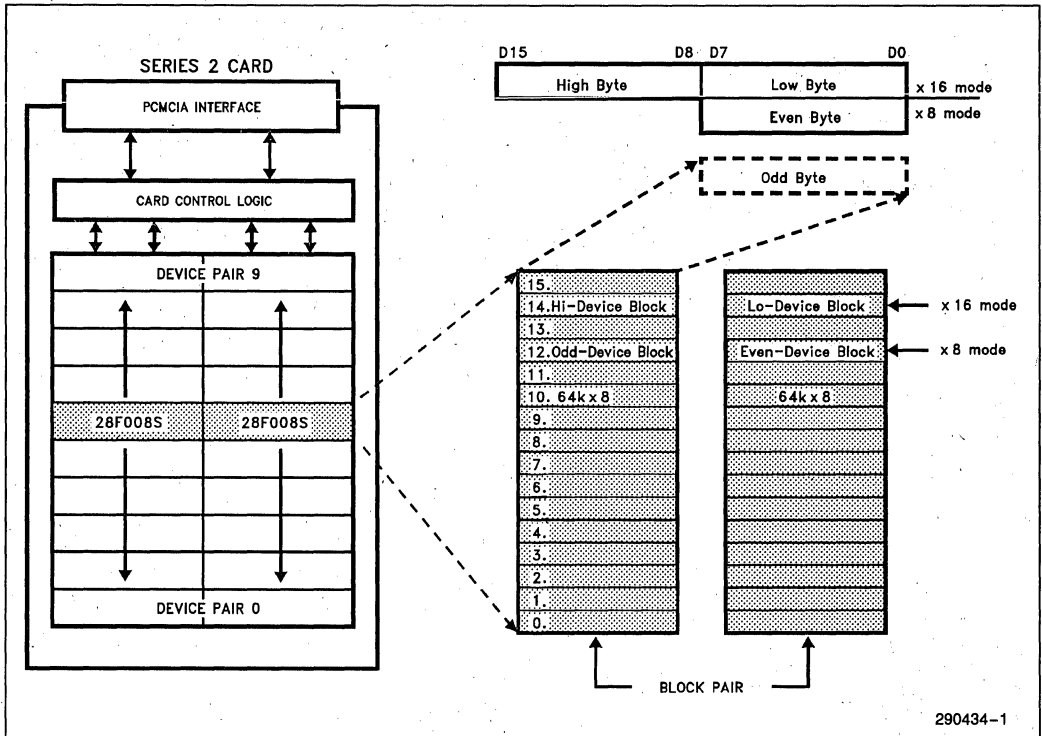


Figure 2. Memory Architecture. Each Device Pair Consists of Sixteen 64 KWord Blocks.

Series 2 Flash Memory Cards offer additional features over the Bulk Erase Flash Card product family (refer to iMC001FLKA, iMC002FLKA and iMC004FLKA data sheets). Some of the more notable enhancements include: high density capability, erase blocking, internal write/erase automation, erase suspension to read, Component Management Registers that provide software control of device-level functions and a deep-sleep mode.

Erase blocking facilitates solid-state storage applications by allowing selective memory reclamation. Multiple 64 Kbyte blocks may be simultaneously erased within the memory card as long as not more than one block per device is erasing. This shortens the total time required for erasure, but requires additional supply current. A block typically requires 1.6 seconds to erase. Each memory block can be erased and completely written 100,000 times.

Erase suspend allows the system to temporarily interrupt a block erase operation. This mode permits reads from alternate device blocks while that same device contains an erasing block. Upon completion of the read operation, erasure of the suspended block must be resumed.

Write/erase automation simplifies the system software interface to the card. A two-step command sequence initiates write or erase operations and provides additional data security. Internal device circuits automatically execute the algorithms and timings necessary for data-write or block-erase operations, including verifications for long-term data integrity. While performing either data-write or block-erase, the memory card interface reflects this by bringing its RDY/$\overline{BSY}$ (Ready/Busy) pin low. This output goes high when the operation completes. This feature reduces CPU overhead and allows software polling or hardware interrupt mechanisms. Writing memory data is achieved in single byte or word increments, typically in 10 μs.

Read access time is 200 ns or less over the 0°C to 60°C temperature range.

The deep-sleep mode reduces power consumption to 5 μA to help extend battery life of portable host systems. Activated through software control, this mode optionally affects the entire flash array (Global PowerDown Register) or specific device pairs (Sleep Control Register).

PCMCIA/JEIDA INTERFACE

The Series 2 Flash Memory Card interface supports the PCMCIA 2.0 and JEIDA 4.1 68-pin card format (see Tables 1 and 2). Detailed specifications are described in the PC Card Standard, Release 2.0, September 1991, published by PCMCIA. The Series 2 Card conforms to the requirements of both Release 1 and Release 2 of the PC Card Standard.

Series 2 Card pin definitions are equivalent to the Bulk-Erase Flash Card except that certain No Connects are now used. A_{22} through A_{24}, RST (Reset), and RDY/$\overline{BSY}$ (Ready/Busy) have pin assignments as set by the PCMCIA standard.

NOTE: The READY/BUSY signal is abbreviated as RDY/$\overline{BSY}$ by PCMCIA (card level) and as RY/$\overline{BY}$ by JEDEC (component level).

The outer shell of the Series 2 card meets all PCMCIA/JEIDA Type 1 mechanical specifications. See Figure 19 for mechanical dimensions.

WRITE PROTECT

A mechanical write protect switch provides the card's memory array with internal write lockout. The Write-Protect (WP) output pin reflects the status of this mechanical switch. It outputs a high signal (V_{OH}) when writes are disabled. This switch does not lock out writes to the Component Management Registers.

BATTERY VOLTAGE DETECT

PCMCIA requires two signals, BVD_1 and BVD_2, be supplied at the interface to reflect card battery condition. Flash Memory Cards do not require batteries. When the power on reset cycle is complete, BVD_1 and BVD_2 are driven high to maintain compatibility.

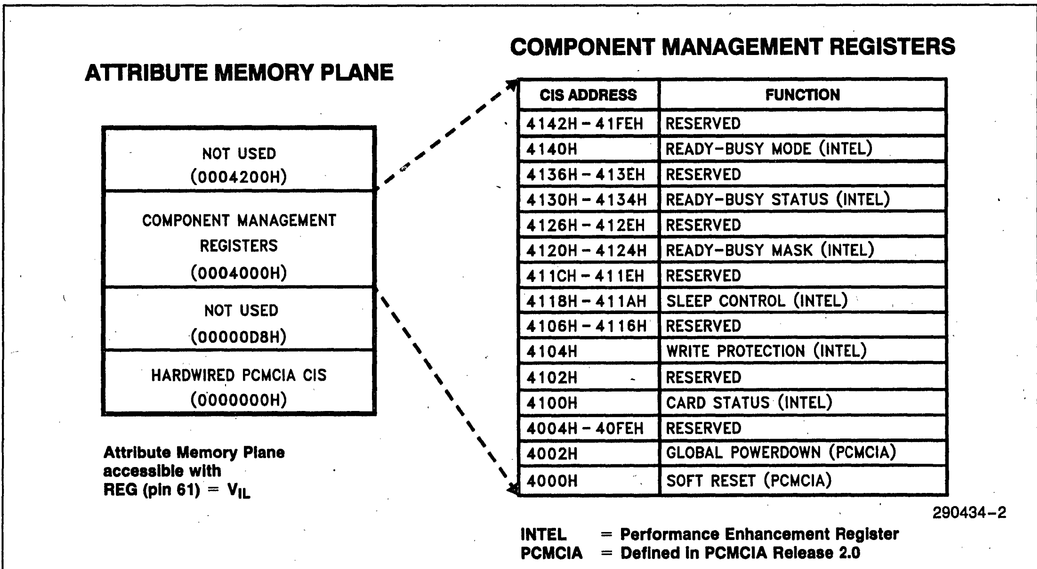
CARD DETECT

Two signals, $\overline{CD_1}$ and $\overline{CD_2}$, allow the host to determine proper socket seating. They reside at opposite ends of the connector and are tied to ground within the memory card.

DESIGN CONSIDERATIONS

The Series 2 Card consists of two separate memory planes: the Common Memory Plane (or Main Memory) and the Attribute Memory Plane. The Common Memory Plane resides in the banks of device pairs and represents the user-alterable memory space.

The Component Management Registers (CMR) and the hardwired Card Information Structure (CIS) reside in the Attribute Memory Plane within the Card Control Logic, as shown in Figure 3. The Card Control Logic interfaces the PCMCIA connector and the internal flash memory array and performs address decoding and data control.



CIS ADDRESS	FUNCTION
4142H – 41FEH	RESERVED
4140H	READY-BUSY MODE (INTEL)
4136H – 413EH	RESERVED
4130H – 4134H	READY-BUSY STATUS (INTEL)
4126H – 412EH	RESERVED
4120H – 4124H	READY-BUSY MASK (INTEL)
411CH – 411EH	RESERVED
4118H – 411AH	SLEEP CONTROL (INTEL)
4106H – 4116H	RESERVED
4104H	WRITE PROTECTION (INTEL)
4102H	RESERVED
4100H	CARD STATUS (INTEL)
4004H – 40FEH	RESERVED
4002H	GLOBAL POWERDOWN (PCMCIA)
4000H	SOFT RESET (PCMCIA)

Figure 3. Component Management Registers Allow S/W Control of Components within Card

ADDRESS DECODE

Address decoding provides the decoding logic for the 2 to 20 Device Chip Enables and the elements of the Attribute Memory Plane. $\overline{REG}$ selects between the Common Memory Plane ($\overline{REG}$ = V_{IH}) and the Attribute Memory Plane ($\overline{REG}$ = V_{IL}).

NOTE:
The Series 2 Card has *active* address inputs A_0 to A_{24} implying that reading and writing to addresses beyond 32 Megabytes causes wraparound. Furthermore, reads to illegal addresses (for example, between 20 and 32 Meg on a 20 Megabyte card) returns 0FFFFh data.

The 28F008SA devices, storing data, applications or firmware, form the Common Memory Plane accessed individually or as device pairs. Memory is linearly mapped in the Common Memory Plane. Three memory access modes are available when accessing the Common Memory Plane: Byte-Wide, Word Wide, and Odd-Byte modes.

Additional decoding selects the hardwired PCMCIA CIS and Component Management Registers mapped in the Attribute Memory Plane beginning at address 000000H.

The 512 memory-mapped even-byte CMRs are linearly mapped beginning at address 4000H in the Attribute Memory Plane.

DATA CONTROL

Data Control Logic selects the path and direction for accessing the Common or Attribute Memory Plane. It controls any of the PCMCIA-defined Word-Wide, Byte-Wide or Odd-Byte modes for either reads or writes to these areas. As shown in Table 3, input pins which determine these selections are $\overline{REG}$, A_0 through A_{24}, $\overline{WE}$, $\overline{OE}$, $\overline{CE}_1$, and $\overline{CE}_2$. PCMCIA specifications allow only even-byte access to the Attribute Memory Plane.

In Byte-Wide mode, bytes contiguous in software actually alternate between two device blocks of a device pair. Therefore, erasure of one device block erases every other contiguous byte. In accordance with the PCMCIA standard for memory configuration, the Series 2 Card does not support confining contiguous bytes within one flash device when in by-8 mode.

Table 3. Data Access Mode Truth Table

Function Mode	$\overline{REG}$	$\overline{CE}_2$	$\overline{CE}_1$	A_0	$\overline{OE}$	$\overline{WE}$	V_{PP2}	V_{PP1}	D_{15}–D_8	D_7–D_0
COMMON MEMORY PLANE										
STANDBY(1)	X	H	H	X	X	X	V_{PPL}(2)	V_{PPL}(2)	HIGH-Z	HIGH-Z
BYTE READ	H	H	L	L	L	H	V_{PPL}(2)	V_{PPL}(2)	HIGH-Z	EVEN-BYTE
	H	H	L	H	L	H	V_{PPL}(2)	V_{PPL}(2)	HIGH-Z	ODD-BYTE
WORD READ	H	L	L	X	L	H	V_{PPL}(2)	V_{PPL}(2)	ODD-BYTE	EVEN-BYTE
ODD-BYTE READ	H	L	H	X	L	H	V_{PPL}(2)	V_{PPL}(2)	ODD-BYTE	HIGH-Z
BYTE WRITE	H	H	L	L	H	L	V_{PPH}	V_{PPH}	X	EVEN-BYTE
	H	H	L	H	H	L	V_{PPH}	V_{PPH}	X	ODD-BYTE
WORD WRITE	H	L	L	X	H	L	V_{PPH}	V_{PPH}	ODD-BYTE	EVEN-BYTE
ODD-BYTE WRITE	H	L	H	X	H	L	V_{PPH}	V_{PPL}(2)	ODD-BYTE	X
ATTRIBUTE MEMORY PLANE										
BYTE READ	L	H	L	L	L	H	X(2)	X(2)	HIGH-Z	EVEN-BYTE
	L	H	L	H	L	H	X(2)	X(2)	HIGH-Z	INVALID
WORD READ	L	L	L	X	L	H	X(2)	X(2)	INVALID DATA(3)	EVEN-BYTE
ODD-BYTE READ	L	L	H	X	L	H	X(2)	X(2)	INVALID DATA(3)	HIGH-Z
BYTE WRITE	L	H	L	L	H	L	X(2)	X(2)	X	EVEN-BYTE
	L	H	L	H	H	L	X(2)	X(2)	X	INVALID OPERATION(3)
WORD WRITE	L	L	L	X	H	L	X(2)	X(2)	INVALID OPERATION(3)	EVEN-BYTE
ODD-BYTE WRITE	L	L	H	X	H	L	X(2)	X(2)	INVALID OPERATION(3)	X

NOTES:
1. Standby mode is valid in Common Memory or Attribute Memory access.
2. To meet the low power specifications, $V_{PP} = V_{PPL}$; however V_{PPH} presents no reliability problems.
3. Odd-Byte data are not valid during access to the Attribute Memory Plane.
4. H = V_{IH}, L = V_{IL}, X = Don't Care.

PRINCIPLES OF OPERATION

Intel's Series 2 Flash Memory Card provides electrically-alterable, non-volatile, random-access storage. Individual 28F008SA devices utilize a Command User Interface (CUI) and Write State Machine (WSM) to simplify block-erasure and data write operations.

COMMON MEMORY ARRAY

Figure 4 shows the Common Memory Plane's organization. The first block pair (64 KWords) of Common Memory, referred to as the Common Memory Card Information Structure Block, *optionally* extends the hardwired CIS in the Attribute Memory Plane for additional card information. This may be written during initial card formatting for OEM customization. Since this CIS Block is part of Common Memory, its data can be altered. Write access to the Common Memory CIS Block is controlled by the Write Protect Control Register which may be activated by system software after power-up. Additionally, the entire Common Memory plane (minus the Common Memory CIS Block) may be software write protected. *Note that the Common Memory CIS Block is not part of the Attribute Memory Plane. Do not assert $\overline{REG}$ to access the Common Memory CIS Block.*

Address	
13FFFFFH 1200000H	Device Pair 9
1000000H	Device Pair 8
0E00000H	Device Pair 7
0C00000H	Device Pair 6
0A00000H	Device Pair 5
0800000H	Device Pair 4
0600000H	Device Pair 3
0400000H	Device Pair 2
0200000H	Device Pair 1
0020000H	Device Pair 0
0000000H	Optional CIS

Figure 4. Common Memory Plane. Use the Optional Common Memory Plane CIS for Custom Card Format Information.

HARDWIRED CIS

The card's structure description resides in the even-byte locations starting at 0000H and going to the CIS ending tuple (FNULL) within the Attribute Memory Plane. Data included in the hardwired CIS consists of tuples. Tuples are a variable-length list of data blocks describing details such as manufacturer's name, the size of each memory device and the number of flash devices within the card.

COMPONENT MANAGEMENT REGISTERS (CMRs)

The CMRs in the Attribute Memory Plane provide special, software-controlled functionality. Card Control Logic includes circuitry to access the CMRs. REG (PCMCIA, pin 61) selects the Attribute Memory Plane (and therefore the CMRs) when equal to V_{IL}.

CMRs are classified into two categories: those defined by PCMCIA R2.0 and those included by Intel (referred to as Performance Enhancement Registers) to enhance the interface between the host system and the card's flash memory array. CMRs (See Figure 3) provide seven control functions—Ready-Busy Interrupt Mode, Device Ready-Busy Status, Device Ready-Busy Mask, Deep-Sleep Control, Software-controlled Write Protection, Card Status and Soft Reset.

SOFT RESET REGISTER (PCMCIA)

(CONFIGURATION OPTION)

The SOFT RESET REGISTER (Attribute Memory Plane Address 4000H, Figure 5) is defined in the PCMCIA Release 2.0 specification as the Configuration Option Register.

Bit 7 is the soft reset bit (SRESET). Writing a 1 to this bit initiates card reset to the power-on default state (see Side Bar page 11). This bit must be cleared to use the CMRs or to access the devices.

SRESET implements in software what the reset pin implements in hardware. On power-up, the card automatically assumes default conditions. Similar to the reset pin (pin 58), this bit clears at the end of a power-on reset cycle or a system reset cycle.

Bits 0 through 6 are not used by this memory card, but power up as zeroes for PCMCIA compatibility.

SOFT RESET REGISTER

(CONFIGURATION OPTION REGISTER)

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 4	BIT 2	BIT 1	BIT 0
4000H	SRESET	PCMCIA CONFIGURATION INDEX RESETS TO ZERO ON POWER UP						

1 = RESET, CLEAR TO ACCESS CARD

Figure 5. SOFT RESET REGISTER (PCMCIA). Sets the Memory Card in the Power-On Default State.

POWER-ON DEFAULT CONDITIONS

- All Devices Powered Up In Standby Mode
- Common Memory Available For Writes
- All Device Ready/Busy Outputs Unmasked
- PCMCIA Ready/Busy Mode Enabled
- Ready/Busy Output Goes To Ready

Global PowerDown Register (PCMCIA)

(Configuration and Status)

The Global PowerDown Register (Attribute Memory Plane Address 4002H, Figure 6) is referred to as the Configuration and Status Register in the PCMCIA Release 2.0 specification.

Bit 2 (PwrDwn) controls global card power-down. Writing a 1 to this bit places each device within the card into "Deep-Sleep" mode. *Devices in Deep-Sleep are not accessible.* Recovery from power-down requires 500 ns for reads and 1 μs for writes.

The PWRDWN bit defaults to 0 on card power-up or reset. Setting or clearing this bit has no affect on the bit settings of the Sleep Control Register.

The remaining Global PowerDown Register bits are defined for Intel's family of I/O cards and are driven low for compatibility.

GLOBAL POWER-DOWN REGISTER

(CONFIGURATION AND STATUS REGISTER)

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2 (1 = POWER DOWN)	BIT 1	BIT 0
4002H	ZEROES					PWRDWN	ZEROES	

Figure 6. GLOBAL POWER-DOWN REGISTER (PCMCIA). The PWRDWN Bit Enables Power-Down of All Flash Memory Devices.

4

CARD STATUS REGISTER

(Read Only Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4100H	ADM	ADS	SRESET	CMWP	PWRDWN	CISWP	WP	RDY/$\overline{BSY}$

Figure 7. CARD STATUS REGISTER (Intel) Provides a Quick Review of the Card's Status

CARD STATUS REGISTER (INTEL)

The Read-Only, CARD STATUS REGISTER (Attribute Memory Plane Address 4100H, Figure 7) returns generalized status of the Series 2 Card and its CMRs.

Bit 0 (RDY/$\overline{BSY}$) reflects the card's RDY/$\overline{BSY}$ (Ready-Busy) output. Software polling of this bit provides data-write or block-erase operation status. A zero indicates a busy device(s) in the card.

Bit 1 (WP) reports the position of the card's Write Protection switch with 1 indicating write protected. It reports the status of the WP pin.

Bit 2 (CISWP) reflects whether the Common Memory CIS is write protected using the WRITE PROTECT REGISTER, with 1 indicating write protected.

Bit 3 (PWRDWN) reports whether the entire flash memory array is in "Deep-Sleep" (PowerDown) mode, with 1 indicating "Deep-Sleep". This bit reflects the PWRDWN bit of the GLOBAL POWER-DOWN REGISTER. Powering down *all* device pairs individually (using the Sleep Control Register), also sets this bit.

Bit 4 (CMWP) reports whether the Common Memory Plane (minus Common Memory CIS) is write protected via the WRITE PROTECT REGISTER with 1 indicating write protected.

Bit 5 (SRESET) reflects the SRESET bit of the SOFT RESET REGISTER. It reports that the card is in Soft Reset with 1 indicating reset. When this bit is zero, the flash memory array and CMRs may be accessed, otherwise clear it via the SRESET REGISTER.

Bit 6 (ADS, ANY DEVICE SLEEP) is the "ORed" value of the SLEEP CONTROL REGISTER. Powering down any device pair sets this bit.

Bit 7 (ADM, ANY DEVICE MASKED) is the "ORed" value of the READY/BUSY MASK REGISTER. Masking any device sets this bit.

WRITE PROTECTION REGISTER (INTEL)

The WRITE PROTECTION REGISTER (Attribute Memory Plane Address 4104H, Figure 8) selects whether the optional Common Memory CIS and the remaining Common Memory blocks are write protected (see Figure 4).

Enable Common Memory CIS write protection by writing a 1 to the CISWP Bit (bit 0).

Enable write protection of the remaining Common Memory blocks by writing a 1 to the CMWP Bit (bit 1).

In the power-on default state, both bits are 0, and therefore not write protected.

Reserved bits (2–7) have undefined values and should be written as zeroes for future compatibility.

WRITE PROTECTION REGISTER

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4104H	RESERVED FOR FUTURE USE						CMWP	CISWP

1 = WRITE PROTECT

Figure 8. WRITE PROTECTION REGISTER (Intel) Eliminates Accidental Data Corruption

SLEEP CONTROL REGISTER (INTEL)

Unlike the GLOBAL POWERDOWN REGISTER, which simultaneously places all flash memory devices into a Deep-Sleep mode, the SLEEP CONTROL REGISTER (Attribute Memory Plane Address 4118H–411AH, Figure 9) allows selective power-down control of individual device pairs.

Writing a 1 to a specific bit of the SLEEP CONTROL REGISTER places the corresponding device pair into the "Deep-Sleep" mode. *Devices in Deep-Sleep are not accessible.* On cards with fewer than 20 Megabytes (10 device pairs), writing a one to an absent device pair has no affect and reads back as zero.

This register contains all zeroes (i.e., not in Deep-Sleep mode) when the card powers up or after a hard or soft reset. Furthermore, the Global Power-Down Register has no affect on the contents of this register. Therefore, any bit settings of the Sleep Control Register will remain unchanged after returning from a global power down (writing a zero to the PWRDWN bit of the Global PowerDown Register).

READY-BUSY STATUS REGISTER (INTEL)

The bits in the Read-only, READY-BUSY Status Register (Attribute Memory Plane Address 4130H-4134H, Figure 10) reflect the status (READY = 1, BUSY = 0) of each device's RY/$\overline{BY}$ output. A busy condition indicates that a device is currently processing a data-write or block-erase operation.

These bits are logically "AND-ed" to form the Ready/Busy output (RDY/$\overline{BSY}$, pin 16) of the PCMCIA interface. On memory cards with fewer than 20 devices, unused Device RY/$\overline{BY}$ Status Register bits appear as ready.

SLEEP CONTROL REGISTER

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
411AH	RESERVED						DEVICES 18/19	DEVICES 16/17
4118H	DEVICES 14/15	DEVICES 12/13	DEVICES 10/11	DEVICES 8/9	DEVICES 6/7	DEVICES 4/5	DEVICES 2/3	DEVICES 0/1

1 = SELECTED DEVICE PAIR IN POWER-DOWN MODE

Figure 9. SLEEP CONTROL REGISTER (Intel) Allows Specific Devices to be Put Into Power-Down Mode

READY-BUSY STATUS REGISTER

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4134H	RESERVED				DEVICE 19	DEVICE 18	DEVICE 17	DEVICE 16
4132H	DEVICE 15	DEVICE 14	DEVICE 13	DEVICE 12	DEVICE 11	DEVICE 10	DEVICE 9	DEVICE 8
4130H	DEVICE 7	DEVICE 6	DEVICE 5	DEVICE 4	DEVICE 3	DEVICE 2	DEVICE 1	DEVICE 0

1 = DEVICE READY, 0 = DEVICE BUSY

Figure 10. READY-BUSY STATUS REGISTER (Intel) Provides Operation Status of All Flash Memory Devices

READY-BUSY MASK REGISTER (INTEL)

The bits of the Read/Write READY-BUSY MASK REGISTER (Attribute Memory Plane Address 4120H–4124H, Figure 11) mask out the corresponding "AND-ed" READY-BUSY STATUS REGISTER bits from the PCMCIA data bus ($RDY/\overline{BSY}$, pin 16) and the CARD STATUS REGISTER $RDY/\overline{BSY}$ Bit (bit 0).

In an unmasked condition (MASK REGISTER bits = 0), any device $RY/\overline{BY}$ output going low pulls the card's $RDY/\overline{BSY}$ output to V_{IL} (BUSY). In this case, all devices must be READY to allow the card's $RDY/\overline{BSY}$ output to be ready (V_{IH}). *This is referred to as the PCMCIA READY-BUSY MODE. An alternate type of READY-BUSY function is described in the next section, READY-BUSY MODE REGISTER.*

READY-BUSY MASK

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4124H	RESERVED				DEVICE 19	DEVICE 18	DEVICE 17	DEVICE 16
4122H	DEVICE 15	DEVICE 14	DEVICE 13	DEVICE 12	DEVICE 11	DEVICE 10	DEVICE 9	DEVICE 8
4120H	DEVICE 7	DEVICE 6	DEVICE 5	DEVICE 4	DEVICE 3	DEVICE 2	DEVICE 1	DEVICE 0

1 = MASK ENABLED

Figure 11. READY-BUSY MASK REGISTER (Intel) Essential for Write Optimization

If the READY-BUSY MASK REGISTER bits are set to ones (masked condition), the RDY/$\overline{\text{BSY}}$ output and the CARD STATUS REGISTER RDY/$\overline{\text{BSY}}$ bit will reflect a READY condition regardless of the state of the corresponding devices. The READY-BUSY MASK REGISTER does not affect the READY-BUSY STATUS REGISTER allowing software polling to determine operation status.

Unmasked is the default condition for the bits in this register. On memory cards with fewer than 20 devices, unused device mask bits appear as masked.

READY-BUSY MODE REGISTER (INTEL)

The READY-BUSY MODE REGISTER (Attribute Memory Plane Address 4140H, Figure 12) provides the selection of two types of system interfacing for the busy-to-ready transition of the card's RDY/$\overline{\text{BSY}}$ pin:

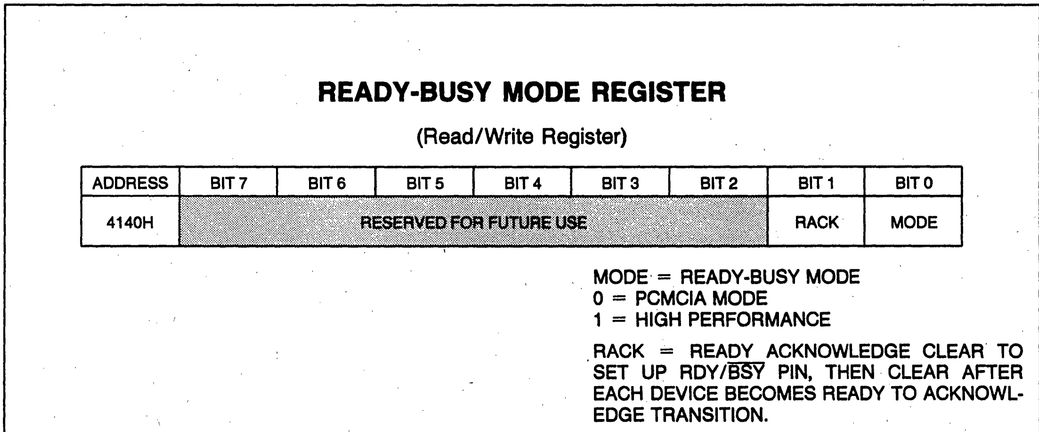
1. The standard PCMCIA READY-BUSY MODE, in which the card's RDY/$\overline{\text{BSY}}$ signal generates a low-to-high transition (from busy to ready) only after *all* busy devices (not including masked devices) have completed their data-write or block-erase operations. This may result in a long interrupt latency.
2. A High-Performance mode that generates a low-to-high (from busy-to-ready) transition after each device becomes ready. This provides the host system with immediate notification that a specific device's operation has completed and that device may now be used. This is particularly useful in a file management application where a block pair, containing only deleted files, is being erased to free up space so new file data may be written.

Enabling the HIGH-PERFORMANCE READY-BUSY MODE requires a three step sequence:

1. Set all bits in the READY/BUSY MASK REGISTER. This prevents ready devices from triggering an unwanted interrupt when step 3 is performed.
2. Write 01H to the READY-BUSY MODE REGISTER. This sets the MODE bit.
3. Write 01H to the READY-BUSY MODE REGISTER. This clears the RACK bit.

The MODE and RACK bits *must* be written in the prescribed sequence, *not* simultaneously. The card's circuitry is designed purposely in this manner to prevent an initial, unwanted busy-to-ready transition. Note that in Step 2, writing to the RACK bit is a Don't Care.

When the High-Performance Mode is enabled, specific READY-BUSY MASK bits must be cleared after an operation is initiated on the respective devices. After each device becomes ready, the RDY/$\overline{\text{BSY}}$ pin makes a low-to-high transition. To catch the next device's completion of an operation, the RACK bit must be cleared.



READY-BUSY MODE REGISTER

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4140H	RESERVED FOR FUTURE USE						RACK	MODE

MODE = READY-BUSY MODE
0 = PCMCIA MODE
1 = HIGH PERFORMANCE

RACK = READY ACKNOWLEDGE CLEAR TO SET UP RDY/$\overline{\text{BSY}}$ PIN, THEN CLEAR AFTER EACH DEVICE BECOMES READY TO ACKNOWLEDGE TRANSITION.

Figure 12. High Performance Ready-Busy Mode REGISTER (Intel) Used to Trigger a Ready Interrupt for Each Device

PRINCIPLES OF DEVICE OPERATION

Individual 28F008SA devices include a Command User Interface (CUI) and a Write State Machine (WSM) to manage write and erase functions in each device block.

The CUI serves as the device's interface to the Card Control Logic by directing commands to the appropriate device circuitry (Table 4). It allows for fixed power supplies during block erasure and data writes. The CUI handles the $\overline{WE}$ interface into the device data and address latches, as well as system software requests for status while the WSM is operating.

The CUI itself does not occupy an addressable memory location. The CUI provides a latch used to store the command and address and data information needed to execute the command. Erase Setup and Erase Confirm commands require both appropriate command data and an address within the block to be erased. The Data Write Setup command requires both appropriate command data and the address of the location to be written, while the Data Write command consists of the data to be written and the address of the location to be written.

The CUI initiates flash memory writing and erasing operations only when V_{PP} is at 12V. Depending on the application, the system designer may choose to make the V_{PP} power supply switchable (available when writes and erases are required) or hardwired to V_{PPH}. When $V_{PP} = V_{PPL}$, power savings are incurred and memory contents cannot be altered. The CUI architecture provides protection from unwanted write and erase operations even when high voltage is applied to V_{PP}. Additionally, all functions are disabled whenever V_{CC} is below the write lockout voltage V_{LKO}, or when the card's Deep-Sleep modes are enabled. The WSM automates the writing and erasure of blocks within a device. This on-chip state machine controls block erase and data-write, freeing the host processor for other tasks. After receiving the Erase Setup and Erase Confirm commands from the CUI, the WSM controls block-erase. Progress is monitored via the device's status register, the card's control logic, and the RDY/$\overline{BSY}$ pin of the PCMCIA interface. Data-write is similarly controlled, after destination address and expected data are supplied.

Table 4. Device Command Set

28F008SA Command[1]	Bus Cycles Req'd	First Bus Ccyle				Second Bus Cycle			
		Operation	Addr[2]	Data		Operation	Addr[2]	Data	
				x8 Mode	x16 Mode			x8 Mode	x16 Mode
Read Array/Reset	1	Write	DA	FFH	FFFFH				
intelligent Identifer	3	Write	DA	90H	9090H	Read	IA	IID[3]	IID[3]
Read Device Status Register	2	Write	DA	70H	7070H	Read	DA	SRD[4]	SRD[4]
Clear Device Status Register	1	Write	DA	50H	5050H				
Erase Setup/Erase Confirm	2	Write	BA	20H	2020H	Write	BA	D0H	D0D0H
Erase Suspend/ Erase Resume	2	Write	DA	B0H	B0B0H	Write	DA	D0H	D0D0H
Write Setup/Write	2	Write	WA	40H	4040H	Write	WA	WD[5]	WD[5]
Alternate Write Setup/Write[6]	2	Write	WA	10H	1010H	Write	WA	WD[5]	WD[5]

NOTES:
1. Commands other than those shown above are reserved by Intel for future device implementations and should not be used.
2. DA = A device-level (or device pair) address within the card.
   BA = Address within the block of a specific device (device pair) being erased.
   WA = Address of memory location to be written.
   IA = A device-level address; 00H for manufacturer code, 01 for device code.
3. Following the intelligent identifier command, two read operations access manufacturer (89H) and device codes (A2H).
4. SRD = Data read from Device Status Register.
5. WD = Data to be written at location WA. Data is latched on the rising edge of $\overline{WE}$.
6. Either 40H or 10H are recognized by the WSM as the Write Setup command.

COMMAND DEFINITIONS

Read Array (FFH) —

Upon initial card power-up, after exit from the Deep-Sleep modes, and whenever illegal commands are given, individual devices default to the Read Array mode. This mode is also entered by writing FFH into the CUI. In this mode, microprocessor read cycles retrieve array data. Devices remain enabled for reads until the CUI receives an alternate command. Once the internal WSM has started a block-erase or data-write operation within a device, that device will not recognize the Read Array command until the WSM has completed its operation (or the Erase Suspend command is issued during erase).

Intelligent Identifier (90H) —

After executing this command, the intelligent identifier values can be read. Only address A_0 of each device is used in this mode, all other address inputs are ignored [(Manufacturer code = 89H for A_0 = 0), (Device code = A2H for A_0 = 1)]. The device will remain in this mode until the CUI receives another command.

This information is useful by system software in determining what type of flash memory device is contained within the card and allows the correct matching of device to write and erase algorithms. System software that fully utilizes the PCMCIA specification will not use the intelligent identifier mode, as this data is available within the Card Information Structure (refer to section on PCMCIA Card Information Structure).

Read Status Register (70H)

After writing this command, a device read outputs the contents of its Status Register, regardless of the address presented to that device. The contents of this register are latched on the falling edge of $\overline{OE}$, $\overline{CE}_1$ (and/or $\overline{CE}_2$), whichever occurs last in the read cycle. This prevents possible bus errors which might occur if the contents of the Status Register changed while reading its contents. $\overline{CE}_1$ (and $\overline{CE}_2$ for odd-byte or word access) or $\overline{OE}$ must be toggled with each subsequent status read, or the completion of a write or erase operation will not be evident. This command is executable while the WSM is operating, however, during a block-erase or data-write operation, reads from the device will automatically return status register data. Upon completion of that operation, the device remains in the Status Register read mode until the CUI receives another command.

The read Status Register command functions when $V_{PP} = V_{PPL}$ or V_{PPH}.

Clear Status Register (50H)

The Erase Status and Write Status bits may be set to "1"s by the WSM and can only be reset by the Clear Status Register Command. These bits indicate various failure conditions. By allowing system software to control the resetting of these bits, several operations may be performed (such as cumulatively writing several bytes or erasing multiple blocks in sequence). The device's Status Register may then be polled to determine if an error occurred during that sequence. This adds flexibility to the way the device may be used.

Additionally, the V_{PP} Status bit (SR.3) MUST be reset by system software (Clear Status Register command) before further block-erases are attempted (after an error).

The Clear Status Register command functions when $V_{PP} = V_{PPL}$ or V_{PPH}. This command puts the device in the Read Array mode.

Write Setup/Write

A two-command sequence executes a data-write operation. After the system switches V_{PP} to V_{PPH}, the write setup command (40H) is written to the CUI of the appropriate device, followed by a second write specifying the address and write data (latched on the rising edge of $\overline{WE}$). The device's WSM controls the data-write and write verify algorithms internally. After receiving the two-command write sequence, the device automatically outputs Status Register data when read (see Figure 13). The CPU detects the completion of the write operation by analyzing card-level or device-level indicators. Card-level indicators include the RDY/$\overline{BSY}$ pin and the READY-BUSY STATUS REGISTER; while device-level indicators include the specific device's Status Register. Only the Read Status Register command is valid while the write operation is active. Upon completion of the data-write sequence (see section on Status Register) the device's Status Register reflects the result of the write operation. The device remains in the Read Status Register mode until the CUI receives an alternate command.

Erase Setup/Erase Confirm Commands (20H)

Within a device, a two-command sequence initiates an erase operation on one device block at a time. After the system switches V_{PP} to V_{PPH}, an Erase Setup command (20H) prepares the CUI for the Erase Confirm command (D0H). The device's WSM controls the erase algorithms internally. After receiving the two-command erase sequence, the device automatically outputs Status Register data when read (see Figure 14). If the command after erase setup is not an Erase Confirm command, the CR sets the Write Failure and Erase Failure bits of the Status Register, places the device into the Read Status Register mode, and waits for another command. The Erase Confirm command enables the WSM for erase (simultaneously closing the address latches for that device's block (A_{16}–A_{19}). The CPU detects the completion of the erase operation by analyzing card-level or device-level indicators. Card-level indicators include the RDY/BSY pin and the READY-BUSY STATUS REGISTER; while device-level indicators include the specific device's Status Register. Only the Read Status Register and Erase Suspend command is valid during an active erase operation. Upon completion of the erase sequence (see section on Status Register) the device's Status Register reflects the result of the erase operation. The device remains in the Read Status Register mode until the CUI receives an alternate command.

The two-step block-erase sequence ensures that memory contents are not accidentally erased. Erase attempts while $V_{PPL} < V_{PP} < V_{PPH}$ produce spurious results and are not recommended. Reliable block erasure only occurs when $V_{PP} = V_{PPH}$. In the absence of this voltage, memory contents are protected against erasure. If block erase is attempted while $V_{PP} = V_{PPL}$, the V_{PP} Status bit will be set to "1".

When erase completes, the Erase Status bit should be checked. If an erase error is detected, the device's Status Register should be cleared. The CUI remains in Read Status Register mode until receiving an alternate command.

Erase Suspend (B0H)/Erase Resume (D0H)

Erase Suspend allows block erase interruption to read data from another block of the device or to temporarily conserve power for another system operation. Once the erase process starts, writing the Erase Suspend command to the CUI (see Figure 15) requests the WSM to suspend the erase sequence at a predetermined point in the erase algorithm. In the erase suspend state, the device continues to output Status Register data when read.

Polling the device's RY/$\overline{BY}$ and Erase Suspend Status bits (Status Register) will determine when the erase suspend mode is valid. It is important to note that the card's RDY/$\overline{BSY}$ pin will also transition to V_{OH} and will generate an interrupt if this pin is connected to a system-level interrupt. At this point, a Read Array command can be written to the device's CUI to read data from blocks **other than those which are suspended.** The only other valid commands at this time are Read Status Register (70H) and Erase Resume (D0H). If V_{PP} goes low during Erase Suspend, the V_{PP} Status bit is set in the Status Register and the erase operation is aborted.

The Erase Resume command clears the Erase Suspend state and allows the WSM to continue with the erase operation. The device's RY/$\overline{BY}$ Status and Erase Suspend Status bits and the card's READY-BUSY Status Register are automatically updated to reflect the erase resume condition. The card's RDY/$\overline{BSY}$ pin also returns to V_{OL}.

Invalid/Reserved

These are unassigned commands having the same effect as the Read Array command. Do not issue any command other than the valid commands specified above. Intel reserves the right to redefine these codes for future functions.

DEVICE STATUS REGISTER

Each 28F008SA device in the Series 2 Card contains a Status Register which displays the condition of its Write State Machine. The Status Register is read at any time by writing the Read Status command to the CUI. After writing this command, all subsequent Read operations output data from the Status Register, until another command is written to the CUI.

Bit 7—WSM Status

This bit reflects the Ready/Busy condition of the WSM. A "1" indicates that read, block-erase or data- write operations are available. A "0" indicates that write or erase operations are in progress.

Bit 6—Erase Suspend Status

If an Erase Suspend command is issued during the erase operation, the WSM halts execution and sets the WSM Status bit and the Erase Suspend Status bit to a "1". This bit remains set until the device receives an Erase Resume command, at which point the CUI resets the WSM Status bit and the Erase Suspend Status bit.

Bit 5—Erase Status

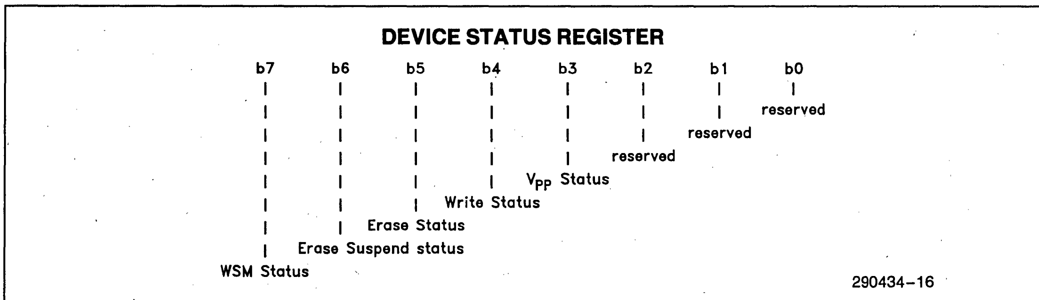
This bit will be cleared to 0 to indicate a successful block-erasure. When set to a "1", the WSM has been unsuccessful at performing an erase verification. The device's CUI only resets this bit to a "0" in response to a Clear Status Register command.

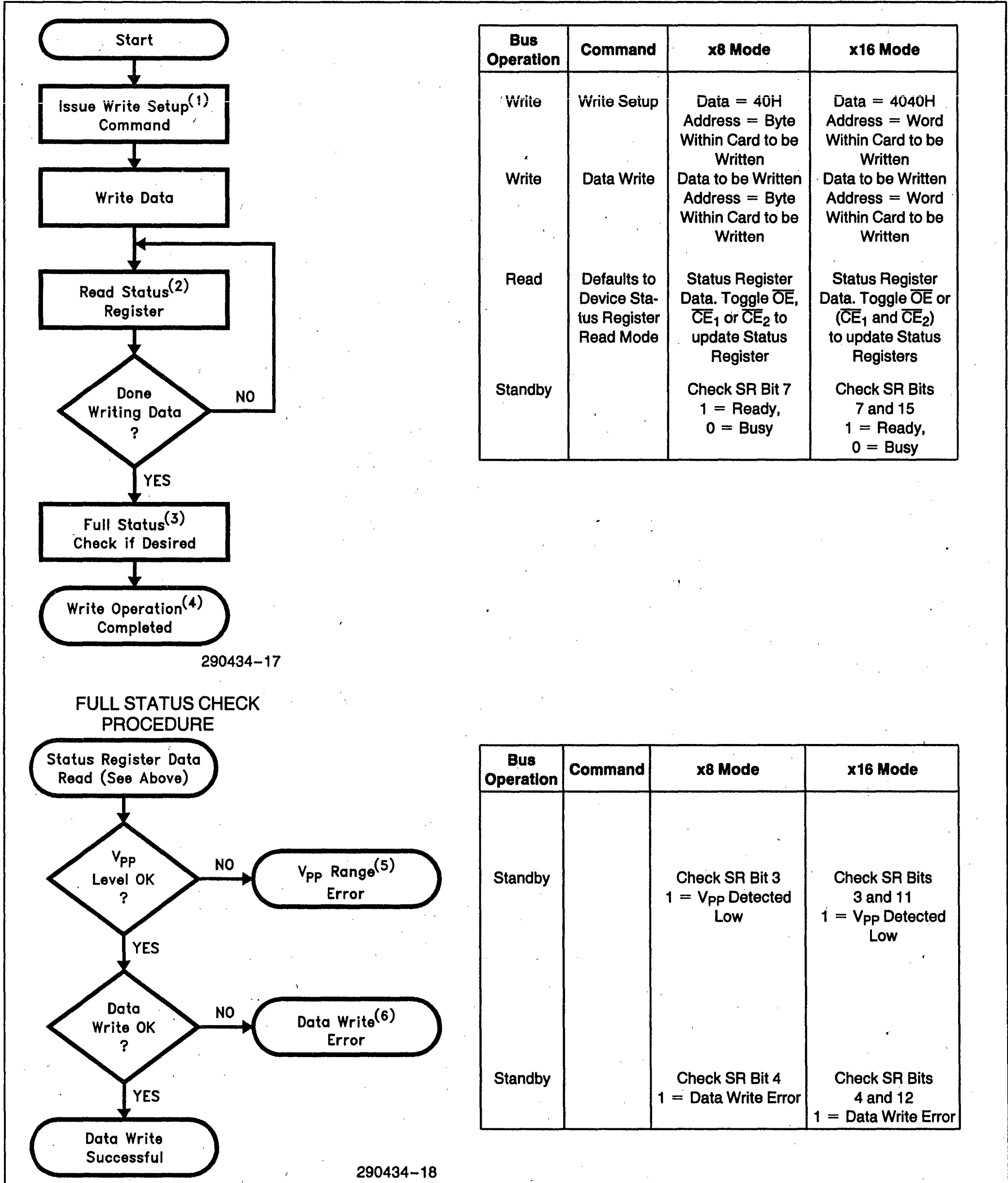
Bit 4—Write Status

This bit will be cleared to a 0 to indicate a successful data-write operation. When the WSM fails to write data after receiving a write command, the bit is set to a "1" and can only be reset by the CUI in response to a Clear Status Register command.

Bit 3—V_{PP} Status

During block-erase and data-write operations, the WSM monitors the output of the device's internal V_{PP} detector. In the event of low V_{PP}, the WSM sets ("1") the V_{PP} Status bit, the status bit for the operation in progress (either write or erase). The CUI resets these bits in response to a Clear Status Register command. Also, the WSM RY/$\overline{BY}$ bit will be set to indicate a device ready condition. This bit MUST be reset by system software (Clear Status Register command) before further data writes or block erases are attempted.





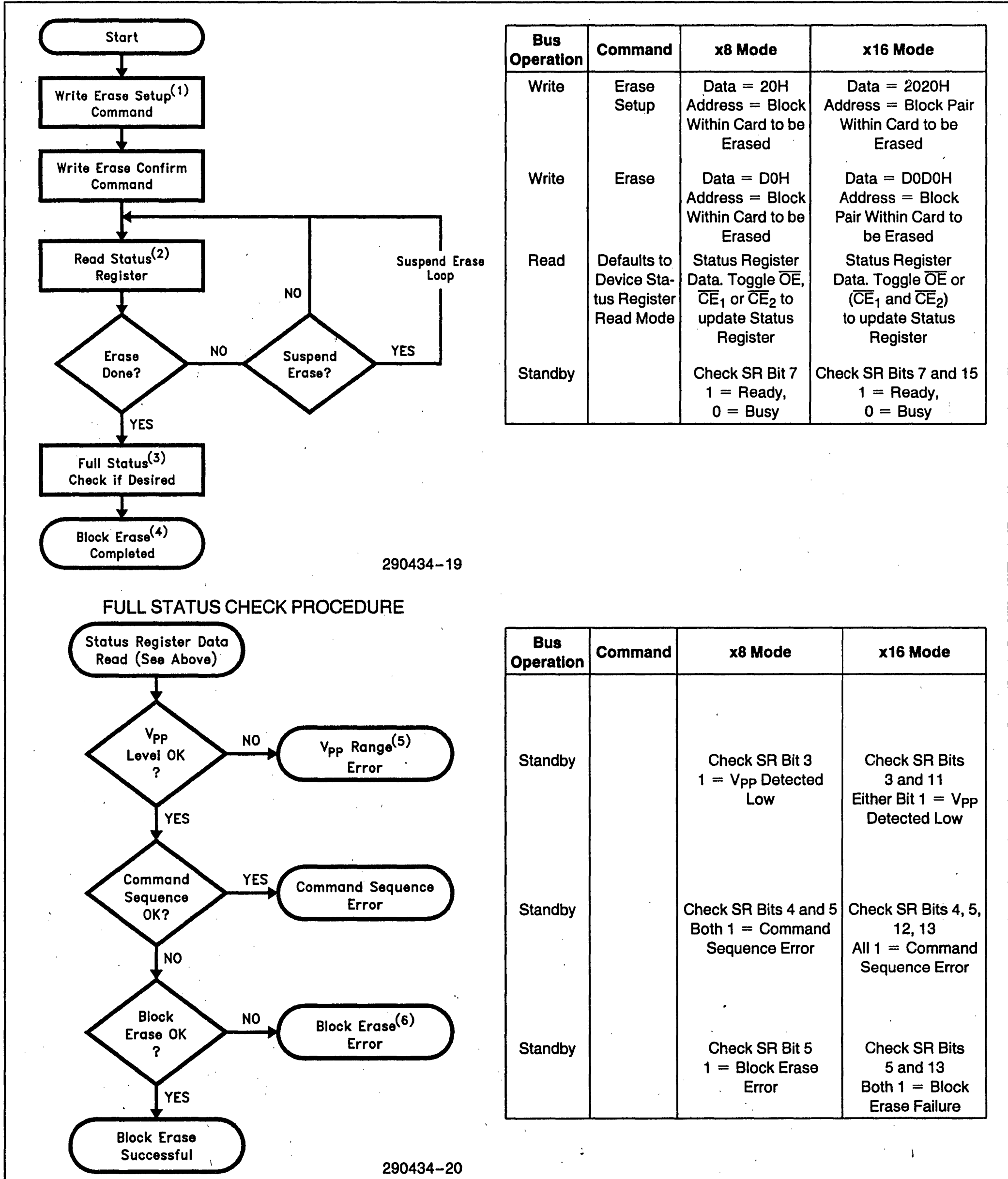
Bus Operation	Command	x8 Mode	x16 Mode
Write	Write Setup	Data = 40H Address = Byte Within Card to be Written	Data = 4040H Address = Word Within Card to be Written
Write	Data Write	Data to be Written Address = Byte Within Card to be Written	Data to be Written Address = Word Within Card to be Written
Read	Defaults to Device Status Register Read Mode	Status Register Data. Toggle $\overline{OE}$, $\overline{CE}_1$ or $\overline{CE}_2$ to update Status Register	Status Register Data. Toggle $\overline{OE}$ or ($\overline{CE}_1$ and $\overline{CE}_2$) to update Status Registers
Standby		Check SR Bit 7 1 = Ready, 0 = Busy	Check SR Bits 7 and 15 1 = Ready, 0 = Busy

Bus Operation	Command	x8 Mode	x16 Mode
Standby		Check SR Bit 3 1 = V_{PP} Detected Low	Check SR Bits 3 and 11 1 = V_{PP} Detected Low
Standby		Check SR Bit 4 1 = Data Write Error	Check SR Bits 4 and 12 1 = Data Write Error

Figure 13. Device-Level Automated Write Algorithm

NOTES:

1. Repeat for subsequent data writes.
2. In addition, the card's READY-BUSY STATUS REGISTER or the RDY/$\overline{BSY}$ pin may be used.
3. Full device-level status check can be done after each data write or after a sequence of data writes.
4. Write FFH (or FFFFH) after the last data write operation to reset the device(s) to Read Array Mode.
5. If a data write operation fails due to a low V_{PP} (setting SR Bit 3), the Clear Status Register command MUST be issued before further attempts are allowed by the Write State Machine.
6. If a data write operation fails during a multiple write sequence, SR Bit 4 (Write Status) will not be cleared until the Command User Interface receives the Clear Status Register command.



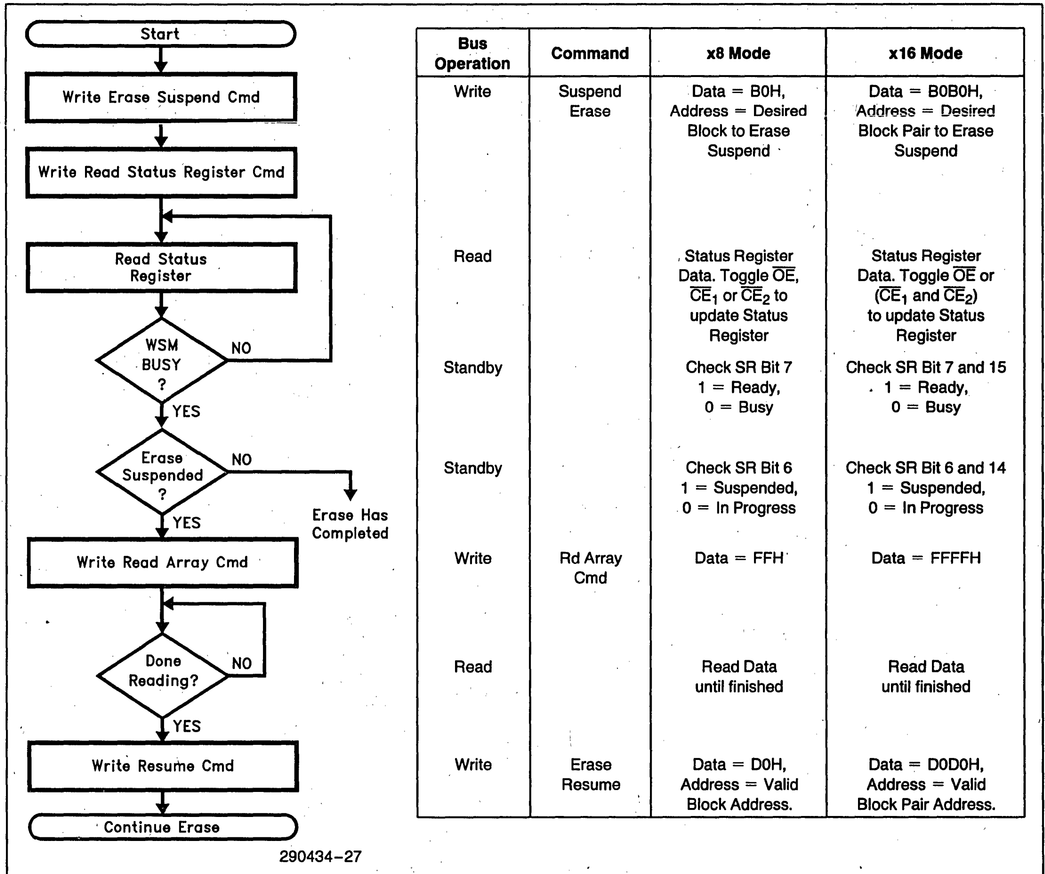
Bus Operation	Command	x8 Mode	x16 Mode
Write	Erase Setup	Data = 20H Address = Block Within Card to be Erased	Data = 2020H Address = Block Pair Within Card to be Erased
Write	Erase	Data = D0H Address = Block Within Card to be Erased	Data = D0D0H Address = Block Pair Within Card to be Erased
Read	Defaults to Device Status Register Read Mode	Status Register Data. Toggle $\overline{OE}$, $\overline{CE}_1$ or $\overline{CE}_2$ to update Status Register	Status Register Data. Toggle $\overline{OE}$ or ($\overline{CE}_1$ and $\overline{CE}_2$) to update Status Register
Standby		Check SR Bit 7 1 = Ready, 0 = Busy	Check SR Bits 7 and 15 1 = Ready, 0 = Busy

Bus Operation	Command	x8 Mode	x16 Mode
Standby		Check SR Bit 3 1 = V_{PP} Detected Low	Check SR Bits 3 and 11 Either Bit 1 = V_{PP} Detected Low
Standby		Check SR Bits 4 and 5 Both 1 = Command Sequence Error	Check SR Bits 4, 5, 12, 13 All 1 = Command Sequence Error
Standby		Check SR Bit 5 1 = Block Erase Error	Check SR Bits 5 and 13 Both 1 = Block Erase Failure

Figure 14. Device-Level Automated Erase Algorithm

NOTES:

1. Repeat for subsequent data writes.
2. In addition, the card's READY-BUSY STATUS REGISTER or the RDY/$\overline{BSY}$ pin may be used.
3. Full device-level status check can be done after each block erase or after a sequence of block erases.
4. Write FFH (or FFFFH) after the last block erase operation to reset the device(s) to Ready Array Mode.
5. If a block erase operation fails due to a low V_{PP} (setting SR Bit 3), the Clear Status Register command MUST be issued before further attempts are allowed by the Write State Machine.
6. If a block erase operation fails during a multiple block erase sequence, SR Bit 4 (Write Status) will not be cleared until the Command User Interface receives the Clear Status Register command.



Bus Operation	Command	x8 Mode	x16 Mode
Write	Suspend Erase	Data = B0H, Address = Desired Block to Erase Suspend	Data = B0B0H, Address = Desired Block Pair to Erase Suspend
Read		Status Register Data. Toggle $\overline{OE}$, $\overline{CE}_1$ or $\overline{CE}_2$ to update Status Register	Status Register Data. Toggle $\overline{OE}$ or ($\overline{CE}_1$ and $\overline{CE}_2$) to update Status Register
Standby		Check SR Bit 7 1 = Ready, 0 = Busy	Check SR Bit 7 and 15 1 = Ready, 0 = Busy
Standby		Check SR Bit 6 1 = Suspended, 0 = In Progress	Check SR Bit 6 and 14 1 = Suspended, 0 = In Progress
Write	Rd Array Cmd	Data = FFH	Data = FFFFH
Read		Read Data until finished	Read Data until finished
Write	Erase Resume	Data = D0H, Address = Valid Block Address.	Data = D0D0H, Address = Valid Block Pair Address.

Figure 15. Erase Suspend/Resume Algorithm. Allows Reads to Interrupt Erases.

POWER CONSUMPTION

STANDBY MODE

In most applications, software will only be accessing one device pair at a time. The Series 2 Card is defined to be in the standby mode when one device pair is in the Read Array Mode while the remaining devices are in the Deep-Sleep Mode. The Series 2 Card's $\overline{CE}_1$ and $\overline{CE}_2$ input signals must also be at V_{IH}. In standby mode, much of the card's circuitry is shut off, substantially reducing power consumption. Typical power consumption for a 20 Megabyte Series 2 card in standby mode is 65 μA.

SLEEP MODE

Writing a "1" to the PWRDWN bit of the GLOBAL POWERDOWN REGISTER places all FlashFile Memory devices into a Deep-Sleep mode. This disables most of the 28F008SA's circuitry and reduces current consumption to 0.2 μA per device. Additionally, when the host system pulls ASIC control logic high and latches all address and data lines (i.e., not toggling), the card's total current draw is reduced to approximately 5 μA (CMOS input levels) for a 20 Megabyte card. On writing a "0" to the PWRDWN bit (Global PowerDown Register) or any individual device pair (Sleep Control Register), a Deep-Sleep mode recovery period must be allowed for 28F008SA device circuitry to power back on.

SYSTEM DESIGN CONSIDERATIONS

POWER SUPPLY DECOUPLING

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current issues—standby, active and transient current peaks, produced by rising and falling edges of CE_1 and CE_2. The capacitive and inductive loads on the card and internal flash memory device pairs determine the magnitudes of these peaks.

The Flash Memory Card features on-card ceramic decoupling capacitors connected between V_{CC} and GND, and between V_{PP1}/V_{PP2} and GND to help transient voltage peaks.

On the host side, the card connector should also have a 4.7 μF electrolytic capacitor between V_{CC} and GND, as well as between V_{PP1}/V_{PP2} and GND. The bulk capacitors will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

POWER UP/DOWN PROTECTION

Each device in the Flash Memory Card is designed to offer protection against accidental erasure or writing, caused by spurious system-level signals that may exist during power transitions. The card will power-up into the Read Array Mode.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}_1$ (and/or $\overline{CE}_2$) must be low for a command write, driving either to V_{IH} will inhibit writes. With its Command User Interface, alteration of device contents only occurs after successful completion of the two-step command sequences.

While these precautions are sufficient for most applications, an alternative approach would allow V_{CC} to reach its steady state value before raising V_{PP1}/V_{PP2} above V_{CC} + 2.0V. In addition, upon powering-down, V_{PP1}/V_{PP2} should be below V_{CC} + 2.0V, before lowering V_{CC}.

HOT INSERTION/REMOVAL

The capability to remove or insert PC cards while the system is powered on (i.e., hot insertion/removal) requires careful design approaches on the system and card levels. To design for this capability consider card overvoltage stress, system power droop and control line stability.

A PCMCIA/JEIDA specified socket properly sequences the power supplies to the flash memory card via shorter and longer pins. This assures that hot insertion and removal will not result in card damage or data loss.

PCMCIA CARD INFORMATION STRUCTURE

The Card Information Structure (CIS) starts at address zero of the card's Attribute Memory Plane. It contains a variable-length chain of data blocks (tuples) that conform to a basic format as shown in Table 5. This section describes each tuple contained within the Series 2 Flash Memory Card.

The Device Information Tuple

This tuple (CISTPL_DEV = 01H) contains information pertaining to the card's speed and size. The Series 2 Card is offered with a 200 or 250 nanosecond access time. Card sizes range between 2 and 20 Megabytes.

Table 5. Tuple Format

Bytes	Data
0	Tuple Code: CISTPL_xxx. The tuple code 0FFH indicates no more tuples in the list.
1	Tuple Link: TPL_LINK. Link to the next tuple in the list. This can be viewed as the number of additional bytes in tuple, excluding this byte. If the link field is zero, the tuple body is empty. If the link field contains 0FFH, this tuple is the last tuple in the list.
2–n	Bytes specific to this tuple.

The Device Geometry Tuple

This tuple (CISTPL_DEVICEGEO = 1EH) is conceptually similar to a DOS disk geometry tuple (CISTPL_GEOMETRY), except it is not a format-dependent property; this deals with the fixed architecture of the memory device(s).

Fields are defined as follows:

DGTPL BUS—Value = n, where system bus width = $2^{(n-1)}$ bytes. N = 2 for standard PCMCIA Release 1.0/2.0 cards.

DGTPL EBS—Value = n, where the memory array's physical memory segments have a minimum erase block size of $2^{(n-1)}$ address increments of DGTPL_BUS-wide accesses.

DGTPL RBS—Value = n, where the memory array's physical memory segments have a minimum read block size of $2^{(n-1)}$ address increments of DGTPL_BUS-wide accesses.

DGTPL WBS—Value = n, where the memory array's physical memory segments have a minimum write block size of $2^{(n-1)}$ address increments of DGTPL_BUS-wide accesses.

DGTPL PART—Value = n, where the memory array's physical memory segments can have partitions subdividing the arrays in minimum granularity of $2^{(n-1)}$ number of erase blocks.

FL DEVICE INTERLEAVE—Value = n, where card architectures employ a multiple of $2^{(n-1)}$ times interleaving of the entire memory arrays with the above characteristics. Non-interleaved cards have values n = 1.

Jedec Programming Information Tuple

This tuple (CISTPL_JEDEC = 18H) contains the Intel manufacturing identifier (89H) and the 28F008SA device ID (A2H).

Level 1 Version/Product Information Tuple

This tuple (CISTPL_VERI = 15H) contains Level-1-version compliance and card-manufacturer information. Fields are described as follows:

TPLLV1 MAJOR—Major version number = 04H.

TPLLV1 MINOR—Minor version number = 01H for release 2.0.

TPLLV1 INFO—

Name of manufacturer	= intel;
Name of product	= SERIES2-"Card size";
Card type	= 2;
Speed	= 150 ns or 200 ns
Register Base	= REGBASE 4000H
Test Codes	= DBBDRELP
Legalities	= COPYRIGHT intel Corporation 1991

The Configurable Card Tuple

This tuple (CISTPL_CONF = 1AH) describes the interface supported by the card and the locations of the Card Configuration Registers and the Card Configuration Table.

Fields are described as follows:

TPCC SZ—Size of fields byte = 01H.

TPCC LAST—Index number of the last entry in the Card Configuration Table = 00H.

TPCC RADR—Configuration Registers Base Address in Reg Space = 4000H.

TPCC RMSK—Configuration Registers Present Mask = 03H.

The End-Of-List Tuple

The end-of-list tuple (CISTPL_END = FFH) marks the end of a tuple chain. Upon encountering this tuple, continue tuple processing as if a long-link to address 0 of common memory space were encountered.

Tuple Address	Value	Description
00H	01H	CISTPL_DEV
02H	03H	TPL_LINK
04H	53H 52H	DEVICE_INFO = FLASH 150 ns DEVICE_INFO = FLASH 200 ns
06H	 06H 0EH 26H 4EH	CARD SIZE 2M 4M 10M 20M
08H	FFH	END OF DEVICE
0AH	1EH	CISTPL_DEVICEGEO
0CH	06H	TPL_LINK
0EH	02H	DGTPL_BUS
10H	11H	DGTPL_EBS
12H	01H	DGTPL_RBS
14H	01H	DGPL_WBS
16H	03H	DGTPL_PART
18H	01H	FL_DEVICE INTERLEAVE
1AH	18H	CISTPL_JEDEC
1CH	02H	TPL_LINK
1EH	89H	INTEL J-ID
20H	A2H	28F008 J-ID
22H	15H	CISTPL_VER1
24H	50H	TPL_LINK
26H	04H	TPLLV1 MAJOR
28H	01H	TPLLV1 MINOR
2AH	69H	TPLLV1 INFO i
2CH	6EH	n
2EH	74H	t
30H	65H	e

Tuple Address	Value	Description
32H	6CH	l
34H	00H	END TEXT
36H	53H	S
38H	45H	E
3AH	52H	R
3CH	49H	I
3EH	45H	E
40H	53H	S
42H	32H	2
44H	2DH	—
46H	30H 30H 31H 32H	2M = 0 4M = 0 10M = 1 20M = 2
48H	32H 34H 30H 30H	2M = 2 4M = 4 10M = 0 20M = 0
4AH	20H	SPACE
4CH	00H	END TEXT
4EH	32H	CARD TYPE 2
50H	41H 42H 45H 5AH 48H 49H 4CH 4FH	A = 2M, 150 ns B = 4M, 150 ns E = 10M, 150 ns Z = 20M, 150 ns H = 2M, 200 ns I = 4M, 200 ns L = 10M, 200 ns O = 20M, 200 ns

4

Tuple Address	Value	Description
52H	20H	SPACE
54H	52H	REGBASE-R
56H	45H	E
58H	47H	G
5AH	42H	B
5CH	41H	A
5EH	53H	S
60H	45H	E
62H	20H	SPACE
64H	34H	4000h 4
66H	30H	0
68H	30H	0
6AH	30H	0
6CH	68H	h
6EH	20H	SPACE
70H	44H	D
72H	42H	B
74H	42H	B
76H	44H	D
78H	52H	R
7AH	45H	E
7CH	4CH	L
7EH	50H	P
80H	00H	END TEXT
82H	43H	COPYRIGHT C
84H	4FH	O
86H	50H	P
88H	59H	Y
8AH	52H	R
8CH	49H	I
8EH	47H	G
90H	48H	H
92H	54H	T
94H	20H	SPACE

Tuple Address	Value	Description
96H	69H	i
98H	6EH	n
9AH	74H	t
9CH	65H	e
9EH	6CH	l
A_0H	20H	SPACE
A2H	43H	CORPORATION C
A4H	4FH	O
A6H	52H	R
A8H	50H	P
AAH	4FH	O
ACH	52H	R
AEH	41H	A
B0H	54H	T
B2H	49H	I
B4H	4FH	O
B6H	4EH	N
B8H	20H	SPACE
BAH	31H	1
BCH	39H	9
BEH	39H	9
C0H	31H	1
C2H	00H	END TEXT
C4H	FFH	END OF LIST
C6H	1AH	CISTPL_CONF
C8H	06H	TPL_LINK
CAH	01H	TPCC_SZ
CCH	00H	TPCC_LAST
CEH	00H	TPCC_RADR
D0H	40H	TPCC_RADR
D2H	03H	TPCC_RMSK
D4H	FFH	END OF LIST
D6H	FFH	CISTPL_END
D8H	00H	INVALID ECIS ADDRESS

OPERATING SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read .... 0°C to +60°C(1)
During Erase/Write .... 0°C to +60°C

Storage Temperature .... −30°C to +70°C

Voltage on Any Pin with
Respect to Ground .... −2.0V to +7.0V(2)

V_{PP1}/V_{PP2} Supply Voltage with
Respect to Ground
during Erase/Write .... −2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage with
Respect to Ground .... −0.5V to +6.0V

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V tor periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
3. Maximum DC input voltage on V_{PP1}/V_{PP2} may overshoot to +14.0V tor periods less than 20 ns.

OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
T_A	Operating Temperature	0	60	°C
V_{CC}	V_{CC} Supply Voltage (5%)	4.75	5.25	V

4

COMMON DC CHARACTERISTICS, CMOS and TTL

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{LI}	Input Leakage Current	1, 3		±1	±20	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1		±1	±20	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND
V_{IL}	Input Low Voltage	1	−0.5		0.8	V	
V_{IH}	Input High Voltage (TTL)	1	2.4		V_{CC} + 0.3	V	
	Input High Voltage (CMOS)		0.7 V_{CC}		V_{CC} + 0.3		
V_{OL}	Output Low Voltage	1	V_{SS}		0.4	V	$V_{CC} = V_{CC}$ Min I_{OL} = 3.2 mA
V_{OH}	Output High Voltage	1	4.0		V_{CC}	V	$V_{CC} = V_{CC}$ Min I_{OH} = 2.0 mA
V_{PPL}	V_{PP} during Read Only Operations	1, 2	0.0		6.5	V	
V_{PPH}	V_{PP} during Read/Write Operations	1	11.4		12.6	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage	1	2.0			V	

NOTES:
1. Values are the same for byte and word wide modes and for all card densities.
2. Block Erases/Data Writes are inhibited when V_{PP} and V_{PPL} and not guaranteed in the range between V_{PPH} and V_{PPL}.
3. Exceptions: With V_{IN} = GND, the leakage on $\overline{CE_1}$, $\overline{CE_2}$, $\overline{REG}$, $\overline{OE}$, $\overline{WE}$, will be ≤ 500 μA due to internal pullup resistors and, with $V_{IN} = V_{CC}$, RST leakage will be ≤ 500 μA due to internal pulldown resistor.

DC CHARACTERISTICS, CMOS

Symbol	Parameter		Notes	Byte Wide Mode			Word Wide Mode			Unit	Test Condition
				Min	Typ	Max	Min	Typ	Max		
I_{CCR}	V_{CC} Read Current		1, 3		45	85		65	120	mA	$V_{CC} = V_{CC}$ Max, Control Signals = GND t_{CYCLE} = 200 ns, I_{OUT} = 0 mA
I_{CCW}	V_{CC} Write Current		1, 3		35	80		45	110	mA	Data Write in Progress
I_{CCE}	V_{CC} Erase Current		1, 2, 3		35	80		45	110	mA	Block (Pair) Erase in Progress
I_{CCS}	V_{CC} Standby Current	4 Meg	1, 4, 6		61	222		61	222	μA	$V_{CC} = V_{CC}$ Max, Control Signals $= V_{IH}$
		10 Meg			63	230		63	230		
		20 Meg			65	242		65	242		
I_{CCSL}	V_{CC} Sleep Current	4 Meg	1, 4, 5		2	25		2	25	μA	
		10 Meg			3	32		3	32		
		20 Meg			5	44		5	44		
I_{PPW}	V_{PP} Write Current ($V_{PP} = V_{PPH}$)		1, 3		10	30		20	60	mA	Data Write in Progress
I_{PPE}	V_{PP} Erase Current ($V_{PP} = V_{PPH}$)		1, 3		10	30		20	60	mA	Block (Pair) Erase in Progress
I_{PPSL}	V_{PP} Sleep Current	4 Meg	1, 5		0.5	4		0.5	4	μA	
		10 Meg			1	10		1	10		
		20 Meg			2	20		2	20		
I_{PPS1}	V_{PP} Standby or Read Current ($V_{PP} \leq V_{CC}$)	4 Meg	1, 6		1.5	13		1.5	13	μA	
		10 Meg			2	19		2	19		
		20 Meg			3	29		3	29		
I_{PPS2}	V_{PP} Standby or Read Current ($V_{PP} = V_{PPH}$)	4 Meg	1, 6		90	203		180	402	μA	
		10 Meg			91	209		181	408		
		20 Meg			92	219		182	418		

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C.
2. The Data Sheet specification for the 28F008SA in Erase Suspend (I_{CCES}) is 5 mA typical and 10 mA max with the device deselected. If the device(s) are read while in Erase Suspend Mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Standby or Sleep currents are not included for non-accessed devices.
4. Address and data inputs to card static. Control line voltages equal to V_{IH} or V_{IL}.
5. All 28F008SA devices in Deep-Sleep (PowerDown) mode.
6. In Byte and Word Mode, all but two devices in Deep-Sleep.

DC CHARACTERISTICS, TTL

Symbol	Parameter		Notes	Byte Wide Mode			Word Wide Mode			Unit	Test Condition
				Min	Typ	Max	Min	Typ	Max		
I_{CCR}	V_{CC} Read Current		1, 3		75	150		100	200	mA	$V_{CC} = V_{CC}$ Max, $t_{CYCLE} = 200$ ns, $I_{OUT} = 0$ mA
I_{CCW}	V_{CC} Write Current		1, 3		60	130		70	160	mA	Data Write in Progress
I_{CCE}	V_{CC} Erase Current		1, 2, 3		60	130		70	160	mA	Block (Pair) Erase in Progress
I_{CCS}	V_{CC} Standby Current	4 Meg	1, 4, 6, 7		20	100		20	100	mA	$V_{CC} = V_{CC}$ Max, Control Signals $= V_{IH}$
		10 Meg									
		20 Meg									
I_{CCSL}	V_{CC} Sleep Current	4 Meg	1, 4, 5, 7		20	100		20	100	mA	
		10 Meg									
		20 Meg									
I_{PPW}	V_{PP} Write Current ($V_{PP} = V_{PPH}$)		1, 3		10	30		20	60	mA	Data Write in Progress
I_{PPE}	V_{PP} Erase Current ($V_{PP} = V_{PPH}$)		1, 3		10	30		20	60	mA	Block (Pair) Erase in Progress
I_{PPSL}	V_{PP} Sleep Current	4 Meg	1, 5		3	20		3	20	µA	
		10 Meg			8	50		8	50		
		20 Meg			16	100		16	100		
I_{PPS1}	V_{PP} Standby or Read Current ($V_{PP} \leq V_{CC}$)	4 Meg	1, 6		3	25		3	25	µA	
		10 Meg			8	55		8	55		
		20 Meg			16	105		16	105		
I_{PPS2}	V_{PP} Standby or Read Current ($V_{PP} = V_{PPH}$)	4 Meg	1, 6		92	215		182	410	µA	
		10 Meg			97	245		186	440		
		20 Meg			105	295		194	490		

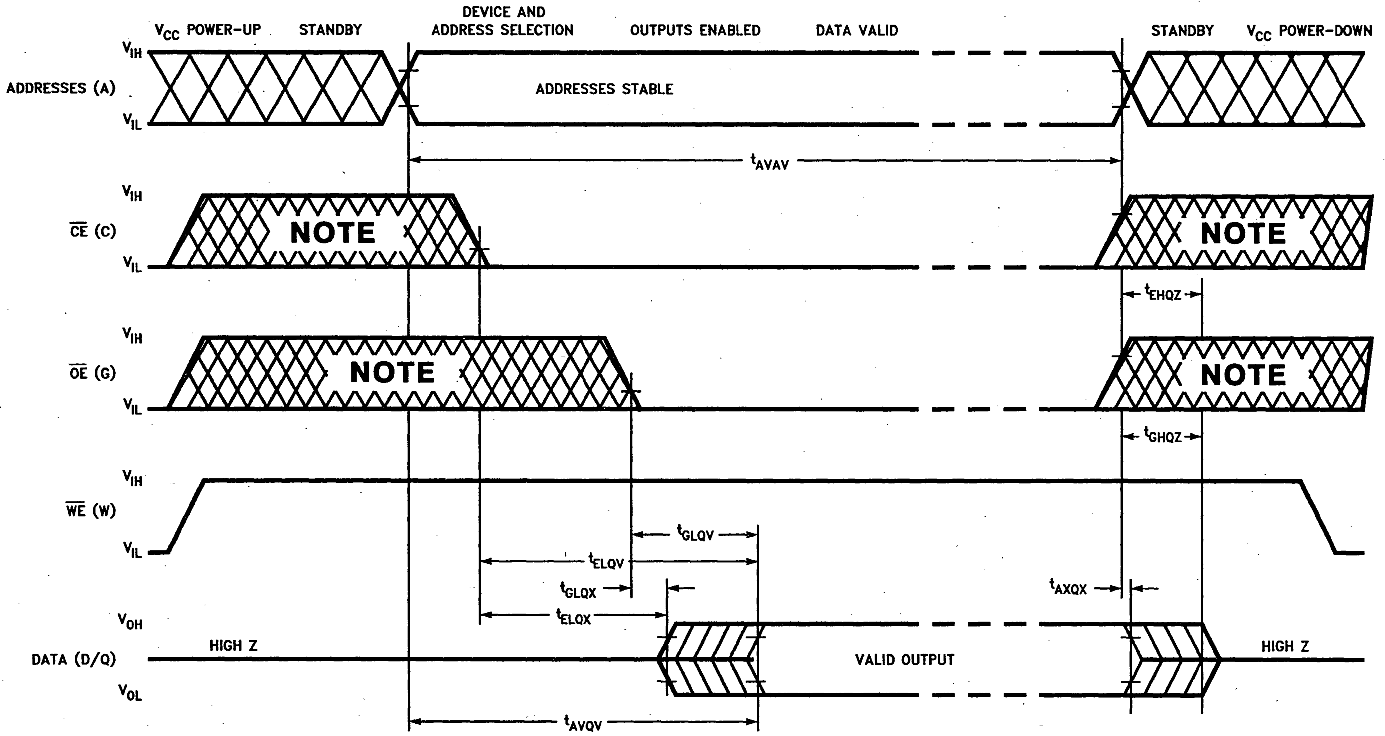
NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at $V_{CC} = 5.0V$, $V_{PP} = 12.0V$, $T = 25°C$.
2. The Data Sheet specification for the 28F008SA in Erase Suspend (I_{CCES}) is 5 mA typical and 10 mA max with the device deselected. If the device(s) are read while in Erase Suspend Mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Standby or Sleep currents are not included for non-accessed devices.
4. Address and data inputs to card static. Control line voltages equal to V_{IH} or V_{IL}.
5. All 28F008SA devices in Deep-Sleep (PowerDown) mode.
6. In Byte and Word Mode, all but two devices in Deep-Sleep.
7. The current consumption from the 28F008SA is insignificant in relation to the ASIC's.

AC CHARACTERISTICS

AC Timing Diagrams and characteristics are guaranteed to meet or exceed PCMCIA Release 2.0 specifications. PCMCIA allows a 300 ns access time for Attribute Memory. Note that read and write access timings to the Series 2 Flash Memory Card's Common and Attribute Memory Planes are identical at 200 ns. Furthermore, there is no delay in switching between the Common and Attribute Memory Planes.

COMMON AND ATTRIBUTE MEMORY, AC CHARACTERISTICS: Read-Only Operations

Symbol		Parameter	Notes	Min	Max	Unit
JEDEC	PCMCIA					
t_{AVAV}	t_{RC}	Read Cycle Time		200		ns
t_{AVQV}	t_a (A)	Address Access Time			200	ns
t_{ELQV}	t_a (CE)	Card Enable Access Time			200	ns
t_{GLQV}	t_a (OE)	Output Enable Access Time			100	ns
t_{EHQX}	t_{dis} (CE)	Output Disable Time from $\overline{CE}$			90	ns
t_{GHQZ}	t_{dis} (CE)	Output Disable Time from $\overline{OE}$			70	ns
t_{GLQX}	t_{en} (CE)	Output Enable Time from $\overline{CE}$		5		ns
t_{ELQX}	t_{en} (OE)	Output Enable Time from $\overline{OE}$		5		ns
t_{AXQX}	t_v (A)	Data Valid from Add Change		0		ns
t_{PHQV}		Powerdown Recovery to Output Delay		500		ns
	t_{su} (V_{CC})	CE Setup Time on Power-Up		1		ms
		First Access after Reset		500		ns



NOTE:
1. The hatched area may be either high or low.

Figure 16. AC Waveform for Read Operations

COMMON AND ATTRIBUTE MEMORY, AC CHARACTERISTICS: Write Operations[1]

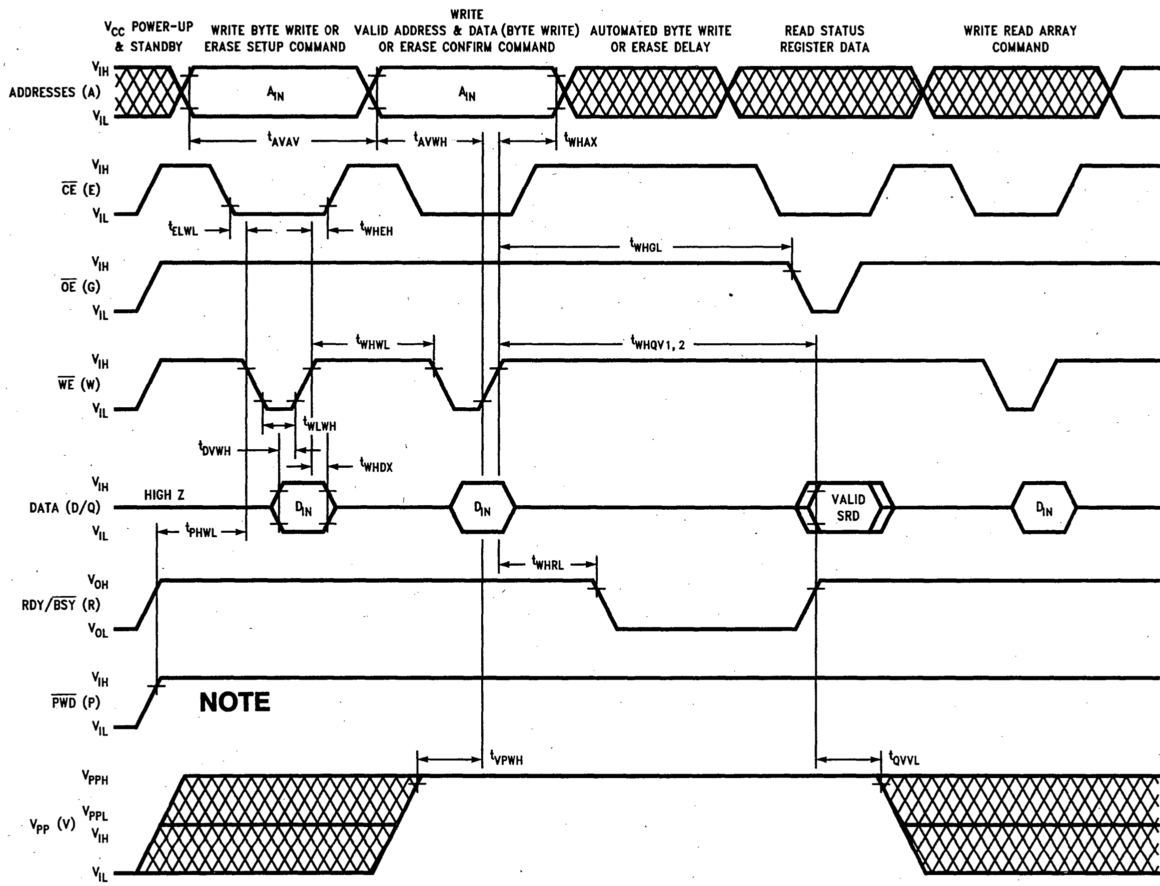
Symbol		Parameter	Notes	Min	Max	Unit
JEDEC	PCMCIA					
t_{AVAV}	t_{WC}	Write Cycle Time		200		ns
t_{WLWH}	t_W (WE)	Write Pulse Width		120		ns
t_{AVWL}	t_{su} (A)	Address Setup Time		20		ns
t_{AVWH}	t_{su} (A-WEH)	Address Setup Time for $\overline{WE}$		140		ns
t_{VPWH}	t_{vps}	V_{PP} Setup to $\overline{WE}$ Going High		100		ns
t_{ELWH}	t_{su} (CE-WEH)	Card Enable Setup Time for $\overline{WE}$		140		ns
t_{DVWH}	t_{su} (D-WEH)	Data Setup Time for $\overline{WE}$		60		ns
t_{WHDX}	t_h (D)	Data Hold Time		30		ns
t_{WHAX}	t_{rec} (WE)	Write Recover Time		30		ns
t_{WHRL}		$\overline{WE}$ High to RDY/$\overline{BSY}$			120	ns
t_{WHQV1}		Duration of Data Write Operation		6		µs
t_{WHQV2}		Duration of Block Erase Operation		0.3		sec
t_{QVVL}		V_{PP} Hold from Operation Complete	2			ns
t_{WHGL}	t_h (OE-WE)	Write Recovery before Read		10		ns
t_{PHWL}		Powerdown Recovery to $\overline{WE}$ Going Low		1		µs

NOTES:
1. Read timing characteristics during erase and data write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only operations.
2. Refer to text on Data-Write and Block-Erase Operations.

BLOCK ERASE AND DATA WRITE PERFORMANCE

Parameter	Notes	Min	Typ[3]	Max	Unit
Block Pair Erase Time[1]	2		1.6	10	sec
Block Pair Write Time	2		0.6	2.1	sec

NOTES:
1. Individual blocks can be erased 100,000 times.
2. Excludes System-Level Overhead.
3. 25°C, 12.0 V_{PP}.



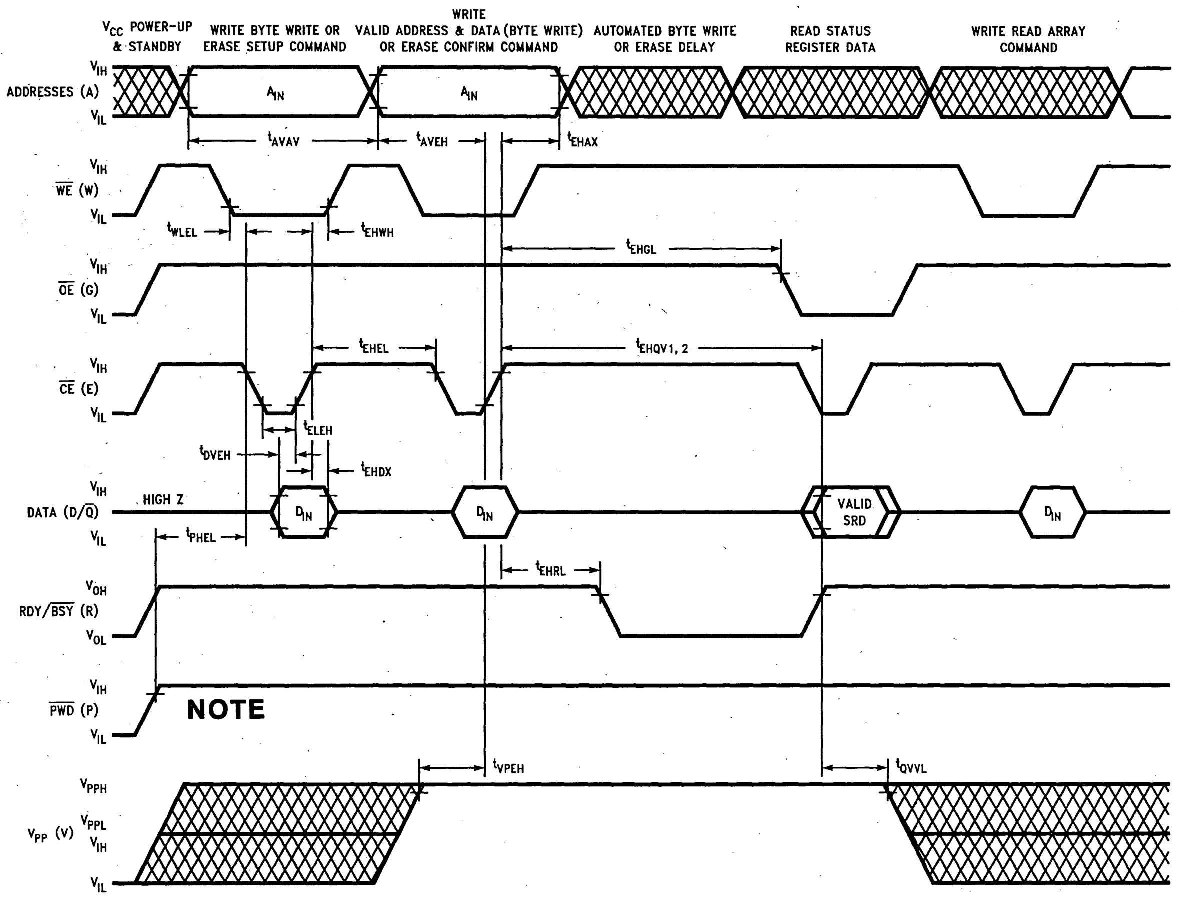
NOTE:
As shown, $\overline{\text{PWD}}$ is a carry-over from the component-level diagram; this signal is generated in the card by the ASIC by writing to the appropriate register.

Figure 17. AC Waveform for Write Operations

COMMON AND ATTRIBUTE MEMORY, AC CHARACTERISTICS: $\overline{CE}$-Controlled Write Operations(1)

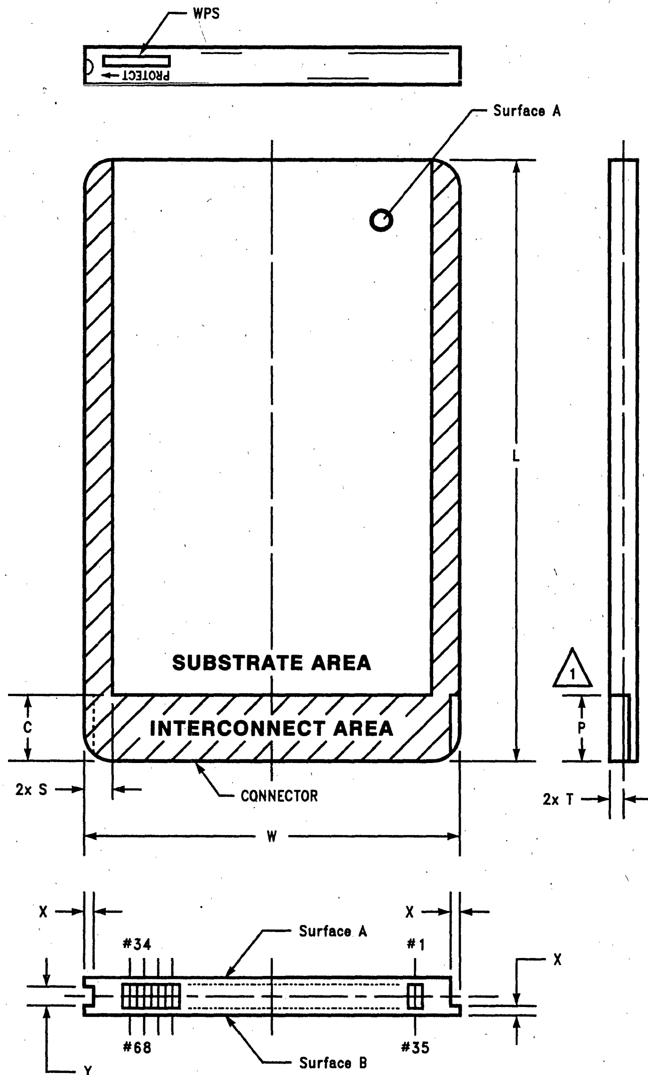
Symbol		Parameter	Notes	Min	Max	Unit
JEDEC	PCMCIA					
t_{AVAV}	t_{WC}	Write Cycle Time	1	200		ns
t_{ELEH}	t_W (WE)	Chip Enable Pulse Width	1	120		ns
t_{AVEL}	t_{su} (A)	Address Setup Time	1	20		ns
t_{AVEH}	t_{su} (A-WEH)	Address Setup Time for $\overline{CE}$	1	140		ns
t_{VPEH}	t_{vps}	V_{PP} Setup to $\overline{CE}$ Going High	1	100		ns
t_{WLEH}	t_{su} (CE-WEH)	Write Enable Setup Time for $\overline{CE}$	1	140		ns
t_{DVEH}	t_{su} (D-WEH)	Data Setup Time for $\overline{CE}$	1	60		ns
t_{EHDX}	t_h (D)	Data Hold Time	1	30		ns
t_{EHAX}	t_{rec} (WE)	Write Recover Time	1	30		ns
t_{EHRL}		$\overline{CE}$ High to RDY/$\overline{BSY}$	1		120	ns
t_{EHQV1}	Duration of Data Write	Duration of Data Write Operation	1	6		µs
t_{EHQV2}	Duration of Erase	Duration of Block Erase Operation	1	0.3		sec
t_{QVVL}		V_{PP} Hold from Operation Complete	1, 2	0		ns
t_{EHGL}	t_h (OE-WE)	Write Recovery before Read	1	10		ns
t_{PHEL}		Powerdown Recovery to $\overline{CE}$ Going Low		1		µs

NOTES:
1. Read timing characteristics during erase and data write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only operations.
2. Refer to text on Data-Write and Block-Erase Operations.



NOTE:
As shown, $\overline{PWD}$ is a carry-over from the component-level diagram; this signal is generated in the card by the ASIC by writing to the appropriate register.

Figure 18. Alternate AC Waveform for Write Operations



C MIN	L ± 0.008	P MIN △1	S MIN	T △2	W ± 0.004	X ± 0.002	Y ± 0.002
0.294 (10.0)	3.370 (85.60)	0.394 (10.0)	0.118 (3.0)	0.065 (1.65)	2.126 (54.0)	0.039 (1.00)	0.063 (1.60)

1. POLARIZATION KEY LENGTH.
2. INTERCONNECT AREA TOLERANCE = ±0.002
   SUBSTRATE AREA TOLERANCE = ±0.004
3. MILLIMETERS ARE IN PARENTHESIS ().

290434–24

Figure 19. Series 2 Flash Memory Card Package Dimensions

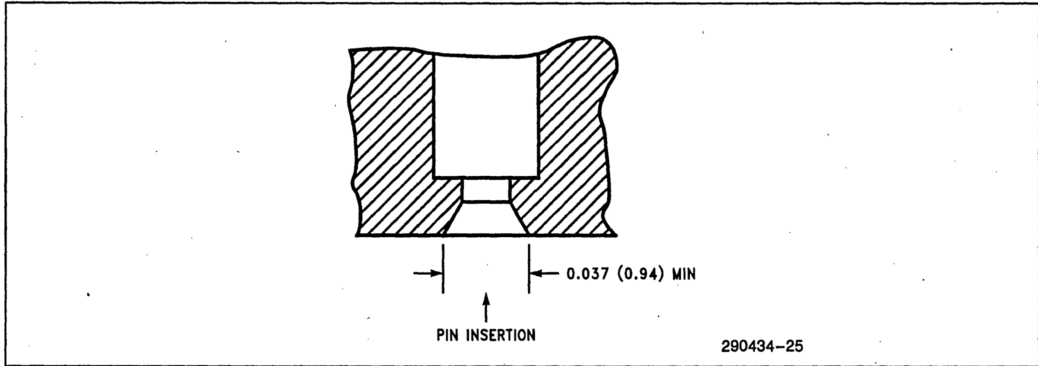


Figure 20. Card Connector Socket

L1 L2 L3 L1

PIN SOCKET CONTACT

1. PIN/SOCKET CONTACT AREA
2. MILLIMETERS ARE IN PARENTHESIS ()

290434-26

L1 MAX	L2	L3 REF
0.020 (0.5)	Pin Type—See Table 1 Detect 0.059 (1.5) ±0.039 General 0.084 (2.1) ±0.064 Power 0.098 (2.5) ±0.078	0.024 (0.6)

Figure 21. Pin/Socket Contact Length with Wipe

4

Table 5. Capacitance $T_A = 25°C, f = 1.0$ MHz

Symbol	Characteristics	Commercial		Unit
		Min	Max	
C_{IN}	Address/Control Capacitance (A_0–A_8, $\overline{CE}_1$, $\overline{CE}_2$)		30	pF
	Address/Control Capacitance (A_9–A_{24}, all others)		20	pF
	V_{CC}, V_{PP}		2	μF
C_{OUT}	Output Capacitance		20	pF

ORDERING INFORMATION

iMC020FLSA,SBXXXXX

WHERE:

i = INTEL
MC = MEMORY CARD
020 = DENSITY IN MEGABYTES (004,010,020 AVAILABLE)
FL = FLASH TECHNOLOGY
S = BLOCKED ARCHITECTURE
A = REVISION
SBXXXXX = CUSTOMER IDENTIFIER

ADDITIONAL INFORMATION	ORDER NUMBER
AP-361 "Implementing the Integrated Registers of the Series 2 Flash Memory Card"	292096
AP-364 "28F008SA Automation and Algorithms"	292099
28F008SA FlashFile™ Memory Data Sheet	290429
ER-27 "The Intel 28F008SA Flash Memory"	294011
ER-28 "ETOX™ III Flash Memory Technology"	294012
AP-359 "28F008SA Hardware Interfacing"	292094
AP-360 "28F008SA Software Drivers"	292095
iMC001FLKA 1-Mbyte Flash Memory Card	290399
iMC002FLKA 2-Mbyte Flash Memory Card	290412
iMC004FLKA 4-Mbyte Flash Memory Card	290388

REVISION HISTORY

Number	Description
02	Added 150 ns TUPLE, Deleted 250 ns TUPLE Corrected Global Power Register Address to 4002H Corrected Write Protection Register Address to 4104H Corrected Ready-Busy Mode Register Address to 4140H I_{CC} Standby Byte Wide Mode MAX/TYP Increased Added Power-On Timing Spec Added First Access after Reset Spec Changed Advanced Information to Preliminary

iSM001FLKA
1 MBYTE (512K x 16) CMOS FLASH SIMM

- **High-Performance**
  - **— 120 ns Maximum Access Time**
  - **— 16.67 MB/s Read Transfer Rate**
- **10,000 Rewrite Cycles Minimum/ Component**
- **Flash Electrical Chip-Erase**
  - **— 1 Second Typical Chip-Erase**
- **16 μs Typical Word Write**
  - **— Up to 1 Mb/s Write Transfer Rate**
- **Inherent Non-volatility**
  - **— No Batteries or Disk Required for Back-up**
  - **— 0W Data Retention Power**
- **CMOS Low Power Consumption**
  - **— 20.3 mA Typical Active Current**
  - **— 0.4 mA Typical Standby Current**
- **Standard 80-Pin Insertable Module**
  - **— 0.050 Centerline Lead Spacing**
  - **— Upgrade Path through 128M bytes**
- **Hardware Presence Detect**
- **Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface**
- **Noise Immunity Features**
  - **— ±10% V_{CC} Tolerance**
  - **— Maximum Latch-Up Immunity Through EPI Processing**
- **12.0V ±5% V_{PP}**
- **Integrated Program/Erase Stop Timer**
- **ETOX™ II Nonvolatile Flash Technology**
  - **— High-Volume Manufacturing Experience**

Intel's iSM001FLKA Flash SIMM (Single In-Line Memory Module) is targeted at high-density read/write non-volatile memory. The iSM001FLKA enables you to optimize board space; to offer incremental memory expansion similar to today's DRAM; and to assure continued access to today's and tomorrow's surface-mount technologies. Intel's iSM001FLKA offers a reliable sold-state alternative for mass storage. The flash memory module is also ideal for high performance code and data storage as well as data recording and accumulation.

The iSM001FLKA, composed of eight 1 Mb flash memories in plastic leaded chip carrier (N28F010), is organized as 524,288 words of 16 bits. The PLCCs are mounted, four to a side, together with 0.1 μF decoupling capacitors on an 80-pin standard, low-profile module.

Extended erase and program cycling capability is designed into Intel's ETOX™ II (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional nonvolatile memory.

Intel's iSM001FLKA Flash SIMM employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 120 ns access time provides no WAIT state performance for a wide range of microprocessors and microcontrollers. Maximum standby current of 0.8 mA translates into power savings when the memory module is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from −1V to V_{CC} + 1V.

October 1992
Order Number: 290244-006

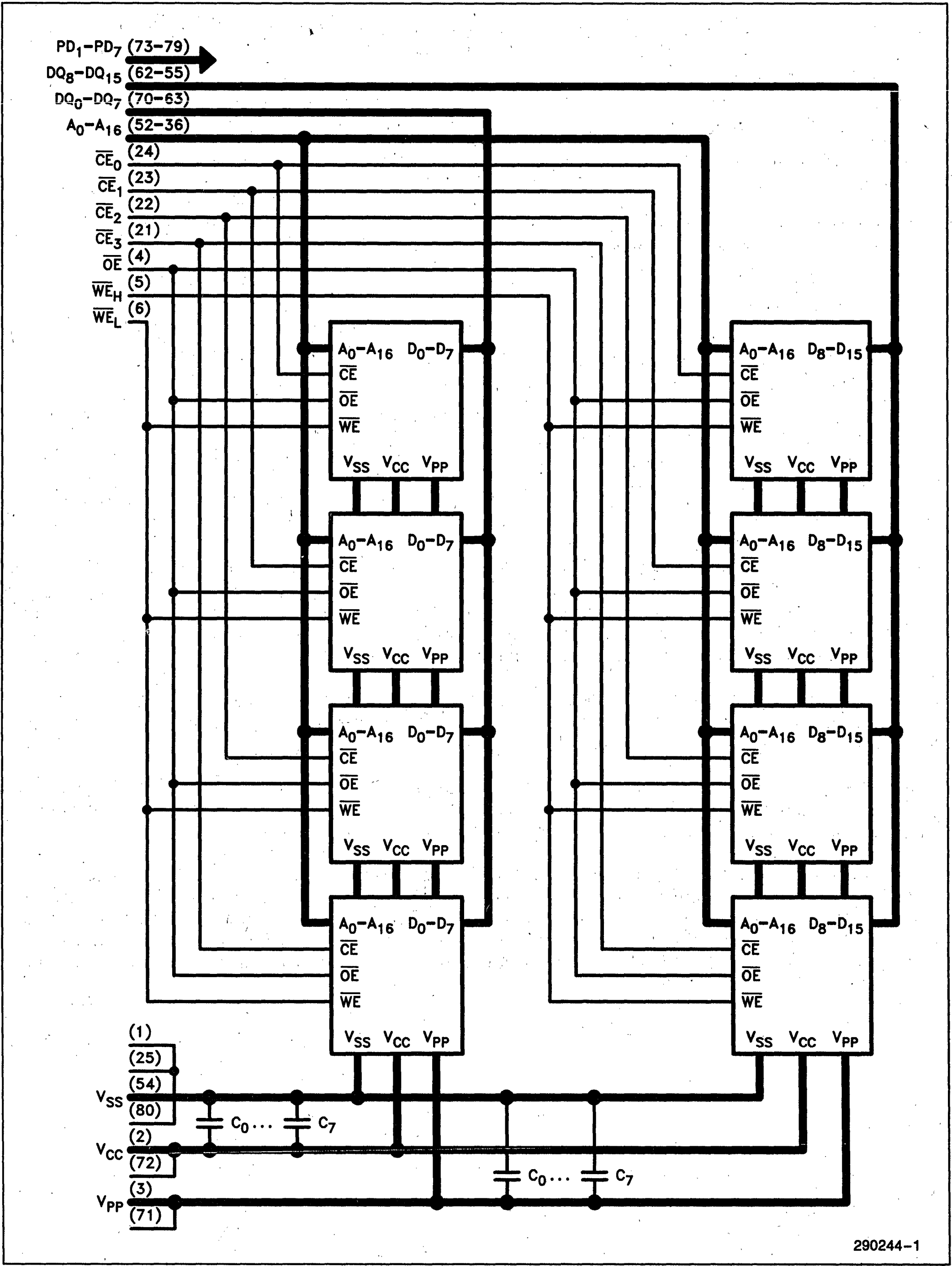


Figure 1. iSM001FLKA Functional Block Diagram

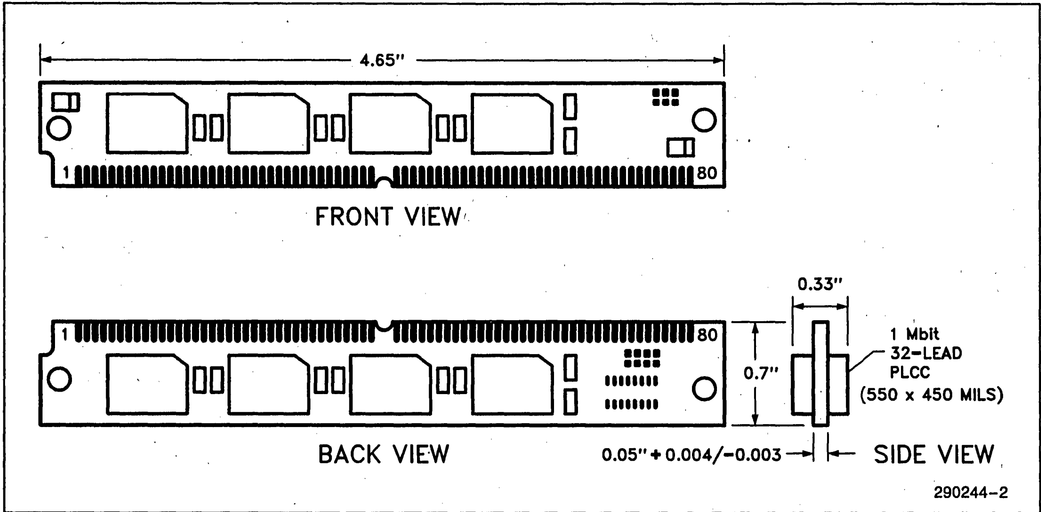


Figure 2. iSM001FLKA Pin Configurations

Table 1. Pinout

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	V_{SS}	21	CE3	41	A_{11}	61	DQ_9
2	V_{CC}	22	CE2	42	A_{10}	62	DQ_8
3	V_{PP}	23	CE1	43	A_9	63	DQ_7
4	$\overline{OE}$	24	CE0	44	A_8	64	DQ_6
5	$\overline{WEH}$	25	V_{SS}	45	A_7	65	DQ_5
6	$\overline{WEL}$	26	RES	46	A_6	66	DQ_4
7	NC	27	RES	47	A_5	67	DQ_3
8	RES	28	RES	48	A_4	68	DQ_2
9	RES	29	RES	49	A_3	69	DQ_1
10	RES	30	NC	50	A_2	70	DQ_0
11	RES	31	NC	51	A_1	71	V_{PP}
12	RES	32	NC	52	A_0	72	V_{CC}
13	RES	33	NC	53	RES	73	PD_1
14	RES	34	NC	54	V_{SS}	74	PD_2
15	RES	35	NC	55	DQ_{15}	75	PD_3
16	RES	36	A_{16}	56	DQ_{14}	76	PD_4
17	NC	37	A_{15}	57	DQ_{13}	77	PD_5
18	NC	38	A_{14}	58	DQ_{12}	78	PD_6
19	NC	39	A_{13}	59	DQ_{11}	79	PD_7
20	NC	40	A_{12}	60	DQ_{10}	80	V_{SS}

Table 2. Pin Description

<table>
<tr><th>Symbol</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>A_0–A_{16}</td><td>INPUT</td><td>ADDRESS INPUTS for memory addresses. Addresses are internally latched during a write cycle.</td></tr>
<tr><td>DQ_0–DQ_{15}</td><td>INPUT/ OUTPUT</td><td>DATA INPUT/OUTPUT: Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.</td></tr>
<tr><td>$\overline{CE}_0$–$\overline{CE}_3$</td><td>INPUT</td><td>CHIP ENABLE: Activates each device's control logic, input buffers, decoders, and sense amplifiers. Each line is unique to one set of 2 devices (word). $\overline{CE}_X$ is active low; $\overline{CE}_X$ high deselects the memory device and reduces power consumption to standby levels. Only one $\overline{CE}_X$ may be active at a time.</td></tr>
<tr><td>$\overline{OE}$</td><td>INPUT</td><td>OUTPUT ENABLE: Gates the devices outputs through the data buffers during a read cycle. $\overline{OE}$ is active low.</td></tr>
<tr><td>$\overline{WE}_H$; $\overline{WE}_L$</td><td>INPUT</td><td>WRITE ENABLE controls writes to the control register and the array.
($\overline{WE}_H$ = High Byte; $\overline{WE}_L$ = Low Byte)
Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse.
NOTE: With $V_{PP} \leq 6.5V$, memory contents cannot be altered.</td></tr>
<tr><td>V_{PP}</td><td></td><td>ERASE/PROGRAM POWER SUPPLY for writing the command register, erasing the entire array, or programming bytes in the array (12V ± 5%).</td></tr>
<tr><td>V_{CC}</td><td></td><td>DEVICE POWER SUPPLY: (5V ± 10%).</td></tr>
<tr><td>V_{SS}</td><td></td><td>GROUND.</td></tr>
<tr><td>NC</td><td></td><td>NO INTERNAL CONNECTION to device. Pin may be driven or left floating.
<table>
<tr><th>Pin</th><th>Function</th></tr>
<tr><td>17</td><td>$\overline{CE}_7$</td></tr>
<tr><td>18</td><td>$\overline{CE}_6$</td></tr>
<tr><td>19</td><td>$\overline{CE}_5$</td></tr>
<tr><td>20</td><td>$\overline{CE}_4$</td></tr>
<tr><td>30</td><td>A_{22}</td></tr>
<tr><td>31</td><td>A_{21}</td></tr>
<tr><td>32</td><td>A_{20}</td></tr>
<tr><td>33</td><td>A_{19}</td></tr>
<tr><td>34</td><td>A_{18}</td></tr>
<tr><td>35</td><td>A_{17}</td></tr>
</table></td></tr>
<tr><td>RES</td><td></td><td>RESERVED for future product enhancements.</td></tr>
<tr><td>PD_1–PD_7</td><td></td><td>PRESENCE DETECT: Denotes word depth (512K) and access time of device. See Table 3, "Presence Detect "PD" Pins" on Page 5.</td></tr>
</table>

Table 3. Presence Detect "PD" Pins

MODULE CAPACITY IDENTIFICATION			
MODULE CAPACITY WORD DEPTH	**PD6**	**PD2**	**PD1**
NO MODULE	O	O	O
256K/32M	O	O	S
512K/64M	O	S	O
1M/128M	O	S	S
2M/256M	S	O	O
4M/512M	S	O	S
8M/1G	S	S	O
16M/2G	S	S	S

MODULE SPEED IDENTIFICATION				
MAXIMUM ACCESS TIME	**PD7**	**PD5**	**PD4**	**PD3**
>300 ns	S	S	S	S
300 ns	S	S	S	O
250 ns	S	S	O	S
200 ns	S	S	O	O
185 ns	S	O	S	S
150 ns	S	O	S	O
135 ns	S	O	O	S
120 ns	S	O	O	O
100 ns	O	S	S	S
85 ns	O	S	S	O
70 ns	O	S	O	S
60 ns	O	S	O	O
50 ns	O	O	S	S
40 ns	O	O	S	O
30 ns	O	O	O	S
ND	O	O	O	O

O = OPEN CIRCUIT ON MODULE
S = SHORT CIRCUIT TO GROUND ON MODULE
ND = NOT DEFINED

4

SINGLE IN-LINE MEMORY MODULE BOARD

PC substrate: Glass Epoxy [0.05″ +0.004/−0.003 nominal thickness]. The iSM001FLKA low-profile SIMM mounts easily between expansion slots. See Appendix A for a list of 80-pin socket suppliers.

APPLICATIONS

With high density, nonvolatility, and extended cycling capability, Intel's iSM001FLKA flash SIMMs offer an innovative alternative to disk and battery-backed static RAM.

Primary applications and operating systems can be stored in flash, eliminating the slow disk-to-DRAM download process. Performance is dramatically enhanced and power consumption is reduced—a consideration particularly important in portable equipment. Flexibility is increased with Flash's electrical chip erasure allowing in-system updates to operating systems and application code.

In diskless workstations and terminals, network traffic is reduced to a minimum and systems are instant-on. Reliability exceeds that of electro-mechanical media. Often in these environments, power glitches force extended re-boot periods for all networked terminals. This mishap is no longer an issue if boot code, operating systems, communication protocols and primary applications are flash-resident in each terminal.

For embedded systems that rely on dynamic RAM/disk for main system memory or nonvolatile backup storage, Flash SIMMs provide a solid state alternative in a minimal form factor. Flash memory provides higher performance, lower power consumption and instant-on capability. Additionally, flash is more rugged and reliable in harsh environments where extreme temperatures and shock can cause disk-based systems to fail.

For systems currently using a high-density static RAM/battery configuration for code updates and data accumulation, flash memory's inherent nonvolatility eliminates the need for battery backup. The possibility of battery failure is removed. This consideration is important for portable equipment and medical instruments, both requiring continuous performance. In addition, flash memory offers a four-to-one cost advantage over SRAM.

Flash memory's electrical chip erasure, byte reprogrammability and complete nonvolatility fit well with data accumulation and recording needs. Electrical chip-erasure gives the designer a "blank-slate" in which to log or record data. Data can be periodically off-loaded for analysis-erasing the slate and repeating the cycle.

Flash SIMMs add additional flexibility to designers by offering end-users incremental expansion memory. As code requirements grow or as memory prices drop, your customers have the option of adding more memory.

PRINCIPALS OF OPERATION

The iSM001FLKA operates as eight N28F010 flash memories connected as shown in the Functional Block Diagram on Page 2.

The iSM001FLKA, organized as 512K x 16, can also be configured for 8- and 32-bit systems. For 32-bit systems, add a second SIMM to your design as currently done with DRAM. For byte-wide operation, buffer the SIMMs DQ_0–DQ_7 and DQ_8–DQ_{15} lines with an octal transceiver; then, tie the buffered outputs together to form the 8-bit bus. Decode the transceiver's enable input with an address line.

The iSM001FLKA features hardware presence detect pins to facilitate memory design. The presence detect pins (PD1–PD7) indicate module word depth and maximum access speed (see Table 3 on the previous page). The pins allow memory-specific wait-state generation upon system initialization. To use the presence capability, pull-up the PD1–PD7 lines through a pull-up resistor. Read the lines through a port and select the appropriate memory depth and speed from a PD data table.

In the absence of high voltage on the modules V_{PP} pins, the iSM001FLKA is a read-only memory array. Manipulation of the module's control pins yields standard read, standby and output disable functions.

Read, standby and output disable operations are also available when high voltage is applied to the V_{PP} pins. In addition, high voltage on the V_{PP} pins enables erasure and programming of the module's devices. All functions associated with altering the memory contents of one or more devices—erase, erase verify, program and program verify—are accessed via each flash device's command register.

Commands are written to a device's command register using standard microprocessor write timings. Register contents serve as input to the devices internal state-machine which controls the erase and programming circuitry. Write cycles to a device also internally latch addresses and data needed for programming or erase operations. With the appropriate command written to a device's register, standard microprocessor read timings output array data, access the int$_e$ligent identifier codes, or output data for erase and program verification.

Table 4. Bus Operations

Operation		V_{PP}(1)	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_0-DQ_{15}
READ-ONLY	Read	V_{PPL}	V_{IL}	V_{IL}	V_{IH}	Data Out
	Output Disable	V_{PPL}	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby	V_{PPL}	V_{IH}	X	X	Tri-State
READ/WRITE	Read	V_{PPH}	V_{IL}	V_{IL}	V_{IH}	Data Out(3)
	Output Disable	V_{PPH}	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby(4)	V_{PPH}	V_{IH}	X	X	Tri-State
	Write	V_{PPH}	V_{IL}	V_{IH}	V_{IL}	Data In(5)

NOTES:
1. Refer to DC Characteristics. When $V_{PP} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes are accessed via a command register write sequence. Refer to Table 5. All other addresses are low.
3. Read operations with $V_{PP} = V_{PPH}$ may access array data or the int$_e$ligent Identifier™ codes.
4. With V_{PP} at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
5. Refer to Table 5 for valid Data-In during a write operation.
6. X can be V_{IL} or V_{IH}.

Integrated Stop Timer

Successive command write cycles define the duration of program and erase operations; specifically, the program or erase time durations are normally terminated by associated program or erase verify commands. An integrated stop timer provides simplified timing control over these operations; thus eliminating the need for maximum program/erase timing specifications. Programming and erase pulse durations are minimums only. When the stop timer terminates a program or erase operation, the device enters an inactive state and remains inactive until receiving the appropriate verify or reset command.

Write Protection

A device's command register is only active when V_{PP} is at high voltage. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable—available only when memory updates are desired. When $V_{PP} = V_{PPL}$, the contents of the register default to the read command, making the iSM001FLKA a read-only memory. In this mode, the memory contents cannot be altered.

Or, the system designer may choose to "hardwire" V_{PP}, making the high voltage supply constantly available. In this instance, all operations are performed in conjunction with the command register. The iSM001FLKA is designed to accommodate either design practice, and to encourage optimization of flash's processor-memory interface.

The following section first discusses byte-wide organization, building a basic understanding of byte-wide bus operations, command definitions, and programming and erasure algorithms. The section concludes with performance enhancements for both 16- and 32-bit systems.

BUS OPERATIONS

Read

Each of the iSM001FLKA's flash memory devices has two control functions, both of which must be logically active, to obtain data. Chip-Enable ($\overline{CE}_X$) is the power control and should be used for device selection. Four chip enables ($\overline{CE}_0-\overline{CE}_3$) control the array's eight devices. Each line is unique to one set of two devices (word). Only one $\overline{CE}_X$ may be active at a time.

Output-Enable ($\overline{OE}$) is the output control and should be used to gate data from a device to the output pins on the module, independent of device selection. One $\overline{OE}$ line serves the iSM001FLKA's flash devices. Figure 7 illustrates read timing waveforms.

When the V_{PP} lines are high (V_{PPH}), a read operation can be used to access array data, to output a device's int$_e$ligent identifier™ code, and to access a device's data for program/erase verification. When V_{PP} is low (V_{PPL}), a read operation can **only** access array data.

Output Disable

With the iSM001FLKA's Output-Enable pin at a logic-high level (V_{IH}), outputs from all devices are disabled. They are placed in a high-impedance state.

STANDBY

With Chip-Enable at a logic-high level, the standby operation disables most of the deselected devices circuitry and substantially reduces device power consumption. The outputs of the deselected devices are place in a high-impedance state, independent of the Output-Enable signal. If a word is deselected during erase, programming, or program/erasure verification, the device draws active current until the operation is terminated.

Intelligent Identifier Operation

The intelligent identifier operation outputs the selected devices' manufacturer code (89H) and device code (B4H). The manufacturer code and device code are read via the devices' command register. Following a write of 90H to a device's command register, a read from address location 0000H outputs the manufacture code (89H). A read from address 0001H outputs the device code (B4H).

Write

Erasure and programming is accomplished via each device's command register, when high voltage is applied to the V_{PP} pins. The contents of each device's register serve as input to its internal state-machine. The state machine outputs dictate the function of each device.

A device's command register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

Two write enable lines are provided, $\overline{WE}_H$ and $\overline{WE}_L$, allowing selective write control of upper and lower bytes.

A device's command register is written by selecting the device (Chip-Enable low), then bringing Write-Enable ($\overline{WE}_H$ or $\overline{WE}_L$) to a logic-low level (V_{IL}). If both WE lines are a logic low, both upper and lower bytes are written. Addresses are latched on the falling edge of the Write-Enable signal, while data is latched on the rising edge of the Write-Enable pulse. Standard microprocessor write timing are used.

Refer to AC Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the module's V_{PP} pins, the contents of all devices' command registers default to 00H, enabling read-only operations.

Placing high voltage on the module's V_{PP} pins allows read/write operation on selected devices. Operations are selected by writing specific data patterns to the device(s) command register. Table 5 defines these register commands.

Table 5. Command Definitions

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	1	Write	X	00H			
Read Intelligent Identifier Codes(4)	3	Write	X	90H	Read	(4)	(4)
Set-up Erase/Erase(5)	2	Write	X	20H	Write	X	20H
Erase Verify(5)	2	Write	EA	A0H	Read	X	EVD
Set-up Program/Program(6)	2	Write	X	40H	Write	PA	PD
Program Verify(6)	2	Write	X	C0H	Read	X	PVD
Reset(7)	2	Write	X	FFH	Write	X	FFH

NOTES:
1. Bus operations are defined in Table 4.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   PA = Address of memory location to be programmed.
   Addresses are latched on the falling edge of the Write-Enable pulse.
3. ID = Data read from location IA during device identification (Mfr = 89H, Device B4H).
   EVD = Data read from location EA during erase verify.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of Write-Enable.
   PVD = Data read from location PA during program verify. PA is latched on the Program command.
4. Following the Read intelligent ID command, two read operations access manufacturer and device codes.
5. Figure 4 illustrates the Quick-Erase™ Algorithm.
6. Figure 3 illustrates the Quick-Pulse Programming™ Algorithm.
7. The second bus cycle must be followed by the desired command register write.

