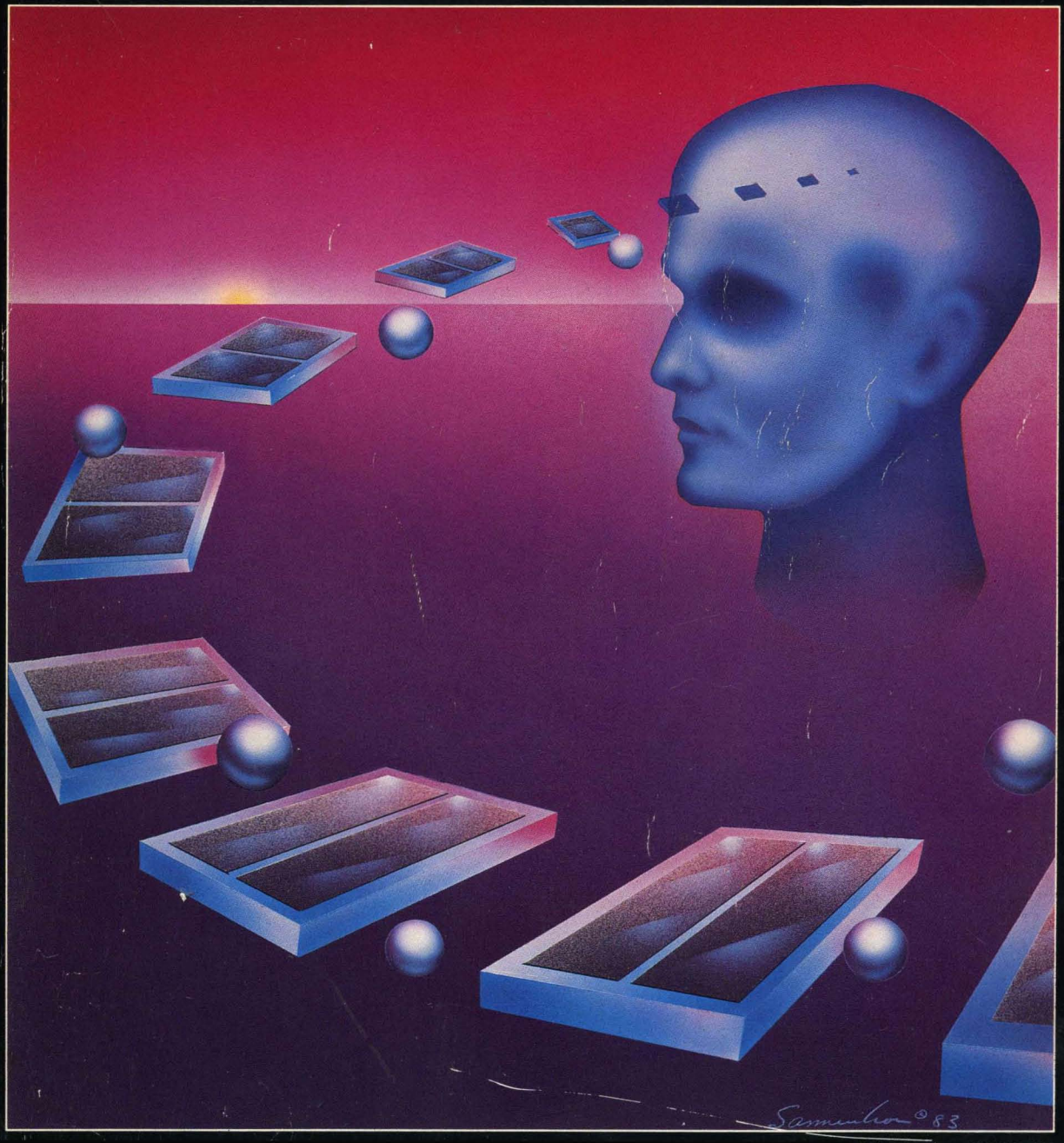
intel

Memory Components Handbook



Order Number: 210830-003

intel®

MEMORY COMPONENTS HANDBOOK

1984

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:

BITBUS, COMMputer, CREDIT, Data Pipeline, GENIUS, i, $\hat{\text{i}}$, ICE, iCS, iDBP, iDIS, I^2ICE, iLBX, i_m, iMMX, Insite, Intel, int$_e$l, int$_e$lBOS, Intelevision, int$_e$ligent Identifier, int$_e$ligent Programming, Intellec, Intellink, iOSP, iPDS, iSBC, iSBX, iSDM, iSXM, Library Manager, MCS, Megachassis, MICROMAINFRAME, MULTIBUS, MULTICHANNEL, MULTIMODULE, Plug-A-Bubble, PROMPT, Promware, QUEST, QUEX, Ripplemode, RMX/80, RUPI, Seamless, SOLO, SYSTEM 2000, and UPI, and the combination of ICE, iCS, iRMX, iSBC, MCS, or UPI and a numerical suffix.

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.

* MULTIBUS is a patented Intel bus.

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

Intel Corporation
Literature Department
3065 Bowers Avenue
Santa Clara, CA 95051

Table of Contents

Alphanumeric Index

PREFACE

This handbook has been prepared to provide a comprehensive grouping of technical literature covering Intel's memory products, with special emphasis on microprocessor applications. In addition, a brief summary of current memory technologies and basic segmentation of product lines is provided.

Memory Overview 1

CHAPTER I: MEMORY OVERVIEW

Joe Altnether

MEMORY BACKGROUND AND DEVELOPMENT

Only ten years ago MOS LSI memories were little more than laboratory curiosities. Any engineer brave enough to design with semiconductor memories had a simple choice of which memory type to use. The 2102 Static RAM for ease of use or the 1103 Dynamic RAM for low power were the only two devices available. Since then, the memory market has come a long way, the types of memory devices have proliferated, and more than 3,000 different memory devices are now available. Consequently, the designer has a lot to choose from but the choice is more difficult, and therefore, effective memory selection is based on matching memory characteristics to the application.

Memory devices can be divided into two main categories: volatile and non-volatile. Volatile memories retain their data only as long as power is applied. In a great many applications this limitation presents no problem. The generic term random access memory (RAM) has come to be almost synonymous with a volatile memory in which there is a constant rewriting of stored data.

In other situations, however, it is imperative that a non-volatile device be used because it retains its data whether or not power is applied. An example of this requirement would be retaining data during a power failure. (Tape and disk storage are also non-volatile memories but are not included within the scope of this book which confines itself to solid-state technologies in an IC form factor.)

Thus, when considering memory devices, it's helpful to see how the memory in computer systems is segmented by applications and then look at the state-of-the-art in these cases.

Read/Write Memory

First examine read/write memory (RAM), which permits the access of stored memory (reading) and the ability to alter the stored data (writing).

Before the advent of solid-state read/write memory, active data (data being processed) was stored and retrieved from non-volatile core memory (a magnetic-storage technology). Solid-state RAMs solved the size and power consumption problems associated with core, but added the element of volatility. Because RAMs lose their memory when you turn off their power, you must leave systems on all the time, add battery backup or store important data on a non-volatile medium before the power goes down.

Despite their volatility, RAMs have become very popular, and an industry was born that primarily fed computer systems' insatiable appetites for higher bit capacities and faster access speeds.

RAM Types

Two basic RAM types have evolved since 1970. Dynamic RAMs are noted for high capacity, moderate speeds and low power consumption. Their memory cells are basically charge-storage capacitors with driver transistors. The presence or absence of charge in a capacitor is interpreted by the RAM's sense line as a logical 1 or 0. Because of the charge's natural tendency to distribute itself into a lower energy-state configuration, however, dynamic RAMs require periodic charge refreshing to maintain data storage.

Traditionally, this requirement has meant that system designers had to implement added circuitry to handle dynamic RAM subsystem refresh. And at certain times, refresh procedures made the RAM unavailable for writing or reading; the memory's control circuitry had to arbitrate access. However, there are now two available alternatives that largely offset this disadvantage. For relatively small memories in microprocessor environments, the integrated RAM or iRAM provides all of the complex refresh circuitry on chip, thus, greatly simplifying the system design. For larger storage requirements, LSI dynamic memory controllers reduce the refresh requirement to a minimal design by offering a monolithic controller solution.

Where users are less concerned with space and cost than with speed and reduced complexity, the second RAM type — static RAMs — generally prove best. Unlike their dynamic counterparts, static RAMs store ones and zeros using traditional flip-flop logic-gate configurations. They are faster and require no refresh. A user simply addresses the static RAM, and after a very brief delay, obtains the bit stored in that location. Static devices are also simpler to design with than dynamic RAMs, but the static cell's complexity puts these non-volatile chips far behind dynamics in bit capacity per square mil of silicon.

The iRAM

There is a way, however, to gain the static RAM's design-in simplicity but with the dynamic RAM's higher

capacity and other advantages. An integrated RAM or iRAM integrates a dynamic RAM and its control and refresh circuitry on one substrate, creating a chip that has dynamic RAM density characteristics, but looks like a static RAM to users. You simply address it and collect your data without worrying about refresh and arbitration.

Before iRAM's introduction, users who built memory blocks smaller than 8K bytes typically used static RAMs because the device's higher price was offset by the support-circuit simplicity. On the other hand, users building blocks larger than 64K bytes usually opted for dynamic RAMs because density and power considerations began to take precedence over circuit complexity issues.

For the application area between these two limits, decisions had to depend on less straightforward tradeoffs. But iRAMs could meet this middle area's needs (See Figure 1).

Read-Only Memory

Another memory class, read-only memory (ROM), is similar to RAM in that a computer addresses it and then retrieves data stored at that address. However, ROM includes no mechanism for altering the data stored at that address — hence, the term read only.

ROM is basically used for storing information that isn't subject to change — at least not frequently. Unlike RAM, when system power goes down, ROM retains its contents.

ROM devices became very popular with the advent of microprocessors. Most early microprocessor applications were dedicated systems; the system's program was fixed and stored in ROM. Manipulated data could vary and was therefore stored in RAM. This application split caused ROM to be commonly called program storage, and RAM, data storage.

The first ROMs contained cell arrays in which the sequence of ones and zeros was established by a metalization interconnect mask step during fabrication. Thus, users had to supply a ROM vendor with an interconnect program so the vendor could complete the mask and build the ROMs. Set-up charges were quite high — in fact, even prohibitive unless users planned for large volumes of the same ROM.

To offset this high set-up charge, manufacturers developed a user-programmable ROM (or PROM). The first such devices used fusible links that could be melted or "burned" with a special programmer system.

Once burned, a PROM was just like a ROM. If the burn program was faulty, the chip had to be discarded. But, PROMs furnished a more cost-effective way to develop program memory or firmware for low-volume purposes than did ROMs.

As one alternative to fusable-link programming, Intel pioneered an erasable MOS-technology PROM (termed an EPROM) that used charge-storage programming. It came in a standard ceramic DIP package but had a window that permitted die exposure to light. When the chip was exposed to ultraviolet light, high energy photons could collide with the EPROM's electrons and scatter them at random, thus erasing the memory.

The EPROM was obviously not intended for use in read/write applications, but it proved very useful in research and development for prototypes, where the need to alter the program several times is quite common. Indeed, the EPROM market consisted almost exclusively of development labs. As the fabrication process became mature, however, and volumes increased, EPROM's lower prices made them attractive even for medium-volume production-system applications.

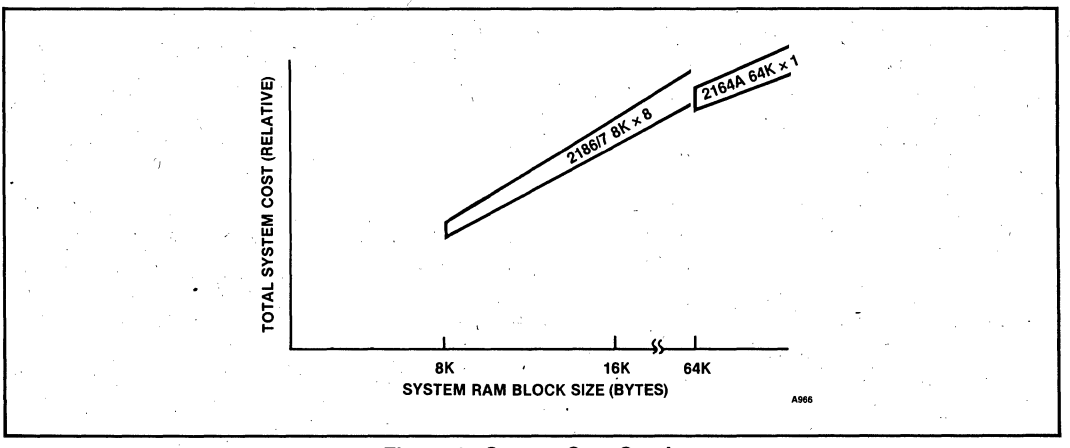


Figure 1. System Cost Graph

Another ROM technology advance occurred in 1980 with the introduction of Intel's 2816 — a 16K ROM that's user programmable and electrically erasable. Thus, instead of removing it from its host system and placing it under ultraviolet light to erase its program, the 2816 can be reprogrammed in its socket. Moreover, single bits or entire bytes can be erased in one operation instead of erasing the entire chip.

Such E^2PROMs (for electrically erasable programmable ROM) are opening up new applications. In point-of-sale terminals, for example, each terminal connects to a central computer but each can also handle moderate amounts of local processing. An E^2PROM can store discount information to be automatically figured in during a sales transaction. Should the discount change, the central computer can update each terminal via telephone lines by reprogramming that portion of the E^2PROM (Figure 2).

In digital instrumentation, an instrument could become self-calibrating using an E^2PROM. Should the instrument's calibration drift outside specification limits, the system could employ a built-in diagnostic to reprogram a parametric setting in an E^2PROM and bring the calibration back within limits.

E^2PROMs contain floating-gate tunnel-oxide (Flotox) cell structure. Based on electron tunneling through a thin (less than 200 Angstroms) layer of silicon dioxide, these cells permit writing and erasing with 21 Volt pulses. During a read operation, the chips use conventional +5 Volt power.

Bubble Memory

A very different device type, bubble memory was once considered the technology that would obsolete RAM components. This view failed to consider the inherent features and benefits of each technology. There is no question that RAMs have staked out a read/write applications area that is vast. Nevertheless, their volatility presents severe problems in more than a few applications. Remote systems, for example, might be unable to accept a memory that is subject to being wiped out should a power failure occur.

Bubble memories use a magnetic storage technique, roughly similar to the core memory concept but on a much smaller size and power-consumption scale. They are non-volatile and physically rugged. Thus, their first clear applications target has been in severe-environment and remote system sites. Portable terminals represent another applications area in which bubbles provide unique benefits.

Considering bubble products, Intel's latest design provides 1,048,576 bits of data storage via a defect-tolerant technique that makes use of 1,310,720 total bits (Figure 3). Internally, the product consists of 256 storage loops of 4,086 bits each. Coupled with available control devices, this single chip can implement a 128K byte memory subsystem.

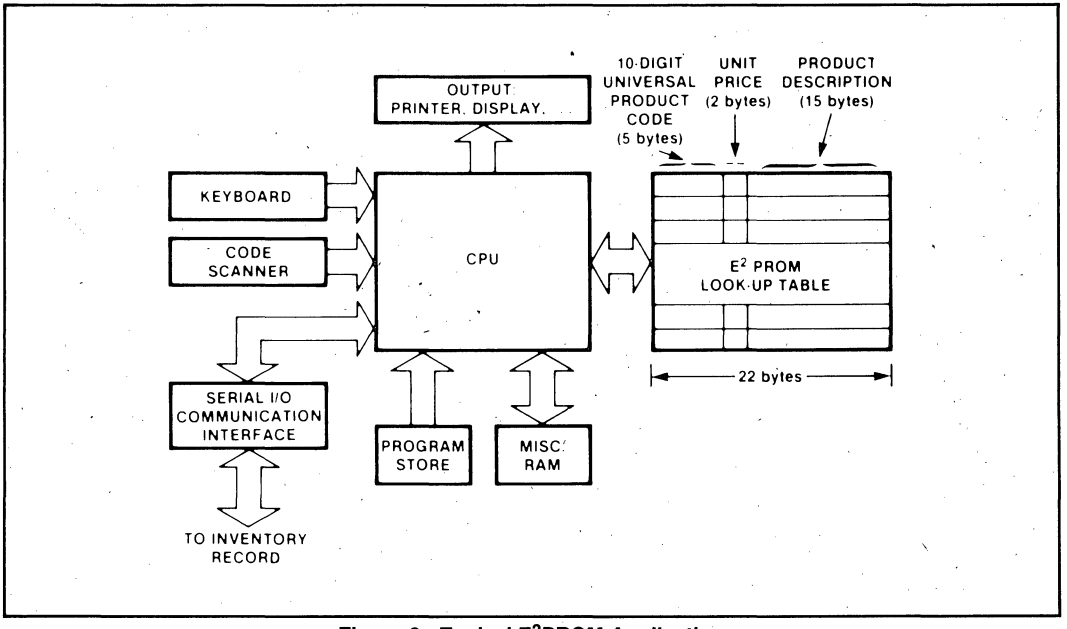


Figure 2. Typical E^2PROM Application

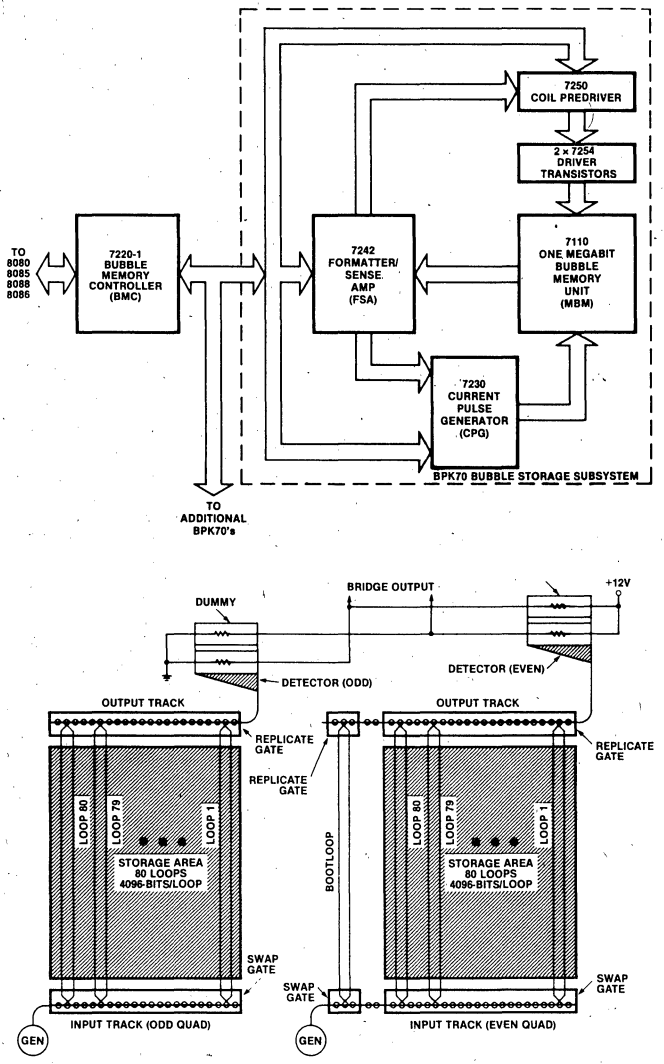


Figure 3. Intel Model 7110 Bubble Memory

SEGMENTATION OF MEMORY DEVICES

Besides the particular characteristics of each device that has been discussed, there are a number of other factors to consider when choosing a memory product, such as cost, power consumption, performance, memory architecture and organization, and size of the memory. Each of these factors plays a important role in the final selection process.

Performance

Generally, the term performance relates to how fast the device can operate in a given system environment. This

parameter is usually rated in terms of the access time. Fast SRAMs can provide access times as fast 20 ns, while the fastest DRAM cannot go much beyond the 100 ns mark. A bipolar PROM has an access time of 35 ns. RAM and PROM access is usually controlled by a signal most often referred to as Chip Select ($\overline{CS}$). $\overline{CS}$ often appears in device specifications. In discussing access times, it is important to remember that in SRAMs and PROMs, the access time equals the cycle time of the system whereas in DRAMs, the access time is always less than the cycle time.

Cost

There are many ramifications to consider when evaluating cost. Cost can be spread over factors such as design-in time, cost per device, cost per bit, size of memory, power consumption, etc.

Cost of design time is directly proportional to design complexity. For example, SRAMs generally require less design-in time than DRAMs because there is no refresh circuitry to consider. Conversely, the DRAM provides the lowest cost per bit because of its higher packing density.

Memory Size

Memory size is generally specified in the number of bytes (a byte is a group of eight bits). The memory size of a system is usually segmented depending upon the general equipment category. Computer mainframes and most of today's minicomputers use blocks of RAM substantially beyond 64K bytes — usually in the hundreds of thousands of bytes. For this size of memory, the DRAM has a significantly lower cost per bit. The additional costs of providing the refresh and timing circuitry are spread over many bits.

The microprocessor user generally requires memory sizes ranging from 2K bytes up to 64K bytes. In memories of this size, the universal site concept allows maximum flexibility in memory design.

Power Consumption

Power consumption is important because the total power required for a system directly affects overall cost. Higher power consumption requires bigger power supplies, more cooling, and reduced device density per board — all affecting cost and reliability. All things considered, the usual goal is to minimize power. Many memories now provide automatic power-down. With today's emphasis on saving energy and reducing cost, the memories that provide these features will gain an increasingly larger share of the market.

In some applications, extremely low power consumption is required, such as battery operation. For these applications, the use of devices made by the CMOS technology have a distinct advantage over the NMOS products. CMOS devices offer power savings of several magnitudes over NMOS. Non-volatile devices such as E^2PROMs are usually independent of power problems in these applications.

Power consumption also depends upon the organization of the device in the system. Organization usually refers to the width of the memory word. At the time of their inception, memory devices were organized as nK x 1 bits. Today, they are available in various configurations such as 4K×1, 16K×1, 64K×1, 1K×4, 2K×8, etc. As the device width increases, fewer devices are required to configure a given memory word — although the total number of bits remains constant. The wider organization can provide significant savings in power consumption, because a fewer number of devices are required to be powered up for access to a given memory word. In addition, the board layout design is simpler due to fewer traces and better layout advantages. The wider width is of particular advantage in microprocessors and bit-slice processors because most microprocessors are organized in 8-bit or 16-bit architectures. A memory chip configured in the nK×8 organization can confer a definite advantage — especially in universal site applications. All non-volatile memories other than bubble memories are organized nK×8 for this very reason.

Types of Memories

The first step to narrowing down your choice is to determine the type of memory you are designing — data store or program store. After this has been done, the next step is to prioritize the following factors:

Performance
Power Consumption
Density
Cost

Global Memory

Generally, a global memory is greater than 64K bytes and serves as a main memory for a microprocessor system. Here, the use of dynamic RAMs for read/write memory is dictated to provide the highest density and lowest cost per bit. The cost of providing refresh circuitry for the dynamic RAMs is spread over a large number of memory bits, thus minimizing the cost impact. Bubbles would also be an excellent choice for global memory where high performance is not required. In addition, bubbles offer low cost per bit and non-volatility.

Local Memory

Local memories are usually less than 64K bytes and reside in the proximity of the processor itself — usually on the same PC board. Two types of memories are

often used in local memory applications: RAMs and E²PROMs/EPROMs. These devices all offer universal site compatibility and density upgrade.

Synchronous and Asynchronous Memories

Historically, there have been several definitions of convenience when describing synchronous and asynchronous memory devices. The question of which definition is the more appropriate boils down to a philosophical decision, and depends on whether the definition is narrowed to component operating parameters or expanded to system operating parameters.

One popular and accepted definition defines the two types of memories by relying on the most apparent difference. The synchronous memory possesses an internal address register which latches the current device address, but the asynchronous device lacks this capability. The logic of this definition is easy to follow: Register transfer or sequential logic is considered synchronous because it is clocked by a common periodic signal — the system clock. Memories with internal address registers are also internally sequential logic arrays clocked by a signal, common throughout the memory system, and are, therefore, synchronous.

By the foregoing definition, asynchronous memories would require the device address be held valid on the bus throughout the memory cycle. Static RAMs fall into this category. In contrast, synchronous memories require the address to be valid only for a very short period of time just before, during, and just after the arrival of the address register clock. DRAMs and clocked static RAMs fall into this category.

With the introduction of the 2186 and 2187 iRAMs, the preceding definition no longer fits, because both devices have on chip address latches. Yet with respect to the system, one device operates synchronously and the other asynchronously.

Therefore, in considering memory devices or systems that operate within a specified cycle time, Intel defines a synchronous memory as one that responds in a predictable and sequential fashion, always providing data within the same time frame from the clock input. This allows a system designer to take advantage of the predictable access time and maximize his system performance by reducing or eliminating WAIT states.

Intel defines an asynchronous memory as one that (within the framework of the memory cycle specifications) does not output data in a predictable and repeatable time frame with respect to system timing. This is generally true of DRAM systems, where a refresh cycle, which occurs randomly skewed to the balance of the system timing, may be in progress at the time of a memory cycle request by the CPU. In this case, provision must be made to re-synchronize the system to the memory — usually with a READY signal. The 2186 iRAMs fit into this category, while the 2187 iRAMs are considered synchronous devices.

These definitions are somewhat broader in scope than those chosen in the past; however, as systems become implemented in silicon, a more global definition is required to encompass those former systems that are now silicon devices.

SUMMARY

Table 1 provides a summary of the various memory devices that have been discussed.

Table 1. Segmentation of Memory Devices

Operating From	Read Speed Fast	Read Speed Slow	Write Speed Fast	Write Speed Slow	Down Load	Size Small	Size Large	Removable (Archive)	System Level
Mass		Bubbles Disk		Bubbles Disk	N/A		Bubbles Disk	Bubbles Disk	Add on RAM Bubbles
Boot	EPROM		N/A	N/A	N/A	All[2]			N/A
Monitor	EPROM		N/A	N/A	N/A	All			N/A
Buffer	Bytewide	Bubbles	Bytewide	Bubbles	N/A	All	Bubbles X1		Add in RAM Bubbles
Diagnostics	E²/EPROM/ RAM		Bytewide	E² PROM	[1]Disk Bubbles	All	All[3]	Bubbles Disk	N/A
Operating System			N/A	N/A	[1]Disk Bubbles	All	X 1	Bubbles Disk	Add in/ Add on RAM
APP/PGM/ Data Store	E²/EPROM/ RAM		Bytewide X1	E² PROM	[1]Disk Bubbles	All	X 1	Bubbles Disk	Add in/ Add on RAM

[1]Down Loaded From Add on/Add in Bubbles
[2]E²/EPROM Bytewides
[3]X 1 Dram Bubbles Disk

Intel Memory Technologies

2

CHAPTER 2: INTEL MEMORY TECHNOLOGIES

Larry Brigham, Jr.

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 overwhelming 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. The majority of MOS memory devices in pro-

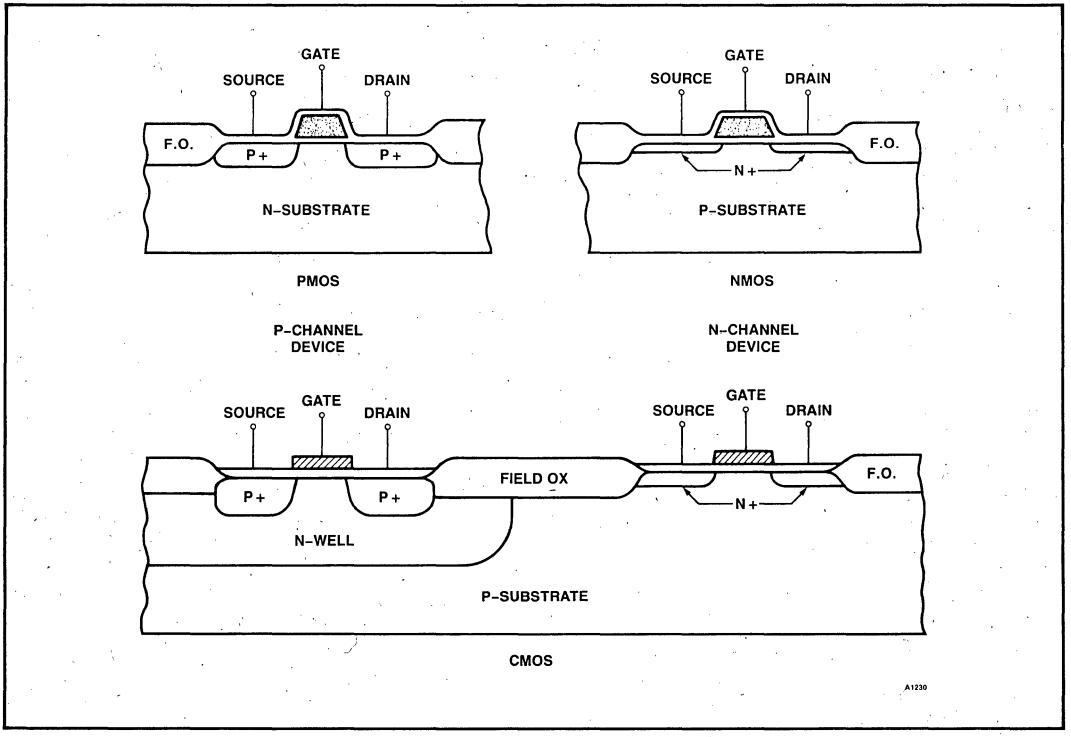


Figure 1. MOS Process Cross-sections

duction today are fabricated with NMOS technologies. CMOS technology has begun to see widespread commercial use in memory devices. It allows for very low power devices and these have been used for battery operated or battery back-up applications. Historically, CMOS has been slower than any NMOS device. Recently, however, CMOS technology has been improved to produce higher speed devices. Up to now, the extra cost processing required to make both transistor types has 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 will make a larger and larger number of memory devices practical in CMOS.

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 5 Volt single supply circuits. HMOS, along with its evolutionary counterparts HMOS II and HMOS III, CHMOS and CHMOS II (and their variants), 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 125 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.

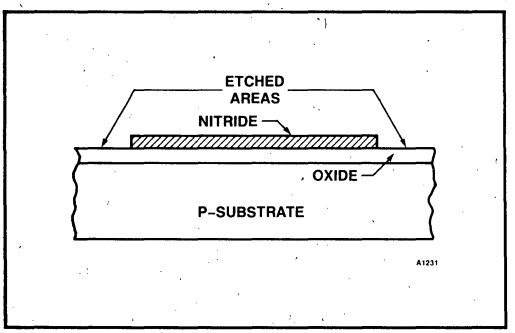


Figure 2. First Mask

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.

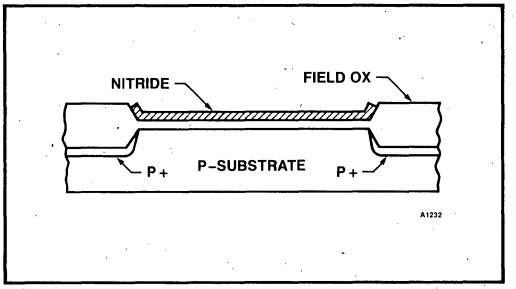


Figure 3. Post Field Oxidation

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 ohms/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).

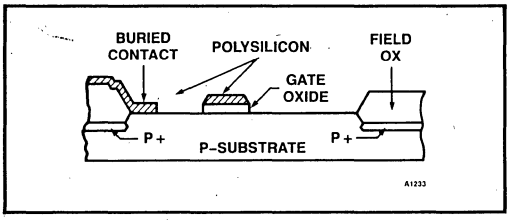


Figure 4. Post Gate Mask

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).

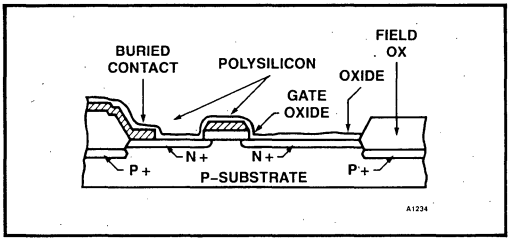


Figure 5. Post Oxidation

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 Al and diffusion or poly.

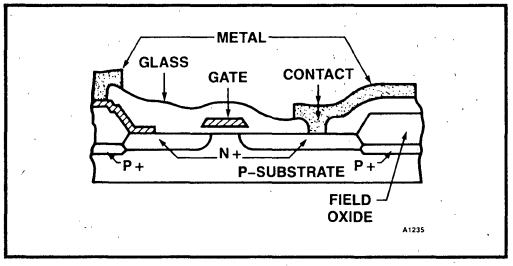


Figure 6. Completed Circuit (without passivation)

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, and E^2PROMs 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.

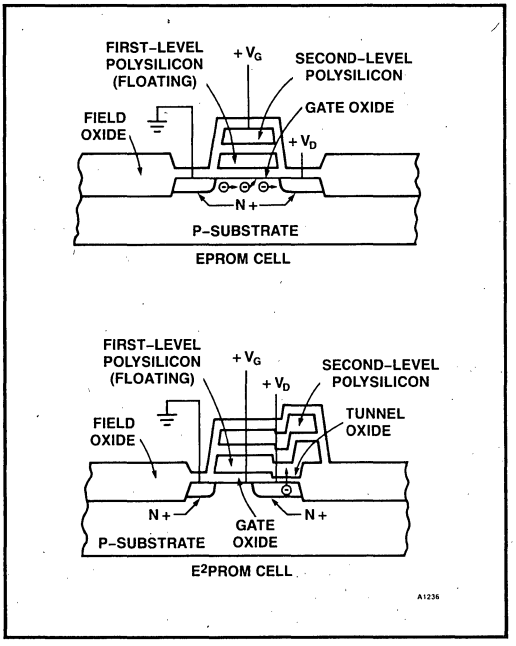


Figure 7. Double Poly Structure

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 for 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.

Table 1. 2164A 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.	
	Mark and Pack			
2.	Final QA	(See attached)		

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

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 $\cong$1/25,000 inch). The n-channel processes of the mid 1970's brought this down to approximately 5 microns, and today minimum geometries are less than 2 microns in production. This dramatic reduction 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 photoresist 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 solid 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 900-1200°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 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 .01 and .1 micron) and very tight control over thickness (typically ± .005 micron or less than ± 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 depositing 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 (HMOS I, HMOS II, HMOS III, CHMOS) 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.

RAMs (Random Access Memories)

3

intel

APPLICATION NOTE

AP-74

March, 1980

High Speed Memory System Design Using 2147H

Joe Altnether

Memory Components
Applications Engineering

INTRODUCTION

The Intel® 2147H is a 4096-word by 1-bit Random Access Memory, fabricated using Intel's reliable HMOS II technology. HMOS II, the second generation HMOS, is Intel's high performance n-channel silicon gate technology, making simple, high speed memory systems a reality. The purpose of this application note is to describe the 2147H operation and discuss design criteria for high speed memory systems.

TECHNOLOGY

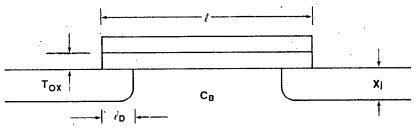
When Intel introduced the HMOS 2147, MOS static RAM performance took a quantum leap by combining scaling, internal substrate bias generation, and automatic powerdown. As a result, the 2147 has an access time of 55ns, density of 4096 bits, and power consumption of .99W active and .165W standby.

The high performance of the 2147 is further enhanced by the 2147H using HMOS II, a scaled HMOS process increasing the speed at the same power level which involves more than scaling dimensions.

Figure 1 shows the cross section of an HMOS device and lists the parameters of scaling, one of which is high device gain. The slew rate of an amplifier or device is proportional to the gain. Because faster switching speeds occur with high gain, the gain is maximized for high speed. Device gain is inversely proportional to the oxide thickness (T_{OX}) and device length (ℓ), consequently, scaling these dimensions increases the gain.

Another factor which influences performance is unwanted capacitance which appears in two forms - - diffusion and Miller. Diffusion capacitance is directly proportional to the diffusion depth (X_j) into the silicon, thus X_j must be reduced. Miller capacitance, the same phenomenon that occurs in the macro world of discrete devices, is proportional to the overlap length of the gate and the source (ℓ_D). Capacitance on the input shunts the high frequency portion of the input signal so that the device can only respond to low frequencies. Secondly, capacitance from the drain to the gate forms a feedback path creating an integrator or low pass filter which degrades the high frequency performance. This effect is minimized by reducing ℓ_D.

One of the limits on scaling is punch through voltage, which occurs when the field strength is too high, causing current to flow when the device is "turned off". Punch through voltage is a function of channel length (ℓ) and doping concentration (C_B), thus channel shortening can be compensated by increasing the doping



PERFORMANCE FACTORS

• HIGH DEVICE GAIN	GAIN $\alpha 1/(T_{OX}\ell)$
• LOW DIFFUSION CAPACITANCE	$C_P \alpha X_j$
• LOW MILLER CAPACITANCE	$C_m \alpha \ell_D$
• LOW BODY EFFECT	$\Delta V_T \alpha \sqrt{C_B}\, T_{OX}$

LIMITS

• PUNCH THROUGH VOLTAGE	$V_{PT} \alpha C_B \ell^2$
• THRESHOLD VOLTAGE	$V_T \alpha \sqrt{C_B}\, T_{OX}$

RESULT

- DECREASE ℓ, T_{OX}, X_j, ℓ_D
- INCREASE C_B

ℓ = CHANNEL LENGTH
T_{OX} = OXIDE THICKNESS
X_j = DIFFUSION DEPTH
ℓ_D = GATE OVERLAP
C_B = CONCENTRATION

Figure 1. HMOS Scaling

concentration. This has the additional advantage of balancing the threshold voltage which was decreased by scaling the oxide thickness for gain.

Comparison

Comparing scaling theory to HMOS II scaling in Table I, note that HMOS II agrees with scaling theory except for the supply voltage. It is left constant at +5V to maintain TTL compatibility. Had the voltage been scaled, the power would have been reduced by $1/K^3$ rather than $1/K$, but the device would not have been TTL compatible.

Table I. Scaling

	Theory	HMOS II
Dimensions	1/K	1/K
Substrate Doping	K	K
Voltage	1/K	1
Device Current	1/K	1
Capacitance A/T	1/K	1/K
Time Delay VC/I	1/K	1/K
Power Dissipation VI	$1/K^2$	1
Power Delay Product	$1/K^3$	1/K

THE DEVICE

The 2147H is TTL compatible, operates from a single +5 volt supply, and is easy to use.

Figure 2 shows the pin configuration and the logic symbol. The 2147H is compatible with the 2147 allowing easy system upgrade. Contained in an industry standard 18-pin dual in-line package the 2147H is organized as 4096 words of 1 bit. To access each of these words, twelve address lines are required. In addition, there are two control signals: $\overline{CS}$, which activates the RAM; and $\overline{WE}$,

which controls the write function. Separate data input and output are available. Logical operation of the 2147H is shown in the truth table. The output is in the high impedance or three-state mode unless the RAM is being read. Power consumption switches from standby to active under control of $\overline{CS}$.

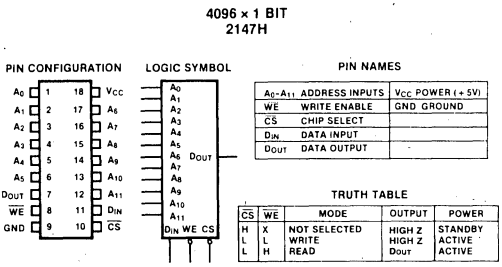


Figure 2. 2147H Logic Diagram

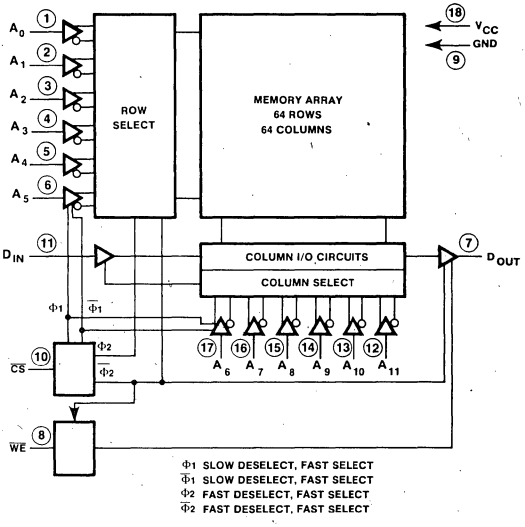


Figure 3. 2147H Block Diagram

Internal structure of the 2147H is shown in the block diagram of Figure 3. The major portions of the device are: addresses, control ($\overline{CS}$ and $\overline{WE}$), the memory array and a substrate bias generator, which is not shown.

The memory is organized into a two-dimensional array of 64 rows and 64 columns of memory cells. The lower-order six addresses decode one of 64 to select the row while the upper-order six addresses decode to select one column. The intersection of the selected row and the selected column locate the desired memory cell. Additional logic in the column selection circuit controls the flow of data to the array and as stated in the truth table, $\overline{WE}$ controls the output buffer.

As shown in Figure 4, the first three stages of the address buffer are designed with an additional transistor. In each stage, the lowest transistors are the active devices, the middle transistors are load devices, while the upper transistors, controlled by Φ_1, are the key to low standby power. Forming an AND function with the active devices, the upper transistors are turned off when the 2147H is not active, minimizing power consumption. Without them, at least one stage of these cascaded amplifiers would always be consuming power.

The signal Φ_1, and its inverse $\overline{\Phi}_1$, are generated from CS. They are part of an innovative design not found in the earlier 2147. Their function is to minimize the effects at short deselect times on the Chip Select access time, t_{ACS}.

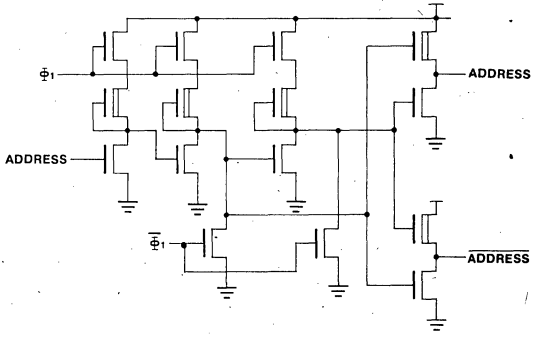


Figure 4. Address Buffer.

For both the 2147 and the 2147H, access is delayed until the address buffers are activated by chip selection. In the standard 2147, priming during deselection compensates for this delay by speeding up the access elsewhere in the circuitry. For short deselect times, however, full compensation does not occur because priming is incomplete. The result is a pushout in t_{ACS} for short deselect times.

In the 2147H, the address buffers are controlled by Φ_1, which is shaped as shown in Figure 5. Φ_1 activates rapidly for fast select time. However, Φ_1 deactivates slowly, keeping the address buffers active during short deselect times to speed access. As shown in Figure 6, this design innovation keeps t_{ACS} pushout to less than 1 ns.

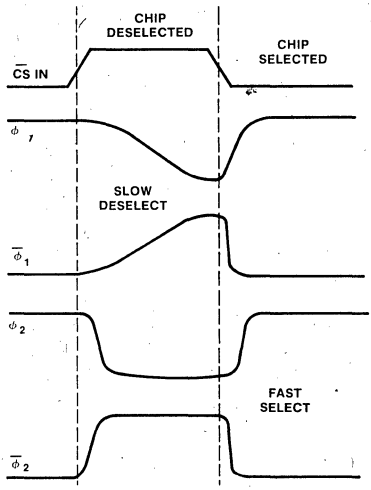


Figure 5. CS Buffer Signals

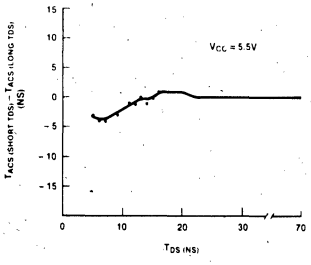


Figure 6. CS Access Vs. Deselect Time

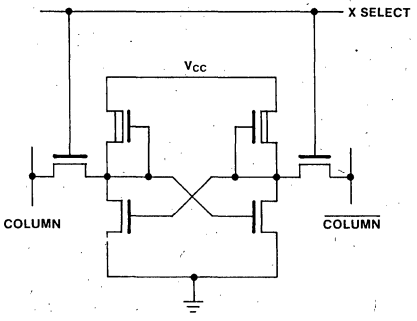


Figure 7. 2147H Memory Cell

Figure 7 shows the standard six-transistor cell. Configured as a bi-stable flip-flop, the memory cell uses two transistors for loads and two for active devices so that the data is stored twice as true and compliment. The two remaining transistors enable data onto the internal I/O bus. Unlike the periphery, the cell is not powered down during deselect time to sustain data indefinitely.

The 2147H has an internal bias generator. Bias voltage allows the use of high resistivity substrate by adjusting the threshold voltages. In addition, it reduces the effect of bulk silicon capacitance. As a result, performance is enhanced. Bias voltage is generated by capacitively coupling the output of a ring oscillator to a charge pump connected to the substrate. Internally generated bias permits the 2147H to operate from a single +5 volt supply, maintaining TTL compatibility.

2147H SUBSTRATE BIAS GENERATOR

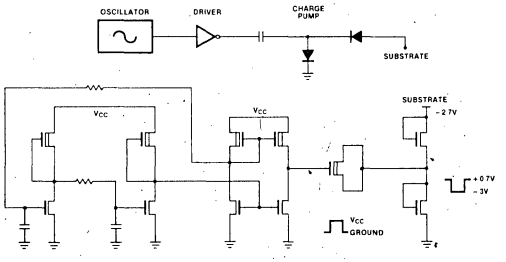


Figure 8. 2147H Substrate Bias Generator

DEVICE OPERATION

READ MODE

With power applied and $\overline{CS}$ at greater than 2V, the 2147H is in the standby mode, drawing less than 30mA. Activating $\overline{CS}$ begins access of the cell as defined by the state of the addresses. Data is transferred from the cell to the output buffer. Because the cell is static, the read operation is non-destructive. Device access and current are shown in Figure 9. Maximum access relative to the leading edge of $\overline{CS}$ is 35 ns for a 2147H-1. Without clocks, data is valid as long as address and control are maintained.

WRITE MODE

Data is modified when the write enable $\overline{WE}$ is activated during a cycle. At this time, data present at the input is duplicated in the cell specified by the address. Data is latched into the cell on the trailing edge of $\overline{WE}$, requiring that setup and hold times relative to this edge be maintained.

Two modes of operation are allowed in a write cycle, as shown in Figure 10. In the first mode, the write cycle is controlled by $\overline{WE}$, while in the other cycle, the cycle is controlled by $\overline{CS}$. In a $\overline{WE}$ controlled cycle, $\overline{CS}$ is held active while addresses change and the $\overline{WE}$ signal is pulsed to establish memory cycles. In the $\overline{CS}$ controlled cycle, $\overline{WE}$ is maintained active while addresses again change and $\overline{CS}$ changes state to define cycle length. This flexible operation eases the use and makes the 2147H applicable to a wide variety of system designs.

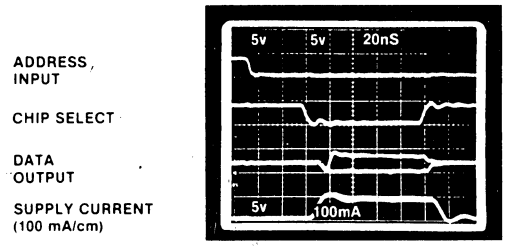
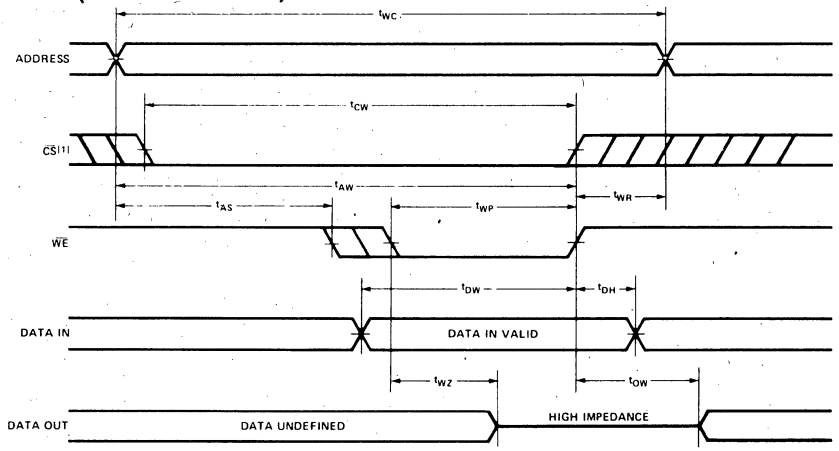


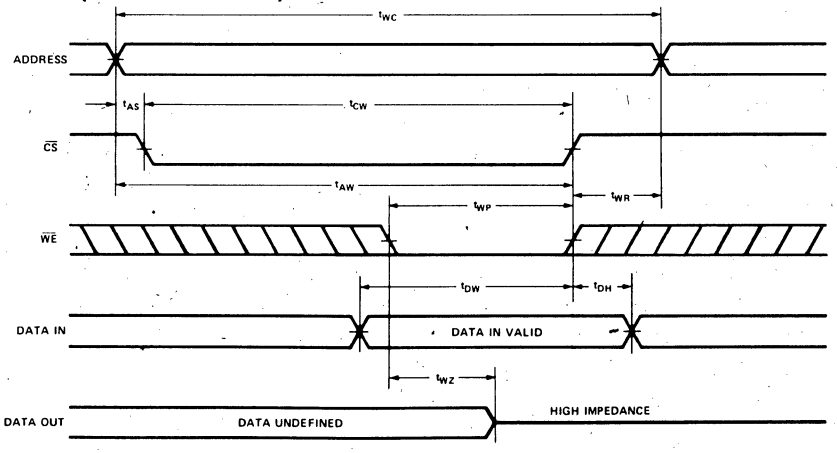
Figure 9. 2147H Access and Power Photo

WAVEFORMS

WRITE CYCLE #1 ($\overline{WE}$ CONTROLLED)



WRITE CYCLE #2 ($\overline{CS}$ CONTROLLED)



Note: 1. If $\overline{CS}$ goes high simultaneously with $\overline{WE}$ high, the output remains in a high impedance state.

Figure 10. Write Cycle Modes of Operation

EFFECT OF POWER DOWN AT THE SYSTEM LEVEL

Power consumed by a memory system is the product of the number of devices, the voltage applied, and the average current:

Equation 1

$$P = NVI_{AVE}$$

where:
P = Power
N = Number of devices
V = Voltage applied
I_{AVE} = Average current/device

Without power down, the average current is approximately the operating current. System power increases linearily with the number of devices. With power down, power consumption increases in proportion to the standby current with increasing number of memory devices. Curves in Figure 11 illustrate the difference which results from the majority of devices being in standby with a very small portion of the devices active or being accessed. For a system with power down, the average current of a device in the system is the sum of total active current and the total standby current divided by the number of devices in the system. For an X1 memory such as the 2147H, the number of active devices in most systems will be equal to the number of bits/word, m. Therefore, the number of devices in standby is the difference between N and M. I_{AVE} is expressed mathematically:

Equation 2

$$I_{AVE} = \frac{mI_{ACT} + (N-m)\, I_{SB}}{N}$$

where:
m = Number of active devices
I_{ACT} = Active current
I_{SB} = Standby current

The graph of Figure 12 shows the relation between average device current and memory size for automatic power down. For large memories the average device current approaches the standby current. Total system power usage, P, is calculated by substituting Equation 2 into Equation 1.

$$P = V[mI_{ACT} + (N-m)\, I_{SB}]$$

Comparison of power consumption of a system with and without power down illustrates the power savings. Assume a 64K by 18-bit memory constructed with 4KX1 devices. Active current of one device is 180mA and standby current is 30mA. Duty cycle is assumed to be 100% and voltage is 5 volts. The number of devices in the system is:

$$N = \frac{64K \text{ words} \times 18 \text{ bits/word}}{4K \text{ bit/device}}$$

$$N = 288 \text{ devices}$$

WITHOUT POWER DOWN:

$$P_{NPD} = 288 \text{ devices} \times 5 \text{ volts} \times 180 \text{ mA/device}$$
$$P_{NPD} = 259.2 \text{ watts}$$

WITH POWER DOWN:

With power down only 18 devices are active — 18 bits/word — and 270 are in standby.

P_{WPD} = 5 volts [18 devices (180mA/device) + 270 devices (30 mA/device)]

P_{WPD} = 56.7 watts

The system with power down devices uses only 22% of the power required by a non-powerdown memory system.

EFFECT OF POWER DOWN AT THE SYSTEM LEVEL

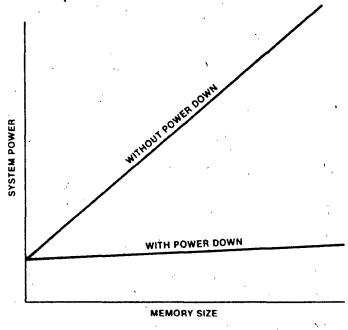


Figure 11. Effect of Power Down at the System

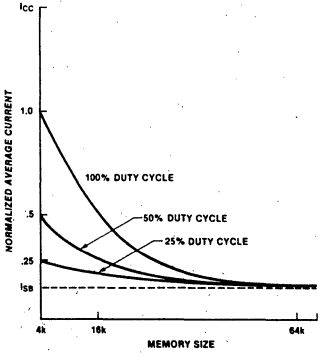


Figure 12. Average Current as a Function of Memory Size

POWER-ON

When power is applied, two events occur that must be considered: substrate bias start up and TTL instability. Without the bias generator functioning (V_{CC} less than 1.0 volts), the depletion mode transistors within the device draw larger than normal current flow. When the bias generator begins operation (V_{CC} greater than 1.0 volts), the threshold of these transistors is shifted, decreasing the current flow. The effect on the device power-on current is shown in Figure 13.

For V_{CC} values greater than 1.0 v., total device current is a function of both the substrate bias start-up characteristic and TTL stability. During power-on, the TTL circuits are attempting to operate under conditions which violate their specifications; consequently the $\overline{CS}$ signals can be indeterminent. One or several may be low, activating one or more banks of memory. The combined effects of this and the substrate bias start-up characteristic can exceed the power supply rating. The V-I characteristic of a power supply with fold back reduces the supply voltage in this situation, inhibiting circuit operation. In addition, the TTL drivers may not be able to supply the current to keep the $\overline{CS}$ signals deactivated.

One of several design techniques available to eliminate the power-on problem is power supply sequencing. Memory supply voltage and TTL supply voltage are separated, allowing the TTL supply to be activated first. When all the $\overline{CS}$ signals have stabilized at 2.0V or greater, the memory supply is activated. In this mode the memory power-on current follows the curve marked $\overline{CS}$ = V_{CC} in Figure 13.

If power sequencing is not practical, an equally effective method is to connect the $\overline{CS}$ signal to V_{CC} through a 1KΩ resistor. Although this does not guarantee a 2.0V $\overline{CS}$ input; emperical studies indicate that the effect is the same.

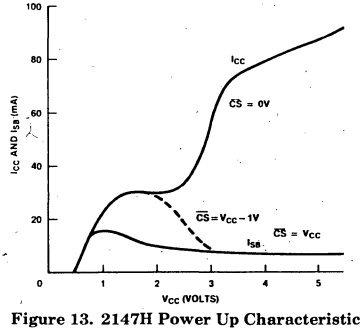


Figure 13. 2147H Power Up Characteristic

ARRAY CHARACTERISTICS

When two or more RAMs are combined, an array is formed. Arrays and their characteristics are controlled by the printed circuit card which is the next most important component after the memory device itself. In addition to physically locating the RAMs, the p.c. board must route power and signals to and from the RAMs.

GRIDDING

A power distribution network must provide required voltage, which from the 2147H data sheet is 5.0 volts ±10% to all the RAMs. A printed circuit trace, being an extremely low DC resistance, should easily route +5v DC to all devices. But as the RAMs are operating, micro circuits within the RAMs are switching micro currents on and off, creating high frequency current transients on the distribution network. Because the transients are high frequency, the network no longer appears as a "pure" low resistance element but as a transmission line. The RAMs and the lumped equivalent circuits of the transmission line are drawn in Figure 14. Each RAM is separated by a small section of transmission line both on the +voltage and the —voltage. Associated with the transmission lines is a voltage attenuation factor. In terms of AC circuits, the voltage across the inductor is the change in current — switching transient — multiplied by the inductance.

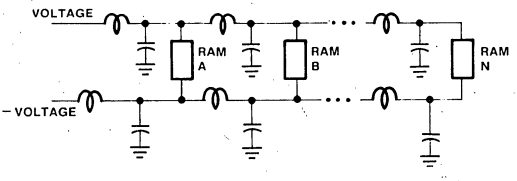


Figure 14. Equivalent Circuit for Distribution

Assuming all RAMs act similarly, the first inductor will see N current transients and the inductor at RAM B sees N-1 transients. The total differential is:

$$\Delta V = \sum_{n=1}^{N} n L \frac{di_n}{dt}$$

That voltage tolerance of ±10% could easily be exceeded with excursions of ±1 volt not uncommon. Measures must be taken to prevent this. The characteristic impedance of a transmission line is shown in Figure 15A.

Connecting two transmission lines in parallel will halve the characteristic impedance. The result is shown in Figure 15B.

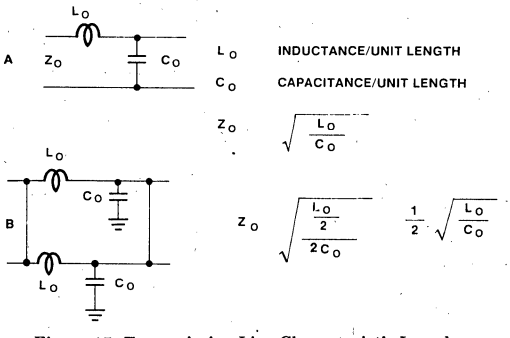


Figure 15. Transmission Line Characteristic Impedance

Paralleling N traces will reduce the impedance to Zo/N. Extrapolation of this concept to its limit will result in an infinite number of parallel traces such that they are physically touching, forming an extremely wide, low impedance trace, called a plane. Distribution of power (+ voltage) and ground (- voltage) via separate planes provides the best distribution.

P.C. boards with planes are manufactured as multi-layer boards sandwiching the power and ground planes internally. Characteristics of a multilayer board can be cost effectively approximated by gridding the power and ground distribution. Gridding surrounds each device with a ring of power and ground distribution forming many parallel paths with a corresponding reduction of impedance. Gridding is easily accomplished by placing horizontal traces of power (and ground) on one side of the pc board and vertical traces on the other, connected by plated through holes to form a grid.

Viewed from the top of the p.c. board, the gridding as in Figure 16 surrounds each device. Pseudo-gridding techniques such as serpentine or interdigitated distribution, as in Figure 17, are not effective because there are no parallel paths to minimize the impedance.

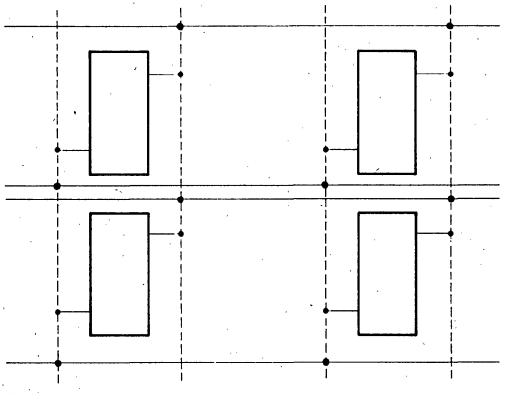


Figure 16. Gridding Plan

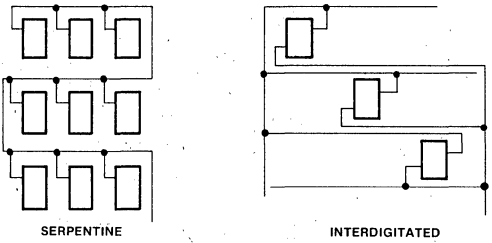


Figure 17. Pseudo-Gridding Techniques

DECOUPLING

One final aspect of power/ground distribution must be considered - decoupling.

Decoupling provides localized charge to minimize instantaneous voltage changes on the power grid due to current changes. These transient current changes are local and high frequency as devices are selected and deselected. Adequate decoupling for the 2147H is accomplished by placing a 0.1 μf ceramic capacitor at every other device as shown in Figure 18. Bulk decoupling is included on the board to filter low frequency noise in the system power distribution. One tantalum capacitor of 22 to 47 μf per 16 devices provides sufficient energy storage. By distributing these capacitors around the board several small currents exist rather than one large current flowing everywhere. Smaller voltage differentials - voltage is proportional to current - are experienced and the voltage remains in the specified operating range. Figure 19 demonstrates the difference with and without gridding.

TERMINATION

Similar reasoning is applied to the a.c. signals: address, control, and data. While they are not gridded or decoupled, they must be kept short and terminated. Similar to the power trace, the signal

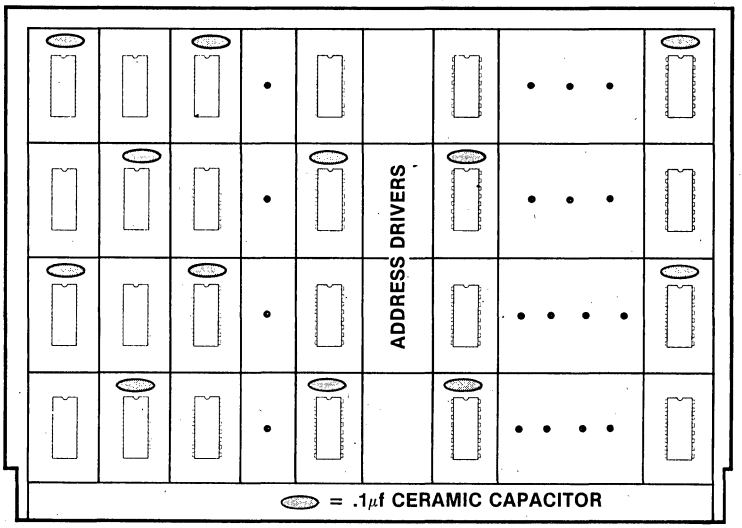
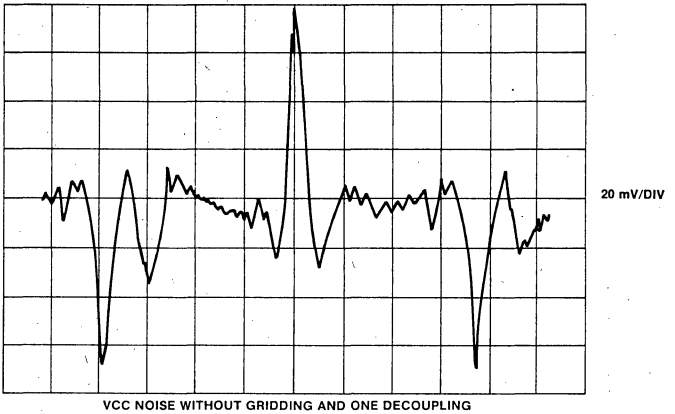
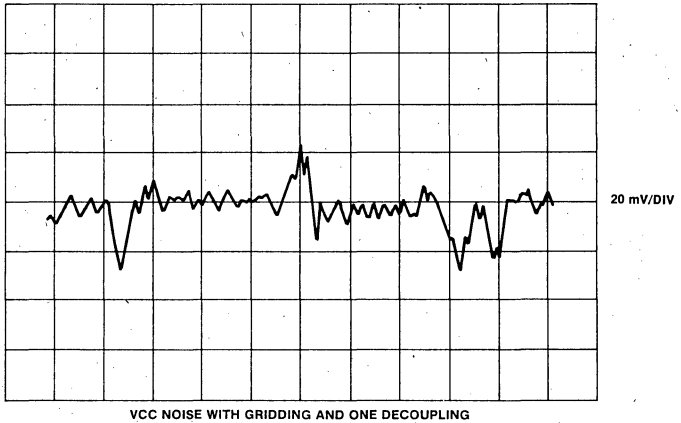


Figure 18. Decoupling



VCC NOISE WITHOUT GRIDDING AND ONE DECOUPLING CAPACITOR PER 4 RAMS



VCC NOISE WITH GRIDDING AND ONE DECOUPLING CAPACITOR PER 2 RAMS

Figure 19. V_{CC} Noise With & Without Gridding

trace will have transmission line characteristics. A simplified circuit is shown in Figure 20.

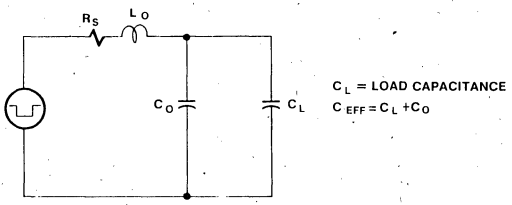


Figure 20. Signal Equivalent Circuit

MOS RAM input is essentially capacitive. Simplifying the capacitance and writing the differential equation.

$$\vartheta = L\frac{di}{dt} + \frac{1}{C}\int idt$$

The solution of this equation is:

$i = K_1 e^{-r_1 t} + K_2 e^{-r_2 t}$

where:

$$r_1 = \frac{R}{2L} + \sqrt{\frac{R^2}{4L^2} - \frac{1}{LC}}$$

$$r_2 = \frac{R}{2L} - \sqrt{\frac{R^2}{4L^2} - \frac{1}{LC}}$$

K_1 = constant

K_2 = constant

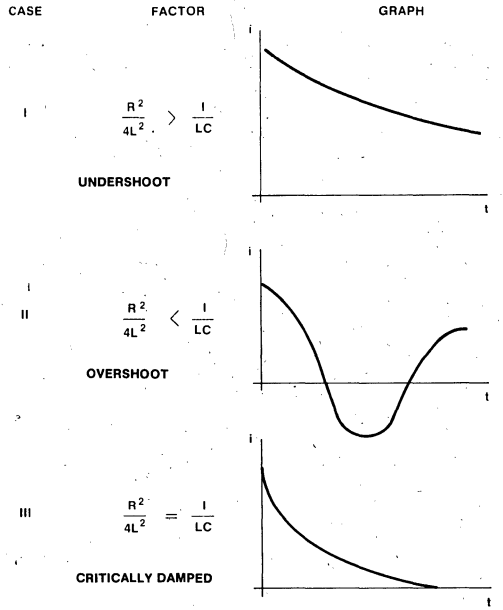


Figure 21. Three Cases of Equation Solution

Dependent on the values of R, L and C, there are three cases shown in Figure 21. In case I, rise and fall times are excessively long. In case III, the current smoothly and clearly changes, while in case II, the current overshoots and rings. If ringing is severe enough, the voltage can cross the threshold voltage of the device as in Figure 22.

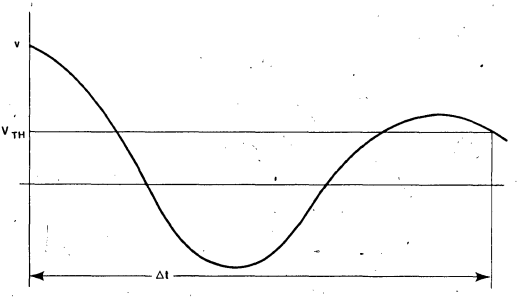


Figure 22. Access Push-Out Due to Ringing

Effective access is stretched out until the wave form settles. System access is the settling time (Δt) plus the specified device access. Case III is the ideal case but in reality a compromise between case I and case II is used because parameters vary in a production environment. Enough series resistance is inserted to prevent ringing but not enough to significantly slow down the access. A series resistance of 33Ω provides this compromise. The exact value is determined emperically but 33Ω is a good first approximation.

SERIES TERMINATION/ PARALLEL TERMINATION

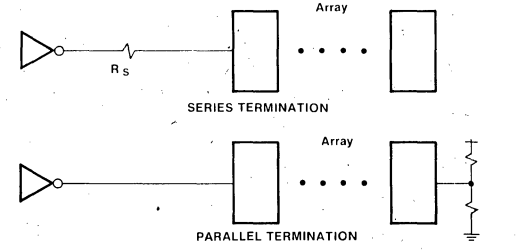


Figure 23. Series and Parallel Termination

Series termination uses one resistor and consumes little power. Current through the resistor creates a voltage differential shifting the levels of input voltage to the devices slightly. This shift is usually insignificant because the 2147H has an extremely high input impedance.

Termination could also be accomplished by a parallel termination as shown in Figure 23.

Parallel termination has the advantage of faster rise and fall times but the disadvantage of higher power consumption and increased board space usage.

SYSTEM DELAYS

RAMs are connected to the system through an interface, comprised of address, data and control signals. Inherent in the interface is propagation delay. Added to the RAM access time, propagation delay lengthens system access time and hence system cycle time. Expressed as an equation:

$$t_{sa} = t_{da} + t_{pd}$$

where: t_{sa} = system access time
t_{da} = device access time
t_{pd} = propagation delay

Device access is a fixed value, guaranteed by the data sheet. System efficiency then, is a function of system access and can be expressed as:

$$Eff = t_{da}/t_{sa}$$

where: Eff = System Efficiency

This can be reduced by substitution for tsa to:

$$Eff = 1/(1 + t_{pd}/t_{da})$$

System efficiency is maximized when propagation delay is minimized. With sub 100 ns access RAMs, efficiency can be reduced to 40-60% because delay through the signal paths is significant when compared to RAM access. Three factors contribute to the delay: logic delay, capacitive loading, and transit time.

LOGIC DELAY

The delay through a logic element is the time required for the output to switch with respect to the input. Actual delay times vary. Maximum TTL delays are specified in catalogs, while minimum delays are calculated as one-half of the typical specification. As an example, a gate with a typical delay of 6 ns has a minimum delay of 3 ns.

A signal propagating through two logically identical paths but constructed from different integrated circuits will have two different propagation times. For example, in Figure 24A one path has minimum delays while the other has maximum delays. Path A-B has a delay of 3.5 ns while A-B^1 has a delay of 11 ns. The time difference between these two signals is skew, which will be important later in the system design. Figure 24B shows skew values for several TTL devices.

CAPACITIVE LOADING

Delay time is also affected by the capacitive load on the device. Typical delay as a function of capacitive load is shown in Figure 25. TTL data sheets specify the delay for a particular capacitive load

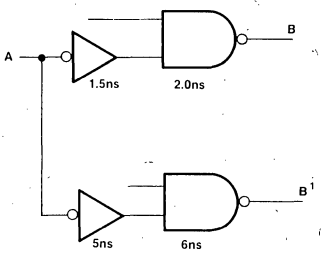


Figure 24A.

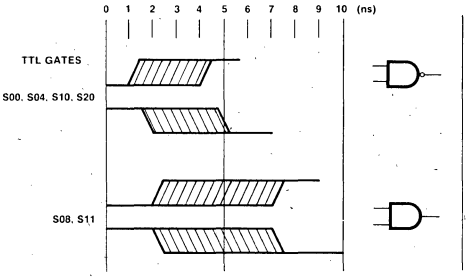


Figure 24B. Skew

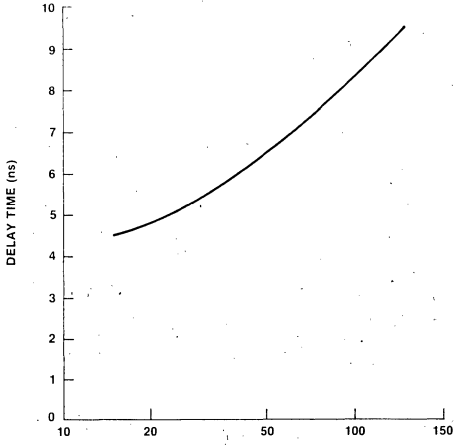


Figure 25. Capacitive Loading

(typically 15pF or 50 pF). Loads greater than specified will slow the device; similarly, loads less than specified will speed up the device.

A value of 0.05 ns/ft is a linear approximation of the function in Figure 25 and is used in the calculations. Loading effect is calculated by subtracting the actual load from the specified load. This difference is multiplied by 0.05 ns/pF and the result algebraically subtracted from the specified delay. As an example, a device has a 4 ns delay driving 50 pF, but the actual load is 25 pF. Then,

50 pF specified
-25 pF actual
25 pF difference

25 pF × 0.05 ns/pF = 1.25 ns

4 ns specified
-1.25ns difference
2.75 ns actual delay

A device specified at 4 ns while driving 50 pF will have a delay of only 2.75 ns when driving 25 pF. Conversely, the same device driving 75 pF would have a propagation time of 5.25 ns.

TRANSIT TIME

Signal transit time, the time required for the signal to travel down the P.C. trace, must also be considered. As was shown in Figure 19, these traces are transmission lines. Classical transmission line theory can be used to calculate the delay:

$$t_p = \sqrt{LC}$$

where: t_p = Travel Time
L = Inductance/unit length of trace
C = Capacitance/unit length of trace

The capacitance term in the equation is modified to include the sum of the trace capacitance and the device capacitance. This equation approximates in the worst case direction; a signal will never "see" all the load capacitance simultaneously, it is distributed along the trace at the devices.

Substituting into the equation:

$$tp^1 = \sqrt{L(C + C_L)}$$

where: tp^1 = Modified delay
C_L = Load capacitance

Algebraically:

$$tp^1 = \sqrt{LC(1 + C_L/C)}$$

$$tp^1 = \sqrt{LC}\ \sqrt{1 + CL/C}$$

and

$$tp^1 = tp\ \sqrt{1 + CL/C}$$

Emperically, tp is 1.8 ns/ft for G-10 epoxy and C is 1.5 pF/in. For a 5-in. trace and a 40 pF load, the delay is calculated to be 4.5 ns. Because this is worst case, an approximated 2 ns/ft can be used. In the following sections, however, the equation will be used. Total delay is the summation of all the delays. Adding the device access, TTL delays and the trace delays result in the system access.

BOARD LAYOUT

The preceding section discussed the effects of trace length and capacitive loading. Proper board layout minimizes these effects.

As shown in Figure 26, address and control lines are split into a right- and left-hand configuration with these signals driving horizontally. This configuration minimizes propagation delay. Splitting the data lines is not necessary, as the data loads are not as great nor are their traces as long as address and control lines. Control and timing fills the remaining space.

Two benefits are derived from this layout. First,

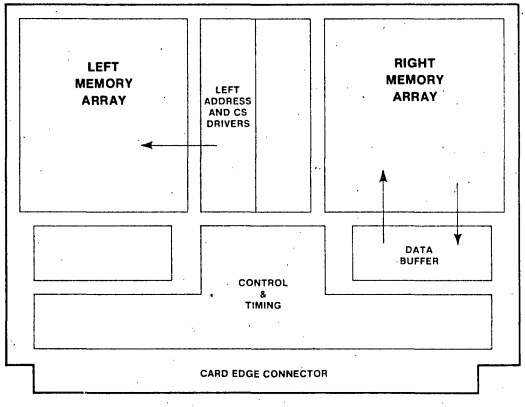


Figure 26. Board Layout

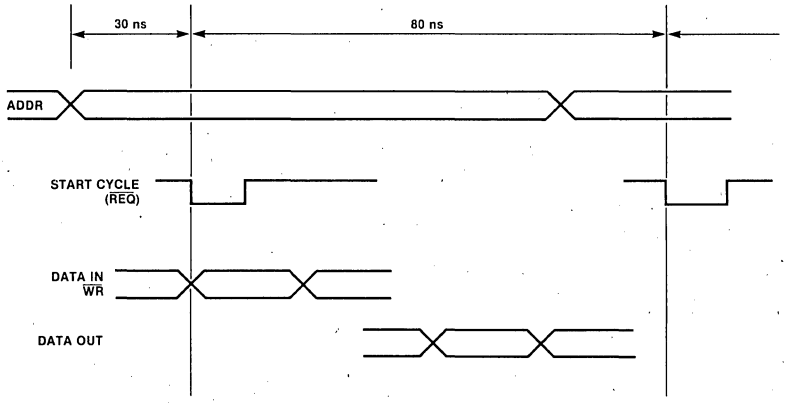


Figure 27. System Timing

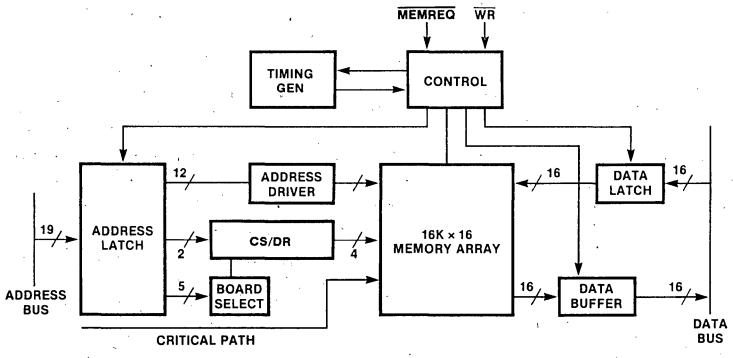


Figure 28. System Block Diagram

the address and control lines are perpendicular to the data lines which minimizes crosstalk. Second, troubleshooting is simplified. A failing row of devices indicates a defective address or control driver; whereas a failing column indicates a faulty data driver.

SYSTEM DESIGN

Using previously discussed rules and guidelines, the design of a typical high speed memory will be reviewed to illustrate these techniques. Configuration of the system is a series of identical memory cards containing 16K words of 16 bits. Timing and control logic is contained on each board. System timing requires an 80 ns cycle as shown in Figure 27. Cycle operation begins when data and control signals arrive at the board. In this design, addresses are shifted 30 ns to be valid before the start of the cycle so that address, data, and control arrive at the memory device at the same time for maximum performance. Data and control signals are coincident with the start of the cycle. Access is not yet specified because it is affected by device access and the unknown propagation delay. Access will be determined in the design.

Figure 28 illustrates the elements of the system in block diagram form. Addresses are buffered and latched at the input to the printed circuit card. Once through the latch, the addresses split to perform three functions: board selection, chip select ($\overline{CS}$) generation, and RAM addressing. Highest order addresses decode the board select, which enables all of the board logic including $\overline{CS}$.

Next higher order addresses decode $\overline{CS}$, while the lowest order addresses select the individual RAM cell. Data enters the board from the bidirectional bus through a buffer/latch, while output data returns to the bidirectional bus via buffers. Only two control signals — cycle request ($\overline{MEMREQ}$) and write ($\overline{WR}$) control the activity on the board.

Figure 29 illustrates the levels of the delay in the

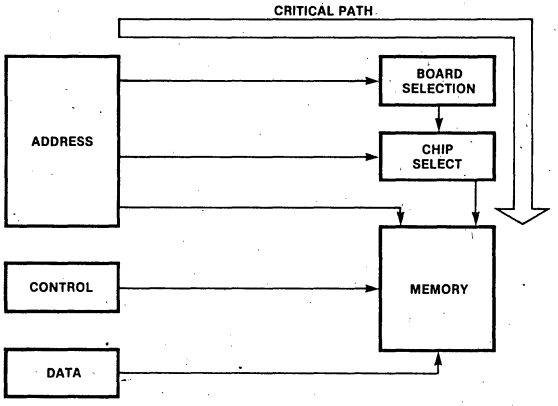


Figure 29. Worst Case Delay Path

system. Data and control have only one level. But examine the address path, it has three levels. Addresses are decoded to activate the logic on the board, select the row of RAM to be accessed and finally locate the specific memory cell. $\overline{CS}$ is in this address path and is crucial for access; without it RAM access cannot begin. But this path has the most levels of decoding with associated propagation delays. Consequently, the address path to $\overline{CS}$ is the critical path and has the greatest effect on system delay and hence must be minimized.

Examination of the system begins with the $\overline{CS}$ portion of the critical path, followed by addresses, data path, and finally timing and control.

CRITICAL PATH

Analysis of the critical path begins with the address latch. The first decision to be made is to the latch type. Latches can be divided into two types: clocked and flow-through. Clocked latches capture the data on the leading or trailing edge of the clock. Associated with the clock is data set-up or hold-time that must be included in the delay time. Accuracy of the clock affects the transit time of the signal because any skew in the clock adds to the delay time. As an example, a typical 74S173 latch has a data set-up time of 5 ns and a maximum propagation delay time from the clock of 17 ns. Total delay time is 22 ns, excluding any clock skew.

Flow-through latches have an enable rather than clock. The enable opens the address window and allows addresses to pass independent of any clock. Delay time is measured from the signal rather than a clock. The Intel® 3404 is a high speed, 6-bit latch operating in a flow-through mode with 12 ns delay. This is acceptable but a faster latch can be fashioned using a 2-to-1 line multiplexer, either a 74S157 or a 74S158. The slower of the two is the 74S157 with 7.5 ns delay. Although the 74S158 is faster with 6 ns delay, it requires an extra inverter in the feedback path as shown in Figure 30. Between the 74S157 and the 74S158 latches, the trade off is speed against board space and power. Individual designers will choose to optimize their designs.

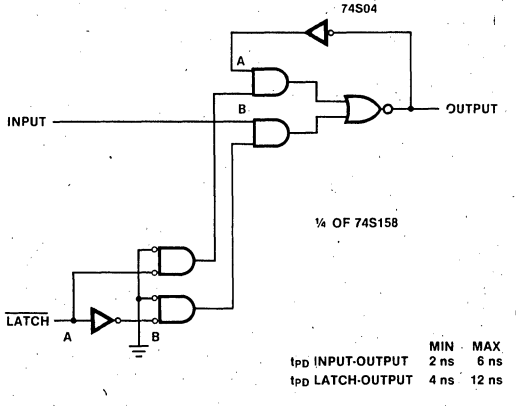


Figure 30. Fast Latch

In either case, care must be exercised in constructing the latch. Output data must be fed back to the input having the shortest internal path — the A input. If the latch is constructed with the output strapped to the B input, the input could be deselected and the feedback loop not yet selected because of the delay through the internal inverter. In this situation data would be lost. Additional delay through the external inverter (74S04) aids in preventing data loss. Inverting addresses has no system effect — except that it's faster than the non-inverting latch. During a write cycle, data will be stored at the compliment of the system address. When this data is to be retrieved, the same address will be complimented, fetching the correct word.

The remaining elements in the critical path to be designed are board selection and $\overline{CS}$ decoding. To minimize the $\overline{CS}$, decode path, the easiest method is to work backwards from $\overline{CS}$. In this manner input signals to a stage are determined and the output from the preceding stage is defined. This saves inserting an inverter at the cost of 5 ns to generate the proper input to a stage.

Starting with the $\overline{CS}$ driver, the design analyzes several approaches to select the fastest one. With four rows of devices, there are four $\overline{CS}$ signals to be generated. A 2-to-4 line decoder like the 74S138 is a possible solution. It is compact, but has two detriments: long propagation delay and insufficient drive capability. Propagation delay from enable is 11 ns. Enable is driven by board selection which arrives later than the binary inputs. Splitting the RAMs into two 4×8 arrays eases the drive requirement but the demultiplexer must still drive eight devices at 5 pF each — or 40 pF total — which adds 1.75 ns to the delay. More importantly, signal drive is required to switch cleanly and maintain levels in spite of crosstalk and reflections. A 74S240 buffer will solve this but in the process consumes an additional 9 ns.

A second and preferred approach is to use a discrete decoder to decode and drive the $\overline{CS}$ signals. Four input NAND buffers — 74S40 — fulfill this function. Addresses A_{12} and A_{13} are inverted via 74S04, providing true and compliment signals to the buffer for decoding. As shown in Figure 31, the delay is 11.5 ns. Propagation delay for the 74S40 is specified into a 50 pF load, eliminating the additional loading delay. Left and right drivers — CSXL and CSXR — are in the same package to minimize skew between left and right bytes of data. All of the decoders are enabled by Board Select to prevent rows of devices on several boards from being simultaneously active. Board Select is a true input, defining the output from the Board Select decoder.

In the Board Select decoder, the high order addresses are matched to hard-wired logic levels generated with switches for flexibility. Changing a switch setting shifts the 16K range of the board. Comparison of the switch setting and the address can be accomplished with an exclusive-OR, a 74S86. NANDing all the exclusive-OR outputs will generate a Board Select signal. Unfortunately, this signal is active-low, requiring an additional inverter as in Figure 32A, and it also consumes 22.5 ns to decode. An MSI solution to board selection is a 4-bit comparator — 74S85 — which

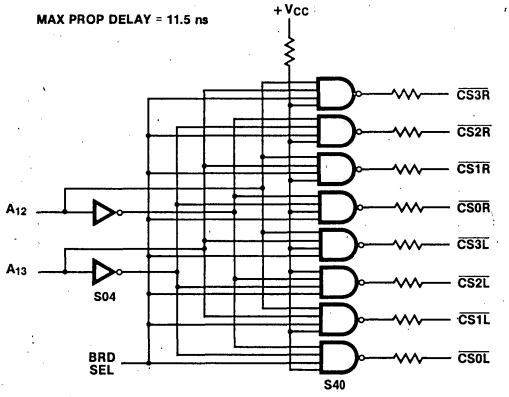


Figure 31. $\overline{CS}$ Decode

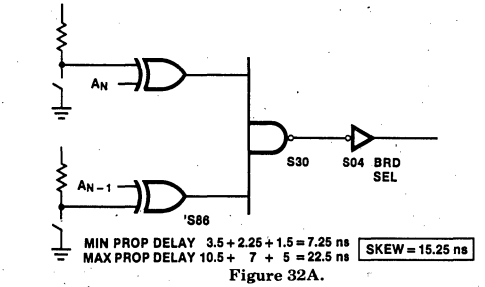


Figure 32A.

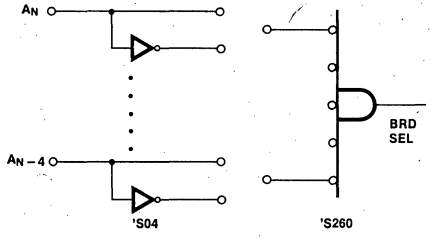


Figure 32B. Board Select

consumes less board area and propagation delay is improved at 16.5 ns.

The best solution is attained by inverting the high order addresses to generate true and compliment signals. the appropriate signal is connected into a 74S260, 5-input NOR. With an active-high output, maximum delay is 11 ns as in Figure 32B.

Critical path timing is the sum of the latch, Board Select, and $\overline{CS}$ delay times. In this example, latch delay is 6 ns, Board Select is 11 ns and $\overline{CS}$ decode is 11.5 ns for a total of 28.5 ns. One additional delay — trace delay — must be included for a complete solution. Each 74S40 drives eight MOS inputs having 5 pF/device for a load of 40 pF. Trace capacitance is calculated on 5 in. of trace. At 1.5 pF/in., trace capacitance is 7.5 pF. Trace delay calculated from equation 3 is 1.9 ns.

$$tp^1 = \frac{1.8\text{ ns}}{\text{ft}} \times \frac{5\text{ in.}}{12\text{ in./ft}} \sqrt{1 + \frac{40\text{ pF}}{7.5\text{ pF}}}$$

$$tp^1 = 1.9\text{ ns}$$

Total worst case maximum critical path delay has been calculated to be 30.4 ns (28.5 ns + 1.9 ns). With the addresses shifted in time by an amount equal to the worst case delay, device and system cycle start are coincident. Start of system access and device access differ only 0.4 ns when the addresses are shifted 30 ns. From the system cycle start, access is stretched by 0.4 ns as shown in Figure 33. Thus, with a 35 ns 2147H-1, data is valid at the output of the device 35.4 ns after the start of the cycle.

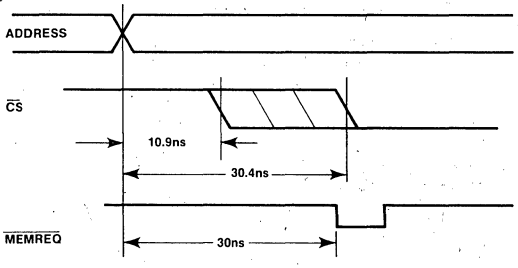


Figure 33. $\overline{CS}$ Decode Time

The minimum delay also must be calculated. With addresses valid prior to the start of the cycle, $\overline{CS}$ decoding can start in the previous cycle. If it occurs too soon, the previous cycle will not be properly completed. Minimum delay time is the sum of the minimum propagation delays plus capacitive loading delay plus trace delay. Capacitive loading delay is less than 0.4 ns and ignored. Minimum delay through the TTL is 9 ns, and added to trace delay results in a total of 10.9 ns.

From address change, the maximum delay in the critical path is 30.4 ns while the minimum is 10.9 ns. The difference between these two times is skew and will be important in later calculations.

ADDRESSES

Lower order addresses (A_0-A_{11}) arrive at the devices earlier than $\overline{CS}$ because they are not decoded. Consequently, the address drivers do not have a critical speed requirement. Once through the 6 ns latch, addresses have 24 ns to arrive at the devices.

While speed is not the primary prerequisite, drive capability is. Address drivers are located in the center of the board, dividing the array into two sections of 32 devices each. For the moment, assume one driver drives 32 devices as in Figure 34A. Each device is rated at 5 pF/input, resulting in a load of 160 pF. In addition, there are four 5-in. traces — one for each row. twenty inches of trace equates to 30 pF. Total capacitive load is 190 pF. A 74S04 is specified at 5 ns delay into 15 pF. The increased capacitive load is 175 pF, which at 0.05 ns/pF increases the delay by 8.75 ns. Under these conditions the worst cast driver relay is 5 ns plus 8.75 ns, totalling 13.75 ns. It is 10 ns earlier than the 24 ns available.

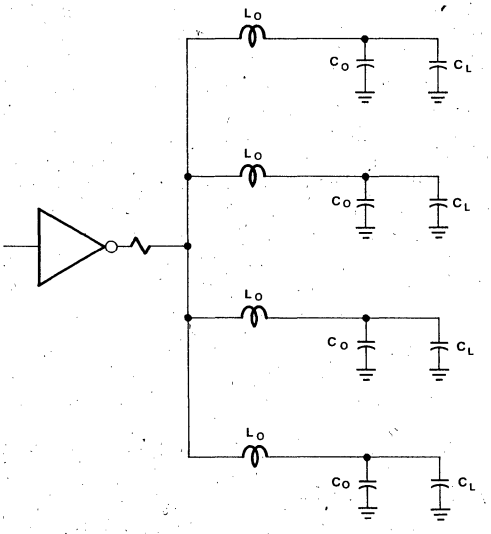


Figure 34A. Address Driver

The first impression is that this is sufficient, but the effect of crosstalk must be considered. For example, as shown in Figure 35, each trace has inductance, and parallel traces take on the

characteristics of transformers. When a signal switches from a one level to a zero level, its driver can sink 20 mA, inducing a transient in an adjacent trace. If the adjacent signal is switching to a one level, only 400 μA of a source current from the driver is available. The induced current will generate a negative spike, driving the signal at a one leval negative. Additional time of 10 to 15 ns is required to recover and re-establish a stable one level. This may prevent stable address at the start of the cycle. Recall:

$$i = C\frac{dv}{dt} \quad \text{or} \quad dt = C\frac{dv}{i}$$

where: i = instantaneous current
C = capacitance

$$\frac{dv}{dt} = \text{voltage time rate of change}$$

The term dv/dt can be maximized by increasing i or decreasing C. Current can be doubled by using a driver like a 74S240, but it draws 150mA supply current. In a large system the increased power is a disadvantage because it requires a larger power supply and additional cooling.

A better alternative is to reduce the capacitance, which results in a corresponding increase in dv/dt for quick recovery. Splitting the loads to 16 devices reduces the capacitance and allows a low power driver, like a 74S04, to be used, as in Figure 34B. This has the double effect of decreased propagation delay and providing sharp rise and fall times.

Now, there are only 10 in. of trace or 15 pF load and 16 devices, representing 80 pF for a total of 95 pF. Again, the S04 delay is 5 ns into 15 pF, but the stretched delay due to 80 pF is only 4.0 ns for a total of 9.0 ns. Stable addresses are guaranteed at the start of the cycle.

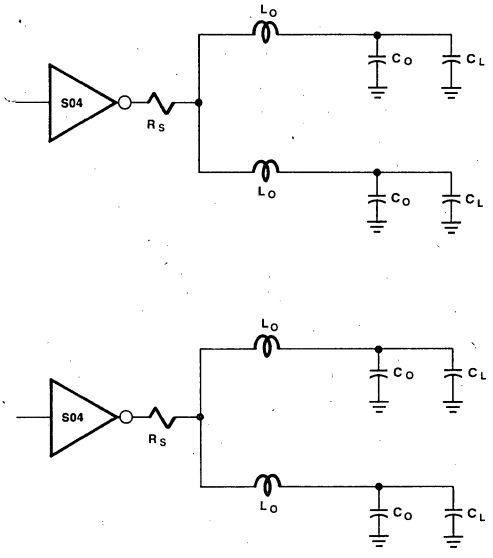


Figure 34B. Address Drivers

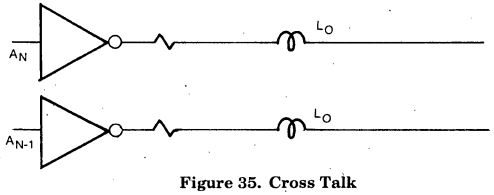


Figure 35. Cross Talk

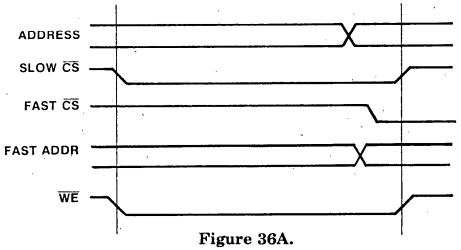


Figure 36A.

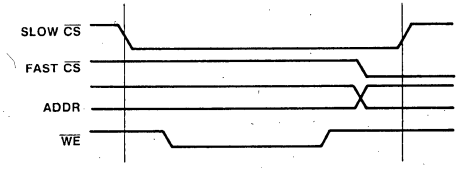


Figure 36B. Race Condition Between Address and WE

DATA PATH

Next in line for analysis is the data path. Reference to the system block diagram shows that the data is latched into the board on a write cycle and buffered out during a read cycle. Data latches are constructed from 74S158 quad two-input multiplexers. Because the data bus is bidirectional, 74S240 three-state drivers are used for output buffers.

All that remains to complete the board access computation is the calculation of the output propagation delay. Output delay of the active RAM is caused by the capacitance loading of its own output plus the three idle RAMs, the input capacitance of the 74S240 bus driver and trace capacitance. Output capacitance of the 2147Hs is 6 pF/device for a subtotal of 24 pF; input capacitance of the 74S240 is 3 pF and trace capacitance of a 5-in. trace is 7.5 pF. total load

capacitance is 34.5 pF, and access time of the 2147H is specified driving a 30 pF load. Calculated loading is close enough to the specified loading to eliminate any significant effect on the access calculations. Had there been a difference, the effect would have been included in the calculation. As previously calculated, transit time of the trace is 1.6 ns. Adding this to the 7 ns delay through the 74S240 bus driver results in an 8.6 ns output propagation delay from the RAM output to the bus.

Total access is 35.4 ns plus 8.6 ns output delay for a total access of 44 ns. The efficiency of this system is:

$$\text{Eff} = \frac{35}{44} \text{or } 80\%$$

TIMING AND CONTROL

Timing and control gating regulates activity on the board to guarantee operation in an orderly fashion. This gating latches addresses, controls the write pulse width and enables the three-state bus drivers. In addition, accurately generated timing compensates for skew effects.

In anticipation of the next cycle, the latch must be opened for the new address. When the current cycle has completed 50 ns, the latches are again opened. The next cycle might not begin 30 ns after the latch is opened because the system may skip one or more memory cycles. Therefore, a signal from the next active cycle must close the latch. In operation, a buffered Memory Request signal latches the addresses.

The write pulse is controlled to guarantee set-up and hold times for data and address and to prevent an overlap of $\overline{CS}$ and write enable from different cycles. To understand the consequences, consider the following example.

Assume two memory banks, one has a minimum $\overline{CS}$ and the other has a maximum delay path in $\overline{CS}$, and both have a minimum address delay. Assume that $\overline{WE}$ is a level generated from a write command as shown in Figure 36A. The operation under examination is a write cycle into the bank with fast $\overline{CS}$ followed by a read cycle into the bank with slow $\overline{CS}$.

Both the write cycle and the read cycle have device specification violations. In the write cycle, the addresses change prior to $\overline{CS}$ and $\overline{WE}$ becoming inactive; that new address location may be written into. In the read cycle, the address change is correct but $\overline{WE}$ is still active and the fast $\overline{CS}$ begins too soon, performing a non-existent write cycle. Clearly, controlling the width of $\overline{WE}$ will solve the problems.

Figure 36B shows the proper operation controlled with timing.

Finally, the data output buffers, controlled by timing signals, are enabled only during a read cycle while the board is selected preventing bus contention with two or more boards in the system. More importantly, timing disables the output prior to the start of the next cycle, allowing input data to be stabilized on the bidirectional data bus in preparation for a write cycle.

TIMING GENERATION

Having discussed the philosophy of timing and control, we can now focus on the specifics of address latching, write pulse generation and output-enable timing. To perform these functions timing can be generated from one of three sources: clock and shift register, monostable multivibrator, or delay line.

CLOCKED SHIFT REGISTER

A clocked shift register circuit is shown in Figure 37 consisting of a D-type flip flop and an 8-bit shift register.

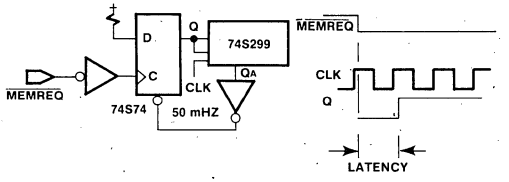


Figure 37. D Flip-Flop and Shift Register

On the leading edge of $\overline{MEMREQ}$, the Q output of the D flip flop is clocked to a one state, enabling a "one" to be propagated through the shift register. The one is clocked into the first stage of the shift register on the first clock edge after the A and B inputs are "ones". After the clock, the output Q_A goes true which subsequently clears the D flip flop, clocking zeros into the register to create a pulse one clock period wide.

The accuracy and repeatability depends primarily on the accuracy and stability of the clock. Crystal clocks can be built with +0.005% tolerance and less than a 1% variation due to temperature.

An inherent difficulty is the synchronization of Memory Request and the clock. At times there will be a latency of one clock cycle between Memory Request and the actual start of the cycle when Memory Request becomes active just after the clock edge. Assuming an 80 ns cycle and 20 ns clock, the latency can be 20 ns or 25% of a cycle stretching both access and cycle accordingly. A second difficulty of this circuit is caused by the asynchronous nature of the clock and the Memory Request. The request becomes active just prior to

the clock and the set-up time of the latch is violated, the output Q_A "hangs" in a quasi-digital state and could double or produce an invalid pulse width; this and the latency hinder effective use in high speed design.

MONOSTABLE MULTIVIBRATOR

The second possible timing generator is a series of monostable multivibrators, using a device such as the AMD Am 26S02 multivibrator. It has a maximum delay from input to output of 20 ns and an approximate minimum of 6 ns. However, with a delay of 20 ns, the monostable multivibrator offers no advantage over the clocked generator. Having a minimum pulse width of 28 ns, the one-shot offers no improvement over the 50 MHz clock, but in fact the performance is worse because it is more temperature and voltage sensitive. The pulse width is dependent on the RC network composed of resistors and capacitors that are temperature sensitive. Consequently, repeatability leaves something to be desired.

DELAY LINE

The third and best choice is a delay line. This design uses STTLDM-406 delay lines from EC[2] with tapped outputs at 5 ns increments. In operation, Memory Request activates an R-S flip flop fabricated from cross coupled NAND gates. The output of this circuit starts the memory cycle. Consequently, the cycle starts 5 ns after Memory Request compared to 20 ns for the other two timing generators. The leading edge travels down the delay lines. When the edge reaches the 25 ns tap, the output is inverted and fed back to the R input of the R-S flip flop, shaping the pulse to width to 25 ns. Twenty-five nanoseconds was chosen to match as close as possible the write pulse width. A 25 ns pulse limits the Memory Request signal width to less than 25 ns to insure proper operation. Otherwise, the R-S flip flop will not clear until Memory Request returns to a one level. As the pulse travels down the delay lines, it acquires additional skew of ±1 ns per delay line package for a total of 6 ns overall. Figure 38 shows several timing pulses and the uncertainty of each edge calculated by worst case timing analysis. The remaining problem is selection of timing edges to operate the device. Now that the timing chain is completely defined, specific details of the address latch, write pulse and output enable can be completed.

ADDRESS LATCH TIMING

An R-S flip flop activated by MEMREQ latches the addresses. A second signal which we will now calculate is used to open the latch. This signal has two boundaries. If the latch opens too late, the access of the cycle will be extended; if it opens too soon, the current cycle will be aborted. Skew through the R-S flip flop is 1.75 ns to 5.5 ns and skew in the latch from enable to output is 4 ns to 12 ns for a total skew of 6 to 17.5 ns. With this skew added to the 30 ns address set-up time, the latch opening signal must be valid at 36 ns best case or

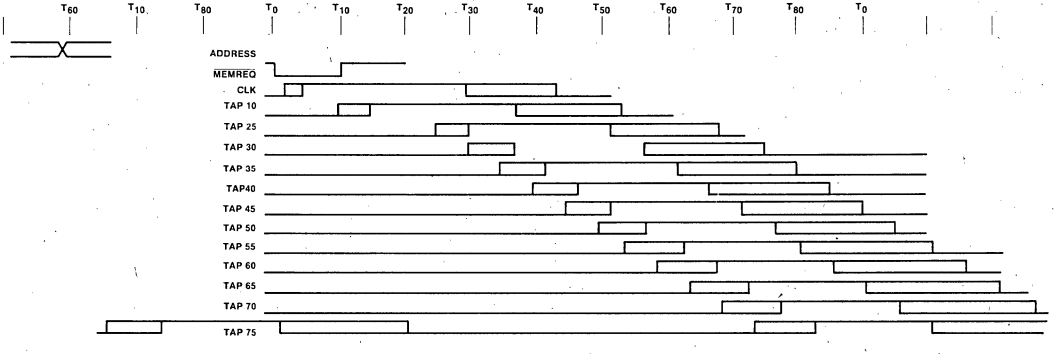


Figure 38. Timing Chain

47.5 ns worst case prior to the start of the memory cycle. Each cycle is 80 ns long, therefore, the latch opening signal must begin 44 ns or 32.5 ns, respectively, in the preceding cycle. From the delay line timing diagram, T35 will satisfy the worst case requirements for opening the latch and T 25 best case. In production, each board is tuned by selecting T25, T30, or T35 to open the latch, guaranteeing it opens between 35 and 30 ns prior to the start of the cycle.

WRITE PULSE TIMING

The next timing to be calculated is the write pulse. Figure 39 shows the three parameters which define the write pulse timing: data set-up time, write pulse width and write recovery time. Data set-up is assured by having data valid through the entire cycle.

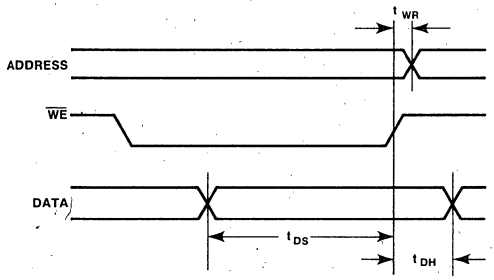


Figure 39. WE Constraints

Placement of $\overline{WE}$ in the cycle is controlled by address change to comply with t_{WR}. From previous calculations the earliest addresses can change is 50 ns, which defines the end of the $\overline{WE}$ signal. Our calculations begin at the device and work back to the timing edge. Eight devices constitute a 40 pF load and a 74S40 is specified for a 50 pF load, reducing delay by 0.5 ns when driving 40 pF. Trace delay and 74S40 delay is 3.5 to 8 ns. Subtracting 8 ns from 50 ns sets the termination of the write timing edge at 42 ns. Using the inversion of T25 will end the write pulse at 43 ns with 7 ns to spare.

Data set-up time is guaranteed because data is valid 6 ns (the worst case delay through the latch) after the start of $\overline{MEMREQ}$.

OUTPUT ENABLE TIMING

There is a 5.5 ns delay through the address driver providing minimum device cycle of 50 ns. As a result the earliest data can disappear from the bus is at 54 ns because of delay through the output circuit. To select the timing tap for the output enable, the skew of the enable circuit is subtracted from the system access time.

Subtracting the 28 ns skew of the buffer enable circuit from the 44 ns access time of the system shows that the latest the timing edge can occur is 16 ns, which is satisfied by edge T10. The trailing edge, however, ends at 37 ns and with minimum propagation delays the bus would become three-stated at 44 ns, coincident with data becoming valid. ORing T20 with T10 will guarantee the output is valid until 54 ns, minimum. Selecting a timing gap between T35 and T50, depending on the propagation delay in the enable circuit, disables the output at 70 ns, allowing input data to be valid for 10 ns prior to start of cycle. The complete schematic is shown in Figure 40.

SUMMARY

The 2147H is an easy-to-use, high speed RAM. The problems in a memory system design are the result of inherent limitations in interfacing. Largest of these is skew, which the designer must strive to minimize. In this example, skew consumed 45 ns of an 80 ns cycle while device access time was extended by only 10 ns, resulting in an 80% efficiency.

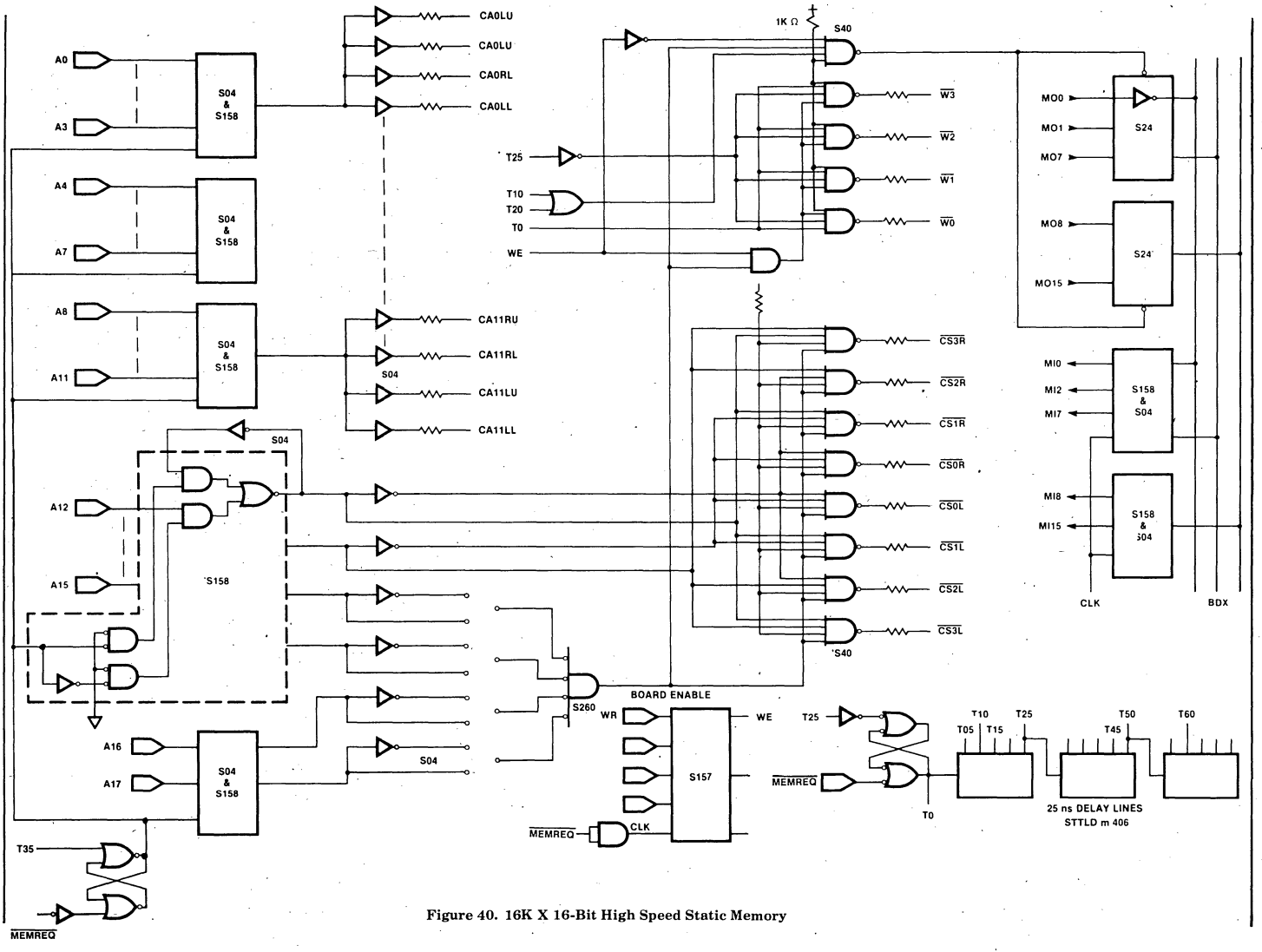


Figure 40. 16K X 16-Bit High Speed Static Memory

intel

APPLICATION NOTE

AP-131

March 1982

Intel 2164A 64K Dynamic RAM Device Description

Memory Components
Application Engineering

1. INTRODUCTION

The Intel® 2164A is a high performance, 65,536-word by 1-bit dynamic RAM, fabricated on Intel's advanced HMOS-D III technology. The 2164A also incorporates redundant elements to improve reliability and yield. Packaged in the industry standard 16-pin DIP configuration, the 2164A is designed to operate with a single +5V power supply with ±10% tolerances. Pin 1 is left as a no-connect (N/C) to allow for future system upgrade to 256K devices. The use of a single transistor cell and advanced dynamic RAM circuitry enables the 2164A to achieve high speed at low power dissipation.

The 2164A is the first commercially available dynamic RAM to be manufactured using redundant elements and also features single +5V operation, low input levels allowing −2V overshoot, a wide t_{RCD} timing window, low power dissipation, and pinout compatibility with future system upgrades. These features make the 2164A easy and desirable to use.

2. DEVICE DESCRIPTION

The 2164A is the next generation high density dynamic RAM from the 2118 +5V, 16K RAM. Pin 1 N/C provides for future system upgrade of 64K to 256K sockets. The 2164A pin configuration and logic symbols are shown in Figure 1.

Sixteen bits are required to address each of the 65,536 data bits. This is accomplished by multiplexing the 16-bit address words onto eight address input pins. The two 8-bit address words are latched into the 2164A by the two TTL level clocks: Row Address Strobe ($\overline{RAS}$) and Column Address Strobe ($\overline{CAS}$). Noncritical timing requirements allow the use of the multiplexing technique while maintaining high performance.

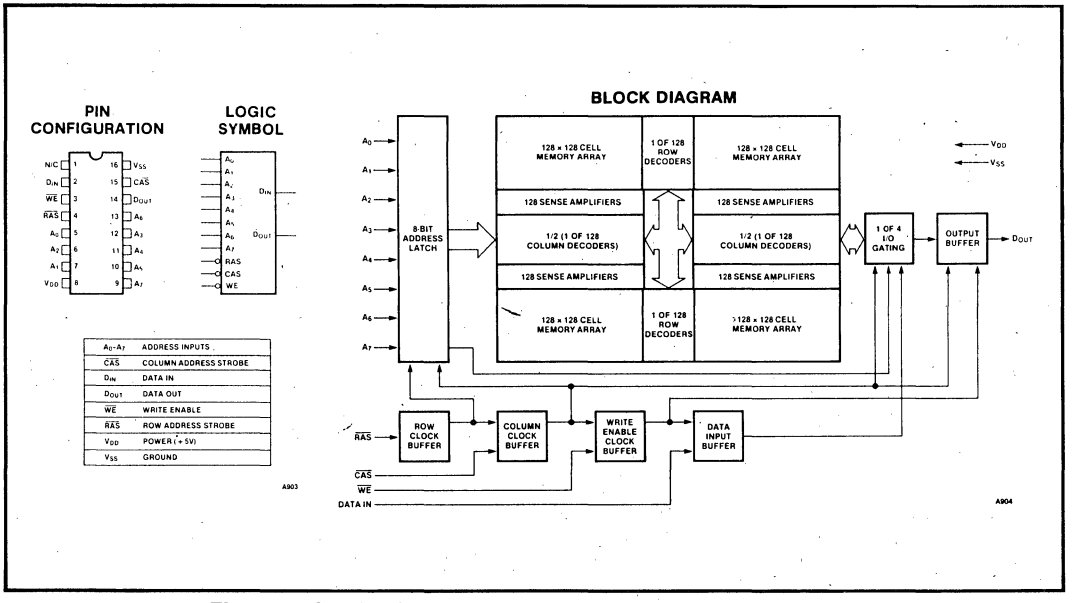
Data is stored in a single transistor dynamic storage cell. Refreshing is required for data retention and is accomplished automatically by performing a memory cycle (read, write or refresh) on the 128 combinations of RA_0 through RA_6 (row addresses) during a 2-ms period. Address input A_7 is a "don't care" during refresh cycles.

3. DEVICE OPERATION

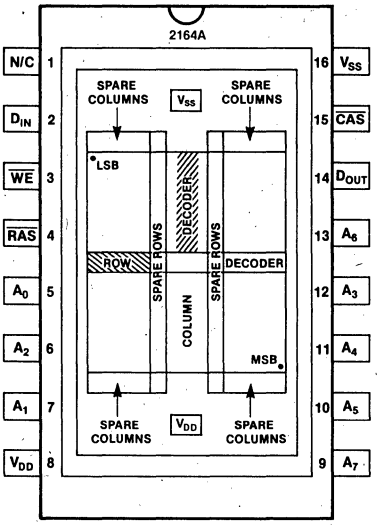
3.1 Addressing

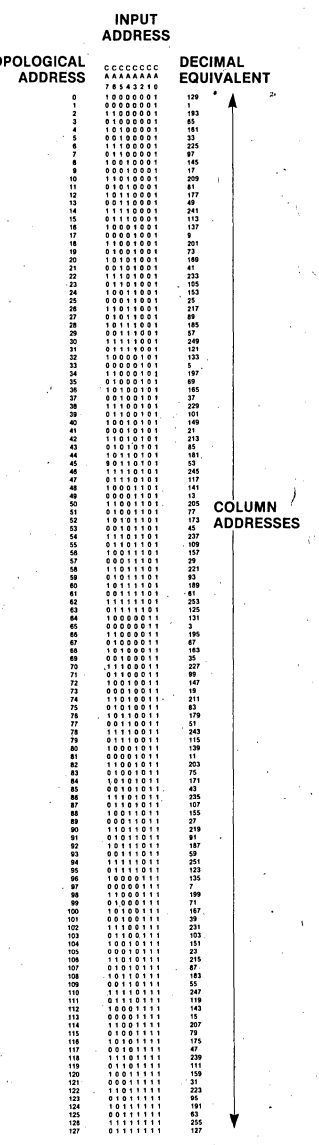
A block diagram of the 2164A is shown in Figure 2. The storage cells are divided into four 16,384-bit memory arrays. The arrays are arranged in a 128-row by 128-column matrix. Each array has 128 sense amplifiers connected to folded bit lines.

Figure 3 depicts a bit map of the 2164A and also shows the Boolean equations necessary to enable sequential addressing of the 16 required address bits (A_0–A_{15}). There is no requirement on the user to sequentially address the 2164A; the bit map and Boolean equations are shown for information only.



Figures 1 & 2. Intel 2164A Pin Assignments and Block Diagram

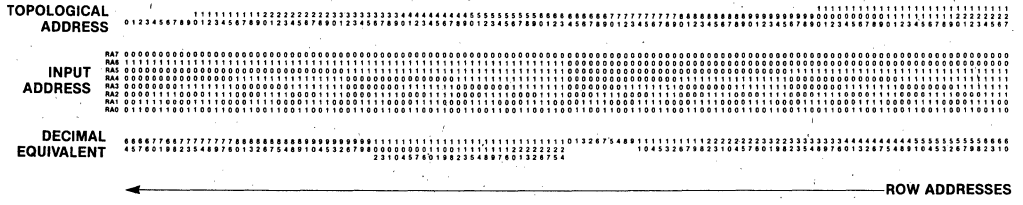




NOTE: Bit Map can be determined from Address Map equations.

2164A

ADDRESS PIN	PROGRAMMED ADDRESS
ROW	ROW ADDRESS SCRAMBLING
RA0	A0R ⊕ A1R
RA1	A1R ⊕ A2R
RA2	A2R ⊕ A7R
RA3	A3R ⊕ A7R
RA4	A4R ⊕ A7R
RA5	$\overline{A5R}$ ⊕ A7R
RA6	$\overline{A6R}$ ⊕ A7R
RA7	A7R
COLUMN	COLUMN ADDRESS SCRAMBLING
CA0	$\overline{AC7}$
CA1	AC6
CA2	AC5
CA3	AC4
CA4	AC3
CA5	AC2
CA6	$\overline{AC1}$
CA7	$\overline{AC0}$
INTERNAL DATA = AOR ⊕ A7C ⊕ D_{IN}	



C835

Figure 3. Intel® 2164A Bit Map

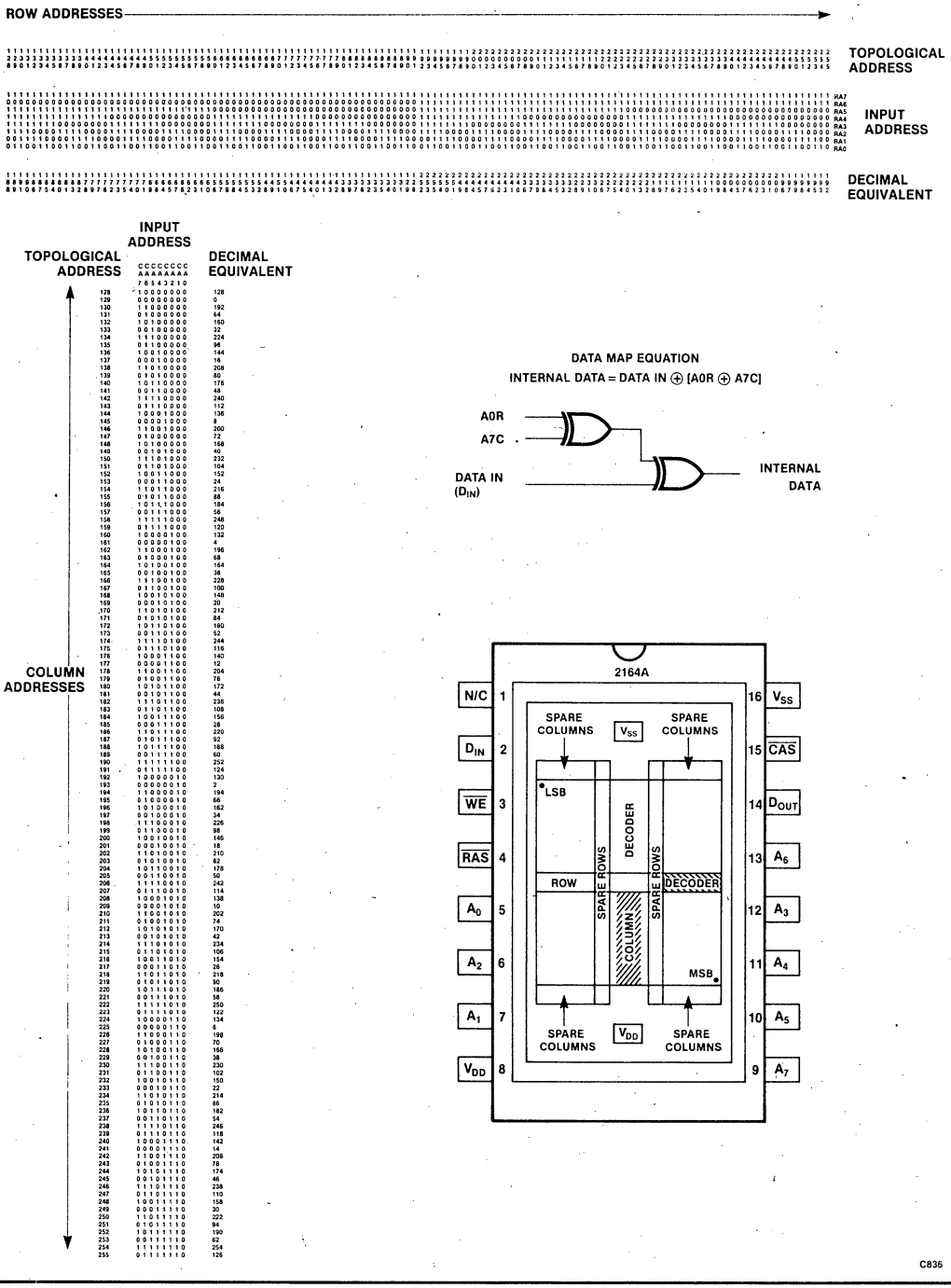


Figure 3. Intel® 2164A Bit Map (continued)

3.2 Active Cycles

When $\overline{RAS}$ is activated, 512 cells are simultaneously sensed. A sense amplifier automatically restores the data. When $\overline{CAS}$ goes active, Column Addresses CA_0–CA_6 choose one of 128 column decoders. CA_7 and RA_7 gate data sensed from the sense amplifiers onto one of the two separate differential I/O lines. One I/O pair is then gated into the Data Out buffer and valid data appears at D_{OUT}.

Because of independent $\overline{RAS}$ and $\overline{CAS}$ circuitry, successive $\overline{CAS}$ data cycles can be implemented for transferring blocks of data to and from memory at the maximum rate — without reapplying the $\overline{RAS}$ clock. This procedure is called Page Mode operation and is described in more detail in Section 4.6. If no $\overline{CAS}$ operation takes place during the active $\overline{RAS}$ cycle, a refresh-only operation occurs: $\overline{RAS}$-only refresh.

3.3 Storage Cell

The basic storage cell is shown in Figure 4. Note that the 2164A uses two dummy cells on each bit line to help compensate for alignment effects. Data is stored in single-transistor dynamic RAM cells. Each cell consists of a single transistor and a storage capacitor. A cell is accessed by the occurrence of row select ($\overline{RAS}$) clocks A_0-A_7 into the address pins, followed by column select ($\overline{CAS}$) multiplexing A_8-A_{15} into the address pins.

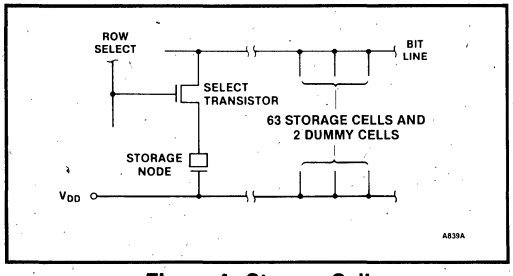


Figure 4. Storage Cell

3.4 Charge Storage in Data Cell

Data is stored in the 2164A memory cells as one of the two discrete voltage levels in the storage capacitor — a high (V_{DD}) and a low (V_{SS}). These levels are sensed by the sense amplifiers and are transmitted to the output buffer. Sensing of stored levels is destructive, so automatic restoration (rewriting or refreshing) must also occur.

The charge storage sensing mechanism for a stored low is described in Figure 5. The V_{DD} storage plate creates a potential well at the storage node. For a stored low, the charge is stored in the cell relative to the storage plate (Figure 5b). The bit sense line is precharged to V_{DD} when $\overline{RAS}$ is high (Figure 5c). During an active cycle, the row select line goes high, and the charge is redistributed (shared) with the bit sense line (Figure 5d). The sense amplifier detects the level from the cell and then reinstates full levels into the data cell via a capacitive bit line restore circuit. At the end of the active cycle, the row select line goes low, trapping the data level charge on the stored cell.

3.5 Data Sensing

The 2164A sense amplifier compares a stored level to a reference level (V_{SS}) in a special, non-addressable storage cell called a dummy cell.

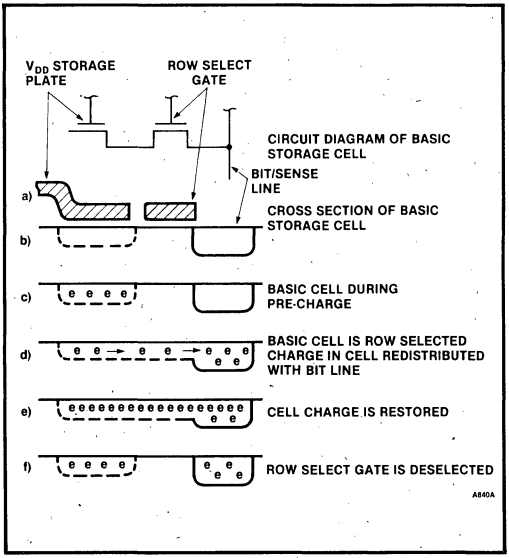


Figure 5. Sensing

Figure 6 depicts a simplified schematic of the 2164A sense amplifier. The sense amp contains a pair of cross-coupled transistors (Q1 and Q2), two isolation transistors (Q3 and Q4), and a common node which goes low with $\overline{SAS}$ (Sense Amp Strobe) and activates the sense amp. The bit-sense lines (BSL and $\overline{BSL}$) run parallel out from the sense amp in a folded bit line approach. Each bit line contains 64 data cells and two dummy cells. The double dummy cell arrangement helps limit the effect of mask alignment on sensing margins by having a dummy cell oriented in the same direction as the data cells.

The folded bit line approach has several advantages, one of which minimizes the effect of interbit line substrate noise and I/O coupling by providing common mode noise rejection. This sense amp arrangement uses metal bit lines and polysilicon word lines.

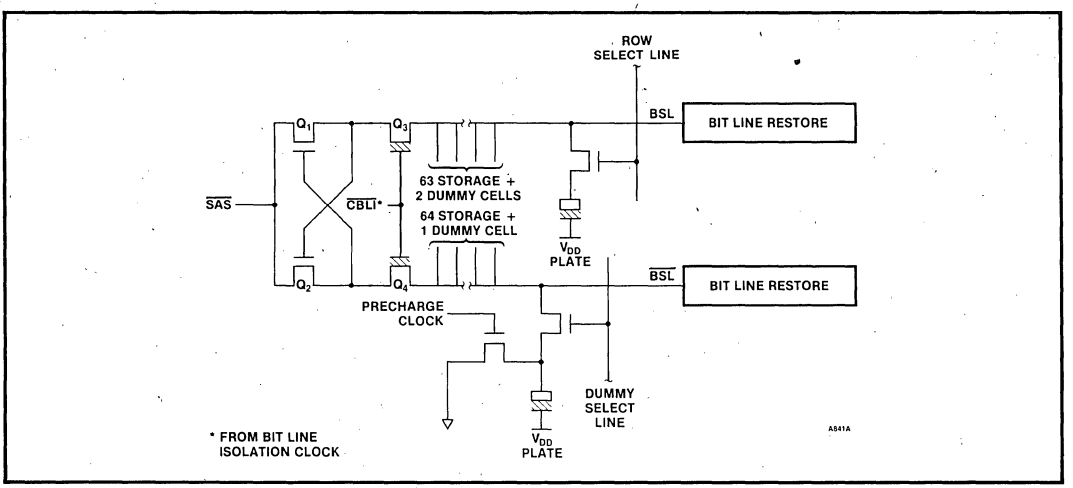


Figure 6. Sense Amp

To eliminate sensing problems, a three-step sensing (Figure 7) is employed in the generation of Sense Amp Strobe clock ($\overline{SAS}$). Device A is triggered by the sense strobe clock. This device pulls down slowly and when fed back, triggers the two gates D and E. When $\overline{SAS}$ is low enough, device B turns on, pulling the $\overline{SAS}$ line lower and at a later time, device C pulls $\overline{SAS}$ down hard. If sensing occurs too quickly, the sense amp becomes sensitive to capacitive imbalance and sensing errors might happen. This design eliminates excessively fast sensing which can occur when two sense strobe clocks are being used.

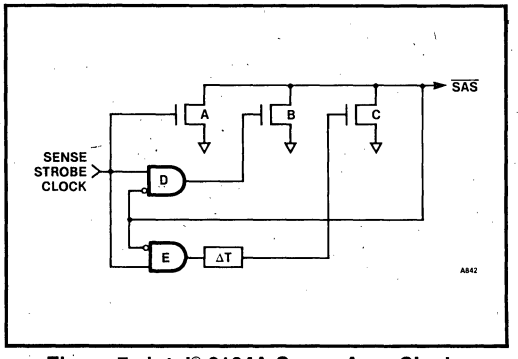


Figure 7. Intel® 2164A Sense Amp Clocks

3.6 Precharge

A precharge period is required after any active cycle to ready the memory device for the next cycle. This occurs while $\overline{RAS}$ is high. The bit lines are precharged to V_{DD}, while the dummy cells are precharged to V_{SS}. During precharge, the row select and dummy select lines are at V_{SS}, isolating the cells from the bit lines. When $\overline{RAS}$ goes low, the precharge clock goes low, ending the precharge period.

3.7 Data Sensing Operation

The row select and dummy select gating are arranged so the selected data and dummy cells are on alternate bit lines of the sense amp (Figure 6). The row select and dummy select lines go high simultaneously, resulting in concurrent charge redistribution on the bit lines. The relationship between the word select lines and the effect of concurrent charge redistribution on the bit lines is shown in Figure 8. An approximate 250 mV differential results from this charge redistribution.

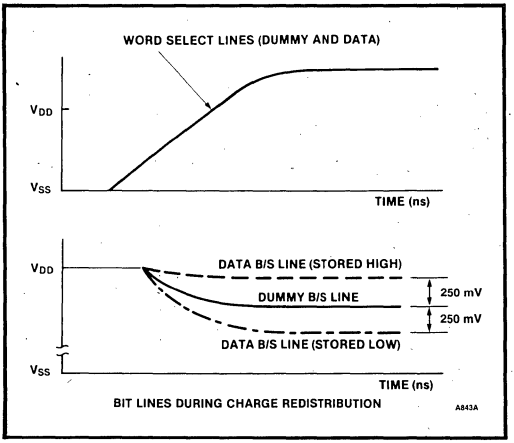


Figure 8. Sensing Voltage Waveforms

After charge redistribution, the sense amp is activated. The sense amp amplifies the differences in the resultant voltages on the bit lines. The line with the lower voltage potential is driven to V_{SS}. The other line remains at a relatively high level, as shown in Figure 9.

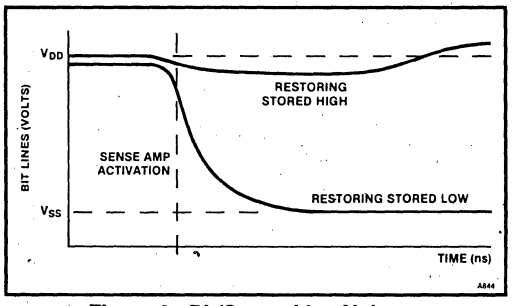


Figure 9. Bit/Sense Line Voltage

The bit line boost circuitry is shown in Figure 10. During sense operations, the boost capacitors are isolated. After sensing, the bit line with a "0" has the capacitor turned off ($V_{GS} \approx 0$) and, conversely, the bit line with a "1" has the capacitor turned on. The boost clock will turn on and boost the 1-level up above V_{DD}, giving maximum charge stored in the cell.

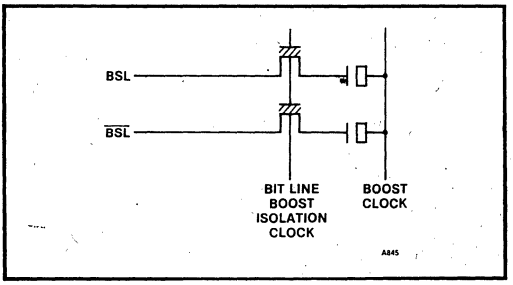


Figure 10. Bit Line Restore

3.8 Data Storage

Figure 11 shows how the I/O lines from each quadrants' sense amps are multiplexed onto the final pair of I/O lines. The I/O is a pair of opposite polarity data lines (I/O and $\overline{I/O}$) which are connected to the Data Input (D_{IN}) and Data Output (D_{OUT}) buffers. Data is differentially placed on the I/O lines during read operation and multiplexed to the final I/O lines. During a write cycle, data is differentially placed on the final I/O lines from D_{IN} and decoded onto the internal I/O lines. Stored levels are determined by CA_7 column and RA_0 row exclusive-ORed product and then exclusive-ORed again with D_{IN} (Figure 3). Stored levels are decoded during D_{OUT} operation and have no effect on device use.

3.9 Address Latches

The 8-bit row and column address words are latched into internal address buffer registers by $\overline{RAS}$ and $\overline{CAS}$. $\overline{RAS}$ strobes in the seven low-order addresses (A_0–A_7) both to select the appropriate data select and dummy select lines and to begin the timing which enables the sense amps. $\overline{CAS}$ strobes in the eight high-order addresses (A_8–A_{15}) to select one of the column decoders and enable I/O operation.

Figure 12 shows a simplified 2164A address buffer. As $\overline{\phi_1}$ goes low, the address input level is trapped via Q1 and similarly, Q2 traps V_{REF}. Since V_{REF} is about halfway between a low (0.8V) and a high (2.4V), either Q3 or Q4 will turn on harder than the other. Then as ϕ_2 becomes active, the cross-coupled latch will change states. As this happens, the load transistor (Q5 or Q6) on the lower side (V_{REF} or A_{IN}) will turn off, minimizing power. As ϕ_3 now becomes active, the address level appears internally at A_X with the complement at $\overline{A_X}$.

The combination of substrate bias and high-speed input buffers allows input overshoots of −2 volts. This is an important specification when designing high-speed switching circuitry driving highly capacitive address busses. Allowing negative overshoots on the address

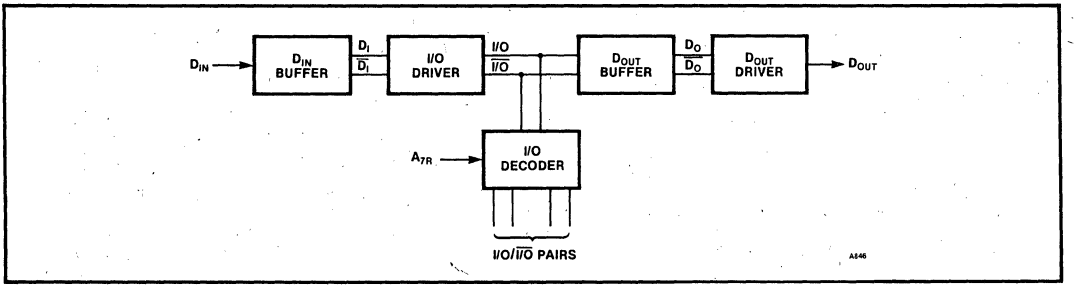


Figure 11. Data I/O

lines means minimum termination of address drivers and increased system performance. This is because a terminated signal (Figure 13) has a slower transition and hence a delay in access time. It is important to note the two advantages to this type of address buffer; first, increased operating speed, and second, a more generous timing window in the multiplexing of the address words.

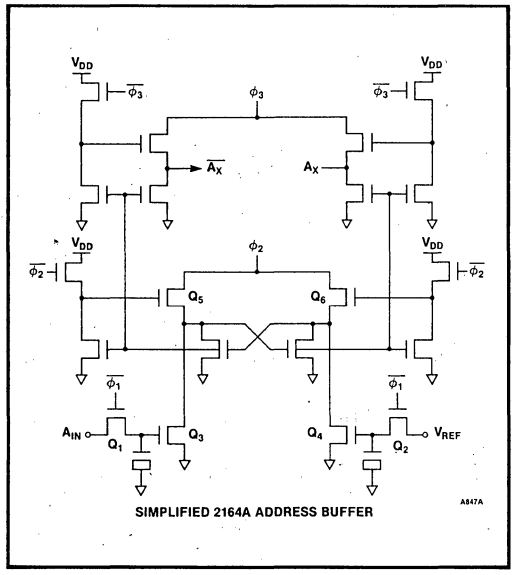


Figure 12. 2164A Simplified Address Buffer Circuitry

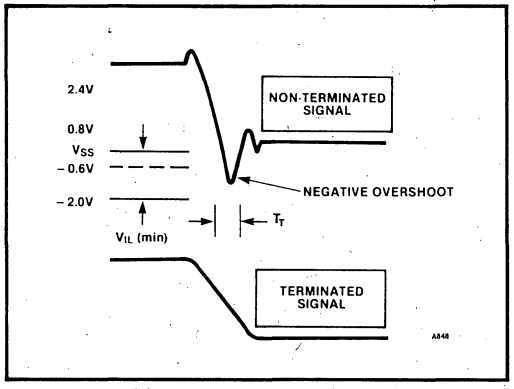


Figure 13. TTL Overshoot

3.10 Data Output Buffer

As shown in Figure 14, the output buffer has a push-pull transistor configuration in which no dc power is dissipated when active.

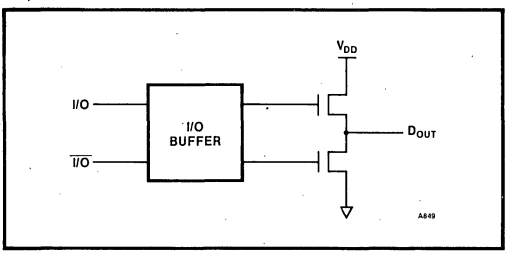


Figure 14. Simplified Output Buffer Circuit

3.11 Data Input/Output Operations

The 2164A contains a Data Input latch which is controlled by the logical NAND function of $\overline{RAS}$, $\overline{CAS}$, and Write Enable ($\overline{WE}$) during the active states (Figure 2). During an early write cycle, where $\overline{WE}$ goes low before $\overline{CAS}$ goes low, the falling edge of $\overline{CAS}$ operates the latch. In a late write (or Read-Modify-Write) cycle, where $\overline{WE}$ goes low after $\overline{CAS}$, the input is latched by the falling edge of $\overline{WE}$.

The 2164A D_{OUT} has three-state capability controlled by $\overline{CAS}$. When $\overline{CAS}$ is at V_{IH}, the output is in a High Impedance (Hi-Z) state. The D_{OUT} states for various operating modes are shown in Table 1. For a Read or Read- Modify-Write cycle, D_{OUT} will remain in the Hi-Z state until the data is valid, whereupon it will go to V_{OH} or V_{OL}, depending on the data.

Table 1. Intel® 2164A Data Output Operation for Various Types of Cycles

Type of Cycle	Data Output State
Read Cycle	Data from Addressed Memory Cell
Early Write Cycle	Hi-Z
$\overline{RAS}$-Only Refresh Cycle	Hi-Z
$\overline{CAS}$-Only Cycle	Hi-Z
Read/Modify/Write Cycle	Data from Addressed Memory Cell
Delayed Write Cycle	Indeterminate
Hidden Refresh Cycle	Data from Addressed Memory Cell
Page Mode Read Cycle (Entry or Internal Cycle)*	Data from Addressed Memory Cell
Page Mode Write Cycle (Entry or Internal Cycle)*	Hi-Z
Page Mode R/M/W Cycle (Entry or Internal Cycle)*	Data from Addressed Memory Cell

* The entry cycle is the first cycle of the page and the internal cycles are the subsequent cycles of the page operation.

For an "Early" Write cycle, D_{OUT} remains in the Hi-Z state which allows "wire-OR" for D_{IN} and D_{OUT}. D_{OUT} is indeterminate for the period between an "Early" Write ($t_{WCS} \geq 0$) and a Read-Modify-Write cycle ($t_{RWD} > t_{RWD}$ min and $t_{CWD} > t_{CWD}$ min). A $\overline{RAS}$-only refresh cycle or a $\overline{CAS}$-only cycle will have no effect on D_{OUT} which will remain in the Hi-Z state. D_{OUT} remains valid from access time until $\overline{CAS}$ goes high. Holding $\overline{CAS}$ low and taking $\overline{RAS}$ high will not affect the state of the D_{OUT}. The D_{OUT} remains valid following a valid Read cycle regardless of the number of subsequent $\overline{RAS}$-only cycles performed on the device up to the t_{CAS} max limit. These secondary $\overline{RAS}$ cycles are $\overline{RAS}$-only refresh cycles to the 2164A.

3.12 Power-On

An initial pause of 500 μs is required after the application of the V_{DD} supply, followed by a minimum of eight (8) initialization cycles (any combination of cycles containing a $\overline{RAS}$ clock such as $\overline{RAS}$-only refresh) prior to normal operation. Eight initialization cycles are required after extended periods of bias (greater than 2 ms) without clocks. The V_{DD} current (I_{DD}) requirement of the 2164A during power on is, however, dependent upon the input levels of $\overline{RAS}$ and $\overline{CAS}$ and the rise time of V_{DD} as shown in Figure 15.

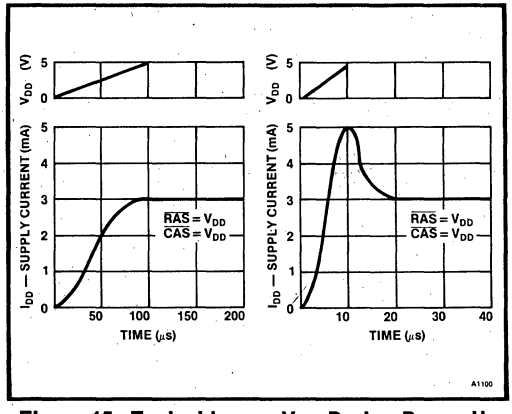


Figure 15. Typical I_{DD} vs. V_{DD} During Power Up

If $\overline{RAS} = V_{SS}$ during power on, the device may go into an active cycle and I_{DD} would show spikes similar to those shown for the $\overline{RAS}/\overline{CAS}$ timings. It is recommended that $\overline{RAS}$ and $\overline{CAS}$ track with V_{DD} during power on or held at a valid V_{IH}.

4. DATA CYCLES/TIMING

A memory cycle begins with a negative transition of $\overline{RAS}$. Both the $\overline{RAS}$ and $\overline{CAS}$ clocks are TTL compatible. The 2164A input buffers convert the TTL level signals to MOS levels inside the device.

$\overline{RAS}$ and $\overline{CAS}$ have minimum pulse widths as specified in the 2164A Data Sheet. These minimum pulse widths and cycle times must be maintained for proper device operation and data integrity. A cycle, once begun, must be within specification.

Figure 16 briefly summarizes the various active cycles which are discussed in paragraphs 4.1 through 4.6.

4.1 Read Cycle

A Read cycle is performed by maintaining $\overline{WE}$ high during a $\overline{RAS}/\overline{CAS}$ operation. The output pin of a selected device remains in a high impedance state until valid data appears at the output within the specified access time.

Device access time, t_{ACC}, is the longer of two calculated intervals:

Eq. (1) $t_{ACC} = t_{RAC}$ or

Eq. (2) $t_{ACC} = t_{RCD} + t_{CAC}$

Access time from $\overline{RAS}$ (t_{RAC}), and access time from $\overline{CAS}$ (t_{CAC}), are device parameters. Row to column address strobe delay time, t_{RCD}, is a system-dependent timing parameter. For example, substituting the device parameters of the 2164A-20 yields:

Eq. (3) $t_{ACC} = t_{RAC} = 200$ ns for 35 ns $\leq t_{RCD} \leq 80$ ns

Eq. (4) $t_{ACC} = t_{RCD} + t_{CAC} = t_{RCD} + 120$ ns for $t_{RCD} > 80$ ns

Note that if 35 ns $\leq t_{RCD} \leq 80$ ns, device access time is determined by equation 3 and is equal to t_{RAC}. If $t_{RCD} > 80$ ns, access time is determined by equation 4. This 45 ns interval (shown in the t_{RCD} inequality in equation 3), in which the falling edge of $\overline{CAS}$ can occur without affecting access time, allows for system timing skew in the generation of $\overline{CAS}$. This allowance for t_{RCD} skew is designed in at the device level to allow the fastest access times to be utilized in practical system designs.

4.2 Write Cycles

4.2.1 EARLY WRITE CYCLE

An early write cycle is performed by bringing $\overline{WE}$ low before $\overline{CAS}$. D_{IN} is written into the selected bit. D_{OUT} remains in the Hi-Z state.

4.2.2 LATE WRITE CYCLE

A late write cycle happens after $\overline{RAS}$ and $\overline{CAS}$ go low. During a late write cycle, t_{RWD} and t_{CWD} ($\overline{RAS}$ and $\overline{CAS}$ delays to Write Enable) minimum timings are not met. Since there is no guarantee that D_{OUT} will remain in a Hi-Z state, the condition of D_{OUT} is indeterminate.

4.3 Read-Modify-Write Cycle (Delayed Write)

A Read-Modify-Write (R-M-W) cycle is performed by bringing $\overline{WE}$ low after $\overline{RAS}$ and $\overline{CAS}$ are low. Here, t_{RWD} and t_{CWD} minimum timings are satisfied. D_{OUT} has had time to become valid and is now latched by $\overline{CAS}$ remaining low. As $\overline{WE}$ goes low, a write begins, transferring the data from D_{IN} to the cell as D_{OUT} remains active with the previous data.

In any type of Write cycle, D_{IN} must be valid at or before the falling edge of $\overline{WE}$ or $\overline{CAS}$, whichever is latest.

4.4 $\overline{CAS}$-Only Cycle

A $\overline{CAS}$-only cycle has no effect on the 2164A. The 2164A remains in the lowest power, standby condition.

4.5 Refresh Cycle

A cycle at each of 128 row addresses will refresh all storage cells. Any memory cycle — Read, Write (Early Write, Delayed Write, R-M-W) or $\overline{RAS}$-only — refreshes the bits selected by the row address combinations of A_0 through A_6. Both 32K halves are refreshed, as the state of A_7 is irrelevant during refresh.

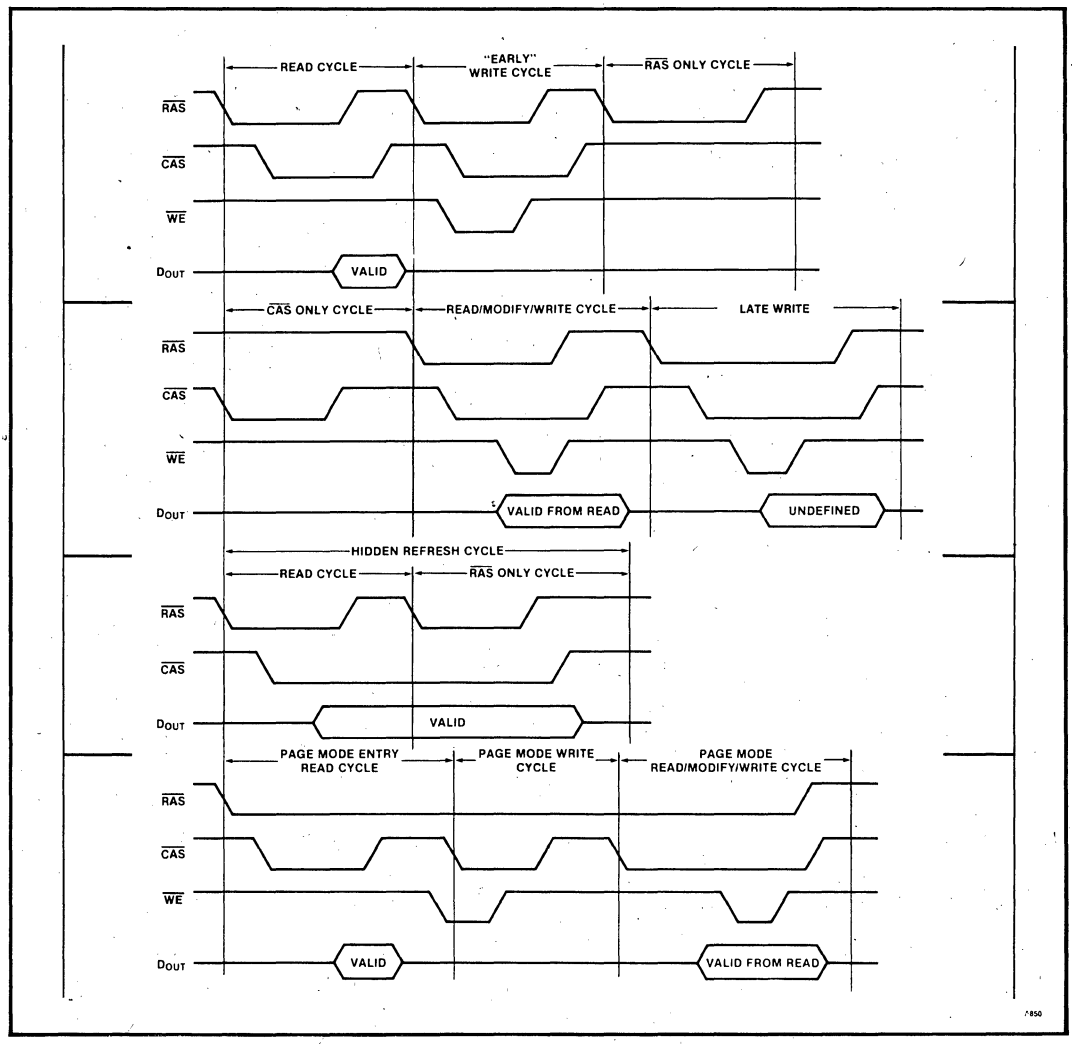


Figure 16. Intel® 2164A Operation of Data Output for Various Active Cycles

4.5.1 READ CYCLE REFRESH

Since A_7 is irrelevant for refresh addressing, a row refreshes 512 cells. The 256 cells in a specific row addressed (A_0–A_6, A_7) are refreshed as are another 256 cells in the row A_0–A_6, $\overline{A_7}$. Therefore, addressing a bit in a row refreshes the 256 cells associated with that row (A_0–A_7). For refresh purposes, row A_0–A_6 and $\overline{A_7}$ is also addressed as another 256 cells. Therefore, successive reads of the 128 row combinations of A_0–A_6 refreshes the entire array of the 2164A.

This refresh mode is useful only when the memory system consists of a single row of devices. When used with more than one row of devices, output bus contention will result.

4.5.2 WRITE CYCLE REFRESH

A Write cycle will perform a refresh. However, the selected cell will be modified to D_{IN}. This may cause a change of state of selected cell, while the other 511 cells are refreshed.

For an Early Write refresh cycle, there will be no output bus contention since the output remains in the Hi-Z state. Bus contention will result for Delayed Write or R-M-W refresh cycles involving more than one row of devices.

4.5.3 $\overline{\text{RAS}}$-ONLY REFRESH

A cycle with $\overline{\text{RAS}}$ active refreshes the 2164A. This is the recommended refresh mode, especially when the memory system consists of multiple rows of memory devices. The D_{OUT}'s may be wired-ORed with no bus contention when $\overline{\text{RAS}}$-only refresh cycles are performed on all rows of devices concurrently. The 2164A D_{OUT} will remain in three-state.

4.5.4 HIDDEN RAS-ONLY REFRESH

The 2164A is designed for "hidden" refresh operation. Hidden refresh accomplishes a refresh cycle following a read cycle without disturbing the D_{OUT}. Once valid, D_{OUT} is controlled solely by $\overline{\text{CAS}}$. After a Read cycle, $\overline{\text{CAS}}$ is held low while $\overline{\text{RAS}}$ goes high for precharge. A $\overline{\text{RAS}}$-only cycle is then performed and D_{OUT} remains valid. However, for operation in this mode, $\overline{\text{CAS}}$ must be decoded along with $\overline{\text{RAS}}$ for the Read and Write cycles. $\overline{\text{CAS}}$ cannot be driven as a common clock to the entire array since it would cause devices being only refreshed to interpret this operation as a $\overline{\text{RAS}}/\overline{\text{CAS}}$ cycle.

4.6 Page Mode Operation

Page Mode operation allows additional columns of the selected device to be accessed at a common row address set. This is done by maintaining $\overline{\text{RAS}}$ low while successive $\overline{\text{CAS}}$ cycles are performed.

Page Mode operation allows a maximum data transfer rate as $\overline{\text{RAS}}$ addresses are maintained internally and do not have to be reapplied. During this operation, Read, Write and R-M-W cycles are possible. Following the entry cycle into Page Mode operation, access is t_{CAC} dependent. The Page Mode cycle is dependent upon $\overline{\text{CAS}}$ pulse width (t_{CAS}) and the $\overline{\text{CAS}}$ precharge period (t_{CPN}).

5. SYSTEM DESIGN CONSIDERATIONS

Calculating total 2164A power consumption is a simple task. To illustrate the method of calculating power, an example system organized as 256K words by 16 bits is assumed.

The first step is to compute the total 2164A current by summing the three individual V_{DD} 2164A supply currents: (1) operating current (I_{DD0}), (2) standby current (I_{DDS}), and (3) refresh current (I_{DDR}). The total 2164A power consumption equals the 2164A current multiplied by the maximum supply voltage (V_{DD}). Total system power consumption is determined by adding the support circuitry power requirements to the total 2164A power.

Examples of these calculations, along with a power/bit determination, are presented in following sections.

5.1 Power Calculations

5.1.1 OPERATING CURRENT (I_{DD0})

Active operating current is determined by the following equation:

Eq. (1) $I_{DD0} = (I_{DD2} + I_{DDLO})K$

Where: I_{DD0} = the operating V_{DD} supply current.

K = the number of active devices (selected at one time by both $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$).

I_{DDLO} = the 2164A output load current (output leakage current plus the load devices input current). For example, if four devices are dot ORed on the output line, the output leakage current is the sum of the input current (I_{IN}) for the load plus the three leakage currents (I_{LO}) for the three devices standby.

5.1.2 STANDBY CURRENT (I_{DDS})

Standby current is determined by the following equation:

Eq. (2) $I_{DDS} = I_{DD1} \times M$

Where: I_{DD1} = the V_{DD} supply current.

M = the number of inactive devices (not selected by $\overline{RAS}$; receiving $\overline{CAS}$-only cycles).

5.1.3 REFRESH CURRENT (I_{DDR})

Refresh current is determined by the following equation:

Eq. (3) $I_{DDR} = (I_{DD3} \times N)\ (t_{RC}/t_{REF})\ (128)$

Where: I_{DD3} = the V_{DD} supply current, $\overline{RAS}$-only cycle.

N = the total number of devices in the system.

t_{RC} = the refresh cycle time.

t_{REF} = the time between refresh cycles.

Since I_{DD3} is not a full-time current, the fraction t_{RC} over t_{REF} represents the duty cycle for one address. There are 128 row addresses active in generating refresh, so the duty cycle is multiplied by 128.

Cycle time has a downward scaling effect on the average operating current according to the following equation:

Eq. (5)

$$I_{DD_{AVE}} = \left[I_{DD2} \times \left(\frac{t_{RC}\ (spec)}{t_{RC}\ (operating)} \right) \right] + \left[I_{DD1} \times \left(1 - \left(\frac{t_{RC}\ (spec)}{t_{RC}\ (operating)} \right) \right) \right]$$

At minimum cycle time, $\dfrac{t_{RC}\ (spec)}{t_{RC}\ (operating)} = 1$,

so that worst case $I_{DD_{AVE}} = I_{DD2}$, but as the cycle time increases, $I_{DD_{AVE}}$ approaches the standby current, becoming 6.3 mA @ 10,000 ns cycle time. Figure 5 in the 2164A data sheet depicts this scaling effect.

5.1.4 TOTAL 2164A POWER

Total 2164A power equals the sum of the three currents multiplied by the worst case supply voltage. This is expressed by the following equation:

Eq. (4) $\text{Power} = (I_{DD0} + I_{DDS} + I_{DDR})\ V_{DD}(\text{max})$

5.1.5 EXAMPLE POWER CALCULATIONS

Assume that we have a 256K word by 16-bit memory system using the 2164A-20 at minimum cycle time. Thus, the following parameters apply:

N = 64 devices in system

K = 16 devices active at one time

M = N-K devices in standby

= 64-16

= 48

Referring to the Intel 2164A Data Sheet[1] and the Intel 8282 Data Sheet[2], we obtain the following values:

I_{DD1} = 5 mA, 2164A-20

I_{DD2} = 45 mA, 2164A-20, t_{RC} = 330 ms

I_{DD3} = 40 mA, 2164A-20, t_{REF} = 2 ms

I_{LO} = 10 μA, 2164A-20

I_{IN} = 200 μA, 8282

To calculate I_{DD0}:

Eq. (1) $I_{DD0} = (I_{DD2} + I_{DDLO})K$

$= (45\ \text{mA} + [3(10\ \mu\text{A}) + 200\ \mu\text{A}])16$

$= 723.68\ \text{mA}$

To calculate I_{DDS}:

Eq. (2) $I_{DDS} = (I_{DD1})M$

$= (5\ \text{mA})48$

$= 240\ \text{mA}$

To calculate I_{DDR}:

Eq. (3) $I_{DDR} = (I_{DD3} \times N)(t_{RC}/t_{REF})(128)$

$= (40\ \text{mA} \times 64)\ \dfrac{330\ \text{ns}}{2\ \text{ms}}\ (128)$

$= (2560\ \text{mA})(.021)$

$= 53.76\ \text{mA}$

To calculate total power:

Eq. (4) $\text{Power} = (I_{DD0} + I_{DDS} + I_{DDR})\ V_{DD}(\text{max})$

$= 5.5\text{V}\ (723.7\ \text{mA} + 240\ \text{mA} + 53.8\ \text{mA})$

$= 5.59\ \text{watts}$

The power/bit is equal to:

Power/Bit = (Total 2164A Power/Number of Devices) (Bits per Device)

$= 5.59(64 \times 65{,}536)$

$= 1.33\ \mu\text{watts/bit}$

5.2 Board Layout

An important consideration in system design is the circuit board layout. A proper layout results in minimum board area while yielding wider power supply and tim-

ing operating margins for increased reliability and easier manufacturability. The key areas of consideration are:

1. Ground (V_{SS}) and power (V_{DD}) gridding
2. Power and ground planes
3. Memory array/control line routing
4. Control logic centralization
5. Power supply decoupling

5.2.1 GROUND AND POWER GRIDDING

Ground and power gridding can contribute to excess noise and voltage drops if not properly structured. An example of an unacceptable method is presented in Figure 17. This type of layout results in accumulated transient noise and voltage drops for the device located at the end of each trace (path).

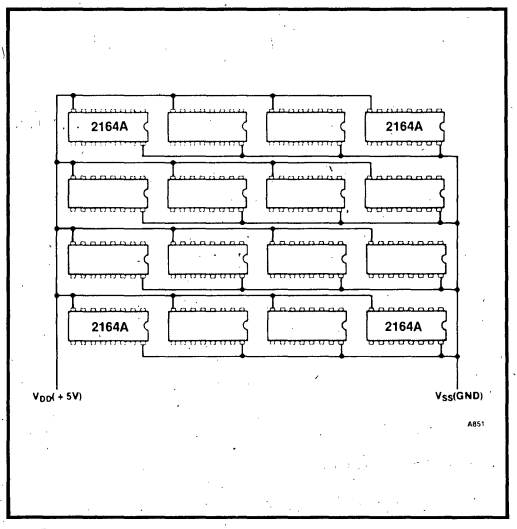


Figure 17. Unacceptable Power Distribution

Transient effects can be minimized by adding extra circuit board traces in parallel to reduce interconnection inductance (Figure 18).

5.2.2 POWER AND GROUND PLANE

A better alternative to power and ground gridding is power and ground planes. Although this requires two additional inner layers to the PC board, noise and supply voltage fluctuations are greatly reduced. If power and ground planes are used, gridding is optional but typically used for increased reliability of power and ground connections and further reduction of electromagnetic noise.

It is preferable on power/ground planes to use circular voids for device pins rather than slotted voids (Figure 19). This provides maximum decoupling and minimum crosstalk between signal traces.

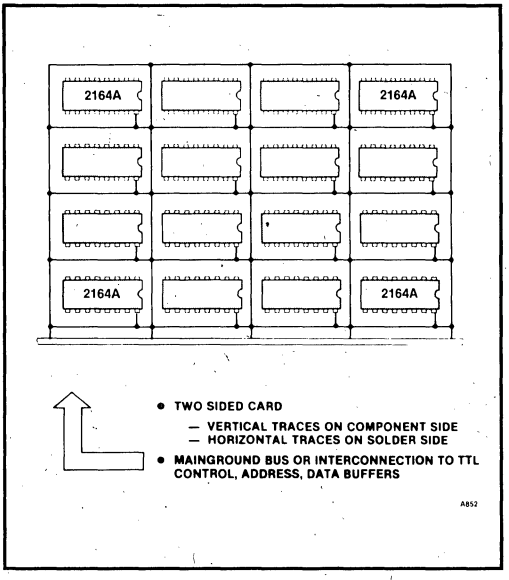


Figure 18. Recommended Power Distribution — Gridding

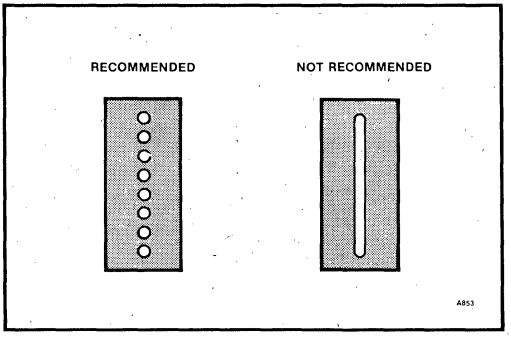


Figure 19. Recommended Voids for Multilayer PC Boards

5.2.3 MEMORY ARRAY/CONTROL LINE ROUTING

Address lines should be kept as short and direct as possible. The lone serpentine line shown in Figure 20 is to be avoided since the devices furthest away from the driver will receive a valid address at a later time than the closer ones. A better way to route address lines is in a comb-like fashion from a central location as shown in Figure 21. Routing control and address signals together from a centralized board area will also minimize skew.

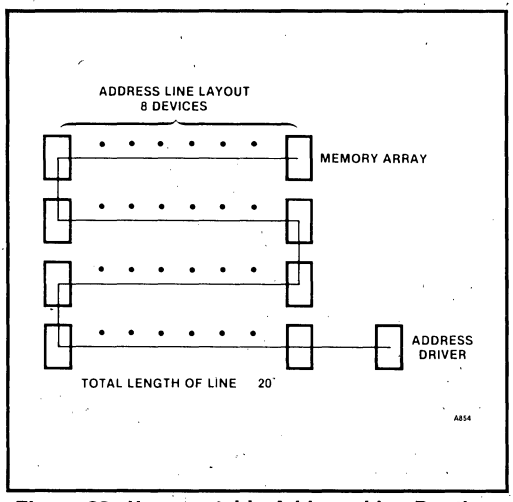


Figure 20. Unacceptable Address Line Routing (Serpentine)

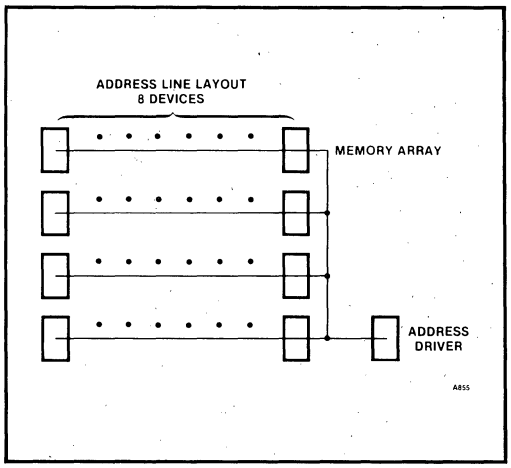


Figure 21. Recommended Address Line Routing

5.2.4 CONTROL LOGIC CENTRALIZATION

Memory control logic should be strategically located in a centralized board position to reduce trace lengths to the memory array. Long trace lines are prone to ringing and capacitive coupling which can cause false triggering of timing circuits. Short lines minimize this condition and also result in less system skew.

A practical memory array layout is shown in Figure 22. Typically, this pattern and its "mirror image" are placed on each side of the memory control logic for a practical memory board design.

5.2.5 POWER SUPPLY DECOUPLING

For best results, decoupling capacitors are placed on the memory array board at each memory location (Figure 22). High frequency 0.1 μF ceramic capacitors are the recommended type, especially for four or more rows of devices. In this arrangement, noise is minimized because of the low impedance across the circuit board traces. Typical V_{DD} noise levels for this arrangement are less than 300 mV.

A large tantalum capacitor (typically one 100 μF per 64 devices) is required at the circuit board edge connector power input pins to recharge the 0.1 μF capacitors between memory cycles.

To calculate decoupling requirements, one considers the current switching of devices from standby to active currents. This involves $I_A = I_{DD2} - I_{DD1}$ (active cycle) and $I_R = I_{DD3} - I_{DD1}$ (refresh cycle). One can then assume some t_B bulk decoupling response time with only one refresh during t_B and minimum cycle time t_C. As a further example, assume only 1/4 of the devices are active at any one time. The amount of charge (Q) requiring decoupling is:

$$Q = I_R t_C + \frac{1}{4} I_A (t_B - t_C).$$

This charge can then be used to calculate the appropriate decoupling capacitance per device. Using Coulomb's law, $Q = CV$, and knowing Q, one picks an acceptable ΔV (< 400 mV) for noise on the V_{DD} lines. The capacitance required is given by $C = Q/\Delta V$. It is important to recognize that C is determined by the current changes in the devices. Minimum cycle time is used for calculating purposes. Lengthening the cycle time will not affect decoupling.

6. THERMAL CHARACTERISTICS

Thermal Characteristics are useful when designing for thermal systems, or for any application where the temperature may go to extremes.

The operating ambient temperature ranges for the 2164A are guaranteed with transverse airflow that exceeds 200 linear feet per minute.

Typical thermal resistance values of the cerdip package at maximum temperature are:

θ_{jA} (@200 fpm air flow) = 47 °C/W
θ_{jc} (still air) = 22 °C/W

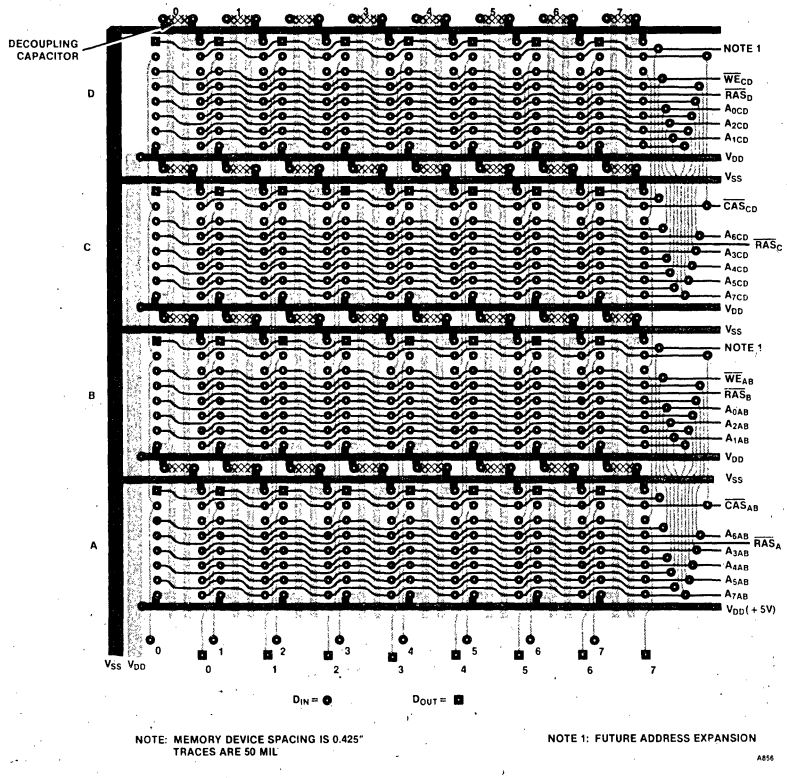


Figure 22. 2164A Memory Array PC Board Layout

7. DESCRIPTION OF REDUNDANT CIRCUITS

The Intel 2164A is the first commercially produced RAM to incorporate redundant elements into the design. Redundancy allows bit-efficient use of silicon by maximizing bits/wafer start. By overstressing and eliminating weak oxide at sort, prior to fusing in redundant elements, long term oxide failures can be greatly reduced. Redundancy makes possible the use of larger die sizes allowing better use of existing fab equipment, and a more conservative layout to utilize larger cell (storage) areas.

In choosing how redundant elements should be organized, single bits, blocks of bits and spare rows and columns were examined. For maximum efficiency, four spare rows and four spare columns were chosen for the 2164A.