Read Command

While V_{PP} is high, for erasure and programming, the selected devices memory contents can be accessed via the read command. The read operation is initiated by writing 00H into the command register of each device. Microprocessor read cycles retrieve array data. The selected devices remain enabled for reads until their command register contents are altered.

The default contents of each device's command register upon V_{PP} power-up is 00H. This default value ensures that no spurious alteration to the iSM001FLKA's memory contents occurs during the V_{PP} power transition. Where the V_{PP} supply is hardwired to the iSM001FLKA's V_{PP} pins, all eight devices power-up and remain enabled for reads until their command-register contents are changed. Refer to the AC Read Characteristics and Waveforms for specific timing parameters.

Intelligent Identifier Command

Flash memories are intended for use in applications where the local CPU alters memory contents. As such, manufacturer- and device-codes must be accessible while the device resides in the target system.

Each flash memory device contains an Intelligent identifier operation. The operation is initiated by writing 90H into the command register. Following the command write, a read cycle from address 0000H retrieves the manufacturer code of 89H. A read cycle from address 0001H returns the device code of B4H. To terminate the operation, it is necessary to write another valid command into the register.

The intelligent identifier and the Presence Detect pins give you complementary information. While the PD pins denote speed and depth, the intelligent identifier operation gives you manufacture and device data.

Set-Up Erase/Erase Commands

Set-up Erase is a command-only operation that stages a selected device for electrical erasure of all bytes in its array. The set-up erase operation is performed by writing 20H to the command register.

To commence chip-erasure, the erase command (20H) must again be written to the register. The erase operation begins with the rising edge of a Write-Enable pulse ($\overline{WE}_H$ or $\overline{WE}_L$) and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, chip-erasure can only occur when high voltage is applied to the V_{PP} pins. In the absence of this high voltage, memory contents are protected against erasure. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all bytes of the selected device(s) in parallel. After each erase operation, all bytes must be verified. The erase verify operation is initiated by writing A0H into the command register of the device. The address for the byte to be verified must be supplied as it is latched on the falling edge of a Write-Enable pulse. The register write terminates the erase operation with the rising edge of its Write-Enable pulse.

Each 28F010 applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased.

4

The erase-verify command must be written to the command register prior to each byte verification to latch its address. The process continues for each byte of the device until a byte does not return FFH data, or the last address is accessed.

In the case where the data read is not FFH, another erase operation is performed. (Refer to Set-up Erase/Erase). Verification then resumes from the address of the last-verified byte. Once all bytes of the device have been verified, the erase step is complete. The device can be programmed. At this point, the verify operation is terminated by writing a valid command (e.g., Program Set-up) to the command register of the device. Figure 4, the Quick-Erase™ algorithm, illustrates how commands and bus operations are combined to perform electrical erasure of each 28F010. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Set-Up Program/Program Commands

Set-up program is a command-only operation that stages a device for byte programming. Writing 40H into the command register of the device performs the set-up operation.

Once the program set-up operation is performed, the next Write-Enable pulse causes a transition to an active programming operation. Addresses are internally latched on the falling edge of the Write-Enable pulse. Data is internally latched on the rising edge of the Write-Enable pulse. The rising edge of Write-Enable also begins the programming operation. The programming operation terminates with the next rising edge of Write-Enable, used to write the program-verify command. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Program-Verify Command

Each 28F010 is programmed on a byte-by-byte basis. Byte programming may occur sequentially or at random. Following each programming operation, the byte just programmed must be verified.

The program-verify operation is initiated by writing C0H into the command register of the device. The register write terminates the programming operation with the rising edge of its Write-Enable pulse. The program-verify operation stages the device for verification of the byte last programmed. No new address information is latched.

Each 28F010 applies an internally-generated margin voltage to the byte. A microprocessor read cycle outputs the data. A successful comparison between the programmed byte and true data means that the byte is successfully programmed. Programming then proceeds to the next desired byte location. Figure 3, the Quick-Pulse Programming™ algorithm (8-bit Systems), illustrates how commands are combined with bus operations to perform byte programming. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or program-command sequences to a device. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

EXTENDED ERASE/PROGRAM CYCLING

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled—an expensive solution.

Intel has designed extended cycling capability into its ETOX II flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field is one-tenth that of common EEPROMs, minimizing the probability of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 MV/cm lower than EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure—increasing time to wearout by a factor of 100,000,000.

Each of the iSM001FLKA's eight 28F010s is specified for a minimum of 10,000 program/erase cycles. Each device is programmed and erased using Intel's Quick-Pulse Programming and Quick-Erase algorithms. Intel's algorithmic approach uses a series of operations (pulses), along with byte verification, to completely and reliably erase and program the device.

For further information, see Reliability Report RR-60 (ETOX II Reliability Data Summary).

QUICK-PULSE PROGRAMMING ALGORITHM

The Quick-Pulse Programming algorithm uses programming operations of 10 μs duration. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The algorithm allows for up to 25 programming operations per byte, although most bytes verify on the first or second operation. The entire sequence of programming and byte verification is performed with V_{PP} at high voltage. Figure 3 illustrates the Quick-Pulse Programming algorithm for 8-bit systems.

QUICK-ERASE ALGORITHM

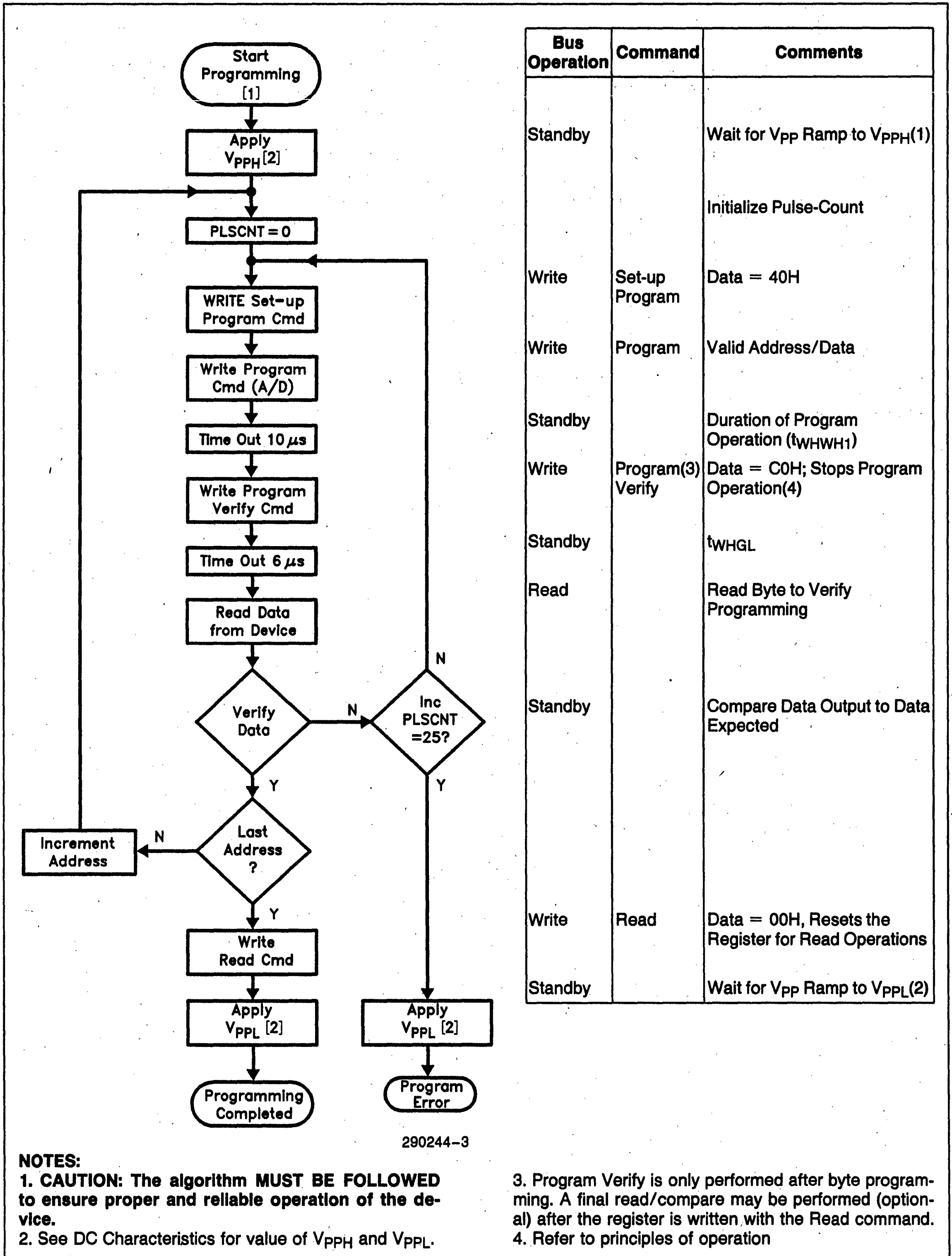
Intel's Quick-Erase algorithm yields fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the Quick-Pulse Programming™ algorithm, to simultaneously remove charge from all bits in the array.

Erasure begins with a read of memory contents. The iSM001FLKA is erased when shipped from the factory. Reading FFH data from each device would immediately be followed by device programming.

For devices being erased and reprogrammed, uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (Data = 00H). This is accomplished, using the Quick-Pulse Programming algorithm, in approximately two seconds.

Erase execution then continues with an initial erase operation. Erase verification (data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. With each erase operation, an increasing number of bytes verify to the erased state. Erase efficiency may be improved by storing the address of the last byte verified in a register. Following the next erase operation, verification starts at that stored address location. Erasure typically occurs in one second. Figure 4 illustrates the Quick-Erase algorithm for 8-bit systems.

4

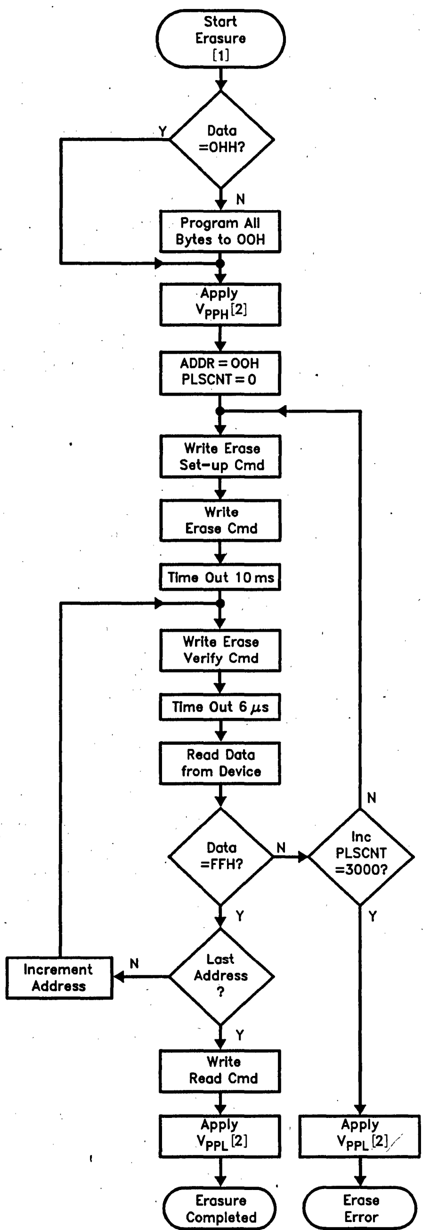


Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Pulse-Count
Write	Set-up Program	Data = 40H
Write	Program	Valid Address/Data
Standby		Duration of Program Operation (t_{WHWH1})
Write	Program(3) Verify	Data = C0H; Stops Program Operation(4)
Standby		t_{WHGL}
Read		Read Byte to Verify Programming
Standby		Compare Data Output to Data Expected
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(2)

NOTES:

1. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**
2. See DC Characteristics for value of V_{PPH} and V_{PPL}.
3. Program Verify is only performed after byte programming. A final read/compare may be performed (optional) after the register is written with the Read command.
4. Refer to principles of operation

Figure 3. Quick-Pulse Programming Algorithm (8-Bit Systems)



Bus Operation	Command	Comments
		Entire Memory Must = 00H Before Erasure
		Use Quick-Pulse Programming™ Algorithm (Figure 4)
Standby		Wait for V_{PP} Ramp to V_{PPH}(2)
		Initialize Addresses and Pulse-Count
Write	Set-up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase(3) Verify	Addr = Byte to Verify; Data = A0H; Stops Erase Operation(4)
Standby		t_{WHGL}
Read		Read Byte to Verify Erasure
Standby		Compare Output to FFH Increment Pulse-Count
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(2)

NOTES:

1. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**
2. See DC Characteristics for value of V_{PPH} and V_{PPL}.
3. Erase Verify is performed only after chip-erasure. A final read/compare may be performed (optional) after the register is written with the read command.
4. Refer to principles of operation.

Figure 4. Quick-Erase Algorithm (8-Bit Systems)

HIGH PERFORMANCE PARALLEL DEVICE ERASURE

Total erase time for the iSM001FLKA is reduced by implementing a parallel erase algorithm (Note 1). You save time by erasing all devices at the same time. However, since flash memories may erase at different rates, you must verify each device separately. This can be done in a word-wise fashion with the command register Reset command and a special masking algorithm.

Take for example the case of two-device (parallel) erasure. The CPU first writes the data word erase command 2020h twice in succession. This starts erasure. After 10 ms, the CPU writes the data word verify command A0A0h to stop erasure and setup erase verification. If both bytes are erased at the given address, then the CPU increments the address (by 2) and then writes the verify command A0A0h again. If neither byte is erased, then the CPU issues the erase sequence again without incrementing the address.

Suppose at the given address only the low byte verifies FFh data? Could the whole chip be erased? The answer is yes. Rather than check the rest of the low byte addresses independently of the high byte, simply use the reset command to mask the low byte from erasure and erase verification on the next erase loop. In this example the erase command would be 20FFh and the verify command would be A0FFh. Once the high byte verifies at that address, the CPU modifies the command back to the default 2020h and A0A0h, increments the address by 2, and writes the verify command to the next address.

See Figure 5 for a conceptual view of the parallel erase flow chart and Appendix B for the detailed version. These flow charts are for 16-bit systems and can be expanded for 32-bit designs.

NOTE:
1. Parallel Erasure and Programming require appropriate choice of V_{PP} supply to support the increased power consumption.

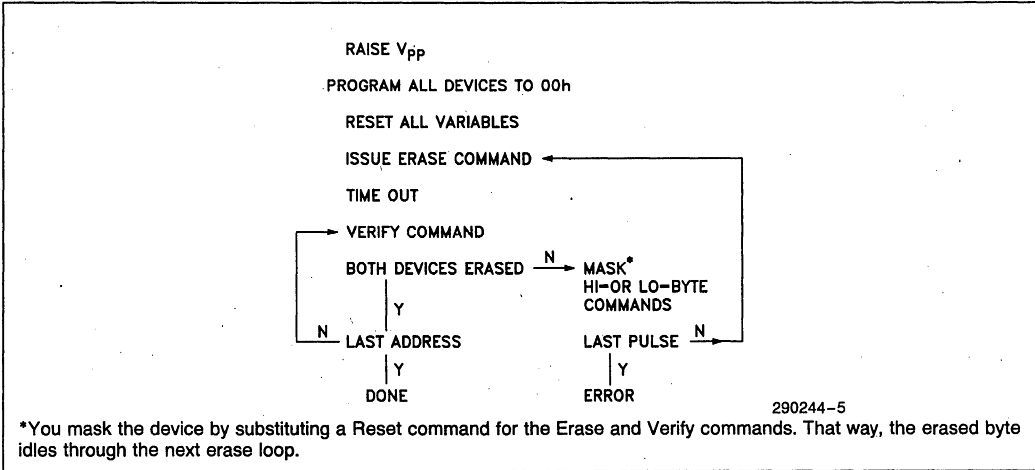
HIGH PERFORMANCE PARALLEL DEVICE PROGRAMMING

Software for word- or double-word programming can be written in two different manners. The first method offers simplicity of design and minimizes software overhead by using a byte programming routine on each device independently (using host CPU's byte addressing mode). The second method offers higher performance by programming the word or double-word data in parallel. This method manipulates the command register instructions for independent byte control. See Figure 6 for conceptual 2-device parallel programming flow chart and Appendix C for the detailed version. Here you can use the host CPU's appropriate word- or double-word addressing modes (i.e., incrementing by 2- or 4-byte addresses, respectively).

NOTE:
Word or double-word programming assumes 2 or 4 8-bit flash memory devices.

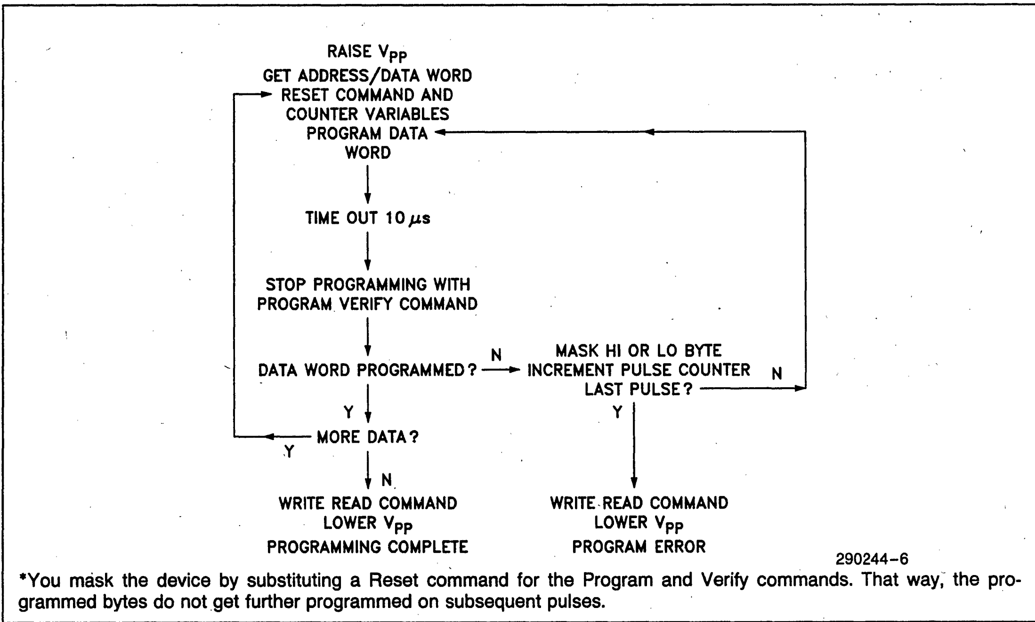
Parallel Programming Algorithm Summary:

- Decreases programming time by programming 2 flash memories (16 bits) in parallel. The algorithm can be expanded for 32-bit systems.
- Eliminates tracking of high/low byte addresses and respective number of program pulses by directing the CPU to write data-words (16-bit) to the command register.
- Maintains word write and read operations. Should a byte on one device program prior to a byte on the other, the CPU continues to write word-commands to both devices. However, it deselects the verified byte with software commands. An alternative is to independently program high and low bytes using hardware select capability (byte-addressing mode of host CPU).



*You mask the device by substituting a Reset command for the Erase and Verify commands. That way, the erased byte idles through the next erase loop.

Figure 5. High Performance Parallel Erasure (Conceptual Overview)



*You mask the device by substituting a Reset command for the Program and Verify commands. That way, the programmed bytes do not get further programmed on subsequent pulses.

Figure 6. Parallel Programming Flow Chart (Conceptual Overview)

4

DESIGN CONSIDERATIONS

Two-Line Output Control

Two-line control provides for:

a. the lowest possible memory power dissipation and,

b. complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address-decoder output should drive chip-enable, while the system's read signal controls all flash-memories and other parallel memories. This assures that only enabled memory devices have active outputs, while deselected devices maintain the low power standby condition.

Power Supply Decoupling

Flash-memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current (I_{CC}) issues—standby, active, and transient current peaks produced by falling and rising edges of chip-enable. The capacitive and inductive loads on the device outputs determine the magnitudes of these peaks.

Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. The iSM001FLKA features a 0.1 μF ceramic capacitor connected between V_{CC} and V_{SS}, and between V_{PP} and V_{SS}.

Also, a 4.7 μF tantalum capacitor decouples the array's power supply between V_{CC} and V_{SS} and between V_{PP} and V_{SS}. The bulk capacitors will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

V_{PP} Trace on Printed Circuit Boards

Programming flash memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the V_{PP} power supply trace. The two V_{PP} pins supply current for programming. Use similar trace widths and layout considerations given the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots. Be sure to connect both module V_{PP} inputs to your 12V supply.

Power Up/Down Protection

The iSM001FLKA is designed to offer protection against accidental erasure or programming during power transitions. Upon power-up, each 28F010 is indifferent as to which power supply, V_{PP} or V_{CC}, powers up first. Power supply sequencing is not required. Internal circuitry in each 28F010 ensures that the command register is reset to the read mode on power up.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will inhibit writes. The control register architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases the usable battery life of your system because each 28F010 does not consume any power to retain code or data when the system is off. Table 4 illustrates the power dissipated when updating each 28F010.

Table 4. 28F010 Typical Update Power Dissipation(4)

Operation	Power Dissipation (Watt-Seconds)
Array Program/Program Verify(1)	0.171
Array Erase/Erase Verify(2)	0.136
One Complete Cycle(3)	0.478

NOTES:

1. Formula to calculate typical Program/Program Verify Power = $[V_{PP} \times \# \text{ Bytes} \times \text{typical } \# \text{ Prog Pulses } (t_{WHWH1} \times I_{PP2} \text{ typical} + t_{WHGL} \times I_{PP4} \text{ typical})] + [V_{CC} \times \# \text{ Bytes} \times \text{typical } \# \text{ Prog Pulses } (t_{WHWH1} \times I_{CC2} \text{ typical} + t_{WHGL} \times I_{CC4} \text{ typical}]$.
2. Formula to calculate typical Erase/Erase Verify Power = $[V_{PP} (V_{PP3} \text{ typical} \times t_{ERASE} \text{ typical} + I_{PP5} \text{ typical} \times t_{WHGL} \times \# \text{ Bytes})] + [V_{CC} (I_{CC3} \text{ typical} \times t_{ERASE} \text{ typical} + I_{CC5} \text{ typical} \times t_{WHGL} \times \# \text{ Bytes})]$.
3. One Complete Cycle = Array Preprogram + Array Erase + Program.
4. "Typicals are not guaranteed but based on a limited number of samples taken from production lots.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read ....0°C to +70°C(1)
During Erase/Program ....0°C to +70°C

Temperature Under Bias ....−10°C to +80°C

Storage Temperature ....−50°C to +100°C

Voltage on Any Pin with
Respect to Ground ....−2.0V to +7.0V(2)

V_{PP} Supply Voltage with
Respect to Ground
During Erase/Program ....−2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage with
Respect to Ground ....−2.0V to +7.0V(2)

Output Short Circuit Current.............100 mA(4)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
3. Maximum DC voltage on V_{PP} may overshoot to +14.0V for periods less than 20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature	0	70	°C	For Read-Only and Read/Write Operations
V_{CC}	V_{CC} Supply Voltage	4.50	5.50	V	

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ	Max		
I_{LI}	Input Leakage Current	3			±8.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	3			±40.0	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1, 3			8.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	2, 3		26	66	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	2, 3		8.0	26	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	2, 3		16.0	36	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	2, 3		16.0	36	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	2, 3		16.0	36	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	3			±80	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	3		0.7	1.6	mA	$V_{PP} > V_{CC}$
					±80	μA	$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	2, 3		16.5	61.2	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	2, 3		12.5	61.2	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	2, 3		4.5	11.2	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	2, 3		4.5	11.2	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	I_{OH} = −2.5 mA $V_{CC} = V_{CC}$ Min
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

NOTES:
1. V_{CC} standby current for 8 devices.
2. Calculations assume only the 2 devices of the 16-bit word are enabled. The remaining 6 devices are in standby. Current will be higher if interleaving is used.
3. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C. These currents are valid for all product versions (packages and speeds).

DC CHARACTERISTICS—CMOS COMPATIBLE

Symbol	Parameter	Notes	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
I_{LI}	Input Leakage Current	3			±8.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	3			±40.0	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1, 3		0.4	0.8	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC} \pm 0.2V$
I_{CC1}	V_{CC} Active Read Current	2, 3		20.3	60.6	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	2, 3		2.3	20.6	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	2, 3		10.3	30.6	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	2, 3		10.3	30.6	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	2, 3		10.3	30.6	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current				±80	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	3		0.7	1.6	mA	$V_{PP} > V_{CC}$
					±80	μA	$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	2, 3		16.5	61.2	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	2, 3		12.5	61.2	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	2, 3		4.5	11.2	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	2, 3		4.5	11.2	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		0.85 V_{CC}			V	I_{OH} = −2.5 mA, $V_{CC} = V_{CC}$ Min
V_{OH2}			V_{CC} − 0.4				I_{OH} = −100 μA, $V_{CC} = V_{CC}$ Min
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

NOTES:
1. V_{CC} standby current for 8 devices.
2. Calculations assume only the 2 devices of the 16-bit word are enabled. The remaining 6 devices are in standby. Current will be higher if interleaving is used.
3. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C. These currents are valid for all product versions (packages and speeds).

CAPACITANCE(1) $T_A = 25°C, f = 1.0 MHz$

Symbol	Parameter	Notes	Limits		Unit	Conditions
			Min	Max		
C_{IN1}	Address Capacitance	2		60	pF	$V_{IN} = 0V$
C_{IN2}	Control Capacitance	2		65	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	2		55	pF	$V_{OUT} = 0V$

NOTES:
1. Trace capacitance calculated, not measured.
2. Address and control capacitance of a typical device is 6 pF.
3. Output capacitance of a typical device is 12 pF.

AC TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ...... 10 ns
Input Pulse Levels .... 0.45V and 2.4V
Input Timing Reference Level .... 0.8V and 2.0V
Output Timing Reference Level ...... 0.8V and 2.0V

AC CHARACTERISTICS—Read-Only Operations(2)

Versions		Notes	ISM001FLKA-120		ISM001FLKA-200		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}/t_{RC}	Read Cycle Time	3	120		200		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time			120		200	ns
t_{AVQV}/t_{ACC}	Address Access Time			120		200	ns
t_{GLQV}/t_{OE}	Output Enable Access Time			50		60	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	3	0		0		ns
t_{EHQZ}	Chip Disable to Output in High Z	3		55		55	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	3	0		0		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	4		30		40	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	3	0		0		ns
t_{WHGL}	Write Recovery Time before Read		6		6		µs

NOTES:
1. Whichever occurs first.
2. Rise/Fall Time ≤ 10 ns.
3. Not 100% tested: Characterization data available.
4. Guaranteed by design.

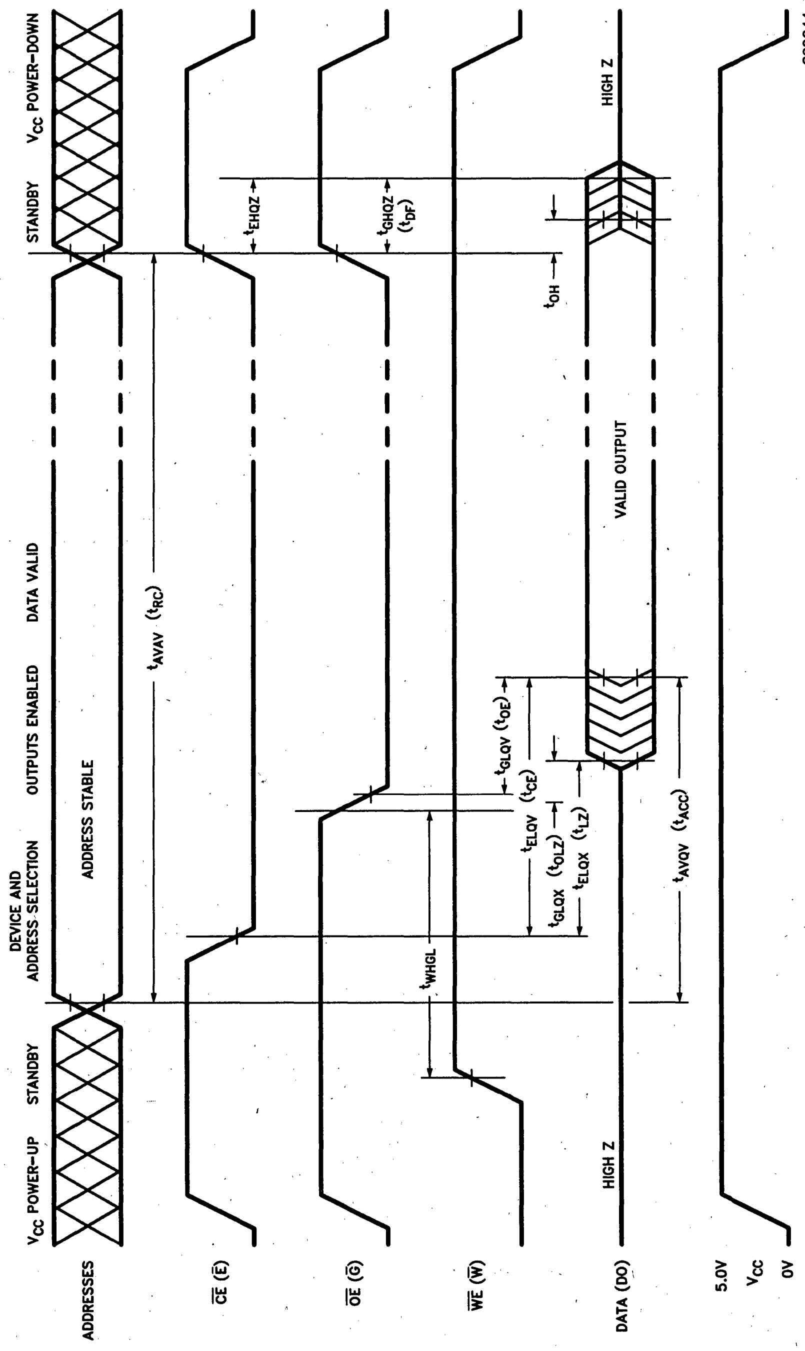


Figure 7. AC Waveforms for Read Operations

AC CHARACTERISTICS—Write/Erase/Program Operations(1, 2)

Versions			ISM001FLKA-120		ISM001FLKA-200		Unit
Symbol	Characteristic	Notes	Min	Max	Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		120		200		ns
t_{AVWL}/t_{AS}	Address Set-up Time		0		0		ns
t_{WLAX}/t_{AH}	Address Hold Time		60		75		ns
t_{DVWH}/t_{DS}	Data Set-up Time		50		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		10		ns
t_{WHGL}	Write Recovery Time before Read		6		6		μs
t_{GHWL}	Read Recovery Time before Write		0		0		μs
t_{ELWL}/t_{CS}	Chip Enable Set-up Time before Write		20		20		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time		0		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width	2	60		60		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		20		ns
t_{WHWH1}	Duration of Programming Operation	3	10		10		μs
t_{WHWH2}	Duration of Erase Operation	3	9.5		9.5		ms
t_{VPEL}	V_{PP} Set-up Time to Chip Enable Low		1.0		1.0		μs

NOTES:
1. Read timing characteristics during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Rise/Fall time ≤ 10 ns.
3. The integrated stop timer terminates the program/erase operations, thereby eliminating the need for a maximum specification.

ERASE AND PROGRAMMING PERFORMANCE

Parameter	Notes	Limits						Unit
		28F010-120			28F010-200			
		Min	Typ	Max	Min	Typ	Max	
Chip Erase Time	1, 3, 4		1	10		1	30	Sec
Chip Program Time	1, 2, 4		2	12.5		2	12.5	Sec
Erase/Program Cycles	1, 5	10,000	100,000		10,000	100,000		Cycles

NOTES:
1. Typicals are not guaranteed, but based on a limited number of samples from production lots. Data taken at 25°C, 12.0V V_{PP}.
2. Minimum byte programming time excluding system overhead is 16 μs (10 μs program + 6 μs write recovery), while maximum is 400 μs/byte (16 μs x 25 loops allowed by algorithm). Max chip programming time is specified lower than the worst case allowed by the programming algorithm since most bytes program significantly faster than the worst case byte.
3. Excludes 00H Programming prior to Erasure.
4. Excludes System-Level Overhead.
5. Refer to RR-60 "ETOX™II Flash Memory Reliability Data Summary" for typical cycling data and failure rate calculations.

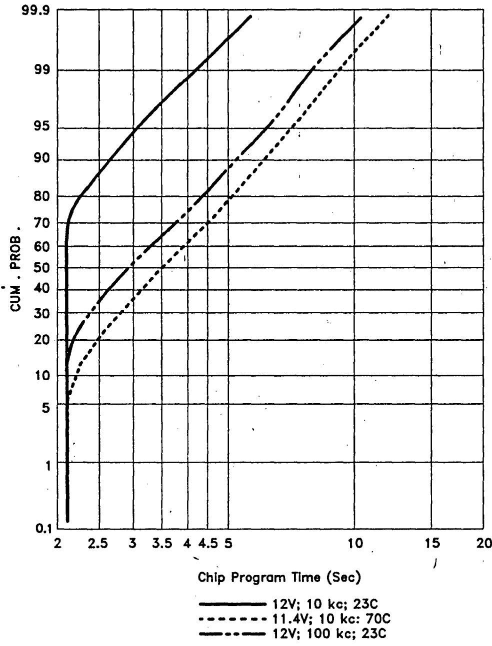


Figure 8. 28F010 Typical Programming Capability

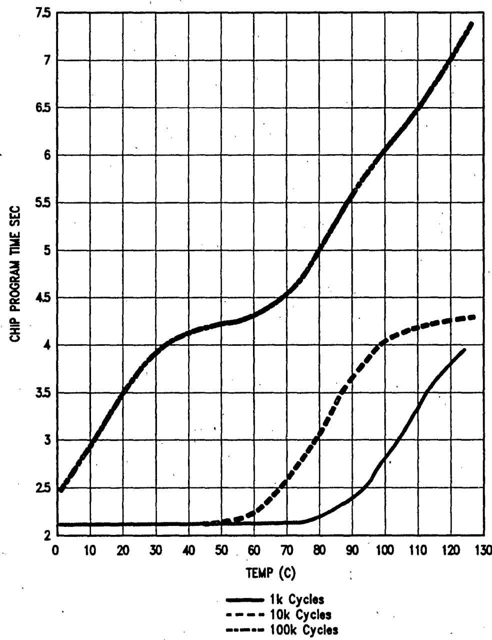


Figure 9. 28F010 Typical Program Time at 12V

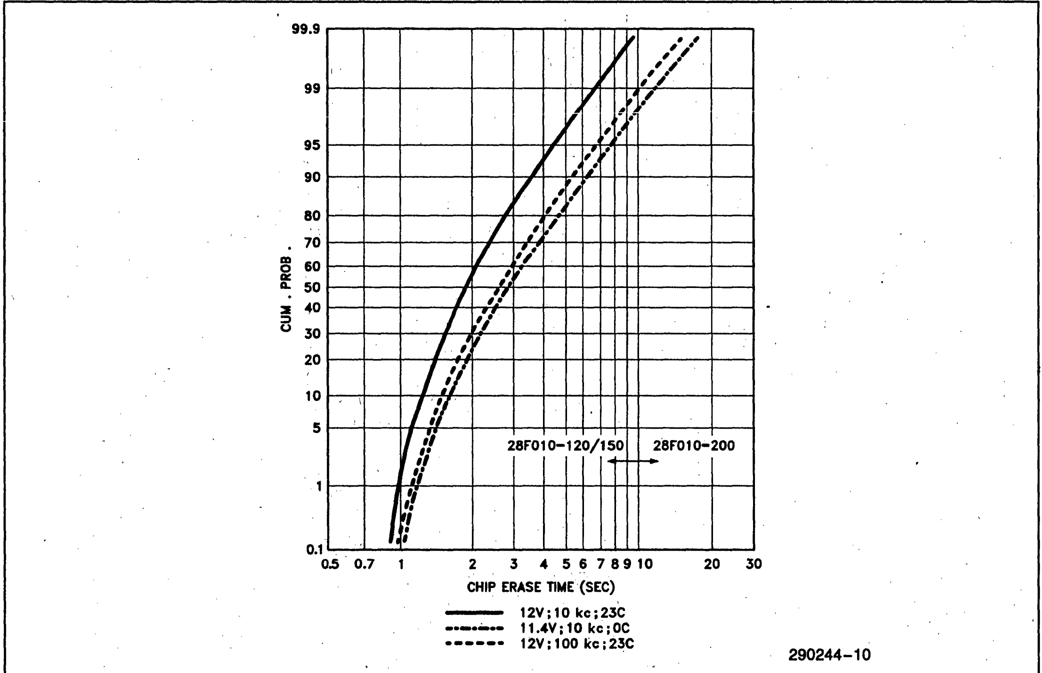


Figure 10. 28F010 Typical Erase Capability

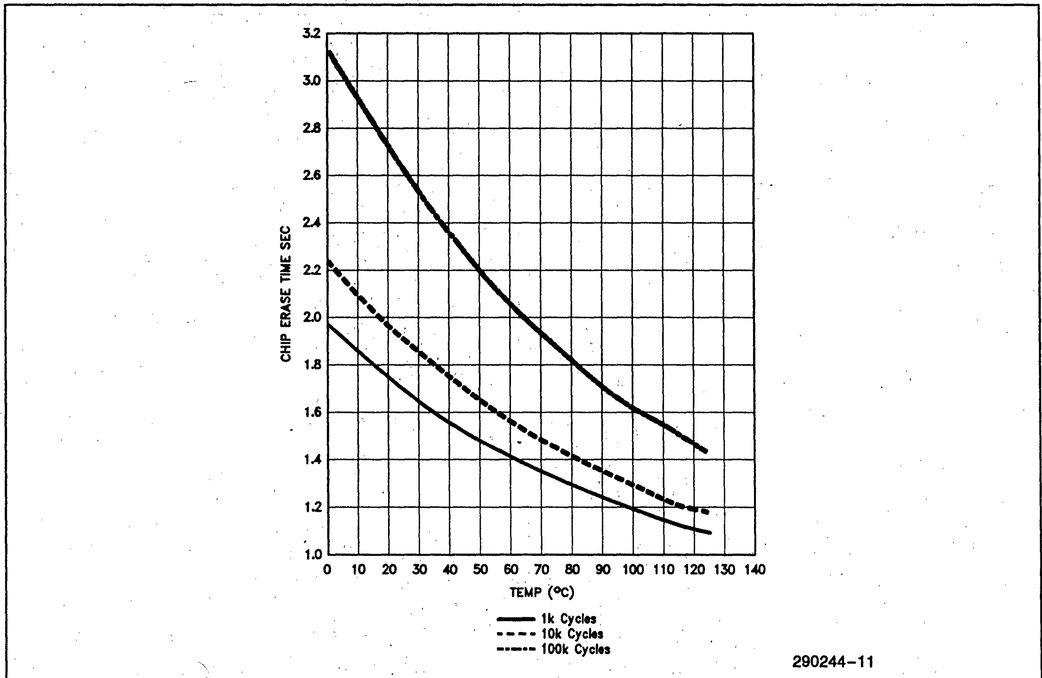


Figure 11. 28F010 Typical Erase Time at 12.0V

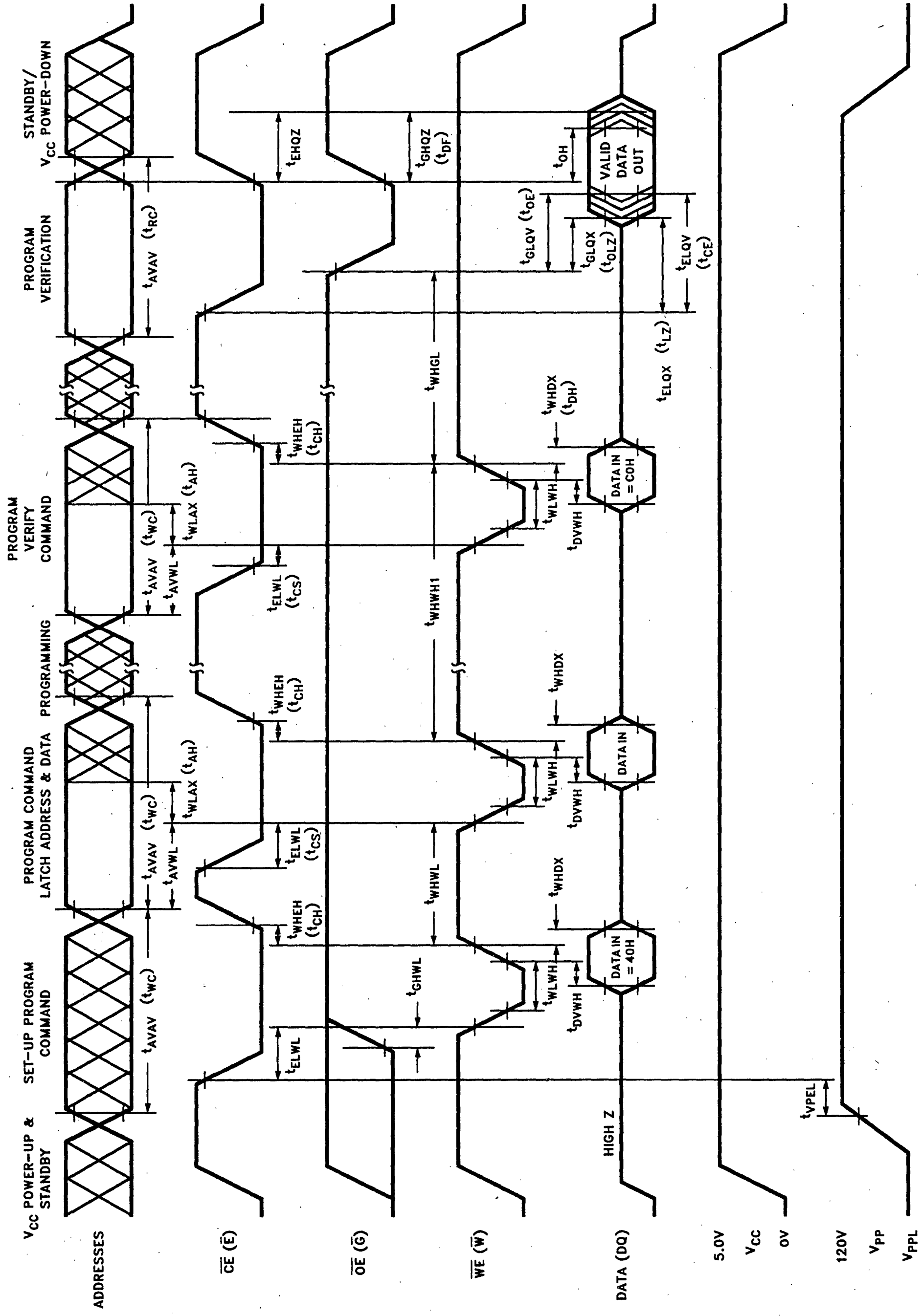


Figure 12. AC Waveforms for Programming Operations

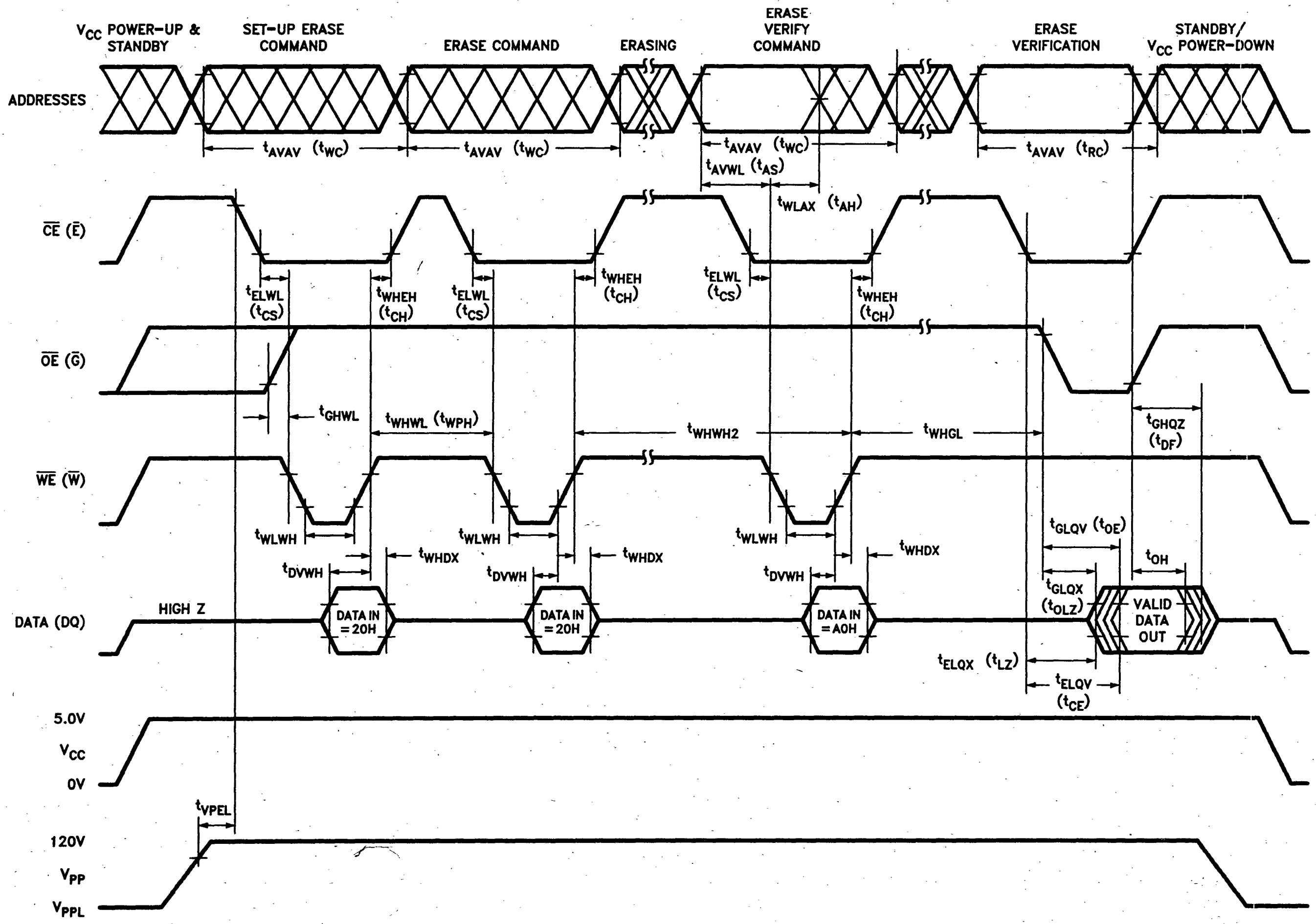
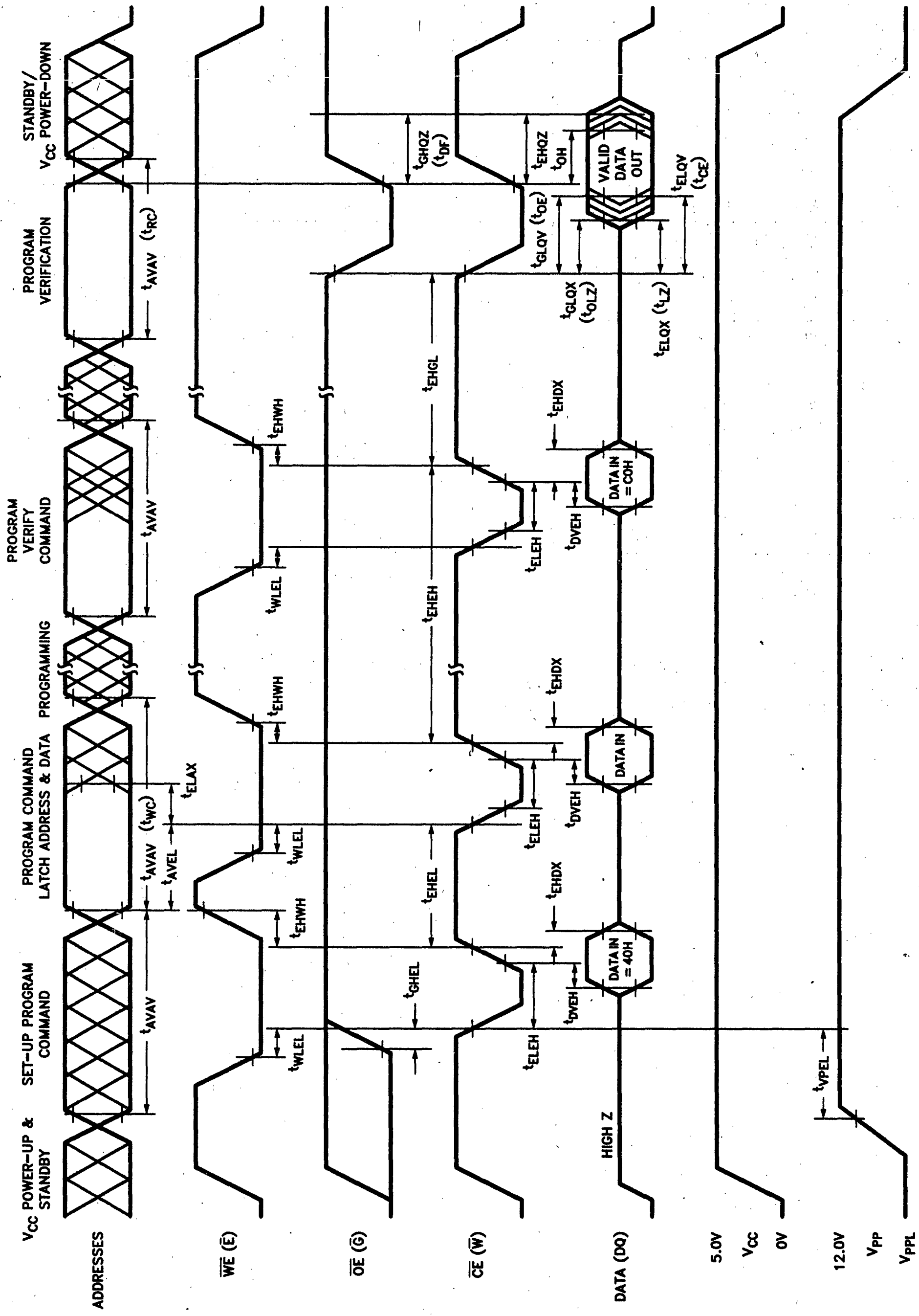


Figure 13. AC Waveforms for Erase Operations

ALTERNATIVE $\overline{CE}$-CONTROLLED WRITES

Versions			28F010-120		28F010-200		Unit
Symbol	Characteristic	Notes	Min	Max	Min	Max	
t_{AVAV}	Write Cycle Time		120		200		ns
t_{AVEL}	Address Set-Up Time		0		0		ns
t_{ELAX}	Address Hold Time		80		95		ns
t_{DVEH}	Data Set-Up Time		50		50		ns
t_{EHDX}	Data Hold Time		10		10		ns
t_{EHGL}	Write Recovery Time before Read		6		6		μs
t_{GHEL}	Read Recovery Time before Write		0		0		μs
t_{WLEL}	Write Enable Set-Up Time before Chip Enable		0		0		ns
t_{EHWH}	Write Enable Hold Time		0		0		ns
t_{ELEH}	Write Pulse Width	1	70		80		ns
t_{EHEL}	Write Pulse Width High		20		20		ns
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low		1.0		1.0		μs

NOTE:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of Chip-Enable and Write-Enable. In systems where Chip-Enable defines the write pulse width (within a longer Write-Enable timing waveform) all set-up, hold and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.



NOTE:
Alternative $\overline{CE}$-Controlled Write Timings also apply to erase operations.

Figure 14. Alternate AC Waveforms for Programming Operations

APPENDIX A
PARTIAL LIST(1) OF 80-PIN SIMM SOCKET COMPANIES

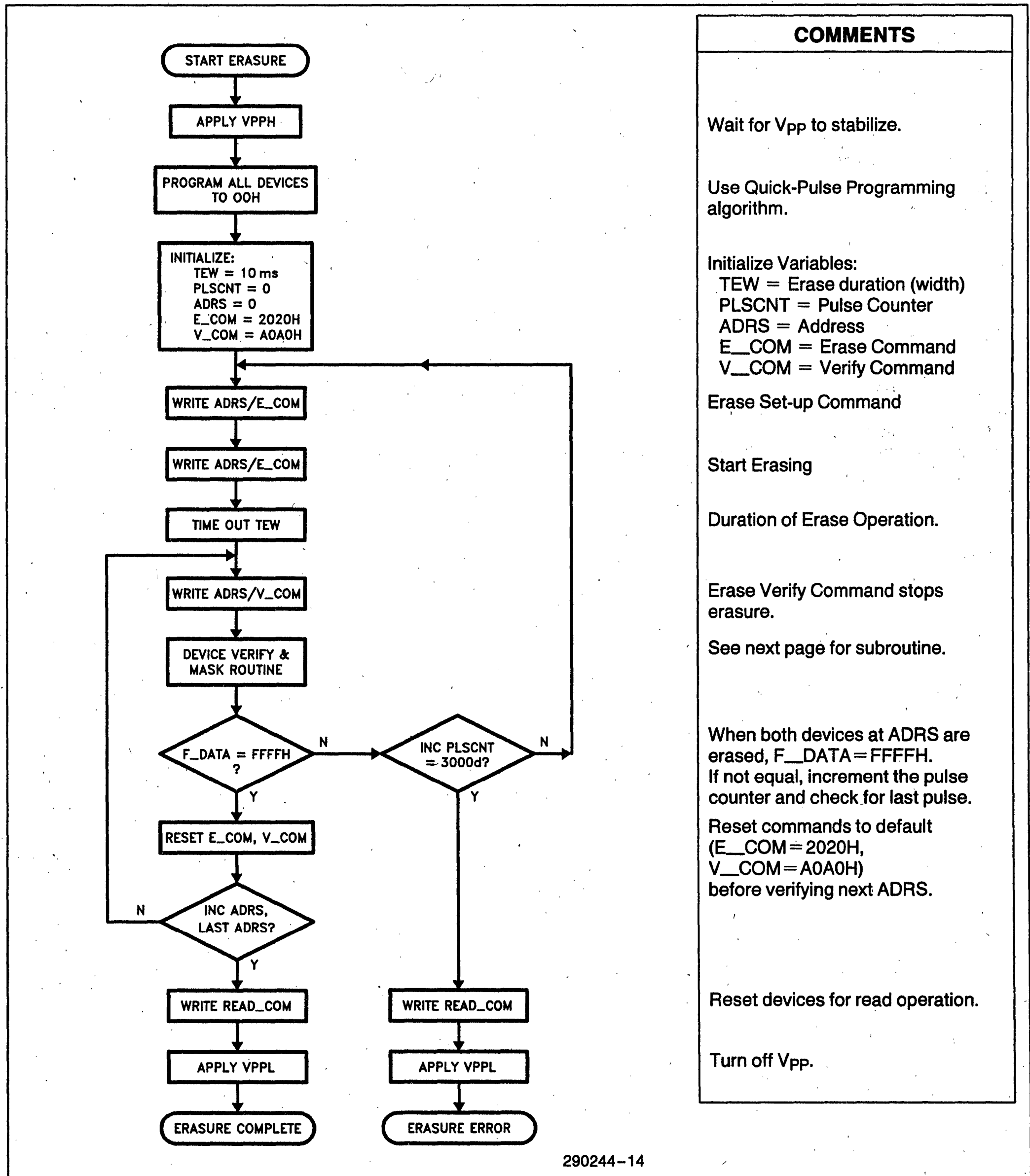
AMP INCORPORATED
HARRISBURG, PA 17105
(800) 522-6752

BURNDY CORPORATION
51 RICHARDS AVENUE
NORWALK, CT 06856
(203) 838-4444

MOLEX
2222 WELLINGTON COURT
LISLE, IL 60532
(708) 969-4550

NOTES:
1. This list is intended for example only, and in no way represents all companies that support 80-pin SIMM Sockets. Intel Corporation assumes no responsibility for circuitry other than circuitry embodies in an Intel product. No other circuit patent licenses are implied.
2. Socket reliability data can be obtained from the above companies upon request.

APPENDIX B
PARALLEL ERASE FLOW CHART



290244–14

Device Erase Verify and Mask Subroutine

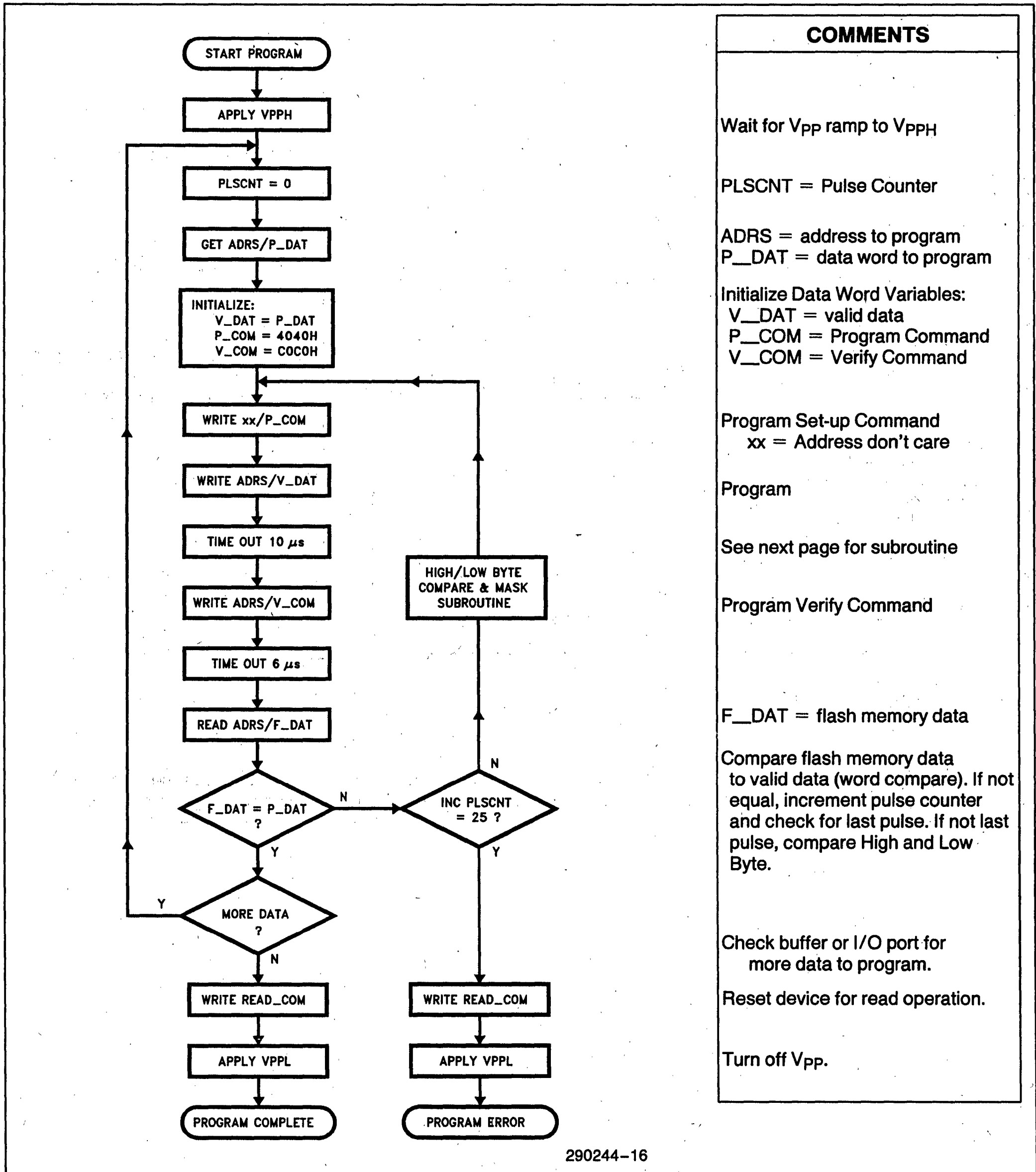
START SUBROUTINE → TIME OUT 6 μs → READ ADRS/F_DATA → F_DATA = FFFFH (Y: to SUBROUTINE RETURN; N: continue) → LO_BYTE = (F_DATA AND 00FFH) → LO_BYTE = FFH ? (Y: E_COM = (E_COM OR 00FFH), V_COM = (V_COM OR 00FFH); N: continue) → HI_BYTE = (F_DATA AND FF00H) → HI_BYTE = FF00H ? (Y: E_COM = (E_COM OR FF00H), V_COM = (V_COM OR FF00H); N: continue) → SUBROUTINE RETURN

290244–15

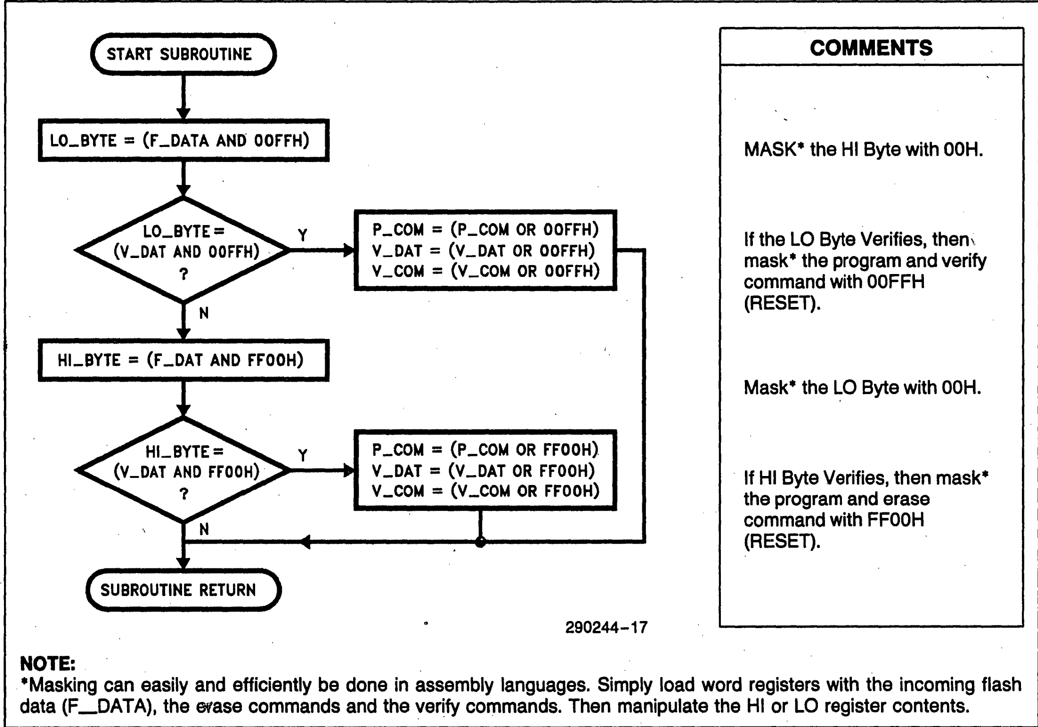
COMMENTS
This subroutine reads the data word (F__DATA). It then masks the HI or LO Byte of the Erase and Verify commands from executing during the next operation.
If both HI and LO bytes verify, then return.
Mask* the HI Byte with 00H.
If the LO Byte verifies erasure, then mask* the erase and verify commands with 00FFH(Reset).
Mask* the LO Byte with 00H.
If the HI Byte verifies erasure, then mask* the erase and verify commands with FF00H(Reset).

NOTE:
*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming flash data (F__DATA), the erase commands and the verify commands. Then manipulate the HI or LO register contents.

APPENDIX C
PARALLEL PROGRAMMING FLOW CHART



Program Verify and Mask Subroutine

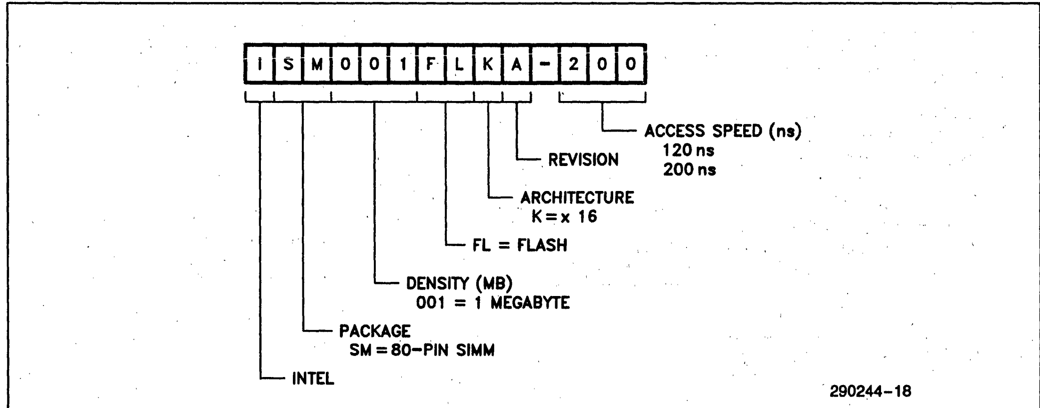


290244-17

NOTE:
*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming flash data (F_DATA), the erase commands and the verify commands. Then manipulate the HI or LO register contents.

4

Ordering Information



Valid Combinations:

iSM001FLKA-120
iSM001FLKA-200

ADDITIONAL INFORMATION

	Order Number
ER-20, "ETOX™ II Flash Memory Technology"	294005
ER-24, "The Intel 28F010 Flash Memory"	294008
RR-60, "ETOX™ II Flash Memory Reliability Data Summary"	293002
AP-316, "Using Flash Memory for In-System Reprogrammable Nonvolatile Storage"	292046
AP-325, "Guide to Flash Memory Reprogramming"	292059
AP-343, "Flash Memory — A Mass Storage Medium"	292079

REVISION HISTORY

Number	Description
-006	Correct Name change Change order number

iSM002FLKA
2 MBYTE (1024K x 16) CMOS FLASH SIMM

- **High-Performance**
  - **150 ns Maximum Access Time**
  - **13.3 MB/s Read Transfer Rate**
- **10,000 Rewrite Cycles Minimum/ Component**
- **Flash Electrical Chip-Erase**
  - **2 Second Typical Chip-Erase**
- **16 μs Typical Word Write**
  - **Up to 1 Mb/s Write Transfer Rate**
- **Inherent Non-volatility**
  - **No Batteries or Disk Required for Back-up**
  - **0W Data Retention Power**
- **CMOS Low Power Consumption**
  - **20.3 mA Typical Active Current**
  - **0.4 mA Typical Standby Current**
- **Standard 80-Pin Insertable Module**
  - **0.050 Centerline Lead Spacing**
  - **Upgrade Path through 128M bytes**
- **Hardware Presence Detect**
- **Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface**
- **Noise Immunity Features**
  - **±10% V_{CC} Tolerance**
  - **Maximum Latch-Up Immunity Through EPI Processing**
- **12.0V ±5% V_{PP}**
- **Integrated Program/Erase Stop Timer**
- **ETOX™ II Nonvolatile Flash Technology**
  - **High-Volume Manufacturing Experience**

Intel's iSM002FLKA flash SIMM (Single In-Line Memory Module) is targeted at high-density read/write nonvolatile memory. The iSM002FLKA enables you to optimize board space; to offer incremental memory expansion similar to today's DRAM; and to assure continued access to today's and tomorrow's surface-mount technologies. Intel's iSM002FLKA offers a reliable sold-state alternative for mass storage. The flash memory module is also ideal for high performance code and data storage as well as data recording and accumulation.

The iSM002FLKA, composed of eight 2 Mb flash memories in plastic leaded chip carrier (N28F020), is organized as 1,048,576 words of 16 bits. The PLCCs are mounted, four to a side, together with 0.1 μF decoupling capacitors on an 80-pin standard, low-profile module.

Extended erase and program cycling capability is designed into Intel's ETOX™ II (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional nonvolatile memory.

Intel's iSM002FLKA flash SIMM employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 150 ns access time provides no WAIT state performance for a wide range of microprocessors and microcontrollers. Maximum standby current of 0.8 mA translates into power savings when the memory module is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from $-1V$ to $V_{CC} + 1V$.

October 1992
Order Number: 290465-001

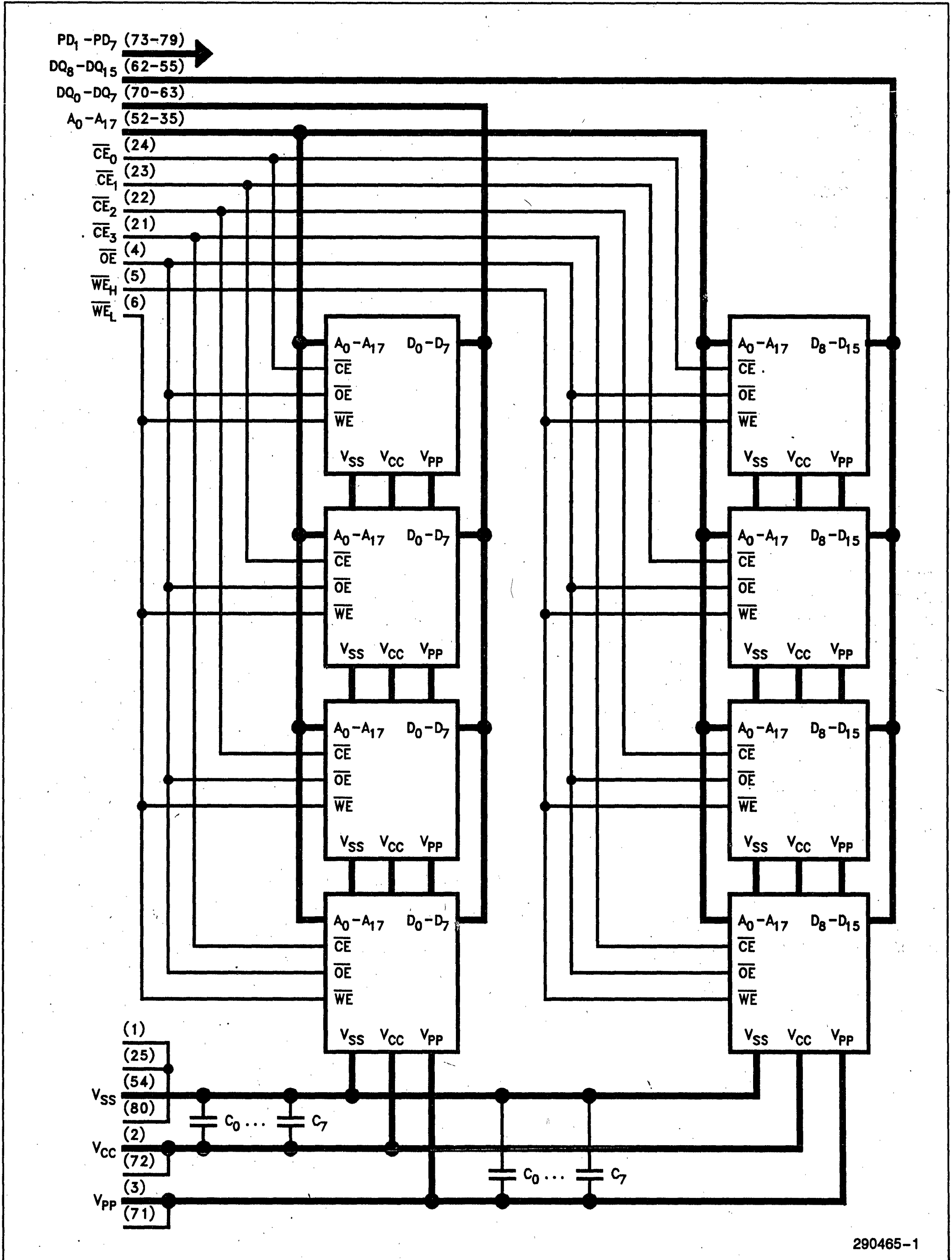


Figure 1. iSM002FLKA Functional Block Diagram

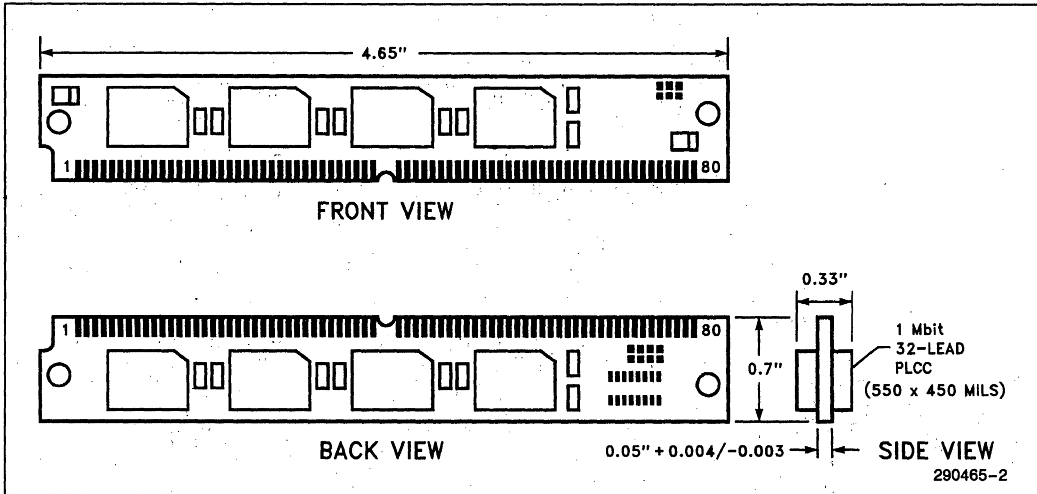


Figure 2. ISM002FLKA Pin Configurations

Table 1. Pinout

1	V_{SS}	21	$\overline{CE3}$	41	A_{11}	61	DQ_9
2	V_{CC}	22	$\overline{CE2}$	42	A_{10}	62	DQ_8
3	V_{PP}	23	$\overline{CE1}$	43	A_9	63	DQ_7
4	$\overline{OE}$	24	$\overline{CE0}$	44	A_8	64	DQ_6
5	$\overline{WEH}$	25	V_{SS}	45	A_7	65	DQ_5
6	$\overline{WEL}$	26	RES	46	A_6	66	DQ_4
7	NC	27	RES	47	A_5	67	DQ_3
8	RES	28	RES	48	A_4	68	DQ_2
9	RES	29	RES	49	A_3	69	DQ_1
10	RES	30	NC	50	A_2	70	DQ_0
11	RES	31	NC	51	A_1	71	V_{PP}
12	RES	32	NC	52	A_0	72	V_{CC}
13	RES	33	NC	53	RES	73	PD_1
14	RES	34	NC	54	V_{SS}	74	PD_2
15	RES	35	A_{17}	55	DQ_{15}	75	PD_3
16	RES	36	A_{16}	56	DQ_{14}	76	PD_4
17	NC	37	A_{15}	57	DQ_{13}	77	PD_5
18	NC	38	A_{14}	58	DQ_{12}	78	PD_6
19	NC	39	A_{13}	59	DQ_{11}	79	PD_7
20	NC	40	A_{12}	60	DQ_{10}	80	V_{SS}

Table 2. Pin Description

Symbol	Type	Name and Function
A_0–A_{17}	INPUT	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
DQ_0–DQ_{15}	INPUT/ OUTPUT	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}_0$–$\overline{CE}_3$	INPUT	**CHIP ENABLE:** Activates each device's control logic, input buffers, decoders, and sense amplifiers. Each line is unique to one set of 2 devices (word). $\overline{CE}_X$ is active low; $\overline{CE}_X$ high deselects the memory device and reduces power consumption to standby levels. Only one $\overline{CE}_X$ may be active at a time.
$\overline{OE}$	INPUT	**OUTPUT ENABLE:** Gates the devices outputs through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}_H$; $\overline{WE}_L$	INPUT	**WRITE ENABLE** controls writes to the control register and the array. ($\overline{WE}_H$ = High Byte; $\overline{WE}_L$ = Low Byte) Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **NOTE:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP}		**ERASE/PROGRAM POWER SUPPLY** for writing the command register, erasing the entire array, or programming bytes in the array (12V ± 5%).
V_{CC}		**DEVICE POWER SUPPLY:** (5V ± 10%).
V_{SS}		**GROUND.**
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating. (see pin table below)
RES		**RESERVED** for future product enhancements.
PD_1–PD_7		**PRESENCE DETECT:** Denotes word depth (512K) and access time of device. See Table 3, "Presence Detect "PD" Pins" on Page 5.

Pin	Function
17	$\overline{CE}_7$
18	$\overline{CE}_6$
19	$\overline{CE}_5$
20	$\overline{CE}_4$
30	A_{22}
31	A_{21}
32	A_{20}
33	A_{19}
34	A_{18}

Table 3. Presence Detect "PD" Pins

MODULE CAPACITY IDENTIFICATION			
MODULE CAPACITY WORD DEPTH	PD6	PD2	PD1
NO MODULE	O	O	O
256K/32M	O	O	S
512K/64M	O	S	O
1M/128M	O	S	S
2M/256M	S	O	O
4M/512M	S	O	S
8M/1G	S	S	O
16M/2G	S	S	S

MODULE SPEED IDENTIFICATION				
MAXIMUM ACCESS TIME	PD7	PD5	PD4	PD3
>300 ns	S	S	S	S
300 ns	S	S	S	O
250 ns	S	S	O	S
200 ns	S	S	O	O
185 ns	S	O	S	S
150 ns	S	O	S	O
135 ns	S	O	O	S
120 ns	S	O	O	O
100 ns	O	S	S	S
85 ns	O	S	S	O
70 ns	O	S	O	S
60 ns	O	S	O	O
50 ns	O	O	S	S
40 ns	O	O	S	O
30 ns	O	O	O	S
ND	O	O	O	O

O = OPEN CIRCUIT ON MODULE
S = SHORT CIRCUIT TO GROUND ON MODULE
ND = NOT DEFINED

4

SINGLE IN-LINE MEMORY MODULE BOARD

PC substrate: Glass Epoxy [0.05″ +0.004/−0.003 nominal thickness]. The iSM002FLKA low-profile SIMM mounts easily between expansion slots. See Appendix A for a list of 80-pin socket suppliers.

APPLICATIONS

With high density, nonvolatility, and extended cycling capability, Intel's iSM002FLKA flash SIMMs offer an innovative alternative to disk and battery-backed static RAM.

Primary applications and operating systems can be stored in flash, eliminating the slow disk-to-DRAM download process. Performance is dramatically enhanced and power consumption is reduced—a consideration particularly important in portable equipment. Flexibility is increased with Flash's electrical chip erasure allowing in-system updates to operating systems and application code.

In diskless workstations and terminals, network traffic is reduced to a minimum and systems are instant-on. Reliability exceeds that of electro-mechanical media. Often in these environments, power glitches force extended re-boot periods for all networked terminals. This mishap is no longer an issue if boot code, operating systems, communication protocols and primary applications are flash-resident in each terminal.

For embedded systems that rely on dynamic RAM/disk for main system memory or nonvolatile backup storage, Flash SIMMs provide a solid state alternative in a minimal form factor. Flash memory provides higher performance, lower power consumption and instant-on capability. Additionally, flash is more rugged and reliable in harsh environments where extreme temperatures and shock can cause disk-based systems to fail.

For systems currently using a high-density static RAM/battery configuration for code updates and data accumulation, flash memory's inherent nonvolatility eliminates the need for battery backup. The possibility of battery failure is removed. This consideration is important for portable equipment and medical instruments, both requiring continuous performance. In addition, flash memory offers a four-to-one cost advantage over SRAM.

Flash memory's electrical chip erasure, byte reprogrammability and complete nonvolatility fit well with data accumulation and recording needs. Electrical chip-erasure gives the designer a "blank-slate" in which to log or record data. Data can be periodically off-loaded for analysis-erasing the slate and repeating the cycle.

Flash SIMMs add additional flexibility to designers by offering end-users incremental expansion memory. As code requirements grow or as memory prices drop, your customers have the option of adding more memory.

PRINCIPALS OF OPERATION

The iSM002FLKA operates as eight N28F020 flash memories connected as shown in the Functional Block Diagram on Page 2.

The iSM002FLKA, organized as 1024K x 16, can also be configured for 8- and 32-bit systems. For 32-bit systems, add a second SIMM to your design as currently done with DRAM. For byte-wide operation, buffer the SIMMs DQ_0–DQ_7 and DQ_8–DQ_{15} lines with an octal transceiver; then, tie the buffered outputs together to form the 8-bit bus. Decode the transceiver's enable input with an address line.

The iSM002FLKA features hardware presence detect pins to facilitate memory design. The presence detect pins (PD1–PD7) indicate module word depth and maximum access speed (see Table 3 on the previous page). The pins allow memory-specific wait-state generation upon system initialization. To use the presence capability, pull-up the PD1–PD7 lines through a pull-up resistor. Read the lines through a port and select the appropriate memory depth and speed from a PD data table.

In the absence of high voltage on the modules V_{PP} pins, the iSM002FLKA is a read-only memory array. Manipulation of the module's control pins yields standard read, standby and output disable functions.

Read, standby and output disable operations are also available when high voltage is applied to the V_{PP} pins. In addition, high voltage on the V_{PP} pins enables erasure and programming of the module's devices. All functions associated with altering the memory contents of one or more devices—erase, erase verify, program and program verify—are accessed via each flash device's command register.

Commands are written to a device's command register using standard microprocessor write timings. Register contents serve as input to the devices internal state-machine which controls the erase and programming circuitry. Write cycles to a device also internally latch addresses and data needed for programming or erase operations. With the appropriate command written to a device's register, standard microprocessor read timings output array data, access the int$_e$ligent identifier codes, or output data for erase and program verification.

Table 4. Bus Operations

Operation \ Pins		V_{PP}(1)	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_0–DQ_{15}
READ-ONLY	Read	V_{PPL}	V_{IL}	V_{IL}	V_{IH}	Data Out
	Output Disable	V_{PPL}	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby	V_{PPL}	V_{IH}	X	X	Tri-State
READ/WRITE	Read	V_{PPH}	V_{IL}	V_{IL}	V_{IH}	Data Out(3)
	Output Disable	V_{PPH}	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby(4)	V_{PPH}	V_{IH}	X	X	Tri-State
	Write	V_{PPH}	V_{IL}	V_{IH}	V_{IL}	Data In(5)

NOTES:
1. Refer to DC Characteristics. When $V_{PP} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes are accessed via a command register write sequence. Refer to Table 5. All other addresses are low.
3. Read operations with $V_{PP} = V_{PPH}$ may access array data or the int$_e$ligent Identifier™ codes.
4. With V_{PP} at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
5. Refer to Table 5 for valid Data-In during a write operation.
6. X can be V_{IL} or V_{IH}.

Integrated Stop Timer

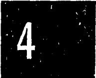
Successive command write cycles define the duration of program and erase operations; specifically, the program or erase time durations are normally terminated by associated program or erase verify commands. An integrated stop timer provides simplified timing control over these operations; thus eliminating the need for maximum program/erase timing specifications. Programming and erase pulse durations are minimums only. When the stop timer terminates a program or erase operation, the device enters an inactive state and remains inactive until receiving the appropriate verify or reset command.

Write Protection

A device's command register is only active when V_{PP} is at high voltage. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable—available only when memory updates are desired. When $V_{PP} = V_{PPL}$, the contents of the register default to the read command, making the iSM002FLKA a read-only memory. In this mode, the memory contents cannot be altered.

Or, the system designer may choose to "hardwire" V_{PP}, making the high voltage supply constantly available. In this instance, all operations are performed in conjunction with the command register. The iSM002FLKA is designed to accommodate either design practice, and to encourage optimization of flash's processor-memory interface.

The following section first discusses byte-wide organization, building a basic understanding of byte-wide bus operations, command definitions, and programming and erasure algorithms. The section concludes with performance enhancements for both 16- and 32-bit systems.



BUS OPERATIONS

Read

Each of the iSM002FLKA's flash memory devices has two control functions, both of which must be logically active, to obtain data. Chip-Enable ($\overline{CE}_X$) is the power control and should be used for device selection. Four chip enables ($\overline{CE}_0$–$\overline{CE}_3$) control the array's eight devices. Each line is unique to one set of two devices (word). Only one $\overline{CE}_X$ may be active at a time.

Output-Enable ($\overline{OE}$) is the output control and should be used to gate data from a device to the output pins on the module, independent of device selection. One $\overline{OE}$ line serves the iSM002FLKA's flash devices. Figure 7 illustrates read timing waveforms.

When the V_{PP} lines are high (V_{PPH}), a read operation can be used to access array data, to output a device's int$_e$ligent identifier™ code, and to access a device's data for program/erase verification. When V_{PP} is low (V_{PPL}), a read operation can **only** access array data.

Output Disable

With the iSM002FLKA's Output-Enable pin at a logic-high level (V_{IH}), outputs from all devices are disabled. They are placed in a high-impedance state.

STANDBY

With Chip-Enable at a logic-high level, the standby operation disables most of the deselected devices circuitry and substantially reduces device power consumption. The outputs of the deselected devices are place in a high-impedance state, independent of the Output-Enable signal. If a word is deselected during erase, programming, or program/erasure verification, the device draws active current until the operation is terminated.

Intelligent Identifier Operation

The intelligent identifier operation outputs the selected devices' manufacturer code (89H) and device code (BDH). The manufacturer code and device code are read via the devices' command register. Following a write of 90H to a device's command register, a read from address location 0000H outputs the manufacture code (89H). A read from address 0001H outputs the device code (BDH).

Write

Erasure and programming is accomplished via each device's command register, when high voltage is applied to the V_{PP} pins. The contents of each device's register serve as input to its internal state-machine. The state machine outputs dictate the function of each device.

A device's command register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

Two write enable lines are provided, $\overline{WE}_H$ and $\overline{WE}_L$, allowing selective write control of upper and lower bytes.

A device's command register is written by selecting the device (Chip-Enable low), then bringing Write-Enable ($\overline{WE}_H$ or $\overline{WE}_L$) to a logic-low level (V_{IL}). If both WE lines are a logic low, both upper and lower bytes are written. Addresses are latched on the falling edge of the Write-Enable signal, while data is latched on the rising edge of the Write-Enable pulse. Standard microprocessor write timing are used.

Refer to AC Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the module's V_{PP} pins, the contents of all devices' command registers default to 00H, enabling read-only operations.

Placing high voltage on the module's V_{PP} pins allows read/write operation on selected devices. Operations are selected by writing specific data patterns to the device(s) command register. Table 5 defines these register commands.

Table 5. Command Definitions

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	1	Write	X	00H			
Read intelligent identifier™ Codes(4)	3	Write	X	90H	Read	(4)	(4)
Set-up Erase/Erase(5)	2	Write	X	20H	Write	X	20H
Erase Verify(5)	2	Write	EA	A0H	Read	X	EVD
Set-up Program/Program(6)	2	Write	X	40H	Write	PA	PD
Program Verify(6)	2	Write	X	C0H	Read	X	PVD
Reset(7)	2	Write	X	FFH	Write	X	FFH

NOTES:
1. Bus operations are defined in Table 4.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   PA = Address of memory location to be programmed.
   Addresses are latched on the falling edge of the Write-Enable pulse.
3. ID = Data read from location IA during device identification (Mfr = 89H, Device BDH).
   EVD = Data read from location EA during erase verify.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of Write-Enable.
   PVD = Data read from location PA during program verify. PA is latched on the Program command.
4. Following the Read intelligent ID command, two read operations access manufacturer and device codes.
5. Figure 4 illustrates the Quick-Erase™ Algorithm.
6. Figure 3 illustrates the Quick-Pulse Programming™ Algorithm.
7. The second bus cycle must be followed by the desired command register write.

Read Command

While V_{PP} is high, for erasure and programming, the selected devices memory contents can be accessed via the read command. The read operation is initiated by writing 00H into the command register of each device. Microprocessor read cycles retrieve array data. The selected devices remain enabled for reads until their command register contents are altered.

The default contents of each device's command register upon V_{PP} power-up is 00H. This default value ensures that no spurious alteration to the iSM002FLKA's memory contents occurs during the V_{PP} power transition. Where the V_{PP} supply is hardwired to the iSM002FLKA's V_{PP} pins, all eight devices power-up and remain enabled for reads until their command-register contents are changed. Refer to the AC Read Characteristics and Waveforms for specific timing parameters.

Intelligent Identifier Command

Flash memories are intended for use in applications where the local CPU alters memory contents. As such, manufacturer- and device-codes must be accessible while the device resides in the target system.

Each flash memory device contains an intelligent Identifier operation. The operation is initiated by writing 90H into the command register. Following the command write, a read cycle from address 0000H retrieves the manufacturer code of 89H. A read cycle from address 0001H returns the device code of BDH. To terminate the operation, it is necessary to write another valid command into the register.

The intelligent Identifier and the Presence Detect pins give you complementary information. While the PD pins denote speed and depth, the intelligent Identifier operation gives you manufacture and device data.

Set-Up Erase/Erase Commands

Set-up Erase is a command-only operation that stages a selected device for electrical erasure of all bytes in its array. The set-up erase operation is performed by writing 20H to the command register.

To commence chip-erasure, the erase command (20H) must again be written to the register. The erase operation begins with the rising edge of a Write-Enable pulse ($\overline{WE}_H$ or $\overline{WE}_L$) and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, chip-erasure can only occur when high voltage is applied to the V_{PP} pins. In the absence of this high voltage, memory contents are protected against erasure. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all bytes of the selected device(s) in parallel. After each erase operation, all bytes must be verified. The erase verify operation is initiated by writing A0H into the command register of the device. The address for the byte to be verified must be supplied as it is latched on the falling edge of a Write-Enable pulse. The register write terminates the erase operation with the rising edge of its Write-Enable pulse.

Each 28F020 applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased.

The erase-verify command must be written to the command register prior to each byte verification to latch its address. The process continues for each byte of the device until a byte does not return FFH data, or the last address is accessed.

In the case where the data read is not FFH, another erase operation is performed. (Refer to Set-up Erase/Erase). Verification then resumes from the address of the last-verified byte. Once all bytes of the device have been verified, the erase step is complete. The device can be programmed. At this point, the verify operation is terminated by writing a valid command (e.g., Program Set-up) to the command register of the device. Figure 4, the Quick-Erase™ algorithm, illustrates how commands and bus operations are combined to perform electrical erasure of each 28F020. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Set-Up Program/Program Commands

Set-up program is a command-only operation that stages a device for byte programming. Writing 40H into the command register of the device performs the set-up operation.

Once the program set-up operation is performed, the next Write-Enable pulse causes a transition to an active programming operation. Addresses are internally latched on the falling edge of the Write-Enable pulse. Data is internally latched on the rising edge of the Write-Enable pulse. The rising edge of Write-Enable also begins the programming operation. The programming operation terminates with the next rising edge of Write-Enable, used to write the program-verify command. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Program-Verify Command

Each 28F020 is programmed on a byte-by-byte basis. Byte programming may occur sequentially or at random. Following each programming operation, the byte just programmed must be verified.

The program-verify operation is initiated by writing C0H into the command register of the device. The register write terminates the programming operation with the rising edge of its Write-Enable pulse. The program-verify operation stages the device for verification of the byte last programmed. No new address information is latched.

Each 28F020 applies an internally-generated margin voltage to the byte. A microprocessor read cycle outputs the data. A successful comparison between the programmed byte and true data means that the byte is successfully programmed. Programming then proceeds to the next desired byte location. Figure 3, the Quick-Pulse Programming algorithm (8-bit Systems), illustrates how commands are combined with bus operations to perform byte programming. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or program-command sequences to a device. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

EXTENDED ERASE/PROGRAM CYCLING

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled—an expensive solution.

Intel has designed extended cycling capability into its ETOX II flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field is one-tenth that of common EEPROMs, minimizing the probability of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 MV/cm lower than EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure—increasing time to wearout by a factor of 100,000,000.

Each of the iSM002FLKA's eight 28F020s is specified for a minimum of 10,000 program/erase cycles. Each device is programmed and erased using Intel's Quick-Pulse Programming and Quick-Erase algorithms. Intel's algorithmic approach uses a series of operations (pulses), along with byte verification, to completely and reliably erase and program the device.

For further information, see Reliability Report RR-60 (ETOX II Reliability Data Summary).

QUICK-PULSE PROGRAMMING ALGORITHM

The Quick-Pulse Programming algorithm uses programming operations of 10 μs duration. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The algorithm allows for up to 25 programming operations per byte, although most bytes verify on the first or second operation. The entire sequence of programming and byte verification is performed with V_{PP} at high voltage. Figure 3 illustrates the Quick-Pulse Programming algorithm for 8-bit systems.

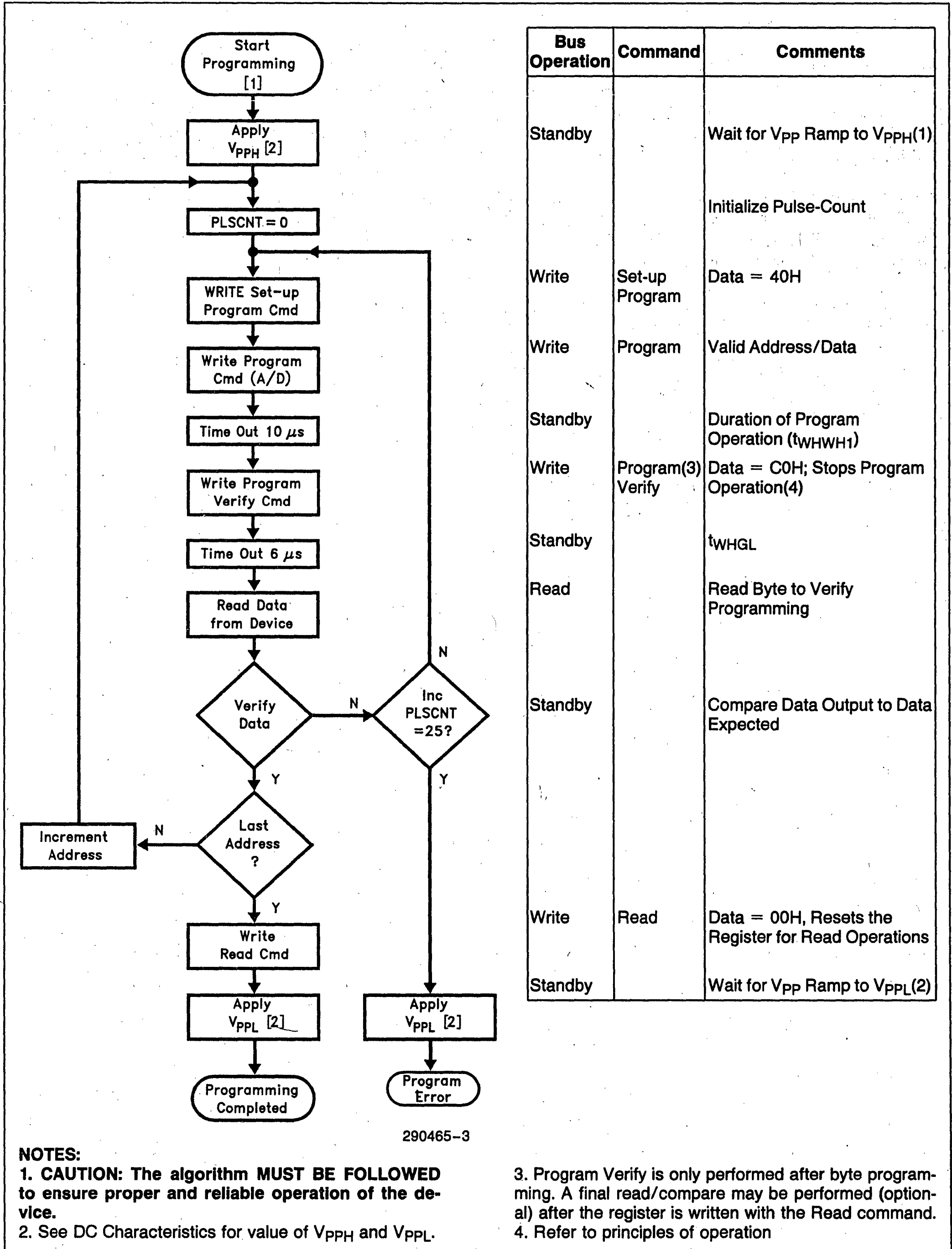
QUICK-ERASE ALGORITHM

Intel's Quick-Erase algorithm yields fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the Quick-Pulse Programming algorithm, to simultaneously remove charge from all bits in the array.

Erasure begins with a read of memory contents. The iSM002FLKA is erased when shipped from the factory. Reading FFH data from each device would immediately be followed by device programming.

For devices being erased and reprogrammed, uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (Data = 00H). This is accomplished, using the Quick-Pulse Programming algorithm, in approximately two seconds.

Erase execution then continues with an initial erase operation. Erase verification (data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. With each erase operation, an increasing number of bytes verify to the erased state. Erase efficiency may be improved by storing the address of the last byte verified in a register. Following the next erase operation, verification starts at that stored address location. Erasure typically occurs in one second. Figure 4 illustrates the Quick-Erase algorithm for 8-bit systems.

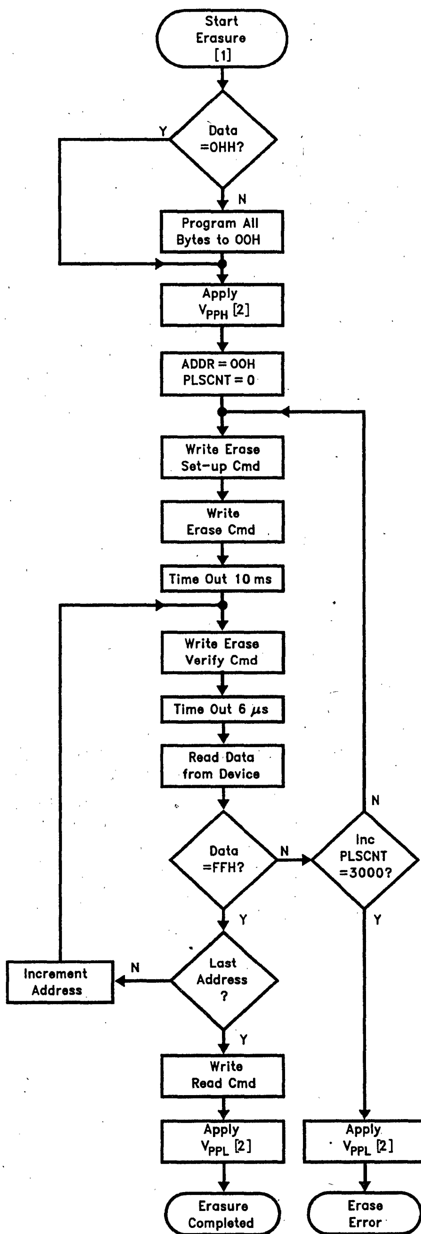


Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Pulse-Count
Write	Set-up Program	Data = 40H
Write	Program	Valid Address/Data
Standby		Duration of Program Operation (t_{WHWH1})
Write	Program(3) Verify	Data = C0H; Stops Program Operation(4)
Standby		t_{WHGL}
Read		Read Byte to Verify Programming
Standby		Compare Data Output to Data Expected
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(2)

NOTES:

1. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**
2. See DC Characteristics for value of V_{PPH} and V_{PPL}.
3. Program Verify is only performed after byte programming. A final read/compare may be performed (optional) after the register is written with the Read command.
4. Refer to principles of operation

Figure 3. Quick-Pulse Programming Algorithm (8-Bit Systems)



Bus Operation	Command	Comments
		Entire Memory Must = 00H Before Erasure
		Use Quick-Pulse Programming™ Algorithm (Figure 4)
Standby		Wait for V_{PP} Ramp to V_{PPH}(2)
		Initialize Addresses and Pulse-Count
Write	Set-up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase(3) Verify	Addr = Byte to Verify; Data = A0H; Stops Erase Operation(4)
Standby		t_{WHGL}
Read		Read Byte to Verify Erasure
Standby		Compare Output to FFH Increment Pulse-Count
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(2)

NOTES:

1. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.

2. See DC Characteristics for value of V_{PPH} and V_{PPL}.

3. Erase Verify is performed only after chip-erasure. A final read/compare may be performed (optional) after the register is written with the read command.

4. Refer to principles of operation.

Figure 4. Quick-Erase Algorithm (8-Bit Systems)

HIGH PERFORMANCE PARALLEL DEVICE ERASURE

Total erase time for the iSM002FLKA is reduced by implementing a parallel erase algorithm (Note 1). You save time by erasing all devices at the same time. However, since flash memories may erase at different rates, you must verify each device separately. This can be done in a word-wise fashion with the command register Reset command and a special masking algorithm.

Take for example the case of two-device (parallel) erasure. The CPU first writes the data word erase command 2020h twice in succession. This starts erasure. After 10 ms, the CPU writes the data word verify command A0A0h to stop erasure and setup erase verification. If both bytes are erased at the given address, then the CPU increments the address (by 2) and then writes the verify command A0A0h again. If neither byte is erased, then the CPU issues the erase sequence again without incrementing the address.

Suppose at the given address only the low byte verifies FFh data? Could the whole chip be erased? The answer is yes. Rather than check the rest of the low byte addresses independently of the high byte, simply use the reset command to mask the low byte from erasure and erase verification on the next erase loop. In this example the erase command would be 20FFh and the verify command would be A0FFh. Once the high byte verifies at that address, the CPU modifies the command back to the default 2020h and A0A0h, increments the address by 2, and writes the verify command to the next address.

See Figure 5 for a conceptual view of the parallel erase flow chart and Appendix B for the detailed version. These flow charts are for 16-bit systems and can be expanded for 32-bit designs.

NOTE:
1. Parallel Erasure and Programming require appropriate choice of V_{PP} supply to support the increased power consumption.

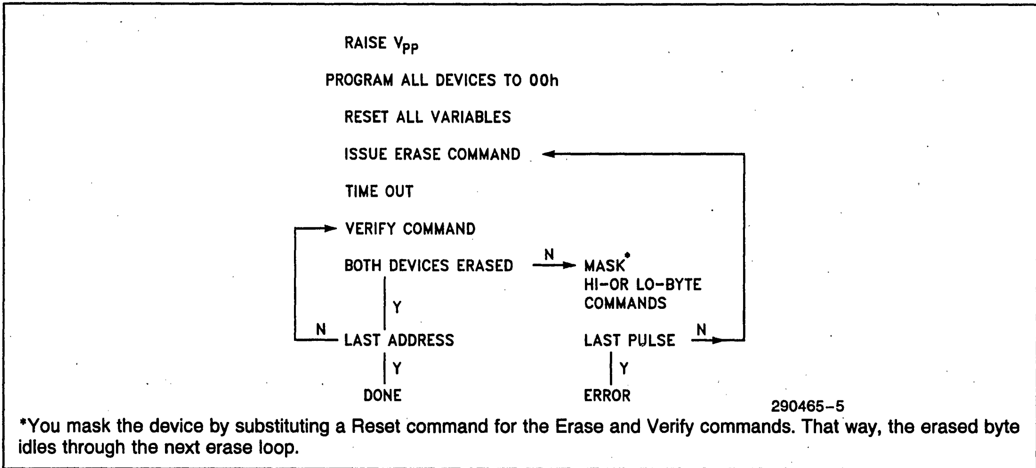
HIGH PERFORMANCE PARALLEL DEVICE PROGRAMMING

Software for word- or double-word programming can be written in two different manners. The first method offers simplicity of design and minimizes software overhead by using a byte programming routine on each device independently (using host CPU's byte addressing mode). The second method offers higher performance by programming the word or double-word data in parallel. This method manipulates the command register instructions for independent byte control. See Figure 6 for conceptual 2-device parallel programming flow chart and Appendix C for the detailed version. Here you can use the host CPU's appropriate word- or double-word addressing modes (i.e., incrementing by 2- or 4-byte addresses, respectively).

NOTE:
Word or double-word programming assumes 2 or 4 8-bit flash memory devices.

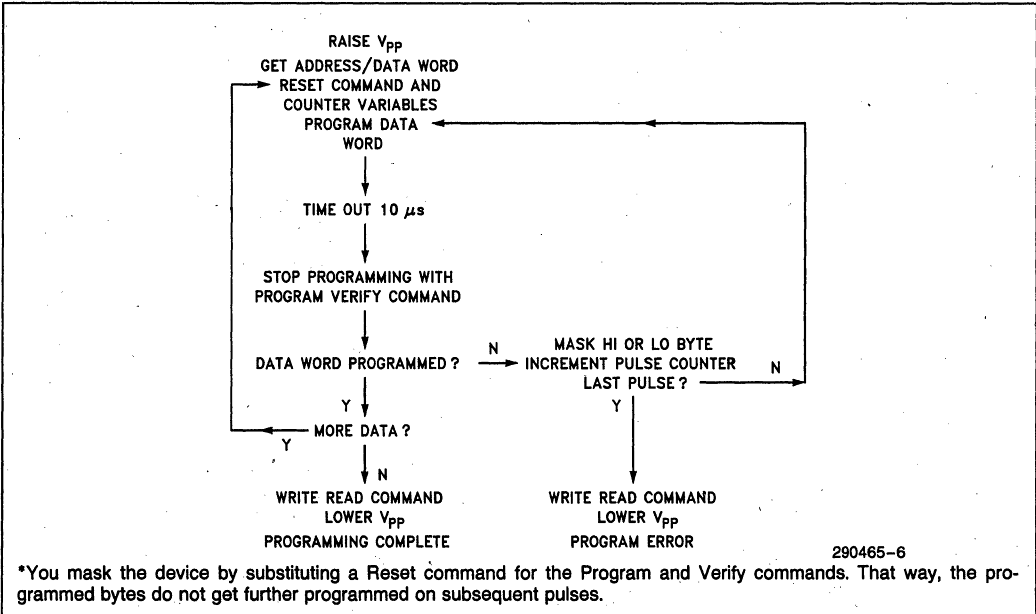
Parallel Programming Algorithm Summary:

- Decreases programming time by programming 2 flash memories (16 bits) in parallel. The algorithm can be expanded for 32-bit systems.
- Eliminates tracking of high/low byte addresses and respective number of program pulses by directing the CPU to write data-words (16-bit) to the command register.
- Maintains word write and read operations. Should a byte on one device program prior to a byte on the other, the CPU continues to write word-commands to both devices. However, it deselects the verified byte with software commands. An alternative is to independently program high and low bytes using hardware select capability (byte-addressing mode of host CPU).



*You mask the device by substituting a Reset command for the Erase and Verify commands. That way, the erased byte idles through the next erase loop.

Figure 5. High Performance Parallel Erasure (Conceptual Overview)



*You mask the device by substituting a Reset command for the Program and Verify commands. That way, the programmed bytes do not get further programmed on subsequent pulses.

Figure 6. Parallel Programming Flow Chart (Conceptual Overview)

DESIGN CONSIDERATIONS

Two-Line Output Control

Two-line control provides for:

a. the lowest possible memory power dissipation and,

b. complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address-decoder output should drive chip-enable, while the system's read signal controls all flash-memories and other parallel memories. This assures that only enabled memory devices have active outputs, while deselected devices maintain the low power standby condition.

Power Supply Decoupling

Flash-memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current (I_{CC}) issues—standby, active, and transient current peaks produced by falling and rising edges of chip-enable. The capacitive and inductive loads on the device outputs determine the magnitudes of these peaks.

Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. The iSM002FLKA features a 0.1 μF ceramic capacitor connected between V_{CC} and V_{SS}, and between V_{PP} and V_{SS}.

Also, a 4.7 μF tantalum capacitor decouples the array's power supply between V_{CC} and V_{SS} and between V_{PP} and V_{SS}. The bulk capacitors will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

V_{PP} Trace on Printed Circuit Boards

Programming flash memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the V_{PP} power supply trace. The two V_{PP} pins supply current for programming. Use similar trace widths and layout considerations given the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots. Be sure to connect both module V_{PP} inputs to your 12V supply.

Power Up/Down Protection

The iSM002FLKA is designed to offer protection against accidental erasure or programming during power transitions. Upon power-up, each 28F020 is indifferent as to which power supply, V_{PP} or V_{CC}, powers up first. Power supply sequencing is not required. Internal circuitry in each 28F020 ensures that the command register is reset to the read mode on power up.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will inhibit writes. The control register architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases the usable battery life of your system because each 28F020 does not consume any power to retain code or data when the system is off. Table 4 illustrates the power dissipated when updating each 28F020.

Table 4. 28F020 Typical Update Power Dissipation(4)

Operation	Power Dissipation (Watt-Seconds)
Array Program/Program Verify(1)	0.34
Array Erase/Erase Verify(2)	0.37
One Complete Cycle(3)	1.05

NOTES:

1. Formula to calculate typical Program/Program Verify Power = [V_{PP} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{PP2} typical + t_{WHGL} × I_{PP4} typical)] + [V_{CC} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{CC2} typical + t_{WHGL} × I_{CC4} typical].
2. Formula to calculate typical Erase/Erase Verify Power = [V_{PP} (V_{PP3} typical × t_{ERASE} typical + I_{PP5} typical × t_{WHGL} × # Bytes)] + [V_{CC} (I_{CC3} typical × t_{ERASE} typical + I_{CC5} typical × t_{WHGL} × # Bytes)].
3. One Complete Cycle = Array Preprogram + Array Erase + Program.
4. "Typicals are not guaranteed but based on a limited number of samples taken from production lots.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read ....0°C to +70°C[1]
During Erase/Program ....0°C to +70°C

Temperature Under Bias .... −10°C to +80°C

Storage Temperature .... −50°C to +100°C

Voltage on Any Pin with Respect to Ground .... −2.0V to +7.0V[2]

V_{PP} Supply Voltage with Respect to Ground During Erase/Program .... −2.0V to +14.0V[2, 3]

V_{CC} Supply Voltage with Respect to Ground .... −2.0V to +7.0V[2]

Output Short Circuit Current............. 100 mA[4]

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
3. Maximum DC voltage on V_{PP} may overshoot to +14.0V for periods less than 20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature	0	70	°C	For Read-Only and Read/Write Operations
V_{CC}	V_{CC} Supply Voltage	4.50	5.50	V	

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ	Max		
I_{LI}	Input Leakage Current	3			±8.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	3			±40.0	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1, 3			8.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	2, 3		26	66	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	2, 3		8.0	26	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	2, 3		16.0	36	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	2, 3		16.0	36	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	2, 3		16.0	36	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	3			±80	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	3		0.7	1.6	mA	$V_{PP} > V_{CC}$
					±80	μA	$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	2, 3		16.5	61.2	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	2, 3		20.5	61.2	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	2, 3		4.5	11.2	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	2, 3		4.5	11.2	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	I_{OH} = −2.5 mA $V_{CC} = V_{CC}$ Min
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

NOTES:
1. V_{CC} standby current for 8 devices.
2. Calculations assume only the 2 devices of the 16-bit word are enabled. The remaining 6 devices are in standby. Current will be higher if interleaving is used.
3. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C. These currents are valid for all product versions (packages and speeds).

DC CHARACTERISTICS—CMOS COMPATIBLE

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ	Max		
I_{LI}	Input Leakage Current	3			±8.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	3			±40.0	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1, 3		0.4	0.8	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC} \pm 0.2V$
I_{CC1}	V_{CC} Active Read Current	2, 3		20.3	60.6	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	2, 3		2.3	20.6	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	2, 3		10.3	30.6	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	2, 3		10.3	30.6	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	2, 3		10.3	30.6	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current				±80	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	3		0.7	1.6	mA	$V_{PP} > V_{CC}$
					±80	μA	$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	2, 3		16.5	61.2	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	2, 3		20.5	61.2	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	2, 3		4.5	11.2	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	2, 3		4.5	11.2	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 5.8 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		0.85 V_{CC}			V	I_{OH} = −2.5 mA, $V_{CC} = V_{CC}$ Min
V_{OH2}			V_{CC} − 0.4				I_{OH} = −100 μA, $V_{CC} = V_{CC}$ Min
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

NOTES:
1. V_{CC} standby current for 8 devices.
2. Calculations assume only the 2 devices of the 16-bit word are enabled. The remaining 6 devices are in standby. Current will be higher if interleaving is used.
3. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C. These currents are valid for all product versions (packages and speeds).

CAPACITANCE(1) $T_A = 25°C, f = 1.0 MHz$

Symbol	Parameter	Notes	Limits		Unit	Conditions
			Min	Max		
C_{IN1}	Address Capacitance	2		60	pF	$V_{IN} = 0V$
C_{IN2}	Control Capacitance	2		65	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	2		55	pF	$V_{OUT} = 0V$

NOTES:
1. Trace capacitance calculated, not measured.
2. Address and control capacitance of a typical device is 6 pF.
3. Output capacitance of a typical device is 12 pF.

AC TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ...... 10 ns
Input Pulse Levels .... 0.45V and 2.4V
Input Timing Reference Level .... 0.8V and 2.0V
Output Timing Reference Level ...... 0.8V and 2.0V

AC CHARACTERISTICS—Read-Only Operations(2)

Versions		Notes	iSM002FLKA-150		Unit
Symbol	Characteristic		Min	Max	
t_{AVAV}/t_{RC}	Read Cycle Time	3	150		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time			150	ns
t_{AVQV}/t_{ACC}	Address Access Time			150	ns
t_{GLQV}/t_{OE}	Output Enable Access Time			55	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	3	0		ns
t_{EHQZ}	Chip Disable to Output in High Z	3		55	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	3	0		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	4		35	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	3	0		ns
t_{WHGL}	Write Recovery Time before Read		6		µs

NOTES:
1. Whichever occurs first.
2. Rise/Fall Time ≤ 10 ns.
3. Not 100% tested: Characterization data available.
4. Guaranteed by design.

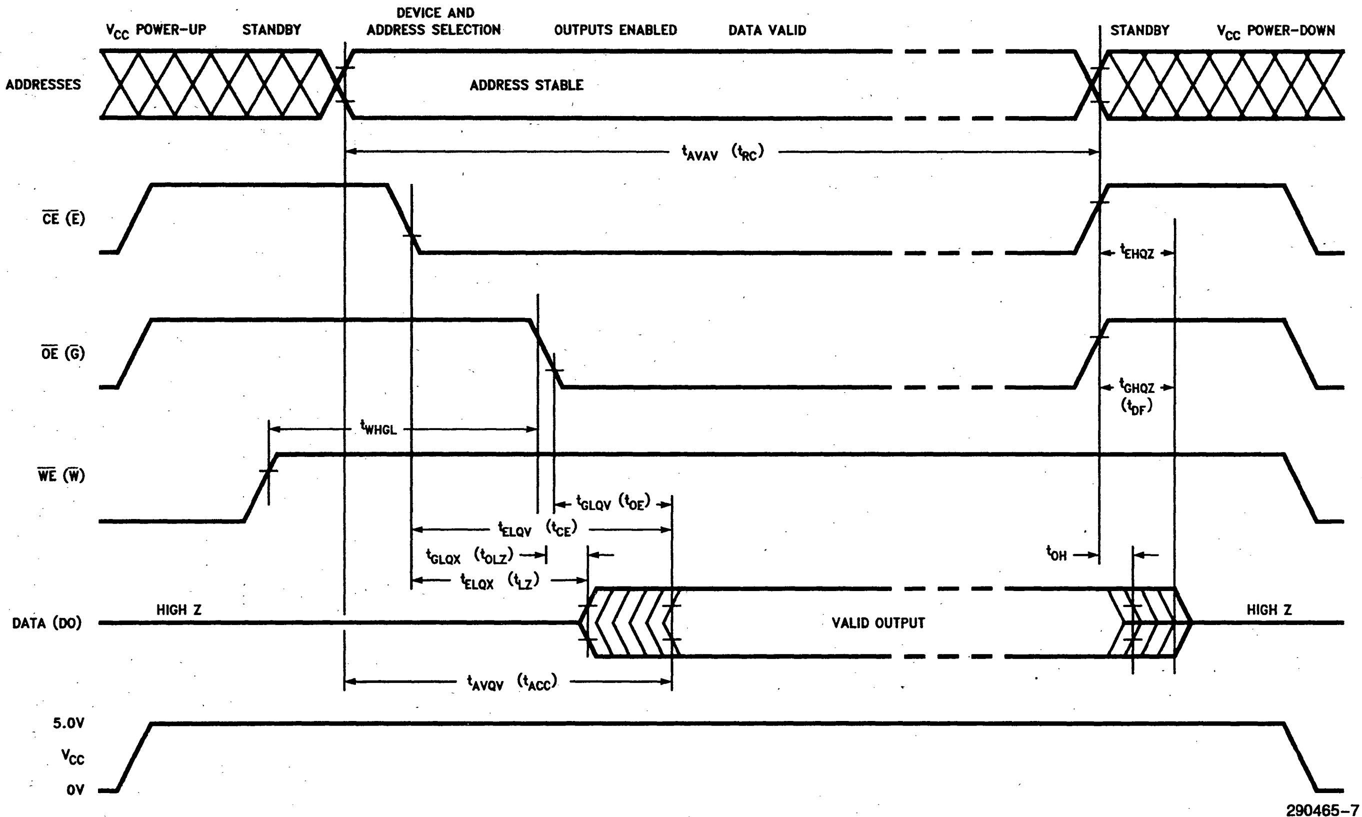


Figure 7. AC Waveforms for Read Operations

AC CHARACTERISTICS—Write/Erase/Program Operations(1, 2)

Versions			ISM002FLKA-150		Unit
Symbol	Characteristic	Notes	Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		150		ns
t_{AVWL}/t_{AS}	Address Set-up Time		0		ns
t_{WLAX}/t_{AH}	Address Hold Time		60		ns
t_{DVWH}/t_{DS}	Data Set-up Time		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		ns
t_{WHGL}	Write Recovery Time before Read		6		μs
t_{GHWL}	Read Recovery Time before Write		0		μs
t_{ELWL}/t_{CS}	Chip Enable Set-up Time before Write		20		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width	2	60		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		ns
t_{WHWH1}	Duration of Programming Operation	3	10		μs
t_{WHWH2}	Duration of Erase Operation	3	9.5		ms
t_{VPEL}	V_{PP} Set-up Time to Chip Enable Low		1.0		μs

NOTES:
1. Read timing characteristics during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Rise/Fall time ≤ 10 ns.
3. The integrated stop timer terminates the program/erase operations, thereby eliminating the need for a maximum specification.

ERASE AND PROGRAMMING PERFORMANCE

Parameter	Notes	Limits			Unit
		28F020-150			
		Min	Typ	Max	
Chip Erase Time	1, 3, 4		2	30	Sec
Chip Program Time	1, 2, 4		4	25	Sec
Erase/Program Cycles	1, 5	10,000	100,000		Cycles

NOTES:
1. Typicals are not guaranteed, but based on a limited number of samples from production lots. Data taken at 25°C, 12.0V V_{PP}.
2. Minimum byte programming time excluding system overhead is 16 μs (10 μs program + 6 μs write recovery), while maximum is 400 μs/byte (16 μs x 25 loops allowed by algorithm). Max chip programming time is specified lower than the worst case allowed by the programming algorithm since most bytes program significantly faster than the worst case byte.
3. Excludes 00H Programming prior to Erasure.
4. Excludes System-Level Overhead.
5. Refer to RR-60 "ETOX™II Flash Memory Reliability Data Summary" for typical cycling data and failure rate calculations.