The address of a faulty element is programmed into the spare element by electrically opening polysilicon fuses during wafer probe. The basic circuit block diagram for a spare row is shown in Figure 23. The key logic node for the spare row is marked by an (A) on the diagram. When the spare row is not in use, node (A) is held permanently low by transistor (T) whose gate is held high by the spare row enable block. When the spare row is to be used, a fuse is opened within the spare row enable block and the pulldown gate is brought to ground so that the programming elements are enabled. Under control of a fuse, either address true or address complement is transmitted through each programming element. Thus, by blowing the proper fuses, the address of a faulty row in the array is programmed into the spare row.

Figure 24 shows the basic configuration of a programming element. V_G and V_{DP} are special high voltage supplies used only during programming. They are brought on-chip by extra pads probed at wafer sort. These pads are not bonded out to the package but instead, V_G is grounded and V_{DP} is tied to V_{DD} by on-chip transistors. No inadvertent programming can occur at the package level because P_1 cannot turn on and current through the fuse is limited by the transistor connecting V_{DP} and V_{DD}. To blow the fuse, the programming address is brought low, which raises the gate of the programming transistor P to a high voltage. A high current flows through the fuse and it opens. When programming is complete, V_G is brought to ground. If the fuse has been blown, current through depletion transistor D1 pulls node (B) to ground and transfer gate T_2 passes X_i onto X_{pi}. If the fuse has **not** been blown, node (B) stays near V_{DP} and $\overline{X_i}$ is transferred onto X_{pi}.

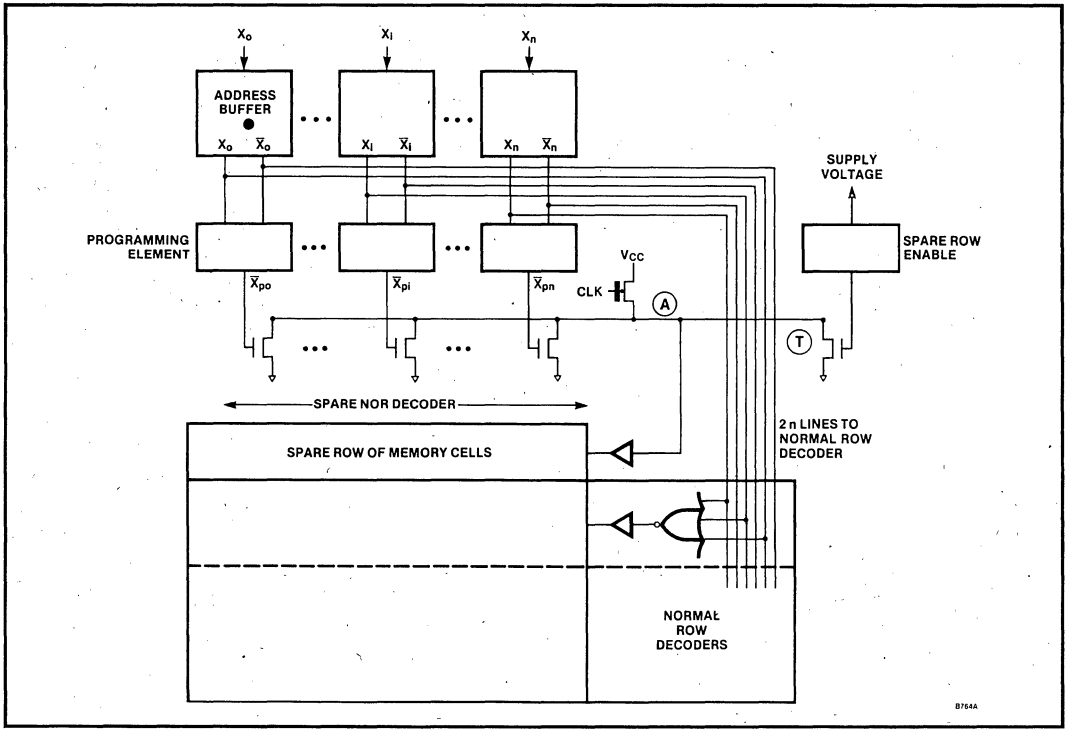


Figure 23. Block Diagram for a Spare Row

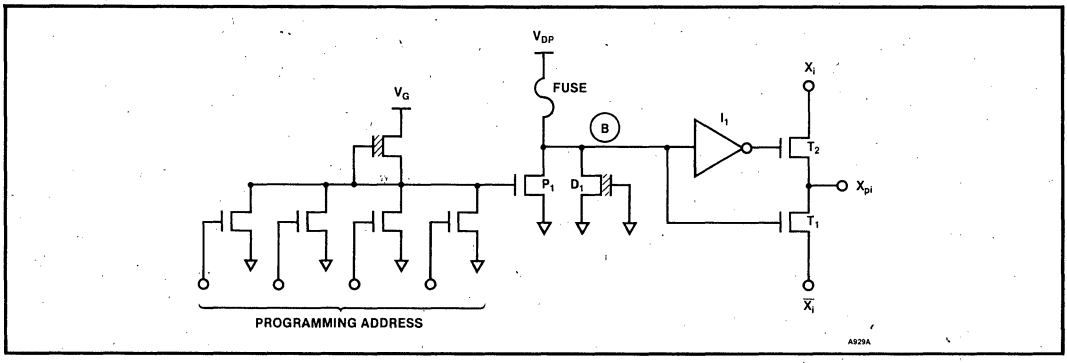


Figure 24. Simplified Circuitry for Programming Element

When the spare row is enabled, one task of the circuit is to deselect the faulty element. Figure 25 illustrates the technique which is used. Whenever any spare select line rises, it causes the "normal element disable" line (NED) to rise as well. NED is connected to one extra input of every normal word select decoder. Thus, when a spare element is selected, it automatically deselects not only the faulty element it replaced, but also every other normal element of the array. The timing of the spare select buffers and the NED generator are optimized to assure that the faulty element is deselected prior to the selection of the spare element.

Another precaution is taken to avoid adverse effects from possible breaks in the faulty select line. If the far end of a broken line were allowed to float, it could present a hazard to data integrity. In the case of a broken word line in the 2164A, word line clamps protect the far end of each row select line from floating high.

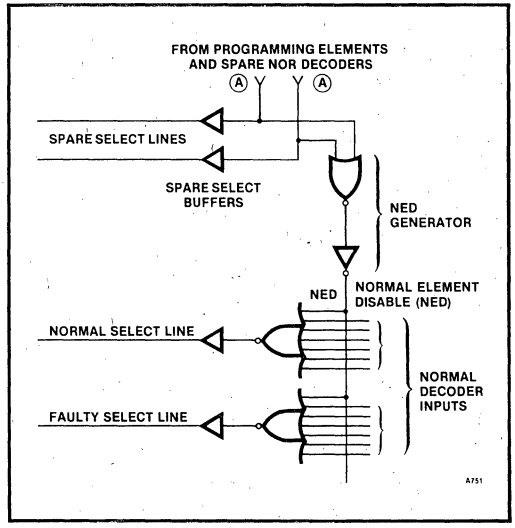


Figure 25. Deselecting a Faulty Element

As mentioned previously, the repair of faulty elements is done during wafer probing. As they come out of fabrication, all spare elements are disabled, allowing full testing of the normal array. Bits are tested not only for hard failures, but also for latent oxide or silicon defects through stressing. The location of any bad bit is stored in the tester's memory. This information is then processed to determine the optimum usage of the spare elements. Then, the spare elements are programmed into their proper logical locations. Finally, the die is tested once more to assure that repair has occurred as planned.

The dice are then assembled as usual. Rigorous class testing is performed to guarantee that the devices meet data sheet specifications in every respect. Both device and system level characterizations have revealed no pattern sensitivity related to the use of redundancy, even when spare elements are intentionally programmed to locations expected to be most susceptible.

Analysis of 2164A devices shows that the worst case patterns do not involve interactions between columns or rows. Replacing the entire row or column introduces no new sensitivity.

The internal delays of redundant element decoding are buried within the internal clocks of the 2164A and have no effect on access time. Figure 26 shows access times for a 2164A before and after repair.

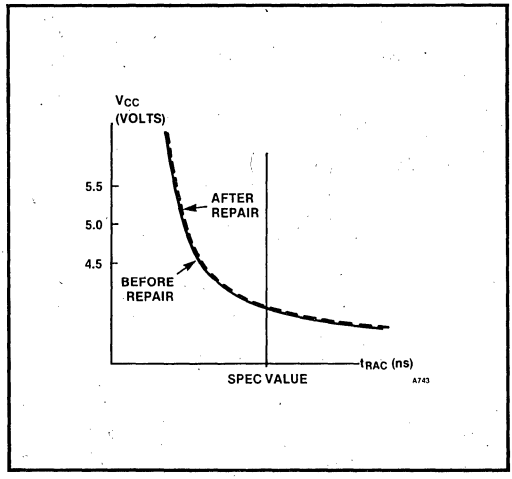


Figure 26. Intel® 2164A t_{RAC} vs V_{CC}

The concept of using redundancy for yield enhancement is well-established. Initially researched by IBM in 1964, Intel has now implemented this concept with the introduction of the 2164A. It is expected that others will follow this lead, and that by the mid-1980's, redundancy will be standard in all memory devices.

8. SUMMARY

The Intel 2164A, made possible by Intel's HMOS-D III technology, introduces a new generation of denser dynamic RAM devices, featuring redundancy, +5V-only TTL-compatible operation, high performance, low power and ease of use. Additional system level design information can be found in Intel Applications Note AP-74, "High Speed Memory System Design Using the 2147H," and AP-133, "Designing Memory Systems For Microprocessors Using the Intel 2164A and 2118 Dynamic Rams."

ADDENDUM

A typical user qualification program of memory devices fits into two categories: device-level qualification and system-level qualification. Occasionally during these programs, failures occur that are not related to the device under evaluation.

At the component level, devices are tested individually for performance to specifications. These tests are usually accomplished with the use of sophisticated software-driven memory testers and environmental handlers. Due to the complexity of the test setup, several problem areas arise. Often testing (software) errors cause failures. Omission of dummy cycles or violation of refresh specifications makes failures invalid. Many times the device under test is remote from the test deck of the system. This can cause excessive power supply noise at the end of the cables. Timing skews, glitches on clock lines and I/O levels at the device are complicated by testing at the end of long cables. Output loading is also critical for the device to perform to specifications.

During system-level qualification, the problems encountered are significantly different. Here the devices are again checked for their performance to specifications. Many devices are simultaneously evaluated whether in a memory system test environment or in an actual system manufactured by the user. Problems can also occur from improper gridding or decoupling on the memory card itself. With the complicated signal paths in a memory system, and the difference between vendor specifications, careful attention must be given to timing and skews not to exceed data sheet values. Errors from timing can result in bus contention or can cause many devices to fail test. Of course, with dynamic RAMs, arbitration between access and refresh modes must be reliable to guarantee the refresh specifications of the RAM.

These problems can be avoided with careful preparation. However, if problems do arise during qualification, don't hesitate to call your local field applications engineer or sales office.

REFERENCES

1. Intel® 2164A Data Sheet, March 1982.

intel

APPLICATION NOTE

AP-132

June 1982

Designing Memory Systems with the 8K x 8 iRAM

John J. Fallin
William H. Righter
Memory Components
Applications Engineering

Order Number: 210443-001

1 INTRODUCTION

1.1 RAM Overview

Matching the correct RAM to microprocessors is fundamental to effective product design. Understanding the advantages and disadvantages of each device type enables a microprocessor system designer to choose the best product for his particular design objective.

Two basic types of semiconductor random access memories (RAMs) are in use at present: static RAMs (SRAMs) and dynamic RAMs (DRAMs). Where large amounts of memory at the lowest cost per bit is required, such as main computer memory, the dynamic RAM holds a commanding position. The extra costs of refresh, timing and arbitration overhead are spread over a very large amount of memory. The static RAM, however, provides a better solution for relatively small memory systems where high performance or simple system design is desired.

A major advantage of dynamic RAMs is low memory component cost. A DRAM uses a simple one-transistor, one-capacitor cell for binary storage. This simple design achieves high integration density and low cost. When a DRAM cell is not being written, read or refreshed, it consumes almost no current. At any given time, the majority of the cells in a DRAM array will be in this condition — yielding low overall power consumption.

One disadvantage of DRAMs are their extensive control and interface requirements. The DRAM control circuitry must generate signals such as $\overline{RAS}$ and $\overline{CAS}$, provide refresh cycles, and handle arbitration. This adds to the component count and overhead costs, both in design and implementation.

Conversely, static RAMs need very little external control circuitry and they interface easily to most microprocessors. An SRAM has no refresh requirement and usually has all of its control signals generated directly by the system microprocessor. A disadvantage of the SRAM is its high cell complexity. A typical static RAM cell requires four to six transistors — resulting in a lower cell density and higher manufacturing cost/bit than DRAMs.

A new type of RAM has now been developed that combines the best features of the SRAM and DRAM and is called the iRAM (integrated RAM). An iRAM is an entire dynamic RAM system integrated onto a single silicon chip, including the memory array, refresh logic, arbitration, and control logic. This new implementation combines the cost, power and density advantages of a DRAM with the ease of use of a static RAM. Because all of the DRAM control logic is internal, the memory system can operate autonomously, controlling its own refresh and arbitration. This greatly simplifies microprocessor interfacing and minimizes additional TTL hardware support. Proper refresh is guaranteed and overall system performance improved.

1.2 iRAM Concept Background

With the advent of VLSI technology and 64K RAM densities, it became possible to further integrate and simplify memory system design. LSI memory controllers integrate all of these components into a single device (such as Intel's 8202A and 8203 dynamic DRAM controllers). Figure 1 shows the major elements of such a dynamic RAM controller.

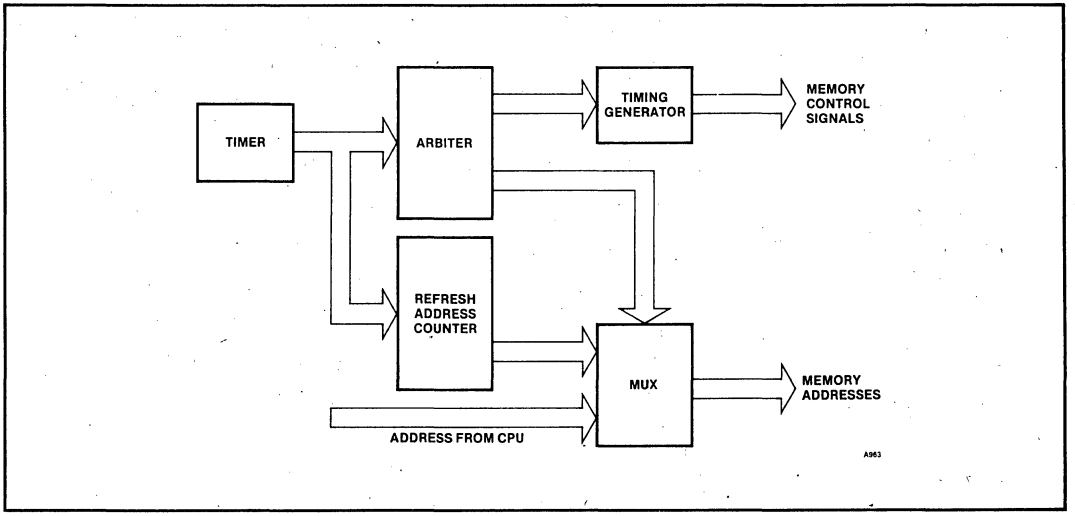


Figure 1. Memory Control Block Diagram

Figure 2 shows a simple microprocessor memory system implemented with three major blocks: the CPU, the memory array, and a memory controller. An example of this configuration is a system comprising an 8088 CPU, and 8203 DRAM controller and a 2164A memory array. To advance this configuration to a higher level of integration would require a decision on whether to place the memory control inside the CPU or within the memory itself.

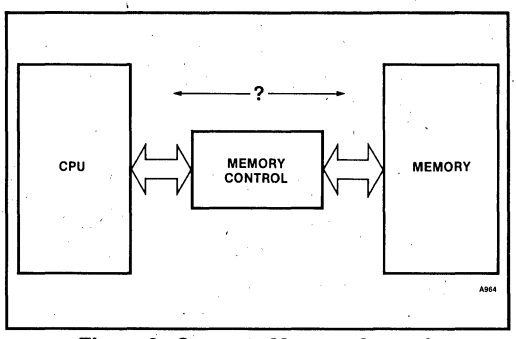


Figure 2. Separate Memory Control

Memory control incorporated within the CPU requires CPU participation in all memory references — just to preserve refresh. This includes DMA (direct memory access) which normally doesn't require or permit CPU intervention. Also, the CPU must run continuously. Single stepping, hold operations, extended WAIT states and the special block data move instructions of some microprocessors must all be carefully avoided to preserve refresh and maintain data integrity of the memory system. While these constraints can be acommodated with careful design, the added overhead does limit the full CPU processing capabilities and overall system performance.

A sensible alternative is to integrate the memory controller circuits into the memory — completely freeing the CPU of this task. While this approach places an additional burden on the device designer, it greatly simplifies the task of the system designer by eliminating the design problems associtated with refresh and timing. This permits a very simple interface to the CPU and yet provides guaranteed refresh, optimized timing, and minimal hardware support requirements.

A microprocessor integrates all the components of a central processing unit into one device. An iRAM integrates all the components of a dynamic RAM memory system into a single device. This is unlike the pseudostatic or quasi-static RAM devices which only incorporate a portion of the refresh circuitry onto the memory chip and still require much control from the CPU. The integration used in the iRAM includes the refresh timer, refresh address control and counter, address multiplexing, and memory cycle arbitration as well as an 8-bit wide memory array. Figure 3 is a pictorial representation of this concept.

1.3 Memory System Size and Cost Constraints

Integrated RAMs are primarily intended for use in microprocessor memories usually less than or approximately equal to 64K bytes, while standard DRAMs with a separate controller are more cost effective in larger memories. The relative costs of systems designed with various device family types are shown in Figure 4. A range is shown for each alternative to represent the change in cost over time. Thus, the 2K × 8 SRAM is a good choice for very small memory systems of less than 8K bytes while DRAMs provide a clear advantage in the region beyond 64K bytes. In the region between 8K and

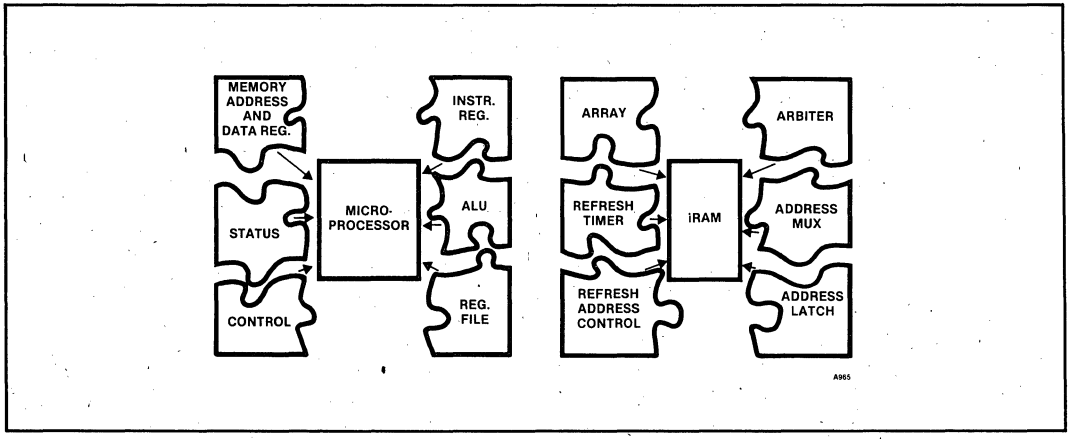


Figure 3. iRAM/Microprocessor Comparison

64K, however, standard DRAMs are usualy not as cost effective because of the overhead involved in the design and cost of the hardware for the controller. Based on these comparisons, iRAMs have a clear advantage for anything other than very small or very large memory systems.

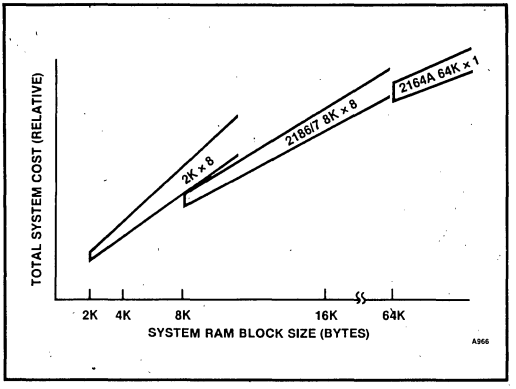


Figure 4. System Cost Graph

1.4 Byte-wide Universal Memory Site

The byte-wide universal memory site concept allows a system designer to create one or more memory sites that can accommodate several types of x8 memories, including RAMs, ROMs, EPROMs, and E^2PROMs. The universal site is depicted in Figure 5. Though based on a 28-pin site, the universal site also supports 24-pin devices. For this site to be truly universal, it should contain provisions for memory densities that have not yet been developed.

Figure 6 shows various memory classes and how they conform to the universal site. The universal site is particularly useful in development of microprocessor systems in which the hardware design of the memory site may be completed early in the design cycle before the RAM/ROM mix has been specified. For example, a RAM might be initially used to store microprocessor instruction code during the development and testing of the system software. This allows code to be run and debugged at full system speed. Initial prototypes and small production runs can place EPROMs in the same sockets, while full scale production may change to PROMs or ROMs. The universal site flexibility also allows an easy upgrade path to next generation (higher density) devices.

A key feature of the universal memory site is the two-line bus control with separate $\overline{CE}$ and $\overline{OE}$ to prevent bus contention in a system. This convention offers a distinct advantage over devices with only one-line control. (Eliminating the effects of bus contention is extremely important and not always easy due to its subleties. Generally, the current and voltage spiking on the power supply rails presents the major problem because this type of noise can lead to a whole host of problems including invalid data, false triggering, race conditions, and reflections, to name a few.)

1.4.1 ONE-LINE CONTROL

With one-line control devices (Figure 7), bus contention occurs when two devices simultaneously occupy a bus (when $\overline{CE}$ of one device goes inactive simultaneously with another devices' $\overline{CE}$ going active). This is the usual situation when chip selects are generated from a decoder. The contention occurs because it takes more time for the output of the deselected device to turn off (switch to high impedence) than the short output buffer turn-on time of the selected device. Because the data lines are wire-ORed to a common data bus, any data bits of opposite polarity will cause bus contention (Figure 8).

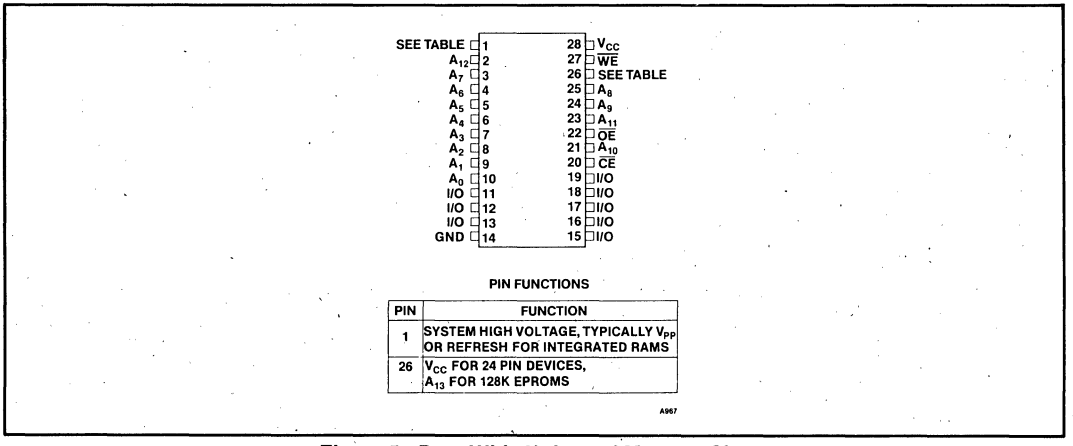


Figure 5. Byte-Wide Universal Memory Site

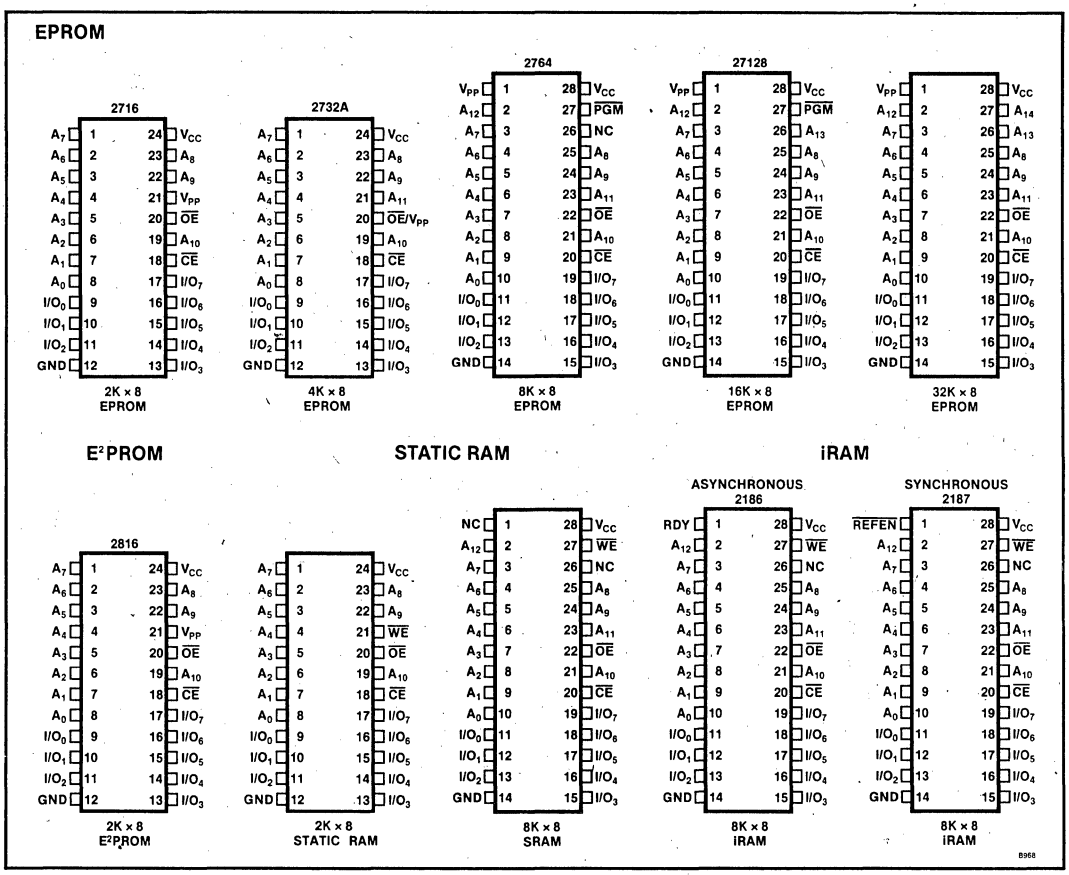


Figure 6. Intel's Line of Universal Products

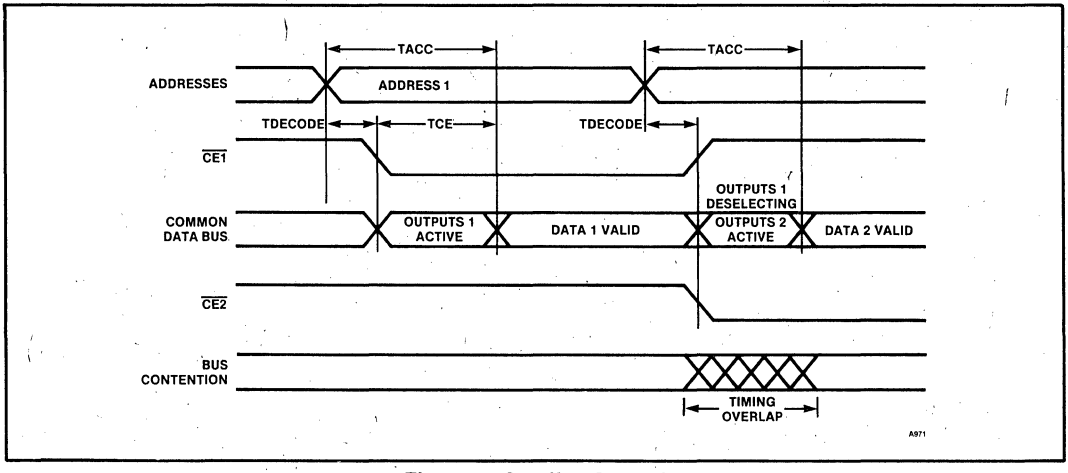


Figure 7. One-line Control

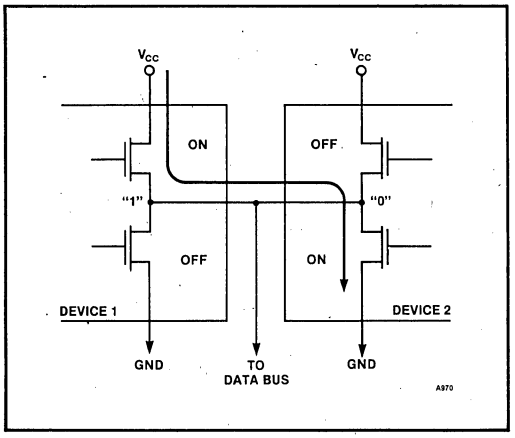


Figure 8. Bus Contention

1.4.2 TWO-LINE CONTROL

Similar to one-line control, two-line control logic allows the $\overline{CE}$ of one device to go inactive simultaneously with another going active. However, the timing diagram in Figure 9 shows that no bus contention occurs because the $\overline{OE}$ of the selected device is not enabled until the outputs of the deselected device have switched off the bus.

The use of an independent output enable is the best way to eliminate bus contention in the system. The use of non-integrated output buffers cannot achieve the same result; they can only confine bus contention to a memory card or memory section of a large card. In addition, as processor speeds increase, greater demands are placed on memory performance and the use of external non-integrated output buffers places still more constraints on memory system performance. In this context, the time between addresses out and data in is a fixed interval for any given processor. All devices inserted in the path, demultiplexers, transceivers, decoders, etc., must be compensated for by a higher speed memory.

2 DEVICE DESCRIPTION

2.1 Overview

The 2186 and 2187 iRAMs are 5-volt only, dynamic RAM 8K × 8 systems integrated on a single chip (Figure 10). The memory devices have been designed for easy use with microcontrollers, multiplexed address/data bus microprocessors, and processors with separate address and data paths. These memories are referred to as integrated RAMs or "iRAMs" because they contain refresh timing and control logic. The 2186/87 iRAMs include the following major features:

- Easy to use on-chip self-refresh, including:
  - — Internal refresh timer
  - — Refresh address counter
  - — High speed arbiter (2186 only)
  - — Refresh address multiplexer
  - — Complete internal timing control
- External refresh control option (2187 only)
- Microprocessor handshake signal (2186 only)
- Outputs drive two low power Schottky TTL loads and 100 pF

The 2186/87 iRAMs are fabricated using an N-channel double layer polysilicon gate process with depletion loads. The four-quadrant memory array is built with conventional one transistor DRAM cells, polysilicon word lines and folded metal bit lines. Each of the four quadrants contains 128 rows and columns. In addition, four redundant columns and four redundant rows are provided. Two pairs of I/O lines from each of the quad-

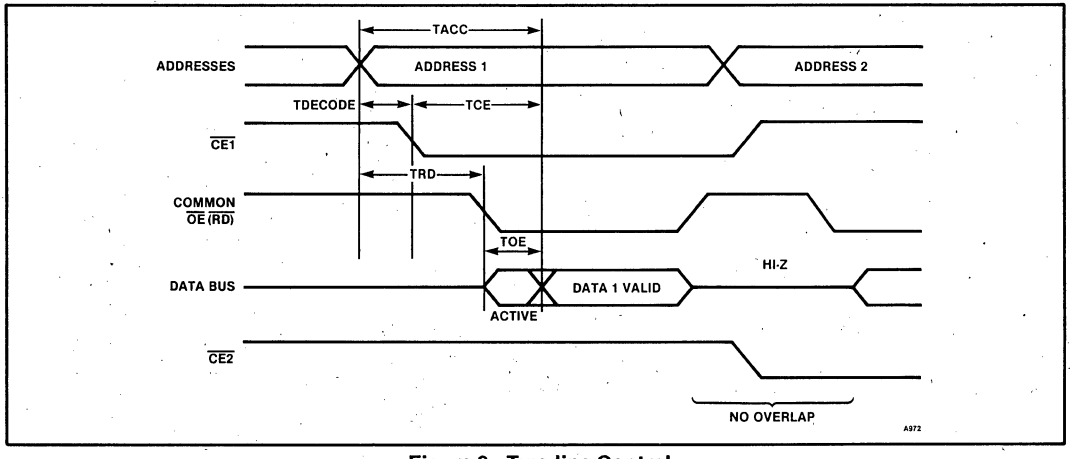


Figure 9. Two-line Control

rants provide a total of eight bits to the data bus. An active restore circuit boosts the bit lines back to a full V_{CC} level after every read or refresh cycle. Boosted word lines and column select lines are used to write a full V_{CC} level into the memory cells. Wide internal operating margins provide a high degree of reliability.

2.2 Device Pinout

The pinout of the 2186 and 2187 is shown in Figure 11. The industry standard 28-pin package conforms to Intel's byte-wide universal memory site (Section 1.4). Pin 1 (labeled "CNTRL") is the only external difference between the 2186 and 2187. On the 2186, Pin 1 is a RDY output — a signal to the system indicating memory status. Pin 1 on the 2187 is a "refresh" strobe ($\overline{\text{REFEN}}$), an input signal for external refresh requests.

Pins 2 thru 10, 21, and 23 thru 25 are the 12 address inputs required to select each of the 8192 bytes. Pins 11 through 13 and 15 through 19 are the eight bits of the bidirectional data bus.

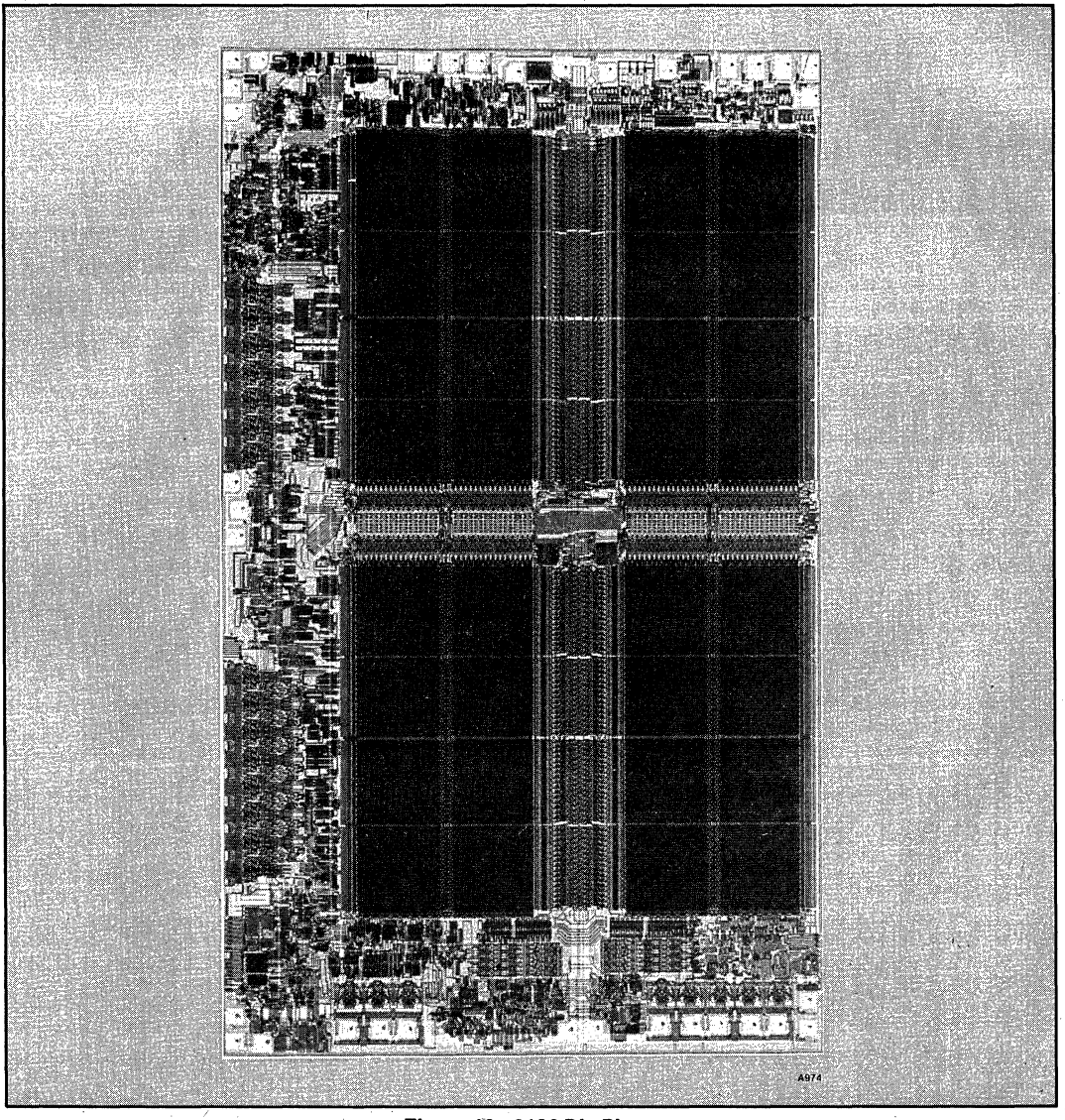


Figure 10. 2186 Die Photo

Pin 27 is the write pulse input strobe ($\overline{WE}$) for data store during Write cycles. Pin 20 is Chip Enable ($\overline{CE}$), which latches addresses and begins the internal memory cycle. Pin 22 is Output Enable ($\overline{OE}$), normally connected to a CPU READ ($\overline{RD}$) line. $\overline{OE}$ enables the iRAM output buffers during a Read cycle.

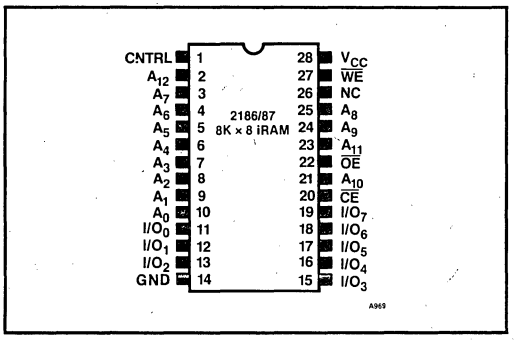


Figure 11. 2186/87 Pinout

2.3 Internal Description

2.3.1 ASYNCHRONOUS AND SYNCHRONOUS REFRESH

The 2186 iRAM contains automatic internal refresh circuitry making it an ideal choice for asynchronous applications. The 2187 does not have the internal arbitration capability as it has been designed for use in synchronous applications.

Pin 1 on the 2186 is the RDY output which serves as the handshake signal (required in asynchronous systems) and is usually bussed to the RDY input circuit of the processor. The RDY output is an open drain device, requiring a 510 ohm pull-up resistor which allows "wire-OR" connections of other device RDY outputs without the need for extra gates.

The 2187 receives external refresh requests via Pin 1 ($\overline{REFEN}$). This input must be strobed 128 times within 2 milliseconds to perserve refresh in the dynamic RAM array. The 2187 iRAM is designed for use in synchronous systems where the user wants control of the refresh cycles. Hence, the designer must provide refresh requests to the iRAM. The 2187 has neither a RDY signal nor any access cycle deferment and because it has no built-in arbitration capabilities, the user must also guarantee that access cycles are not requested during refresh cycles.

Refresh addresses are generated internally in both devices by an onboard refresh address counter. In addition, both devices have an internal refresh timer which, for the 2187, becomes active in a power-down mode.

2.3.2 FUNCTIONAL BLOCK DIAGRAM

Figure 12 shows a functional block diagram of the iRAM.

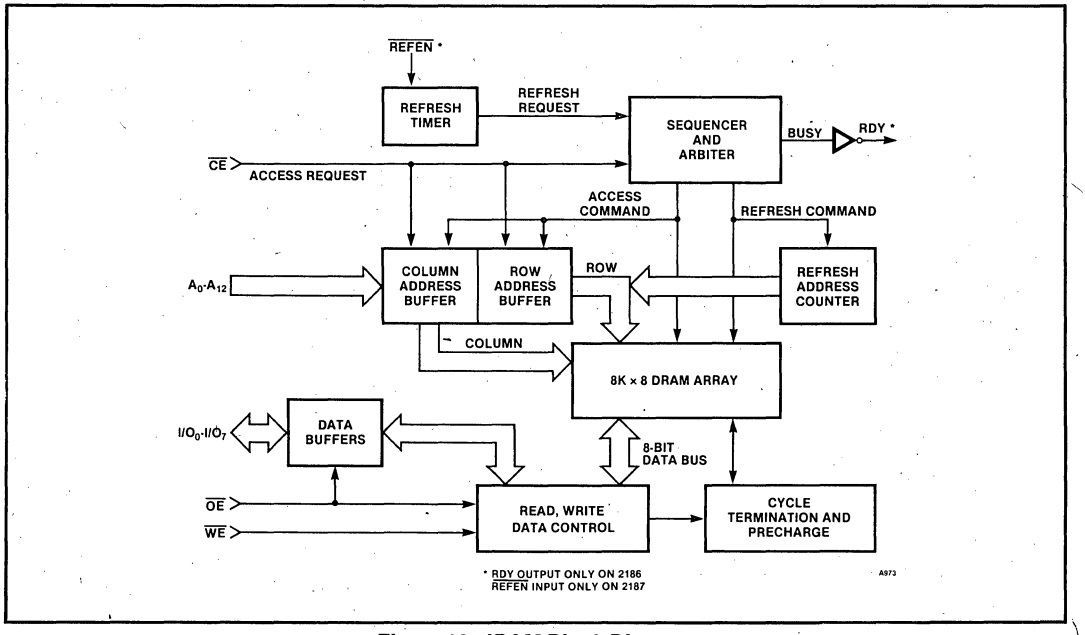


Figure 12. iRAM Block Diagram

2.3.2.1 Refresh Timer

The refresh timer requests refresh cycles as required. The refresh timer has been designed to track with temperature and process variations. The design optimizes the rate at which refreshes occur while still guaranteeing data integrity.

2.3.2.2 Sequencer and Arbiter Circuits

The sequencer and arbiter circuits accept refresh requests from the refresh timer and memory cycle requests from the $\overline{CE}$ input. The internal refresh command and the external memory accesses are asynchronous and either may occur at any time with respect to the other. If one does occur while the other is in progress, the request is queued and the cycle performed after the existing cycle has completed. If a refresh cycle is already in progress at the time an access request occurs, the RDY signal on pin 1 is pulled to V_{OL} informing the system that the access cycle is being deferred. In this instance, the normal cycle will be delayed until after the refresh cycle has been completed. RDY will remain low until shortly before valid data becomes available, after which the cycle is completed in a normal manner. The internal high speed arbiter resolves any conflict wherein an internal refresh command and an external access occur simultaneously. This circuit also generates the RDY handshake signal in the 2186. The sequencer/arbiter circuit also decides which type of memory cycle is to occur and controls the operation.

2.3.2.3 Address Buffers and Refresh Address Counter

External addresses A_0-A_{12} are directed to internal row and column address buffers to generate internal byte addresses. Refresh addresses are generated by an internal refresh address counter and are multiplexed internally with the external row addresses.

2.3.2.4 Data Buffers

Controlled by signals from the read/write data control circuit, the three-state bidirectional data buffers receive or transmit eight data bits.

2.3.2.5 Read, Write Data Control

The read/write data control circuit controls and directs the flow of data between the 8K × 8 DRAM memory array and the data buffers.

2.3.2.6 Cycle Terminator and Precharge

The cycle terminator and precharge circuits ensure proper termination of all memory cycles and precharge the dynamic circuitry in preparation for the next cycle.

3 DEVICE DESCRIPTION

All timing signals used throughout this document are denoted by various alpha character strings to indicate certain basic conditions or parameters. Understanding signal name derivation will enable the reader to arrive at a correct interpretation of any signal name encountered. Figure 13 illustrates the meaning of various letters used in a signal name.

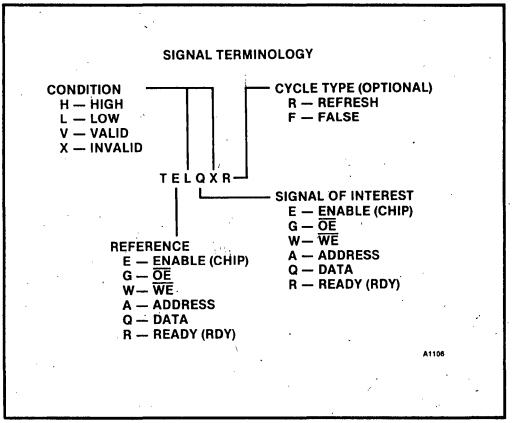


Figure 13. Timing Signal Terminology

Each control signal is given a one-letter designer; i.e., $\overline{CE}$ is represented by E, $\overline{OE}$ by G, etc. Each of these letters is followed by another letter describing the state of the foregoing. In addition, a timing descriptor may have a letter added to the end to describe a special case. For example, TELGL is the time from $\overline{CE}$ low to $\overline{OE}$ low, while TEHELF is the time from $\overline{CE}$ high to the next $\overline{CE}$ low during a false memory cycle; TELQVR is the time from $\overline{CE}$ low to data valid for a not ready condition.

The 2186 and 2187 are edge-triggered devices that recognize a timing edge as a signal to start an operation. Because of this, $\overline{CE}$ must be allowed to make only one transition per cycle, otherwise the device cycle time (TELEL) will be violated. The 2186 and 2187 latch all external addresses on the leading edge of $\overline{CE}$. Data is latched into the device on the **leading** edge of $\overline{WE}$ as opposed to the trailing edge write requirement which is common among static RAMs.

The 2186 provides four major types of cycles: read, write, false memory, and refresh.

Two major modes of operation exist for both read and write cycles; $\overline{CE}$ pulsed mode and $\overline{CE}$ long mode. For pulsed mode $\overline{CE}$ operation, the low $\overline{CE}$ time (TELEH) must be less than or equal to $TELGL(TELWL)_{max} + TGLEH(TWLEH)_{min}$, while long $\overline{CE}$ mode requires a longer $\overline{CE}$. (For more detailed timing information, consult the 2186 and 2187 data sheets.)

3.1 Read Cycle

A read cycle (Figure 14) is initiated by both $\overline{CE}$ and $\overline{OE}$ going low during the same cycle. Depending on the low time of $\overline{CE}$, either a pulsed or long $\overline{CE}$ mode will occur.

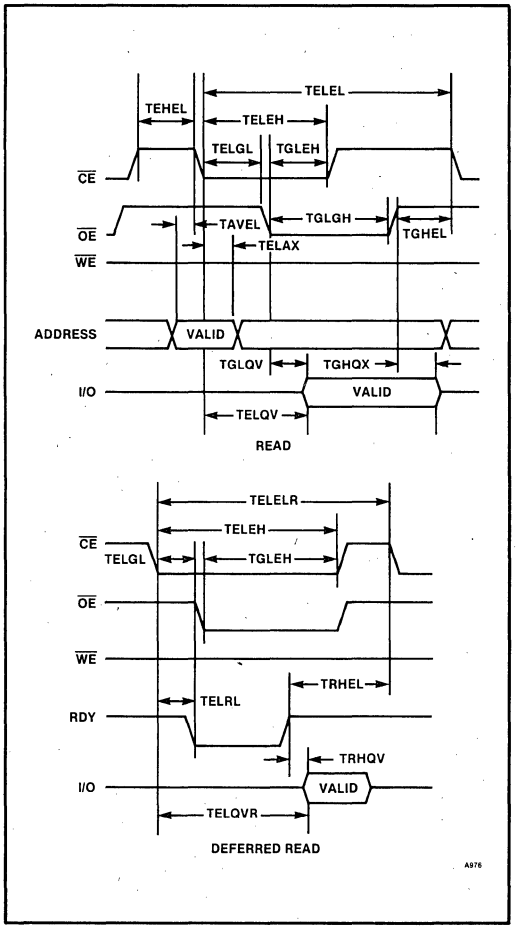


Figure 14. Read Cycle Timing

3.1.1 PULSED MODE $\overline{CE}$ READ

For pulsed mode, a $\overline{CE}$ read cycle is initiated on the falling edge of $\overline{CE}$ at which time either a refresh is or is not in progress.

Refresh cycle not in progress

With a refresh cycle not in progress, the memory cycle can immediately commence (non-deferred read cycle). After the falling edge of $\overline{CE}$, $\overline{OE}$ must go low within a specified period of time (TELGL). If this latter condition is not met, a false memory cycle (FMC) will occur (see Section 3.3). At some point after $\overline{OE}$ goes low, data will become valid and remain so for as long as $\overline{OE}$ is active, independent of $\overline{CE}$.

Refresh cycle is in progress

If a refresh cycle is in progress at the time $\overline{CE}$ goes low, the read cycle will be delayed (deferred read cycle) until after the refresh cycle has completed. In this event, the 2186 will respond very quickly with a RDY low output (TELRL). After the refresh cycle is completed, the read cycle will commence and data will be available at a given time after RDY returns high (TRHQV). As was the case with the non-deferred read cycle, TELGL must be met or an FMC will occur.

3.1.2 LONG $\overline{CE}$ MODE READ

For long $\overline{CE}$, a read cycle mode is initiated on the falling edge of $\overline{CE}$. Similarly to pulsed mode $\overline{CE}$, both deferred and non-deferred write cycles may occur where a deferred cycle causes RDY to be pulled low.

In the long $\overline{CE}$ mode of operation, $\overline{CE}$ must be held low for a given period of time after $\overline{OE}$ goes low (TGLEH). Violation of this specification will cause an FMC to occur. At a given time after $\overline{OE}$ goes low, valid data will become and remain available throughout the duration of $\overline{OE}$'s active period, independent of $\overline{CE}$.

Note that deferred access cycles are not allowed for the 2187.

3.2 Write Cycle

A write cycle (Figure 15) occurs when both $\overline{CE}$ and $\overline{WE}$ go low during the same cycle. As is the case for the read cycle, either a pulsed or a long $\overline{CE}$ mode can occur.

3.2.1 PULSED MODE $\overline{CE}$ WRITE

In the pulsed mode, a $\overline{CE}$ write cycle is initiated on the falling edge of $\overline{CE}$. At this time, a refresh cycle may or may not be in progress.

Refresh cycle not in progress

With a refresh cycle not in progress, the memory cycle can immediately commence (non-deferred write cycle). After the falling edge of $\overline{CE}$, $\overline{WE}$ must go low within a specified period of time. If this latter condition is not met, a false memory cycle (FMC) will occur (see Section 3.3). On the falling edge of $\overline{WE}$, data is latched into the device.

Refresh cycle is in progress

If a refresh cycle is in progress at the time $\overline{CE}$ goes low, the write cycle will be delayed (deferred write cycle) until after the refresh cycle has completed. In this event, RDY is brought low and held there until the refresh cycle has completed. Note that data is still latched into the 2186 on the falling edge of $\overline{WE}$.

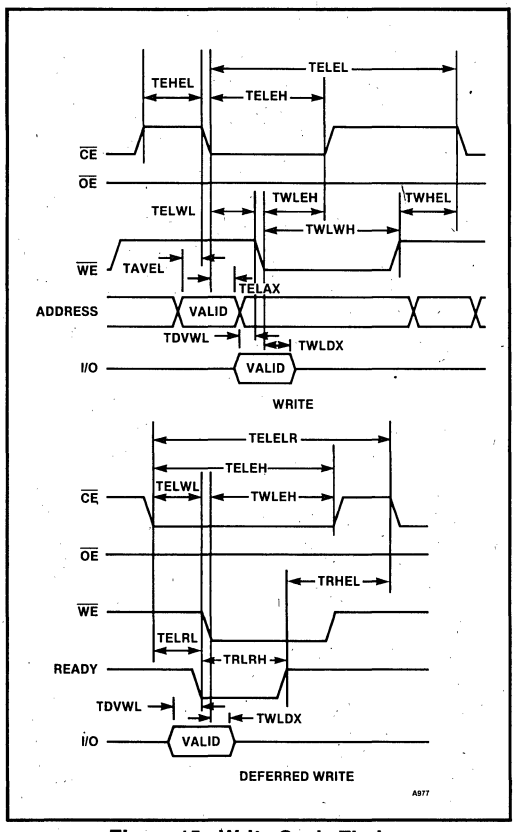


Figure 15. Write Cycle Timing

3.2.2 LONG $\overline{CE}$ MODE WRITE

A long mode $\overline{CE}$ write cycle is initiated on the falling edge of $\overline{CE}$. As is the case for a pulsed mode $\overline{CE}$, both deferred and non-deferred write cycles may occur with RDY being pulled low in the deferred cycle.

For the long $\overline{CE}$ mode of operation, $\overline{CE}$ must be held low for a given period of time after $\overline{WE}$ goes low (TWLEH). Violation of this specification will cause an FMC to occur. On the falling edge of $\overline{WE}$, data is latched into the device.

3.3 False Memory Cycle (FMC)

A false memory cycle (Figure 16) occurs when $\overline{CE}$ is active and neither $\overline{OE}$ or $\overline{WE}$ go low. In this case, the cycle will automatically be terminated on the trailing edge of $\overline{CE}$. This is a valid mode of operation in which precharge and data integrity are guaranteed.

As an added feature of the false memory cycle, a refresh cycle is performed on the row which is selected by the seven external row addresses.

Note that the $\overline{CE}$ high time (TEHELF) required after an FMC is somewhat longer than the corresponding period required for a read or write cycle (TEHEL).

As is the case with a read or write cycle, FMC cycles can be deferred. RDY response time (TELRL) and recovery time (TRHEL) are the same as for the read and write cycles.

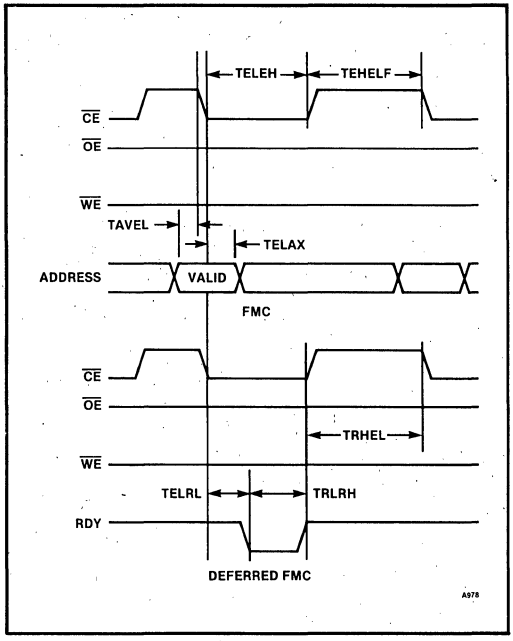


Figure 16. False Memory Cycle Timing

3.4 Refresh Modes

Both the 2186 and 2187 can be refreshed by reading or writing all 128 rows (A_0 through A_6) within a two millisecond period. Several specific modes of refresh operation exist for each part as outlined below.

3.4.1 2186 AUTOMATIC INTERNAL REFRESH

Refresh is totally automatic and requires no external control. In addition, the refresh address is computed internally and does not have to be supplied externally. A high speed arbitration circuit resolves any potential conflict arising between simultaneous access and refresh cycle requests. If a refresh cycle is in progress at the time $\overline{CE}$ becomes active (low), the 2186 will respond with a RDY low output. If, on the other hand, an access or false memory cycle is in progress at the time the internal refresh timer times out, the refresh request will be queued and then performed after the present cycle is complete. Note that RDY will not go low during a refresh unless the RAM is selected by $\overline{CE}$.

3.4.2 2187 EXTERNAL REFRESH

A high-to-low transition on the $\overline{\text{REFEN}}$ input will cause a refresh cycle to be initiated (Figure 17). In this mode $\overline{\text{REFEN}}$ must always be strobed 128 times in a two millisecond period. The $\overline{\text{REFEN}}$ input may be strobed in distributed or burst mode. Refresh addresses are supplied by an internal refresh address counter.

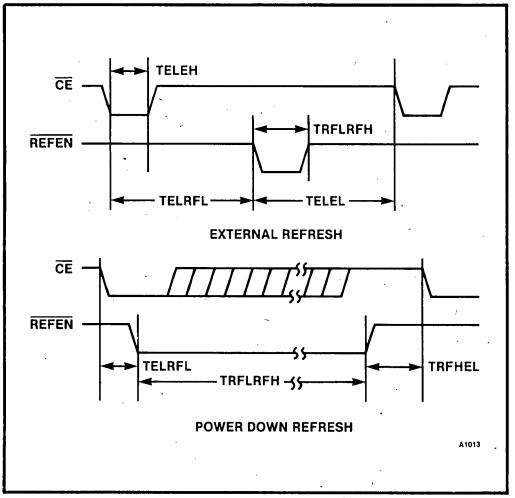


Figure 17. External Refresh Timing

3.4.3 2187 POWER-DOWN AUTOMATIC INTERNAL REFRESH

If $\overline{\text{REFEN}}$ is kept low for greater than one timer period, the internal refresh timer will be activated. Refresh in this mode is totally automatic and requires no external stimulus. $\overline{\text{REFEN}}$ must return high within a specified interval prior to the next memory access cycle (TRFHEL). Attempting to access the 2187 during a refresh cycle is not a valid mode of operation.

3.5 Single-step Operation

Both the 2186 and 2187 can support "single-step" operation for microprocessor system diagnostics. Single-step operation is defined as inserting an unspecified number of WAIT states in the middle of a normal RAM access.

3.5.1 2186

The 2186 supports single-step operation in microprocessor applications which hold $\overline{\text{OE}}$ or $\overline{\text{WE}}$ valid (low) for indefinite periods. Data will remain valid on the bus as long as $\overline{\text{OE}}$ is valid. $\overline{\text{WE}}$ latches data on its falling edge. Automatic refreshes will continue to be performed as needed, even while $\overline{\text{OE}}$ or $\overline{\text{WE}}$ is held low. During this extended cycle, the internal array is free to be refreshed with no threat of access/refresh cycle conflicts. Because of this, RDY will not respond to these extended cycle refreshes.

3.5.2 2187

The 2187 supports single-step operation by following the beginning of a memory cycle with $\overline{\text{REFEN}}$ going and remaining low. Refresh cycles continue to occur periodically as long as $\overline{\text{REFEN}}$ is held low, even if $\overline{\text{OE}}$ or $\overline{\text{WE}}$ remain low indefinitely. Data remains valid on the bus as long as $\overline{\text{OE}}$ is valid. $\overline{\text{WE}}$ latches data on its falling edge. Again, after $\overline{\text{REFEN}}$ returns to a high state, a minimum amount of time (TRFHEL) must be allowed before the next high-to-low transition of $\overline{\text{CE}}$.

3.6 Power-up

3.6.1 2186

To guarantee power-up, all control inputs must be inactive (high) for a 100 microsecond period after V_{CC} is within specification. No dummy cycles are required.

3.6.2 2187

The 2187 power-up is accomplished by holding REFEN active low for 100 microseconds after V_{CC} is within specification. All inputs must be stable and within specification. $\overline{\text{CE}}$, $\overline{\text{WE}}$, and $\overline{\text{OE}}$ must remain inactive (high) during user power-up.

4 INTERFACE CIRCUITRY

There are three key interface circuit considerations when designing with iRAMs.

1. The first consideration is the need for a single edge ("glitchless") transition of chip enable ($\overline{\text{CE}}$) per cycle — because the leading edge transition (active low) of $\overline{\text{CE}}$ latches addresses into the iRAMs and initiates several internal device clocks. Also, there is a minimum specification for $\overline{\text{CE}}$ inactive time (to allow for proper precharge of internal dynamic circuitry).
2. The second consideration concerns write cycles. Because iRAMs write data on the leading edge of $\overline{\text{WE}}$, there is the need for valid data at the memory device before the $\overline{\text{WE}}$ line is activated.
3. The third consideration is the value of same site compatibility with byte-wide SRAMs, EPROMs, ROMs, and E^2PROMs. In particular, allowance for the trailing edge write requirements of SRAMs should be made.

Modest additional circuitry permits compatibility with SRAMs as second sources or allows the iRAM to substitute for ROM or EPROM during debug stages. Several

circuit examples to meet the various requirements of interfacing microprocessors to iRAMs will be described.

Figure 18 shows circuitry for generation of a "glitchless" $\overline{CE}$ from standard 8086 bus signals. Figure 19 shows the circuit timing. This dual J-K flip-flop arrangement guarantees a number of operating conditions. The flip-flops generate a stable $\overline{CE}$ for the iRAMs by enabling the 8205 decoder only after valid addresses have arrived, but early enough to allow the 2186 iRAMs' RDY signal to respond in time to insert a WAIT-state (if required). The circuit also ensures that a minimum $\overline{CE}$ high time is provided. (This is especially important during false memory cycles (FMC) where the $\overline{CE}$ high time specification stretches beyond that of normal cycles).

Also of significance is the compatibility of this circuitry with SRAMs and EPROMs. This includes requiring $\overline{CE}$ to remain valid throughout the cycle.

The interface circuit is simply a two-bit counter designed to start a count sequence when flip-flop A is preset by ALE going high. The $\overline{Q}$ output of flip-flop B along with M/$\overline{IO}$ ($\overline{S2}$ for max mode) is used to enable the $\overline{CE}$ decoder to provide a $\overline{CE}$ to the desired iRAM.

The READY signal is ANDed with the $\overline{Q}$ output of flip-flop A and input into flip-flop B. As long as READY is low, the K input of the flip-flop driving B will stay low, keeping it from being reset. This in turn acts to keep $\overline{CE}$ active. This input allows $\overline{CE}$ to stretch during a WAIT state to meet the requirements or SRAMs or EPROMs that may occupy the same memory site. However, the iRAMs do not require that $\overline{CE}$ be held low for extended cycles.

The circuit in Figure 20 (only for the 8088 — enclosed in dashed lines) offers an alternative. This circuit provides an Enable signal ($\overline{E}$) for the $\overline{CE}$ decoder which is synchronized with ALE. This Enable signal along with

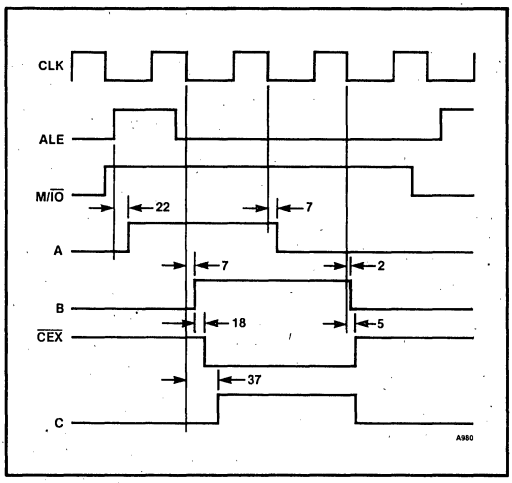


Figure 19. Interface Circuit No. 1 Timing

M/$\overline{IO}$ ($\overline{S2}$ for max mode), is used to enable and address decoder to provide $\overline{CE}$'s to the iRAMs.

A certain amount of skew can occur between the falling edge of ALE and the falling edge of the clock. Two situations can occur: (1) ALE goes low before the falling edge of the clock, the $\overline{E}$ enable line to the decoder remains high until the falling edge of the clock, and (2) ALE goes low at or after the falling edge of the clock, in which case the $\overline{E}$ enable line is immediately activated and enables the decoder. Note that the $\overline{RESET}$ line is used to clear the M/$\overline{IO}$ flip-flop. This causes the 74S138 to be disabled, satisfying the power-up requirements of the 2186 ($\overline{CE}$ remains high). Also, a pull-up resistor is connected to the $\overline{RD}$ line. This ensures that $\overline{OE}$ remains high during RESET (the 8086 three-states $\overline{RD}$ during RESET).

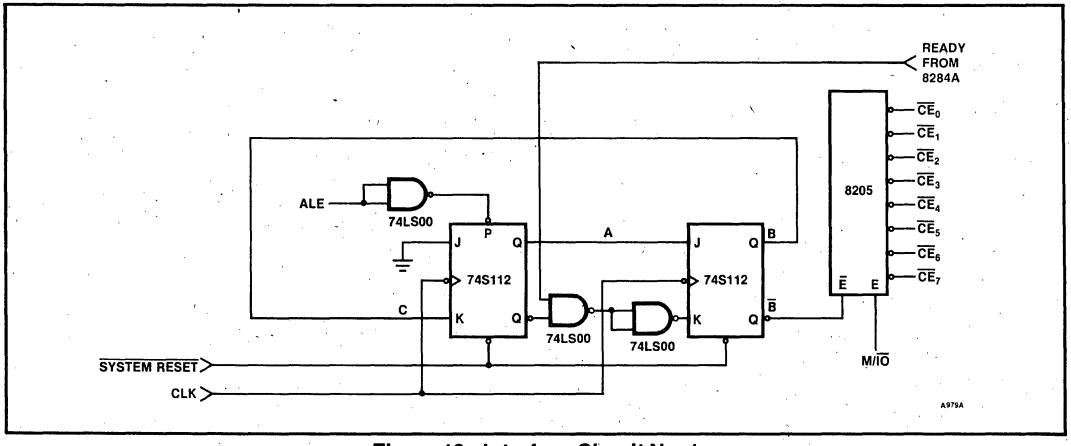


Figure 18. Interface Circuit No. 1

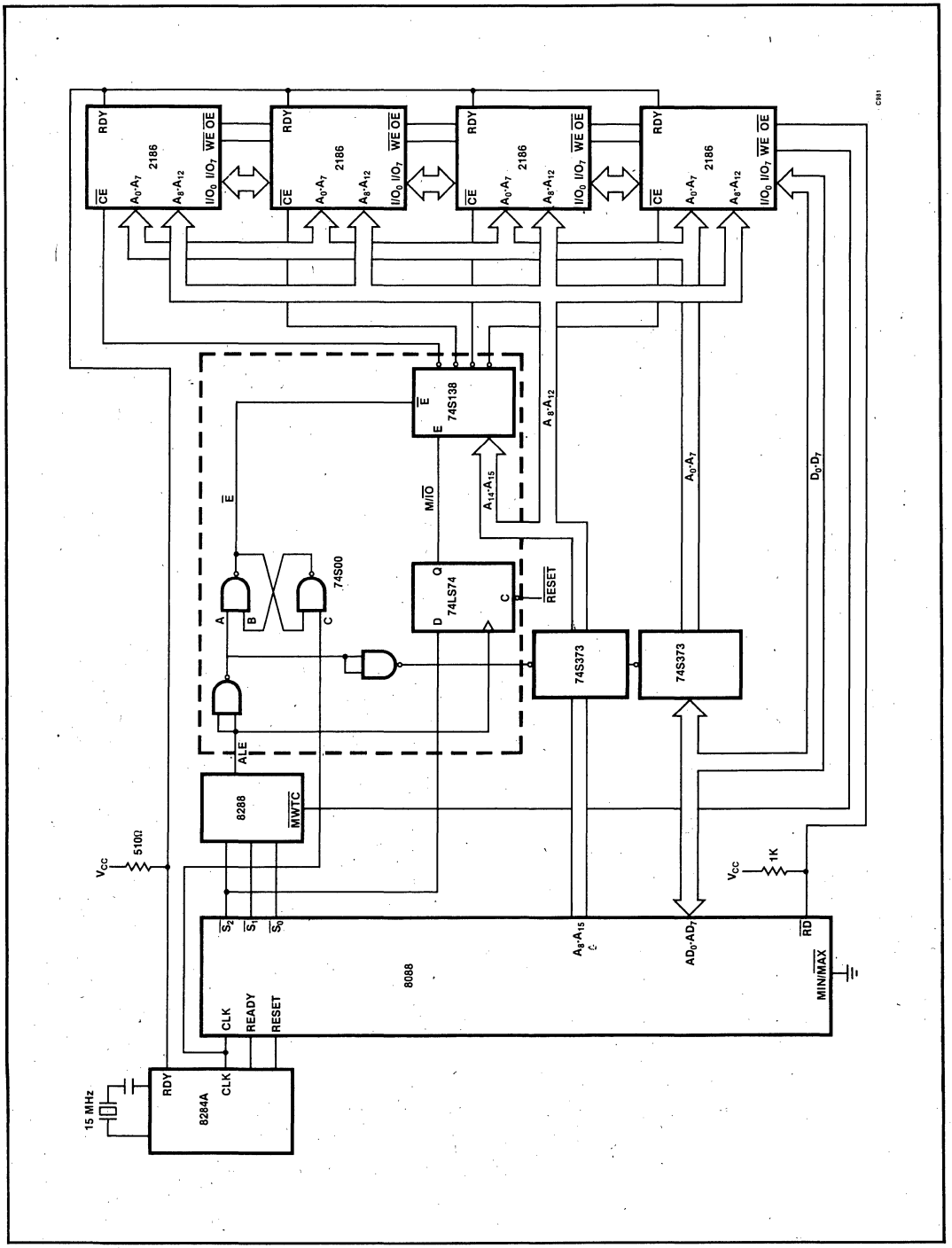


Figure 20. Interface Circuit No. 2

Either circuit will provide all of the interface needed for a 5 MHz 8086 or 8088 max mode system, because MWTC can be used to provide both leading and trailing edge writes. For a min mode system, the circuit in Figure 21 can be used to provide a leading edge write, and the circuit in Figure 22 can be used to provide both a leading and trailing edge write.

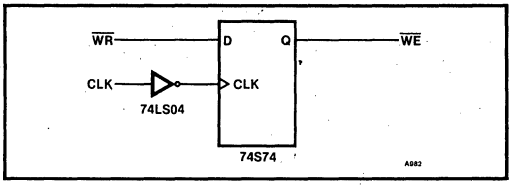


Figure 21. Leading Edge Write

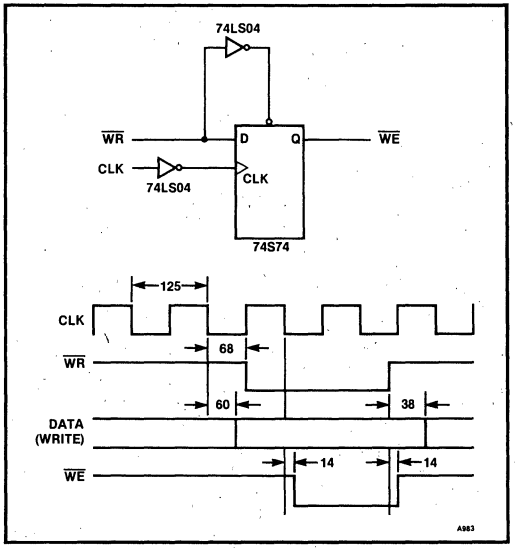


Figure 22. Leading and Trailing Edge Write

A simple, one-gate alternative to the preceding example along with the appropriate timings is shown in Figure 23. This cross-coupled NAND arrangement operates in much the same way as the $\overline{CE}$ generation circuit presented earlier, acting to synchronize the $\overline{WR}$ pulse with the clock. This circuit will provide for both leading and trailing edge writes.

5 SPECIFIC APPLICATION EXAMPLES

This section describes some typical memory interface designs using three types of CPUs: an 8-bit microcontroller, an 8-bit microprocessor, and a 16-bit microprocessor. Design examples are included for both the 2186 and the 2187.

5.1 8-Bit Microcontroller

Figure 24 shows a two-chip microcomputer system using the 8751/8051. This system features 4K bytes of EPROM/PROM and 8K bytes of data storage using the 2186 iRAM. Interface to the multiplexed bus is simplified because the 2186 latches addresses from its external bus on the falling edge of $\overline{CE}$, eliminating the need for latches. In this configuration, the ALE output from the microcontroller is gated with P2.7, and used to generate $\overline{CE}$ of the 2186. The gating of ALE with P2.7 is important for the following reasons: when the 8051 does any type of memory operation, it outputs ALE onto its external bus. This includes internal program memory fetches, in which the ALE cycle time (Figure 25) is only half of what it would be for an external data memory fetch. During these "short" cycles, ALE must be inhibited from generating a $\overline{CE}$ to the 2186, or else the 2186 cycle time with WAIT specification (TELELR) would be violated. To carry this out, P2.7 is initially set to a "1" which is done automatically upon RESET. This "1" will be present on the output during all times except external data memory fetches from addresses below 8000H, at which time P2.7 will go low, allowing ALE to

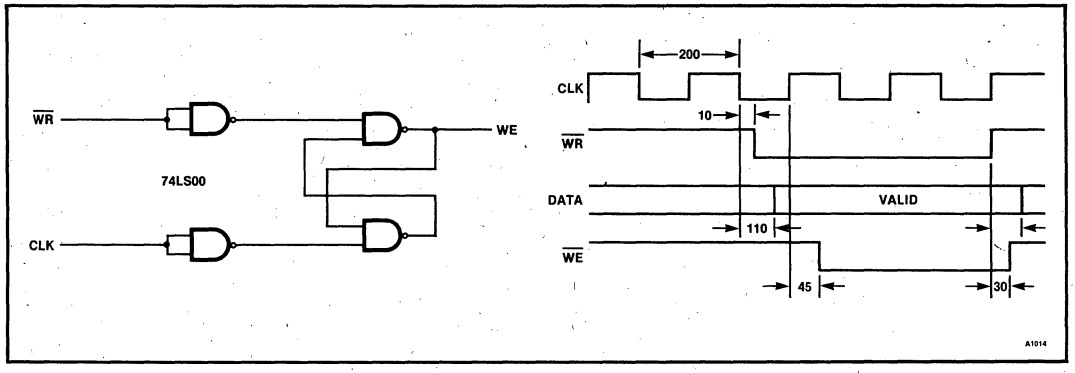


Figure 23. Simplified Write Enable Circuitry

provide a $\overline{CE}$ to the 2186. After completion of the external data memory fetch, P2.7 will revert to its preset value of "1".

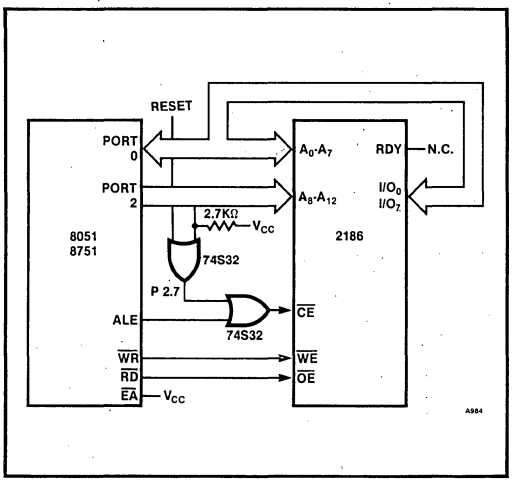


Figure 24. Asynchronous 8051 System

Note that a pull-up resistor is used to ensure that P2.7 will return to a "1" before the next trailing edge of ALE. Timings on the ALE are specified so that all $\overline{CE}$-related parameters on the 2186 are guaranteed, including address setup (TAVEL) and hold times (TELAX), and $\overline{CE}$ high time (TEHEL). The $\overline{RD}$ and $\overline{WR}$ outputs of the 8051 are tied directly to the $\overline{WE}$ and $\overline{OE}$ inputs to the iRAM. Data to be written is guaranteed to be valid before the leading edge of $\overline{WR}$ for the 8051. This provides the leading edge write needed by the 2186/87.

Although a RDY input does not exist for the 8051, a 2186 can still be used for data memory. At 8 MHz the 8051 does not require data back from the data memory until 800 ns after the trailing edge of ALE. The 2186-25 specifies worst case access time at 675 ns from the trailing edge of $\overline{CE}$, which in this system, corresponds to ALE. Even if the 2186 is just starting a refresh cycle when the 8051 requests an access, it will still have time to complete the refresh cycle, and access valid data by the time the 8051 requires it. Note that during RESET, $\overline{CE}$ is kept high to satisfy the power-up requirements of the 2186.

The access time required of program memory is somewhat faster than that needed for data memory. Because of this, the 2186 cannot be used in an asynchronous refresh mode as program memory for a full speed system. However, operation could be guaranteed if the system clock were slowed down.

The synchronous 2187 iRAM can be used as program storage for an 8051 running at 10 MHz by utilizing a method known as clock stretching. The circuitry, as shown in Figure 26, allows the 8051 clock to be stopped in a high state whenever the 2187 requires a refresh cycle. This stretched period is performed at the beginning of a cycle while ALE is high.

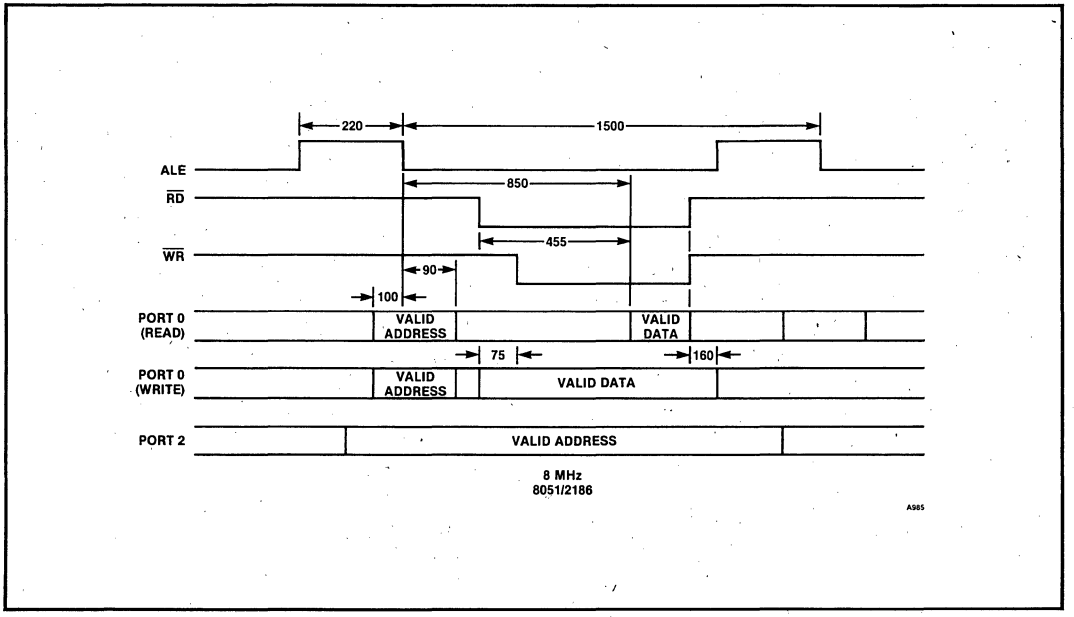


Figure 25. Asynchronous 8051 System Timing

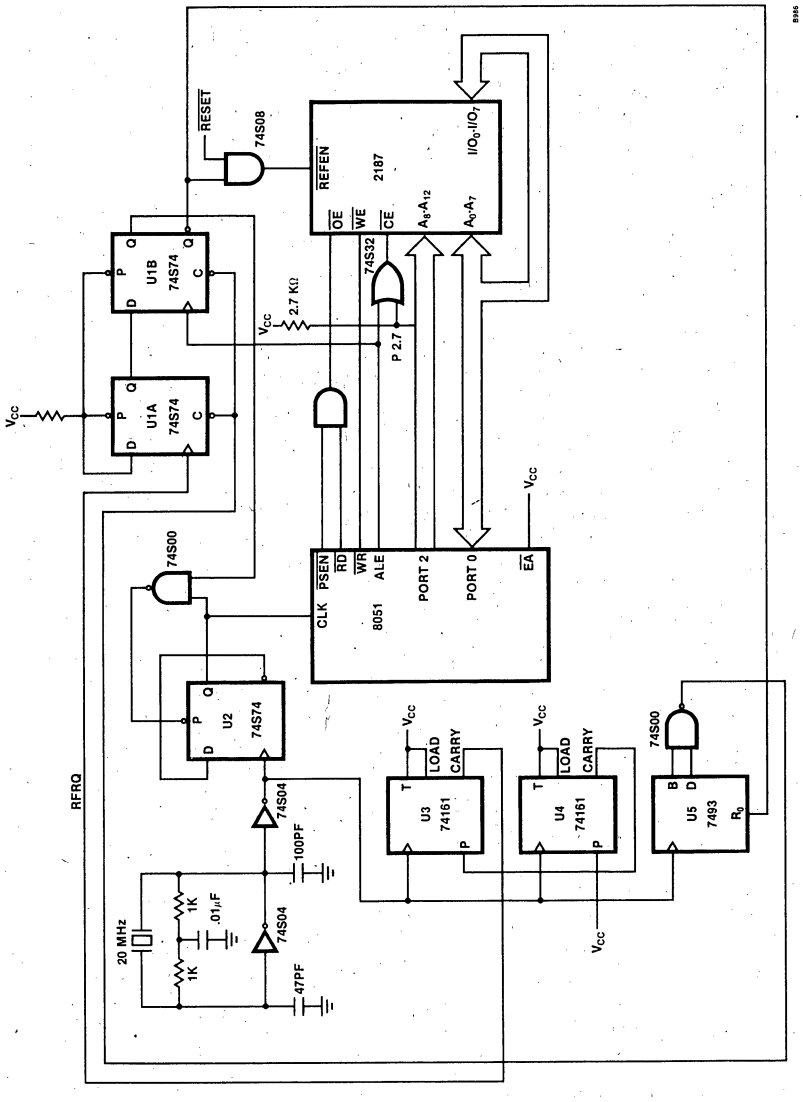


Figure 26. Synchronous 8051 System

Operation of the clock stretching circuitry is straightforward (Figure 27). Under normal operation, U2 acts as a frequency divider for the clock. U3 and U4 count clock pulses, and when a full count occurs, a refresh cycle request is issued (RFRQ). This request sets U1A. On the next high transition of ALE, this request is clocked into U1B, where it causes $\overline{\text{REFEN}}$ to become active. A refresh cycle within the 2187 begins at this time.

At the same time that $\overline{\text{REFEN}}$ becomes active, U5 is released from a clear state to start counting clocks, acting as an interval timer to allow time for the refresh cycle to occur.

On the first high transition of the system clock after U1B is set, U2 will be preset, maintaining the already high state of the clock. This high level is maintained until U5 has counted 10 clock cycles, at which point it acts to reset the clock stretching circuitry and allow the clock to return to a toggling condition.

The clock stretching circuitry used in this system could be utilized to a greater extent than just handling iRAM refresh cycles. For example, it might be useful for some type of DMA operation, or for use with slow peripherals. Also note that no address latches are needed with this system. To satisfy the power-up requirements of the 2187, $\overline{\text{REFEN}}$ must be held low for 100 μsec after V_{CC} is within its specified value. This is accomplished by driving $\overline{\text{REFEN}}$ low during RESET.

In a typical operation, a down-loader program would reside onboard the 8051 in PROM. This program would write program instructions into data memory. These instructions could then be "fetched" out of the same memory which would now be acting as program storage. This overlaying of program and data store is accomplished by allowing either $\overline{\text{PSEN}}$ or $\overline{\text{RD}}$ to enable the 2187 for a READ. Thus, it is possible to create a intermixed data and instruction field.

5.2 8088/2186 8-Bit Microprocessor Design Example

An example of an 8088/2186 iRAM design is shown in Figure 28. The 8088 is connected in a straightforward manner to the 2186 iRAM array. The low order addresses are latched from the multiplexed address/data bus of the CPU by ALE and are connected to the array. The CPU $\overline{\text{RD}}$ provides $\overline{\text{OE}}$ for the iRAMs while the $\overline{\text{MWTC}}$ from the 8288 bus controller serves as the $\overline{\text{WE}}$ for the memory. A stable chip select is generated by circuitry enclosed within the dashed lines. This circuit runs without WAIT states at 5 MHz using the 250 ns 2186-25.

5.3 8086/2186 16-Bit Microprocessor Design Example

The 5 MHz min mode system shown in Figures 29 and 30 depicts a typical interface of 2186 iRAMs with an 8086 16-bit microprocessor. With this arrangement, up to 128K words can be addressed.

To guarantee a stable $\overline{\text{CE}}$, the first interface circuit described in Section 4 is used. The output of this dual J-K flip-flop arrangement is used to enable the 8205 $\overline{\text{CE}}$ decoder.

A False Memory Cycle (FMC) is generated by this circuit during byte write cycles because both devices in the 16-bit word receive $\overline{\text{CE}}$, but only one device (or byte) receives a $\overline{\text{WE}}$. The other device enters an FMC without any consequences at the system level.

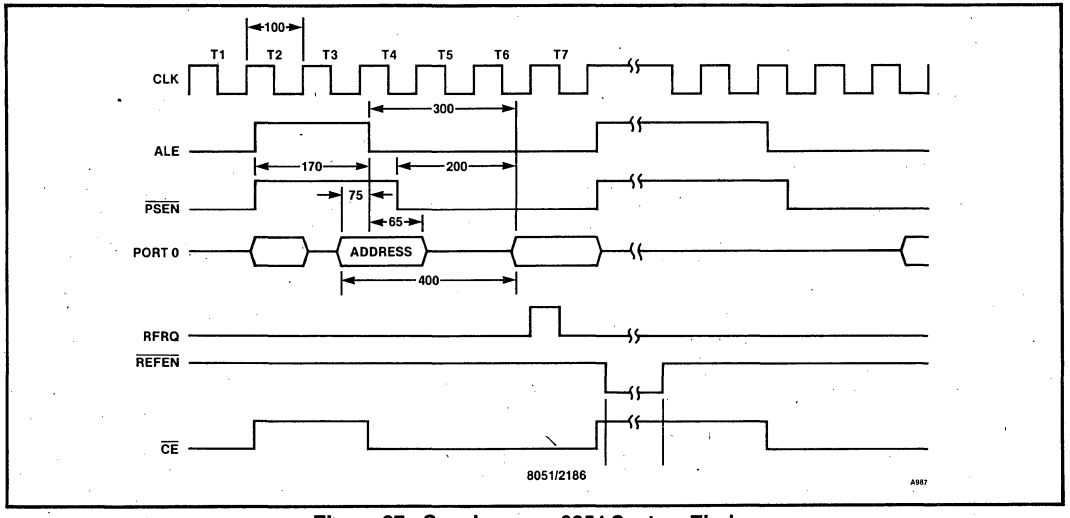


Figure 27. Synchronous 8051 System Timing

In min mode, the 8086 does not guarantee that valid data is present before the leading edge of $\overline{WR}$. A technique to delay this edge in order to provide the iRAMs with a properly timed $\overline{WE}$ must be included in the system. The cross coupled NAND arrangement described in Section 4 is used to provide both leading and trailing edge write compatibility.

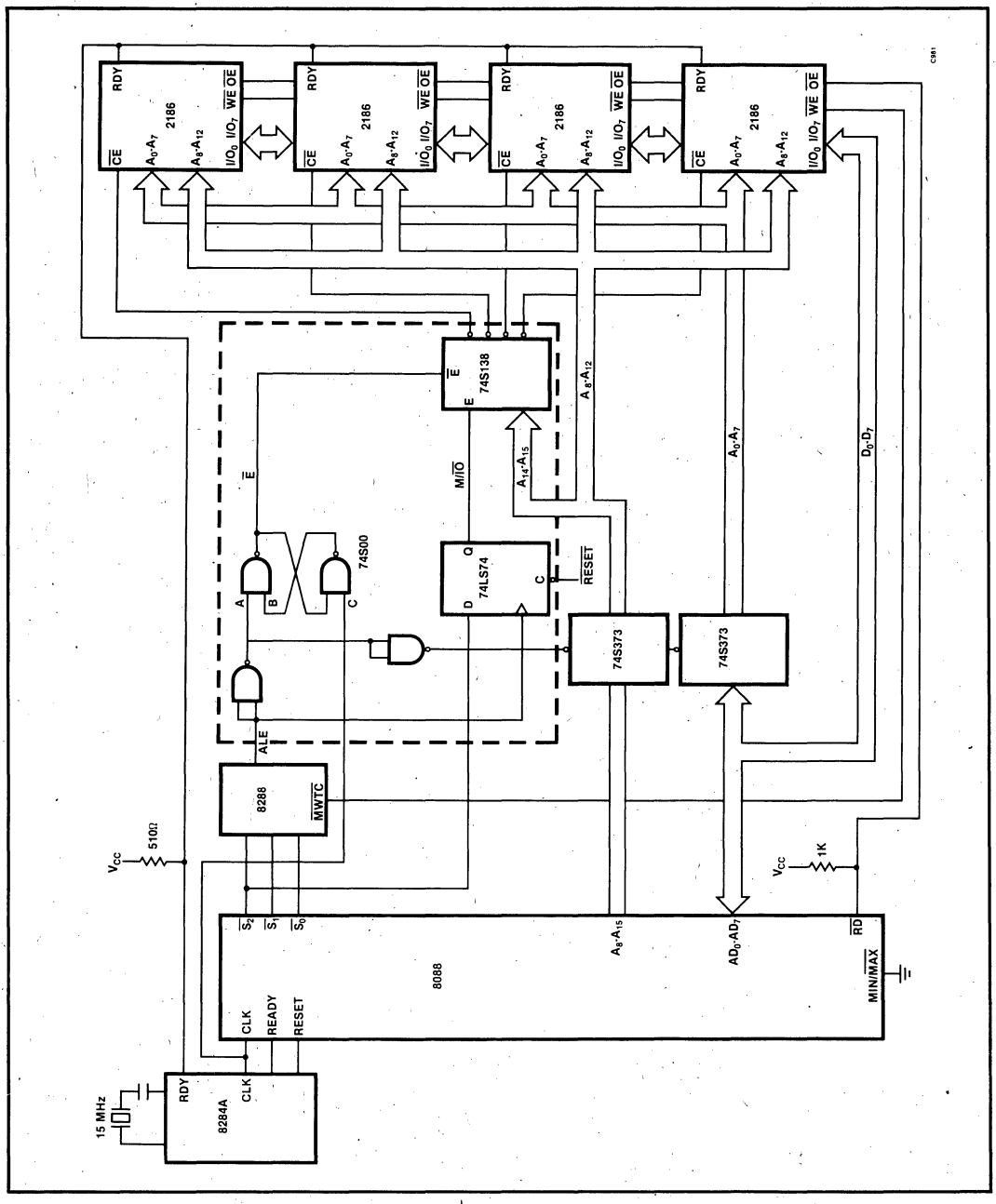


Figure 28. 8088/2186 Microprocessor System

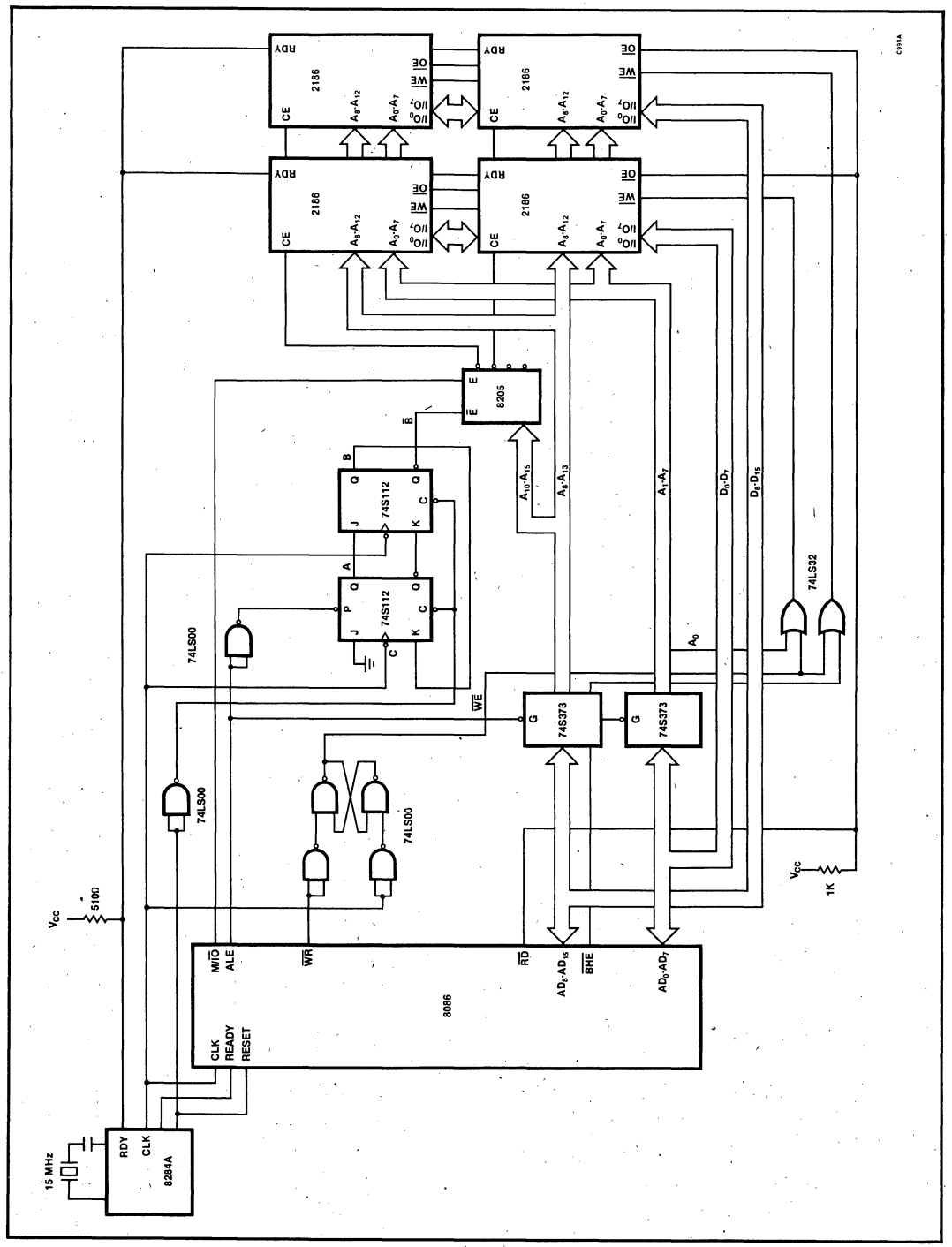


Figure 29. 8086 Min Mode System

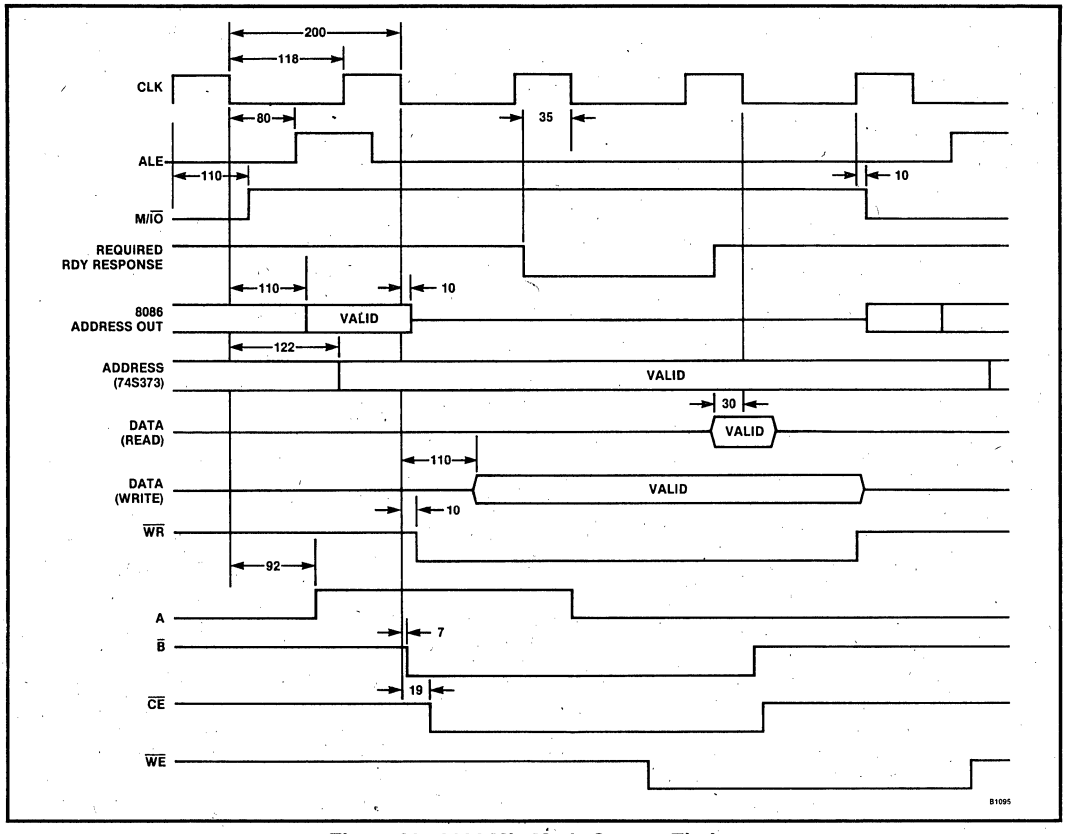


Figure 30. 8086 Min Mode System Timing

If an 8086 max mode system is to be used, the $\overline{WE}$ delay circuitry is not needed. In this case, the normal $\overline{WR}$ provided by the 8288 bus controller meets the leading edge write requirement. A diagram is shown in Figure 31. Note that a D-type flip-flop is used to latch $\overline{S2}$. This is important, because during certain 8086 operations, such as execution of a software HALT, $\overline{S2}$ is not guaranteed to remain valid up to the trailing edge of ALE. To overcome this, $\overline{S2}$ is latched on the leading edge of ALE, as done here.

5.4 Graphics Example

All of the applications examples presented thus far are non-specific; that is, all demonstrate how to connect the iRAMs to various microprocessors in the most general terms without regard to the total application. The design that follows shows the 2186/2187 iRAM in a specific application: a color graphics display memory.

In this example (Figure 32), the color display resolution is 65,536 (256 × 256 pixels) × 4 bits. The four bits select the color of the pixel by addressing a color lookup and video priority table. This programmable table permits up to 16 colors (out of 256 possible) per display frame. It also assigns priority. For example, a red disk crosses a green on the display. Does the red cross in front of the green disc, the green in front of the red, or does the area of the overlap become yellow? The priority encoding assigns answers to these questions.

By industry standards, this 256 × 256 pixel display has low-end to medium display resolution. For those unfamiliar with the capabilities at this level, visit a local video game parlor and examine some of the dazzling displays on the state-of-the-art video games such as Williams Electronics Defender. Advanced machines such as this are only beginning to approach this display density.

The iRAM used in this example is the synchronous 2187. Due to the sequential addressing scheme of video displays, video memory typically requires no additional circuitry for refresh. The 2187 is no exception, and in this design the $\overline{REFEN}$ pin is tied high. The sequential scanning by the video address generator automatically refreshes the internal array of the iRAM.

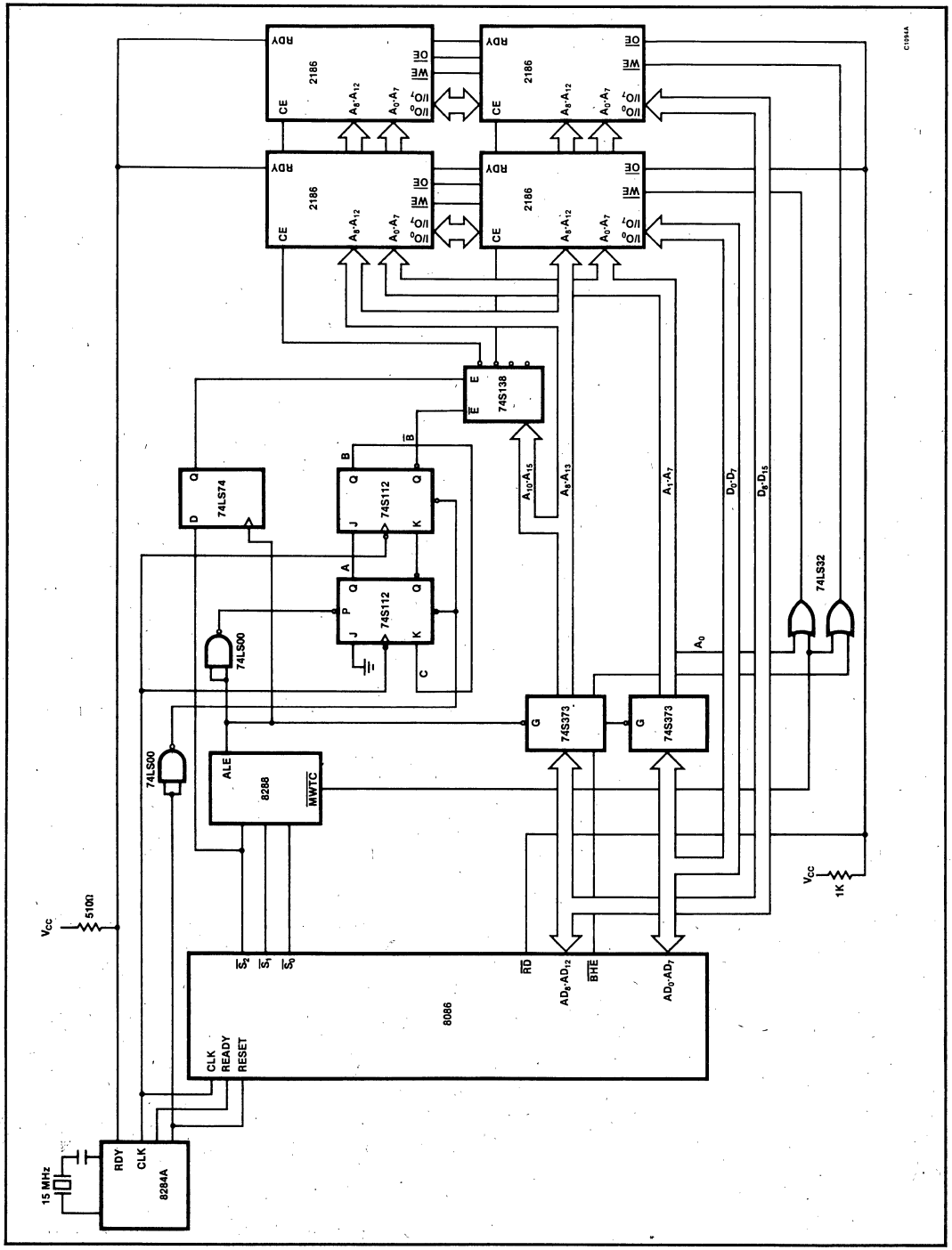


Figure 31. 8086 Max Mode System

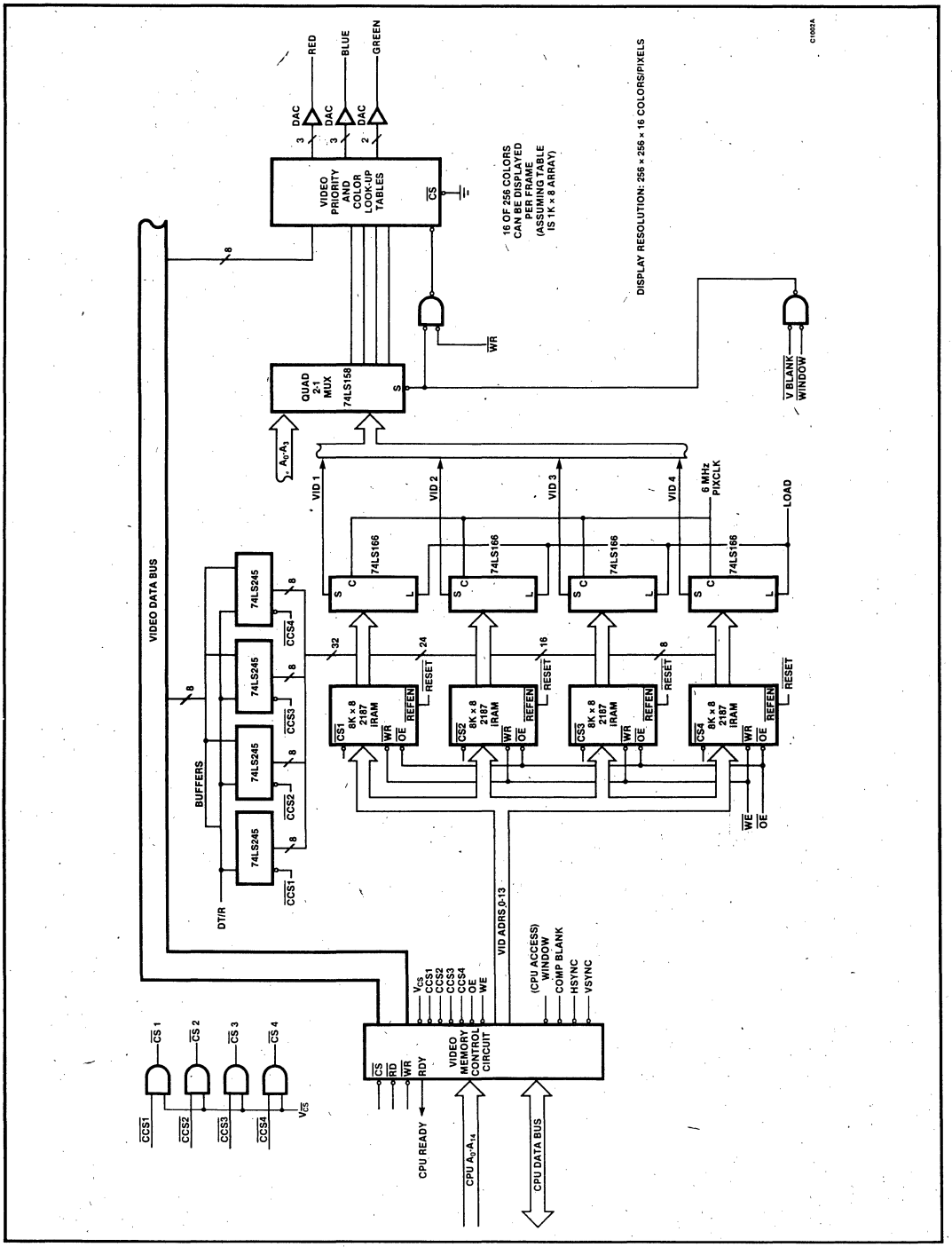


Figure 32. Graphics System

Figure 32 is a simplified diagram. A detailed analysis of the circuit and timing will not be discussed. Briefly, the circuit functions as follows:

CPU addresses A_{14} and A_{15} are decoded to generate one of four iRAM chip selects so that the (assumed 8-bit) CPU can read or write information to the individual memory planes (iRAMs). These chip selects are gated so that all four iRAMs can be simultaneously enabled by the V_{CS} signal from the video timing circuitry. A similar circuit (not shown) would allow $\overline{OE}$ for the iRAMs to be generated by either the CPU or the video timing generator. The iRAM addresses are generated by multiplexing the CPU addresses with video timing addresses. The 32-bit output from the iRAMs is loaded into four 8-bit shift registers and are serially shifted out as four bits of video information used to address the color lookup table. The four lines (Vid_1-Vid_4) are multiplexed with CPU addresses A_0-A_3 to create the actual addresses of the lookup table. Comprised of two 2148H RAMs, the eight data lines of the lookup table are directed to three digital-to-analog converters for generating 16 of 256 different display colors.

Due to the byte-wide organization of the iRAMs, there is plenty of time between video read cycles to allow CPU access to the memory. With a pixel rate of 6 MHz, the byte-wide iRAM has a video read rate of 6/8 MHz or once every 1.33 microseconds. Only 350 ns of this time is needed for a video read cycle. The balance of the time (approximately 1 μsec) can be used by the CPU to access the memory. This interleaving of CPU cycles with video timing cycles, combined with allowing the CPU unrestricted access to the memory during both horizontal and vertical blanking (retrace) periods permits the real time screen update required in an animated display.

5.5 External Refresh Systems

5.5.1 BURST REFRESH

Figure 33 shows an example of a burst mode refresh controller. Timings for this system are shown in Figure 34. To ensure data integrity for a 2187, $\overline{REFEN}$ must be strobed at least 128 times in each 2 ms period. After each high-to-low transition of $\overline{REFEN}$, one cycle time must be allowed before $\overline{REFEN}$ (or $\overline{CE}$) again becomes active.

The system shown in Figure 33 accomplishes refresh by interrupting the processor once each 1.63 ms (200 ns clock period divided by 8192). Upon acknowledgment of this interrupt, $\overline{TEST}$ is driven high, allowing $\overline{REFEN}$ to be generated once every three clock cycles. $\overline{TEST}$ is also routed back to the $\overline{TEST}$ pin of the 8086 to indicate that a burst is in progress. The 8086 samples the state of the $\overline{TEST}$ pin and loops in an idle state until the $\overline{TEST}$ goes low. This is accomplished using the WAIT instruction.

Two 4-bit counters are used to count the $\overline{REFEN}$ pulses. After 128 pulses, the count goes high. On the next rising edge of the system clock, $\overline{TEST}$ is reset to a zero, blocking any further $\overline{REFEN}$ pulses, clearing the counters, and signaling to the processor that the burst is complete. Note that one non-access cycle should be inserted after $\overline{TEST}$ is set low to ensure that sufficient time has been allowed for the last refresh cycle to complete.

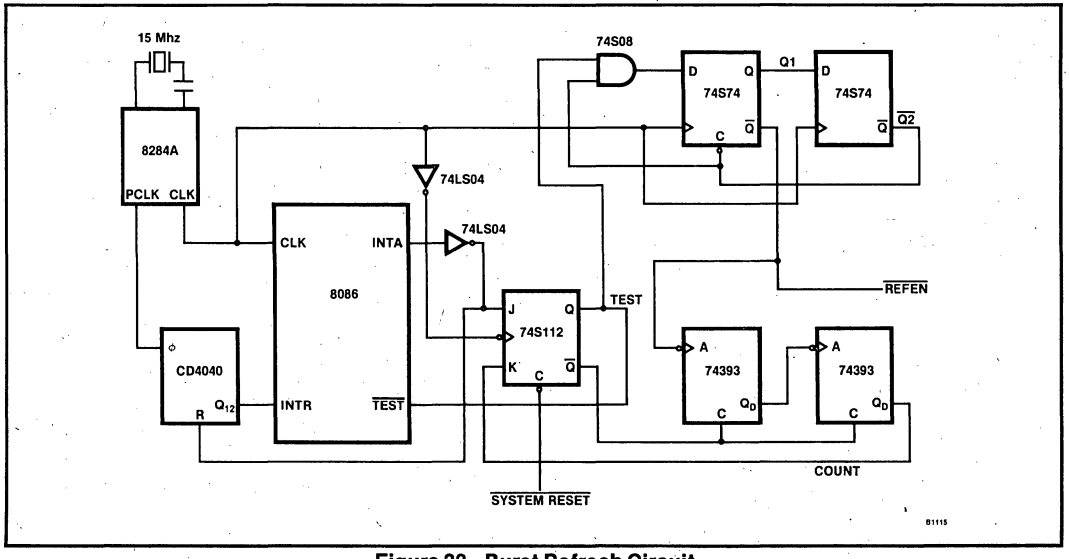


Figure 33. Burst Refresh Circuit

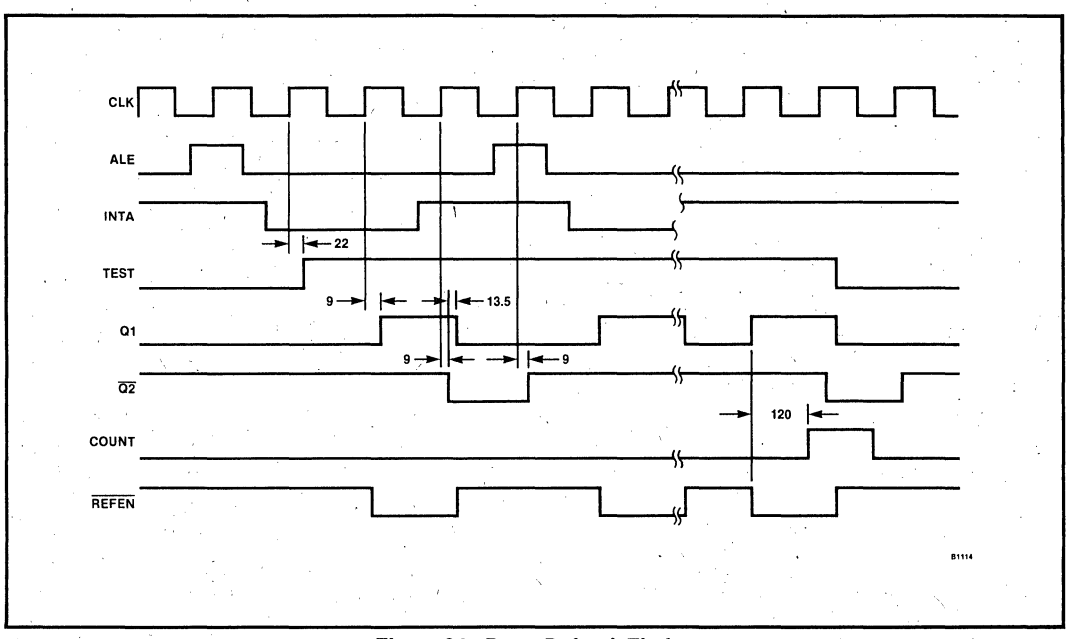


Figure 34. Burst Refresh Timing

5.5.2 SYNC REFRESH SYSTEM

The system in Figure 35 represents one way in which synchronous refresh could be employed using the 2187. In this configuration, memory is divided into four banks, selected via the two least significant addresses. To ensure data integrity, each of the four banks must receive 128 $\overline{\text{REFEN}}$ pulses every 2 ms. In this system if any one bank is accessed, each of the other three banks receives a refresh pulse. Minimum cycle time cannot quite be attained because the cycle time for the refresh cycle is the same as that for an access cycle, and the fact that a one-gate delay exists between $\overline{\text{CE}}$ to one device and the $\overline{\text{REFEN}}$ to the others. At least 16 ns must be added to the minimum cycle time of 425 ns. This number is derived by taking the propagation delay difference between a "fast" 74155 and a "slow" 74155, and adding the maximum delay through a 74S11. This gives the $\overline{\text{CE}}$ to $\overline{\text{REFEN}}$ delay time. This extra delay is not really critical in most systems; the minimum cycle time for a 5 MHz 8086 is 800 ns.

With the circuitry described, data integrity would be jeopardized if one bank were accessed consecutively too many times, since the accessed bank would receive no $\overline{\text{REFEN}}$ pulses. Assuming a 500 ns cycle time, one bank would have to be accessed at least 30 consecutive times to jeopardize data. This is the worst case. In actual operation, consecutive accesses to one bank could be many more than this, as long as operation during any 2 ms period provides 128 $\overline{\text{REFEN}}$ pulses to all banks. Due to the nature of bank selection used (A0:A1 decoding), more than a couple of consecutive accesses to any one bank are highly unlikely.

One caution to note, however, has to do with power-down refresh. If $\overline{\text{REFEN}}$ is kept low for longer than one timer period, the timer will begin to time out. In this event, a period of time (RFHEL) must be allowed before $\overline{\text{CE}}$ can go low again after $\overline{\text{REFEN}}$ returns high. This is to ensure, that if a timer initiated refresh cycle started just as $\overline{\text{REFEN}}$ returned to a high state, it will have time to complete before an access cycle is started.

6 SYSTEM CONCEPTS

6.1 System Reliability

New applications for microprocessor systems appear almost every day. They appear in microwave ovens, automobiles, word processors, home computers, video games, vending machines, lighting controls, medical equipment, etc. The list goes on and on. Failures on these systems cover equally broad ranges: acute annoyance (such as losing your last quarter to the coffee machine), financial loss (a double debit is added to your bank statement by an electric teller machine), and life threatening system failures (the electronic carburator control on your car fails, opening the throttle wide open).

In many applications, reliability is important enough to be designed into the system. The computer memory system is one of the system components for which reliability is important. Also it is one of the few system elements which can be easily designed to enhance its reliability. Since memory system reliability is inversely proportional to the number of devices in the system, a system of a given size should be designed with as few components as possible. For example, a 32K byte system could be designed with sixteen 16K 2118 DRAMs. The system MTBF (Mean Time Between Failures — the "up" time of a system) could be calculated from the combined device soft and hard error rates (See Intel Application Note AP-73 *"ECC #2 Memory System Reliability With ECC"* for a model to calculate system MTBF's). The point is that, whatever the calculated system MTBF, the 2186 will be several times more reliable in a system due to the lower device count.

A few example calculations are tabulated in Table 1. Essentially what is shown is what the maximum acceptable device soft error rate is for a specified system MTBF. For example, if a design using 8K × 8 RAMs requires a memory system MTBF for two years, and the system size is 16K bytes, then the design allows a device with a soft error rate of 3.1%/1K-hrs. The 2186/87 soft error rate goal is more than an order of magnitude better than that! From the chart it can be seen that a 64K byte extra-reliable memory system with a 10 year MTBF requires a device with a soft error rate or 0.15%/1K-hrs. Clearly the 2186/87 family of iRAMs is reliable over the entire spectrum of typical application memory sizes.

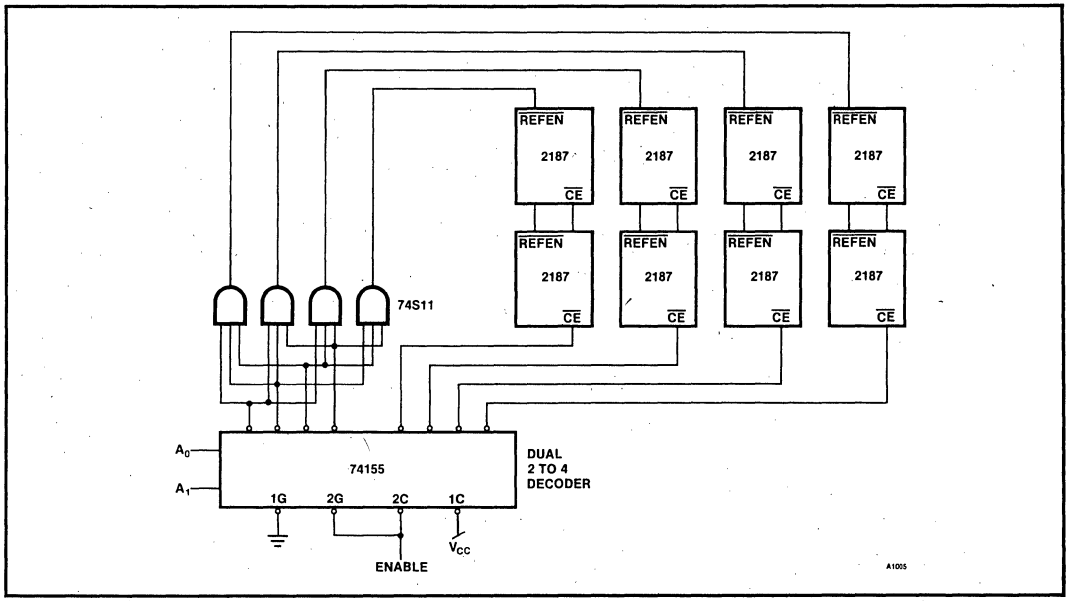


Figure 35. Synchronous Refresh Scheme

Table 1. 2186/87 SER Data

No. of Syst. Rows	No. of Dev.	Sys. Size	Eff. Cycle Time*	Maximum Allowable SER (%/K-Hrs.)			
				1 Yr. MTBF (8800 Hrs.)	2 Yrs. MTBF (17600 Hrs.)	5 Yrs. MTBF (44000 Hrs.)	10 Yrs. MTBF (88000 Hrs.)
1	1	8K	7.00	11.34	5.66	2.25	1.12
2	2	16K	9.66	6.23	3.10	1.22	.60
3	3	24K	11.06	4.29	2.13	.84	.41
4	4	32K	11.93	3.27	1.62	.64	.31
8	8	64K	11.52	1.67	.82	.31	.15

* All times in microseconds
System has a 7μsec device cycle time.
Hard error rate = 0.02%/1K-Hrs.

6.2 Circuit Design Considerations

Integrating components into systems requires a keen awareness of basic concepts on the part of the designer.

Techniques for designing optimal performance memory systems have been thoroughly covered in other literature. Two useful documents that cover these procedures are AP-74 *"High Speed Memory System Design Using the 2147H"* and AP-133 entitled *"Designing Memory Systems for Microprocessors Using the Intel 2164A and 2118 Dynamic RAMs."* There are essentially three areas of major concern in a memory system design:

- Timing delay calculations in the critical path (worst case timing analysis)
- Memory circuit trace layout
- Power distribution and decoupling

The following sections summarize these techniques as they apply to the 2186 and 2187 iRAMs.

6.2.1 DELAY CALCULATIONS

All memory designs require a timing analysis to ensure proper operation and compatibility of the memory and the processor. Timing skews, capacitive delays and propagation delays all have to be accounted for in a proper analysis. Propagation delay design rules for TTL are furnished in the manufacturer's data book. The maximum delay is the data book maximum and the typical delay (usually useless for design) is the data book typical. Intel has determined in work with TTL device manufacturers that the minimum propagation delay is ½ the data book typical value.

Skew is defined as simply the difference between the maximum and minimum propagation delays through devices in a parallel path. Figure 36 is a simple example. Best case propagation of signal A is 6 nanoseconds versus worst case delay of signal B which is 16 nanoseconds. This condition equates to 10 nanoseconds of skew (Figure 37) which adds directly to system access or cycle time. The worst case number of 16 ns would be used for timing analysis in this type of delay calculation; however, often the best case is the most important. For example, as in Figure 38, the skew of concern deals with the best case arrival of a write pulse versus worst case arrival of data to a memory device.

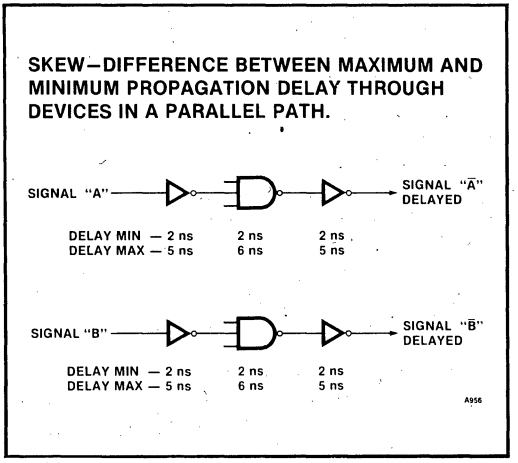


Figure 36. Skew

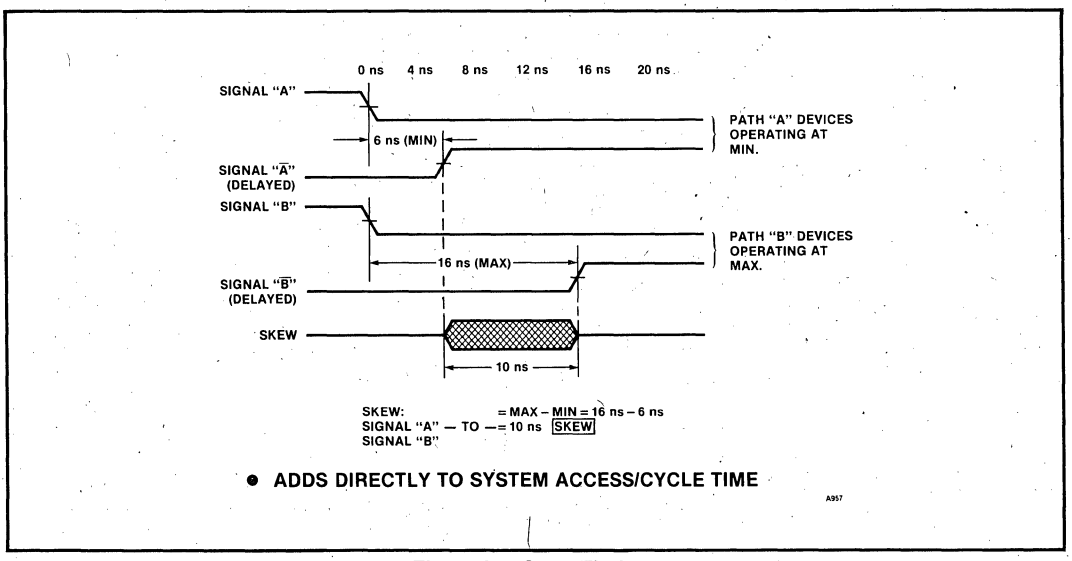


Figure 37. Skew Timing

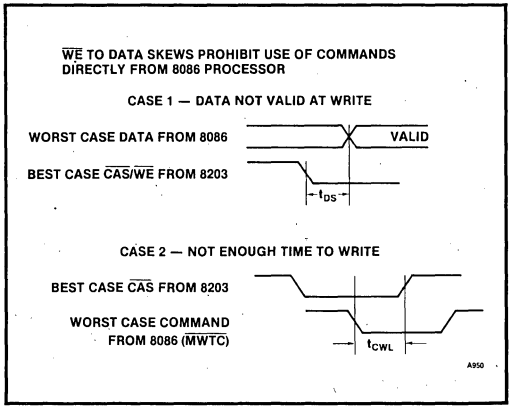


Figure 38. Worst Case Timing

Unbalanced capacitive loading on address or control line drivers also contribute to skew. Capacitance contributes to risetime degradation on these signals. The unbalanced loading causes differing rise times as shown in Figure 39. The different rise times reach a logic threshhold at different times, contributing to skew. In all of these examples, skew contributes to the overall delay, and the goal of the designer is to minimize these skews. A few simple rules will help to achieve this in 2186/87 memory system design:

- Select logic gates for minimum delay per function
- Place parallel paths in the same package (device to device skew is much less within same package - 0.5 ns max for STTL)
- Balance the output loading of device drivers to equalize capacitive delays.

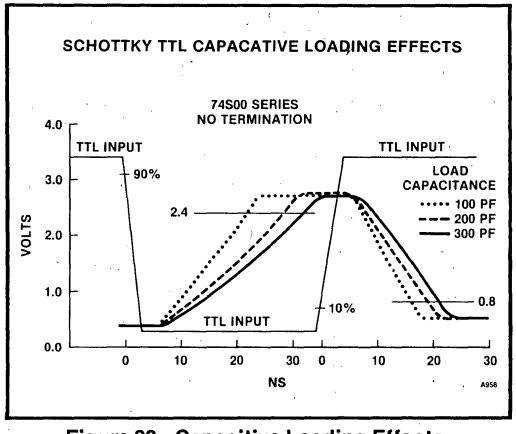


Figure 39. Capacitive Loading Effects

As previously stated, capacitance contributes to signal risetime degradation. To determine the delay due to capacitance, use the following standard derating factors:

Schottky TTL = 0.05 ns/pF
Low Power Schottky TTL = 0.1 ns/pF
Standard TTL = 0.75 ns/pF

Add up all of the capacitance connected to a driver, including the circuit-printed trace capacitance of 2 pF per inch, subtract out the manufacturer's capacitance drive specification, (typically 15 pF) then multiply this capacitance by the derating factor for the driver. This net result is the additional delay due to capacitance. The equation is:

$$D_C = [\Sigma C_{IO} + \Sigma C_{PCB} - C_{SPEC}]T_D$$

where: D_C = delay due to capacitance
ΣC_{IO} = sum of all input/output connections attached to driver
ΣC_{PCB} = 2 pF × number of inches of circuit trace attached to driver
C_{SPEC} = specified drive capacitance of driver
T_D = capacitive derating factor

6.2.2 TRACE LAYOUT

Address lines need to be kept as short and direct as possible. Route address lines in a comb-like fashion from a central location. Routing control and address signals together from a centralized board area will also minimize skew.

Allow for proper termination of all address and control lines because these circuit traces are actually transmission lines. A series resistor close to the driver is the recommended termination technique. Thirty-three ohms is a good typical value, although actual values are usually determined empirically. Figure 40 shows P.C.B. artwork that embodies these rules as well as proper power and ground gridding with decoupling as described in the following section.

6.2.3 POWER SUPPLY DISTRIBUTION AND DECOUPLING

Ground and power busses can contribute to excess noise and voltage drops if not properly structured. The power and ground network do not appear as a pure low resistance element but rather as a transmission line, because the current transients created by the RAMs are high frequency in nature.

Transient effects can be minimized by adding extra circuit board traces in parallel to reduce interconnection inductance. Extrapolation of this concept to its limits will result in an infinite number of parallel traces, or an extremely wide low impedance trace, called a plane. Arranging power and ground voltages by plane provides the best distribution; however, correct gridding can cost effectively approximate the benefits of planar distribu-

tion by surrounding each device with a ring of power and ground traces (Figure 40).

Consider two aspects of the memory device that contribute to power system noise: the active/standby power modes of the RAMs, and the drive requirements of the data I/O buffers. In a typical microprocessor-based system, address space is divided into blocks of RAM, ROM/EPROM, and I/O. When the microprocessor is not accessing a given RAM, the RAM is usually deselected and in a power standby mode. When a previously unselected RAM is selected, a large current surge is experienced. Because the connections supplying power to the device will involve resistance and inductance, a voltage variation will occur in association with the current surge in accordance with the equation:

$$V = Ri + Ldi/dt,$$

where V = instantaneous voltage,
L = inductance,
R = resistance,
and i = instantaneous current

Because a RAM may be selected and deselected hundreds of thousands of times a second, the transient noise generation is significant and must be dealt with during design.

Another factor that contributes to current surges are the drive requirements of the memory devices data I/O buffers. Consider first an I/O buffer outputting a logic one. To accomplish this, the buffer must supply a current to charge the capacitance of the line that it's driving to a logic one level. This operation places a higher current requirement than normal on the V_{CC} bus. Conversely, if the I/O buffer is outputting a logic zero, it must discharge all of the capacitance on the line to ground. This produces a current surge to the ground bus, possibly raising the local V_{SS} potential above ground during the transient.

The solution to this problem is to use a solid plane V_{CC} and ground bus on a P.C. board or use a proper power and ground grid combined with adequate decoupling.

Adequate decoupling is also important in circuit design to minimize transient effects on the power supply system. For best results with the 2186/87, decoupling capacitors are placed on the memory array board at every device location (Figure 40). High frequency 0.1 μF

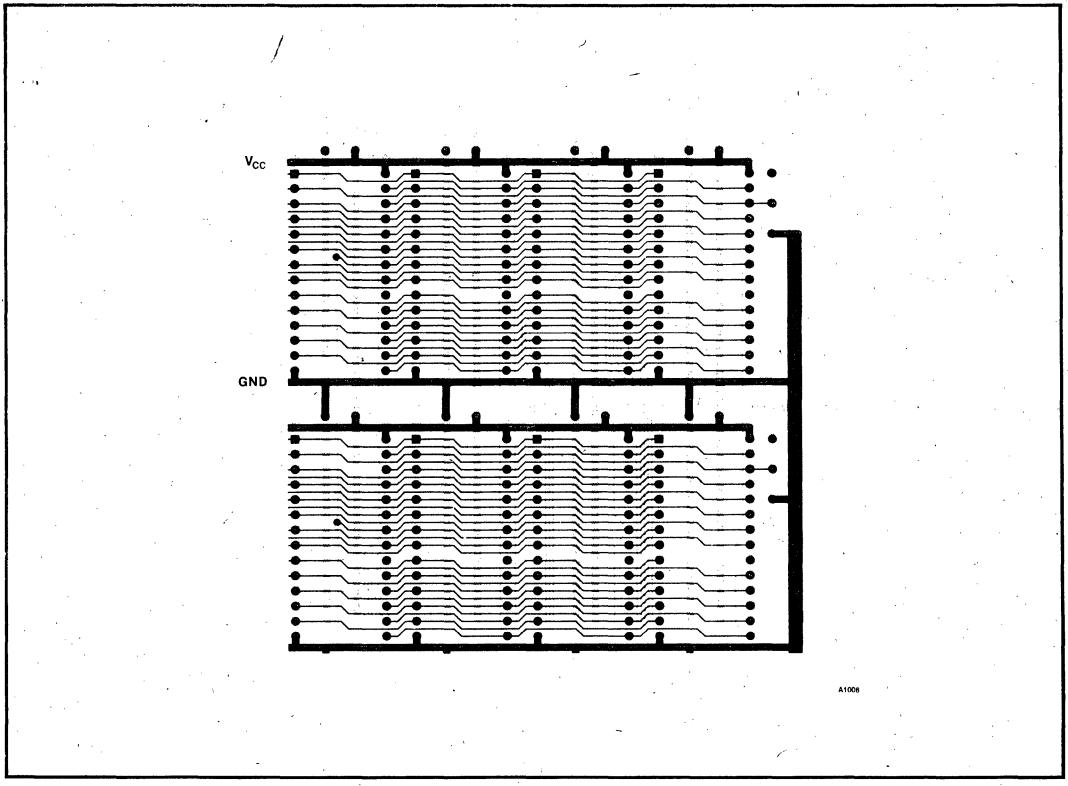


Figure 40. Example of Power and Ground Gridding

ceramic capacitors are the recommended type. Also included should be a large bulk decoupling capacitor in the 50 to 100 μF range, placed where power is supplied to the memory system grid. In this arrangement, each memory is effectively decoupled and the noise is minimized because of the low impedance across the circuit board traces.

7 SUMMARY

Intel's iRAMs provide a new approach to memory design that allows the system designer to take advantage of DRAM density, power consumption, and price without the added cost of designing the refresh control circuitry. The 2186 and 2187 are the premier members of this new byte-wide product family, designed for flexible operation in virtually any microprocessor memory system. By comforming to Intel's universal memory site concept, these iRAMs are compatible with a wide variety of byte-wide memory devices including SRAMs, EPROMs, and E^2PROMs.

In summary, Intel provides another innovative memory product, the 2186/87 iRAMs — basic building blocks for microprocessor memory solutions.

intel

APPLICATION NOTE

AP-133

April 1982

Designing Memory Systems for Microprocessors Using the Intel 2164A and 2118 Dynamic RAMs

William H. Righter
Joseph P. Altnether

Order Number: 210431-001

PREFACE

This application note has been developed to provide the memory system designer with a detailed description of microprocessor memory system design using Intel Dynamic RAMs, the 16K 2118, 64K 2164A, and the 8203 Dynamic RAM Controller. The 8086 bus interface to memory components is described and three major examples are presented and analyzed — ranging from simple to complex: the simple solution, the 5 MHz No-WAIT State and the 10 MHz No-WAIT State systems. To assist the designer, complete logic schematics, timing diagrams and system design considerations are also included in this application note.

1 INTRODUCTION

Matching the correct RAM to microprocessor application requirements is fundamental to effective product design. A good understanding of the advantages and disadvantages of each technological approach and device type will enable a memory system designer to best choose the product that provides the optimal benefit for his particular design objective.

Two basic types of random access memories (RAMs) have existed since the inception of MOS memories: static RAMs (SRAMs) and dynamic RAMs (DRAMs). Where highest performance and simplest system design is desired, the static RAM can provide the optimum solution for smaller memory systems. However, the dynamic RAM holds a commanding position where large amounts of memory and the lowest cost per bit are the major criteria.

The major attributes of dynamic RAMs are low power and low cost — a direct result of the simplicity of the storage cell. This is achieved through the use of a single transistor and a capacitor to store a single data bit (Figure 1).

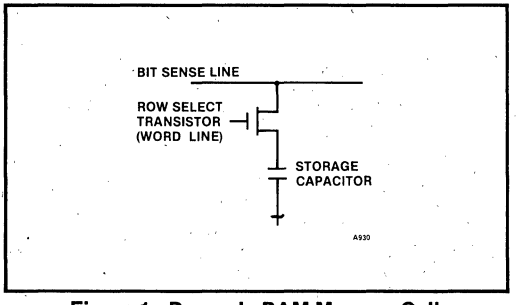


Figure 1. Dynamic RAM Memory Cell

The absence or presence of charge stored in the capacitor equates to a one or a zero respectively. The capacitor is in series with the transistor eliminating the need for a continuous current flow to store data. In addition, the input buffers, the output driver and all the circuitry in the RAM have been designed to operate in a sequentially clocked mode, thus consuming power only when being accessed. The net result is low power consumption. Also, a single transistor dynamic cell as compared to a four or six-transistor cell of a static RAM, occupies less die area. This results in more die per wafer.

Because the manufacturing cost of a wafer is fixed, more die per wafer translate into lower cost. For example, assume a wafer costs $250 to manufacture. Yielding 250 die per wafer means each die costs one dollar. But, if only 125 die are yielded, the cost per die is two dollars. The rationale of the quest for smaller die size is obvious; the simple dynamic memory cell fulfills this quest.

Unfortunately, the simple cell has a drawback: the capacitor is not a pure element and it has leakage. If left alone, leakage current would cause the loss of data. The solution is to refresh the charge periodically. A refresh cycle reads the data before it degrades too far and then rewrites the data back into the cell. RAM organization is tailored to aid the refresh function. As an example, the Intel® 2164A 64K RAM is organized internally as four 16K RAM arrays, each comprised of 128 rows by 128 columns. Consequently the row address accesses 128 columns in each of the four quadrants. However, let's concern ourselves with only one quadrant. Prior to selection, the bit sense line was charged to a high voltage. Via selection of the word line (row addresses) 128 bits are transferred onto their respective bit lines. Electrons will migrate from the cell onto the bit line destroying the stored charge. Each one of the 128 bit lines has a separate sense amplifier associated with it. Charge on the bit line is sensed, amplified and returned to the cell. Each time the RAM clocks in a row address, one row of the memory is refreshed. Sequencing through all the row addresses within 2 ms will keep the memory refreshed.

In spite of the advantages of minimal cost per bit and low power, the dynamic RAM has often been shunned in microprocessor systems. Up until now, dynamic RAMs have required a good deal of complicated circuitry to support the refresh requirements, and associated timing and interfacing needs. Circuitry for arbitration of simultaneous data and refresh requests, for example, has posed significant design problems. These requirements all add to the component count and system overhead costs, both in design and implementation.

The development of the Intel family of dynamic RAM controllers has brought a new level of design simplicity to dynamic RAM memory systems. These new devices include the solutions to the problems of arbitration, timing, and address multiplexing associated with dynamic RAMs.

This application note describes two basic memory systems employing the use of the Intel® 2164A and 2118 dynamic RAMs in conjunction with the Intel® 8203 Dynamic RAM Controller and the Intel 2164A, 64K dynamic RAM with a high speed TTL controller.

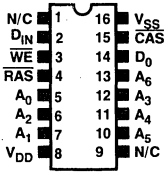
1.1 2118 16K RAM

The Intel 2118 is a high performance 16,384 word by 1 bit dynamic RAM, fabricated on Intel's n-channel HMOS technology. The Intel 2118 is packaged in the industry standard 16-pin DIP configuration, and only requires a single +5V power supply (with ±10% tolerances) and ground for operation, i.e., V_{DD} (+5V) and V_{SS} (GND). The substrate bias voltage, usually

designated V_{BB}, is internally produced by a back bias generator. The single +5V power supply and reduced HMOS geometries result in lower power dissipation and higher performance.

1.1.1 2118 DEVICE DESCRIPTION

The 2118 pin configuration and performance ratings are shown in Figure 2. Note that pins 1 and 9 are N/C (no-connects). This allows for future expansion up to 256K bits in the same device (package). For a rigorous device description, refer to AP-75, "Application of the Intel 2118 16K Dynamic RAM."



	2118-10	2118-12	2118-15
Access Time (ns)	100	120	150
Cycle Time (ns)	235	270	320
Operating Current (mA)	27	25	23
Standby Current (mA)	2	2	2

A931

Figure 2. Intel® 2118 Pinout

1.2.2 2118 ADDRESSING

Fourteen addresses are required to access each of the 16,384 data bits. This is accomplished by multiplexing the addresses onto seven address input pins. The two 7-bit address words are sequentially latched into the 2118 by the two TTL level clocks: Row Address Strobe ($\overline{RAS}$) and Column Address Strobe ($\overline{CAS}$). Noncritical timing requirements allow the use of the multiplexing technique while maintaining high performance. For example, a wide t_{RCD} window ($\overline{RAS}$ to $\overline{CAS}$ delay) allows relaxation of the timing sequence for $\overline{RAS}$, address change, and $\overline{CAS}$ while still permitting a fast t_{RAC} (Row Access Time).

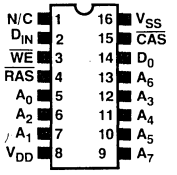
Data is stored in a single transistor dynamic storage cell. Refreshing is required for data retention and is accomplished automatically by performing a memory cycle (read, write or refresh) at all row addresses every 2 milliseconds.

1.2 2164A 64K RAM

The Intel 2164A is a high performance 65,536 word by 1 bit dynamic RAM, fabricated on Intel's advanced HMOS-D III technology. The 2164A also incorporates redundant elements. Packaged in the industry standard 16-pin DIP configuration, the 2164A is designed to operate with a single +5V power supply with ±10% tolerences. Pin 1 is left as a no-connect (N/C) to allow for future system upgrade to 256K devices. The use of a single transistor cell and advanced dynamic RAM circuitry enables the 2164A to achieve high speed at low power dissipation.

1.2.1 2164A DEVICE DESCRIPTION

The 2164A is the next generation high density dynamic RAM from the 2118 +5V, 16K RAM. The 2164A pin configuration and performance ratings are shown in Figure 3. For a detailed device description, refer to AP-131, "Intel 2164A 64K Dynamic RAM Device Description."



	2164A-15	2164A-20
Access Time (ns)	150	200
Cycle Time (ns)	260	330
Operating Current (mA)	55	45
Standby Current (mA)	5	5

A932

Figure 3. Intel® 2164A Pinout

1.2.2 2164A ADDRESSING

Sixteen address lines are required to access each of the 65,536 data bits. This is accomplished by multiplexing the 16-bit address words onto eight address input pins. The two 8-bit address words are latched into the 2164A by the two TTL level clocks: Row Address Strobe ($\overline{RAS}$) and Column Address Strobe ($\overline{CAS}$). Noncritical timing requirements allow the use of the multiplexing technique while maintaining high performance.

Data is stored in a single transistor dynamic storage cell. Refreshing is required for data retention and is accomplished automatically by performing a memory cycle (read, write or refresh) on the 128 combinations of A_0 through A_6 (row addresses) during a 2 ms period. Address input A_7 is a "don't care" during refresh cycles. Thus, designing a system for 256 cycle refresh at 4 ms in a distributed mode automatically provides 128 cycle refresh at 2 ms and a more universal system design.

1.3 Compatibility of the 2118 and the 2164A

In 2118 memory systems designed for upgradability, it is now possible to take advantage of the direct upgrade

path to the 2164A. The common pinout and similarities in A.C. and D.C. operating characteristics of most systems make this upgrade easy and straightforward. A simple jumper change to bring the additional multiplexed address into the memory array, a check for proper decoupling, and the replacement of the 2118's with 2164A's usually completes the job. In the two sections that follow, both device and system level compatibility issues are examined, key parameters are compared, and implications discussed. A data sheet for each device should be handy to aid in understanding the following material.

1.3.1 DEVICE COMPATIBILITY

Both the 2118 and 2164A are packaged in the industry standard 16-pin DIP. Observation of the device's pinout configurations shows that the only difference is the additional multiplexed address address input on pin 9 of the 2164A. This extra input is required to address the additional memory within. Notice the N/C (no connect) on pin 1 of the 2164A. This allows for another direct upgrade path to the 256K DRAM device, with pin 1 used as the next address input. The first and most obvious specifications to compare are the speed and cycle times. Clearly, when discussing compatibility and upgradability the same speed devices must be examined. A glance at the respective data sheets shows that the 2118-15 and the 2164A-15 are the current devices available that are speed and cycle time compatible, and further discussion will center on these two specific device types.

1.3.1.1 D.C. and Operating Characteristics

Both the 2164A and the 2118 function in the same temperature environment (0-70°C) with a single 5 volt ±10% power supply. All signal input voltage level specifications are identical. The input load currents and the output leakage currents are also the same. The operating currents (I_{DD1}, I_{DD2}, I_{DD3}, I_{DD4}) of the 2164A are greater than the 2118 because of the increased density of the 2164A. One other parametric difference worth pointing out is the maximum capacitive load of the control lines on the 2164A. The maximum specification is 8 pF on the $\overline{RAS}$ and $\overline{CAS}$ lines, each respectively 1 pF greater than the 2118.

1.3.1.2 A.C. Characteristics

As mentioned above, the t_{RAC} (access time from $\overline{RAS}$) spec of the 2164A-15 is a perfect match to the 2118-15. Generally, the other A.C. timing specs of the 2164A meet or exceed those of the 2118. Both the read and write cycle times (t_{RC}) of the 2164A-15 are 60 ns less than the 2118. The read-modify-write cycle of the 2164A runs 130 ns faster than the 2118. All parameters in the write cycle (reference 2164A data sheet page 3) of the 2164A exceed those of the 2118-15, as well as those timings specific to the read and refresh cycles. Noteworthy are the t_{RWL} (write command to $\overline{RAS}$ lead time) and t_{CWL} (write command to $\overline{CAS}$ lead time) specifications of the 2164A. These are 60 ns less than those of the 2118, allowing more flexibility in timing generation of the write cycle. One other improvement is t_{PC} (page mode read or write cycle) which is 125 ns. This parameter allows, for the first time, a two-fold performance advantage for page mode called extended page mode. This is offered as an option to read or write an entire page (row) of data during a single $\overline{RAS}$ cycle. By providing a fast t_{PC} and long $\overline{RAS}$ pulse width (t_{RPM2}), the 2164A-15 S6493 permits high-speed transfers of large blocks of data, such as required in bit-mapped graphics applications.

There are a few of the 2164A timing specifications however, that exceed those of the 2118. These are:

t_{CAC} (access from $\overline{CAS}$) = 85 ns, 5 ns greater than 2118

t_{RAH} (row address hold time) = 20 ns, 5 ns greater than 2118

t_{CAH} (col address hold time) = 25 ns, 5 ns greater than 2118

t_{RCD} ($\overline{RAS}$ to $\overline{CAS}$ delay time) = 30 to 65 ns, versus the 2118, 25 to 70 ns

Usually only the t_{RAH} specification has significance in system applications. This and all other system level compatibility issues are discussed in the following section.

1.3.2 SYSTEM LEVEL COMPATIBILITY

When designing a new system, the current (I_{DD}) requirements of the 2164A do not present any particular problems. Simply proceed with the normal power requirement analysis, and specify the power supply accordingly. (A method for determining memory system power requirements is detailed in Intel application note AP-131 titled: *Intel 2164A 64K Dynamic RAM Device Description.*) In a system being upgraded with 2164A devices, check the new power supply reqirements against the current power supply specifications to insure compatibility. Worth pointing out is the fact that in a 2118 system arranged as 64K by 16-bit word (32 devices) the power/bit of the 2118-15 is 2.6 microwatts/bit (see AP-75, pp. 11-12). Replacing the 2118's with 2164A DRAMS creates a 256K by 16-bit word (again, 32 devices) and the power per bit is 1.33 microwatts/bit (see AP-131, pp. 11-12). The quadrupling in memory size does not quadruple power supply requirements.

For a 64K by 16-bit to 256K by 16-bit conversion, the additional power required is 2.89 watts. (5.59 watts for the 2164A system — 2.7 watts for the 2118 system). On

the other hand, to build a 64K by 16-bit system with 2118 requires 2.7 watts versus only 1.4 watts for the 2164A, meaning that for a given system size, there is a significant system power system savings by implementing the design with the 2164A.

The difference in current (I_{DD}) specifications leads to another system consideration, that of decoupling. The larger current transients generated as a dynamic RAM internally powers up as a response to refresh cycles or active cycles requires decoupling to keep noise off the power grid and to prevent a transitory local voltage drop across devices. Specifics of calculating local and bulk decoupling requirements are presented in Section 6.3.4, but in general Intel recommends .1 μF high frequency ceramic capacitors for every 2164A device, and 100 μF bulk decoupling for every 32 devices.

In comparison to the 2118, the $\overline{RAS}$, $\overline{CAS}$ lines of the 2164A RAM have 1 pF additional load. This seems trivial on a device level, but in a system the extra capacitance adds approximately .1 ns/pF propagation delay (assuming low power Schottky drivers) to the overall system access path. With 16 devices per driver, this extra load adds up to a measurable increase in propagation delay. Determining additional delay due to capacitance is standard engineering practice in a new design. When upgrading a current memory system with 2164A DRAMs, the additional delay also has to be considered. Refer to section 6.2 for the formula to determine if the additional loading is a concern in any specific application.

Of the four timing specifications where the 2164A-15 exceeds the 2118-15 usually only t_{RAH} specification is of concern. If, however, the system being upgraded is $\overline{CAS}$ access limited rather than $\overline{RAS}$ access, then check the timing to determine if the extra 5 ns on t_{CAC} will require system re-tuning. The column address hold specification (t_{CAH}) needs also be checked in this case. In the majority of DRAM systems, the access speed of importance is t_{RAC}, the $\overline{RAS}$ access time. When optimizing a memory system to achieve the design's fastest access time, set the t_{RCD} spec to a value less than t_{RCD} maximum. In these high performance systems, be sure that the tighter 5 ns in the 2164A t_{RCD} spec window doesn't push out the system access time by that amount, or if it does, that it still conforms to the system timing requirements.

Reliablity qualification data for the 2164A and 2118 are identical with projections of less than .1%/1K-hrs for soft errors caused by α particles and less than .02%/1K-hrs for hard failures. This leads to a distinct system reliability advantage of the 2164A over the 2118. System reliability is qualified as MTBF (mean time between failure). This is the "up-time" of the system and is defined as $1/n\lambda$ where n is the number of devices in the system, and λ is the device failure rate. This equation (MTBF = $1/n\lambda$) says that system reliability is inversely proportional to the number of devices in the system. Therefore, a 1 Megabyte system (or any given system size) built with 2164A devices is four times more reliable as one built with 2118s.

In summary, when upgrading a system to 64K devices, increase the decoupling, check the power supply, and tweak the timing only if necessary, then enjoy the improved system reliability. When engineering a new design, become familiar with and be aware of the specification differences between the 2118 and the 2164A.

2 MICROPROCESSOR SYSTEM

To effectively design a microcomputer memory, an understanding of both the RAM and the microprocessor is necessary. Since Intel microprocessors have been well-documented in other publications, this applications note will mainly focus upon operation during bus cycles as related to the memory interface.

2.1 iAPX 86 Bus Operation

The iAPX 86 bus is divided into two parts: control bus and time-multiplexed address data bus. The bus is the microprocessor's only avenue for dialog with the system. The processor communicates with both the memory and I/O via the bus. As a result, it must necessarily differentiate between a memory cycle and an I/O cycle. In the minimum mode, this differentation is accomplished with the signal M/$\overline{IO}$ which remains valid during the entire cycle. Therefore, this signal need not be latched. In the maximum mode, the processor commits to a bus cycle by means of three status bits transmitted to the bus controller which generates the control signals.

The bus cycle is divided into four times, referred to as t-states, independent of the mode. Duration of this t-time (t_{CLCL}) is the reciprocal of the clock frequency into the microprocessor. During each of these states, a distinct suboperation occurs. In t_1, the address becomes valid and the system is informed of the type of bus cycle, memory or I/O. In addition, a clock called ALE (Address Latch Enable) is generated to enable the system to latch the address. This is required because the address will disappear in anticipation of data on the bus. Intended to strobe a flow-through latch, ALE becomes active after the address is valid and deactivated prior to the address becoming invalid. At the end of t_2, the Ready input is sampled. If it is low, the processor will "idle," repeating the t_3 state until the Ready line is high, allowing the memory or I/O to synchronize with the microprocessor. In t_3, the read or write operation commences and the high order status bits become valid.

Finally in t_4, the machine cycle is terminated; input data is latched into the processor in a read cycle or in a write cycle output data disappears. The relationship of the signals for minimum and maximum modes are shown in Figures 4 and 5. Exact timing relationships will be developed throughout the text. The design problem involves making the microprocessor signals intelligible to the dynamic RAM.

The timing analysis is to be given with a read cycle for the minimum mode configuration of Figure 6. Unlike static RAMs which access from whenever every input signal is stable, dynamic RAMs begin a cycle on a clock edge after addresses are stable. This will introduce a certain amount of delay in the logic path. The exact amount depends on the complexity of the memory controller. Two paths to access will be considered: first, the control signal to data input and second, address stable to data input. In the read cycle there are four control signals; M/$\overline{IO}$ used to differentiate memory and I/O cycles, $\overline{RD}$ used to control the output enable, DT/$\overline{R}$ and $\overline{DEN}$ are controls for data flow. Of these only M/$\overline{IO}$ is a concern to the memory design. Without WAIT states, no cycle can be longer than four clock periods.

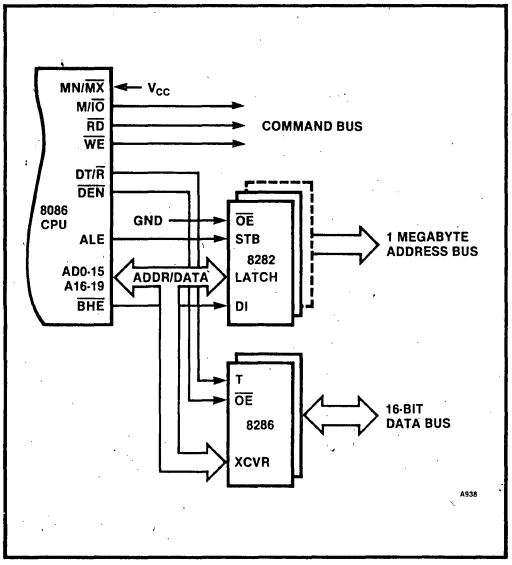


Figure 6. Minimum Mode Operation

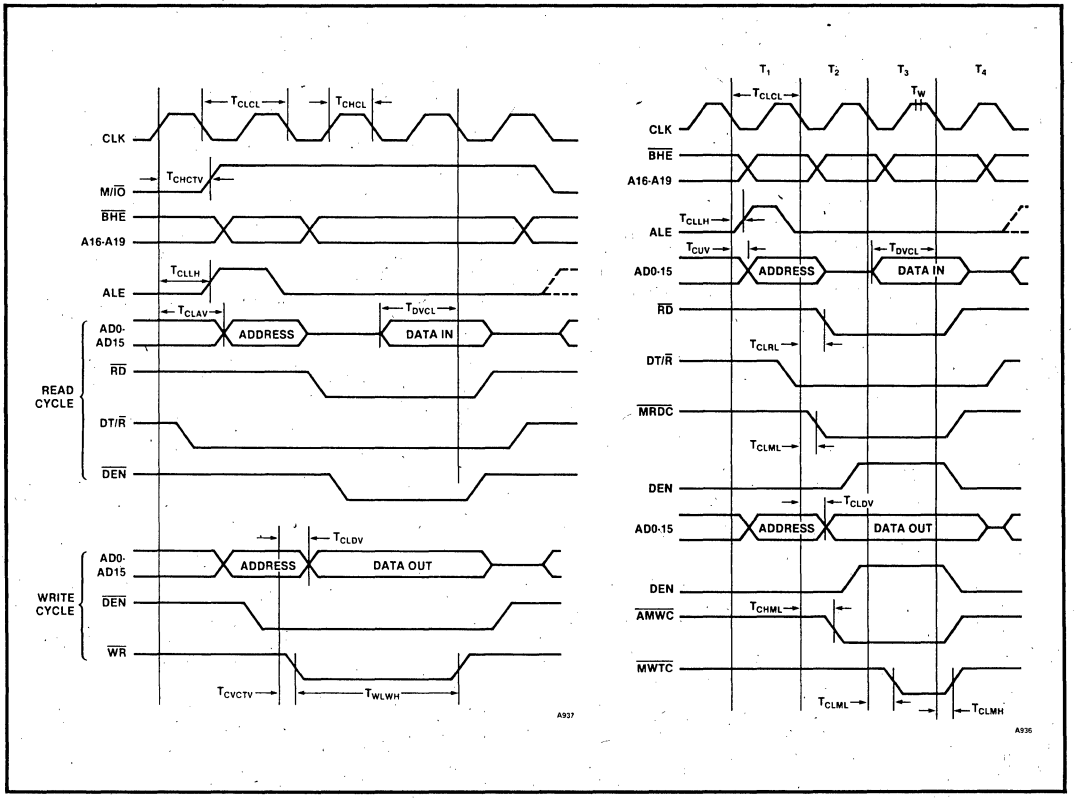


Figure 4. 8086 Bus Timing — Minimum Mode and Figure 5. 8086 Bus Timing — Maximum Mode

Referring to Figure 5, the following is obtained:

$MEMCY \leq 4\, t_{CLCL}$

M/$\overline{IO}$ is stable t_{CHCTV} from the previous clock high time t_{CHCL}, but;

$t_{CHCL} = 1/3t_{CLCL} + 2$

For the 5 MHz clock, $t_{CLCL} = 200$ ns

solving for t_{CHCL},
$t_{CHCL} = 68$ ns
But t_{CHCTV} is 110 ns.

As a result, M/$\overline{IO}$ is a stable worst case 32 ns after the start of a memory cycle. For an 8086-2, t_{CLCL} is 125 ns and t_{CHCTV} is 60 ns. Similarly, M/$\overline{IO}$ is stable 17 ns after the start of the cycle.

Address calculations must include the buffer delay (Figure 7). Stable addresses from the processor are available t_{CLAV} into the cycle and ALE is active t_{CLLH} into the cycle (Table 1). Addresses are on the bus t_{CLAV} plus t_{IVOV} (latch delay) or t_{CLLH} plus t_{SHOV} (buffer delay from strobe). The worst case number (t_{ADDR}) is the greater of these two numbers.

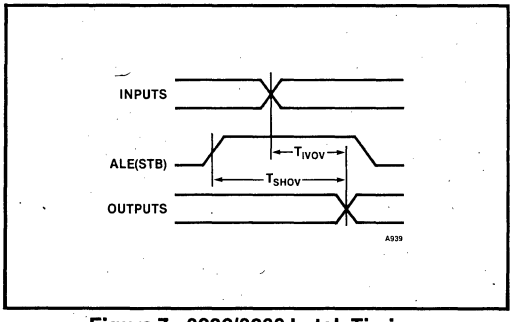


Figure 7. 8282/8283 Latch Timing

Table 1. Address Latch Delays — Min Mode

	5 MHz		8 MHz		10 MHz	
t_{CLAV} (ns) =	110		60		50	
t_{IVOV} (ns) =	+ 22		+ 22		+ 22	
Flow Thru (ns) =	132		82		72	
t_{CLLH} (ns) =		80		50		40
t_{SHOV} (ns) =		+ 40		+ 40		+ 40
Latch Delay (ns) =		120		90		80

Flow through delay is the limiting factor of the 5 MHz system, whereas delay from the latch strobe (ALE) is the limiting factor in the fastest processors. Finally, data must be inputted t_{DVCL} plus t_{IVOV} to the data buffer prior to the fourth t-state. Access from stable addresses is:

$$t_{ACC} = 3t_{CLCL} - t_{ADDR} - (t_{DVCL} + t_{IVOV})$$

Using this equation and the results from Table 1, t_{ACC} can be calculated.

Table 2. t_{ACC} Calculations — Min Mode

	5 MHz		8 MHz		10 MHz	
$3t_{CLCL}$ (ns) =		600		375		300
t_{ADDR} (ns) =	132		90		80	
t_{DVCL} (ns) =	+ 30		+ 20		5	
t_{IVOV} (ns) =	+ 22		+ 22		+ 22	
SUBTOTAL (ns) =	184	− 184	132	− 132	107	− 107
t_{ACC} (ns) =		416		243		193

Table 3 shows the system access time from stable address to input data required. This time is the summation of the RAM access time plus the control logic delay time.

Table 3. Data Setup Time — Min Mode

	5 MHz	8 MHz	10 MHz
t_{CVCTV} (ns) =	110	70	50
t_{CLDV} (ns) =	− 110	− 70	− 50
t_{IVOV} (ns) =	− 22	− 22	− 22
t_{DS} (ns) =	− 22	− 22	− 22

During a write cycle, access is not the issue, but the write pulse width, the data setup and hold time with respect to the write pulse are of concern. The pulse width is simply t_{WLWH}, while data set-up time must be calculated from a clock edge. Dynamic RAMs latch input data on the falling edge of the write enable pulse, so the calculation is critical. Data is valid t_{CLDV} plus the buffer delay t_{IVOV} in t_2 while the write pulse begins t_{CVCTV} in t_2. Worst case condition is a skew such that t_{CLDV} is a maximum delay while t_{CVCTV} has a minimum delay.

$$t_{DS} = t_{CVCTC} - (t_{CLDV} + t_{IVOV})$$

From the calculations in Table 3, the leading edge of the write pulse must be delayed in the minimum mode. These calculations will be used later.

Having examined the major timing parameters of the minimum mode configuration, let's now check the maximum mode timings.

In the maximum mode configuration of Figure 8, the system has another component — an 8288 bus controller — which generates ALE and the read and write control signals. In this configuration a memory read cycle is not committed until t_{CLML} into t_2 whereas in the minimum mode operation, the information was known in t_1. In this respect, a maximum mode system access cycle is less than $3t_{CLCL}$.

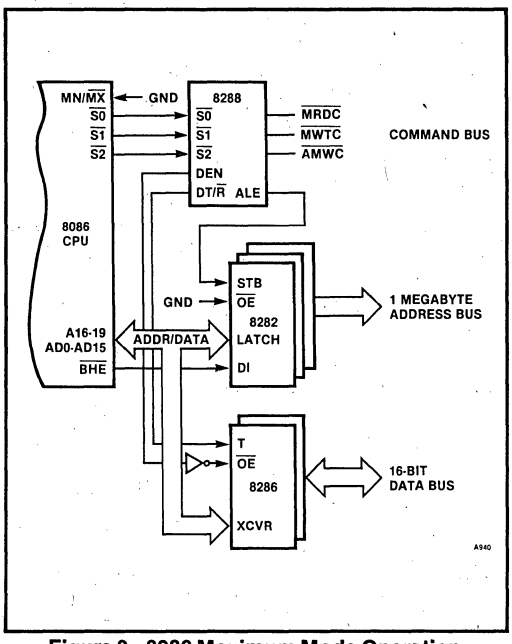


Figure 8. 8086 Maximum Mode Operation

To determine address delay, we will, again, examine the data flow path and the delay from the latch opening. The greater of these two numbers is the worst case time delay (t_{ADDR}).

$$\text{Flow-thru Delay} = t_{CLAV} + t_{IVOV}$$
$$\text{Latch Delay} = t_{CLLH} + t_{SHOV}$$

Using these equations and previous data, Table 4 shows how Flow-thru Delay can be calculated.

Table 4. Flow-through Delay — Max Mode

	5 MHz		8 MHz		10 MHz	
t_{CLAV} (ns) =		110		60		50
t_{IVOV} (ns) =		+ 22		+ 22		+ 22
Flow Thru Delay =		132		82		72
t_{CLLH} (ns) =	15		15		15	
t_{SHOV} (ns) =	+ 40		+ 40		+ 40	
Delay from ALE =	55		55		55	

In each case in Table 4, the limiting delay is flow-thru-time. Access time from address can now be determined. Again, data must be valid t_{DVCL} plus the input buffer delay (t_{IVOV}) before the end of t_3. For maximum mode access from the address valid time is:

$$t_{ACC} = 3t_{CLCL} - t_{ADDR} - (t_{DVCL} + t_{IVOV})$$

Using this equation and previous data (Table 4), Table 5 shows how t_{ACC} can be calculated.

Table 5. t_{ACC} Calculations — Max Mode

	5 MHz		8 MHz		10 MHz	
$3t_{CLCL}$ (ns) =		600		375		300
t_{ADDR} (ns) =	132		82		72	
t_{DVCL} (ns) =	+ 30		+ 20		5	
t_{IVOV} (ns) =	+ 22		+ 22		+ 22	
SUBTOTAL (ns) =	184	− 184	124	− 124	99	− 99
t_{ACC} (ns) =		416		251		201

Access from the read command ($\overline{MRDC}$) must also be determined. $\overline{MRDC}$ is valid t_{CLML} from t_2, causing access (t_{CA}) from $\overline{MRDC}$ to be:

$$t_{CA} = 2t_{CLCL} - t_{CLML} - (t_{DVCL} + t_{IVOV})$$

Using this equation, Table 6 shows the access calculations.

Table 6. Access From Memory Read Command

	5 MHz		8 MHz		10 MHz	
$2t_{CLCL}$ (ns) =		400		250		200
t_{CLML} (ns) =	35		35		35	
t_{DVCL} (ns) =	+ 30		+ 20		5	
t_{IVOV} (ns) =	+ 22		+ 22		+ 22	
SUBTOTAL (ns) =	87	− 87	77	− 77	62	− 62
t_{CA} (ns) =		313		173		138

Access from the memory read command ($\overline{MRDC}$) is much more stringent than address access. Consequently both access paths must be consideed in system design. The write cycle has the same limitation as access from memory read command. Memory write is identified by $\overline{MWTC}$ having the same timing as the memory read command. Address timing is the same for both the read and write cycles. The write pulse, t_{WP} is generated by $\overline{MWTC}$ with a pulse width of one clock cycle plus maximum t_{CLML} plus the minimum overlap into the next cycle (t_{CLMH}).

$$t_{WP} = t_{CLCL} + t_{CLML} - t_{CLMH}$$

For the 5 MHz, 8 MHz, and 10 MHz system, t_{WP} is calculated as shown in Table 7.

Table 7. t_{WP} Calculations — Max Mode

	5 MHz		8 MHz		10 MHz	
t_{CLCL} (ns) =		200		125		100
t_{CLML} (ns) =	35		35		35	
t_{CLMH} (ns) =	− 10		− 10		− 10	
SUBTOTAL (ns) =	25	− 25	25	− 25	25	− 25
t_{WP} (ns) =		175		100		75

Data setup time (t_{DS}) to the leading edge of the write pulse occurs approximately one t_{CLCL} time later. From t_{CLCL}, the maximum t_{DVCL} plus the minimum t_{CLML} must be subtracted:

$$t_{DS} = t_{CLCL} - (t_{CLDV} + t_{CLML})$$

Now t_{DS} can be computed as shown in Table 8 by using data from previous calculations and the data sheet.

Table 8. Data Setup (t_{DS}) Calculations — Max Mode

		5 MHz	8 MHz	10 MHz
t_{CLCL} (ns)	=	200	125	100
t_{CLDV} (ns)	=	−110	− 60	− 50
t_{CLML} (ns)	=	− 10	− 10	− 10
t_{DS} (ns)	=	80	55	40

Using $\overline{MWTC}$ as the write pulse allows sufficient data set-up time for the dynamic RAMs. These, then, are the basic timing equations for the system of Figures 6 and 8. They are general in that timing requirements for different clock frequencies (i.e., 9 MHz) can be calculated using them. Armed with these equations, the designer can now shape the control and address signal in the time domain with a memory controller to meet the dynamic RAM requirements.

In addition to converting address, $\overline{MRDC}$ and $\overline{MWTC}$ into $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, etc., to satisfy both the processor and memory, another task called refresh must be performed by the memory controller.

Performing the interface translation, providing refresh and controlling the signal timing to the RAM requires a controller that consists of six elements as shown in Figure 9. Of these, the most basic is the oscillator because it fulfills two functions: providing a time base for refresh interval timing and establishing precise times for $\overline{RAS}$, $\overline{CAS}$, etc., to the RAM. The operating frequency must be high enough to provide sufficient increments between timing signals. The relationship of timing signals will be multiple periods of the clock frequency. In addition, the oscillator drives a countdown or divide by N circuit to measure the time between refresh cycles. Refresh can be either burst or distributed. In the burst mode, a refresh request would occur once every two milliseconds to meet the dynamic RAMs' needs. For a 16K or 64K RAM with 128 refresh cycle requirement, all 128 refresh cycles would be performed consecutively. A disadvantage of this method is that the memory is "out of service" for a long period of time. Assume a 350 ns cycle time, then the time required to perform refresh is 350 ns multiplied by 128 cycles or 44.8 microseconds operating with a 5 MHz 8086; this translates to 224 consecutive WAIT states.