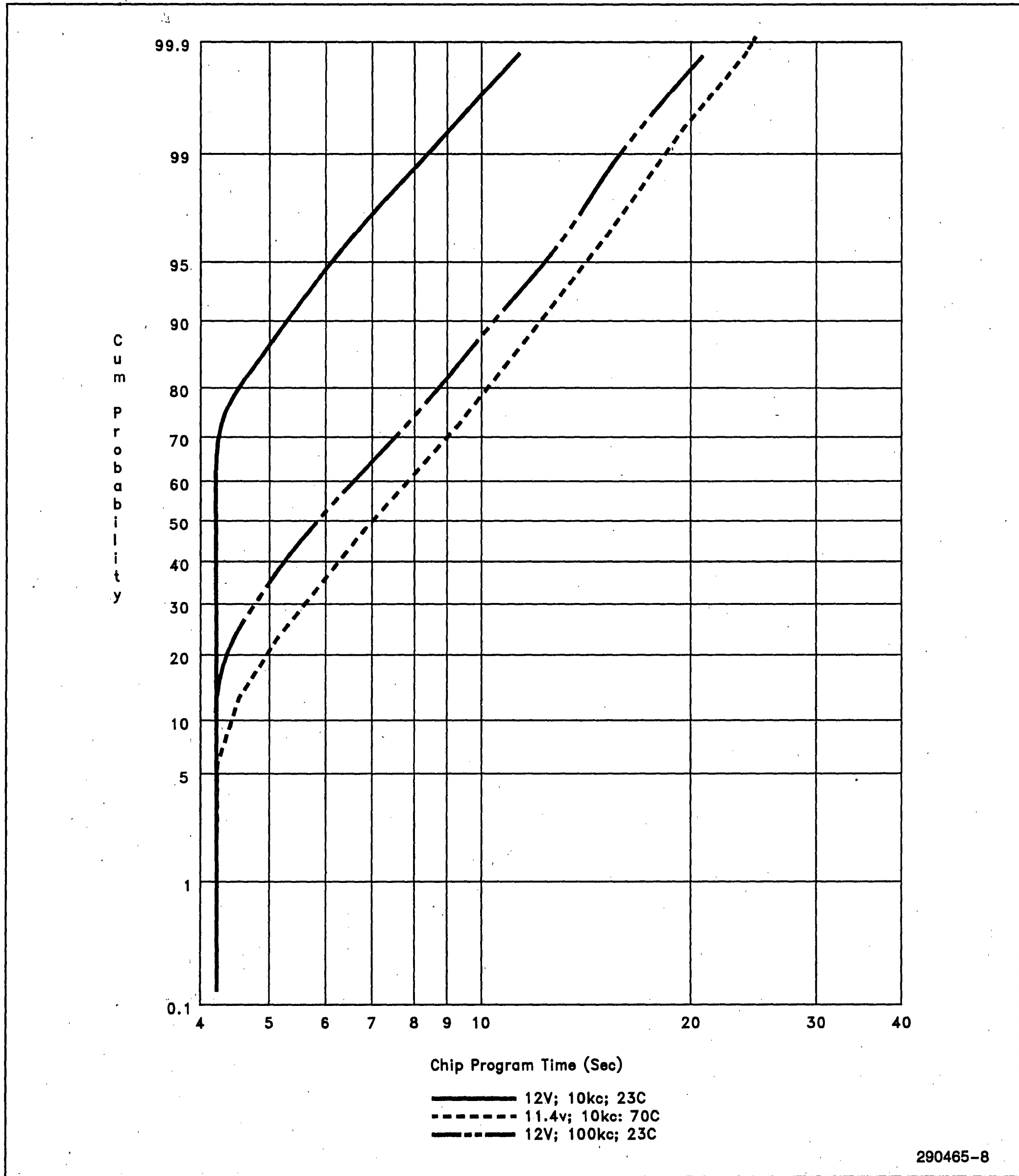


Figure 8. 28F020 Typical Programming Capability

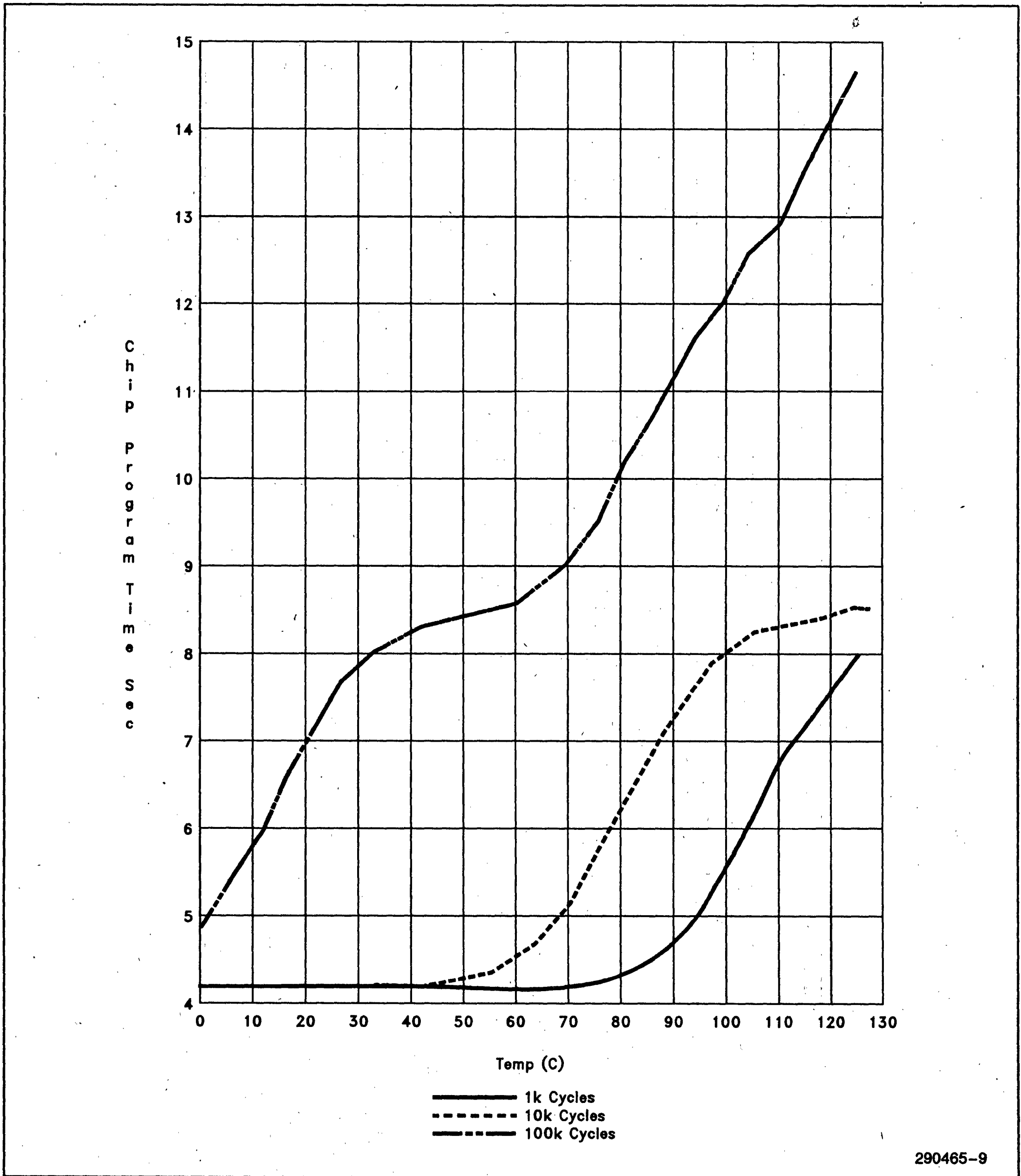
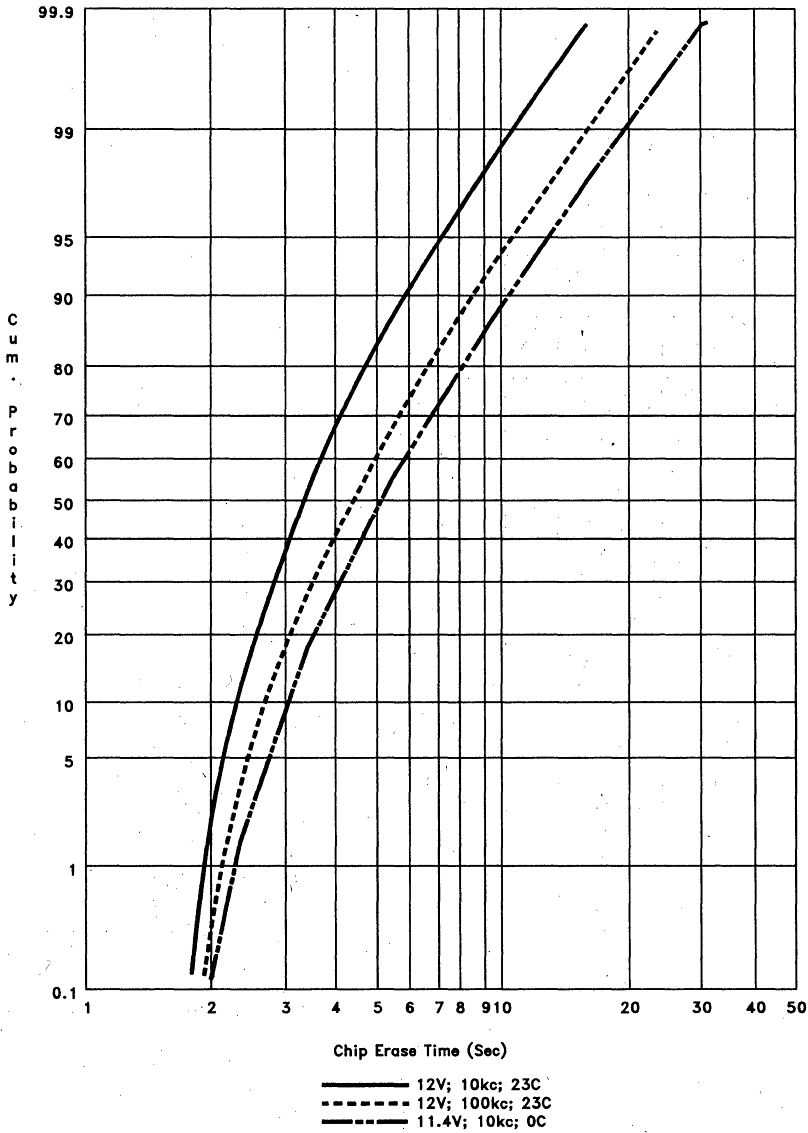
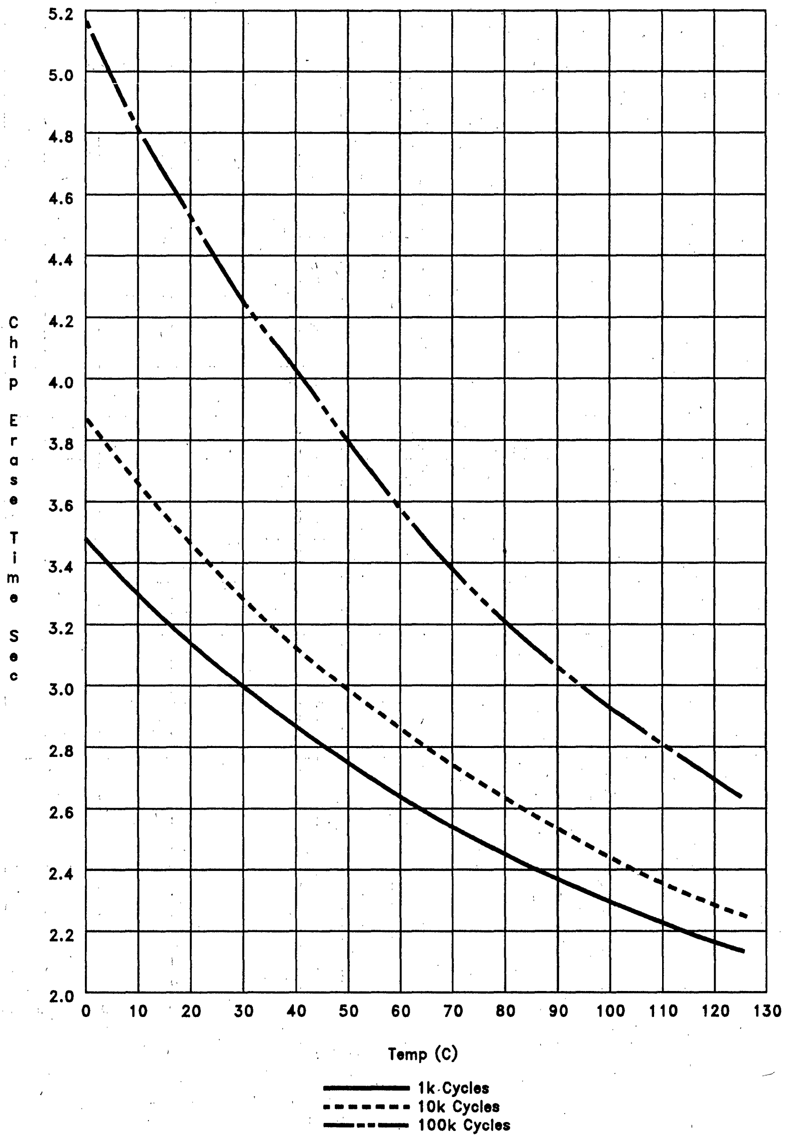


Figure 9. 28F020 Typical Program Time at 12V



NOTE:
Does not include Pre-Erase Program.

Figure 10. 28F020 Typical Erase Capability



NOTE:
Does not include Pre-Erase Program.

Figure 11. 28F020 Typical Erase Time at 12.0V

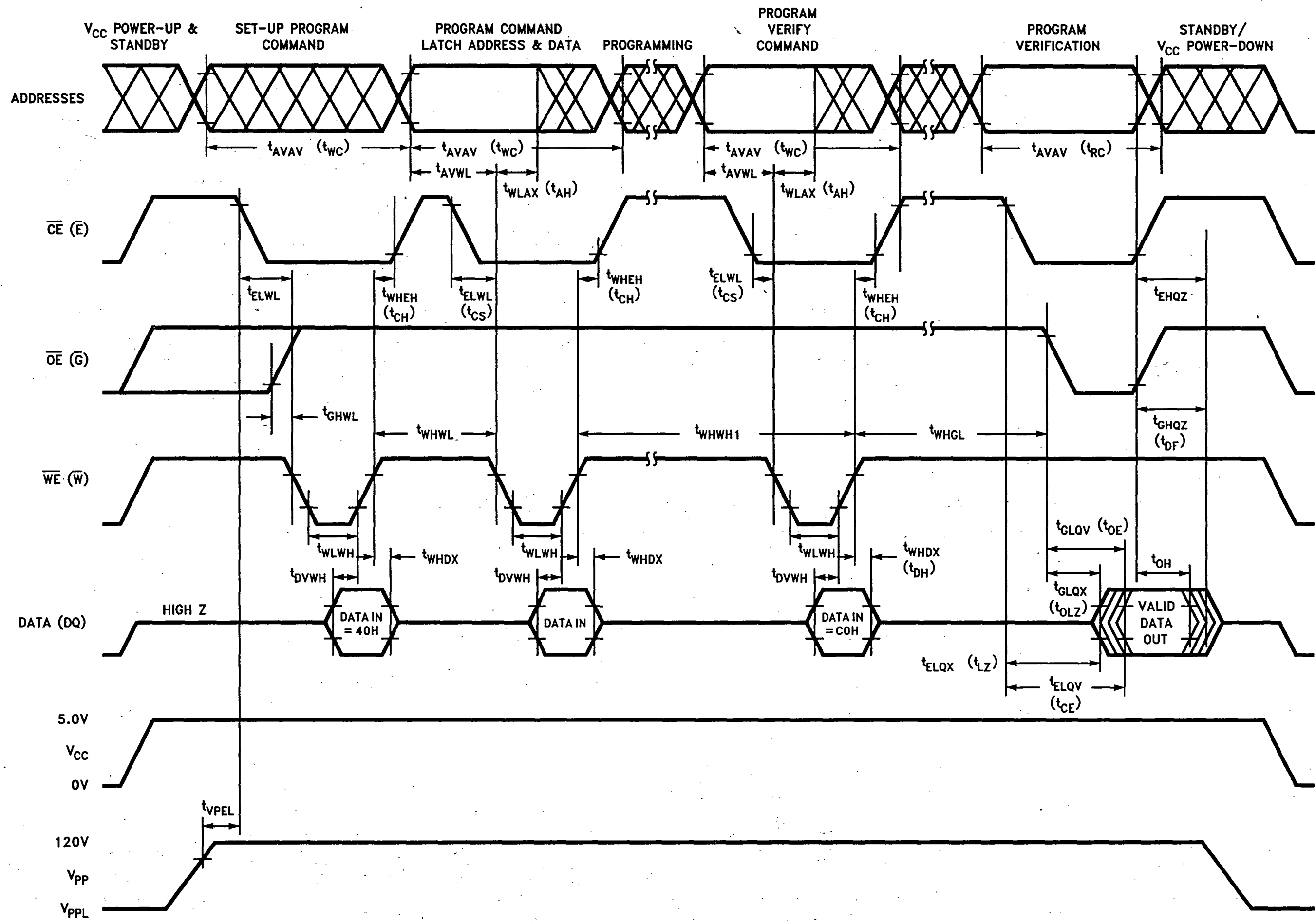


Figure 12. AC Waveforms for Programming Operations

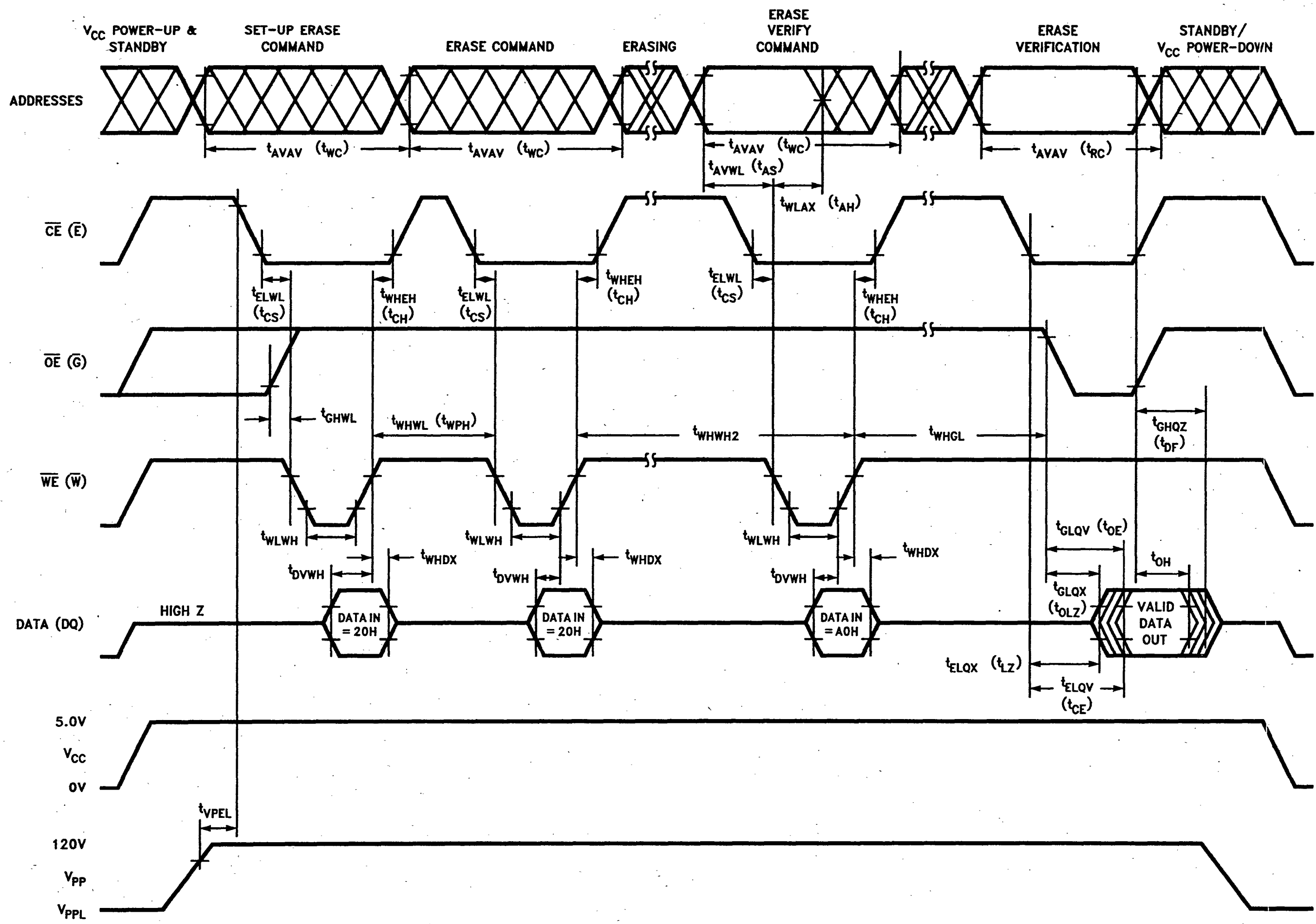


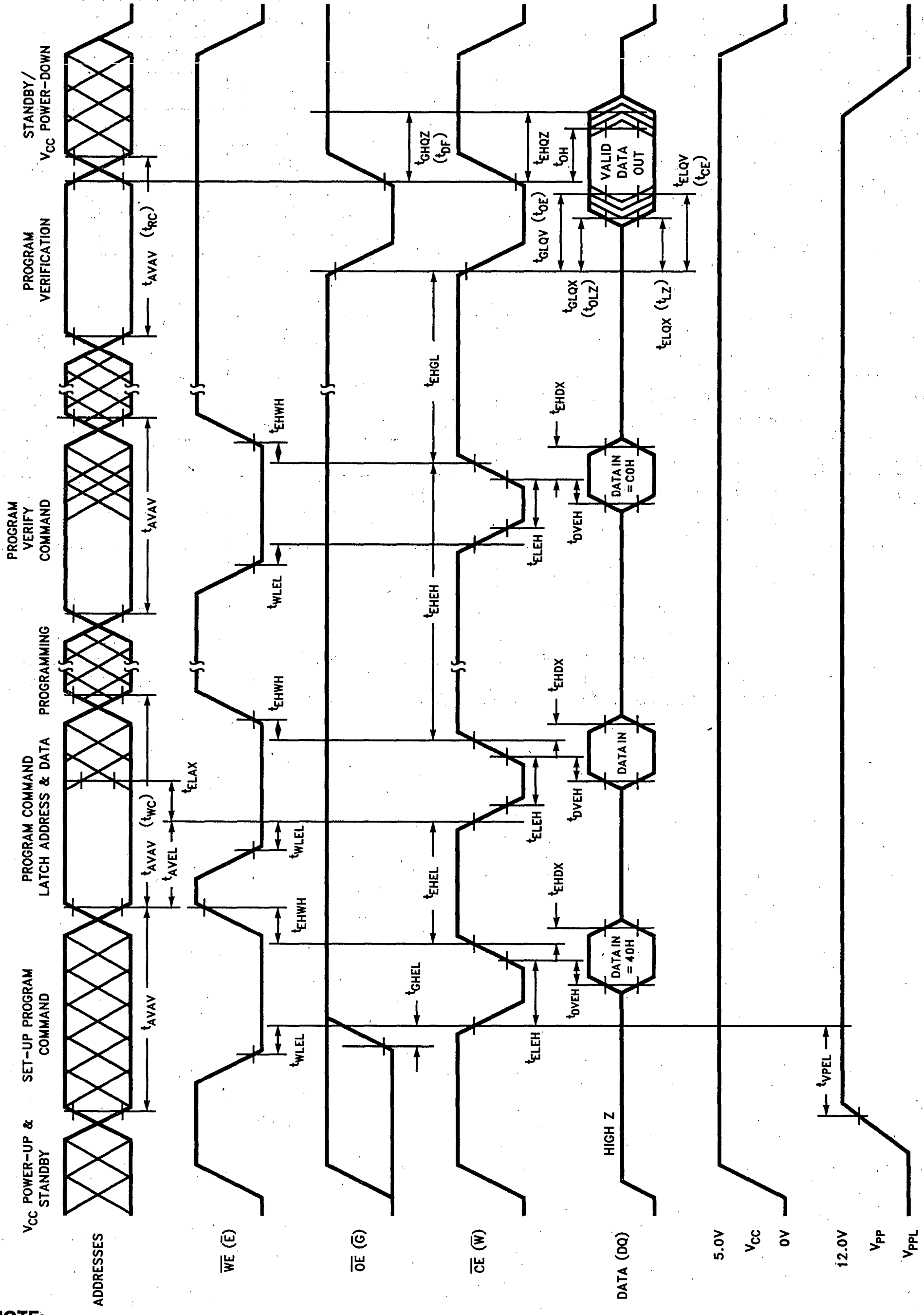
Figure 13. AC Waveforms for Erase Operations

ALTERNATIVE $\overline{CE}$-CONTROLLED WRITES

Versions			28F020-150		Unit
Symbol	Characteristic	Notes	Min	Max	
t_{AVAV}	Write Cycle Time		150		ns
t_{AVEL}	Address Set-Up Time		0		ns
t_{ELAX}	Address Hold Time		80		ns
t_{DVEH}	Data Set-Up Time		50		ns
t_{EHDX}	Data Hold Time		10		ns
t_{EHGL}	Write Recovery Time before Read		6		µs
t_{GHEL}	Read Recovery Time before Write		0		µs
t_{WLEL}	Write Enable Set-Up Time before Chip Enable		0		ns
t_{EHWH}	Write Enable Hold Time		0		ns
t_{ELEH}	Write Pulse Width	1	70		ns
t_{EHEL}	Write Pulse Width High		20		ns
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low		1.0		µs

NOTE:
1. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of Chip-Enable and Write-Enable. In systems where Chip-Enable defines the write pulse width (within a longer Write-Enable timing waveform) all set-up, hold and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.

4



NOTE:
Alternative $\overline{CE}$-Controlled Write Timings also apply to erase operations.

Figure 14. Alternate AC Waveforms for Programming Operations

APPENDIX A
PARTIAL LIST[1] OF 80-PIN SIMM SOCKET COMPANIES

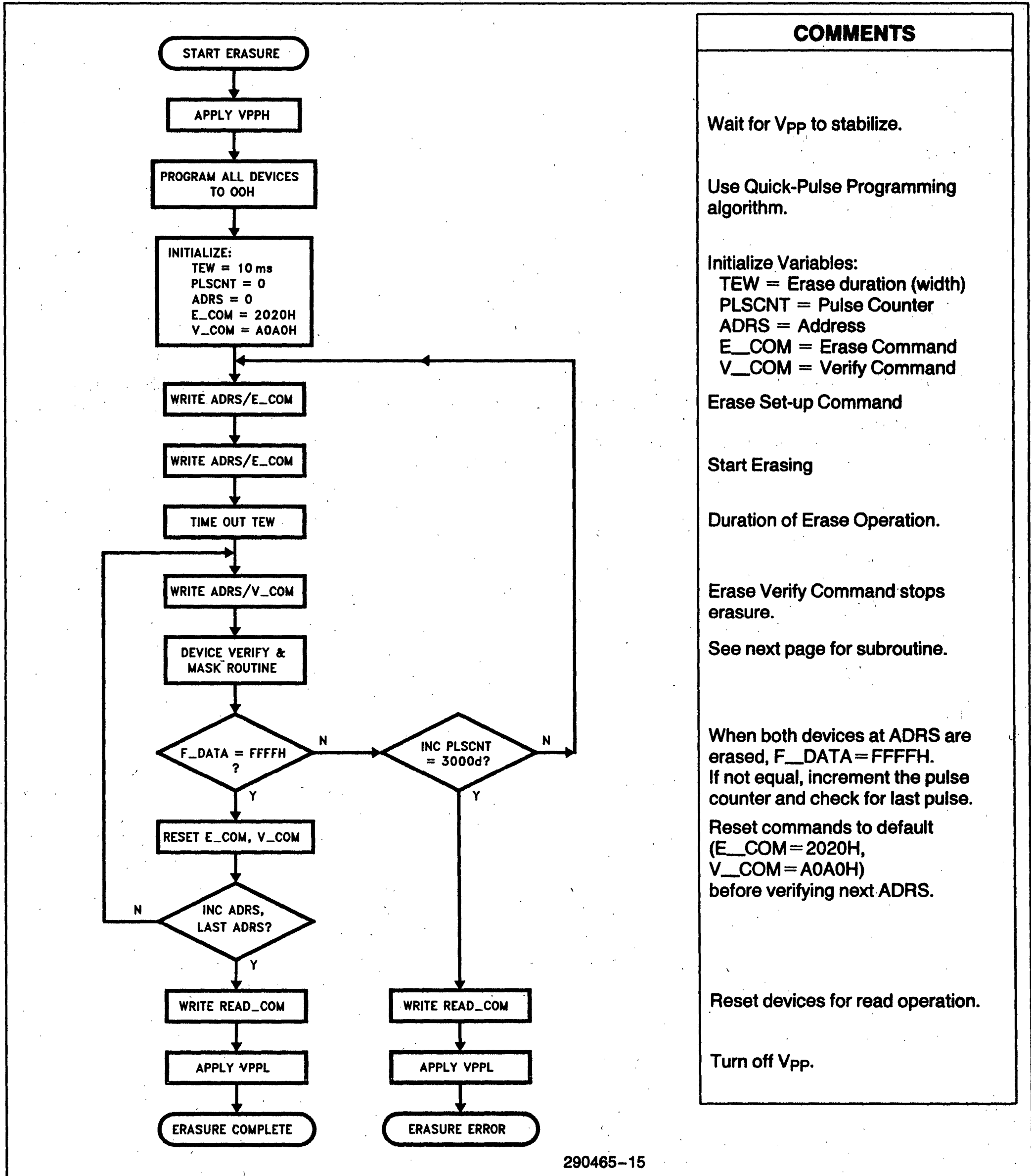
AMP INCORPORATED
HARRISBURG, PA 17105
(800) 522-6752

BURNDY CORPORATION
51 RICHARDS AVENUE
NORWALK, CT 06856
(203) 838-4444

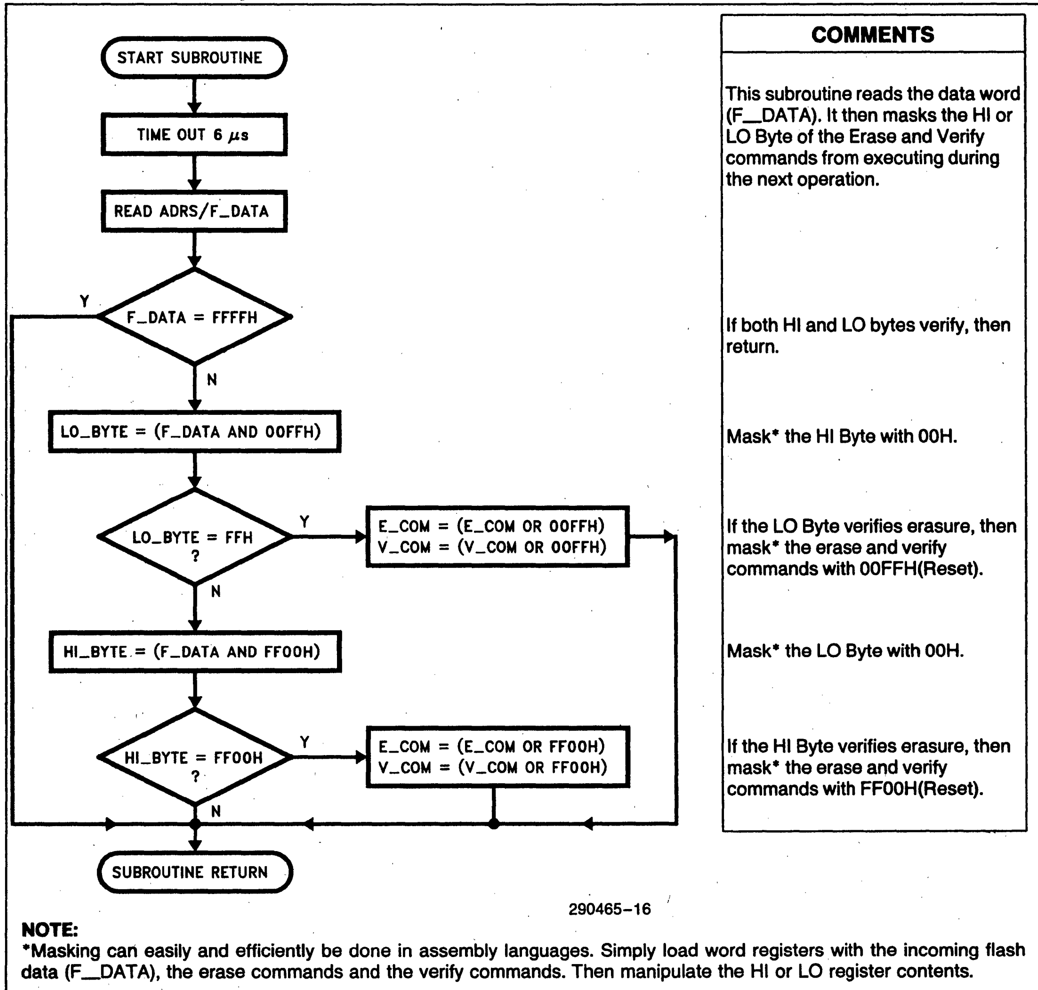
MOLEX
2222 WELLINGTON COURT
LISLE, IL 60532
(708) 969-4550

NOTES:
1. This list is intended for example only, and in no way represents all companies that support 80-pin SIMM Sockets. Intel Corporation assumes no responsibility for circuitry other than circuitry embodies in an Intel product. No other circuit patent licenses are implied.
2. Socket reliability data can be obtained from the above companies upon request.

APPENDIX B
PARALLEL ERASE FLOW CHART

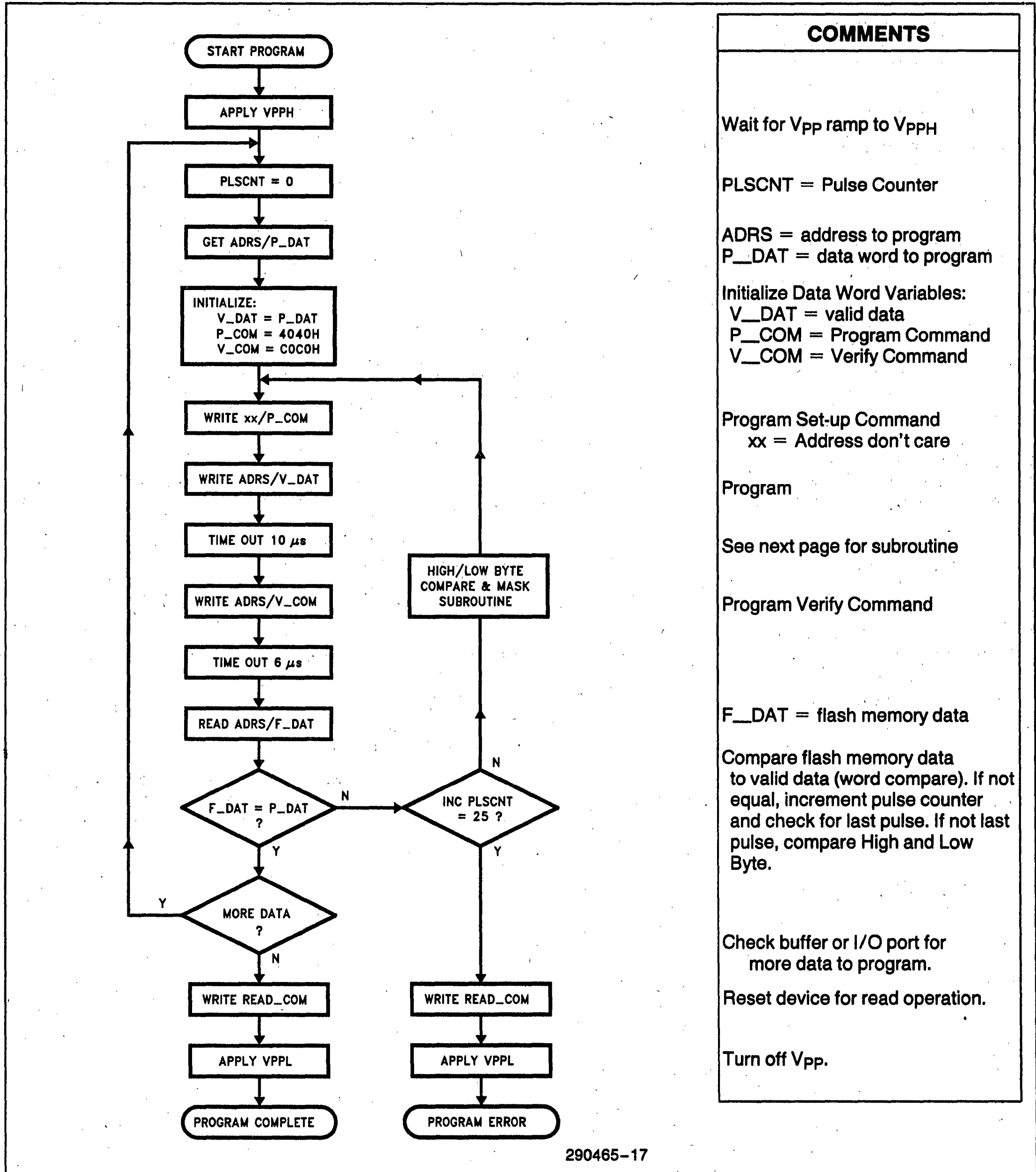


Device Erase Verify and Mask Subroutine



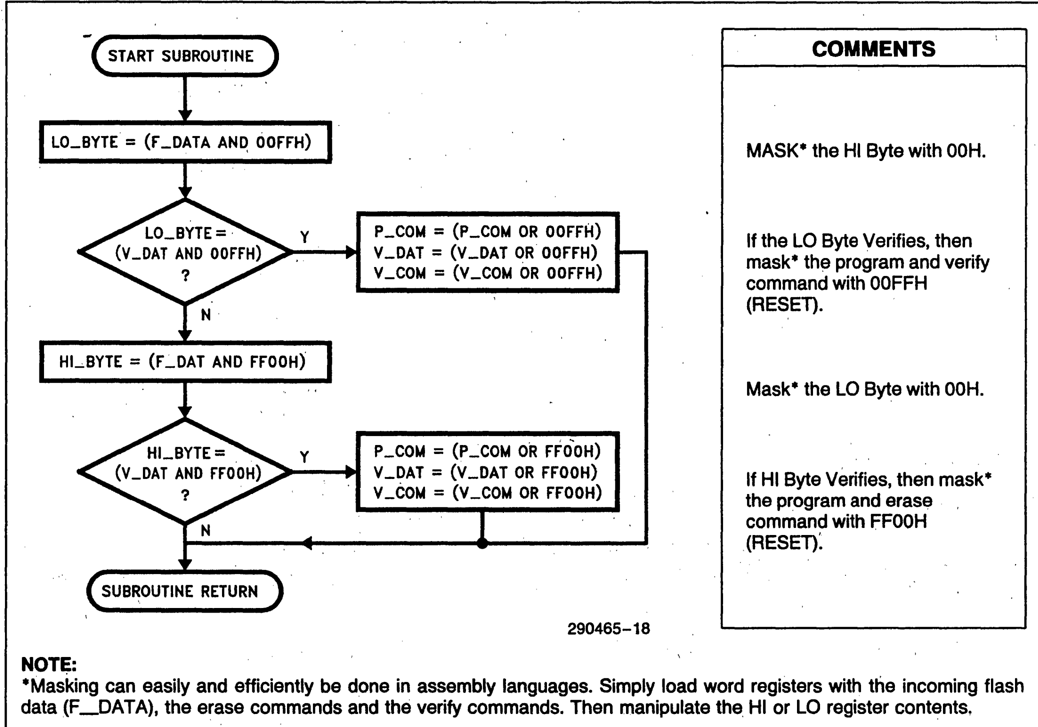
NOTE:
*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming flash data (F_DATA), the erase commands and the verify commands. Then manipulate the HI or LO register contents.

APPENDIX C
PARALLEL PROGRAMMING FLOW CHART



290465-17

Program Verify and Mask Subroutine

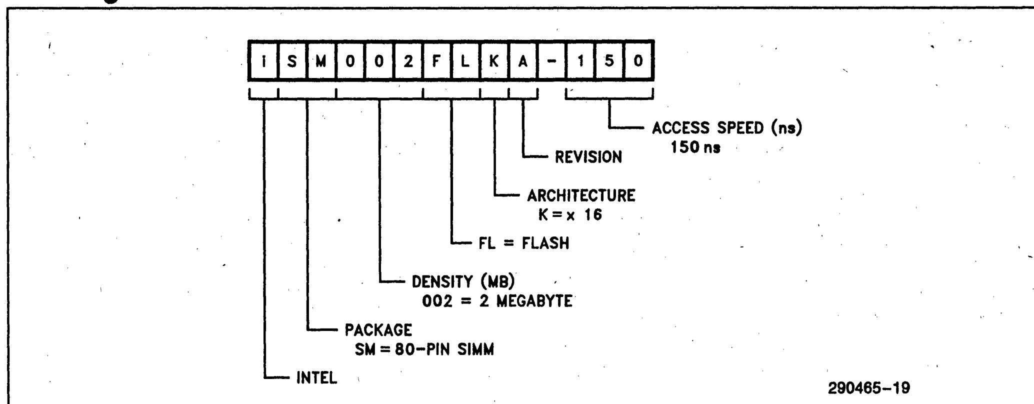


290465-18

NOTE:
*Masking can easily and efficiently be done in assembly languages. Simply load word registers with the incoming flash data (F_DATA), the erase commands and the verify commands. Then manipulate the HI or LO register contents.

4

Ordering Information



Valid Combinations:

iSM002FLKA-150

ADDITIONAL INFORMATION

	Order Number
ER-20, "ETOX™ II Flash Memory Technology"	294005
ER-24, "The Intel 28F020 Flash Memory"	294008
RR-60, "ETOX™ II Flash Memory Reliability Data Summary"	293002
AP-316, "Using Flash Memory for In-System Reprogrammable Nonvolatile Storage"	292046
AP-325, "Guide to Flash Memory Reprogramming"	292059
AP-343, "Flash Memory — A Mass Storage Medium"	292079

September 1992

4

Solutions for High Density Applications Using Intel Flash Memory

MARKUS A. LEVY
DALE ELBERT
APPLICATIONS ENGINEERING
INTEL CORPORATION

Order Number: 292079-004

Solutions For High Density Applications Using Intel Flash Memory

CONTENTS

PAGE

INTRODUCTION 4-201

ADVANCED PACKAGING 4-201

Plastic Leaded Chip Carrier (PLCC) 4-201

Thin Small Outline Package (TSOP) 4-202

Memory Cards 4-204

Single In-Line Memory Module (SIMM) .. 4-206

HARDWARE DESIGN IMPLEMENTATIONS 4-209

Design Example—A Paged-Mapped Memory Board 4-209

Optional Board Features 4-220

Initializing Software for the Paged Memory Board 4-224

Linear Addressing 4-226

I/O Addressing 4-227

Capacitive Loading 4-229

SOFTWARE DESIGN IMPLEMENTATIONS 4-230

Data Recording 4-230

Interleaving 4-230

Write-Once-Read-Many (WORM) Drives 4-234

Disk Emulation 4-235

Creating a Bootable Drive 4-238

***WHY FLASH?*—CHARACTERISTICS OF INTEL FLASH MEMORY** 4-238

Power Consumption Comparison (Watts) 4-238

Data Access Time 4-238

Reliability 4-239

Weight 4-240

SUMMARY 4-240

APPENDIX A 4-241

SCHEMATICS 4-247

INTRODUCTION

Mass storage encompasses many different technologies. Though commonalities exist, mass storage strives for nonvolatility, low cost per bit, and high density. Disk drives provide the best known example. However, many environments now require higher performance and reliability with lower power consumption, even at the expense of capacity. Flash memory uniquely meets these demands.

Flash memory can be used as a mass storage medium in applications including factory automation, notebook computers, high-end workstations, point of sale terminals, and data acquisition systems. Even desktop computers benefit from solid-state storage. The motivation to incorporate flash memory in any of these applications becomes obvious to the system designer who understands flash memory's benefits and density projections.

In an effort to understand these benefits, this document includes both conceptual and application oriented discussions. These discussions will be kept to a minimum with the real focus being on specific design techniques and considerations.

ADVANCED PACKAGING

Mass storage is synonymous with high density. Disk drives have increased the bit density of the rotating media via material improvements and closer tolerances. For semiconductors, density requires advanced packaging as well as higher capacity silicon (improved photolithography). Intel's Flash Memory devices are based on the company's EPROM Tunnel Oxide (ETOX™) technology that enables the high degree of scaling required to achieve high density.

Intel offers the high density flash memories in several package types. The standard packages are the Plastic Dual In-line Package (PDIP), the Plastic Leaded Chip Carrier (PLCC), and the Thin Small Outline Package (TSOP). Advanced modular packaging in the form of PCMCIA compatible memory cards and Single In-line Memory Modules (SIMM) provide the total solution.

Which package is best for your application?

Plastic Leaded Chip Carrier (PLCC)

The engineer striving to reduce board space is already using surface-mounted technology, such as PLCC. The PLCC is seen frequently on PC add-in cards and motherboards. Compared to the DIP, PLCC uses as little as 35% the overall board space. Its small size, compared to the DIP, is attributed to the terminal center-to-center spacing—50 mils versus 100 mils—as well as its four-sided pinout. No drilling or lead-cutting is necessary as leads are soldered directly to pads on the circuit board. The PLCC's 50-mil pad pitch is compatible with most circuit board manufacturing equipment. Additionally, components can be mounted on both sides of the board. However,the four-sided PLCC generally requires the use of a multi-layered board to lay out conductor traces for maximum compaction.

Thin Small Outline Package (TSOP)

When overall space constraints are critical, the TSOP is the best choice. This is best exemplified by IC memory cards. Low height is the key attribute of the TSOP, measuring 1.2 mm versus 3.5 mm for the PLCC. (Mechanical drawings in Appendix.) State-of-the-art center-to-center terminal spacing of 0.5 mm yields a smaller package and narrower conductor traces than the PLCC or DIP. In comparison, the volume of the TSOP is 172.8 mm^3 versus 656.3 mm^3 for the PLCC and 1872.3 mm^3 for the DIP.

The TSOP is available in standard and reverse pin configurations (Figure 1). Pins are located on only two ends of the package. This approach simplifies trace layout while reducing the number of board layers because traces can be routed out the non-leaded sides of the devices. Very dense board layouts are accommodated because components can literally be laid out end-to-end and side-by-side. Figure 2 displays an optimal layout best utilizing the TSOP's attributes. The close spacing allows one bypass capacitor to be used for two devices (provided they are not simultaneously selected). This optimal component layout can be mirror-imaged through the board to easily double the memory capacity.

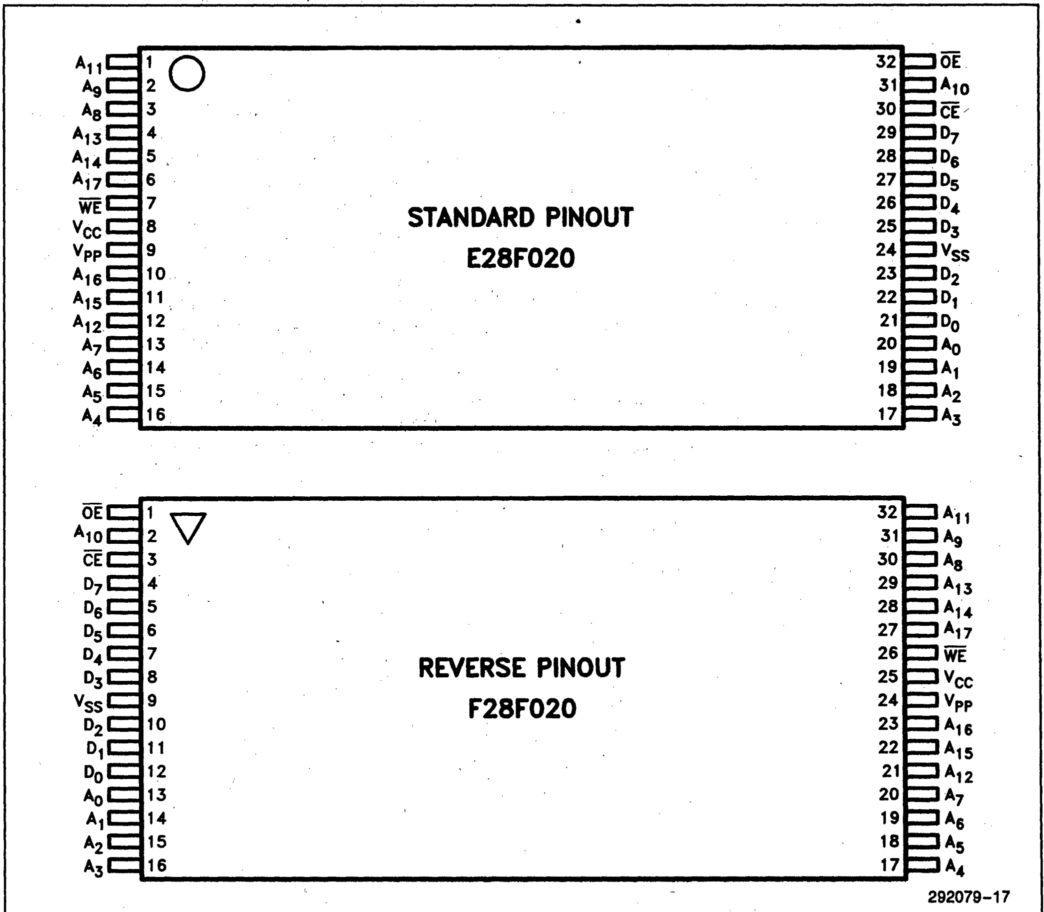


Figure 1. 28F020 32-Lead TSOP—Standard and Reverse Pinouts

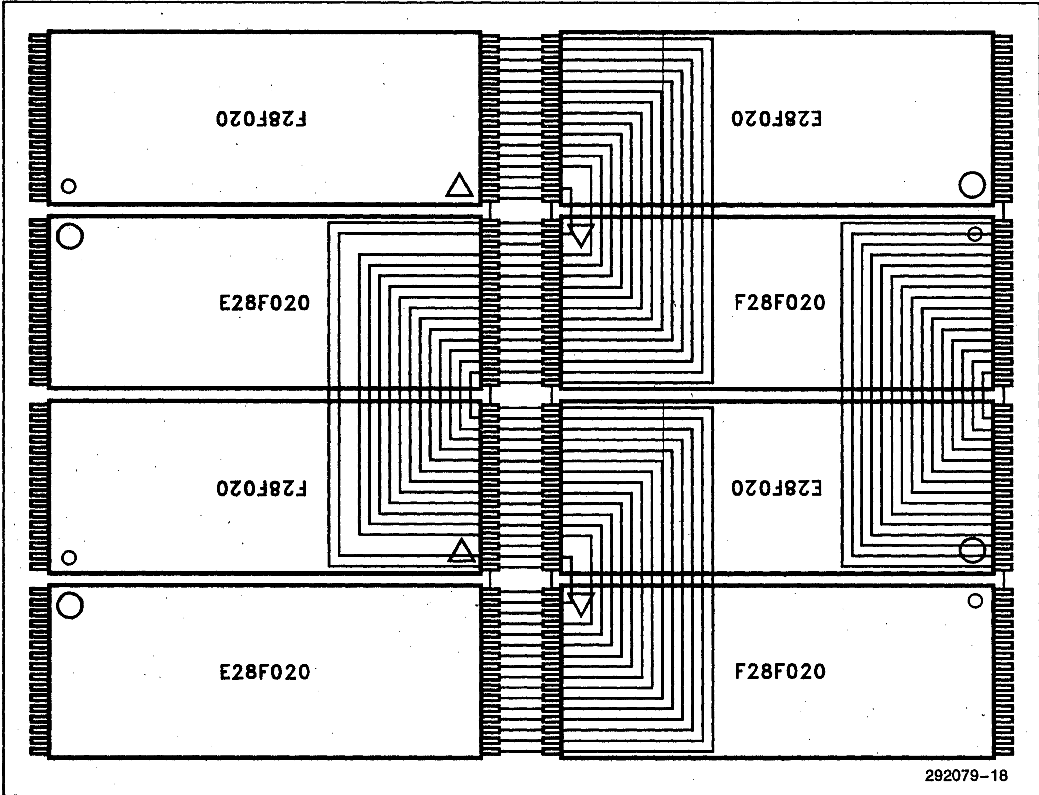


Figure 2. TSOP Optimal Layout: Highest Density Configuration (Conceptual)

Memory Cards

Many laptop and notebook computer manufacturers are pursuing the IC memory card to incorporate a removable mass storage medium. This is an ideal application for the Intel Flash Memory TSOP, due to the package's minimal height.

Solid-State Memory Alternatives

ROM and SRAM are currently the dominant IC card memory technologies. ROM has the advantage of being inexpensive, but is not changeable. When newer software revisions (e.g. Lotus* 123, Wordstar**, etc.) are available, the user must buy a new ROM card for each upgrade. Intel Flash Memory's reprogrammability minimizes the user's expense and the OEM's inventory risk.

SRAM is reprogrammable but batteries are required to maintain data, risking data loss. Like magnetic disks, flash memory is truly nonvolatile and thus has virtually infinite storage time with power off (10 years minimum, 100 years typical). Additionally, SRAM is expensive and not a high density solution. Intel Flash Memory provides a denser, more cost effective and reliable solution.

System level cost is about the same for Intel Flash Memory and SRAM + battery—

Flash memory requires 12V for programming and erasing. If a 12V supply is not available, 5V can easily be boosted. (See Application Note AP-316.) SRAM + battery requires battery state detect circuitry.

Card level cost differences are substantial (Figure 3)—

SRAM must have a battery to retain data. It also requires a V_{CC} monitor and Write Lockout circuitry. Intel's Flash Memory only requires Write Lockout circuitry (switching V_{PP} to 0V is an alternative write protect). This leads to increased area for memory components. More importantly, Intel's Flash Memory density is 4 times that of static RAM, yielding for lower cost per bit.

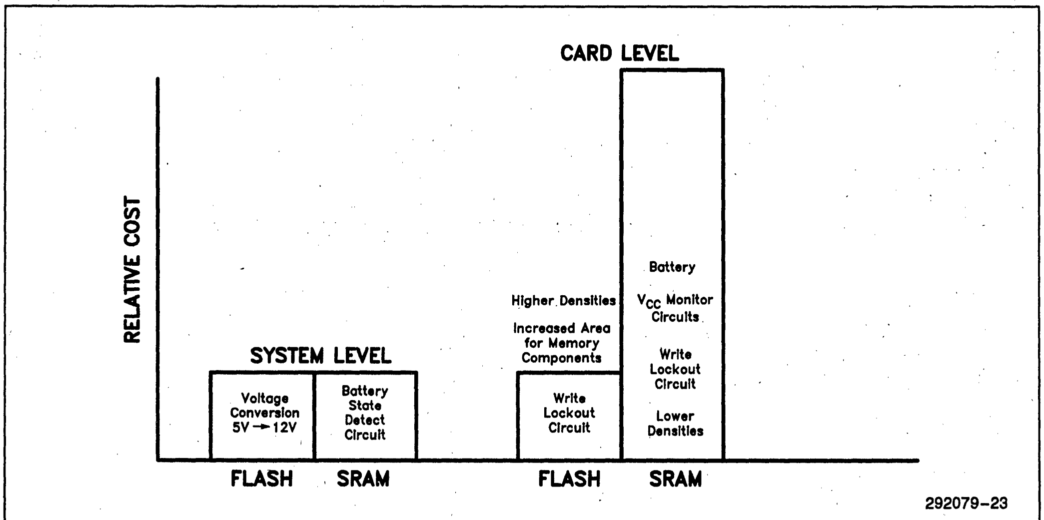


Figure 3. Support Circuitry Cost Comparison

*LOTUS® is a registered trademark of LOTUS Development Corporation.
**WORDSTAR® is a registered trademark of MICROPRO.

Designing a PCMCIA/JEIDA Standard Memory Card

Choosing among IC card design options depends on card architecture (standardization), memory capacity, data bus width, card intelligence, V_{PP} generation, and reliability.

What are the advantages of a standardized memory card pinout?

From the computer system's viewpoint, a standardized pinout enables the use of multiple third-party memory cards. This ensures competitive pricing and wide availability. From the memory card point of view, standardization allows use in a variety of systems.

The Personal Computer Memory Card International Association/Japan Electronic Industry Development Association (PCMCIA/JEIDA) 68-pin format is the emerging IC memory card standard. Several proprietary formats are also available from their respective manufacturers, but these same manufacturers now offer PCMCIA/JEIDA versions. The PCMCIA/JEIDA standard specifies physical, electrical, information structure, and data format characteristics of the card. This standard accommodates either 8- or 16-bit data bus widths.

The following 2 Mbyte memory card design provides a byte-addressable interface using 8-28F020s (2 Mbit, 256k x 8 devices) as shown in Figure 8. While TTL equivalent interfacing is shown, most cards will use gate arrays to reduce chip count. Address lines A18 and A19 are decoded with a 2-to-4 decoder (74HC139) to generate high and low byte chip select signals for each of the 4 pairs of flash memory devices (one pair = high and low byte). The PCMCIA/JEIDA format specifies inputs $\overline{CSL}$ and $\overline{CSH}$ (along with the A0 address line) which select the low and high byte, respectively.

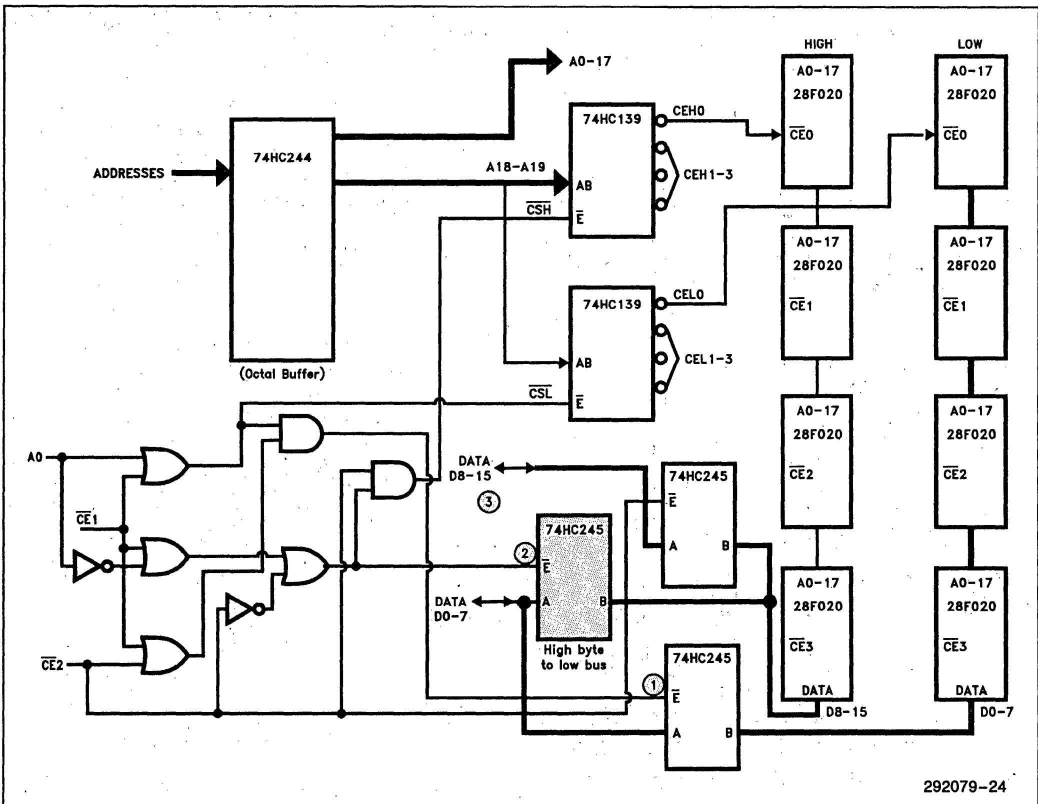


Figure 4. Decoding for PCMCIA/JEIDA Standard Bus Interface

According to the PCMCIA/JEIDA standard, the memory card is designed with the flexibility to have both an 8-bit or a 16-bit interface, dependent upon the machine it is plugged into. When the memory card is plugged into an 8-bit system, the high byte transceiver is multiplexed to the low byte of the system. In Figure 4, the highlighted transceiver (#2), maps the upper byte to the lower byte of the data bus (i.e., D_{8-15} to D_{0-7}). Signals are decoded according to the truth table in the Appendix. (1, 2, and 3 denote transceiver numbers of Figure 8.)

One can double the memory capacity and select from among 8 pairs of flash memory devices by using a 3 to 8 decoder with inputs A_{18-20}. Notice that additional transceivers are not needed to support the additional data fanout. (See section on capacitive loading.)

Single In-Line Memory Module (SIMM)

The SIMM is optimal where minimized board space and upgrade capability are required. Compared to using 8 discrete PLCCs plus capacitors (3019.4mm^2), the equivalent memory capacity SIMM (926.1mm^2) consumes 70% less motherboard real estate.

The SIMM can be built as an 80-pin, 0.050 mil centerline lead spaced, insertable module designed with a 16-bit wide data bus interface. The SIMM pin configuration allows convenient implementation:

- No Address or Data Bus Multiplexing—RAS# and CAS# are not needed;
- Reserved Pins—For product expansion and enhancements: Upgrade capability to 128 Mbytes;
- Presence Detect Eliminates Jumpering—Simplifies user installation.

The 80-pin definition of the flash memory SIMM includes 7 pins for Presence Detect (PD). (See Appendix). The PD pins are read to determine module memory capacity and speed of the devices. The PD pins are either **O**pen circuit or **S**horted to ground. By attaching a pull-up resistor to each pin, **O**pen circuits will read as a binary 1 and Shorts as a binary 0. Before implementing the presence detect feature, define your system criteria:

How many modules will be used?

Decide how much total memory your system is to contain. The limit is dictated by the space available, as well as cost.

Flash memory SIMMs can easily accommodate different memory capacities and speeds. Could your system handle mismatched SIMMs?

There are two basic design implementations for interpreting presence detect information. The first approach requires that matching SIMMs are used. The PD pins of all SIMMs are tied to one transceiver that is read as an I/O port (Figure 5).

Invalid reads occur if the user installs mismatched SIMM configurations. Any PD pin shorted to ground makes an open circuit pin appear as a binary zero (0). Mixing module speeds is acceptable, but the PD pins reflect the slower module.

The second approach, allowing any mixture of flash SIMMs, requires more hardware and software for interpretation. The PD pins from each SIMM have separate transceivers, resistors, and I/O ports (Figure 6). Flexibility is increased at the expense of board real estate.

Assume your system accommodates several SIMMs but complete population is not needed. Can the system handle empty sockets?

SIMM upgrade capability is not limited to increases in memory density. A system may be designed with several SIMM sockets on the circuit board. To keep initial end-customer costs down, the system ships with only one SIMM installed. This provides the option of populating the empty sockets at a later time. The PD pins are designed to eliminate jumper or software setups by the end-user when SIMM upgrades are made.

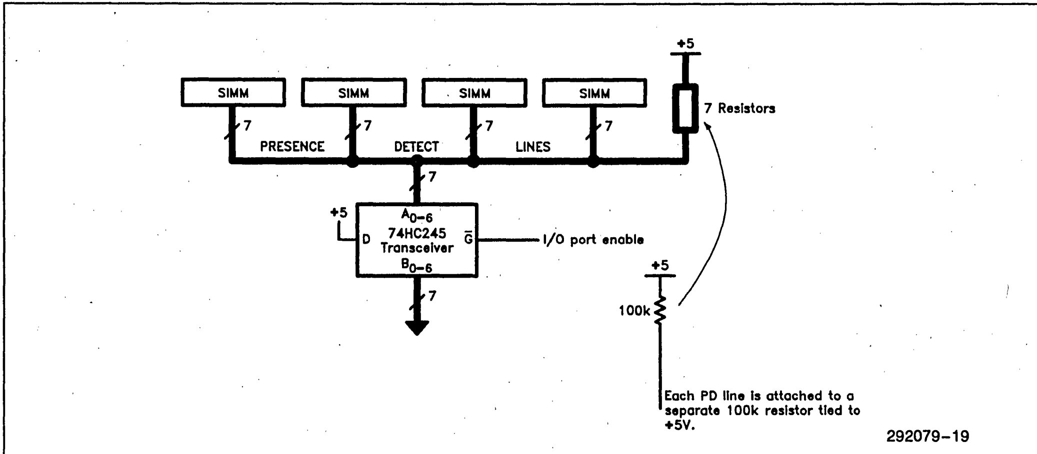


Figure 5. All SIMMs Should Reflect the Same PD Configuration

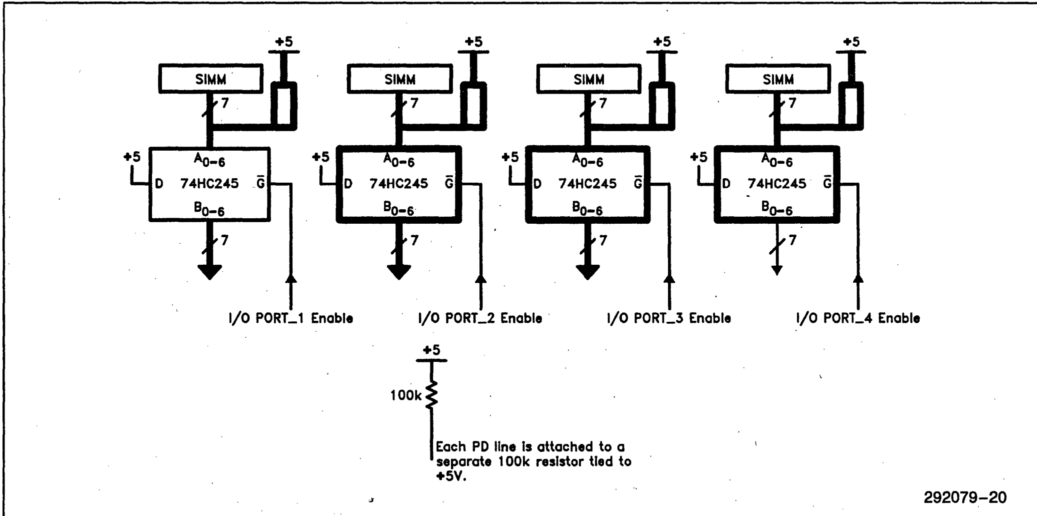


Figure 6. Multiple I/O Transceivers Are Needed if Mismatched SIMMs Are Used

Using the previous scenario, will it matter which socket is used? In other words, what is the installation procedure?

With respect to the PD feature, it does not matter which sockets are full. (However, most designers request that sockets are filled in sequential order to minimize hardware and softwear requirements.) To explain this, look again at the bit-level interpretation of the PD pins. An empty socket also appears as an open circuit. Your software can determine a full (or empty) socket in one of two ways:

Method One (Figure 5)—Reading the PD pins is insufficient. An empty socket will reflect the value of the full socket. Your software will have to read the chip level device identifier hardwired in each flash memory device. (See Intel Flash Memory data sheets regarding inteligent identifiers.) Reading an invalid device identifier from a SIMM address signifies an empty socket. Software demonstrating the use of this method to determine memory capacity is discussed further in the section on "Verifying Paged Memory Board Functionality".

Method Two (Figure 6)—Each SIMM's PD pins are read separately. Reading all ones (the result of all Open circuits) signifies an empty socket. The chip level device identifiers should still be read to establish the number of flash memory devices on the SIMM.

Presence Detect for WAIT-State Interpretation

Using Method One or Method Two from above, the device speed information is read from the pins. This information can be interpreted by software to issue the proper command to the system's programmable WAIT-state generator. By guaranteeing the use of matching SIMMs, the WAIT-state generator would not have to be reprogrammed each time a different SIMM is accessed.

A hardware driver alternative implements an 85C220 EPLD configured with an internal counter (Figure 7). The rising edge of the clock, following Chip Enable going active, latches the count value derived from the PD speed pins (Figure 8).

Each subsequent rising edge of the clock input decrements the counter. A $\overline{READY}$ signal is output to the CPU (or the system's READY logic circuitry) when the count reaches zero (0). The $\overline{READY}$ signal remains active until LOAD (Chip Enable, $\overline{CE}$) goes inactive at the completion of the bus cycle.

The clock signal for the internal EPLD counter is derived directly from the CPU, therefore the count rate and WAIT-states will be system dependent. An EPLD Advanced Design File was generated to demonstrate this application. (See Appendix A.) This is a straightforward approach until designing systems, such as power-saving laptops, that have changeable system clock rates.

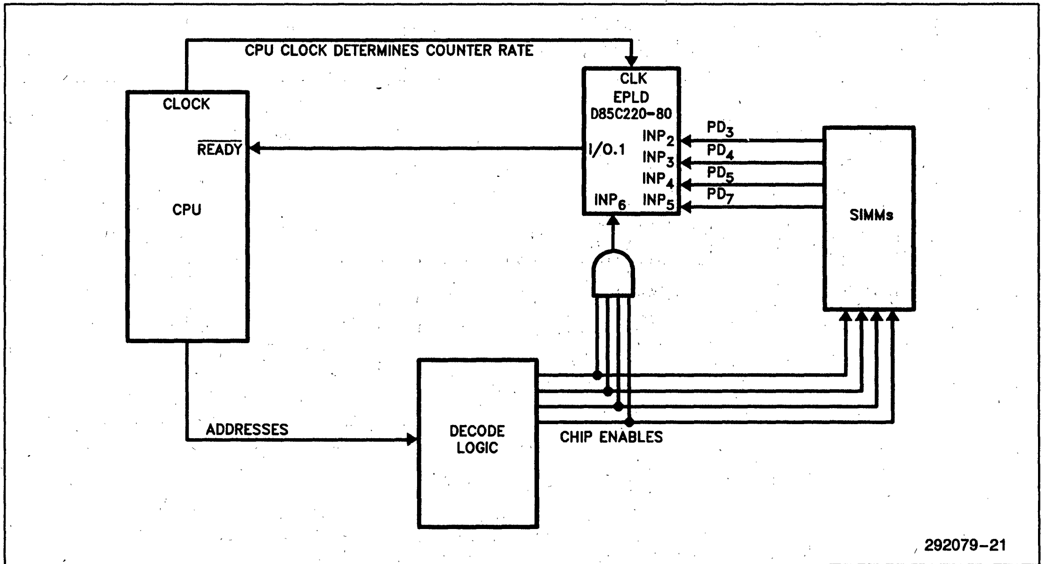


Figure 7. WAIT-State Generator Using an EPLD Configured as a Counter

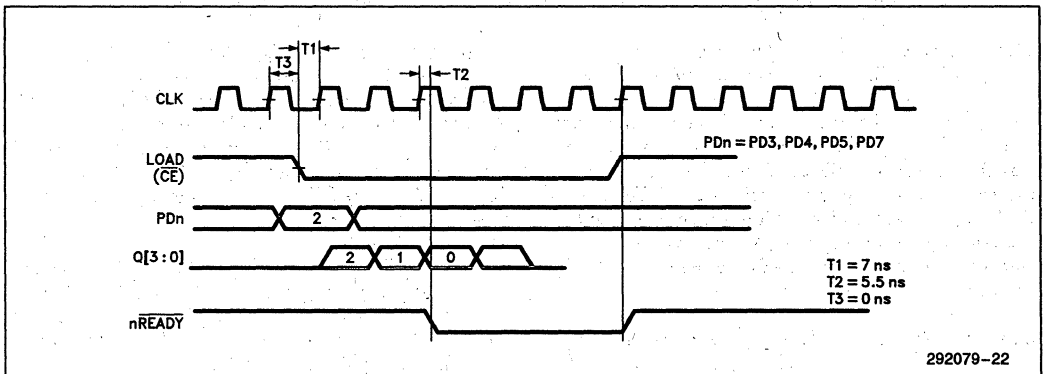


Figure 8. Timing for SIMM Presence Detect WAIT-State Generator

HARDWARE DESIGN IMPLEMENTATIONS

Paged, linear, and I/O are the three fundamental addressing methods that can be used for accessing an array of memory devices. Linear addressing offers the fastest and most direct access to a memory array. It consumes the largest portion of the system's memory and is only practical in a 386™ microprocessor (or other 32-bit processor) family system because of the large memory space available above 1 Mbyte. The I/O mapped memory array consumes the smallest amount of the system address space but has the lowest performance. A page-mapped memory array, also called a sliding AT window, is a hybrid of the linear and I/O designs. The memory array is usually very large relative to the system interface, consisting of pages typically ranging in size from 8 Kbytes to 64 Kbytes. (LIM-EMS use four to twelve 16 Kbyte pages.)

Design Example—A Paged-Mapped Memory Board

A paged design employs addressing techniques similar to the Lotus-Intel-Microsoft expanded memory specification (LIM-EMS). It allows access to one or more sections (or pages) of the flash memory array at a time. This minimal interface is particularly useful within the DOS 1 Mbyte memory space. The DOS map (Figure 9) shows 128 Kbytes of memory space available in the Optional I/O Adapter ROM area. LIM-EMS, LAN, the flash memory design discussed in the following sections, and other accessory cards can use this area.

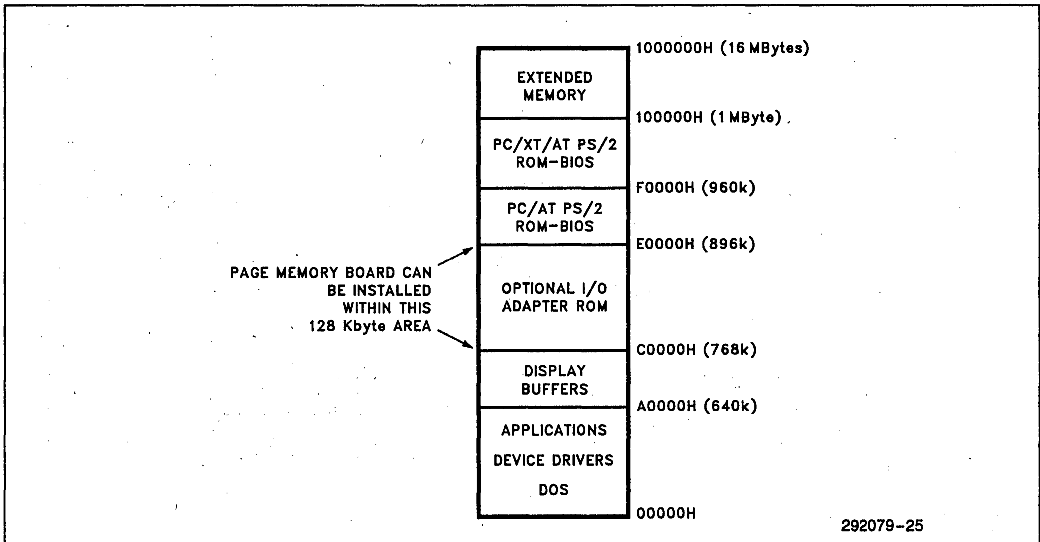


Figure 9. DOS Memory Map

Figure 10 shows the block diagram of the page-mapped flash memory board design. (Except for the addressing method, all the functional components of this board could be used on a linear or I/O mapped flash memory array.) This PC-AT*** compatible design example consists of a flash memory array (using SIMMs) and the corresponding memory and I/O decoding, V_{PP} generation, and the interface to the system bus. (Component numbers shown with the following diagrams correlate with the actual schematics in the appendix.) A page size of 64 Kbytes is used. Depending on the system's configuration, memory contention may require a smaller page size. (Note that the LIM EMS 4.0 standard uses 4 contiguous 16 Kbyte pages. Multiple pages can exist as space permits.)

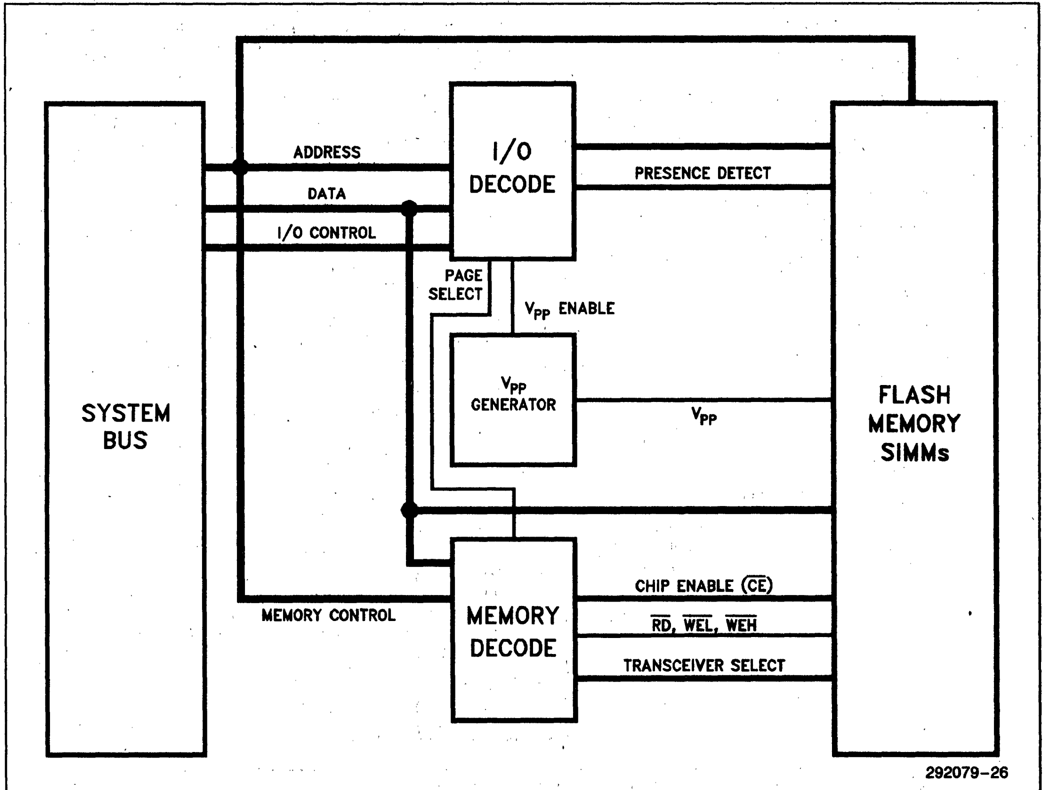


Figure 10. Page-Mapped Flash Memory Board

NOTE:
A similar hardware platform using Intel Flash Memory Cards is contained and described in the Hardware Reference Manual that can be ordered separately through Intel literature (Order Number 296871).

***PC-AT® is a registered trademark of International Business Machine Corporation.

The Decoding Scheme

The Intel Flash Memory on this board is installed in 4 SIMM sockets. With a fully populated board, the memory capacity ranges from 4 Mbytes to 16 Mbytes depending on the SIMM density used.

Depending on the density, up to eight chip enables, $\overline{CE}_0$–$\overline{CE}_7$, are used on a SIMM (4 $\overline{CE}$s for 8-chip, 8 $\overline{CE}$s for 16-chip SIMMs). Standard decoding techniques generate separate chip enables, output enables, and write enables. This method has the disadvantage of having to accommodate a large number of traces. The addressing scheme incorporated in this design minimizes the number of board traces neeed to select individual devices. Device selection is made on a row-column basis where: rows are Output Enables ($\overline{OE}$s), Write Lows ($\overline{WRL}$s), and Write Highs ($\overline{WRH}$s) and columns are Chip Enables ($\overline{CE}$s). (For low-powered systems, this method may be unacceptable because each chip enable activates a maximum of 8 components.) These signals are generated by decoding the page lines P_3–P_7 (Figure 11, U22). (See Page Number section.) Pages within a component are selected by tying P_0, P_1 and P_2, respectively, into pins 37 (A_{15}), 36 (A_{16}), and 35 (A_{17}) on the SIMM (Pin 35 is a no-connect (NC)).

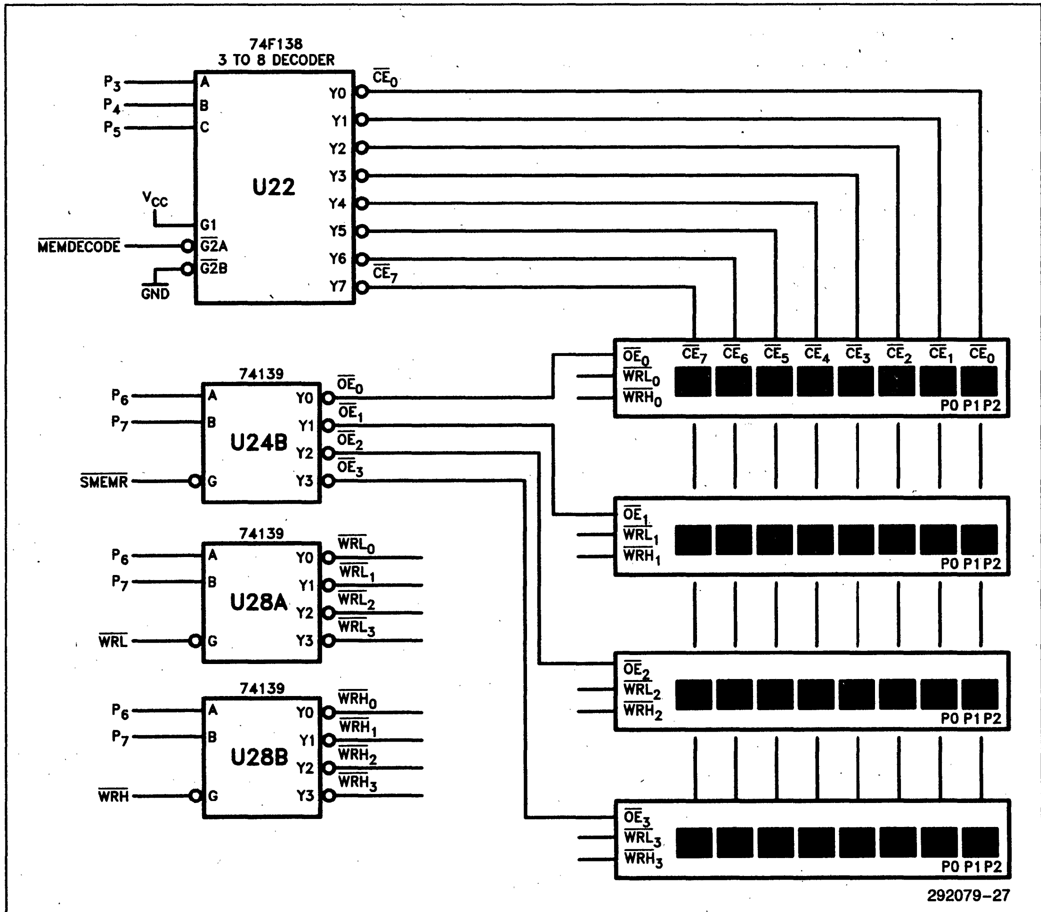


Figure 11. Individual Components Selected by Row-Column Addressing Saves Board Traces

Planning for upgrades also presents another interesting situation. The 1 Mbyte SIMM has four chip enables ($\overline{CE_0}$–$\overline{CE_3}$), one for each pair of components. The pair of components represent high and low bytes and are selected by $\overline{WRH}$ and $\overline{WRL}$, respectively.

The 1 Mbyte SIMM represents sixteen 64-Kbyte pages (eight 128-Kbyte components). To accommodate upgrade capability, pages will not be contiguous because a "4-page hole" exists every 4 pages (P2 is attached to a no connect). To overcome this rearrange the page select lines with jumpers (Figure 12):

1 Mbyte	SIMM	(8 * 28F010s)	JP2, JP4, JP6, JP8
2 Mbyte	SIMM	(16 * 28F010s)	JP2, JP3, JP7, JP9
2 Mbyte	SIMM	(8 * 28F020s)	JP1, JP3, JP5, JP7
4 Mbyte	SIMM	(16 * 28F020s)	JP2, JP3, JP6, JP7

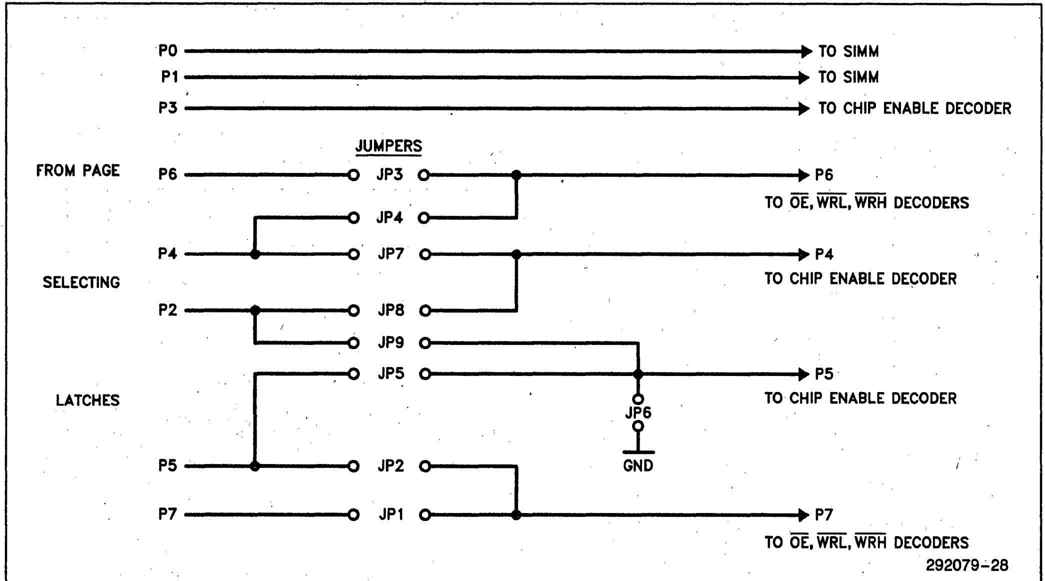


Figure 12. SIMM Density Upgrading Jumpers

If a system is designed that uses PCMCIA/JEIDA standard memory cards instead of SIMMs, this decoding is greatly simplified. The memory card is treated like a large memory array. Using a 64 Kbyte page size as an example:

Address lines A_{0-15} are supplied directly from the system address bus (after buffering). Address lines A_{16-23}, which select the pages, are sent as data to a latch before entering the memory card (Figure 13).

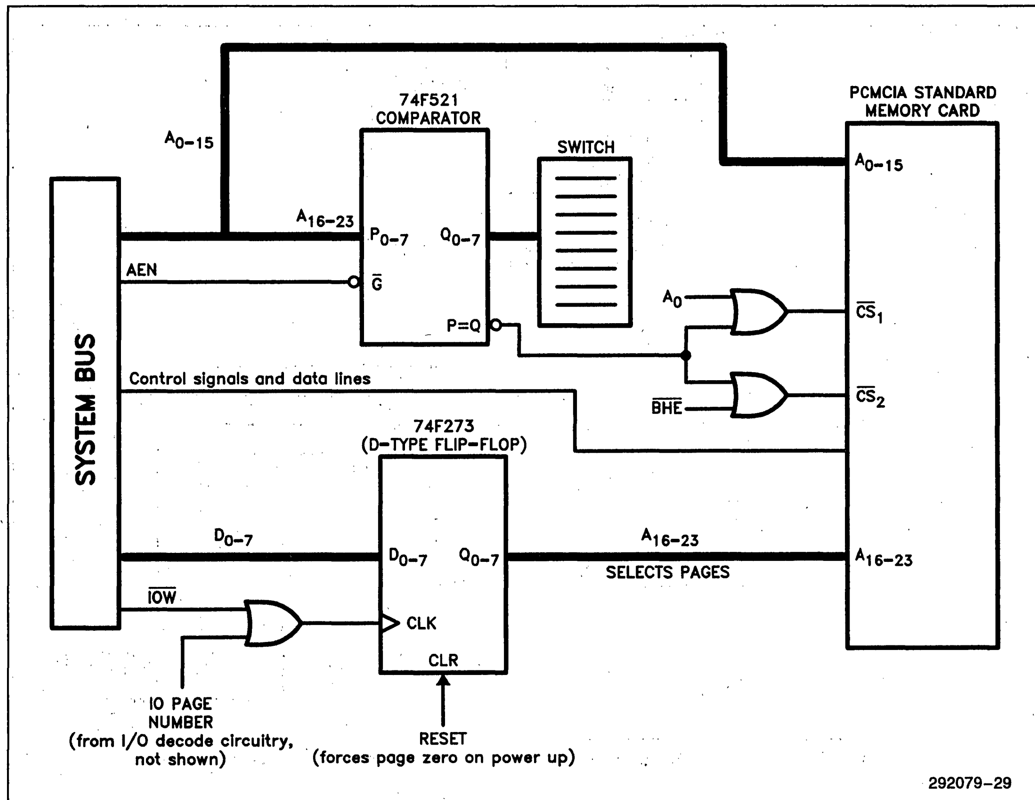


Figure 13. Memory Card Interfacing

The page inputs to the "Chip Enable" decoder (Figure 14, U22) are redefined as follows:

P3 = P3, P4 = P2, P5 = GND,

P6 = P4, and P7 = P5.

For a better understanding, you should verify the bit combinations while stepping through the first few pages. Notice that the sequencing of page numbers does not correspond linearly with the Chip Enables. This is not significant because the data is read the same way it is written.

Page Numbers	Chip Enable
0, 1, 2, 3	0
4, 5, 6, 7	2
8, 9, 10, 11	1
12, 13, 14, 15	3

NOTE:
Linear Page Selection Results in Nonlinear Chip Enable.

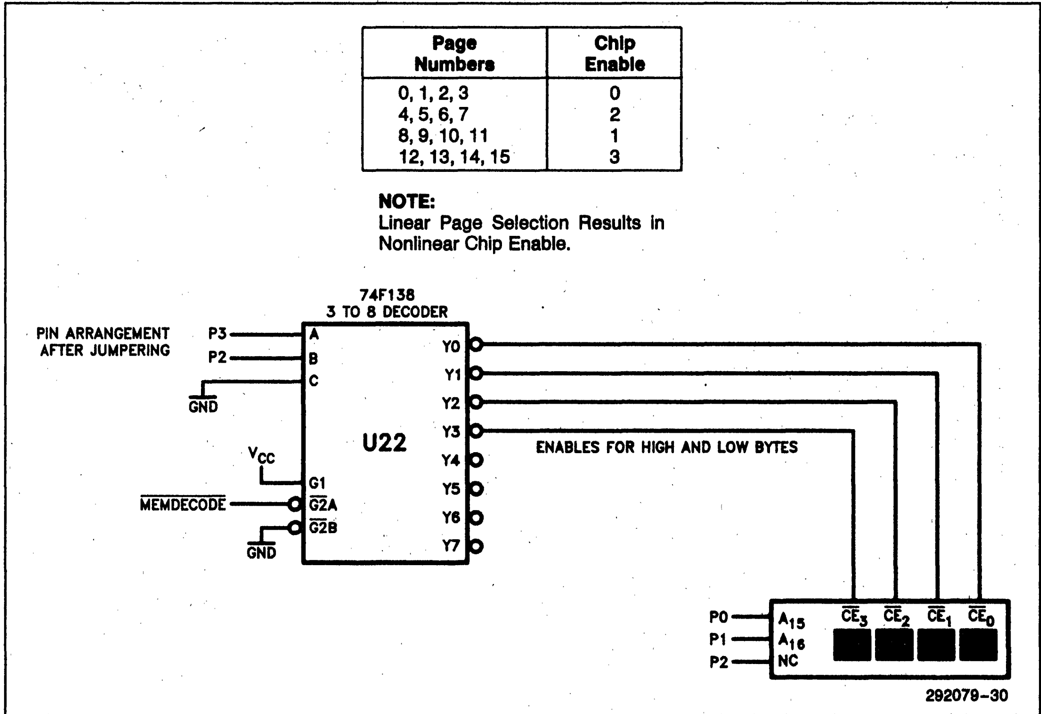
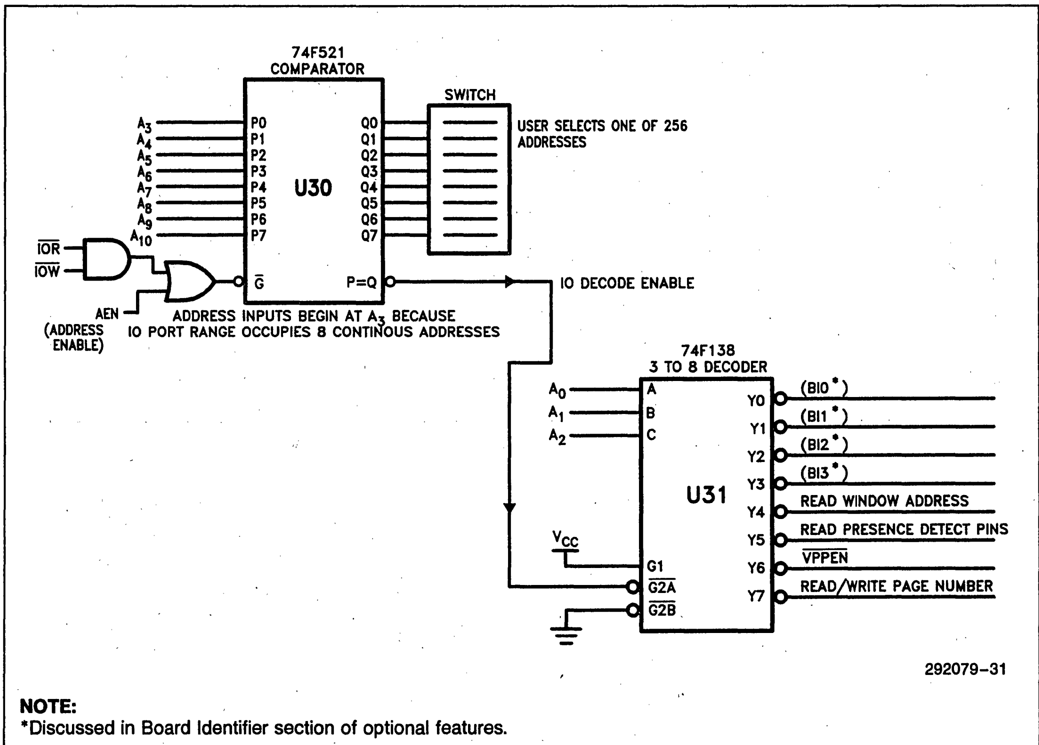


Figure 14. Component Selection Relative to Page Number for 1 Mbyte SIMM

I/O Decoding

Multiple functions can be implemented with I/O decoding access. Some examples include: reading the current window address, reading the presence detect pins, enabling V_{PP}, and reading/writing the page number. The eight consecutively addressed I/O ports on this board (4 reserved for optional features) are located at a user-selectable address. This base I/O port address is setup on an 8-byte boundary by using A_{3-10} as inputs to the 74F521 comparator (Figure 15, U30). When any of the eight consecutive I/O port addresses matches the dip switch settings (and AEN is low), the comparator outputs the I/O Decode Enable (to decoder U31).

AEN (address enable), the chip select for the 74F521 comparator, is supplied by the PC I/O channel. It distinguishes processor bus cycles from DMA bus cycles. A high on AEN indicates that a DMA (or DRAM refresh) cycle is in progress and we must stay off the bus. The enables for the 74F138 IODECODER (U31) are provided by $\overline{\text{IODECODE}}$ ENABLE along with the "ANDing" of $\overline{\text{IOR}}$ and $\overline{\text{IOW}}$. This decoder selects the I/O ports that access the page window address, the SIMM presence detect pins, the V_{PP} Enable, and the page number. Each of these I/O ports are described in detail:



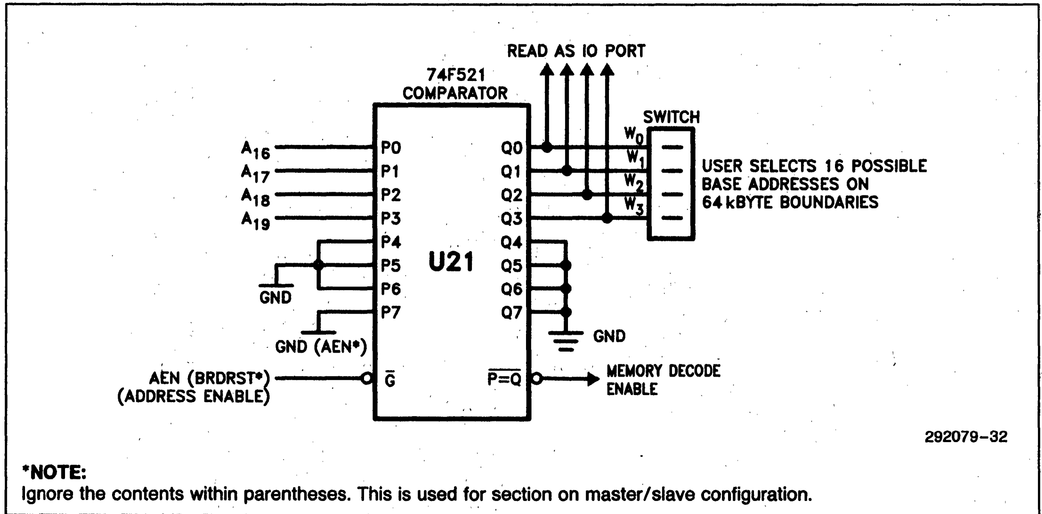
NOTE:
*Discussed in Board Identifier section of optional features.

Figure 15. User Selectable I/O Base Address for I/O Decoding

The Window Address

The user-selectable window address can be set up on any 64K boundary below 1 Mbyte. (The memory window should be placed between C0000h and E0000h to be DOS compatible.) A DIP switch (connected to a transceiver for reading) and the four address lines A_{16-19} are the inputs to the 74F521 comparator (Figure 16, U21). There are 16 possible window addresses. The comparator outputs the "Memory Decode Enable" signal when an address is selected that is within the 64 Kbyte window. This signal (with AEN low) allows board level memory decode.

The location of this 64 Kbyte window can be moved above 1 Mbyte by adding A_{20-23} to the comparator's inputs P_4 to P_7 of the 74F521. Bits D_{4-7} of the data bus can be connected to the comparator's pins Q_4 to Q_7 to allow reading of the full base memory address.



*NOTE:
Ignore the contents within parentheses. This is used for section on master/slave configuration.

Figure 16. User Selects Base Memory Address

Presence Detect

The method shown earlier in Figure 3, is used to configure the PD pins in this design. SIMMs can be added incrementally only in similar densities. The SIMM PD pins are read by selecting the appropriate I/O address that enables the 74F245 transceiver.

V_{PP} Generation

V_{PP} is generated locally (on this board) to ensure a stable, switchable 12V (±5%) supply. (Many systems generate their own 12V power supply. However, it should not be used if its regulation is greater than 5%.) On power-up, system reset, or when V_{CC} is below 4.5V, V_{PP} is forced off. It is enabled (or disabled) by writing to the I/O port address (Figure 15, U31) that generates the $\overline{VPPEN}$ signal. This on/off capability is essential for battery-operated equipment and eliminates the need for $\overline{WE}$ filtering (as discussed below). (See Intel data sheet for V_{PP} standby current.) The $\overline{VPPEN}$ signal "ORed" with the system I/O write, $\overline{IOW}$, functions as the clock signal for the 74HC74 D-flip flop (Figure 17, U42A). The D-input is latched when $\overline{IOW}$ goes high. Writing a one or a zero turns V_{PP} on or off, respectively.

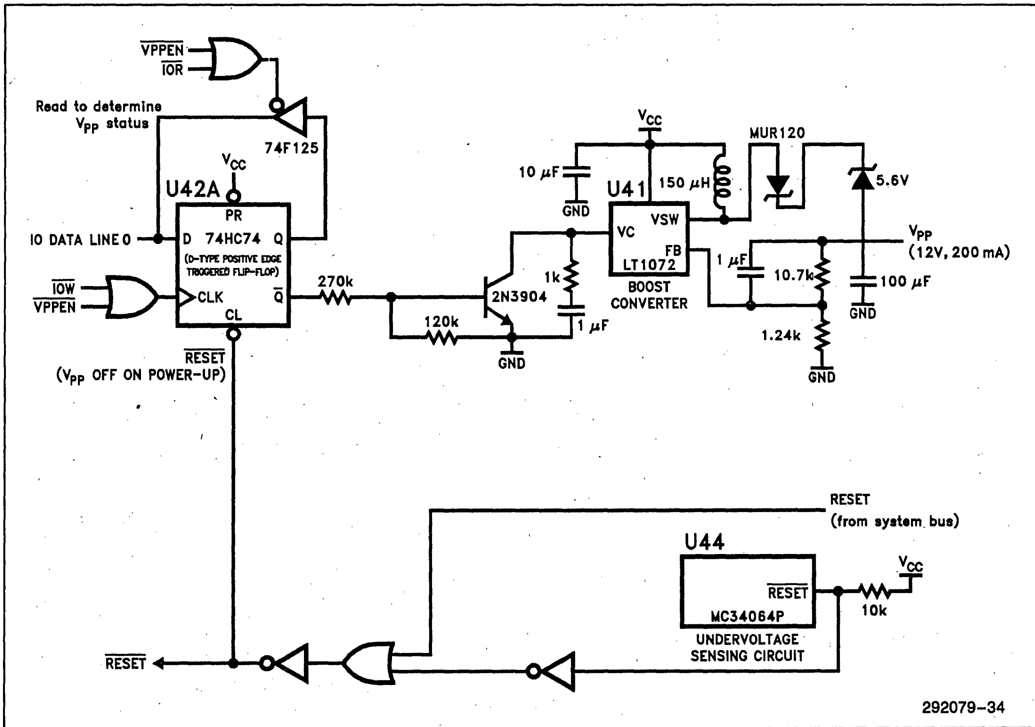


Figure 17. V_{PP} and $\overline{RESET}$ Generation

Linear Technology's LT1072 (U41) switching regulator is used as a 5V to 12V boost converter. The FB input regulates the voltage output. The 10.7k and 1.24k resistors are used to establish the correct reference voltage to obtain 12V. The 100 μF capacitor at the output is used to handle up to 200 mA. (See Linear Technology's LT1072 data sheet for more information.) Typically this will be much more than needed and a smaller capacitor can be used. However, this will accommodate interleaving of 8 components but may not be practical in a battery-operated system. (See section on Interleaving in the Software Design Implementation chapter.) Additionally, sufficient time should be allowed when switching V_{PP} on. The delay is a factor of the load on the line and the quality of the passive components chosen. The diode, MUR120, keeps the inductor from absorbing current from the charged output capacitor. The 5.6V zener diode ensures that when V_{PP} is less than 5.6V, the V_{PP} output is held at 0V. (This is optional if $V_{PP} \leq$ 5V is tolerable.)

During system power-up, some probability exists that noise may generate spurious writes which are actually the sequence of flash memory commands that initiate erasure or programming. Power-up protection in this design is provided by disabling V_{PP} until voltages have stabilized. The Motorola component, MC34064P (U44), is an undervoltage sensing circuit that begins functioning when V_{CC} is above 1V. Between 1V and 4.6V, the $\overline{\text{RESET}}$ output is active. The $\overline{\text{RESET}}$ output or a system RESET clears the 74HC74 (U42A), keeping V_{PP} off when V_{CC} is less than 4.6V. Alternatively, this signal, or a supply's "POWERGOOD" signal, may gate $\overline{\text{WE}}$ or $\overline{\text{CE}}$, as is common with battery-backed SRAM or EEPROM designs. As an example, the $\overline{\text{RESET}}$ output of the MC34064P can be tied to the active-high enable of the decoder to disable any $\overline{\text{CE}}$s until V_{CC} = 4.6V, as shown in Figure 18.

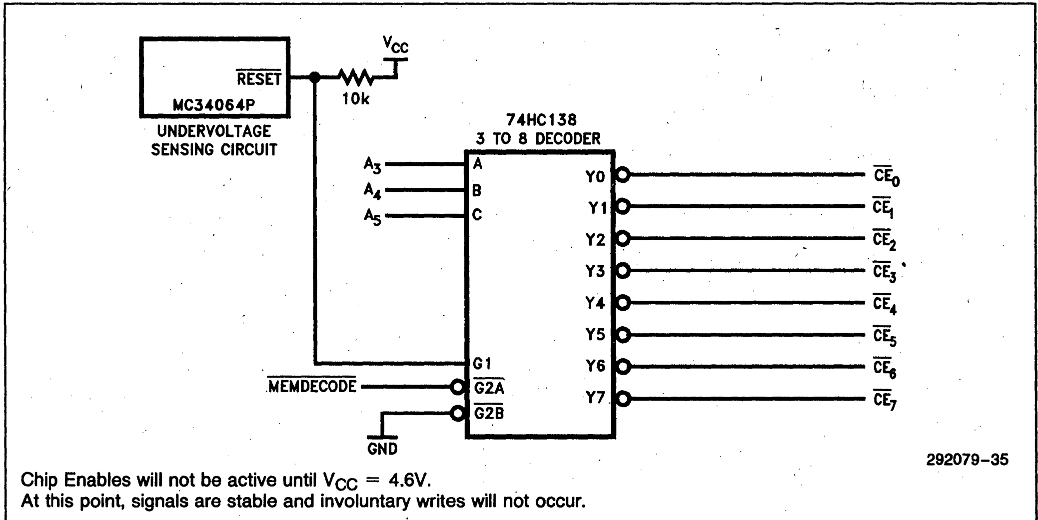


Figure 18. Protecting the Circuit from Involuntary Erasure and Programming. Use an Undervoltage Sensing Circuit, or a System's "POWERGOOD" Signal, to Control Chip Enables

How is V_{PP} Switched on (Refer to Figure 17):

Latching a one into the 74HC74 D-input (U42A) puts a zero on the output $\overline{Q}$. This turns off the transistor 2N3904. When the 2N3904 is off, the VC input of the LT1072 (U41) is 5V and the VOLTAGE SWITCH (VSW) output generates 12V.

Page Number Selection and Reading

It is standard practice to use an I/O port to generate the page number for this type of memory array. The potential number of pages that can be selected is determined by the size of the data bus as well as the amount of decoding the system can practically handle. In this design, this I/O port allows selection of 256 64-Kbyte pages, for a total of 16 Mbytes of flash memory. The page number is written to the 74F273, Octal D-Type Flip-Flop (Figure 19, U37). It is latched by the rising edge clock signal derived by the "ORing" of the corresponding 74F138 decode signal (I/O PAGE NUMBER) and the system $\overline{IOW}$.

Page zero is automatically selected on power-up because the 74F273 clear input is connected to $\overline{RESET}$ (generated as part of the V_{PP} circuitry). This feature ensures that the board will power up in page zero. Given the proper software, this board can be turned into the system's bootable drive. (See section on Software Design Implementations.)

The current page number can be obtained by reading the same I/O port. The I/O decoder output, I/O PAGE NUMBER, "ORed" with the system $\overline{IOR}$, produces the signal enabling the 74F245 bus transceiver (that is tied to the output of the 74F273).

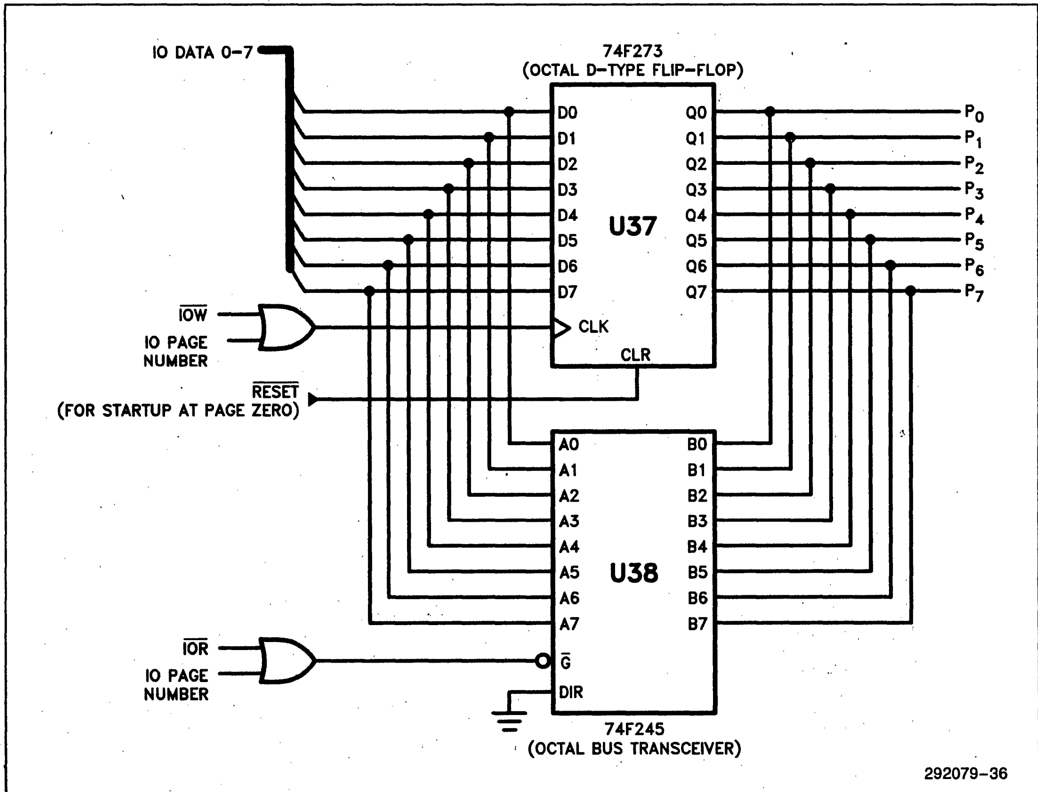


Figure 19. Selecting or Reading Page Number

Design Considerations

The SIMMs high and low bytes are enabled by $\overline{WEH}$ and $\overline{WEL}$, respectively. Using a high and low byte transceiver for each SIMM limits the capacitive loading and prevents performance degradation of the data bus. (This becomes important when upgrading to flash memory SIMMs that have 16 components. See section on capacitive loading.) Also, the PC I/O channel bus specification requires that no more than 2 TTL loads be present on any one line. Therefore, the SIMM transceivers must be routed through two additional transceivers at the PC bus interface (refer to "Switchable Data Bus Width" section). In this paged memory board design, the SIMM transceivers are enabled by a 2 to 4 decoder which uses page pins P_6 and P_7 as decode signals. The enable for the decoder is supplied by the $\overline{MEMDECODE}$ signal; transceivers are disabled unless an address within the 64 Kbyte page is accessed.

Optional Board Features

So far we have described the components required to design a functional flash memory array. Optional features can be added to make this board more versatile in an application environment:

Switchable Data Bus Width

This feature allows the board to execute in a PC XT* (8-bit bus) or a PC AT system (16-bit bus). Memory card designs for adopting the PCMCIA/JEIDA format must include similar provisions as shown earlier. At the PC-I/O channel interface, (for use in an 8-bit system), an extra transceiver is used to redirect the upper data bus (D_{8-15}) to the lower data bus (Figure 20, U9). The $\overline{16BIT}$ signal is generated from a ground on the PC AT I/O channel extension; it will be high (because of the pull-up resistor) when a PC XT is used. (The $\overline{16BIT}$ signal can be read by software through the 8th bit of the Presence Detect port.)

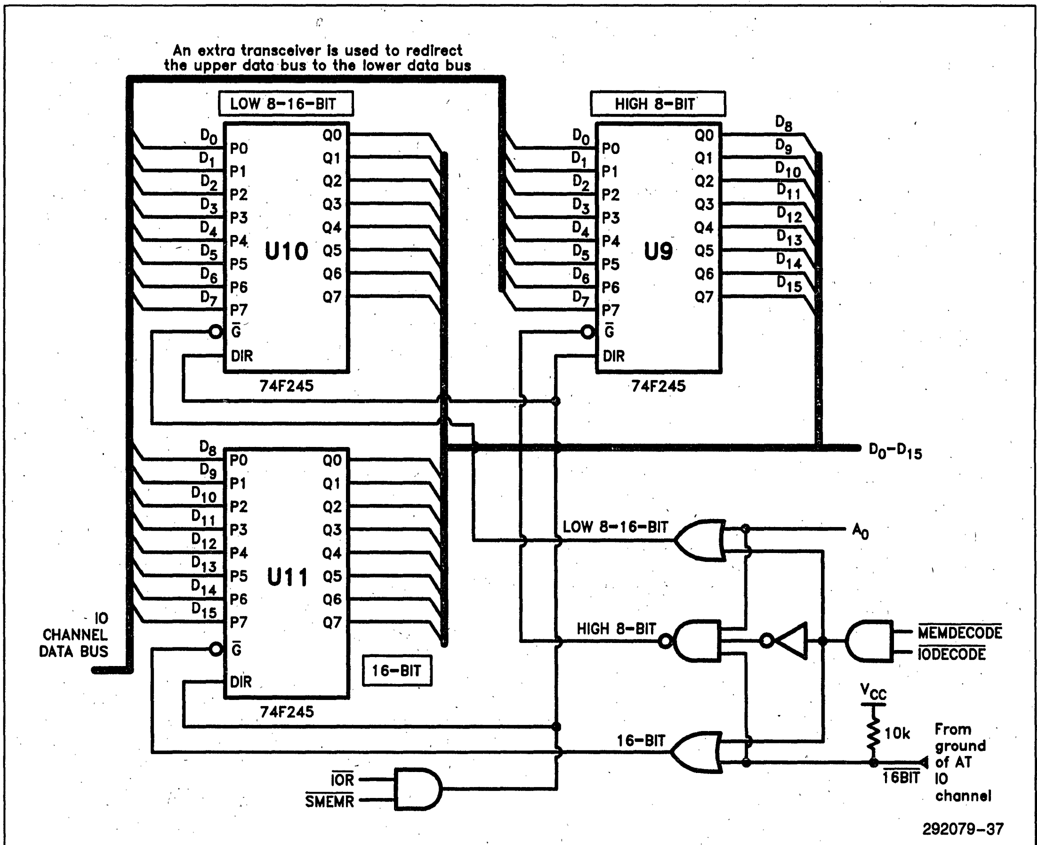


Figure 20. I/O Channel Transceiver Interface for 8- or 16-Bit Data Bus Selection

*PCXT® is a registered trademark of International Business Machine Corporation.

Access to a word (2 bytes) requires two bus cycles to generate two addresses in an 8-bit system. As an example referring to Figure 20, when accessing a memory word at address zero (0):

$\overline{16BIT} = 1, \overline{MEMDECODE} = 0;$

During access to the low byte → A0 = 0, so the signal "LOW 8/16 BIT" is active;

During access to the high byte → A0 = 1, so the signal "HIGH 8 BIT" is active.

The high byte from the SIMM is multiplexed onto the low byte of the system bus.

The circuitry at the SIMM transceiver interface determines whether to use the Bus High Enable ($\overline{SBHE}$) signal or A_0 to select the high byte. The $\overline{16BIT}$ signal selects the "A" or "B" inputs of the 74F157 multiplexer (Figure 21, U27). Regardless of the bus size, the $\overline{WRL}$ signal is generated on a system memory write ($\overline{SMEMW}$) to an even address (A0 = 0). During a 16-bit write, the $\overline{WRH}$ signal is generated by a system memory write to the high bus ($\overline{BHE}$). However in an 8-bit system, where $\overline{SBHE}$ is absent, the $\overline{WRH}$ signal is generated by a system memory write to an odd addressed byte (A_0 = 1).

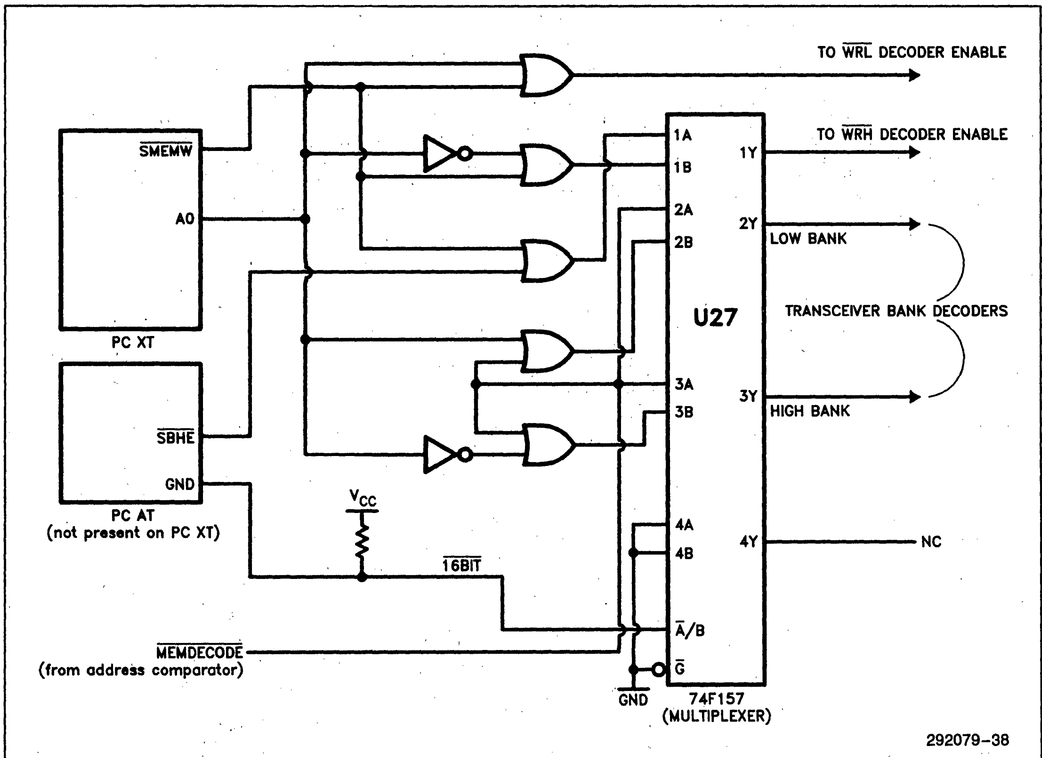


Figure 21. 8- or 16-Bit Data Bus Selection at the SIMM Transceiver Interface

The eight transceivers for the four SIMMs are selected by signals T_{0-7}. Even (T_0, T_2, T_4, T_6) and odd (T_1, T_3, T_5, T_7) numbered signals decode for the SIMM low and high bytes, respectively. The signals T_{0-7} are derived by decoding P_6 and P_7 (Figure 22, U24A) and the transceiver bank decoders (Figure 21, U27).

For a 16-bit system, the $\overline{\text{MEMDECODE}}$ signal selects both the low and high banks. For an 8-bit system, the low bank is selected by generating an even address ($A_0 = 0$) in conjunction with the $\overline{\text{MEMDECODE}}$ signal. Since $\overline{\text{SBHE}}$ is absent (in an 8-bit system), the high bank is selected by an odd address ($A_0 = 1$) in conjunction with the $\overline{\text{MEMDECODE}}$ signal.

Master/Slave Configuration

This feature allows the system to accommodate more than one board. The board reset signal, $\overline{\text{BRDRST}}$, of Figure 23 is used to enable the board. The comparator (Figure 16, U21) that generates the MEMORY DECODE ENABLE must be reconfigured:

1. AEN is connected to P_7;
2. $\overline{\text{BRDRST}}$ is connected to the chip enable, $\overline{\text{G}}$.

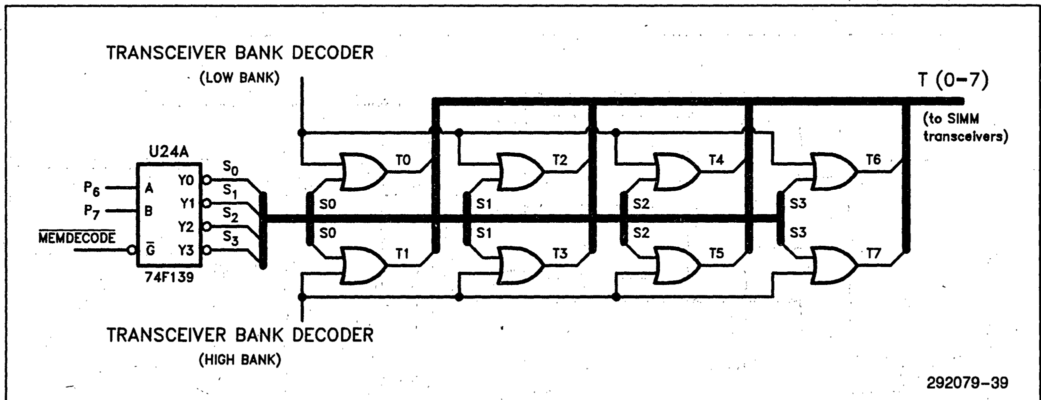


Figure 22. Transceiver Selection at the SIMM Interface

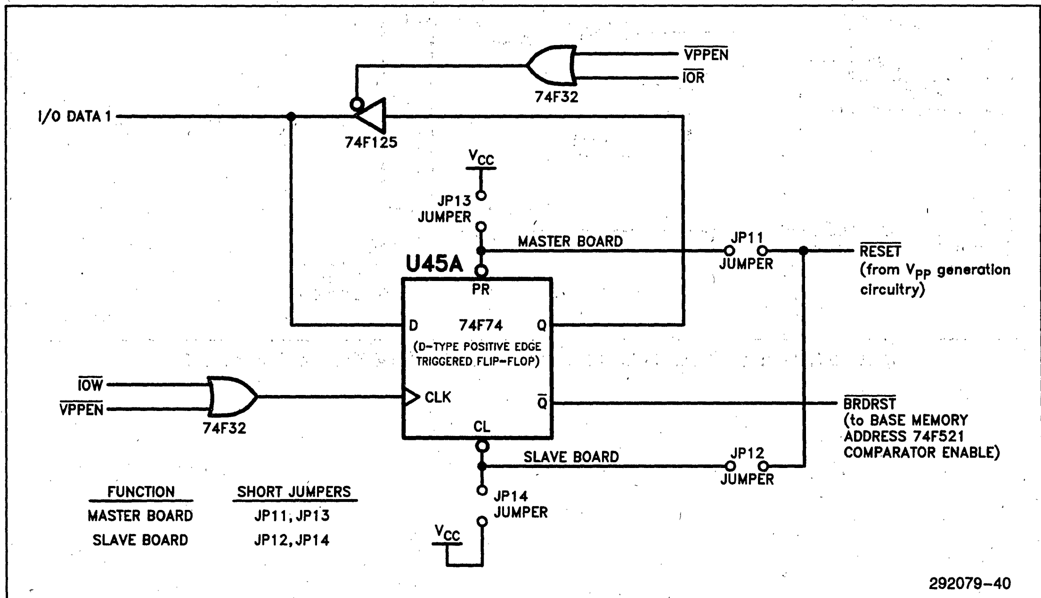


Figure 23. Master/Slave Configuration for Multiple Boards

The jumper settings determine if the board is "active" on system reset ($\overline{BRDRST}$ will be low). The Master/Slave port is shared with V_{PP} enble; therefore to change the "active" status of the board, write to the $\overline{VPPEN}$ I/O port. Software should first read this port to determine the status of "V_{PP} Enable", then use the appropriate mask technique to activate or deactivate the board.