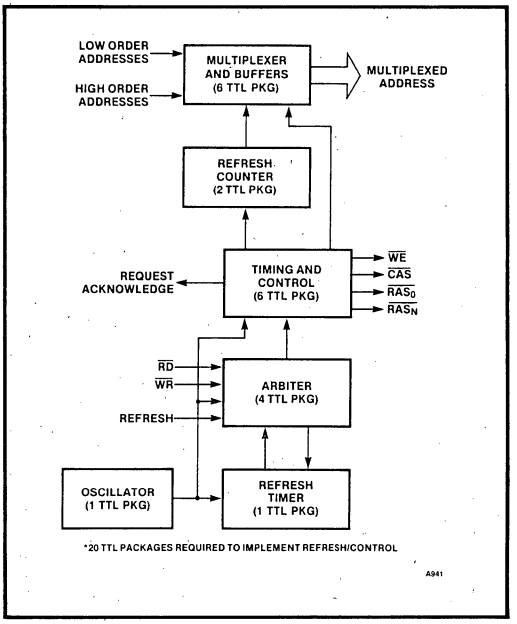


Figure 9. Refresh Timing and Control Block Diagram

Consequently, a large delay is injected every 2 ms. On the other hand, distributed refresh steals a single cycle, 128 times periodically throughout the 2 ms. Evenly distributed, a refresh cycle occurs once every 15 microseconds. Again assume a 350 ns refresh cycle, and our 5 MHz system need only inject two WAIT states (worst case) each time. Thus distributed refresh is preferable in almost all microprocessor systems.

Guaranteeing that all 128 refresh addresses are exercised is the task of the refresh address counter. It consists of an eight-stage binary counter. After the refresh cycle has been completed, the counter is advanced one count. Incrementing after refresh eliminates any concern regarding address settling or setup time as the counter outputs are changing. This would be a concern if the counter were incremented as the refresh cycle started.

Because the counter cycles through all 128 addresses every 2 milliseconds, it isn't required to be in a specific state after power on, i.e., it need not start at address 0 after power on.

Next is the arbiter — which can be the bane of every memory design. Deciding whether a memory cycle is an access cycle or a refresh cycle is the function of the arbiter. Refresh requests are derived from the oscillator which operates asynchronously with the system clock. The arbiter will grant the request when a refresh request is made and no memory cycle is occuring or pending. If

an access cycle is in progress, the arbiter must inhibit the refresh cycle until the current cycle is completed. The same logic process occurs if a refresh cycle is in progress and access is requested. This sequence flows smoothly most of the time. The difficulty arises when refresh and access are requested simultaneously. In every arbiter there exists an infinitely small but very real time period when the arbiter cannot make a decision, much less the correct one. Consider the arbiter in Figure 10 — a simple cross-coupled NAND or an R-S flip-flop.

If both requests are made simultaneously, both would be granted — an impossibility!

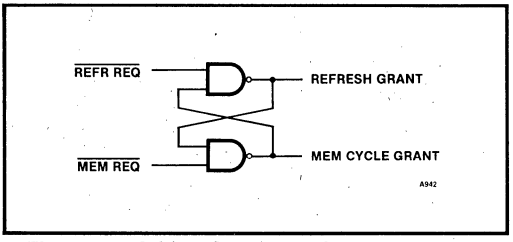


Figure 10. Arbiter Cross-coupled NAND Gates

Another arbiter frequently used is a D-type flip-flop as in Figure 11. Here arbitration is attempted between the clock and the D input. Violating the setup or hold time with respect to the clock can cause the output to enter a quasi-stable state of non-TTL levels for as long as 75 ns. This timing is too long for many high performance systems.

Effective solutions have reduced performance to maximize reliability. One such method is a two stage clocked flip-flop per Figure 12.

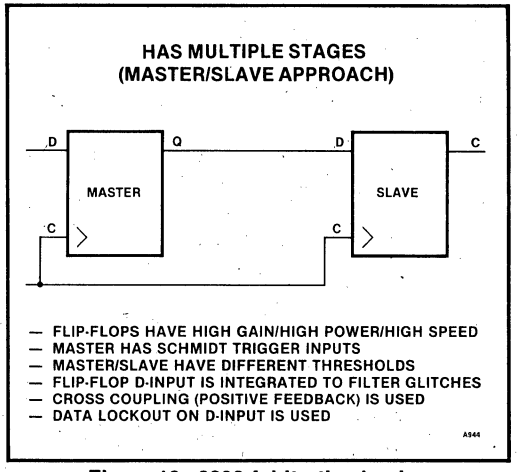


Figure 12. 8203 Arbitration Logic

In this configuration arbitration is performed at the second stage so that even if the first stage "hangs" all will be settled by the clocking of the second stage.

The timing and control section is the core of the controller. Under its guidance, addresses are switched for multiplexing. $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ are produced and sequenced in a fashion understandable by the RAMs. One other fea-

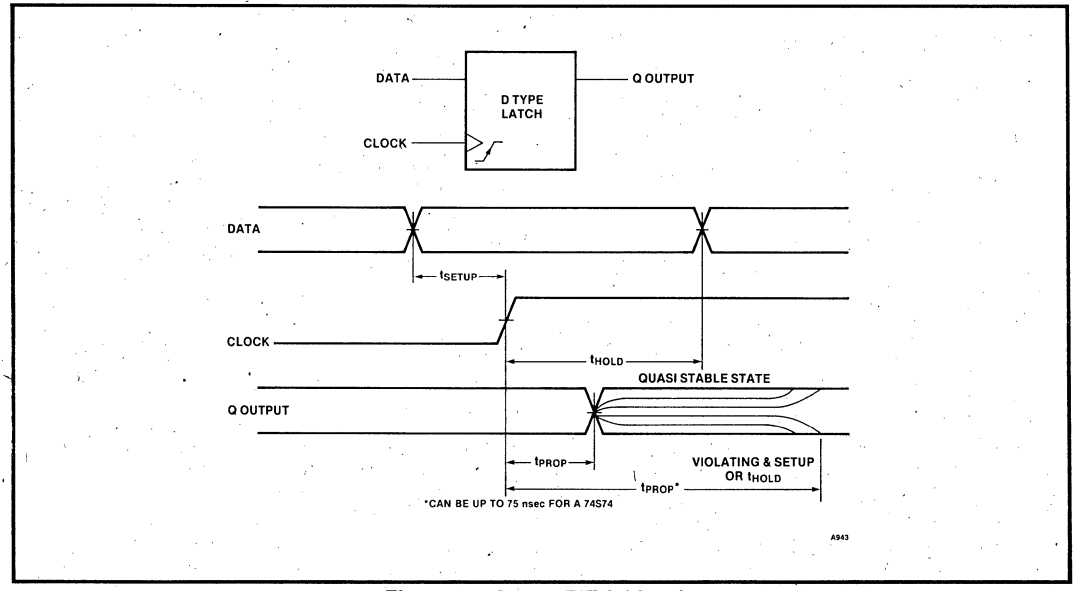


Figure 11. D-type F/F Arbitration

ture required is a handshake signal with the processor to indicate whether or not the memory is ready to be accessed. This is usually implemented with a System Acknowledge ($\overline{SACK}$) (an early signal in the cycle) which indicates a receipt by the controller of a memory access request, or by a Transfer Acknowledge ($\overline{XACK}$, a signal occuring later in the memory cycle), indicating the valid memory data is available.

The final piece of the memory controller is the address multiplexers and buffers to drive the memory addresses. During the normal memory cycle the parallel addresses from the bus must be reduced by one half through time multiplexing. In addition refresh addresses must be applied to the array through this same address path. Buffers are shown to drive the capacitance of the array with signals having sharp rise and fall times.

Figure 9 also shows the quantity of TTL packages required to implement such a controller. Twenty TTL packages are usually required for a controller.

To design a controller with discrete TTL components can take several man months of design effort. Typically, four weeks for design, two weeks for timing analysis, four weeks to build and debug prototypes, six weeks for circuit board layout, and another four weeks to add additional features or to tweak the original design. Obviously, the Intel 8203 DRAM controller is a desirable alternative.

2.2 8203 Dynamic RAM Memory Controller

The Intel 8203 is a Schottky bipolar device housed in a 40-pin dual in-line package. It provides a complete dynamic RAM controller for microprocessor systems and expansion memories. All of the system control signals are provided to operate and refresh the 2117, 2118 and 2164A dynamic RAMs. To accomplish this, the 8203 provides the following features:

- Directly addresses one-half megabyte of 2164A (with external drivers)
- Provides address multipexing and $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ strobes
- Provides a refresh timer and an 8-bit refresh address counter
- Refresh may be internally selected for automatic refresh in a distributed fashion
- Refresh may be externally requested to provide for synchronous or transparent refresh
- Compatible with Intel 8080A, 8085A, iAPX 88 and iAPX 86 families of microprocessors
- Provides system acknowledge and transfer acknowledge signals
- Allows asynchronous memory and refresh cycle requests
- Provisions for external clock or crystal oscillator

A block diagram of the 8203 is given in Figure 13 which illustrates how these features are integrated.

2.2.1 OSCILLATOR

The Intel 8203 generates its timing from an internal shift register which is crystal controlled. This method provides highly accurate control of the timing required for dynamic RAMs. This method is superior to a monostable mulitvibrator approach where transients and unit-to-unit timing accuracies are difficult to control.

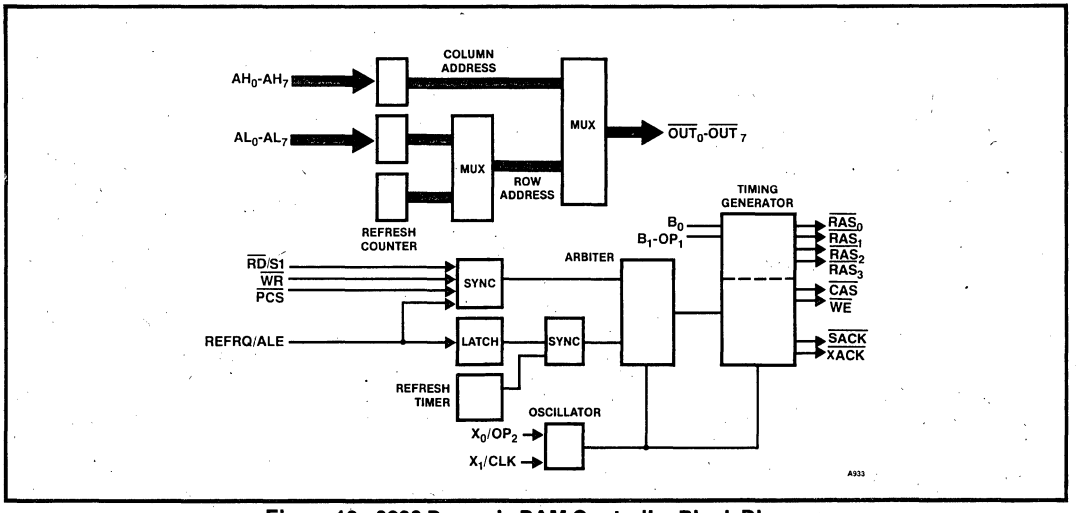


Figure 13. 8203 Dynamic RAM Controller Block Diagram

2.2.2 ARBITER

The arbiter resolves all conflicts between any cycles that are requested simultaneously. These cycles can be generated from one of four places:

1. Read Cycle Request — $\overline{RD}$/S1 input
2. Write Cycle Request — $\overline{WR}$ input
3. External Refresh Request — REFRQ/ALE
4. Internal Refresh Request — (refresh timer shown in Figure 13)

If a refresh cycle is in progress and a read or write cycle is requested, the requesting device receives a "not ready" until the present cycle is completed. After completion of the present refresh cycle a response from the 8203 called System Acknowledge, or $\overline{SACK}$, will notify the requesting device of availability for use. If a read or write request occurs simultaneously with a refresh request, the read or write cycle will be performed first, then the refresh cycle. Read and write cycle requests cannot occur simultaneously during normal operation. If the 8203 is deselected, only an internal or external refresh cycle request will be accepted. Once selected, it will continue with the present memory cycle if one is being performed. (Hence the chip select input is called protected chip select, $\overline{PCS}$, because the current cycle is always completed regardless of any other pending request.)

2.2.3 REFRESH TIMER AND COUNTER

The refresh timer is a counter that increments on each pulse from the clock input until it reaches a preset number causing an internal refresh request to occur. Note that this causes the refresh rate to be 8203 clock cycle dependent. External refresh requests will cause the refresh timer to reset, but will not disable it.

The internal address counter contains the address that will be used during the next refresh cycle. The counter is incremented after each refresh, counting up to 256 before resetting to zero after all RAM addresses have been refreshed. All current generation Intel DRAMs require a 128-cycle refresh, hence, the most significant bit is ignored. However, this extra bit allows use of 256 cycle 4 ms refresh devices without changing the current memory system design.

2.2.4 MULTIPLEXER

The multiplexer is controlled by the timing and control logic. It presents to the address bus one of the following:

1. The contents of the refresh counter when there is a refresh cycle
2. $AL_{0\text{-}6}$ on a $\overline{RAS}$ pulse
3. $AH_{0\text{-}6}$ on a $\overline{CAS}$ pulse

The outputs from the multiplexer are inverted from the address inputs. This is immaterial to the dynamic RAM array and does not require inversion for proper system operation.

2.2.5 TIMING AND CONTROL

The timing and control logic allows either a read, write or refresh cycle to occur. After any read or write cycle request, $\overline{SACK}$ (System ACKnowledge) goes active if the cycle was not requested during a refresh cycle. If it was, $\overline{SACK}$ is delayed until $\overline{XACK}$, thereby requesting WAIT states from the cycle requester.

Figure 14 is a diagram of the 8203 pinout. Table 9 lists the pin numbers, the symbols, and the function of each pin when the 8203 is configured for the 64K option.

The 8203 has two ways of providing dynamic RAM refresh:

1. Internal (failsafe) refresh
2. External refresh

Both types of 8203 refresh cycles activate all of the $\overline{RAS}$ outputs, while $\overline{CAS}$, $\overline{WE}$, $\overline{SACK}$, and $\overline{XACK}$ remain inactive.

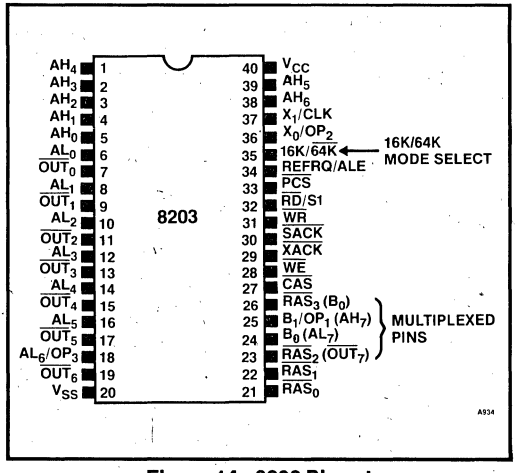


Figure 14. 8203 Pinout

2.2.6 REFRESH CYCLES

Internal refresh is generated by the on-chip refresh timer. The timer uses the 8203 clock to ensure that refresh of all rows of the dynamic RAM occurs every 2 milliseconds. If REFRQ is inactive, the refresh timer will request a refresh cycle every 10-16 microseconds.

External refresh is requested via the REFRQ input (pin 34). External refresh control is not available when the Advanced-Read mode is selected. External refresh requests are latched, then synchronized to the 8203 clock.

Table 9. Pin Description (64K Option)

Symbol	Pin No.	Type	Name and Function
AL_0	6	Input	Address Low: CPU address inputs used to generate memory row address.
AL_1	8	Input	
AL_2	10	Input	
AL_3	12	Input	
AL_4	14	Input	
AL_5	16	Input	
AL_6	18	Input	
AL_7	24	Input	
AH_0	5	Input	Address High: CPU address inputs used to generate memory column address.
AH_1	4	Input	
AH_2	3	Input	
AH_3	2	Input	
AH_4	1	Input	
AH_5	39	Input	
AH_6	38	Input	
AH_7	25	Input	
BO	26	Input	Bank Select Input: Used to gate the appropriate $\overline{RAS}_0$-$\overline{RAS}_1$ output for a memory cycle.
$\overline{PCS}$	33	Input	Protected Chip Select: Used to enable the memory read and write inputs. Once a cycle is started, it will not abort even if $\overline{PCS}$ goes inactive before cycle completion.
$\overline{WR}$	31	Input	Memory Write Request
$\overline{RD}$	32	Input	Memory Read Request
REFRQ	34	Input	External Refresh Request
$\overline{OUT}_0$	7	Output	Output of the Multiplexer: These outputs are designed to drive the addresses of the dynamic RAM array. (Note that the $\overline{OUT}_{0\text{-}7}$ pins do not require inverters or drivers for proper orientation.)
$\overline{OUT}_1$	9	Output	
$\overline{OUT}_2$	11	Output	
$\overline{OUT}_3$	13	Output	
$\overline{OUT}_4$	15	Output	
$\overline{OUT}_5$	17	Output	
$\overline{OUT}_6$	19	Output	
$\overline{OUT}_7$	23	Output	
$\overline{WE}$	28	Output	Write Enable: Drives the write enable inputs of the dynamic RAM array.
$\overline{CAS}$	27	Output	Column Address Strobe: This output is used to latch the column address into the dynamic RAM array.
$\overline{RAS}_0$	21	Output	Row Address Strobe: Used to latch the row address into bank of dynamic RAMs, selected by the 8203 Bank Select Pin (B_0).
$\overline{RAS}_1$	22	Output	
$\overline{XACK}$	29	Output	Transfer Acknowledge: This output is a strobe indicating valid data during a read cycle or data written during a write cycle. $\overline{XACK}$ can be used to latch valid data from the RAM array.
$\overline{SACK}$	30	Output	System Acknowledge: This output indicates the beginning of a memory access cycle. It can be used as an advanced transfer acknowledge to eliminate WAIT states. (Note: if a memory access request is made during a refresh cycle, $\overline{SACK}$ is delayed until $\overline{XACK}$ in the memory access cycle.)
X_0/OP_2	36	Input	Crystal Inputs: These inputs are designed for a quartz crystal to control the frequency of the oscillator. X_1/CLK becomes a TTL input for an external clock if X/OP is tied to V_{CC}.

The arbiter will allow the refresh request to start a refresh cycle only if the 8203 is not in a cycle.

Internally, if a memory request and a refresh request reach the arbiter at the same time, the 8203 will honor the refresh request first. However, the external refresh synchronization takes longer than the memory request synchronization so, relative to the 8203 input signals, a simultaneous memory request and external refresh request will result in the memory request being honored first. This 8203 characteristic can be used to "hide" refresh cycles during system operation. A circuit similiar to Figure 15 can be used to decode the CPU's instruction fetch status to generate an external refresh request. The refresh request is latched while the 8203 performs the instruction fetch: the refresh cycle will start immediately after the memory cycle is completed, even if the $\overline{RD}$ input has not gone inactive. If the CPU's instruction decode time is long enough, the 8203 can complete the refresh cycle before the next memory request is generated.

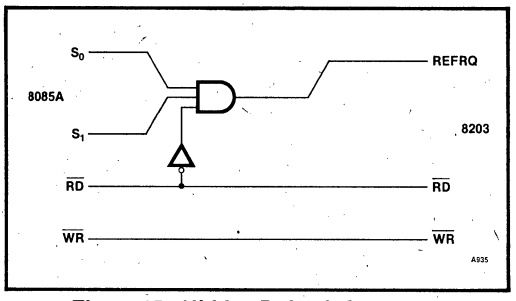


Figure 15. Hidden Refresh Generator

After each refresh cycle, the 8203 increments the refresh counter, reloads the refresh timer, and clears the external refresh latch. If the external refresh request is held active, the latch will be set again, and another refresh cycle will be generated. If, however, a memory request is pending, it will be honored before the second refresh request. This feature prevents refresh from locking out the memory request.

Certain system configurations require complete external refresh control. If external refresh is requested faster than the minimum internal refresh timer (t_{REF}) then, in effect, all refresh cycles will be caused by the external refresh request, and the internal refresh timer will never generate a refresh request.

2.2.7 READ CYCLES

The 8203 can accept two different types of memory Read requests:

1. Normal Read, via the $\overline{RD}$ input
2. Advanced Read, using the S1 and ALE inputs

The user can select the desired Read request configuration via the B1/OP1 hardware strapping option on pin 25.

Normal Reads are requested by activating the $\overline{RD}$ input, and keeping it active until the 8203 responds with an $\overline{XACK}$ pulse. The $\overline{RD}$ input can go inactive as soon as the command hold time (t_{CHS}) is met.

Advanced Read cycles are requested by pulsing ALE while S1 is active; if S1 is inactive (low) ALE is ignored. Advanced Read timing is similiar to Normal Read timing, except the falling edge of ALE is used as the cycle start reference.

If a read cycle is requested while a refresh cycle is in progress, then the 8203 will set the internal delayed-$\overline{SACK}$ latch. When the Read cycle is eventually started, the 8203 will delay the active $\overline{SACK}$ transition until $\overline{XACK}$ goes active. This delay was designed to compensate for the CPU's READY setup and hold times. The delayed-$\overline{SACK}$ latch is cleared after every READ cycle.

Based on system requirements, either $\overline{SACK}$ or $\overline{XACK}$ can be used to generate the CPU READY signal. $\overline{XACK}$ will normally be used; if the CPU can tolerate an advanced READY, then $\overline{SACK}$ can be used. If $\overline{SACK}$ arrives too early to provide the appropriate number of WAIT states, then either $\overline{XACK}$ or a delayed form of $\overline{SACK}$ should be used.

2.2.8 WRITE CYCLES

Write cycles are similiar to Normal Read cycles, except for the $\overline{WE}$ output. $\overline{WE}$ is held inactive for Read cycles, but goes active for Write cycles. All 8203 Write cycles are "early write" cycles; $\overline{WE}$ goes active before $\overline{CAS}$ goes active by an amount of time sufficient to keep the dynamic RAM output buffers turned off.

For a more detailed analysis of the 8203, refer to Application Note AP-97A, entitled "Interfacing Dynamic RAMs to iAPX 86/88 Systems Using the Intel 8202A and 8203."

3 SIMPLE SOLUTION

An example of the ease of interfacing DRAMs to microprocessors with the 8203 is shown in Figure 16. This is an example of the 8203 and 2118's or 2164A's configured as local memory to a min mode iAPX 88 System. The CPU's local bus is demultiplexed by an 8283 which latches the addresses and presents them to the 8203. Notice the lack of TTL support circuitry. The only additional components are a latch for the dynamic RAM output data and a OR gate to steer the $\overline{WE}$ signal on byte writes. The 8203 handles all the interface requirements of the

DRAM array, rendering a very simple solution to a dynamic memory design.

Figure 17 is an 8203/2164A memory system configured as a global resource to a max-mode iAPX 86 microprocessor system. Although there are several more TTL components involved, the buffers and transceivers are a requirement for proper system bus interface design. In terms of controlling the memory, the 8203 and 2164A interface is as simple as in the previous example. The ability of the 16 bit 8086 to perform byte operations requires two gates (shown on the diagram of Figure 17 between the 8203 and the 2164A array) to steer the write pulse output from the 8203 to either the high or low byte or both bytes as directed by A0 and $\overline{BHE}$ (Byte High Enable).

These examples balance ease of use and design throughput time with performance. The designs shown typically require one to two WAIT states. With one WAIT state,

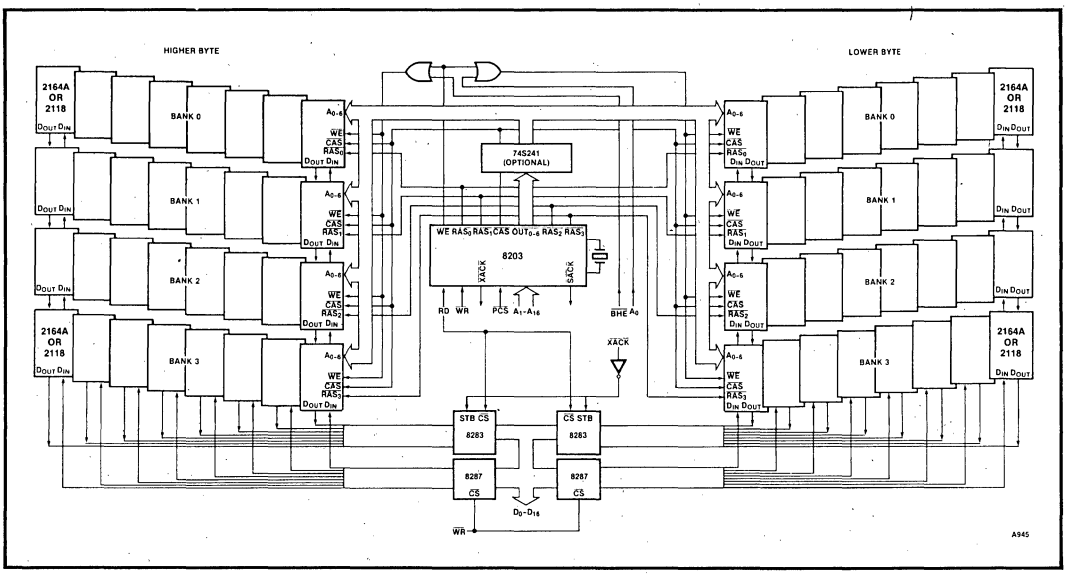


Figure 16. 8203/2118 Local Memory System

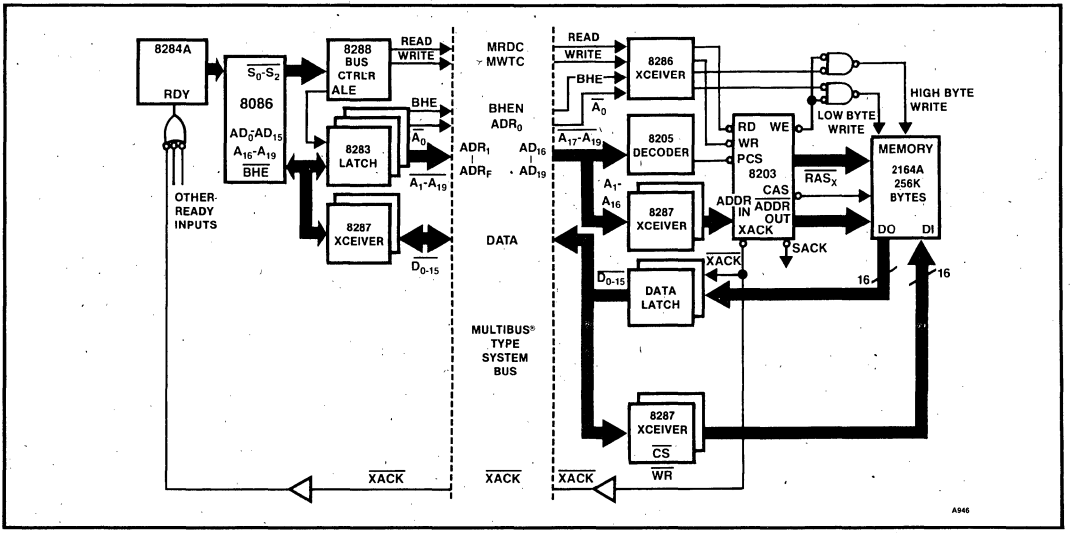


Figure 17. 8203/2164A Global Memory System

processor performance is reduced to 91.7%, and with two WAIT states it drops to 83.7%. This may be acceptable in many applications, but where it is not, a modest additional design effort can yield zero WAIT states.

4 5 MHz NO-WAIT STATE SYSTEM1

4.1 Circuit Description

The DRAM/8203 microprocessor memory system discussed up to this point met all of our design criteria except one — optimum performance. In minimum mode operation, inherent delays in the system $\overline{RD}$ and $\overline{WR}$ commands resulted in a READY signal that was too late to avoid processor WAIT states. Attaining zero WAIT states requires minimizing these delays by transmitting advanced read ($\overline{RD}$) or write ($\overline{WR}$) commands. This is not a simple task in minimum mode operation because the iAPX 86 processor produces the $\overline{RD}$ and $\overline{WR}$ signals in a fixed relationship after ALE occurs. However, operating in a max-mode, the iAPX 86 outputs three status bits ($\overline{S0}$, $\overline{S1}$, $\overline{S2}$) which occur ahead of the ALE signal. (Refer to the timing diagram shown in Figure 18.) With proper logic circuitry, these status bits can be used to initiate the advanced signals required.

The following discussion describes a 5 MHz no-WAIT state microprocessor memory system designed for optimum performance. Figure 19 shows an iAPX 86 maximum mode system modified for zero WAIT states. The circuitry added to the system previously described is enclosed in the dashed lines. The 8205 decodes the three status bits ($\overline{S0}$, $\overline{S1}$, $\overline{S2}$) and outputs an advanced read or write signal at pin 13 or 14, respectively. These signals flow through the corresponding 74S158 (a 2:1 mux con-

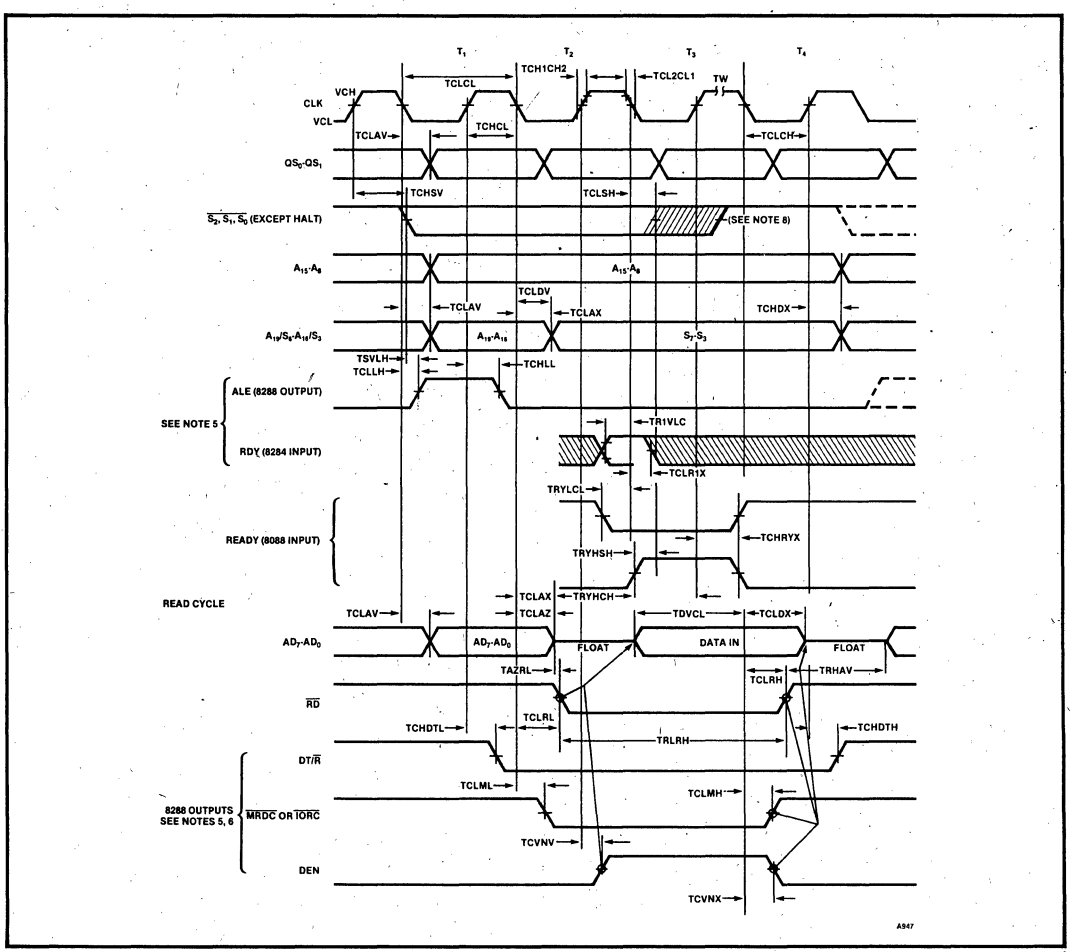


Figure 18. 8086 Bus Timing — Maximum Mode System (using 8288)

figured as a high speed flow through latch) and are latched on the falling edge of ALE from the 8288. Latch outputs (ADV $\overline{WRC}$ and ADV $\overline{RDC}$) are connected to the 8203 $\overline{WR}$ and $\overline{RD}$ inputs. The two latches are cleared by clocking the trailing edge of either the memory read command ($\overline{MRDC}$) or memory write command ($\overline{MWTC}$) through a 74S74 flip-flop. System acknowledge ($\overline{SACK}$ — used in place of $\overline{XACK}$ because it occurs sooner) is returned to the 8284A which provides a synchronous ready signal to the iAPX 86. The advanced memory write command, $\overline{AMWC}$, clocked to provide appropriate timing with $\overline{CAS}$, is ORed with $\overline{WE}$ to obtain the $\overline{WR}$ for the 2118's. The $\overline{S2}$ status bit is latched by the 74S158 on the trailing edge of ALE.

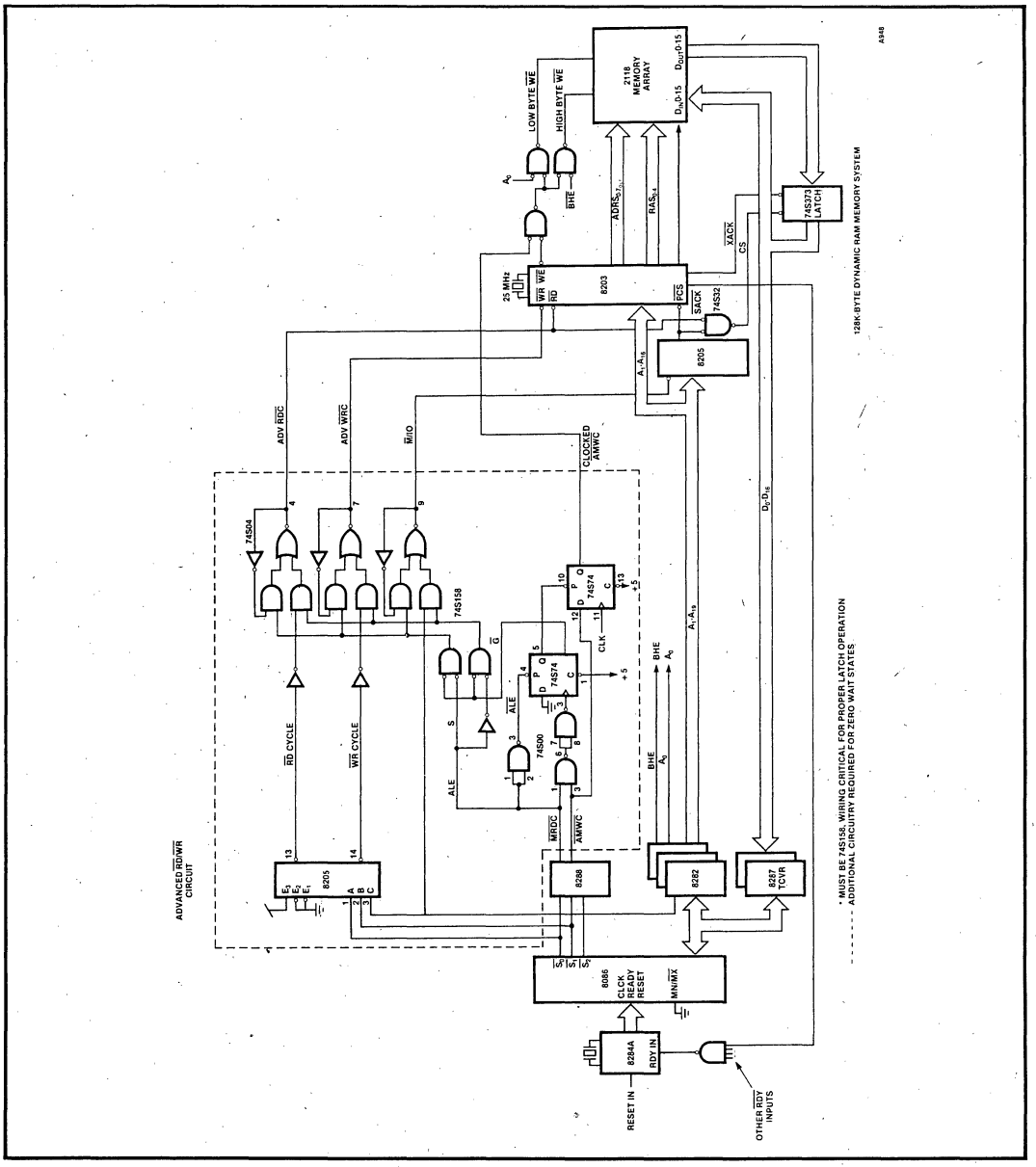


Figure 19. 5 MHz No-WAIT State Microprocessor Memory System

4.2 Analysis and Description of System Timing

Read cycle worst case analysis is shown in Figure 20 which only considers the maximum time delays. The four processor t states are indicated by t_1 through t_4. To accomplish zero WAIT states, valid data must reach the iAPX 86 by the end of t_3 minus 30 ns. The latest read data arrives at the iAPX 86 (next to the last waveform) within this time frame. Timing relationships are as follows:

The ADV $\overline{RDC}$ flows through the 74S158 latch and reaches the 8203 within 6 ns after the rise of ALE. The latest $\overline{PCS}$ is generated by decoding CPU addresses and arrives within 133 ns. The $\overline{SACK}$ signal is then returned within 127 ns from $\overline{PCS}$. The buffered $\overline{SACK}$ is used as the READY signal to the iAPX 86, resulting in zero WAIT states (except when the 8203 is performing a refresh cycle). The maximum $\overline{PCS}$ to $\overline{CAS}$ delay is shown to be 245 ns. Also accounted for is the maximum access time from $\overline{CAS}$ to data valid of 80 ns and a propagation delay of 45 ns for valid data to reach the processor.

In the write cycle, the relationship between data and $\overline{WE}$ at the memory and the relationship between the leading edge of $\overline{WE}$ and the trailing edge of $\overline{CAS}$ (t_{CWL}) must be

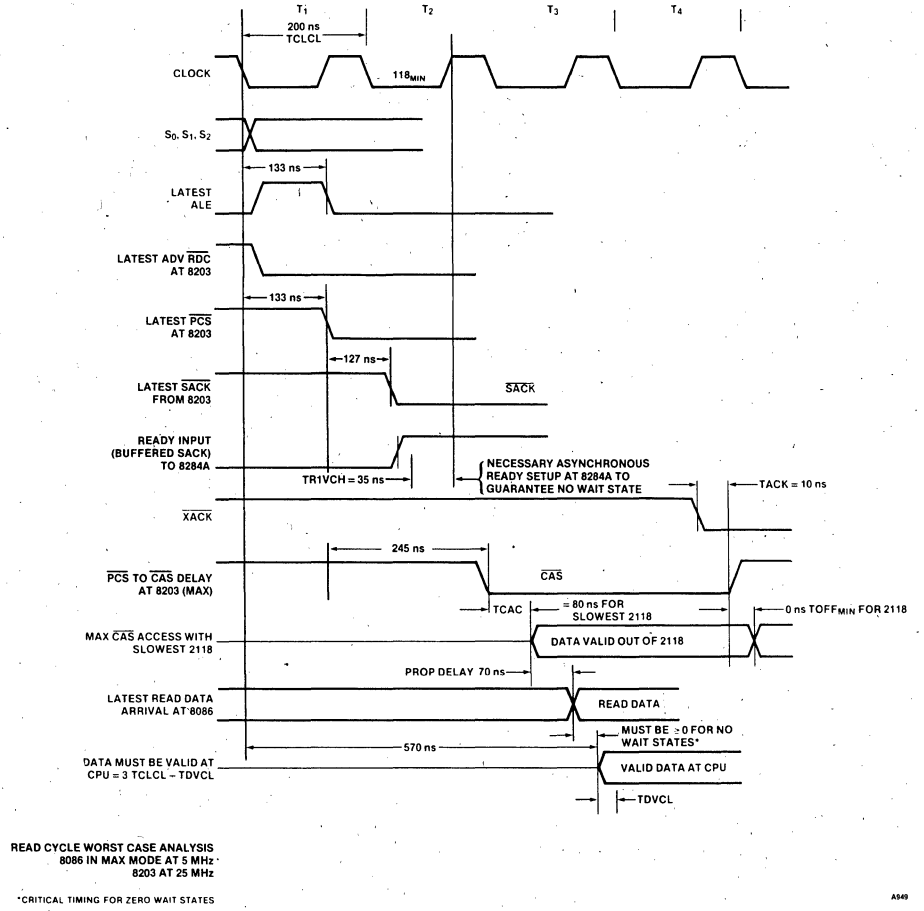


Figure 20. Read Cycle Timing Analysis (5 MHz)

preserved. Since DRAMs write data on the leading edge of the write pulse, data must be valid before the fall of $\overline{WE}$. Timing analysis of the skew of the normal memory write command ($\overline{MWTC}$) to valid data shows that worse case, it is possible to have data arrive after the falling edge of $\overline{WE}$ (case 1 of Figure 21). Using the other write pulse available from 8288 bus controller, the advanced memory write command ($\overline{AMWC}$), led to the problem depicted in Figure 21, case 2, violation of the DRAM specification t_{CWL}. From these observations, the need for the clocked $\overline{AMWC}$ pulse becomes apparent. By delaying the $\overline{AMWC}$ pulse until the next rising edge of the system clock and then gating this signal with the $\overline{WR}$ output from the 8203, a "best-fit" write pulse is created that meets all timing requirements.

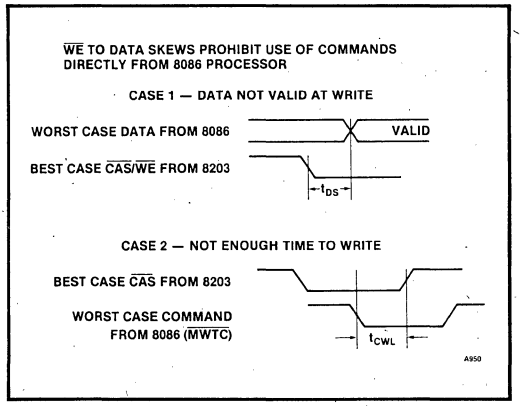


Figure 21. Write Cycle Problems

Figure 22 depicts the worst case analysis of the write cycle. The timing relationships are similar to those for the read cycle with a few exceptions. The advanced write command, ADV $\overline{WRC}$, flows through 74S158 and is latched by the fall of ALE. The earliest $\overline{CAS}$ occurs 145 ns after the $\overline{PCS}$. Valid data is output from the CPU within 210 ns and reaches the memory 35 ns later. The advanced memory write command, $\overline{AMWC}$, and associated progation delays must satisfy the t_{CWL} requirement of the 2118's which starts at the beginning of the $\overline{AMWC}$ pulse and terminates with the end of $\overline{CAS}$. The write enable, $\overline{WE}$, from the 8203, is ANDed with $\overline{AMWC}$ to obtain the $\overline{WR}$ for memory.

4.3 Compatibility of the 2118 and 2164A

The 5 MHz no-WAIT state system was designed with the 2118-15 DRAM. By following the guide lines in section 1.3 and examining tight timing areas specific to this application, it can be shown that the system is expandable and works equally well by using two rows of 2164A-15 parts in place of four rows of 2118-15 parts. The 8203, when configured in the 64K mode, guarantees proper generation and arrival of timing signals to the memory. Since the controller is $\overline{CAS}$ access (t_{CAC}) limited, the t_{CAC} spec of the 2118 and the 2164A must be compared for the read cycle. t_{CAC} on the 2164A-15 is 85 ns, 5 ns greater than the 2118. This means that valid data will arrive at the 8086 processor 5 ns later, for the worse case, using the 64K device. The read cycle timing analysis shows this is still well within the 570 ns requirement of the 8086. During the write cycle, two parameters were of concern in the 5 MHz system:

t_{DS} (data set-up before $\overline{CAS}$)

t_{CWL} (leading edge of write to trailing edge of $\overline{CAS}$)

Since the t_{DS} spec is the same for both devices (0 ns), the original timing analysis for this parameter is still valid and the 2164A fits. The t_{CWL} spec for the 2164A-15 is 40 ns. This is 60 ns less than the 2118-15, so that substituting the 2164A actually relieves a tight timing spot in this design. The additional delay added to control line paths due to larger input capacitances of the 2164A is accounted for in the 8203 specification (the 8203 is specified to directly drive four rows of 2118's, only two rows of 2164A's for this reason). After adding decoupling to meet the 2164A-15 requirements, the 2164A memory system is up and running, doubling memory size and reducing device count by one-half.

4.4 System Reliability

The majority of microcomputer systems are designed into applications where system failure ranges from irritating (such as a vending machine failure) to a financial loss (such as a double debit from an electronic teller machine). While these are not life threatening failures, reliability is important enough to be designed into the system.

A memory system is one of the system components for which reliability is important. Also it is one of the few system elements which can be easily altered to enhance its reliability. The inclusion of some additional hardware allows the CPU to keep check on the integrity of the data in memory. Figure 23 represents a five TTL chip solution that, when added to the 5 MHz design example, allows error detection in the memory.

Because the 16-bit 8086 has the ability to do selective high or low byte writes in addition to full word operations, parity needs to be generated and checked on the byte level. This requires two extra memory devices per row to store the parity bits of the high and low bytes.

Parity is generated by exclusive ORing all the data bits in each byte (accomplished by the 74S280) which results in a parity bit. This parity bit is the encoding bit of each byte. Because there are eight data bits, the parity bit C is: $C = b_1 \oplus b_2 \ldots . b_7 \oplus b_8$ where b = value in the bit positions.

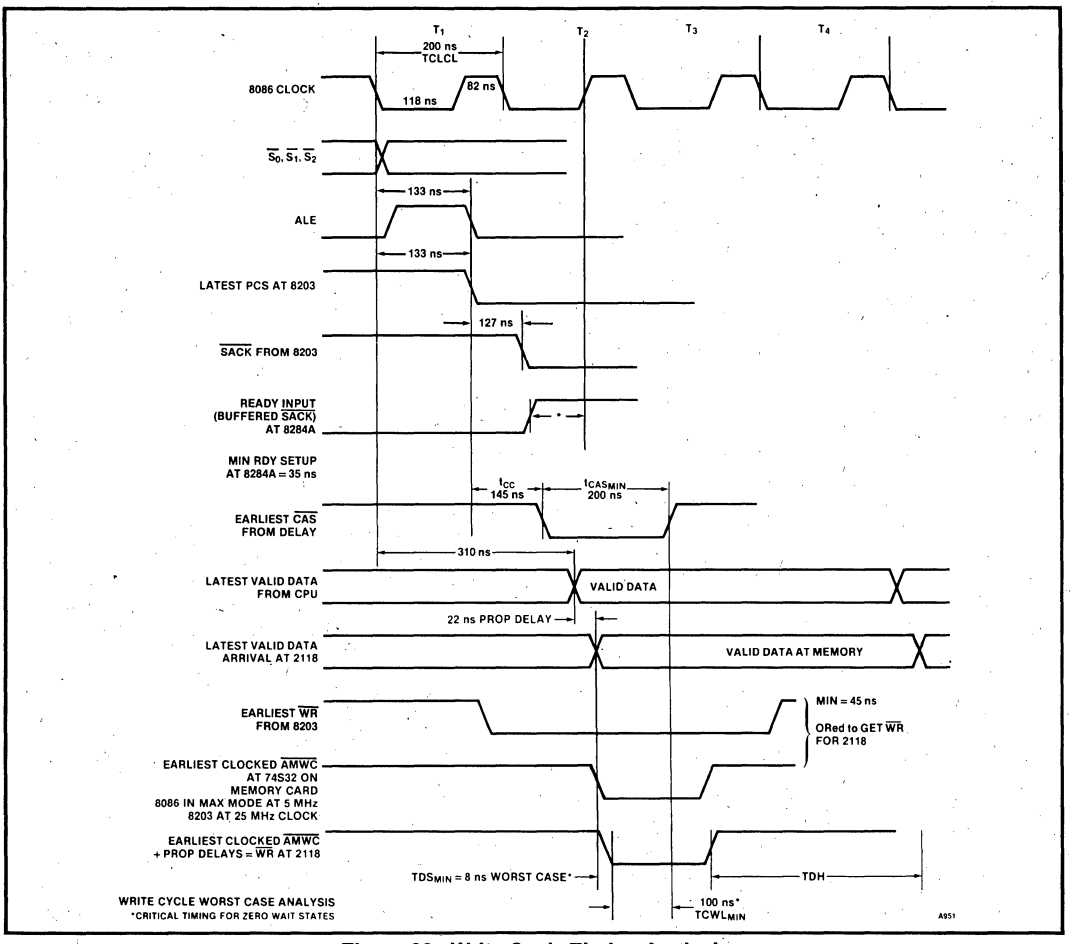


Figure 22. Write Cycle Timing Analysis

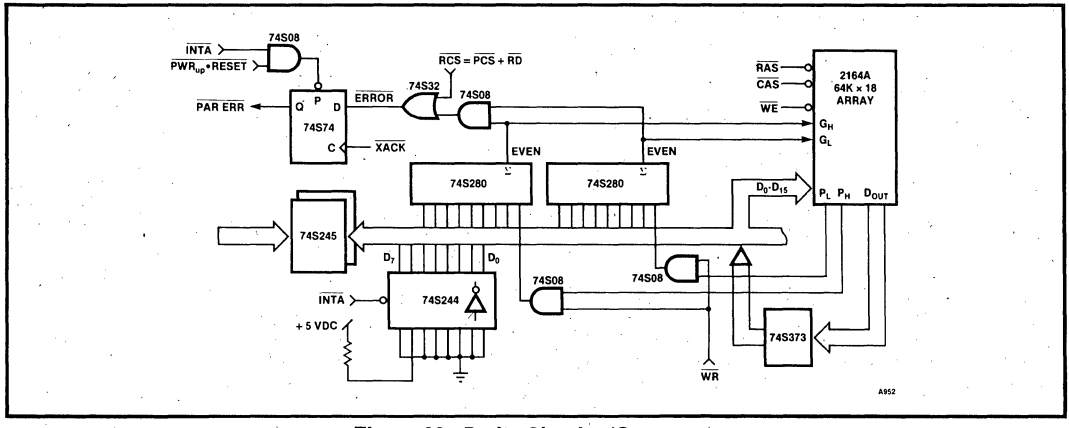


Figure 23. Parity Checker/Generator

The parity bit combines with the bits from the original data byte to form the encoded half-word (9-bit byte). Encoded words always have either "odd" parity, which is an odd number of 1s (an odd weight) or "even" parity which is an even number of 1s (an even weight). Odd and even parity are never intermixed, so that the encoded words have either odd or even parity — never both.

When the encoded word is fetched, the parity bits are removed from the word and saved. Two new parity bits are generated from each byte. Comparing these new parity bits with the stored parity bit determines if a single bit error has occurred in either byte.

Consider the two bit data word whose value is "01". Exclusive-NORing the two data bits generates a parity bit which causes the encoded word to have odd parity:

$C = \overline{0 \oplus 1}$

$C = 0$

The encoded word becomes:

Data	Generated Parity Bit
01	0

Assume that an error occurs and the value of the word becomes "110." Stripping off the parity bit and generating a new parity bit:

transmitted parity = 0

transmitted word = 11

New parity of transmitted word = $\overline{1 \oplus 1}$ = 1; generated parity ≠ transmitted parity.

Note that the error could have occurred in the parity bit and the final result would have been the same. An error in the encoding bit as well as in the data bits can be detected.

Although parity detects the error, no correction is possible. This is because each valid word can generate the same error state. Illustration of this is shown in Table 10.

Table 10. Possible Errors

Possible Correct Word with Parity	Single Bit Error
0 0 1	0 1 1
1 1 1	0 1 1
0 1 0	0 1 1

Each of the errors is identical to the others and reconstruction of the original word is impossible.

Parity fails to detect an even number of errors occurring in the word. If a double bit error occurs, no error is detected because two bits have changed state, causing the weight of the word to remain the same.

Using the encoded word "010" one possible double bit error (DBE) is:

1 1 1
└— Parity

Checking parity:

$C = \overline{1 \oplus 1} = 1$

The transmitted parity and the regenerated parity agree. Therefore the technique of parity can detect only an odd number of errors.

In the circuit of Figure 23, parity is generated and checked in the same devices — the 74S280 pair. Should a parity error occur in either the high or low byte (or both) the error flip-flop is set, causing an interrupt to the 8086 to occur. When the 8086 responds with INTA (interrupt acknowledge) the flip-flop is reset. INTA also enables the 74S244 which gates the interrupt number onto the data bus. The interrupt request signal to the CPU indicates a memory error has occurred. The nature of the interrupt procedure is heavily dependent on the user application, but typically ranges from retry or recovery routines to simply turning on the parity error light and proceeding.

One other software consideration for this circuit is the requirement to initialize all the memory to a known state. This initialization is needed to properly encode all the memory to even parity. This is typically done upon power-up by writing zeros into all memory locations prior to program storage.

In summary, single bit parity will detect the majority of errors, but cannot be used to correct errors. Using parity introduces a measure of confidence in the system. Should a single bit error occur, it will be detected.

For a detailed treatment of error detection and also techniques for error correcting, refer to Intel application notes AP-46, "Error Detecting and Correcting Codes Part #1," and Application Note AP-73, "ECC #2 Memory System Reliability with Error Correction."

4.5 Alternatives to 8203 Refresh Control Designs

There are essentially four choices available when selecting a technique for refresh control circuitry. These are:

Separate controller

CPU Hardware Control

CPU Software Control

Circuitry Internal to the RAM

Figure 24 is an implementation of a separate controller design. This is a typical non-LSI version that requires 11 TTL packages, an 8282A octal latch, a 3242 address multiplexer/refresh counter, two bidirectional bus drivers, an 8212 octal latch and two active delay lines.

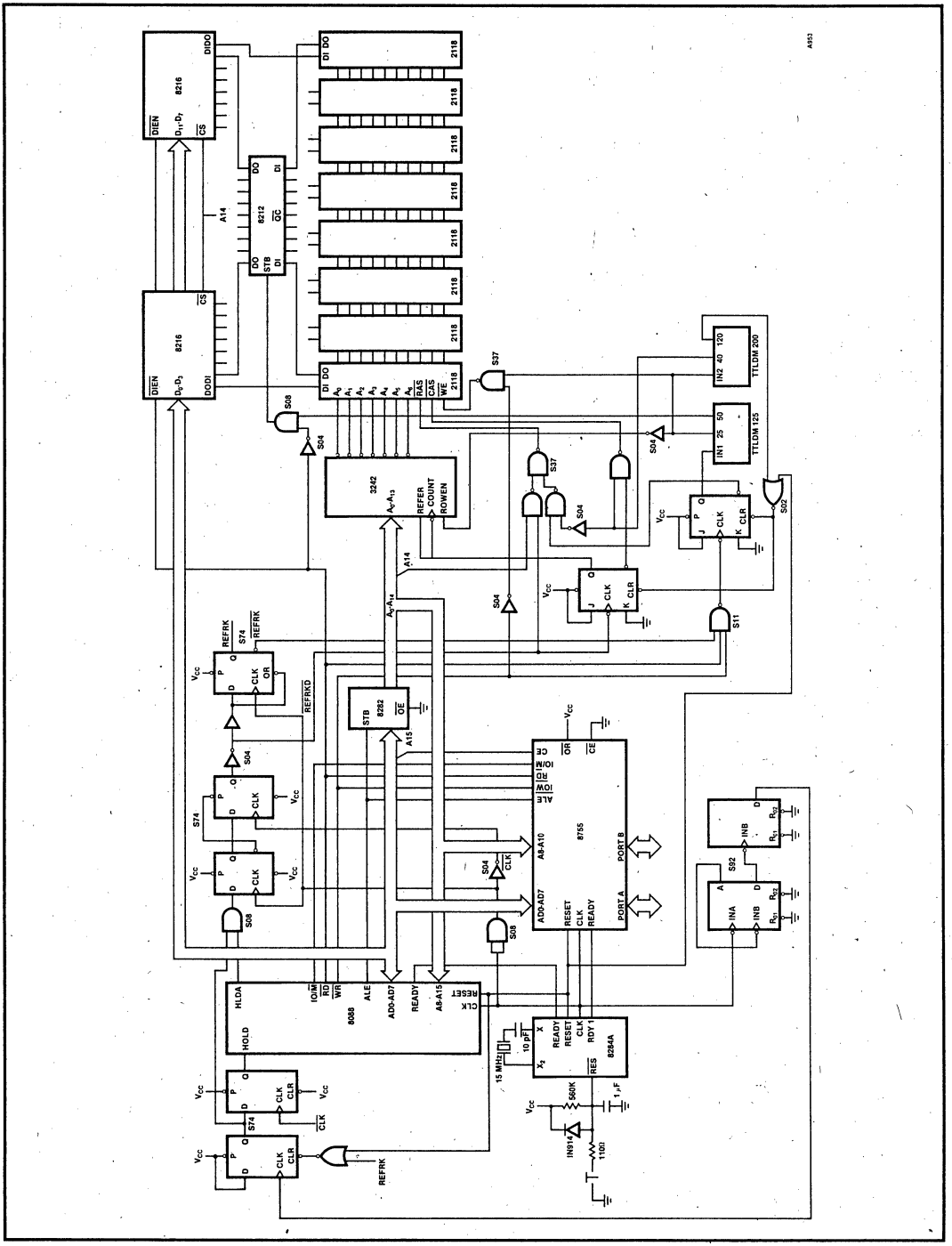


Figure 24. Discrete DRAM Controller

Nothing is gained by using discrete packages where a LSI device can be designed in. The plethora of TTL does require a larger engineering effort exemplified by the circuit complexity and timing analysis for this circuit (Figure 25). In terms of performance, the extra engineering effort can be fruitless — the CPU in this example is forced into the HOLD condition every time a refresh cycle occurs, even if the memory is not being accessed. This waiting period lasts 1.23 microseconds for every refresh cycle performed. Contrast this with the 8203 circuit which runs without WAIT states (unless a refresh cycle is in progress when the CPU requests a memory access, in which case one WAIT state is inserted). The advantages of using the 8203 should be obvious by now.

Additional hardware closely coupled to the CPU timing refresh for the microprocessor operation is one alternative to 8203 design. Some implementations include the extra hardware within the microprocessor; rendering a low cost, simple design. Wide restrictions govern the usage of such a system however, precluding this type of design in many applications.

To cite a few disadvantages:

- CPU must run continuously — no single step, HOLD, or extended WAIT states
- Multiprocessor operation is difficult
- CPU must always participate in memory operations

CPU software control of refresh is another alternative. This approach increases software development and maintenance costs and may not be offset by the very low or no hardware overhead for refresh. One method requires real-time analysis of all modules and possible directions of the program, with branch-to-refresh instructions included in all paths so that a refresh procedure is executed at least every 2 ms. An option on this technique requires a single interrupt time, which, when it times out, interrupts the CPU, causing it to revert to the burst refresh software routine.

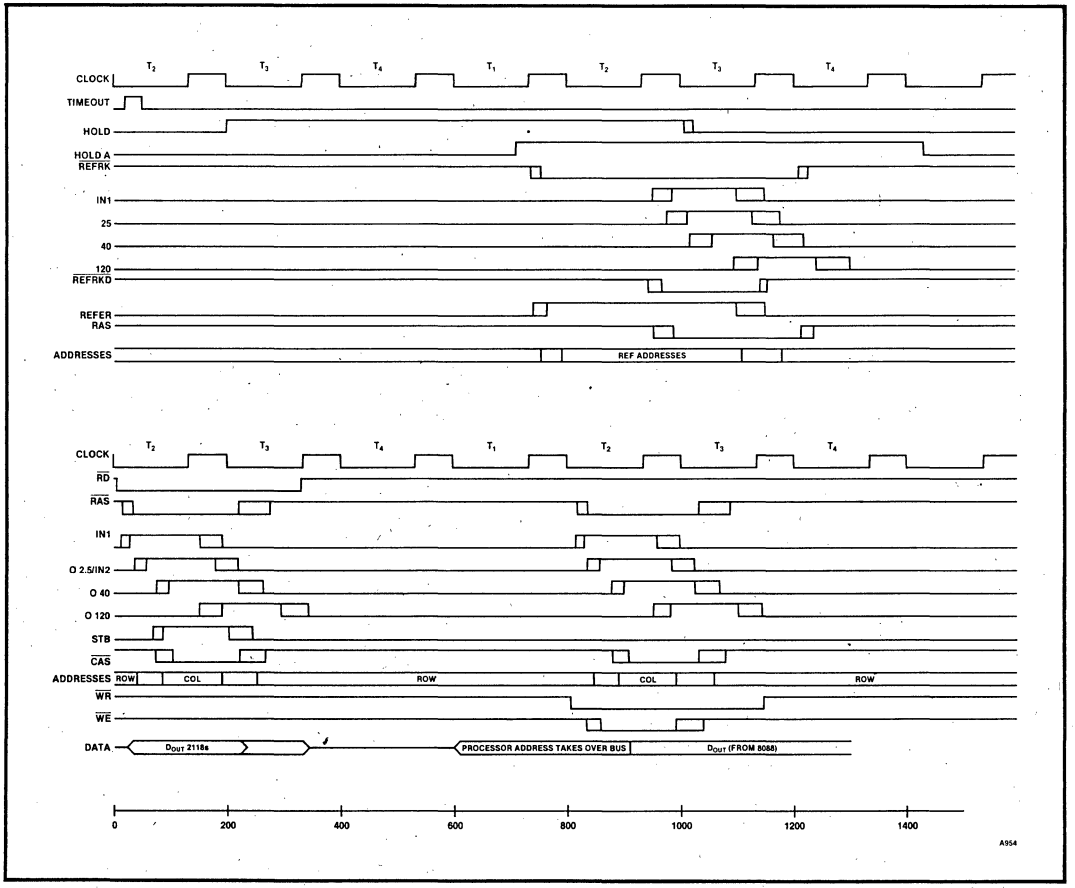


Figure 25. Timing Analysis Discrete Controller

Figure 26 shows an ASM-86 implementation of a burst refresh procedure. Accomplishing refresh in software is simple: save all registers used, perform a read at each of the 128 row addresses, then restore all registers and return.

The pure software approach makes it very difficult to make program changes and is limited to special applications. Also, since the refresh cycles are actually read cycles, the memory consumes more power for refresh than in standard refresh cycles. Both software refresh methods require that the CPU is always running, and hence shares many of the disadvantages of a CPU hardware refresh design.

One approach to memory system refresh control is to forge the entire system in silicon, incorporating the dynamic RAM array and all of the refresh control circuitry into one device. This, however, represents a departure from classical, dynamic RAM system design methodologies and as such, are outside the scope of this application note.

```
;*************************************************************
;
;            BURST REFRESH ROUTINE IN ASM86
;                  VERSION 1.0
;            MC APPLICATIONS LAB JAN 82
;
;*************************************************************

CSEG SEGMENT
ASSUME CS:CSEG,DS:CSEG

;************** BURST REFRESH INTERRUPT ROUTINE ***************

; This procedure does software refresh from an interrupt by
; performing dummy reads on the first 128 device (row)
; addresses.

; HARDWARE ASSUMPTIONS: RAS is common throughout the array
; with CAS decoded for a row select. An external timer
; generates the refresh interrupt every 2 milliseconds.

;*************************************************************

BURST:                          ;SAVE REGISTER CONTENTS
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    SI
        MOV     BX,BASEADRS     ;PLACE SEG PNTR OF TARGET BOARD ROW IN BX
BURS1:  MOV     DS,BX           ;INIT DATA SEG TO START OF A BOARD ROW
        MOV     CX,REFCOUNT     ;SET LOOP COUNTER TO NUMBER OF DEVICE ROWS
        MOV     SI,ADRCOUNT     ;INIT MEM INDEX PNTR
REF:    MOV     AX,DS:[SI]      ;READ 16 BIT WORD (DUMMY READ IS A REFRESH)
        DEC     SI
        DEC     SI              ;DECREMENT REFRESH ADDRESS PNTR TO NEXT WORD
        LOOP    REF             ;LOOP ONCE FOR EACH DEVICE ROW

; 128 ROWS HAVE BEEN READ, (REFRESHED) SO EXIT

EXIT:   POP     SI              ;RESTORE REGISTERS
        POP     CX
        POP     BX
        POP     AX
        IRET                    ;RETURN FROM INTERRUPT

BASEADRS EQU 0000   ;SET TO SEGMENT ADDRESS OF MEMORY
REFCOUNT EQU 128    ;SET TO NUMBER OF DEVICE ROWS (128 FOR 2118)
ADRCOUNT EQU 256    ;SET TO TWICE NUMBER OF DEVICE ROWS

CSEG ENDS

END
```

Figure 26. PLM-86/ASM-86 Burst Refresh, Sheet 1 of 2

```
REFRSH:
 DO;
  BURSTREF: PROCEDURE;     /* PROCEDURE PROVIDES A BURST REFRESH BY READING
                              ALL 128 DEVICE ROWS ON ALL BOARD LEVEL ROWS*/


  INCADR: PROCEDURE(PTR) POINTER;  /* INCREMENTS REFRESH ADDRESS POINTER */

   DECLARE  PTR POINTER,
            ADR BASED PTR (2) WORD;

   ADR(1)=ADR(1)+2;                /*INC WORD ADDRESS*/
   RETURN PTR;
  END INCADR;


  INCBD: PROCEDURE (PTR)POINTER;  /* INCREMENTS BOARD LEVEL ADDRESS POINTER */

   DECLARE PTR POINTER,
         ADDR BASED PTR (2) WORD;

   ADDR(0)=ADDR(0)+03FFFH;
   IF ADDR(1)=0 THEN ADDR(0)=ADDR(0)+1;
   RETURN PTR;
  END INCBD;


  DECLARE (BDROW$PTR,REF$PTR,START$PTR,LAST$PTR )POINTER;
  DECLARE (REF BASED REF$PTR,RDDATA ) WORD;
  DECLARE (DEVROWS) BYTE;


      /* READ 128 ADDRESSES ON ALL BOARD ROWS */

  DO;
   START$PTR=20000H;
   LAST$PTR=3FFF0H;
   BDROW$PTR=START$PTR;
   REF$PTR=START$PTR;
    DO WHILE BDROW$PTR<=LAST$PTR;
     DEVROWS=128;
      DO WHILE DEVROWS>=0;
       RDDATA=REF;
       REF$PTR=INCADR(REF$PTR);
       DEVROWS=DEVROWS-1;
      END;
     BDROW$PTR =INCBD(BDROW$PTR);
     REF$PTR=BDROW$PTR;
    END;
  END;

 END BURSTREF;

          /* MAIN */

 DO;
  CALL BURSTREF;
 END;

END REFRSH;
```

Figure 26. PLM-86/ASM-86 Burst Refresh, Sheet 2 of 2

One last technique for refresh control exists that doesn't fit into any of the above catagories and is worth bringing to light. Its use is heavily application dependent, hence has the most severe limitations, but if it meets the design requirements, its the most cost effective of all. The memory must be configured so that all row addresses will be strobed within 2 ms. Figure 27 is a block diagram of an application where this is possible since successive memory access addresses are predictable and defined. The circuit depicts a simpified graphics terminal display memory interface. Assuming a requirement of a 512×512 display resolution, the memory array is arranged as two rows of eight 2118 devices. During each read cycle, one byte is loaded from the memory into the shift register and is serially clocked out as video. A single $\overline{RAS}$ is common to the array and $\overline{CAS}$ is decoded to each row. This configuration simultaneously refreshes one row while reading data from the other row. A disadvantage of this arrangement is additional power supply and decoupling requirements, since one row is always making a transition to active current (ΔI_A) while the other draws refresh cycle current (ΔI_R). Refer to Section 6.3.4 on decoupling for calculations. The following is determined:

Pixel Clock (Hz) = (N + R) * L * F = 21.450 MHz

where N = Number of displayed dots per line = 512

L = Number of horizontal lines per frame = 532 (512 visible lines + 20 line times allowed for vertical retrace)

F = Frame rate of 60 Hz

R = Number of pixel clock times allowed for horizontal retrace time = 160 (Usually empirically determined. This number establishes the width of the margins on the left and right sides of the CRT display.)

Memory Cycle Rate = Byte read rate of the memory = 2.68 MHz

$$M_{cyc}\ (Hz) = \frac{21.450\ MHz}{8}$$

$$\frac{\text{pixel rate}}{\text{pixels/byte}} = 2.68\ MHz$$

$$T_{cyc} = \frac{1}{2.68\ MHz} = 373\ ns/cycle$$

The 2118-15 meets this T_{cyc} cycle time requirement.

Since the memory array is sequentially addressed, the memory is automatically refreshed every 128 consecutive cycles.

Checking refresh timings: 128 cycles × 373 ns/cycle = 47.74 microseconds between total refresh for each device, easily within the 2 ms specification.

The worst case refresh occurs during vertical retrace time when:

retrace time = 31.3 microseconds/line × 20 lines = 627 microseconds

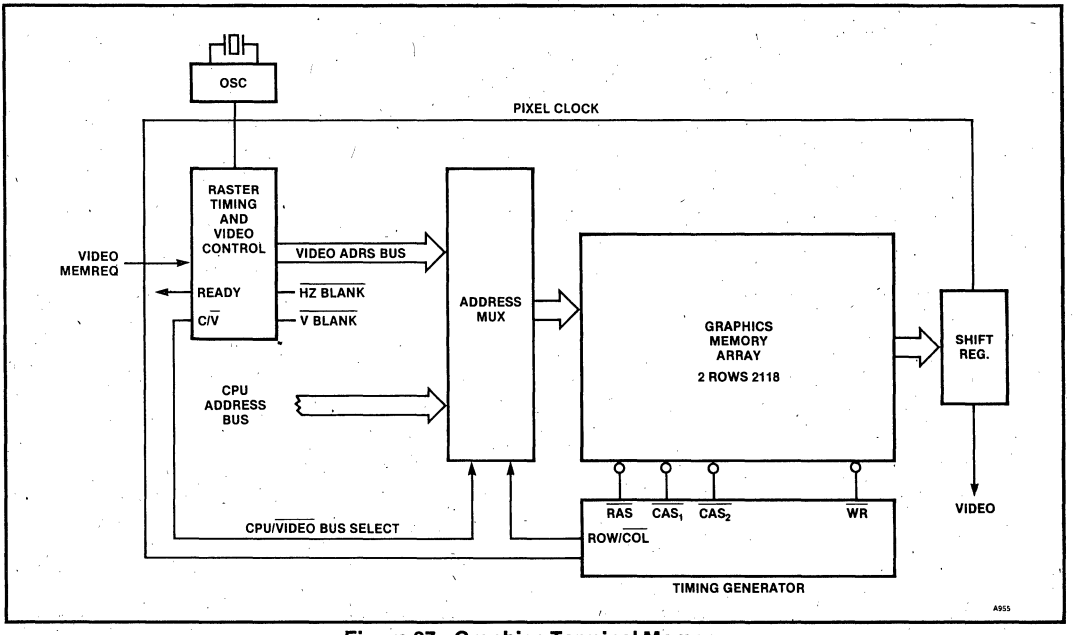


Figure 27. Graphics Terminal Memory

worst case refresh rate = 627 microseconds + 47.7 microseconds = 674.7 microseconds, still well within the 2 ms specification.

Writing is performed during horizontal or vertical retrace. More efficient designs would interleave memory, eliminating the processor being in WAIT mode until the memory is open. Here, and in some other limited applications, refresh can occur automatically by design, and with no software or hardware overhead.

5 10 MHz NO-WAIT STATE SYSTEM

For fast high performance microprocessors such as the 10 MHz 8086, an LSI controller for dynamic RAM interfacing is unacceptable, due to the requirement for WAIT states and resultant impact on performance. Until faster LSI controllers appear, discrete controller designs are required. In the example that follows, high performance design techniques are coupled with Intel high performance RAMs to yield a 10 MHz no-WAIT state 8086/2164A system.

The key requirements are:

ALE to data in:	219 ns
READY response:	89 ns
2164A t_{RAC}	150 ns

The solution and implementation that follows, configures the 8086-1 in max-mode, incorporates a synchronous arbiter while providing a quasi-synchronous refresh (refresh that is synchronous to the system clock, but not to the microprocessor).

5.1 System Refresh

Rather than being constrained to the design configurations of purely synchronous or asynchronous refresh arbitration, a quasi-synchronous scheme was chosen — taking advantage of the benefits of both, and avoiding some of the drawbacks of implementing either one exclusively. Synchronizing the refresh arbitration to the system clock ensures that its operations are inherently and closely coupled to CPU operation and allowing critical timing edges to always be predicted through worst case analysis. However, unlike totally synchronous systems, if the CPU in this example were to enter a HOLD, HALT, or otherwise stopped state, refresh cycles would continue to keep valid data in the memory, independent of the CPU operation. Also, synchronization of refresh requests to the system clock make the task of the arbiter very easy. Memory cycle requests and refresh cycle requests never occur at the same time (Figures 28 and 29, timing analysis). As a result, there is no chance that a random cycle request can arrive in a narrow time window that would violate data setup and data hold time of a flip-flop arbiter. This is a major problem in purely asynchronous designs.

5.2 System Block Diagram

Figure 30 is a block diagram of the basic functions required for this system; refresh interval timer, refresh address counter, arbiter synchronization, address multiplexing and timing generation. Included also in the diagram are the memory and CPU status decoders, data latches and transceivers, bus control and clock generation.

The function of the refresh interval timer is to place requests for refresh cycles, distributed in approximately 15 microsecond intervals, so that each row of the memory devices receives a refresh within 2 milliseconds. This timer is comprised of two four-bit synchronous binary counters and two flip-flops. The timer circuits divide the 10 MHz system clock by 150, then latches the count carry bit to hold until recognized, through the arbiter, by the refresh latch.

The refresh address counter generates the refresh addresses that are submitted to the address multiplexer during a refresh cycle. The counter is incremented once at the end of each refresh cycle to update the refresh address. The outputs are wire-ORed to the microprocessor address bus and are active only during a refresh cycle, at which time the current count is presented to the address multiplexer as the refresh address.

Timing generation for the memory array produces the control signals for the address multiplexer and the gating signals that provide for the properly timed arrival to the memory of $\overline{RAS}$, $\overline{CAS}$, and addresses. In this design example, it is essentially a delay circuit with variable taps to permit fine tuning of the memory inputs so as to allow no-WAIT states by the microprocessor for a memory cycle. The strobe used to latch valid data from the memory is also provided by the timing generator.

The 2164A dynamic RAM requirement of multiplexed row and column addresses is met by the address multiplexer. Here, the proper selection and transmission of row/refresh or column addresses is accomplished by control of the select line timing generation circuit.

In this design (Figure 31), arbitration is easily performed, i.e., once a cycle type is latched into its respective flip-flop (refresh latch or memory access latch) its request is presented to the input of an AND gate that will allow the request to pass through if a request of the other type is not currently in execution. Once the request passes the AND gate, the hardware is committed to a cycle of the requesting type and blocks any subsequent request until the current cycle is complete.

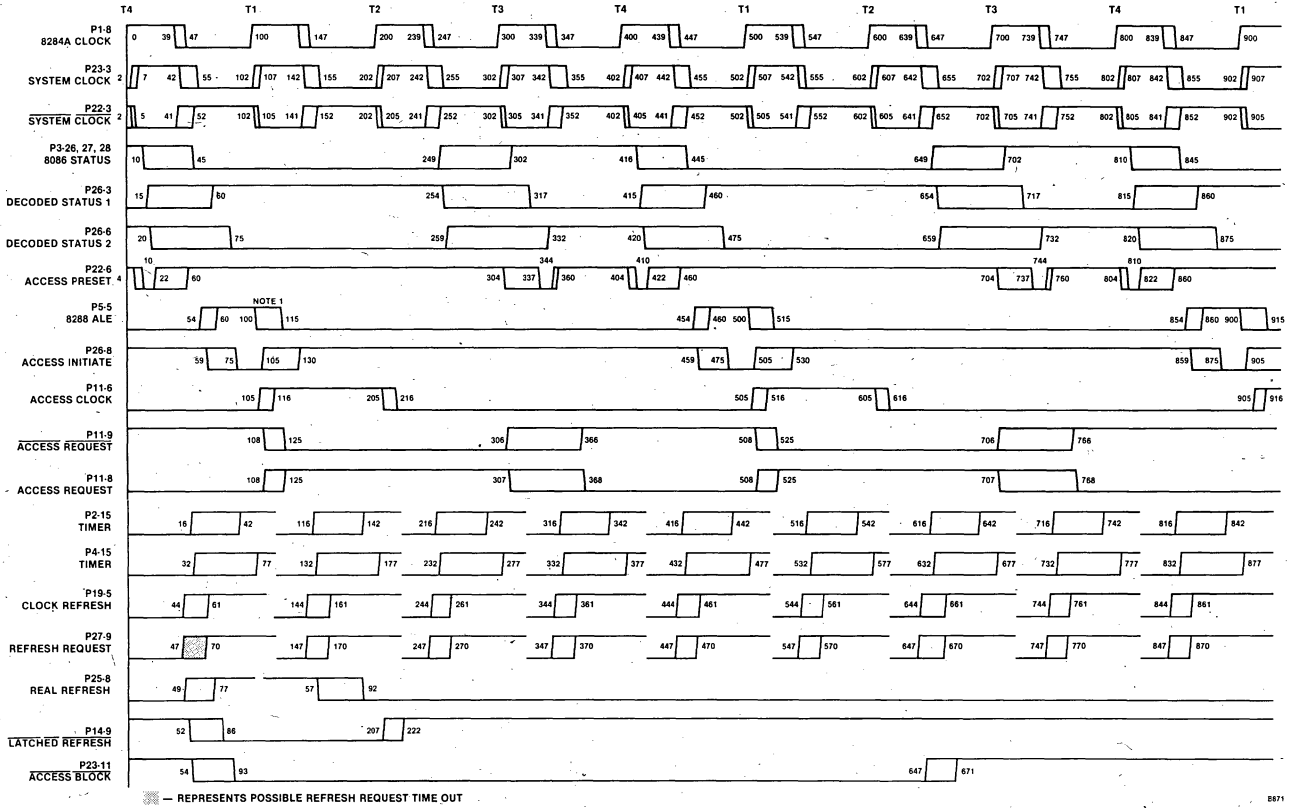


Figure 28. Refresh Cycle Followed by a Read/Write Cycle

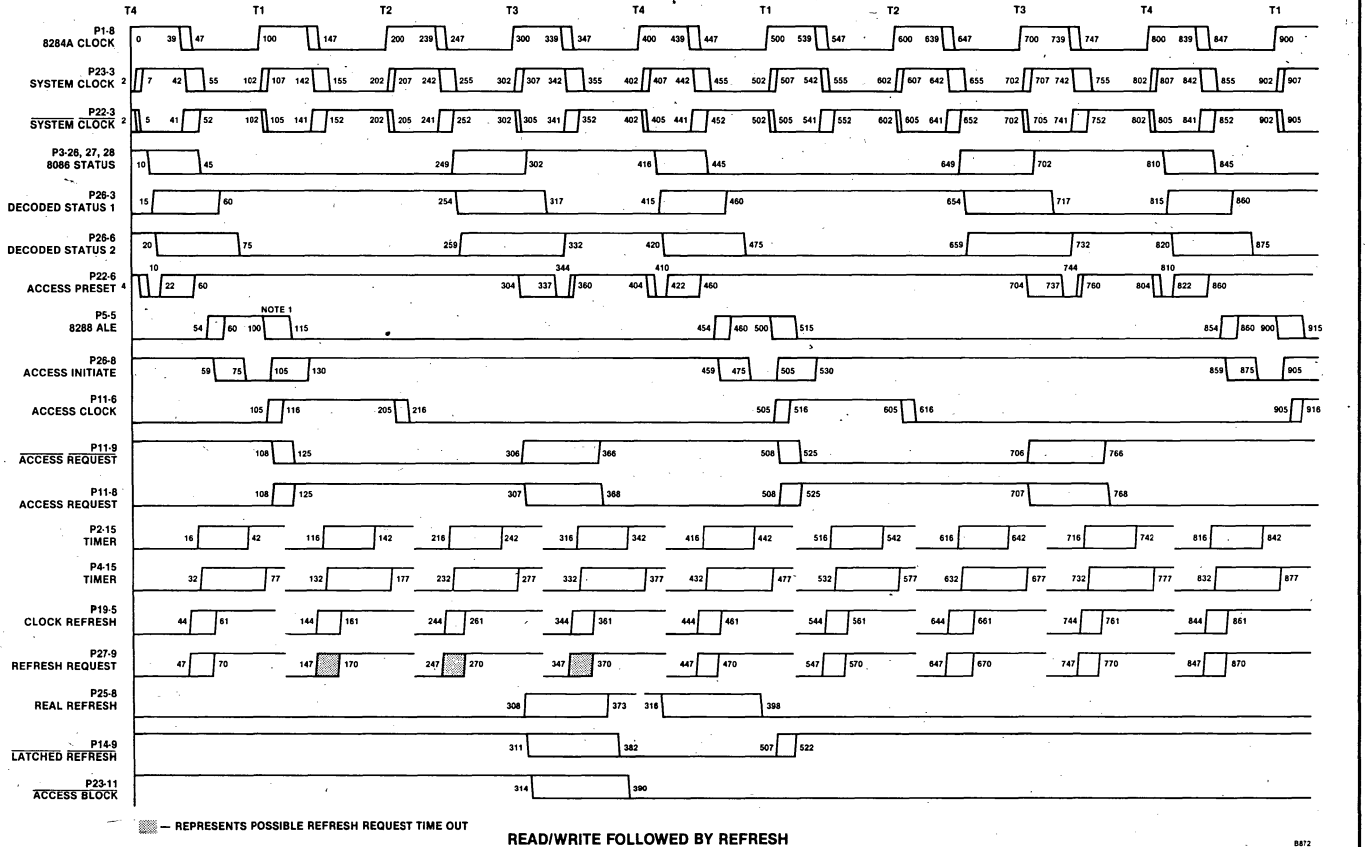


Figure 29. Read/Write Cycle Followed by a Refresh Cycle

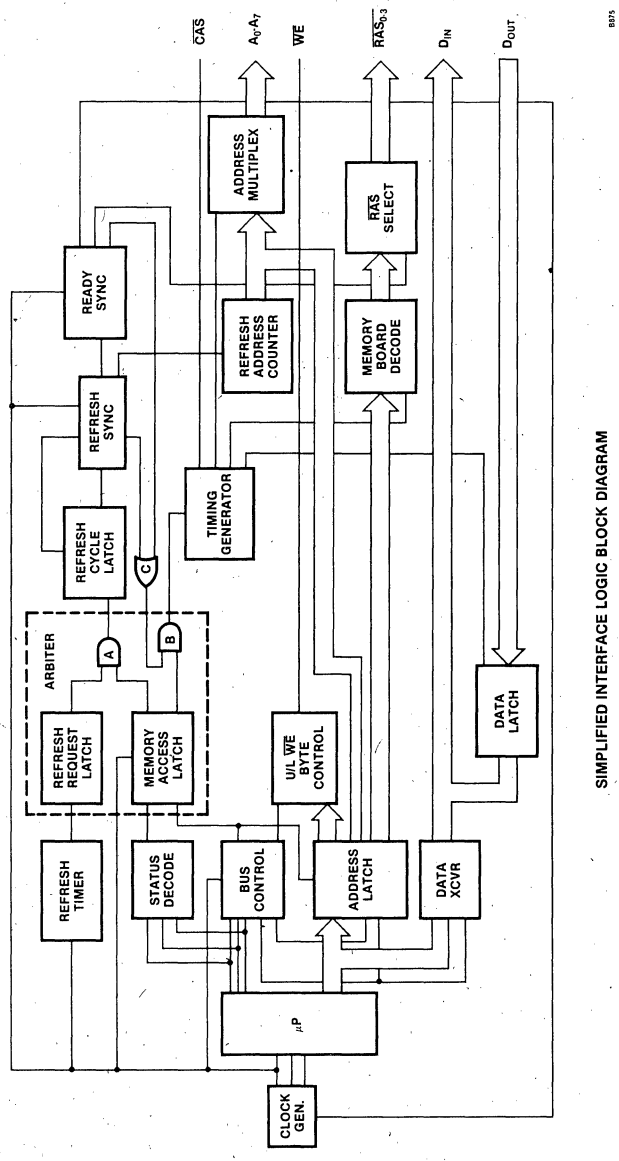


Figure 30. Simplified Interface Logic Block Diagram

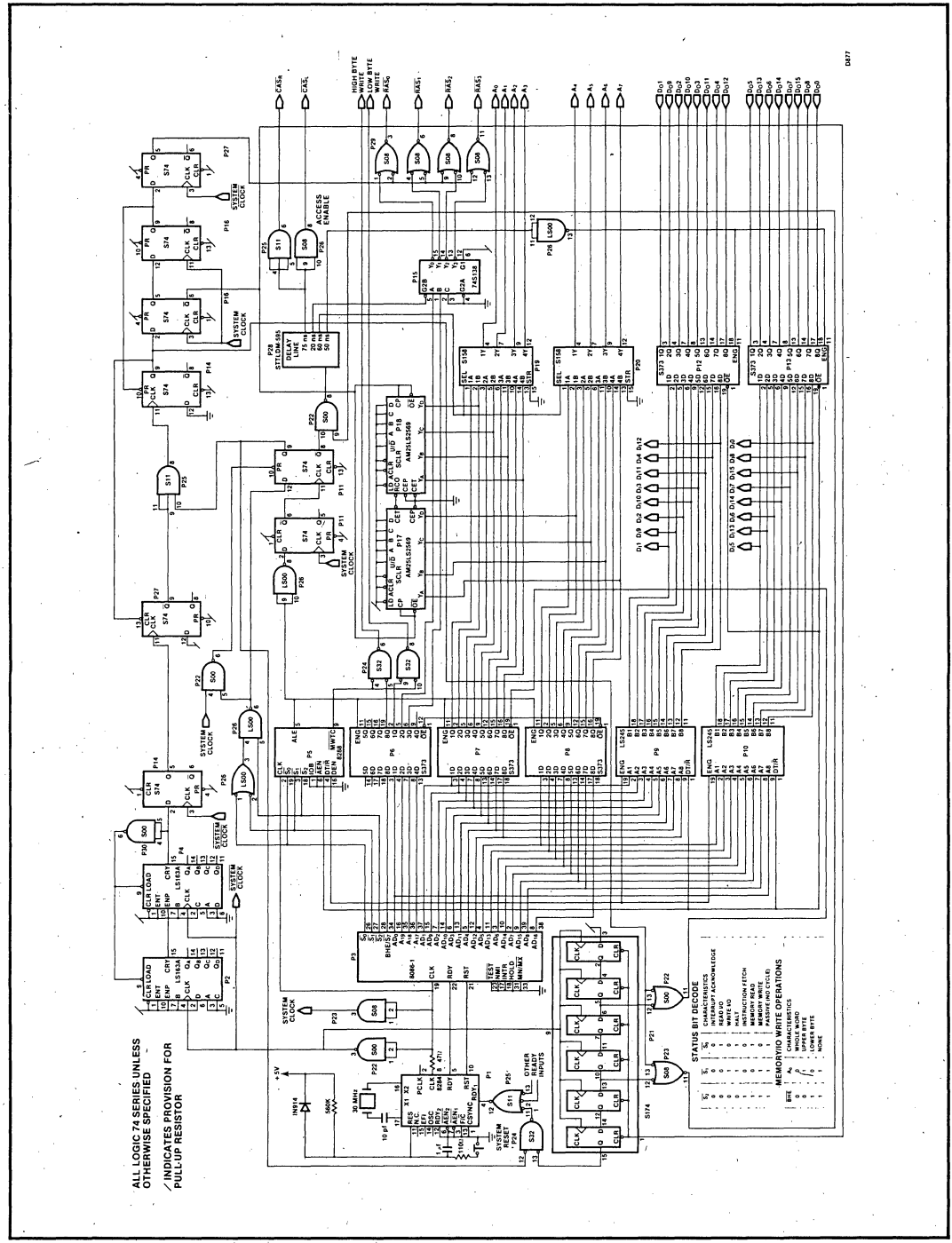


Figure 31. Logic Diagram — 10 MHz System

For example, suppose the CPU status decoder indicates a memory cycle is pending and there is no refresh cycle in progress. The status decoder outputs a bit indicating this condition to the memory access latch and is latched on the falling edge of ALE (address latch enable). After propagating through the latch, this latched memory access bit is presented to the input of AND gate B (where it will carry through the gate initiating a memory cycle, since there is no refresh cycle in progress) and its complement to AND gate A where it will block a refresh request from propagating through until the memory cycle is complete. As another example, assume that a refresh cycle is pending. The refresh timer times out, latches its output signal into the refresh request latch which subsequently presents this latched refresh request to the input of AND gate A. Here the signal is either held up or passed through depending upon the current CPU status. Assuming that there is no memory cycle in progress or that one has just ended, the AND gate passes the refresh request through to the refresh cycle latch, committing the hardware to initiate a refresh cycle and blocking any memory request that may occur until the end of the refresh cycle.