Board Identifier

The board identifier, occupying 4 additional I/O ports, is used for two functions:

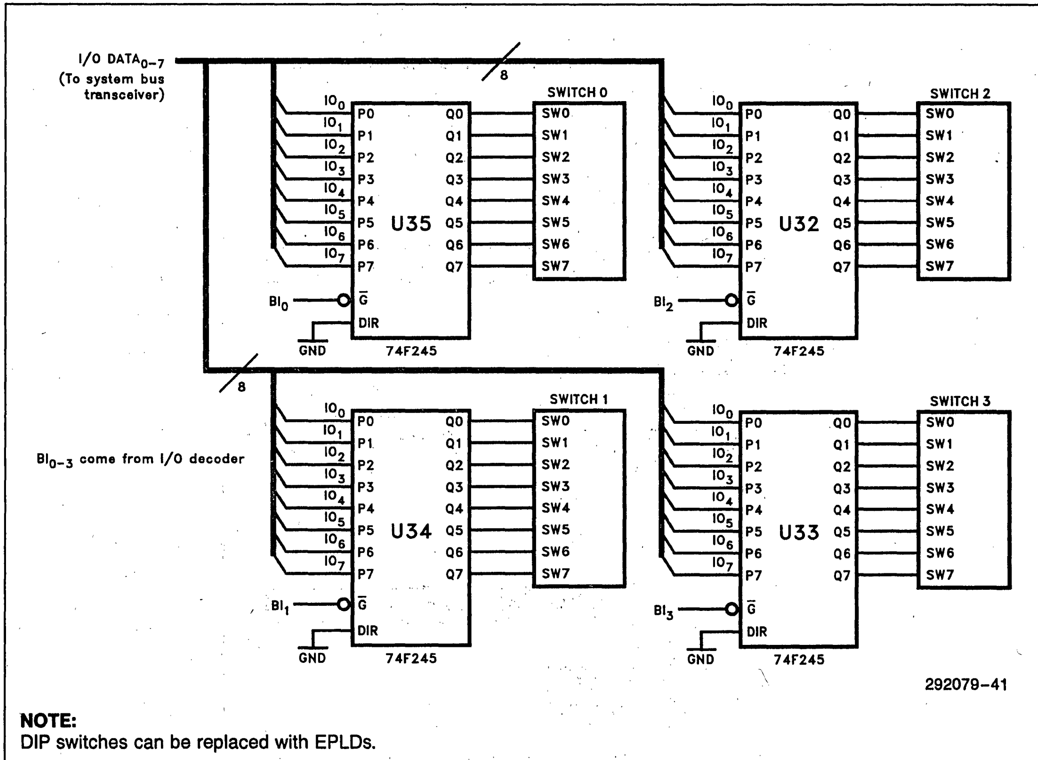
1. To locate the board within the system I/O space, and
2. To identify the board version to assure the software matches the hardware.

The hardware consists of 4 DIP switches and associated 74F245 transceivers (Figure 24, U33–U36). Each switch is read by selection of its I/O address (Figure 15, use $\overline{BI_0}-\overline{BI_3}$). The DIP switches can be replaced by EPLDs that permanently "hardwire" the settings. In this case, the identifier is changed by reprogramming the EPLDs.

Zero-Wait-State Selection

The zero-WAIT selection feature is only applicable in a PC AT system. Driving a low input to the $\overline{0WS}$ pin of the PC I/O channel within 21.5 ns of $\overline{MEMR}$ or $\overline{MEMW}$ going low keeps the system from inserting the standard WAIT-states into the I/O channel bus cycle. On the page memory board, the 0WS signal is generated by the Boolean equation:

$$(\overline{SMEMW} \cdot \overline{SMEMR}) + \overline{MEMDECODE} = 0WS.$$



NOTE:
DIP switches can be replaced with EPLDs.

Figure 24. Hardware Used to Locate and Identify Page Memory Board

Initializing Software for the Paged Memory Board

(The assembly language software is included in the Appendix.)

In the following sections, algorithms will be shown that verify the page-memory board's functionality. To access this board, first find the location of the base I/O address. From Figure 15, the board's I/O ports are accessed as offsets of the I/O base address:

Board Identifier *n*	@ Base Address + *n* (n = 0, 1, 2, 3)
Window Base Address	@ Base Address + 4
Presence Detect Pins	@ Base Address + 5
Master/Slave and $V_{\overline{PPEN}}$	@ Base Address + 6
Page Number	@ Base Address + 7

Next, the Window Base Address I/O port is used to locate the "page" in DOS's memory space. It is then necessary to determine the density of the SIMMs and the total memory available.

Locating the Base I/O Address

Use the board identifiers to locate the base I/O address. The software reads I/O locations until the correct byte sequence is found (Figure 25). Some discretion should be made when choosing the board's I/O address. (See table of I/O port usage in Appendix.) The PC XT and PC AT specification allocates 32 I/O ports at 0300h to 031Fh for prototype cards. We will use this address range for this example. Because the I/O ports for the paged-memory board must begin on an 8-byte boundary, the only possible base addresses are 300h, 308h, 310h, 318h.

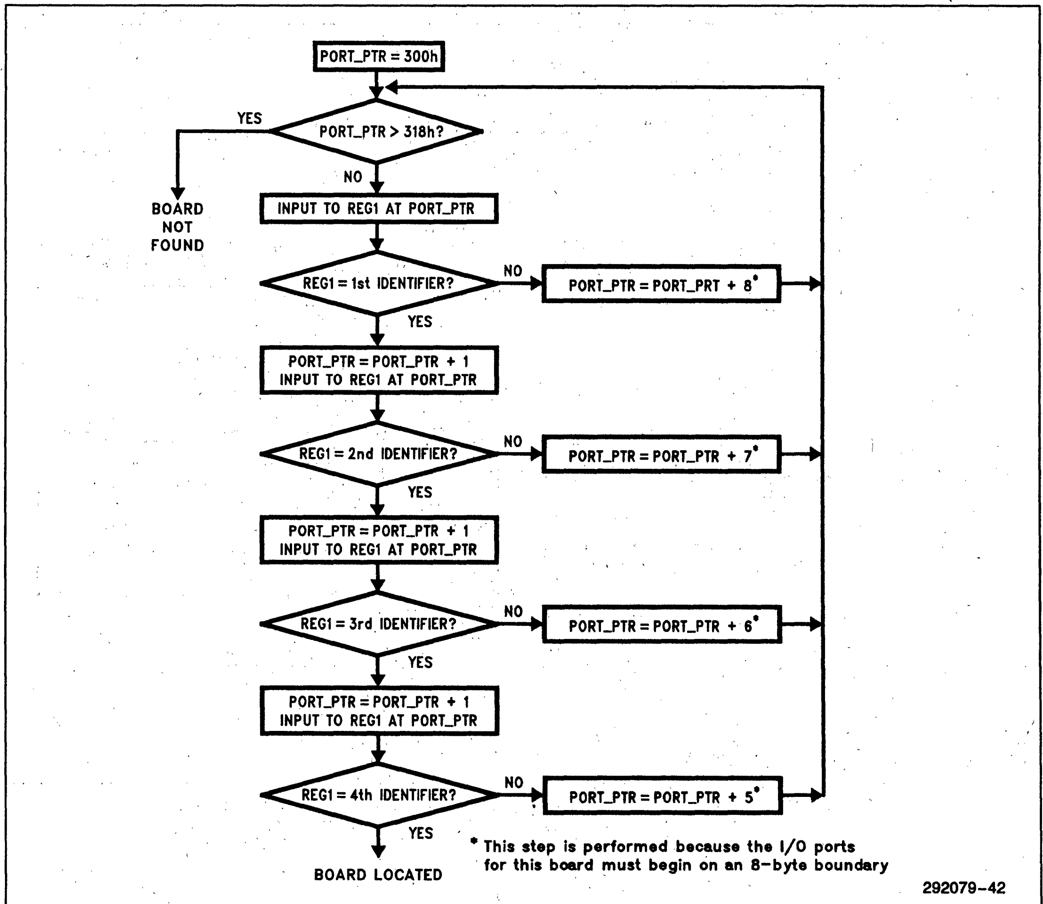


Figure 25. Locating the Page-Memory Board

Locating the Base Memory Address

The base memory address gives the location of the page within the system's memory space. The address switch settings for A_{16}–A_{19} are read from the correct I/O port, Base Address + 4 (Figure 15). After reading these address lines they are stored in the ES segment register used as a pointer to access that memory segment. A_{16}–A_{19} must be shifted into the upper nibble of the ES register to allow proper address generation.

Determining Memory Capacity

First ensure the board is set to read from Page 0. The PD pins are read and translated, using a lookup table of SIMM densities, to a functional value. Then the device identifiers should be read to determine:

1. The number of components on each SIMM;
2. the number of SIMMs installed on the board;
3. and which sockets are used.

SHIFT_REG = 1
SOCKETS = 0
PAGE # = 0
READ PRESENCE DETECT PINS
MASK OFF UNUSED BITS
PUT PD6 IN BIT POSITION 2

Initialize variables.

Only interested in SIMM density info.
Combine density info bits within byte.

REG VALUE = 2, 3 OR 4? — NO: NOT A VALID SIMM; YES: continue

The value will be 2.3, or 4 verifying the presence of a 1, 2, or 4 Mbyte SIMM.

REG IS OFFSET INTO LOOKUP TABLE
READ TABLE FOR SIMM DENSITY

Density is stored as multiples of 1 Kbyte.

READ FLASH DEVICE ID

Determine individual component density.

ID = OBDH OR OB4H? — NO / YES

1 or 2 Mbit Intel flash memory devices on SIMM.

SOCKETS = SOCKETS or'd WITH SHIFT_REG
DIVIDE SIMM DENSITY BY DEVICE DENSITY

SIMM socket is full
"Sockets" used as a bit map.
As an example, if SOCKETS = 0Fh, all 4 sockets are full.

Derives component count on SIMM.

SHIFT_REG = SHIFT_REG * 2
CHANGE PAGE #.

Move binary 1 to next bit address.

To access next SIMM location.

SHIFT_REG = 8? — NO: return to READ FLASH DEVICE ID

Check all sockets.

ALL SOCKETS ACCOUNTED FOR

292079–43

Figure 26. Determining SIMM and Component Densities and Locations

Linear Addressing

Linear addressing directly maps the flash memory array into the system's memory space. "Instantaneous Access" of the entire array is the obvious advantage over paging. Additionally, the decode circuitry is simplified. Figure 27 shows an example for accessing 16 Intel Flash Memory 28F020s arranged in a 4 Mbyte linear array.

The number of address lines used, as well as the decoder type (2 to 4, 3 to 8, etc.), is determined by the flash memory device size. The address lines A_1-A_{18} are used for byte selection within each device (256 Kbytes * 8).

The decodes for the individual devices can be designed in a row-column method similar to that used for the page memory board. An alternative design uses an individual chip enable for each of the 16 devices.

The enable for the 74HC138 (3 to 8 decoder) is governed by a 74F521 comparator. System address inputs to the comparator are chosen to locate this array on a 4 Mbyte boundary. (The array base address could be located on a non-4 Mbyte boundary but this would add to the decoding complexity.) With the inputs chosen in this example ($A_{22}-A_{23}$), the array base address will be between address 0 and 12 Mbytes to confine this memory array within the PC AT defined address space of 16 Mbytes. $A_{19}-A_{21}$ are inputs to the decoder which generates one of the eight chip enables ($\overline{CE}$). (Use a 74F245 transceiver for the data bus of every 8 flash memory devices. The address lines also need buffering when connected to a PC bus.)

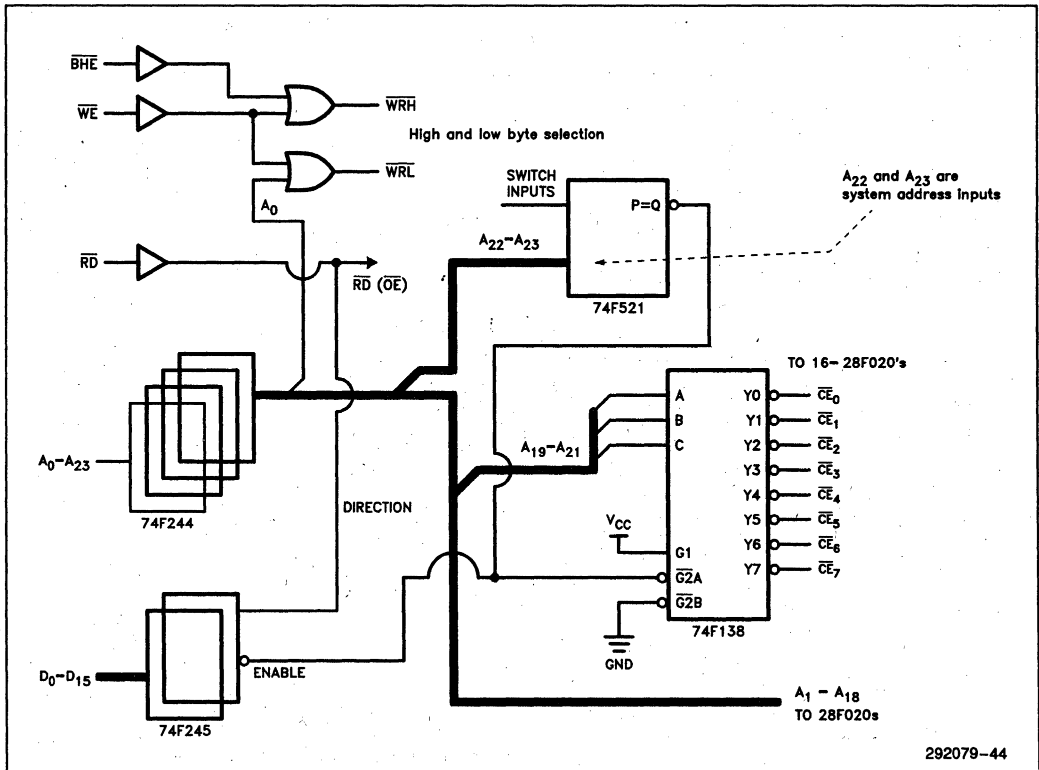


Figure 27. Linear Addressing Hardware Block Diagram

I/O Addressing

From the standpoint of the system's address space usage, I/O addressing provides a conservative solution. As an example, four gigabytes of a flash memory array can be addressed through only two I/O ports. An I/O write sends the flash memory addresses out on the data bus. This "data" is latched (using '574s) and made available to the flash memory devices and decoding circuitry (Figure 28). A third I/O port, used as an enable for the flash memory device decoder and transceivers, helps conserve power when the array is not being accessed.

Relative to linear addressing, I/O addressing generally has limited access speed capability because of the I/O "bottleneck". Read speed can be increased to match linear addressing by replacing the '574 latches with '191 counters.

In the following circuit example, decoding for I/O is accomplshed with a 74F138, 3 to 8 decoder (Figure 29, U1). The base address for these I/O ports is on an 8-byte boundary. When any one of the 8 I/O addresses is selected, the comparator (U2) generates the enable signal (if AEN is low) for the decoder.

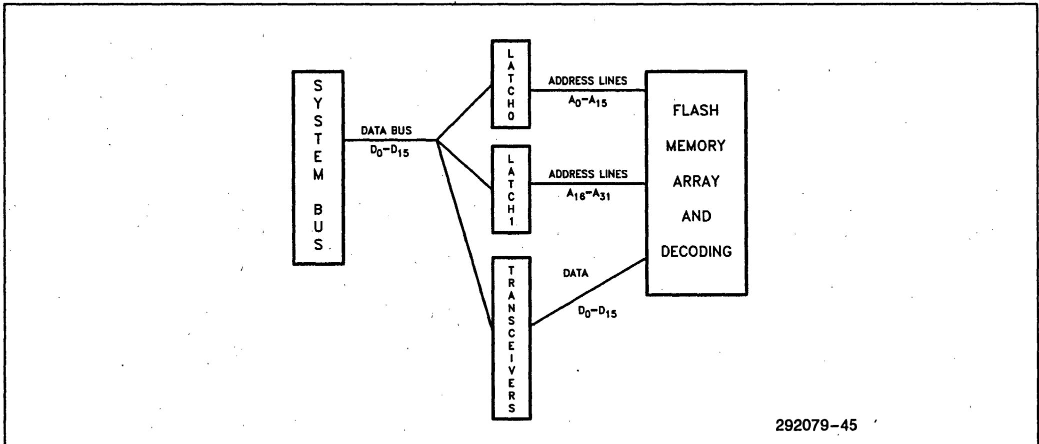


Figure 28. Data Bus Generates Flash Memory Addresses

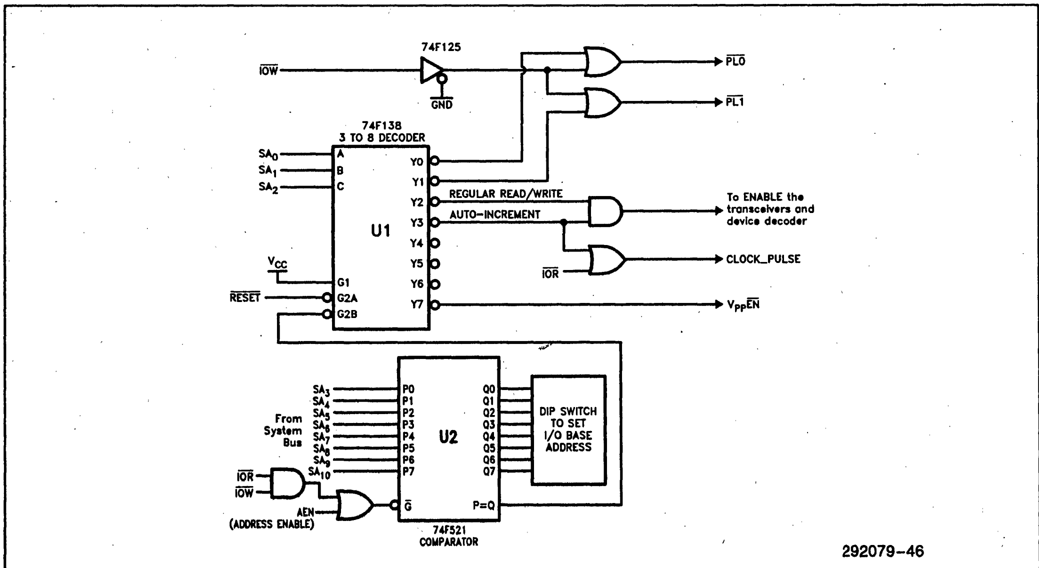
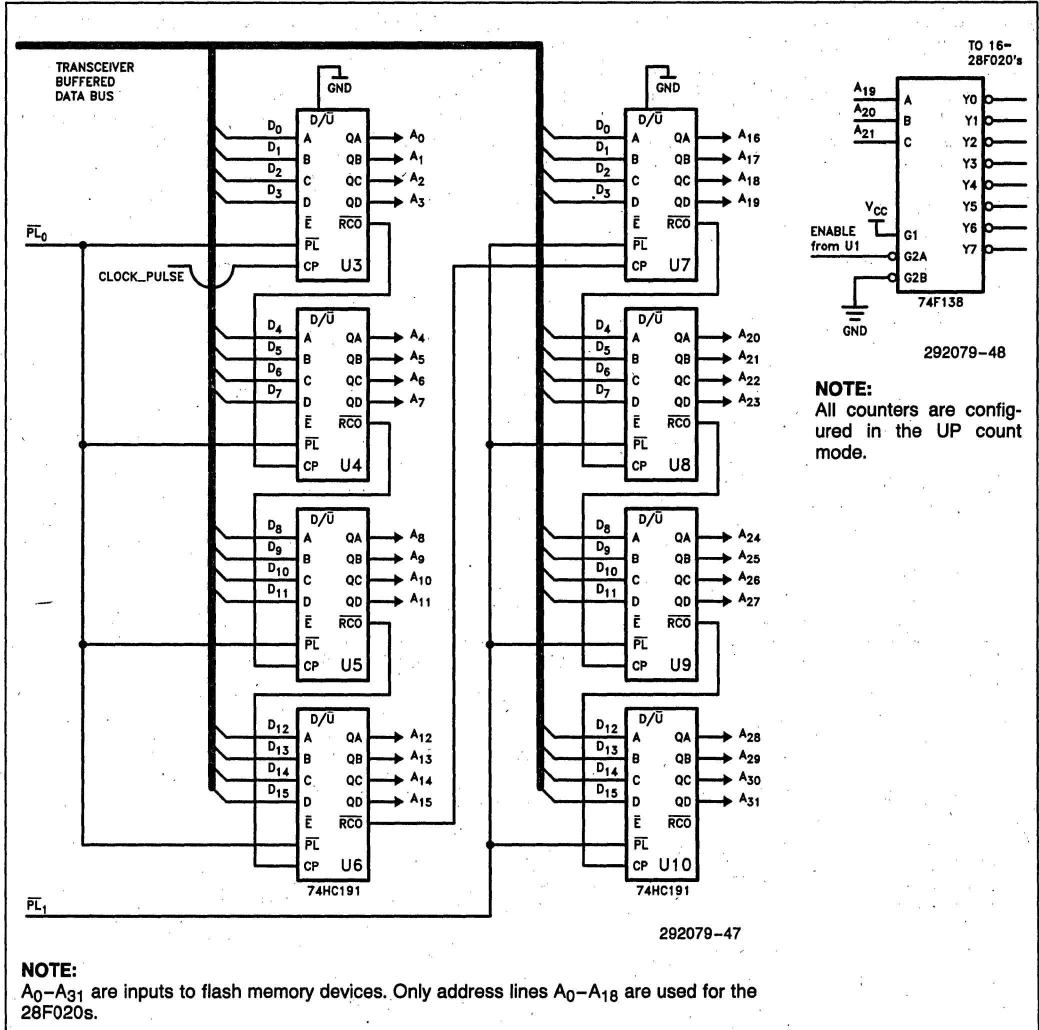


Figure 29. I/O Decode and Enable Circuitry

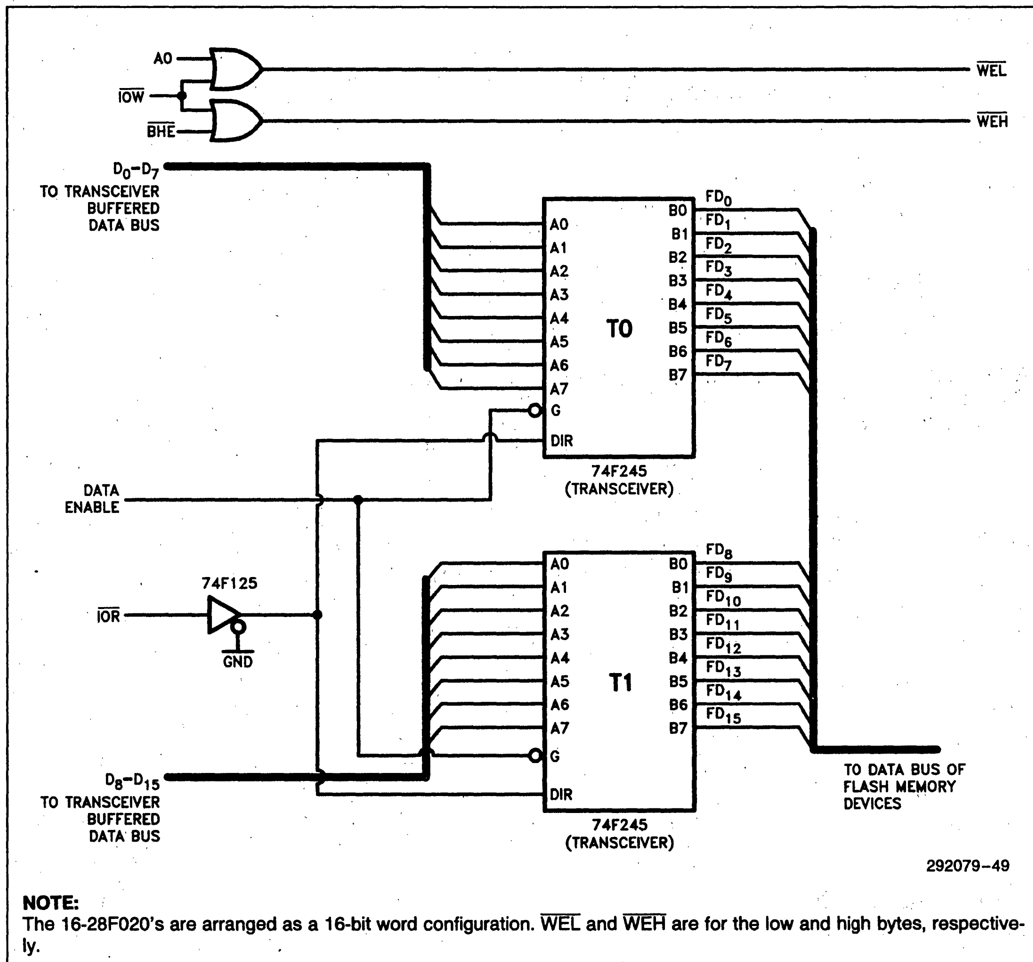
An I/O write to the first and second ports generates parallel load signals, $\overline{PL_0}$ and PL_1. These signals latch the "data" (addresses) into the 4-bit counters (Figure 30, U3–U10). This latched data represents the address for the flash memory devices.

A read or write from the selected flash memory address is performed when the third I/O port is accessed (Figure 29, U1); this generates an enable for the flash memory device decoder and associated transceivers (Figure 31, T_0 and T_1).



NOTE:
A_0–A_{31} are inputs to flash memory devices. Only address lines A_0–A_{18} are used for the 28F020s.

Figure 30. Counter Circuitry



NOTE:
The 16-28F020's are arranged as a 16-bit word configuration. WEL and WEH are for the low and high bytes, respectively.

Figure 31. Transceiver Enable Circuitry

The fourth I/O port activates the circuitry that obtains very high performance from an I/O board. A read from the fourth I/O port address generates the clock signal for the 74HC191s, CLOCK_PULSE. The counter increments on the rising edge of the clock (read signal), selecting the next flash memory address. This rising edge occurs at the end of the I/O read cycle and the data has already been read. This method is analogous to address pipelining. It is perfect for a "string" read because continuous reads from the fourth I/O port automatically increments the address to access the next word of data stored in the flash memory array.

Capacitive Loading

Capacitive loading is an important consideration for a solid-state mass storage device. If proper buffering techniques are not followed, performance degradation will occur.

The specifications for Intel's Flash Memory devices are based on a test capacitive load of 100 pF. Each data line contributes 12 pF, therefore 8 devices connected to one data transceiver will not experience speed derating (12 pF * 8 = 96 pF). Additional flash memory devices

on that transceiver will increase the loading seen by any one device.

Degradation is calculated as follows (Q = Amount of Charge, T = Time, C = Capacitance, V = Voltage, and I = Current):

COULOMBS LAW STATES:

$$Q = I\Delta T$$

AND GIVEN THE RELATION:

$$V = \Delta Q/C \rightarrow I = C\,\Delta V/\Delta T$$

FROM THIS RELATION, THE CHANGE IN ACCESS TIME CAN BE EXPRESSED IN TERMS OF CAPACITIVE LOAD:

$$\Delta T = C\,\Delta V/I$$

For example, using four SIMMs, each with 8 components in a 16-bit configuration (4 components on high byte and 4 components on low byte), each Intel Flash Memory device sees a load of 15 devices (12 pF * 15 = 180 pF). This loading is 80 pF in excess of the device specification so therefore:

$$\text{Time Change} = \text{Additional Capacitance} \times \frac{(V_{CC} - V_{OL})}{I_{OL}}$$

$$= 80\text{ pF} \times \frac{(5.0 - 0.4)\text{V}}{5.8\text{ mA}} = 64\text{ ns}$$

(Reflecting worst case conditions.)

SOFTWARE DESIGN IMPLEMENTATIONS

Each hardware implementation discussed above can be used in several types of mass storage applications. The general categories include: data recoders, Write-Once-Read-Many (WORM) drives for storing application programs and fixed data, and magnetic disk emulators.

Data Recording

The applications for data recording represent an endless list. Examples include digital imaging, digital photography, point-of-sale terminals, patient monitors, and flight recorders. These systems will use Intel Flash Memory as a more economical and reliable replacement for SRAM + battery. Alternatively, mechanical disks will also be replaced by Intel's Flash Memory when higher reliability, lower power consumption, higher performance, and lighter weight are required.

Interleaving

Although the basic concept of data recording is similar from system to system, variations in implementation exist. For instance, some applications require high-speed data acquisition. Data programming rates are improved considerably by employing interleaving techniques. The majority of time spent programming or erasing a flash memory device results from the delay times in the software algorithms. (It is advised to review the standard algorithms first. See any Intel Flash Memory data sheet for Quick-Pulse Programming™ algorithm.) Interleaving takes advantage of these delay times to begin programming consecutive devices.

There are hardware and software mechanisms for interleaving. The flash memory array for hardware interleaving requires special decoding techniques (Figure 32). Contrary to linear decoding, the system address lines A_0–A_3 are decoded to provide the chip select signals and individual bytes are selected with the address lines A_4–A_{20}. (For the Intel 28F010.) This decoding technique allows software to automatically access sequential devices by writing or reading sequential memory addresses. (Data accumulated with program interleaving will not be stored consecutively within a single device.)

The interleaving algorithm to program the 2 Mbyte flash memory array is shown in Figure 34 and 35. The basic goal is to utilize the delay times. To simplfy the algorithm for this discussion, the data will be programmed on a byte-wide basis. Word-wide and double word-wide techniques, discussed later, will further increase programming speeds.

During multi-component programming, the number of pulses required could vary between different devices. Code is reduced if the programming loop does not have to selectively "decide" if a byte has programmed correctly (verified). However, continual programming of a programmed byte is not necessary and should be avoided. This is done by masking the command sent to that particular device. The RAM table in Figure 33 is used as a data and flash memory command buffer. After a programmed byte has verified, its associated data and commands in the RAM table are written with the value 0FFH (RESET command for Intel flash memory). The data is also written as an 0FFH since this is null program data.

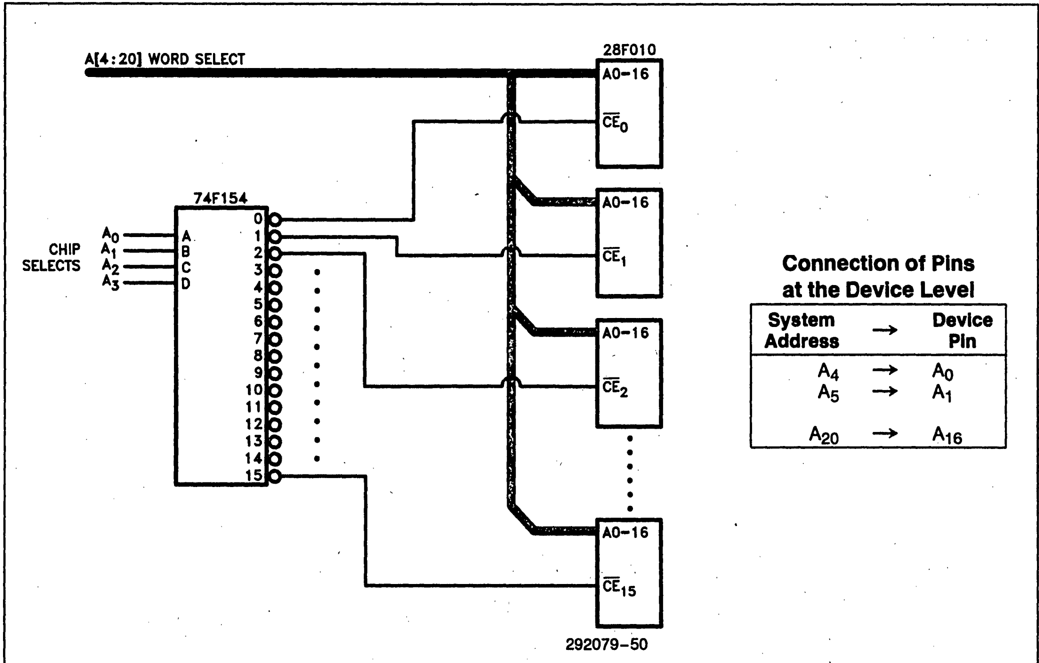
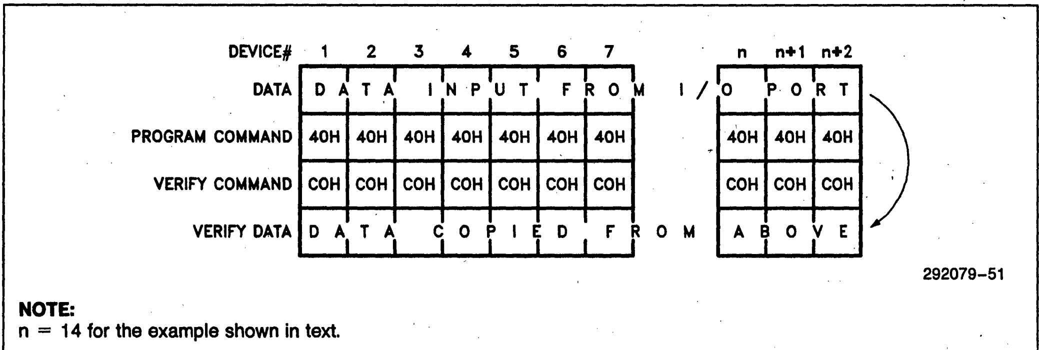


Figure 32. Hardware Interleaving Block Diagram



NOTE:
n = 14 for the example shown in text.

Figure 33. RAM Array Used as Data Buffer and Command Mask Storage

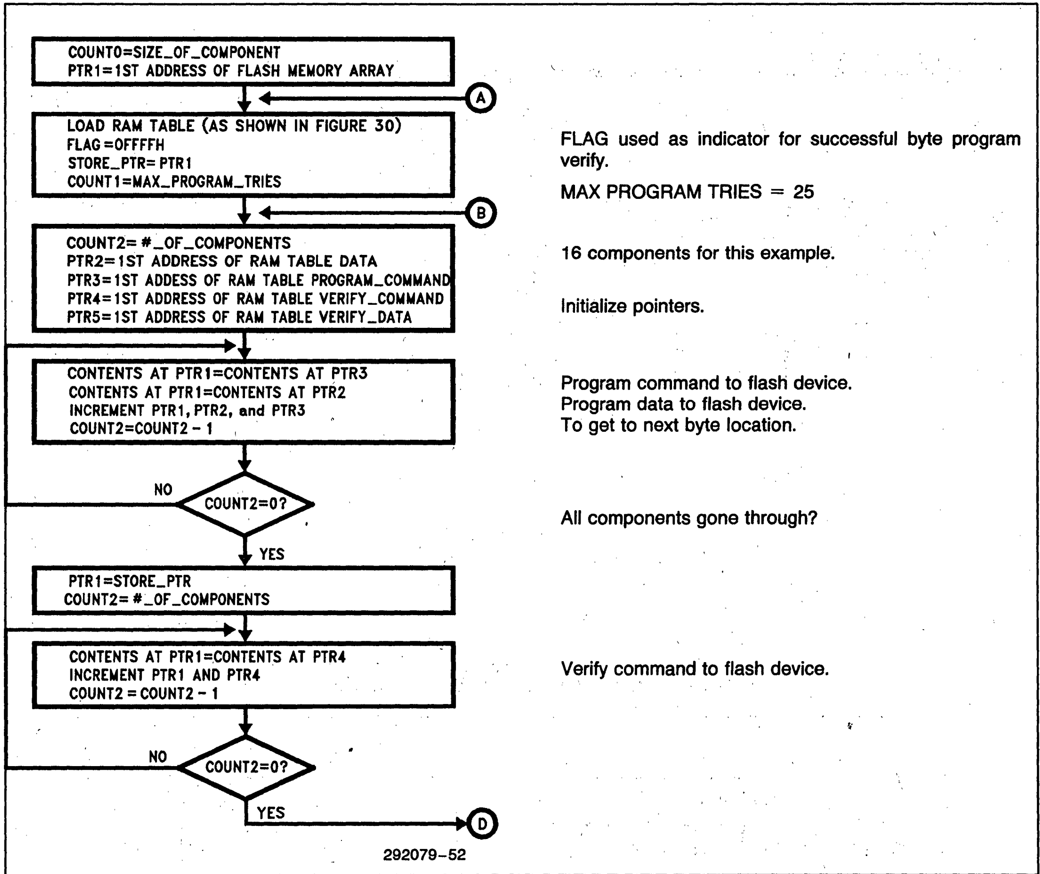


Figure 34. Program Interleaving Algorithm

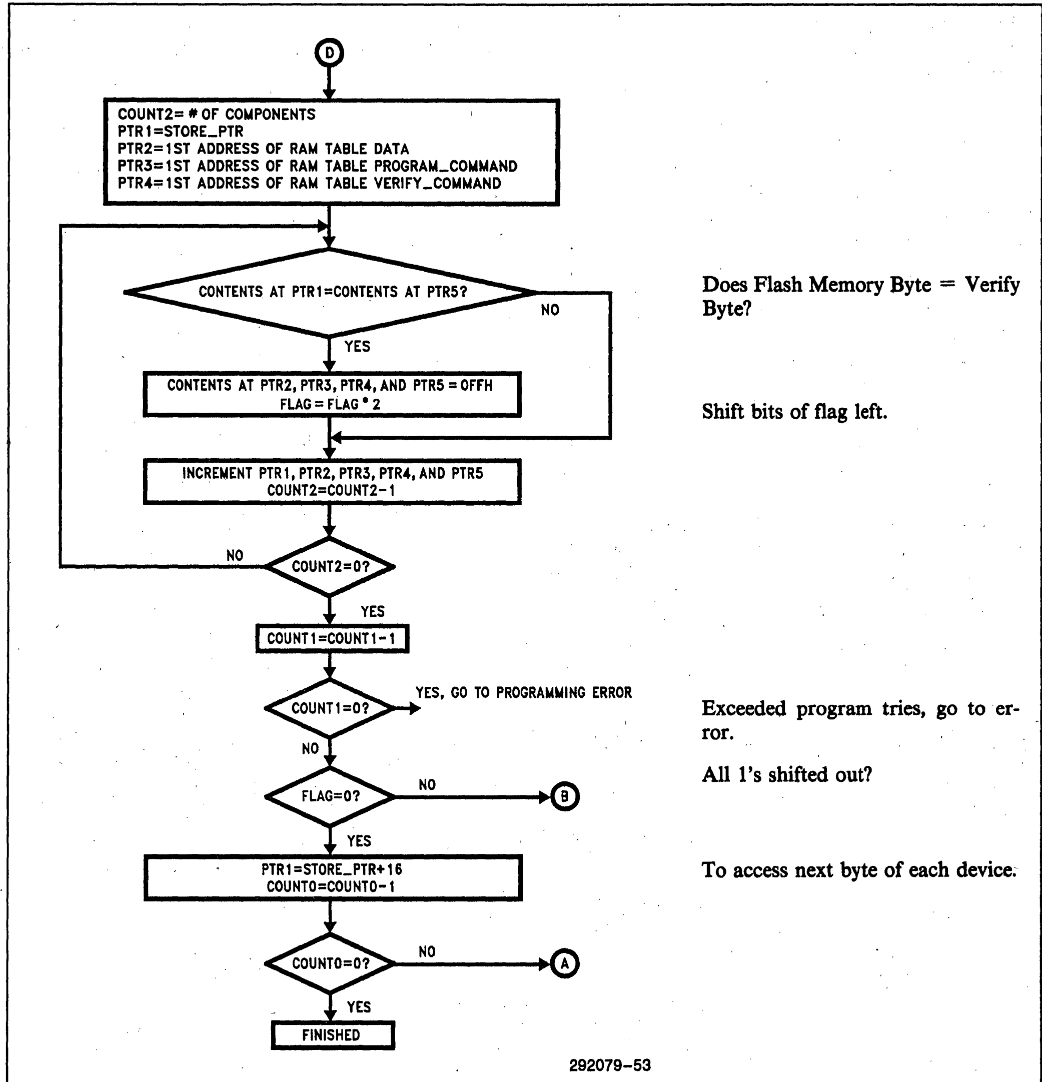


Figure 35. Program Interleaving Algorithm (Continued)

Software and hardware interleaving are very similar. Software interleaving is performed using conventional decoding and addressing methods. Instead of incrementing flash memory addresses by one to access the next byte (as with hardware decoding), the address is incremented by the size of the component. While allowing the use of "general-purpose" (non-interleaved) hardware, software interleaving requires reading back the data in the same, non-sequential fashion as was used for recording.

Interleaved erase is useful for erasing an array of flash memory devices. This approach greatly reduces the total subsystem format time. As specified in the erase algorithm, each erase pulse requires a 10 ms delay. (See Quick-Erase™ algorithm in Intel Flash Memory data sheet.) Without interleaving, the processor is idle during this delay time. As with program interleaving, this time is used to begin the erasure of consecutive devices, thereby reducing the overall erase time.

Further program and erase time can be saved by supplementing the byte-wide algorithm with 16- or 32-bit interleaving. Extra data and commands are added to the RAM Mask Table. The major difference in the algorithms involves the verify operation. Depending on the bus width, 2 or 4 bytes are verified simultaneously as shown in Figure 36 (for a 16-bit algorithm).

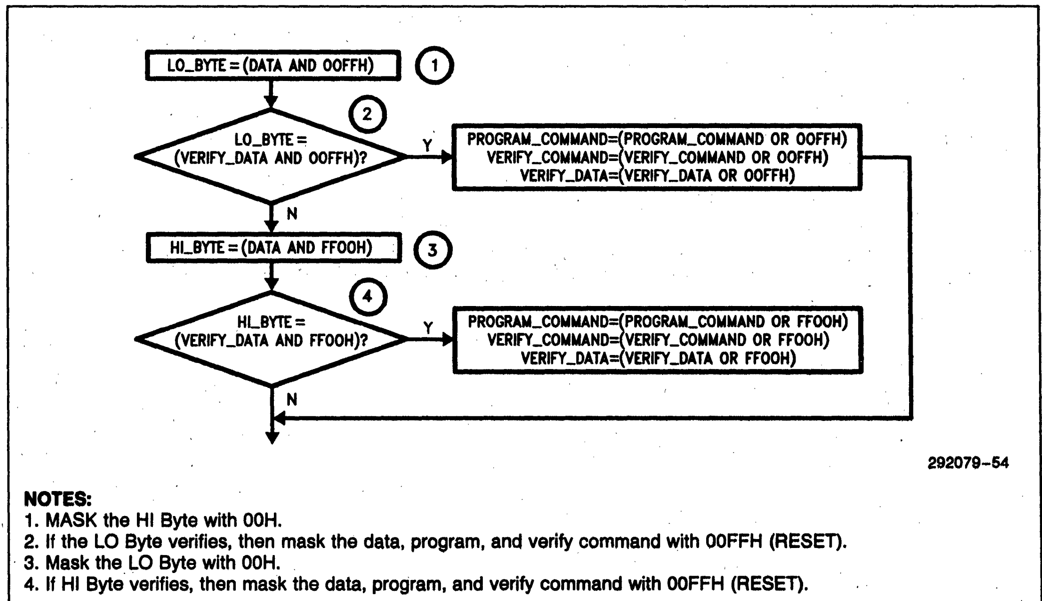
Power Requirements for Interleaving

Current consumption is an important consideration for interleaving. During programming, each device typically consumes 9 mA (1 mA I_{CC} and 8 mA I_{PP}) while programming or erasing; this translates to about 100 mW. If interleaving with 16 devices, about 144 mA (16 devices * 9 mA) or 1.6W, is drawn. Battery powered systems will have a practical limit on the number of components in the interleaving loop. Failure to accommodate these current levels, resulting in V_{PP} voltage drop, will compromise programming and erase reliability.

Write-Once-Read-Many (WORM) Drives

The optical disk is an example of a typical WORM drive application. Its strengths are extremely high densities and low cost per bit. However, it is an unacceptable solution for a low powered, lightweight laptop computer system. It is this environment that solid-state drives offer the greatest benefit. Solid-state ROMs have historically been used in laptop systems to store software programs that seldom change. When the software does change, the ROM "application hardfile" is discarded and a new one is programmed.

Unlike the ROM drive, Intel Flash Memories can be reused and reprogrammed in a true WORM fashion. A computer user can load favorite software programs on the flash memory drive. Adding revised programs to the drive is accomplished by writing to the next free space or by erasing and reprogramming the entire drive. Software drivers can be written to implement this functionality in most operating systems.



NOTES:
1. MASK the HI Byte with 00H.
2. If the LO Byte verifies, then mask the data, program, and verify command with 00FFH (RESET).
3. Mask the LO Byte with 00H.
4. If HI Byte verifies, then mask the data, program, and verify command with 00FFH (RESET).

Figure 36. 16-Bit Masking for Verify Operation

Disk Emulation

Microsoft has a flash memory file system for DOS. It stores and retrieves data or application programs in a manner that, to the end user, appears similar to a disk drive. New files are written sequentially from beginning of memory. However, when the disk is full, it reclaims memory space for storing additional files.

When an application accesses a disk through INT 21H, the MS-DOS* kernel checks the drive letter (Figure 37). If the drive has been declared as a flash memory disk, a built-in redirector services the call. (This is very similar for networked drive accesses.) Otherwise, if the drive letter is that of a floppy or hard disk, the call is handled by the standard DOS file system. The File System provides the link between DOS and the Flash Memory and Hardware device driver. It changes DOS file system commands into a form understood by this unique file structure.

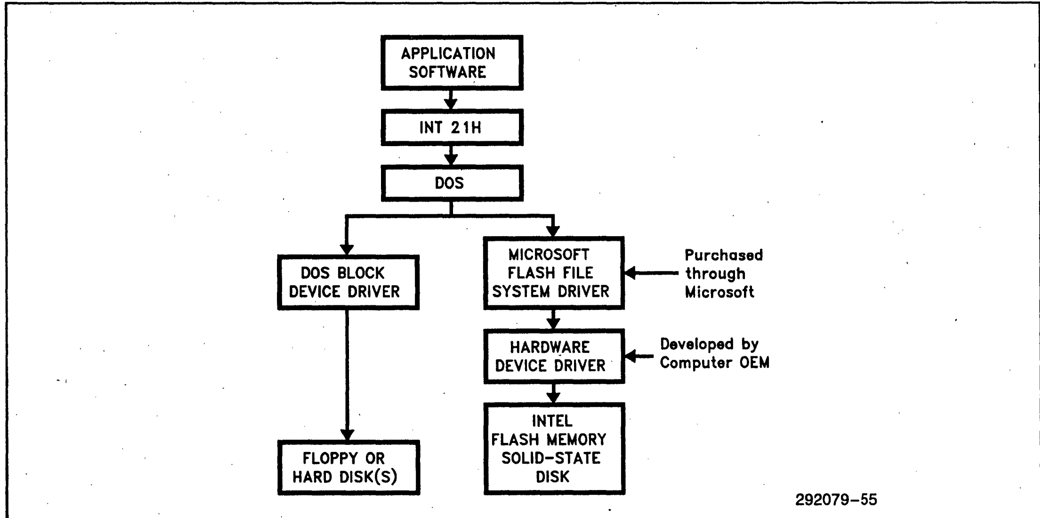


Figure 37. Disk Interface Levels

*MS-DOS® and Microsoft® are registered trademarks of Microsoft Corporation.

The Flash File System Driver is the "intelligence" of this file system. It searches for:

1. A Boot Record that identifies the file system and version, and locates the start of the data area;
2. The Root Directory Entry Record and many Directory and File Entry Records.

The file system driver is independent of the hardware interface to the flash memory disk. The hardware device driver, developed by the OEM or BIOS software vendor, interfaces the flash memory disk to the flash file system. It is responsible for the low level calls to the Intel flash memory devices. The actual implementation of the interface is dependent on the hardware configuration of the disk (I/O, paged, and linear addressing are examples).

To minimize fragmentation losses and allow arbitrary extension of files, the flash memory file system uses variable sized blocks rather than the standard sector/cluster method of more traditional file systems. The fundamental structure employed to offer this flexibility is based on linked list concepts; files are chained together using address pointers located within directory entries for each file.

Files and directories are written to the flash memory disk using sequentially free memory locations—a stack-like operation (Figure 38). Furthermore, file sizes can be variable, abandoning the traditional sector/cluster approach of DOS. When "the stack" is full, (containing deleted files), the intelligent software algorithm performs a cleanup operation to reclaim the "dirty" space.

File and subdirectory information is attached to the beginning of each file, unlike the standard DOS approach of directory and FAT placement. As directory and file entries are added, they are located by building a linked-list. Besides containing the customary fields (e.g., name, extension, time, date of creation, etc.), a directory and file entry contains a status byte and various pointers used for the linked-list process. The status byte, besides indicating whether a file/subdirectory exists or is deleted, is also used to signify valid sibling and/or child pointers and to determine if a directory entry pertains to a file or a directory.

When a directory or file is requested or added, the flash memory disk is searched beginning at the head of the linked-list. The chain is followed from pointer to pointer until the correct entry is found. If the search arrives at the chain's end (an FNULL is encountered), the system responds analogously to DOS with a "File not found" message.

This linked-list chain consists of two basic types of pointers: sibling and child. Sibling pointers are used to locate directories or files at the same hierarchial level. Child pointers are used to locate subdirectories or the first file of a particular directory. The following examples elaborate these concepts.

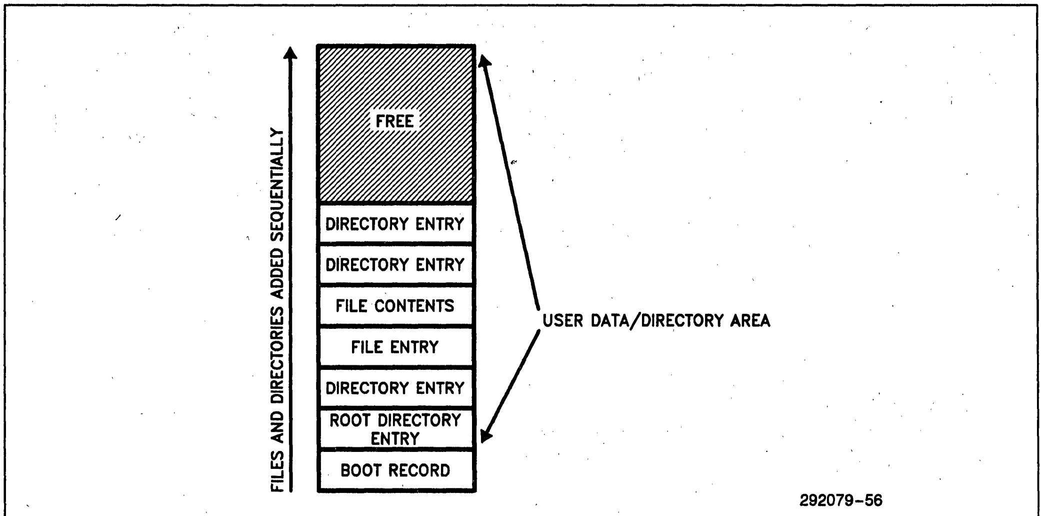


Figure 38. FFS Storage

In Figure 39, Directories B and C are subdirectories of Directory A. Specifically, Directory C is a sibling of Directory B and both are children of Directory A. FNULL indicates the end of the chain.

Figure 40 shows two files (File A and File B) added to a directory (Directory A). File A and File B are at the same level, therefore they are siblings. A file's file entry contains an extent location pointer that indicates the start of its data area.

When a file appears multiple times (because of deleted versions) on the flash memory disk, the file system must find the most recent version. The status byte contains bit fields that indicate whether that particular file is a valid or deleted file. The directory information of a deleted file is used for pointers of the linked list and the search would proceed until the most recent version is found.

A key point to be made for using this method of file storage is that the user is in control of the rate in which the disk becomes full; using the flash memory disk predominantly for application code storage and non-temporary data files reduces the frequency of "cleanup". However, flash memory will typically perform 100,000 cycles and eliminates reliability concerns when used as a hard or floppy disk replacement.

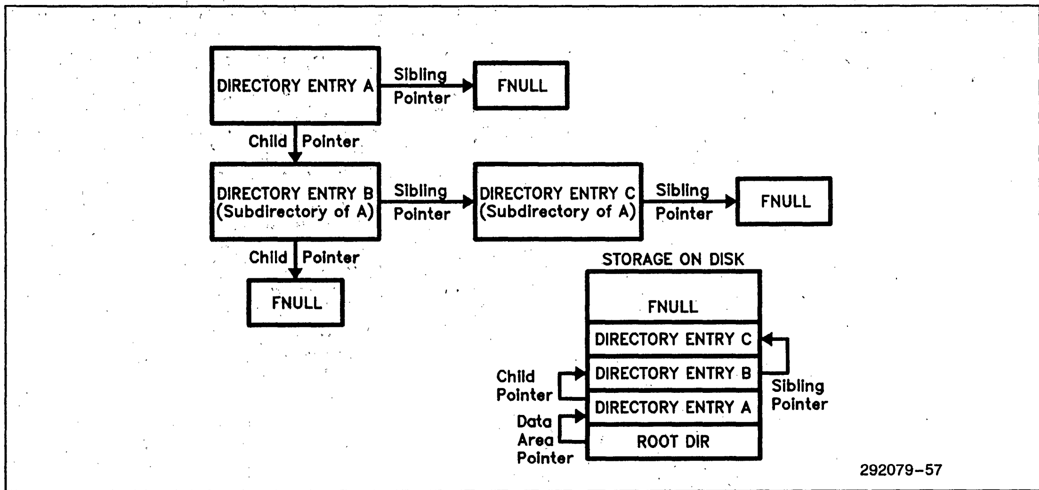


Figure 39. Directory Arrangement

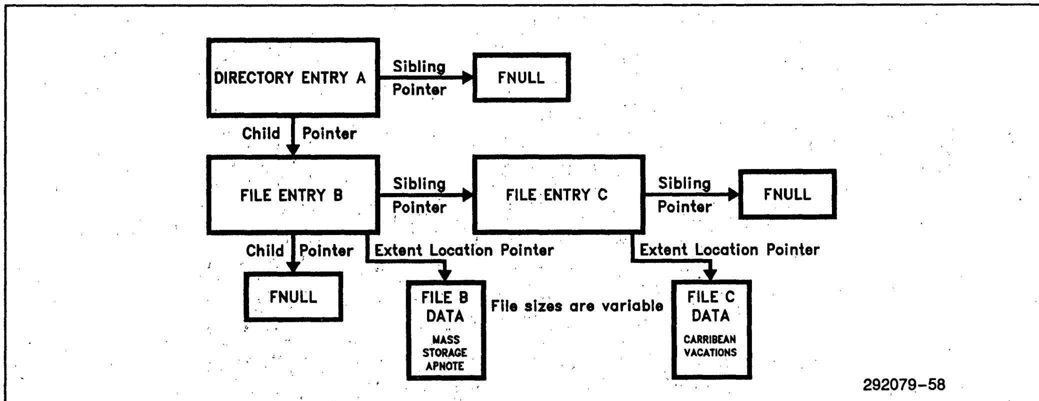


Figure 40. File Arrangement

Creating a Bootable Drive

The startup time of the PC can be decreased by booting from a flash memory disk instead of the magnetic disk. To do this, a "disk-image" is installed on the flash memory disk which is located in the system memory space between C0000H and E0000H (Figure 10). (The "disk image" contains the Boot Record, Directory, and FAT.) This memory space is referred to as Expansion ROM. During the system Power-On-Self-Test (POST), the system searches this memory area for the ROM adapter signature, 055AAh, marking the beginning of the disk image. Once this signature is found, the BOOTSTRAP process begins. The software to create and install this "disk image" is available as a product from Microsoft Corporation as ROM executable MS-DOS.

WHY FLASH?

CHARACTERISTICS OF INTEL FLASH MEMORY

Power consumption, weight, performance, and reliability are the key criteria for a system design. The discussion of Intel flash memory as a mass storage medium would not be complete without a performance analysis and comparison to other technologies.

Power Consumption

Portability of a computer demands battery longevity and consequently minimal power consumption. Small form factor disk drives are being designed specifically for the size and power requirements of laptops.

A drive has three basic operating modes: active, power savings, and standby. The active mode consists of reading, writing, and ready. Ready condition allows "instantaneous" transitions into the read or write states. In the power-savings mode only the drive motor continues to run. Standby shuts off all functionality except for the circuitry needed to "spin-up" the drive. From the standby mode, extra power and considerable time, is required to "spin-up" the disk.

Power Consumption Comparison (Watts)

(Based on typical performance characteristics. The 20 Mbyte Flash Memory disk is based on the use of 80-28F020s. Only two of the forty devices are accessed at a time, the remainder are in standby mode.)

Active Modes	Hard Disk Drive (2.5", 20 Mbytes)	INTEL Flash Memory (20 Mbytes)
Ready	1.7–2.0	0.05 (Same as Standby)
Read	3.5-4.0	0.15
Write	3.5–4.0	0.25
Power Savings	1.5	0.05 (Same as Standby)
Standby	0.1–0.5	0.05
Spinup (from Standby)	9.3	0

For a battery-powered system, 3–4 hours of operation is unacceptable. Battery longevity is achieved by using Intel Flash Memory solid-state storage as a disk replacement. The following table relates battery life and the different functions of disk operation. A "AA" battery with a capacity of 2215 mAH is used for the comparison. Obviously, for a truly accurate representation, other components of the system should be included. But from the data storage point of view, the flash memory disk will operate many more hours than the hard disk drive on a set of batteries.

Hours of Operation for a "AA" Battery (Based on Data from Previous Table and 2215 mAH Battery Capacity)

	Hard Disk Drive (2.5", 20 Mbytes)	INTEL Flash Memory (20 Mbytes)
Read	0.83	22.15
Write	0.83	13.29
Standby	6.64	66.45

Data Access Time

Reading data from a magnetic disk is a very slow process compared to a solid-state disk (SSD). Disk transfer time is lengthy due to four time components: spin-up, seek time, latency, and data transfer time. Spin-up is a factor to consider for battery-powered systems, where most disk accesses are begun from the standby mode. During the seek time, the arm is repositioned to the correct track. Latency is the delay from arm repositioning until the first sector of the transfer moves under the

read/write head. This is dependent on the speed of rotation. The actual transfer of data is the third component. The standard SCSI interface transfers data between 5 Mbits and 10 Mbits per second, with which flash memory compares very favorable.

For this example it is reasonable to assume a transfer rate of 1.0 Mbytes per second. Using a full word-wide (x16) bus bandwidth (120 ns access speed of the device), flash achieves a transfer rate of 16.6 Mbytes per second.

Read Speed Comparisons

	Hard Disk (Standard SCSI Interface)	Floppy Disk	Flash Memory (16-Bit Bus, 120 ns Access)
Seek Time	28 ms		0
Latency	8.3 ms	100 ms	0
Transfer Rate	1.0 Mbyte/s	62 Kbyte/s	16.6 Mbyte/s
Total for 10 Kbyte File	46.54 ms	261.3 ms	0.62 ms

(Floppy disk drive specifications combine access into one category.)

In this example, the flash memory disk has 75 times the read performance over the hard disk. Smaller files result in even greater differences. Additionally, the 5 second spin-up of the hard disk gives the flash memory disk over 8,000 times the performance!

A byte will typically program in Intel Flash Memory in one pulse. (See Intel Flash Memory Data sheet for programming algorithm.) Based on this and the parameters used in the example above, a 10 Kbyte file is written to the flash memory disk in 87.04 ms. Because writes to a hard disk typically begin from spin-down, the flash memory disk is still over 50 times faster. Since reads are 80% of disk access, flash memory's user-perceptible performance advantage is substantial.

Reliability

The definition of hard disk mean-time-before-failure (MTBF) is extremely ambiguous. There are no industry-wide standards for making a reliable calculation. Disk drive manufacturers choose whichever method best suits their product's reliability perception.

One method uses the overall mean failure. The MTBF of all critical components is computer analyzed and the lowest one is selected. A second method tests 100 drives. The hours of the first ones to fail are multiplied by the number of drives. How many reads or writes are performed? Is the disk stopped and started during the process? Standard answers do not exist.

The vagueness of the test procedures makes it difficult to compare the MTBF for a flash memory solid-state disk and a hard disk. Based on the fact that disk usage is 80% reads and 20% writes, a reasonable comparison can be made. (What is not taken into account is that disks are an 'infinite' write, but finite read medium. Continuous reading causes reduced magnetic field strength, a failure mechanism hidden by re-writing the disk.)

Intel's Flash Memory typically performs 100,000 erase/program cycles. (Failure does not occur at this point. The only noticeable change is a gradual increase in program and erase times.) Assume a flash memory disk size of 4 Mbytes that functions like a WORM drive; it is erased and reused after filling.

Based on a typical disk MTBF of 50,000 hours and the 80/20% division, 10,000 hours are used for writing files. Assume the average file size written to disk is 10 Kbytes. A 4 Mbyte flash disk can store approximately 400 $\times$ 10 Kbyte files (4 Mbyte/10K = 408) before erasure is necessary.

These 400 files could be writen to the disk 40 $\times$ 10^6 times — (400 files $\times$ 100,000 cycles = 40 $\times$ 10^6). The result is that within a 10,000 hour period, one 10 Kbyte file could be written once every 0.9 seconds.

$$\frac{10{,}000 \text{ hours}}{40 \times 10^6 \text{ Files}} \times \frac{3600 \text{ Seconds}}{1 \text{ Hour}} = \frac{0.9 \text{ Seconds}}{\text{File}}$$

It would be more realistic (although still extremely aggressive) to assume that this 10 Kbyte file is written to this disk every 10 minutes. At 100,000 cycles, 40 $\times$ 10^6 files will have been written. The MTBF can be calculated as follows:

$$40 \times 10^6 \text{ Files} \times \frac{10 \text{ Minutes}}{\text{File}} \times \frac{1 \text{ Hour}}{60 \text{ Minutes}} = 6.6 \times 10^6 \text{ Hours}$$

This is an MTBF of over 6 million hours! (See Reliability Report RR60 for more details.)

A flash memory solid-state disk outlasts its mechanical counterpart by at least two orders of magnitude, especially if head parking problems and limited start/stop cycles of the mechanical disk are taken into account.

Weight

Lowering the power consumption of your portable system also lowers the weight. Reduced battery demands mean smaller and lighter batteries and power supplies. Weight savings is also gained by the proper choice of the mass storage medium. The small 20 Mbyte 2.5" disk drives weigh between 9 and 21 ounces. The equivalent capacity of flash memory using 80-2 Mbit TSOPs (which individually weigh 1.16 x 10^{-2} ounces) weighs 0.93 ounces plus the weight of the circuit board. (See section on Intel flash memory packaging.) This difference is critical when the computer weight requirement is under five (5) pounds.

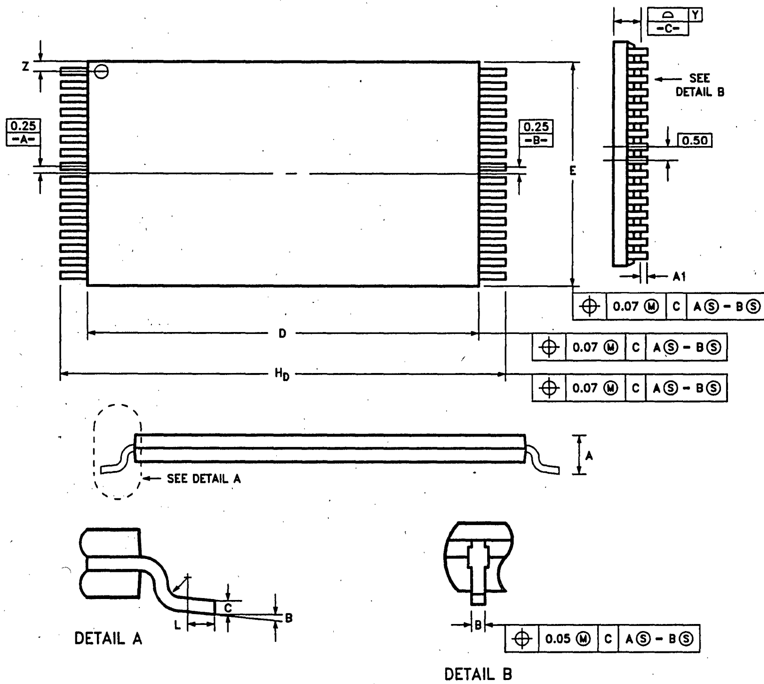
SUMMARY

The advent of Intel Flash Memory has led to the evolution of solid-state mass storage. This application note has provided the building blocks that will allow the innovative manufacturer to remain on the forefront of technology.

- Advanced packaging, such as the TSOP, IC memory cards, and SIMM, is necessary for high-density applications.
- Intel Flash Memory allows flexible system interfacing by using I/O, paged, or linear addressing methods.
- Software variations enable an unlimited number of mass storage applications for Intel Flash Memory.
- Intel Flash Memory offers superior performance over the magnetic disk.

APPENDIX A

28F020 TSOP Dimensions



292079-60

Symbol	Description	Dimensions in mm		
		Min	Nom	Max
A	Package Height			1.20
A_1	Standoff	0.05		
A_2	Package Body Height	0.96	1.01	1.06
B	Lead Width	0.15	0.20	0.30
C	Lead Thickness	0.10	0.15	0.20
D	Package Body Length	18.20	18.40	18.60
E	Package Body Width	7.80	8.00	8.20
H_D	Terminal Dimension	19.80	20.00	20.20
L	Lead Tip Length	0.30	0.33	0.35
N	Lead Count		32	
Y	Seating Plane Coplanarity	0.00		0.10
Z	Lead to Package Offset	0.20	0.25	0.30
Ø	Lead Tip Angle	1	3	5

28F020 TSOP Physical Dimensions Drawings and Specifications

4

TSOP Sockets and Wands

32-Lead TSOP test sockets are available from the following vendors:

Enplas,
Part Number: 0TS-32-0.5-02
Distributed by:
Tesco International Inc.
1825 S. Grant Street, Suite 745
San Mateo, CA 94402
(415) 572-1683

32-Lead TSOP to DIP adapter sockets are available from the following vendor:

California Integration Coordinators Inc.,
Part Numbers: CIC-32TS-32D-AG-ENP-GANG-S,
CIC-32TS-32D-AG-ENP-GANG-R
656 Main Street
Placerville, CA 95667
(916) 626-6168

Emulation Technologies,
Part Numbers: AS-32-32-01TS-GENP-GANG-R
(F Version-Reverse),
AS-32-32-01TS-GENP-GANG-S
(E Version-Standard)
2344 Walsh Ave., Bldg. F
Santa Clara, CA 95051
(408) 982-0660

Suction wands for transferring units are available from the following vendor:

H-Square Corp.
1289-H Reamwood Avenue
Sunnyvale, CA 94089

1	V_{SS}	21	$\overline{CE3}$	41	A_{11}	61	DQ9
2	V_{CC}	22	$\overline{CE2}$	42	A_{10}	62	DQ8
3	V_{PP}	23	$\overline{CE1}$	43	A_9	63	DQ7
4	$\overline{OE}$	24	$\overline{CE0}$	44	A_8	64	DQ6
5	$\overline{WEH}$	25	V_{SS}	45	A_7	65	DQ5
6	$\overline{WEL}$	26	NC	46	A_6	66	DQ4
7	NC	27	NC	47	A_5	67	DQ3
8	RES	28	NC	48	A_4	68	DQ2
9	RES	29	NC	49	A_3	69	DQ1
10	RES	30	NC	50	A_2	70	DQ0
11	RES	31	NC	51	A_1	71	V_{PP}
12	RES	32	NC	52	A_0	72	V_{CC}
13	RES	33	NC	53	RES	73	PD1
14	RES	34	NC	54	V_{SS}	74	PD2
15	RES	35	NC	55	DQ15	75	PD3
16	RES	36	A_{16}	56	DQ14	76	PD4
17	RES	37	A_{15}	57	DQ13	77	PD5
18	RES	38	A_{14}	58	DQ12	78	PD6
19	RES	39	A_{13}	59	DQ11	79	PD7
20	RES	40	A_{12}	60	DQ10	80	V_{SS}

Figure 43. 1 Mbyte SIMM Pinout

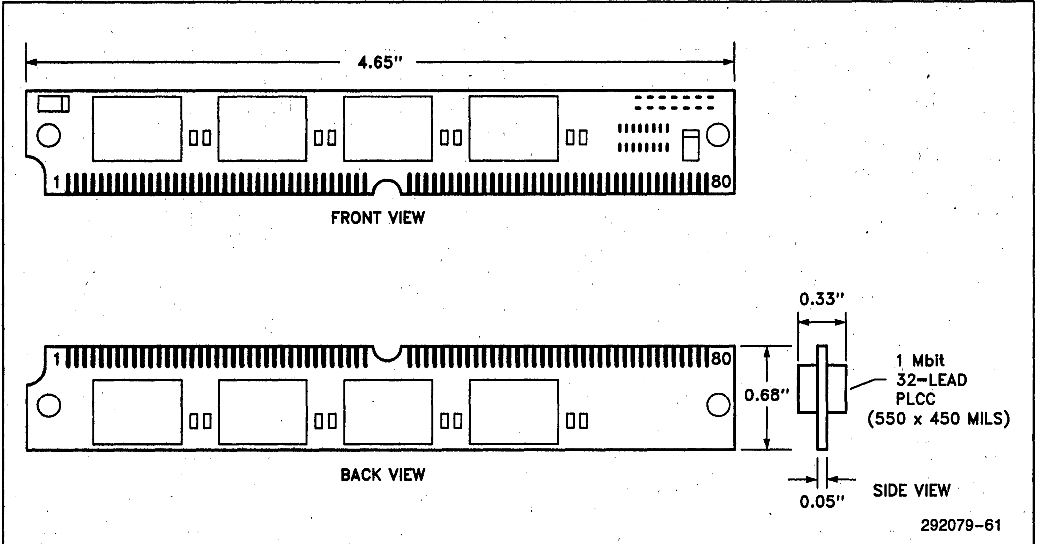


Figure 44. 1 Mbyte SIMM Dimensions

Module Capacity Idenfitication

Module Capacity Word Depth	PD6	PD2	PD1
No Module	O	O	O
256K/32M	O	O	S
512K/64M	O	S	O
1M/128M	O	S	S
2M/256M	S	O	O
4M/512M	S	O	S
8M/1G	S	S	O
16M/2G	S	S	S

NOTE:
These PD pins are JEDEC defined, not future product commitments.

Module Speed Identification

Maximum Access Time	PD7	PD5	PD4	PD3
>300 ns	S	S	S	S
300 ns	S	S	S	O
250 ns	S	S	O	S
200 ns	S	S	O	O
185 ns	S	O	S	S
150 ns	S	O	S	O
135 ns	S	O	O	S
120 ns	S	O	O	O
100 ns	O	S	S	S
85 ns	O	S	S	O
70 ns	O	S	O	S
60 ns	O	S	O	O
50 ns	O	O	S	S
40 ns	O	O	S	O
30 ns	O	O	O	S
ND	O	O	O	O

O = Open Circuit On Module
S = Short Circuit to Ground on Module
ND = Not Defined

```
EPLD ADF for Presence Detect WAIT-State Generator
PLFG Applications    1-800-323-EPLD
Intel Corp.
June 6, 1990
U999
002
85C220
Pre-loadable wait state down counter/READY generator
OPTIONS: TURBO = ON

PART: 85C220

INPUTS: CLK@1, nLOAD@2, PD7@3, PD6@4, PD5@5, PD4@6
OUTPUTS: nREADY@19, Q3@18, Q2@17, Q1@16, Q0@15, nDL@14

NETWORK:
    CLK = INP(CLK)              % System clock input %
    nLOAD = INP(nLOAD)          % Load count input %
    PD7 = INP(PD7)              % PD[7:4] Wait state %
    PD6 = INP(PD6)              % count size inputs %
    PD5 = INP(PD5)              % to lookup table %
    PD4 = INP(PD4)
    nREADY, nREADY = RORF(nREADYd,CLK,GND,GND,VCC) % /READY Output %

    Q3,Q3 = RORF (Q3d,CLK,GND,GND,VCC)          % counter outputs . . . %
    Q2,Q2 = RORF (Q2d,CLK,GND,GND,VCC)          % not externally %
    Q1,Q1 = RORF (Q1d,CLK,GND,GND,VCC)          % necessary %
    Q0,Q0 = RORF (Q0d,CLK,GND,GND,VCC)
    nDL,nDL = RORF (nDLd,CLK,GND,GND,VCC)

EQUATIONS:
    Q0d = Q0EQN * !READY * !COUNT_ZERO          % count if not ready %
        + Q0 * (READY + !LOAD)                  % hold if ready %
        + X0 * LOAD * !READY * COUNT_ZERO;      % read inputs on LOAD %
        Q0EQN = !Q0;

    Q1d = Q1EQN * !READY * !COUNT_ZERO
        + Q1 * (READY + !LOAD)
        + X1 * LOAD * !READY * COUNT_ZERO;
        Q1EQN = Q1 * Q0 + !Q1 * !Q0;

    Q2d = Q2EQN * !READY * !COUNT_ZERO
        + Q2 * (READY + !LOAD)
        + X2 * LOAD * !READY * COUNT_ZERO;
        Q2EQN = Q2 * (Q1 + Q0) + !Q2 * !Q1 * !Q0;

    Q3d = Q3EQN * !READY * !COUNT_ZERO
        + Q3 * (READY + !LOAD)
        + X3 * LOAD * !READY * COUNT_ZERO;
        Q3EQN = Q3 * (Q2 + Q1 + Q0) + !Q3 * !Q2 * !Q1 * !Q0

    nREADYd' = !Q3 * !Q2 * !Q1 * LOAD * !nDL;   % Anticipate counter %
                                                % to provide READY%
    nDLd = nLOAD;                               % hold until LOAD is%
                                                % taken away %
```

EPLD ADF for Presence Detect WAIT-State Generator

```
    COUNT_ZERO = !Q3 * !Q2 * !Q1 * !Q0;
    LOAD = NLOAD';
    READY = nREADY';

    X3 = GND;                                   % Wait State Scrambler %
    X2 = 3CNT;                                  % lookup table %
    X1 = 8CNT + 7CNT + 6CNT + 5CNT;
    X0 = 5CNT;

    8CNT =  PD7 * !PD6 * !PD5 * !PD4;
    7CNT = !PD7 *  PD6 *  PD5 *  PD4;
    6CNT = !PD7 *  PD6 *  PD5 * !PD4;
    5CNT = !PD7 *  PD6 * !PD5 *  PD4;
    4CNT = !PD7 *  PD6 * !PD5 * !PD4;
    3CNT = !PD7 * !PD6 *  PD5 *  PD4;
    2CNT = !PD7 * !PD6 *  PD5 * !PD4;

END$
```

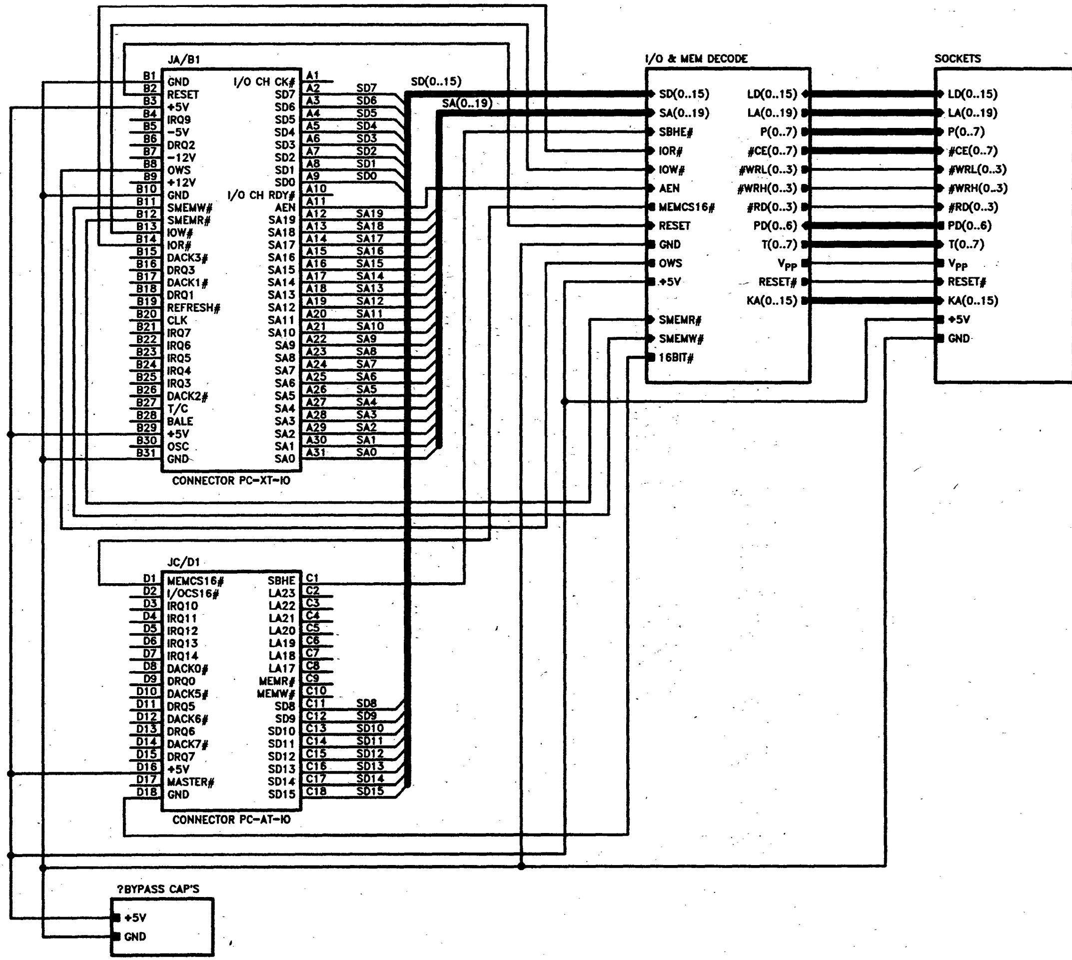
EPLD ADF for Presence Detect WAIT-State Generator (Continued)

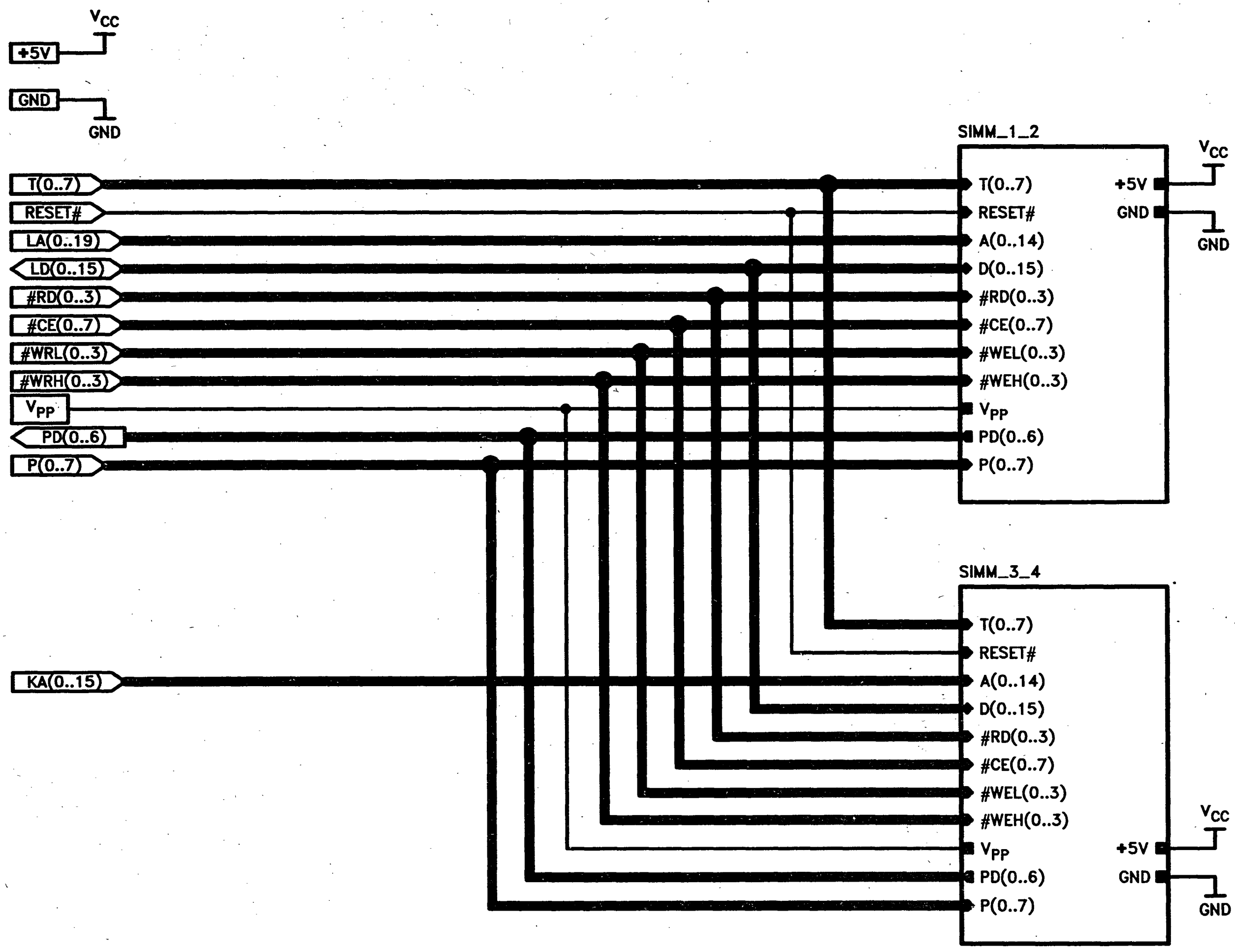
Decoding Truth Table for "Multiplexing" Data Bus of PCMCIA/JEIDA Memory Card

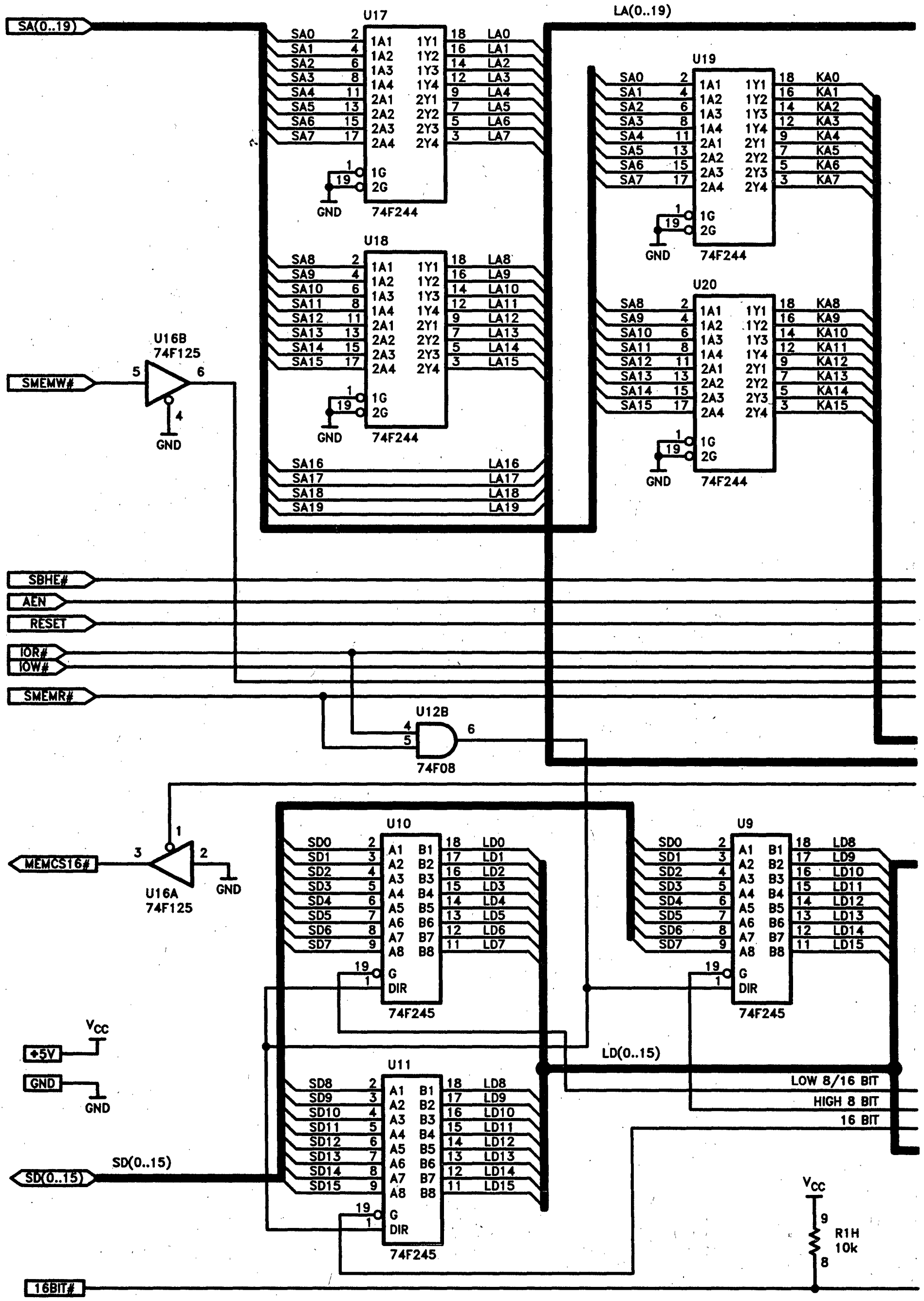
System Bus Width	Data Transfer	$\overline{CSH}$	$\overline{CSL}$	A_0	1	2	3
8 or 16	None	1	1	x	1	1	1
8 or 16	Lo-Byte	1	0	0	0	1	1
8	Hi-Byte to Lo-Byte	1	0	1	1	0	1
16	Hi-Byte	0	1	x	1	1	0
16	Low and High Byte	0	0	x	0	1	0

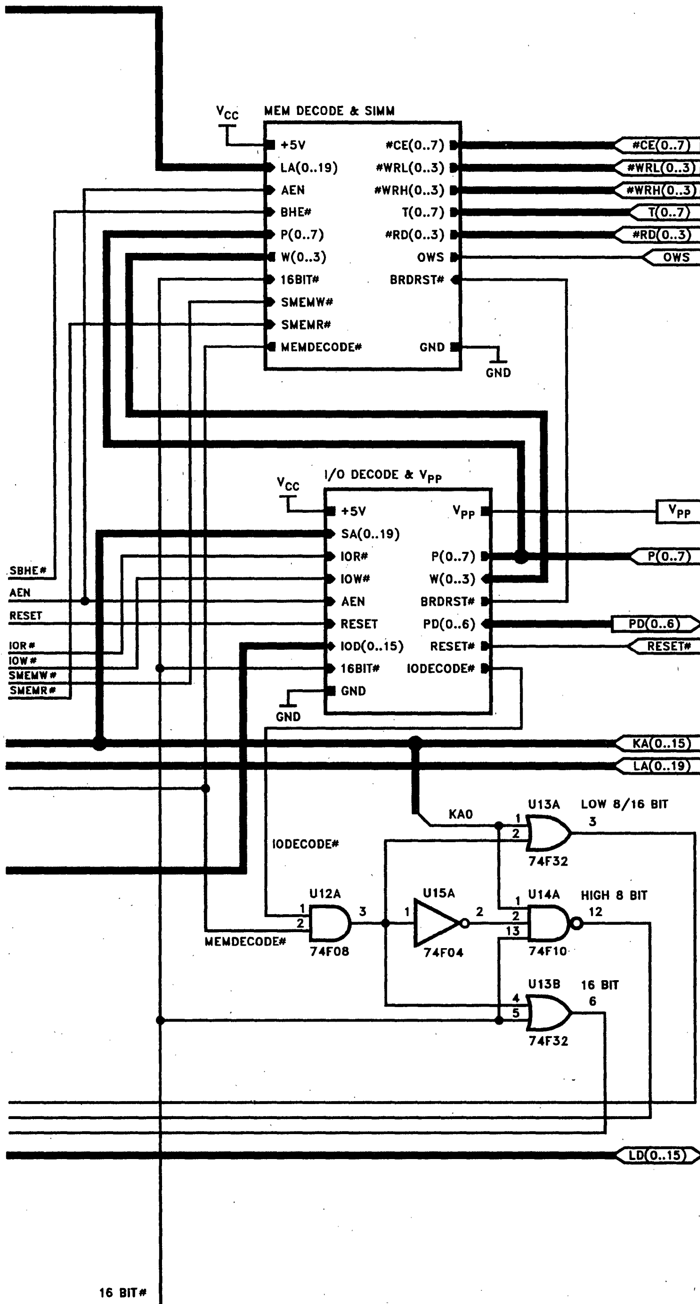
NOTE:
References Figure 8 in Memory Card Section.

PAGE MEMORY BOARD SCHEMATICS


JA/B1
CONNECTOR PC-XT-IO
B1 GND
B2 RESET
B3 +5V
B4 IRQ9
B5 -5V
B6 DRQ2
B7 -12V
B8 OWS
B9 +12V
B10 GND
B11 SMEMW#
B12 SMEMR#
B13 IOW#
B14 IOR#
B15 DACK3#
B16 DRQ3
B17 DACK1#
B18 DRQ1
B19 REFRESH#
B20 CLK
B21 IRQ7
B22 IRQ6
B23 IRQ5
B24 IRQ4
B25 IRQ3
B26 DACK2#
B27 T/C
B28 BALE
B29 +5V
B30 OSC
B31 GND
A1 I/O CH CK#
A2 SD7
A3 SD6
A4 SD5
A5 SD4
A6 SD3
A7 SD2
A8 SD1
A9 SD0
A10 I/O CH RDY#
A11 AEN
A12 SA19
A13 SA18
A14 SA17
A15 SA16
A16 SA15
A17 SA14
A18 SA13
A19 SA12
A20 SA11
A21 SA10
A22 SA9
A23 SA8
A24 SA7
A25 SA6
A26 SA5
A27 SA4
A28 SA3
A29 SA2
A30 SA1
A31 SA0
SD(0..15)
SA(0..19)
JC/D1
CONNECTOR PC-AT-IO
D1 MEMCS16#
D2 I/OCS16#
D3 IRQ10
D4 IRQ11
D5 IRQ12
D6 IRQ13
D7 IRQ14
D8 DACK0#
D9 DRQ0
D10 DACK5#
D11 DRQ5
D12 DACK6#
D13 DRQ6
D14 DACK7#
D15 DRQ7
D16 +5V
D17 MASTER#
D18 GND
C1 SBHE
C2 LA23
C3 LA22
C4 LA21
C5 LA20
C6 LA19
C7 LA18
C8 LA17
C9 MEMR#
C10 MEMW#
C11 SD8
C12 SD9
C13 SD10
C14 SD11
C15 SD12
C16 SD13
C17 SD14
C18 SD15
I/O & MEM DECODE
SD(0..15)
SA(0..19)
SBHE#
IOR#
IOW#
AEN
MEMCS16#
RESET
GND
OWS
+5V
SMEMR#
SMEMW#
16BIT#
LD(0..15)
LA(0..19)
P(0..7)
#CE(0..7)
#WRL(0..3)
#WRH(0..3)
#RD(0..3)
PD(0..6)
T(0..7)
VPP
RESET#
KA(0..15)
SOCKETS
LD(0..15)
LA(0..19)
P(0..7)
#CE(0..7)
#WRL(0..3)
#WRH(0..3)
#RD(0..3)
PD(0..6)
T(0..7)
VPP
RESET#
KA(0..15)
+5V
GND
?BYPASS CAP'S
+5V
GND
292079-16

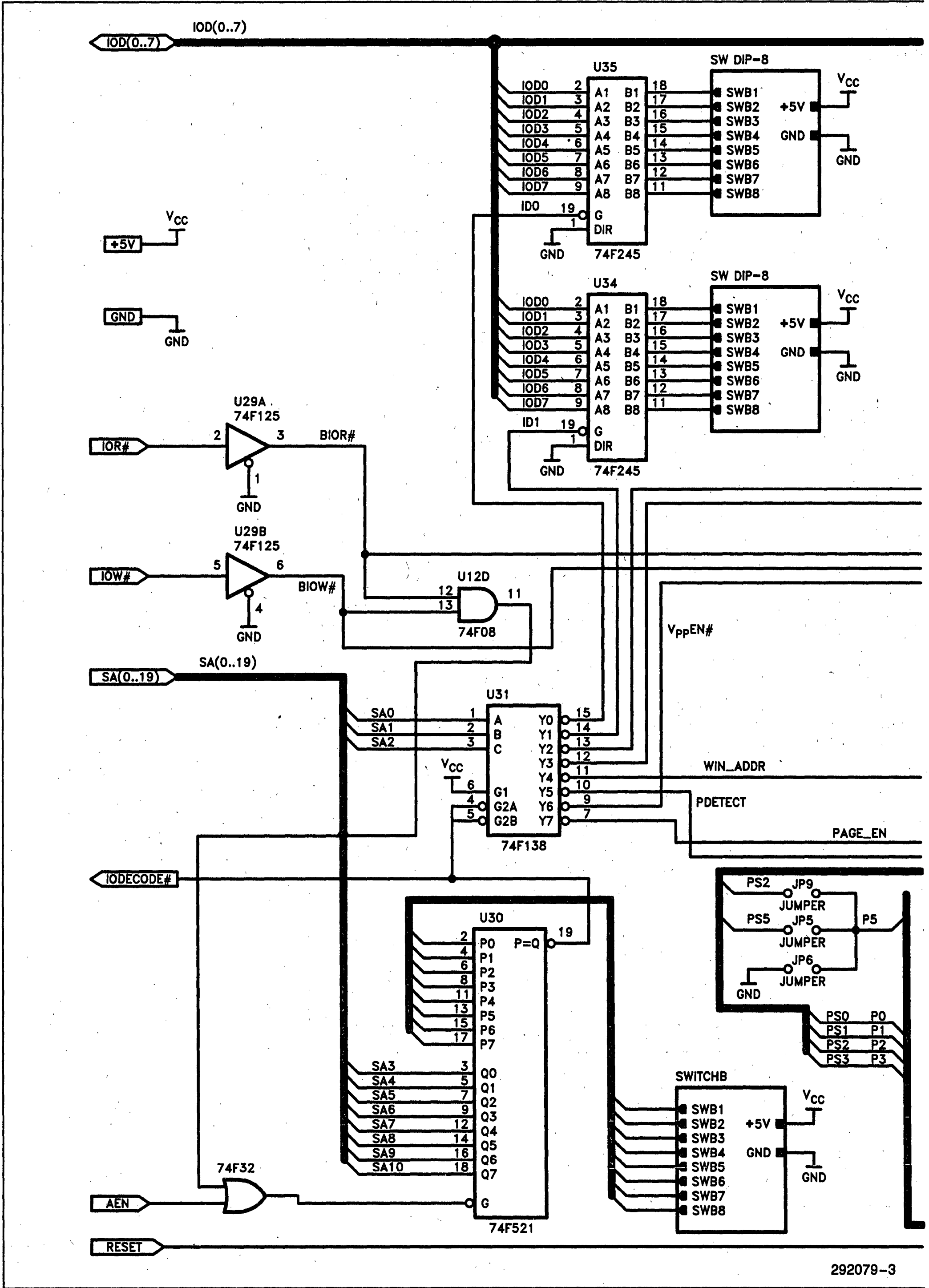

VCC
+5V
GND
GND
T(0..7)
RESET#
LA(0..19)
LD(0..15)
#RD(0..3)
#CE(0..7)
#WRL(0..3)
#WRH(0..3)
VPP
PD(0..6)
P(0..7)
KA(0..15)
SIMM_1_2
T(0..7)
RESET#
A(0..14)
D(0..15)
#RD(0..3)
#CE(0..7)
#WEL(0..3)
#WEH(0..3)
VPP
PD(0..6)
P(0..7)
+5V
GND
VCC
GND
SIMM_3_4
T(0..7)
RESET#
A(0..14)
D(0..15)
#RD(0..3)
#CE(0..7)
#WEL(0..3)
#WEH(0..3)
VPP
PD(0..6)
P(0..7)
+5V
GND
VCC
GND
292079-15

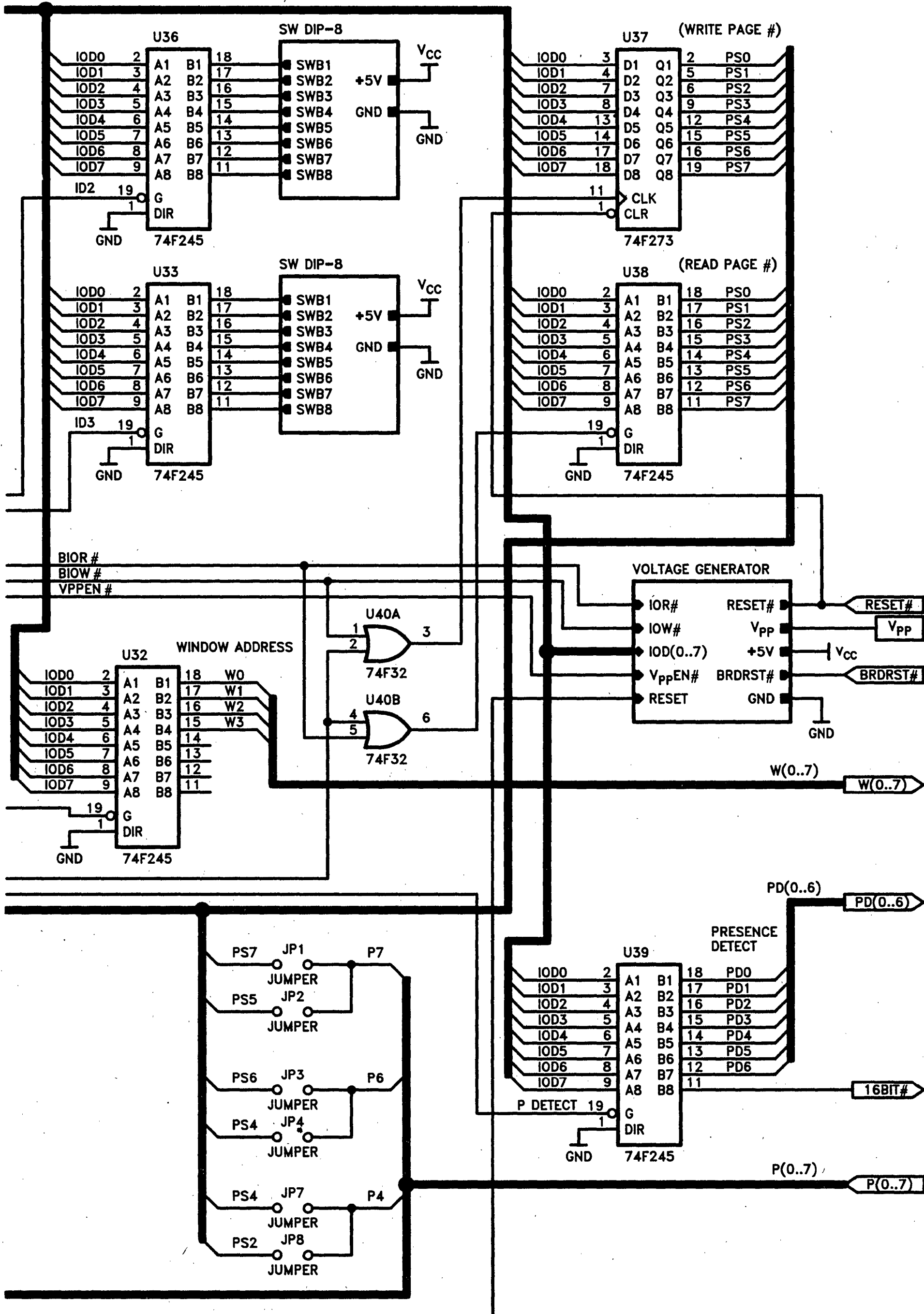

SA(0..19)
LA(0..19)
U17
74F244
U18
74F244
U19
74F244
U20
74F244
GND
U16B
74F125
SMEMW#
SBHE#
AEN
RESET
IOR#
IOW#
SMEMR#
U12B
74F08
MEMCS16#
U16A
74F125
U10
74F245
U9
74F245
U11
74F245
VCC
+5V
GND
SD(0..15)
LD(0..15)
LOW 8/16 BIT
HIGH 8 BIT
16 BIT
R1H
10k
16BIT#
292079-1



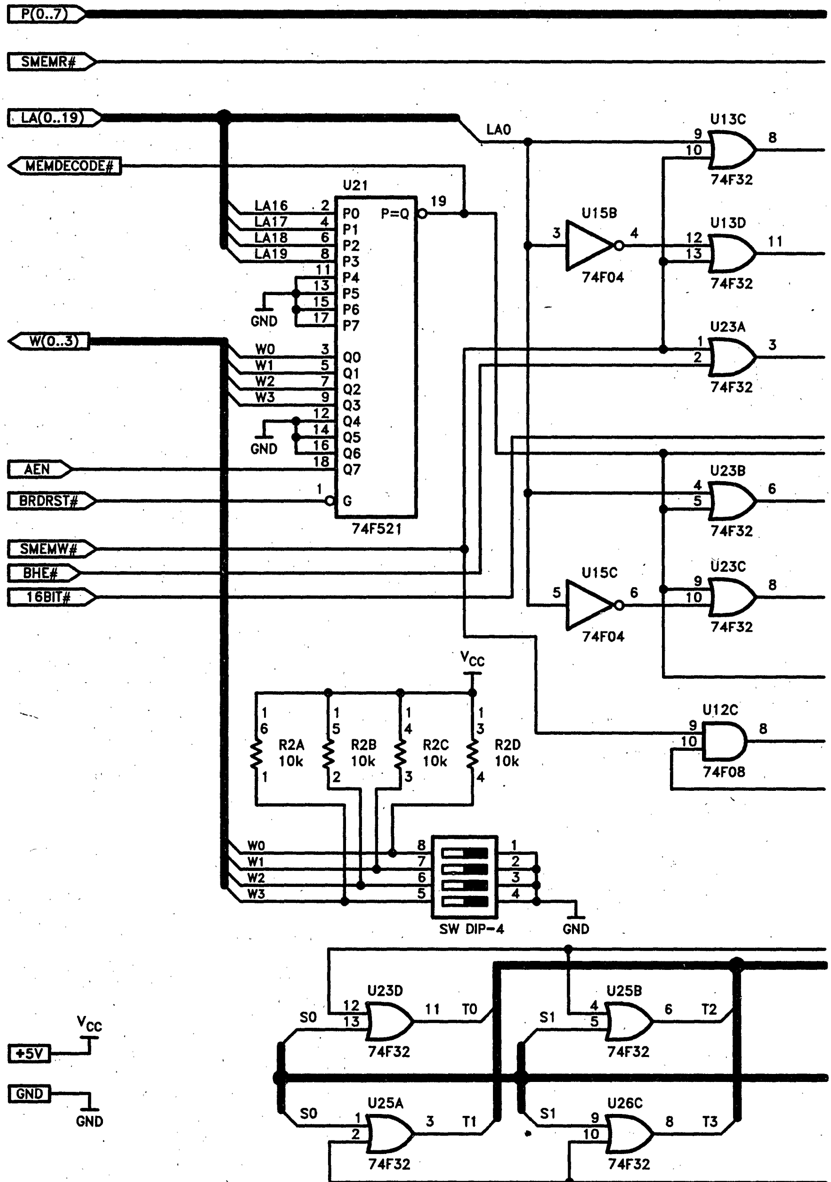
292079-2

4

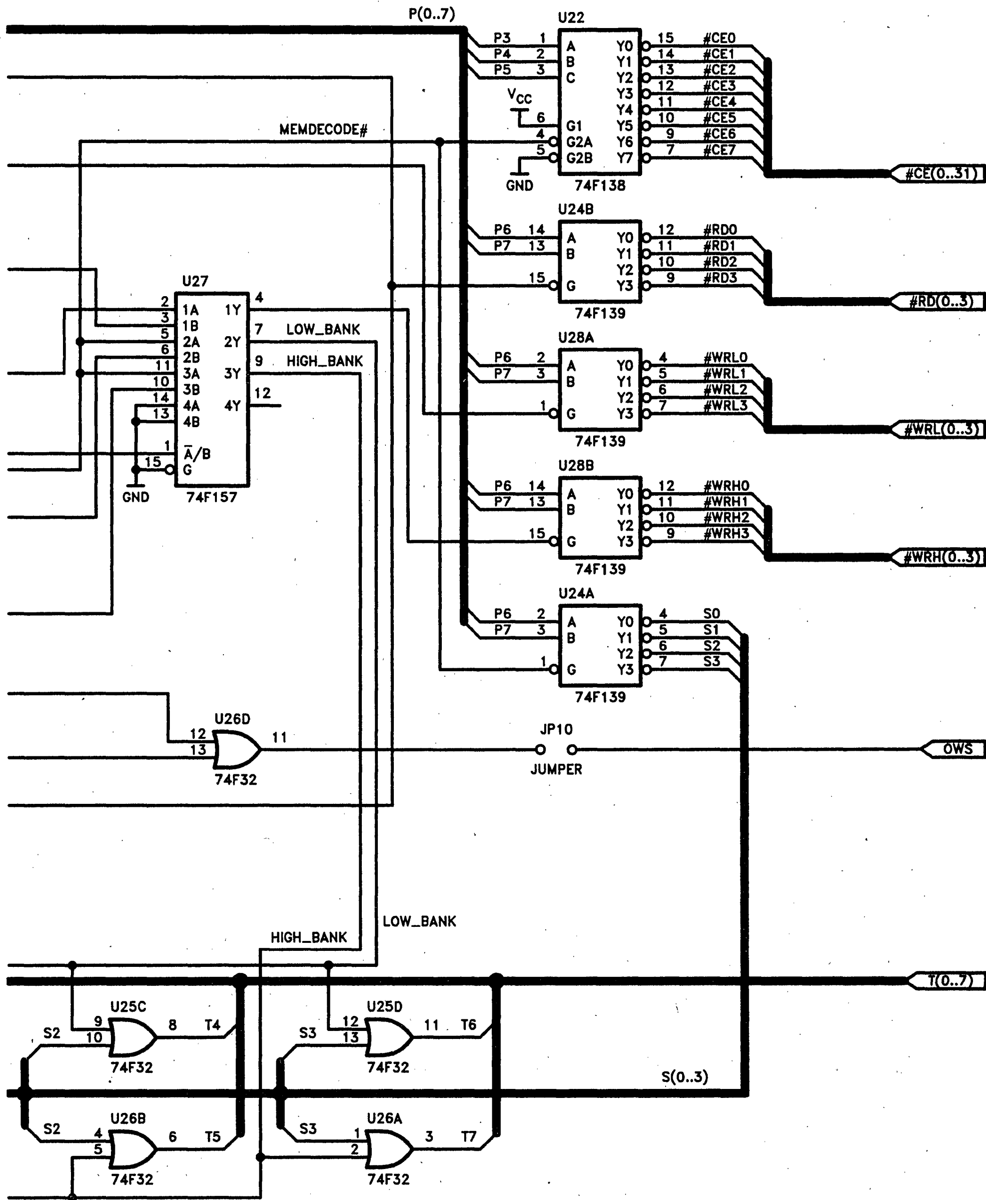

IOD(0..7)
U35
SW DIP-8
IOD0
IOD1
IOD2
IOD3
IOD4
IOD5
IOD6
IOD7
IDO
74F245
GND
+5V
VCC
U34
ID1
U29A
74F125
IOR#
BIOR#
U29B
IOW#
BIOW#
U12D
74F08
VPPEN#
SA(0..19)
U31
SA0
SA1
SA2
74F138
WIN_ADDR
PDETECT
PAGE_EN
IODECODE#
U30
P=Q
SA3
SA4
SA5
SA6
SA7
SA8
SA9
SA10
74F521
74F32
AEN
RESET
SWITCHB
PS2
PS5
JP9
JP5
JP6
JUMPER
P5
PS0
PS1
PS3
292079-3



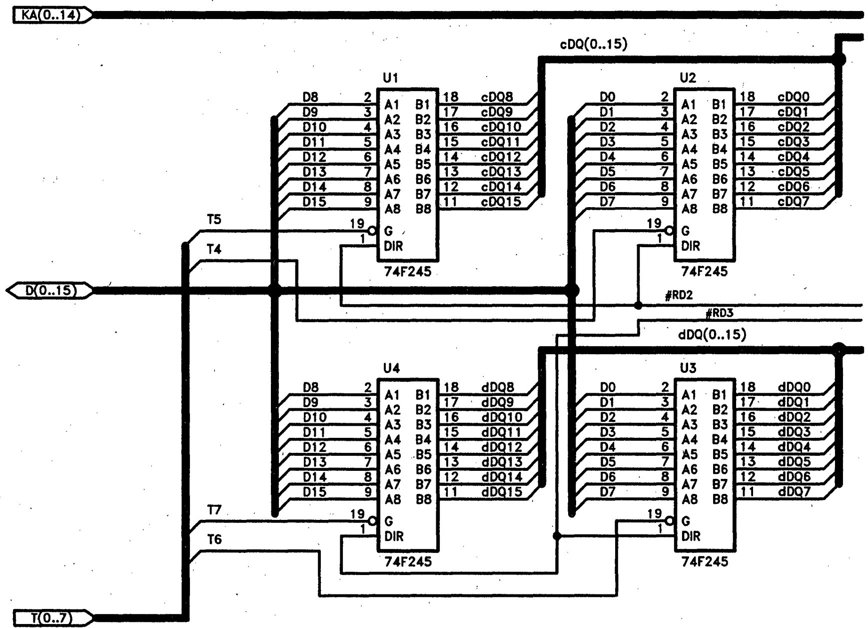
292079–4

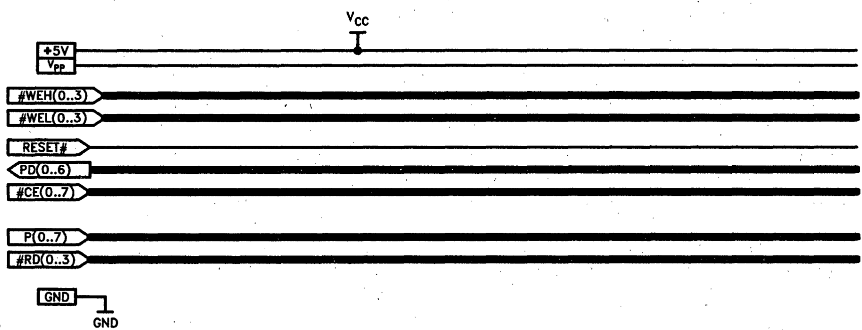


292079-9

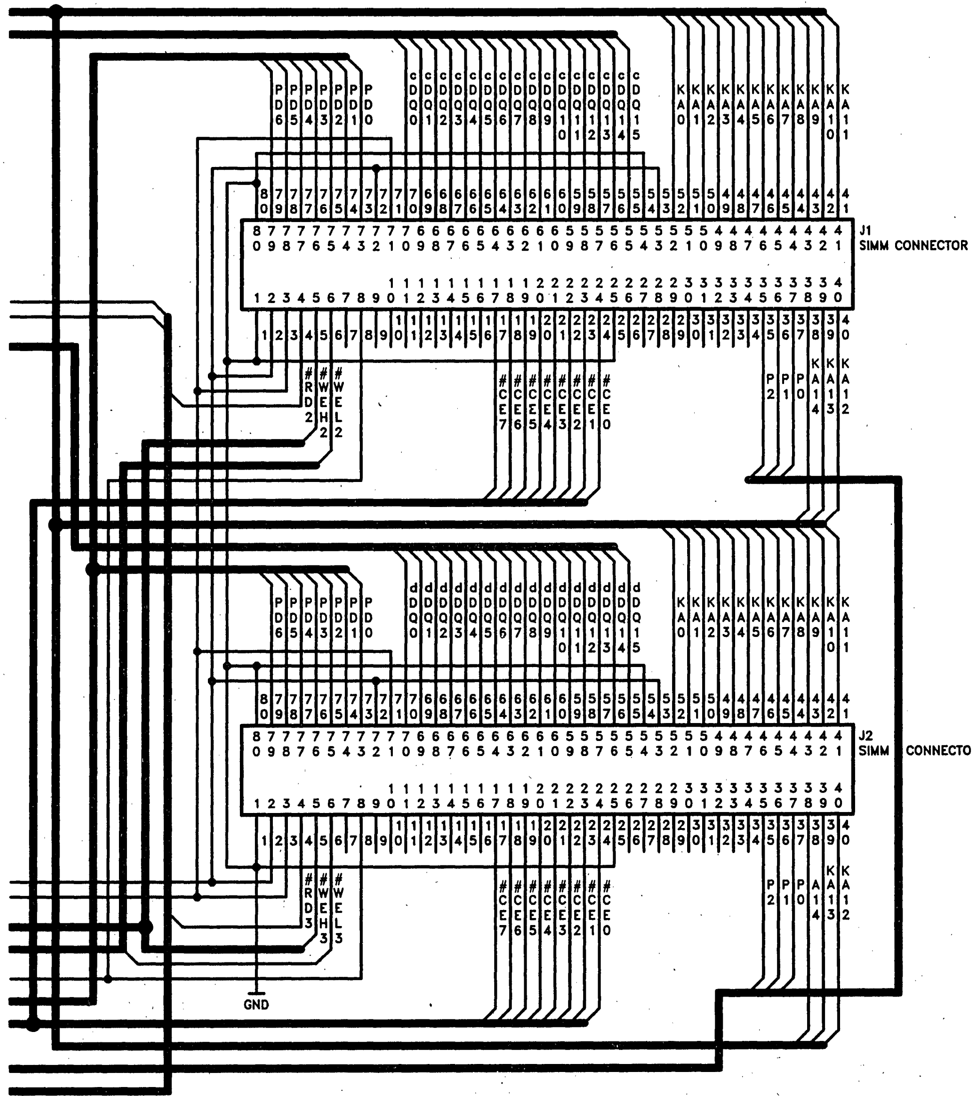


4

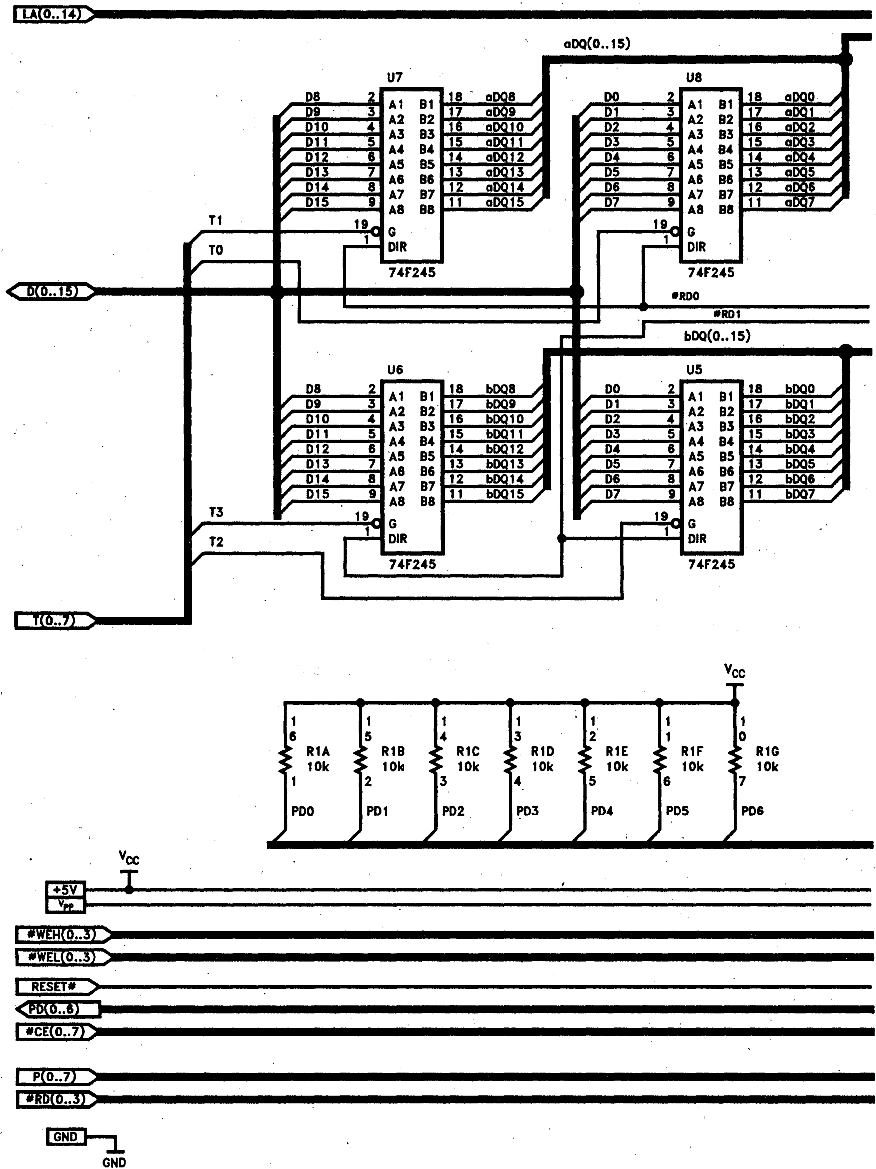




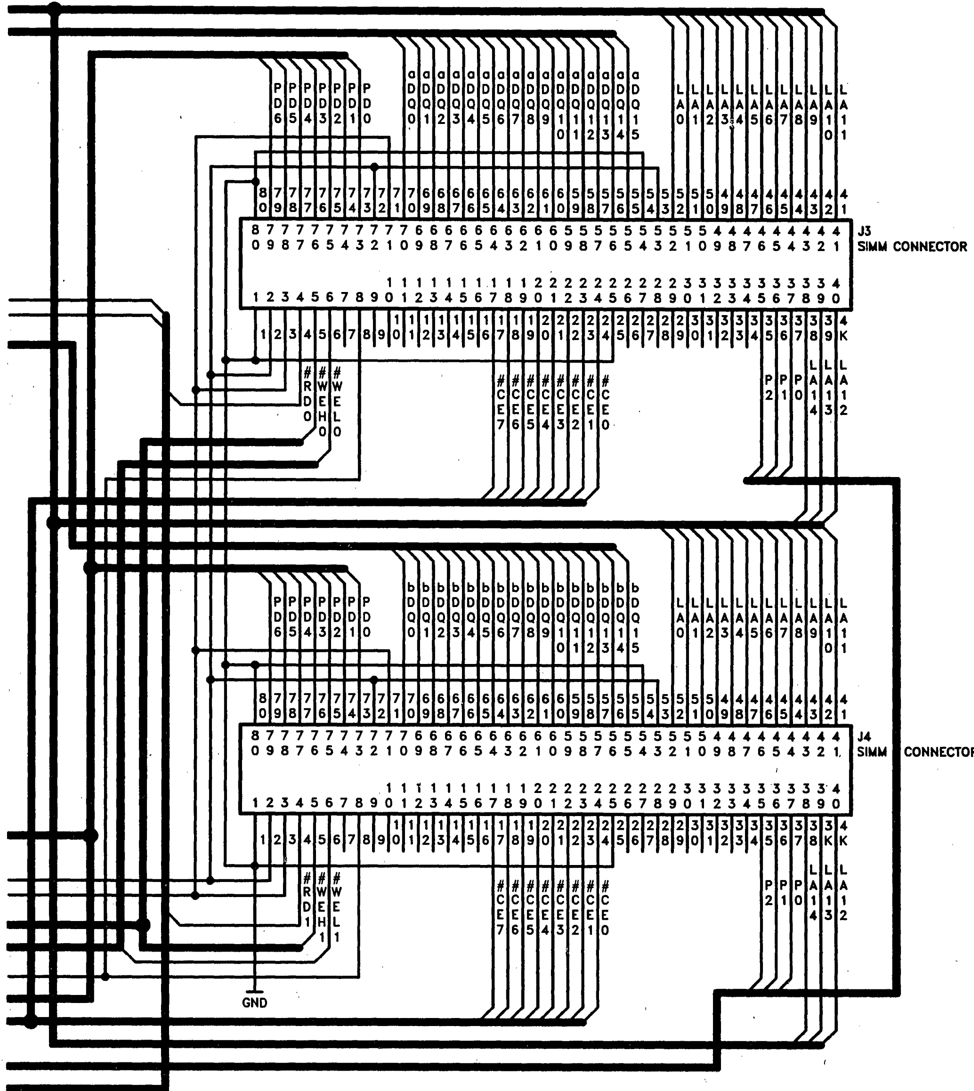
292079-5



4

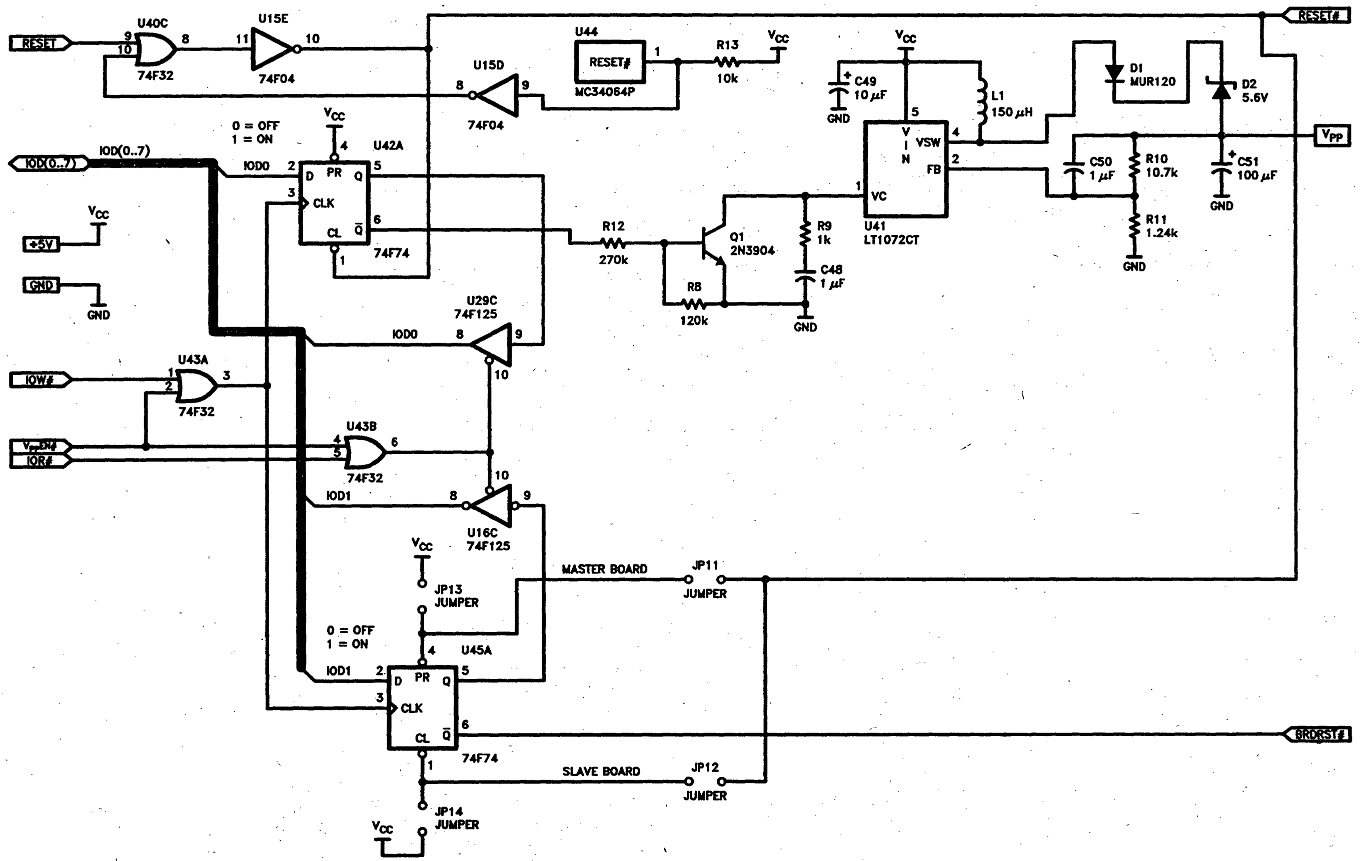


292079–7

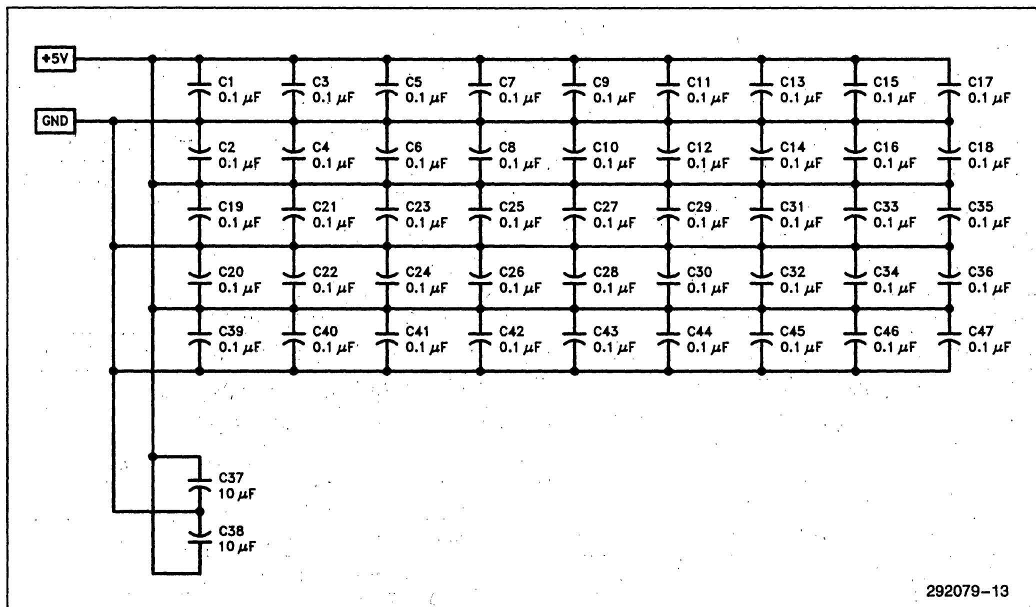


292079-8

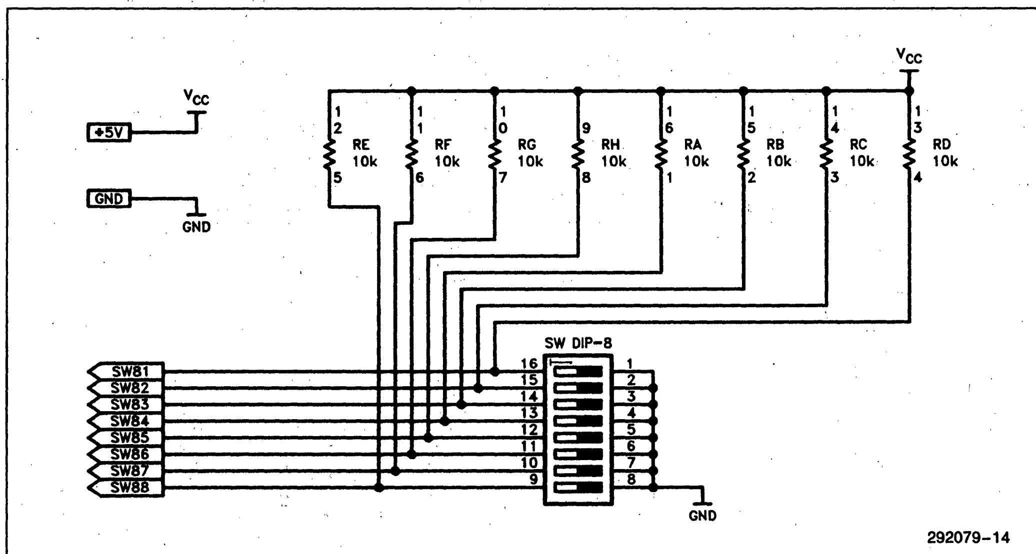
4



292079-11



292079–13



292079–14

I/O Port Usage for PCAT

Range	Usage	Range	Usage
0000–000fh	DMA Controller 1, 8237A	0390–0393h	Cluster (Adapter 0)
0020–0021h	Interrupt Controller 1, 8259A	03a0–03a9h	BSC Communications (Primary)
0040–005fh	Programmable Timer, 8254	03b0–03bfh	Monochrome/Parallel Printer Adapter
0060–006fh	Keyboard Controller, 8042		
0070–007fh	CMOS Real-Time Clock, NMI Mask	03c0–03cfh	EGA (Primary)
		03d0–03dfh	CGA
0080–009fh	DMA Page Registers, 74LS612	03f0–03f7h	Floppy Disk Controller
00a0–00a1h	Interrupt Controller 2, 8259A	03f8–03ffh	Serial Communications (COM 1)
00c0–00dfh	DMA Controller 2, 8237A	06e2–06e3h	Data Acquisition (Adapter 1)
00f0–00ffh	Math Coprocessor	0790–0793h	Cluster (Adapter 1)
01f0–01f8h	Fixed Disk	0ae2–0ae3h	Data Acquisition (Adapter 2)
0200–020fh	Game Controller	0b90–0b93h	Cluster (Adapter 2)
0278–027fh	Parallel Printer Port 2	0ee2–0ee3h	Data Acquisition (Adapter 3)
02b0–027dfh	EGA (Alternate)	1390–1393h	Cluster (Adapter 3)
02e1h	GPIB (Adapter 0)	2390–2393h	Cluster (Adapter 4)
02e2–02e3h	Data Acquisition (Adapter 0)	42e1h	GPIB (Adapter 2)
02f8–02ffh	Serial Communications (COM2)	62e1h	GPIB (Adapter 3)
0300–031fh	Prototype Card	82e1h	GPIB (Adapter 4)
0360–036fh	PC Network	a2e1h	GPIB (Adapter 5)
0378–037fh	Parallel Printer Port 1	c2e1h	GPIB (Adapter 6)
0380–038ch	SDLC Communications	e2e1h	GPIB (Adapter 7)

ASSEMBLY LANGUAGE CODE FOR PAGED BOARD

```
;****************************************************************************
;Locating the Base I/O Address.
;BOARD_NOT_FOUND is an error procedure and is not shown.

    BRD_IDO             dw 4 dup (?)
    Window_Base         dw ?
    Presence_Detect     dw ?
    VPPEN               dw ?
    Page_Number         dw ?

    mov dx,02F8h            ;Set port pointer to 02F8H.

KEEP_LOOKING:
    add dx,8                ;First valid address after adding.
    cmp dx,318              ;Port pointer = 8 less than highest port address?
    jg board_not_found      ;Hey, you forgot to install the board!!!

    in al,dx                ;Read port data into al register.
    cmp al,0Dh              ;Does register = 1st identifier value?
    jne KEEP_LOOKING        ;Not equal --> Not located yet

    inc dx
    in al,dx
    cmp al,0Ah              ;Does register = 2nd identifier value?
    jne KEEP_LOOKING

    inc dx
    in al,dx
    cmp al,01h              ;Does register = 3rd identifier value?
    jne KEEP_LOOKING

    inc dx
    in al,dx
    cmp al,0Eh              ;Does register = last identifier value?
    jne KEEP_LOOKING        ;TOO BAD, you almost had it!

 ;FOUND-Good Job!
    sub dx,3                ;Restored to base address.
    mov BRD_IDO,dx          ;Save the value in RAM.
```

Locating the Base I/O Address

NOTE:
A review of 8086 asembly language programming fundamentals might be necessary at this point.

```
;Locating the Base Memory Address

;**************************************************************************

;This Information is loaded into a segment register to access data

;**************************************************************************
    mov ax,0               ;Clear register
    mov dx,Window_Base     ;Port pointer = I/O to read base memory address
    in al,dx               ;Read port
    mov bx,256
    mul ax                 ;Shifts address information left.
    mov es,ax              ;es used as segment register for board.
```

```
;*************************************************************************
;Determining Memory Capacity

    Density_Lookup_Table dw ?,?,0fffh,7ffh,03ffh
    DENSITY dw ?

    mov ax,0                  ;Clears register.
    mov dx,Page_Number        ;Port pointer accesses page number.
    mov al,0
    out dx,al                 ;Write a zero to page number hardware.
    mov dx,Presence_Detect    ;Port pointer accesses presence detect pins.
    in al,dx
    and al,23H                ;Clears all but density information.
    cmp al,20H                ;Checks if PD6 is set.
    jng skip_or
    or al,4                   ;If greater than 20H, set bit 2 of al.

;Go to density lookup table, translate value from PD pins, store in RAM
;variable DENSITY. Density value must be 2, 3, or 4 to be valid.
 skip_or:
    cmp al,4
    je okay
    cmp al,3
    je okay
    cmp al,2
    je okay
    jmp Unknown_Device        ;Invalid or no SIMMs present, routine not shown.

;Base address of density lookup table stored in bx register.
    mov bx,offset Density_Lookup_Table
    mov si,ax                 ;si register will be pointer into density table
;Density values for 1M-4M, multiples of 1 Kbytes, stored in lookup table.
    mov ax,[bx+si]            ;Density read into ax register
    mov DENSITY,ax
;Read the device identifier. Use the es segment register for the base
;address of the memory.
;28F010 = 0B4h,28F020 = 0BDh
    mov ax,DENSITY            ;Put RAM held density info into ax.
    mov bx,1
    mov es:[bx],90H           ;Write ID command.
    mov bx,es:[bx]            ;Read device identifier.
    cmp bx,0B4h
    je 1MEG
```

Determining Memory Capacity

4

```
    cmp bx,0BDh
    je 2MEG

    jmp Unknown_device      ;If other than 28F010 or 28F020.

;Divide SIMM density by component density to determine number of components
;on SIMM.

1MEG:
    mov bx,3FFh
    div ax                  ;Divide ax/bx, # of components stored in al.
    jmp NEXT_OPERATION

2MEG:
    mov bx,7FFh
    div ax
    jmp NEXT_OPERATION

;Read from the next SIMM location to verify its presence.
;As an example, assume that the SIMM's density is 1 Mbyte.
;A 1 Mbyte SIMM has 16 pages.
    mov dx,Page_Number
    mov al,16
    out dx,al               ;Switch to Page 16 for next SIMM location.
    mov bx,1
    mov ex:[bx],90H         ;Write ID Command to first device of next SIMM.
    mov ax,es:[bx]          ;Read the identifier. Invalid data = empty socket
                            ;Repeat the process for all SIMMs.

;*****************************************************************************
```

Determining Memory Capacity (Continued)

intel®

APPLICATION NOTE

AP-361

October 1992

Implementing the Integrated Registers of the Series 2 Flash Memory Card

MARKUS LEVY
SENIOR TECHNICAL APPLICATIONS ENGINEER

Order Number: 292096-002

4

Implementing the Integrated Registers of the Series 2 Flash Memory Card

CONTENTS PAGE

INTRODUCTION 4-269

SERIES 2 COMPONENT MANAGEMENT REGISTERS 4-270

PCMCIA Release 2.0 Defined 4-270

Soft Reset Register 4-270

Global Powerdown Register 4-272

Performance Enhancement Registers .. 4-273

Sleep Control Register 4-273

Ready-Busy Status Register 4-274

Ready-Busy Mask Register 4-276

Ready-Busy Mode Register 4-277

Write Protection Register 4-280

Card Status Register 4-281

SUMMARY 4-282

GLOSSARY 4-283

RELATED DOCUMENTS 4-283

INTRODUCTION

Intel's first generation flash memory cards(1) forever changed the vision of solid-state storage. Electrically rewritable, non-volatile, reliable, yet economical in high densities, these cards provided a unique solution for the portable computing industry demanding such media. The second generation of flash memory cards provide even higher densities, lower power consumption and a higher level of functionality. The Series 2 Flash Memory Card delivers a major technology breakthrough by supporting densities up to 20 MBytes(2), an integrated memory control register set (**Component Management Registers** or **CMRs**) and PCMCIA 2.0/ExCA™ compliance.

Intel's 8-Megabit FlashFile™ Memory, 28F008SA, provides the foundation for the Series 2 Flash Memory Card. Its properties include data-write and block-erase automation, sixteen 64 KByte, separately-erasable blocks, a Ready/busy output pin, and a Powerdown mode. Within the Series 2 Card, high-functionality ASICs link the flash memory devices with the PCMCIA-specified electrical interface. These ASICs handle buffering, decoding and all control signals. They also contain the CMRs and hardwired Card Information Structure (CIS) used by system software to enhance device-level functions.

The OEM has many hardware and software alternatives for using the Series 2 Card. From a hardware perspective, the Intel 82365SL offers the most practical solution for controlling the PCMCIA socket in a PC solid-state drive application. This component, called the PC Card Interface Controller, provides the ExCA compliant hardware interface between the host system and the Series 2 Cards (and all other ExCA-compliant cards). As shown in Figure 1, the fundamental glue logic consists of a V_{PP} generator and V_{CC} control, a latching transceiver and address and decode signal buffers. Embedded systems not requiring an ExCA-compliant socket, can provide proper card signals with discrete circuitry.

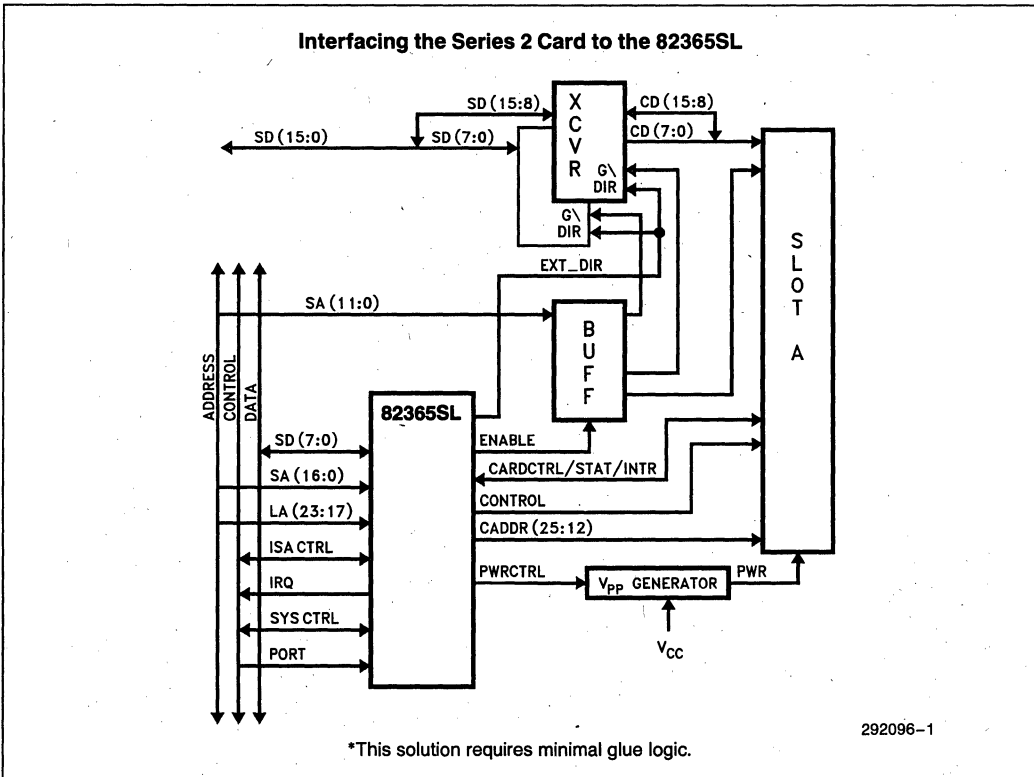


Figure 1. The 82365SL Establishes ExCA™ Compatibility with Minimal Glue Logic

NOTES:
1. The Bulk-Erase iMC001FLKA, iMC002FLKA, and iMC004FLKA (One, Two and Four Megabytes, respectively).
2. Higher density cards may be realized in the future as component densities go beyond 8 Megabits.

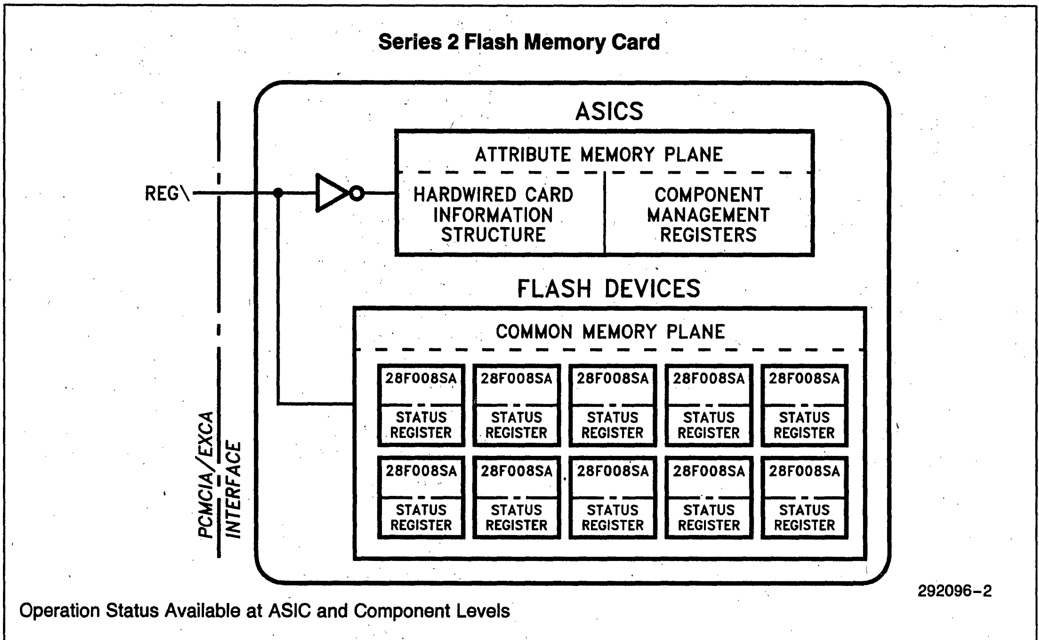


Figure 2. Selecting the Attribute or Common Memory Planes

Computer systems using the Series 2 Card as a solid-state disk drive employ file management software, such as Microsoft's* Flash File System with ExCA software. This software capitalizes on the architectural benefits of flash memory. It includes drivers that interface directly to the Series 2 Card. Beyond specifying the hardware architecture, ExCA provides a software solution that consists of modular software pieces designed for easy adaptation to the various hardware platforms and memory technologies. The various pieces of the ExCA system may be obtained from your BIOS vendor. Essentially, this means that a system OEM is relieved of having to implement the integrated registers of the Series 2 Flash Memory Card.

This application note supplements the information contained in the Series 2 Flash Memory Card Data Sheet. It benefits OEMs developing their own Series 2 Flash Memory Card software pieces, including custom flash file management software and software for embedded systems running non-DOS applications. Specifically, it describes the software aspects of implementing the card's CMRs which provide software control of many 28F008SA functions, elevating the system designer above device-level issues used by higher-level file system software.

SERIES 2 COMPONENT MANAGEMENT REGISTERS

The CMRs optimize the Series 2 Flash Memory Card's performance by supplying a software-controlled interface to the individual devices within the card. As shown in Figure 2, they are accessed as memory-mapped I/O in the Attribute Memory Plane by pulling the card's Register Select pin low ($\overline{REG}$, pin 61)(3). CMRs can be divided into two basic categories; those defined by the PCMCIA Release 2.0 specification and Intel defined "Performance Enhancement Registers".

PCMCIA RELEASE 2.0 DEFINED

Soft Reset Register
(Configuration Option Register)

During card operation, it may be necessary to place the card into a known state by resetting the 28F008SA-level Status Registers and the CMRs in the ASICs to their power-on conditions (Figure 3). Specifically, in the

NOTE:
3. No switch-over setup-time from Common Memory is needed when PCMCIA timing requirements are met.

Component Managment Registers(4)

Defined by the PCMCIA R2.0 specification
- **Soft Reset Register (5)—(R/W)**
- **Global Powerdown Register (6)—(R/W)**

PERFORMANCE ENHANCEMENT REGISTERS designed to deliver control benefits tied directly to the Intel 28F008SA:

- **Sleep Control Registers—(R/W)**
- **Ready-Busy Status Registers—(RO)**
- **Ready-Busy Mode Registers—(R/W)**
- **Ready-Busy Mask Registers—(R/W)**
- **Write Protection Registers—(R/W)**
- **Card Status Register—(RO)**

ASICs, this reset affects the PwrDwn bit (**Global Powerdown Register**), the **Sleep Control Register**, the **Ready-Busy Mode Register**, the **Ready-Busy Mask Register**, and the CISWP and CMWP bits (**Write Protection Register**). There are several ways to enter power-on status:

1. Issuing a hardware reset, with a complete system reset or socket reset through the interface hardware, affects the entire system or the Series 2 Card, respectively.
2. During normal operation of many portable systems, such as those employing the 386SL™ microprocessor, tremendous power savings are realized by entering a suspend state. In this state, power to the card's socket is removed. After reapplying power, the card automatically attains its power-on status. Therefore, before removing power from the Series 2 Card, system software must save the contents of the Component Management Registers. It should also be pointed out, that a startup period must elapse to allow all internal circuitry to stabilize before accessing the card. This time period depends on host system power supply capabilities.(7)
3. The third method utilizes a software-controlled mechanism built into the Series 2 Card. This option, activated with the **Soft Reset Register**, provides a simple approach for placing the card in its power-on state without time delay.

The **Soft Reset Register** (Figure 4) contains a soft reset (SRESET) bit that performs a function similar to the hardware reset invoked by the card's RESET pin ($\overline{RST}$, pin 58)(8). Achieve the reset condition by issuing a two-step write sequence to the SRESET bit (i.e. toggling from 0 to 1 and back to 0).

During reset (SRESET = 1), the ASICs drive the flash memory array into the deep-sleep mode. This aborts any device operations in progress and resets each device's Status Register. After initiating a soft reset, the SRESET bit *must* be cleared (zero) to enable access to the flash memory array or write to another CMR. The host system can clear this bit by writing in a zero or issuing a hardware reset.

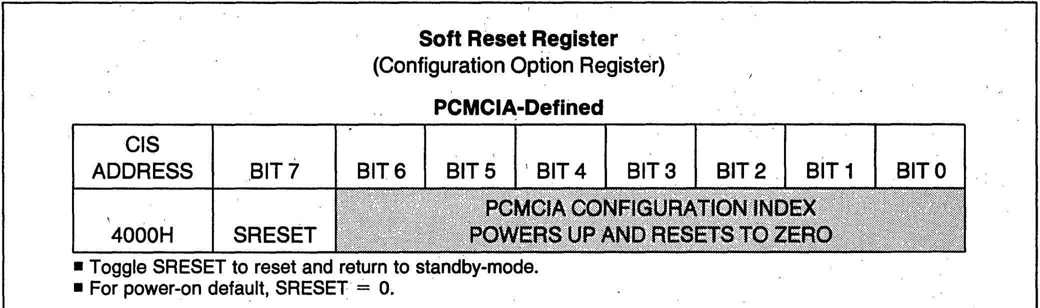
Power-On Conditions*

ALL DEVICES IN STANDBY MODE.
SOFTWARE WRITE-PROTECT DISABLED.
ALL DEVICES' READY/$\overline{BUSY}$ OUTPUTS UNMASKED.
PCMCIA-READY/$\overline{BUSY}$ MODE ENABLED.
READY/$\overline{BUSY}$ OUTPUT PIN GOES TO READY.

NOTE:
Generated by Hardware Reset or Toggling SRESET Bit.

Figure 3

NOTES:
4. R = READ, W = WRITE, RO = READ ONLY
5. Referred to as Configuration Option Register by PCMCIA R2.0.
6. Referred to as Configuration and Status Register by PCMCIA R2.0.
7. As specified by PCMCIA Release 2.0.
8. Soft reset puts all devices into power-down mode and requires a recovery time after returning from soft reset (500 ns for reads and 1 μs for writes).



Soft Reset Register
(Configuration Option Register)

PCMCIA-Defined

CIS ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4000H	SRESET	PCMCIA CONFIGURATION INDEX POWERS UP AND RESETS TO ZERO						

- Toggle SRESET to reset and return to standby-mode.
- For power-on default, SRESET = 0.

Figure 4. Useful for placing the card into a known state

The other two fields (not implemented with the Series 2 Card), defined in this register by the PCMCIA R2.0 specification, include the Configuration Index and the LevlREQ. After powerup or soft reset, the Configuration Index contains zeros to maintain compatibility as a Memory-Only Interface. The LevlREQ bit is hardwired to zero.

GLOBAL POWERDOWN REGISTER

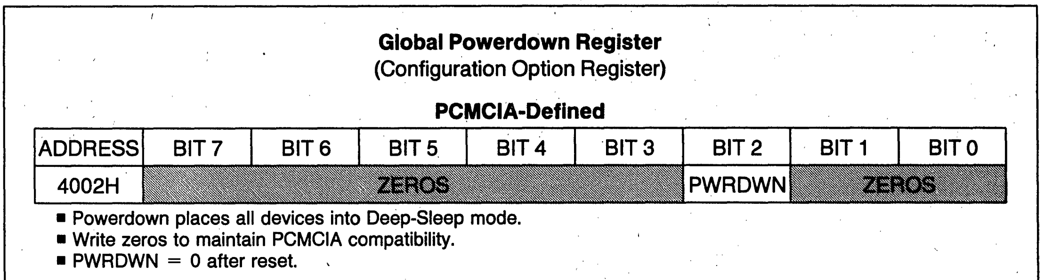
PCMCIA R2.0 Defined

(Configuration and Status Register)

The portable system designer strives to minimize power consumption in every conceivable way. Solid-state storage devices using Intel Flash Memory deliver significant power consumption reductions (when compared to the mechanical disk) and therefore play an important part of the system design considerations. The portable system aimed at power conservation looks to shut down portions of system circuitry not in use (i.e. the solid-state drive not accessing files, the screen's backlight when the keyboard has not been touched in a certain amount of time, etc.). Powering down the entire socket achieves a minimal power usage status. However, the powerup recovery time from this approach produces varying delays.

The Series 2 Card offers the optimal solution with the **Global Powerdown Register** (Figure 5). Writing a one (1) to the Power-Down Bit (PwrDwn, bit 2) of this register puts all internal devices into the Deep-Sleep Mode by pulling every device's $\overline{\text{PWD}}$ input low[9]. In the Deep-Sleep mode, a 20 Megabyte Series 2 Card consumes 90% less current versus the standby mode current[10].

When the host system drives the two card enable pins high[11], the Series 2 ASIC circuitry blocks system-level address and data signals from the internal devices. Additionally, latching address buffers and data transceiv-



Global Powerdown Register
(Configuration Option Register)

PCMCIA-Defined

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4002H	ZEROS					PWRDWN	ZEROS	

- Powerdown places all devices into Deep-Sleep mode.
- Write zeros to maintain PCMCIA compatibility.
- PWRDWN = 0 after reset.

Figure 5

NOTES:
9. The remaining fields in this register (Changed, SigChg, IOis8, Audio, Intr and Rsvd) are tied low in the Series 2 Card for PCMCIA compatibility and for simplifying software masking.
10. I_{CCS} = 30 μA vs, I_{CCSL} = 0.2 μA; refer to 28F008SA Data Sheet. The ASICs consume 1 μA.
11. $\overline{\text{CE1}}$ (pin 7) and $\overline{\text{CE2}}$ (pin 42) = V_{IH}

SAMPLE 80X86 CODE TO HANDLE RECOVERY-PERIOD TIMING

```
   GLOBAL_PWD      EQU     4002H        ;Global PowerDown Register
   NOT_PWRDWN      EQU     0H

   MOV AX, MEM_CARD_BASE        ;Load card address
   MOV ES, AX
   MOV DI, GLOBAL_PWD           ;Pointer setup

;Software assumes already in REG# mode access
   MOV BYTE PTR ES:[DI], NOT_PWRDWN ;Clears PWRDWN bit

   MOV CX, RECOVERY_TIME        ;Based on speed of processor
 FOR_A_WHILE:
   LOOP FOR_A_WHILE
```

Figure 6. Assembly Language Code for Returning from "Deep-Sleep" Mode

ers on the host side eliminate address and data signal switching at the Series 2 Card input buffers further reducing power consumption levels. In other words, to achieve the lowest power consumption levels, these signals should not be floated or tristated.

After clearing the PwrDwn bit, the device-recovery times must be met before accessing the flash memory. As shown in Figure 6, the recovery period can be implemented using a simple software algorithm(12).

Prior to entering the Powerdown Mode, your software must check operation status for data-writes or block-erases in progress(13). Any operations in progress will be terminated when powering down the flash array. The 28F008SA does not maintain Status Register contents in the Powerdown Mode. Therefore, when the card returns to standby mode, all devices will report **successful** status (Status Register = 80H) indicating the need for software drivers to use the powerdown function intelligently.

PERFORMANCE ENCHANCEMENT REGISTERS

Sleep Control Register

(Performance Enhancement Register)

The powerdown functionality of the **Global Powerdown Register** has a global affect on all devices. In many solid-state storage applications, reading or writing files only requires access to select device pairs and the remaining devices could be kept in Deep-Sleep status until needed.

Sleep Control Registers
Performance Enhancement Register

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
411AH	RESERVED						DEVICES 18/19	DEVICES 16/17
4118H	DEVICES 14/15	DEVICES 12/13	DEVICES 10/11	DEVICES 8/9	DEVICES 6/7	DEVICES 4/5	DEVICES 2/3	DEVICES 0/1

- For reset, all devices powered up (bits = 0).
- On cards less than 20 megabytes, absent devices read as "0"s.
- Bits cleared to zero by SRESET and RESET.

Figure 7. Allows Selective Powerdown of Devices within the Series 2 Card

NOTE:
12. PCMCIA does not specify a maximum recovery time. Recovery times, varying for different card technologies, must be handled on a case-by-case basis.
13. Polling the individual device's Status Register, the Ready/Busy Status Register, or the RDY/$\overline{BSY}$ bit in the Card Status Register.

28F008SA Status Register Bit Definition

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
WSM STATUS	ERASE SUSPEND STATUS	ERASE STATUS	WRITE STATUS	VPP STATUS	RESERVED		

Figure 8. Read during Write or Erase Operations to Determine Status

The **Sleep Control Register** (Figure 7) offers this option; each bit provides power down for a specific device pair. Except for the global vs individual affect, this register functions identically to the **Global Powerdown Register**. The global powerdown can be enabled while individual devices are sleeping. Disabling the global PWRDWN does not affect prior bit settings of the **Sleep Control Register**.

In many applications using the Series 2 Card, the card will be in the Standby Mode a large percentage of the time. This avoids device recovery times associated with complete socket power off or entering the Deep-Sleep Mode. In the Standby Mode, the **Sleep Control Register** offers the greatest advantage over the **Global Powerdown Register**. With the capability of controlling individual device pairs, a power savings improvement of approximately 16 times (based on typical current values) will be seen. This is derived from the following information:

- 28F008SA devices in Deep-Sleep; $I_{CC} = 0.2\ \mu A$, $I_{PP} = 0.1\ \mu A$.
- 28F008SA devices in Standby; $I_{CC} = 30\ \mu A$, $I_{PP} = 1\ \mu A$.
- ASICs in Standby and Sleep; $I_{CC} = 1\ \mu A$.
- With device-pair control, unaccessed devices remain in Deep Sleep.

Although the other operating modes (read, data-write, or block-erase) also experience power savings by using the **Sleep Control Register**, the effects are not as significant relative to the higher current requirements of those modes.

When using the **Sleep Control Register**, software must account for the same device-recovery time of the global powerdown method. To access files (or data) that span multiple device pairs (and experience uninterrupted access), software can perform a "look-ahead" function to determine which device pairs must be powered up.

READY-BUSY STATUS REGISTER

Performance Enhancement Register

The automated data-write and block-erase capability of the Intel 28F008SA FlashFile Memory results in a significant performance improvement. Furthermore, automation simplifies system-level interfacing as the user only delivers the proper command and monitors the operation's READY/BUSY status. Referring to the 28F008SA Data Sheet (or Figure 8), operation status can be obtained from the device's *Status Register* or RY/$\overline{\text{BY}}$ pin. The device's Status Register allows software polling for ready status in addition to write and erase status. The RY/$\overline{\text{BY}}$ pin can be used to generate an interrupt when making a busy to ready transition. Regardless of the method used for determining ready status, the Status Register should be read to determine whether an operation was successful.

In the Series 2 Card, where multiple devices are present and multiple simultaneous operations can occur, software polling each device's Status Register requires extra software and time. Furthermore, the PCMCIA interface only has one RDY/$\overline{\text{BSY}}$ pin which obviously prevents 20 devices from hooking their individual RY/$\overline{\text{BY}}$ out to the system. The ASICs within the card take these signals and feed them into the **$\overline{\text{BUSY}}$ Status Register** (Figure 9). This facilitates multiple device-pair operations by allowing an analysis of all devices simultaneously. After initiating the data-write and block-erase operations, the system can switch the card to the Attribute Memory Plane to access these registers. Alternatively, each device's RY/$\overline{\text{BY}}$ signal funnels into a single "wired or" signal that becomes the PCMCIA-RDY/$\overline{\text{BSY}}$ pin driving an interrupt or polled through an I/O port.

When performing single device pair operations, Ready/Busy status should be accessed directly from the Status Register of the flash memory devices for the following reasons: 1) A device's Status Register must be read anyway to determine the result of an operation; 2) This saves several instructions required to switch to the Attribute Memory Plane.

Ready-Busy Status Register
Performance Enhancement Register

CIS ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4134H	RESERVED				DEVICE 19	DEVICE 18	DEVICE 17	DEVICE 16
4132H	DEVICE 15	DEVICE 14	DEVICE 13	DEVICE 12	DEVICE 11	DEVICE 10	DEVICE 9	DEVICE 8
4130H	DEVICE 7	DEVICE 6	DEVICE 5	DEVICE 4	DEVICE 3	DEVICE 2	DEVICE 1	DEVICE 0

- Each bit corresponds to a device's RY/$\overline{BY}$ signal.
- Devices not present (i.e. < 20 Megabytes) return ready status.

Figure 9. Monitors Individual Device's RY/$\overline{BY}$ Pins

Example for Monitoring Ready/Busy Status

(Assume ES contains memory card base address)

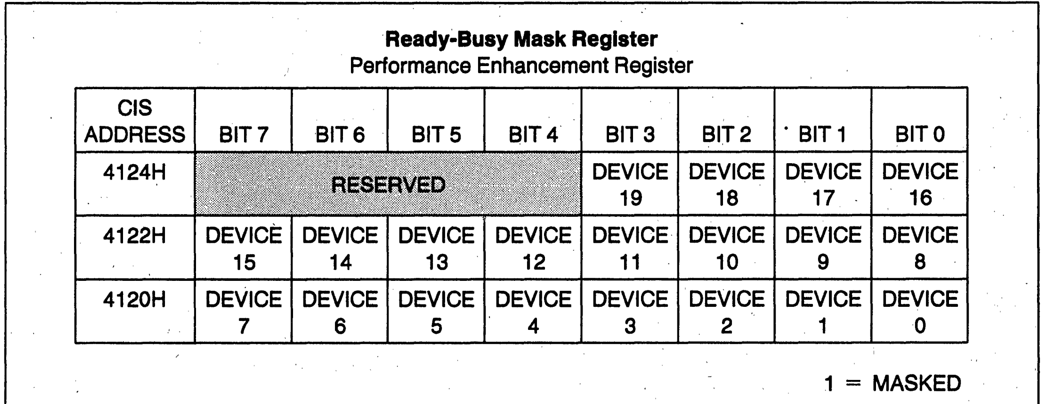
```
RDY_BSY_STATUS          EQU 4130H ;Register address
DEVICE_0                EQU 01H   ;Settings in register for specific devices
DEVICE_1                EQU 02H
DEVICE_2                EQU 04H
DEVICE_3                EQU 08H
DEVICE_4                EQU 10H
DEVICE_5                EQU 20H

XOR AX, AX                           ;Zero AX Register
MOV DI, RDY_BSY_STATUS

;Insert code to start write operation in first 3 Device Pairs
;i.e. Devices 0, 1, 2, 3, 4, 5.

OR AX, DEVICE_0
OR AX, DEVICE_1
OR AX, DEVICE_2
OR AX, DEVICE_3
OR AX, DEVICE_4
OR AX, DEVICE_5

;Assume card already in REG mode.
TEST BYTE PTR ES:[DI], AX         ;Zero flag cleared when programming
                                  ;devices are ready.
```

READY-BUSY MASK REGISTER Performance Enhancement Register

As described earlier, completion of a data-write or block-erase operation can be determined by attaching the card's RDY/BSY pin into a system interrupt. This frees the host system to perform alternate tasks after initiating an operation. In other words, device-level automation allows Series 2 Card operations to become background tasks.

Occasions exist where the interrupt generated from a device becoming ready produces unacceptable latency times. For instance, data-write operations, completing in only 10 μs, realize a performance penalty dealing with interrupt latencies longer than the write time itself. The data-write operations would achieve a higher level of performance by using software polling techniques[14]. On the other hand, block-erase operations typically require one second. Therefore, these operations perform well as background tasks because the interrupt latency constitutes a small fraction of the total time.

The above discussion implies that the system interrupt should be disabled for data-writes and enabled for block-erases. What if an application requires simultaneous writes and erases? The Series 2 Flash Memory Card handles this situation with its Ready-Busy Mask Register (Figure 10). Setting the appropriate mask bits in the Ready-Busy Mask Register blocks the corresponding device's RY/$\overline{\text{BY}}$ signals. With a device's mask bit set, the card's RDY/BSY pin and Card Status Register (bit 0) always reflect a ready condition, regardless of the operation status. Figure 11 displays a conceptual mask circuit for a single device. The mask settings have no effect on the card's Ready-Busy-Status Registers (providing direct access to each device's RY/$\overline{\text{BY}}$ output) or the Device Status Register. This allows software polling in the usual manner.



Ready-Busy Mask Register
Performance Enhancement Register

CIS ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4124H	RESERVED				DEVICE 19	DEVICE 18	DEVICE 17	DEVICE 16
4122H	DEVICE 15	DEVICE 14	DEVICE 13	DEVICE 12	DEVICE 11	DEVICE 10	DEVICE 9	DEVICE 8
4120H	DEVICE 7	DEVICE 6	DEVICE 5	DEVICE 4	DEVICE 3	DEVICE 2	DEVICE 1	DEVICE 0

1 = MASKED

Figure 10. Allows Masking of Individual Device's Ready/Busy Signals

Selecting the Appropriate Device to Mask

Assume the register set DI:DX contains a 32-bit physical address into SERIES 2 card.
Each device pair represents 2 Megabytes (i.e. 200000H).

```
MOV CL, 5   ;Load shift count
SHR DI, CL  ;Result in DI is device pair number to mask.

;Now determine whether to mask device pair for word operations or use Bit 0 of
the DX portion to determine high or low device (odd or even) for byte
operations.
```

NOTE:
14. Polling the individual device's Status Register, the Ready/Busy Status Register, or the RDY/BSY bit in the Card Status Register.

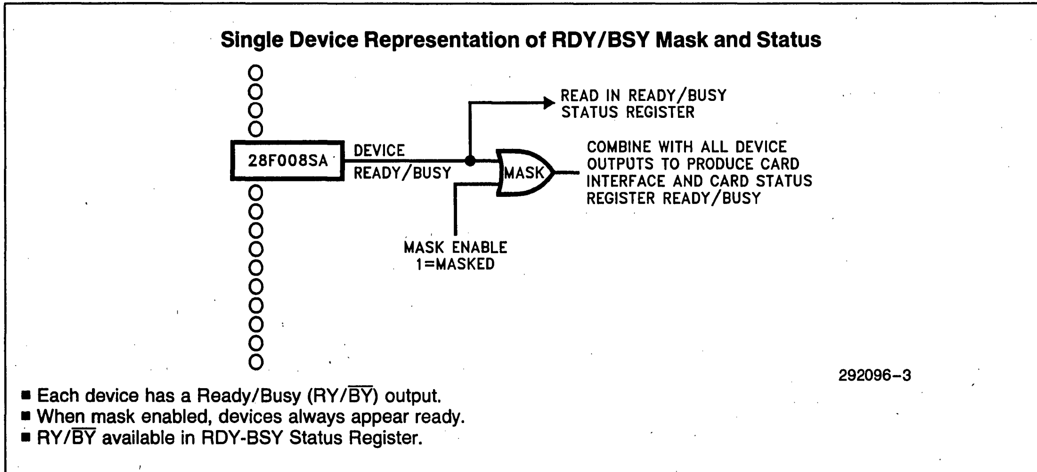


Figure 11. The Ready-Busy Mask is Very Useful for Write Optimization

READY-BUSY MODE REGISTER

Performance Enhancement Register

The PCMCIA specification for the Ready/Busy interface states that "*the RDY/$\overline{BSY}$ line is driven low by the memory card to indicate that the memory-card circuits are busy and unable to accept a data-transfer operation.*" Contrary to the PCMCIA specification, device-level data-write and block-erase automation enables the Series 2 Card to perform multiple operations simultaneously. Using the PCMCIA-specified method of RDY/$\overline{BSY}$ functionality for multiple device operations, the RDY/$\overline{BSY}$ interrupt does not notify the system until all devices finish because busy devices hold the RDY/$\overline{BSY}$ signal low, as shown in Figure 12. Multiple block erases (typical block erase time of 1 second) could present an unacceptable pushout if system software waits for the first available "clean" block to write data.

The Series 2 Card offers an alternative Ready/Busy mode (High-Performance Ready/Busy mode, alias "Levy"-mode) removing the performance impact of the PCMCIA mode. Circuitry internal to the ASIC catches every "READY-going" edge from each device. After an individual device becomes ready (Ready/Busy signal goes high), the system has the opportunity to immediately service the interrupt. System software must now toggle the CLEAR bit (bit 1) in the Ready-Busy Mode Register (Figure 14) to reactivate the Ready/Busy signal. Figure 13 demonstrates the resulting waveform.

The Series 2 Card powers up in the PCMCIA-mode. Switching into the High Performance mode requires a two step process, as shown in Figure 15. ASIC circuitry design prevents being able to write a zero to the RACK bit on the same cycle as entering the High-Performance RDY/BSY Mode. This intentional design technique eliminates the possibility of receiving a noise generated RDY/$\overline{BSY}$ rising edge, which would trigger an unwanted interrupt.

PCMCIA-Defined RDY/BSY Waveform for Multiple Device Operations

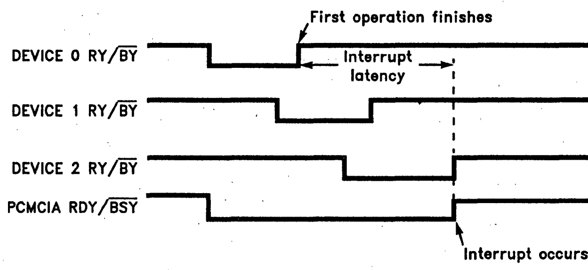
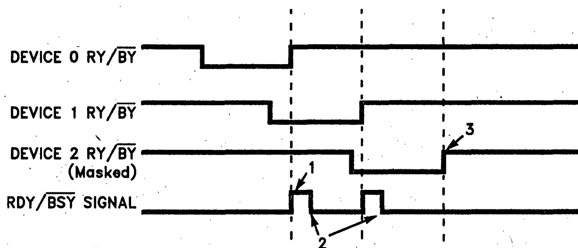


Figure 12. PCMCIA-Defined RDY/BSY Waveform for Multiple-Device Operations

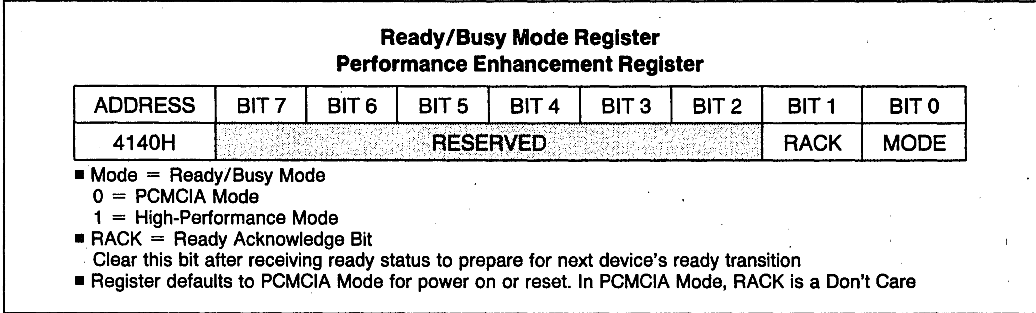
High-Performance RDY/BSY for Multiple Device Operations



NOTES:

1. Device 0 operation completes. RDY/$\overline{BSY}$ generates system interrupt. A masked RY/$\overline{BY}$ is zero. Unmasking simultaneously or after RY/$\overline{BY}$ going high, still enables a low-to-high transition on RDY/$\overline{BSY}$ to generate interrupt.
2. Software clears bit 1 of Ready-Busy Mode Register pulling RDY/$\overline{BSY}$ signal low.
3. Last device operation completes. Masked RDY/$\overline{BSY}$signal does not generate interrupt. Software must poll to detect operation completion of masked device(s).

Figure 13. High-Performance Mode Catches Each Device Going Ready



Ready/Busy Mode Register
Performance Enhancement Register

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4140H	RESERVED						RACK	MODE

- Mode = Ready/Busy Mode
  0 = PCMCIA Mode
  1 = High-Performance Mode
- RACK = Ready Acknowledge Bit
  Clear this bit after receiving ready status to prepare for next device's ready transition
- Register defaults to PCMCIA Mode for power on or reset. In PCMCIA Mode, RACK is a Don't Care

Figure 14. To Prevent Accidental Ready Transitions, a Three Step Sequence must be Followed to Enter High-Performance Mode

As discussed in the previous section, the block-erase operation benefits from the interrupt capabilities of the RDY/$\overline{BSY}$ signal. However, if your software only erases one device pair at any time, the PCMCIA-RDY/$\overline{BSY}$ Mode will be sufficient for two reasons: 1) Both devices started simultaneously will complete the erase operation almost at the same time; 2) in 16-bit access mode, both devices of the pair must be erased before writing.

To block-erase in multiple devices:

1. Be sure to mask all devices (in Ready/Busy Mask Register).
2. If not already done, place the Series 2 Card in the High-Performance Mode (refer to Figure 15).
3. Issue the block-erase command sequence to the appropriate devices.

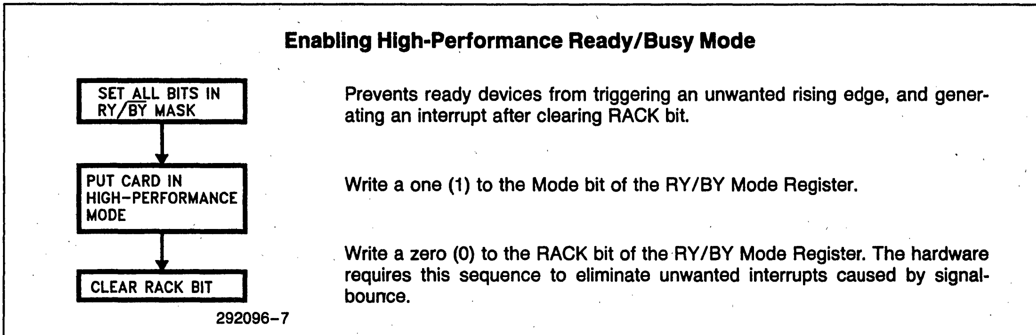


Figure 15. Entering High-Performance Mode

4. Unmask appropriate **Ready-Busy Mask Register** bits. The circuitry catches devices with already completed erase operations with the conceptual setup shown in Figure 13. Use a RAM-based variable or register for an erase-block queue to monitor erasing devices.

The interrupt service routine (ISR) can be as simple as removing the erase block from the queue. It could also be used to notify the system that this block is free to use. Regardless of the ISR implementation, it should include the following basic procedures:

1. Set all RY/BY masks in the **Ready-Busy Mask Register**. This prevents additional interrupts within the ISR (i.e. prevent re-entrant interrupt). Keep track of mask setup to reinstate before ISR exit.
2. Check the queue of erasing devices and read the Ready-Busy Status Register to determine which device completed the operation.
3. Service the erased block(s). Even though one erased block generated the interrupt, more blocks may have completed erasing at this point.
4. Clear RACK in the Ready-Busy Mode Register.
5. Before exiting the ISR, reset the mask. This "catches" devices that went ready during the ISR and will cause a re-entrant ISR. However, at this point in the ISR, this will not affect system or software integrity.

WRITE-PROTECTION REGISTER

The Series 2 Card contains a PCMCIA-defined, hardwired Card Information Structure (CIS) accessed in the Attribute Memory Plane. This data structure provides fundamental, unchanging information pertaining to the card. It includes card size, type of components, access speed, etc. Situations exist where the user needs to include custom-format information, such as card partitioning and operating system specific information.

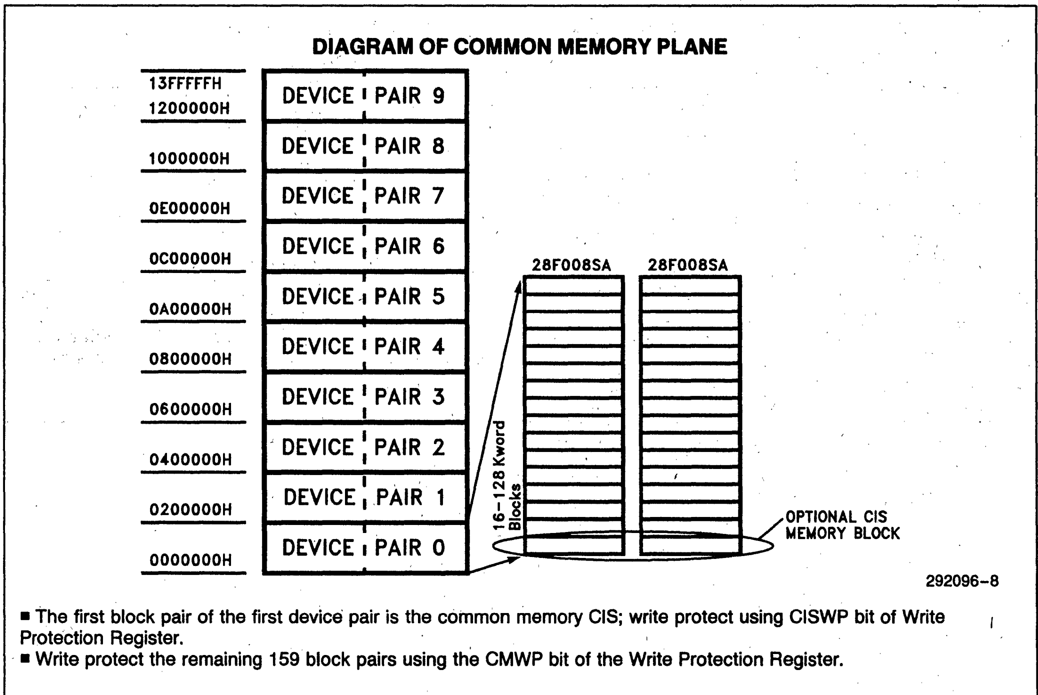
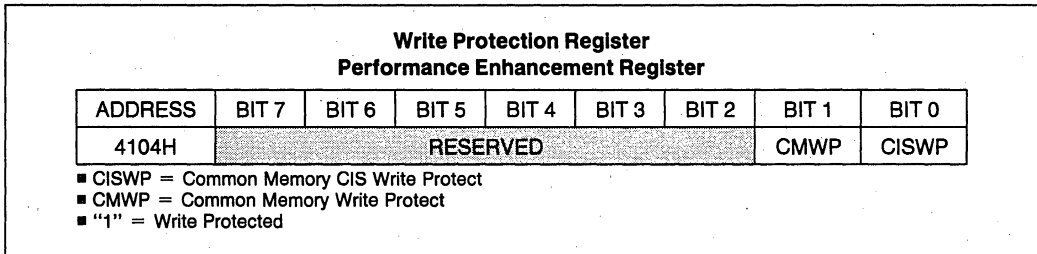


Figure 16. The WRITE PROTECT REGISTER Blocks Writes to the Two Sections of the Common Memory Plane

This information can be loaded in the Common Memory CIS during card format (refer to Figure 16). Typically, once this information is written, it would rarely change. The Series 2 Card provides a means of locking this area of memory, as well as the remainder of the Common Memory array with the **Write Protection Register** (Figure 17). The **Write Protection Register** has an advantage over the mechanical write protect switch in that it allows software to control user write access to the card's data (the mechanical switch can be easily switched off enabling card writes). For example, a pen-based system may use this feature to protect its read-only operating system stored within the Series 2 Card.

The CIS Write Protect Bit (CISWP, bit 0) prevents writes to the Common Memory CIS blocks. When software determines that this block of memory contains valid, *custom-format* information (contains PCMCIA tuple data structure), the CISWP Bit could be set to prevent accidental data corruption by another application. Note that if an End-User format utility is provided, this software must be careful not to destroy the custom format information which could be accessed if the CISWP Bit was deactivated. The Common Memory Write Protect Bit (CMWP, bit 1) prevents writes to the remainder of the Common Memory Plane (i.e. minus the Common Memory CIS blocks). To "software"-write-protect the *entire* Common Memory Plane, both bits must be set.



Write Protection Register
Performance Enhancement Register

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4104H	RESERVED						CMWP	CISWP

- CISWP = Common Memory CIS Write Protect
- CMWP = Common Memory Write Protect
- "1" = Write Protected

Figure 17. Provides a Software Implementation of the Write Protect Switch

CARD STATUS REGISTER

Performance Enhancement Register

This (Read-Only) register provides quick access to generalized conditions within the Series 2 Card (Figure 18). It provides a shorthand method for checking the following functions:

- Ready/Busy Status
- Ready/Busy Masking
- Deep-Sleep Modes
- Setting of Mechanical Write-Protect Switch
- Software Write Protect Status
- Soft Reset Status

Where the RY/$\overline{BY}$ Bit (bit 0, **Card Status Register**) displays the operation status of the cumulative devices within the card, the **Ready-Busy Status Registers** reflects the status of each individual device. Bit 0 (RDY/$\overline{BSY}$) mirrors the card's RDY/$\overline{BSY}$ (Ready/Busy) output pin, also reflecting any Ready/Busy masking conditions. Two circumstances warrant the use of this bit: 1) When the hardware interrupt triggered by the RDY/$\overline{BSY}$ pin produces an unacceptably long latency period, this bit should be software polled instead to increase performance; 2) When multiple devices have data-write/block-erase operations in progress, reading this cumulative Ready/Busy status will be quicker than reading multiple status registers within each device. However, when the application requires immediate access to each device as it finishes an operation, individual Device Status Registers or the card's **Ready-Busy Status Register** must be used.

Card Status Register
Performance Enhancement Register

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4100H	ADM	ADS	SRESET	CMWP	PWRDWN	CISWP	WP	RDY/$\overline{BSY}$

- RDY/$\overline{BSY}$ = Reflects PCMCIA interface RDY/BSY pin, 0 = busy
- WP = Mechanical Write Protect Switch, 0 = off
- CISWP = Common Memory CIS Write Protect, 0 = off
- PWRDWN = Powerdown Reflects PWRDWN in Global Power Down Reg, 1 = Power Off
- CMWP = Common Mode Write Protect, 0 = Off
- SRESET = Soft Reset Reflects SRESET in SOFT RESET Reg, 1 = Soft Reset
- ADS = Any Device Pair Powered Down, OR'd Condition of Sleep Control Reg, 1 = Power Off
- ADM = Any Device Masked, OR'd Condition of Ready-Busy Mask Reg, 1 = Masked

Figure 18. Provides Generalized Card and Device Information

Bit 1 reflects the card's mechanical switch position (1 = Write Protected). This switch disables any writes to the card. Two software strategies can be implemented for this bit: 1) Assume the card's Write-Protect switch is off. Attempt to write to the card and only check the Write-Protect status if the data-write fails (which it will if the switch is on); 2) Check the switch first to avoid the possibility of failing a data-write. The choice depends on the application. For a solid-state disk continuously updating files, the former is more appropriate because the Write-Protect switch will probably be off.

Bits 2 (CISWP = Common Memory CIS) and 4 (CMWP = Common Memory Write Protect) are direct (Read Only) inputs from the **Write-Protect Register**. These bits should be checked in a manner similar to that for Bit 1 (WP). For more detail refer to the **Write-Protection Register** section.

The PwrDwn Bit (bit 3) provides a (Read Only) version of the PwrDwn Bit in the **Global Powerdown Register** (1 = PwrDwn). Only the Attribute Memory Plane is available with the Powerdown feature enabled, allowing access to the **Component Management Registers**.

The SRESET Bit (bit 5) provides a (Read-Only) version of the SRESET Bit in the **Soft Reset Register** (1 = Locked in soft reset state).

SUMMARY

The Series 2 Flash Memory Card delivers the hardware capabilities required for implementing a solid-state storage device. Software engineers will find the features of this card both flexible and powerful when coupled with flash-optimized filing systems, such as Flash File System from Microsoft. This application note has discussed the various methods of using the **Component Management Registers** to facilitate designs incorporating the SERIES 2 card.

- PCMCIA-Defined Registers provide generalized assistance for memory card interfacing.
- Performance Enhancement Registers boost software control over the card's internal flash memory devices.

GLOSSARY

Attribute Plane: Memory plane within the card selected by pulling the $\overline{REG}$ pin low. This random access memory contains the CIS and Component Management Registers.

Block-Erase: Erasing sections of a single flash memory device.

Bulk-Erase: Erasing the entire flash device simultaneously.

Common Memory: The memory card's main memory array.

Common Memory-Card Information Structure: The first block pair of the first device pair. Useful for storing custom format information, such as partitioning of the card.

Component Management Registers (CMR): Memory-mapped I/O registers used to control device-level functions.

Deep-Sleep Mode: A special very low power mode useful for saving power when not accessing the flash memory components.

Device-Pair: Arrangement of the 8-bit 28F008SA devices in the SERIES 2 card in a word-wide manner.

ExCA: System Implementation (hardware and software) of PCMCIA R2.0.

Hardwired Card Information Structure (CIS): Embedded into the Attribute Memory Plane to describe non-changing information about the SERIES 2 Card (i.e. density, speed).

Levy Mode: alias for the High-Performance mode for Ready/Busy notification.

Personal Computer Memory Card International Association (PCMCIA): The organization formed to promote interchangeability of IC cards by providing a standardized mechanical, electrical and metaformat interface.

Performance Enhancement Registers: Memory-Mapped I/O registers included by Intel in the Series 2 Card to boost performance by providing software control of the internal 28F008SA functions.

Ready/Busy: Indicator used to determine when a data-write or block-erase operation has completed. Symbolized by RY/$\overline{BY}$ for the 28F008SA and RSY/$\overline{BSY}$ at the Series 2 Card interface.

Status Register: A register internal to a 28F008SA FlashFile™ Memory device used to determine write and erase operation status.

RELATED DOCUMENTS

28F008SA, 8 Megabit, FlashFile™ Memory Data Sheet
Series 2 Flash Memory Card Data Sheet
82365SL, PC Card Interface Controller Data Sheet
PCMCIA PC Card Standard Release 2.0
Exchangeable Card Architecture Specification

4

RELIABILITY REPORT

RR-70

August 1992

Flash Memory Card Quality/Reliability Data Summary

Order Number: 293007-002

Flash Memory Card Quality/Reliability Data Summary

CONTENTS PAGE

INTRODUCTION 4-286

QUALITY ASSESSMENT 4-286

RELIABILITY STRESS 4-286

MECHANICAL STRESSES 4-289

CONTAMINATION 4-289

CARD CARE SUGGESTIONS 4-289

DATA SUMMARY 4-290

CONTENTS PAGE

APPENDIX A 4-291

External Dimensions 4-292

Pin Configurations 4-292

Pin Descriptions 4-293

Functional Block Diagram 4-294

4

INTRODUCTION/SCOPE

Quality and Reliability: The Cornerstone of Business

In the world of electronic hardware, no facet is more important to the user of a system than the reliability of its individual components and subsystems. Being the leader in FLASH, we are proud of the continuous quality and reliability leadership position in the marketplace we have maintained.

In the spirit of service to our customers and their customers, this publication has been assembled for your convenience and reference. The scope of this document is limited to Intel's latest Flash Card Products consisting of the 1/2/4 Meg densities. The FLASH device specific product/process reliability summaries can be found in RR-60. The data provided herein is the product of just one of Intel's qualification and reliability monitoring systems. The purpose of this report is to supplement Intel's Quality and Reliability handbook with product specific data. For additional information, please contact your Field Sales or Customer Quality Engineer.

Quality versus Reliability

The traditional concepts segregating quality from reliability is one of time. Quality is a measure of the ability of a product to meet performance expectations at a single point in time. This "point in time" is usually interpreted as your initial board power-up or incoming inspection. Reliability, on the other hand, is a measure of a product's ability to maintain its "time zero" quality throughout its life cycle. A reliability failure usually occurs after your product has shipped to your customer.

The cost of poor quality can be objectively totalled within your organization. It includes the cost of detection and in-house repair. However, the cost of poor reliability has a much higher cost. Besides an inherently higher repair cost per defective unit, reliability failures create customer concern about design and/or workmanship standards used in the manufacture of the product. Loss of goodwill with your customers can have many long term negative effects on your business. Therefore, Intel advocates that you make reliability a key consideration for the selection of your system's components.

The Roots of Reliability

The manufacture of a reliable VLSI semiconductor device using a modern technology is a dynamic and evolutionary process. Success of this process is highly dependent upon the interplay between knowledgeable and experienced manufacturing engineers, materials physicists, and responsible/responsive management. Only the correct combination can consistently deliver high volumes of reliable product. In this model, the experienced process engineer selects and defines the stresses to be performed and the performance criteria to be met, utilizing appropriate statistical tools and limits. The materials physicist then determines the root causes of failure, if and when failure occurs, and provides effective solutions and or containment recommendations. Finally, management provides the resources for the entire process from initial monitor to root cause corrective action.

MONITORS: THE CONTROL MECHANISM OF RELIABILITY

A Comprehensive Program is the Key

Intel has developed and implemented many types of reliability monitoring systems. Since continuous delivery of reliable product is of paramount importance, most of the monitors are in-line and are designed to provide as close to "real time" feedback on the reliability of the product in-process as possible. The monitors are located throughout the fab, assembly and test areas. The data from these monitors are an indication of process health and overall statistical control. They are not necessarily directly correlatable to the reliability of the product that will ship to your location. For this reason, a final finished product monitor which randomly selects product is used as the yardstick to measure the success of our factory in meeting your customer's reliability goals.

QUALITY ASSESSMENT

All new cards undergo a full temperature range quality test assessment. This involves a specific sample size of product to be 100% temperature range tested. The quality goal to be met is consistent with the Intel Corporate stated quality goals.

RELIABILITY STRESS

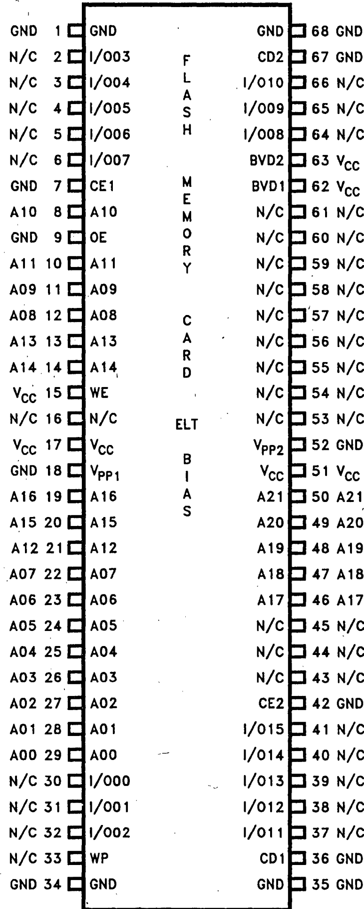
ELT Extended Life Test

ELT is performed on programmed cards. V_{CC} = 5.25V and the stress temperature is held to a maximum of 85°C due to card and connector thermal considerations. It is functionally exercised in a sequential addressing pattern and outputs are exercised but not loaded. See Figure 1 for the typical bias and timing conditions. The end point electrical tests are conducted within a fixed period of time to worst case data sheet parameters. The Memory Components Division also periodically takes variables data on selected data sheet parameters to monitor the stability of the process.

ELT Configuration

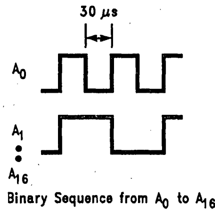
Device: Memory Card
Stepping:
Description: 1 Meg, 2 Meg, and 4 Meg
Technology: CHMOS III

Configuration Type: ELT
Temperature: 85°C
Package: Memory Card



293007–2

Option Pins: All option pins parallel to each other, programmable to V_{CC}



293007–1

Figure 1. Typical ELT Bias and Timing Sequence

ESD

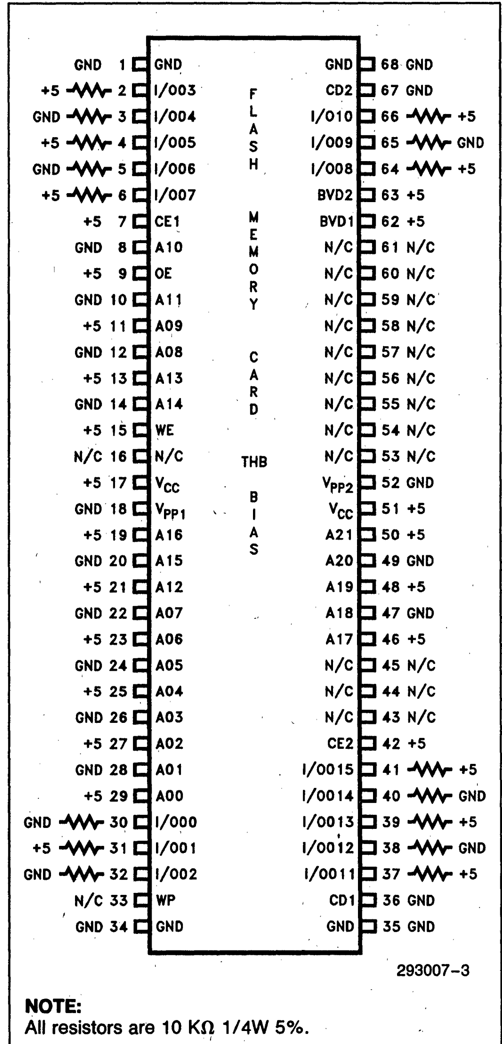
ESD Testing—This test is performed to validate the products tolerance to Electro Static Discharge damage. All products incorporate ESD protection networks on appropriate pins. Three types of ESD testing were performed. Mil Spec to ±2K, charged device to ±1.5 KV and Case Zapping to ±2 KV. The cards contain internal spring connections to ensure the case lids are grounded.

Temperature Cycling (T/C)

Temperature cycling evaluates not only the card's mechanical ability to remain a closed memory card system but further evaluates the mechanical solder joint integrity of card assembly process. 1K cycles of −40°C to +85°C cycling are required with 20 minutes/cycle. Electrical endpoints are tested across the data sheet temperature range.

Moisture Resistance (85°C/85% RH)

During the 85/85 test the cards are subjected to a high temperature, high humidity environment. The objective is to accelerate corrosion failure mechanisms thru an electrolytic process. This is accomplished through a combination of moisture penetration of the plastic, voltage potentials and contamination which, if present, would combine with the moisture to act as an electrolyte. See Figure 2 for typical 85/85 Bias Diagram.



NOTE:
All resistors are 10 KΩ 1/4W 5%.

Figure 2. Typical 85/85 Baising

MECHANICAL STRESS

Vibration

Test is done to evaluate memory cold surface mount (SM) reliability. Test condition: SSH2 for 2 hours.

Drop Test

This test is to simulate the conditions a flash memory card could be subjected to if inadvertently dropped. The test consists of 10 repeated drops from an elevation of 4 feet to a tile floor in the X, Y and Z orientation. Lid popping or frame/card damage are not allowed. Testing is followed by visual examination and ambient electrical/function testing.

Pressure/Crush Test

This test is to simulate the effect of a crushing force applied to the card. This condition could be as a result of inadvertently sitting on the card while in a persons pocket. The test consists of a 100 kg load for a duration of 40 seconds pressed thru a rubber pad on top or bottom surface of the card. Stressing is followed by a ambient electrical/functional test.

Socketing Test

This test verifies the connectors life expectancy of the card. It simulates repeated insertion/removal into a system. It is 10k insertions duration.

Switch Test

The flash memory card incorporates a write protect slide switch. The switch is cycled for endurance to 1k cycles.

Bend/Flex Test

This test is to simulate the effect inadvertent flexure or bending of the card. It consists of five consecutive scan twists in both the X–Z and Y–Z directions. Ambient electrical/function testing is performed to verify the card.

External/Internal Dimensions

External case and connector dimensions are verified to specification. Internal examinations are performed for device alignment and solder joint quality.

CONTAMINATION

Ionograph

Internal PC Band components are evaluated using an ionograph for ionic content. PC B without cases/frames are emersed in the ionograph fluid and ionic contamination levels are measured. The cards are required to contain less than 15 μgm/sq. in. NaClor equivalent.

Visual Flux Residue

Memory cards are visually examined at 10x for SMT process flux residues. This examination gives an indication of the throughness of the PCB cleaning step after IR reflow processing.

CARD CARE SUGGESTIONS

Several simple precautions should be observed when using/storing your flash memory card.

Connector Opening

Keep this area free of dirt, food and moisture. The connector holes could become clogged/obstructed and possible damage to the main system male connector could result.

Storage

When possible keep the card in a cool dry place. Laying the card in bright, hot sunlight could cause the card to warp making system insertion difficult. Store the card in its protective sleeve when not in use. This will help keep the connector clean.

Never sit on, drop or emerse the card in water.

4

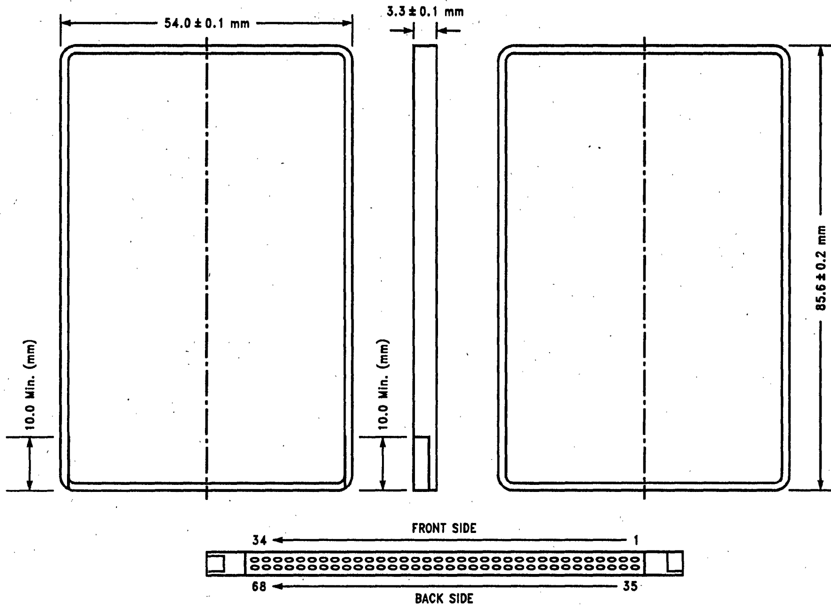
QUALIFICATION SUMMARY 1, 2 & 4 MB FLASH MEMORY CARD

Test	Condition	Level III
ELECTRICAL		
Temp QV	−10°C/+70°C, 5V±10%, <500 DPM @ 60%	NR
ELT	85°C, 1K hours, 5.25V	NR
ESD	Mil Spec ±2 KV	0/5
	Charged Device ±1.5 KV	0/5
	Case Zap ±2 KV	0/5
ENVIRONMENTAL		
T/C	−40°C to +85°C, 1K Cycles	0/66
85/85	1K Hours at Alternate Pin Bias	0/66
MECHANICAL		
Vibration	55 Hz, 2 Hour	0/10
Drop	4′ to Tile Floor X, Y and Z, 10X	0/10
Pressure	100 kg Load, 40s thru 2 mm Rubber Sheet	0/10
Socketing	10K Insertions	0/10
Switch	1K Cycles	0/10
Bend	0.5 cm Twist, 5X	0/10
Connector	Dimensions per Spec	0/10
External	Dimensions per Spec	0/10
Internal	Visual Solder Joint—Device Alignment	0/10
CONTAMINATION		
Ionograph	<15 μgm/Sq. In. MaCl Equivalent	0/10
Visual	Flux Residue	0/10
ASIC		
ASIC	Failure Rate (%1k Hrs)	0.0018
E/F 28F010 TSOP		
T/C	Condition C 2K Cycles	0/254
T/S	Condition C 1K Cycles	0/233
85/85	85°C/85% RH 2K Hours	0/382
Steam	121°C 2ATM 336 Hours	0/446
ELT	125°C 1K Hours	0/284

NOTES:
All samples equally divided from 3 manufacturing lots.
NR = Not Required if same assembly factory
* = Fully populated card

APPENDIX A

External Dimensions



293007-4

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	GND	18	V_{PP1}	35	GND	52	V_{PP2}
2	D_3	19	A_{16}	36	$\overline{CD}_1$	53	NC
3	D_4	20	A_{15}	37	D_{11}	54	NC
4	D_5	21	A_{12}	38	D_{12}	55	NC
5	D_6	22	A_7	39	D_{13}	56	NC
6	D_7	23	A_6	40	D_{14}	57	NC
7	$\overline{CE}_1$	24	A_5	41	D_{15}	58	NC
8	A_{10}	25	A_4	42	$\overline{CE}_2$	59	NC
9	$\overline{OE}$	26	A_3	43	NC	60	NC
10	A_{11}	27	A_2	44	NC	61	$\overline{REG}$[1]
11	A_9	28	A_1	45	NC	62	$\overline{BVD}_2$[2]
12	A_8	29	A_0	46	A_{17}	63	$\overline{BVD}_1$[2]
13	A_{13}	30	D_0	47	A_{18}	64	D_8
14	A_{14}	31	D_1	48	A_{19}	65	D_9
15	$\overline{WE}$	32	D_2	49	A_{20}	66	D_{10}
16	NC	33	WP	50	A_{21}	67	$\overline{CD}_2$
17	V_{CC}	34	GND	51	V_{CC}	68	GND

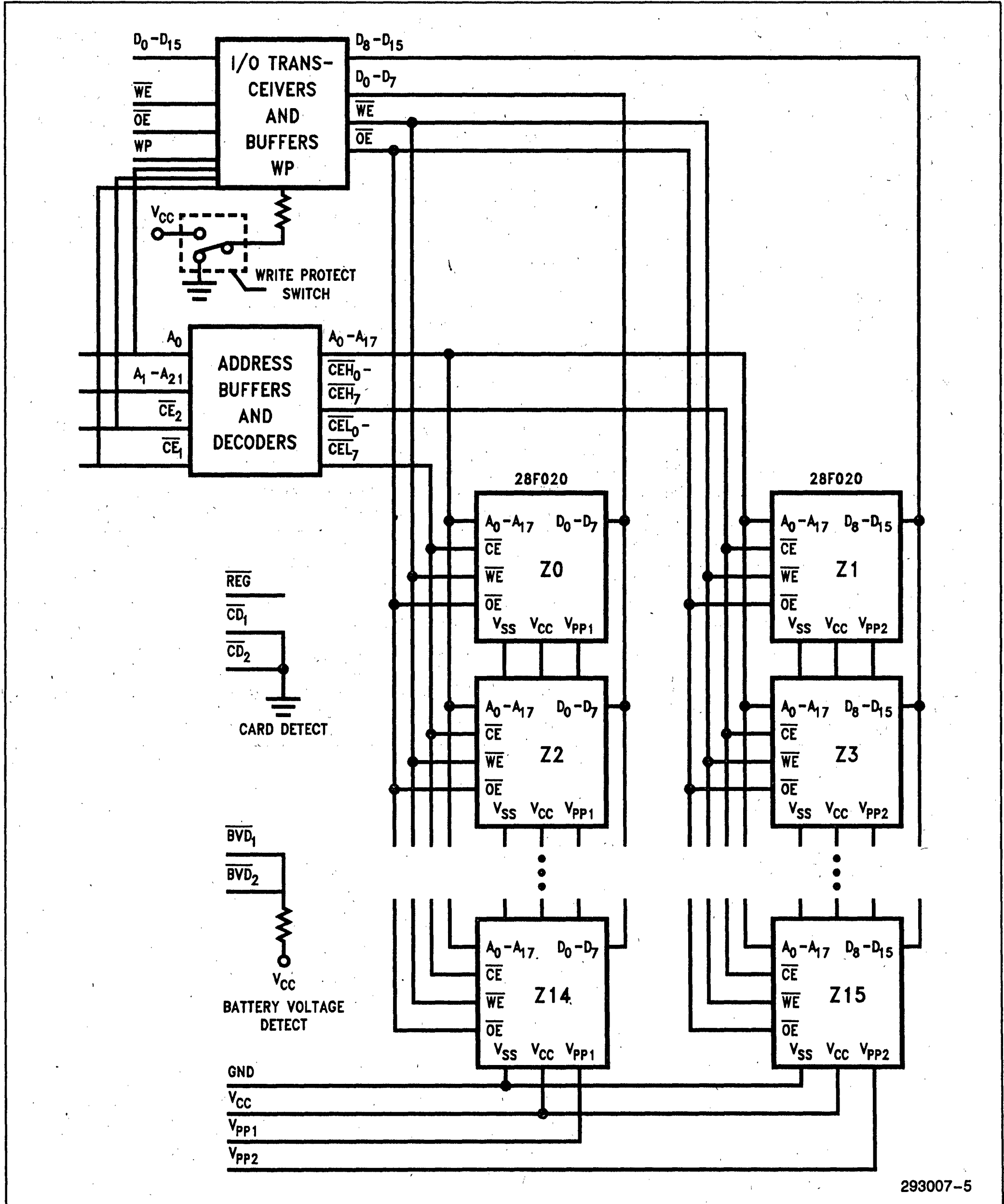
NOTES:
1. $\overline{REG}$ = Register Memory Select = No Connect (NC), unused. When $\overline{REG}$ is brought low, PCMCIA/JEIDA standard card information structure data is expected. This is accomplished by formatting the card with this data.
2. $\overline{BVD}$ = Battery Detect Voltage = No Connect (NC), unused.

Pin Configurations

Pin Description

Symbol	Type	Name and Function
A_0–A_{21}	INPUT	**ADDRESS INPUTS** for memory locations. Addresses are internally latched during a write cycle.
D_0–D_{15}	INPUT/ OUTPUT	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the card is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}_1$, $\overline{CE}_2$	INPUT	**CARD ENABLE:** Activates the card's high and low byte control logic, input buffers, zone decoders, and associated memory devices. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory card and reduces power consumption to standby levels.
$\overline{OE}$	INPUT	**OUTPUT ENABLE:** Gates the cards output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	INPUT	**WRITE ENABLE:** Controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **NOTE:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP1}, V_{PP2}		**ERASE/WRITE POWER SUPPLY** for writing the command register, erasing the entire array, or writing bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY:** (5V ± 5%)
GND		**GROUND**
$\overline{CD}_1$, $\overline{CD}_2$	OUTPUT	**CARD DETECT:** The card is detected at $\overline{CD}_{1,2}$ = Ground.
WP	OUTPUT	**WRITE PROTECT:** All write operations are disabled with WP = active high.
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating.
$\overline{BVD}_1$, $\overline{BVD}_2$	OUTPUT	**BATTERY VOLTAGE DETECT. NOT REQUIRED.**

Block Diagram



May 1990

4

PC Standard in the Cards

BY TOM WOLFE

© 1990 Intel Corporation, 1990
Reprinted with permission from *EETimes*, May 1990

Order Number 295048-001

AGREEMENT AT HAND FOR IC-BASED STORAGE MEDIUM

PC standard in the cards

BY DAVID LAMMERS

Tokyo — Agreement could be imminent on a Japan–U.S. standard for the "PC Card," a 2 × 3-inch IC-based card to be used as a removable data-storage medium for portable computers.

Expectations are high that this transPacific standard will do for notebook and laptop computers what MS-DOS, the floppy disk and the IBM PC did for desktop machines: allow software to be sold for, and data to be exchanged over, a medium compatible across a broad range of portables from a long list of manufacturers worldwide.

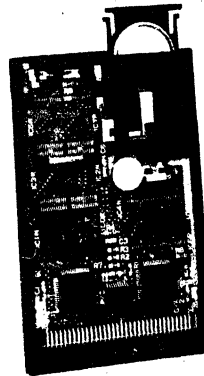
With notebook and low-end laptops expected to constitute half of Japan's PC market by 1994—and perhaps a third of PC sales around the world by then—approval of the standard is especially important to U.S. computer and software companies. Proponents of the PC Card concept hope that, with the standard approved, software vendors will quickly begin porting applications to the cards and users here and in Japan will embrace the new technology.

The PC Card standard is being forged by the Personal Computer Memory Card International Association (PCMCIA) and the MITI-affiliated Japan Electronics Industry Development Association (JEIDA), which includes about 40 major Japanese companies. The 70-member PCMCIA includes nearly all of the personal computer industry's movers and shakers, with IBM, Lotus Development Corp. and Microsoft Corp. playing particularly active roles.

Today and tomorrow in Seattle, Microsoft will host the May meeting of the PCMCIA, at which members are expected to approve a draft agreed to in Tokyo on May 10 by PCMCIA members and the memory card working group of JEIDA. The agreement specifies the JEIDA V. 4.0 format, 68-pin card; the DOS file format; a means for the system to know what kind of card it is dealing with; and other hardware and system-software specifications.

It's expected that Poqet Computer's (Sunnyvale, Calif.) Poqet PC, a palmtop unit that accepts the IC cards, will spearhead penetration of the U.S. market.

Dan Sternglass, founder of Databook Inc. (Ithaca, N.Y.), which manufactures a series of IC-card reader/writers and programmers, said: "What's going to drive the market first are portable systems, starting with the Poqet. We still have to see how much of the market will be penetrated by the handheld-type computers. Then,



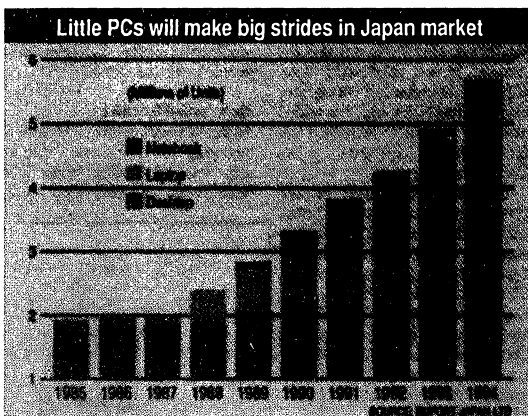
Fujitsu's version of the credit-card-size 'PC Card.'

PC Card standard drafted

if everyone owns a handheld, IC cards might be used in desktops."

A host of notebook machines coming to market in Japan is expected to fuel use of the new cards there.

Last week at the Japan Business Show, NEC Corp., Fujitsu Ltd. and Mitsubishi Electric Corp. all introduced powerful new notebook computers that include IC reader/writers compatible with the new 68-pin standard. Fujitsu offered a half dozen applications in ROM-based IC card format, along with various data file cards using SRAMs.



New notebook wave

Those A-4-sized systems are in the 6-lb. (2.7 kg) range, similar in size to the popular "Dynabook" from Toshiba. One model of NEC's PC98 Note is also the first built around Intel's 386SX processor, partly because it expects that users of notebook computers will want to run the same Windows interface they use on their desktops.

Though the Dynabook does not include an IC card slot, future Toshiba systems will. Both the chairman of the JEIDA working group and the software subcommittee are Toshiba executives.

Though several companies are developing notebook machines in the United States, the portable field here is currently focused on the larger, heavier laptop PCs, like those made by Zenith Data Systems (now part of the Bull Group), Compaq and Tandy.

But that could change. According to Japanese sources, IBM Corp. is expected late this year to introduce a notebook computer, now under development at IBM Japan, that would use IC cards manufactured at a new IBM plant in Toronto. By using the PC Card, IBM might try to leapfrog its competitors and make a comeback in portables, just as it's trying to do in workstations. The June meeting of the PCMCIA will be hosted by IBM in Toronto.

For now, hopes for the PC Card's success in the U.S. market rest mainly with the Poqet computer. Poqet is pioneering IC card use with versions of Lotus 1-2-3, an integrated package called AlphaWorks from Alpha Software (Burlington, Mass.), and other applications. The system it now uses is upward-compatible with the new standard.

John Reimer, the Fujitsu Microelectronics (San Jose, Calif.) memory card manager who serves as the PCMCIA's chairman, said he got interested in IC cards because of Fujitsu's investment in Poqet Computer. Fujitsu is doing back-up manufacturing for Poqet in Japan and is a card supplier to Poqet.

Reimer said he expects the success of Poqet's $2,000 system to drive demand for IC cards in the United States for the time being. But, he said, ultimately, "every executive will want to have some kind of notebook computer," and that will create the market for IC cards.

Not an expansion card

The PC Card should not be confused with the memory expansion cards that some vendors offer for adding DRAM to laptops. Partly to avoid confusion with these DRAM cards, PCMCIA uses the name "PC Card" and has developed a logo that will mark the cards that comply to the standard.

PC Cards, rather than being analogous to add-on memory, are actually a form of removable media, like the 3½-inch diskettes being used in today's laptops. Like floppies, they

How standard came about

Tokyo — The people who hammered out the IC card standard between Japan and the United States described it as an exercise in quick compromises—and a demonstration that good will exists in abundance between Japan and America.

Basically, the standard took most of the hardware specifications developed over the past five years by the Japanese and added software standards prompted by the U.S.'s Personal Computer Memory Card International Association (PCMCIA).

Fujitsu Microelectronics memory card manager John Reimer said Poqet Computer executives realized a year ago that a standard for the cards would expand the software base for their palmtop machine. Reimer—described as "the driving force" behind the formation of the PCMCIA—sent out letters in June 1989 about the new association and got quick acceptance from U.S. companies. About 70 companies joined PCMCIA.

Reimer: Instigator.

Late last year, the Americans sent a letter to the Japan Electronic Industry Development Association (JEIDA), an association that focuses on personal computers. The JEIDA working committee, already five years in existence, sent 10 Japanese representatives to the PCMCIA's January meeting, in Dallas.

Japanese and American executives began crossing the ocean each month, attending each other's meetings. The Americans, accustomed to voting on issues after a period of discussion, worried that the Japanese would "want to keep talking, talking, talking, until they reached a consensus," Reimer said.

Instead, the Japanese accepted U.S. proposals about the pin lengths for the 68-pin connector; Japanese software companies—including Microsoft Japan, Just Systems and Ascii Corp.—provided important input to the software discussions.

—*David Lammers*

4

Japan/U.S. PC Card standard at hand

not only store programs and data but will allow dissimilar notebook machines to share that data, thanks to the standard. Further, by using PC Cards, notebook computers could exchange data with pocket computers or even with new versions of the electronic organizers which have sold millions in Japan, but have been based to date on proprietary card schemes.

IC cards are seen as the key to eventually replacing floppy disk drives in notebook computers. Ryozo Yamashita, an ASCII Corp. (Tokyo) engineering vice president, attended several PCMCIA meetings in the United States and said he grew tired of carrying the six-pound Dynabook along in his rucksack.

"With a floppy disk drive in the computer, there is not much more than can be done to reduce the weight. And a floppy drive consumes a lot of power."

But before the ubiquitous floppy is designed out of notebook computers, software companies must port more software to PC Cards. Yamashita said he believes the market will be driven first by the Poqet computer (though he believes the Poqet keyboard needs improvement) and later this year by less expensive portable computers.

"By the end of this year the IC memory card will be used as the primary media on pocket-type computers from many companies," Yamashita said, with the fall Comdex show in November a target date. With the PC Card, he noted, the system CPU can directly access the memory on the card itself. "That will eliminate the huge memory needed on the main unit," he said.

Once a large installed base of hardware is on the market, more software will be ported to PC Cards, he said. However, software companies are cautious because of the high cost of putting software into silicon. A 1-Mbyte ROM card that costs $50-70 now may come down to half that over the next year, as 8- and 16-Mbit mask ROM chips proliferate. But compared with distributing applications on floppies, ROM IC cards are a big financial risk, especially for the thousands of small software companies.

Japan's software houses, including ASCII, have a lot of experience selling Nintendo game software stored in ROM, Yamashita said. But Nintendo software can sell in millions of units, while the computer market is marked by higher prices but smaller unit volumes. The big merit of IC card software, he said, is that it cannot be copied by individuals, giving software companies the incentive to strive for potentially higher margins.

One other potential hurdle for getting software into PC Cards could be settled at this week's PCMCIA meeting. There, Microsoft and Lotus Development Corp., two of the biggest promoters of the standard, will try to work out their differences on how to implement "execute in place" (XIP). XIP permits a small system to run software stored on a PC Card and access memory on the same card, rather than relying solely on the system's memory. XIP is an important issue for the optimal execution of large programs adapted to IC cards, such as Lotus 1-2-3 running on the Poqet system. The issue brings to the IC card level a bigger issue: how to get around the 640-kbyte barrier of the original PC architecture while maintaining PC compatibility, said Yoshinobu Akimoto, an engineer at Microsoft Japan.

Dynabook engineer Terry Moore (left), key figure in standards development, with president Dan Sternglass and PC Card.

Mike Dreyfoos, chief engineer of Microsoft's MS-DOS division, who is active in the PCMCIA, and Jim Prelack, a Lotus Development executive who serves as president of PCMCIA, are both said to be taking a "market oriented" approach toward resolving the snag.

An informal meeting on the XIP issue, held here May 14, resulted in some progress, sources here said. Even without an immediate agreement on XIP, companies can take the basic standard and begin porting software and building IC card-based hardware.

"Both companies [Lotus and Microsoft] realize we've got to get the show on the road," said Fujitsu's Reimer.

U.S. to get the jump

T. Shigeta, a senior staff manager at Microsoft Japan, said the U.S. market may adopt the software cards faster than the Japan market. In Japan, Fujitsu, NEC and other companies all support proprietary versions of MS-DOS, making applications incompatible. That fracture is continuing down to the notebook and palmtop systems, which will support proprietary versions of MS-DOS.

"The big issue is not only the high cost of the [IC card-based] software, but having to support different cards for the different architectures here in Japan," Shigeta said.

He believes the data cards will sell well in Japan. "The importance of this standard is that notebook-, laptop- and desktop-type computers will be able to exchange data on the cards."

He predicted small ISVs will maintain a cautious stance toward IC card-based software. The lack of software support has hurt previous attempts to market IC card-only notebook computers, including NEC's "UltraLite," sold in the U.S. market, Epson's "Note Executive" and Sharp's "Brain."

Asked if Microsoft will port its applications to the cards, Shigeta said, "I can't say anything explicit, but from the level of our activity in JEIDA and PCMCIA, you can see that we see a bright future in IC memory cards."

Ryosuke Takahashi heads up the five-person IC card team at DuPont Japan Ltd. As a neutral player in between Japan's competing electronics companies, DuPont buys memories, has them assembled by third-party suppliers and markets the IC cards to Japan's computer makers. DuPont also supplies most of the two-piece (header and female) connectors used in the JEIDA format cards.

A 1988 market study done by DuPont and Nomura Research Institute predicted that the IC card market in Japan would grow by a 33 percent compound average growth rate, rising to about $1 billion in 1995. That's about five times larger than the total expected for 1990, and the Japan market estimate preceded the unexpected joint standard with the U.S.

In about two years, when flash EPROM-based cards are in wider use, the price of most of the cards will drop to half, Takahashi said. Now, a 512-kbyte SRAM card is sold to OEMs for about 40,000 yen, or about $240.

Takahashi believes that palmtop-size computers will be the biggest market for the next couple of years, with most notebook computers continuing to use floppy disk drives. Beyond that, some companies may migrate to IC card-based notebooks, sans floppy drive.

But other markets will be important. Already, robots and measurement equipment, laser printers, and medical equipment use IC cards. "My personal view is the digital still cameras will be a big market for IC cards in years to come, replacing film," Takahashi said. Toshiba and Fuji Film already have a camera on the market that uses IC cards, and Sony may change from a floppy to a card based still camera. The FBI put in a major order for Sony's camera last week.

ASCII's Yamashita said a potentially huge market for IC cards is in distribution of specialized information. A number of Japanese software, printing and publishing companies have initiated the International Card Media Publishing Association. Stock exchange data and financial news, train timetables and other forms of changeable data could be stored on IC cards. One idea is to provide vending machines that would download data on to a card at a train kiosk or newsstand.

"When flash memory gets cheap enough, then you might stop by and download certain kinds of news and view it on the train; information could be personalized," Yamashita said.

The way to look at IC cards is as the next step in the evolution of computer media, from paper tape to magnetic tape and floppy/hard/optical disks, and now to a silicon-based media, he said.

Mask ROMs normally are used to store software in IC cards. The market for Nintendo game cartridges has helped drive the price of a 1-Mbyte ROM card down to about $60 to $70. That may drop by half over the next year.

Before IC cards become popular the cost of data storage cards must come down, an area where flash EPROMs are expected to play a key role.

"All of the PCMCIA members expect that flash will replace a good chunk of the SRAM-based cards," said Reimer, noting that Intel Corp., Texas Instruments, Inc. and Toshiba Corp.—the larger companies in the flash memory field—are active members of PCMCIA. Fujitsu and other Japanese companies have major flash development efforts under way.

William Howe, president of Intel Japan, said IC-card related product announcements from Intel, based on flash EPROMs, "are not very far away." Though he said Japanese semiconductor companies have accelerated their own flash development efforts, they are turning to Intel for flash EPROMs to be built into IC cards.

Takemae said flash memories will make an impact on IC card pricing, probably beginning next year.

"In the last few months the interest from our customers in flash (for use in IC cards) has increased by an order of magnitude. It's not just Company A or Company B, it's everybody," Howe said.

Though flash is considerably more expensive than EPROM memory now, Howe said he expects the price to come down to 10 to 15 percent above the tags on EPROMs, and far less than the price of SRAMs now used in IC data cards.

Yoshihiro Takemae, a Fujitsu Ltd. semiconductor manager who served as chairman of the JEIDA hardware subcommittee, said flash memories will make an impact on IC card pricing, probably beginning next year. The PC Card pin layout scheme reserves pin No. 18 for programming the card, which would accommodate the 12 volts needed to electrically rewrite a flash memory.

While research continues into ways in which sectors of a flash EPROM can be selectively erased, Takemae said, "I think that deletions can better be handled by the software. People should think about the format and handling of the IC memory card just as they think about floppy disks now."

The hardware specifications included a write-protect switch, the position of the battery, a green-yellow-red light system to indicate the strength of the battery, and a variety of electrical specifications, all of which can be obtained from the PCMCIA once the standard is published.

The decision to move from 60-pin cards, which had been used by several Japanese companies, had been agreed upon earlier by the JEIDA group in its V 3.0 specifications.

The cards are about 3.3 mm thick, so that four-layer, double-sided cards can be housed. Fujitsu and other companies have been putting 20 to 24 chips on the double-sided cards, using TSOP (thin small-outline packages), an emerging form of surface-mount packaging, Reimer said.

Takemae said the most difficult issue facing the hardware group was how to deal with "hot insertion/hot removal," i.e., pulling out a card while the system is still operating, which can result in data loss. A major future issue is how to develop an I/O specification so that interface cards can be built into portable computers, "talking to" fax machines, telephones and pagers, printers, and other external devices.

Yamashita, of ASCII, and Terry Moore at Databook worked together to develop the META card interface format, with input from Dreyfoos of Microsoft.

META is a header format that tells the system what kind of card (such as an application or data card) is in the slot, what kind of semiconductor memory—and how much of it—is on the card, and so on.

"We really have worked hard so the consumer can just plug in the card and make it look like a floppy disk. We want this to be a consumer product," Sternglass said.

August 1990

COMPUTER TECHNOLOGY REVIEW ■ MASS STORAGE

Flash Memory Operates 10-20 Times Longer

by **Markus Levy**

Major technology changes in the 1990's will place new demands on memory devices. Mainframe computing performance is now available in a laptop PC, and hand-held solid-state calculators have become sophisticated organizers. The convergence of these two trends has led to the evolution of the notebook PC and the emergence of Flash memory.

With a minimum battery life of 20 hours and weight not exceeding five lbs, the completely solid-state notebook PC will reliably handle all performance requirements of the traveler. Flash memory helps make the design goals of the notebook computer a reality by replacing the majority of the memory technologies in the system. Psion, a leader in the notebook computer market, has created a product in which Flash memory is used for BIOS, OS, and secondary storage. This completely solid-state, DOS-compatible machine weighs only 4.5 lbs and operates on eight AA batteries for 25 hours.

To incorporate rapidly improving power management techniques for battery-powered systems, the BIOS must be software updatable—remotely over a modem or with a floppy disk sent by the OEM. As such, the EPROM no longer fulfills its classic role for code storage. However, designed with a similar memory cell structure based on ETOX technology (EPROM Tunnel Oxide), Intel Flash memory provides equivalent reliability and nonvolatility with the added advantage of one sec, chip-level, electrical erasability (hence the name 'flash'). Flash memory can occupy the EPROM's socket with minor hardware modifications, primarily, 12V (Vpp) and write enable (W/E) must be supplied to enable the software controlled erase and program operations.

Traditionally, when the computer boots up, the operating system (namely DOS) is read from the disk and downloaded to DRAM. Digital Research and Microsoft offer ROM-executable versions of DOS. Originally designed for the unchangeable ROM, this product now accommodates Flash memory which can easily be reloaded with newer revisions without removal from the system. Flash-executable DOS benefits the notebook computer because it reduces the system RAM required for DOS from 70K to 15K, reflecting both power and component savings. Additionally, the system bootup is almost instantaneous, commonly referred to as 'instant-on.'

Solid State Secondary Storage

Solid-state secondary storage has had the greatest overall impact on the notebook computer (See Fig). In this environment, the power consumption, reliability, size, and weight of the mechanical disk drive is unacceptable. For example, the active and standby modes of the small form factor (2 1/2-in.), 20 Mbyte disk drive typically consume 4W and 0.5W, respectively. As a comparison, the active and standby modes of the equivalent capacity of Flash memory, consisting of low power CMOS circuitry, typically consume only 0.15W and 0.04W, respectively. Obviously, for a truly accurate analysis, other components of the system should be included, but from the data storage point of view alone, the Flash memory disk will operate 10-20 times longer than the mechanical disk on a set of batteries.

Reliability issues will always exist with mechanical media in any type of portable equipment because of shock and vibration, but it is difficult to perform a theoretical analysis on this subject. Suffice it to say, that from an MTBF standpoint (as measured by disk drive manufacturers under normal operating conditions), a mechanical disk will typically run 50,000 hours. A Flash memory device (capable of 100,000 erase/program cycles) should continue to function past 1.6 million hours—a difference of two

© INTEL CORPORATION, 1990.
Reprinted with permission from Computer Technology Review.

Order Number: 295054-001

Flash Memory Should Offer 1.6 Million Hr MTBF

orders of magnitude.

Size and weight are also critical factors in the notebook computer. Two Mbits of Flash memory is now available in a thin small outline package (TSOP) with a height of 1.2 mm. Minimally, 16 of these tiny packages can be put into a pocket-sized IC memory card (15,789 cm^3 vs 215,384 cm^3 for the 2 1/2-in. mechanical disk drive) to make up a four Mbyte disk, an adequate supply of memory for the notebook computer. Flash memory is not the only technology used as a solid-state alternative to secondary storage. ROM and battery-backed SRAM drives are actually more common because of familiarity. However, each has inherent drawbacks. ROMs have historically been used in laptop systems to store unchangeable, preloaded software programs. To upgrade with software revisions, the ROM application hardfile is discarded and a new card is purchased—an undesirable expense for the user.

Battery-backed SRAMs enable the flexibility to continuously modify files. SRAMs are used both as floppy and hard drive replacements, only where very low densities are required. Besides not being practical for high-density applications, SRAMs also draw concern from unpredictable battery life.

Unlike the ROM drive, flash memories can be reprogrammed many times. Unlike SRAMs, the single transistor memory cell of flash (compared to 4-6 transistors for SRAM) is very scalable for photolithographic processes, promoting very high density devices. In an environment where high density is synonymous with secondary storage, flash will outsell and outlast volatile SRAMs because of cost and reliability advantages.

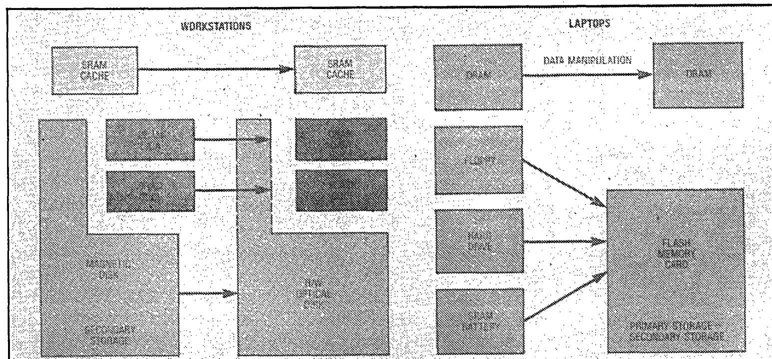


Fig Whereas in workstations Flash memory is used as cache for the OS and code, the laptop uses Flash memory for direct execution.

Using Flash memory

The adoption of Flash memory in a solid-state disk comes with the design challenge of interfacing a bulk-erasable memory with a file system requiring byte-level alterability. The simplest solution is to use a ROM-like approach and use the drive as an application hardfile with the extra benefit of being able to erase and reuse the disk. Microsoft has made major advances over this approach by developing a special file system it calls Flash File System. Based on linked-list techniques, this DOS-compatible file system, with superior performance over the mechanical disk, takes advantage of the chip-level erasability of Flash memory.

Although we have only discussed Flash memory applications in the notebook computer, its usage spreads well beyond. BIOS modification in desktop computers is also unavoidable due to increasing system complexity. Primarily aimed at fixing bugs, this technique also alleviates compatibility problems that might arise from the installation of the myriad of add-in boards and software packages. In addition, the OEM can promote upgrade service as a market distinction, as done by NCR and Olivetti.

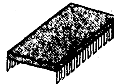
Flash memory disks are useful as application caches in high-end systems because of their nonvolatility and RAM-disk equivalent access speeds. Many types of industrial equipment are using Flash memory for code storage and data accumulation, replacing all forms of disk drives, both mechanical and solid-state.

Flash memory will continue to play a dominant role in the evolution of the notebook computer as well as every other application requiring a nonvolatile, reprogrammable, reliable, high density, and low cost memory. The flexibility of this new memory technology is driving costs down and generating an important alternative to disk memory. ■

Markus Levy is technical marketing engineer at Intel Corp. (Folsom, CA).

Nonvolatile, in-circuit-reprogrammable memories

Vendors of nonvolatile, in-circuit-reprogrammable, semiconductor memories want you to replace electromechanical memory components, such as DIP switches and disk drives, with chips. To spur your move toward silicon, IC vendors are rapidly expanding nonvolatile memory capacities beyond 1M bits and slashing the cost per bit.



Steven H Leibson,
Senior Regional Editor

Computer designers abandoned core memory in the early 1970s because silicon memory (RAMs and EPROMs) offered equal or better performance for less money. However, system designers also lost something in the bargain. Semiconductor RAMs furnish volatile memory; they remember only as long as they receive power. EPROMs need no power to retain their contents but you can't reprogram an EPROM without first erasing it for 20 minutes under a UV lamp. Consequently, disk drives now fill the nonvolatile memory needs of many computer systems.

Unfortunately, disk drives also have liabilities. They are much larger and heavier than ICs, they're more easily damaged and slower than semiconductor memory, they require a relatively complex disk controller, and they consume a lot of power. Disk drives may make no sense at all in small, portable, or embedded µP-based systems. Recognizing the inherent limitations of EPROMs and disk drives for nonvolatile storage, the makers of parallel-access, in-circuit-reprogrammable, nonvolatile memory ICs plan to restore nonvolatility to your processor's main memory, as in the days of core. These vendors are pushing device capacities beyond 1M bits (to 16M bits for hybrid modules) and dropping costs to $10/megabit as a means of swaying you to their way of thinking.

When you decide to use some form of nonvolatile memory IC in your next design, you immediately confront a range of technology choices. You can pick from battery-backed static RAMs (SRAMs), nonvolatile RAMs, EEPROMs, and flash EEPROMs. Your choice will depend on several factors including how much nonvolatile memory you need, how often you need to change the nonvolatile memory's contents, and how much you're willing to spend. The selection guide in **Table 1** can help you choose the type of device that best suits your needs. Choosing a particular device within a given device type then becomes a somewhat simpler problem. Unfortunately, none of the nonvolatile-memory technologies listed in **Table 1** offers the ideal nonvolatile-memory characteristics of unlimited endurance, fast storage times, and low cost. All exhibit some form of wearout failure and all require that you make some compromises.

If you need nonvolatile storage that closely mimics the infinite read/write characteristics of RAM,

© INTEL CORPORATION, 1991
Reprinted with permission from EDN

Order Number: 295062-001

then battery-backed SRAMs may well be your best choice. These devices work just like SRAM because they're built with SRAM chips. In addition to the RAMs, these devices incorporate one or two batteries and a power-management circuit that switches over from system power to battery power when necessary.

Table 2 lists battery-backed SRAMs offered by two vendors: Dallas Semiconductor and SGS-Thomson. These companies offer products whose capacities range from 2k to 128k bytes. Some of the products listed also incorporate an electronic time-of-day clock and calendar. The products' integral battery allows the clock and calendar to keep track of the time and date even when a system is turned off.

However, the integral battery is not just an asset, it is also the technology's main liability. Batteries die. **Table 2** lists the minimum number of years you can expect the battery to last for each device. Most of these products retain data for about a decade. After the battery is drained, the SRAM will operate as a RAM, but the product's nonvolatile abilities disappear.

The batteries in battery-backed SRAMs cannot be recharged, so you must replace the entire device to restore nonvolatile memory to your system. If you prefer to provide your own battery (perhaps one that's rechargeable), SGS-Thomson offers the MK48C02, which is a 2k-byte SRAM with integrated power management. The device has separate battery pins, which allow you to connect a backup power source to the RAM in addition to the normal power supply. The 48C02 costs less than $10 (1000).

The battery-backed SRAM's integral battery is not only an asset, it is also the technology's main liability.

Other types of nonvolatile semiconductor memories based on floating-gate electron storage (nonvolatile RAMs, EEPROMs, and flash EEPROMS) feature much longer data retention than battery-backed SRAMs. Although most of the data sheets for these floating-gate devices claim 10-year data retention, the technology is actually much better. In fact, Xicor's nonvolatile RAM and EEPROM data sheets specify 100-year retention times. Most nonvolatile-memory vendors will readily negotiate longer retention-time specs, if your application requires them. Apparently, many of these vendors don't want you to think that you're paying extra for the longer retention specs so they use the de facto "industry standard" spec of 10 years.

You should also be aware of the difference in retention-time specifications between floating-gate memory products and battery-backed SRAMs. The retention-time clock for floating-gate memories starts when you store data in the IC. For battery-backed memories, the retention-time spec refers to the elapsed time since the device was first powered up.

If the limited life of a battery concerns you, but you still need infinite read/write capabilites, consider using nonvolatile RAM for your nonvolatile memory. By merging SRAM and EEPROM arrays on one chip, nonvolatile RAMs do not require a battery and provide many of the same advantages provided by battery-backed SRAMs. While the system supplies power, the nonvolatile RAM's SRAM array pro-

Table 1—Nonvolatile-memory characteristics

	Nonvolatile-memory technology				
Characteristic	Battery-backed static RAM	Nonvolatile static RAM	Full-featured EEPROM	Flash EEPROM	EPROM
Write granularity	1 byte	1 byte (RAM), entire device (EEPROM)	1 byte or 1 page	1 byte or 1 page	1 byte
Rewrite granularity	1 byte	1 byte (RAM), entire device (EEPROM)	1 byte or 1 page	Entire device (Note 1)	Entire device (Note 2)
Relative storage speed	Very fast	Very fast (RAM), slow (EEPROM)	Slow (1 byte), medium (1 page)	Slow (1 byte), medium (1 page)	Slow
Programming and erase power requirements	5V	5V	5V	5V, or 5V and 12V	Not reprogrammable in system
Write endurance (cycles)	Unlimited	Unlimited (RAM) 10,000 to 100,000 (EEPROM)	10,000 to 100,000	100 to 10,000	<1000
Bit capacity per device	16k to 1M	256 to 64k	4k to 1M	256k to 2M	8k to 4M
Relative cost per bit	Highest	High	Medium	Low	Lowest

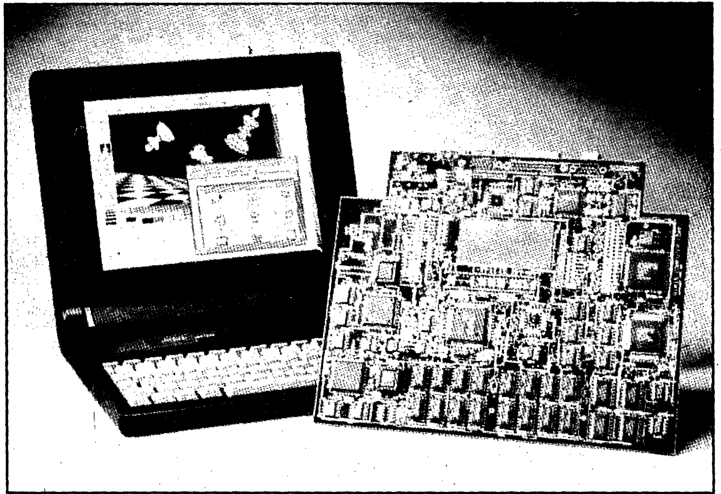
Notes: 1. Sector-erasable flash EEPROMs have 1-sector rewrite granularity.
2. EPROMS cannot be erased electrically. They must be erased with a UV lamp.

4

vides unlimited read/write capabilities. At any time, you can transfer the entire contents of the RAM array to the EEPROM array with one command. You can also transfer information stored in the nonvolatile RAM's EEPROM array to its RAM array with one command. In addition, nonvolatile RAMs automatically transfer their EEPROM's contents to their RAM array when power is applied.

Is 27 years long enough?

Like any floating-gate memory technology, the nonvolatile RAM's EEPROM array accepts only a limited number of changes before it will fail. However, as **Table 3** shows, that limit is at least 10,000 storage cycles. If you only transfer the RAM array to the EEPROM array when shutting the system down, that limitation should not represent much of a problem. For a product that is switched off once a day, the 10,000-cycle nonvolatile RAMs should last more than 27 years.



To hold the BIOS in its VH-286 notebook PC, Airus Computer Corp (Chicago, IL) uses a flash EEPROM from Seeq, which allows the company to change its BIOS over the telephone via a modem.

Unfortunately, if you use the nonvolatile RAM to save data prior to the loss of system power, you'll need circuitry that can foresee the power loss sufficiently early enough to complete the data transfer from RAM to EEPROM. Such predictive circuitry can be tough or impossible to design, so you may need to initiate transfers more frequently in case the system's power vanishes unexpectedly. If so, 10,000-cycle and even 100,000-cycle endurance may become a real limitation. However, if you only need to store infrequently updated items, such as con-

Table 2—Representative battery-backed static RAM modules

Manufacturer	Model	Size (bits)	Organization (bytes)	Access time range (nsec)	Minimum battery life (years)	Cost ($) (1000)	Comments
Dallas Semiconductor	DS1220	16k	2k	150 to 200	10	6.25 to 7.20	Industrial-temperature version available
	DS1225	64k	8k	70 to 200	10	8.75 to 9.10	Industrial-temperature version available
	DS1230	256k	32k	100 to 200	10	18.15 to 22.50	
	DS1243	64k	8k	200	10	16.90	Real-time clock/calendar
	DS1245	1M	128k	100 to 120	10	58.75 to 61.25	
	DS1387	32818	4k+50	370	10	13.75	Real-time clock/calendar, plugs into IBM PC/AT clock socket
	DS2217	1M+parity	128k+1 parity bit	200	10	70	30-pin, single-in-line package
	DS2230	512k	64k	200	10	39.60	Dual-ported device: one 8-bit port and one serial port, 40-pin single-in-line package
SGS-Thomson	MK48T02	16k	2k	120 to 250	11	17.10 to 27.12	Real-time clock/calendar
	MK48Z02	16k	2k	120 to 250	11	7.50 to 12.45	Industrial-temperature version available
	MK48T08	64k	8k	150 to 250	10	22.50 to 25.50	Real-time clock/calendar
	MK48Z08	64k	8k	70 to 200	11	13.13 to 18	
	MK48Z30	256k	32k	100 to 120	10	27 to 31.50	Available the second quarter of 1991
	MK48T87	512	64	240	10	11.25	Real-time clock/calendar, multiplexed address/data bus

figuration information or calibration data, the nonvolatile RAM may be a perfect choice.

Nonvolatile RAMs meld the infinite read/write capabilities of RAM with the nonvolatile storage of EEPROM. This combination offers more flexibility than other types of nonvolatile semiconductor memory, but you must pay for this flexibility. Smaller nonvolatile RAMs only cost a few dollars, so they are frugal replacements for DIP switches, jumpers, and other mechanical contrivances often used to store small amounts of semipermanent data on a board. However, the nonvolatile RAM's relatively high cost per bit makes it unattractive for large storage requirements. Further, the inherent complexity of the nonvolatile RAM's combined static-RAM/EEPROM cell limits the amount of memory that an IC vendor can fabricate on one IC using nonvolatile RAM cells. For these two reasons, you can buy only relatively small nonvolatile RAMs. Simtek's 16k- and 64k-bit parts offer significantly more storage space than nonvolatile RAMs from other vendors. Even so, 8k bytes doesn't seem to go as far these days as it once did.

For a product that is switched off once a day, the 10,000-cycle nonvolatile RAMs should last more than 27 years.

If you can restructure your design requirements so that you don't need unlimited read/write capabilities, you can conceptually eliminate the nonvolatile RAM's SRAM. You then end up with a part that is all EEPROM. Full-featured EEPROMs offer more capacity at a lower cost per bit than nonvolatile RAMs. **Table** 4 lists a large sample of available full-featured EEPROMs. (Note: **Table** 4 lists only bit-parallel EEPROMS. EDN plans to cover serial EEPROMs later this year.)

Restructuring your design requirements can be easier than you might think because a full-featured EEPROM differs from an EEPROM array of a nonvolatile RAM in one key aspect: You can change individual storage locations in a full-featured EEPROM independently. Consequently, you can write to one storage cell of a full-featured EEPROM several thousand times without reducing the endurance of the device's other memory locations. So, if your application changes stored information frequently, but doesn't change the same locations each time, the relatively limited endurance of full-featured EEPROMs may pose no problems. Alternatively, if you don't need to use an EEPROM's full capacity, you can adopt a scheme that uses only part of the EEPROM's memory until you near the endurance limits and then switches to a different section of the device. One of the EEPROM's memory locations can serve as a pointer to the section currently in use.

Nonvolatile RAMs can be faster than EEPROMs for multiple-byte

4

Table 3—Representative nonvolatile static RAMs

Manufacturer	Model	Size (bits)	Organization (bytes)	Access time range (nsec)	Store cycle time (entire array) (msec)	Array endurance (store cycles)	Cost ($) (1000)	Comments
Catalyst	CAT22C10	256	64x4 bits	200 to 300	10	10,000	3 (10,000)	Industrial-temperature version available
	CAT22C12	1024	256x4	200 to 300	10	10,000	3.75 (10,000)	Industrial-temperature version available
Seiko Instruments	S-2210R	256	64x4 bits	300	10	10,000	3.02	Industrial-temperature version available
	S-2212R	1024	256x4 bits	300	10	10,000	4.95	Industrial-temperature version available
Simtek	10C48	16k	2k	25 to 55	10	10,000	8.13 to 26.72	Pin-driven EEPROM store/recall, industrial- and military-temperature versions available
	10C68	64k	8k	25 to 45	10	10,000	18.87 to 59.37	Pin-driven EEPROM store/recall, industrial- and military-temperature versions available
	11C48	16k	2k	25 to 55	10	10,000	10.19 to 32.06	Software-driven EEPROM store/recall, industrial- and military-temperature versions available
	11C68	64k	8k	25 to 45	10	10,000	18.87 to 59.37	Software-driven EEPROM store/recall, industrial- and military-temperature versions available
Xicor	X2001P	1k	128	200 to 300	10	100,000	5.10	Industrial-temperature version available
	X2004P	4k	512	200 to 300	10	100,000	11.82	Industrial-temperature version available
	X2210P	256	64x4 bits	300	10	100,000	2.95	Industrial-temperature version available
	X2212P	1k	256x4 bits	300	10	100,000	4.92	Industrial-temperature version available

storage operations, but you can overcome this problem as well. Because the nonvolatile RAM fills its entire EEPROM array in one storage cycle, it has a high "effective storage rate" per byte. Some data-acquisition and real-time processing applications that use nonvolatile storage to record data at periodic intervals can't tolerate the milliseconds-long intervals between each store operation if the storage time exceeds the sampling interval. Nonvolatile RAMs circumvent this problem by allowing the application to accumulate data in the IC's RAM array and then store all of the accumulated data in one operation.

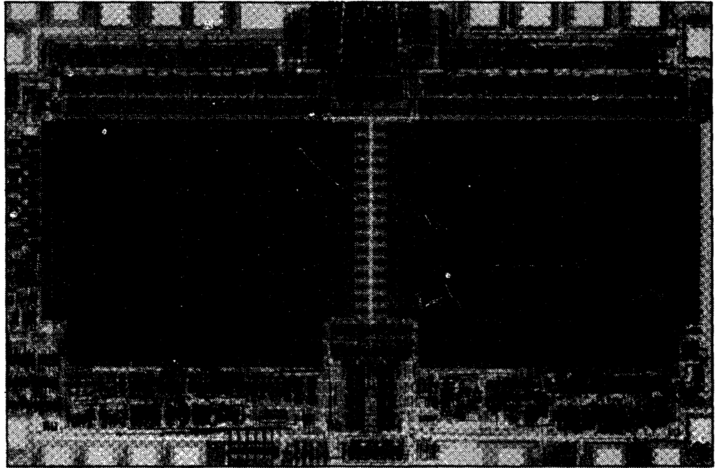


Fig 1—Two EEPROM sections allow this Xicor XC88C64 to store data and provide read capability simultaneously. This feature may allow you to reduce the number of memory devices in your design to one.