The sole purpose of the CPU status decode block is to inform the arbiter (as soon as possible) as to whether or not the pending machine cycle is going to be a memory cycle.

The bus controller provides the memory write command ($\overline{MWTC}$) and is steered to a high and/or low byte write by A0 and $\overline{BHE}$ in the byte control block. Address latch enable (ALE) used for latching valid addresses off the multiplexed bus, data enable ($\overline{DEN}$) used to enable the data transceivers, and data transmit/receive (DT/$\overline{R}$) used to control the direction of the data transceivers, are all provided by the bus control block.

The refresh sync and ready sync blocks generate several control signals for a number of functions that must execute to carry a refresh cycle to its natural end, all in synchronization with the system clock. The first signals generated are address disable — used to switch the CPU address latches into a high impedance state, and access block — used to block a memory cycle request at AND gate B. On the next rising clock edge a control signal is output that will switch the refresh address counters onto the address bus and enable a string of shift registers that comprise the ready sync to start shifting the READY bit through. Then, on the next rising edge of the clock, the refresh cycle latch is cleared, and finally on the falling edge of the clock the refresh signal is output from the ready sync block which is used by the $\overline{RAS}$ select block to enable all the $\overline{RAS}$ lines at once, simultaneously performing refresh on all four memory rows.

5.3 Schematic

Refer to the logic schematic (Figure 31) and to the block diagram in Figure 30, during the following discussion involving the conversion of logic blocks to TTL logic.

The refresh interval timer is comprised of devices P2 and P4, two 74LS163 four-bit synchronous binary counters, and one F/F from P14, a 74S74 flip-flop. The counters are cascaded and free-running, being incremented by the system clocks so as to output a refresh request pulse every 15 microseconds. This pulse is stored by P27 F/F, the refresh request latch, which is part of the arbiter.

The refresh counter is a pair of AM25LS2569 three-state binary up/down counters (located at P17 and P18) that sequence from 0 to 2^8-1 (255) and then roll over to start again. The MSB (most significant bit) of the counter is unused. With the devices' clock input tied to their $\overline{OE}$, the counters are automatically incremented at the end of a refresh cycle when the outputs are switched off the address bus by $\overline{OE}$ going high. This sets up the count to the next refresh address.

Memory address multiplexing is comprised of a pair of 74S158 quad 2:1 multiplexers (P19, P20). Inverted data output devices were selected because of their shorter propagation delay. The arrival of addresses to the memory is one of the tight timing constraints for zero WAIT states. The select line is controlled by the timing generator during a memory read or write cycle and is used to switch from row to column addresses at the appropriate time. During a refresh cycle, the select line does not change; thus, only the refresh addresses, which are wire-ORed to the row addresses are presented to the memory array.

The arbiter in this system is designed with two 74S74 F/Fs, one from P11 and the other from P27, and two gates: a 74S11 AND gate at P25 and a 74S00 NAND gate at P22. As previously discussed, the arbiter makes the decision of whether to run a memory R/W cycle or a refresh cycle, then commits the hardware to initiate the cycle decided upon. Classically a difficult choice, the task is greatly simplified by the quasi-synchronous nature of this design. Memory and refresh cycle requests never occur at or near the same time and the worst case data setup and hold times at each F/F are easily predicttable and are designed to avoid violations of these specifications. The relatively simple nature of this arbitration circuit is demonstrated by the small device count and simplicity of the method involved.

The status decode block is implemented with two NAND gates from 74S00 at P26 and one NAND gate from P22. Low power Schottky devices were required because of the limited (2 mA) drive capability of the 8086 status

lines. Through observation of the truth table for the status bits S0-S2 on the schematic and the following logic, it is apparent that NAND gate P26, pin 6 goes low during memory read, memory write, or instruction fetch cycles. This active low memory cycle status bit is latched into the access latch on the trailing edge of the clocked ALE (from S74 F/F at P11) and informs the arbiter that this memory cycle is in progress. For any other type of CPU cycle, device P26, pin 6 is high, which enables NAND gate P22, pin 5 to allow the next rising edge of the clock to preset the memory access latch, indicating to the arbiter that this is not a memory cycle.

The bus control block functions are executed with an Intel 8288 bus controller. In this circuit, ALE, DT/$\overline{R}$, $\overline{DEN}$ and $\overline{MWTC}$ are all generated at P5 from system clock and CPU status bits inputs. The $\overline{MWTC}$ is used for the write pulse to the memory array, being directed to the higher or low byte by the pair of 74S32 gates at P24 which comprised the U/$\overline{L}$ $\overline{WE}$ byte control block. ALE is transmitted to P11 latch control (ENG) input of the 74S373 three-state address latches P6-P8, thus latching valid addresses from the multiplexed CPU bus. DT/$\overline{R}$ and $\overline{DEN}$ are wired to pins 1 and 19 respectively of the pair of 8-bit 74LS245 data transceivers at P9 and P10, with DT/$\overline{R}$ controlling the direction of data flow through the devices and $\overline{DEN}$ used to enable the device output drivers in the direction selected by DT/$\overline{R}$.

Timing generation for memory array related signals are all derived from a STTLDM-595* active delay line at P28. Activated only during a memory cycle via a single input from the arbiter, this one pulse is delayed 25 ns to become the ACCESS ENABLE signal (the source of $\overline{RAS}$), 50 ns to enable the flow through memory data latches, 60 ns before switching the address multiplexer and finally delayed 75 ns before becoming the source of $\overline{CAS}$.

The ACCESS ENABLE line is connected to P5 of the 74S138 three-to-eight decoder located at P15. Configured as a two-to-four decoder by grounding the C-input and placing high order addresses A17 and A18 on the A and B inputs, P15 selects which of the four memory rows will receive a $\overline{RAS}$ signal. Once a proper output is selected, the ACCESS ENABLE signal is directed through the 74S138 to the correct row after being buffered through a 74S08 at P29. Note that one input of all the gates at P29 $\overline{RAS}$ buffers are connected together to the refresh signal. This allows simultaneous strobing of all memory $\overline{RAS}$ during a refresh cycle.

It is evident from the examples presented that the Intel 2118 and 2164A high performance DRAMs match any speed microprocessor memory requirement, fulfilling the needs at all performance levels. In particular, the 2164A DRAMs used in this 10 MHz design easily conform to the rigid requirements of this high performance system.

6 HIGH PERFORMANCE SYSTEM DESIGN CONSIDERATIONS

Designing a high performance, high speed memory system requires consideration of the following areas:

1. Skew
2. Propagation Delay
3. General Circuit Design Techniques
4. Worst-case timing analysis

6.1 Skew

Skew is the difference between maximum and minimun propagation delay through devices in a parallel path. For example, refer to Figure 32. Here signal A and signal B propagate through the same number and types of gates, each transversing a parallel path. For both signals the total mimimum delay is 6 ns and the total maximum delay is 16 ns. However, diagramming the worst case (Figure 33), the skew between these signals can be as much as 10 ns. This time (skew) adds directly to the system access/cycle time.

Capacitive loading of the STTL drivers will cause rise time degradation in the memory array, and will contribute to skew, caused by heavily loaded versus lightly loaded signals. Figure 34 displays the effects of capacitive loading of the Schottky TTL. Obviously skew needs to be minimized.

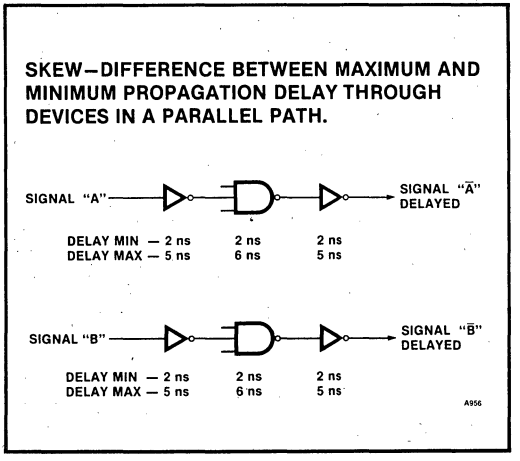


Figure 32. Skew — Variations Between Max/Min Propagation Delay

* Available from EC², San Luis Obispo, California

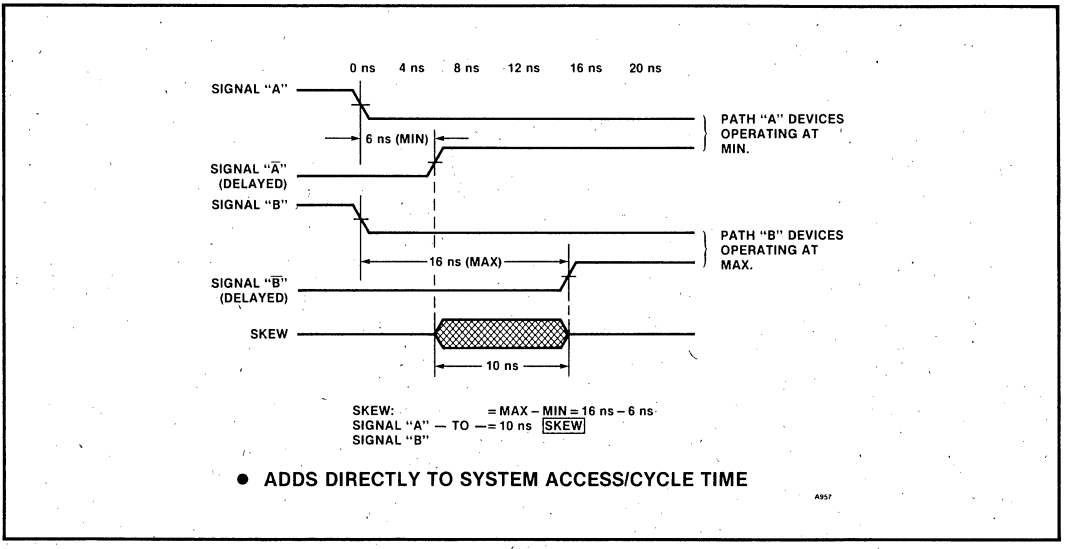


Figure 33. Skew — Adds Directly to System Access/Cycle Time

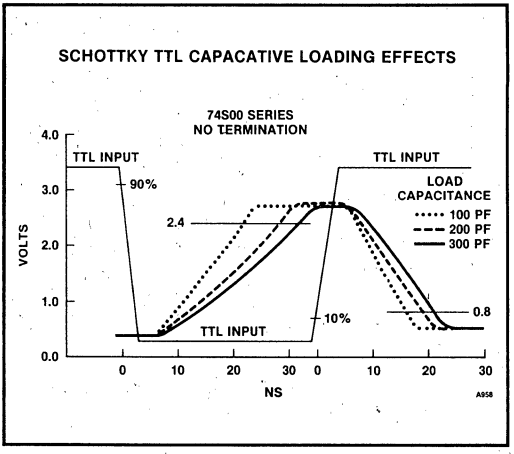


Figure 34. Schottky TTL Capacitive Loading Effects

The goal to minimize skew is achieved by observing the following guidelines:

- Select logic gates for minimum delay per function
- Place parallel paths in the same package (Device to device skew within the same package = .5 ns max for STTL, 2.0 ns max for high current drivers, i.e., 74S240.)
- Balance the output loading to equalize the capacitive delays
- Use delay lines with tight t_{prop} and t_{rise} tolerances (± 1 ns)
- Drive address and clocks from a common area on the P.C.B. to avoid circuit board trace skew due to unequal lengths of signal distribution (Figure 35).
- Localize the timing generation

6.2 Propagation Delay

Propagation delay must be determined in the critical paths to guarantee the design goals of circuit optimization and maximum performance. The following rules are generally used to determine propagation delay through the TTL devices:

- t_{prop} MAX = Data Book maximum
- t_{prop} Typical = Data Book typical
- t_{prop} MIN = ½ Data Book typical

Capacitive loads add to the propagation delays specified in the data books. The additional delay can be calculated in the following manner:

- Additional Delay = $D_C \times (C_{load} - C_{spec})$, where
  C_{load} = sum of all input capacitance plus PCB traces (≈ 2 pF/in),
  C_{spec} = specified capacitance of the driver, and
  D_C = the derating factor for the driver logic family
  - — Schottky TTL = 0.5 ns/pF
  - — Low power Schottky TTL = .1 ns/pF
  - — High current Schottky TTL = .25 ns/pF
  - — TTL = .75 ns/pF

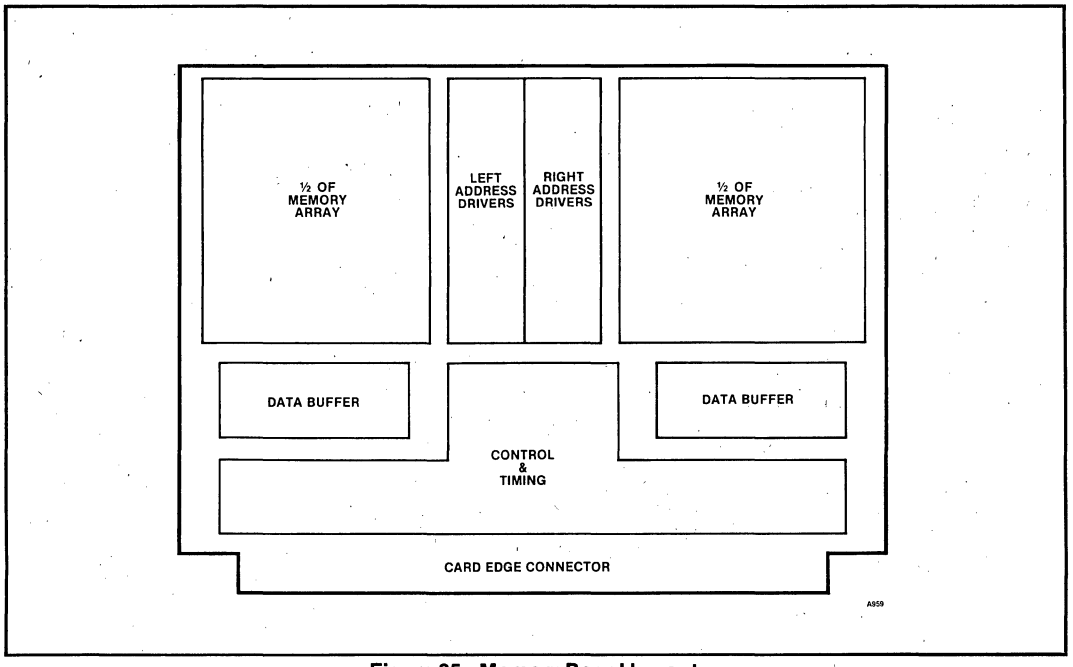


Figure 35. Memory Board Layout

6.3 Circuit Design Techniques

Optimum circuit design demands attention to the physical details of a 2164A memory system. A properly produced layout will minimize board area while yielding wider operating margins on timing and power supply requirements. The key areas of consideration are:

1. Ground and power gridding
2. Memory array/control line trace routing
3. Control logic centralization
4. Power supply decoupling

6.3.1 GROUND AND POWER GRIDDING

The power and ground network do not appear as a pure low resistance element, but rather as a transmission line because the current transients created by the RAMs are high frequency in nature. The RAMs are the lumped equivalent circuits of the power and ground transmission lines are shown in Figure 36.

The characteristic impedance of a transmission line is shown in Figure 37A. By connecting two transmission lines in parallel, the characteristic impedence is halved. The result is shown in Figure 37B.

Transient effects can be minimized by adding extra circuit board traces in parallel to reduce interconnection inductance.

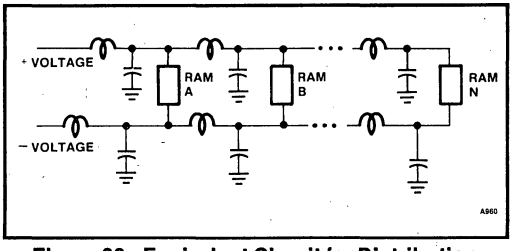


Figure 36. Equivalent Circuit for Distribution

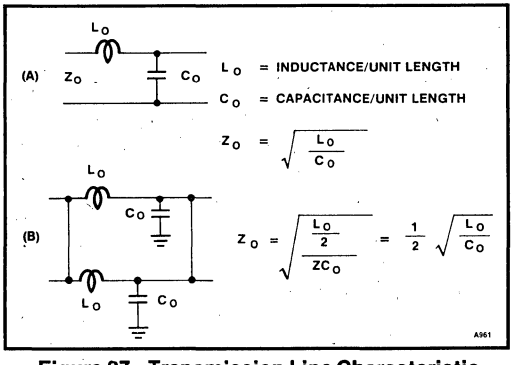


Figure 37. Transmission Line Characteristic Impedance

Extrapolation of this concept to its limit will result in an infinite number of parallel traces, or an extremely wide low impedance trace, called a plane. Distribution of power and ground voltages by plane provides the best distribution, however correct gridding can effectively approximate the benefits of planar distribution by surrounding each device with a ring of power and ground (Figure 38).

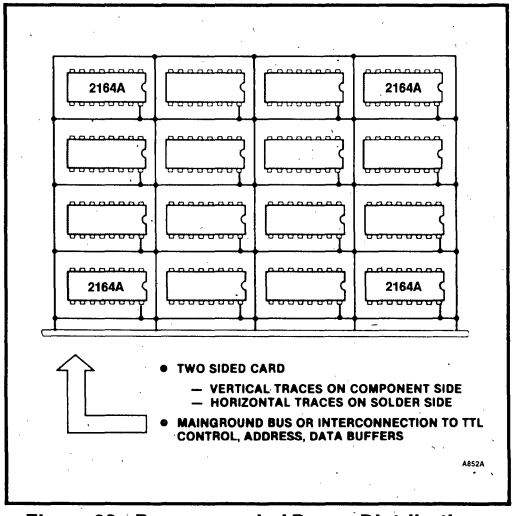


Figure 38. Recommended Power Distribution — Gridding

Improper ground and power gridding can contribute to excess noise and voltage drops if not properly structured. An example of an unacceptable method is presented in Figure 39. This type of layout promotes accumulated transient noise and voltage drops for the device located at the end of each trace (path).

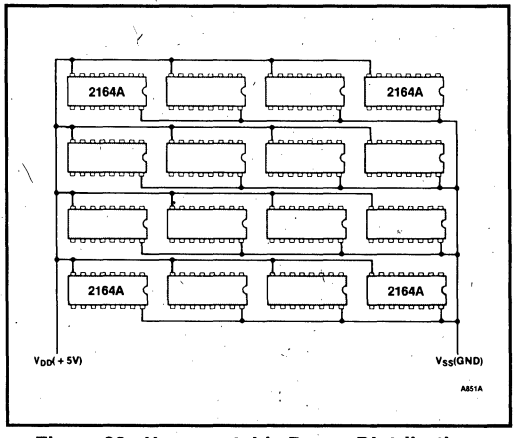


Figure 39. Unacceptable Power Distribution

6.3.2 MEMORY ARRAY/CONTROL LINE ROUTING

Address lines need to be kept as short and direct as possible. The lone serpentine line depicted in Figure 40 should be avoided, since the devices farthest away from the driver will receive a valid address at a later time than the closer ones. A better way to route address lines is in a comb like fashion from a central location as depicted in Figure 41. Routing control and address signals together from a centralized board area will also minimize skew.

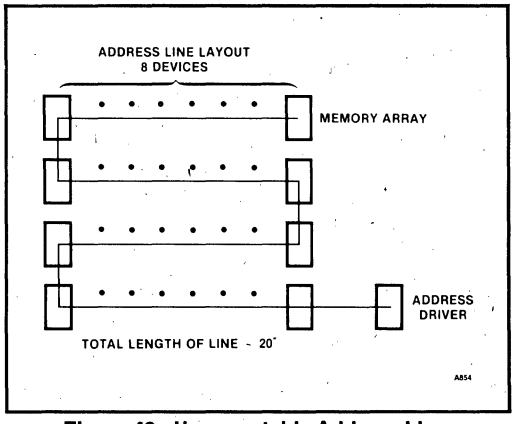


Figure 40. Unacceptable Address Line Routing (Serpentine)

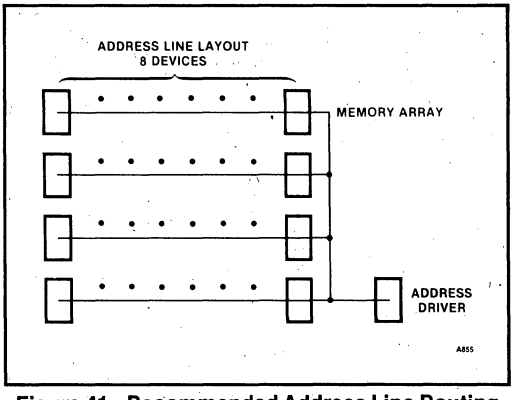


Figure 41. Recommended Address Line Routing

Allow for proper termination of all address and control lines, since a P.C.B. trace becomes a transmission line when:

$2t_{pd} \geq t_r$ or t_f

where: t_p = propagation delay down the line
t_r = rise time
t_f = fall time

The maximum unloaded line lengths not displaying transmission line characteristics are listed in Table 11. The values assume propagation delay of $\delta = 1.7$ ns/ft.

Table 11. Transmission Characteristics

Logic Family	Rise Time	Fall Time	Max. Length
54/74L	14 - 18 ns	4 - 6 ns	14.1 inches
54/74	6 - 9 ns	4 - 6 ns	14.1 inches
54H/74H	4 - 6 ns	2 - 3 ns	7.0 inches
54LS/74LS	4 - 6 ns	2 - 3 ns	7.0 inches
54S/74S	1.8 - 2.8 ns	1.6 - 2.6 ns	5.6 inches
10K ECL	1.5 - 2.2 ns	1.5 - 2.3 ns	5.3 inches
100K ECL	0.5 - 1.1 ns	0.5 - 1.1 ns	1.8 inches

The maximum length of a loaded transmission line is:

$$L_{max} = \sqrt{\left(\frac{C_D}{C_O}\right)^2 + \left(\frac{t_R \text{ or } t_F}{\delta}\right)^2} - \frac{C_D}{2C_O}$$

where C_D = Capacitive load/unit length
and C_O = Capacitance/unit length

6.3.3 CONTROL LOGIC CENTRALIZATION

Memory control logic should be strategically located in a centralized board position to reduce trace lengths to the memory array (Figure 35).

Long trace lines are prone to ringing and capacitive coupling, which can cause false triggering of timing circuits. Short lines minimize this condition and also result in less system skew.

A practical memory array layout is presented in Figure 42. Typically, this pattern and its "mirror image" are placed on each side of the memory control logic for a practical memory board design.

6.3.4 POWER SUPPLY DECOUPLING

For best results with the 2164A, decoupling capacitors are placed on the memory array board at every device location (Figure 42). High frequency 0.1 μF ceramic capacitors are the recommended type. In this arrangement each memory is effectively decoupled and the noise is minimized because of the low impedence across the

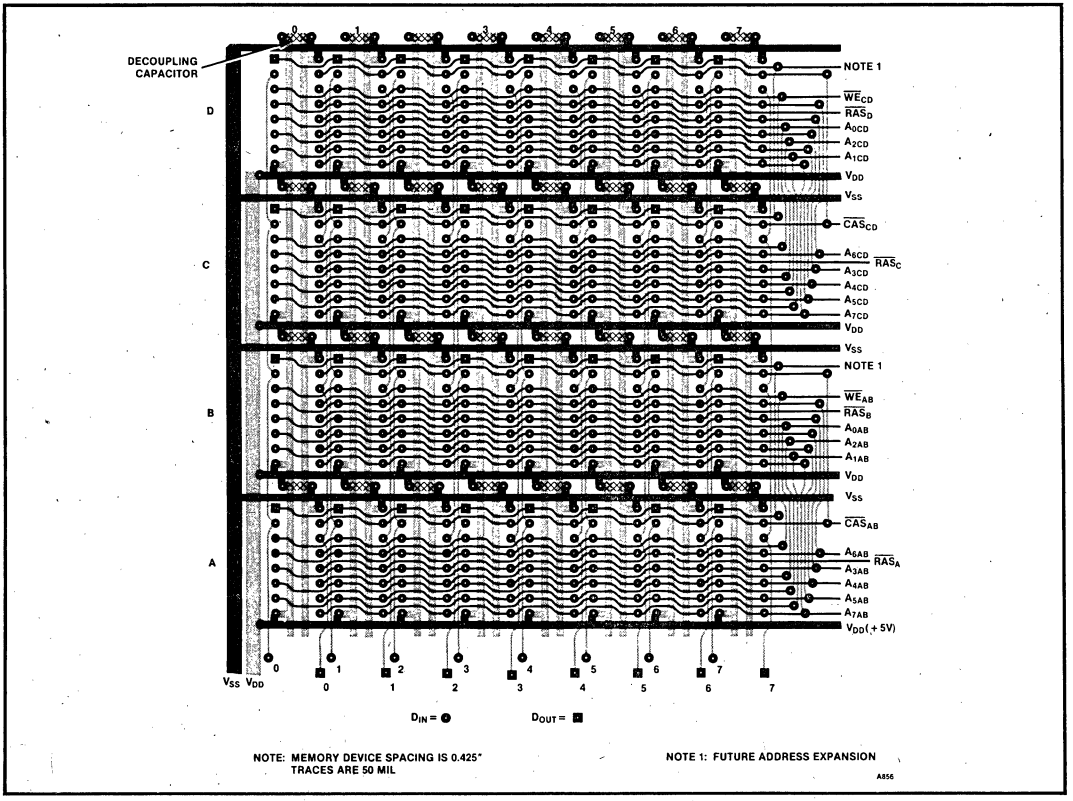


Figure 42. 2164A Memory Array P.C. Board Layout

circuit board traces. Typical V_{DD} noise levels for this array are less than 300 mV.

A large tantalum capacitor (typically one 100 μF per 32 devices) is required for the 2164A at the circuit board edge connector power input pins to recharge the 0.1 μF capacitors between memory cycles.

Decoupling is of considerable importance in circuit design in order to minimize transient effects on the power supply system. In order to determine the values for proper decoupling capacitors, the required amount of charge storage for a capacitor must first be determined in the following manner:

$Q = \Delta I \Delta T$

where: Q = charge in coulombs
ΔI = change in current is amperes
ΔT = change in time is seconds

and: $\Delta V = Q/C$

where: ΔV = voltage change in volts
and C = capacitance in farads

Assuming the following system parameters:

5 mA to 55 mA current switch for regular cycle
5 mA to 45 mA current switch for refresh cycle
1 microsecond bulk decoupling response time
260 ns cycle time
¼ of devices selected (one of four rows)

An example calculation proceeds as follows:

Q = (45-5 mA) (.3 μsec) + ¼ (55-5) mA (.7 μsec)
Q = 20.75 nanocoulombs

if V_{DD} is restricted to 100 mV (2%) then

$$C = \frac{20.8\text{ nC}}{100\text{ mV}} = .21\ \mu\text{F/Device}$$

if V_{DD} is allowed to 500 mV (10%) then

$$C = \frac{20.8\text{ nC}}{500\text{ mV}} = .042\ \mu\text{F/Device}$$

Bulk decoupling requirements are determined in a similar way:

Assuming the following:

50 μsec power supply response time
15.6 μsec refresh rate
Three refresh cycles/50 μsec period
I_{DD} standby = 5.77 mA

$$I_{DD}\ \text{STDBY} = \frac{45\text{ mA (.3 }\mu\text{sec)} + 5\text{ mA (15.3 }\mu\text{sec)}}{15.6\ \mu\text{sec}}$$

$$= 5.77\text{ mA}$$

An example calculation with ¼ devices active proceeds as shown:

$$Q = [50 - (3)(.3)]\ \mu\text{sec} \times 49.23\text{ mA } (¼) = 604\text{ nC}$$

$$\text{if V} = 100\text{ mV then C} = \frac{604\text{ nC}}{100\text{ mV}} = 6.0\ \mu\text{F device}$$

$$\text{if V} = 500\text{ mV then C} = \frac{604\text{ nC}}{500\text{ mV}} = 1.2\ \mu\text{F device}$$

The data shown in Table 12 defines the decoupling requirements of 2164A-15 and 2118-15 dynamic RAMs for a 300 ns cycle time over various device selections for a given percentage.

Cycle time has a downward scaling effect on the average operating current according to the following equation:

$$I_{DD_{AVE}} = \left[I_{DD2} \times \left(\frac{t_{RC}\text{ (spec)}}{t_{RC}\text{ (operating)}} \right) \right] + \left[I_{DD1} \times \left(1 - \frac{t_{RC}\text{ (spec)}}{t_{RC}\text{ (operating)}} \right) \right]$$

At minimum cycle time, $\frac{t_{RC}\text{ (spec)}}{t_{RC}\text{ (operating)}} = 1$,

so that worst case $I_{DD_{AVE}} = I_{DD2}$, but as the cycle time increases, $I_{DD_{AVE}}$ approaches the standby current,

Table 12. Decoupling Chart

	% Selected Devices	$\Delta V_{DD} = 2\%$		Cycle Time	$\Delta V_{DD} = 10\%$		
		C_D	C_B		C_D	C_B	
2164A-15	100	0.47	24.0	300 ns	0.11	4.8	0.1 μfd/device will work if ¼ devices are active at one time. + 100 μfd every 32 devices.
	50	0.29	12.0	300 ns	0.059	2.4	
	25	0.21	6.0	300 ns	0.042	1.2	
	12.5	0.16	3.0	300 ns	0.033	0.6	
2118-15	100	0.19	9.2	300 ns	0.038	1.84	0.1 μF/2 devices + 27 μF every 32 devices (assuming ¼ of devices active)
	50	0.10	4.6	300 ns	0.019	0.92	
	25	0.064	2.3	300 ns	0.013	0.46	
	12.5	0.048	1.15	300 ns	0.01	0.3	

becoming 6.3 mA @ 10,000 ns cycle time. Figure 5 in the 2164A data sheet depicts this scaling effect. Be sure to use the correct I_{DD} value based on specific worst case cycle time when computing specific decoupling requirements.

6.4 Timing Analysis — Determining the Worst Case

Once the control logic is designed, worst case system delays must be determined to guarantee proper circuit operation. There are two ways to perform these calculations:

1. A statistical worst case analysis (or the Monte Carlo method) which assumes that all devices probably won't be in their worst case condition at the same time.

   It is determined by the following formula:

   STATISTICAL WORST CASE

   $= \sqrt{\Sigma(A)^2 + (B)^2 + (C)^2}$ MAX STTL DELAYS + TYPICAL STTL DELAYS

   $+ \sqrt{\Sigma (A)^2 + (B)^2 + (C)^2}$ SKEW DELAYS

   + Σ DELAYS DUE TO CAPACITIVE LOADING

   + *MAXIMUM* DELAY ACCESSING MEMORY DEVICE

   WHERE (A), (B) OR (C) = MAX-TYP OR TYP-MIN

2. A true worst case analysis, using specified maximum and minimum delays for peripheral circuits plus all delays due to capacitive loading from device inputs and distributive capacitance in PC board etched con ductors. The following formula appears here:

   WORST CASE

   = Σ MAX STTL DELAYS + SKEW DELAYS (PERIPHERAL DEVICES)

   + Σ DELAYS DUE TO CAPACITIVE LOADING (INPUTS + P.C.B. TRACES)

   + *MAXIMUM* DELAY ACCESSING MEMORY DEVICE (T_{RAC} OR T_{CAC})

Since the statistical approach can be justified only in large systems with hundreds or thousands of components, the timing calculations used in all of the previous examples are based on a true worst case analysis. Capacitive delay is formulated from the equations in Section 6.1.2.

In summary, the following rules and guidelines apply to worst case analysis:

1. All propagation delays are from the industry TTL books.

   Max = Data book maximum

   Typ = Data book typical

   Min = ½ Data Book Typical

2. Skew device to device in same package = 0.5 ns Max for Schottky TTL and 2 ns for 74S240.
3. STTLDM-595 is a special delay line with active outputs. Propagation delay = ± 1 ns per tap (i.e., 75 ± 1 ns). (10 MHz system.)
4. Capacitive loads add 0.5 ns/pF to propagation delays specified in device spec (i.e., 74S04 is specified at 5.0 ns Max @ 15 pF. At 25 Pf propagation delay is 5.5 ns) Schottky TTL input capacitance is 3 pF. PCB traces are 2 pF/inch.
5. PCB etch delay adds little or no skew to array address/control timing signals. It adds 4 ns, however, in the overall access time data path.
6. Timing components are immediately adjacent to each other, making PCB etch delays in delay timing chain negligible (exception is timing tap used to terminate delay line latch).

7. SUMMARY

The Intel 2164A and 2118 DRAMs meet all microprocessor system requirements, offering high density, speed, low power and ease of use. Follow the system design guidelines presented to create a harmonious microprocessor memory design.

8. REFERENCES

AP-75	Application of the Intel 2118 16K Dynamic RAM
AP-131	2164A 64K Dynamic RAM Device Description
AP-92A	Interfacing Dynamic RAMs to iAPX 86/88 Systems Using the Intel 8202A and 8203
AP-46	Error Detecting and Correcting Codes Part #1
AP-73	ECC #2 Memory System Reliability with Error Correction

intel

APPLICATION NOTE

AP-46

Error Detecting and Correcting Codes Part 1

JOE ALTNETHER
MEMORY COMPONENTS APPLICATIONS

9800752

INTRODUCTION

Complex electronic systems require the utmost in reliability. Especially when the storage and retrieval of critical data demands faultless operation, the system designer must strive for the highest reliability possible. Extra effort must be expended to achieve this high reliability. Fortunately, not all systems must operate with these ultra reliability requirements.

The majority of systems operate in an area where system failure ranges from irritating, such as a video game failure, to a financial loss, such as a misprinted check. While these failures are not hazardous, reliability is important enough to be designed into the system.

A memory system is one of the system components for which reliability is important. Also, it is one of the few system components which can be altered to greatly enhance its reliability. The purpose of this report is to examine different methods of error encoding, especially Error Correction Codes (ECC), to increase the reliability of the memory system.

SYSTEM RELIABILITY

Individual device reliability is the foundation of memory system reliability. Reliability is expressed as mean time between failures. The mean time between failures (MTBF) of a system is a function of the number of devices and the device failure rate. Failure rate of the memory device can be obtained from the reliability report on the specific device. MTBF of the device is:

$$T_D = \frac{1}{\lambda} \qquad [1]$$

where T_D = MTBF of the *device*

λ = *device* failure rate (%/1000 hrs)

and MTBF of the *system* is approximately:

$$T_S = \frac{T_D}{D} \qquad [2]$$

where T_S = MTBF of the *system*

D = number of devices in the *system*

As the number of devices required to construct a system becomes larger, the system MTBF becomes smaller.

A plot of system MTBF as a function of the number of memory devices is shown in Figure 1 for different failure rates. Included for reference are the failure rates of the Intel® 2104A 4Kx1 RAM and the Intel® 2117 16Kx1 RAM. Using RAMs which are organized one bit wide, the amount of devices required for a system is calculated by multiplying the number of words by the word length

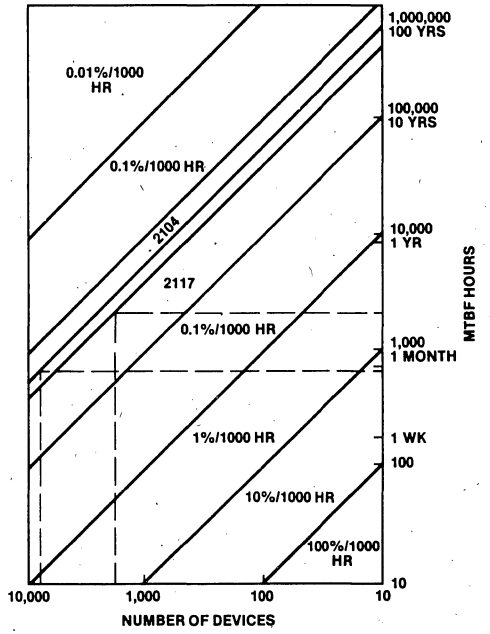


Figure 1. System Reliability vs Number of Devices

and dividing by the size of the RAM. To illustrate, assume a 1 megaword memory system with a word width of 32 bits, implemented with Intel® 2104A 4Kx1 RAMs. The number of required devices is:

$$D = \frac{1{,}048{,}576 \times 32}{4{,}096} = 8{,}192 \text{ devices}$$

Prediction of failure for this system, shown in Figure 1, is 667 hours or 28 days — assuming continuous use and worst case temperature.

Equation 2 showed that system MTBF is increased when fewer devices are used. A one megaword memory having 32 bit wide words can be constructed with Intel 2117 16K RAMs. In this case one fourth as many devices are required — 2048 devices. From Equation 2, the expected MTBF should be four times as large — 2668 hours. It is not. The failure rate from Figure 1 for this system is 2000 hours. Different device failure rates account for this difference. The failure rate of the 16K is not *yet* equal to that of the 4K. Memory device reliability is a function of time as shown in Figure 2. Reliability improvement often is a result of increased experience in manufacturing and testing. In time, the failure rate of the 16K will reach that of the 4K and one fourth as many devices will result in a system MTBF approximately four times better.

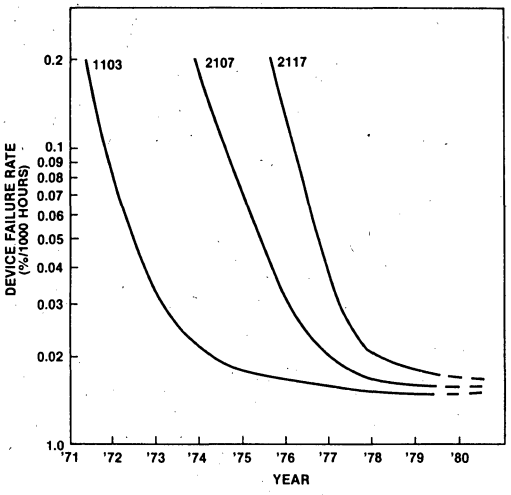


Figure 2. Device Failure Rate as a Function of Time.

The failure rate of a system without error correction will follow a similar curve over time. Indeed, in very large systems built with large numbers of devices, the *system* failure rate may be intolerable, even with very reasonable *device* failure rates. To increase the system reliability beyond the device reliability, *redundancy coding techniques* have been developed for detecting and correcting errors.

REDUNDANCY CODES

Redundancy codes add bits to the data word to provide a validity check on the entire word. These additional bits, used to detect whether or not an error has occurred, are called encoding bits. With M data bits and K encoding bits, the encoded word width is N bits. Shown in Figure 3 is the form of the encoded word.

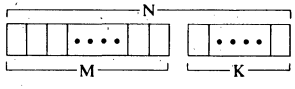


Figure 3. Encoded Word Form

Mathematically, N is related to M and K by:

$$N = M + K \qquad [3]$$

where N = number of bits in the encoded word

M = number of data bits

K = number of encoding bits

Exactly how K is related to M, and the number of required K bits depends on several factors which will be described later.

One measure of a code is its efficiency. Efficiency is the ratio of the number of bits in the encoded word to the number of bits of data:

$$E = \frac{N}{M}$$

Substituting $N = M + K$:

$$E = \frac{M + K}{M} \qquad [4]$$

where E = efficiency

All of the data are contained in the M bits. The K bits contain no data, only validity checks. To maximize the amount of data in the encoded word, the number of K bits must be minimized. Examination of Equation 4 shows that the minimum value of K is zero. With K equal to zero, the efficiency is unity. Efficiency is maximized, but the word has no encoding bits. Therefore, it has no capability to detect an error.

As an example, consider a two bit word. It can assume 2^2 or 4 states, which are:

State 1	00
State 2	01
State 3	10
State 4	11

Figure 4. All States of a Two-Bit Word

All possible states have been used as data; consequently any error will cause the error state to be identical to a valid data state.

The mechanics of the encoding bits create encoded words such that every valid encoded word has a set of error words which differ from all valid encoded words. When an error occurs, an error word is formed and this word is recognized as containing invalid data.

By adding one K bit to the two bit word error detection becomes possible. The value of the K bit will be such that the encoded word has an odd number of ONES. As will be explained later, this technique is "odd" parity.

The sum of the ONES in a word is the *weight* of the word. Parity operates by differentiating between odd and even weights. The encoded word will always have an odd weight as a result of having an odd number of ONES.

If a single bit error occurs, one bit in the encoded word will change state and the word will have an even weight. Then in this example, all encoded states with an even weight — an even number of ones — are error states.

The value of the encoding bit or parity bit is found by counting the number of ones — calculating the weight — and setting the value of K to make the weight of the encoded word odd. Referring to Figure 4, State 1 was 00,

the weight of this word is 0, so K is set to 1 and the weight of the encoded word is odd. State 2 is 01, the weight is odd already, so K is set to 0. The weight of State 3 is identical to that of State 2 so K is again set to 0. Finally, State 4 has an even weight (1 + 1 = 2), thus K is 1. The encoded states of the two bit data word are listed in Figure 5.

	Data	Encoding Bit
State 1	00	1
State 2	01	0
State 3	10	0
State 4	11	1
	M	K
	N	

Figure 5. Code Bits for All Possible States of a Two-Bit Word

To illustrate the error detection, Figure 6a lists all states of the encoded data word and all possible single bit errors. Because the encoded word is 3 bits long, there are only 3 possible single bit errors for each encoded state.

	A	B	C	D
Encoded States	001	010	100	111
Error States	000	000	000	011
	011	011	101	101
	101	110	110	110

Figure 6a. All Possible Single-Bit Errors

Notice that every error state has an even weight, while the valid encoded states have odd weights.

Converting all the values of these states to decimal equivalents makes the errors more obvious as shown in Figure 6b.

Valid States	1	2	4	7
Error States	0	0	0	
	3	3		3
	5		5	5
		6	6	6

Figure 6b. Decimal Representation of Errors

No error state is the same as any valid encoded state. Identical error states can be found in several columns. The fact that some error states are identical prevents identification of the bit in error, and hence correction is impossible. Importantly though, error detection has occurred.

Figure 6a demonstrates another property of codes. Every error state differs from its valid encoded state by one bit, whereas each of the encoded states differs from the others by two bits. Examine the encoded states labeled B and D in Figure 6a and shown in Figure 7.

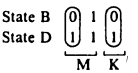


Figure 7. Bit Difference.

These two states have two bit positions which differ. This *difference* is defined as *distance* and these two states have a distance of two. Distance, then, is the number of bits that differ between two words. The encoded words have a minimum distance of two. Longer encoded words may have distances greater than two but never less than two if error detection is desired. The error states have a minimum distance of one from their valid encoded state.

A minimum distance of two between encoded states is required for error detection. A re-examination of a word with no encoding bits shows that the states have a minimum distance of 1 (see Figure 8). No error detection is possible because any single bit error will result in a valid word.

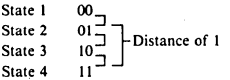


Figure 8. Minimum Distance of a Two-Bit Word

PARITY

A minimum distance of two code is implemented with Parity. Refer to previous section for an explanation. Parity is generated by exclusive-ORing all the data bits in the word, which results in a parity bit. This parity bit is the K encoding bit of the word. If the word contains M data bits, the parity bit is:

$$C = b1 \oplus b2 \oplus b3 \oplus \ldots \oplus bm$$

where C = parity bit

b = value in the bit position

The parity bit combines with the original data bits to form the encoded word as shown in Figure 9. Encoded words always have either "odd" parity, which is an odd number of 1s (an odd weight) or "even" parity which is an even number of 1s (an even weight). Odd and even parity are never intermixed, so that the encoded words all have either odd or even parity — never both.

When the encoded word is fetched, the parity bit is removed from the word and saved. A new parity bit is generated from the M bits. Comparing this new parity bit with the stored parity bit determines if a single bit error has occurred.

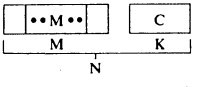


Figure 9. Encoded Word Form

Consider the two bit data word whose value is "01." Exclusive-NORing the two data bits generates a parity bit which causes the encoded word to have odd parity:

$$\overline{C} = \overline{0 \oplus 1}$$

$$\overline{C} = 0$$

The encoded word becomes:

$$\underset{\overline{0\ 1}}{M} \quad \underset{\overline{0}}{K}$$

parity

LSB of data

Assume that an error occurs and the value of the word becomes "110." Stripping off the parity bit and generating a new parity bit:

$$\text{transmitted parity} = 0$$

$$\text{transmitted word} = 11$$

$$\text{new parity of transmitted word} = \overline{1 \oplus 1} = 1$$

$$\text{generated parity} \neq \text{transmitted parity}$$

Note that the error could have occurred in the parity bit and the final result would have been the same. An error in the encoding bit as well as in the data bits can be detected.

Although parity detects the error, no correction is possible. This is because each valid word can generate the same error state. Illustration of this is shown in Figure 10.

Correct Word with Parity	Possible Single Bit Error
0 0 1	0 1 1
1 1 1	0 1 1
0 1 0	0 1 1

Figure 10. Possible Errors

Each of the errors is identical to the others and reconstruction of the original word is impossible.

Parity fails to detect an *even* number of errors occurring in the word. If a double bit error occurs, no error is detected because two bits have changed state, causing the weight of the word to remain the same.

Using the encoded word "010" one possible double bit error (DBE) is:

1 1 1

parity

Checking parity:

$$\overline{C} = \overline{1 \oplus 1} = 1$$

The transmitted parity and the regenerated parity agree. Therefore the technique of parity can detect only an *odd* number of errors.

In summary, single bit parity will detect the majority of errors, but cannot be used to correct errors. Using parity introduces a measure of confidence in the system. Should a single bit error occur, it will be detected.

ERROR CORRECTION

Classical texts on error coding contain proofs showing that a minimum distance of three between encoded words is necessary to correct errors. While this fact does not describe the code, it does give an indication of the form of the code.

Correcting errors is not as difficult as it first appears. As a result of a paper published by R. W. Hamming on error correction the most widely used type of code is the "Hamming" code. Using the same technique as parity, Hamming code generates K encoding bits and appends them to the M data bits. As shown in Figure 11, this N bit word is stored in memory.

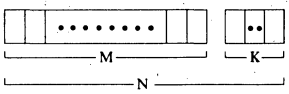


Figure 11. Encoded Word Form

Thus far the mechanism is similar to parity. The only difference is the number of K bits and how they relate to the M data bits.

When the word is read from memory, a new set of code bits (K′) is generated from the M′ data bits and compared to the fetched K encoding bits. Comparison is done by exclusive-ORing as shown in Figure 12. Like parity the result of the comparison — called the syndrome word — contains information to determine if an error has occurred. Unlike parity, the syndrome word also contains information to indicate which bit is in error.

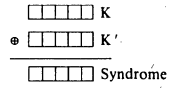


Figure 12. Syndrome Generation

The *syndrome word* is therefore K bits wide. The syndrome word has a range of 2^K values between 0 and $2^K - 1$. One of these values, usually zero, is used to indicate that no error was detected, leaving $2^K - 1$ values to indicate which of the N bits was in error. Each of these $2^K - 1$ values can be used to uniquely describe a bit in error. The range of K must be equal to or greater than N. Mathematically, the formula is:

$$2^K - 1 \geq N$$

$$\text{but } N = M + K$$

$$\text{and } 2^K - 1 \geq M + K \qquad [5]$$

Equation 5 gives the number of K bits needed to correct a single bit error in a word containing M data bits. Ranges of M for various values of K are calculated and listed in Table I.

K	Single Correct/ Single Detect ⩽ M <		Single Correct/ Double Detect ⩽ M <	
4	4	11	1	3
5	12	26	4	10
6	27	57	11	25
7	58	120	26	56
8	121	245	57	119

Table I.
Range of M for Single Correct/Single Detect or Double Detect Codes for Values of K

To detect and correct a single bit error in a 16 bit data word, five encoding bits must be used. As a result, the total number of bits in the encoded word is 21 bits.

Efficiencies of single detect — parity — and single detect/single correct codes as a function of the number of data bits are shown in Figure 13. For large values of M, the efficiency of single detect/correct is approximately equal to that of the single detect code — parity.

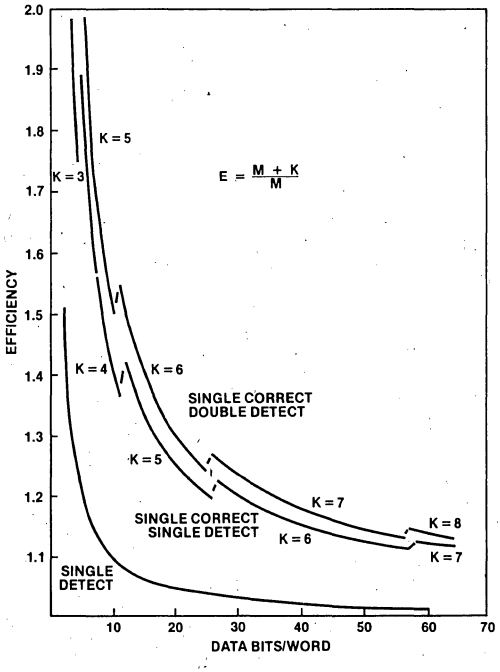


Figure 13. Code Efficiency vs Data Word Size

CODE DEVELOPMENT

Contained in the syndrome word is sufficient information to specify which bit is in error. After decoding this information, error correction is accomplished by inverting the bit in error. All bits, including the encoding bits — called check bits — are identified by their positions in the word.

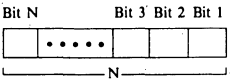


Figure 14. Positional Representation of Bits in the Word

Bits in the N bit word are organized as shown in Figure 14. Bit numbers shown in decimal form are converted to binary numbers. From equation 5, this binary number will be K bits wide. In Figure 15 is an example using a 16 bit data word. Because there are 16 data bits, M equals 16, K equals 5 and N equals 21. Shown in Figure 15 the word is binary equivalent of the position. Notice that where the M and the K bits are located is not yet specified.

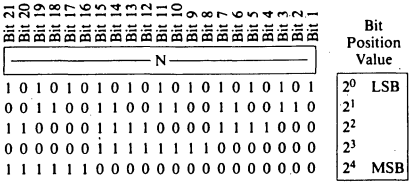


Figure 15. Binary Value of Bit Position.

The syndrome word is the difference between the fetched check bits and the regenerated check bits. Identification of the bit in error by the syndrome word is provided by the binary value of the bit position. The syndrome word is generated by exclusive-ORing the fetched check bits with the regenerated check bits. Any new check bits that differ from the old check bits will set 1s in the syndrome word. To identify bit 3 as a bit in error, the syndrome word will be 00011, which is the binary value of the bit position. Weight is determined only by the 1s in the bit position chart in Figure 15, so they are replaced with an X and the 0s are deleted. The result is shown in Figure 16.

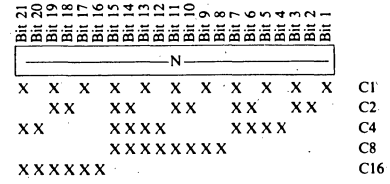


Figure 16. Relationship of Data Bits and Check Bits.

Check bit function is now defined by equating the check bits to the powers of 2 in the binary positions. Each check bit will operate on every bit position that has an X in the row shown in Figure 16. Five bit positions — 1, 2, 4, 8, and 16 — have only one X in their columns. The corresponding check bits are in these respective locations. Check bit C1 is stored in Bit Position 1, C2 is stored in Bit Position 2, and C4, C8, and C16 are stored in positions 4, 8, and 16 respectively. Because each of these positions has one X in the column, the check bits are independent of one another. If a check bit fails, the syndrome word will contain a single "1." A data bit failure will be identified by two or more "1s" in the syndrome word.

The data bits are filled in the positions between the check bits. The least significant bit (LSB) of data is located in position 3.

Data Bit 2 is stored in position 5 — position 4 is a check bit. Figure 17 shows the positions of data bits and check bits for sixteen bits of data.

When the check bits are generated for storage, bits 1, 2, 4, 8, and 16 are omitted from the generation circuitry because they do not yet exist, being the result of generation.

Parity check on the specified bits is used to generate the check bits. Each check bit is the result of exclusive-ORing the data bits marked with an "X" in Figure 18. Check bits are generated by these logic equations:

$$C1 = M1 \oplus M2 \oplus M4 \oplus M5 \oplus M7 \oplus M9 \oplus M11 \oplus M12 \oplus M14 \oplus M16$$

$$C2 = M1 \oplus M3 \oplus M4 \oplus M6 \oplus M7 \oplus M10 \oplus M11 \oplus M13 \oplus M14$$

$$C4 = M2 \oplus M3 \oplus M4 \oplus M8 \oplus M9 \oplus M10 \oplus M11 \oplus M16 \oplus M16$$

$$C8 = M5 \oplus M6 \oplus M7 \oplus M8 \oplus M9 \oplus M10 \oplus M11$$

$$C16 = M12 \oplus M13 \oplus M14 \oplus M15 \oplus M16$$

How the Hamming code corrects an error is best shown with an example. In this example, a data word will be assumed, check bits will be generated, an error will be forced, new check bits will be generated, and the syndrome word will be formed. Assuming the 16-bit data word

0101 0000 0011 1001

Check bits are generated by overlaying the data word on the Hamming Chart of Figure 16 and performing an odd parity calculation on the bits matching the "Xs."

The simplest mechanism to calculate the check bits is shown in Figure 18. The data word is aligned on the chart. Because weight and hence parity are affected only by "1s," only columns containing "1s" are circled for identification. The check bits are the result of odd parity generated on the rows. For example, the C1 row has three "Xs" circled; therefore C1 is 0 to keep the row parity odd. In this example, all other rows contain an even number of circled "Xs;" therefore the remaining check bits are "1s." These check bits are incorporated into the data word, forming the encoded word. Performing this function, the 21 bit encoded word is:

	C16		C8		C4		C2	C1
0101	0	1	000	0011	1	100	1	1	1	0

Forcing an error with bit position 7 — data bit 4:

	C16		C8		C4		C2	C1
0101	0	1	000	0011	1	000	1	1	1	0

A new set of check bits is generated on the error word as shown in Figure 18 and is:

C16	C8	C4	C2	C1
1	1	0	0	1

When the new check bits are exclusive-ORed with the old check bits, the syndrome word is formed:

	C16	C8	C4	C2	C1	
	1	1	0	0	1	New check bits
⊕	1	1	1	1	0	Old check bits
	0	0	1	1	1	

The result is 00111, indicating that bit position 7 — data bit 3 — is in error. Bit position of the error is indicated directly by the syndrome word.

While this "straight" Hamming code is simple, implementing it in hardware does present some problems. First, the number of bits exclusive-ORed to generate parity is not equal for all check bits. In the preceding example, the number of bits to be checked ranges from 10 to 5. The propagation delay of a 10 input exclusive-OR is much longer than that of a 5 input exclusive-OR. The system must wait for the longest propagation delay path, which slows the system. Equalizing the number of bits checked will optimize the speed of the encoders.

16	15	14	13	12		11	10	9	8	7	6	5		4	3	2		1			Data Bits
					C16								C8				C4		C2	C1	Check Bits
21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Position

Figure 17. Data and Check Bit Positions in the Encoded Word.

16	15	14	13	12	C16	11	10	9	8	7	6	5	C8	4	3	2	C4	1	C2	C1	
X		X		X		X		X		X		X		X		X		X		X	C1
		X	X			X	X			X	X			X	X			X	X		C2
X	X					X	X	X	X					X	X	X	X				C4
						X	X	X	X	X	X	X	X								C8
X	X	X	X	X	X																C16

Figure 18a. Hamming Chart.

Bit Position	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
Data Bit	16	15	14	13	12	C16	11	10	9	8	7	6	5	C8	4	3	2	C4	1	C2	C1		
	X		X		X		X		X		X		X		X		X		X			C1	0
		X		X			X	X			X	X			X	X			X			C2	1
	X	X					X	X	X	X					X	X	X					C4	1
							X	X	X	X	X	X	X									C8	1
	X	X	X	X	X																	C16	1
Word as Stored	0	1	0	1	0		0	0	0	0	0	1	1		1	0	0		1				
Word as Fetched	0	1	0	1	0	1	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0		
	X		X		X		X		X		X		X		X		X		X			C1	1
		X	X	X			X	X			X	X			X	X			X			C2	0
	X	X					X	X	X	X					X	X	X					C4	0
							X	X	X	X	X	X	X									C8	1
	X	X	X	X	X																	C16	1

Figure 18b. Check Bit Generation.

Secondly, two bits in error can cause a correct bit to be indicated as being in error. For example, if check bits C1 and C2 failed, data bit 1 would be flagged as a bit in error.

Because of these two difficulties, the Error Correction Code (ECC) most commonly used is a "modified" Hamming code is most widely used which will detect double bit errors and correct single bit errors.

SINGLE BIT CORRECT/ DOUBLE BIT DETECT CODES

Modern algebra can be used to prove that a minimum distance of four is required between encoded words to detect two errors or correct a single bit error. An excellent text on this subject is *Error Correcting Codes* by Peterson and Weldon.

One possible double bit error is two check bits. Using straight Hamming code, the circuit would "correct" the wrong bit. Double error detection techniques — modified Hamming codes — prevent this by separating the encoded words by a minimum distance of four. As a result each data bit is protected by a minimum of three check bits, so that the syndrome word always has an odd weight. Therefore, even weight syndrome words cannot be used. When two check bits fail, the syndrome word has two "1s" or an even weight. Even weight is detectable as a double bit error by performing a parity check on the syndrome word. If two data bits fail, again the syndrome word has an even weight — a detectable error.

Adding one additional check bit to the correction check bits provides the capability to detect double bit errors. The number of encoding or check bits required to detect double bit errors and correct single bit errors is:

$$2^M \leqslant \frac{2^{N-1}}{N}$$

Substituting M + K for N:

$$2^{K-1} \geqslant M + K \qquad [6]$$

Equation 6 is similar to equation 5, which describes single bit correct and detect except for the left side of the inequality, which shows one additional encoding bit is required. For single bit detect and correct the left side of the inequality was 2^K. Table I also lists the ranges of M for values of K, for a direct comparison to single bit detect and single bit correct codes.

Figure 13 includes the efficiency curve for single bit correct/double bit detect (SBC/DBD) codes for values of M. As would be expected, because of the additional encoding bit the efficiency is slightly lower. For large values of M, the efficiency of this code approaches unity like the two other curves.

Syndrome words for the SBC/DBD code are developed like the straight Hamming code, except that syndrome words do not map directly to bit positions. The syndrome word has an odd weight and does not increment like straight Hamming code. In addition, implementation considerations can impose constraints. For example, the 74S280 parity generator is a nine input device. If a check bit is generated from ten bits, extra hardware is required.

Empirical methods can be used to form the syndrome words. All possible states of the encoding bits are listed and those with an even weight are stricken from the list. Again like Hamming code, states which have a weight of one are used for syndrome words for check bits. For a sixteen bit data word, six check bits are required. Figure 19 lists the possible states of syndrome words for a 16 bit data word.

C6	C5	C4	C3	C2	C1
1	1	1	0	0	0
1	1	0	1	0	0
1	1	0	0	1	0
1	1	0	0	0	1
1	0	1	1	0	0
1	0	1	0	1	0
1	0	1	0	0	1
1	0	0	1	1	0
1	0	0	1	0	1
1	0	0	0	1	1
0	1	1	1	0	0
0	1	1	0	1	0
0	1	1	0	0	1
0	1	0	1	1	0
0	1	0	1	0	1
0	1	0	0	1	1
0	0	1	1	1	0
0	0	1	1	0	1
0	0	1	0	1	1
0	0	0	1	1	1
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0

Figure 19. Possible Syndrome Words

In Figure 19 only twenty syndrome words for data bits are listed, because the possible words with a weight of 5 were eliminated so that every data bit would have only three bits protecting it. This simplifies the hardware implementation. If there are more than 20 data bits, states with a weight of 5 must be used. All states listed in Figure 19 are valid syndrome words, so that the problem becomes one of selecting the optimum set of syndrome words. To minimize circuit propagation delay the number of data bits checked by each encoding bit should be as close as possible to all the others.

The syndrome words can be mapped to any bit position, providing that identical code generations are done at storage and retrieval times. Syndrome word mapping may be arranged to solve system design problems. For example, in byte oriented systems the lower order syndrome bits are identical, so that the circuit design may be simplified by using these syndromes to determine which bit is in error, and the higher order syndromes to determine which byte is in error. Double bit detect/single bit correct code is implemented in hardware as a straight Hamming code would be.

DESIGN EXAMPLE

To illustrate code development, the design example uses single bit correct/double bit detect code on a 16 bit data word. In addition to the memory, the ECC system has five components: write check bit generator, read check bit generator, syndrome generator, syndrome decoder, and bit correction. Connected together as shown in Figure 20, these components comprise the basic system. Features can be added to the system to enhance its performance. Some systems include error logs as a feature. Because the address of the error and the errors are known, the address and the syndrome word are saved in a non-volatile memory. At maintenance time this error log is read and the indicated defective devices are replaced. Being a basic design, this example does not include an error log.

Write check bits are generated when data are written into the memory, while read check bits are generated when data are read from the memory. Off-the-shelf TTL is used to implement the design. Check bits are generated by performing parity on a set of data bits, so that this function is performed by 74S280 9-bit parity generators. One parity generator for each check bit is required. Because the read and write check bit generations are the same, the circuits are similar. One minor difference should be noted. In this example, the check bit will be formed from parity on eight data bits. The 74S280 parity generator has nine inputs; therefore, the write check bit generator will have the extra input grounded while the read generator has as an input the fetched check bit. Developed directly in the read check bit generator is the syndrome bit, which saves one level of gating. Figure 21 shows the identical results of generating the syndrome bit by exclusive-ORing the fetched check bit with the regenerated check bit and forming the syndrome bit in the read check bit generator.

Implementing the syndrome generator word in this way reduces the circuit propagation delay by approximately 10 nanoseconds. This implementation imposes a restriction on the code to be used — the check bit must be formed from no more than eight data bits.

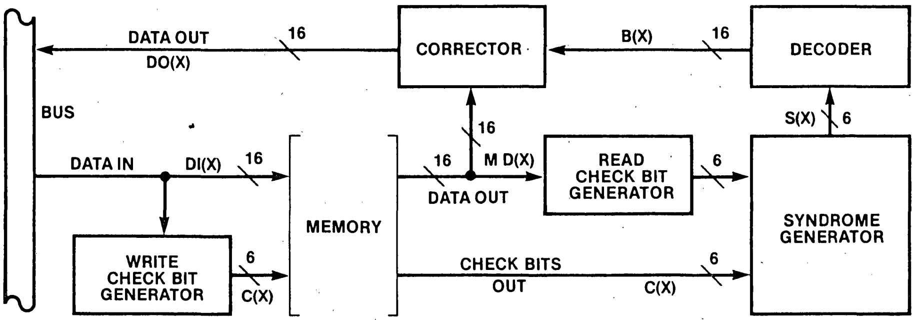


Figure 20. Block Diagram of ECC System.

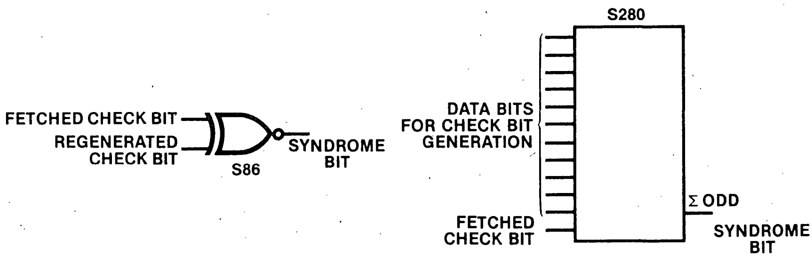


Figure 21. Syndrome Bit Generation.

Figure 19 listed the possible syndrome words for a 16 bit data word. These are relisted in Figure 22 with the syndrome words for the check bits and the zeros deleted.

			1			1		1	1			1		1	1		1	1	1	C1
		1			1		1		1		1		1		1	1		1	1	C2
	1			1			1	1		1			1	1		1	1		1	C3
1				1	1	1				1	1	1				1	1	1		C4
1	1	1	1							1	1	1	1	1	1					C5
1	1	1	1	1	1	1	1	1	1											C6

Figure 22. Possible Syndrome Words with Three Check Bits.

While there are twenty possibilities for syndrome words, only 16 are needed. Each row contains ten "1s" and each column contains three "1s." Four columns are eliminated but in a way that each row contains eight "1s." When the columns are matched to data bits, the "1s" in each row define inputs to the 74S280 parity generators for the given check bit. Eliminating the two columns from each end results in sixteen columns with each row having eight "1s." These remaining sixteen columns which match the data bits are rearranged in Figure 23 for convenience of printed circuit board layout and assigned to the data bits. The syndrome words for check bits are also shown for complete code development.

Data Bit

M16	M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	C1	C2	C3	C4	C5	C6	
X				X	X	X		X		X			X		X	X						C1
	X	X				X	X		X		X			X	X		X					C2
X	X		X		X		X					X	X	X				X				C3
X	X	X	X	X						X	X	X							X			C4
		X	X	X	X	X	X	X	X											X		C5
								X	X	X	X	X	X	X	X						X	C6

Figure 23.

With this information the check bit generators can be designed. Figure 24 depicts write check bit generators while Figure 25 depicts read check bit generators.

Double bit error detection is accomplished by generating parity on the syndrome bits. Except for the syndrome word of 000000 — no error — even parity will be the result of a double bit error. Hardware implementation is shown in Figure 26. OR-ing the syndrome detects the zero state, which has even parity and prevents flagging this state as a double bit error.

Decoding the syndrome word must be done to invert the one bit in error. Combinational logic will decode only those syndrome states which select the one of sixteen bits for correction. Figure 28 shows the logic of the decoder.

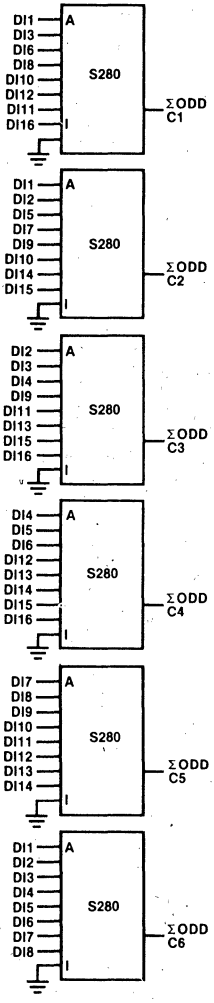


Figure 24. Write Check Bit Generators

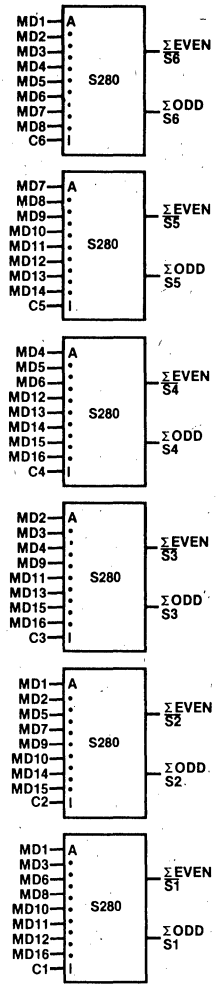


Figure 25. Read Check Bit Generators

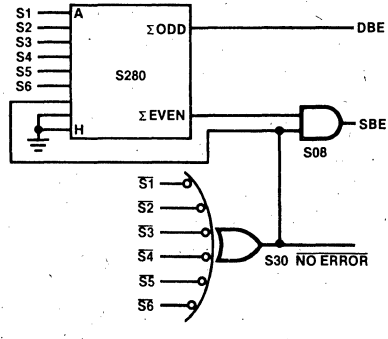


Figure 26. Double Error Decoder

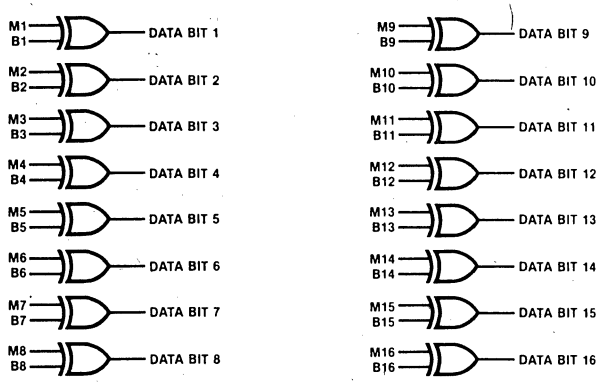


Figure 27.Correction Circuit.

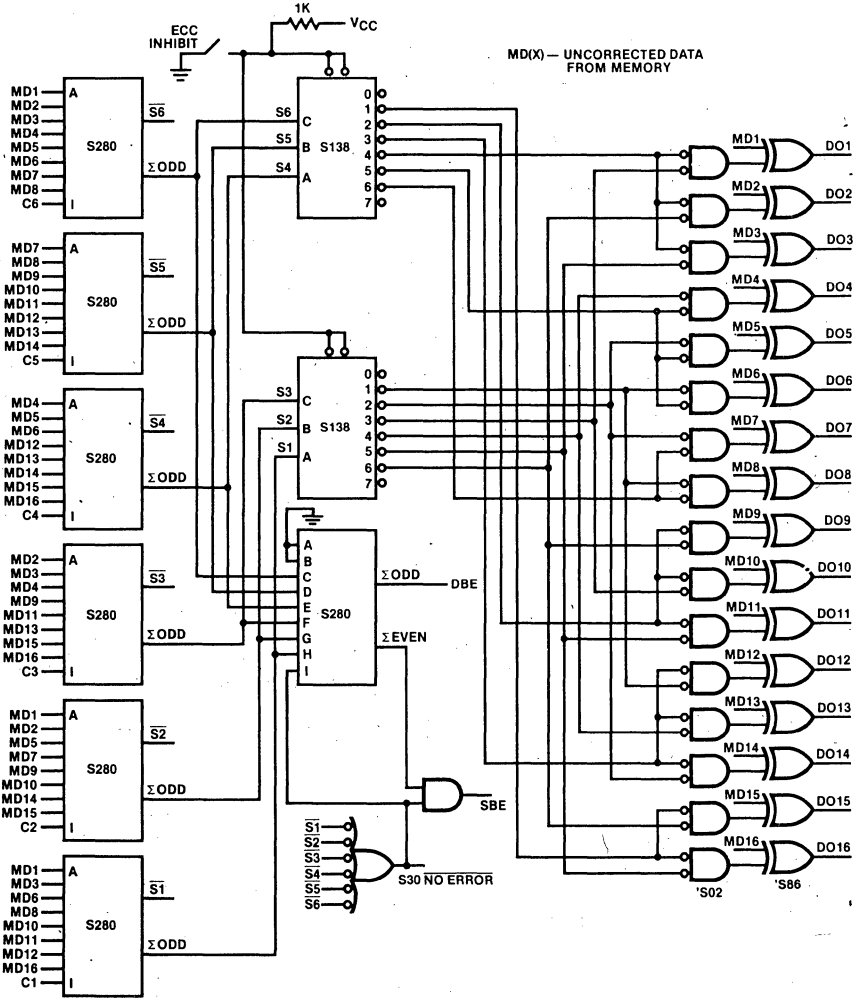


Figure 28. Complete Correction Circuit

Enabling the correction logic, the decoded B(x) signals become "high" to invert the output of the 74S86 exclusive-OR circuits. If the B(x) signals are "low" the output of the correction is the same level as the input. The correction circuit is shown in Figure 29.

Connecting the five circuits as shown in the block diagram of Figure 20 completes the error correction circuitry.

SUMMARY

An unprotected memory has a system MTBF which is approximately equal to the device MTBF divided by the number of devices. Redundancy codes are used to protect memories. While parity is a redundancy code, it only indicates that an error has occurred. A "modified" Hamming code can correct single bit errors and detect double bit errors, truly enhancing the system MTBF.

This report has laid the foundation of ECC basic concepts. Building on this foundation, the next report will address the mathematics for calculating the enhancement factor of ECC in a system environment.

REFERENCES

1. "2107A/2107B N-Channel Silicon Gate MOS 4K RAMs," Reliability Report RR-7, Intel Corporation, September, 1975.
2. "2115/2125 N-Channel Silicon Gate 1K MOS RAMs," Reliability Report RR-14, Intel Corporation, 1976.
3. "2104A N-Channel Silicon Gate 4K Dynamic RAM," Reliability Report RR-15, Intel Corporation, September, 1977.
4. "2116 N-Channel Silicon Gate 16K Dynamic RAM," Reliability Report RR-16, Intel Corporation, August, 1977.
5. R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Technical Journal,* Vol. 26 (April 1950), pp. 147-160.
6. Len Levine and Ware Meyers, "Semiconductor Memory Reliability with Error Detecting and Correcting Codes," *Computer,* October, 1976, pp. 43-50.
7. —, "Modern Algebra for Coding," *Electro-Technology,* April, 1965, pp. 59-66.
8. William W. Peterson and E. J. Weldon Jr., *Error Correcting Codes,* MIT Press, Cambridge, Mass., 1972.

intel

APPLICATION NOTE

AP-73

AUGUST, 1980

ECC #2 Memory System Reliability with ECC

Dennis Marston
Memory Products Division-Applications Lab

1. INTRODUCTION

This Application Note explains reliability analysis as applied to a typical memory system. (It follows Intel Application Note AP-46, which reviewed basic ECC, **Error Corrections Code,** concepts.) A number of examples demonstrate techniques to calculate reliability of a model memory system, with and without ECC — emphasizing system reliability as a function of the number of devices in a system and the individual device failure rates.

Since a system with ECC can correct a single bit failure and detect double bit errors within an accessed word, it has a decided advantage over a system without ECC. A soft error rate of two or three times device hard failure rate has significantly less effect on the Mean Time Between Failures (MTBF) for a system with error correction. This is quantified as the Enhancement Factor, EF — the ratio of MTBF for two identical systems, one with and one without ECC. The Enhancement Factor can be predicted by the application of statistical analysis.

The general model presented in this Application Note numerically predicts the chance of memory system failures during a specified length of time. It also provides insights into the relationship of device failure mechanisms and soft errors to memory system reliability. Intel® 2117 Dynamic RAM is used in the example memory system. The reliability data for distribution of hard failures was obtained from the 2117 Reliability Report (Intel RR-20).

2. MEMORY CONFIGURATION

2.1 Device

System reliability begins with the smallest physical unit, the memory device. Each device can be considered a system itself, with the smallest functional unit being a single storage cell. Device internal structures have inherent failure mechanisms affecting individual memory cells.

The structure of a typical RAM device consists of two-dimensional coordinate-addressed arrays of memory cells arranged in rows and columns, such as the Intel® 2117 Dynamic RAM shown in Figure 2. This device contains 16384 cells arranged in a 128 row by 128 column matrix; each cell is selected by an encoded 7-bit row and 7-bit column address.

2.2 System

An array of memory devices on one or more circuit boards forms a typical memory system. A system is defined by n bits per word, x words per page and p pages per system. Note that a "page" is defined as the number of memory words formed by a minimum set of memory components.

For example, 16K by 1 RAMs would have a minimum page size of 16384 words.

Figure 1 represents such a system, with the horizontal axis corresponding to parallel, address-accessed data bits and the vertical axis corresponding to the series stacking of words and pages. This memory structure is used for the model system.

3. ERROR CLASSIFICATION

The 2117 failure mechanisms illustrated in Figure 3 are fairly representative for today's RAM devices. These can be categorized as **hard failures** and **soft errors.**

3.1 Hard Failures

Hard failures are permanent physical defects, such as shorts, open leads, micro-cracks or other intrinsic flaws. They are classified as single cell failures, row failures, column failures, combined row-column failures, half-chip failures and full-chip failures.

The failure type distribution within a device is a function of the device design. Typical ratios are 50% single cell failures, 40% row or column failures, 10% combined failures and less than 0.1% half-chip or full-chip failures. (Refer to Figure 4.) The accumulative independent events are expressed as a single numeric value for the combined failure rate of the device (EQ:1a). The standard mathematical symbol for device failure rate is the Greek letter Lambda, λ; i.e., λ = 0.027%/1000 hrs.

EQ:1a $$\lambda_{hrd} = \lambda_{single} + \lambda_{row} + \lambda_{column} + \lambda_{row/col} + \lambda_{halfchip} + \lambda_{fullchip}$$

3.2 Soft Errors

In contrast to hard failures, soft errors are characterized as being random in nature, non-recurring, non-destructive single cell errors.

Traditional soft errors are caused by noisy system environments, poor system design, or rare combinations of noise, data patterns, and temperature effects which push the RAM beyond its normal specified range of operation. This type of soft error has not been included in the analysis to follow because it is associated with system level problems and the rate of failure is difficult to quantify; in any case it is assumed to be quite small.

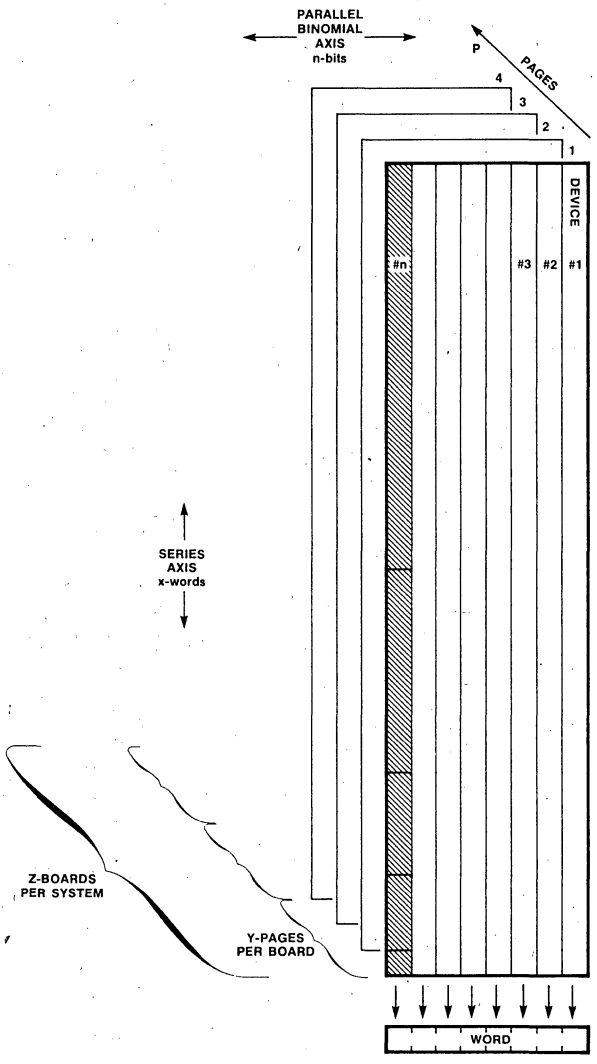


Figure 1. Memory Configuration

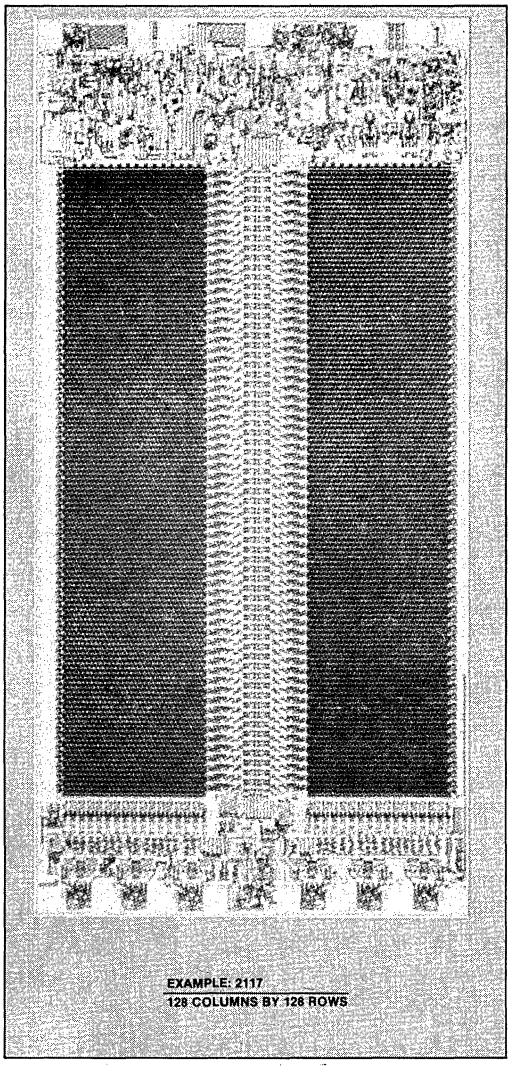


Figure 2. Random Access Memory Device

Other soft errors are caused by ionizing radiation of alpha particles changing memory cell charge in semiconductor substrates with high impedance nodes. The data bit error is realized during a memory read to the failing cell. These errors are purged by rewriting (restoring) the correct data bit information to the cell. The failure rate for this type of soft error is stated separately from hard failures because of its unique properties.

The total device failure rate becomes:

EQ:1b $\lambda_{dev} = \lambda_{hrd} + \lambda_{sft}$

The pie graph in Figure 5 depicts the combined distribution of both hard and soft errors.

4. RELIABILITY

Reliability, as used in this application note, is defined as "the probability that a component will operate within specified limits, for a given period of time"[1]. The definition includes the term "probability", a quantitative measure for chance or likelihood of occurrence, of a particular form of event — in this case, operation without failure within specified limits. In addition to the probabilistic aspect, the reliability definition also involves length of operational time.

Since reliability is concerned with events which occur in the time domain, they are classified as incidental failures, which do not cluster around any mean life period, but occur at random time intervals. The exact time of failure cannot be predicted; however, the probability of occurrence or non-occurrence of a statistical mean in a given operating frame of time can be analyzed by the theories of probability. Since exact formulae exist for predicting the frequency of occurrence of events following various statistical distributions, the chance or probability of specified events can be derived.

4.1 Component Reliability

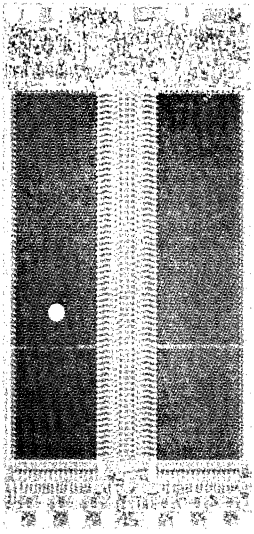
Memory systems are operated where failures occur randomly due only to chance causes. The fundamental principles of reliability engineering predict the failure rate of a group of devices which will follow the so-called bathtub curve in Figure 6. The curve is divided into three regions: Infant Mortality, Random Failures, and Wearout Failures. All classes of failure mechanisms can be assigned to these regions.

Infant Mortality, as the name implies, represents the early life failures of a device. These failures are usually associated with one or more manufacturing defects. Memory device failures occurring as the result of Infant Mortality have been eliminated by corrective actions relating design, inspection, and test methods.

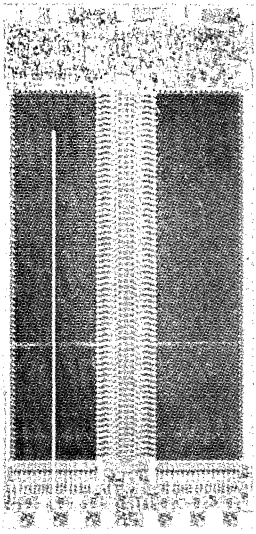
Wearout failures occur at the end of the device's useful life and are characterized by a rising failure rate with time as the device's "wearout" both physically and electrically. This does not occur for hundreds of years for integrated circuits.

The Random Failure portion of the curve represents the useful period of device life. As stated, memory devices are operated in systems during this period when failures occur randomly. The number of failures occurring during any time interval within the "Random" period is related only to the total number of memory components

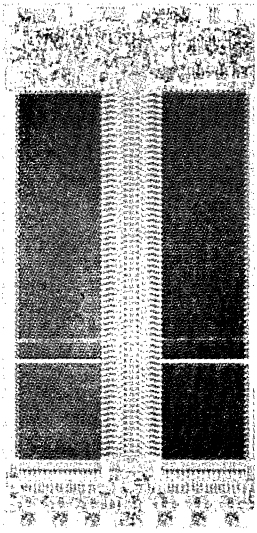
[1] *Reliability Mathematics* — Amstadter



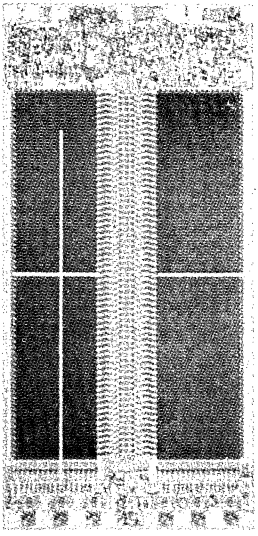
SINGLE CELL
(1 CELL)



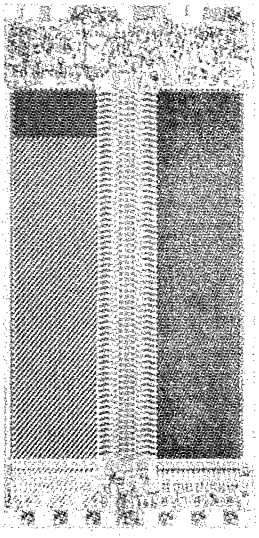
ROW
(128 CELLS)



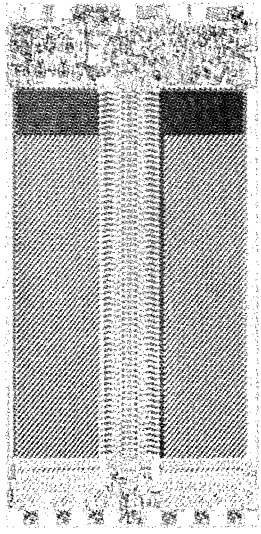
COLUMN
(128 CELLS)



ROW-COLUMN
(256 CELLS)



HALF-DEVICE
(8192 CELLS)



FULL-DEVICE
(16,384 CELLS)

Figure 3. Failure Geometry — 2117 Example

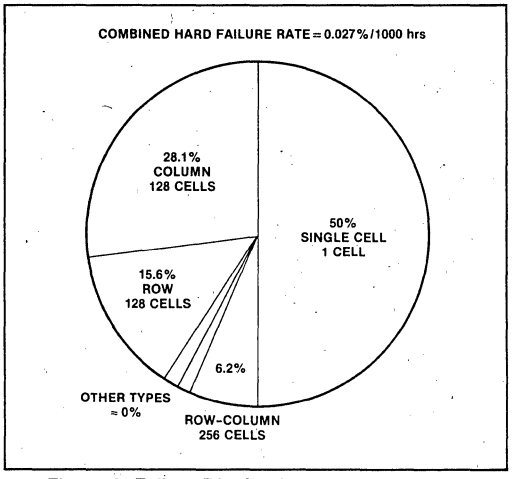


Figure 4. Failure Distribution — 2117 Example

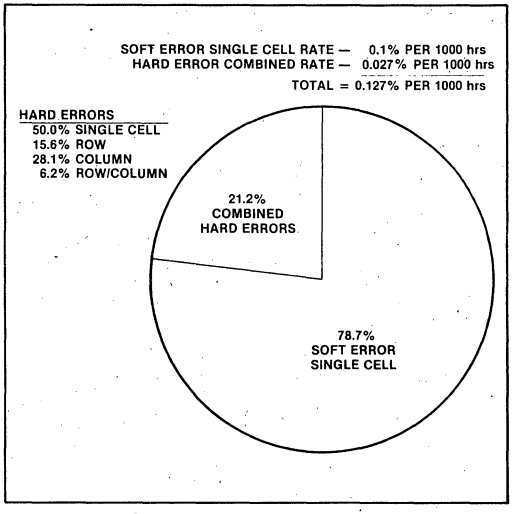


Figure 5. Combined Distribution of Failure Type

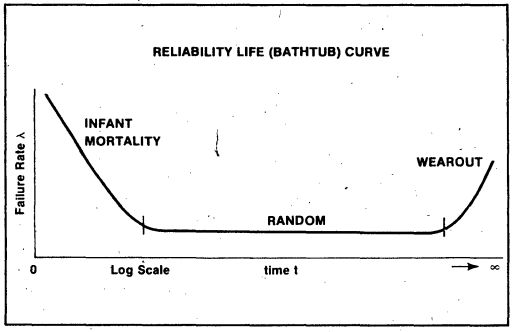


Figure 6. Reliability Life Curve

operating. If sufficient numbers are operated, and the measured interval is long enough, failure rate approaches some relative constant value. For any given component type, the failure rate value will depend on operating and external environmental conditions (voltage, temperature, timing, etc.) and will be characteristic of this set of conditions. When the conditions change, the failure rate will correspondingly change.

For example, if 500 devices are tested for 1,500 hours and two failures were observed during the test interval, then the failure rate is two failures per 750,000 device-hours or one failure per 375,000 device-hours. For commonality, device failure rates are expressed as a percentage value per 1000 device-hours. The above example then becomes .00266 failures per 1000 device-hours or λ_{dev} = 0.27% per 1000 hours. This is an overly simplified statement on determining the device failure rate. Many tests, designed to stress the devices over operating conditions and margins, are used in the final analysis for the specification of device failure rates.

4.1.1 RELIABILITY FUNCTION R(t)

The Reliability Function, R(t), follows an inverse, natural logarithmic curve, which expresses the rate of change for a memory component from an operational state to a failure or error condition. The curve is a familiar one to the physical scientists because of its relationship to growth and decay.

The general function for reliability is given in EQ:2 where the exponent ($\lambda \cdot t$) represents the device failure "lambda" times the independent time variable "t". The graph in Figure 7 shows the shape of the R-function curve.

EQ:2 $R(t) = e^{-\lambda t}$

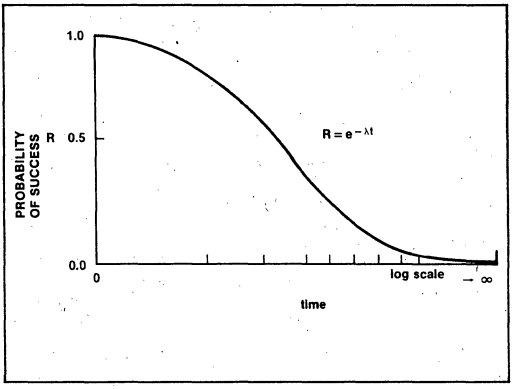


Figure 7. R(t) - Reliability Function

For any constant failure rate the value of reliability depends only on time. The limits of the reliability function R(t) are:

$$R(0) = 1.0 \text{ and } R(\infty) = 0.0$$

The distribution is a one-parameter type; in that once the failure rate is established, the reliability function is completely defined. For high or low failure rates the general shape of the curve remains the same, but is adjusted along the time axis.

4.2 System Reliability

Just as there is a functional relationship between the components and the system, there is a functional relationship between component reliability and system reliability. If a failure in any one of the components of a system causes the entire system to fail, the system is a "Series System" (Figure 8).

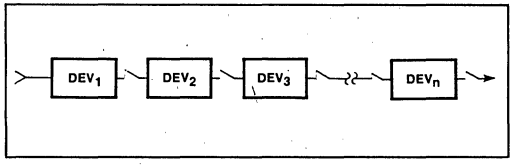


Figure 8. System of Series Components

If all the component devices must fail before the system fails, the system is a "Parallel System" (Figure 9).

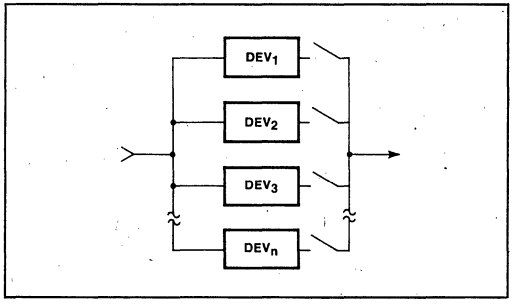


Figure 9. Parallel System

If a system has 'n' components which operate in parallel, but 'j' out of the 'n' components need to be functional for the system to operate, then this system is referred to as a "Parallel Binomial System" (Figure 10).

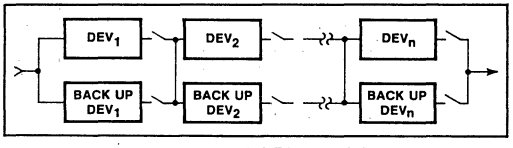


Figure 10. Parallel Binomial System

4.2.1 EQUATION FOR A SERIES SYSTEM

The Reliability Function for a series system is the product of the reliabilities of the individual components. If "n" components with corresponding failure rate of $\lambda_1, \lambda_2, \lambda_3, , , , \lambda\eta$ operate in series to form a system then the equation for system reliability is:

EQ:3 $R(t)_{sys} = R(t)_1 \cdot R(t)_2 \cdot R(t)_3 \cdot , , , R(t)_\eta$
where $R(t)_i = e^{-\lambda_i \cdot t}$

If each of the n components has the same device failure rate lambda, then the system reliability equation reduces to:

EQ:4 $R(t)_{sys} = R(t)^\eta = e^{-\eta\lambda t}$

4.2.2 EQUATION FOR A PARALLEL BINOMIAL SYSTEM

One of the fundamental concepts of reliability engineering is the Binomial Theorem. The theorem is used for computing the reliability of complex redundant systems, where "j" out of "n" units are required to operate for system success. The binomial distribution expresses the probabilities of two states of an event, "a" and "b", where the event is permutated "n" ways. The general form of the binomial distribution is $(a + b)^\eta$, and is expanded to:

EQ:5 $$a^\eta + \eta a^{\eta-1} \cdot b + \frac{\eta(\eta-1)a^{\eta-2} \cdot b^2}{2!} + \frac{\eta(\eta-1)(\eta-2)a^{\eta-3} \cdot b^3}{3!} + \cdots + b^\eta$$

It is applicable to a memory system operating in parallel; i.e., when there are only two possible states or results of an event — when a component of the system either conforms to requirements or is discrepant.

If we assign to one state the function of reliability — R(t), then the other state is Q(t), the function of non-reliability, which is the probability of being inoperative.

Recall that R(t) is a unity function, which ranges from 1.0 to 0.0, as a function of time. Since the sum of R(t) and Q(t) make up the whole "event", then EQ:6 defines Q(t). This relationship is also illustrated in Figure 11.

EQ:6 $R(t) + Q(t) = 1$, then $Q(t) = 1 - R(t)$

By substituting R(t) and Q(t) respectively for a and b, where R(t) is the probability of a device being good, Q(t) is the probability of the same device being defective, and "n" the number of units in parallel, then:

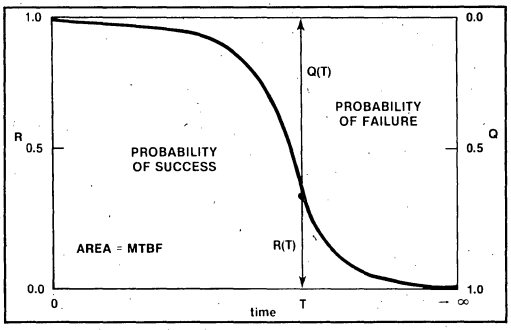


Figure 11. Q(t) = 1 - e $^{-\lambda t}$

EQ:7 $[\ R + Q\ ]^{\eta} = 1$

Note, for simplicity, all references to (t) for the reliability and non-reliability functions will not be indicated, but implied.

It follows that the expansion of $[R + Q]^{\eta}$ must also equal unity example

EQ:8 $$R^{\eta} + \eta R^{\eta-1}\cdot Q + \frac{\eta(\eta-1)R^{\eta-2}\cdot Q^2}{2!} + \frac{\eta(\eta-1)(\eta-2)R^{\eta-3}\cdot Q^3}{3!} + \cdots + Q^{\eta} = 1$$

We can next examine the meaning of each term in the series on the left side of EQ:8. Suppose that there are "n" identical components of a system, of which the probability of a component being operative is R, and that the probability of its being inoperative is Q or (1 − R). If there is only one component (n = 1), then the probability of its being not defective is simply R.

If there are two components (n = 2), then the probability of both being operative is $R \times R = R^2$; and if there were three components, then the probability of all three being good is R^3. Consequently, if there are "n" components, the chance of all "n" units being operative is R^{η} and the first term in the series R^{η} is the probability of all components being operational.

Next, suppose there are two components X and Y, one is operative and one has failed. There are two ways that this can occur: X is operational and Y fails, with the probability $R_x \cdot Q_y$; or X fails and Y is operational, with the probability $Q_x \cdot R_y$. Since these are mutually exclusive and constitute all possible combinations of one operative component and one failure, the total probability is $(R_xQ_y) + (Q_xR_y)$, or 2RQ.

If there are three components X, Y, and Z, of which two are operative and one fails, then three possible combinations exist: X and Y are operational and Z fails, X and Z operational and Y fails, and Y and Z operational and X fails. The probability of each combination is $(R_xR_yQ_z) + (R_xQ_yR_z) + (Q_xR_yR_z)$.

Again, since each combination is mutually exclusive and together they constitute all possible combinations, the probability of two operational devices and one failure is $3R^2 \cdot Q$. Similarly, if there are n component-devices, the probability of all but one being operative is $nR^{\eta-1} \cdot Q$. Thus, the second term of the binomial expansion series is the probability of exactly one device failure, and all other devices being good.

By extending these derivations to cover each succeeding term, we find that the third term is the probability of exactly two failed components, the fourth term is the probability of exactly three failures and so on. There are n + 1 terms in the expansion, and the last term Q is the probability all components are inoperative.

The reliability of a group of redundant items depends not only on the reliability of each individual item and on the number of items in redundant configuration, but also on how many are required to operate to achieve system success. If all are required, then the first term of the binomial series represents system success. In this case there is really no redundancy. However, if all but one are required (one failure permitted), then success is achieved if no failures occur or exactly one failure occurs within word accessed from a page of memory. The system reliability is then the sum of the first two terms of the series.

If two failures are permitted, then the sum of the first three terms represents the probability of system success. In general, if r failures are permitted, system success is the sum of the first r + 1 terms.

The general equation then for a binomial system, permitting one error, which is representative of a memory system with single bit error correction — ECC per accessed word is expressed as:

EQ:9 $$R_T(t) = \underbrace{R^{\eta}}_{1st} + \underbrace{\eta \cdot R^{\eta-1} \cdot Q}_{2nd} - \text{binomial terms}$$

Note that the remaining terms of the binomial expansion represent all combinations of failures that are greater than one failure, up to and including all components failing. RT(t) is still a unity function of reliability and has a converse QT(t), where QT(t) = 1 − RT(t). Thus, QT represents the 3rd through n-th terms of the binomial.

5. RELIABILITY ANALYSIS USING PAGE/SYSTEM APPROACH

The analysis of the model system in Figure 1 begins with EQ:2 at the smallest non-redundant failure level; by using standard rules for series and parallel reliability, the combination of these device exponential expressions will yield the system reliability equation. The method of approach will be to calculate the reliability of a page of memory and treat subsequent pages as a series system where:

EQ:10 $R(t)_{system} = [R(t)_{page}]^P$

For clarity, the reliability of power supplies, fans, backplane connections, TTL support logic, etc. will not be included. These items can be merged in the final analysis by the reader as additional series system equations for each type.

5.1 Memory System Without ECC

The analysis of reliability of a memory system "without" any form of ECC is simply the first term of the binomial equation EQ:9. Since this term represents reliability of all components in a page of memory without redundancy, it is equivalent to a "series system" equation (EQ:4). Therefore, the equation for a page of memory without ECC is:

EQ:11 $R(t)_{PAGE_{necc}} = R(t)^{\eta}_{DEV_{necc}} = e^{-\lambda \cdot n \cdot t}$

where "n" is the number of components in the page and λ_{dev} is the device combined failure rate.

The reliability for the memory system of "p" pages is:

EQ:12

$$R(t)_{SYS_{necc}} = [R(t)_{PAGE_{necc}}]^p = [R(t)_{DEV}]^{p \cdot \eta}$$

5.2 Memory System With ECC

The analysis of reliability of a memory system "with ECC" — (single bit error correction) is more complex. The fundamental difference between the two memory systems is that in a non-corrected system, any error — no matter the type, single cell failure, row failure, soft error, etc. — is considered a system failure. In a memory system with ECC, a system level failure only occurs when more than one bit has failed in an accessed word.

Thus in the analysis of a System with ECC, we must deal with the probabilities of each failure type occurring in random combinations which align within a word of memory to cause multiple bit failures as shown in Figure 12.

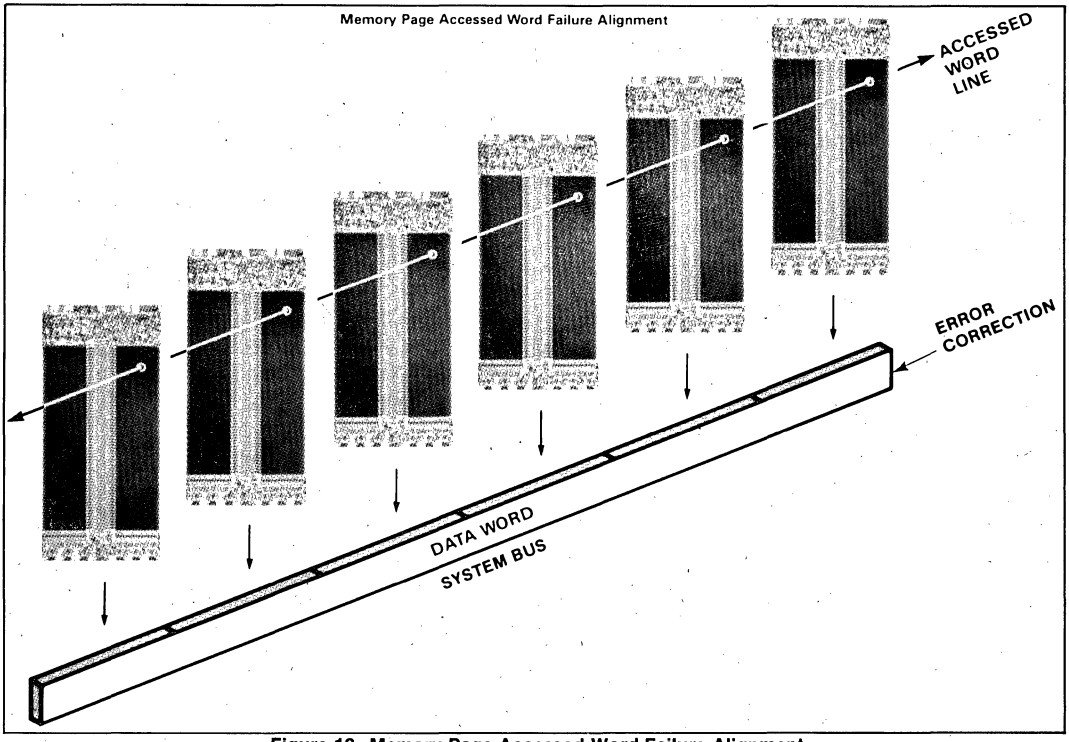


Figure 12. Memory Page Accessed Word Failure Alignment

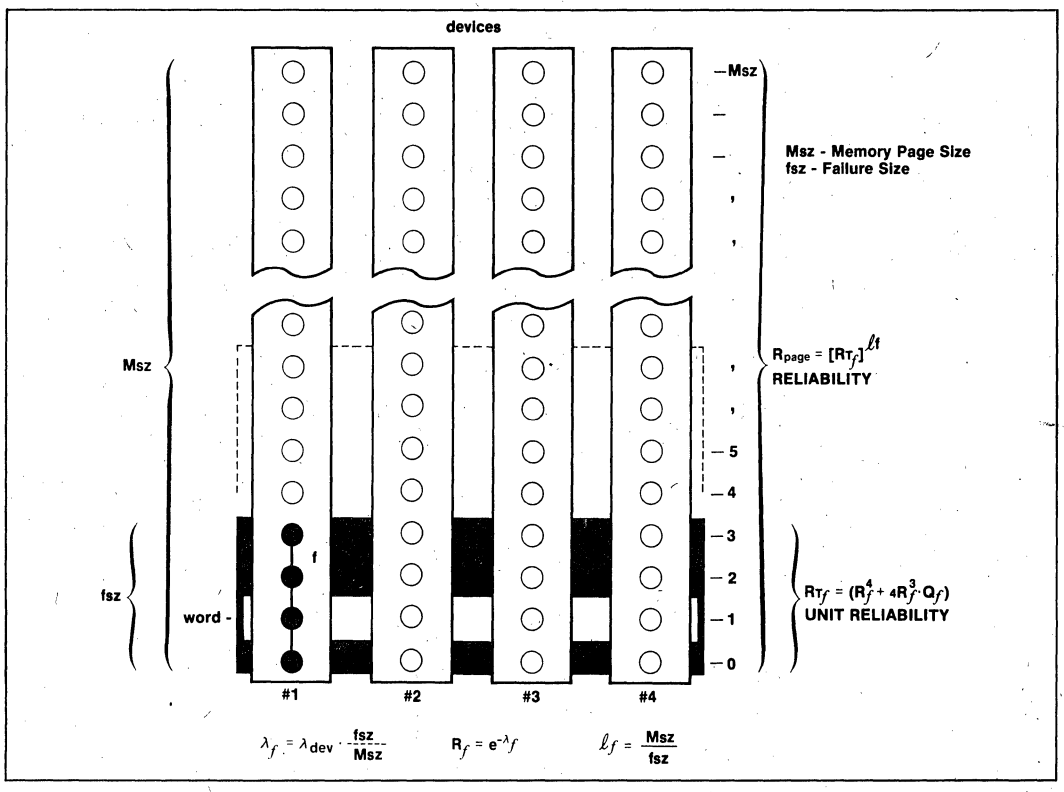


Figure 13. Single Failure Type Illustration

For example, consider a single cell hard failure in one device in a system using 16K RAMs. The chance of a similar failure in the same cell of a different device is 1/16384 times the device failure rate for single cells. For n devices in the data word the total chance is n/16384 for a single cell match.

The application of the binomial distribution (EQ:9) requires further differentiation in the analysis of the example memory system. EQ:9 is restricted to one failure mode, in that it typically assumes a failure renders the whole device inoperative. This is not the case with memory components where each device in itself can be thought of as a system of memory cells, with the smallest unit being the single cell.

Multiple devices have multiple failure modes, but usually when a failure occurs only a portion of the memory component is inoperative. Therefore, the application of EQ:9 must represent the **unit of failure** and be mutually inclusive with all other components along the accessed word (parallel axis) of the memory page.

The example in Figure 13 shows a four device memory array where each component has a single failure mechanism of type f, which affects fsz number cells during a failure. The unit failure rate λf is the ratio of {fsz/Msz} times the device failure rate λ_{dev}. Only that portion of the failure area, the shaded area in Figure 13, is mutually-inclusive with the failure when it occurs. Any additional failures outside the shaded area are mutually-exclusive, causing no double-bit failures in conjunction with "*f*."

The Reliability Function, RT, therefore, represents only a portion of the memory page as indicated by the shaded area fsz in Figure 13. If "*f*" were the only failure type, then the reliability for the full page is simply a series equation with RT raised to the exponent ℓ, the ratio Msz/fsz.

Derived from the binomial equation EQ:9, the expression for reliability for a single page of memory with one bit redundancy — (ECC) —, and only one failure type "x" is given as:

EQ:13

$$R(t)_{PAGEecc} = \left[R(t)_x^{\eta} + \eta \cdot R(t)_x^{\eta-1} \cdot Q(t)_x\right]^{\ell_x}$$

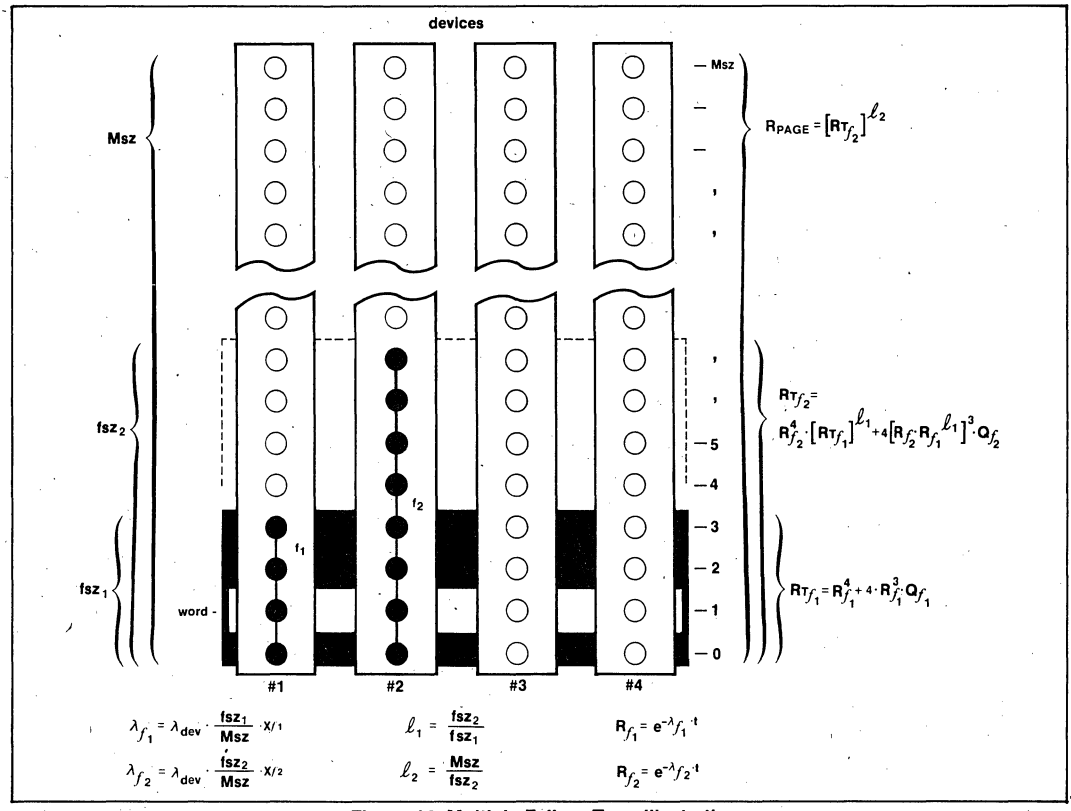


Figure 14. Multiple Failure Type Illustration

Now that the binomial equation technique has been applied to a single failure type, let's expand the process to cover more than one failure type. By the process of combining or permutating these failure types, the Reliability Function can be calculated. Figure 14 shows a four component system with the probability that two failure types f_1 or f_2 can occur in each component. Both failure types affect fsz_1 and fsz_2 number of cells during a failure, respectively. The calculation begins with evaluating the probability of f_1 occuring (EQ:14a) and merging by a second calculation the probability of failure type f_2. (EQ:14b).

EQ:14a $$R_{T_1} = R_{f_1}^{\eta} + \eta R_{f_1}^{\eta-1} \cdot Q_{f_1}$$

EQ:14b $$R_{T_2} = R_{f_2}^{\eta} \cdot [R_{T_1}^{\ell_2}] + \eta [R_{f_2} \cdot R_{f_1}^{\ell_2}]^{\eta-1} \cdot Q_{f_2}$$

NOTE: with λ_{dev} representing more than one failure type, f_1 and f_2, λ_{dev} must be proportioned to the "failure-type-distribution" in determining the unit failure rates λf_1 and λf_2. The term Xf_1 and Xf_2 are introduced to quantify the failure type distribution as a percentage. (Ref: EQ:1 and Figure 5).

EQ:14c is the unit failure rate equation for f_1 and f_2 in this case.

EQ:14c

$$\lambda f_1 = \lambda_{dev} \cdot X f_1 \cdot \frac{fsz_{f_1}}{Msz} \qquad \lambda f_2 = \lambda_{dev} \cdot X f_2 \cdot \frac{fsz_{f_2}}{Msz}$$

The total reliability for the page in Figure 13b is given by equation 14d.

EQ:14d $$R(t)_{page} = [R_{T_2}]^{\frac{Msz}{fsz_2}}$$

By expanding on this process the equation for a system of memory components with these failure types: f_1, f_2, f_3 is given in EQ:15.

EQ:15

$$R_{T_3} = R_{f_3}^{\eta} \cdot [R_{T_2}]^{\ell_3} + \eta [R_{f_3} (R_{f_2} (R_{f_1})^{\ell_2})^{\ell_3}]^{\eta-1} \cdot Q_{f_3}$$

We can now formulate a general set of equations for multiple (f_1) failure types in an error corrected system.

5.2.1 EQUATIONS FOR THE MODEL

The full model under analysis in this report has six failure types, as described in the section on Error Classification. The reliability calculations for a page of memory must permutate all combinations of these six failure types. It is accomplished by the set of equations in EQ:16.

EQ:16

$$\left[\begin{array}{l} N \\ i \leftarrow \\ 1 \end{array} \left| \begin{array}{l} \ell_i = \dfrac{fsz_i}{fsz_{i-1}} \\ \lambda f_i = \lambda_{dev} \cdot X_i \cdot \dfrac{fsz_i}{Msz} \end{array} \right. \right]$$

$$R(t)_{PAGE_{ecc}} = \left\{ \begin{array}{l} N \\ i \leftarrow \\ 1 \end{array} \left| \begin{array}{l} R_i = e^{-\lambda f_i \cdot t},\ Q_i = 1 - R_i \\ RS_i = R_i \cdot (RS_{i-1})^{\ell_i} \\ RT_i = R_i^{\eta} \cdot (RT_{i-1})^{\ell_i} + \eta (RS_i)^{\eta-1} \cdot Q_i \end{array} \right. \right\}^{\frac{Msz}{fsz_N}}$$

$$R(t)_{SYSTEM\,ecc} = [R(t)_{PAGE\,ecc}]^{Pages}$$

restrictions: $RS_0 = RT_0 = fsz_0 = 1$.

The process begins at the word level with soft errors and gradually increases the area of evaluation to single cell hard failures, then row or column failures, combined row/column failures, half-chip failures, and finally full-chip failures.

Illustrated in Figures 15 and 16 are the six iterative steps to merge all combinations of failure types — f_1, f_2, f_3, f_4, f_5, f_6.

The first step calculates the chance of a single word of the memory page not having more than one soft error.

The second step calculates the probability of not having more than one single-cell hard failure and merges step #1, for a combined result that no more than one failure caused by either soft error or single-cell failure has occurred within the single word analyzed.

The third step calculates for row failures and merges with step #2 all combinations of the three failure types. Using the 2117 example memory system from Figure 6 to illustrate this point — a row or column failure affects 128 memory words — the combined result from step #2, which analyzed a single word, is raised by the exponent 128 as a series equation. The combined result for step #3 is the probability of not having a system failure due to any of the failure types f_1, f_2, f_3, in any given word for a 128-word block.

This process continues up to step six, which is the calculation for all six failure types occuring in all combinations that would cause a system failure within the page of memory. The analysis of each step therefore raises the results of each previous step by the exponent ℓ_i.

5.2.2 THE ENHANCEMENT FACTOR

5.2.2.1 Mean Time Between Failures

The Mean Time Between Failures (MTBF) for a memory system, with or without ECC, is given in EQ:17. MTBF is calculated by integrating the system reliability function, $R(t)_{sys}$, from t = 0 to infinity.

EQ:17 $$MTBF_{sys} = \int_0^{\infty} R(t)_{sys} \cdot dt$$

On the average a system will fail once every $MTBF_{sys}$ hours. The relationship between MTBF and the R function is shown in Figure 17.

The bottom line conclusions on the effect that error-correction has on a given memory system is calculated by comparing the resultant MTBF $_{sys-ecc}$ projection with the MTBF $_{sys-necc}$ of a similar system without ECC. The improvement of a memory system with error correction logic over a comparable system without is expressed by EQ:18 as the enhancement factor EF.

EQ:18 $$EF = \frac{MTBF_{sys\text{-}ecc}}{MTBF_{sys\text{-}necc}}$$

5.2.2.2 Mean Time To Failures

The Mean Time To Failure (MTTF) is similar in concept to MTBF, but differs in that it represents the effects of maintenance on an error corrected memory system. When a maintenance policy is adopted which allows for the replacement of failed components before the system fails, system failure is postponed (depending on how often the system is inspected and maintained). With this policy a memory system fails less frequently than it does without maintenance; it is assured that every new operating period after inspection starts with full redundancy restored. The maintained system Mean Time To Failure thus becomes greater than $MTBF_{sys}$.

If preventive maintenance is performed at an arbitrary time T, then EQ:19 expresses mean time to failure.

EQ:19 $$MTTF = \frac{\int_0^{T} R(t)_{sys\text{-}ecc} \cdot dt}{1 - R(T)_{sys\text{-}ecc}}$$

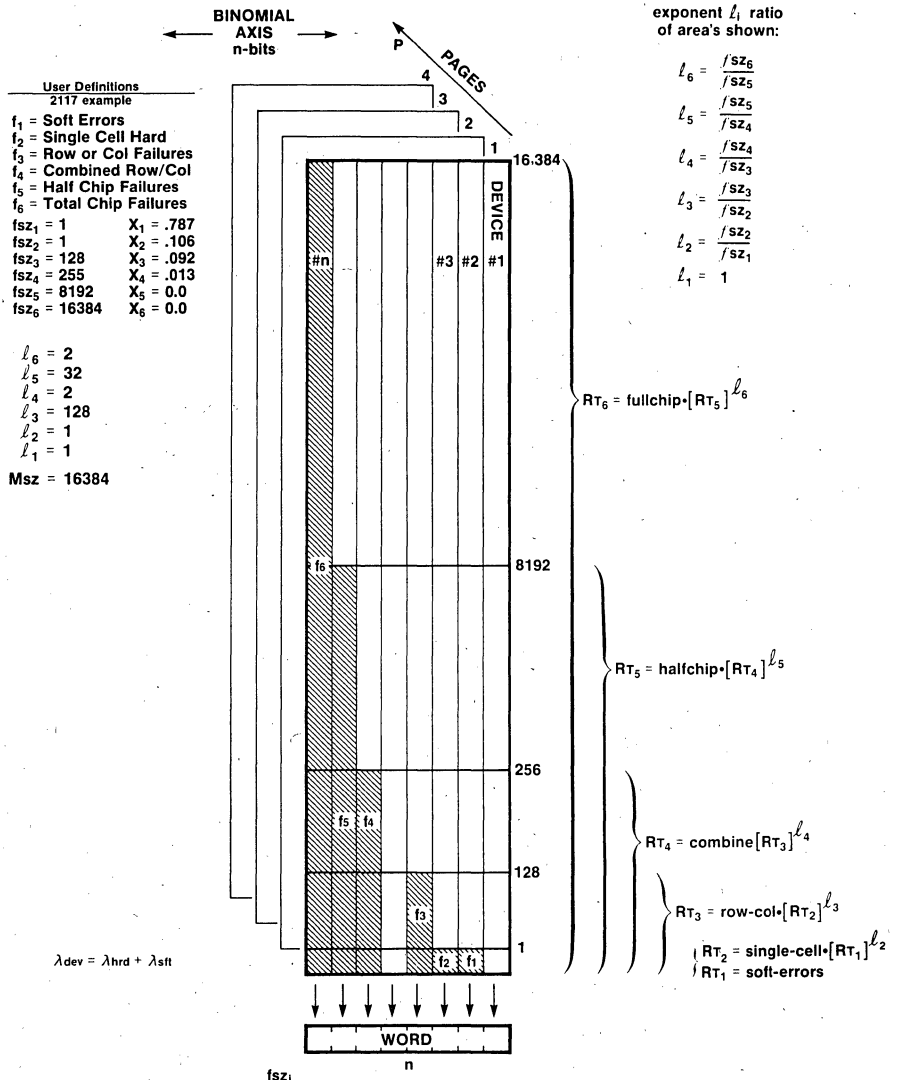

BINOMIAL AXIS n-bits
P
PAGES
4
3
2
1
DEVICE
16,384
8192
256
128
1
#n
#3 #2 #1
f6
f5 f4
f3
f2 f1
WORD
n
User Definitions
2117 example
f1 = Soft Errors
f2 = Single Cell Hard
f3 = Row or Col Failures
f4 = Combined Row/Col
f5 = Half Chip Failures
f6 = Total Chip Failures
fsz1 = 1 X1 = .787
fsz2 = 1 X2 = .106
fsz3 = 128 X3 = .092
fsz4 = 255 X4 = .013
fsz5 = 8192 X5 = 0.0
fsz6 = 16384 X6 = 0.0
ℓ6 = 2
ℓ5 = 32
ℓ4 = 2
ℓ3 = 128
ℓ2 = 1
ℓ1 = 1
Msz = 16384
exponent ℓi ratio of area's shown:
ℓ6 = fsz6 / fsz5
ℓ5 = fsz5 / fsz4
ℓ4 = fsz4 / fsz3
ℓ3 = fsz3 / fsz2
ℓ2 = fsz2 / fsz1
ℓ1 = 1
RT6 = fullchip•[RT5]^ℓ6
RT5 = halfchip•[RT4]^ℓ5
RT4 = combine[RT3]^ℓ4
RT3 = row-col•[RT2]^ℓ3
RT2 = single-cell•[RT1]^ℓ2
RT1 = soft-errors
λdev = λhrd + λsft
λfi = λdev • Xi • fszi / Msz
RPAGEecc = { 6 | Ri = e^-λi•t , Qi = 1 - Ri
i | RSi = Ri • (RSi-1)^ℓi
1 | RTi = Ri^η • (RTi-1)^ℓi + η(RSi)^η-1 • Q1 } ⇒RT6
1st term
2nd term
RSYS = [RT6]^P

Figure 15.

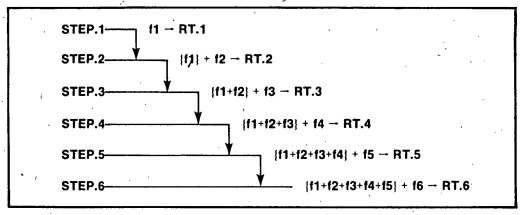


Figure 16.

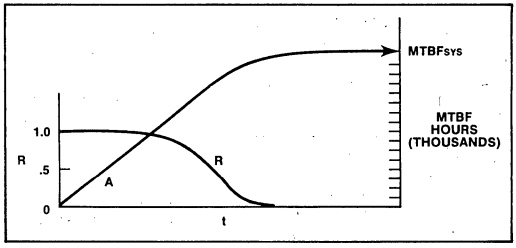


Figure 17.

Figures 18 and 19 show the relationship of MTTF to the R function and MTTF to MTBF respectively.

The enhancement of a memory system with maintenance over a comparable system without ECC is expressed in EQ:20.

EQ:20 $$EF_{mnt} = \frac{MTTF}{MTBF_{sys\text{-}ecc}}$$

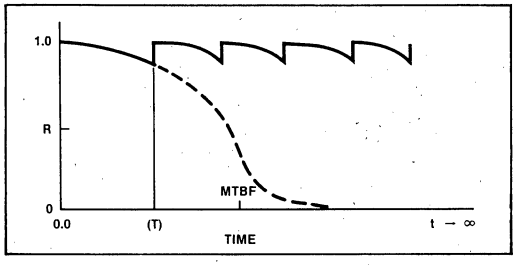


Figure 18.

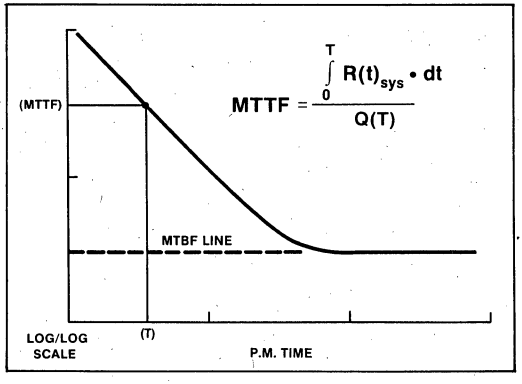


Figure 19.

5.2.3 SOFT ERROR SCRUBBING

In the previous sections on MTBF and MTTF, soft errors and hard errors were treated the same. They both accumulated to cause system failure or were removed at scheduled preventive maintenance (PM) intervals.

However, soft errors can have their own special maintenance function. Recall that soft errors can be purged from a system with ECC by rewriting (restoring) the correct data bit information to the failing memory cell. (Provided that no other bit within the word containing the soft error has failed.) Thus it is possible for the system to maintain itself by software, etc. This special maintenance function of scrubbing soft errors at predetermined intervals is incorporated into the system reliability equations by merely resetting the time parameter t for the soft error portion of the equations.

Figure 20 shows the relationship of soft error scrubbing on MTBF and the system R functions.

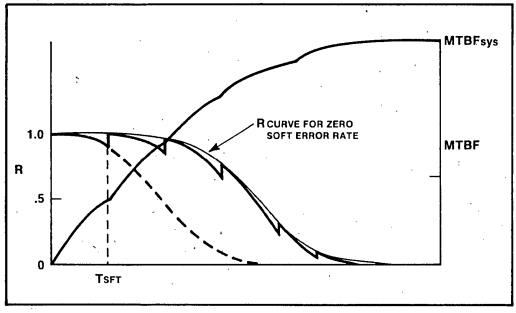


Figure 20.

5.2.4 APPLYING THE MODEL EQUATIONS

The basic set of equations for a model are derived from EQ:16. The application of these equations is best suited for implementation on a computer. An example computer program is available on request.

Figure 21 illustrates a simplified block diagram of the model.

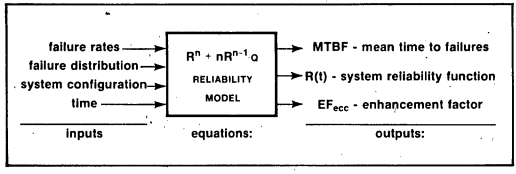


Figure 21.

The required user inputs are for component parameters — total memory size, number of rows and columns, hard failure rate, soft error rate, and

failure mode distribution; for system parameters — memory word size, ECC check bits, number of pages, interval of time, and soft error scrub time.

Output is a set of discrete values of the reliability function representing the complete memory system as a function of time.

The integral functions for MTTF and MTBF are evaluated by the trapezoidal rule of integration.

EQ:21

$$MTBF = \sum_{i=1}^{\infty} \frac{1}{2} \cdot [R_{sys_{i-1}} + R_{sys_i}] \cdot \Delta Time$$

where $R_{sys_0} = 1$

Based on the Intel® 2117 Dynamic Ram, the following three sections — (I, II, III) — compare various system configurations and failure rate parameters.

I. Table 1 shows the comparison of six memory configurations, ranging from 32K-bytes to 16 Megabytes. The Input parameters used were those listed in Table 2.

Table 1. Memory Configuration versus MTBF

FAILURE RATE = .127% / 1000 hrs

configuration	MTBF, non-ecc	MTBF, ecc	E.F.
16-bit word by 1 pg	49 k hrs	1170 k hrs	24
16-bit word by 128 pgs	390 hrs	95 k hrs	249
32-bit word by 1 pg	24 k hrs	658 k hrs	27
32-bit word by 128 pgs	195 hrs	53 k hrs	278
64-bit word by 1 pg	12 k hrs	355 k hrs	29
64-bit word by 128 pgs	98 hrs	29 k hrs	299

Table 2. Model Input Parameters

Combined HARD FAILURE RATE = 0.027% / 1000 hours

Failure distributions:

single cell	= 50.0%
row cells	= 15.6%
column cells	= 28.1%
row-column cells	= 6.3%
half-chip	= 0.0%
full-chip	= 0.0%
total	100%

SOFT ERROR FAILURE RATE = 0.1% / 1000 hrs - est.

These results show an enhancement factor of approximately 27 for a single page of memory and over 278 for 128 pages.

II. Table 3 shows the comparison of six memory configurations, between two soft error rates.

Table 3. Memory Configurations versus SE Rates

HARD FAILURE RATE = 0.027% / 1000 hrs

configuration	SOFT ERROR RATE .2% / 1000 hrs MTBF, ecc	SOFT ERROR RATE .5% / 1000 hrs MTBF, ecc
16-bit word by 1 pg	880 k hrs	575 k hrs
16-bit word by 128 pgs	70 k hrs	44 k hrs
32-bit word by 1 pg	492 k hrs	322 k hrs
32-bit word by 128 pgs	39 k hrs	24 k hrs
64-bit word by 1 pg	265 k hrs	173 k hrs
64-bit word by 128 pgs	21 k hrs	13 k hrs

III. Table 4 shows the comparison of a memory device with one failure type. The failure types compared are devices with a single cell failure modes and full-chip failure modes.

System A has devices with only "single cell" failure types and System B has only "full-chip" type. All other parameters are identical. Both system failure rates are 0.027%/1000 hrs.

Table 4. Single Cell versus Full Chip Failures

configuration:	SYSTEM A with single cell MTBF	SYSTEM B with full-chip MTBF
64-bit by 1 page	8.3 m hrs	103 k hrs
64-bit by 128 pages	730 k hrs	6 k hrs

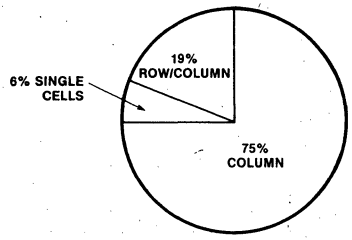
5.2.5 DISTRIBUTION

Error correction in a system does not alter or change the actual occurrence of failures. Failures still occur at the $MTBF_{necc}$ period based on the distribution in Figure 5. (For the example system, the soft error rate is three times the hard failure rate — .1% vs. .027% — which represents a soft error occurring 78% of the time.)

However, the fact that a multibit failure is required to cause a system failure in a system with ECC modifies the failure distribution; soft errors have much less effect than hard failures on system performance. Figure 22 demonstrates this by showing a modified distribution based on average cells per failure, the Rate Geometry Product, RGP.

INDIVIDUAL FAILURE RATE DISTRIBUTION OF A GIVEN TYPE TIMES THE NUMBER OF CELLS AFFECTED EQUALS AVERAGE CELLS PER FAILURE

FAILURE TYPE	% DISTRIBUTION		CELLS		AVERAGE CELLS
SOFT ERROR:					
SINGLE CELL	78.7%	×	1	=	0.79
HARD ERRORS:					
SINGLE CELL	10.6%	×	1	=	0.11
ROW	6.0%	×	128	=	7.93
COLUMN	3.4%	×	128	=	4.35
ROW/COLUMN	1.3%	×	256	=	3.32
	100%		TOTAL		16.5



DISTRIBUTION BY NUMBER OF AVERAGE CELLS

Figure 22.

The illustration shows the statistical average cell failure for each type derived by taking the product of the component failure rate distribution times the number of cells affected. For the 2117 example device, the total average cell failure is 16.2 of which 11.8 are column and row failures.

Intuitively, it can be seen that row and column failures are the most predominant, while the least predominant are soft errors and single cell hard errors.

6. SUMMARY

This Application Note presents step-by-step procedures for calculating system reliability. In a system without ECC, a fault of any type can cause system failure — predominantly types with the highest failure rates. In a system with ECC, only multi-bit errors within the same word cause system failure — predominantly types with the highest average cell errors as defined by the Rate Geometry Product. An Enhancement Factor, comparing a system without ECC to one with ECC, can be used to determine if error correcting techniques are advantageous for any specific memory system.

References

1. Randall C. Cork, "Reliability with Error-Detecting and Correcting Codes in Semiconductor Memories," Ph.D. dissertation, Arizona State University, 1975.
2. Bertram L. Amstadter, Reliability Mathematics Fundamentals; Practices; Procedures, McGraw Hill, N.Y., 1971.
3. Byron L. Newton, Statistics for Business, Science Research Associates, 1973.
4. Carl-Erik W. Sundberg, member IEEE, "Erasure and Error Decoding for Semiconductor Memories", IEEE 1978.
5. Peter Elias, "Error Free Coding", MIT.
6. S.K. Wang & K. Lovelace, "Improvement of Memory Reliability by Single-Bit-Error Correction", Texas Instruments Inc.

APPENDIX A

APPENDIX A

EQ:1a $\lambda_{hrd} = \lambda_{single} + \lambda_{row} + \lambda_{column} + \lambda_{row/col} + \lambda_{halfchip} + \lambda_{fullchip}$

EQ:1b $\lambda_{dev} = \lambda_{hrd} + \lambda_{sft}$

EQ:2 $R(t) = e^{-\lambda t}$

EQ:3 $R(t)_{sys} = R(t)_1 \cdot R(t)_2 \cdot R(t)_3 \cdot , , , R(t)_\eta$
where $R(t)_i = e^{-\lambda_i \cdot t}$

EQ:4 $R(t)_{sys} = R(t)^\eta = e^{-\eta\lambda t}$

EQ:5 $a^\eta + \eta a^{\eta-1} \cdot b + \frac{\eta(\eta-1)a^{\eta-2} \cdot b^2}{2!} + \frac{\eta(\eta-1)(\eta-2)a^{\eta-3} \cdot b^3}{3!} + \cdots + b^\eta$

EQ:6 $R(t) + Q(t) = 1$, then $Q(t) = 1 - R(t)$

EQ:7 $[\, R + Q \,]^\eta = 1$

EQ:8 $R^\eta + \eta R^{\eta-1} \cdot Q + \frac{\eta(\eta-1)R^{\eta-2} \cdot Q^2}{2!} + \frac{\eta(\eta-1)(\eta-2)R^{\eta-3} \cdot Q^3}{3!} + \cdots + Q^\eta = 1$

EQ:9 $R_T(t) = \underbrace{R^\eta}_{1st} + \underbrace{\eta \cdot R^{\eta-1} \cdot Q}_{2nd} - \text{binomial terms}$

EQ:10 $R(t)_{system} = [R(t)_{page}]^P$

EQ:11 $R(t)_{PAGE_{necc}} = R(t)^\eta_{DEV_{necc}} = e^{-\lambda \cdot n \cdot t}$

EQ:12 $R(t)_{SYS_{necc}} = [\, R(t)_{PAGE_{necc}} \,]^p = [\, R(t)_{DEV} \,]^{p \cdot \eta}$

EQ:13 $R(t)_{PAGEecc} = \left[R(t)_x^\eta + \eta \cdot R(t)_x^{\eta-1} \cdot Q(t)_x \right]^{\ell_x}$

EQ:14a $R_{T_1} = R^\eta_{f_1} + \eta R^{\eta-1}_{f_1} \cdot Q_{f_1}$

EQ:14b $R_{T_2} = R^\eta_{f_2} \cdot [R_{T_1}^{\ell_2}] + \eta [R_{f_2} \cdot R_{f_1}^{\ell_2}]^{\eta-1} \cdot Q_{f_2}$

EQ:14c $\lambda_{f_1} = \lambda_{dev} \cdot X_{f_1} \cdot \frac{fsz_{f_1}}{Msz} \qquad \lambda_{f_2} = \lambda_{dev} \cdot X_{f_2} \cdot \frac{fsz_{f_2}}{Msz}$

EQ:14d $R(t)_{page} = [R_{T_2}]^{\frac{Msz}{fsz_2}}$

EQ:15 $R_{T_3} = R^\eta_{f_3} \cdot [R_{T_2}]^{\ell_3} + \eta [R_{f_3} (R_{f_2} (R_{f_1})^{\ell_2})^{\ell_3}]^{\eta-1} \cdot Q_{f_3}$

EQ:16

$$\left[\begin{array}{l} \overset{N}{\underset{1}{i \leftarrow}} \left| \begin{array}{l} \ell_i = \dfrac{fsz_i}{fsz_{i-1}} \\ \lambda f_i = \lambda_{dev} \cdot X_i \cdot \dfrac{fsz_i}{Msz} \end{array} \right. \end{array} \right]$$

$$R(t)_{PAGE_{ecc}} = \left\{ \begin{array}{l} \overset{N}{\underset{1}{i \leftarrow}} \left| \begin{array}{l} R_i = e^{-\lambda f_i \cdot t},\ Q_i = 1 - R_i \\ RS_i = R_i \cdot (RS_{i-1})^{\ell_i} \\ R_{T_i} = R_i^\eta \cdot (R_{T_{i-1}})^{\ell_i} + \eta (RS_i)^{\eta-1} \cdot Q_i \end{array} \right. \end{array} \right\}^{\frac{Msz}{fsz_N}}$$

$$R(t)_{SYSTEM\,ecc} = [R(t)_{PAGE\,ecc}]^{Pages}$$

restrictions: $RS_0 = RT_0 = fsz_0 = 1.$

EQ:17 $$MTBF_{sys} = \int_0^{\infty} R(t)_{sys} \cdot dt$$

EQ:18 $$EF = \frac{MTBF_{sys\text{-}ecc}}{MTBF_{sys\text{-}necc}}$$

EQ:19 $$MTTF = \frac{\int_0^{T} R(t)_{sys\text{-}ecc} \cdot dt}{1 - R(T)_{sys\text{-}ecc}}$$

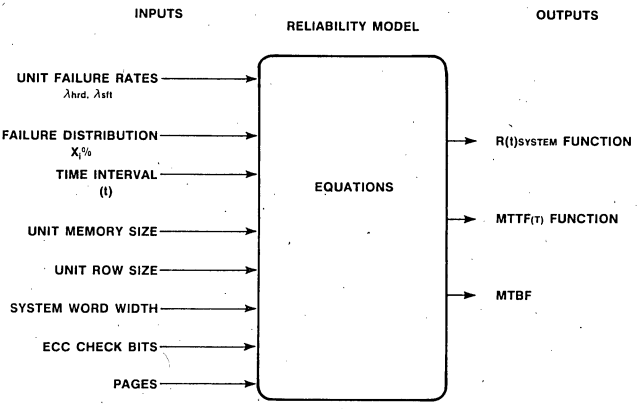
EQ:20 $$EF_{mnt} = \frac{MTTF}{MTBF_{sys\text{-}ecc}}$$

EQ:21 $$MTBF = \sum_{i=1}^{\infty} \text{½} \cdot [R_{sys_{i-1}} + R_{sys_i}] \cdot \Delta Time$$

where $R_{sys_0} = 1$

APPENDIX B

APPENDIX B


INPUTS
RELIABILITY MODEL
OUTPUTS
UNIT FAILURE RATES
λhrd, λsft
FAILURE DISTRIBUTION
Xi%
TIME INTERVAL
(t)
UNIT MEMORY SIZE
UNIT ROW SIZE
SYSTEM WORD WIDTH
ECC CHECK BITS
PAGES
EQUATIONS
R(t)SYSTEM FUNCTION
MTTF(T) FUNCTION
MTBF

```
FORTRAN IV       V02.04

      C#########################################################
      C     #     ECC RELIBILITY MODEL     REV 6B  FEB79    #
      C     #                                                 #
      C     #     INTEL CORP                                  #
      C     #     MEMORY PRODUCTS DIVISION                    #
      C     #     APPLICATIONS LAB                            #
      C     #     ALOHA, OREGON                               #
      C     #                                                 #
      C     #                                                 #
      C     #     ERROR CORRECTION RELIABILITY
      C     #     APPLICATIONS NOTE                           #
      C     #################################################
      C
0001           IMPLICIT REAL*8 (D,R,S,T,Z)
0002           DIMENSION KM(2),LH(2),KL(4),LQ(4)
0003           BYTE LL(2),LR(2),IBUF(80),ILIST(80)
0004           INTEGER*4 IIPTR,LPTR(13),IABORT,IHELP,LMFLGS(3),IIXFG
0005           COMMON /ECC1/RXZ,RXS,RXR,RXC,RXF,RXE,RXH,RXT,RCNF,SZER,SXX
0006           COMMON /ECC2/RMSZ,RCSZ,RWD,BPG,RZD,RZDD,REC,RZER,R1,RTM,RTSF
0007           COMMON /ECC3/IM,ILLM,IULM,RSQ,JSFLG,EPGX,ISFLG,ST,R,RTH,R2
0008           COMMON /ECC4/ISW,RFF,IPM,RTTF,RZTTL,ICST,RALMT,IDBK,IQFG,IUCD
0009           COMMON /ECC5/ISFG,REC1,REC2,IEFLG,RZSYS,ILIM,IDFLG,RAVE
0010           COMMON /ECC6/ITIN,ITOUT,ILP
0011           COMMON /ECC7/RZZ,RZS,RZR,RZC,RZF,RZE,RZH,RZT,RZDX
0012           COMMON /ECC8/ECZ,ECS,ECR,ECC,ECF,ECE,ECH,ECT,ECX
0013           COMMON /ECC9/EW,EW1,EW2,RW,RW1,RW2,S,T,TSFT,THRD
0014           COMMON /ECCA/EPG,EBD,EPSZ,ECA,EPX,EPY,EPZ
0015           COMMON /ECCC/I,IMN,RPRT,RTD,RTPG,RTX,RZDZ,RXX,RSPC1,RSPC2
0016           DATA KL/'  ','* ',' $','*$'/,IABORT/'ABOR'/,IHELP/'HELP'/
0017           DATA KM/'KB','MB'/,LL/' ','<'/,LR/' ','>'/
0018           DATA LMFLGS/'SYS','MPD','MSO'/
      C                    1       2       3       4       5       6       7
0019           DATA LPTR/'LIST','SIZE','RATE','DIST','COMM','DUMP','FLAG',
     *  'HXDR','CYCL','PURG','NECC','SECC','DECC'/
      C         8       9      10      11      12      13
0020           DATA LQ/'Q1','Q2','QX','QZ'/
0021           DATA LH/' -','M-'/,IBEL/1799/
0022           DATA ITIN/5/,ITOUT/7/,ILIM/10/,IDFLG/1/,ILP/6/
0023           DATA RXZ/.7874D0/,RXS/.50D0/,RXR/.156D0/,RXC/.281D0/,RXE/.062D0
0024           DATA RXH/0.0D0/,RXT/0.0D0/,RCNF/.37D0/,SZER/.1D-4/
0025           DATA RRXS/50.D0/,RRXR/15.6D0/,RRXC/28.1D0/,RRXE/6.2D0/
0026           DATA RRXH/0.0D0/,RRXT/0.0D0/,SXX/1.0D0/
0027           DATA RMSZ/16384.0D0/,RCSZ/128.0D0/,RWD/64.0D0/,BPG/128.0/
0028           DATA RZD/0.00027D0/,RZSE/0.001D0/,REC/-1.0D0/,RZER/0.0D0/
0029           DATA RRZD/0.027D0/,RRZSE/0.1D0/,RRZTTL/0.0D0/,RRZSYS/0.0D0/
0030           DATA R1/1.0D0/,RTM/2500.0D0/,RTSF/1000.0D0/,RAVE/228.0D0/
0031           DATA IM/0/,ILLM/0/,IULM/30/,RSQ/2.0D0/,JSFLG/1/,EPGX/1.0/
0032           DATA ISFLG/1/,ST/0.0D0/,R/100.0D0/,RTH/100000.0D0/,R2/2.0D0/
0033           DATA ISW/1/,RFF/0.0D0/,IPM/10/,RTTF/0.0D0/,RZTTX/0.0D0/
0034           DATA ICST/2/,RALMT/0.01D0/,IDBK/1/,IQFG/1/,IUCD/0/,ISFG/1/
0035           DATA REC1/8.0D0/,REC2/15.0D0/,IEFLG/0/,RZSYX/0.0D0/
0036           DATA ILFG/1/,RTMSO/0.0D0/,RZTMP/0.0D0/,JXFG/1/
0037           DATA RZS1/0.0D0/,RZS2/0.0D0/,RRZS1/0.0D0/,RRZS2/0.0D0/
0038           DATA TTMCYL/5.0D2/,TMCYL/5.0D2/,TTRCYL/1.5D4/,TRCYL/1.5D4/
```

```
FORTRAN IV      V02.04

0039          DATA TREF/7.D3/,RRX1A/66.D0/,RX1A/.66D0/,RRX1B/33.D0/,RX1B/.33D0/
0040          RPSZ=RZER
0041          SLM=RZER
0042          EPG=BPG
      C
      C#################### RELIABILITY EQUATIONS ############################
      C
      C       R[T] = N*QT*[ RT*(RE*(RF*(RS*(RZ)**MS)**MF)**ME)**MT ]**(N-1)  +
      C
      C            (RT**N)*[ N*QE*[ RF*(RF*(RS*(RZ)**MS)**MF)**ME ]**(N-1)  +
      C
      C            (RE**N)*[ N*QF*[ RF*(RS*(RZ)**MS)**MF ]**(N-1)  +
      C
      C               (RF**N)*[ N*QS*[ RS*(RZ)**MS ]**(N-1)  +
      C
      C                  (RS**N)*[ N*QZ*[ RZ ]**(N-1)  +
      C
      C                        (RZ**N) ]**MS ]**MF ]**ME ]**MT
      C
      C       WHERE:  RZ = SOFT ERROR              RS = SINGLE CELL
      C           RF = COLUMN                  RE = ROW /COLUMN
      C           RH = HALF CHIP               RT = TOTAL CHIP
      C
      C           MS -> SINGLE CELL TO SOFT ERROR RATIO  - 1
      C           MF = COLUMN TO SINGLE CELL RATIO
      C           ME = ROW/COLUMN TO COLUMN RATIO
      C           MH = HALF CHIP TO ROW/COLUMN RATIO
      C           MT = TOTAL CHIP TO HALF CHIP RATIO
      C
0043          WRITE (ITOUT,10)
0044    10    FORMAT (T2,'<<<< ERROR CORRECTION RELIABILITY >>>>',/,
             C T4,'INTEL CORP.  MPD/MCO     DJM FEB79',//,
             C T4,'FOR PROGRAM DESCRIPTION ENTER >  HELP')
      C#############################################################################
      C
      C                        INPUT PARAMETERS
      C
      C#############################################################################
      C
0045    100   CONTINUE
0046          WRITE (ITOUT,90) IBEL
0047    90    FORMAT (A2,T5,'POINTER,  INDEX,  TIME.,  PAGE,  BOARD')
0048    101   FORMAT (T2,'** LIST OUTPUT PARAMETERS **',/,
             C T2,'* LOWER,  UPPER,  SKIP,  UNCOND,  MAINT,  .CONF')
0049    102   FORMAT (T2,'** COMPONENT & MEMORY SYSTEM PARAMETERS **',/,
             C T2,'* RAMSIZE.,  COLSIZE,  WORDSIZE,  CHECKBITS')
0050    103   FORMAT (T2,'** DEVICE & SYSTEM FAILURE RATES **',/,
             C T2,'* .HARD%,  .SOFT%,  .TTL%,  .SYSTEM%')
0051    104   FORMAT (T2,'** DEVICE HARD FAILURE TYPE DISTRIBUTION **',/,
             C T5,'HINT:  SC ROW COL CMB HLF FULL',/,
             C T2,'* X2.%,  X3.%,  X4.%,  X5.%,  X6.%,  X7.%')
0052    105   FORMAT (T2,'** HEADER COMMENT **',/,T2,'*')
0053    106   FORMAT (T2,'#')
0054    107   FORMAT (T2,'** ERROR **',1X,I2)
```

```
FORTRAN IV       V02.04

0055   109  FORMAT (T2,'** CYCLE **      SE CYCLES & DISTRIBUTION'
           C ,/,T5,'SOFT ERROR ALGORITHM - CYCLES IN NS',/,
           C T2;'* MEMORY.   REFRESH.   BITLINE.%  SNGLE-CELL.%',/,T2,'*')
       C
0056           READ (ITIN,95) IIPTR,III,RRTM,IRPG,IRBD
0057   95      FORMAT (A4,1X,I6,F10.2,I5,I5)
0058           IPTR=0
0059           DO 94,J=1,13
0060           IF (IIPTR.EQ.LPTR(J)) IPTR=J
0062   94      CONTINUE
0063           IF (IIPTR.EQ.IABORT) STOP
0065           IF (IIPTR.EQ.IHELP) CALL HELP
0067           IF (IPTR.EQ.0) GO TO 100
0069           IF (IPTR.LT.10) GO TO 96
0071           RTM=RRTM
0072           EPG=IRPG
0073           EBD=IRBD
0074           I=III
       C      ..... DETERMINE WHAT TEST .....
0075           IMM=IPTR-10
0076   96      CONTINUE
0077           IDMP=1
0078           IF (IPTR.EQ.6) IDMP=2
0080           IF (IPTR.GE.11) GO TO 200
0082           GO TO (98,115),IDMP
0083   98      CONTINUE
0084           GO TO (110,120,130,140,150,160,170,180,190,195),IPTR
0085   110     WRITE (ITOUT,101)
0086           READ (ITIN,112) ILLM,IULM,ISW,IUCD,JS,RCNF
0087   112     FORMAT (5(I6),F10.8)
       C      ... DISABLE FUDGE FACTOR ...
0088           RFF=0.0
0089           IEFLG=0
0090           IF ((ILLM.LE.0).OR.(IULM.LE.0).OR.(JS.LT.0)) IEFLG=1
0092           IF ((RCNF.LE.0.0).OR.(RFF.LT.0.0)) IEFLG=2
0094           IF ((IUCD.LT.0).OR.(JS.LT.0)) IEFLG=3
0096           IF (IEFLG.EQ.0) GO TO 100
0098           WRITE (ITOUT,107) IEFLG
0099   115     WRITE (ITOUT,117) LPTR(1),ILLM,IULM,ISW,IUCD,JS,RCNF,RFF
0100   117     FORMAT (T2,A4,':  ',5(I6,1X),F10.8,1X,F8.0)
0101           WRITE (ITOUT,106)
0102           GO TO (100,125),IDMP
       C
0103   120     CONTINUE
0104           WRITE (ITOUT,102)
0105   121     READ (ITIN,122) RMSZ,JCSZ,JWD,JEC
0106   122     FORMAT (F8.0,3(I5))
0107           IEFLG=0
0108           IF ((RMSZ.LT.1.).OR.(JCSZ.LT.1).OR.(JWD.LT.1)) IEFLG=
0110           RCSZ=JCSZ
0111           RWD=JWD
0112           REC=JEC
0113           IF (IEFLG.EQ.0) GO TO 100
0115           WRITE (ITOUT,107) IEFLG
```

```
FORTRAN IV      V02.04

0116   125  WRITE (ITOUT,127) LPTR(2),RMSZ,JCSZ,JWD,JEC
0117   127  FORMAT (T2,A4,': ',F8.0,1X,3(I5,1X))
0118        WRITE (ITOUT,106)
0119        GO TO (100,135),IDMP
0120   130  CONTINUE
0121        WRITE (ITOUT,103)
0122        READ (ITIN,132) RRZD,RRZSE,RRZTTL,RRZSYS
0123   132  FORMAT (4(F12.8))
0124        IEFLG=0
0125        IF ((RRZD.EQ.0.0).AND.(RRZSE.EQ.0.0)
           * .AND.(RRZTTL.EQ.0.0).AND.(RRSYS.EQ.0.0)) IEFLG=1
0127        IF ((RRZTTL.LT.0.0).OR.(RRZSYS.LT.0.0).OR.(RRZD.LT.0.0)
           1 .OR.(RRZSE.LT.0.0)) IEFLG=2
0129        IF (IEFLG.EQ.0) GO TO 136
0131        WRITE (ITOUT,107) IEFLG
0132   135  WRITE (ITOUT,137) LPTR(3),RRZD,RRZSE,RRZTTL,RRZSYS
0133   137  FORMAT (T2,A4,': ',4(F12.8,1X))
0134        WRITE (ITOUT,106)
0135        GO TO (100,145),IDMP
       C    ... CONVERT FROM PERCENT ...
0136   136  RZD=RRZD/100.
0137        RZSE=RRZSE/100.
0138        RZTTX=RRZTTL/100.
0139        RZSYX=RRZSYS/100.
0140        GO TO 100
0141   140  CONTINUE
0142        WRITE (ITOUT,104)
0143        READ (ITIN,142) RRXS,RRXR,RRXC,RRXE,RRXH,RRXT,RPSZ
0144   142  FORMAT (6(F11.8),F8.0)
       C    .... DISABLE PARTIALS ....
0145        RPSZ=0.0
0146        IEFLG=0
0147        RXS=RRXS/100.
0148        RXR=RRXR/100.
0149        RXC=RRXC/100.
0150        RXE=RRXE/100.
0151        RXH=RRXH/100.
0152        RXT=RRXT/100.
0153        SXX=RXS+RXR+RXC+RXE+RXH+RXT
0154        IF (SXX.GT.R1) IEFLG=1
0156        IF (RPSZ.LT.0.0) IEFLG=2
0158        IF (IEFLG.EQ.0) GO TO 100
0160        WRITE (ITOUT,107) IEFLG
0161   145  WRITE (ITOUT,147) LPTR(4),RRXS,RRXR,RRXC,RRXE,RRXH,RRXT,RPSZ
0162   147  FORMAT (T2,A4,': ',6(F10.6,1X),F8.0)
0163        WRITE (ITOUT,106)
0164        GO TO (100,155),IDMP
0165   150  WRITE (ITOUT,152)
0166   152  FORMAT (T2,'* INPUT BUFFER')
0167        READ (ITIN,154) ICHRS,(IBUF(IB),IB=1,ICHRS)
0168   154  FORMAT (T2,Q,72A1)
0169        GO TO 100
0170   155  WRITE (ITOUT,156) LPTR(5),(IBUF(IB),IB=1,72)
0171   156  FORMAT (T2,A4,': ',72A1)
```

```
FORTRAN IV      V02.04

0172            GO TO (100,193),IDMP
0173     160    CONTINUE
0174     170    CONTINUE
       C        FLAGS
0175            WRITE (ITOUT,172)
0176     172    FORMAT (T2,'* FLAGS -  SET TTL & SYSTEM FAILURE RATE MODE',
              C /,T10,'- & FLAG FOR ONE OR TWO DIMENSION MERGE',/,
              C T2,'* NNN, 0#')
0177            READ (ITIN,173) IIXFG,IIQFG
0178     173    FORMAT (A3,1X,A2)
0179            IEFLG=0
0180            JXFG=0
0181            DO 174 IJ=1,3
0182            IF (IIXFG.EQ.LMFLGS(IJ)) JXFG=IJ
0184            IF (IIQFG.EQ.LQ(IJ)) IQFG=IJ
0186     174    CONTINUE
0187            WRITE (ITOUT,179) LMFLGS(JXFG),LQ(IQFG),IIXFG,IIQFG
0188     179    FORMAT (T2,'##SPECIAL CK ->',2X,A3,1X,A2,5X,A3,1X,A2,/)
0189            IF ((IQFG.LT.1).OR.(IQFG.GT.3)) IEFLG=1
0191            IF ((JXFG.LT.1).OR.(JXFG.GT.3)) IEFLG=2
0193            IF (IEFLG.EQ.0) GO TO 100
0195     175    WRITE (ITIN,176) IEFLG
0196     176    FORMAT (T2,'** ERROR -',I2,' RETRY, HINT: SYS, MPD, OR MSO')
0197            GO TO 170
0198     180    CONTINUE
       C        HEADER
0199            IMM=0
0200            EPG=IRPG
0201            EBD=IRBD
0202            REC=III
0203            WRITE (ITOUT,182) JXFG,REC,EPG,EBD
0204     182    FORMAT (T2,'* HEADER > FLG-',I3,2X,'CK-',F4.0,2X,'PG-',
              C F4.0,3X,'BD-',F4.0)
0205            GO TO 200
0206     190    CONTINUE
0207            WRITE (ITOUT,109)
0208            READ (ITIN,192) TTMCYL,TTRCYL,RRX1A,RRX1B
0209     192    FORMAT (2(F12.0),2(F8.4))
0210            IEFLG=0
0211            IF ((TTMCYL.LE.0.0).OR.(TTRCYL.LE.0.0)) IEFLG=1
0213            IF ((RRX1A+RRX1B).GT.100.) IEFLG=2
0215            IF ((RRX1A.LT.0.).OR.(RRX1B.LT.0.)) IEFLG=3
0217            IF (IEFLG.NE.0) GO TO 193
0219            TMCYL=TTMCYL
0220            TRCYL=TTRCYL
0221            RX1A=RRX1A/100.
0222            RX1B=RRX1B/100.
0223            GO TO 100
0224     193    WRITE (ITOUT,194) TTMCYL,TTRCYL,RRX1A,RRX1B
0225     194    FORMAT (T2,'* ',2(F12.0,1X),2X,2(F8.4,1X))
0226            GO TO 100
0227     195    CONTINUE
       C        PURGE
0228            DO 197,INIT=1,5
```

```
FORTRAN IV      V02.04

0229          WRITE (ILP,196) (ILIST(NN),NN=1,72)
0230    196   FORMAT (T2,72A2)
0231    197   CONTINUE
0232          WRITE (ILP,198)
0233    198   FORMAT (1H1)
0234          GO TO 100
      C
      C##############################################################################
      C
      C                    INITIALIZE PARAMETERS
      C
      C##############################################################################
0235    200   CONTINUE
0236          IF (IEFLG.EQ.0) GO TO 202
0238          WRITE (ITIN,201) IEFLG
0239    201   FORMAT (T2,'ERROR COND EXISTS - ABORT ',I2)
0240          GO TO 100
0241    202   CONTINUE
0242          SLM=JS*RTM
0243          RE=0.00
0244          IF (REC.LT.RZER) GO TO 208
0246          RE=REC
0247          GO TO 209
0248    208   CONTINUE
0249          IF (IMM.EQ.2) RE=REC1
0251          IF (IMM.EQ.3) RE=REC2
0253    209   RW=RWD+RE
0254    210   CONTINUE
      C                      ..... NAME CHANGES FOR SPEED REASON'S
0255          EW=RW
0256          RW1=RW-1.0
0257          EW1=RW1
0258          RW2=RW-2.0
0259          EW2=RW2
      C     .... SOFT ERROR ALGORITHM BY CYCLE TIMES ....
0260          SMCYL=RMSZ
0261          SMTIM=SMCYL*TMCYL
0262          SRCYL=SMTIM/TRCYL
0263          SRPG=(EPG*EBD)-1.
0264          SECYL=(SMTIM+(SMTIM*SRPG))/(SMCYL+(SRCYL*SRPG))
0265          SNRMLZ=TREF/SECYL
0266          RZDD=(RX1A*RZSE*SNRMLZ)+(RX1B*RZSE)
0267          IF (JXFG.NE.2) RZDD=RZSE
0269          RZDX=RZD+RZDD
0270          RXZ=RZDD/RZDX
0271          RXF=RXR+RXC
0272          RRSZ=RMSZ/RCSZ
0273          RFSZ=(RCSZ+RRSZ)/2.0
0274          RESZ=RMSZ/(RCSZ+RRSZ)
0275          RHSZ=2.0
0276          RTSZ=1.0
0277          ECZ=1.0
0278          ECS=1.0
0279          ECR=RRSZ
```

```
FORTRAN IV      V02.04

0280          ECC=RCSZ
0281          ECF=RFSZ
0282          ECE=(RRSZ+RCSZ)/RRSZ
0283          ECH=RMSZ/((RRSZ+RCSZ)*2.0)
0284          ECT=2.0
0285          T=RZER
0286          S=RZER
0287          RINC=RTM/RTSF
0288          RMTBF=RZER
0289          RMTTF=RZER
0290          RMNT=RZER
0291          LG=1
0292          IMFLG=1
0293          ISFLG=1
0294          JSFLG=1
0295          IXFLG=1
0296          ILFLG=1
0297          RMIL=1000000.
0298          IF (RTM.GE.100000.0) IMFLG=2
0300          RZZ=(RXZ*RZDX)/RMSZ
0301          RZS=(RXS*RZD)/RMSZ
0302          RZR=(RXR*RZD)/RRSZ
0303          RZC=(RXC*RZD)/RCSZ
0304          RZF=(RXF*RZD)/RFSZ
0305          RZE=(RXE*RZD)/RESZ
0306          RZH=(RXH*RZD)/RHSZ
0307          RZT=(RXT*RZD)
0308          RHRD=1.0-RXZ
0309          RTMP=(RXS/RMSZ)+(RXF/RFSZ)+(RXE/RESZ)+(RXH/RHSZ)+(RXT/RTSZ)
0310          RAVE=RXZ+((RTMP*RMSZ)*RHRD)+RFF
0311          AZ=RXZ
0312          AS=RHRD*RXS
0313          AF=RHRD*RXF*(RMSZ/RFSZ)
0314          AE=RHRD*RXE*(RMSZ/RESZ)
0315          AH=RHRD*RXH*(RMSZ/RHSZ)
0316          AT=RHRD*RXT*(RMSZ/RTSZ)
0317          AXX=AS+AF+AE+AH+AT
0318          BZ=(AZ/(AZ+AXX))*R
0319          BHRD=100.00-BZ
0320          BS=(AS/AXX)*R
0321          BF=(AF/AXX)*R
0322          BE=(AE/AXX)*R
0323          BH=(AH/AXX)*R
0324          BT=(AT/AXX)*R
0325          RZDZ=RZDX*(RAVE/RMSZ)
0326          ECA=RMSZ/RAVE
0327          RZREV=0.0
0328          RZTOL=0.0
0329          RTPM=RTM/IPM
0330          ROLD=1.0
0331          ISFLG=1
0332          RTSED=RTTF
0333          RSPC1=1.0
0334          RSPC2=2.0
```

```
FORTRAN IV       V02.04

0335          EPX=((RMSZ-RPSZ)/RMSZ)*EPG
0336          EPY=(RPSZ/RMSZ)*EPG
0337          EPZ=0.0
0338          IF (RPSZ.LE.RMSZ) GO TO 211
0340          EP=(RPSZ-RMSZ)/RMSZ
0341          EPZ=EP*EPG
0342          EPY=(1.0-EP)*EPG
0343          EPX=0.0
0344          IF (RPSZ.LE.2.0*RMSZ) GO TO 211
0346          EPX=0.0
0347          EPY=0.0
0348          EPZ=0.0
0349    211   CONTINUE
0350          EPSZ=RPSZ
0351          RZTMP=0.0
0352          IF ((RTTF.NE.RZER).OR.(IMM.EQ.1)) GO TO 212
0354          RZTMP=RWD*EPG*EBD*((RZD*SXX)+RZDD)
0355          IF (RZTMP.GT.0.0) RTSED=1000.0/RZTMP
0357    212   CONTINUE
0358          RZTTL=0.0
0359          RZSYS=0.0
0360          RTMSO=0.0
0361          GO TO (213,214,214),JXFG
      C       NORMAL  TTL SYSTEM CALCULATION.....
0362    213   RZTTL=RZTTX
0363          RZSYS=RZSYX
0364          IF (RZSYS.GT.0.0) RTMSO=1000.0/RZSYS
0366          GO TO 216
      C       MSO MODE  RTMSO CALCULATION FOR HEADER ONLY .....
0367    214   RZTMP=(EBD*RZTTX)+RZSYX
0368          IF (RZTMP.GT.0.0) RTMSO=1000.0/RZTMP
0370          IF (JXFG.LT.3) GO TO 216
0372          RZSYS=RZTMP
0373    216   CONTINUE
0374          IMN=1
0375          IF (RPSZ.GT.RZER) IMN=2
0377    217   CONTINUE
0378          IF ((III.EQ.0).OR.(IMM.EQ.0)) GO TO 220
0380          T=III*RINC
0381          S=T
0382          IF (JS.EQ.0) GO TO 395
0384          GO TO (218,219),ISFG
0385    218   S=JS*RINC
0386          GO TO 395
0387    219   IF (RTSF.GT.0.0) S=SLM/RTSF
0389          GO TO 395
      C
      C###############################################################################
      C
      C                    PRINT HEADER
      C
      C###############################################################################
0390    220   WRITE (ILP,221) (IBUF(IB),IB=1,72)
0391    221   FORMAT (T2,72A1,/,T2,8('----------'))
```

```
FORTRAN IV      V02.04

0392          QC=RWD*EPG*EBD
0393          RSS=(QC*RMSZ)/8192.
0394          IFLG=1
0395          IF (RSS.LT.1000.0) GO TO 16
0397          IFLG=2
0398          RSS=RSS/1000.0
0399    16    CONTINUE
0400          KMM=KM(IFLG)
0401          BPG=EPG*EBD
0402          WRITE (ILP,13) LMFLGS(JXFG),LQ(IQFG),RSS,KMM,RWD,RE,EPG,EBD,
             * RWD*BPG,RE*BPG,RMSZ,RCSZ,RZTTX*R,RZSYX*R,RTSED,RTMSO
0403    13    FORMAT (T2,A4,'/',A2,T20,
             C 'ECC PROBABILITY PROGRAM "INTEL-MPD/MC."',
             C //,T2,'MEMORY SYS: SIZE->  ',F6.1,A2,T36,'WORD WIDTH-> ',F4.0,
             C '+',F3.0,T58,'NO. PAGES-> ',F4.0,'X',F4.0,/,T2,'COMPONENT',
             C ': TOTAL->',F7.0,'+',F7.0,T38,'RAM SIZE-> ',F8.0,T58,
             C 'COL SIZE-> ',F4.0,/,T2,'SYSTEM DATA:',2X,'TTL RATE -> ',
             C F8.5,'%/1K-HRS,',2X,'SYSTEM RATE -> ',F8.5,'%/1K-HRS',
             C //,T2,'FAILURE DATA:',T17,'MTBF.NECC ->',F11.2,'HRS,',T45,
             C 'MTBF.SYS -> ',F11.2,'HRS')
0404          WRITE (ILP,224) RPSZ,RZD*R,SLM,RZDD*R,RTM,RAVE
0405    224   FORMAT (T2,'HARD ERRORS:',T19,'PARTIAL ->',F6.0,'CELLS/PG',
             C T45,'RATE -> ',F10.6,'% / 1000 HRS',/,T2,'SOFT ERRORS:',T16,
             C '"*"',T20,'MAINT  -> ',F10.0,'HRS,',T45,'RATE -> ',F10.6,
             C '% / 1000 HRS',/,T2,'ANALYSIS DATA:',T20,'PERIOD -> ',F10.2,
             C 'HRS,',T45,'AVE CELL FAILURE ->',F8.1)
0406          K=100
0407          WRITE (ILP,11)
0408    11    FORMAT (T2,'FAILURE TYPE RATIOS:   ',5('----------'),/,T2,
             C '=TYPE=',T14,'=DISTRIBUTION=GEOMETRY=UNIT.RATE/1K HRS=',
             C 'AVE.CELLS=ECC.DISTR=EXPS=')
0409          WRITE (ILP,12) RXZ*R,RMSZ,RZZ*R,AZ,BZ,RHRD*R,BHRD,RXS*R,
             C RMSZ,RZS*R,AS,BS,ECS,RXF*R,RFSZ,RZF*R,AF,BF,ECF
0410    12    FORMAT (T3,'SOFT ERROR  ->[',F7.3,'%]/',F7.0,' = ',E12.5,'%,',
             C F11.3,2X,'[',F6.2,'%]',/,T3,'HARD ERRORS ->[',F7.3,'%]',
             C T65,'[',F6.2,'%]',/,
             C T3,'SINGLE CELL ->',F8.4,'% /',F7.0,' = ',E12.5,'%,'F11.3,2X,
             C F6.2,'%',2X,F4.0,/,T3,'ROW OR COL  ->',F8.4,'% /',F7.0,' = ',
             C E12.5,'%,',F11.3,2X,F6.2,'%',2X,F4.0)
0411          WRITE (ILP,17) RXE*R,RESZ,RZE*R,AE,BE,ECE,RXH*R,RHSZ,RZH*R,
             C AH,BH,ECH,RXT*R,RTSZ,RZT*R,AT,BT,ECT
0412    17    FORMAT (T3,'COLUMN/ROW  ->',F8.4,'% /',F7.0,' = ',
             C E12.5,'%,',F11.3,2X,F6.2,'%',2X,F4.0,/,T3,'HALF CHIP   ->',
             C F8.4,'% /',F7.0,' = ',E12.5,'%,'F11.3,2X,F6.2,'%',2X,F4.0,/,
             C T3,'TOTAL CHIP  ->',F8.4,'% /',F7.0,' = ',E12.5,'%,',
             C F11.3,2X,F6.2,'%',2X,F4.0,/)
0413          IF (IMM.NE.0) GO TO 390
0415          WRITE (ILP,380)
0416    380   FORMAT (1H1,T2,8('---------'))
0417          GO TO 100
0418    390   CONTINUE
      C
      C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      C
```

```
FORTRAN IV      V02.04

      C                    EQUATION LOOP
      C
      C
0419          WRITE (ILP,14) LPTR(IMM+10),LH(IMFLG)
0420    14    FORMAT (T2,'PERIOD',T11,'PM@T: ',T24,'R(T) ',A4,T44,'MTTF',
     C T52,'ENHANCEMENT',T65,'% - R(T)',/,T2,'-----',T8,
     C '<',A2,' HRS>',T23,'=FUNCTION=',T42,'<  HRS >',T52,
     C '  FACTOR  ',T65,'    @T  ')
0421          I=0
0422    395   TSFT=S
0423          THRD=T
0424    400   CONTINUE
0425          JG=1
0426          RENH=RZER
      C
      C##############################################################################
      C
      C              RELIABILITY EQUATIONS
      C
      C
0427          CALL TEST (IMM)
      C
      C##############################################################################
      C
      C                    OUTPUT DATA
      C
      C##############################################################################
0428    500   CONTINUE
0429          IF (III.EQ.0) GO TO 510
0431          WRITE (ITOUT,502) LPTR(IMM+10),I,T*RTSF,RTPG
0432    502   FORMAT (T2,'** ',A4,5X,'I => ',I4,5X,'T => ',F10.2,10X,
     C 'R => ',F10.7)
0433          GO TO 100
0434    510   IZFLG=0
0435          RTIM=T*RTSF
0436          RTMX=RTIM
0437          IF (IMFLG.EQ.2) RTMX=RTIM/RMIL
0439          IF (I.EQ.0) GO TO 522
0441          RINT=((ROLD+RTPG)/2.0)*RTM
0442          IF (ISFLG.EQ.2) RINT=RZER
0444          RMTTF=RMTTF+RINT
0445          IF (1.0-RTPG.LE.SZER) GO TO 517
0447          RMNT=RMTTF/(1.0-RTPG)
0448          GO TO 520
0449    517   IZFLG=2
0450    520   ROLD=RTPG
0451    522   CONTINUE
0452          IF ((RTPG.LE.RCNF).AND.(LG.EQ.1)) JG=2
0454          KLI=KL(IXFG)
0455          IFLG=0
0456          IF (((I/ISW)*ISW.NE.I).OR.(I.LT.ILLM)) IFLG=1
0458          IF (I.GT.IULM) IFLG=1
0460          IF (I.EQ.IUCD) IFLG=0
0462          RI=I
```

```
FORTRAN IV      V02.04

      C      IF (I.EQ.0) GO TO 525
      C      IF (RTPG.LE.0.0) GO TO 525
      C      RZREV=(DLOG(1.0/RTPG))/T
      C      RZTOL=((RZTOL*(RI-1.0))+RZREV)/RI
0463   525   CONTINUE
0464         IF ((I.NE.0).AND.(IFLG.EQ.1)) GO TO 550
0466         L1=LL(JG)
0467         L2=LR(JG)
0468         RCFD=RTPG*100.00
0469         IF (RTSED.GE.1.0) RENH=RMNT/RTSED
0471         IF ((ISFLG+JSFLG.EQ.3).AND.(ILFG.EQ.1)) GO TO 535
0473         GO TO (531,532),IMFLG
0474   531   WRITE (ILP,505) I,RTMX,KLI,RTPG,RMNT,RENH,L1,RCFD,L2
0475   505   FORMAT (T2,I4,T7,F8.0,A2,T24,F8.5,T40,F10.0,T52,F8.0,
            C T65,A1,F5.1,'%',A1)
0476         GO TO 535
0477   532   WRITE (ILP,506) I,RTMX,KLI,RTPG,RMNT,RENH,L1,RCFD,L2
0478   506   FORMAT (T2,I4,T7,F8.2,A2,F8.5,T40,F10.0,T52,F8.0,
            C T65,A1,F5.1,'%',A1)
0479   535   CONTINUE
0480         IF (JG.EQ.2) LG=2
0482   550   CONTINUE
      C
      C##################################################################################
      C
      C                    CALCULATE NEXT 'T' INTERVAL
      C
      C##################################################################################
      C
0483         GO TO (560,570),ISFG
0484   560   IXFLG=1
0485         IXFG=1
0486         IF (JSFLG+ISFLG.LT.4) GO TO 562
0488         JSFLG=1
0489         ISFLG=1
0490   562   CONTINUE
0491         GO TO (564,566),JSFLG
0492   564   T=T+RINC
0493         S=S+RINC
0494         I=I+1
0495         GO TO 568
      C                        ..... SCRUB SOFT ERRORS ....
0496   566   S=RZER
0497   568   CONTINUE
0498         TSFT=S
0499         THRD=T
0500         IF ((S.GE.(SLM/RTSF)).AND.(SLM.NE.RZER)) IXFLG=2
0502         IF (JSFLG.EQ.2) ISFLG=2
0504         IF (IXFLG.EQ.2) JSFLG=2
0506         IF ((LG.EQ.2).AND.(ISFLG.EQ.1)) JSFLG=1
0508         IXFG=ISFLG
0509         GO TO 580
0510   570   CONTINUE
      C          .... SPECIAL MODE AVERAGE SOFT ERROR RATE
```