Pages save storage time

Some full-featured EEPROMs have page buffers that can store a contiguous page of memory locations in one operation. EEPROM page sizes range from 32 to 128 bytes. Using the page-write feature, you can accumulate a page worth of data and then store that page in the EEPROM in about the same time required to store 1 byte. The full-featured EEPROM's page buffer, therefore, provides advantages similar to those of the nonvolatile RAM's RAM array.

This page-write feature works similarly for all vendors' full-featured EEPROMs that have page buffers. The EEPROM senses the first write cycle and starts an internal timer. If another write cycle occurs before the timer reaches a predetermined value (usually 100 μsec), the additional byte enters the page buffer, the timer resets, and the EEPROM postpones the actual storing of the information in the EEPROM array. Thus you can quickly fill the EEPROM's page buffer without activating the milliseconds-long storage cycle.

When you stop writing to the page buffer, the timer times out and the EEPROM stores the entire page in one operation. This scheme effectively reduces the time required to store an individual byte by the reciprocal of the page size. For example, if an EEPROM stores 64 bytes from its page buffer in one cycle, the "effective" time required to store 1 byte is 1/64 of the page-write storage time.

You can't use just one

Most full-featured EEPROMs have another shortcoming that complicates their use: They don't respond to read cycles while performing a store operation; the EEPROM's contents become inaccessible for several msecs at a time. If you store executable code in an EEPROM with this trait, your processor will not be able to fetch instructions from the EEPROM while it is storing information. Consequently, you cannot use just a full-featured EEPROM to store your code in systems that modify the code space because access to the code will be interrupted for long periods of time. You can solve this problem by combining an EPROM and an EEPROM, or two EEPROMs. If you use two EEPROMs, be certain to initiate store operations in only one device at a time. You can also solve this problem using one of Xicor's X88C64s, which incorporates two independent EEPROM arrays (**Fig 1**). This device allows access to one of its EEPROM arrays while it is storing data in the other.

The X88C64 represents the latest step in the evolution of full-featured EEPROM design. Early EEPROMs were hard to use because they incorporated no address or data latches to hold values during the long store operations, they required high programming voltages, they were sensitive to power-supply transitions when a system was powered up or down, and they required external timing circuitry or software to control the store cycle. As the IC vendors developed expertise with EEPROM technology, they eliminated these design obstacles by adding circuitry to the EEPROMs.

However, the circuitry added for each new feature also adds cost to the device. Because of their cost per bit, full-featured EEPROMs pose no great threat to disk drives. By paring the full-featured EEPROM to the barest essentials, many ven-

dors of full-featured EEPROMs now offer a relatively new type of device, the flash EEPROM, which will give both EPROMs and disk drives some competition. **Table 5** lists flash EEPROMs offered by some sources.

To make flash EEPROMs inexpensive, some IC designers remove all circuitry that isn't absolutely necessary. You get a less expensive memory IC, but you also assume more responsibility for programming and erasing the device. For example, many flash EEPROMs lack on-chip charge pumps and require an external 12V source for programming and erasing operations. Full-featured EEPROMs and flash devices from Atmel, Catalyst, and Texas Instruments generate

Table 4—Representative full-featured EEPROMs

Manufacturer	Model	Size (bits)	Organization (bytes)	Read/access time range (nsec)	Max write time (msec)	Page size (bytes)	Endurance (write cycles)	Cost ($) (1000)	Comments
Atmel	AT28C04	4k	512	150 to 250	1	NA	10,000	3.20 to 6.50	0.2-msec fast-write and 100,000-cycle versions available
	AT28C16	16k	2k	150 to 250	1	NA	10,000	4.50 to 14	0.2-msec fast-write, 100,000-cycle, 45-nsec, and low-power versions available
	AT28C64	64k	8k	150 to 250	1	NA	10,000	6.50 to 17	0.2-msec fast-write, 100,000-cycle, 55-nsec, and low-power versions available
	AT28PC64	64k	8k	150 to 350	2	32	10,000	6.50 to 17	100,000-cycle version available
	AT28C256	256k	32k	150 to 350	10	64	10,000	40 to 59	3-msec fast-write and 100,000-cycle versions available
	AT28C010	1M	128k	120 to 250	10	128	10,000	350 to 550	100,000-cycle version available
	AT28C1024	1M	64k×16 bits	120 to 250	5 (typ)	64 words	10,000	420 to 650	100,000-cycle version available
Catalyst	CAT28C16A	16k	2k	200	10	NA	10,000	3 (10,000)	
	CAT28C16V3	16k	2k	700	20	NA	10,000	3.75 (10,000)	Operates with power supplies as low as 3V
	CAT28C64A	64k	8k	120 to 200	10	32	10,000	5 (10,000)	Industrial-temperature version available
	CAT28C256	256k	32k	200 to 300	10	64	10,000	25 (10,000)	
Dense-Pac Microsystems	DPE256Q8	2M	256k	70 to 250	10	64	10,000	380 to 760	Industrial- and military-temperature versions available
	DPE256S8N	2M	256k	135 to 300	10	128	10,000	1075 to 2150	Industrial- and military-temperature versions available
	DPE3232V	1M	128k×8, 64k×16, or 32k×32 bits	70 to 250	10	64	10,000	200 to 400	Industrial- and military-temperature versions available
	DPE32X16A	512k	64k×8 or 32k×16 bits	70 to 250	10	64	10,000	120 to 1250	Industrial- and military-temperature versions available
	DPE41288	1M	128k	90 to 250	10	64	10,000	220 to 450	Industrial- and military-temperature versions available
	DPE45128	4M	512k	100 to 350	10	64	10,000	800 to 1600	Industrial- and military-temperature versions available
	DPE6434	2M	256k×8, 128k×16, or 64k×32 bits	90 to 250	10	64	10,000	400 to 800	Industrial- and military-temperature versions available
	DPE832V	55 to 250	32k×8, 16k×16, or 8k×32 bits	55 to 250	10	32	10,000	140 to 275	Industrial- and military-temperature versions available, pin-grid-array package
	DPE8M624	90 to 250	128k×8 or 64k×16 bits	90 to 250	10	64	10,000	220 to 440	Industrial- and military-temperature versions available
Exel	XLS2804	4k	512	250 to 450	10	NA	10,000	4.21 (250 nsec)	
	XLS28C16	16k	2k	100 to 250	5	NA	10,000	3.41 (200 nsec)	

the necessary programming and erase voltages internally.

You can view the requirement for an external supply of 12V as both a liability and an asset. Certainly it's something extra that you must supply. However, you don't necessarily need or want to design that extra power supply into your product. If you want to allow flash EEPROM programming only under certain, controlled conditions, you can put the 12V supply in a programming station instead of the product itself. A cable can carry the 12V along with the programming signals. This scheme ensures that no inadvertant device programming can occur while your product is in use.

To save transistors, most flash

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Max write time (msec)	Page size (bytes)	Endurance (write cycles)	Cost ($) (1000)	Comments
Exel (continued)	XLS2864	64k	8k	250 to 450	10	32	10,000	10.88 (250 nsec)	
	XLS28C64	64k	8k	150 to 250	5	64	10,000	10.88 (1500 nsec)	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	NA	10,000	6	
	[illegible]	[illegible]	[illegible]	[illegible]	10	64	10,000	27	
Mosaic	MEM832WJ	256k	32k	90 to 250	10	64	100,000	79 to 168	Military part
	MEM864VX	512k	64k	90 to 250	10	64	100,000	175 to 349	Military part, single-in-line module
	ME128SC	1M	128k	120 to 250	12	64	10,000	560 to 840	Military part
	Puma 2E1000	1M	128k×8, 64k×16, or 32k×32 bits	90 to 250	12	64	100,000	560 to 1480	Military part, pin-grid-array package
NEC	[illegible]	[illegible]	[illegible]	[illegible]	10	NA	100,000	4.50	
	[illegible]	[illegible]	[illegible]	[illegible]	10	32	100,000	7.80	
	[illegible]	[illegible]	[illegible]	[illegible]	10	64	10,000	30	
Samsung	KM28C16	16k	2k	150 to 250	2	32	10,000	2.45	Industrial-temperature version available
	KM28C64	64k	8k	200 to 250	5	32	10,000	4.40	Industrial-temperature version available
	KM28C256	256k	32k	150 to 250	5	64	10,000	23	100,000-cycle and industrial-temperature versions available
Seeq	2816A	16k	2k	[illegible]	10	NA	10,000	5.25 (300 nsec)	2-msec fast-write and 1,000,000-cycle versions available
	[illegible]	64k	8k	[illegible]	10	64	10,000	8.25 (300 nsec)	
	28C256	256k	32k	[illegible]	10	64	10,000	30 (300 nsec)	90-nsec version available
	[illegible]	[illegible]	128k	[illegible]	10	128	10,000	595 (250 nsec)	5-msec fast-write version available
Simtek	STK28C256	256k	32k	70 to 120	10	64	100,000	55 to 80	Military-temperature version available
Xicor	[illegible]	[illegible]	8k	[illegible]	10	NA	10,000	13.52	
	[illegible]	[illegible]	[illegible]	[illegible]	10	32	10,000	<10	Multiplexed address/data bus, write protection for 1k-byte blocks
	[illegible]	[illegible]	32k	[illegible]	10	64	10,000	29.50	
	[illegible]	[illegible]	64k	[illegible]	10	128	10,000	100	
	[illegible]	1M	128k	[illegible]	10	256	10,000	275	

EEPROMs also lack the circuitry which allows you to erase individual locations. When you erase them, they erase completely. However, you don't always need to erase a flash EEPROM before writing to it. You can always write to an erased, but as-yet-unwritten, location, which results in a write time comparable to that of a full-featured EEPROM. However, once you've written to a location, you must erase most flash EEPROMs entirely before you can rewrite that same location. Thus the flash EEPROM's rewrite time can be very long indeed.

Flash EEPROMs from Catalyst, Philips Components-Signetics, Seeq, and Texas Instruments allow you to erase individual memory sectors and Atmel's flash products (dubbed PEROMs for programmable, erasable ROMs) automatically erase a location before writing new bits into that location. Sector erasure can save your application time

Table 5—Representative flash EEPROMs

Manufacturer	Model	Size (bits)	Organization (bytes)	Read/access time range (nsec)	Max byte write time (msec) (Note 1)	Max erase time (msec) (Note 2)	Sector size (bytes) (Note 3)	Endurance (erase cycles)	Cost ($)	Comments
Advanced Micro Devices	AM28F010	1M	128k	90 to 200	0.4	10,000	NA	10,000	26.50 (100) (200 nsec)	
Atmel	AT29C256	256k	32k	150 to 250	10 (64 bytes)	10	NA	1000	12 to 24 (1000)	Needs only 5V, programs in 64-byte blocks, autoerase during write.
	AT29C010	1M	128k	150 to 250	10 (64 bytes)	10	NA	100	29 to 45 (1000)	Needs only 5V, programs in 64-byte blocks, autoerase during write.
Catalyst	CAT28F512V5	512k	64k	120 to 200	0.4	100,000/ sector	2k	1000	58 (10,000)	Needs only 5V.
	CAT28F512	512k	64k	120 to 200	0.4 (1 or 4 bytes)	10,000	NA	1000	15 (10,000)	Can perform 4-byte writes. 10,000-cycle endurance version available.
	CAT28F010	1M	128k	120 to 200	0.4 (1 or 4 bytes)	10,000	NA	1000	22 (10,000)	Can perform 4-byte writes. 10,000-cycle endurance version available.
Dense-Pac Microsystems	DPZ1MS16P	16M	2M×8 or 1M×16 bits	150 to 250	0.4	30,000	NA	10,000	550 to 75u (1000)	Industrial- and military-temperature versions available.
	DPZ2MS516P	32M	4M×8 or 2M×16 bits	200 to 300	0.4	30,000	NA	10,000	900 to 1200 (1000)	Industrial- and military-temperature versions available.
Intel	28F010	1M	128k	120 to 200	0.4	30,000	NA	10,000	12.05 to 16.95 (1000)	Automotive-temperature version available.
	28F020	2M	256k	150 to 200	0.4	30,000	NA	10,000	32.05 to 35.25 (1000)	
Mitsubishi	M5M28F101P	1M	128k	100 to 150	0.4	10,000	NA	10,000	20 (1000) (100 nsec)	
Philips-Components/ Signetics	48F010	1M	128k	200 to 300	11.53	13,425	1k	100	15 (1000)	1000-cycle version available.
Seeq	48F512	512k	64k	200 to 300	11.53	13,425	512	100	13.50 (1000)	1000-cycle version available.
	48F010	1M	128k	200 to 300	11.53	13,425	1k	100	22 (1000)	1000-cycle and military versions available.
SGS-Thomson	M28F256	256k	32k	100 to 200	2.65	840	NA	100	3.50 (10,000)	1000- and 10,000-cycle versions available.
Texas Instruments	TMS29F256	256k	32k	170 to 300	60 (1 to 64 bytes)	150	NA	100	11 (1000)	Needs only 5V. 1000- and 10,000-cycle versions available.
Toshiba	TC58257AP	256k	32k	170 to 250	2.5	2050	NA	100	10.95 to 12.05 (1000)	
	TC58F1001P	1M	128k	150 or 200	2.65	3,000,000	NA	100	15 (100)	10,000-cycle version available.

Notes: 1. A page-write feature allows some flash EEPROMs to program more than one location in the same time required to program one location.
2. For most flash EEPROMs, you must first write zeroes into every memory location to place an equal electrical charge in each memory cell. This pre-erasure operation will greatly extend the time needed to erase large-capacity parts. Most parts will erase in far less time than the maximum listed. Atmel's AT29C256 and AT29C010 automatically erase before writing into a location.
3. You can erase flash EEPROMs with sectors on a sector basis so you need not reprogram the entire part after an erasure.

because most flash erasure algorithms require you to write zeros into each location before erasing the part. Each write operation takes hundreds of μsecs or msecs to complete, so writing to the smaller number of locations contained in a sector can mean faster erasure times.

Erase flash EEPROMs carefully

Writing zeros into each memory location charges each of the flash EEPROM's memory cells to the same electrical potential (a charged cell represents a logical zero) so that subsequent erasure will drain an equal amount of free electrons from each cell. Failure to equalize the charge in each flash-EEPROM cell prior to erasure can result in the overerasure of some cells by dislodging bound electrons in the floating gate and driving them out. When a floating gate becomes depleted in this way, the affected transistor can no longer be turned off; overerasure literally destroys some flash EEPROMs. Flash EEPROMs from Seeq are not sensitive to overerasure because the devices' split-gate cell design inherently precludes such problems. As a result, Seeq's flash EEPROM erasure algorithm does not require you to write zeros to all locations before erasing.

Most flash EEPROMs also lack built-in timing circuits so your host CPU must time the programming and erasure algorithms. This feat isn't always as easy as you may think. **Fig 2** illustrates the complexity of Intel's programming algorithm. Flash EEPROMs are relatively new, so the algorithms for programming and erasing the parts vary from vendor to vendor. However, there is an industry-standard identification sequence for flash EEPROMs, so your software can determine what vendor's part it's dealing with if you want to write one piece of code that can program and erase several types of flash memory. When you're implementing the programming and erasure algorithms, you must be careful to plan for inadvertent resets, long delays caused by interrupts, and any other events which may distract the host CPU and disrupt the proper timing of flash operations. Remember, failure to time the erasure properly can permanently damage some flash EEPROMs.

Despite these liabilities, you will still want to consider flash EEPROMs for your nonvolatile memory requirements because they cost less per bit than the other types of nonvolatile, in-circuit-reprogrammable semiconductor memories. The cost for 1M-bit flash memories has dropped by a factor of 10 in the past year and most flash EEPROM vendors state that they will overtake the nonvolatile-memory price leader, the EPROM, in the near future. In fact, some computer manufacturers, such as Airus

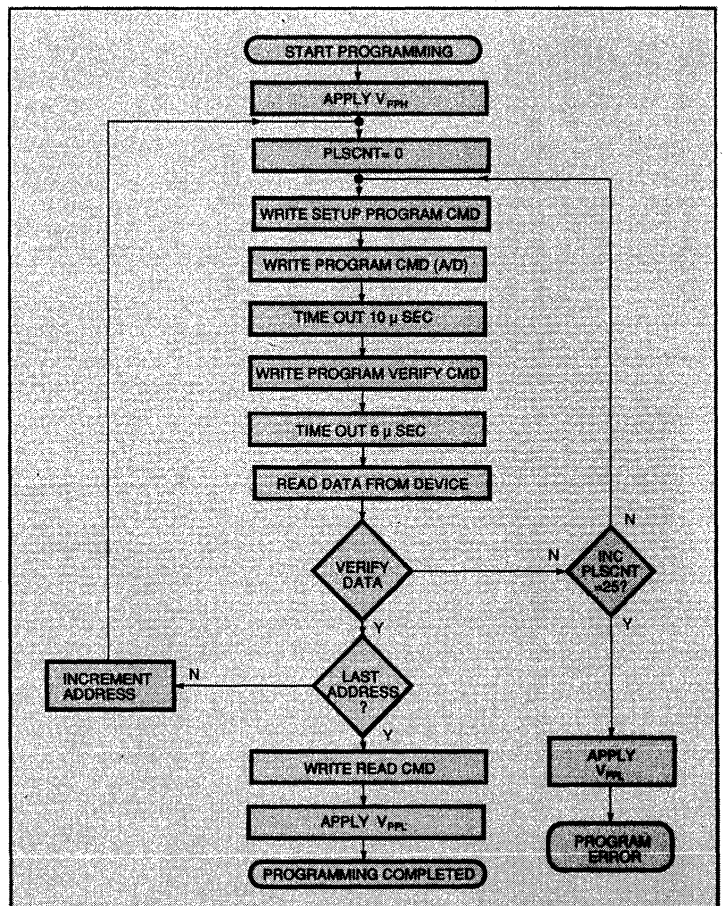


Fig 2—The flash EEPROM's low part cost demands another price: They require a host CPU to execute a fairly complex programming algorithm as illustrated by this programming flowchart for Intel's 28F010 1M-bit flash device.

(Chicago, IL) and Solbourne (Longmont, CO), are already using flash EEPROMs instead of EPROMs to store executable code in their products. The cost advantages of in-circuit reprogramming already outweigh the part-cost differential between EPROMs and EEPROMs if the EPROMs must be changed just once. The labor cost incurred by opening a product and changing the EPROMs tilts the balance in favor of flash memory.

In addition, flash EEPROMs allow you to eliminate the IC socket that you often use with EPROMs just in case you need to change your code. Elimination of the socket further reduces the parts-cost difference between EPROMs and flash EEPROMs and boosts system reliability because sockets can be one of the least reliable parts of a system. You can solder flash EEPROMs directly onto your circuit board and still accommodate code changes through in-circuit reprogramming.

EPROM replacement represents a big market for flash EEPROMs because all μP-based and many microcontroller-based systems contain at least a few EPROMs that flash devices can replace one for one. However, flash EEPROM vendors, such as Intel and Toshiba, have set their sights on an even bigger market. They will eventually replace a large number of disk drives with flash EEPROMs as well. At first glance, that goal seems more like an impossible dream than a realistic marketing plan. Disk drives are well established as the storage medium of choice for a range of computer systems. So was core memory before semiconductor RAM appeared. No technology in the electronics industry lives forever.

To overtake disk drives, flash-memory vendors must first overcome a few obstacles. For example, flash EEPROMs do not exactly match the characteristics and capabilities of disk drives so they cannot exactly replace them. Read/access times are much faster for flash devices than for disk drives, which is a real advantage. Storage times can also be much faster for flash EEPROMs, but they can also be much slower if you need to erase them before writing new information. In addition, flash EEPROM vendors aren't yet ready to match the low cost per bit you achieve with disk drives. However, they are preparing for the battle.

As part of that preparation, Intel recently introduced two plug-in memory cards based on its flash EEPROMs. The company's 1M-byte iMC001FLKA and 4M-byte iMC004FLKA cards measure 85.6×54×3.3 mm and conform to the 68-pin specification developed by the Personal Computer Memory Card International Association. Because memory cards do not work like disk drives, Intel has worked with Microsoft (Redmond, WA) to develop a Flash File System that allows a DOS-based computer to store and retrieve files from Intel's memory cards. The 1M- and 4M-byte cards sell for $298 and $1198 (1000), respectively. A developers' kit containing a 1M-byte memory card, an IBM PC interface board, and an evaluation copy of Microsoft's Flash File System costs $499.95. At today's prices, flash EEPROMs clearly do not threaten disk drives for most applications. However, if your design must meet rugged or low-power requirements, these memory cards give you a viable alternative to consider.

As usual, IC technology refuses to stand still. In the past year, flash EEPROM prices dropped by an order of magnitude. That change occurred without any increase in device density. As flash EEPROM vendors start doubling and redoubling device capacities, they plan to drop the cost-per-bit of their products until they compete with disk drives. If this plan still seems overly ambitious to you, consider this: More than one IC vendor plans to make the flash EEPROM its IC technology driver in the 1990s.

EDN

FLASH MEMORY CARDS

This breakthrough technology enables the mass production of a cost-effective, high-density in-system alterable nonvolatile memory.

Portable computer and handheld equipment designers are continually called upon to develop products that have the same functionality and performance of desktop systems. At odds with these market requirements are the system characteristics of small size, light weight, low power consumption, high reliability and ruggedness. Typically, it is the memory storage media of these systems that has the largest impact on system functionality and performance.

A solid-state memory technology that addresses these seemingly competing requirements is flash memory. This breakthrough technology enables the mass production of a cost-effective, high-density in-system alterable nonvolatile memory.

This ideal combination of characteristics provide the basis for a memory card that redefines the functionality and performance of portable PCs and equipment. Enter the flash memory card. (See Figure 1).

What About Disk Technology?

From hand-held and notebook PCs to data collection terminals to calculators, there exists the need for a solid-state mass storage technology that consumes less power, weighs less, lasts longer and provides higher performance than disk-based technology. A high-density flash memory card provides these advantages over small form factor hard disk drives. Figure 2 compares Intel's 4MB Flash Memory Card with a 2.5" Hard Disk Drive.

The disk-to-DRAM or SRAM serial download process becomes one of the

About the Author:

Lou Hebert is the Senior Product Marketing Engineer in the Flash Memory Operation of the Memory Components Division of Intel. He has been involved in IC-related work for 10 years, the last 5 of which have been with Intel. Hebert received his BSEE Degree from the University of California at Santa Barbara in 1980, and an MBA Degree from California State University at Sacramento in 1985.

© 1991 INTEL CORPORATION
Reprinted with permission from Memory Card Magazine, March/April 1991

Order Number 295065-001

ADVANTAGES OF FLASH MEMORY CARDS vs. HARD DISK DRIVES

Characteristic	Flash Card	2.5" Hard Disk Drive
■ Performance	250 nanoseconds	20 miliseconds
■ Power (Active/Standby)	250mW/5mW	3W/500mW
■ Weight	28 Grams	150 Grams
■ Size	0.95 Cubic Inch	6 Cubic Inches
■ Mechanical Shock	>50 Gs	5 Gs
■ Reliability (MTBF)	1 x 10^{6} Hours	1 x 10^{5} Hours

Figure 2.

largest performance inhibitors. Portable disk-based systems incur an additional performance hit when disk drives are powered down and then brought back up again to conserve battery power. Disk seek and access times increase when power management software idles disk drives to save precious system battery life. Using a flash memory card as a ROM-disk or application cache increases read-performance by a factor of more than 100 times. In addition, a flash memory card allows for execute-in-place (XIP) for code execution and data reading. Here, the CPU takes advantage of flash memories parallel architecture and random access to execute code directly from the flash memory card. This architecture increases performance of low-end systems.

Power consumption is one of the most important design factors for portable, battery-operated PCs and equipment. Flash memory cards consume hundreds of miliwatts of power in the active mode compared to watts for disk drives. Flash memory cards consume less power in the standby mode and power-up much quicker into the active mode. Using low-power CMOS technology provides the designer with the option of designing in smaller commercial batteries or providing longer operating periods with larger, rechargeable battery packs.

Some portable applications simply cannot accommodate the weight and volume consumed by a conventional disk drive. In contrast, flash memory cards are approximately one-fifth the weight and consume one-sixth the volume of the smallest hard disk drives.

Lastly, ruggedness is a very important reliability concern, especially in the portable environment. Mechanical by nature, disk drives wear out faster than solid-state technology. Being solid-state, flash memory cards tolerate a larger mechanical shock and operate longer than disk drives by at least a factor of ten.

Better Solution Than Other Solid-State Alternatives

Other existing solid-state cards (SRAM, OTP/EPROM, EEPROM) already address the disadvantages of disk-based technology. However, the other memory card technologies do not address all of the requirements of the portable computing market. Why then, are flash memory cards more beneficial? The flash memory card provides the best combination of attributes not available from other solid-state memory technologies. These characteristics include high-density, low power consumption, rewritability, nonvolatility, cost-effectiveness and the ability to store both code and data. It is this combination of properties that enable the design of higher performance systems with more functionality.

Different Approaches Lead to System Level Tradeoffs

There exist approximately four major approaches to flash memory technology: 1) Intel's ETOX™ (EPROM Tunnel Oxide) approach, 2) the triple-poly approach, 3) the stepped gate approach and 4) the tunnel oxide approach. ETOX and stepped gate are based on EPROM technology, Tunnel Oxide is an

Characteristic	INTEL ETOX™	STEPPED GATE	TUNNEL OXIDE	TRIPLE-POLY
normalized cell size	1.0	1.3	1.7	1.7
external erase/write voltage	12-volts	12-volts	5-volts	12-volts
internal erase/write voltage	12-volts	12-volts	18-22 volts	18-22 volts
highest device densities available	2Mbits	1Mbit	256Kbits	1Mbit
current card density capability	4MBytes	2MBytes	512KBytes	2MBytes
endurance (# of cycles)	10K-100K	1K	100-1K	1K-10k
base technology	EPROM	EPROM	EEPROM	hybrid

Figure 3.

4

EEPROM derivitive, and Triple Poly is a hybrid of several technologies. Also, ETOX is in high volume production today. Figure 3 illustrates the characteristics of these four flash memory technology approaches.

Flash memory or Pseudo Flash memory technologies require algorithmic software control for erase/write operations and cannot perform individual byte-alterability.

Differences between the various approaches lie with cell size, erase and write times and algorithms, power supply requirements and endurance (number of program and erase cycles). Tradeoffs existing between these variables have a large impact on overall system functionality and performance. Thus, the designer must carefully analyze these differentiators to determine which technology best meets the requirements of the target application.

For example, larger applications programs suggest higher-density mass storage. The smallest cell size and the least amount of overhead circuitry on the silicon chip will provide the most cost-effective, high-density memory device and card. Intel's ETOX approach (5-volt read/12-volt erase/write) provides the smallest flash memory cell size when compared to the alternative technologies. The stepped gate, tunnel oxide and triple-poly approaches have larger cell sizes than the ETOX approach. These approaches achieve the same density levels over time, but at a larger die-area and thus cost penalty.

Tradeoffs come with density as well. The ETOX, stepped gate, and Triple-Poly approaches require a V_{PP} 12-volts for erase/write operations at the cell level. This requires a 12-volt power supply or some form of DC to DC 12-volt conversion. Though 12-volts seems inconvenient, economics show this technique to be the most cost-effective. Indeed, the cost of the 12-volt power supply amortizes quickly when using large amounts of mass storage memory.

The five-volt-only tunnel oxide approach provides convenience, but limits density on each individual device. The approach uses on-chip charge-pumps to convert system-supplied 3- or 5-volts to an even higher internal voltage (18-22 volts) than the fixed, lower 12-volt techniques. The on-chip charge-pumps reduce usable space for additional memory cell locations for a given device area. This, in turn, reduces overall density capability and lowers cost-effectiveness for a given photolithography. Software write-protect schemes consume additional area offsetting the 5-volt benefit. In essence, the combination of larger cell sizes, on-chip charge pumps and software write-protect circuitry limit memory device and thus, memory card density.

Lastly, flash memory cards used for data recording and data file storage require high endurance or cycling capability (the ability to erase and rewrite memory card contents many times). The key to high endurance lies with the quality of the tunnel oxide, the magnitude of the electric field placed across the tunnel oxide and whether or not the cell erases and writes through the same oxide region.

The stepped gate and tunnel oxide approaches perform erase and write operations through the same junction. The triple-poly and tunnel oxide approaches use high voltage (18-22 volts) at the cell level placing higher stress across the charge transfer junction. These factors increase the probability of oxide wearout over time and reduce endurance capability. Intel's ETOX approach erases and writes through separate junctions and, with 12-volts, places a lower electric field across the oxide. The 12-volt-induced electric field reduces the stress on the oxide junction orders of magnitude less than the triple-poly or tunnel oxide approaches.

Designers need to weigh the pros and cons of each flash memory approach as they relate to the requirements of the target application. Further, the approach selected will effect system implementation.

> Microsoft's Flash File System operates as a redirected drive under ROM-executable MS-DOS. More efficient than disk-based sectoring, this filing system uses a linked-list structure that distributes files and locator pointers throughout the entire flash memory card. This, in turn, optimizes the use of the available memory locations. Interfacing flash memory technology into the DOS world allows access to the huge library of DOS application software. Figure 4 illustrates a system implementation model interfacing Microsoft's Flash File System to a flash memory card.
>
> This marriage of innovative filing software and flash memory card technology lays the foundation for powerful system architectures that do not have to rely on disk technology for high-density nonvolatile storage.

Straightforward System Implementation

In the memory card form factor, flash memory excels in four application areas: 1) Updatable application and firmware code storage, 2) Data acquisition and recording, 3) Data file and data table storage and 4) as a code and data transport media between host and target systems. Given the advantages provided by flash memory cards, how do designers implement flash memory cards into today's portable designs?

Applications that require memory card removal for transport to another system need a standardized interface. The PCMCIA/JEIDA (Personal Computer Memory Card International Association/Japanese Electronics Industry Development Association) standards organizations prescribe such a standard for interfacing solid-state memory and I/O cards. Defined as a 68-pin hardware interface, today's versions of the standard allow for SRAM, ROM and FLASH memory card densities up to 64MBs. Systems can distinguish between these cards via a Card Information Structure (CIS) stored on each individual memory card.

The level of participation in the standards committees by major systems, hardware and software companies lends credence to the

standards effort and reduces risk for those implementing the standard. The interchangeability of the three different types of memory cards, along with an upgrade path, reduces risk for OEMs as well as end-users.

OEMs implementing the PCMCIA/JEIDA interface into a design incur low cost as well. Indeed, the OEM need only supply interfacing hardware and software for the flash memory card. The compatibility of memory interfacing within the PCMCIA/JEIDA standard leaves the option open as to which density of flash memory card best fits the application.

Hardware implementations also require software drivers (erase and write algorithms) that allow the system to erase and write to the flash memory card. These low-level software drivers interface the system bus and higher-level operating software to the flash memory card via the PCMCIA/JEIDA hardware interface.

PCMCIA/JEIDA specifies unique pin length assignments for protective power supply sequencing, allowing "hot" insertion. For those designs implementing 12-volt power supplies, the 12-volt power supply can be switched on and off to conserve battery life. Some flash memory cards provide a V_{PP} lockout feature that provides code/data protection while power supply levels stabilize.

File management schemes, such as Microsoft's Flash File System, allow the storage and retrieval of data files on flash memory cards in MS-DOS* based systems. (See sidebar.)

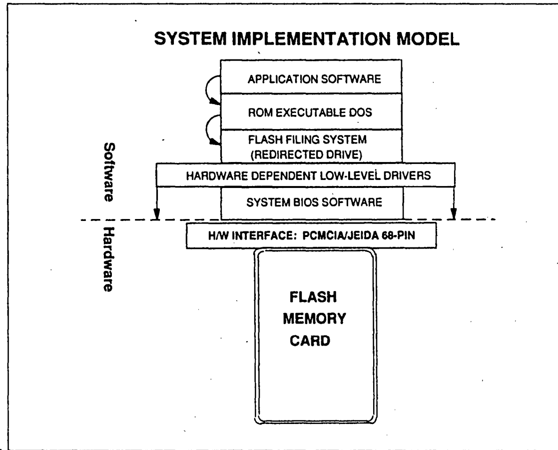


Figure 4.

Can Flash Memory Cards Catch Disk Technology?

Moving notebook PC and portable equipment designs to the next level of functionality and performance requires a removable solid-state mass storage medium. With the ideal combination of attributes, flash memory cards extend the capabilities of portable disk-based designs. Further, solid-state technology scales at a faster rate than disk-based technology allowing high-volume learning and cost reduction. As such, the density and cost of flash memory cards should approach low-density disks drives sometime in the 1996 timeframe. To illustrate, figure 5 shows the OEM price comparison between flash memory cards and small form factor hard disk drives at the 40MB density.

The Solid-State "Disk" Technology of the Future

Cost-effective flash memory cards used as mass storage may very well change the architecture of the PC as it is known today. Flash memory cards provide the "dream come true" for disk-based users: a removable hard disk or, from another viewpoint, a high-density floppy disk. The additional benefits associated with solid-state technology allow for new levels of performance and functionality in notebook PCs and portable equipment. Indeed, with ever-increasing densities and steep cost-learning curves, flash memory cards provide a breakthrough in the development of revolutionary new products. ☐

* *MS-DOS is a registered trademark of Microsoft Corporation.*

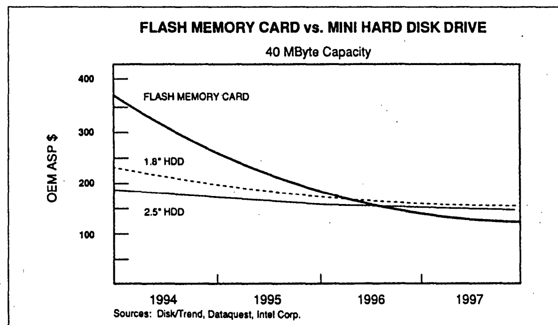


Figure 5.

4

eXecute-In-Place

Programs stored in cards can be executed directly by the processor when the appropriate mapping methods are employed.

MS-DOS computers typically execute programs from system RAM space. Often, these programs are loaded from magnetic-based, serial-transfer, I/O media like floppy or hard disks, or sometimes tape drives. Since memory cards are computer-readable random-access devices, programs stored in cards can be executed directly by the processor when the appropriate mapping methods are employed (see figure 1). An eXecute-In-Place (XIP) working group within the Personal Computer Memory Card Industry Association (PCMCIA) technical committee developed some basic concepts and a list of requirements that lay the implementation groundwork. Two basic goals are:

- Low-Cost XIP Hardware
- "Lowest Common Denominator" Compatibility Approach

Although laptop, notebook, and palmtop systems have a great deal of latitude supplying memory space for XIP within the regions of the first megabyte of memory, desktop systems do not. Typical ISA, EISA, and MCA systems have many constraints that complicate freeing regions larger than 64Kbytes for XIP. Therefore, XIP should be defined to work in the "lowest common denominator" configuration. If a system can supply extra memory regions for XIP beyond the "lowest common denominator" configuration then that is optional. This enables XIP applications to run on either a desktop or a laptop. The minimal 64Kbyte XIP configuration minimizes system costs with its low complexity and by supporting 1Mbyte-constrained systems. In order to meet these goals, the XIP requirements were defined as follows:

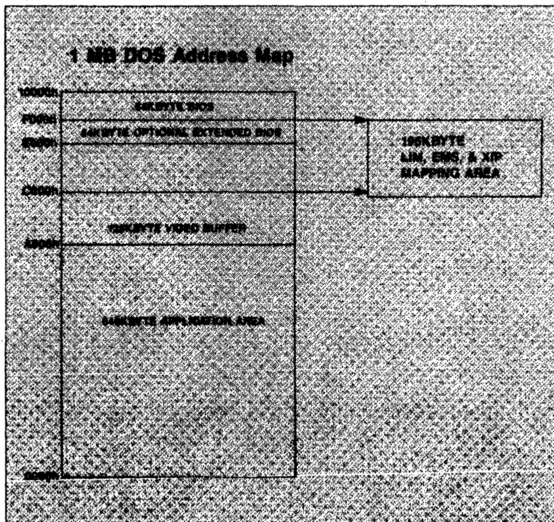


Figure 1.

Requirement 1

A system supporting XIP must be able to allow the mapping of an XIP memory block on a 16Kbyte boundary of system address space. Typically, these 16Kbyte boundaries will be above the 640Kbyte user-application address range. However, this is not mandated.

About the Author

Don Verner is a Senior Technical Marketing Engineer for the Flash Memory Operation of the Memory Components Division of Intel. He has been involved with the electronics industry for the last 7 years, the last two years with Intel. He received his BSET Degree from Purdue University.

© 1991 INTEL CORPORATION
Reprinted with permission from Memory Card Magazine, March/April 1991

Order Number 295066-001

Requirement 2

A system supporting XIP must be able to simultaneously map a minimum of four 16Kbyte pages into the system address space.

Requirement 3

A system supporting XIP mapping must be able to map the minimum of four 16Kbyte pages contiguously. Taken together, requirements 1 through 3 imply that the system be able to support mapping of four 16Kbyte pages into a 64Kbyte page frame. This minimum page frame must be able to start on a 16Kbyte boundary within the processor address space.

This requirement exactly matches a minimal Lotus-Intel-Microsoft (LIM) implementation usually referred to as the LIM 3.2 page frame.

Requirement 4

XIP memory mapping hardware will view the 64Mbyte physical address space of PCMCIA PC memory card as 4096 regions that are each 16Kbytes long. Each 16Kbyte region is call a "page." Obviously, every page starts on a 16Kbyte boundary (See figure 2, XIP Memory Card Address Map).

A minimal XIP mapping hardware implementation must be able to map any four of the 4096 pages in the PC memory cards partition into the processor's address space.

Requirement 5

Since a PC memory card can be "partitioned" into several file systems and an XIP partition, the previous requirements have a side effect on the physical location of an XIP partition within a card. An XIP partition within a PCMCIA compliant card must start and end on a 16Kbyte physical address boundary within the card. Only one XIP partition is allowed per memory card.

For cards that have individual, chip-erasable devices, like Intel's 1Mbyte and 4Mbyte flash memory cards, an XIP partition should not overlap devices that contain another partition. Instead, the XIP partition should start on an unwritten device. This prevents an erasure of one partition destroying data in another partition. If the XIP partition ends without filling up a chip-erasable device, the whole device is still considered part of the XIP partition.

The granularity of the XIP pages, the processor address boundaries where the pages can be mapped, and the page's physical address boundaries within the PC memory card simplify the design of the mapping hardware.

Requirement 6

XIP must be compatible with LIM 4.0. That is to say, the presence of the XIP driver and XIP applications in a system must not "break" either the LIM 4.0 driver or applications that depend on LIM 4.0.

Summary

The XIP specification made tremendous progress over the past year as evidenced by the clean, complete-draft showing at the March PCMCIA meeting. There is good consensus between Japanese Electronics Industry Development Association (JEIDA) and PCMCIA on the XIP specification. Only minor changes need to be made to the final draft for inclusion in the May pre-release of PCMCIA Release 2.0 specification. ☐

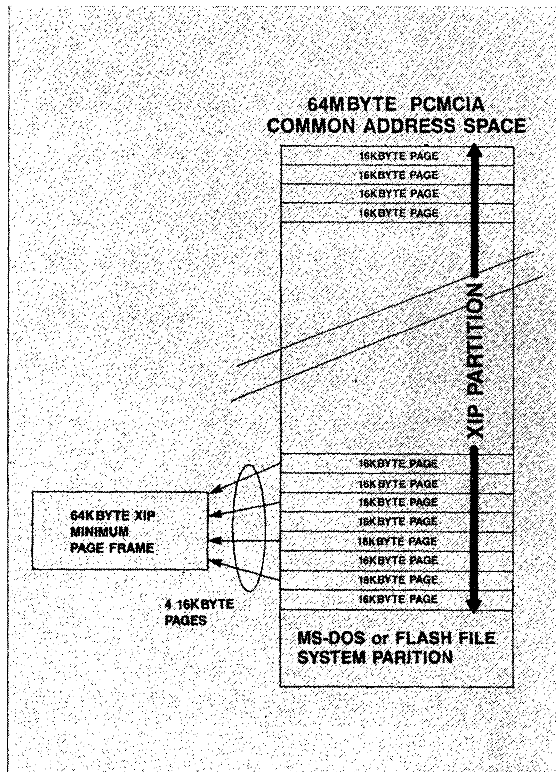


Figure 2.

FEATURE ARTICLE

Markus Levy

Interfacing Microsoft's Flash File System

Using Flash Memory Under MS-DOS

Back in issue 18, I discussed the design for a flash memory array based upon a PC/AT add-in board employing SIMMs or IC memory cards. Now we are ready to turn your flash memory platform into a DOS-compatible solid-state disk. In this article I'll show you how to interface Microsoft's special Flash File System (FFS) so you can store and retrieve files on this nonvolatile solid-state disk.

Before discussing the structural elements and benefits of the FFS, let's review some basic standard MS-DOS concepts. This will provide the motivation for implementing a special file system for flash memory.

A LOOK AT STANDARD MS-DOS

At the highest level, applications make requests to the MS-DOS function dispatcher through interrupt 21h. Arguments identifying the service desired and its options are typically passed in registers. To relieve application programs of the necessity of managing disk storage space, MS-DOS provides a file system manager. This series of MS-DOS services keeps track of disk storage using file and directory structures.

All block-device accesses by MS-DOS are made through a standardized device driver. MS-DOS makes requests to a block device by passing a data structure, called a request header, to the device driver. The request header contains the desired command and required arguments and is located by the device driver using a special routine, typically named "Strategy." For a read or write request, the logical sector to start the access and the number of logical sectors to transfer are provided. The device driver performs translations from the requested logical sector to the physical location on the device.

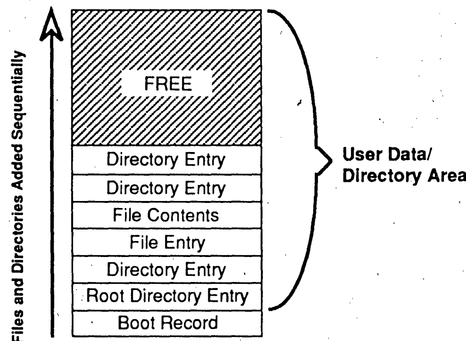


Figure 1—*Flash files are located by following linked-list pointers within each file and directory entry.*

For each block device, MS-DOS maintains four areas: a boot record, a File Allocation Table (FAT), a root directory, and a data area. The boot record is the first section on the disk. It contains a structure known as the BIOS Parameter Block (BPB) supplying MS-DOS with information about the disk, including sector size, sectors per cluster, number of FATs, directory size (number of files), and so on.

MS-DOS requires each disk to have a FAT to keep track of sector/cluster allocation. The FAT contains information on all the clusters that are allocated, free or unusable. It is an array of cluster pointers and each entry has a one-to-one correspondence with a cluster on the disk.

Grouping sectors to form clusters increases efficiency in terms of the memory required to manage the FAT. A cluster can consist of a different number of sectors. Four sectors (or 2048 bytes) per cluster is typical on a hard disk. The larger the cluster or allocation unit, the more potential of wasting space for files not sized to a multiple of the number of bytes in a cluster. For instance, a 20-byte file stored on a disk with a cluster size of 2048 bytes, wastes 2028 bytes.

Following the FAT is the root directory. Directory entries consist of the file name, attributes, time, date, file size, initial cluster number, and reserved bytes. When allocating space to a file, the initial cluster number is updated to point to the initial cluster number used by the file. The value at that location in the FAT points to the next cluster used by the file or contains an end-of-file marker. Thus, the allocation chain is a forward linked list. When extending a file and another cluster is required, MS-DOS replaces the end-of-file marker with a pointer to the next cluster, which is set to an end-of-file marker.

An increase in file size, or any type of file revision, results in a byte-alterable modification to the disk di-

© 1991 INTEL CORPORATION
Reprinted with permission from Circuit Cellar, Inc., June/July 1991

Order Number: 295067-001

rectory. In addition to file size change, modifications include time of last change, date, attributes, and file renaming.

THE NEED FOR AN ALTERNATIVE FILE SYSTEM

Recall that flash memory is a bulk-erase memory. The FAT and directory structures created for the byte-alterable magnetic disk are not ideal for a flash memory solid-state disk (SSD). Updating a FAT or directory entry requires complete erasure of the flash memory components containing the changing bytes. It is possible to implement a flash memory disk based on this approach, but the write latency times are unacceptable for general use. When a file is added, deleted, or modified, the directory could be copied to a RAM buffer and modified to reflect the change. After the flash memory device that contains this directory is erased, the modified directory is copied back.

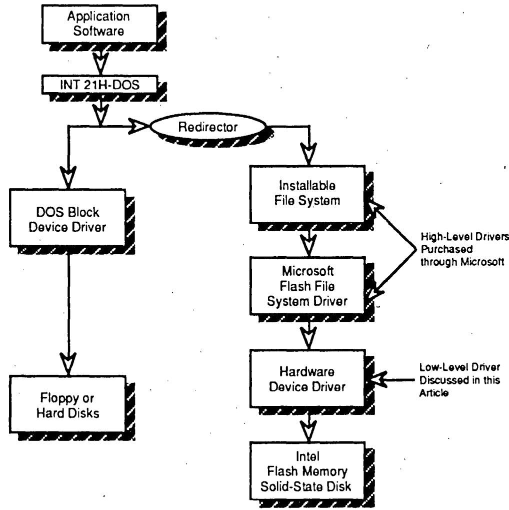


Figure 2—*A two-level architecture provides a consistent application interface while allowing for a variety of flash-memory hardware platforms.*

Disk imaging is another method of implementing the standard MS-DOS FAT scheme on a flash memory SSD. Using this method, files are first copied to a floppy disk. Then a special utility performs a disk copy transferring the FAT, directory, and all files to the flash memory SSD. This approach is useful for building an application cache that is FAT file system compatible, but all flexibility is lost.

Microsoft has developed a special file system utilizing the attributes of flash memory. To minimize fragmentation losses and allow arbitrary extension of files, the flash memory file system uses variable-sized blocks rather than the sector/cluster method of standard MS-DOS file systems. This flexibility is provided by employing a linked-list structure; that is, chaining files together using address pointers located within directory entries for each file.

Files and directories are written to the flash memory SSD using sequentially free memory locations—a stack-like operation (Figure 1). When the "stack" is full, the desired files are copied to another disk and the current disk is erased for reuse.

File and subdirectory information is attached to the beginning of each file, unlike the standard MS-DOS approach of directory and FAT placement. As directory and file entries are added, they are located by building a linked list. Besides containing the standard fields (e.g., name, extension, time, date of creation), a directory or file entry contains a status byte and various pointers used for the linked-list structure. The status byte, besides indicating whether a file/subdirectory exists or is deleted, also signifies valid sibling and/or child pointers and if a directory entry pertains to a file or a directory.

When a directory or file is requested, the flash memory SSD is searched beginning at the head of the linked list. The chain is followed from pointer to pointer until the correct entry is found. If the search arrives at the chain's end (an FNULL identifier is encountered), the system responds analogously to MS-DOS with a "File not found" message.

When deleted versions of a file appear on the flash memory SSD, the file system finds the most recent version. The status byte contains bit fields that indicate whether a particular file is valid or deleted. The directory information of a deleted file is still used for pointers of the linked list and the search proceeds until it finds the most recent and valid version.

FLASH FILE SYSTEM: ARCHITECTURE OVERVIEW

The Flash File System consists of two components: `IFS.SYS` (Installable File System) and `FEFS.SYS` (Flash EPROM File System). When an application accesses a disk through interrupt 21h, the MS-DOS kernel checks the drive letter. If the drive has been declared as a flash memory SSD, `IFS.SYS` intercepts the request through a proprietary interrupt 2Fh redirector interface and passes it to

Character Device Header

DW	Block Header	; Pointer to Next Device Driver
DW	0	
DW	1100000000000000B	; Attributes (CHAR, IOCTL control)
DW	Strategy	; Offset of Strategy Procedure
DW	Interrupt	; Offset of Interrupt Procedure
DB	'FIFIDEV£'	; Device Name Used by FEFS to Locate the LLD

Figure 3—*The Flash File System doesn't use a FAT and directory structure, but is uses a character device driver which must contain the proper header.*

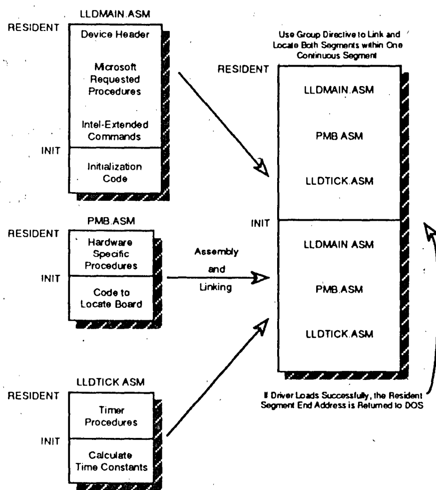


Figure 4—*The INIT portion of the device driver is used only once, at initialization, then "discarded."*

the FEFS.SYS driver. FEFS.SYS implements the FFS logic, developing and maintaining the linked-list structures.

The Microsoft FFS is implemented as a two-level architecture, where IFS.SYS and FEFS.SYS represent the high-level driver communicating with a low-level driver (LLD) that is hardware specific (Figure 2). This architecture provides a consistent application program interface while allowing for a variety of flash memory hardware platforms.

The LLD implements a set of device primitives for use by the high-level driver. This is not that different from the FAT file system as we know it. In that environment, MS-DOS implements a high-level, FAT file system driver interacting with a set of device primitives implemented as interrupt 13h.

WRITING THE LOW-LEVEL DRIVER

The remainder of this article will explain how to write this low-level driver providing the bridge between Microsoft's Flash File System and the page-memory board described in the December issue. While I would need forty pages to completely explain all the details of this implementation, there is enough here for you to understand the LLD software that can be obtained from the Circuit Cellar BBS. Also, if you have no intentions of implementing a DOS-compatible SSD, many of the device primitives from the low-level driver can be extracted for any type of flash memory application.

The LLD consists of three components: an MS-DOS device driver, the procedures called by Microsoft's FFS, and the page-memory board hardware-specific procedures. I have written the LLD in several modules to simplify any modifications necessary to accommodate hardware variations. The module LLDMAIN.ASM contains the MS-DOS interface routines, the Microsoft-requested procedures, and special Intel-extended procedures. PMB.ASM contains procedures specific to the page-memory board hardware, such as setting the page number and turning on V_{pp}. All the functions contained in this module control the I/O functions on the board. LLDTICK.ASM contains procedures associated with the timing requirements of the flash memory components, as well as routines used to provide a V_{pp} turn-off delay.

The programming and erase voltage, V_{pp}, is generated on the page-memory board using a DC-DC converter. When this converter is switched on, it takes anywhere from 20 to 100 milliseconds for V_{pp} to arrive at a stable voltage. This time depends on the amount of capacitive loading and the circuit used, as some have faster start times. If you are designing your hardware for a desktop system, V_{pp} can remain switched on. However, in battery-powered systems, V_{pp} should be switched off when not in use to conserve power. To accommodate these applications, I have written a procedure to generate a V_{pp} turn-off delay. This is similar to that for a floppy disk in that after two seconds of nonuse, V_{pp} is switched off. If several blocks of data are being written consecutively, your software will not have to delay waiting for V_{pp} to ramp up every time a new block is written.

The V_{pp} turn-off delay is calculated by installing an interrupt 1Ch (time-of-day clock) filter. This interrupt is generated every 18.2 milliseconds. Before servicing the original interrupt, our filter increments a count value. When that count value reaches 36, the procedure to turn V_{pp} off is called.

Since the LLD is an installable device driver, it requires a standard

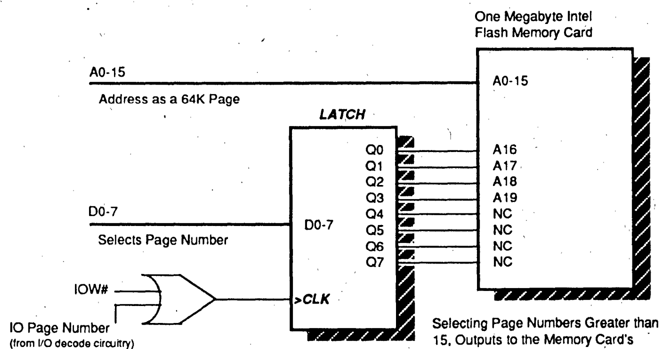


Figure 5—*Device density is dynamically determined, with page numbers beyond the valid size of the device handled by page wrapping.*

MS-DOS–compatible interface that provides a request header and entry points into the Strategy and Interrupt procedures. This portion of code primarily performs the initialization of the device driver.

Unlike the magnetic disk, the FFS does not implement a FAT and directory structure with sectors. Therefore, it is designed as a character device driver rather than a block device driver and must contain a character device driver header (Figure 3). During installation of the character device, MS-DOS passes a command number in a request header to the Interrupt procedure which dispatches a call to the `InitDriver` routine (command zero). This procedure is performed only once, immediately after the device driver is loaded into memory. The initialization procedure of the character device, `InitDriver`, is primarily responsible for locating the page-memory board, computing the time constants for the flash memory program and erase algorithms, and determining the quantity of flash memory available.

INITIALIZATION

Recall from the page-memory board design that the first four I/O ports are read to obtain the board's signature. A procedure, `SearchPMB`, called by `InitDriver`, reads sequential I/O ports until finding this signature. When the signature is found, the base port address is stored in a memory location to be used for future I/O port access. If the board is not present, the driver aborts its installation and returns the system memory to MS-DOS. This is done by passing an offset of zero back to MS-DOS in the INIT request header. Each of the three modules (`LLDMAIN.ASM`, `PMB.ASM`, `LLDTICK.ASM`) are divided into a `RESIDENT` and an `INIT` segment. These segments are joined using the group directive to ensure they are linked and loaded consecutively in memory. Using this technique, if the page-memory board is found, the ending offset of the resident portion of the device driver is passed back to MS-DOS (Figure 4). This "throws

away" the INIT portion which is not needed after initialization.

The page-memory board has four sockets for Intel Flash Memory single in-line memory modules (SIMMs). The hardware functions with one or more sockets populated, so during initialization the available memory must be determined. Similarly, with the Intel Flash Memory Card, an interesting situation is faced in accommodating density variations. This is best understood using the memory card as an example.

One- and four-megabyte card densities are available today, but the page-memory board's card socket handles up to 64 megabytes (based on the Personal Computer Memory Card International Association, PCMCIA, specifications). Assume that a one-megabyte card has been installed (although our software doesn't know this yet) containing eight one-megabit components (28F010). The first step in determining the module's density is reading the flash memory device ID from offset zero of page zero to obtain the device size. Comparison against a table of IDs versus densities allows the calculation of the page number needed to access the next component.

Theoretically, this process would continue, adding up the total number of components and multiplying that number by the component size to calculate the total module density. However, the page-memory board responds to setting the page register beyond the valid page range of the flash memory installed (Figure 5). For example, the one-megabyte memory card accommodates sixteen (numbered 0–15) 64K-byte pages. Pages 0, 16, 32, and so on, access the same memory location because of the wrap-around phenomenon. This inaccurately determines an infinite module size. How does our software know when to stop? Notice in the code (Listing 1) that the first device (page zero) is left in the READ_ID mode. Looping through the rest of memory, a wrap-around is detected when the device ID is already present in the first location of the device. This condition is used to terminate the loop.

Besides initialization, the MS-DOS device driver for FFS supports IOCTL reads and writes. The device driver uses the IOCTL commands to return control information to the program regarding the device. FFS issues the IOCTL commands to get and set the entry point for the low-level driver. Unlike block devices, character devices are located with a file open request. When FEFS.SYS installs, it links into the low-level driver by opening the file FIFIDEV£ (£ is the British Pound Sterling or IBM extended ASCII 9Ch), and performing an IOCTL read to obtain the pointer to the entry of the low-level driver.

Character devices do not support drive letters. You must use a "pseudo-block" device header during driver installation to reserve DOS letters for use by the FFS (Listing 2). After the character device installs, the block-device driver links the device driver into the device driver chain. Drive letters are established by providing fake BPB information in the block device driver header to pass back to MS-DOS.

```
GetPhyChar PROC NEAR
        ASSUME CS:PROG, DS:PROG, ES:NOTHING

        CALL    TurnVppON     ; Turn Vpp ON for READID_CMD
        XOR     AX,AX         ; AX contains page number
        CALL    SetPage       ; Routine to set page

        MOV     ES,FrameSeg   ; Point to page frame segment
        XOR     DI,DI         ; Point to start of device

        MOV     ES:WORD PTR[DI],(READID_CMD SHL 8) OR READID_CMD

        MOV     AX,ES:[DI]    ; Read manufacturer's ID
        CMP     AX,(INTEL_ID SHL 8) OR INTEL_ID ; Intel Devices?
        JE      GPC2          ; Yes, continue
GPC1:
        MOV     ES:WORD PTR[DI], (READ_CMD SHL 8) OR READ_CMD
        STC                   ; Indicate error and exit
        JMP     GPCExit
GPC2:
        MOV     BX,2          ; num of 64K pages/device
                              ;   for 1 Meg devices

; -----Compute total size of media

        XOR     AX,AX         ; Clear regs for accumulation
        MOV     CX,AX         ; Count num of device pairs
        SHL     BX,1          ; 2 Devices per 16 bits
GPC3:
        INC     CX            ; Count num of device pairs
        ADD     AX,BX         ; Point to next device pair
        CALL    SetPage       ; Select page for next device
                              ; Already in READ_ID mode?
        CMP     ES:WORD PTR [DI],(INTEL_ID SHL 8) OR INTEL_ID
        JNE     GPC4          ; No, Continue
        JMP     GPC5          ; Yes, We're done
GPC4:
        MOV     ES:WORD PTR [DI],(READID_CMD SHL 8) OR READID_CMD
                              ; Intel Devices?
        CMP     ES:WROD PTR [DI],(INTEL_ID SHL 8) OR INTEL_ID
        JNE     GPC5

; Set READ mode and check next device
        MOV     ES:WORD PTR [DI],(READ_CMD SHL 8) OR READ_CMD
        JMP     GPC3
GPC5:
        MOV     ES:WORD PTR [DI],(READ_CMD SHL 9) OR READ_CMD
        MOV     AX,CX         ; Get num of device pairs
        SHL     AX,1          ; Compute number of devices
        MOV     BX,2          ; Multiply by number of 64k pages
        MUL     BX
        MOV     TotalSizeHi,AX ; Set total size
        MOV     TotalSizeLo,0 ; Size is always a multiple of 64k
GPCExit:
        CALL    TurnVppOFF    ; Turn off Vpp

        RET
GetPhyChar ENDP
```

Listing 1—*Page wrap-around is detected by leaving the first device encountered in READ_ID mode. When a preexisting ID is found, the program knows to stop looping.*

By issuing the IOCTL read command, FEFS.SYS obtained the entry point to the low-level driver, which has been named FlashEntry. The FlashEntry procedure is to the FFS what the Interrupt procedure is to an MS-DOS device driver. It determines a command's validity before dispatching. FlashEntry handles calls from FEFS.SYS and applications that communicate directly with the flash memory, such as a formatter.

When writing the procedures to handle the FEFS.SYS commands, it is important to follow the entry and exit protocols. This is analogous to interrupt routines expecting parameters and status information to be passed within certain registers. Take for example the procedure ReadLogBlock that reads a block of data from the flash memory SSD into a buffer. FFS is informed by MS-DOS that the destination buffer is located at ES:BX. The CX register contains the number of bytes to read. A 32-bit pointer into flash memory is supplied by DI:DX. The unit number is passed in the AL register. Upon return from this procedure, the carry flag is expected to have the status of the operation, whether it was successful or unsuccessful.

There are thirteen procedures, including ReadLogBlock, that are defined by the Microsoft FFS technical specification. To more fully comprehend the file system, a brief discussion of each is helpful:

GetMediaCheck—Determines media status (same, changed, missing, unknown).

ReadLogBlock—Read a block of data at a logical address (which must be converted to a physical address).

WriteLogBlock—Writes a block of data from a buffer to the flash memory SSD.

CheckWrite—Checks a block of data for writability. If a byte is already programmed, but still has "one" bits, that byte can be rewritten to change the "ones" to a zero. This is useful for modifying the status byte.

EraseSSD—Erases data on a selected unit.

GetMediaInfo—Returns the description (type and size) of the installed media.

```
; Block Device Header use to set up drive letters with MS-DOS

Blockheader     DD    -1                    ; Last device in file
                DW    0000000000000000B     ; Block device
                DW    Strategy
                DW    BlockInt
bBlkUnits       DB    8 DUP (0)             ; Initialized in BlockInt

; These structures are passed back to MS-DOS to simulate
;  BIOS Parameter Block values:

BPB1            BPB   <512,1,1,2,128,1024,0F8H,2> ; Dummy values
BPB2            BPB   <512,1,1,2,128,1024,0F8H,2> ; Dummy values
```

Listing 2—*To reserve an MS-DOS drive letter, a different header must be used during driver installation. The new header acts like a block device, rather than a character device.*

FirstFree—Finds the first non-FFh byte from the end of the SSD and, as a result, finds the first available space.

SetMediaCB—Registers a procedure as the call-back procedure which is called by the LLD when a flash memory card is inserted or removed

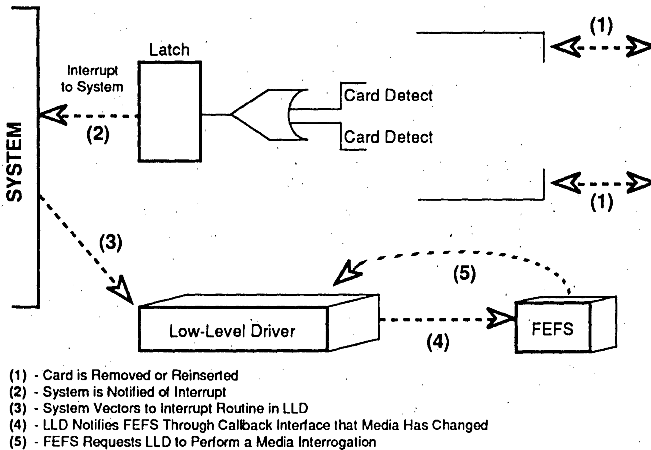


Figure 6—*The Call-back Interface supports changeable media in a system using Flash Cards.*

(see the discussion on Removable Media).

`GetMediaCB`—Returns the present media call-back procedure pointer. If zero is returned, no media call-back procedure exists.

`WriteLogByte`—Write a byte of data.

`ReadLogByte`—Read a byte of data.

`SetEraseCB`—Registers a procedure address as the call-back procedure called while erasing an SSD. It is used to display progress information.

`GetEraseCB`—Gets the address of the procedure called while erasing the SSD. A zero value returned indicates no erase call-back procedure has been registered.

```
BOOT_RCD          STRUC            ; Flash File System Boot Record
                                   ;  (and root dir) copied in
                                   ;  with formatter.
wStdID            DW 0F1A5H        ; Flash Media ID (Spells FLASH)
wUniqueIDLo       DW ?             ; Uniques ID for this Flash Memory
wUniqueIDHi       DW ?
WWriteVersion     DW ?             ; Flash version required to write
wReadVersion      DW ?             ; Flash version required to read
bPointerSize      DB 0             ; Num of bits in link list ptrs
pRoot             DB 3 DUP (0) ; Pointer to root directory
bVolumeLabel      DB 11 DUP (oFFH) ; Volume label
wEraseCyclesLo    DW OFFFFH        ; Num times device erased (low)
wEraseCyclesHi    DW OFFFFH        ; Num times device erased (high)

bManfID           DB ?             ; Manufacturer's ID
bDeviceID         DB ?             ; Device ID
wDeviceSize       DW ?             ; # of 64k pages in each device
wNumDevices       DW ?             ; # dev in SIMM or memory card
wTotalSizeLo      DW ?             ; Size of media (low word)
wTotalSizeHi      DW ?             ; Size of media (high word)
wTotalPages       DW ?             ; Total number of 64k pages
bIntelID          DB '1.0'         ; Indicates Intel format

pSibling          DB 3 DUP (OFFH) ; Start offset of data area
bRootName         DB 'ROOT    '; First entry must have this name
bRootExt          DB '   '
bStatus           DB OFBH          ; Directory entry
pPrimary          DB 3 DUP (OFFH)
pSecondary        DB 3 DUP (OFFH)
bAttributes       DB OFFH
wTime             DW ?
wDate             DW ?

BOOT_RCD          ENDS
```

Listing 3—*Any formatting utility must be hardware dependent. This listing shows the boot record copied into the SSD at the beginning of the FFS partition.*

SETTING UP THE PAGE NUMBER

During initialization, our software calculated the total flash memory available, and this information is passed back to `FEFS.SYS` (which knows nothing about the actual hardware used to access this memory). Because a page-memory scheme is used, the linear address supplied in DI:DX must be converted to a page number and offset within that page. A procedure, called `SetupRW`, performs this conversion and sets the hardware accordingly. In reality, if you are using a 64K-byte page size, this conversion is very simple: [DI] contains the page number and [DX] contains the offset. This is only tricky if you have designed your hardware with a smaller page size. In this case you would take DI and DX, move them into DX and AX, respectively, and divide DX:AX by the page size. AX would now contain the page number and DX the offset.

REMOVABLE MEDIA

If you have designed your page-memory board for the Intel Flash Memory card instead of SIMMs, you can include support for media change. The PCMCIA specifies the use of card-detect pins on the memory card. These pins can be wired to a latch that is set whenever the card is removed or inserted. This card-change line can be read before every access or can be tied to an interrupt line to allow instant acknowledgement of the card change. For both cases, Microsoft has provided a call-back interface.

A call-back interface allows the LLD to call a procedure internal to `FEFS.SYS` when the media changes. This function is similar to a software interrupt except it is private and cannot be accessed by other software. At initialization, `FEFS.SYS` requests the LLD to perform a "Set Media Call-back." In this procedure, the LLD stores the value in ES:BX which has been set up by `FEFS.SYS` as a pointer

to its internal media status call-back procedure. Now when the memory card is changed, the low-level driver, initiated by polling or interrupt, calls this procedure to let the FFS know that it needs to perform a media interrogation (Figure 6). This is analogous to removing a floppy disk in that MS-DOS must read the BPB and related information.

FORMATTING

A standard format utility is not incorporated in Microsoft's FFS because a formatter is hardware dependent. As such, a formatter was written that communicates directly with the LLD through `FlashEntry`. The formatter starts off by using the `EraseSSD` procedure. After erasing the SSD, a boot record (Listing 3) is copied to the SSD at the beginning of the FFS partition. During initialization, `wStdID` in the boot record is read to determine if the flash memory is formatted for the flash file system (notice that "F1A5h" spells Flash). If the SSD is FFS-compatible, the information in the remainder of the boot record describes the characteristics of the flash memory as well as the starting point for the FFS data area.

DEBUGGING

DOS Debug can be used for simple debugging purposes such as reading and writing I/O ports to test basic hardware functionality. Device drivers are difficult to debug because they are loaded during the DOS boot process. However, System Debugger by Sandpaper Software works well for this purpose. To use System Debugger to debug this device driver, the system must be set up so that a warm reboot may be taken at any time. After the debugger is loaded, breakpoints are set and a program called `BOOT.COM` is executed. This reboots the system without removing the debugger. When the breakpoint is reached, the debugger pops up and displays the information you need. One thing to note is that this debugger is designed for '386 systems because it uses the CPU's internal break registers and extended memory.

This flash file system project will provide a great education into the world of device drivers. Some of the concepts may seem a little tricky at first, but all of a sudden, a light bulb will go on, and you will be impressed by the functionality of this unique file system. Furthermore, once the project is complete, you will be impressed by the performance of the flash memory solid-state disk. This technology is rapidly gaining market acceptance and you have become a part of the secondary storage revolution. ✣

Markus Levy is an application engineer at Intel Corporation in Folsom, California, and holds a B.S.E.E. from California State University. His specialties include software and hardware implementations of solid-state disks in portable computers.

Flash Memory Card Redefines Portable Computing

Portability: It is the computing industry's watchword for the '90s. Achieving true portability presents everyone, from the component manufacturer to the system OEM to the software developer, with a host of challenges. The systems that meet these challenges will integrate some of the most advanced technologies available today. However, because portable computers are considered consumer products, they cannot compromise cost-effectiveness for the sake of technology. In short, they must offer consumers useful levels of performance, be easy to use, rugged, lightweight, reliable, yet not prohibitively expensive.

Today, the portable computer market is still in its infancy. However, if current predictions hold true, this infancy is now on the brink of an explosion. According to Dataquest, the portable computer market will experience a five-fold increase from 1991 to 1994, growing from 2.2 million units to 11 million during this four-year period. This represents an extraordinary opportunity for electronics manufacturers who can provide a product or technology solution that becomes an essential ingredient within the portable computing market.

Even at this early stage, the reprogrammable portable computing market is dividing itself into three major segments: laptop, notebook and palmtop. Although overlap exists, they are fundamentally distinguished in this way:

Type	Weight	Characteristics
Laptop	6-15 lbs.	Functions as a small desktop. Full screen and keyboard. Typically features hard and floppy disk drives.
Notebook	4-6 lbs.	Compacted screen and keyboard. Smaller size precludes having both a floppy and hard disk drive.
Palmtop	1-4 lbs.	Limited display, small keyboard. Handheld-size precludes use of any disk.

Similarly, the number of embedded portable computing applications and their respective unit volumes are growing steadily. A substantial portion of this growth is aided by the wide availability of development tools and applications software designed to work within a disk-based computing environment.

Because embedded systems span such a wide variety of applications and each is subject to different market variables, growth is more difficult to accurately quantify. Suffice it to say that from laser printers and robotics to measurement and medical equipment, the opportunities for suppliers to this market are abundant.

ORDER NUMBER 296839-001

If Not Disks, What?

Portable or not, reprogrammable or embedded, all computing systems require memory — and in successively greater amounts as software becomes more sophisticated and versatile. As noted above, due to a huge installed base and traditional computer software techniques, disk technology is still the default memory medium for two of the three reprogrammable portable computing segments: laptop and notebook. This is in spite of several inherent drawbacks to disk drive technology: its propensity for failure when exposed to rugged or harsh environmental conditions, high power requirements, large size and its need for redundant, directly executable memory since disks inherently cannot provide direct access to their contents.

For palmtops, however, the memory scenario precludes disk drives because of their severe size, weight and power constraints, as well as exposure to the harshest user environment. Although laptop and notebook computers also face size and power constraints, their larger size allows them some latitude to incorporate disk technology if, in fact, it were able to provide the required levels of functionality and reliability. The conclusion drawn from this is that palmtop manufacturers have no choice but to be the earliest adopters of disk alternatives, and that the other segments also stand to benefit greatly from a suitable solution.

In the embedded world, many systems face demands similar to palmtops: They must be as small, lightweight and low-power as possible yet provide sufficient memory density for maximum performance and versatility. Further, because many embedded computers are used for industrial applications, disk drives cannot begin to withstand the inherent harsh environmental conditions.

What serves as a suitable alternative to disk technology in the portable computing environment? The strongest contender to date is a technology known as the solid-state memory card.

Several Means to a Common End

Memory cards are approximately the same width and length as a credit card and about four times as thick. Mounted inside the card may be as many as 16 memory ICs; therefore the card's total capacity is dependent upon both the density of the resident memory components as well as how many devices it contains.

Since their recent inception, IC memory cards have featured several different solid-state semiconductor technologies, each with its own particular strengths and weaknesses. For example, one of the first implementations employed ready-only memory, or ROM, devices. ROM technology is advantageous because of its status as a mature, high-volume, low-cost, reliable commodity product. It is also inherently nonvolatile (meaning that its contents remain intact even when power is removed), and is readily available in high densities. Its usefulness in a memory card format, however, is sorely limited since once manufactured, the card's contents can never be changed. As such, ROM cards can only be used for software that will never change, such as specific application programs. Although inexpensive at the outset, a ROM card can suddenly become a costly mistake if its contents ever do require updating — since it must then be discarded and replaced.

Another solid-state technology frequently used for memory cards is static RAM (random-access memory). Static RAM provides high-speed, virtually limitless reprogramming flexibility, making it a suitable medium for both data file and software program storage. Two fundamental characteristics limit its desirability as a memory card technology, however.

First, static RAM by itself is a volatile technology, requiring power at all times to retain its contents. Static RAM cards must therefore include a built-in battery as a back-up power source for those times when either system power shuts off or the card is removed from the system. Battery back-up introduces an element of unreliability since sooner or later it will fail, causing inconvenience at least, and disaster at worst.

Secondly, static RAM's performance comes at a high silicon technology cost. Static RAM cells require from four to six transistors to store one bit of information, increasing total silicon area, constraining the achievement of higher densities, and increasing cost.

To a lesser extent, some manufacturers have developed cards based on EEPROM (electrically erasable programmable ROM) or EPROM (erasable

programmable ROM) technology. Like static RAMs, EEPROMs provide update flexibility yet are nonvolatile, eliminating the need for battery back-up. However, EEPROM is a relatively complex technology that yields a larger die, lower density, higher cost and questionable reliability.

Cards based on EPROMs, also a nonvolatile technology, provide only slightly more flexibility than ROM cards. In contrast to ROM programming, which is accomplished through a time-consuming mask development process and cost-effective only in sufficient volumes, an OEM can program an EPROM-based card as needed and accommodate any changes or varying quantities up to that time. Once programmed, however, no further changes may be made by either the OEM or end user who is again faced with the distinct possibility of an obsolete card should a software change be required.

The Enabling Technology

As many system manufacturers have discovered at the component level, portable computer OEMs are now beginning to realize as they explore memory cards: Flash memory technology successfully combines the best traits of its predecessors. Specifically, Intel's ETOX™ Flash Memory technology is inherently nonvolatile and random access. Based on a single-transistor EPROM cell, ETOX Flash Memory technology achieves high densities in a comparatively small silicon area, facilitating high volume manufacturing and ensuring optimum cost-effectiveness.

Because flash memory's combination of features and characteristics make it an ideal solution for memory cards, Intel is now offering its high-density flash devices in this format. Intel Flash Memory cards conform to the PCMCIA/JEIDA (Personal Computer Memory Card International Association/Japan Electronic Industry Development Association) 68-pin connector standard to ensure that cards are compatible with both portable and desktop computers.

Disk Drives Falter in Face of Portability

While memory card density, flexibility and overall performance are greatly improved with the advent of flash memory, disk and disk drive manufacturers continue to advance their own technologies. One might wonder why anyone really needs an alternative.

As noted earlier, disk drives are prone to failure — particularly when subjected to the rough and tumble environments of portable computers. Because the flash memory card is hermetically sealed and solid-state, there are no moving parts, nor vulnerability to dust, moisture, shock or vibration, giving the memory card substantially higher reliability over disk drives. In fact, while the average disk drive would be severely damaged if its host computer were dropped even a short distance, experiencing a gravity force shock of approximately 3-5 Gs, testing has proven that Intel Flash Memory cards are undisturbed by as much as 50 Gs. Additionally, Intel's Flash Memory Card features approximately 1.6 million hours mean time between failure, or MTBF.

As for power, portable computing systems that rely on batteries as their power source require that it be used efficiently when necessary and conserved otherwise. Disk drives do neither particularly well. For example, during active mode, a 2.5" disk drive typically consumes 4 watts; the same drive still needs .5 watts during standby. In contrast, a system using flash memory cards would need no more than .15 watts when active and .04 watts during standby. Although a variety of other system-level factors determine overall power consumption, it is still clear that a flash-based system would operate more than 20 times longer than a system using a mechanical disk.

In addition to power consumption, however, are the considerations of performance and functionality. Even here, disk drives fall short in several key areas when compared to flash memory cards. Specifically, users cannot directly access data stored on a disk; that data must first be downloaded into system RAM. This process is made up of four steps: spin-up, seek time, latency and data transfer. In a benchmark test using a 16-MHz 386 PC AT*, accessing a 100 page file created on a popular word processing package from a hard disk takes longer than 11 seconds with a 1 Mbyte/second transfer rate, while a floppy disk, transferring data at .06 Mbyte/second, takes more than one minute.

In contrast, even though many system architectures today still require programs or files tored in a flash memory card to be downloaded into a high-speed RAM cache, a flash memory card can access the same 100 page file more than three times faster than a hard disk and 20 times faster than a floppy disk, while enjoying a data transfer rate of 8 Mbytes/second. Additionally, when writing this same file to the flash memory card, the test results showed a speed improvement of nearly 2X over the hard disk drive.

Probably the most obvious requirement of portable computing systems is that they be compact and lightweight. Particularly for the systems at the smallest end of the spectrum, disk drives thwart achievement of this goal. Whereas a credit-card sized medium can fit into virtually any hand-held machine, disk drives require significantly more space for the drive, the disk itself, the support circuitry and the system RAM needed to store downloaded programs and files. In addition, its higher power requirements necessitate a large system battery which again consumes more space and adds weight. For example, 2.5" disk drives weigh an average of 6.5 ounces. On the other hand, flash memory cards weigh in at approximately 1.0 ounce. Add in the weight of the large battery packs needed to run disk-based laptops, compared to small, "AA" batteries used in a flash-based system, and you have a much higher total system weight.

Unquestionably, the most desirable attribute of disk technology is its pure density. As a medium for the desktop computer, where users may store half a dozen application programs and years' worth of data files, today's 40+ Mbyte disk drives provide the optimum solution. But as soon as users need to transport their computing capability, pure density is no longer the most important factor. If we consider how most portable systems are actually used, we see that the above characteristics — ruggedness, low power, size/weight and performance — take precedence. Today, a user equipped with a few flash memory cards has sufficient capacity to store operating system software, several application programs and numerous data files. Further, as Intel continues to develop higher density products, potentially achieving a card capacity of 512 Mbytes

by the year 2000, density is no longer a limiting factor.

The other disk drive stronghold is software, software and more software. Nearly every piece of software in use today was written with disk storage and execution in mind. Microsoft*'s disk operating system, known as MS-DOS*, was developed to optimize the serial performance characteristics of disk drives and allow broad-based compatibility between systems and software.

Fortunately, Intel Flash Memory cards are supported by a ROM-executable version of DOS and flash file system software, assuring users access to the huge library of DOS-compatible software available today.

Intel's Flash Memory Card Solution

Beyond the ETOX Flash Memory card itself, Microsoft developed the Flash File System, or FFS. FFS is a DOS driver that allows a computer to read from or sequentially write to the flash memory card. With flash memory cards and FFS, the disk drive's time-consuming serial downloading process is eliminated. However, using traditional DOS commands, users may store and retrieve data or application software exactly as they would from a conventional disk drive with the only perceptible difference being faster access.

The final element of Intel's Flash Memory Card Solution are card drives. These computer peripheral devices, from companies such as Databook, are designed to read and write flash memory cards, allowing users to easily transfer programs and data files between their desktop and portable computers. Card drives are the link to truly achieving "mobile office" capability.

As flash technology continues to increase in density, flash memory cards stand to reach a capacity of 128 megabytes by 1997. Such density will allow users to store successively larger files, more sophisticated application programs and multiple operating systems. Combined, Intel's flash memory card, FFS and the card drive comprise the optimum memory solution for today's portable computing systems while pointing the way to the ultimate mobile office of tomorrow.

*Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

*PC AT is a registered trademark of IBM.

November 1992

4

Intel FlashFile™ Memory The Key to Diskless Mobile PCs

JANET WOODWORTH
MEMORY COMPONENTS DIVISION

Order Number: 297115-001

INTRODUCTION

As the PC evolves into what is truly a "personal" computer—one that can be held in your hand—a completely different system memory architecture will emerge. Step aside ROM, DRAM, floppy disk, and hard disk; Intel's FlashFile™ memory is here. FlashFile memory will finally make it possible to build a thin, 2-pound notebook computer that runs for many hours on a few AA batteries. But before these mobile PCs are built, designers must learn some new ways to configure system memory.

In April 1992, Intel introduced a new flash memory architecture with a combination of functionality and price that redefines mobile computing. This new architecture, when implemented in new system memory configurations, enables nonvolatile executable system memory and removable file and program storage. Intel's new flash architecture lets designers create a portfolio of products that will clearly differentiate them from their competition.

WHY A NEW MEMORY ARCHITECTURE?

The ideal memory system is:

- Dense (stores lots of code and data in a small amount of space and weighs very little)
- Fast (lets you read and write data quickly)
- Inexpensive (low cost per megabyte)
- Nonvolatile (data remains when power is removed)
- Power Conscious (prolongs battery life and reduces heat)
- Reliable (retains data when exposed to extreme temperature and mechanical shock)

Since PCs were introduced over 10 years ago, designers have grappled with how to construct memory systems that offer all these attributes. They have wisely elected to use to optimum combination of solid-state memory and magnetic storage, i.e., DRAMs plus magnetic hard disks. DRAMs are dense and inexpensive, yet slower than the processors they serve, and they are volatile. SRAMs are used in caching schemes to compensate for DRAM's slowness. While SRAMs keep pace with today's high-performance microprocessors, they are not as dense as DRAM, are inherently more expensive, and volatile. Magnetic hard disks are very dense, inexpensive on a cost-per-megabyte basis, and nonvolatile, but they are painfully slow, power hungry and subject to damage from physical shock.

ENTER FLASH MEMORY

Because Intel's ETOX™ III flash memory cell is 30-percent smaller than equivalent DRAM cells, the company expects it to track DRAM density closely. Intel's new 28F008SA FlashFile Memory can store 8 megabits, or one megabyte, of data today. Flash memory is more scalable than DRAM due to its simple cell structure, so as DRAM technology shrinks towards 0.25 microns and 64 megabits, flash will pace and ultimately overtake DRAM's technology treadmill. In fact, expect to see 256-Mbit flash memory by the end of the '90s.

FLASH MEMORY IS FAST

Don't be misled by technology-to-technology speed comparisons. Designing your system memory around flash will break the code/data bottleneck created by connecting a mechanical memory such as disks to a high-performance electronic system. For instance, data seek time for a 1.8″ magnetic hard disk is 20 ms, plus an 8 ms average rotational delay, while flash is less than 0.1 ms. At the chip level, current read speeds for flash are about 90 ns. Thus, downloading from flash to system RAM or directly executing from flash will dramatically enhance system speed.

FLASH MEMORY IS INEXPENSIVE

At the 8-Mbit density, Intel flash pricing matches DRAM and Intel expects to continue decreasing price as both densities and volumes increase.

FLASH IS NONVOLATILE

Unlike SRAM or pseudo-SRAM (SRAM with built-in battery), flash needs no battery backup. Further, Intel's flash devices retain data typically for over 100 years, well beyond the useful lifetime of even the most advanced computer.

FLASH IS POWER CONSCIOUS

FlashFile Memory in a hard-disk drive configuration consumes less than one two-hundredth the average power of a comparable magnetic disk drive based on the typical user model. At the chip level, the 28F008SA has a DEEP POWERDOWN mode that reduces power consumption to less than 0.2 μA.

FLASH IS RUGGED AND RELIABLE

On average, today's hard-disk drives can withstand up to 10 Gs of operating shock; Intel FlashFile memory can withstand as much as 1000 Gs. FlashFile compo-

nents can operate at up to 70°C while magnetic drives are limited to 55°C. Intel FlashFile memory can be cycled 100,000 times per block or segment. By employing wear-leveling techniques, a 20-Mbyte flash array can provide over 30 million hours before failure.

WHY NOW?

Flash memory is not a new technology. Intel has been the flash technology and market leader since 1988. Then why hasn't flash taken the mobile PC market by storm yet? Why now?

One reason that 1992 is the pivotal year for flash-based systems is the sharply increased demand for highly mobile computers. The other reason is that a number of key capabilities, in development for some time, reached maturity together.

1. Intel Introduces FlashFile™ Memory

MS-DOS*, the ubiquitous operating system for PCs, was developed specifically to optimize the sectoring scheme inherent to disk technology. Intel's first generation "bulk-erase" flash required that all of the chip be erased before data could be re-written: a natural fit for updatable firmware and data acquisition, but not for data file storage or disk emulation. Intel FlashFile memory, based on a block-erase architecture, divides the flash memory space into segments that are somewhat analogous to the zones recognized by MS-DOS. For instance, the Intel 28F008SA contains sixteen identical, individually-erasable, 64-Kbyte blocks. This organization has been carefully optimized to maximize cycling capability while preserving the smallest granularity possible. The ability to segment block memory into individual segments allows disk-like data-file storage.

2. Standardization of Delivery System and Interface

Thanks to work by the Personal Computer Memory Card International Association (PCMCIA), and the Japanese Electronics Industry Development Association (JEIDA), there is now an internationally recognized standard for memory cards. PCMCIA cards are the size of a business card but about four times as thick. Intel is widely promulgating its Exchangeable Card Architecture (ExCA™), a hardware and software implementation of the PCMCIA system interface. When used with the proper BIOS, ExCA/PCMCIA-compatible cards will be completely interchangeable between systems and vendors, and can be equated to solid-state floppy disks, albeit with many advantages.

Flash-based solid-state disks, intended to replace magnetic hard disks in certain applications, with IDE interfaces will be "plug compatible" with existing systems that are already designed with IDE magnetic drives in mind.

3. Flash File System

Intel has worked very closely with Microsoft* to implement a DOS flash memory extension called Flash File System (FFS) that transparently handles swapping of data between flash blocks, much as DOS now handles swaps between disk sectors. With Flash File System, the user inputs a DOS comand and doesn't need to think about whether a magnetic disk or a flash memory is being used. Flash File System employs wear leveling algorithms that prevent any block from being cycled excessively, thus ensuring millions of hours of use across multiple chips.

4. Off-the-Shelf Hardware Interface

The introduction of the Intel 82365SL™ PC Card Interface Controller provides a ready-made interface between the PC's ISA bus and up to two PCMCIA sockets. It is a key component for memory and I/O card implementations since the designer is relieved from building the interface from scratch.

5. Cost Reductions

Magnetic drives do not scale well; that is, it becomes increasingly difficult to improve or even retain density as platter size shrinks. Thus, every reduction in drive size requires complete retooling and costly learning. Also, the complex controller circuitry provides a price floor under which magnetic drives cannot drop. Since flash is scalable, at some point in the near future, small magnetic drives are likely to become more expensive per megabyte than flash cards and are certain to have less capacity. But even today, the value of a particular memory technology is a result of more than just dollars per megabyte.

Notes market analysis expert Dataquest:

> "The question is, "Can you put a floppy disk drive in a palmtop PC to take advantage of the cost disparity (between disk and flash)?" The answer is, 'No.' There is not enough power (or space). The issue then, is not cost. Here, the removable storage medium dictates the product's capabilities and its success or failure in the marketplace. Without a memory card, a palmtop is nothing more than an electronic organizer. It is the memory card that transforms a palmtop into a full-fledged personal computer." . . . Nick Samaras, SAMS Newsletter.

All of the aforementioned features, Intel FlashFile memory's block-erase architecture, PCMCIA standards, ExCA, Flash File System, 82365SL ISA-PCMCIA interface controller and reduced costs, are deliverables ... now. And not a moment too soon based on the tremendous market opportunity created by the increasing demand for truly mobile computers. Dataquest predicts that the worldwide sale of portable PCs will increase from six million units in 1992 to nearly 30 million units in 1995. While laptop PCs are only expected to increase by about two million units, notebooks, pen-based, and handheld PCs will increase from three million units in 1992 to nearly 25 million in 1995, an eight-fold increase. This extraordinary growth will be greatly assisted by FlashFile memory.

ENABLING THE TRULY MOBILE COMPUTER

In the world of the desktop PC, DRAM is used for executable code storage and data manipulation. Since DRAM is volatile, if power is lost, both programs and data are lost, hence the need for a nonvolatile magnetic hard disk. With the addition of the hard disk, programs and data are stored on the hard disk and swapped in and out of DRAM as needed. Some part of the DRAM is reserved for use as a register to store temporary results during compute-intensive operations. Today's PCs are typically configured with 4 megabytes of DRAM and at least a 40-Mbyte disk.

FlashFile memory fully supports this system configuration when used simply as a magnetic drive replacement. Instructions and data are still swapped to DRAM but at a much faster rate. Plus, execution speed can be enhanced if the DRAM is replaced with SRAM.

In the solid-state computer, the "DRAM + magnetic hard drive" are replaced by a "flash memory + SRAM". The key to this architecture is the ability to eXecute-InPlace (XIP). Program instructions stored in the flash memory are read directly by the processor. Results are written directly to the flash memory. Compute-intensive operations that require the fastest memory and byte-alterability use high-speed SRAM or pseudo SRAM. Most of what we now think of as the "DRAM" is replaced by low-cost flash and only a relatively small part of the DRAM is replaced by SRAM.

The flash memory space is made even more storage-efficient through the use of compression techniques which offer at least 2:1 compression. For example, one 20-Mbyte flash card that uses 2:1 compression offers the same storage as a 40-Mbyte hard disk!

The advantages of a flash-based computer include:

- Blazingly fast speed
- Instant-on and instant-resume
- Ultra-light PC (2–4 lbs.)
- Very secure data retention
- Flexible firmware

As you can see, by changing the system memory architecture to a flash-based one, designers will be able to build a new generation of PCs that meets the needs of the computer user of both today and tomorrow.

Progress has been made toward implementing this approach with the introduction of Hewlett-Packard's successful HP95LX DOS-compatible palmtop. MS-DOS and Lotus 1-2-3* are stored in ROM. Internally, pseudo-static RAM is used, and a PCMCIA memory socket is provided. Lotus 1-2-3 was re-written to allow ROM-based storage so it could execute in place. Other ROM-executable versions of popular operating systems are expected to be available shortly.

DESIGNING YOUR SYSTEM WITH FLASHFILE MEMORY

Details of the three Intel flash applications and implementations—flash cards, silicon disks, and Resident Flash Array (RFA)—are presented below.

APPLICATION NUMBER 1: MEMORY CARDS

Memory cards are the most rugged and reliable of the removable memory media. A card can be slipped into a shirt pocket and moved from location to location. With high-density flash cards, you can download files from the desktop and use the card in your notebook or palmtop.

Memory cards have been around for some time. The first cards to be introduced were ROM-only cards used in video games and pocket organizers. These were produced in various formats prior to the formation of PCMCIA and JEIDA. Later cards included battery-backed SRAM and EEPROM. Neither became very popular due to their high cost of $500–$600 per megabyte and limited capacity. Flash cards overcome the cost barrier and they are certain to be multiply sourced, assuring availability and competitive pricing. A 20-Mbyte flash card has three times the real storage density of a 20-Mbyte 1.8″ magnetic drive (0.95 Mbyte/cm^3 vs 0.34 Mbyte/cm^3) and it has 10 times the weight density (2 Mbyte/gm vs 0.21 Mbyte/gm). The PCMCIA has complete industry support, and enhanced versions, such as PCMCIA Version 2.0, are designed to be backward-compatible with earlier versions.

As part of its flash product family, Intel's new Series 2 memory cards are the first to utilize chips processed on its 0.8-micron flash technology. Storing up to 20 megabytes, these cards are designated Series 2 to differentiate them from the earlier bulk-erase flash cards. The cards consist of 4 to 20, 28F008SA TSOP FlashFile memory devices. Each 28F008SA contains 16 distinct, individually-erasable, 64-Kbyte blocks. Therefore, each card contains from 64 to 320 blocks.

With the release of PCMCIA Version 2.0 in September of 1991, the PCMCIA-compatible field grew somewhat larger. The PCMCIA interface grew from memory-only to one that supports many types of I/O devices. Intel's system-level implementation of PCMCIA 2.0, called ExCA, ensures that if there are two ExCA sockets available, one can be used for a flash memory card and one for a modem; and the cards may be interchanged.

How difficult is it to design-in an ExCA socket? Not very. Intel's open ExCA specification details the system implementation. Other than the physical incorporation of the socket and card, the only required hardware is an ISA-to-PCMCIA interface such as Intel's 82365SL chip, and an ExCA compliant BIOS from vendors like SystemSoft, Award and Phoenix. You'll also need a flash file management system like Microsoft's Flash File System. Intel's block-erase architecture, along with the DOS Filing System and ExCA BIOS, makes it easy to incorporate ExCA features. In addition, ExCA-compliant systems will allow system-to-system interoperability much like floppy disks.

APPLICATION NUMBER 2: FLASH-BASED SOLID STATE DISK

The implementation of block-erase flash as a "solid-state disk" (SSD) is something of a misnomer. It is not a disk at all, rather a flash module that has the same form, fit and function as a 2.5″ or smaller magnetic drive.

A flash-based SSD implementation is the most direct route to adapting flash to an existing design. A built-in IDE interface would make it plug-compatible. But what a difference a silicon disk will make! A 1.8″ drive typically uses one watt-hour/hour while a silicon disk uses as little as 0.035 watt-hour/hour. This kind of power savings makes it possible to reduce battery size and weight considerably. Or, consider reliability. We've already discussed differences in susceptibility to shock and temperature extremes. In addition, an SSD theoretically has a mean-time-between-failure (MBTF) of 250,000 hours, compared to 100,000 hours for the magnetic drive.

With all these advantages, when should you use memory cards and when is use of a flash-based SSD preferable?

First and foremost, the SSD is considered to be installable while memory cards are removable and transportable. In other words, the SSD is meant to be installed and then left alone, while memory cards are designed for constant removal and reinsertion. In operation, the only change a user would notice in a notebook computer equipped with a flash-based SSD is that access speed is unprecedented.

The flash-based solid-state drive is one very good way to get to market early with flash technology. In February 1992, Conner Peripherals, Inc., and Intel announced the signing of a joint product and technology development contract focused on designing and bringing to market proprietary FlashFile memory-based SSD storage products.

Incidentally, manufacturers of magnetic drives are starting to take notice. In a manner much like the tail wagging the dog, 1.8" magnetic hard disks with PCMCIA interfaces are currently being developed.

APPLICATION NUMBER 3: RESIDENT FLASH ARRAY

The one approach that offers totally new capabilities is the Resident Flash Array (RFA). This is an arrangement of from 8 to 20, 8-Mbit FlashFile memories. In the long term, it replaces some of the motherboard's DRAM. This is the approach that is applicable to all levels of PC, from desktop to palmtop. For near-term applications, however, RFA is an ideal way of making code or ROM-executable operating systems such as DOS or Windows* updatable to protect the end-user's software investment. Also, when used as a resident application program and data-file storage medium on the local memory bus, RFA provides a high-performance, low-power solution.

The Resident Flash Array provides the highest possible performance of any option, especially since the processor can be closely coupled to it; and hence, would not be encumbered by IDE or PCMCIA interfaces, or even the ISA bus itself. The flash memory and the processor will sit side-by-side.

The proliferation of flash memory card-based systems will accelerate the process of converting disk-oriented applications to a flash-executable orientation. Those manufacturers who elect to be early adopters of Intel FlashFile memory will be able to develop a new generation of PC—the truly "personal" computer you can hold in your hand.

NOTE:

ETOX, ExCA and FlashFile are trademarks of Intel Corporation.
*Microsoft and MS-DOS are registered trademarks; Windows is a trademark of Microsoft Corporation.
*Lotus and 1-2-3 are registered trademarks of Lotus Development Corporation.

Intel Flash Memory Card iMC001FLKA/002FLKA/004FLKA

The Intel Flash Memory Card is the optimal integrated memory solution for the portable PC environment. It offers the ability to store and revise application code as well as store sequential data files on an all-silicon memory media. The Flash Memory Card supports the PCMCIA/JEIDA standard 68-pin connector, allowing code and data file transportability between portable and desktop PCs. Using the Intel Flash Memory Card with Independent Software Vendor's (ISV) flash filing systems, OEM system manufacturers can now provide the end-user with more rugged portable PCs that perform at higher levels, consume less power and weigh much less than traditional disk-based portable PCs.

Intel Flash Memory Cards

Product Highlights

- **Inherent nonvolatility—no batteries required for data/code retention**
- **Low power CHMOS* technology**
- **High-performance read access of 200 ns**
- **Flash electrical zone-erase**
- **Write protect switch to prevent accidental data corruption**
- **More reliable than disk**
- **Command register architecture for microprocessor/microcontroller compatible write interface**
- **ETOX™ II one-micron Flash Memory technology—5 volt read, 12 volt erase/write, high-volume manufacturing experience**
- **PCMCIA/JEIDA standard 68-pin connector**

Product Description

The functionality of the 1 Megabyte (iMC001FLKA), 2 Megabyte (iMC002FLKA) and 4 Megabyte (iMC004FLKA) Flash Memory Cards are based on Intel's current product line of ETOX II nonvolatile Flash Memories. Each memory card consists of an array of Intel Flash Memories in TSOP and two ASIC chips that buffer address/data lines, decode zone $\overline{CE}$, and provide system interfacing control logic and timing for $\overline{WE}/\overline{CE}/\overline{OE}$.

The Intel Flash Memory Card conforms to the PCMCIA/JEIDA 68-pin standard for memory cards. The cards dimensions are 85.6 mm x 54.0 mm x 3.3 mm.

The Intel Flash Memory Card is ideal for storing application code where the end-user wants the flexibility to reconfigure the portable PC application software. Microsoft's** Flash File System (FFS) software facilitates sequential data file storage, card erasure, and when coupled with the Intel Memory Card, effectively creates an all-silicon system that is more durable, reliable and lasts longer than disk based memory systems.

*CHMOS is a patented process of Intel Corporation
**Microsoft is a trademark of Microsoft Corporation.

Order Number 296840-003

Intel Series 2 Flash Memory Cards iMC004FLSA, iMC010FLSA, iMC020FLSA

Intel's second generation Flash Memory Cards feature block erasure providing data file and application storage in removable nonvolatile memory. Series 2 Flashcards enable the design of diskless sub-notebook, palmtop, and pen-based PCs that have the processing power of today's desktop computers but weigh under 4 pounds (1.8 Kilograms) and can run for an extended time on a set of "AA" batteries. The Flashcards weigh little, extend system battery life and withstand rugged environments where small form factor rotating disks are not suitable or just can't fit. The cards adhere to the PCMCIA 2.0/JEIDA

297102-1

Product Highlights

- Flash Block-Electrical Erase
  - Enables High-Performance Disk Emulation
- 4, 10 and 20 MByte Versions of Addressable Flash
  - High Capacity in a Very Small Form Factor
- PCMCIA 2.0/JEIDA 4.1 Standard 68-Pin Interface
  - Provides System-to-System Interchangeability
- ExCA™ Compatible for System-to-System Interoperability
- Byte or Word-Wide Operation
- Ready/Busy Signal in Conjunction with Automatic Erase/Write
- Erase Suspend/Resume
- Inherent Nonvolatility—No Batteries Required
- Solid-State Reliability
- ETOX™ III 0.8 Micron Flash Memory Technology—3.3/5.0V V_{CC}, 12V V_{PP}, High-Volume Manufacturing Experience

Product Description

The functionality of the Series 2 Flashcards are based on Intel's ETOX III 8 Megabit FlashFile™ Memories. Each memory card consists of a PCMCIA connector, an array of 28F008SAs in TSOP and two card control logic devices. The control logic devices provide the system interface and control the internal flash memories.

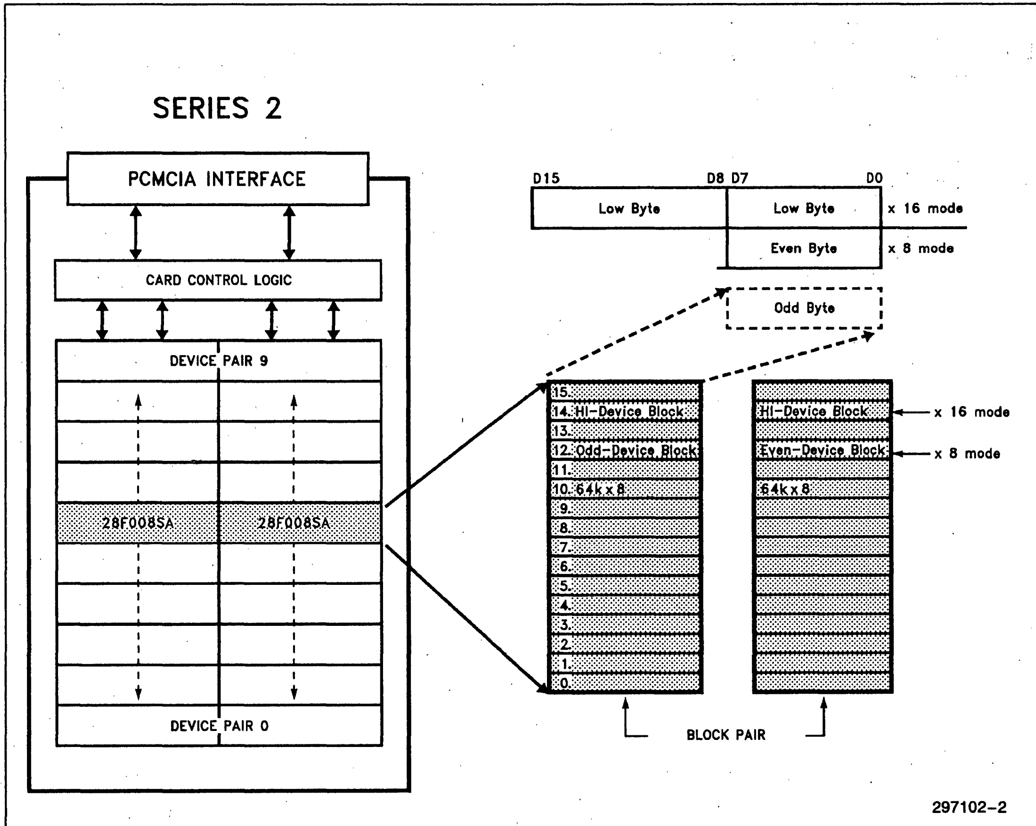
The card features fast random access, Byte-wide and Word-wide operation, low power modes and a Card Information Structure (CIS) for easy identification of card characteristics.

The flashcard's solid-state characteristics, small form factor and nonvolatility make it ideal for applications where rugged removable firmware and data storage are a requirement.

Combined with file management software, such as Microsoft's* Flash File System, Series 2 Flashcards provide removable high-performance disk emulation in mobile PCs.

ExCA™, ETOX™, and FlashFile™ are trademarks of Intel Corporation.

*Microsoft is a trademark of Microsoft Corporation.



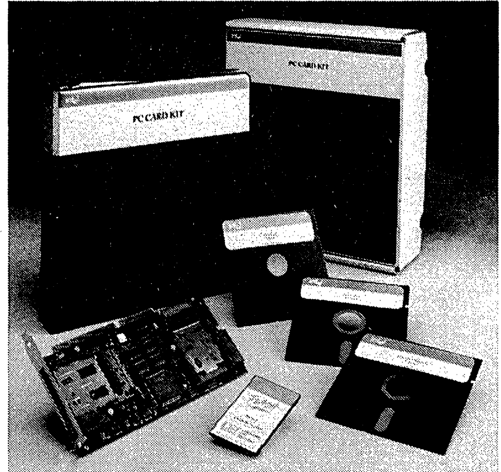
Key Specifications

- — 200 ns Random Read Access
- — 10 μs Typical Byte Write
- — 1.5 Second Typical Block Erase
- — PCMCA Type I Card (85.6mm × 54.0mm × 3.3mm)
- — 100K Cycles per Block
- — 3.3V/5V V_{CC}/12V V_{PP}

PRODUCT BRIEF

Intel PC CARD Kit Product Brief

Intel's PC CARD Developer's Kit provides the mobile computer designer the hardware, software and system interface to evaluate the functionality and benefits of Intel's Series 2 Flash Memory Cards. An ExCA™ add-in board in an IBM PC-AT* desktop PC using MS-DOS serves as the development platform. The hardware in the PC CARD Kit is compliant with Intel's Exchangeable Card Architecture. The PC CARD Kit supersedes Intel's Flash Memory Systems Developer's Kit (iMSDK001FLKA).**



297159–1

Kit Contents

- **One 4-Megabyte Intel Series 2 Flash Memory Card**
- **One PC-AT ExCA Developer's Board with two PCMCIA/JEIDA 68-Pin Interface Connectors**
- **Details on how to Obtain an Evaluation Copy of Microsoft's** Flash File System Software**
- **PC CARD Developer's Kit User's Guide**
- **Technical Documentation Describing Intel's Series 2 Flash Memory Cards and FlashFile™ Components**
- **A Copy of Intel's ExCA Specification**
- **82365SL Diagnostic Software**
- **PC Card Developer's Kit Registration Certificate**

Kit Description

A PC-AT add-in board based on ExCA hardware, two PCMCIA slots, Microsoft's Flash File System, one Series 2 Flashcard, and associated Flash driver and diagnostic software provide a hands-on evaluation/development tool for mobile computer designs.

The Kit User's Guide provides hardware and software installation instructions and advice on how to maximize kit usage.

This kit will enable systems designers to:

1. Become familiar with Solid State Disk Emulation using the Series 2 Flash Memory Card
2. Write DOS data files and programs to the Flashcard using Microsoft's Flash File System
3. Evaluate Intel's Exchangeable Card Architecture (ExCA)
4. Design Intel's Series 2 Flashcards into your next mobile computer design.

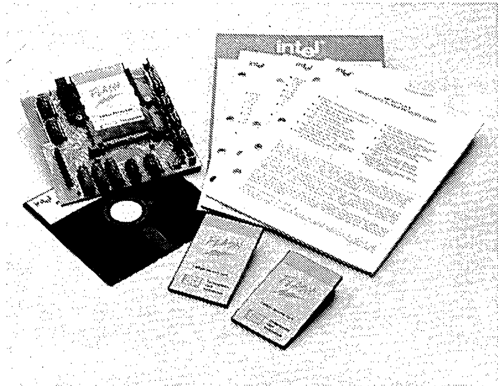
The ExCA Developer's Board will accept software upgrades that allow the integration of future Intel Flash Memory and I/O Cards as they become available.

*PC-AT is a trademark of IBM Corporation.
**Microsoft and MS-DOS are trademarks of Microsoft Corporation.
ExCA and FlashFile are trademarks of Intel Corporation.

Order Number: 297159-001

Intel Flash Memory Card/SIMM Evaluation Module (D, FLASHEVAL3)

Intel's Flash Memory Card/SIMM Evaluation Module (D, FLASHEVAL3), provides the system designer with a cost-effective prototyping development tool. This module is a plug in board, used with the D, FLASHEVAL2 Kit. It allows the user to become familiar with the functionality of Intel's Memory Card and SIMM with the ability to edit, read, program and erase Hex/Binary files. The D, FLASHEVAL3 module also offers low volume programming and program verification during system prototyping. Intel will continue to offer ongoing support for new densities and additional capabilities by providing software updates to registered kit users.



297156-1

Kit Contents

- **(1) Memory Card/SIMM Socket Module**
- **(1) iFLASH2 Software Diskette Version 2.0**
- **(1) iFLASH2 User's Manual**
- **(1) 1 MByte Intel Flash Memory Card**
- **(1) 1 MByte Intel SIMM**
- **(1) Registration Card**
- **Technical Documentation**

Kit Description

The Intel Flash Memory Card SIMM Evaluation Module (D, FLASHEVAL3) provides a PC-driven flash memory programming solution.

The Intel D, FLASHEVAL3 module, used with the D, FLASHEVAL2 Kit, provides the hardware, software and system interface necessary to evaluate and integrate the Intel Flash Memory Card and/or SIMM solution into your next design.

The Kit User's Manual provides extensive hardware and software installation instructions to program, erase and read HEX/Binary files on Memory Cards and SIMMs.

4

NOTE:
D, FLASHEVAL3 must be used in conjunction with the D, FLASHEVAL2 hardware.

Order Number: 297156-002

Automotive Memory

5

A28F256A
256K (32K x 8) CMOS FLASH MEMORY
Automotive

- **Extended Automotive Temperature Range −40°C to +125°C**
- **Flash Electrical Chip-Erase**
  - **1 Second Typical Chip-Erase**
- **Quick-Pulse Programming™ Algorithm**
  - **10 μs Typical Byte-Program**
  - **0.5 Second Chip-Program**
- **1,000 Erase/Program Cycles Minimum Over Automotive Temperature Range**
- **12.0V ±5% V_{PP}**
- **High-Performance Read**
  - **120/150 ns Maximum Access Time**
- **CMOS Low Power Consumption**
  - **30 mA Maximum Active Current**
  - **100 μA Maximum Standby Current**
- **Integrated Program/Erase Stop Timer**
- **Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface**
- **Noise Immunity Features**
  - **±10% V_{CC} Tolerance**
  - **Maximum Latch-Up Immunity through EPI Processing**
- **ETOX™-II Flash Nonvolatile Memory Technology**
  - **EPROM-Compatible Process Base**
  - **High-Volume Manufacturing Experience**
- **JEDEC-Standard Pinouts**
  - **32-Pin Plastic-DIP**
  - **32-Lead PLCC**

(See Packaging Spec., Order #231369)

Intel's 28F256A CMOS flash-memory offers the most cost-effective and reliable alternative for updatable non-volatile memory. The 28F256A adds electrical chip-erasure and reprogramming to familiar EPROM technology. Memory contents can be erased and reprogrammed: in a test socket; in a PROM-programmer socket; on-board during subassembly test; in-system during final test; and in-system after-sale. The 28F256A increases memory flexibility, while contributing to time- and cost-savings. The 28F256A is targeted for alterable code- or data-storage applications where traditional EEPROM functionality (byte-erasure) is either not required or not cost-effective. The 28F256A can also be applied where EPROM ultraviolet erasure is impractical or time-consuming.

The 28F256A is a 256-kilobit non-volatile memory organized as 32768 bytes of 8 bits. Intel's 28F256A is offered in 32-pin Plastic-DIP or 32-lead PLCC packages. Pin assignments conform to JEDEC standards.

Extended erase and program cycling capability is designed into Intel's ETOX™-II (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional EEPROMs. With the 12.0V V_{PP} supply, the 28F256A performs a minimum of 1,000 erase and program cycles well within the time limits of the Quick-Pulse Programming™ and Quick-Erase™ algorithms.

Intel's 28F256A employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 150 nanosecond access time provides no-WAIT-state performance for a wide range of microprocessors and microcontrollers. Maximum standby current of 100 μA translates into power savings when the device is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from −1V to V_{CC} + 1V.

With Intel's ETOX-II process base, the 28F256A levers years of EPROM experience to yield the highest levels of quality, reliability, and cost-effectiveness.

In order to meet the rigorous environmental requirements of Automotive Applications, Intel offers the 28F256A in extended Automotive temperature range. Read and Write Characteristics are guaranteed over the range of −40°C to +125°C ambient.

September 1992
Order Number: 290168-006

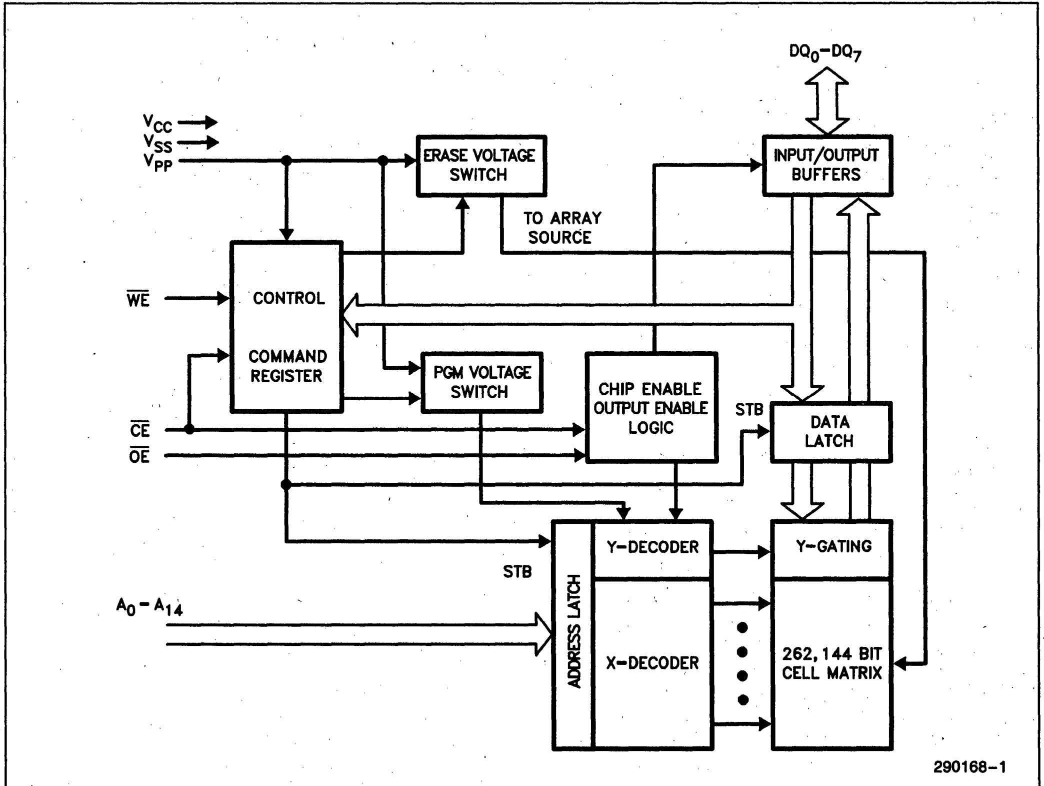


Figure 1. 28F256A Block Diagram

AUTOMOTIVE TEMPERATURE FLASH MEMORIES

The Intel Automotive Flash memories have received additional processing to enhance product characteristics. The Automotive temperature range is −40°C to +125°C during the read/write/erase/program operations.

Speed Versions	Packaging Options	
	Plastic-DIP	PLCC
-120	AP	AN
-150	AP	AN

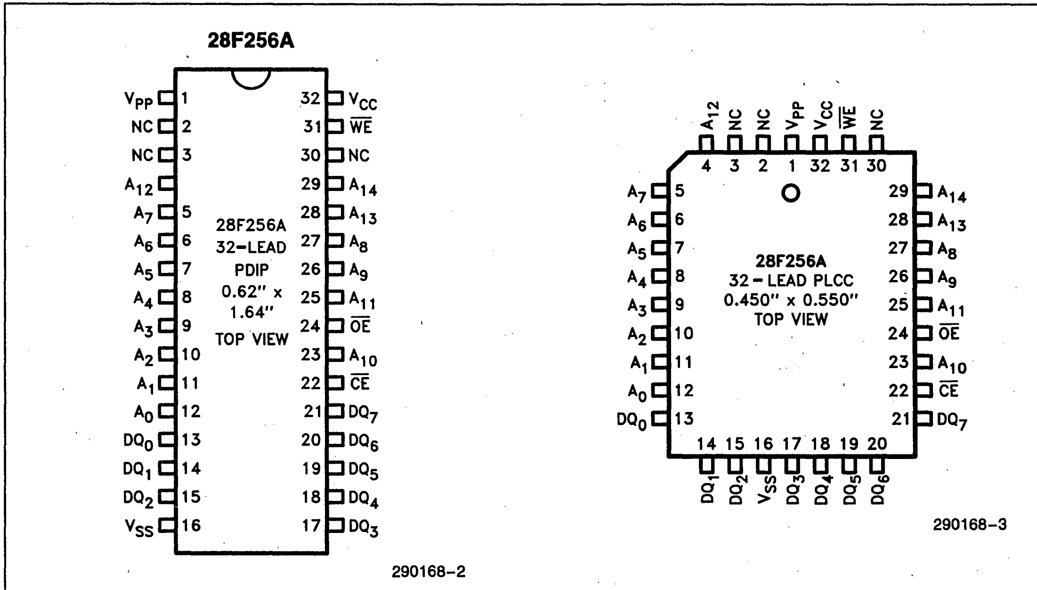


Figure 2. 28F256A Pin Configurations

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{14}	INPUT	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
DQ_0–DQ_7	INPUT/OUTPUT	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}$	INPUT	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels.
$\overline{OE}$	INPUT	**OUTPUT ENABLE:** Gates the devices output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	INPUT	**WRITE ENABLE:** Controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **Note:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP}		**ERASE/PROGRAM POWER SUPPLY** for writing the command register, erasing the entire array, or programming bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY** (5V ±10%)
V_{SS}		**GROUND**
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating.

APPLICATIONS

The 28F256A flash-memory adds electrical chip-erasure and reprogrammability to EPROM non-volatility and ease of use. The 28F256A is ideal for storing code or data-tables in applications where periodic updates are required. The 28F256A also serves as a dense, nonvolatile data acquisition and storage medium.

The need for code updates pervades all phases of a system's life—from prototyping to system manufacture to after-sale service. In the factory, during prototyping, revisions to control code necessitate ultraviolet erasure and reprogramming of EPROM-based prototype codes. The 28F256A replaces the 15- to 20-minute ultraviolet erasure with one-second electrical erasure. Electrical chip-erasure and reprogramming occur in the same workstation or PROM-programmer socket.

Diagnostics, performed at subassembly or final assembly stages, often require the socketing of EPROMs. Socketed test codes are ultimately replaced with EPROMs containing the final program. With electrical chip-erasure and reprogramming, the 28F256A is soldered to the circuit board. Test codes are programmed into the 28F256A as it resides on the circuit board. Ultimately, the final code can be downloaded to the device. The 28F256A's in-circuit alterability eliminates unnecessary handling and less-reliable socketed connections, while adding greater test flexibility.

Material and labor costs associated with code changes increase at higher levels of system integration—the most costly being code updates after sale. Code "bugs", or the desire to augment system functionality, prompt after-sale code updates. Field revisions to EPROM-based code require the removal of EPROM components or entire boards.

Designing with the in-circuit alterable 28F256A eliminates socketed memories, reduces overall material costs, and drastically cuts the labor costs associated with code updates. With the 28F256A, code updates are implemented locally via an edge-connector, or remotely over a serial communication link.

The 28F256A's electrical chip-erasure, byte reprogrammability, and complete nonvolatility fit well with data accumulation needs. Electrical chip-erasure gives the designer a "blank-slate" in which to log data. Data can be periodically off-loaded for analysis—erasing the slate and repeating the cycle. Or, multiple devices can maintain a "rolling window" of accumulated data.

A high degree of on-chip feature integration simplifies memory-to-processor interfacing. Figure 3 illustrates the interface between the MCS®-51 microcontroller and one 28F256A flash-memory in a minimum chip-count system. Figure 4 depicts two 28F256As tied to the 80C186 system bus. In both instances, the 28F256A's architecture minimizes interface circuitry needed for complete in-circuit updates of memory contents. (Comprehensive system design information is included in AP-316, "Using

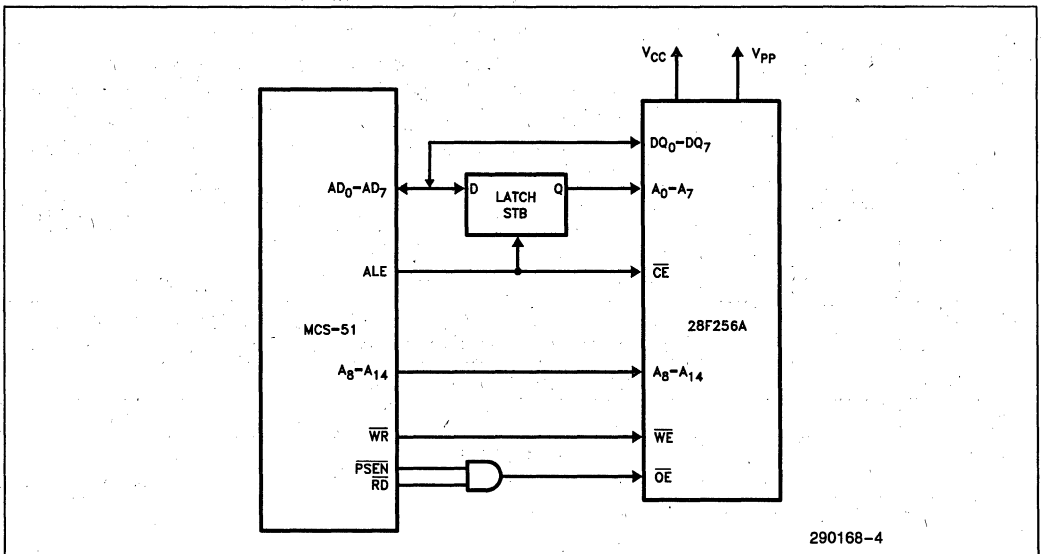


Figure 3. 28F256A in an MCS®-51 System

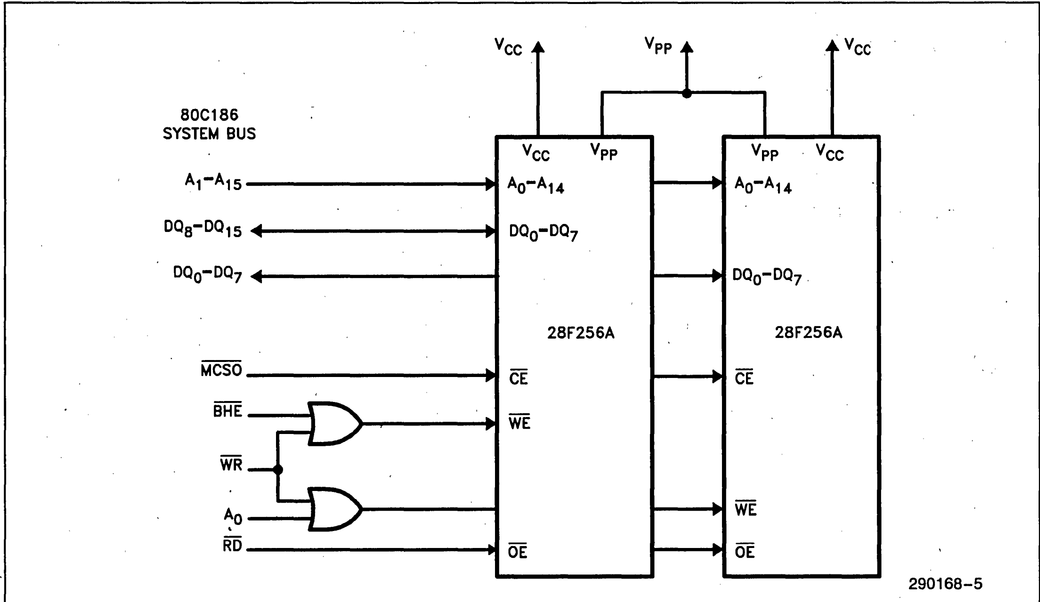


Figure 4. 28F256A in an 80C186 System

the 28F256A Flash Memory for In-System Reprogrammable Nonvolatile Storage", Order Number 292046-002).

With cost-effective in-system reprogramming and extended cycling capability, the 28F256A fills the functionality gap between traditional EPROMs and E^2PROMs. EPROM-compatible specifications, straightforward interfacing, and in-circuit alterability allows designers to easily augment memory flexibility and satisfy the need for updatable nonvolatile storage in today's designs.

PRINCIPLES OF OPERATION

Flash-memory augments EPROM functionality with in-circuit electrical erasure and reprogramming. The 28F256A introduces a command register to manage this new functionality. The command register allows for: 100% TTL-level control inputs; fixed power supplies during erasure and programming; and maximum EPROM compatibility.

In the absence of high voltage on the V_{PP} pin, the 28F256A is a read-only memory. Manipulation of the external memory-control pins yields the standard EPROM read, standby, output disable, and intelligent Identifier operations.

The same EPROM read, standby, and output disable operations are available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} enables erasure and programming of the device. All functions associated with altering memory contents—intelligent Identifier, erase, erase verify, program, and program verify—are accessed via the command register.

Commands are written to the register using standard microprocessor write timings. Register contents serve as input to an internal state-machine which controls the erase and programming circuitry. Write cycles also internally latch addresses and data needed for programming or erase operations. With the appropriate command written to the register, standard microprocessor read timings output array data, access the intelligent Identifier codes, or output data for erase and program verification.

Integrated Program/Erase Stop Timer

Successive command write cycles define the durations of program and erase operations; specifically, the program or erase time durations are normally terminated by associated program or erase verify commands. An integrated stop timer provides simplified timing control over these operations; thus eliminating the need for maximum program/erase timing specifications. Programming and erase pulse durations are minimums only. When the stop timer terminates a program or erase operation, the device enters an inactive state and remains inactive until receiving the appropriate verify or reset command.

Table 2. 28F256A Bus Operations

Operation		V_{PP}(1)	A_0	A_9	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_0–DQ_7
READ-ONLY	Read	V_{PPL}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out
	Output Disable	V_{PPL}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby	V_{PPL}	X	X	V_{IH}	X	X	Tri-State
	intelligent ID Manufacturer(2)	V_{PPL}	V_{IL}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = 89H
	intelligent ID Device(2)	V_{PPL}	V_{IH}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = B9H
READ/WRITE	Read	V_{PPH}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out(4)
	Output Disable	V_{PPH}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby(5)	V_{PPH}	X	X	V_{IH}	X	X	Tri-State
	Write	V_{PPH}	A_0	A_9	V_{IL}	V_{IH}	V_{IL}	Data In(6)

NOTES:
1. V_{PPL} may be ground, a no-connect with a resistor tied to ground, or $\leq$ 6.5V. V_{PPH} is the programming voltage specified for the device. Refer to D.C. Characteristics. When $V_{PP} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes may also be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
3. V_{ID} is the intelligent Identifier high voltage. Refer to DC Characteristics.
4. Read operations with $V_{PP} = V_{PPH}$ may access array data or the intelligent ID.
5. With V_{PP} at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
6. Refer to Table 3 for valid Data-In during a write operation.
7. X can be V_{IL} or V_{IH}.

Write Protection

The command register is only alterable when V_{PP} is at high voltage. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable—available only when memory updates are desired. When high voltage is removed, the contents of the register default to the read command, making the 28F256A a read-only memory. Memory contents cannot be altered.

Or, the system designer may choose to "hardwire" V_{PP}, making the high voltage supply constantly available. In this instance, all operations are performed in conjunction with the command register. The 28F256A is designed to accommodate either design practice, and to encourage optimization of the processor-memory interface.

The two-step Program/Erase write sequence to the command register provides additional software write protection.

BUS OPERATIONS

Read

The 28F256A has two control functions, both of which must be logically active, to obtain data at the outputs. Chip-Enable ($\overline{CE}$) is the power control and should be used for device selection. Output-Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Figure 7 illustrates read timing waveforms.

When V_{PP} is low (V_{PPL}), the read only operation is active. This permits reading the data in the array and outputting the intelligent Identifier codes (See Table 2). When V_{PP} is high (V_{PPH}), the default condition of the device is the read-only mode. This allows reading the data in the array. Further functionality is achieved through the Command Register as shown in Table 3.

Output Disable

With Output-Enable at a logic-high level (V_{IH}), output from the device is disabled. Output pins are placed in a high-impedance state.

Standby

With Chip-Enable at a logic-high level, the standby operation disables most of the 28F256A's circuitry and substantially reduces device power consumption. The outputs are placed in a high-impedance state, independent of the Output-Enable signal. If the 28F256A is deselected during erasure, programming, or program/erase verification, the device draws active current until the operation is terminated.

intelligent Identifier Operation

The intelligent Identifier operation outputs the manufacturer code (89H) and device code (B9H). Programming equipment automatically matches the device with its proper erase and programming algorithms.

With Chip-Enable and Output-Enable at a logic low level, raising A9 to high voltage, V_{ID} (See DC Characteristics), activates the operation. Data read from locations 0000H and 0001H represent the manufacturer's code and the device code, respectively.

The manufacturer- and device-codes can also be read via the command register, for instances where the 28F256A is erased and reprogrammed in the target system. Following a write of 90H to the command register, a read from address location 0000H outputs the manufacturer code (89H). A read from address 0001H outputs the device code (B9H).

Write

Device erasure and programming are accomplished via the command register, when high voltage is applied to the V_{PP} pin. The contents of the register serve as input to the internal state-machine. The state-machine outputs dictate the function of the device.

The command register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command. The command register is written by bringing Write-Enable to a logic-low level (V_{IL}), while Chip-Enable is low. Addresses are latched on the falling edge of Write-Enable, while data is latched on the rising edge of the Write-Enable pulse. Standard microprocessor write timings are used.

Refer to A.C. Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the V_{PP} pin, the contents of the command register default to 00H, enabling read-only operations.

Placing high voltage on the V_{PP} pin enables read/write operations. Device operations are selected by writing specific data patterns into the command register. Table 3 defines these 28F256A register commands.

Table 3. Command Definitions

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	1	Write	X	00H			
Read intelligent ID(4)	1	Write	X	90H	Read	IA	ID
Set-up Erase/Erase(5)	2	Write	X	20H	Write	X	20H
Erase Verify(5)	2	Write	EA	A0H	Read	X	EVD
Set-up Program/Program(6)	2	Write	X	40H	Write	PA	PD
Program Verify(6)	2	Write	X	C0H	Read	X	PVD
Reset(7)	2	Write	X	FFH	Write	X	FFH

NOTES:
1. Bus operations are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   PA = Address of memory location to be programmed.
   Addresses are latched on the falling edge of the Write-Enable pulse.
3. ID = Data read from location IA during device identification (Mfr = 89H, Device = B9H).
   EVD = Data read from location EA during erase verify.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of Write-Enable.
   PVD = Data read from location PA during program verify. PA is latched on the Program command.
4. Following the Read intelligent ID command, two read operations access manufacturer and device codes.
5. Figure 6 illustrates the Quick-Erase Algorithm.
6. Figure 5 illustrates the Quick-Pulse Programming Algorithm.
7. The second bus cycle must be followed by the desired command register write.

Read Command

While V_{PP} is high, for erasure and programming, memory contents can be accessed via the read command. The read operation is initiated by writing 00H into the command register. Microprocessor read cycles retrieve array data. The device remains enabled for reads until the command register contents are altered.

The default contents of the register upon V_{PP} power-up is 00H. This default value ensures that no spurious alteration of memory contents occurs during the V_{PP} power transition. Where the V_{PP} supply is hard-wired to the 28F256A, the device powers-up and remains enabled for reads until the command-register contents are changed. Refer to the A.C. Read Characteristics and Waveforms for specific timing parameters.

Intelligent Identifier Command

Flash-memories are intended for use in applications where the local CPU alters memory contents. As such, manufacturer- and device-codes must be accessible while the device resides in the target system. PROM programmers typically access signature codes by raising A9 to a high voltage. However, multiplexing high voltage onto address lines is not a desired system-design practice.

The 28F256A contains an intelligent Identifier operation to supplement traditional PROM-programming methodology. The operation is initiated by writing 90H into the command register. Following the command write, a read cycle from address 0000H retrieves the manufacturer code of 89H. A read cycle from address 0001H returns the device code of B9H. To terminate the operation, it is necessary to write another valid command into the register.

Set-up Erase/Erase Commands

Set-up Erase is a command-only operation that stages the device for electrical erasure of all bytes in the array. The set-up erase operation is performed by writing 20H to the command register.

To commence chip-erasure, the erase command (20H) must again be written to the register. The erase operation begins with the rising edge of the Write-Enable pulse and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, chip-erasure can only occur when high voltage is applied to the V_{PP} pin. In the absence of this high voltage, memory contents are protected against erasure. Refer to A.C. Erase Characteristics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all bytes of the array in parallel. After each erase operation, all bytes must be verified. The erase verify operation is initiated by writing A0H into the command register. The address for the byte to be verified must be supplied as it is latched on the falling edge of the Write-Enable pulse. The register write terminates the erase operation with the rising edge of its Write-Enable pulse.

The 28F256A applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased.

The erase-verify command must be written to the command register prior to each byte verification to latch its address. The process continues for each byte in the array until a byte does not return FFH data, or the last address is accessed.

In the case where the data read is not FFH, another erase operation is performed. (Refer to Set-up Erase/Erase). Verification then resumes from the address of the last-verified byte. Once all bytes in the array have been verified, the erase step is complete. The device can be programmed. At this point, the verify operation is terminated by writing a valid command (e.g. Program Set-up) to the command register. Figure 6, the Quick-Erase™ Algorithm, illustrates how commands and bus operations are combined to perform electrical erasure of the 28F256A. Refer to A.C. Erase Characteristics and Waveforms for specific timing parameters.

Set-up Program/Program Commands

Set-up program is a command-only operation that stages the device for byte programming. Writing 40H into the command register performs the set-up operation.

Once the program set-up operation is performed, the next Write-Enable pulse causes a transition to an active programming operation. Addresses are internally latched on the falling edge of the Write-Enable pulse. Data is internally latched on the rising edge of the Write-Enable pulse. The rising edge of Write-Enable also begins the programming operation. The programming operation terminates with the next rising edge of Write-Enable, used to write the program-verify command. Refer to A.C. Programming Characteristics and Waveforms for specific timing parameters.

Program-Verify Command

The 28F256A is programmed on a byte-by-byte basis. Byte programming may occur sequentially or at random. Following each programming operation, the byte just programmed must be verified.

The program-verify operation is initiated by writing C0H into the command register. The register write terminates the programming operation with the rising edge of its Write-Enable pulse. The program-verify operation stages the device for verification of the byte last programmed. No new address information is latched.

The 28F256A applies an internally-generated margin voltage to the byte. A microprocessor read cycle outputs the data. A successful comparison between the programmed byte and true data means that the byte is successfully programmed. Programming then proceeds to the next desired byte location. Figure 5, the 28F256A Quick-Pulse Programming Algorithm, illustrates how commands are combined with bus operations to perform byte programming. Refer to A.C. Programming Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or program-command sequences. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

EXTENDED ERASE/PROGRAM CYCLING

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled—an expensive solution.

Intel has designed extended cycling capability into its ETOX-II flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge-carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field is one-tenth that of common EEPROMs, minimizing the probability of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 mV/cm lower than EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure—increasing time to wearout by a factor of 100,000,000.

The device is programmed and erased using Intel's Quick-Pulse Programming and Quick-Erase algorithms. Intel's algorithmic approach uses a series of operations (pulses), along with byte verification, to completely and reliably erase and program the device.

For further reliability information, see Reliability Report RR-60 (ETOX II Reliability Data Summary).

QUICK-PULSE PROGRAMMING ALGORITHM

The Quick-Pulse Programming algorithm uses programming operations of 10 μs duration. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The algorithm allows for up to 25 programming operations per byte, although most bytes verify on the first or second operation. The entire sequence of programming and byte verification is performed with V_{PP} at high voltage. Figure 5 illustrates the Quick-Pulse Programming algorithm.

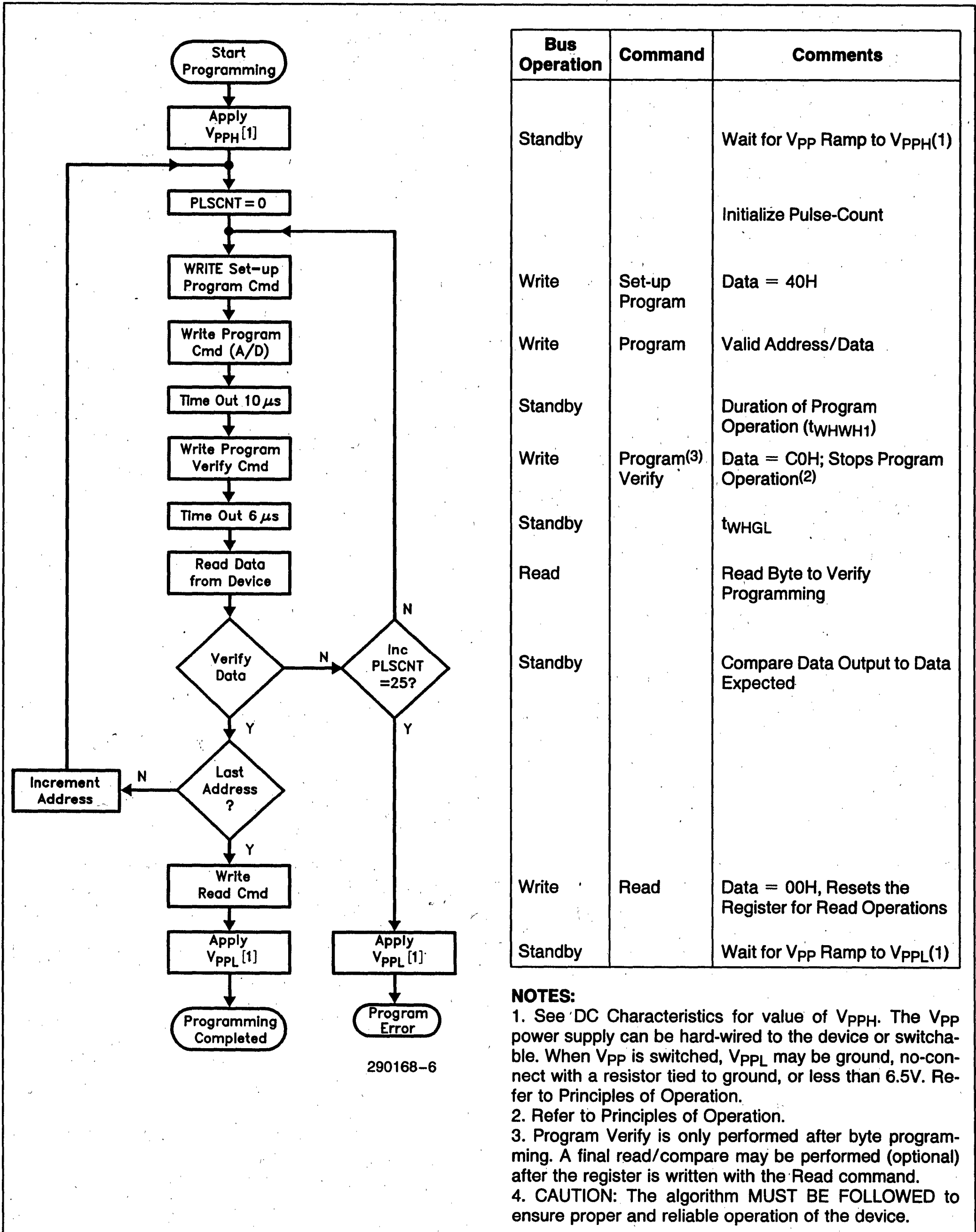
QUICK-ERASE ALGORITHM

Intel's Quick-Erase algorithm yields fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the Quick-Pulse Programming™ algorithm, to simultaneously remove charge from all bits in the array.

Erasure begins with a read of memory contents. The 28F256A is erased when shipped from the factory. Reading FFH data from the device would immediately be followed by device programming.

Uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (Data = 00H). This is accomplished, using the Quick-Pulse Programming algorithm, in approximately, one-half second.

Erase execution then continues with an initial erase operation. Erase verification (data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. With each erase operation, an increasing number of bytes verify to the erased state. Erase efficiency may be improved by storing the address of the last byte verified in a register. Following the next erase operation, verification starts at that stored address location. Erasure typically occurs in one second. Figure 6 illustrates the Quick-Erase Algorithm.

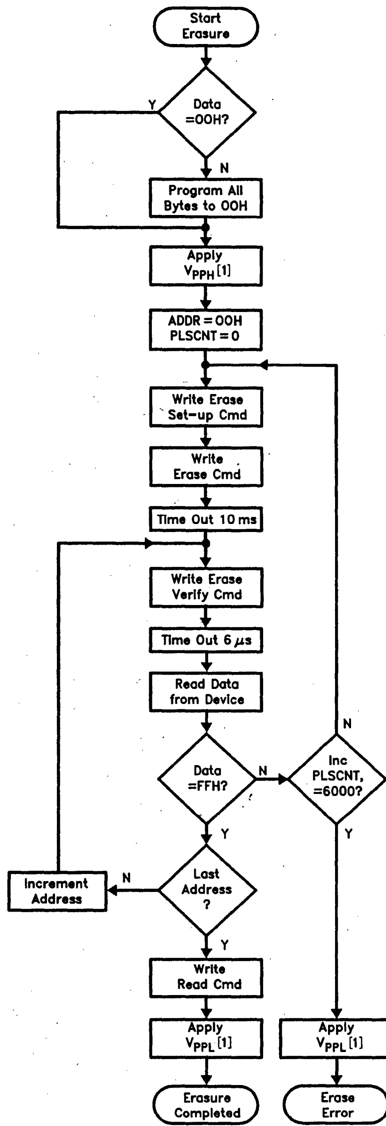


Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Pulse-Count
Write	Set-up Program	Data = 40H
Write	Program	Valid Address/Data
Standby		Duration of Program Operation (t_{WHWH1})
Write	Program(3) Verify	Data = C0H; Stops Program Operation(2)
Standby		t_{WHGL}
Read		Read Byte to Verify Programming
Standby		Compare Data Output to Data Expected
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

NOTES:

1. See DC Characteristics for value of V_{PPH}. The V_{PP} power supply can be hard-wired to the device or switchable. When V_{PP} is switched, V_{PPL} may be ground, no-connect with a resistor tied to ground, or less than 6.5V. Refer to Principles of Operation.
2. Refer to Principles of Operation.
3. Program Verify is only performed after byte programming. A final read/compare may be performed (optional) after the register is written with the Read command.
4. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.

Figure 5. 28F256A Quick-Pulse Programming Algorithm



Bus Operation	Command	Comments
		Entire memory must = 00H before erasure
		Use Quick-Pulse Programming Algorithm (Figure 5)
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Addresses and Pulse-Count
Write	Set-up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase(3) Verify	Addr = Byte to Verify; Data = A0H; Stops Erase Operation(2)
Standby		t_{WHGL}
Read		Read Byte to Verify Erasure
Standby		Compare Output to FFH Increment Pulse-Count
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

NOTES:

1. See DC Characteristics for value of V_{PPH}. The V_{PP} power supply can be hard-wired to the device or switchable. When V_{PP} is switched, V_{PPL} may be ground, no-connect with a resistor tied to ground, or less than 6.5V. Refer to Principles of Operation.
2. Refer to Principles of Operation.
3. Erase Verify is performed only after chip-erasure. A final read/compare may be performed (optional) after the register is written with the read command.
4. CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.

Figure 6. 28F256A Quick-Erase Algorithm

DESIGN CONSIDERATIONS

Two-Line Output Control

Flash-memories are often used in larger memory arrays. Intel provides two read-control inputs to accommodate multiple memory connections. Two-line control provides for:

a. the lowest possible memory power dissipation and,
b. complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address-decoder output should drive chip-enable, while the system's read signal controls all flash-memories and other parallel memories. This assures that only enabled memory devices have active outputs, while deselected devices maintain the low power standby condition.

Power Supply Decoupling

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current (I_{CC}) issues—standby, active, and transient current peaks produced by falling and rising edges of chip-enable. The capacitive and inductive loads on the device outputs determine the magnitudes of these peaks.

Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 μF ceramic capacitor connected between V_{CC} and V_{SS}, and between V_{PP} and V_{SS}.

Place the high-frequency, low-inherent-inductance capacitors as close as possible to the devices. Also, for every eight devices, a 4.7 μF electrolytic capacitor should be placed at the array's power supply connection, between V_{CC} and V_{SS}. The bulk capacitor will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

V_{PP} Trace on Printed Circuit Boards

Programming flash-memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the V_{PP} power supply trace. The V_{PP} pin supplies the memory cell current for programming. Use similar trace widths and layout considerations given the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots.

Power Up/Down Protection

The 28F256A is designed to offer protection against accidental erasure or programming, caused by spurious system-level signals that may exist during power transitions. Also, with its control register architecture, alteration of memory contents only occurs after successful completion of the two-step command sequences. Power supply sequencing is not required. Internal circuitry of the 28F256A ensures that the command register architecture is reset to the read mode on power up.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will inhibit writes. The control register architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

28F256A Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases the usable battery life of your system because the 28F256A does not consume any power to retain code or data when the system is off. Table 4 illustrates the power dissipated when updating the 28F256A.

Table 4. 28F256A Typical Update Power Dissipation(4)

Operation	Power Dissipation (Watt-Seconds)	Notes
Array Program/Program Verify	0.043	1
Array Erase/Erase Verify	0.083	2
One Complete Cycle	0.169	3

NOTES:

1. Formula to calculate typical Program/Program Verify Power = [V_{PP} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{PP2} typical + t_{WHGL} × I_{PP4} typical)] + [V_{CC} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{CC2} typical + t_{WHGL} × I_{CC2} typical)].
2. Formula to calculate typical Erase/Erase Verify Power = [V_{PP} (I_{PP3} typical × t_{ERASE} typical + I_{PP5} typical × t_{WHGL} × # Bytes)] + [V_{CC}(I_{CC3} typical × t_{ERASE} typical + I_{CC5} typical × t_{WHGL} × # Bytes)].
3. One Complete Cycle = Array Preprogram + Array Erase + Program.
4. "Typicals" are not guaranteed, but based on a limited number of samples from production lots.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
- During Read .... −40°C to +125°C(1)
- During Erase/Program .... −40°C to +125°C

Temperature Under Bias .... −40°C to +125°C

Storage Temperature .... −65°C to +150°C

Voltage on Any Pin with Respect to Ground .... −2.0V to +7.0V(2)

Voltage on Pin A_9 with Respect to Ground .... −2.0V to +13.5V(2, 3)

V_{PP} Supply Voltage with Respect to Ground During Erase/Program .... −2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage with Respect to Ground .... −2.0V to +7.0V(2)

Output Short Circuit Current............. 100 mA(4)

Maximum Junction Temperature (T_J) .... 140°C

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for automotive product defined by this specification.
2. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
3. Maximum DC voltage on A_9 or V_{PP} may overshoot to +14.0V for periods less than 20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature	−40	+125	°C	For Read-Only and Read/Write Operations
V_{CC}	V_{CC} Supply Voltage	4.50	5.50	V	

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1			1.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	30	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE (Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{PP1}	V_{PP} Read Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10	μA	$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	μA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 2.1 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	I_{OH} = −2.5 mA $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	$A_9 = V_{ID}$
I_{ID}	V_{CC} ID Current	1		10	30	mA	$A_9 = V_{ID}$
	V_{PP} ID Current			90	500	μA	
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—CMOS COMPATIBLE

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		50	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC}$ ±0.2V
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	30	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress

DC CHARACTERISTICS—CMOS COMPATIBLE (Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 2.1 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		0.85 V_{CC}			V	I_{OH} = −2.5 mA, $V_{CC} = V_{CC}$ Min
V_{OH2}			V_{CC} − 0.4				I_{OH} = −100 μA, $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	
I_{ID}	V_{CC} ID Current	1		10	30	mA	A_9 = ID
I_{ID}	V_{PP} ID Current	1		90	500	μA	A_9 = ID
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Programs are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/ Write Operations		11.40		12.60	V	V_{PP} = 12.0V
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

CAPACITANCE(3) T_A = 25°C, f = 1.0 MHz

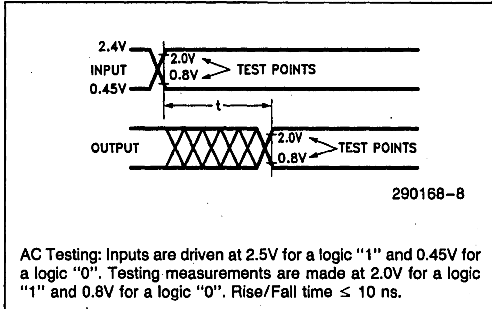
Symbol	Parameter	Limits		Unit	Conditions
		Min	Max		
C_{IN}	Address/Control Capacitance		8	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance		12	pF	V_{OUT} = 0V

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C.
2. Not 100% tested: characterization data available.
3. Sampled, not 100% tested.
4. "Typicals" are not guaranteed, but are based on a limited number of samples from production lots.

AC TEST CONDITIONS

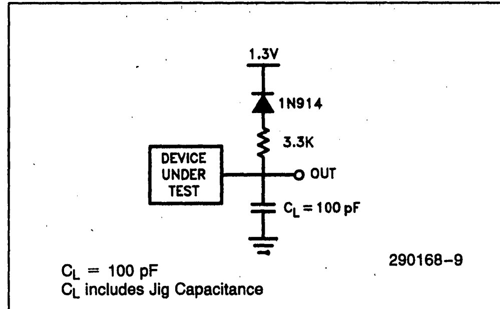
Input Rise and Fall Times (10% to 90%) . . . . . . 10 ns
Input Pulse Levels . . . . 0.45V and 2.4V
Input Timing Reference Level . . . . 0.8V and 2.0V
Output Timing Reference Level . . . . . . 0.8V and 2.0V

AC TESTING INPUT/OUTPUT WAVEFORM



AC Testing: Inputs are driven at 2.5V for a logic "1" and 0.45V for a logic "0". Testing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0". Rise/Fall time ≤ 10 ns.

AC TESTING LOAD CIRCUIT



C_L = 100 pF
C_L includes Jig Capacitance

AC CHARACTERISTICS—Read-Only Operations(2)

Versions			28F256A-120		28F256A-150		Unit
Symbol	Characteristic	Notes	Min	Max	Min	Max	
t_{AVAV}/t_{RC}	Read Cycle Time	3	120		150		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time			120		150	ns
t_{AVQV}/t_{ACC}	Address Access Time			120		150	ns
t_{GLQV}/t_{OE}	Output Enable Access Time			50		55	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	3	0		0		ns
t_{EHQZ}	Chip Disable to Output in High Z	3		50		55	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	3	0		0		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	4		30		35	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	1, 3	0		0		ns
t_{WHGL}	Write Recovery Time before Read		6		6		µs

NOTES:
1. Whichever occurs first.
2. Rise/Fall Time ≤ 10 ns.
3. Not 100% tested: characterization data available.
4. Guaranteed by design.

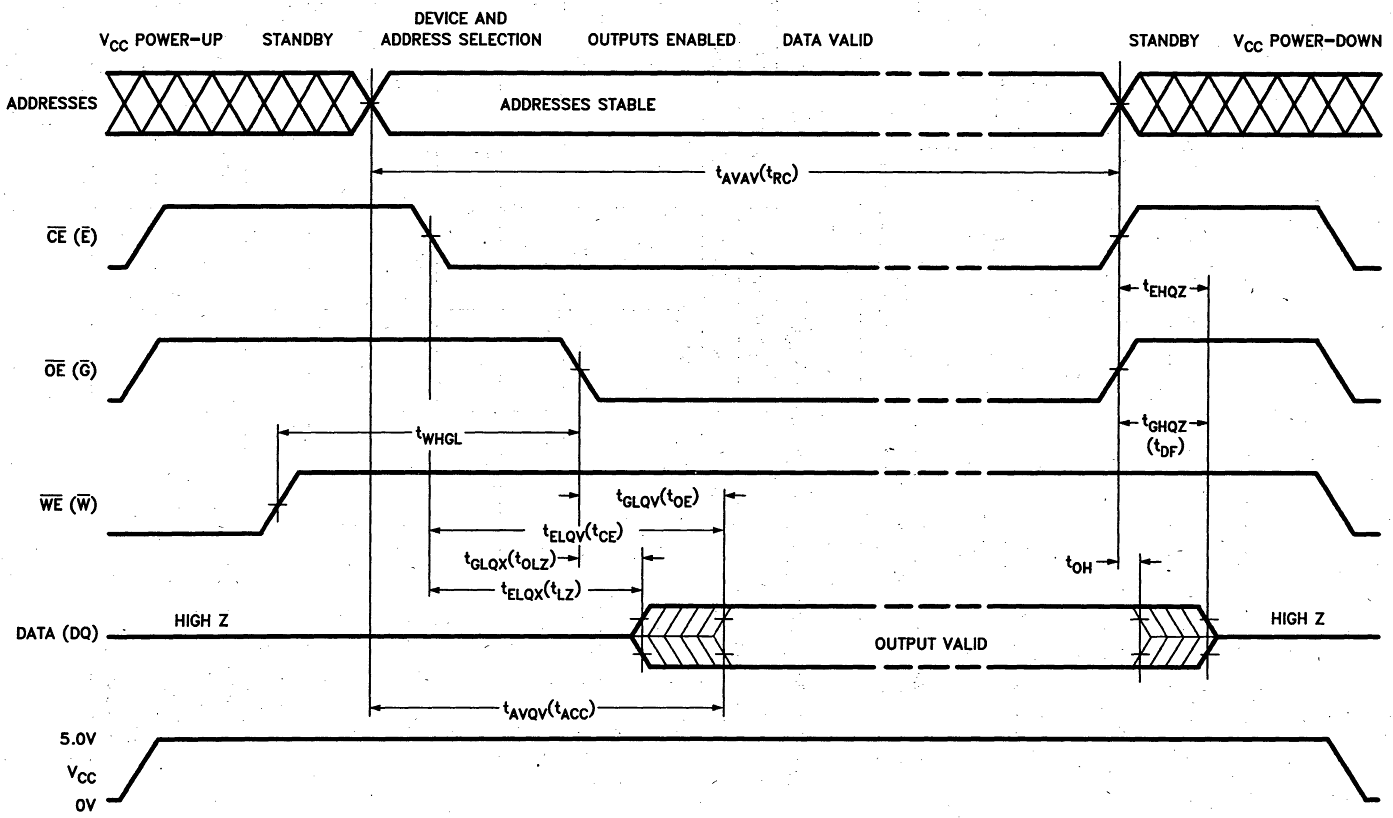


Figure 7. AC Waveforms for Read Operations

AC CHARACTERISTICS—Write/Erase/Program Operations(1, 3)

Versions			28F256A-120		28F256A-150		Unit
Symbol	Characteristic	Notes	Min	Max	Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		120		150		ns
t_{AVWL}/t_{AS}	Address Set-Up Time		0		0		ns
t_{WLAX}/t_{AH}	Address Hold Time	2	60		60		ns
t_{DVWH}/t_{DS}	Data Set-up Time		50		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		10		ns
t_{WHGL}	Write Recovery Time before Read		6		6		μs
t_{GHWL}	Read Recovery Time before Write		0		0		μs
t_{ELWL}/t_{CS}	Chip Enable Set-Up Time before Write	2	20		20		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time		0		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width	2	80		80		ns
t_{ELEH}	Alternative Write Pulse Width	2	80		80		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		20		ns
t_{WHWH1}	Duration of Programming Operation	4	10		10		μs
t_{WHWH2}	Duration of Erase Operation	4	9.5		9.5		ms
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low		1.0		1.0		ms

NOTES:
1. Read timing characteristics during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of Chip-Enable and Write-Enable. In systems where Chip-Enable defines the write pulse width (within a longer Write-Enable timing waveform) all set-up, hold, and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.
3. Rise/Fall time ≤ 10 ns.
4. The integrated stop timer terminates the programming/erase operations, thereby eliminating the need for a maximum specification.

5

ERASE AND PROGRAMMING PERFORMANCE

Parameter	Notes	Limits			Unit	Comments
		Min	Typ	Max		
Chip Erase Time	1, 3, 4, 6		1	60	Sec	Excludes 00H Programming Prior to Erasure
Chip Program Time	1, 2, 4		0.5	3.1	Sec	Excludes System-Level Overhead
Erase/Program Cycles	1, 3, 5	1,000	100,000		Cycles	

NOTES:
1. "Typicals" are not guaranteed, but based on a limited number of samples taken from production lots. Data taken at T = 25°C, V_{PP} = 12.0V, V_{CC} = 5.0V.
2. Minimum byte programming time excluding system overhead is 16 μs (10 μs program + 6 μs write recovery), while maximum is 400 μs/byte (16 μs × 25 loops allowed by algorithm). Max chip programming time is specified lower than the worst case allowed by the programming algorithm since most bytes program significantly faster than the worst case byte.
3. Excludes 00H Programming prior to erasure.
4. Excludes system-level overhead.
5. Refer to RR-60 "ETOX Flash Memory Reliability Data Summary" for typical cycling data and failure rate calculations.
6. Maximum erase specification is determined by algorithmic limit and accounts for cumulative effect of erasure at T = −40°C, 1,000 cycles, V_{PP} = 11.4V, V_{CC} = 4.5V.

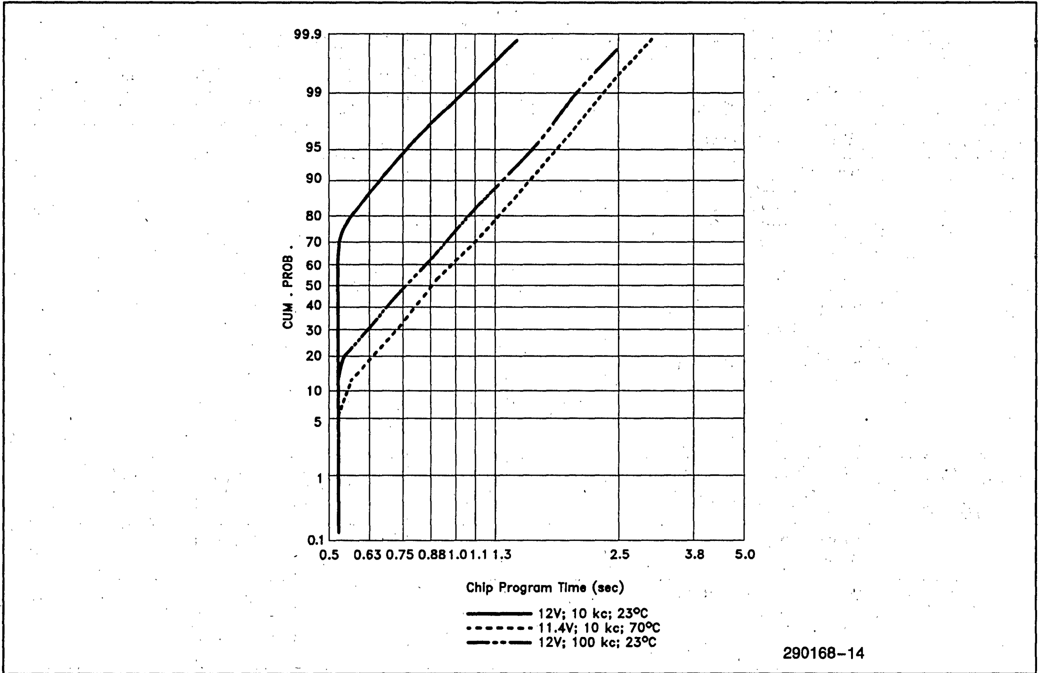


Figure 8. 28F256A Typical Programming Capability
See Note 1, Page 5-19

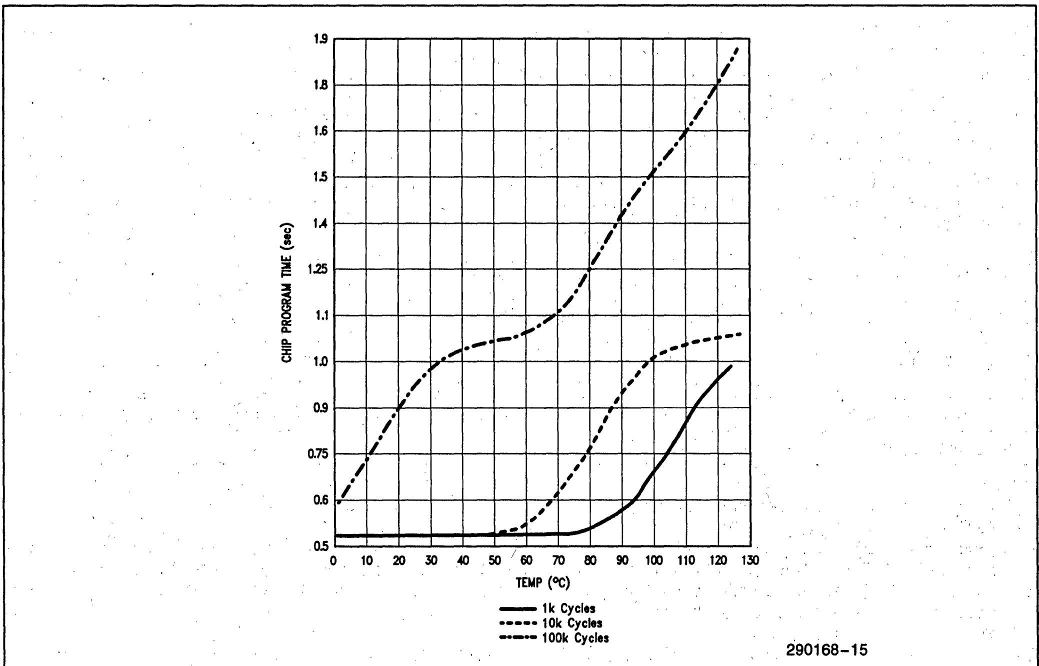


Figure 9. 28F256A Typical Program Time at V_{PP} = 12.0V

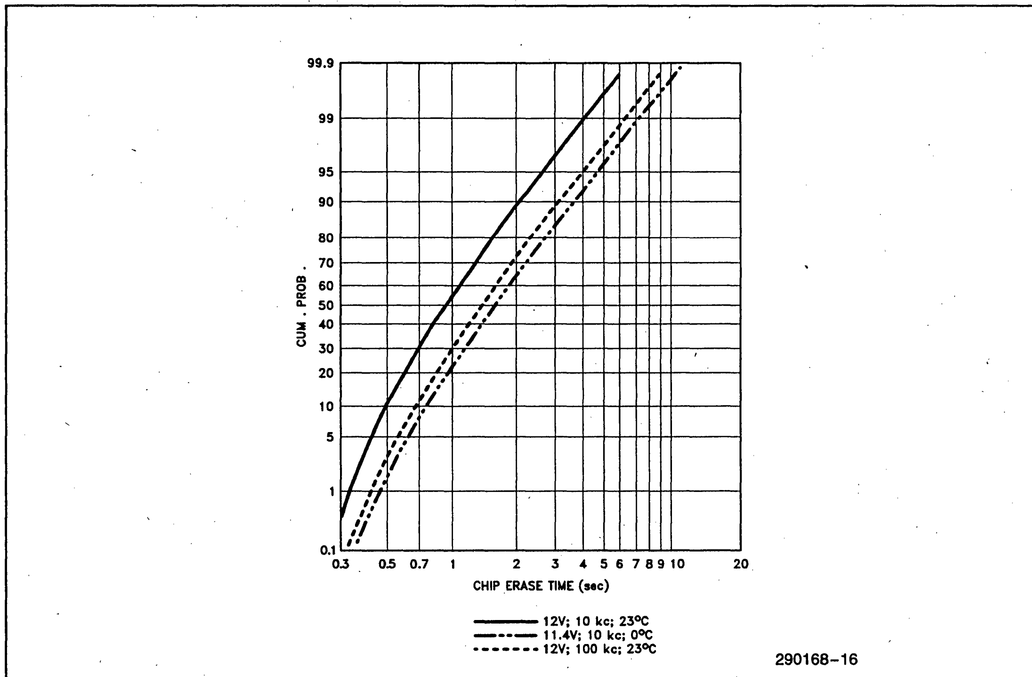


Figure 10. 28F256A Typical Erase Capability
See Note 1, Page 5-19

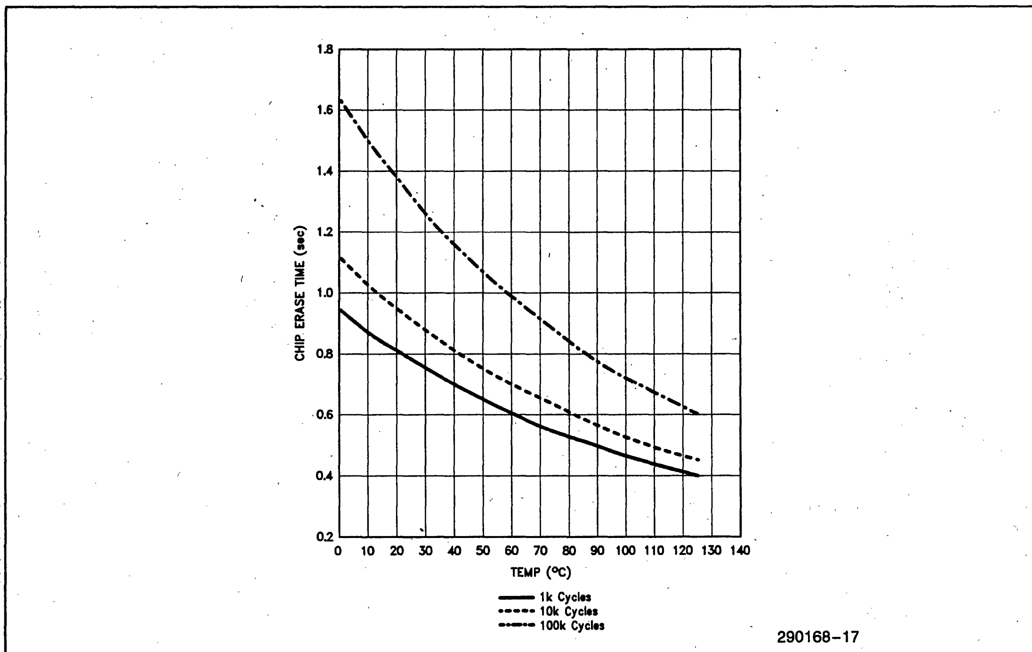


Figure 11. 28F256A Typical Erase Time at V_{PP} = 12.0V

5

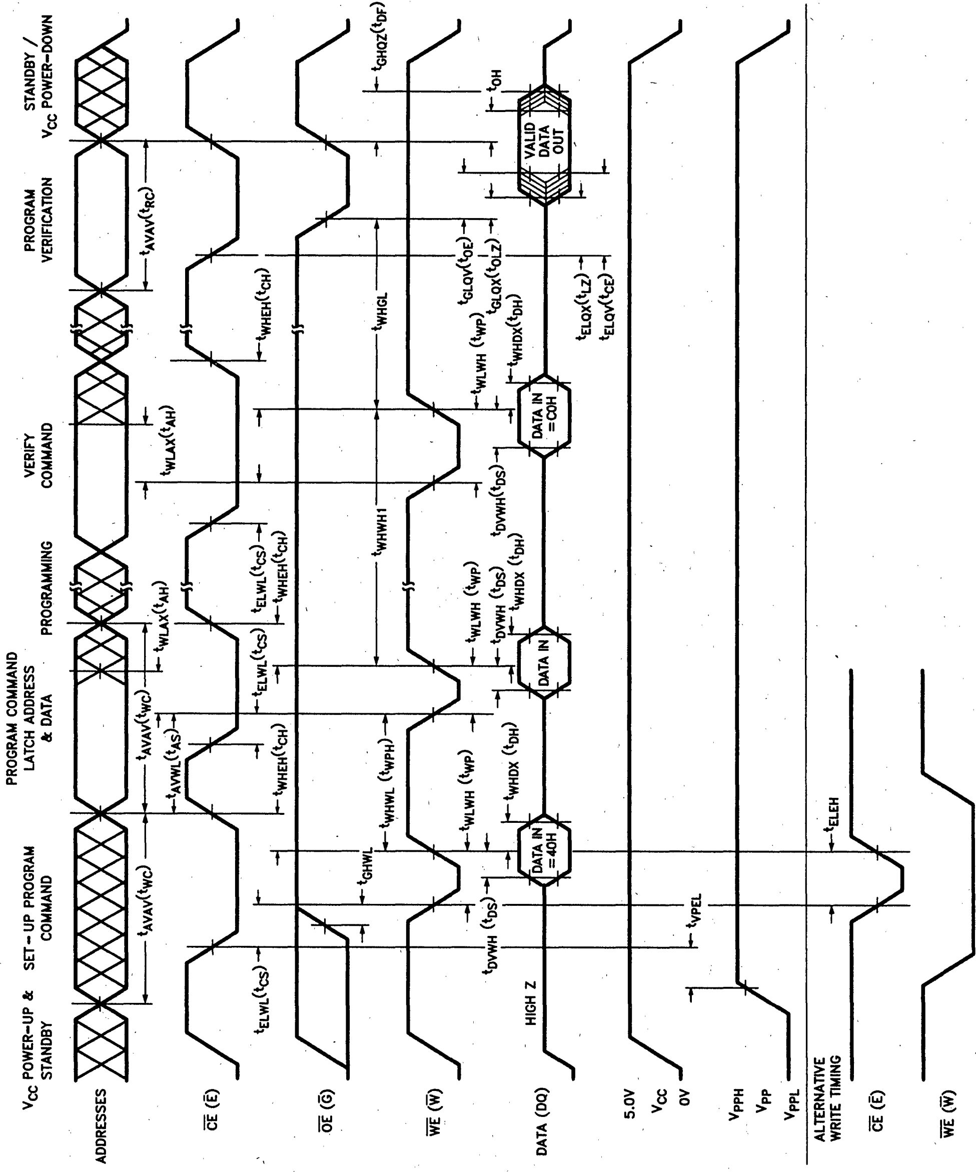


Figure 12. AC Waveforms for Programming Operations

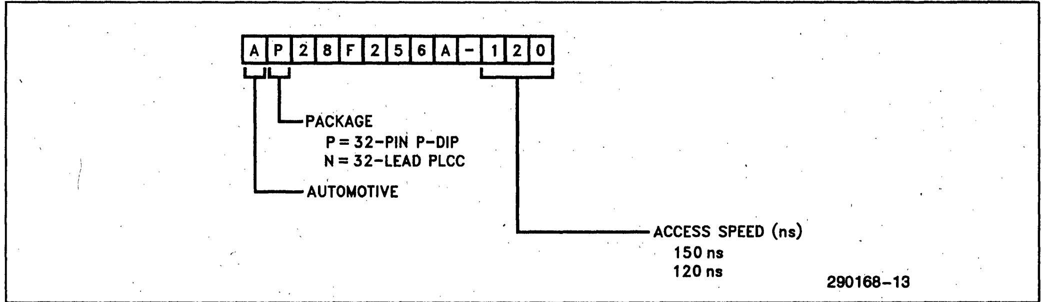
V_{CC} POWER-UP & STANDBY | SET-UP ERASE COMMAND | ERASE COMMAND | ERASURE | ERASE VERIFY COMMAND | ERASE VERIFICATION | STANDBY / V_{CC} POWER-DOWN

ADDRESSES
$t_{AVAV}(t_{WC})$
$t_{AVAV}(t_{WC})$
$t_{AVWL}(t_{AS})$
$t_{AVAV}(t_{WC})$
$t_{AVAV}(t_{RC})$
$t_{WLAX}(t_{AH})$
$\overline{CE}$ ($\overline{E}$)
$t_{ELWL}(t_{CS})$
$t_{WHEH}(t_{CH})$
$t_{ELWL}(t_{CS})$
$t_{ELWL}(t_{CS})$
$t_{WHEH}(t_{CH})$
$t_{WHEH}(t_{CH})$
$\overline{OE}$ ($\overline{G}$)
t_{GHWL}
t_{WHWL} (t_{WPH})
t_{WHWH2}
t_{WHGL}
$t_{GHQZ}(t_{DF})$
$\overline{WE}$ ($\overline{W}$)
t_{WLWH} (t_{WP})
t_{WLWH} (t_{WP})
t_{WLWH} (t_{WP})
$t_{GLQV}(t_{OE})$
t_{DVWH} (t_{DS})
t_{DVWH} (t_{DS})
$t_{DVWH}(t_{DS})$
$t_{GLQX}(t_{OLZ})$
t_{OH}
t_{WHDX} (t_{DH})
t_{WHDX} (t_{DH})
$t_{WHDX}(t_{DH})$
DATA (DQ)
HIGH Z
DATA IN =20H
DATA IN =20H
DATA IN =A0H
VALID DATA OUT
$t_{ELQX}(t_{LZ})$
$t_{ELQV}(t_{CE})$
5.0V
V_{CC}
0V
t_{VPEL}
V_{PPH}
V_{PP}
V_{PPL}

290168-12

Figure 13. AC Waveforms for Erase Operations

Ordering Information



VALID COMBINATIONS:

AP28F256A-120	AP28F256A-120
AP28F256A-150	AN28F256A-150

ADDITIONAL INFORMATION

	Order Number
AP-316, "Using the 28F256A Flash Memory for In-System Reprogrammable Non-Volatile Storage"	292046
ER-21, "The Intel 28F256 Flash Memory"	294004
RR-60, "ETOX™ II Flash Memory Reliability Data Summary"	293002
AP-325, "Guide to Flash Memory Reprogramming"	292059

REVISION HISTORY

Number	Description
005	Changed Erase/Program Cycles to 1,000 minimum.
006	Added 120 ns Speed

A28F512
512K (64K x 8) CMOS FLASH MEMORY
(Automotive)

- **Extended Automotive Temperature Range: −40°C to +125°C**
- **Flash Electrical Chip-Erase**
  - **1 Second Typical Chip-Erase**
- **Quick-Pulse Programming Algorithm**
  - **10 μs Typical Byte-Program**
  - **1 Second Chip-Program**
- **1,000 Erase/Program Cycle Minimum Over Automotive Temperature Range**
- **12.0V ±5% V_{PP}**
- **High-Performance Read**
  - **120/150 ns Maximum Access Time**
- **CMOS Low Power Consumption**
  - **30 mA Maximum Active Current**
  - **100 μA Maximum Standby Current**
- **Integrated Program/Erase Stop Timer**
- **Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface**
- **Noise Immunity Features**
  - **±10% V_{CC} Tolerance**
  - **Maximum Latch-Up Immunity through EPI Processing**
- **ETOX™-II Flash Nonvolatile Memory Technology**
  - **EPROM-Compatible Process Base**
  - **High-Volume Manufacturing Experience**
- **JEDEC-Standard Pinouts**
  - **32-Pin Plastic DIP**
  - **32-Lead PLCC**

  (See Packaging Spec., Order #231369)

Intel's 28F512 CMOS flash memory offers the most cost-effective and reliable alternative for read/write random access nonvolatile memory. The 28F512 adds electrical chip-erasure and reprogramming to familiar EPROM technology. Memory contents can be rewritten: in a test socket; in a PROM-programmer socket; on-board during subassembly test; in-system during final test; and in-system after-sale. The 28F512 increases memory flexibility, while contributing to time- and cost-savings.

The 28F512 is a 512-kilobit nonvolatile memory organized as 65,536 bytes of 8 bits. Intel's 28F512 is offered in 32-pin cerdip or 32-lead PLCC packages. Pin assignments conform to JEDEC standards.

Extended erase and program cycling capability is designed into Intel's ETOX™ II (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional EEPROMs. With the 12.0V V_{PP} supply, the 28F512 performs a minimum of 1,000 erase and program cycles well within the time limits of the Quick-Pulse Programming™ and Quick-Erase™ algorithms.

Intel's 28F512 employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 150 nanosecond access time provides no-WAIT-state performance for a wide range of microprocessors and microcontrollers. Maximum standby current of 100 μA translates into power savings when the device is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from −1V to V_{CC} + 1V.

With Intel's ETOX II process base, the 28F512 levers years of EPROM experience to yield the highest levels of quality, reliability, and cost-effectiveness.

In order to meet the rigorous environmental requirements of Automotive applications, Intel offers the 28F512 in extended Automotive temperature range. Read and Write Characteristics are guaranteed over the range of −40°C to +125°C ambient.

September 1992
Order Number: 290265-004

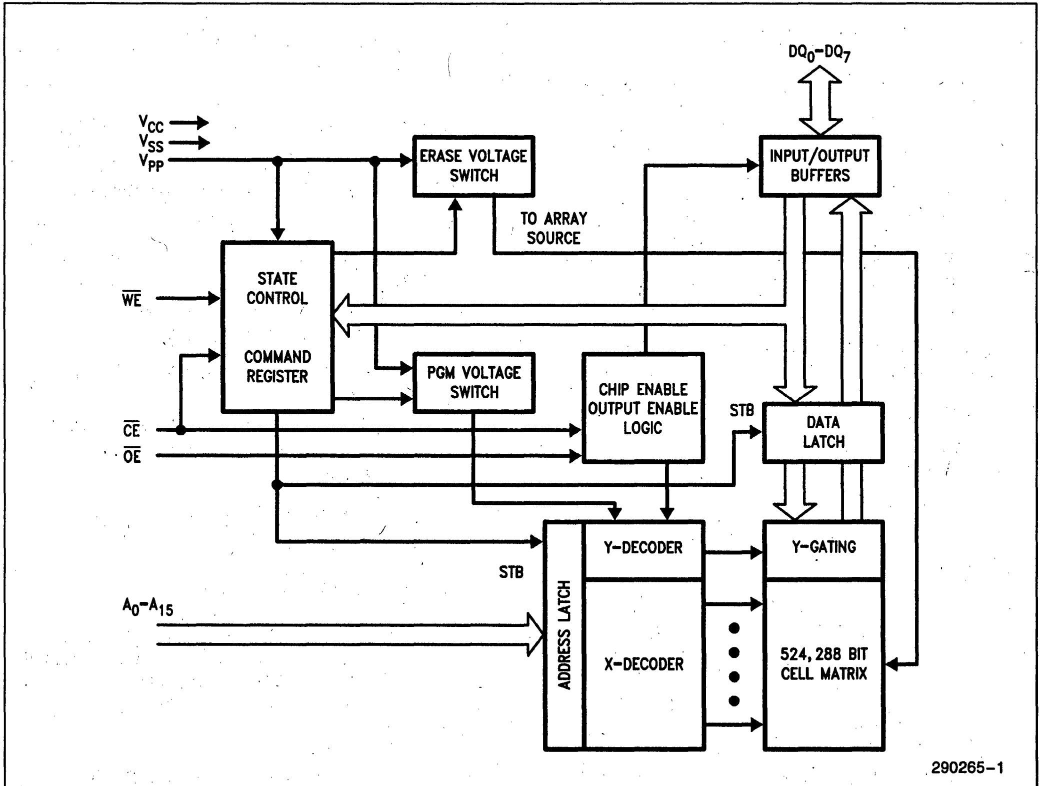


Figure 1. 28F512 Block Diagram

AUTOMOTIVE TEMPERATURE FLASH MEMORIES

The Intel Automotive FLASH Memories have received additional processing to enhance product characteristics. The automotive temperature range is −40°C to +125°C during the read/write/erase/program operations.

Speed Versions	Packaging Options	
	Plastic DIP	PLCC
-120	AP	AN
-150	AP	AN

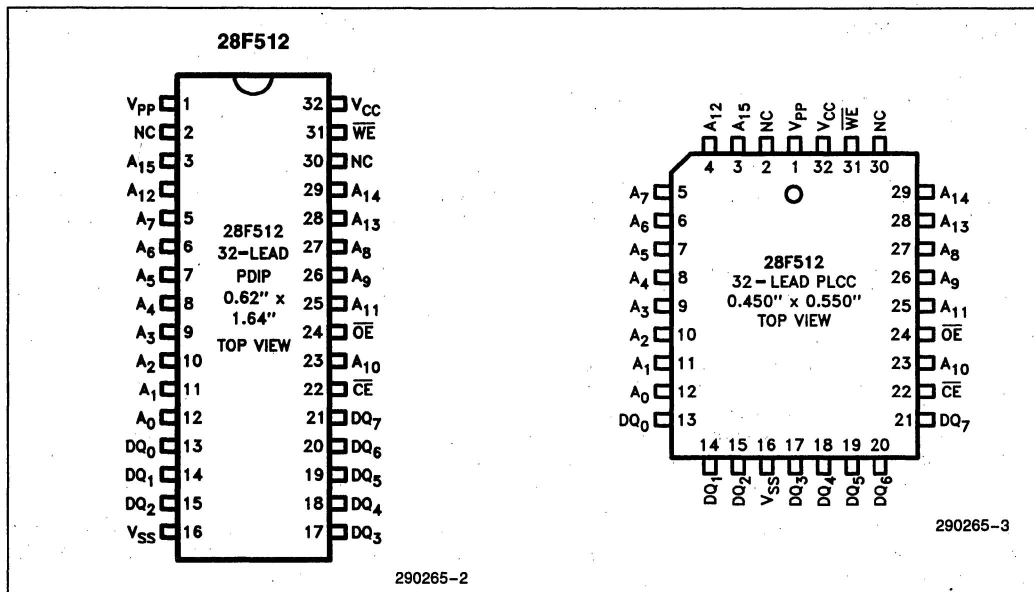


Figure 2. 28F512 Pin Configurations

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{15}	INPUT	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
DQ_0–DQ_7	INPUT/OUTPUT	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}$	INPUT	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels.
$\overline{OE}$	INPUT	**OUTPUT ENABLE:** Gates the devices output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	INPUT	**WRITE ENABLE:** Controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **Note:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP}		**ERASE/PROGRAM POWER SUPPLY** for writing the command register, erasing the entire array, or programming bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY** (5V ±10%)
V_{SS}		**GROUND**
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating.

5

APPLICATIONS

The 28F512 flash-memory adds electrical chip-erasure and reprogrammability to EPROM non-volatility and ease of use. The 28F512 is ideal for storing code or data-tables in applications where periodic updates are required. The 28F512 also serves as a dense, nonvolatile data acquisition and storage medium.

The need for code updates pervades all phases of a system's life—from prototyping to system manufacture to after-sale service. In the factory, during prototyping, revisions to control code necessitate ultraviolet erasure and reprogramming of EPROM-based prototype codes. The 28F512 replaces the 15- to 20-minute ultraviolet erasure with one-second electrical erasure. Electrical chip-erasure and reprogramming occur in the same workstation or PROM-programmer socket.

Diagnostics, performed at subassembly or final assembly stages, often require the socketing of EPROMs. Socketed test codes are ultimately replaced with EPROMs containing the final program. With electrical chip-erasure and reprogramming, the 28F512 is soldered to the circuit board. Test codes are programmed into the 28F512 as it resides on the circuit board. Ultimately, the final code can be downloaded to the device. The 28F512's in-circuit alterability eliminates unnecessary handling and less-reliable socketed connections, while adding greater test flexibility.

Material and labor costs associated with code changes increase at higher levels of system integration—the most costly being code updates after sale. Code "bugs", or the desire to augment system functionality, prompt after-sale code updates. Field revisions to EPROM-based code require the removal of EPROM components or entire boards.

Designing with the in-circuit alterable 28F512 eliminates socketed memories, reduces overall material costs, and drastically cuts the labor costs associated with code updates. With the 28F512, code updates are implemented locally via an edge-connector, or remotely over a serial communication link.

The 28F512's electrical chip-erasure, byte reprogrammability, and complete nonvolatility fit well with data accumulation needs. Electrical chip-erasure gives the designer a "blank-slate" in which to log data. Data can be periodically off-loaded for analysis—erasing the slate and repeating the cycle. Or, multiple devices can maintain a "rolling window" of accumulated data.

With high density, nonvolatility, and extended cycling capability, the 28F512 offers an innovative alternative for mass storage.

Integrating main memory and backup storage functions into directly executable flash memory boosts system performance, shrinks system size, and cuts power consumption. Reliability exceeds that of electromechanical media, with greater durability in extreme environmental conditions.

A high degree of on-chip feature integration simplifies memory-to-processor interfacing. Figure 3 depicts two 28F512s tied to the 80C186 system bus. The 28F512's architecture minimizes interface circuitry needed for complete in-circuit updates of memory contents.

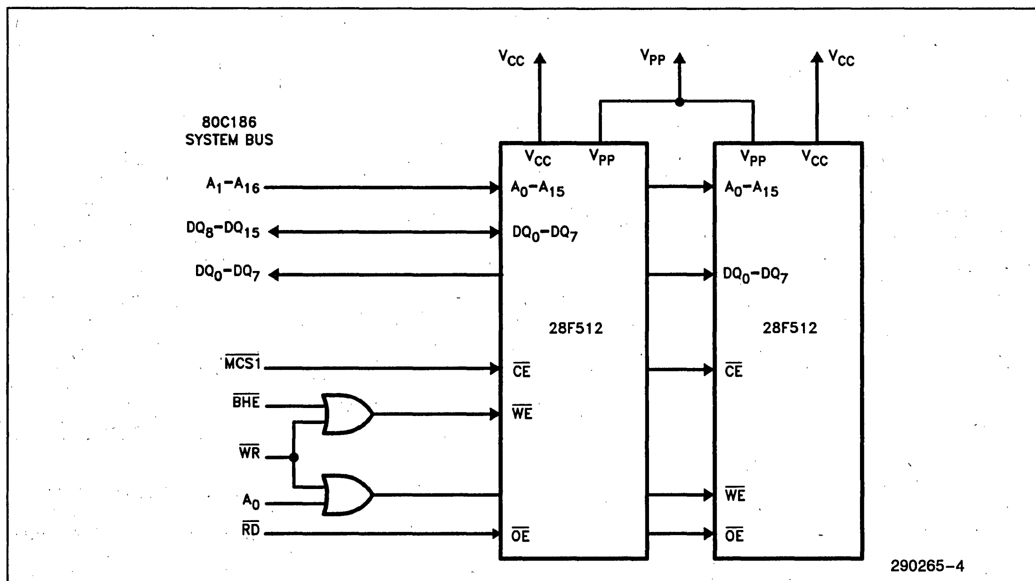


Figure 3. 28F512 in an 80C186 System

With cost-effective in-system reprogramming and extended cycling capability, the 28F512 fills the functionality gap between traditional EPROMs and EEPROMs. EPROM-compatible specifications, straightforward interfacing, and in-circuit alterability allows designers to easily augment memory flexibility and satisfy the need for updatable nonvolatile storage in today's designs.

PRINCIPLES OF OPERATION

Flash-memory augments EPROM functionality with in-circuit electrical erasure and reprogramming. The 28F512 introduces a command register to manage this new functionality. The command register allows for: 100% TTL-level control inputs; fixed power supplies during erasure and programming; and maximum EPROM compatibility.

In the absence of high voltage on the V_{PP} pin, the 28F512 is a read-only memory. Manipulation of the external memory-control pins yields the standard EPROM read, standby, output disable, and Intelligent Identifier operations.

The same EPROM read, standby, and output disable operations are available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} enables erasure and programming of the device. All functions associated with altering memory contents—Intelligent Identifier, erase, erase verify, program, and program verify—are accessed via the command register.

Commands are written to the register using standard microprocessor write timings. Register contents serve as input to an internal state-machine which controls the erase and programming circuitry. Write cycles also internally latch addresses and data needed for programming or erase operations. With the appropriate command written to the register, standard microprocessor read timings output array data, access the Intelligent Identifier codes, or output data for erase and program verification.

Integrated Program/Erase Stop Timer

Successive command write cycles define the durations of program and erase operations; specifically, the program or erase time durations are normally terminated by associated program or erase verify commands. An integrated stop timer provides simplified timing control over these operations; thus eliminating the need for maximum program/erase timing specifications. Program and erase pulse durations are minimums only. When the stop timer terminates a program or erase operation, the device enters an inactive state and remains inactive until receiving the appropriate verify or reset command.

Write Protection

The command register is only alterable when V_{PP} is at high voltage. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable—available only when memory updates are desired. When high voltage is removed,

Table 2. 28F512 Bus Operations

Operation	Pins	V_{PP}(1)	A_0	A_9	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_0–DQ_7
READ-ONLY	Read	V_{PPL}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out
	Output Disable	V_{PPL}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby	V_{PPL}	X	X	V_{IH}	X	X	Tri-State
	Intelligent Identifier (Mfr)(2)	V_{PPL}	V_{IL}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = 89H
	Intelligent Identifier (Device)(2)	V_{PPL}	V_{IH}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = B8H
READ/WRITE	Read	V_{PPH}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out(4)
	Output Disable	V_{PPH}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby(5)	V_{PPH}	X	X	V_{IH}	X	X	Tri-State
	Write	V_{PPH}	A_0	A_9	V_{IL}	V_{IH}	V_{IL}	Data In(6)

NOTES:
1. V_{PPL} may be ground, a no-connect with a resistor tied to ground, or $\leq$ 6.5V. V_{PPH} is the programming voltage specified for the device. Refer to D.C. Characteristics. When $V_{PP} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes may also be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
3. V_{ID} is the Intelligent Identifier high voltage. Refer to DC Characteristics.
4. Read operations with $V_{PP} = V_{PPH}$ may access array data or the Intelligent Identifier codes.
5. With V_{PP} at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
6. Refer to Table 3 for valid Data-In during a write operation.
7. X can be V_{IL} or V_{IH}.

the contents of the register default to the read command, making the 28F512 a read-only memory. Memory contents cannot be altered.

Or, the system designer may choose to "hardwire" V_{PP}, making the high voltage supply constantly available. In this instance, all operations are performed in conjunction with the command register. The 28F512 is designed to accommodate either design practice, and to encourage optimization of the processor-memory interface.

The two-step program/erase write sequence to the Command Register provides additional software write protection.

BUS OPERATIONS

Read

The 28F512 has two control functions, both of which must be logically active, to obtain data at the outputs. Chip-Enable ($\overline{CE}$) is the power control and should be used for device selection. Output-Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Figure 6 illustrates read timing waveforms.

When V_{PP} is low (V_{PPL}), the read only operation is active. This permits reading the data in the array and outputting the Intelligent Identifier codes (see Table 2). When V_{PP} is high (V_{PPH}), the default condition of the device is the read only mode. This allows reading the data in the array. Further functionality is achieved though the Command Register as shown in Table 3.

Output Disable

With Output-Enable at a logic-high level (V_{IH}), output from the device is disabled. Output pins are placed in a high-impedance state.

Standby

With Chip-Enable at a logic-high level, the standby operation disables most of the 28F512's circuitry and substantially reduces device power consumption. The outputs are placed in a high-impedance state, independent of the Output-Enable signal. If the 28F512 is deselected during erasure, programming, or program/erase verification, the device draws active current until the operation is terminated.

Intelligent Identifier Operation

The Intelligent Identifier operation outputs the manufacturer code (89H) and device code (B8H). Pro-

gramming equipment automatically matches the device with its proper erase and programming algorithms.

With Chip-Enable and Output-Enable at a logic low level, raising A9 to high voltage V_{ID} (see DC Characteristics) activates the operation. Data read from locations 0000H and 0001H represent the manufacturer's code and the device code, respectively.

The manufacturer- and device-codes can also be read via the command register, for instances where the 28F512 is erased and reprogrammed in the target system. Following a write of 90H to the command register, a read from address location 0000H outputs the manufacturer code (89H). A read from address 0001H outputs the device code (B8H).

Write

Device erasure and programming are accomplished via the command register, when high voltage is applied to the V_{PP} pin. The contents of the register serve as input to the internal state-machine. The state-machine outputs dictate the function of the device.

The command register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

The command register is written by bringing Write-Enable to a logic-low level (V_{IL}), while Chip-Enable is low. Addresses are latched on the falling edge of Write-Enable, while data is latched on the rising edge of the Write-Enable pulse. Standard microprocessor write timings are used.

Refer to A.C. Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the V_{PP} pin, the contents of the command register default to 00H, enabling read-only operations.

Placing high voltage on the V_{PP} pin enables read/write operations. Device operations are selected by writing specific data patterns into the command register. Table 3 defines these 28F512 register commands.

Table 3. Command Definitions

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	1	Write	X	00H			
Read Intelligent Identifier Code(4)	2	Write	X	90H	Read	IA	ID
Set-up Erase/Erase(5)	2	Write	X	20H	Write	X	20H
Erase Verify(5)	2	Write	EA	A0H	Read	X	EVD
Set-up Program/Program(6)	2	Write	X	40H	Write	PA	PD
Program Verify(6)	2	Write	X	C0H	Read	X	PVD
Reset(7)	2	Write	X	FFH	Write	X	FFH

NOTES:
1. Bus operations are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   PA = Address of memory location to be programmed.
   Addresses are latched on the falling edge of the Write-Enable pulse.
3. ID = Data read from location IA during device identification (Mfr = 89H, Device = B8H).
   EVD = Data read from location EA during erase verify.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of Write-Enable.
   PVD = Data read from location PA during program verify. PA is latched on the Program command.
4. Following the Read Intelligent ID command, two read operations access manufacturer and device codes.
5. Figure 5 illustrates the Quick-Erase™ algorithm.
6. Figure 4 illustrates the Quick-Pulse Programming™ algorithm.
7. The second bus cycle must be followed by the desired command register write.

Read Command

While V_{PP} is high, for erasure and programming, memory contents can be accessed via the read command. The read operation is initiated by writing 00H into the command register. Microprocessor read cycles retrieve array data. The device remains enabled for reads until the command register contents are altered.

The default contents of the register upon V_{PP} power-up is 00H. This default value ensures that no spurious alteration of memory contents occurs during the V_{PP} power transition. Where the V_{PP} supply is hard-wired to the 28F512, the device powers-up and remains enabled for reads until the command-register contents are changed. Refer to the A.C. Read Characteristics and Waveforms for specific timing parameters.

Intelligent Identifier Command

Flash-memories are intended for use in applications where the local CPU alters memory contents. As such, manufacturer- and device-codes must be accessible while the device resides in the target system. PROM programmers typically access signature codes by raising A9 to a high voltage. However, multiplexing high voltage onto address lines is not a desired system-design practice.

The 28F512 contains an Intelligent Identifier operation to supplement traditional PROM-programming methodology. The operation is initiated by writing 90H into the command register. Following the command write, a read cycle from address 0000H retrieves the manufacturer code of 89H. A read cycle from address 0001H returns the device code of B8H. To terminate the operation, it is necessary to write another valid command into the register.

Set-up Erase/Erase Commands

Set-up Erase is a command-only operation that stages the device for electrical erasure of all bytes in the array. The set-up erase operation is performed by writing 20H to the command register.

To commence chip-erasure, the erase command (20H) must again be written to the register. The erase operation begins with the rising edge of the Write-Enable pulse and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, chip-erasure can only occur when high voltage is applied to the V_{PP} pin. In the absence of this high voltage, memory contents are protected against erasure. Refer to A.C. Erase Characteristics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all bytes of the array in parallel. After each erase operation, all bytes must be verified. The erase verify operation is initiated by writing A0H into the command register. The address for the byte to be verified must be supplied as it is latched on the falling edge of the Write-Enable pulse. The register write terminates the erase operation with the rising edge of its Write-Enable pulse.

The 28F512 applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased.

The erase-verify command must be written to the command register prior to each byte verification to latch its address. The process continues for each byte in the array until a byte does not return FFH data, or the last address is accessed.

In the case where the data read is not FFH, another erase operation is performed. (Refer to Set-up Erase/Erase). Verification then resumes from the address of the last-verified byte. Once all bytes in the array have been verified, the erase step is complete. The device can be programmed. At this point, the verify operation is terminated by writing a valid command (e.g. Program Set-up) to the command register. Figure 5, the Quick-Erase™ algorithm, illustrates how commands and bus operations are combined to perform electrical erasure of the 28F512. Refer to A.C. Erase Characteristics and Waveforms for specific timing parameters.

Set-up Program/Program Commands

Set-up program is a command-only operation that stages the device for byte programming. Writing 40H into the command register performs the set-up operation.

Once the program set-up operation is performed, the next Write-Enable pulse causes a transition to an active programming operation. Addresses are internally latched on the falling edge of the Write-Enable pulse. Data is internally latched on the rising edge of the Write-Enable pulse. The rising edge of Write-Enable also begins the programming operation. The programming operation terminates with the next rising edge of Write-Enable, used to write the program-verify command. Refer to A.C. Program-

ming Characteristics and Waveforms for specific timing parameters.

Program-Verify Command

The 28F512 is programmed on a byte-by-byte basis. Byte programming may occur sequentially or at random. Following each programming operation, the byte just programmed must be verified.

The program-verify operation is initiated by writing C0H into the command register. The register write terminates the programming operation with the rising edge of its Write-Enable pulse. The program-verify operation stages the device for verification of the byte last programmed. No new address information is latched.

The 28F512 applies an internally-generated margin voltage to the byte. A microprocessor read cycle outputs the data. A successful comparison between the programmed byte and true data means that the byte is successfully programmed. Programming then proceeds to the next desired byte location. Figure 4, the 28F512 Quick-Pulse Programming™ algorithm, illustrates how commands are combined with bus operations to perform byte programming. Refer to A.C. Programming Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or program-command sequences. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

EXTENDED ERASE/PROGRAM CYCLING

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled—an expensive solution.

Intel has designed extended cycling capability into its ETOX-II flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field is one-tenth that of common EEPROMs, minimizing the probability of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 MV/cm lower than EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure—increasing time to wearout by a factor of 100,000,000.

The device is programmed and erased using Intel's Quick-Pulse Programming and Quick-Erase algorithms. Intel's algorithmic approach uses a series of operations (pulses), along with byte verification, to completely and reliably erase and program the device.

QUICK-PULSE PROGRAMMING ALGORITHM

The Quick-Pulse Programming algorithm uses programming operations of 10 µs duration. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The algorithm allows for up to 25 programming operations per byte, although most bytes verify on the first or second operation. The entire sequence of programming and byte verification is performed with V_{PP} at high voltage. Figure 4 illustrates the Quick-Pulse Programming algorithm.

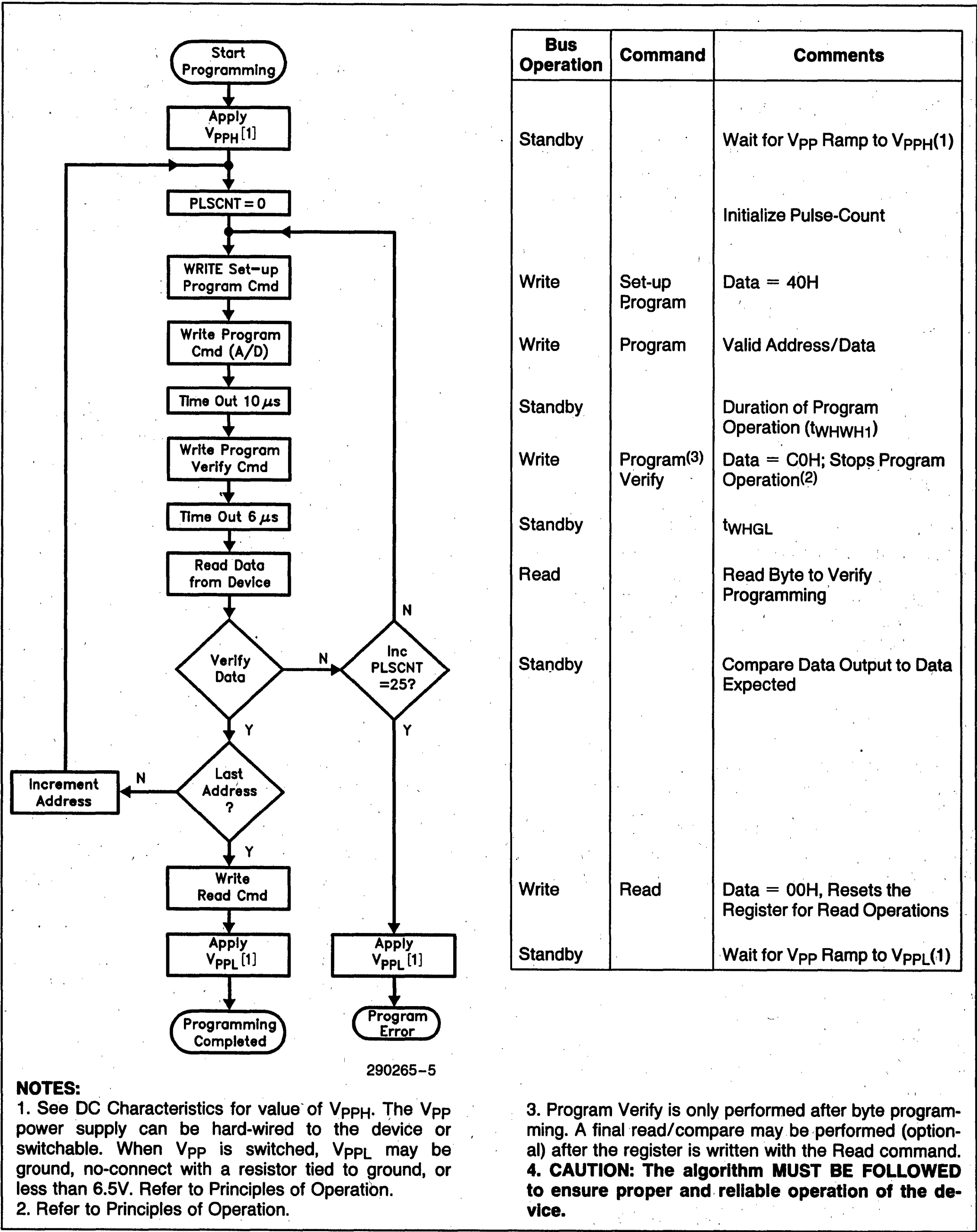
QUICK-ERASE ALGORITHM

Intel's Quick-Erase algorithm yields fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the Quick-Pulse Programming algorithm, to simultaneously remove charge from all bits in the array.

Erasure begins with a read of memory contents. The 28F512 is erased when shipped from the factory. Reading FFH data from the device would immediately be followed by device programming.

For devices being erased and reprogrammed, uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (Data = 00H). This is accomplished, using the Quick-Pulse Programming algorithm, in approximately one second.

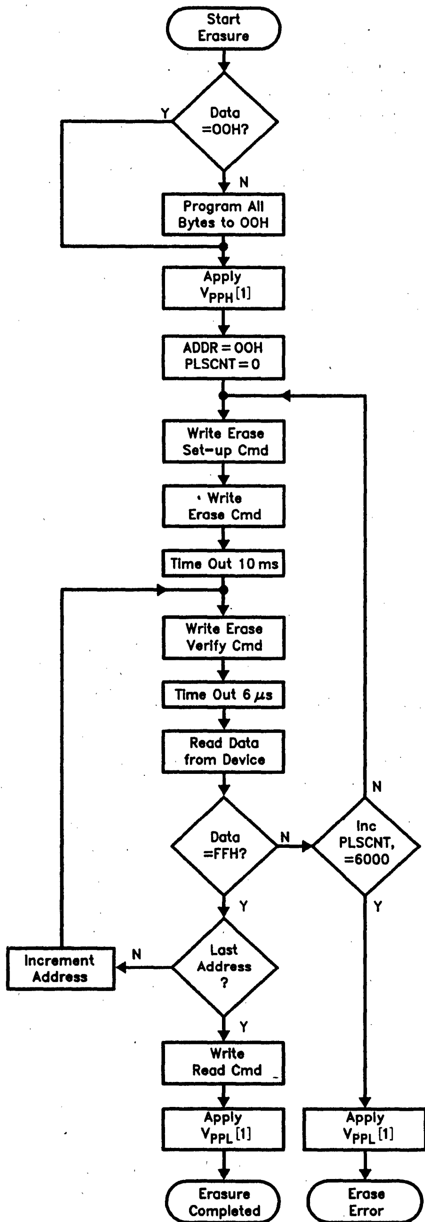
Erase execution then continues with an initial erase operation. Erase verification (data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. With each erase operation, an increasing number of bytes verify to the erased state. Erase efficiency may be improved by storing the address of the last byte verified in a register. Following the next erase operation, verification starts at that stored address location. Erasure typically occurs in one seconds. Figure 5 illustrates the Quick-Erase algorithm.



Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Pulse-Count
Write	Set-up Program	Data = 40H
Write	Program	Valid Address/Data
Standby		Duration of Program Operation (t_{WHWH1})
Write	Program(3) Verify	Data = C0H; Stops Program Operation(2)
Standby		t_{WHGL}
Read		Read Byte to Verify Programming
Standby		Compare Data Output to Data Expected
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

NOTES:
1. See DC Characteristics for value of V_{PPH}. The V_{PP} power supply can be hard-wired to the device or switchable. When V_{PP} is switched, V_{PPL} may be ground, no-connect with a resistor tied to ground, or less than 6.5V. Refer to Principles of Operation.
2. Refer to Principles of Operation.
3. Program Verify is only performed after byte programming. A final read/compare may be performed (optional) after the register is written with the Read command.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

Figure 4. 28F512 Quick-Pulse Programming Algorithm



Bus Operation	Command	Comments
		Entire memory must = 00H before erasure
		Use Quick-Pulse Programming Algorithm (Figure 4)
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Addresses and Pulse-Count
Write	Set-up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase(3) Verify	Addr = Byte to Verify; Data = A0H; Stops Erase Operation(2)
Standby		t_{WHGL}
Read		Read Byte to Verify Erasure
Standby		Compare Output to FFH Increment Pulse-Count
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

NOTES:

1. See DC Characteristics for value of V_{PPH}. The V_{PP} power supply can be hard-wired to the device or switchable. When V_{PP} is switched, V_{PPL} may be ground, no-connect with a resistor tied to ground, or less than 6.5V. Refer to Principles of Operation.
2. Refer to Principles of Operation.
3. Erase Verify is performed only after chip-erasure. A final read/compare may be performed (optional) after the register is written with the read command.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

Figure 5. 28F512 Quick-Erase Algorithm

DESIGN CONSIDERATIONS

Two-Line Output Control

Flash-memories are often used in larger memory arrays. Intel provides two read-control inputs to accommodate multiple memory connections. Two-line control provides for:

a. the lowest possible memory power dissipation and,

b. complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address-decoder output should drive chip-enable, while the system's read signal controls all flash-memories and other parallel memories. This assures that only enabled memory devices have active outputs, while deselected devices maintain the low power standby condition.

Power Supply Decoupling

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current (I_{CC}) issues—standby, active, and transient current peaks produced by falling and rising edges of chip-enable. The capacitive and inductive loads on the device outputs determine the magnitudes of these peaks.

Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 μF ceramic capacitor connected between V_{CC} and V_{SS}, and between V_{PP} and V_{SS}.

Place the high-frequency, low-inherent-inductance capacitors as close as possible to the devices. Also, for every eight devices, a 4.7 μF electrolytic capacitor should be placed at the array's power supply connection, between V_{CC} and V_{SS}. The bulk capacitor will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

V_{PP} Trace on Printed Circuit Boards

Programming flash-memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the V_{PP} power supply trace. The V_{PP} pin supplies the memory cell current for programming. Use similar trace widths and layout considerations given the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots.

Power Up/Down Protection

The 28F512 is designed to offer protection against accidental erasure or programming, caused by spurious system-level signals that may exist during power transitions. Also, with its control register architecture, alteration of memory contents only occurs after successful completion of the two-step command sequences. Power supply sequencing is not required. Internal circuitry of the 28F512 ensures that the command register architecture is reset to the read mode on power up.

A system designer must guard against active writes for V_{CC} voltages above the V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will prohibit writes.

The control register architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

28F512 Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases the usable battery life of your system because the 28F512 does not consume any power to retain code or data when the system is off. Table 4 illustrates the power dissipated when updating the 28F512.

Table 4. 28F512 Typical Update Power Dissipation(4)

Operation	Power Dissipation (Watt-Seconds)	Notes
Array Program/ Program Verify	0.085	1
Array Erase/ Erase Verify	0.092	2
One Complete Cycle	0.262	3

NOTES:
1. Formula to calculate typical Program/Program Verify Power = [V_{PP} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{PP2} typical + t_{WHGL} × I_{PP4} typical)] + [V_{CC} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{CC2} typical + t_{WHGL} × I_{CC4} typical)].
2. Formula to calculate typical Erase/Erase Verify Power = [V_{PP} (I_{PP3} typical × t_{ERASE} typical + I_{PP5} typical × t_{WHGL} × # Bytes)] + [V_{CC} (I_{CC3} typical × t_{ERASE} typical + I_{CC5} typical × t_{WHGL} × # Bytes)].
3. Once Complete Cycle = Array Preprogram + Array Erase + Program.
4. "Typicals" are not guaranteed, but based on a limited number of samples from production lots.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read . . . . $-40°C$ to $+125°C$(1)
During Erase/Program . . . . $-40°C$ to $+125°C$

Temperature Under Bias . . . . $-40°C$ to $+125°C$

Storage Temperature . . . . $-65°C$ to $+150°C$

Voltage on Any Pin with
Respect to Ground . . . . $-2.0V$ to $+7.0V$(2)

Voltage on Pin A_9 with
Respect to Ground . . . . $-2.0V$ to $+13.5V$(2, 3)

V_{PP} Supply Voltage with
Respect to Ground
During Erase/Program . . . . $-2.0V$ to $+14.0V$(2, 3)

V_{CC} Supply Voltage with
Respect to Ground . . . . $-2.0V$ to $+7.0V$(2)

Output Short Circuit Current . . . . 100 mA(4)

Maximum Junction Temperature (T_J) . . . . 140°C

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for automotive product as defined by this specification.
2. Minimum DC input voltage is $-0.5V$. During transitions, inputs may undershoot to $-2.0V$ for periods less than 20 ns. Maximum DC voltage on output pins is $V_{CC} + 0.5V$, which may overshoot to $V_{CC} + 2.0V$ for periods less than 20 ns.
3. Maximum DC voltage on A_9 or V_{PP} may overshoot to $+14.0V$ for periods less than 20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature	−40	125	°C	For Read-Only and Read/Write Operations
V_{CC}	V_{CC} Supply Voltage	4.50	5.50	V	

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1			1.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ $f = 6$ MHz, $I_{OUT} = 0$ mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	30	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10	μA	$V_{PP} \leq V_{CC}$

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE (Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ	Max		
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage				0.45	V	$I_{OL} = 2.1$ mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	$I_{OH} = -2.5$ mA $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 inteligent Identifier Voltage		11.50		13.00	V	$A_9 = V_{ID}$
I_{ID}	V_{CC} ID Current	1		10	30	mA	$A_9 = V_{ID}$
	V_{PP} ID Current			90	500	μA	
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

5

DC CHARACTERISTICS—CMOS COMPATIBLE

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		50	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC} \pm 0.2V$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, $I_{OUT} = 0$ mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	30	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress

DC CHARACTERISTICS—CMOS COMPATIBLE (Continued)

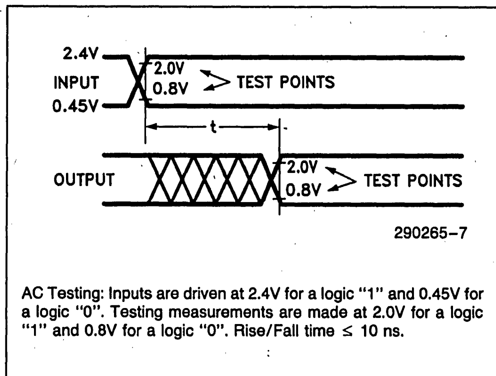
Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typ	Max		
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10		$V_{PP} \leq V_{CC}$
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Program Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 2.1 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		0.85 V_{CC}			V	I_{OH} = −2.5 mA, $V_{CC} = V_{CC}$ Min
V_{OH2}			V_{CC} − 0.4				I_{OH} = −100 μA, $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifier Voltage		11.50		13.00	V	
I_{ID}	V_{CC} ID Current	1		10	30	mA	A_9 = ID
I_{ID}	V_{PP} ID Current	1		90	500	mA	A_9 = ID
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Programs are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	V_{PP} = 12.0V
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

CAPACITANCE(3) T_A = 25°C, f = 1.0 MHz

Symbol	Parameter	Limits		Unit	Conditions
		Min	Max		
C_{IN}	Address/Control Capacitance		8	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance		12	pF	V_{OUT} = 0V

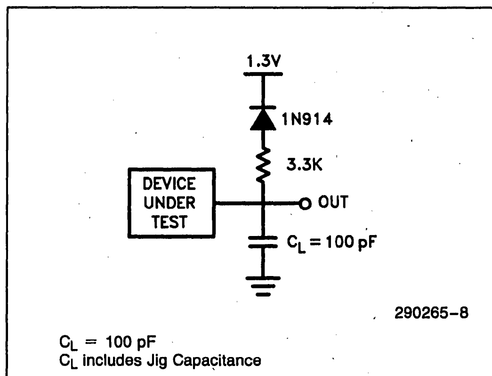
NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C.
2. Not 100% tested: characterization data available.
3. Sampled, not 100% tested.
4. "Typicals" are not guaranteed, but are based on a limited number of samples from production lots.

AC TESTING INPUT/OUTPUT WAVEFORM



AC Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Testing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0". Rise/Fall time ≤ 10 ns.

AC TESTING LOAD CIRCUIT



C_L = 100 pF
C_L includes Jig Capacitance

AC TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ...... 10 ns
Input Pulse Levels .... 0.45V and 2.4V
Input Timing Reference Level .... 0.8V and 2.0V
Output Timing Reference Level ...... 0.8V and 2.0V

AC CHARACTERISTICS—Read-Only Operations(2)

Versions		Notes	28F512-120		28F512-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}/t_{RC}	Read Cycle Time	3	120		150		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time			120		150	ns
t_{AVQV}/t_{ACC}	Address Access Time			120		150	ns
t_{GLQV}/t_{OE}	Output Enable Access Time			50		55	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	3	0		0		ns
t_{EHQZ}	Chip Disable to Output in High Z	3		50		55	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	3	0		0		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	4		30		35	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change(1)	3	0		0		ns
t_{WHGL}	Write Recovery Time before Read		6		6		µs

NOTES:
1. Whichever occurs first.
2. Rise/Fall Time ≤ 10 ns.
3. Not 100% tested characterization data available.
4. Guaranteed by design.

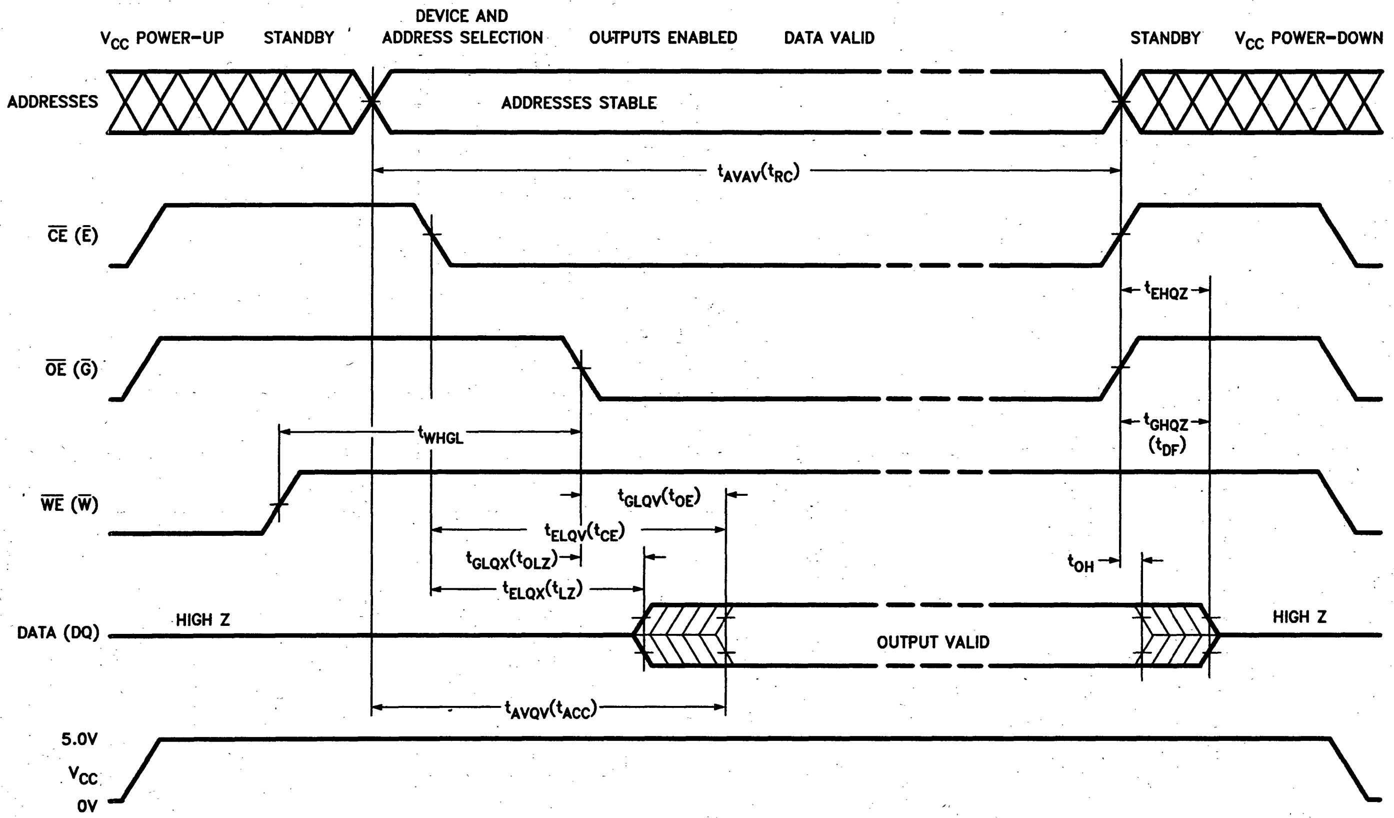


Figure 6. AC Waveforms for Read Operations

AC CHARACTERISTICS—Write/Erase/Program Operations(1, 3)

Versions		Notes	28F512-120		28F512-150		Unit
Symbol	Characteristic		Min	Max	Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		120		150		ns
t_{AVWL}/t_{AS}	Address Set-Up Time		0		0		ns
t_{WLAX}/t_{AH}	Address Hold Time	2	60		60		ns
t_{DVWH}/t_{DS}	Data Set-up Time		50		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		10		ns
t_{WHGL}	Write Recovery Time before Read		6		6		μs
t_{GHWL}	Read Recovery Time before Write		0		0		μs
t_{ELWL}/t_{CS}	Chip Enable Set-Up Time before Write	2	20		20		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time		0		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width(2)	2	80		80		ns
t_{ELEH}	Alternative Write(2) Pulse Width	2	80		80		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		20		ns
t_{WHWH1}	Duration of Programming Operation	4	10		10		μs
t_{WHWH2}	Duration of Erase Operation	4	9.5		9.5		ms
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low		1.0		1.0		ms

NOTES:
1. Read timing characteristics during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of Chip-Enable and Write-Enable. In systems where Chip-Enable defines the write pulse width (within a longer Write-Enable timing waveform) all set-up, hold, and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.
3. Rise/Fall time ≤ 10 ns.
4. The integrated stop timer terminates the programming/erase operations, thereby eliminating the need for a maximum specification.

ERASE AND PROGRAMMING PERFORMANCE

Parameter	Limits			Notes	Unit	Comments
	Min	Typ	Max			
Chip Erase Time		1	60	1, 3, 4, 6	Sec	Excludes 00H Programming Prior to Erasure
Chip Program Time		1	6.25	1, 2, 4	Sec	Excludes System-Level Overhead
Erase/Program Cycles	1,000	100,000		1, 5	Cycles	

NOTES:
1. "Typicals" are not guaranteed, but based on a limited number of samples taken from production lots. Data taken at $T = 25°C$, $V_{PP} = 12.0V$, $V_{CC} = 5.0V$.
2. Minimum byte programming time excluding system overhead is 16 μs (10 μs program + 6 μs write recovery), while maximum is 400 μs/byte (16 μs × 25 loops allowed by algorithm). Max chip programming time is specified lower than the worst case allowed by the programming algorithm since most bytes program significantly faster than the worst case byte.
3. Excludes 00H programming prior to erasure.
4. Excludes system-level overhead.
5. Refer to RR-60 "ETOX Flash Memory Reliability Data Summary" for typical cycling data and failure rate calculations.
6. Maximum erase specification is determined by algorithmic limit and accounts for cumulative effect of erasure at $T = -40°C$, 1,000 cycles, $V_{PP} = 11.4V$, $V_{CC} = 4.5V$.

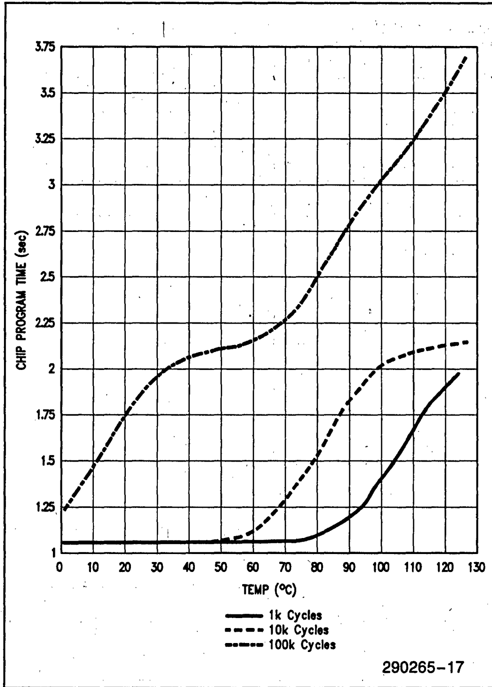


Figure 7. 28F512 Typical Program Time at 12V

See Note 1, Page 5-43.

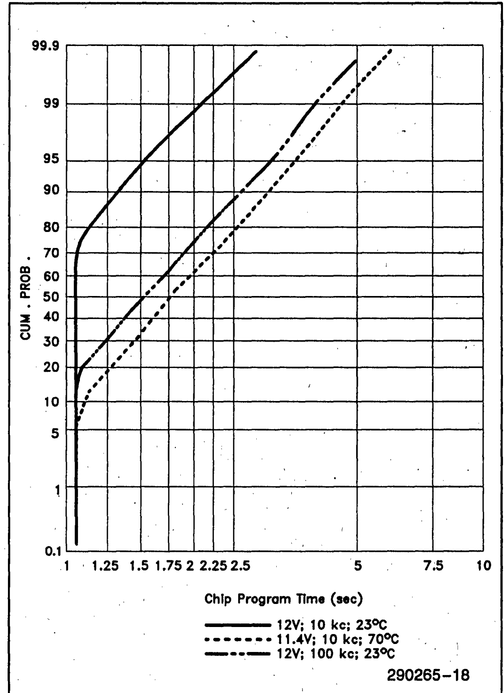
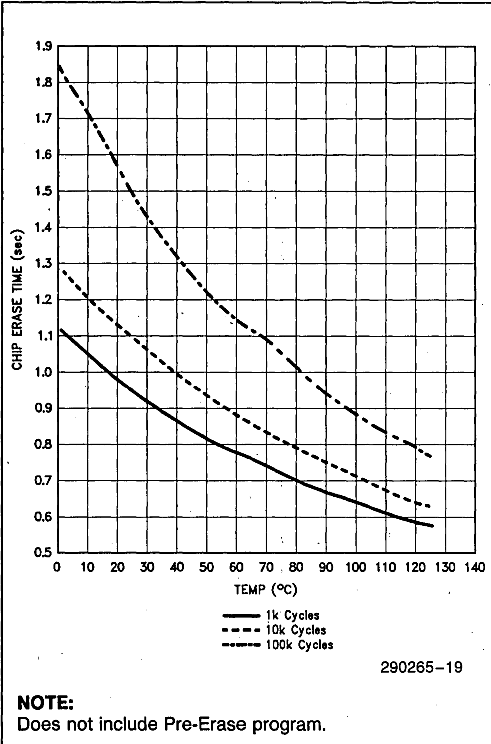


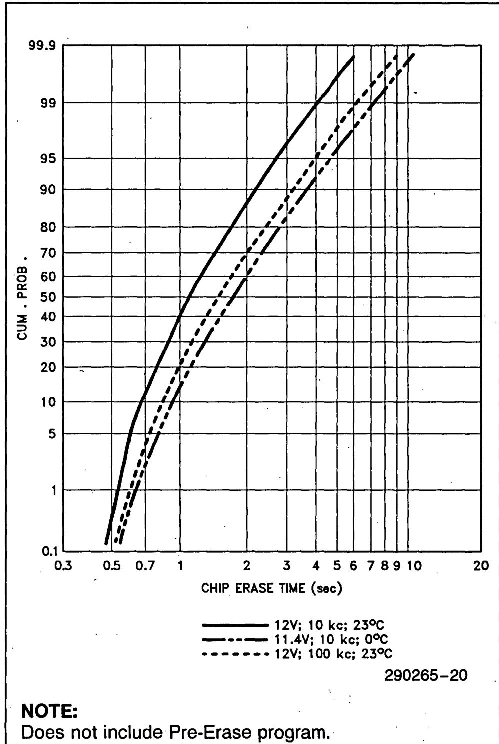
Figure 8. 28F512 Typical Programming Capability



NOTE:
Does not include Pre-Erase program.

Figure 9. 28F512 Typical Erase Time at 12V

See Note 1, Page 5-43.



NOTE:
Does not include Pre-Erase program.

Figure 10. 28F512 Typical Erase Capability

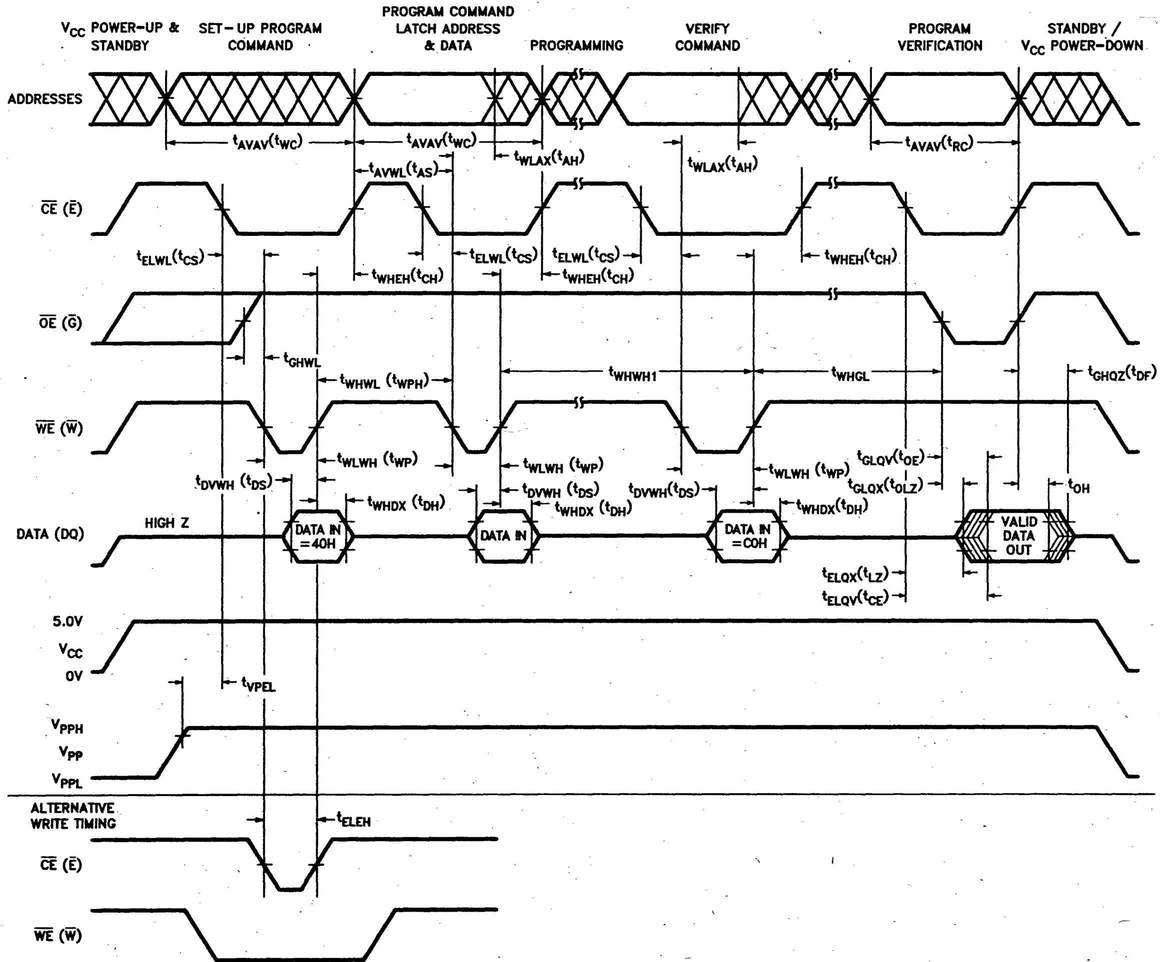


Figure 11. AC Waveforms for Programming Operations

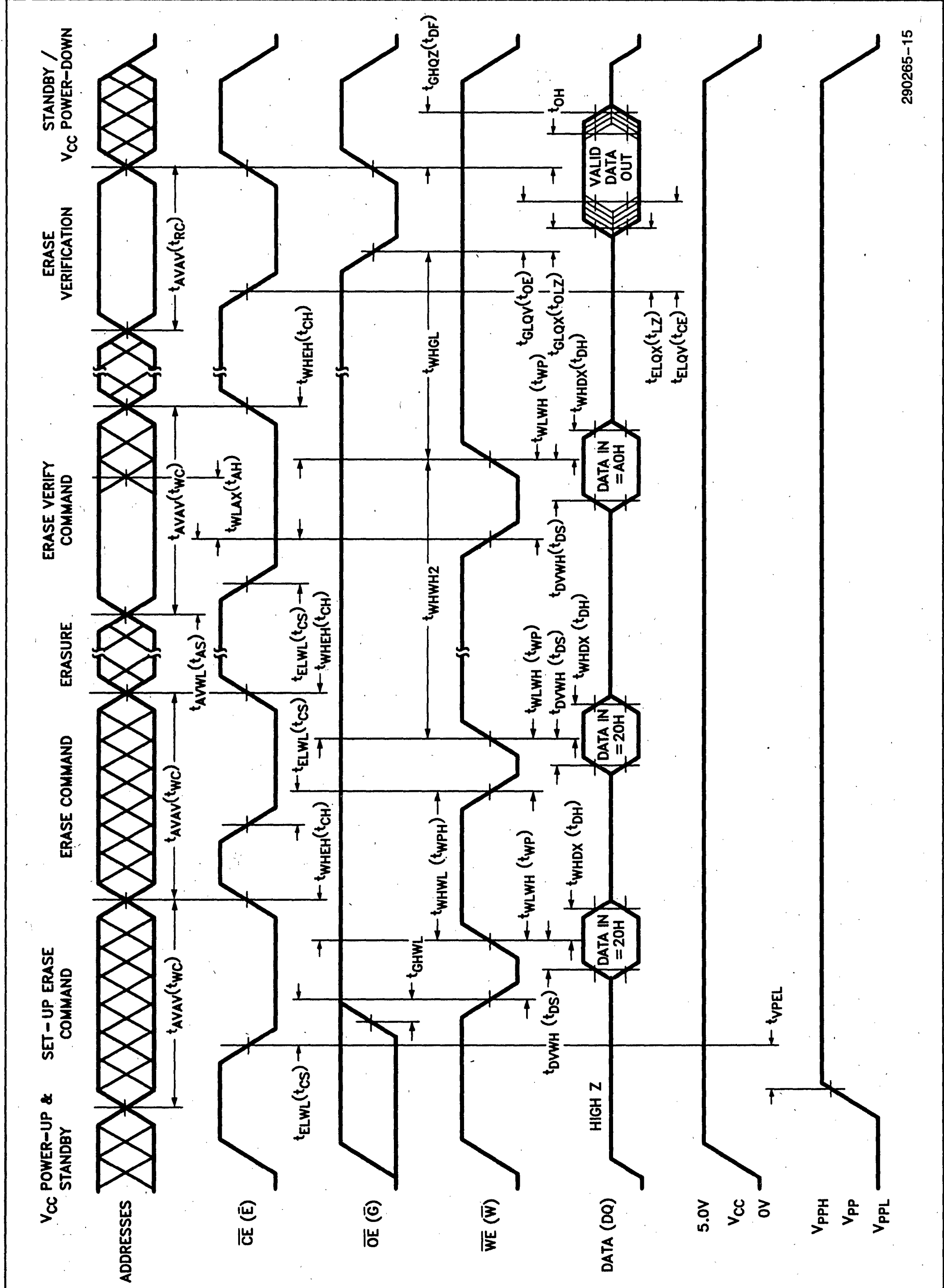
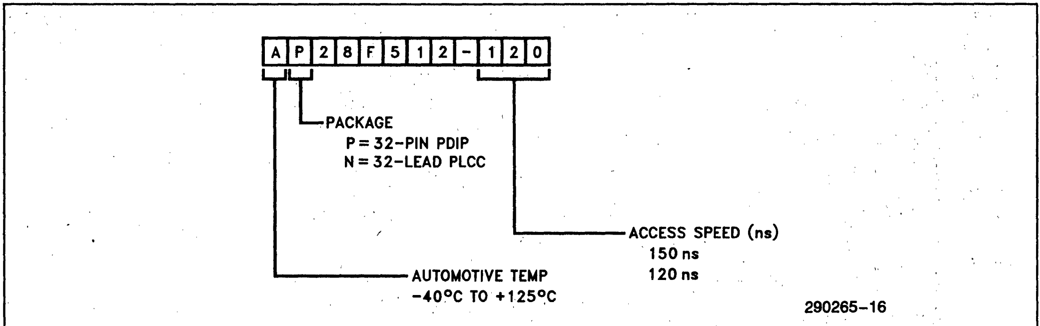


Figure 12. AC Waveforms for Erase Operations

Ordering Information



Valid Combinations:

AP28F512-150	AN28F512-150
AP28F512-120	AN28F512-120

ADDITIONAL INFORMATION

	Order Number
ER-20, "ETOX™ II Flash Memory Technology"	294005
ER-23, "The Intel A28F512 Flash Memory"	294007
RR-60, "ETOX™ Flash Memory Reliability Data Summary"	293002
AP-316, "Using Flash Memory for In-System Reprogrammable Nonvolatile Storage"	292046
AP-325 "Guide to Flash Memory Reprogramming"	292059

REVISION HISTORY

Number	Description
002	Changed Erase/Program Cycles to 1,000 minimum.
003	Added 120 ns speed

A28F010
1024K (128K x 8) CMOS FLASH MEMORY
(Automotive)

- **Extended Automotive Temperature Range: −40°C to +125°C**
- **Flash Electrical Chip-Erase**
  - **—1 Second Typical Chip-Erase**
- **Quick-Pulse Programming Algorithm**
  - **—10 μs Typical Byte-Program**
  - **—2 Second Chip-Program**
- **1,000 Erase/Program Cycles Minimum over Automotive Temperature Range**
- **12.0V ±5% V_{PP}**
- **High-Performance Read**
  - **—150 ns Maximum Access Time**
- **CMOS Low Power Consumption**
  - **—30 mA Maximum Active Current**
  - **—100 μA Maximum Standby Current**
- **Integrated Program/Erase Stop Timer**
- **Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface**
- **Noise Immunity Features**
  - **—±10% V_{CC} Tolerance**
  - **—Maximum Latch-Up Immunity through EPI Processing**
- **ETOX™-II Flash Nonvolatile Memory Technology**
  - **—EPROM-Compatible Process Base**
  - **—High-Volume Manufacturing Experience**
- **JEDEC-Standard Pinouts**
  - **—32-Pin Plastic DIP**
  - **—32-Lead PLCC**

(See Packaging Spec., Order #231369)

Intel's 28F010 CMOS flash memory offers the most cost-effective and reliable alternative for read/write random access nonvolatile memory. The 28F010 adds electrical chip-erasure and reprogramming to familiar EPROM technology. Memory contents can be rewritten: in a test socket; in a PROM-programmer socket; on-board during subassembly test; in-system during final test; and in-system after-sale. The 28F010 increases memory flexibility, while contributing to time- and cost-savings.

The 28F010 is a 1024-kilobit nonvolatile memory organized as 131,072 bytes of 8 bits. Intel's 28F010 is offered in 32-pin Plastic DIP or 32-lead PLCC packages. Pin assignments conform to JEDEC standards.

Extended erase and program cycling capability is designed into Intel's ETOX™ II (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional EEPROMs. With the 12.0V V_{PP} supply, the 28F010 performs a minimum of 1,000 erase and program cycles well within the time limits of the Quick-Pulse Programming™ and Quick-Erase™ algorithms.

Intel's 28F010 employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 150 nanosecond access time provides no-WAIT-state performance for a wide range of microprocessors and microcontrollers. Maximum standby current of 100 μA translates into power savings when the device is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from −1V to V_{CC} + 1V.

With Intel's ETOX II process base, the 28F010 levers years of EPROM experience to yield the highest levels of quality, reliability, and cost-effectiveness.

In order to meet the rigorous environmental requirements of automotive applications, Intel offers the 28F010 in extended automotive temperature range. Read and write characteristics are guaranteed over the range of −40°C to +125°C ambient.

August 1992
Order Number: 290266-003

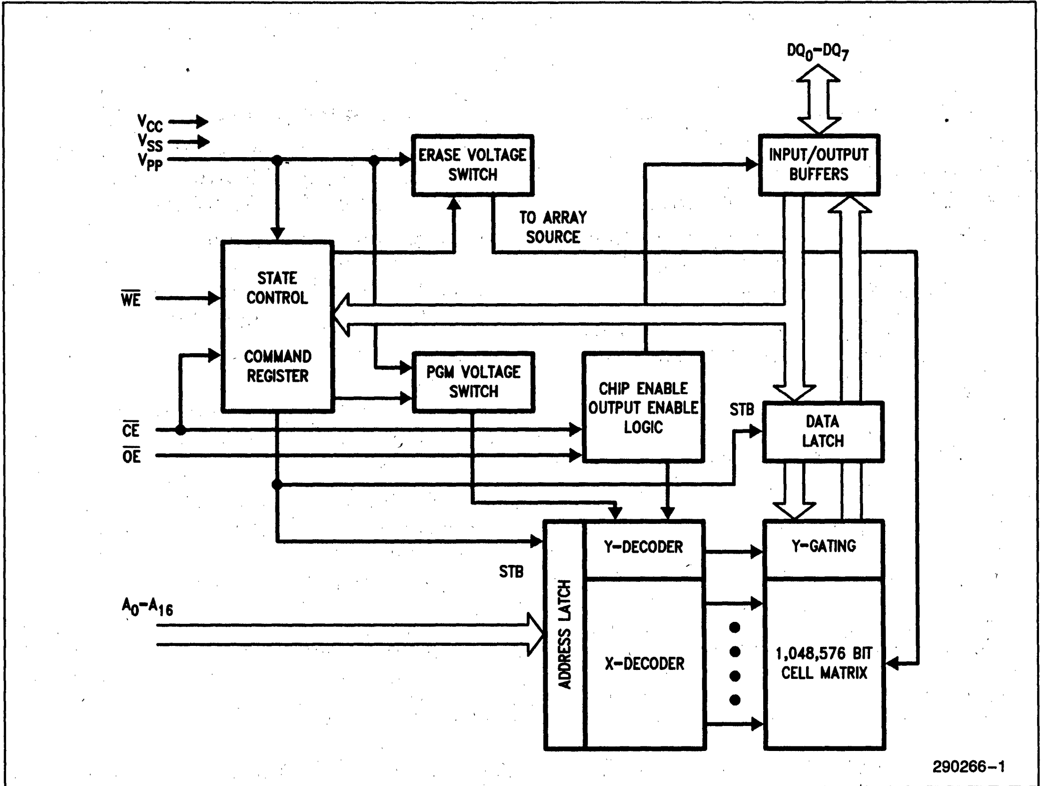


Figure 1. 28F010 Block Diagram

AUTOMOTIVE TEMPERATURE FLASH MEMORIES

The Intel Automotive FLASH memories have received additional processing to enhance product characteristics. The automotive temperature range is −40°C to +125°C during the read/write/erase/program operations.

Speed Versions	Packaging Options	
	Plastic DIP	PLCC
150	AP	AN

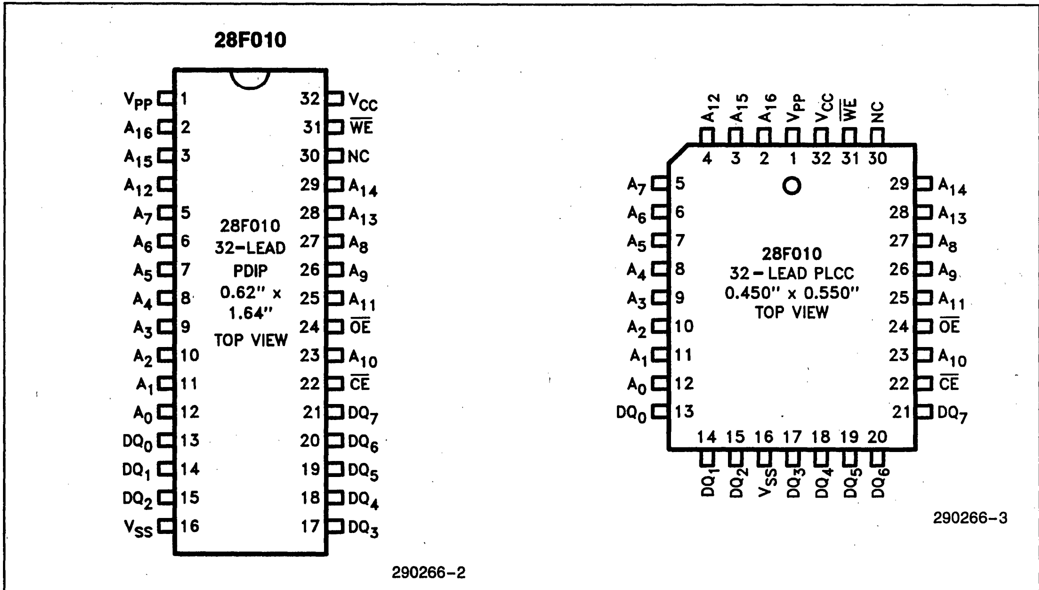


Figure 2. 28F010 Pin Configurations

Table 1. Pin Description

Symbol	Type	Name and Function
A_0–A_{16}	INPUT	**ADDRESS INPUTS** for memory addresses. Addresses are internally latched during a write cycle.
DQ_0–DQ_7	INPUT/OUTPUT	**DATA INPUT/OUTPUT:** Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
$\overline{CE}$	INPUT	**CHIP ENABLE:** Activates the device's control logic, input buffers, decoders and sense amplifiers. $\overline{CE}$ is active low; $\overline{CE}$ high deselects the memory device and reduces power consumption to standby levels.
$\overline{OE}$	INPUT	**OUTPUT ENABLE:** Gates the devices output through the data buffers during a read cycle. $\overline{OE}$ is active low.
$\overline{WE}$	INPUT	**WRITE ENABLE:** Controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the $\overline{WE}$ pulse. **Note:** With $V_{PP} \leq 6.5V$, memory contents cannot be altered.
V_{PP}		**ERASE/PROGRAM POWER SUPPLY** for writing the command register, erasing the entire array, or programming bytes in the array.
V_{CC}		**DEVICE POWER SUPPLY** (5V ±10%)
V_{SS}		**GROUND**
NC		**NO INTERNAL CONNECTION** to device. Pin may be driven or left floating.

5

APPLICATIONS

The 28F010 flash-memory adds electrical chip-erasure and reprogrammability to EPROM non-volatility and ease of use. The 28F010 is ideal for storing code or data-tables in applications where periodic updates are required. The 28F010 also serves as a dense, nonvolatile data acquisition and storage medium.

The need for code updates pervades all phases of a system's life—from prototyping to system manufacture to after-sale service. In the factory, during prototyping, revisions to control code necessitate ultraviolet erasure and reprogramming of EPROM-based prototype codes. The 28F010 replaces the 15- to 20-minute ultraviolet erasure with one-second electrical erasure. Electrical chip-erasure and reprogramming occur in the same workstation or PROM-programmer socket.

Diagnostics, performed at subassembly or final assembly stages, often require the socketing of EPROMs. Socketed test codes are ultimately replaced with EPROMs containing the final program. With electrical chip-erasure and reprogramming, the 28F010 is soldered to the circuit board. Test codes are programmed into the 28F010 as it resides on the circuit board. Ultimately, the final code can be downloaded to the device. The 28F010's in-circuit alterability eliminates unnecessary handling and less-reliable socketed connections, while adding greater test flexibility.

Material and labor costs associated with code changes increase at higher levels of system integration—the most costly being code updates after sale. Code "bugs", or the desire to augment system functionality, prompt after-sale code updates. Field revisions to EPROM-based code require the removal of EPROM components or entire boards.

Designing with the in-circuit alterable 28F010 eliminates socketed memories, reduces overall material costs, and drastically cuts the labor costs associated with code updates. With the 28F010, code updates are implemented locally via an edge-connector, or remotely over a serial communication link.

The 28F010's electrical chip-erasure, byte reprogrammability, and complete nonvolatility fit well with data accumulation needs. Electrical chip-erasure gives the designer a "blank-slate" in which to log data. Data can be periodically off-loaded for analysis—erasing the slate and repeating the cycle. Or, multiple devices can maintain a "rolling window" of accumulated data.

With high density, nonvolatility, and extended cycling capability, the 28F010 offers an innovative alternative for mass storage. Integrating main memory and backup storage functions into directly executable flash memory boosts system performance, shrinks system size, and cuts power consumption. Reliability exceeds that of electromechanical media, with greater durability in extreme environmental conditions.

A high degree of on-chip feature integration simplifies memory-to-processor interfacing. Figure 3 depicts two 28F010s tied to the 80C186 system bus. The 28F010's architecture minimizes interface circuitry needed for complete in-circuit updates of memory contents.

With cost-effective in-system reprogramming and extended cycling capability, the 28F010 fills the functionality gap between traditional EPROMs and EEPROMs. EPROM-compatible specifications, straightforward interfacing, and in-circuit alterability allows designers to easily augment memory flexibility and satisfy the need for updatable nonvolatile storage in today's designs.

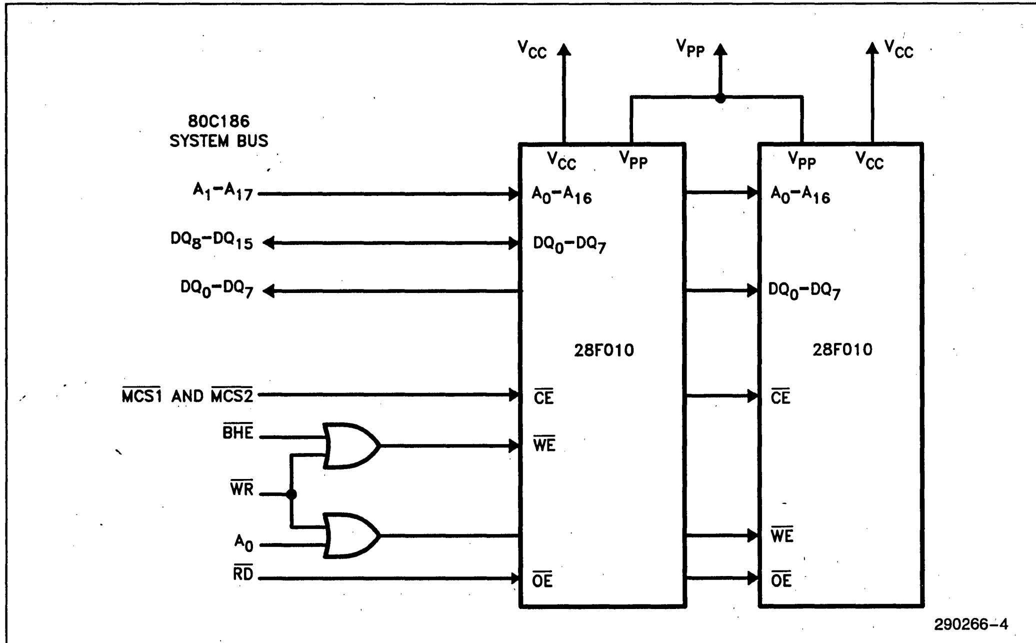


Figure 3. 28F010 in a 80C186 System

PRINCIPLES OF OPERATION

Flash-memory augments EPROM functionality with in-circuit electrical erasure and reprogramming. The 28F010 introduces a command register to manage this new functionality. The command register allows for: 100% TTL-level control inputs; fixed power supplies during erasure and programming; and maximum EPROM compatibility.

In the absence of high voltage on the V_{PP} pin, the 28F010 is a read-only memory. Manipulation of the external memory-control pins yields the standard EPROM read, standby, output disable, and Intelligent Identifier operations.

The same EPROM read, standby, and output disable operations are available when high voltage is applied to the V_{PP} pin. In addition, high voltage on V_{PP} enables erasure and programming of the device. All functions associated with altering memory contents—Intelligent Identifier, erase, erase verify, program, and program verify—are accessed via the command register.

Commands are written to the register using standard microprocessor write timings. Register contents serve as input to an internal state-machine which controls the erase and programming circuitry. Write cycles also internally latch addresses and data needed for programming or erase operations. With the appropriate command written to the register, standard microprocessor read timings output array data, access the Intelligent Identifier codes, or output data for erase and program verification.

Integrated Program/Erase Stop Timer

Successive command write cycles define the durations of program and erase operations; specifically, the program or erase time durations are normally terminated by associated program or erase verify commands. An integrated stop timer provides simplified timing control over these operations; thus eliminating the need for maximum program/erase timing specifications. Program and erase pulse durations are minimums only. When the stop timer terminates a program or erase operation, the device enters an inactive state and remains inactive until receiving the appropriate verify or reset command.

Write Protection

The command register is only alterable when V_{PP} is at high voltage. Depending upon the application, the system designer may choose to make the V_{PP} power supply switchable—available only when memory updates are desired. When high voltage is removed,

Table 2. 28F010 Bus Operations

	Pins / Operation	V_{PP}(1)	A_0	A_9	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_0-DQ_7
READ-ONLY	Read	V_{PPL}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out
	Output Disable	V_{PPL}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby	V_{PPL}	X	X	V_{IH}	X	X	Tri-State
	Intelligent Identifier (Mfr)(2)	V_{PPL}	V_{IL}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = 89H
	Intelligent Identifier (Device)(2)	V_{PPL}	V_{IH}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = B4H
READ/WRITE	Read	V_{PPH}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out(4)
	Output Disable	V_{PPH}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby(5)	V_{PPH}	X	X	V_{IH}	X	X	Tri-State
	Write	V_{PPH}	A_0	A_9	V_{IL}	V_{IH}	V_{IL}	Data In(6)

NOTES:
1. V_{PPL} may be ground, a no-connect with a resistor tied to ground, or $\leq$ 6.5V. V_{PPH} is the programming voltage specified for the device. Refer to D.C. Characteristics. When $V_{PP} = V_{PPL}$ memory contents can be read but not written or erased.
2. Manufacturer and device codes may also be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
3. V_{ID} is the Intelligent Identifier high voltage. Refer to DC Characteristics.
4. Read operations with $V_{PP} = V_{PPH}$ may access array data or the Intelligent Identifier codes.
5. With V_{PP} at high voltage, the standby current equals $I_{CC} + I_{PP}$ (standby).
6. Refer to Table 3 for valid Data-In during a write operation.
7. X can be V_{IL} or V_{IH}.

the contents of the register default to the read command, making the 28F010 a read-only memory. Memory contents cannot be altered.

Or, the system designer may choose to "hardwire" V_{PP}, making the high voltage supply constantly available. In this instance, all operations are performed in conjunction with the command register. The 28F010 is designed to accommodate either design practice, and to encourage optimization of the processor-memory interface.

The two-step Program/Erase write sequence to the Command Register provides additional software write protection.

BUS OPERATIONS

Read

The 28F010 has two control functions, both of which must be logically active, to obtain data at the outputs. Chip-Enable ($\overline{CE}$) is the power control and should be used for device selection. Output-Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Figure 6 illustrates read timing waveforms.

When V_{PP} is low (V_{PPL}), the read only operation is active. This permits reading the data in the array and outputting the Intelligent Identifier codes (see Table 2). When V_{PP} is high (V_{PPH}), the default condition of the device is the read only mode. This allows reading the data in the array. Further functionality is achieved though the Command Register as shown in Table 3.

Output Disable

With Output-Enable at a logic-high level (V_{IH}), output from the device is disabled. Output pins are placed in a high-impedance state.

Standby

With Chip-Enable at a logic-high level, the standby operation disables most of the 28F010's circuitry and substantially reduces device power consumption. The outputs are placed in a high-impedance state, independent of the Output-Enable signal. If the 28F010 is deselected during erasure, programming, or program/erase verification, the device draws active current until the operation is terminated.

Intelligent Identifier Operation

The Intelligent Identifier operation outputs the manufacturer code (89H) and device code (B4H). Programming equipment automatically matches the device with its proper erase and programming algorithms.

With Chip-Enable and Output-Enable at a logic low level, raising A9 to high voltage V_{ID} (see DC Characteristics) activates the operation. Data read from locations 0000H and 0001H represent the manufacturer's code and the device code, respectively.

The manufacturer- and device-codes can also be read via the command register, for instances where the 28F010 is erased and reprogrammed in the target system. Following a write of 90H to the command register, a read from address location 0000H outputs the manufacturer code (89H). A read from address 0001H outputs the device code (B4H).

Write

Device erasure and programming are accomplished via the command register, when high voltage is applied to the V_{PP} pin. The contents of the register serve as input to the internal state-machine. The state-machine outputs dictate the function of the device.

The command register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command.

The command register is written by bringing Write-Enable to a logic-low level (V_{IL}), while Chip-Enable is low. Addresses are latched on the falling edge of Write-Enable, while data is latched on the rising edge of the Write-Enable pulse. Standard microprocessor write timings are used.

Refer to A.C. Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

COMMAND DEFINITIONS

When low voltage is applied to the V_{PP} pin, the contents of the command register default to 00H, enabling read-only operations.

Placing high voltage on the V_{PP} pin enables read/write operations. Device operations are selected by writing specific data patterns into the command register. Table 3 defines these 28F010 register commands.

Table 3. Command Definitions

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	1	Write	X	00H			
Read Intelligent Identifier Codes(4)	2	Write	X	90H	Read	IA	ID
Set-up Erase/Erase(5)	2	Write	X	20H	Write	X	20H
Erase Verify(5)	2	Write	EA	A0H	Read	X	EVD
Set-up Program/Program(6)	2	Write	X	40H	Write	PA	PD
Program Verify(6)	2	Write	X	C0H	Read	X	PVD
Reset(7)	2	Write	X	FFH	Write	X	FFH

NOTES:
1. Bus operations are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code.
   EA = Address of memory location to be read during erase verify.
   PA = Address of memory location to be programmed.
   Addresses are latched on the falling edge of the Write-Enable pulse.
3. ID = Data read from location IA during device identification (Mfr = 89H, Device = B4H).
   EVD = Data read from location EA during erase verify.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of Write-Enable.
   PVD = Data read from location PA during program verify. PA is latched on the Program command.
4. Following the Read Intelligent ID command, two read operations access manufacturer and device codes.
5. Figure 5 illustrates the Quick-Erase Algorithm.
6. Figure 4 illustrates the Quick-Pulse Programming Algorithm.
7. The second bus cycle must be followed by the desired command register write.

Read Command

While V_{PP} is high, for erasure and programming, memory contents can be accessed via the read command. The read operation is initiated by writing 00H into the command register. Microprocessor read cycles retrieve array data. The device remains enabled for reads until the command register contents are altered.

The default contents of the register upon V_{PP} power-up is 00H. This default value ensures that no spurious alteration of memory contents occurs during the V_{PP} power transition. Where the V_{PP} supply is hard-wired to the 28F010, the device powers-up and remains enabled for reads until the command-register contents are changed. Refer to the A.C. Read Characteristics and Waveforms for specific timing parameters.

Intelligent Identifier Command

Flash-memories are intended for use in applications where the local CPU alters memory contents. As such, manufacturer- and device-codes must be accessible while the device resides in the target system. PROM programmers typically access signature codes by raising A9 to a high voltage. However, multiplexing high voltage onto address lines is not a desired system-design practice.

The 28F010 contains an inteligent Identifier operation to supplement traditional PROM-programming methodology. The operation is initiated by writing 90H into the command register. Following the command write, a read cycle from address 0000H retrieves the manufacturer code of 89H. A read cycle from address 0001H returns the device code of B4H. To terminate the operation, it is necessary to write another valid command into the register.

Set-up Erase/Erase Commands

Set-up Erase is a command-only operation that stages the device for electrical erasure of all bytes in the array. The set-up erase operation is performed by writing 20H to the command register.

To commence chip-erasure, the erase command (20H) must again be written to the register. The erase operation begins with the rising edge of the Write-Enable pulse and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-step sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, chip-erasure can only occur when high voltage is applied to the V_{PP} pin. In the absence of this high voltage, memory contents are protected against erasure. Refer to A.C. Erase Characteristics and Waveforms for specific timing parameters.

Erase-Verify Command

The erase command erases all bytes of the array in parallel. After each erase operation, all bytes must be verified. The erase verify operation is initiated by writing A0H into the command register. The address for the byte to be verified must be supplied as it is latched on the falling edge of the Write-Enable pulse. The register write terminates the erase operation with the rising edge of its Write-Enable pulse.

The 28F010 applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased.

The erase-verify command must be written to the command register prior to each byte verification to latch its address. The process continues for each byte in the array until a byte does not return FFH data, or the last address is accessed.

In the case where the data read is not FFH, another erase operation is performed. (Refer to Set-up Erase/Erase). Verification then resumes from the address of the last-verified byte. Once all bytes in the array have been verified, the erase step is complete. The device can be programmed. At this point, the verify operation is terminated by writing a valid command (e.g. Program Set-up) to the command register. Figure 5, the Quick-Erase™ algorithm, illustrates how commands and bus operations are combined to perform electrical erasure of the 28F010. Refer to A.C. Erase Characteristics and Waveforms for specific timing parameters.

Set-up Program/Program Commands

Set-up program is a command-only operation that stages the device for byte programming. Writing 40H into the command register performs the set-up operation.

Once the program set-up operation is performed, the next Write-Enable pulse causes a transition to an active programming operation. Addresses are internally latched on the falling edge of the Write-Enable pulse. Data is internally latched on the rising edge of the Write-Enable pulse. The rising edge of Write-Enable also begins the programming operation. The programming operation terminates with the next rising edge of Write-Enable, used to write the program-verify command. Refer to A.C. Program-

ming Characteristics and Waveforms for specific timing parameters.

Program-Verify Command

The 28F010 is programmed on a byte-by-byte basis. Byte programming may occur sequentially or at random. Following each programming operation, the byte just programmed must be verified.

The program-verify operation is initiated by writing C0H into the command register. The register write terminates the programming operation with the rising edge of its Write-Enable pulse. The program-verify operation stages the device for verification of the byte last programmed. No new address information is latched.

The 28F010 applies an internally-generated margin voltage to the byte. A microprocessor read cycle outputs the data. A successful comparison between the programmed byte and true data means that the byte is successfully programmed. Programming then proceeds to the next desired byte location. Figure 4, the 28F010 Quick-Pulse Programming™ algorithm, illustrates how commands are combined with bus operations to perform byte programming. Refer to A.C. Programming Characteristics and Waveforms for specific timing parameters.

Reset Command

A reset command is provided as a means to safely abort the erase- or program-command sequences. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

EXTENDED ERASE/PROGRAM CYCLING

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled—an expensive solution.

Intel has designed extended cycling capability into its ETOX-II flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field is one-tenth that of common EEPROMs, minimizing the probability of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 MV/cm lower than EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure—increasing time to wearout by a factor of 100,000,000.

The device is programmed and erased using Intel's Quick-Pulse Programming™ and Quick-Erase algorithms. Intel's algorithmic approach uses a series of operations (pulses), along with byte verification, to completely and reliably erase and program the device.

QUICK-PULSE PROGRAMMING ALGORITHM

The Quick-Pulse Programming algorithm uses programming operations of 10 μs duration. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The algorithm allows for up to 25 programming operations per byte, although most bytes verify on the first or second operation. The entire sequence of programming and byte verification is performed with V_{PP} at high voltage. Figure 4 illustrates the Quick-Pulse Programming algorithm.

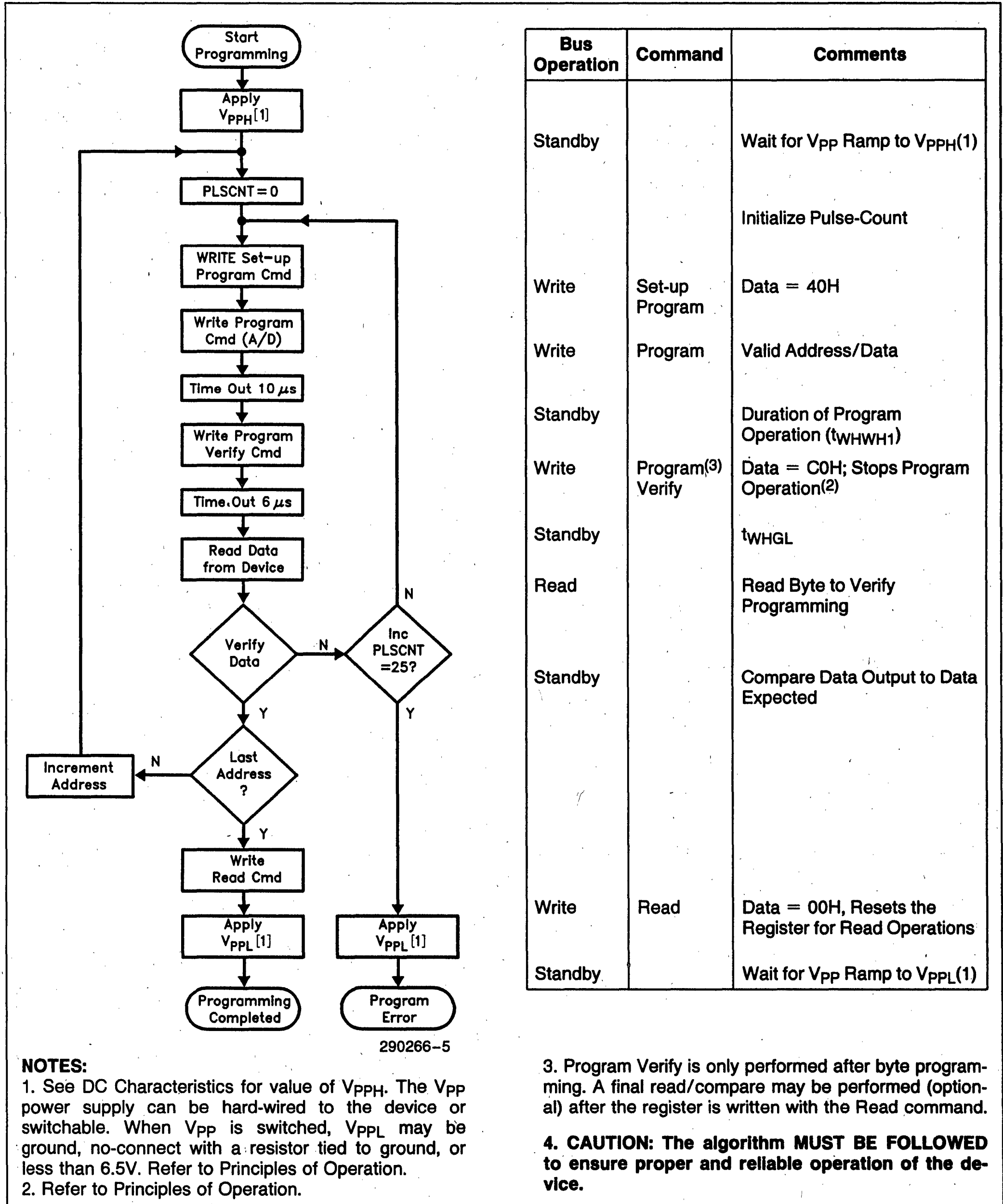
QUICK-ERASE ALGORITHM

Intel's Quick-Erase algorithm yields fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the Quick-Pulse Programming™ algorithm, to simultaneously remove charge from all bits in the array.

Erasure begins with a read of memory contents. The 28F010 is erased when shipped from the factory. Reading FFH data from the device would immediately be followed by device programming.

For devices being erased and reprogrammed, uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (Data = 00H). This is accomplished, using the Quick-Pulse Programming algorithm, in approximately two seconds.

Erase execution then continues with an initial erase operation. Erase verification (data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. With each erase operation, an increasing number of bytes verify to the erased state. Erase efficiency may be improved by storing the address of the last byte verified in a register. Following the next erase operation, verification starts at that stored address location. Erasure typically occurs in one second. Figure 5 illustrates the Quick-Erase algorithm.

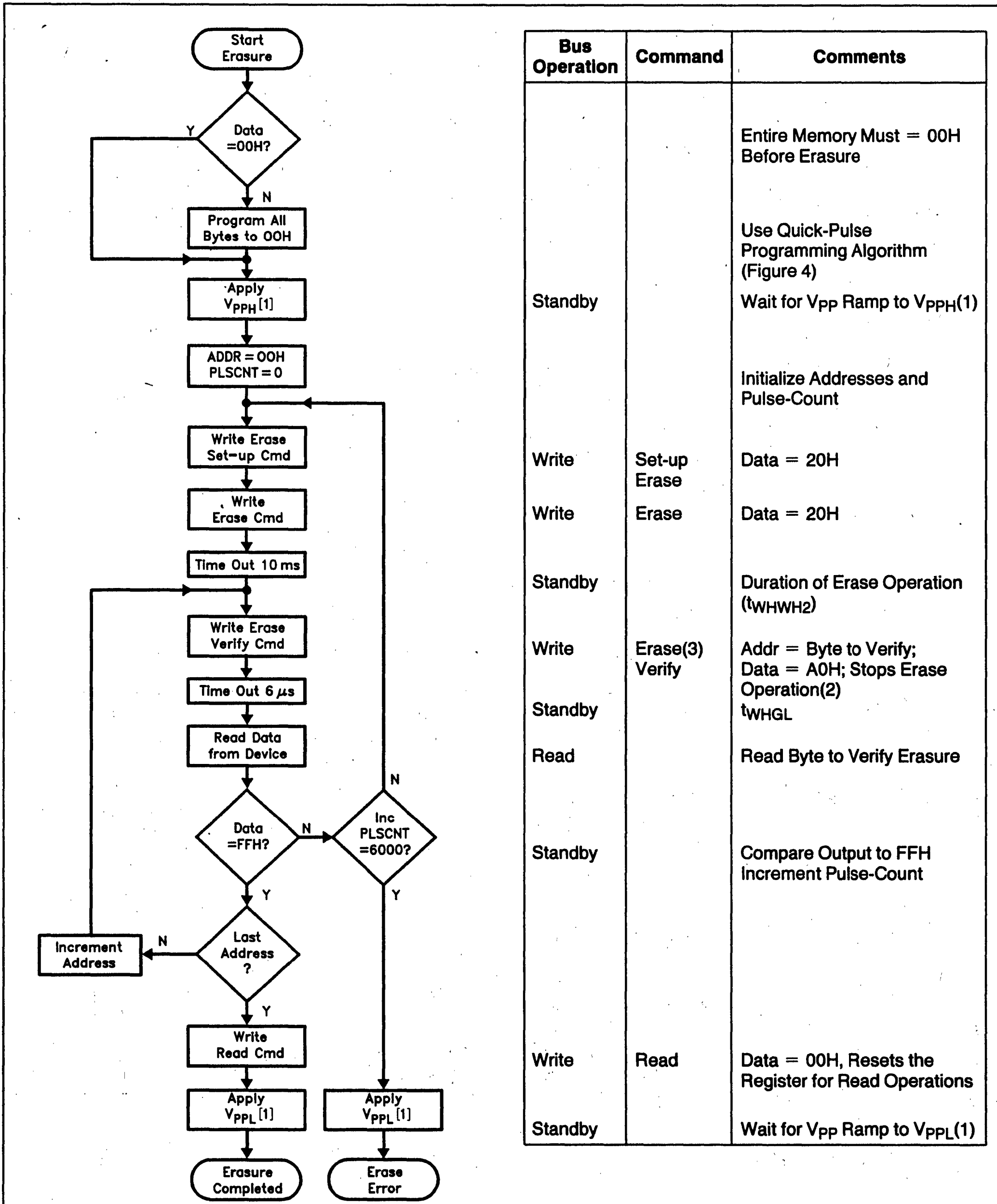


Bus Operation	Command	Comments
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Pulse-Count
Write	Set-up Program	Data = 40H
Write	Program	Valid Address/Data
Standby		Duration of Program Operation (t_{WHWH1})
Write	Program(3) Verify	Data = C0H; Stops Program Operation(2)
Standby		t_{WHGL}
Read		Read Byte to Verify Programming
Standby		Compare Data Output to Data Expected
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

NOTES:

1. See DC Characteristics for value of V_{PPH}. The V_{PP} power supply can be hard-wired to the device or switchable. When V_{PP} is switched, V_{PPL} may be ground, no-connect with a resistor tied to ground, or less than 6.5V. Refer to Principles of Operation.
2. Refer to Principles of Operation.
3. Program Verify is only performed after byte programming. A final read/compare may be performed (optional) after the register is written with the Read command.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

Figure 4. 28F010 Quick-Pulse Programming Algorithm



Bus Operation	Command	Comments
		Entire Memory Must = 00H Before Erasure
		Use Quick-Pulse Programming Algorithm (Figure 4)
Standby		Wait for V_{PP} Ramp to V_{PPH}(1)
		Initialize Addresses and Pulse-Count
Write	Set-up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase(3) Verify	Addr = Byte to Verify; Data = A0H; Stops Erase Operation(2)
Standby		t_{WHGL}
Read		Read Byte to Verify Erasure
Standby		Compare Output to FFH Increment Pulse-Count
Write	Read	Data = 00H, Resets the Register for Read Operations
Standby		Wait for V_{PP} Ramp to V_{PPL}(1)

NOTES:

1. See DC Characteristics for value of V_{PPH}. The V_{PP} power supply can be hard-wired to the device or switchable. When V_{PP} is switched, V_{PPL} may be ground, no-connect with a resistor tied to ground, or less than 6.5V. Refer to Principles of Operation.
2. Refer to Principles of Operation.
3. Erase Verify is performed only after chip-erasure. A final read/compare may be performed (optional) after the register is written with the read command.
4. **CAUTION: The algorithm MUST BE FOLLOWED to ensure proper and reliable operation of the device.**

Figure 5. A28F010 Quick-Erase Algorithm

DESIGN CONSIDERATIONS

Two-Line Output Control

Flash-memories are often used in larger memory arrays. Intel provides two read-control inputs to accommodate multiple memory connections. Two-line control provides for:

a. the lowest possible memory power dissipation and,

b. complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address-decoder output should drive chip-enable, while the system's read signal controls all flash-memories and other parallel memories. This assures that only enabled memory devices have active outputs, while deselected devices maintain the low power standby condition.

Power Supply Decoupling

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current (I_{CC}) issues—standby, active, and transient current peaks produced by falling and rising edges of chip-enable. The capacitive and inductive loads on the device outputs determine the magnitudes of these peaks.

Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 μF ceramic capacitor connected between V_{CC} and V_{SS}, and between V_{PP} and V_{SS}.

Place the high-frequency, low-inherent-inductance capacitors as close as possible to the devices. Also, for every eight devices, a 4.7 μF electrolytic capacitor should be placed at the array's power supply connection, between V_{CC} and V_{SS}. The bulk capacitor will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

V_{PP} Trace on Printed Circuit Boards

Programming flash-memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the V_{PP} power supply trace. The V_{PP} pin supplies the memory cell current for programming. Use similar trace widths and layout considerations given the V_{CC} power bus. Adequate V_{PP} supply traces and decoupling will decrease V_{PP} voltage spikes and overshoots.

Power Up/Down Protection

The 28F010 is designed to offer protection against accidental erasure or programming, caused by spurious system-level signals that may exist during power transitions. Also, with its control register architecture, alteration of memory contents only occurs after successful completion of the two-step command sequences. Power supply sequencing is not required. Internal circuitry of the 28F010 ensures that the command register architecture is reset to the read mode on power up.

A system designer must guard against active writes for V_{CC} voltages above the V_{LKO} when V_{PP} is active. Since both $\overline{WE}$ and $\overline{CE}$ must be low for a command write, driving either to V_{IH} will prohibit writes. The control register architecture provides an added level of protection since alteration of memory contents only occurs after successful completion of the two-step command sequences.

28F010 Power Dissipation

When designing portable systems, designers must consider battery power consumption not only during device operation, but also for data retention during system idle time. Flash nonvolatility increases the usable battery life of your system because the 28F010 does not consume any power to retain code or data when the system is off. Table 4 illustrates the power dissipated when updating the 28F010.

Table 4. 28F010 Typical Update Power Dissipation(4)

Operation	Power Dissipation (Watt-Seconds)	Notes
Array Program/ Program Verify	0.171	1
Array Erase/ Erase Verify	0.136	2
One Complete Cycle	0.478	3

NOTES:

1. Formula to calculate typical Program/Program Verify Power = [V_{PP} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{PP2} typical + t_{WHGL} × I_{PP4} typical)] + [V_{CC} × # Bytes × typical # Prog Pulses (t_{WHWH1} × I_{CC2} typical + t_{WHGL} × I_{CC4} typical].
2. Formula to calculate typical Erase/Erase Verify Power = [V_{PP} (V_{PP3} typical × t_{ERASE} typical + I_{PP5} typical × t_{WHGL} × # Bytes)] + [V_{CC} (I_{CC3} typical × t_{ERASE} typical × I_{CC5} typical × t_{WHGL} × # Bytes)].
3. One Complete Cycle = Array Preprogram + Array Erase + Program.
4. "Typicals" are not guaranteed, but based on a limited number of samples from production lots.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read . . . . −40°C to +125°C(1)
During Erase/Program . . . . −40°C to +125°C

Temperature Under Bias . . . . −40°C to +125°C

Storage Temperature . . . . −65°C to +150°C

Voltage on Any Pin with
Respect to Ground . . . . −2.0V to +7.0V(2)

Voltage on Pin A_9 with
Respect to Ground . . . . −2.0V to +13.5V(2, 3)

V_{PP} Supply Voltage with
Respect to Ground
During Erase/Program . . . . −2.0V to +14.0V(2, 3)

V_{CC} Supply Voltage with
Respect to Ground . . . . −2.0V to +7.0V(2)

Output Short Circuit Current . . . . 100 mA(4)

Maximum Junction Temperature (T_J) . . . . . . +140°C

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for automotive product defined by this specification.
2. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
3. Maximum DC voltage on A_9 or V_{PP} may overshoot to +14.0V for periods less than 20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.

OPERATING CONDITIONS

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T_A	Operating Temperature	−40	+125	°C	For Read-Only and Read/Write Operations
V_{CC}	V_{CC} Supply Voltage	4.50	5.50	V	

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1			1.0	mA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{IH}$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, I_{OUT} = 0 mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	30	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{CC4}	V_{CC} Program Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{CC5}	V_{CC} Erase Verify Current	1, 2		5.0	30	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10	μA	$V_{PP} \leq V_{CC}$

DC CHARACTERISTICS—TTL/NMOS COMPATIBLE (Continued)

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage				0.45	V	$I_{OL} = 2.1$ mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage		2.4			V	$I_{OH} = -2.5$ mA $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifer Voltage		11.50		13.00	V	$A_9 = V_{ID}$
I_{ID}	V_{CC} ID Current	1		10	30	mA	$A_9 = V_{ID}$
	V_{PP} ID CURRENT			90	500	μA	
V_{PPL}	V_{PP} during Read-Only Operations		0.00		6.5	V	**NOTE:** Erase/Program are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations		11.40		12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.5			V	

DC CHARACTERISTICS—CMOS COMPATIBLE

Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{LI}	Input Leakage Current	1			±1.0	μA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or V_{SS}
I_{LO}	Output Leakage Current	1			±10	μA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or V_{SS}
I_{CCS}	V_{CC} Standby Current	1		50	100	μA	$V_{CC} = V_{CC}$ Max $\overline{CE} = V_{CC} \pm 0.2V$
I_{CC1}	V_{CC} Active Read Current	1		10	30	mA	$V_{CC} = V_{CC}$ Max, $\overline{CE} = V_{IL}$ f = 6 MHz, $I_{OUT} = 0$ mA
I_{CC2}	V_{CC} Programming Current	1, 2		1.0	30	mA	Programming in Progress
I_{CC3}	V_{CC} Erase Current	1, 2		5.0	30	mA	Erasure in Progress
I_{PPS}	V_{PP} Leakage Current	1			±10	μA	$V_{PP} \leq V_{CC}$
I_{PP1}	V_{PP} Read Current or Standby Current	1		90	200	μA	$V_{PP} > V_{CC}$
					±10		$V_{PP} \leq V_{CC}$

DC CHARACTERISTICS—CMOS COMPATIBLE (Continued)

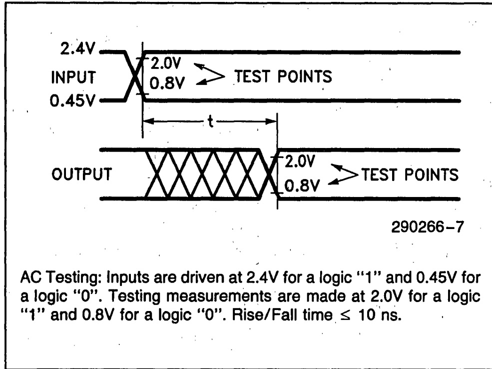
Symbol	Parameter	Notes	Limits			Unit	Test Conditions
			Min	Typical	Max		
I_{PP2}	V_{PP} Programming Current	1, 2		8.0	30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP3}	V_{PP} Erase Current	1, 2		4.0	30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PP4}	V_{PP} Program Verify Current	1, 2		2.0	5.0	mA	$V_{PP} = V_{PPH}$ Program Verify in Progress
I_{PP5}	V_{PP} Erase Verify Current	1, 2		5.0	5.0	mA	$V_{PP} = V_{PPH}$ Erase Verify in Progress
V_{IL}	Input Low Voltage	−0.5			0.8	V	
V_{IH}	Input High Voltage	0.7 V_{CC}			V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL} = 2.1 mA $V_{CC} = V_{CC}$ Min
V_{OH1}	Output High Voltage	0.85 V_{CC}				V	I_{OH} = −2.5 mA, $V_{CC} = V_{CC}$ Min
V_{OH2}		V_{CC} − 0.4					I_{OH} = −100 μA, $V_{CC} = V_{CC}$ Min
V_{ID}	A_9 Intelligent Identifier Voltage		11.50		13.00	V	
I_{ID}	V_{CC} ID Current	1		10	30	mA	A_9 = ID
I_{ID}	V_{PP} ID Current	1		90	500	μA	A_9 = ID
V_{PPL}	V_{PP} during Read-Only Operations	0.00			6.5	V	**NOTE:** Erase/Programs are Inhibited when $V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during Read/Write Operations	11.40			12.60	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage	2.5				V	

CAPACITANCE[3] T_A = 25°C, f = 1.0 MHz

Symbol	Parameter	Limits		Unit	Conditions
		Min	Max		
C_{IN}	Address/Control Capacitance		8	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance		12	pF	V_{OUT} = 0V

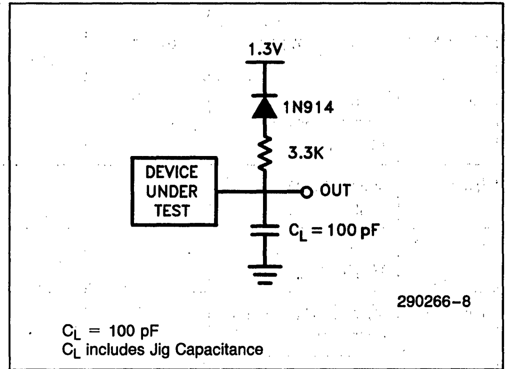
NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T= 25°C.
2. Not 100% tested: characterization data available.
3. Sampled, not 100% tested.
4. "Typicals" are not guaranteed, but are based on a limited number of samples from production lots.

AC TESTING INPUT/OUTPUT WAVEFORM



AC Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Testing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0". Rise/Fall time ≤ 10 ns.

AC TESTING LOAD CIRCUIT



C_L = 100 pF
C_L includes Jig Capacitance

AC TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ...... 10 ns
Input Pulse Levels .... 0.45V and 2.4V
Input Timing Reference Level .... 0.8V and 2.0V
Output Timing Reference Level ...... 0.8V and 2.0V

AC CHARACTERISTICS—Read-Only Operations(2)

Versions		Notes	28F010-150		Unit
Symbol	Characteristic		Min	Max	
t_{AVAV}/t_{RC}	Read Cycle Time	3	150		ns
t_{ELQV}/t_{CE}	Chip Enable Access Time			150	ns
t_{AVQV}/t_{ACC}	Address Access Time			150	ns
t_{GLQV}/t_{OE}	Output Enable Access Time			55	ns
t_{ELQX}/t_{LZ}	Chip Enable to Output in Low Z	3	0		ns
t_{EHQZ}	Chip Disable to Output in High Z	3		55	ns
t_{GLQX}/t_{OLZ}	Output Enable to Output in Low Z	3	0		ns
t_{GHQZ}/t_{DF}	Output Disable to Output in High Z	4		35	ns
t_{OH}	Output Hold from Address, $\overline{CE}$, or $\overline{OE}$ Change	1, 3	0		ns
t_{WHGL}	Write Recovery Time before Read		6		µs

NOTES:
1. Whichever occurs first.
2. Rise/Fall Time ≤ 10 ns.
3. Not 100% tested: characterization data available.
4. Guaranteed by design.

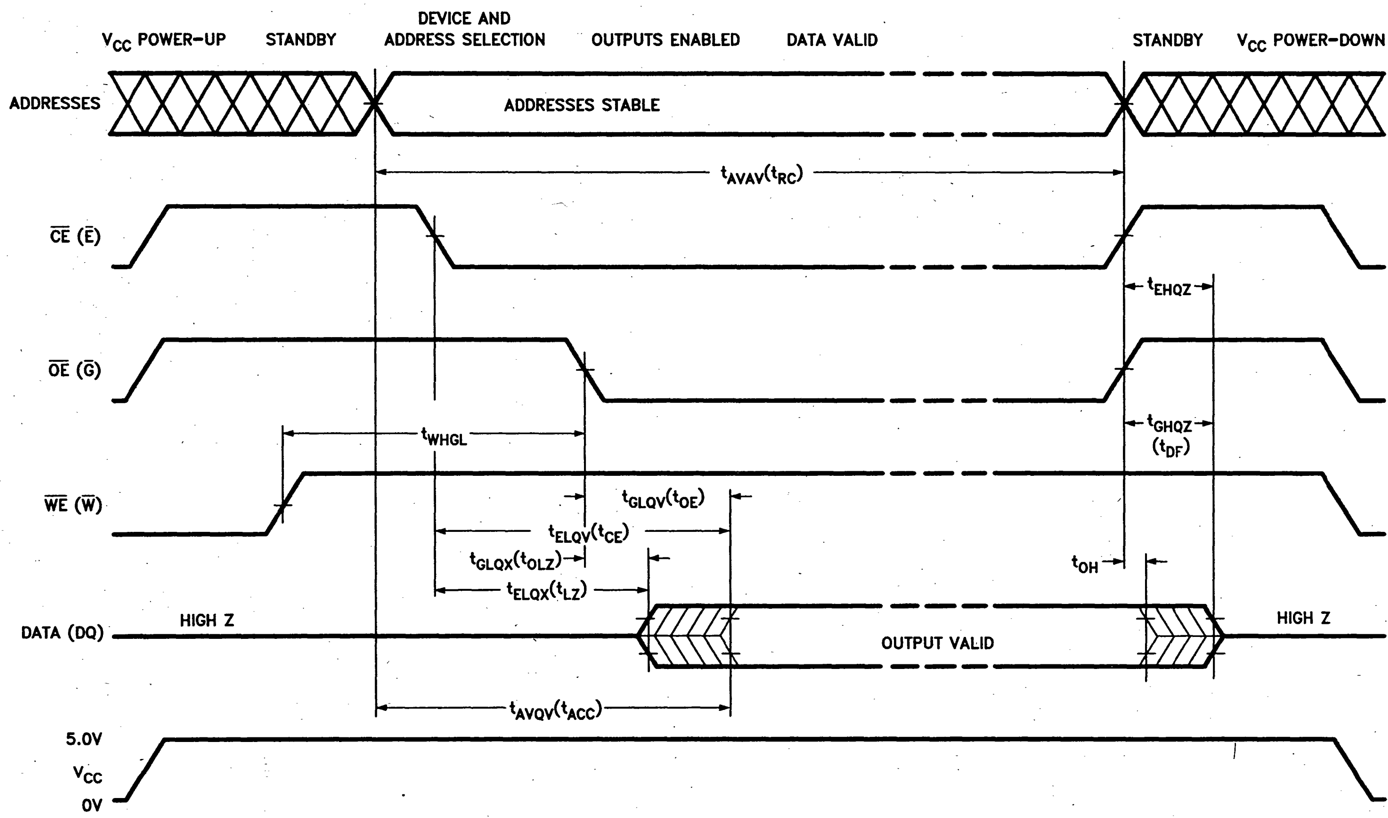


Figure 6. AC Waveforms for Read Operations

AC CHARACTERISTICS—Write/Erase/Program Operations(1, 3)

Versions		Notes	28F010-150		Unit
Symbol	Characteristic		Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		150		ns
t_{AVWL}/t_{AS}	Address Set-Up Time		0		ns
t_{WLAX}/t_{AH}	Address Hold Time	2	60		ns
t_{DVWH}/t_{DS}	Data Set-up Time		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		ns
t_{WHGL}	Write Recovery Time before Read		6		μs
t_{GHWL}	Read Recovery Time before Write		0		μs
t_{ELWL}/t_{CS}	Chip Enable Set-Up Time before Write	2	20		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width(2)	2	80		ns
t_{ELEH}	Alternative Write(2) Pulse Width	2	80		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		ns
t_{WHWH1}	Duration of Programming Operation	4	10		μs
t_{WHWH2}	Duration of Erase Operation	4	9.5		ms
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low		1.0		ms

NOTES:
1. Read timing characteristics during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only Operations.
2. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of Chip-Enable and Write-Enable. In systems where Chip-Enable defines the write pulse width (within a longer Write-Enable timing waveform) all set-up, hold, and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.
3. Rise/Fall time ≤ 10 ns.
4. The internal stop timer terminates the programming/erase operations, thereby eliminating the need for a maximum specification.

ERASE AND PROGRAMMING PERFORMANCE

Parameter	Notes	Limits			Unit	Comments
		Min	Typ	Max		
Chip Erase Time	1, 3, 4, 6		1	60	Sec	Excludes 00H Programming Prior to Erasure
Chip Program Time	1, 2, 4		2	12.5	Sec	Excludes System-Level Overhead
Erase/Program Cycles	1, 5	1,000	100,000		Cycles	

NOTES:
1. "Typicals" are not guaranteed, but based on a limited number of samples taken from production lots. Data taken at $T = 25°C$, $V_{PP} = 12.0V$, $V_{CC} = 5.0V$.
2. Minimum byte programming time excluding system overhead is 16 μsec (10 μsec program + 6 μsec write recovery), while maximum is 400 μsec/byte (16 μsec x 25 loops allowed by algorithm). Max chip programming time is specified lower than the worst case allowed by the programming algorithm since most bytes program significantly faster than the worst case byte.
3. Excludes 00H programming prior to erasure.
4. Excludes system-level overhead.
5. Refer to RR-60 "ETOX Flash Memory Reliability Data Summary" for typical cycling data and failure rate calculations.
6. Maximum erase specification is determined by algorithmic limit and accounts for cumulative effect of erasure at $T = -40°C$, 1,000 cycles, $V_{PP} = 11.4V$, $V_{CC} = 4.5V$.

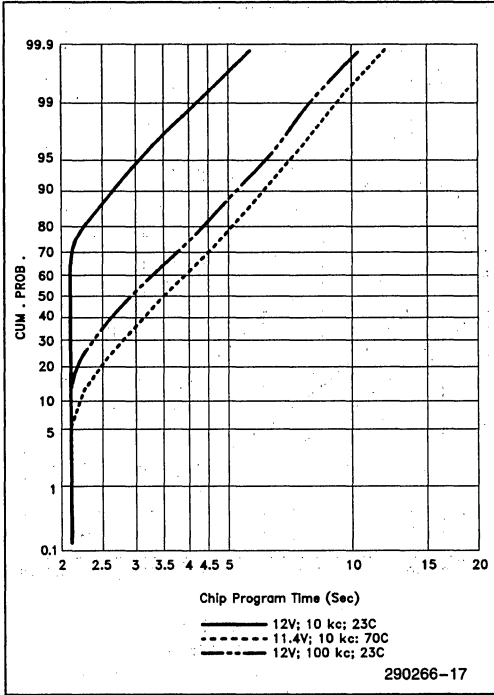


Figure 7. 28F010 Typical Programming Capability
See Note 1, Page 5-66.

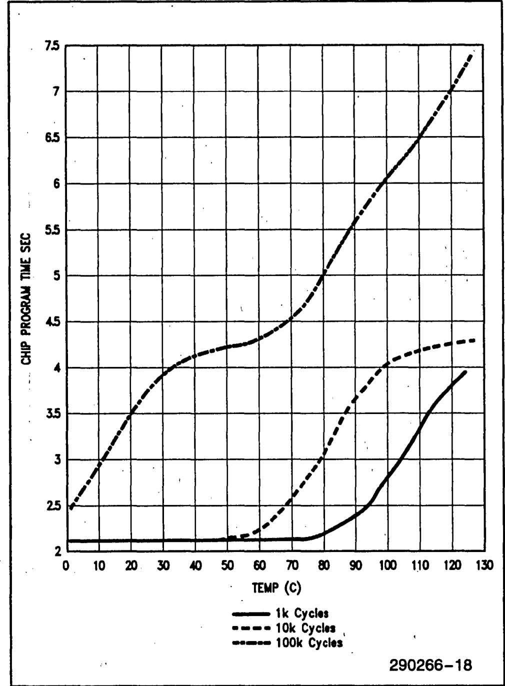


Figure 8. 28F010 Typical Program Time at 12V

5

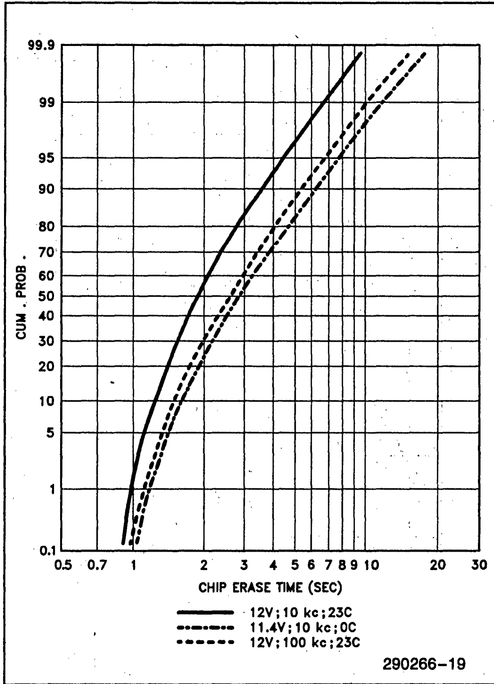


Figure 9. 28F010 Typical Erase Capability
See Note 1, Page 5-66.

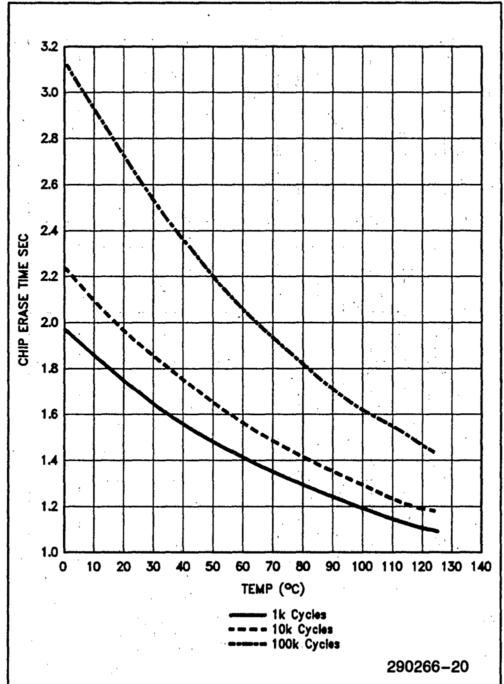


Figure 10. 28F010 Typical Erase Time at 12V

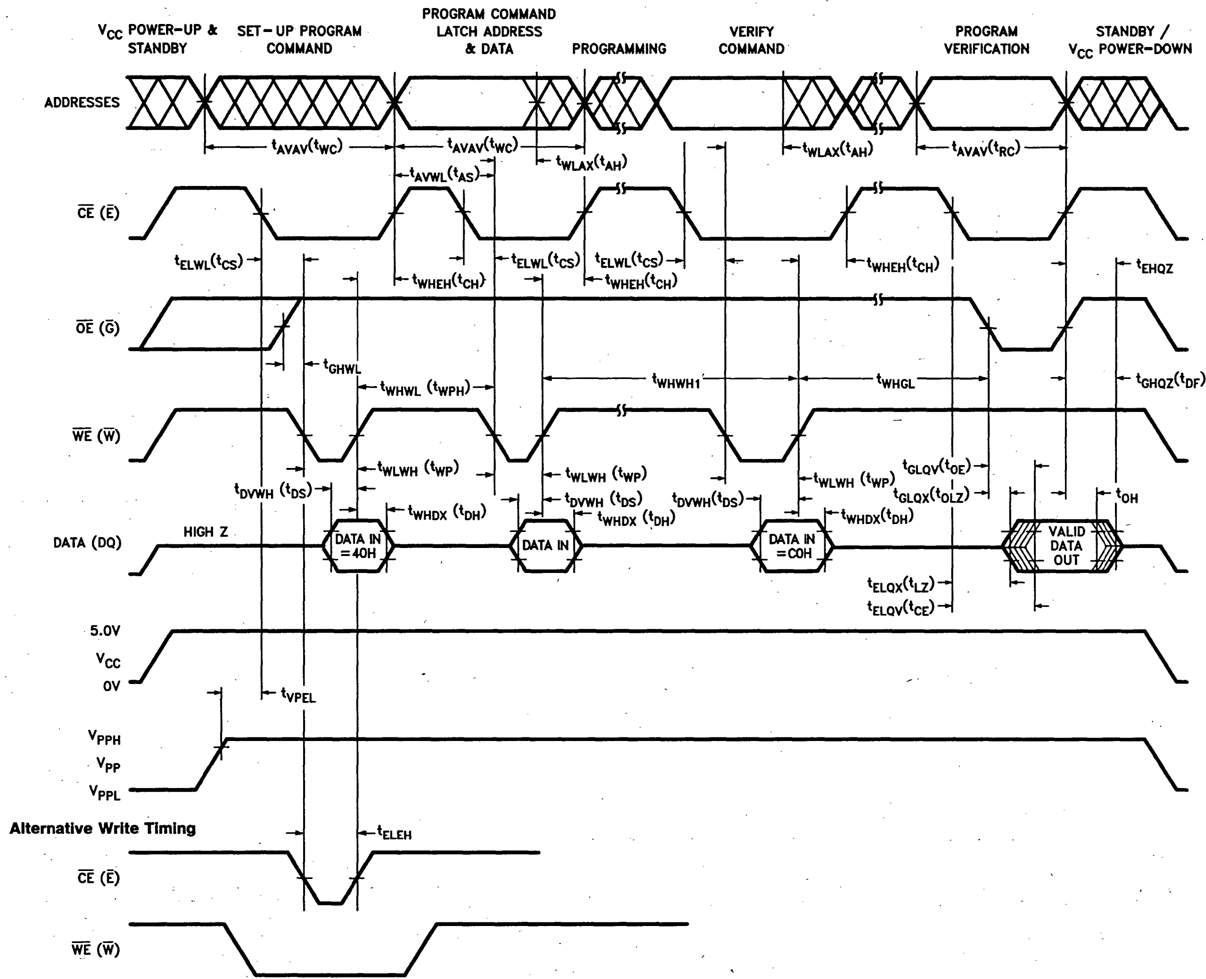


Figure 11. AC Waveforms for Programming Operations

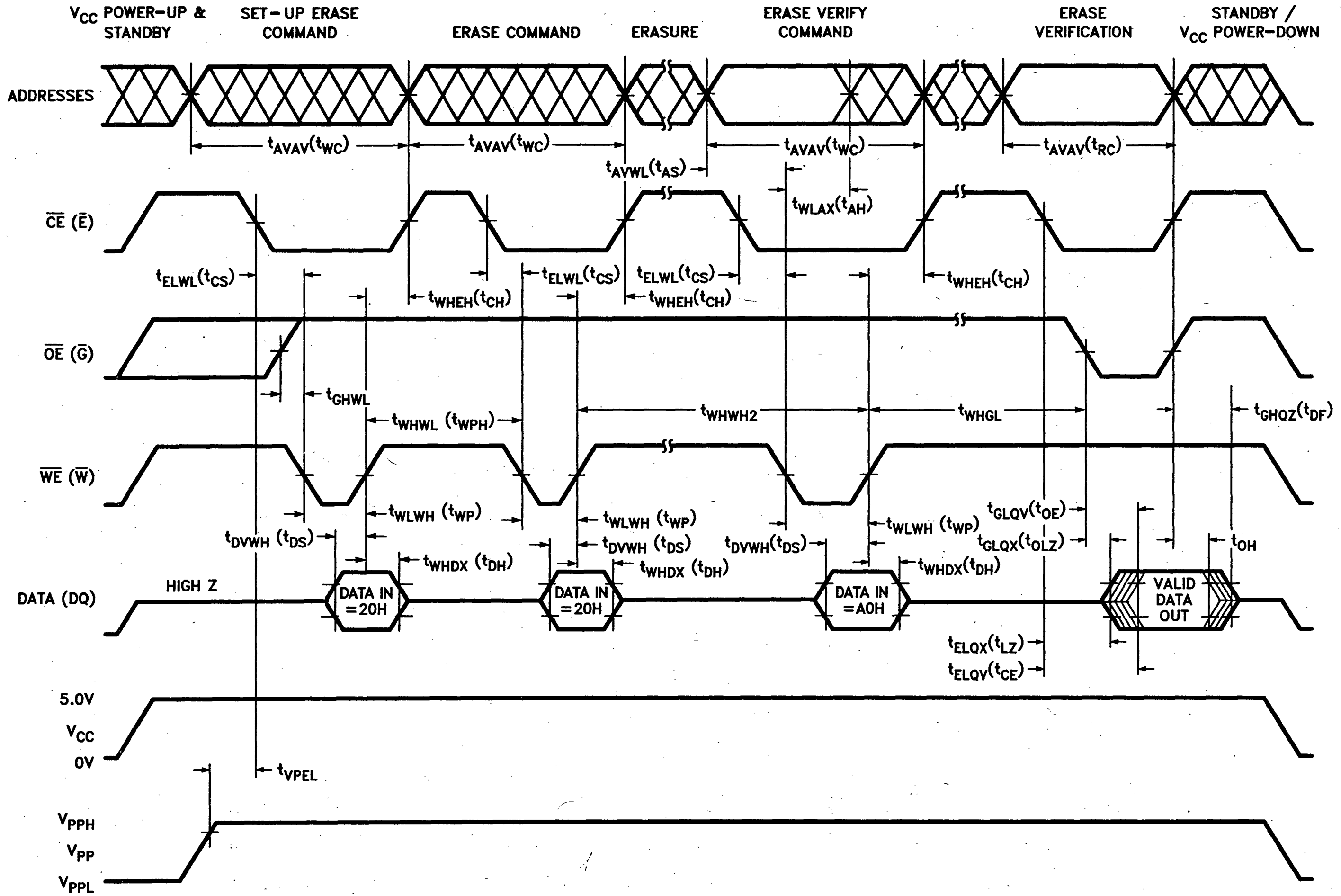
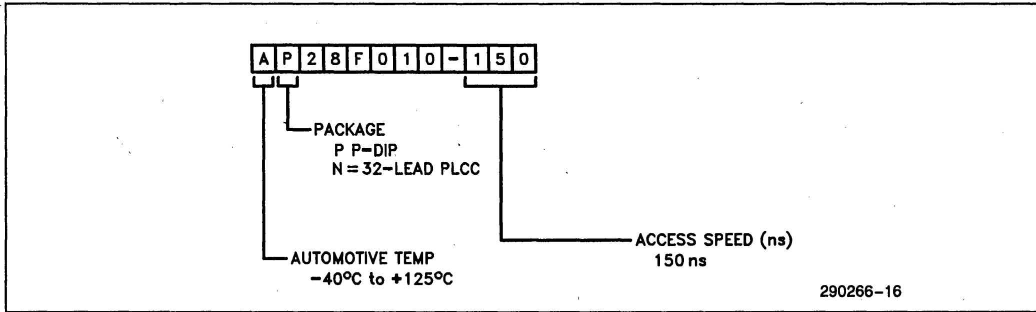


Figure 12. AC Waveforms for Erase Operations

Ordering Information



Valid Combinations:

AP28F010-150 AN28F010-150

ADDITIONAL INFORMATION

	Order Number
ER-20, "ETOX™II Flash Memory Technology"	294005
ER-24, "The Intel 28F010 Flash Memory"	294008
RR-60, "ETOX™ Flash Memory Reliability Data Summary"	293002
AP-316, "Using Flash Memory for In-System Reprogrammable Nonvolatile Storage"	292046
AP-325, "Guide to Flash Memory Reprogramming"	292059

REVISION HISTORY

Number	Description
003	Changed Erase/Program Cycles to 1,000 minimum

A27C256
256K (32K x 8) CHMOS EPROM

Automotive

- **Extended Automotive Temperature Range: −40°C to +125°C**
- **CHMOS/NMOS Microcontroller and Microprocessor Compatible**
  - **Universal 28 Pin Memory Site, 2-line Control**
- **120 ns Maximum Access Time**
- **CMOS and TTL Compatible**
- **Low Power**
  - **30 mA Max. Active**
  - **100 μA Max. Standby**
- **Fast Programming**
  - **Quick-Pulse Programming Algorithm**
  - **Programming Time as Fast as 4 Seconds**
- **Noise Immunity Features**
  - **±10% V_{CC} Tolerance**
  - **Maximum Latch-up Immunity through EPI Processing**
- **Available in 28-Pin Cerdip Package**
  - **28-Pin Plastic Dip Package**
  - **Compact 32-Lead PLCC**

Intel's A27C256 is a 5V only, 262,144-bit Erasable Programmable Read Only Memory, organized as 32,768 words of 8 bits. Its standard pinouts provide for simple upgrades to 512 Kbits in the future in both DIP and SMT.

The A27C256 is ideal in embedded control applications based on advanced 16-bit CPUs. Fast 120 ns access times allow no-wait-state operation with the 12 MHz 80286. The A27C256 also excels in reprogrammable environments where the system designer must strike an optimal density/performance balance. For example, bootstrap and diagnostic routines run 1-wait-state on a 16 MHz 386™ microprocessor.

Intel offers two DIP profile options to meet your prototyping and production needs. The windowed ceramic dip (CERDIP) package provides erasability and reprogrammability for prototyping and early production. Once the design is in full production, the plastic dip (PDIP) one-time programmable part provides a lower cost alternative that is well adapted for auto insertion.

In addition to the JEDEC 28-pin DIP package, Intel also offers a 32-lead PLCC version of the A27C256. This one-time-programmable surface mount device is ideal where board space consumption is a major concern or where surface mount manufacturing technology is being implemented across an entire production line.

The A27C256 is equally at home in both TTL and CMOS environments. The Quick-Pulse programming algorithm improves speed as much as 100 times over older methods, further reducing cost for system manufacturers.

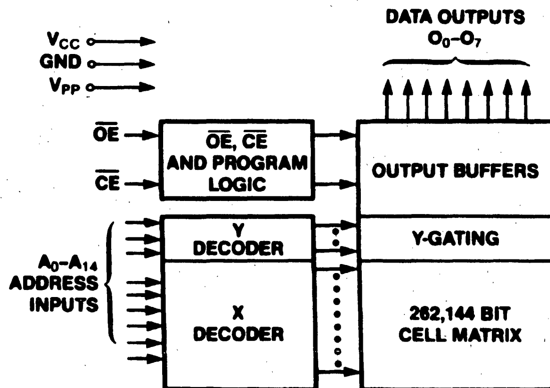


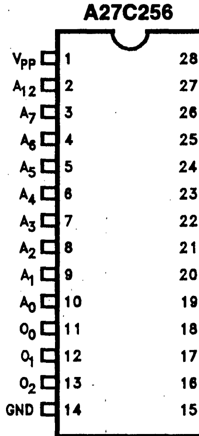
Figure 1. Block Diagram

September 1992
Order Number: 290120-005

Pin Names

A_0–A_{15}	ADDRESSES
O_0–O_7	OUTPUTS
$\overline{OE}$	OUTPUT ENABLE
$\overline{CE}$	CHIP ENABLE
$\overline{PGM}$	PROGRAM
NC	NO CONNECT
DU	DON'T USE

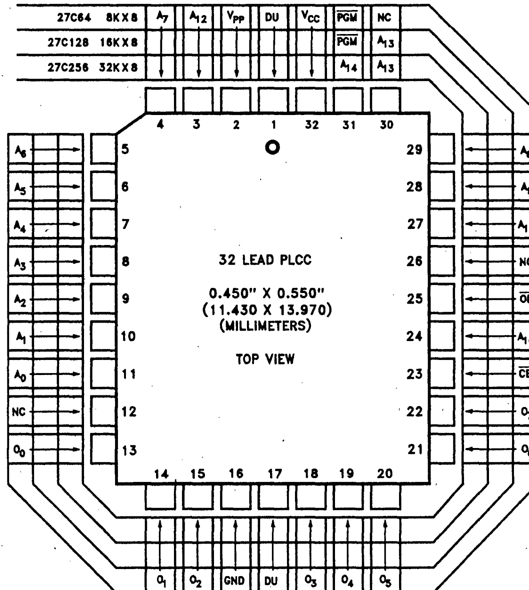
27512 27C512	27128A 27C128	2764A 27C64	2732A	2716
A_{15}	V_{PP}	V_{PP}		
A_{12}	A_{12}	A_{12}		
A_7	A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND



2716	2732A	27C64 27C64	27128A 27C128	27512 27C512
		V_{CC}	V_{CC}	V_{CC}
		$\overline{PGM}$	$\overline{PGM}$	A_{14}
V_{CC}	V_{CC}	NC	A_{13}	A_{13}
A_8	A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

290120–2

Figure 2. DIP Pin Configuration



290120–3

Figure 3. PLCC Lead Configuration

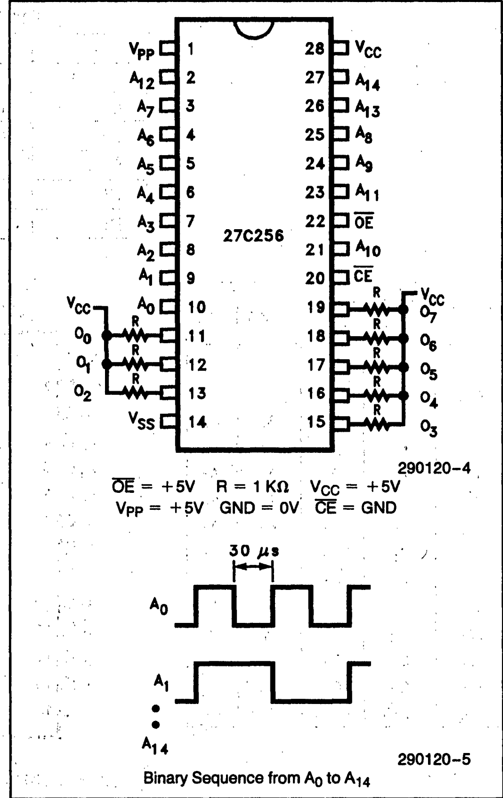
5

AUTOMOTIVE TEMPERATURE EPROMs

The Intel AUTOMOTIVE EPROM family receives additional processing to enhance product characteristics. AUTOMOTIVE processing is available for several densities allowing the appropriate memory size to match system requirements. AUTOMOTIVE EPROMs are available with 168 ±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This processing meets or exceeds most industry burn-in specifications. The AUTOMOTIVE product family is available in −40°C to 125°C operating temperature range versions. Like all Intel EPROMs, the AUTOMOTIVE EPROM family is inspected to 0.1% electrical AQL. This allows reduction or elimination of incoming testing.

Options

Speed	Packaging		
	CerDIP	PLCC	PDIP
−120V10	AD	AN	AP
−200V10	AD	AN	AP



Burn-In Bias and Timing Diagrams

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature . . . . −40°C to 125°C
Temperature Under Bias . . . . −40°C to 125°C
Storage Temperature . . . . −65°C to 150°C
Voltage on Any Pin (except A_9, V_{CC} and V_{PP}) with Respect to GND . . . . −2V to 7V(2)
Voltage on A_9 with Respect to GND . . . . −2V to 13.5V(2)
V_{PP} Supply Voltage with Respect to GND . . . . −2V to 14.0V(2)
V_{CC} Supply Voltage with Respect to GND . . . . −2V to 7.0V(2)
Maximum Junction Temperature (T_J) . . . . 140°C

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

READ OPERATION DC CHARACTERISTICS(1) V_{CC} = 5.0V ±10%

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Conditions
I_{LI}	Input Load Current	7		0.01	1.0	μA	V_{IN} = 0V to V_{CC}
I_{LO}	Output Leakage Current				±10	μA	V_{OUT} = 0V to V_{CC}
I_{SB}	V_{CC} Standby Current				1.0	mA	$\overline{CE}$ = V_{IH}
					100	μA	$\overline{CE}$ = V_{CC} ±0.2V
I_{CC}	V_{CC} Operating Current	3			30	mA	$\overline{CE}$ = V_{IL} f = 5 MHz
I_{PP}	V_{PP} Operating Current	3			200	μA	V_{PP} = V_{CC}
I_{OS}	Output Short Circuit Current	4, 6			100	mA	
V_{IL}	Input Low Voltage		−0.5		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.45	V	I_{OL}=2.1 mA
V_{OH}	Output High Voltage		2.4			V	I_{OH}= −400 μA
V_{PP}	V_{PP} Operating Voltage	5	V_{CC} − 0.7		V_{CC}	V	

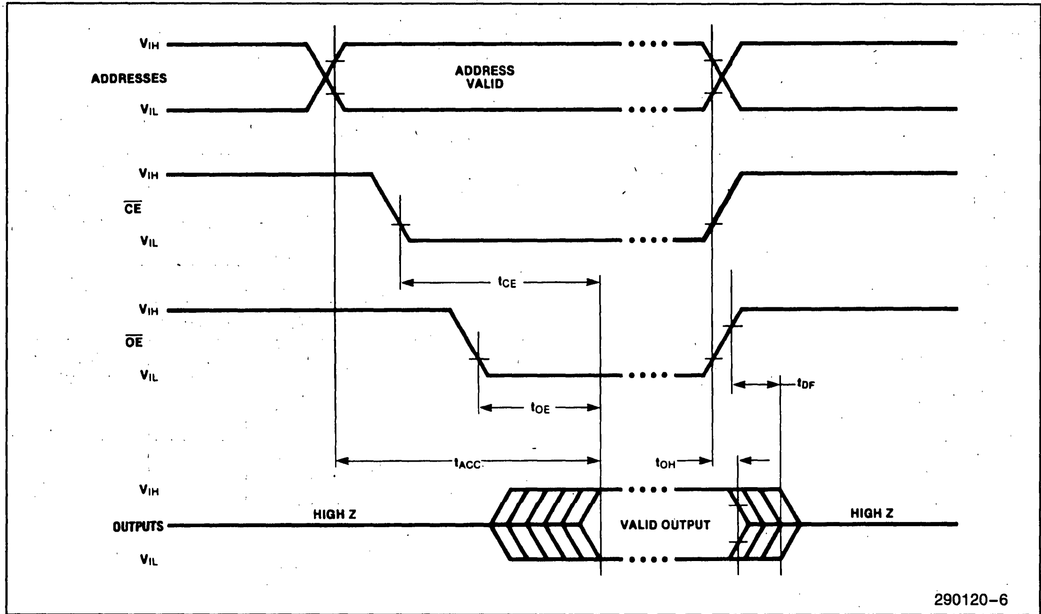
NOTES:
1. Minimum DC voltage is −0.5V on input/output pins. During transitions, this level may undershoot to −2.0V for periods <20 ns. Maximum DC voltage on input/output pins is V_{CC} + 0.5V which, during transitions, may overshoot to V_{CC} + 2.0V for periods <20 ns.
2. Maximum active power usage is the sum I_{PP} + I_{CC}. Maximum current value is with outputs O_0 to O_7 unloaded.
3. Output shorted for no more than one second. No more than one output shorted at a time.
4. V_{PP} may be connected directly to V_{CC}, or may be one diode voltage drop below V_{CC}. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
5. Sampled, not 100% tested.
6. Typical limits are at V_{CC} = 5V, T_A = 25°C.

READ OPERATION AC CHARACTERISTICS(1) V_{CC} = 5.0V ± 10%

Versions		V_{CC} ± 10%	A27C256-120V10		A27C256-200V10		Unit
Symbol	Parameter	Notes	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay			120		200	ns
t_{CE}	$\overline{CE}$ to Output Delay	2		120		200	ns
t_{OE}	$\overline{OE}$ to Output Delay	2		55		75	ns
t_{DF}	$\overline{OE}$ High to Output High Z	3		30		55	ns
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change-Whichever is First	3	0		0		ns

NOTES:
1. See AC Input/Output Reference Waveform for timing measurements.
2. $\overline{OE}$ may be delayed up to $t_{CE}-t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.
3. Sampled, not 100% tested.

AC WAVEFORMS



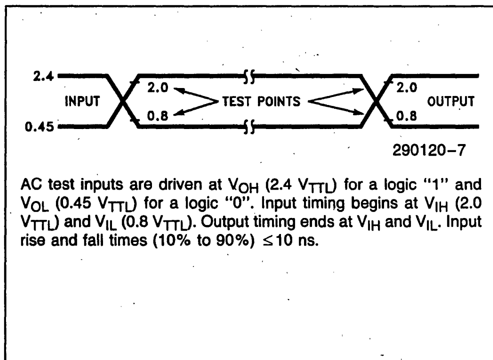
290120-6

CAPACITANCE(1) $T_A = 25°C$, $f = 1.0$ MHz

Symbol	Parameter	Max	Units	Conditions
C_{IN}	Address/Control Capacitance	6	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	12	pF	$V_{OUT} = 0V$

NOTE:
1. Sampled, not 100% tested.

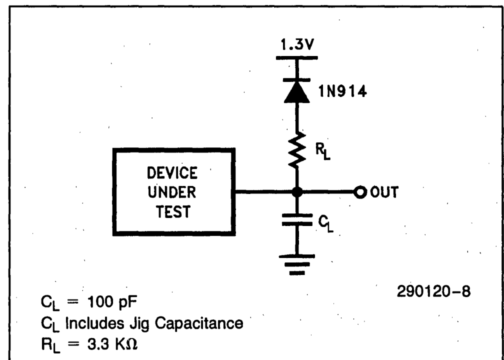
AC INPUT/OUTPUT REFERENCE WAVEFORM



290120-7

AC test inputs are driven at V_{OH} (2.4 V_{TTL}) for a logic "1" and V_{OL} (0.45 V_{TTL}) for a logic "0". Input timing begins at V_{IH} (2.0 V_{TTL}) and V_{IL} (0.8 V_{TTL}). Output timing ends at V_{IH} and V_{IL}. Input rise and fall times (10% to 90%) ≤10 ns.

AC TESTING LOAD CIRCUIT



290120-8

C_L = 100 pF
C_L Includes Jig Capacitance
R_L = 3.3 KΩ

5

DEVICE OPERATION

The Mode Selection table lists A27C256 operating modes. Read Mode requires a single 5V power supply. All inputs, except V_{CC} and V_{PP}, and A_9 during Intelligent Identifier Mode, are TTL or CMOS.

Table 1. Mode Selection

Mode	Notes	$\overline{CE}$	$\overline{OE}$	A_9	A_0	V_{PP}	V_{CC}	Outputs
Read	1	V_{IL}	V_{IL}	X	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable		V_{IL}	V_{IH}	X	X	V_{CC}	V_{CC}	High Z
Standby		V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	2	V_{IL}	V_{IH}	X	X	V_{PP}	V_{CP}	D_{IN}
Program Verify		V_{IH}	V_{IL}	X	X	V_{PP}	V_{CP}	D_{OUT}
Program Inhibit		V_{IH}	V_{IH}	X	X	V_{PP}	V_{CP}	HIGH Z
Intelligent Identifier -Manufacturer	2, 3	V_{IL}	V_{IL}	V_{ID}	V_{IL}	V_{CC}	V_{CC}	89 H
Intelligent Identifier -Device	2, 3, 4	V_{IL}	V_{IL}	V_{ID}	V_{IH}	V_{CC}	V_{CC}	8D H

NOTES:
1. X can be V_{IL} or V_{IH}.
2. See DC Programming Characteristics for V_{CP}, V_{PP} and V_{ID} voltages.
3. A_1–A_8, A_{10-14} = V_{IL}.
4. Programming equipment may also refer to this device as the A27C256A. Older devices may have device ID = 8CH.

Read Mode

The A27C256 has two control functions, both must be enabed to obtain data at the outputs. $\overline{CE}$ is the power control and device select. $\overline{OE}$ controls the output buffers to gate data to the outputs. With addresses stable, the address access time (t_{ACC}) equals the delay from $\overline{CE}$ to output (t_{CE}). Outputs display valid data t_{OE} after $\overline{OE}$'s falling edge, assuming t_{ACC} and t_{CE} times are met.

V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.

Two Line Output Control

EPROMs are often used in larger memory arrays. Intel provides two control inputs to accommodate multiple memory connections. Two-line control provides for:

a) lowest possible memory power dissipation

b) complete assurance that data bus contention will not occur

To efficiently use these two control inputs, an address decoder should enable $\overline{CE}$, while $\overline{OE}$ should be connected to all memory devices and the system's $\overline{READ}$ control line. This assures that only selected memory devices have active outputs while deselected memory devices are in Standby Mode.

Standby Mode

Standby Mode substantially reduces V_{CC} current. When $\overline{CE}$ = V_{IH}, the outputs are in a high impedance state, independent of $\overline{OE}$.

Program Mode

Initially, and after each erasure, all EPROM bits are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" are programmed, the data word can contain both "1's" and "0's". Ultraviolet light erasure is the only way to change "0's" to "1's".

Program Mode is entered when V_{PP} is raised to 12.75V. Data is introduced by applying an 8-bit word to the output pins. Pulsing $\overline{CE}$ low while $\overline{OE} = V_{IH}$ programs that data into the device.

Program Verify

A verify should be performed following a program operation to determine that bits have been correctly programmed. With V_{CC} at 6.25V a substantial program margin is ensured. The verify is performed with $\overline{CE}$ at V_{IH}. Valid data is available t_{OE} after $\overline{OE}$ falls low.

Program Inhibit

Program Inhibit Mode allows parallel programming of multiple EPROMs with different data. $\overline{CE}$-high inhibits programming of non-targeted devices. Except for $\overline{CE}$ and $\overline{OE}$, parallel EPROMs may have common inputs.

Intelligent Identifier Mode

The Intelligent Identifier Mode will determine an EPROM's manufacturer and device type, allowing programming equipment to automatically match a device with its proper programming algorithm.

This mode is activated when a programmer forces 12V + 0.5V on A_9. With $\overline{CE}$, $\overline{OE}$, A_1-A_8, and $A_{10}-A_{14}$ at V_{IL}, $A_0 = V_{IL}$ will present the manufacturer code and $A_0 = V_{IH}$ the device code. This mode functions in the 25°C ± 5°C ambient temperature range required during programming.

UPGRADE PATH

Future upgrade to the 512 Kbit density is easily accomplished due to the standardized pin configuration of the A27C256. A jumper between A_{15} and V_{CC} allows upgrade using the V_{PP} pin. Systems designed for 256 Kbit program memories today can be upgraded to 512 Kbit in the future with no circuit board changes.

SYSTEM CONSIDERATIONS

EPROM power switching characteristics require careful device decoupling. System designers are interested in 3 supply current issues: standby current levels (I_{SB}), active current levels (I_{CC}), and transient current peaks produced by falling and rising edges of $\overline{CE}$. Transient current magnitudes depend on the device output's capacitive and inductive loading. Two-Line Control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 μF ceramic capacitor connected between its V_{CC} and GND. This high frequency, low inherent-inductance capacitor should be placed as close as possible to the device. Additionally, for every 8 devices, a 4.7 μF electrolytic capacitor should be placed at the array's power supply connection between V_{CC} and GND. The bulk capacitor will overcome voltage slumps caused by PC board trace inductances.

ERASURE CHARACTERISTICS

Erasure begins when EPROMs are exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain flourescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room level fluorescent lighting can erase an EPROM in approximately 3 years, while it takes approximately 1 week when exposed to direct sunlight. If the device is exposed to these lighting conditions for extended periods, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure is exposure to ultraviolet light of wavelength 2537Å. The integrated dose (UV intensity x exposure time) for erasure should be a minimum of 15 Wsec/cm^2. Erasure time is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The EPROM should be placed within 1 inch of the lamp tubes. An EPROM can be permanently damaged if the integrated dose exceeds 7258 Wsec/cm^2 (1 week @ 12000 μW/cm^2).

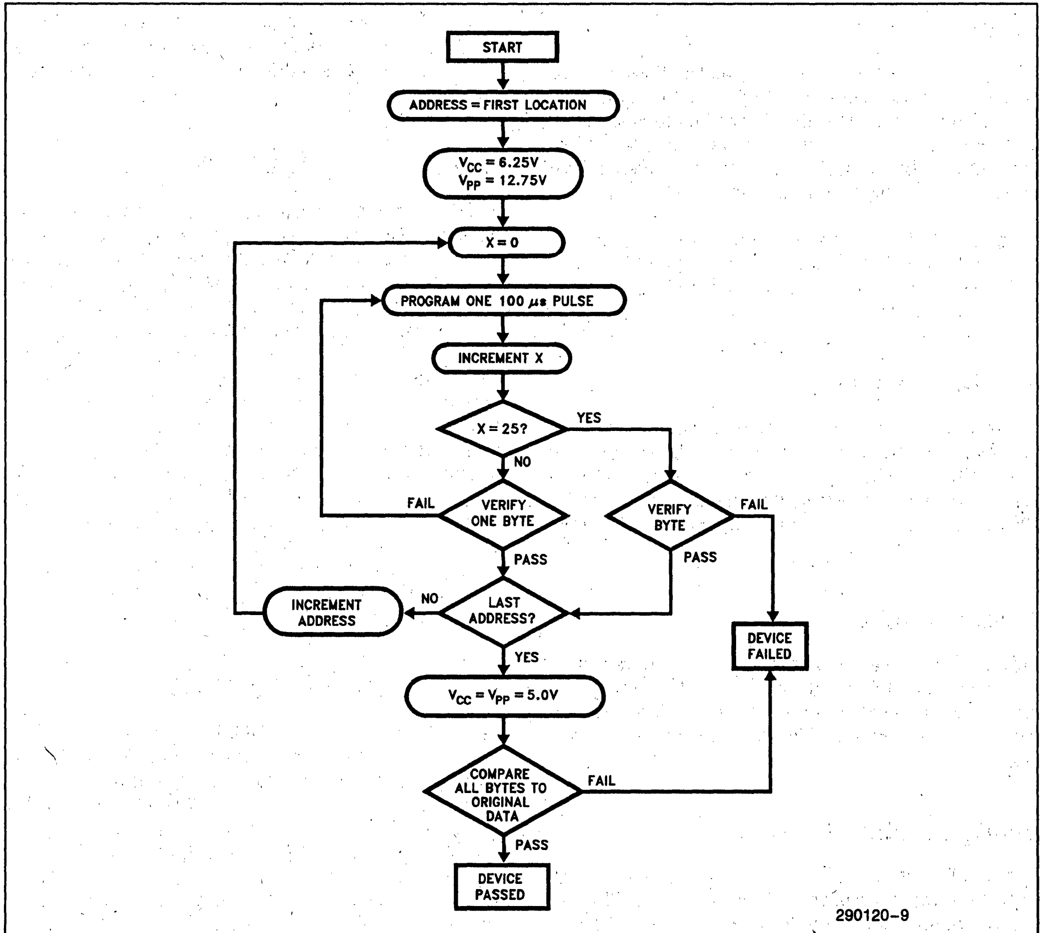


Figure 4. Quick-Pulse Programming Algorithm

Quick-Pulse Programming Algorithm

The Quick-Pulse Programming algorithm programs Intel's A27C256. Developed to substantially reduce programming throughput, this algorithm can program the A27C256 as fast as 4 seconds. Actual programming time depends on programmer overhead.

The Quick-Pulse programming algorithm employs a 100 μs pulse followed by a byte verification to determine when the addressed byte has been sucessfully programmed. The algorithm terminates if 25 attempts fail to program a byte.

The entire program pulse/byte verify sequence is performed with V_{PP} = 12.75V and V_{CC} = 6.25V. When programming is complete, all bytes are compared to the original data with V_{CC} = V_{PP} = 5.0V.

In addition to the Quick-Pulse Programming Algorithm, the A27C256 has also been characterized for the Quick-Board Programming Algorithm. The Quick-Board Programming Algorithm was developed for specific automotive applications using Intel's 1.0 micron EPROM products. Contact the factory or an automotive sales representative for any information regarding the Quick-Board Programming Algorithm.

DC PROGRAMMING CHARACTERISTICS $T_A = 25°C \pm 5°C$

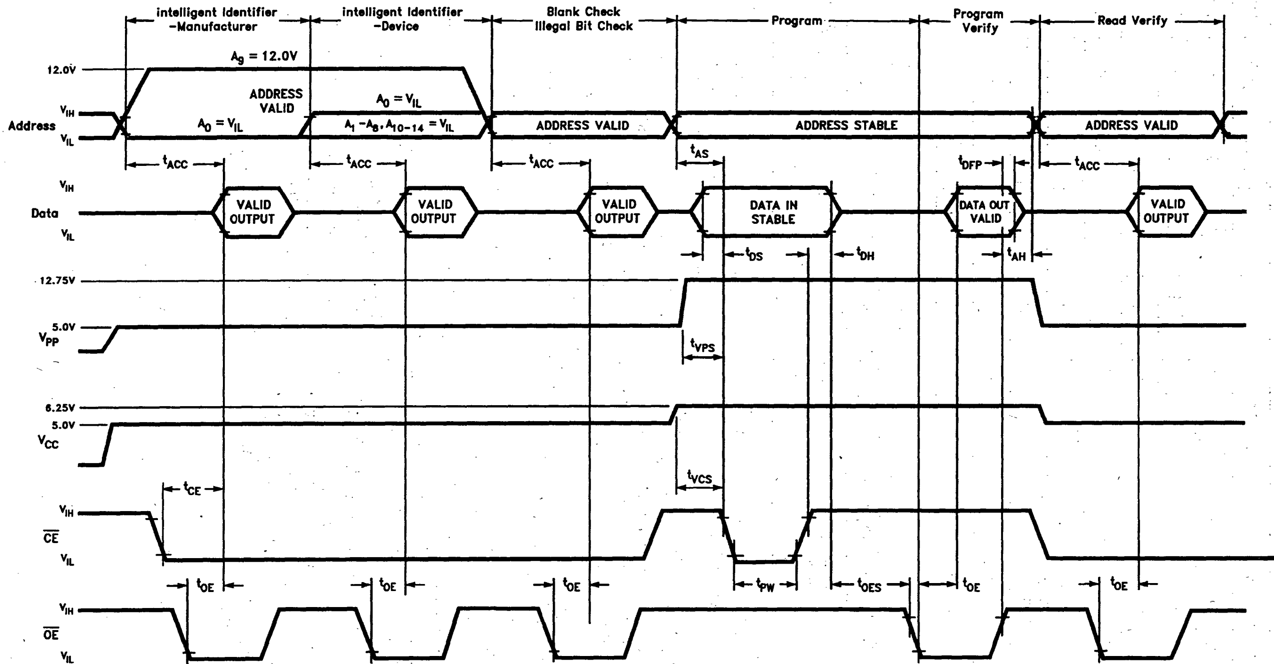
Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{LI}	Input Load Current				1.0	µA	$V_{IN} = V_{IL}$ or V_{IH}
I_{CP}	V_{CC} Program Current	1			30	mA	$\overline{CE} = V_{IL}$
I_{PP}	V_{PP} Program Current	1			50	mA	$\overline{CE} = V_{IL}$
V_{IL}	Input Low Voltage		−0.1		0.8	V	
V_{IH}	Input High Voltage		2.4		6.5	V	
V_{OL}	Output Low Voltage (Verify)				0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage (Verify)		3.5			V	$I_{OH} = -2.5$ mA
V_{ID}	A_9 Intelligent Identifier Voltage		11.5	12.0	12.5	V	
V_{PP}	V_{PP} Program Voltage	2, 3	12.5	12.75	13.0	V	
V_{CP}	V_{CC} Supply Voltage (Program)	2	6.0	6.25	6.5	V	

AC PROGRAMMING CHARACTERISTICS(4) $T_A = 25°C \pm 5°C$

Symbol	Parameter	Notes	Min	Typ	Max	Unit
t_{VCS}	V_{CP} Setup Time	2	2			µs
t_{VPS}	V_{PP} Setup Time	2	2			µs
t_{AS}	Address Setup Time		2			µs
t_{DS}	Data Setup Time		2			µs
t_{PW}	$\overline{CE}$ Program Pulse Width		95	100	105	µs
t_{DH}	Data Hold Time		2			µs
t_{OES}	$\overline{OE}$ Setup Time		2			µs
t_{OE}	Data Valid from $\overline{OE}$	5			150	ns
t_{DFP}	$\overline{OE}$ High to Output High Z	5, 6	0		130	ns
t_{AH}	Address Hold Time		0			µs

NOTES:
1. Maximum current value is with outputs O_0 to O_7 unloaded.
2. V_{CP} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
3. When programming, a 0.1 µF capacitor is required across V_{PP} and GND to suppress spurious voltage transients which can damage the device.
4. See AC Input/Output Reference Waveform for timing measurments.
5. t_{OE} and t_{DFP} are device characteristics but must be accommodated by the programmer.
6. Sampled, not 100% tested.

PROGRAMMING WAVEFORMS



A87C257
256K (32K x 8) CHMOS LATCHED EPROM

Automotive

- **Extended Automotive Temperature Range: −40°C to +125°C**
- **CHMOS/NMOS Microcontroller and Microprocessor Compatible**
  - **—87C257-Integrated Address Latch**
  - **—Universal 28 Pin Memory Site, 2-line Control**
- **120 ns Maximum Access Time**
- **CMOS and TTL Compatible**
- **Low Power**
  - **—30 mA Max. Active**
  - **—100 μA MAx. Standby**
- **Fast Programming**
  - **—Quick-Pulse Programming Algorithm**
  - **—Programming Time as Fast as 4 Seconds**
- **Noise Immunity Features**
  - **—±10% V_{CC} Tolerance**
  - **—Maximum Latch-up Immunity Through EPI Processing**
- **Available in 28-Pin Cerdip Package**
  - **—Compact 32 Lead PLCC**

(See Packaging Spec., Order #231369)

Intel's 87C257 CHMOS EPROM is a 256K-bit 5V only memory organized as 32,768 8-bit words. It employs advanced CHMOS*III-E circuitry for systems requiring low power, high speed performance, and noise immunity. The 87C257 is optimized for compatibility with multiplexed address/data bus microcontrollers such as Intel's 16 MHz 80C51, 80C152, 80C252, and 8 MHz 80C196.

The 87C257 incorporates latches on all address inputs to minimize chip count, reduce cost, and simplify design of multiplexed bus systems. The 87C257's internal address latch allows address and data pins to be tied directly to the processor's multiplexed address/data pins. Address information (inputs A_0–A_{14}) is latched early in the memory-fetch cycle by the falling edge of the ALE input. Subsequent address information is ignored while ALE remains low. The EPROM can then pass data (from pins O_0–O_7) on the same bus during the last part of the memory-fetch cycle.

The 87C257 is offered in a ceramic DIP and PLCC packages, providing flexibility in prototyping and R&D environments. The 87C257 employs the Quick-Pulse Programming™ Algorithm for fast and reliable programming.

Intel's EPI processing achieves the highest degree of latch-up protection. Address and data pin latch-up prevention is provided for stresses up to 100 mA from −1V to V_{CC} + 1V.

In order to meet the rigorous environmental requirements of automotive applications, Intel offers the 87C257 in extended Automotive temperature range. Operational characteristics are guaranteed over the range of −40°C to +125°C ambient.

*HMOS and CHMOS are patented processes of Intel Corporation.

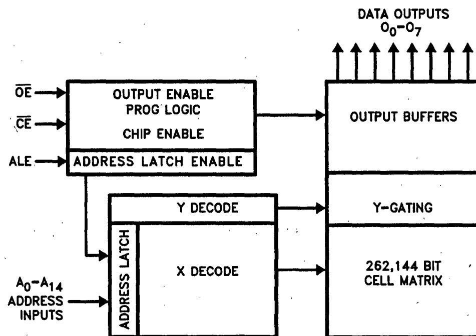


Figure 1. Block Diagram

5

August 1992
Order Number: 290142-004

Pin Names

A_0–A_{14}	ADDRESSES
O_0–O_7	OUTPUTS
$\overline{OE}$	OUTPUT ENABLE
$\overline{CE}$	CHIP ENABLE
ALE/V_{PP}	Address Latch Enable/V_{PP}
N.C.	NO CONNECT
D.U.	DON'T USE

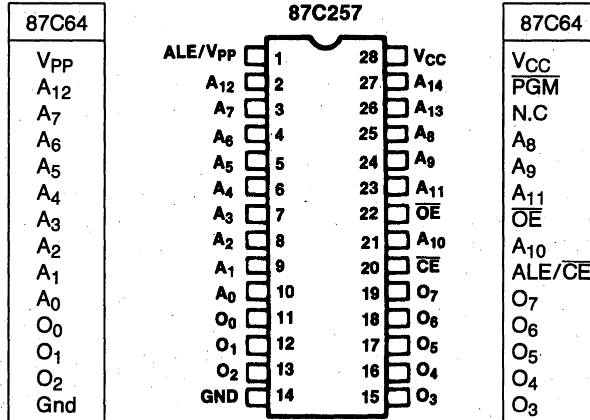


Figure 2. DIP Pin Configuration

NOTE:
Intel "Universal Site"-Compatible EPROM Pin Configurations are Shown in the Blocks Adjacent.

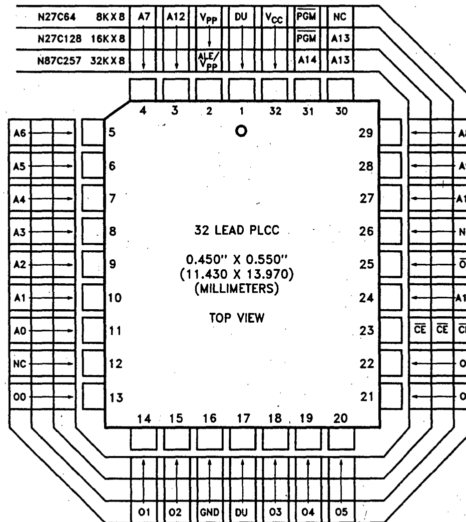


Figure 3. PLCC Lead Configuration

NOTE:
Intel "Universal Site"-Compatible EPROM Pin Configurations are Shown in the Blocks Adjacent.

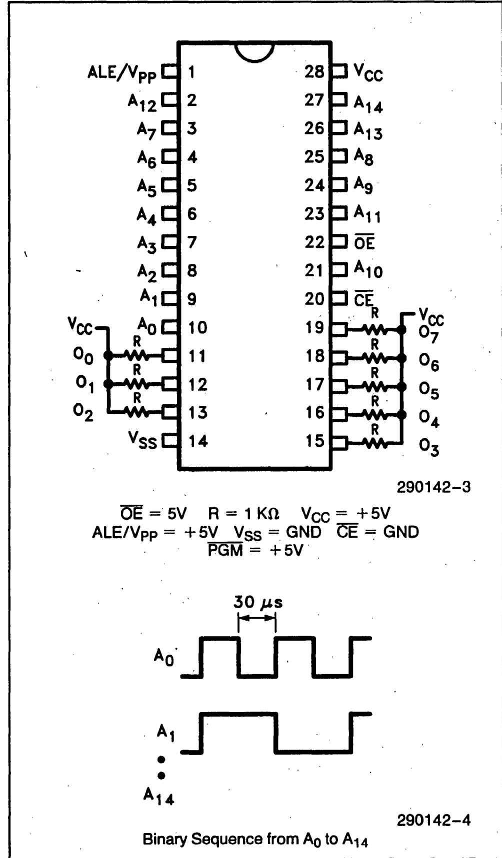
AUTOMOTIVE TEMPERATURE EPROMs

Intel automotive EPROMs have received additional processing to enhance product characteristics. The automotive temperature range is −40°C to +125°C during operating modes.

AUTOMOTIVE OPTIONS

Versions

Speed Versions	Packaging Options	
	Cerdip	PLCC
−120V10	AD	AN
−200V10	AD	AN



Burn-In Bias and Timing Diagrams

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature During
Read . . . . −40°C to +125°C
Temperature Under Bias . . . . −40°C to +125°C
Storage Temperature . . . . −65°C to +150°C
Voltage on any Pin with
Respect to Ground . . . . −2V to +7V(1)
Voltage on A_9 with
Respect to Ground . . . . −2V to +13.5V(1)
V_{PP} Supply Voltage with Respect to Ground
During Programming . . . . −2V to +14.0V(1)
V_{CC} Supply Voltage with
Respect to Ground . . . . −2V to +7.0V(1)

Maximum Junction Temperature (T_J) . . . . 140°C
Maximum Thermal Resistance
Junction to Ambient (θ_{JA}):
Cerdip . . . . 40°C/W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

WARNING: *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

READ OPERATION

DC CHARACTERISTICS TTL and NMOS Inputs −40°C ≤ T_A ≤ +125°C

Symbol	Parameter		Notes	Min	Typ(2)	Max	Units	Test Condition
I_{LI}	Input Load Current				0.01	±1.0	μA	V_{IN} = 0V, 5.5V
I_{LO}	Output Leakage Current				0.01	±10	μA	V_{OUT} = 0V, 5.5V
I_{SB}	V_{CC} Current Standby with Inputs—	Switching				10	mA	$\overline{CE}$ = ALE = V_{IH}
		Stable				1.0	mA	$\overline{CE}$ = V_{IH}, ALE = V_{IL}
I_{CC1}	V_{CC} Current Active		4			30	mA	$\overline{CE}$ = V_{IL}, ALE = V_{IH} f = 5 MHz, I_{OUT} = 0 mA
V_{IL}	Input Low Voltage (±10% Supply)		1	−0.5		0.8	V	
V_{IH}	Input High Voltage (±10% Supply)			2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage					0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage			2.4			V	I_{OH} = −400 μA
I_{OS}	Output Short Circuit Current		5			100	mA	

DC CHARACTERISTICS CMOS Inputs −40°C ≤ T_A ≤ +125°C

Symbol	Parameter		Notes	Min	Typ(2)	Max	Units	Test Condition
I_{LI}	Input Load Current				0.01	±1.0	μA	V_{IN} = 5.5V
I_{LO}	Output Leakage Current				0.01	±10.0	μA	V_{OUT} = 05, 5.5V
I_{SB}	V_{CC} Current Standby with Inputs—	Switching	3			6	mA	$\overline{CE}$ = ALE = V_{IH}
		Stable				100	μA	$\overline{CE}$ = V_{IH}, ALE = V_{IL}
I_{CC1}	V_{CC} Current Active		4			15	mA	$\overline{CE}$ = V_{IL}, ALE = V_{IH} f = 5 MHz, I_{OUT} = 0 mA
V_{IL}	Input Low Voltage (±10% Supply)			−0.2		0.8	V	
V_{IH}	Input High Voltage (±10% Supply)			0.7 V_{CC}		V_{CC} + 0.2	V	
V_{OL}	Output Low Voltage					0.4	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage			V_{CC} − 0.8			V	I_{OH} = −2.5 mA
I_{OS}	Output Short Circuit Current		5			100	mA	

NOTES:
1. Minimum DC input voltage is −0.5V. During transitions, the inputs may undershoot to −2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V which may overshoot to V_{CC} + 2V for periods less than 20 ns.
2. Typical limits are at V_{CC} = 5V, T_A = +25°C.
3. $\overline{CE}$ is V_{CC} ±0.2V. All other inputs can have any value within spec.
4. Maximum current value is with outputs O_0 to O_7 unloaded.
5. Output shorted for no more than one second. No more than one output shorted at a time. I_{OS} is sampled but not 100% tested.

READ OPERATION

AC CHARACTERISTICS(1) $-40°C \leq T_A \leq +125°C$

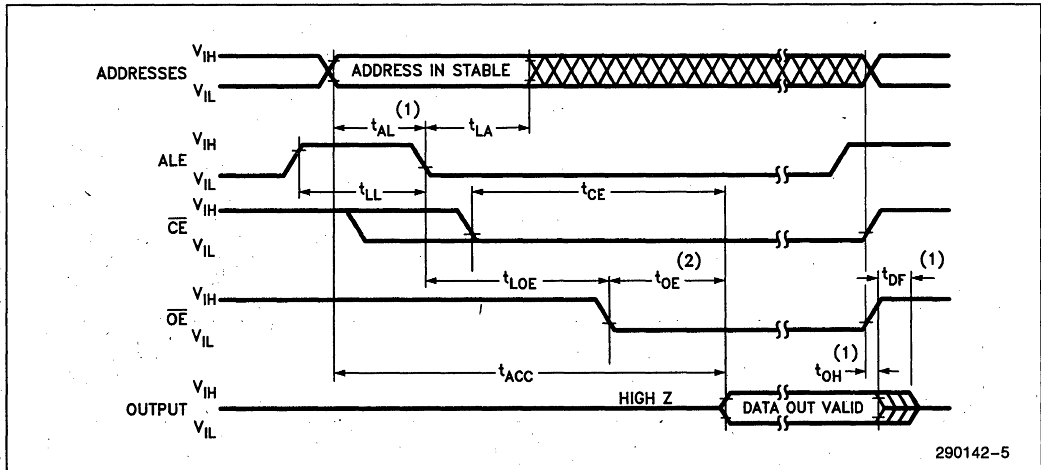
Versions	V_{CC} ±10%	87C257-120V10		87C257-200V10		Units
Symbol	Characteristic	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay		120		200	ns
t_{CE}	$\overline{CE}$ to Output Delay		120		200	ns
t_{OE}	$\overline{OE}$ to Output Delay		55		75	ns
t_{DF}(2)	$\overline{OE}$ High to Output High Z		30		40	ns
t_{OH}(2)	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change-Whichever is First	0		0		ns
t_{LL}	Latch Deselect Width	50		50		ns
t_{AL}(2)	Address to Latch Set-Up	15		15		ns
t_{LA}	Address Hold from LATCH	30		30		ns
t_{LOE}	ALE to Output Enable	30		30		ns

NOTES:
1. See AC Testing Input/Output Waveforms for timing measurements.
2. Guaranteed and sampled.

AC CONDITIONS OF TEST

Input Rise and Fall Times (10% to 90%) ...... 10 ns
Input Pulse Levels .... V_{OL} to V_{OH}
Input Timing Reference Level 1.5V
Output Timing Reference Level .... V_{IL} and V_{IH}

AC WAVEFORMS



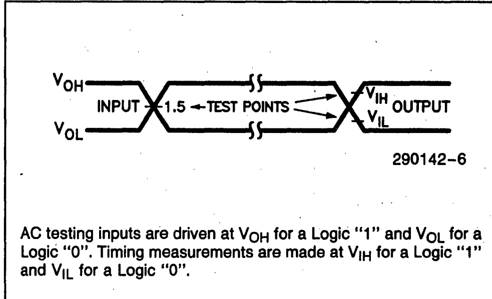
NOTES:
1. This parameter is only sampled and is not 100% tested.
2. $\overline{OE}$ may be delayed up to $t_{CE}-t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.

CAPACITANCE(1) $T_A = 25°C, f = 1.0 MHz$

Symbol	Parameter	Max	Units	Conditions
C_{IN}	Address/Control Capacitance	6	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	12	pF	$V_{OUT} = 0V$

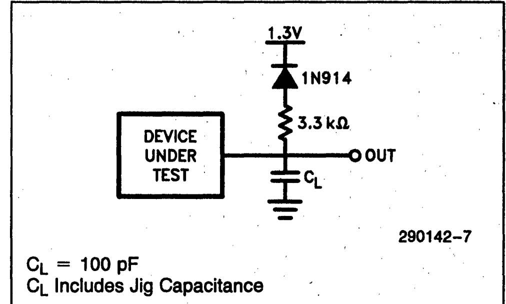
NOTE:
1. Sampled. Not 100% tested.

AC TESTING INPUT/OUTPUT WAVEFORM



AC testing inputs are driven at V_{OH} for a Logic "1" and V_{OL} for a Logic "0". Timing measurements are made at V_{IH} for a Logic "1" and V_{IL} for a Logic "0".

AC TESTING LOAD CIRCUIT



C_L = 100 pF
C_L Includes Jig Capacitance

DEVICE OPERATION

Table 1 lists 87C257 operating modes. Read mode requires a single 5V power supply. All input levels are TTL or CMOS except A_9 in Intelligent Identifier mode and V_{PP}.

Table 1. Mode Selection

Mode \ Pins	$\overline{CE}$	$\overline{OE}$	A_9	A_0	ALE/ V_{PP}	V_{CC}	Outputs
Read	V_{IL}	V_{IL}	X(1)	X	X	5.0V	D_{OUT}
Output Disable	V_{IL}	V_{IH}	X	X	X	5.0V	High Z
Standby	V_{IH}	X	X	X	X	5.0V	High Z
Programming	V_{IL}	V_{IH}	X	X	(Note 4)	(Note 4)	D_{IN}
Program Verify	V_{IH}	V_{IL}	X	X	(Note 4)	(Note 4)	D_{OUT}
Optional Program Verify	V_{IL}	V_{IL}	X	X	V_{CC} (Note 4)	(Note 4)	D_{OUT}
Program Inhibit	V_{IH}	V_{IH}	X	X	(Note 4)	(Note 4)	High Z
Intelligent Identifier(3) -Manufacturer	V_{IL}	V_{IL}	V_H(2)	V_{IL}	X	V_{CC}	89 H
Intelligent Identifier(3) -87C257	V_{IL}	V_{IL}	V_H(2)	V_{IH}	X	V_{CC}	24 H

NOTES:
1. X can be V_{IL} or V_{IH}.
2. $V_H = 12.0V \pm 0.5V$.
3. A_1–A_8, $A_{10-12} = V_{IL}$, A_{13-14} = X.
4. See Table 2 for V_{CC} and V_{PP} programming voltages.

Read Mode

The 87C257 has two control functions; both must be logically active to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and the device-select. Output enable ($\overline{OE}$) gates data to the output pins by controlling the output buffer. When the address is stable (ALE = V_{IH}) or latched (ALE = V_{IL}), the address access time (t_{ACC}) equals the delay from $\overline{CE}$ to output (t_{CE}). Outputs display valid data t_{OE} after the falling edge of $\overline{OE}$, assuming t_{ACC} and t_{CE} times are met.

The 87C257 reduces the hardware interface in multiplexed address-data bus systems. Figure 4 shows a low power, small board space, minimal chip 87C257/microcontroller design. The processor's multiplexed bus (AD_{0-7}) is tied to the 87C257's address and data pins. No separate address latch is needed because the 87C257 latches all address inputs when ALE is low.

The ALE input controls the 87C257's internal address latch. As ALE transitions from V_{IH} to V_{IL}, the last address present at the address pins is retained. The $\overline{OE}$ control can then enable EPROM data onto the bus.

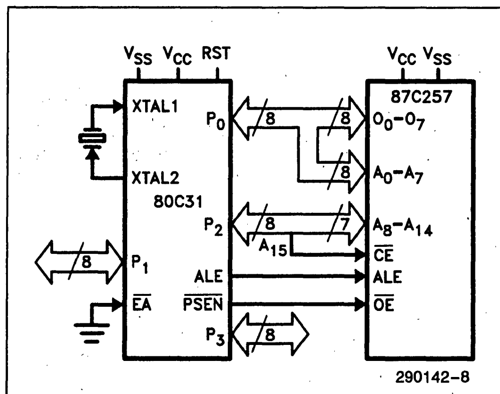


Figure 4. 80C31 with 87C257 System Configuration

Standby Mode

The standby mode substantially reduces V_{CC} current. When $\overline{CE}$ = V_{IH}, the standby mode places the outputs in a high impedance state, independent of the $\overline{OE}$ input.

Two Line Output Control

EPROMs are often used in larger memory arrays. Intel provides two contol inputs to accommodate multiple memory connections. Two-line control provides for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address decoder should enable $\overline{CE}$, while $\overline{OE}$ should be connected to all memory-array devices and the system's $\overline{READ}$ control line. This assures that only selected memory devices have active outputs while deselected memory devices are in low-power standby mode.

SYSTEM CONSIDERATIONS

EPROM power switching characteristics require careful device decoupling. System designers are interested in three supply current (ICC) issues—standby current levels, active current levels, and transient current peaks produced by falling and rising edges of Chip Enable. Transient current magnitudes depend on the device outputs' capacitive and inductive loading. Two-Line Control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 μF ceramic capacitor connected between its V_{CC} and GND. This high frequency, low inherent-inductance capacitor should be placed as close as possible to the device. Additionally, for every eight devices, a 4.7 μF electrolytic capacitor should be placed between V_{CC} and GND at the array's power supply connection. The bulk capacitor will overcome voltage slumps caused by PC board trace inductances.

PROGRAMMING MODES

Caution: Exceeding 14V on V_{PP} will permanently damage the device.

Initially, and after each erasure, all EPROM bits are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" are programmed, the data word

can contain both "1s" and "0s". Ultraviolet light erasure is the only way to change "0s" to "1s".

The programming mode is entered when V_{PP} is raised to its programming voltage (see Table 2). Data is programmed by applying an 8-bit word to the output pins (O_{0-7}). Pulsing $\overline{CE}$ to TTL-low while $\overline{OE} = V_{IH}$ will program data. TTL levels are required for address and data inputs.

Program Inhibit

The Program Inhibit mode allows parallel programming of multiple EPROMs with different data. With V_{PP} at its programming voltage, a $\overline{CE}$-low pulse programs the desired EPROM. $\overline{CE}$-high inputs inhibit programming of non-targeted devices. Except for $\overline{CE}$ and $\overline{OE}$, parallel EPROMs may have common inputs.

Program Verify

With V_{PP} and V_{CC} at their programming voltages, a verify (read) determines that bits are correctly programmed. The verify is performed with $\overline{CE} = V_{IH}$ and $\overline{OE} = V_{IL}$. Valid data is available t_{OE} after $\overline{OE}$ falls low.

Intelligent Identifier Mode

The Intelligent Identifier Mode will determine an EPROM's manufacturer and device type. Programming equipment can automatically match a device with its proper programming algorithm.

This mode is activated when programming equipment forces 12V $\pm$0.5V on the EPROM's A_9 address line. With A_1–A_8, A_{10}–A_{12} = V_{IL} (A_{13-14} are don't care), address line $A_0 = V_{IL}$ will present the manufacturer's code and $A_0 = V_{IH}$ the device code (see Table 1). When $A_9 = V_H$, ALE need not be toggled to latch each identifier address. This mode functions in the 25°C $\pm$5°C ambient temperature range required during programming.

ERASURE CHARACTERISTICS (FOR CERDIP EPROMS)

Exposure to light of wavelength shorter than 4000 Angstroms (Å) begins EPROM erasure. Sunlight and some fluorescent lamps have wavelengths in the 3000–4000Å range. Constant exposure to room-level fluorescent light can erase an EPROM in about 3 years (about 1 week for direct sunlight). Opaque labels over the window will prevent unintentional erasure under these lighting conditions.

The recommended erasure procedure is exposure to 2537Å ultraviolet light. The minimum integrated dose (intensity x exposure time) is 15 Wsec/cm^2. Erasure time using a 12000 $\mu W/cm^2$ ultraviolet lamp is approximately 15 to 20 minutes. The EPROM should be placed about 1 inch from the lamp. The maximum integrated dose is 7258 Wsec/cm^2 (1 week @ 12000 $\mu W/cm^2$). High intensity UV light exposure for longer periods can cause permanent damage.

CHMOS NOISE CHARACTERISTICS

System reliability is enhanced by Intel's CHMOS EPI-process techniques. Protection on each data and address pin prevents latch-up; even with 100 mA currents and voltages from $-1V$ to $V_{CC} + 1V$. Additionally, the V_{PP} pin is designed to resist latch-up to the 14V maximum device limit.

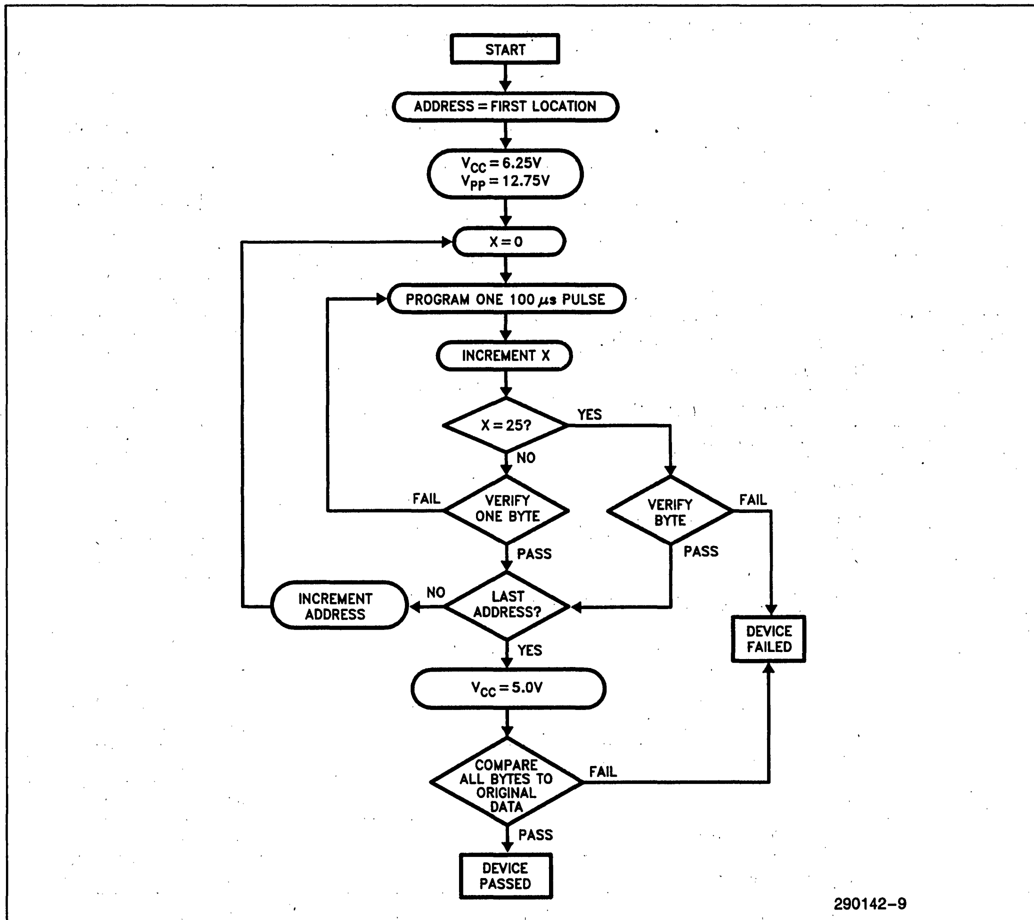


Figure 5. Quick-Pulse Programming Algorithm

Quick-Pulse Programming Algorithm

The Quick-Pulse Programming algorithm programs Intel's 87C257 EPROM. Developed to substantially reduce production programming throughput time, this algorithm can program a 87C257 in under four seconds. Actual programming time depends on the PROM programmer used.

The Quick-Pulse Programming algorithm uses a 100 microsecond initial-pulse followed by a byte verification to determine when the addressed byte is correctly programmed. The algorithm terminates if 25 100μs pulses fail to program a byte. Figure 5 shows the Quick-Pulse Programming algorithm flowchart.

The entire program-pulse/byte-verify sequence is performed with V_{CC} = 6.25V and V_{PP} = 12.75V. When programming is complete, all bytes should be compared to the original data with V_{CC} = 5.0V.

Alternate Programming

Intel's 27C256 Quick-Pulse Programming algorithms will also program the 87C257. By overriding a check for the Intelligent Identifier, older or non-upgraded PROM programmers can program the 87C257. See Intel's 27C256 data sheets for programming waveforms of these alternate algorithms.

In addition to the Quick-Pulse Programming Algorithm, the 87C257 has also been characterized for the Quick-Board Programming Algorithm. The Quick-Board Programming Algorithm was developed for specific automotive applications using Intel's 1.0 micron EPROM products. Contact the factory or an automotive sale representative for any information regarding the Quick-Board Programming Algorithm.

DC PROGRAMMING CHARACTERISTICS $T_A = 25°C \pm 5°C$

Table 2

Symbol	Parameter	Limits			Test Conditions
		Min	Max	Unit	
I_{LI}	Input Current (All Inputs)		1.0	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.2	0.8	V	
V_{IH}	Input High Level	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage During Verify		0.4	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	V_{CC} − 0.8		V	I_{OH} = −400 μA
I_{CC2}(3)	V_{CC} Supply Current		30	mA	
I_{PP2}(3)	V_{PP} Supply Current (Program)		50	mA	$\overline{CE} = V_{IL}$
V_{ID}	A_9 Intelligent Identifier Voltage	11.5	12.5	V	
V_{PP}(1)	Programming Voltage	12.5	13.0	V	
V_{CC}(1)	Supply Voltage During Programming	6.0	6.5	V	

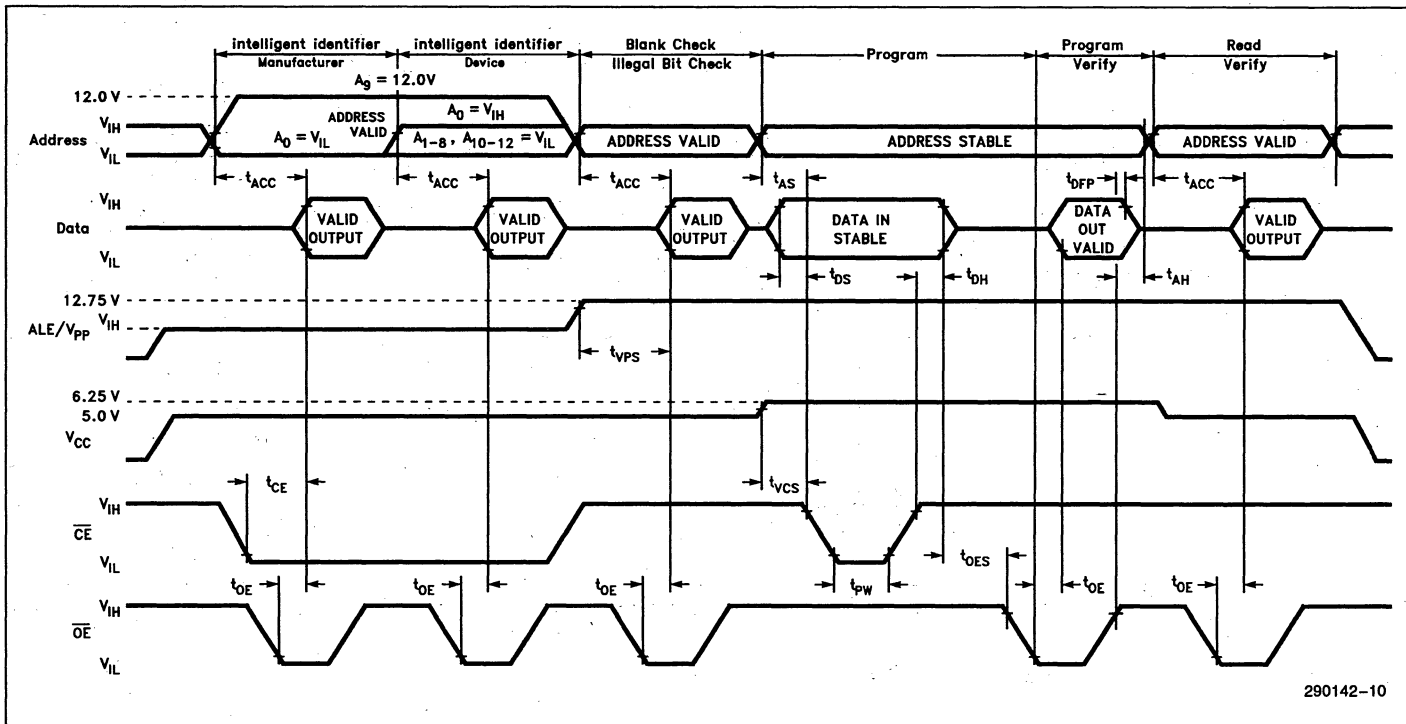
AC PROGRAMMING CHARACTERISTICS

$T_A = 25°C \pm 5°C$; see Table 2 for V_{CC} and V_{PP} voltages.

Symbol	Parameter	Limits				Conditions
		Min	Typ	Max	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}(2)	$\overline{OE}$ High to Output Float Delay	0		130	ns	
t_{VPS}(1)	V_{PP} Setup Time	2			μs	
t_{VCS}(1)	V_{CC} Setup Time	2			μs	
t_{PW}	$\overline{CE}$ Program Pulse Width	95	100	105	μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram.
3. The maximum current value is with outputs O_0 to O_7 unloaded.

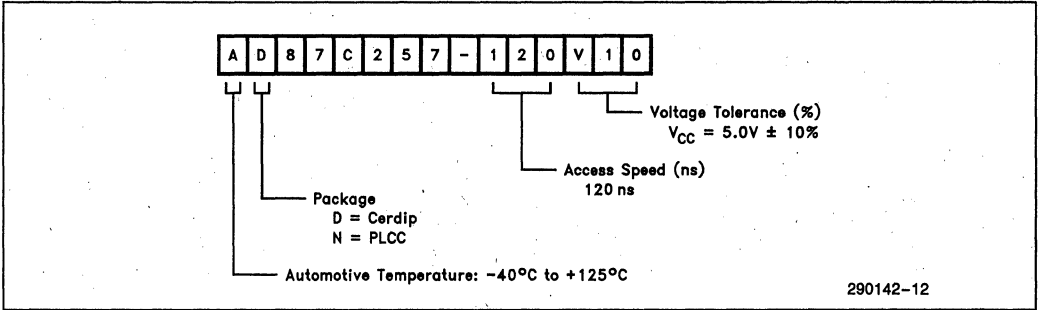
PROGRAMMING WAVEFORMS



NOTES:

1. The input timing reference level is $V_{IL} = 0.8V$ and $V_{IH} = 2V$.
2. t_{OE} and t_{DFP} are device characteristics but must be accommodated by the programmer.
3. To prevent device damage during programming, a 0.1 μF capacitor is required between V_{PP} and ground to suppress spurious voltage transients.
4. During programming, the address latch function is bypassed whenever $V_{PP} = 12.75V$ or $A_9 = V_H$. When V_{PP} and A_9 are at TTL levels, the address latch function is enabled, and the device functions in read mode.
5. V_{PP} can be 12.75V during Blank Check and Final Verify; if so, $\overline{CE}$ must be V_{IH}.

ORDERING INFORMATION



Valid Combinations:
AD87C257-120V10 AN87C257-120V10
AD87C257-200V10 AN87C257-200V10

REVISION HISTORY

Number	Description
004	Added the 120 ns speed bin

Memory Products

In today's vast array of semiconductor memory choices, the design engineer's job is to match memory characteristics to the application. Since 1971, Intel has offered a variety of memory devices to suit a wide range of applications.

Because of their inherent features, Intel's Flash Memory products are dramatically altering traditional computer memory solutions. Today's hand-held palmtop computers, personal organizers and notebook PCs need a solid-state storage technology that provides the optimum combination of performance, size, weight, low power and shock resistance demanded by today's mobile computer user. As a high density, non volatile, read/write semiconductor technology, Intel Flash Memory is the ideal memory medium to meet these seemingly competing requirements whether implemented in memory cards, solid-state disks or components.

This handbook is devoted to techniques and information to help design semiconductor memory into an application or system. Informative data sheets contain many comprehensive charts, block diagrams, operating characteristics and programming modes. Application notes provide diagrams and hardware design information. In addition, several interesting article reprints are included.

intel

Order Number: 210830-012
ISBN: 1-55512-172-1
Printed in USA/193/35K/RRD CC
Memory Products