FORTRAN IV V02.04

```
        IXFG=3
        T=T+RINC
        I=I+1
        GO TO (574,576),IXFLG
  574   S=S+RINC
        GO TO 578
  576   S=RZER
        IXFLG=1
  578   CONTINUE
        THRD=T
        TSFT=S
        IF (SLM.EQ.RZER) GO TO 580
        TSFT=R1+(SLM/1000.)
        IF (S.GE.(SLM/RTSF)) IXFLG=2
        IF (IXFLG.EQ.2) IXFG=4
  580   CONTINUE
C
C             .... END EQUATION LOOP ....
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
        IFLG=0
        GO TO (585,590),IDFLG
  585   CONTINUE
C                       .... ITERATE TILL LIST COUNT ....
        IFLG=1
        IF ((I.GT.IULM).OR.(RTPG.LE.RALMT)) GO TO 650
        GO TO 400
  590   CONTINUE
C                       .... ITERATE TILL R(T) BELOW LIMIT ....
        IF ((LG.EQ.1).OR.(ILFLG.EQ.2)) GO TO 660
C               ....  ACCELERATE FAILURE RATE ....
        RINC=RINC*IDBK
        ILFLG=2
  660   CONTINUE
        IFLG=2
        IF (RTPG.LE.RALMT) GO TO 650
        IFLG=3
        IF (I.LT.IULM*ICST) GO TO 400
        WRITE (ILP,595)
  595   FORMAT (T2,'****')
  600   CONTINUE
  650   CONTINUE
C
        RENH=0.0
        IF (RTPG.GT.RALMT) WRITE (ILP,595)
        IF (RTSED.GT.1.0) RENH=RMTTF/RTSED
C
C##############################################################################
C
C           THIS IS IT .....    SYSTEM MTBF
C
C##############################################################################
        WRITE (ILP,675) I,RTMX,RTPG,RMTTF,RENH
  675   FORMAT (T38,'=MEMORY MTBF=',4X,'=EF=',/,T2,I4,T7,F10.2,
```

```
FORTRAN IV      V02.04

          C T24,F8.5,T40,F12.2,T52,,F8.0,/,T2,8('--------'),/)
      C
0560          RZTOL=1000./RMTTF
0561          GO TO (676,677,678),JXFG
0562    676   RZTMP=RZTOL+RZSYS
0563          RTSYS=1000.0/RZTMP
0564          GO TO 679
0565    677   RZTMP=RZTOL+RZSYX+(EBD*RZTTX)
0566          RTSYS=1000.0/RZTMP
0567          GO TO 679
0568    678   RTSYS=RMTTF
0569          RZTMP=0.0
0570    679   CONTINUE
0571          IF (RTSED.GT.0.0) RENH=RTSYS/RTSED
0573          WRITE (ILP,680) IFLG,RTSYS,RENH,RZTMP
0574    680   FORMAT (T2,'FIN',1X,I3,T24,'=SYSTEM MTBF=',T40,F12.2,T52,F8.0,
          C T65,E12.5,/,1H1)
0575          GO TO 100
0576          END

FORTRAN IV      V02.1-1

0001          REAL FUNCTION DRTI*8(RZX,RTM,EL,RT)
0002          IMPLICIT REAL*8 (R)
0003          DATA R1/1.0D0/
      C#############################################################################
      C
      C           RS(T) FUNCTION
      C
      C#############################################################################
0004          DRTI=(R1/DEXP(RZX*RT))*(RTM**EL)
0005          RETURN
0006          END

FORTRAN IV      V02.1-1

0001          REAL FUNCTION DRTO*8(RZX,RXI,RXO,RN,EL,RT)
0002          IMPLICIT REAL*8 (R)
0003          DATA R1/1.0D0/
      C#############################################################################
      C
      C           BINOMIAL EQUATION FUNCTION
      C
      C#############################################################################
0004          EN=RN
0005          EN1=RN-R1
0006          RR=R1/DEXP(RZX*RT)
0007          RXN=RR**EN
0008          RQX=R1-RR
0009          RTRM1=RXN*(RXO**EL)
0010          RTRM2=RN*RQX*(RXI**EN1)
0011          DRTO=RTRM1+RTRM2
0012          RETURN
0013          END
```

```
FORTRAN IV      V02.1-1

0001          REAL FUNCTION DRTX*8(RZX,RXI,R10,R20,RN,EL,RT)
0002          IMPLICIT REAL*8 (R)
0003          DATA R1/1.0D0/,R2/2.0D0/
      C#################################################################################
      C
      C       BINOMIAL EQUATION FUNCTION FOR DOUBLE BIT CORRECTION
      C
      C#################################################################################
0004          EN=RN
0005          RN1=RN-R1
0006          EN1=RN1
0007          RN2=RN-R2
0008          EN2=RN2
0009          E2=R2
0010          RR=R1/DEXP(RZX*RT)
0011          RXN=RR**EN
0012          RQX=R1-RR
0013          RTRM1=RXN*(R20**EL)
0014          RTRM2=RN*RQX*(RR**EN1)*(R10**EL)
0015          RTRM3=(RN*RN1*(RQX**E2)*(RXI**EN2))*0.50
0016          DRTX=RTRM1+RTRM2+RTRM3
0017          RETURN
0018          END
```

```
FORTRAN IV      V02.1-1

0001          SUBROUTINE TEST(IMM)
      C
      C##############################################################################
      C
      C     EQUATIONS FOR:
      C                   NON ERROR CORRECTED SYSTEM
      C                   SINGLE BIT CORRECTED SYSTEM
      C                   DOUBLE BIT CORRECTED SYSTEM
      C                   USE OF PARTIALS IN A SYSTEM
      C
      C##############################################################################
0002          IMPLICIT REAL*8 (D,R,S,T,Z)
      C
0003          COMMON /ECC1/RXZ,RXS,RXR,RXC,RXF,RXE,RXH,RXT,RCNF,SZER,SXX
0004          COMMON /ECC2/RMSZ,RCSZ,RWD,BPG,RZD,RZDD,REC,RZER,R1,RTM,RTSF
0005          COMMON /ECC3/IM,ILLM,IULM,RSQ,JSFLG,EPGX,ISFLG,ST,R,RTH,R2
0006          COMMON /ECC4/ISW,RFF,IPM,RTTF,RZTTL,ICST,RALMT,IDBK,IQFG,IUCD
0007          COMMON /ECC5/ISFG,REC1,REC2,IEFLG,RZSYS,ILIM,IDFLG,RAVE
0008          COMMON /ECC6/ITIN,ITOUT,ILP
0009          COMMON /ECC7/RZZ,RZS,RZR,RZC,RZF,RZE,RZH,RZT,RZDX
0010          COMMON /ECC8/ECZ,ECS,ECR,ECC,ECF,ECE,ECH,ECT,ECX
0011          COMMON /ECC9/EW,EW1,EW2,RW,RW1,RW2,S,T,TSFT,THRD
0012          COMMON /ECCA/EPG,EBD,EPSZ,ECA,EPX,EPY,EPZ
0013          COMMON /ECCB/ZT,ZR,ZF,ZE,EZL
0014          COMMON /ECCC/I,IMN,RPRT,RTO,RTPG,RTX,RZDZ,RXX,RSPC1,RSPC2
      C
0015          GO TO (410,420,430),IMM
      C
0016     410  CONTINUE
      C          ######## SINGLE ERROR DETECT EQUATIONS ########
      C
0017          RZO=(1.0/DEXP(RZDD*S))**EW
0018          RXO=(1.0/DEXP(SXX*RZD*THRD))**EW
0019          RYO=(1.0/DEXP(RZTTL*THRD))**EW
      C
0020          RTO=(RZO*RYO*RXO)**EPG
0021          RXO=(1.0/DEXP(RZTTL*THRD))**EBD
0022          RTSYS=(1.0/DEXP(RZSYS*THRD))
0023          RTPG=((RTO*RXO)**EBD)*RTSYS
0024          GO TO 500
      C
0025     420  CONTINUE
      C          ######## SINGLE BIT ERROR CORRECTION EQUATIONS  #####
      C
0026          RZI=DRTI(RZZ,R1,ECZ,TSFT)
0027          RZO=DRTO(RZZ,RZI,R1,RW,ECZ,TSFT)
      C
0028          RSI=DRTI(RZS,RZI,ECS,THRD)
0029          RSO=DRTO(RZS,RSI,RZO,RW,ECS,THRD)
      C
0030          RFI=DRTI(RZF,RSI,ECF,THRD)
0031          RFO=DRTO(RZF,RFI,RSO,RW,ECF,THRD)
      C
0032          REI=DRTI(RZE,RFI,ECE,THRD)
```

```
FORTRAN IV      V02.1-1

0033            REO=DRTO(RZE,REI,RFO,RW,ECE,THRD)
      C
0034            RHI=DRTI(RZH,REI,ECH,THRD)
0035            RHO=DRTO(RZH,RHI,REO,RW,ECH,THRD)
      C
0036            RTI=DRTI(RZT,RHI,ECT,THRD)
0037            RTO=DRTO(RZT,RTI,RHO,RW,ECT,THRD)
      C
0038            RXI=DRTI(RZTTL,RTO,EPG,THRD)
0039            RXO=DRTO(RZTTL,RXI,RTO,RW,EPG,THRD)
      C
0040            RTSYS=1.0/DEXP(RZSYS*THRD)
      C
0041            GO TO (425,422,422),IQFG
      C             .... SPECIAL EQUATION FOR 2-D EFFECTS ....
0042     422    RQR=1.0-(1.0/DEXP(RXR*RZD*THRD*EW))
0043            RQC=1.0-(1.0/DEXP(RXC*RZD*THRD*EW))
0044            RQF=1.0-(1.0/DEXP(RXF*RZD*THRD*EW))
0045            RQE=1.0-(1.0/DEXP(RXE*RZD*THRD*EW))
0046            GO TO (425,424,423),IQFG
0047     423    RSPC1=((1.0-(RQR*RQC))*(1.0-(RQF*RQE)))**EPG
0048            GO TO 425
0049     424    SQX=RQR
0050            IF (RQC.LT.RQR) SQX=RQC
0052            SQZ=RQF
0053            IF (RQE.LT.RQF) SQZ=RQE
0055            RSPC1=((1.0-SQX)*(1.0-SQZ))**EPG
      C
0056     425    RTPG=((RXO*RSPC1)**EBD)*RTSYS
      C
0057            GO TO (500,428),IMN
      C             .... EQUATIONS FOR USE OF PARTIALS ....
0058     428    RPRTO=(1.0/DEXP(RW1*RZDX*THRD))**EPY
0059            RTPX=(RXO*RSPC1)**EPX
      C
0060            RTPG=((RTPX*RPRTO)**EBD)*RTSYS
0061            GO TO 500
      C
0062     430    CONTINUE
      C         ######## DOUBLE BIT ERROR CORRECTION EQUATIONS    ######
      C
0063            RZI=DRTI(RZZ,R1,ECZ,TSFT)
0064            RZO=DRTO(RZZ,RZI,R1,RW1,ECZ,TSFT)
0065            RZX=DRTX(RZZ,RZI,R1,R1,ECZ,TSFT)
      C
0066            RSI=DRTI(RZS,RZI,ECS,THRD)
0067            RSO=DRTO(RZS,RSI,RZO,RW1,ECS,THRD)
0068            RSX=DRTX(RZS,RSI,RZO,RZX,RW,ECS,THRD)
      C
0069            RFI=DRTI(RZF,RSI,ECF,THRD)
0070            RFO=DRTO(RZF,RFI,RSO,RW1,ECF,THRD)
0071            RFX=DRTX(RZF,RFI,RSO,RSX,RW,ECF,THRD)
      C
0072            REI=DRTI(RZE,RFI,ECE,THRD)
```

```
FORTRAN IV      V02.1-1

0073          REO=DRTO(RZE,REI,RFO,RW1,ECE,THRD)
0074          REX=DRTX(RZE,REI,RFO,RFX,RW,ECE,THRD)
      C
0075          RHI=DRTI(RZH,REI,ECH,THRD)
0076          RHO=DRTO(RZH,RHI,REO,RW1,ECH,THRD)
0077          RHX=DRTX(RZH,RHI,REO,REX,RW,ECH,THRD)
      C
0078          RTI=DRTI(RZT,RHI,ECT,THRD)
0079          RTO=DRTO(RZT,RTI,RHO,RW1,ECT,THRD)
0080          RTX=DRTX(RZT,RTI,RHO,RHX,RW,ECT,THRD)
      C
0081          RXI=DRTI(RZTTL,RTI,EPG,THRD)
0082          RXO=DRTO(RZTTL,RXI,RTO,RW1,EPG,THRD)
0083          RXX=DRTX(RZTTL,RXI,RTO,RTX,RW,EPG,THRD)
      C
0084          RTSYS=R1/DEXP(RZSYS*THRD)
      C
0085          GO TO (434,431,431),IQFG
      C             .... SPECIAL EQUATION FOR 2-D EFFECTS ....
0086    431   RTP=DRTI(RZR,R1,R1,THRD)
0087          RQR=1.0-DRTO(RZR,RTP,R1,RW1,R1,THRD)
0088          RTP=DRTI(RZC,R1,R1,THRD)
0089          RQC=1.0-DRTO(RZC,RTP,R1,RW1,R1,THRD)
0090          RTP=DRTI(RZF,R1,R1,THRD)
0091          RQF=1.0-DRTO(RZF,RTP,R1,RW1,R1,THRD)
0092          RTP=DRTI(RZE,R1,R1,THRD)
0093          RQE=1.0-DRTO(RZE,RTP,R1,RW1,R1,THRD)
0094          GO TO (434,433,432),IQFG
0095    432   RSPC2=((1.0-(RQR*RQC))*(1.0-(RQF*RQE)))**EPG
0096          GO TO 434
0097    433   SQX=RQR
0098          IF (RQC.LT.RQR) SQX=RQC
0100          SQZ=RQF
0101          IF (RQE.LT.RQF) SQZ=RQE
0103          RSPC2=((1.0-SQX)*(1.0-SQZ))**EPG
      C
0104    434   RTPG=((RXX*RSPC2)**EBD)*RTSYS
0105          GO TO (500,435),IMN
      C             .... EQUATIONS FOR USE OF PARTIALS ....
0106    435   CONTINUE
0107          GO TO (439,436,436),IQFG
0108    436   RQR=1.0-(1.0/DEXP(RXR*RZD*THRD*EW))
0109          RQC=1.0-(1.0/DEXP(RXC*RZD*THRD*EW))
0110          RQF=1.0-(1.0/DEXP(RXF*RZD*THRD*EW))
0111          RQE=1.0-(1.0/DEXP(RXE*RZD*THRD*EW))
0112          GO TO (439,438,437),IQFG
0113    437   RSPC1=((1.0-(RQR*RQC))*(1.0-(RQF*RQE)))**EPG
0114          GO TO 439
0115    438   RSPC1=((1.0-SQX)*(1.0-SQZ))**EPG
0116    439   RPRTO=(1.0/DEXP(RW2*RZDX*THRD))**EPZ
0117          RPRT1=(RTO*RSPC1)**EPY
0118          RPRT2=(RTX*RSPC2)**EPX
      C
0119          RTPG=((RPRTO*RPRT1*RPRT2)**EBD)*RTSYS
```

```
FORTRAN IV      V02.1-1

0120            GO TO 500
      C
0121    440     CONTINUE
      C         FUTURE TESTS HERE
0122    500     CONTINUE
0123            RETURN
0124            END
FORTRAN IV      V02.1-1

0001            SUBROUTINE HELP
0002            COMMON /ECC6/ITIN,ITOUT,ILP
0003            WRITE (ITOUT,20)
0004    20      FORMAT (/,T10,2('*********'),2X,'HELP TEXT  ',2('*********'),//
               C ,T2,'PRAMETER...RANGE.....COMMENTS.....',/,
               C T15,'NECC - NON ECC EVALUATION RUN.',/,T15,'SECC - SINGLE',
               C ' BIT ERROR CORRECTION RUN.',/,T15,'DECC - DOUBLE BIT ERROR'
               C ' ERROR CORRECTION RUN.')
0005            WRITE (ITOUT,25)
0006    25      FORMAT (T2,'INDEX:',T12,4('.........'),T50,'<DATA TYPE INTEGER>'
               C ,/,T18,'0 - SPECIFIES FULL-OUTPUT NORMAL RUN.',/,T17,'>0 -',
               C ' SPECIFIES SINGLE POINT CALCULATION OF R-FUNCTION @ T',/,
               C T22,'WHERE T = INDEX * TIME,   PURPOSE IS TO ASSIST USER',/,
               C T22,'DETERMINE BEST TIME INTERVAL FOR RUNS.')
0007            WRITE (ITOUT,30)
0008    30      FORMAT (T2,'TIME:',T12,4('.........'),T50,'<DATA TYPE FLOATING',
               C ' PT.>',/,T16,'>0.  - SPECIFIES INTERVAL OF TIME BETWEEN RUN-',
               C 'TIME',/,T22,'EVALUATION POINTS')
0009            WRITE (ITOUT,35)
0010    35      FORMAT (T2,'PAGE:',T12,4('.........'),T50,'<DATA TYPE INTEGER>',
               C /,T17,'>1 - NUMBER OF MEMORY ROWS PER BOARD.',/,
               C T2,'BOARDS:',T12,4('.........'),T50,'<DATA TYPE INTEGER>',/,
               C T17,'>1 - NUMBER OF BOARDS PER MEMORY SYSTEM.',/)
0011            WRITE (ITOUT,40)
0012    40      FORMAT (T20,' ** HIT <RETURN> TO CONTINUE **')
0013            READ (ITIN,45) IDUM
0014    45      FORMAT (A2)
0015            WRITE (ITOUT,50)
0016    50      FORMAT (//,T20,' # ADDITIONAL POINTER PARAMETERS #',//,
               C T2,'POINTER:',T12,4('.........'),T50,'<DATA TYPE LITERAL>',/,
               C T15,'LIST  -  LIST OUTPUT PARAMETERS.',/,
               C T15,'SIZE  -  MEMORY COMPONENT & SYSTEM PARAMETERS.',/,
               C T15,'RATE  -  COMPONENT & SYSTEM FAILURE RATES.',/,
               C T15,'DIST  -  COMPONENT FAILURE-TYPE DISTRIBUTION.',/,
               C T15,'COMM  -  OUTPUT RUN-TIME COMMENT LINE')
0017            WRITE (ITOUT,55)
0018    55      FORMAT (T15,'ABORT -  EXIT PROGRAM.',/,
               C T15,'DUMP  -  DISPLAY  < LIST, SIZE, RATE, DIST, COMM >',/,
               C T15,'PURGE -  PRINT REST OF RUN-TIME OUTPUT BUFFER',//,
               C T15,'FLAG  -  USE OF TTL & SYSTEM FAILURE RATES,  Q-FLAG',/,
               C T22,'SYS = TTL @ BOARD LEVEL, SYSTEM USED WITH MEMORY',/,
               C T22,'MPD = TTL N.U., SYSTEM RATE LISTED IN HEADER ONLY',/,
               C T28,'SOFT ERROR RATE SPECIAL MPD ALGORITHM - MEM CYCLES',/,
               C T22,'MSO = (TTL X BOARDS) + SYSTEM COMBINED WITH MEMORY',/,
               C T22,'Q1  =  ONE DIMENSIONAL ARRAY MODEL',/,
               C T22,'Q2  =  SAME AS Q1, PLUS SPECIAL TWO DIMENSIONAL FIX',/,
               C T10,5('*********'),///)
0019            RETURN
0020            END
      C         END OF PROGRAM
```

```
SYS /Q1              ECC PROBABILITY PROGRAM "INTEL-MPD/MC. "

MEMORY SYS:  SIZE->     32. OKB        WORD WIDTH->  16. + 6.  NO. PAGES->    1. X   1.
COMPONENT:  TOTAL->     16. +      6.   RAM SIZE->    16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0. 00000%/1K-HRS,   SYSTEM RATE ->  0. 00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->    49212. 60HRS,  MTBF. SYS ->          0. 00HRS
HARD ERRORS:       PARTIAL ->     0. CELLS/PG   RATE ->    0. 027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT   ->           0. HRS,  RATE ->    0. 100000% / 1000 HRS
ANALYSIS DATA:     PERIOD ->   100000. 00HRS,  AVE CELL FAILURE ->     16. 2
FAILURE TYPE RATIOS:  ----------------------------------------------------
=TYPE=       =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR  ->[ 78. 740%]/ 16384.  =  0. 61035E-05%,      0. 787  [  4. 87%]
 HARD ERRORS ->[ 21. 260%]                                         [ 95. 13%]
 SINGLE CELL -> 50. 0000% / 16384.  =  0. 82397E-06%,      0. 106    0. 69%     1.
 ROW OR COL  -> 43. 7000% /   128.  =  0. 92180E-04%,     11. 892   77. 36%   128.
 COLUMN/ROW  ->  6. 2000% /    64.  =  0. 26156E-04%,      3. 374   21. 95%     2.
 HALF CHIP   ->  0. 0000% /     2.  =  0. 00000E+00%,      0. 000    0. 00%    32.
 TOTAL CHIP  ->  0. 0000% /     1.  =  0. 00000E+00%,      0. 000    0. 00%     2.

PERIOD   PM@T:          R[T]. SECC          MTTF      ENHANCEMENT   % - R(T)
-----  <M- HRS>        =FUNCTION=          < HRS >     FACTOR        @T
    0     0. 00    1. 00000                    0.          0.      100. 0%
    1     0. 10    0. 99332             14922026.        303.       99. 3%
    2     0. 20    0. 97389              7584078.        154.       97. 4%
    3     0. 30    0. 94293              5149121.        105.       94. 3%
    4     0. 40    0. 90200              3939900.         80.       90. 2%
    5     0. 50    0. 85289              3221005.         65.       85. 3%
    6     0. 60    0. 79748              2747323.         56.       79. 7%
    7     0. 70    0. 73770              2413825.         49.       73. 8%
    8     0. 80    0. 67537              2168007.         44.       67. 5%
    9     0. 90    0. 61217              1980705.         40.       61. 2%
   10     1. 00    0. 54958              1834422.         37.       55. 0%
   11     1. 10    0. 48884              1718017.         35.       48. 9%
   12     1. 20    0. 43095              1624061.         33.       43. 1%
   13     1. 30    0. 37666              1547397.         31.       37. 7%
   14     1. 40    0. 32650              1484334.         30.     < 32. 6%>
   15     1. 50    0. 28075              1432149.         29.       28. 1%
   16     1. 60    0. 23956              1388787.         28.       24. 0%
   17     1. 70    0. 20290              1352665.         27.       20. 3%
   18     1. 80    0. 17062              1322533.         27.       17. 1%
   19     1. 90    0. 14248              1297395.         26.       14. 2%
   20     2. 00    0. 11819              1276437.         26.       11. 8%
   21     2. 10    0. 09741              1258993.         26.        9. 7%
   22     2. 20    0. 07979              1244505.         25.        8. 0%
   23     2. 30    0. 06496              1232508.         25.        6. 5%
   24     2. 40    0. 05258              1222606.         25.        5. 3%
   25     2. 50    0. 04232              1214463.         25.        4. 2%
   26     2. 60    0. 03388              1207796.         25.        3. 4%
   27     2. 70    0. 02698              1202358.         24.        2. 7%
   28     2. 80    0. 02137              1197945.         24.        2. 1%
   29     2. 90    0. 01685              1194379.         24.        1. 7%
   30     3. 00    0. 01322              1191512.         24.        1. 3%
****
                                    =MEMORY MTBF=    =EF=
   31     3. 00        0. 01322        1175755. 35      24.
-----------------------------------------------------------------------

FIN   1                  =SYSTEM MTBF=     1175755. 35      24.       0. 85052E-03
```

```
SYS /Q1              ECC PROBABILITY PROGRAM "INTEL-MPD/MC. "

MEMORY SYS:  SIZE->      4. 1MB        WORD WIDTH->  16. + 6.  NO. PAGES->    1. X128.
COMPONENT:  TOTAL->  2048. +    768.    RAM SIZE->   16384.  COL. SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0. 00000%/1K-HRS,   SYSTEM RATE ->  0. 00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->      384. 47HRS,  MTBF. SYS ->          0. 00HRS
HARD ERRORS:       PARTIAL ->     0. CELLS/PG   RATE ->    0. 027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT   ->          0. HRS,  RATE ->    0. 100000% / 1000 HRS
ANALYSIS DATA:     PERIOD ->     8000. 00HRS,  AVE CELL FAILURE ->     16. 2
FAILURE TYPE RATIOS:  ------------------------------------------------------
=TYPE=        =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR   ->[ 78. 740%]/ 16384.  =   0. 61035E-05%,      0. 787  [  4. 87%]
 HARD ERRORS ->[ 21. 260%]                                           [ 95. 13%]
 SINGLE CELL  -> 50. 0000% / 16384.  =   0. 82397E-06%,      0. 106    0. 69%    1.
 ROW OR COL   -> 43. 7000% /   128.  =   0. 92180E-04%,     11. 892   77. 36%  128.
 COLUMN/ROW   ->  6. 2000% /    64.  =   0. 26156E-04%,      3. 374   21. 95%    2.
 HALF CHIP    ->  0. 0000% /     2.  =   0. 00000E+00%,      0. 000    0. 00%   32.
 TOTAL CHIP   ->  0. 0000% /     1.  =   0. 00000E+00%,      0. 000    0. 00%    2.

PERIOD    PM@T:          R[T]. SECC          MTTF       ENHANCEMENT   % - R(T)
------ < - HRS>         =FUNCTION=        < HRS >       FACTOR          @T
     0        0.          1. 00000             0.           0.       100. 0%
     1     8000.          0. 99446       1439611.        3744.        99. 4%
     2    16000.          0. 97804        722580.        1879.        97. 8%
     3    24000.          0. 95132        484470.        1260.        95. 1%
     4    32000.          0. 91518        366101.         952.        91. 5%
     5    40000.          0. 87080        295638.         769.        87. 1%
     6    48000.          0. 81955        249140.         648.        82. 0%
     7    56000.          0. 76294        216349.         563.        76. 3%
     8    64000.          0. 70256        192135.         500.        70. 3%
     9    72000.          0. 63998        173652.         452.        64. 0%
    10    80000.          0. 57670        159190.         414.        57. 7%
    11    88000.          0. 51411        147663.         384.        51. 4%
    12    96000.          0. 45341        138346.         360.        45. 3%
    13   104000.          0. 39562        130737.         340.        39. 6%
    14   112000.          0. 34153        124475.         324.      < 34. 2%>
    15   120000.          0. 29171        119297.         310.        29. 2%
    16   128000.          0. 24653        115001.         299.        24. 7%
    17   136000.          0. 20616        111433.         290.        20. 6%
    18   144000.          0. 17059        108471.         282.        17. 1%
    19   152000.          0. 13968        106017.         276.        14. 0%
    20   160000.          0. 11318        103990.         270.        11. 3%
    21   168000.          0. 09076        102322.         266.         9. 1%
    22   176000.          0. 07202        100958.         263.         7. 2%
    23   184000.          0. 05656         99849.         260.         5. 7%
    24   192000.          0. 04397         98954.         257.         4. 4%
    25   200000.          0. 03382         98237.         256.         3. 4%
    26   208000.          0. 02575         97668.         254.         2. 6%
    27   216000.          0. 01941         97220.         253.         1. 9%
    28   224000.          0. 01448         96872.         252.         1. 4%
    29   232000.          0. 01069         96603.         251.         1. 1%
    30   240000.          0. 00782         96397.         251.         0. 8%
                                      =MEMORY MTBF=      =EF=
    31   240000. 00       0. 00782         95643. 55      249.
-------------------------------------------------------------------------------

FIN    1                  =SYSTEM MTBF=    95643. 55      249.       0. 10455E-01
```

```
SYS /Q1              ECC PROBABILITY PROGRAM "INTEL-MPD/MC. "

MEMORY SYS:  SIZE->     64. OKB        WORD WIDTH->  32. + 7.  NO. PAGES->    1. X   1.
COMPONENT:  TOTAL->     32. +      7.    RAM SIZE->   16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0. 00000%/1K-HRS,  SYSTEM RATE ->  0. 00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->    24606. 30HRS,  MTBF. SYS ->          0. 00HRS
HARD ERRORS:       PARTIAL ->     0. CELLS/PG   RATE ->    0. 027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT   ->           0. HRS,  RATE ->    0. 100000% / 1000 HRS
ANALYSIS DATA:     PERIOD ->    66000. 00HRS,  AVE CELL FAILURE ->     16. 2
FAILURE TYPE RATIOS:  ------------------------------------------------
=TYPE=       =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR  ->[ 78. 740%]/ 16384.  =  0. 61035E-05%,       0. 787  [   4. 87%]
 HARD ERRORS ->[ 21. 260%]                                          [  95. 13%]
 SINGLE CELL -> 50. 0000% / 16384.  =  0. 82397E-06%,       0. 106     0. 69%    1.
 ROW OR COL  -> 43. 7000% /   128.  =  0. 92180E-04%,      11. 892    77. 36%  128.
 COLUMN/ROW  ->  6. 2000% /    64.  =  0. 26156E-04%,       3. 374    21. 95%    2.
 HALF CHIP   ->  0. 0000% /     2.  =  0. 00000E+00%,       0. 000     0. 00%   32.
 TOTAL CHIP  ->  0. 0000% /     1.  =  0. 00000E+00%,       0. 000     0. 00%    2.

PERIOD    PM@T:          R[T]. SECC          MTTF      ENHANCEMENT   % - R(T)
------ < - HRS>         =FUNCTION=        < HRS >      . FACTOR  .     @T
     0        0.          1. 00000             0.            0.      100. 0%
     1    66000.          0. 99070       7066011.          287.       99. 1%
     2   132000.          0. 96388       3604737.          146.       96. 4%
     3   198000.          0. 92173       2458257.          100.       92. 2%
     4   264000.          0. 86699       1890484.           77.       86. 7%
     5   330000.          0. 80276       1554231.           63.       80. 3%
     6   396000.          0. 73219       1333790.           54.       73. 2%
     7   462000.          0. 65828       1179584.           48.       65. 8%
     8   528000.          0. 58374       1066829.           43.       58. 4%
     9   594000.          0. 51088        981755.           40.       51. 1%
    10   660000.          0. 44151        916096.           37.       44. 2%
    11   726000.          0. 37700        864581.           35.       37. 7%
    12   792000.          0. 31821        823686.           33.     < 31. 8%>
    13   858000.          0. 26564        790958.           32.       26. 6%
    14   924000.          0. 21942        764629.           31.       21. 9%
    15   990000.          0. 17941        743388.           30.       17. 9%
    16  1056000.          0. 14528        726238.           30.       14. 5%
    17  1122000.          0. 11655        712400.           29.       11. 7%
    18  1188000.          0. 09267        701259.           28.        9. 3%
    19  1254000.          0. 07305        692319.           28.        7. 3%
    20  1320000.          0. 05712        685174.           28.        5. 7%
    21  1386000.          0. 04431        679492.           28.        4. 4%
    22  1452000.          0. 03411        674998.           27.        3. 4%
    23  1518000.          0. 02607        671465.           27.        2. 6%
    24  1584000.          0. 01979        668705.           27.        2. 0%
    25  1650000.          0. 01492        666562.           27.        1. 5%
    26  1716000.          0. 01118        664910.           27.        1. 1%
    27  1782000.          0. 00832        663644.           27.        0. 8%
                                      =MEMORY MTBF=      =EF=
    28  1782000. 00       0. 00832        658122. 51        27.
-------------------------------------------------------------------------------

FIN    1                 =SYSTEM MTBF=    658122. 51        27.      0. 15195E-02
```

```
SYS /Q1              ECC PROBABILITY PROGRAM "INTEL-MPD/MC. "

MEMORY SYS:  SIZE->      8. 2MB       WORD WIDTH->  32. + 7.  NO. PAGES->    1. X128.
COMPONENT:  TOTAL->   4096. +    896.    RAM SIZE->    16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0. 00000%/1K-HRS,   SYSTEM RATE ->  0. 00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->      192. 24HRS,  MTBF. SYS ->          0. 00HRS
HARD ERRORS:       PARTIAL ->     0. CELLS/PG   RATE ->    0. 027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT   ->           0. HRS,  RATE ->    0. 100000% / 1000 HRS
ANALYSIS DATA:      PERIOD ->     5000. 00HRS,  AVE CELL FAILURE ->     16. 2
FAILURE TYPE RATIOS:  ------------------------------------------------------
=TYPE=        =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR   ->[ 78. 740%]/ 16384.  =  0. 61035E-05%,       0. 787  [  4. 87%]
 HARD ERRORS  ->[ 21. 260%]                                          [ 95. 13%]
 SINGLE CELL  -> 50. 0000% / 16384.  =  0. 82397E-06%,       0. 106    0. 69%     1.
 ROW OR COL   -> 43. 7000% /   128.  =  0. 92180E-04%,      11. 892   77. 36%   128.
 COLUMN/ROW   ->  6. 2000% /    64.  =  0. 26156E-04%,       3. 374   21. 95%     2.
 HALF CHIP    ->  0. 0000% /     2.  =  0. 00000E+00%,       0. 000    0. 00%    32.
 TOTAL CHIP   ->  0. 0000% /     1.  =  0. 00000E+00%,       0. 000    0. 00%     2.

PERIOD    PM@T:          R[T]. SECC          MTTF      ENHANCEMENT  % - R(T)
------ < - HRS>         =FUNCTION=        < HRS >      FACTOR         @T
    0        0.          1. 00000             0.          0.        100. 0%
    1     5000.          0. 99306        718156.        3736.         99. 3%
    2    10000.          0. 97257        360766.        1877.         97. 3%
    3    15000.          0. 93940        242201.        1260.         93. 9%
    4    20000.          0. 89494        183350.         954.         89. 5%
    5    25000.          0. 84095        148393.         772.         84. 1%
    6    30000.          0. 77946        125392.         652.         77. 9%
    7    35000.          0. 71269        109232.         568.         71. 3%
    8    40000.          0. 64283         97356.         506.         64. 3%
    9    45000.          0. 57202         88344.         460.         57. 2%
   10    50000.          0. 50219         81346.         423.         50. 2%
   11    55000.          0. 43499         75818.         394.         43. 5%
   12    60000.          0. 37176         71398.         371.         37. 2%
   13    65000.          0. 31351         67835.         353        < 31. 4%>
   14    70000.          0. 26090         64949.         338.         26. 1%
   15    75000.          0. 21425         62605.         326.         21. 4%
   16    80000.          0. 17364         60702.         316.         17. 4%
   17    85000.          0. 13888         59159.         308.         13. 9%
   18    90000.          0. 10963         57913.         301.         11. 0%
   19    95000.          0. 08542         56913.         296.          8. 5%
   20   100000.          0. 06569         56116.         292.          6. 6%
   21   105000.          0. 04987         55486.         289.          5. 0%
   22   110000.          0. 03737         54992.         286.          3. 7%
   23   115000.          0. 02765         54609.         284.          2. 8%
   24   120000.          0. 02019         54316.         283.          2. 0%
   25   125000.          0. 01456         54093.         281.          1. 5%
   26   130000.          0. 01037         53927.         281.          1. 0%
   27   135000.          0. 00729         53804.         280.          0. 7%
                                     =MEMORY MTBF=     =EF=
   28   135000. 00       0. 00729         53412. 22      278.
-------------------------------------------------------------------------

FIN    1                =SYSTEM MTBF=     53412. 22      278.       0. 18722E-01
```

```
SYS /01              ECC PROBABILITY PROGRAM "INTEL-MPD/MC. "

MEMORY SYS:  SIZE->    128. OKB        WORD WIDTH->  64. + 8.  NO. PAGES->    1. X   1.
COMPONENT:  TOTAL->     64. +      8.    RAM SIZE->    16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0. 00000%/1K-HRS,   SYSTEM RATE ->  0. 00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->    12303. 15HRS,  MTBF. SYS ->          0. 00HRS
HARD ERRORS:       PARTIAL ->     0. CELLS/PG   RATE ->    0. 027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT   ->           0. HRS,  RATE ->    0. 100000% / 1000 HRS
ANALYSIS DATA:     PERIOD ->    33000. 00HRS,  AVE CELL FAILURE ->      16. 2
FAILURE TYPE RATIOS:  ------------------------------------------------------
=TYPE=        =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR  ->[ 78. 740%]/ 16384.  =   0. 61035E-05%,        0. 787  [  4. 87%]
 HARD ERRORS ->[ 21. 260%]                                            [ 95. 13%]
 SINGLE CELL -> 50. 0000% / 16384.  =   0. 82397E-06%,        0. 106     0. 69%     1.
 ROW OR COL  -> 43. 7000% /   128.  =   0. 92180E-04%,       11. 892    77. 36%   128.
 COLUMN/ROW  ->  6. 2000% /    64.  =   0. 26156E-04%,        3. 374    21. 95%     2.
 HALF CHIP   ->  0. 0000% /     2.  =   0. 00000E+00%,        0. 000     0. 00%    32.
 TOTAL CHIP  ->  0. 0000% /     1.  =   0. 00000E+00%,        0. 000     0. 00%     2.

PERIOD    PM@T:          R[T]. SECC          MTTF      ENHANCEMENT   % - R(T)
-----  < - HRS>         =FUNCTION=         < HRS >      FACTOR          @T
    0        0.          1. 00000              0.          0.        100. 0%
    1    33000.          0. 99197        4092667.        333.         99. 2%
    2    66000.          0. 96871        2084463.        169.         96. 9%
    3    99000.          0. 93192        1418701.        115.         93. 2%
    4   132000.          0. 88376        1088549.         88.         88. 4%
    5   165000.          0. 82663         892655.         73.         82. 7%
    6   198000.          0. 76308         763911.         62.         76. 3%
    7   231000.          0. 69556         673564.         55.         69. 6%
    8   264000.          0. 62639         607238.         49.         62. 6%
    9   297000.          0. 55758         556949.         45.         55. 8%
   10   330000.          0. 49083         517906.         42.         49. 1%
   11   363000.          0. 42747         487054.         40.         42. 7%
   12   396000.          0. 36848         462357.         38.       < 36. 8%>
   13   429000.          0. 31451         442398.         36.         31. 5%
   14   462000.          0. 26592         426159.         35.         26. 6%
   15   495000.          0. 22280         412889.         34.         22. 3%
   16   528000.          0. 18504         402018.         33.         18. 5%
   17   561000.          0. 15240         393104.         32.         15. 2%
   18   594000.          0. 12450         385797.         31.         12. 5%
   19   627000.          0. 10093         379818.         31.         10. 1%
   20   660000.          0. 08120         374936.         30.          8. 1%
   21   693000.          0. 06487         370964.         30.          6. 5%
   22   726000.          0. 05146         367744.         30.          5. 1%
   23   759000.          0. 04056         365147.         30.          4. 1%
   24   792000.          0. 03176         363061.         30.          3. 2%
   25   825000.          0. 02472         361395.         29.          2. 5%
   26   858000.          0. 01912         360071.         29.          1. 9%
   27   891000.          0. 01471         359025.         29.          1. 5%
   28   924000.          0. 01125         358203.         29.          1. 1%
   29   957000.          0. 00856         357561.         29.          0. 9%
                                      =MEMORY MTBF=     =EF=
   30   957000. 00       0. 00856         354499. 33      29.
-------------------------------------------------------------------------

FIN    1                =SYSTEM MTBF=     354499. 33      29.       0. 28209E-02
```

```
SYS /Q1            ECC PROBABILITY PROGRAM "INTEL-MPD/MC."

MEMORY SYS:  SIZE->     16.4MB       WORD WIDTH->  64. + 8.  NO. PAGES->    1. X128.
COMPONENT:  TOTAL->   8192. +  1024.    RAM SIZE->   16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0.00000%/1K-HRS,   SYSTEM RATE ->  0.00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->       96.12HRS,  MTBF. SYS ->         0.00HRS
HARD ERRORS:       PARTIAL ->    0. CELLS/PG   RATE ->    0.027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT   ->          0. HRS,  RATE ->    0.100000% / 1000 HRS
ANALYSIS DATA:      PERIOD ->    2500.00HRS,  AVE CELL FAILURE ->     16.2
FAILURE TYPE RATIOS:  ------------------------------------------------------
=TYPE=       =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR  ->[ 78.740%]/ 16384.  =   0.61035E-05%,        0.787  [  4.87%]
 HARD ERRORS ->[ 21.260%]                                          [ 95.13%]
 SINGLE CELL -> 50.0000% / 16384.  =   0.82397E-06%,        0.106    0.69%     1.
 ROW OR COL  -> 43.7000% /   128.  =   0.92180E-04%,       11.892   77.36%   128.
 COLUMN/ROW  ->  6.2000% /    64.  =   0.26156E-04%,        3.374   21.95%     2.
 HALF CHIP   ->  0.0000% /     2.  =   0.00000E+00%,        0.000    0.00%    32.
 TOTAL CHIP  ->  0.0000% /     1.  =   0.00000E+00%,        0.000    0.00%     2.

PERIOD    PM@T:          R(T). SECC          MTTF     ENHANCEMENT   % - R(T)
------ < - HRS>         =FUNCTION=        < HRS >    . FACTOR  .     @T   .
     0       .0. M-        1.00000              0.          0.       100.0%
     1     2500.           0.99401         416361.      4332.        99.4%
     2     5000.           0.97629         209042.      2175.        97.6%
     3     7500.           0.94751         140217.      1459.        94.8%
     4    10000.           0.90869         106020.      1103.        90.9%
     5    12500.           0.86119          85676.       891.        86.1%
     6    15000.           0.80658          72264.       752.        80.7%
     7    17500.           0.74658          62816.       654.        74.7%
     8    20000.           0.68298          55851.       581.        68.3%
     9    22500.           0.61752          50543.       526.        61.8%
    10    25000.           0.55187          46400.       483.        55.2%
    11    27500.           0.48749          43106.       448.        48.7%
    12    30000.           0.42566          40453.       421.        42.6%
    13    32500.           0.36741          38295.       398.      < 36.7%>
    14    35000.           0.31350          36528.       380.        31.3%
    15    37500.           0.26445          35074.       365.        26.4%
    16    40000.           0.22053          33876.       352.        22.1%
    17    42500.           0.18183          32888.       342.        18.2%
    18    45000.           0.14822          32075.       334.        14.8%
    19    47500.           0.11947          31407.       327.        11.9%
    20    50000.           0.09521          30862.       321.         9.5%
    21    52500.           0.07503          30419.       316.         7.5%
    22    55000.           0.05847          30061.       313.         5.8%
    23    57500.           0.04506          29774.       310.         4.5%
    24    60000.           0.03434          29546.       307.         3.4%
    25    62500.           0.02588          29367.       306.         2.6%
    26    65000.           0.01929          29227.       304.         1.9%
    27    67500.           0.01422          29120.       303.         1.4%
    28    70000.           0.01037          29037.       302.         1.0%
    29    72500.           0.00748          28975.       301.         0.7%
                                        =MEMORY MTBF=    =EF=
    30    72500.00         0.00748          28758.48     299.
-------------------------------------------------------------------------

FIN    1                =SYSTEM MTBF=       28758.48     299.       0.34772E-01
```

intel

ARTICLE REPRINT

AR-189

November 1981

Keep memory design simple yet cull single-bit errors

M. Bazes
L. Farrell
B. May
M. Mebel
Intel Corporation

AFN-02124A
ORDER NUMBER: 210287-001

A new error-correction chip with dual-bus architecture interfaces easily with dynamic RAMs. Memory-system reliability soars and the additional parts count is relatively modest.

Keep memory design simple yet cull single-bit errors

In memory-system design, the demand for greater reliability is reflected by an increasing interest in error-detection and correction circuitry. Several semiconductor manufacturers have recently introduced error-detection and correction chips. They share a common architecture that features a multiplexed data bus. But the Intel 8206 error-detection and correction unit (EDCU) is different: This LSI device, fabricated in HMOS II, allows error correction to be added to memory systems with minimal overhead.

A single 8206 handles 8 or 16-bit data widths, and up to five 8206's can be cascaded to handle all multiples of eight bits (up to 80 bits). The 8206 corrects single-bit errors in a maximum of 65 ns for 16-bit systems and typically replaces 20 to 40 ICs, depending upon the number of features in the system.

Common error detection circuits simply recognize that data has a parity error. Correction circuits use the Hamming code as an extension of parity to detect and give the position of the error, allowing it to be corrected.

Single-bit correction and multiple-bit detection is the typical implementation, reflecting the tradeoff between the probability of errors in a system and the cost of additional memory. For a 16-bit system, single-bit error correction and double-bit error detection is implemented by using 6 additional check bits, for an overhead of 37% (Table 1). Adding single-bit error correction to a system improves system reliability by at least a factor of 24 (Table 2).

M. Bazes, Design Engineer
L. Farrell, Marketing Manager
B. May, Applications Engineer
M. Mebel, Design Engineer
Intel Corp., 3065 Bowers Ave.
Santa Clara, CA 95051

Error correction is used extensively in mainframe and minicomputer design where memory sizes of several megabytes are common. Here the probability of error is directly related to the error rate of the individual RAMs and the number of RAMs in the system. As the number of RAMs increases, so does the system error rate.

With today's microprocessors, like the Intel eight-bit iAPX 88 and 16-bit iAPX 86 (each can directly address 1 Mbyte), typical RAM memory sizes are 100 kbytes and climbing. As a result, microprocessor system designers are looking to add error correction as simply as possible.

New bus architecture

The 8206 is the first 16-bit EDCU to use separate input and output data buses, a feature that simplifies system design, saves board space, and reduces parts count. The new architecture is made possible by packaging the 8206 in a JEDEC type A 68-pin leadless chip carrier. Figure 1 shows the 8206's functional blocks.

During read cycles, data and check bits enter via the data input (DI) and check-bit input (CBI) pins, where they are optionally latched by the STB input. The data then take two parallel paths. The first path is to the data-output (DO/WDI) pins, where the uncorrected data are available 32 ns later. The second path is to the check-

September 30, 1981

bit generator, where check bits generated from the data are compared with the check bits read from the memory.

The result of the comparison is the *syndrome*, a 5-to-8-bit value identifying which bit (if any) was in error. The syndrome is then decoded to a 1-of-16 bit strobe which is used to "flip" the bit in error (assuming the $\overline{CRCT}$ input is active). Syndrome decoding also tells the 8206 whether to assert the error flags. The 16 data output pins are enabled on a byte basis by the $\overline{BM}$ inputs.

For write cycles, data enter the write data input (DO/WDI) pins and goes to the check-bit generator. The check bits are then written to the check-bit memory by the check-bit output (SYO/CBO/PPO) pins. These pins also output the syndrome bits during read or read-modify-write cycles.

Note that only the 8206's R/$\overline{W}$ pin is typically used for control during a memory cycle. This pin informs the 8206 whether the cycle is a read (generate new check bits and compare to those from memory) or a write (generate new check bits only). During a read-modify-write cycle, a falling edge of R/$\overline{W}$ tells the 8206 to latch the syndrome bits internally and output check bits to be written back into memory. The strobe input (STB) may optionally be used to latch data and check bits internally.

The 8206's dual-bus architecture saves the additional control lines and the sequencing logic required

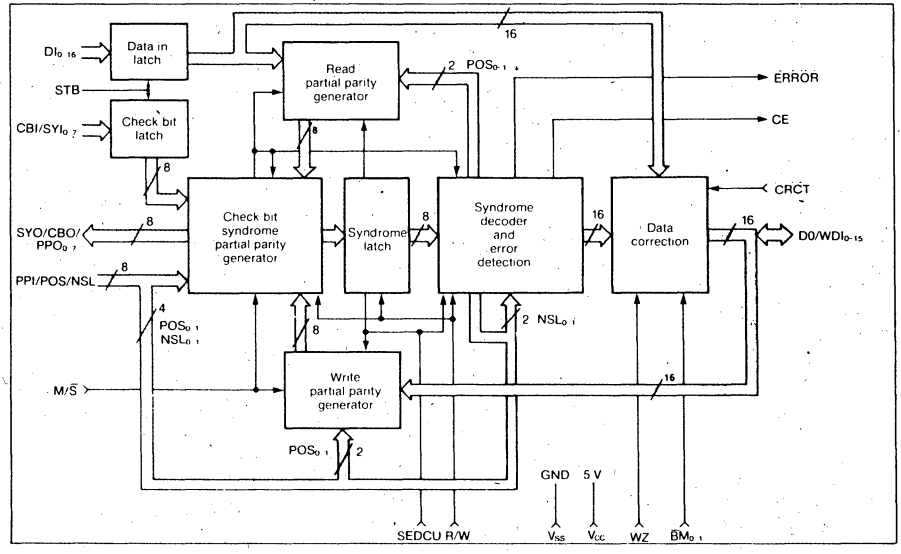
Table 1. Check bits required for single-bit correction, multiple-bit detection.

Data word bits	Check bits	Overhead % (# check bits/# data bits)
8	5	62
16	6	37
32	7	22
64	8	12
80	8	10

Table 2. Single-bit error correction increases memory reliability a minimum of 24 times.

*Memory size	MTBF (no error correction)	MTBF (single-bit error correction)	MTBF improvement ratios
32 kbytes	5.6 Years	133.6 Years	24
64 "	2.7 "	75.1 "	28
128 "	1.4 "	40.5 "	29
5 Mbytes	16 Days	10.8 "	246
8 "	8 "	6.1 "	278
16 "	4 "	3.3 "	301

*Based on a 16 kbit dynamic RAM with a failure rate of 0.127% every 1000 hours.
Note: MTBF, though related to memory size, also depends on memory organization (e.g. word width, number of pages) that is not detailed in this table.



1. The 8206's two 16-bit data buses, one for data from the RAM (DI_{0-15}) and one for data to the system bus (DO_{0-15}), minimize the external control logic required.

by single-bus EDCUs. The principal advantages of dual-bus architecture can be illustrated by looking at the three types of memory cycles: reads, writes, and read-modify-writes.

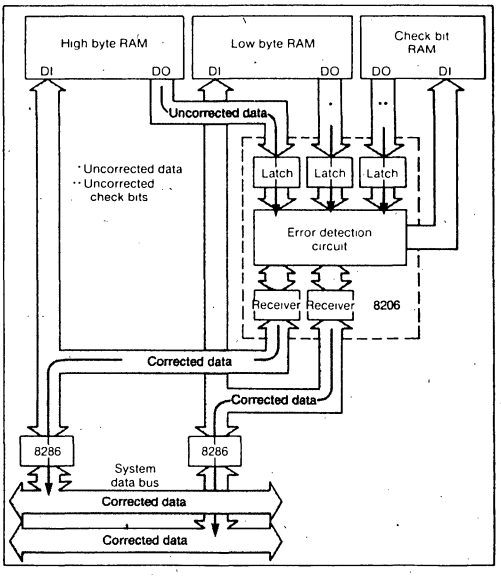
In a read cycle (Fig. 2), data and check bits are received from the RAM outputs by the DI and CBI pins. New check bits are generated from the data bits and compared to the check bits read from the RAM. An error in either the data or the check bits read from memory means the generated check bits will not match the read check bits. If an error is detected, the $\overline{\text{ERROR}}$ flag is activated and the correctable error (CE) flag tells the system if the error is (or is not) correctable.

With the $\overline{\text{BM}}$ inputs high, the corrected word appears at the DO pins (if the error was correctable), or the unmodified word appears (if the error was uncorrectable). Note that for this correction cycle there is no control or timing logic required. The 8206's dual buses isolate the RAM outputs from the EDCU outputs. Special transceivers that prevent contention between the uncorrected RAM data and corrected EDCU data are not needed.

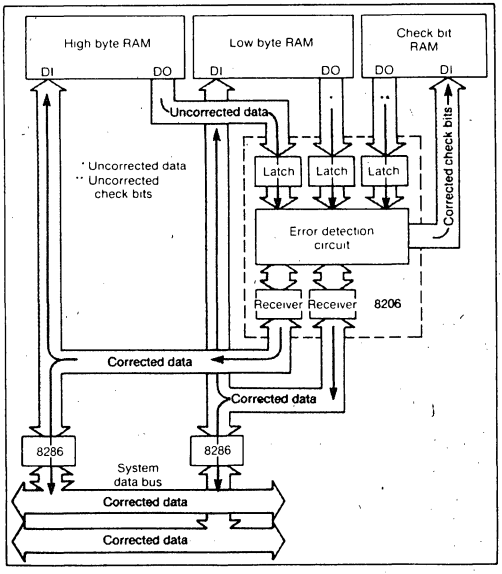
A syndrome word, five to eight bits in length and containing all necessary information about the existence and location of an error, is provided at the syndrome output (SYO/CBO/PPO) pins. Error logging is accomplished by latching the syndrome and the memory address of the word in error. The syndrome decoding of Table 3 can be used as a table lookup by the CPU.

If an error is detected during a read, the read cycle is extended to a read-modify-write cycle where the corrected data is rewritten to the same location. This offers several advantages:

- Since soft errors are random, independent processes, the longer a soft error is allowed to remain in memory, the greater the probability that a second soft error will occur in the memory word, resulting in an uncorrectable double-bit error. By writing the correct data back to RAM, the mean "lifetime" of soft errors is reduced, greatly reducing the chance of double-bit errors, and increasing reliability.
- "Error scrubbing" (going through the entire memory and correcting any soft errors) may be done as a background software task. For instance, the 8086 microprocessor's load string (LODS) instruction can consecutively read all addresses in RAM. Any soft errors will be corrected. Scrubbing further increases system reliability.
- Error logging may be used to detect hard errors. (A soft error is seen once when the affected word is read and is then corrected, while a hard error is seen again and again.) An error logger shows a consistent pattern if a hard error is present in a particular word. A system may be configured to generate an interrupt when the 8206 detects an error.



2. The 8206 requires no control logic or timing inputs to perform read-with-correction cycles.



3. The 8206 can correct both data bits and check bits.

This last advantage allows the operating system to re-read the address where the error occurred. If the same error re-occurs, it is assumed to be a hard error, and while the system can continue to function, maintenance is indicated. The operating system may mark that page of memory as "bad" until its PC card

has been serviced. Alternatively, the memory system may reconfigure itself and map the bit where the hard error occurred to a spare dynamic RAM whenever the affected memory page is accessed.

When a correctable error occurs during a read cycle (Fig. 3), the system's dynamic RAM controller (or CPU) examines the 8206 ERROR and CE outputs to determine if a correctable error occurred. If it did, the dynamic RAM controller (or CPU) forces R/$\overline{W}$ low, telling the 8206 to latch the generated syndrome and drive the corrected check bits onto the SYO/CBO/PPO outputs. The corrected data is already available on the DO/WDI pins. The dynamic RAM controller then writes the corrected data and check bits into memory. Once again the 8206's dual buses allow this cycle to be implemented without special bus transceivers.

The 8206 may be used to perform read-modify-writes in one or two RAM cycles. If it is done in two cycles, the 8206 latches are used to hold the data and check bits from the read cycle to be used in the immediately following write cycle.

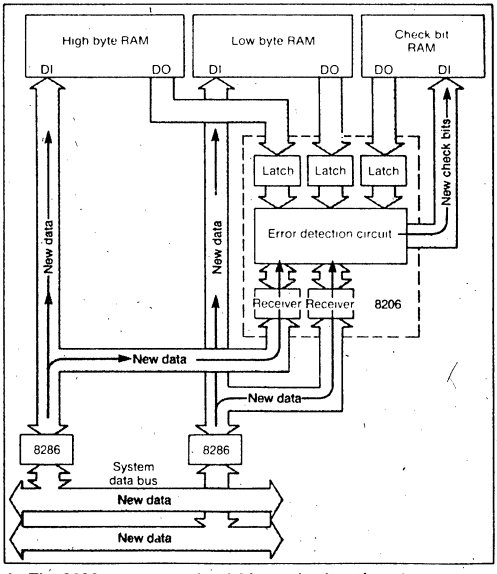
Write cycle corrections

For a full-word write (Fig. 4) where an entire word is written to memory, data are written directly to the RAM. This same data enter the 8206 through the DO/WDI pins where five to eight check bits are generated. The check bits are then sent to the RAM through the SYO/CBO/PPO pins for storage along with the data word.

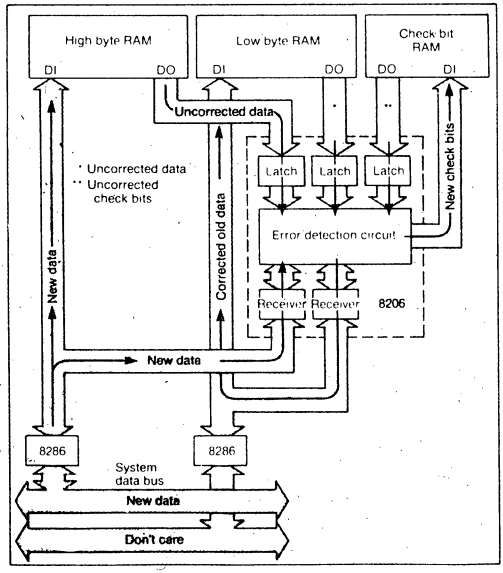
A byte write (Fig. 5) is implemented as a read-modify-write cycle. Since the Hamming code works only on entire words, to write one byte of the word, it is necessary to read the entire word to be modified, perform error correction, merge the new byte into the old word inside the 8206, generate check bits for the new word, and write the whole word plus check bits into RAM.

Error correction on the old word is important. Suppose a bit error occurs in the half of the old word that was not changed. This old byte would be combined with the new byte, and check bits would be generated for the whole word, including the bit in error. The bit error now becomes "legitimate"; no error will be detected when this word is read, and the system may crash. Obviously, it is important to eliminate this bit error before new check bits are generated.

The 8206 may alternatively be used in a "check-only" mode with the correct ($\overline{CRCT}$) pin left inactive. With the correction facility turned off, the delay of generating and decoding the syndromes is avoided, and the propagation delay from memory outputs to 8206 outputs is significantly shortened. In the event of an error, the 8206 activates the $\overline{ERROR}$ flag to the CPU or dynamic RAM controller, which can then perform one of several optins: lengthen the current cycle for correction, restart the instruction, perform a diagnostic routine, or activate the $\overline{CRCT}$ input to enable error correction. Even with the $\overline{CRCT}$ pin



4. The 8206 generates check bits and writes them to memory.



5. The "new data" byte is supplied by the CPU, while the 8206 supplies the corrected old byte. The 8206 also generates new check bits.

inactive, the 8206 generates and decodes the syndrome bits, so that data may be corrected rapidly if the $\overline{CRCT}$ is activated.

Multiple 8206 systems

A single 8206 handles eight or 16 bits of data and five or six check bits, respectively. Up to five 8206's can be cascaded for 80-bit data words with eight check bits. When cascaded, one 8206 operates as a master, and all others work as slaves (Fig. 6).

As an example, during a read cycle in a 32-bit system with one master and one slave, the slave calculates "partial parity" on its portion of the word and presents it to the master through the partial-parity output (SYO/CBO/PPO) pins. The master receives the partial parity at its partial-parity input (PPI/POS/NSL) pins and combines the partial parity from the slave with the parity it calculated from its own portion of the word to generate the syndrome. The syndrome is then returned from the master to the slave for error correction.

The 8206 uses a modified Hamming code which was optimized for multi-chip EDCU systems. The code is such that partial parity is computed by all 8206's in parallel. No 8206 requires more time for logic propagation than any other, hence no single device becomes a bottleneck in the parity operation.

The 8206 is easy to use with all kinds of dynamic RAM controllers. Because of its dual-bus architecture, the amount of control logic needed is very small.

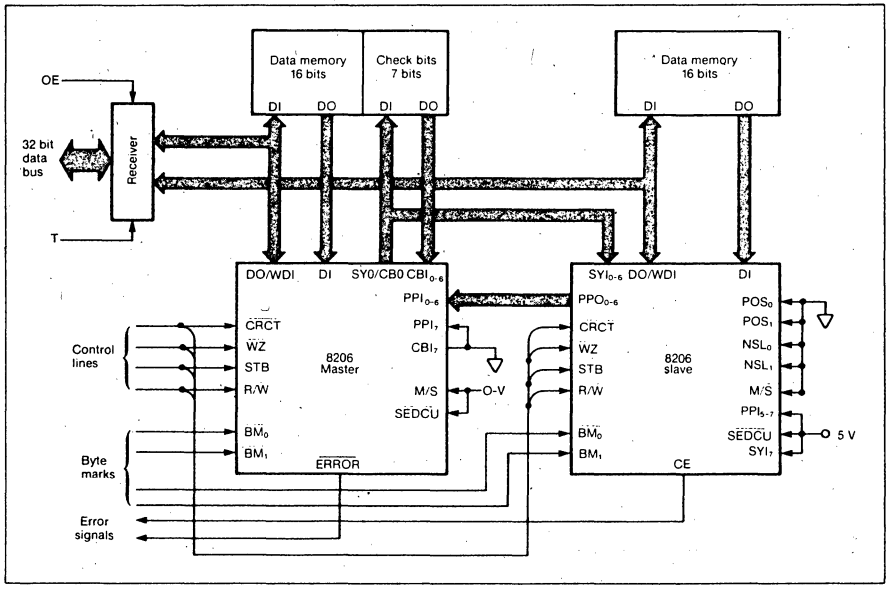
Figure 7a shows a memory design using the 8206 with Intel's 8203 64-kbit dynamic RAM controller and 2164 64-kbit dynamic RAM. As few as three additional ICs complete the memory control function (Fig. 7b).

For simplicity, all memory cycles are implemented as single-cycle read-modify-writes (Fig. 8). This cycle differs from a normal read or write primarily in when the RAM Write Enable ($\overline{WE}$) is activated. In a normal write cycle, $\overline{WE}$ is activated early in the cycle. In a read cycle, $\overline{WE}$ is inactive.

A read-modify-write cycle consists of two phases. In the first phase, $\overline{WE}$ is inactive, and data are read from the RAM; for the second phase, $\overline{WE}$ is activated and the (modified) data is written into the same word in the RAM. Dynamic RAMs have separate data input and output pins so that modified data may be written, even as the original data is being read. Therefore, data may be read and written in only one memory cycle.

In order to perform read-modify-writes in one cycle, the 8203 dynamic RAM's CAS strobe must be active long enough for the 8206 to access and correct data from the RAM, and write the corrected data back into RAM. CAS active time (t_{CAS}) depends on the 8203's clock frequency. The clock frequency and dynamic RAM must be chosen to satisfy:

$$t_{CAS(min)}^{8203} \geq t_{CAC}^{RAM} + TDVQV^{8206} + TQVQV^{8206} + t_{DS}^{RAM} + t_{CWL}^{RAM}$$



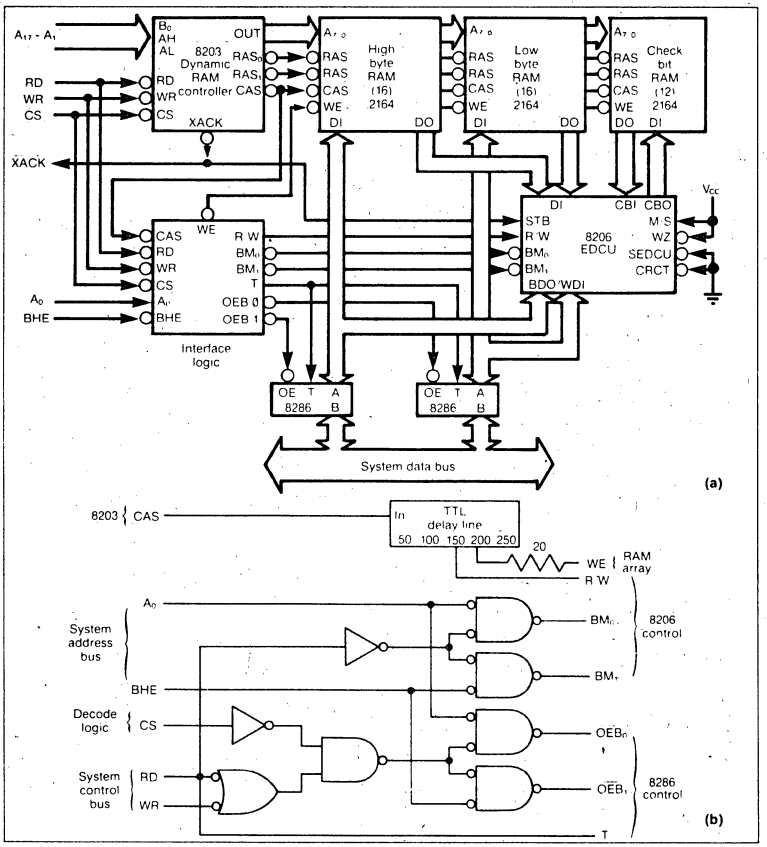
6. No additional logic is required for this 32-bit master-slave system. The slave calculates partial parity on its half of the data, and the master determines which of the 32 data bits and 7 check bits is in error.

Table 3. Syndrome decoding identifies and corrects all single-bit errors.

Syndrome bits 7	6	5	4	0 0 1 0 2 0 3 0	1 0 0 0 0 0	0 1 0 0 0 0	1 1 0 0 0 0	0 0 1 1 0 0	1 0 1 0 0 0	0 1 1 1 0 0	1 1 0 0 0 0	0 0 1 1 0 1	1 0 1 1 0 1	0 1 1 1 0 1	1 1 0 0 1 1	0 0 1 1 1 1	1 0 1 1 1 1	0 1 1 1 1 1	1 1 1 1 1 1
0	0	0	0	N	CB0	CB1	D	CB2	D	D	18	CB3	D	D	0	D	1	2	D
0	0	0	1	CB4	D	D	5	D	6	7	D	D	3	16	D	4	D	D	17
0	0	1	0	CB5	D	D	11	D	19	12	D	D	8	9	D	10	D	D	67
0	0	1	1	D	13	14	D	15	D	D	21	20	D	D	66	D	22	23	D
0	1	0	0	CB6	D	D	25	D	26	49	D	D	48	24	D	27	D	D	50
0	1	0	1	D	52	55	D	51	D	D	70	28	D	D	65	D	53	54	D
0	1	1	0	D	29	31	D	64	D	D	69	68	D	D	32	D	33	34	D
0	1	1	1	30	D	D	37	D	38	39	D	D	35	71	D	36	D	D	U
1	0	0	0	CB7	D	D	43	D	77	44	D	D	40	41	D	42	D	D	U
1	0	0	1	D	45	46	D	47	D	D	74	72	D	D	U	D	73	U	D
1	0	1	0	D	59	75	D	79	D	D	58	60	D	D	56	D	U	57	D
1	0	1	1	63	D	D	62	D	U	U	D	D	U	U	D	61	D	D	U
1	1	0	0	D	U	U	D	U	D	D	U	76	D	D	U	D	U	U	D
1	1	0	1	78	D	D	U	D	U	U	D	D	U	U	D	U	D	D	U
1	1	1	0	U	D	D	U	D	U	U	D	D	U	U	D	U	D	D	U
1	1	1	1	D	U	U	D	U	D	D	U	U	D	D	U	D	U	U	D

N = No error
CBX = Error in check bit X (correctable)
X = Error in data bit X (correctable)

D = Double-bit error (detected but not corrected)
U = uncorrectable multi-bit error



7. The 256-kbyte system (a) has 32 64-kbyte dynamic RAMs for data plus 12 dynamic RAMs for error correction. The dynamic RAMs are controlled by the 8203 dynamic RAM controller while error correction control is supplied by the 8206. Interface logic (b) allows the 8203/8206 system to implement read-modify-write cycles by generating Write Enable ($\overline{WE}$) to the RAMs, Read/Write ($R/\overline{W}$) to the 8206, and byte-control signals.

The 8203 itself performs normal reads and writes. To perform read-modify-writes, simply change the timing of the $\overline{WE}$ signal. In Fig. 7b, $\overline{WE}$ is generated by the interface logic—the 8203 $\overline{WE}$ output is not used. All other dynamic RAM control signals come from the 8203. A 20-Ω damping resistor reduces the $\overline{WE}$ signal ringing. These damping resistors are included on-chip for all 8203 outputs.

The interface logic generates the R/$\overline{W}$ input tothe 8206. This signal is high for read cycles and low for write cycles. During a read-modify-write cycle, R/$\overline{W}$ is first high, then low.

The falling edge of R/$\overline{W}$ tells the 8206 to latch its syndrome bits internally and generate corrected check bits to be written to RAM. Corrected data are already available from the DO pins. No control signals at all are required to generate corrected data. R/$\overline{W}$ is generated by delaying CAS from the 8203 with TTL-buffered delay line. This delay ($t_{DELAY\ 1}$) must satisfy:

$$t_{DELAY\ 1} \geq t_{CAC}{}^{RAM} + TDVRL^{8206}$$

The 8206 uses multiplexed pins to output the syndrome word and then the check bits. The R/$\overline{W}$ signal may be used to latch the syndrome word externally for error logging. The 8206 also supplies two useful error signals: $\overline{ERROR}$ indicates an error is present in the data or check bits; CE tells if the error is correctable (single bit) or uncorrectable (multiple bits).

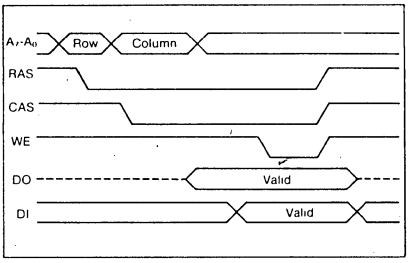
After R/$\overline{W}$ goes low, sufficient time is allowed for the 8206 to generate corrected check bits, then the interface logic activates $\overline{WE}$ to write both corrected data and check bits into RAM. $\overline{WE}$ is generated by delaying $\overline{CAS}$ from the 8203 with the same delay line used to generate R/$\overline{W}$. This delay, $t_{DELAY\ 2}$, must be long enough to allow the 8206 to generate valid check bits, but not so long that the spec of the RAM (t_{CWL}) is violated. This is expressed by:

$$t_{DELAY\ 1} + TRVSV^{8206} \leq t_{DELAY\ 2} \leq t_{CAS(min)}^{8203} - t_{CWL}{}^{RAM}$$

Errors in both data and check bits are automatically corrected, without special 8206 programming.

Since the 8203 terminates $\overline{CAS}$ to the RAMs at a fixed interval after the start of a memory cycle, a latch is usually needed to maintain data on the bus until the 8086 completes the read cycle. This is conveniently done by connecting XACK from the 8203 to the STB input of the 8206, latching the read data and check bits inside the 8206.

The 8086, like all 16-bit CPUs, is capable of reading and writing single-byte data to memory. As just explained, the Hamming code works only on entire words, so in byte writes, and new byte and old byte must be merged, and new check bits written for the composite word. This is difficult with most EDC chips, but it is easy with the 8206.



8. In all memory cycles, the row and column addresses are strobed to the RAMs by $\overline{RAS}$ and $\overline{CAS}$. Sometime after the data out is valid, the control logic in Fig. 7b generates Write Enable ($\overline{WE}$) to write the data back into the RAMs.

Further qualifications on 8206 operation

Referring again to Fig. 7b, the 8206 byte-mark inputs ($\overline{BM_0}$, $\overline{BM_1}$), are generated from A0 and BHE, respectively (off the 8086's address bus) to tell the 8206 which byte is being written. The 8206 performs error correction on the entire word to be modified, but 3-states its DO/WDI pins for the byte to be written; this byte is provided from the data bus by enabling the corresponding 8286 transceiver. The 8206 then generates check bits for the new word.

During a read cycle, $\overline{BM_0}$ and $\overline{BM_1}$ are forced inactive (i.e., the 8206 outputs both bytes even if 8086 is only reading one). This is done since all cycles are implemented as read-modify-writes, so both bytes of data (plus check bits) must be present at the RAM data in pins to be rewritten during the second phase of the read-modify-write cycle. Only those bytes actually being read by the 8086 are driven on the data bus by enabling the corresponding 8286 transceiver.

The 8286's Output Enables ($\overline{OEB_0}$, $\overline{OEB_1}$) are qualified by the 8086's $\overline{RD}$, $\overline{WR}$ commands and the 8203's $\overline{CS}$ command. This serves two purposes: It prevents data bus contention during read cycles and it prevents contention between the transceivers and the 8206 DO pins at the beginning of a write cycle.

Thanks to the use of a 68-pin leadless chip carrier, the 8206 error detection and correction unit is able to implement an architecture with separate 16-pin input and output buses. Thus single-bit error correction may be added to a system with a minimum of control signals or external logic.□

intel

ARTICLE REPRINT

AR-197

JANUARY, 1982

Better processor performance via global memory

JOSEPH P. ALTNETHER
Intel Corporation

ORDER NUMBER: 210378

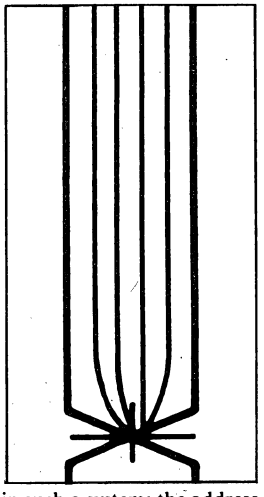
BETTER PROCESSOR PERFORMANCE VIA GLOBAL MEMORY

Wait states are eliminated by joining global and local memories through five TTL components

by Joseph P. Altnether

At least 60% of today's designs incorporate microcomputers, which have become one of the most widespread components in a variety of electronic equipment ranging from video games to navigational flight computers. Microcomputers comprise several elements. One of the more important of these is the memory. In early systems (and even in some of today's low performance microcontrollers), the memory is interfaced and accessed exactly like any other peripheral. Such an architecture is shown in Fig 1. For this type of application, data store (random access memory), control store (electrically programmable read only memory/read only memory), and input/output reside on a single bus connected directly to the central processing unit. This kind of application is usually a dedicated system performing only one function, such as control of a vending machine.

Memory consists of control store and data store. The former occupies most of the memory and contains about 16k bytes of program; the latter is small and contains less than 4k bytes. A major design goal is simplicity, which can be best achieved when the components of control store and data store are compatible. It is much simpler and certainly more efficient to use the same set of address decoders and drivers, as well as data transceivers, for both control and data store. This is achieved with common pinout and functionality between random access memory (RAM) and electrically programmable read only memory/read only memory (EPROM/ROM). Therefore, the memory should be an 8-byte wide RAM. Several disadvantages are inherent in such a system: the address space is limited; and because all elements—including the central processing unit (CPU)—reside on a common bus, the CPU, as the bus controller, suspends processing to control bus operations.



Enhancing the system

The performance of this system can be enhanced by upgrading to a microprocessor and storing a variety of programs in permanent bulk memory. In this kind of system, control store consists of a RAM containing up to 64k bytes (Fig 2). This memory is much larger because it serves a dual function: data store and control store. Programs to be executed are downloaded via a boot program residing in EPROM. The system overcomes the memory addressing space deficit of the previous system but still retains the disadvantage of having all memory

Joseph P. Altnether is technical marketing manager for memory products at Intel Corp, Aloha, OR 97005. Before that, he worked as an applications engineer at Intel and a memory system designer for nuclear/medical instruments at G. D. Searle. Mr Altnether has a BSEE from St Louis University.

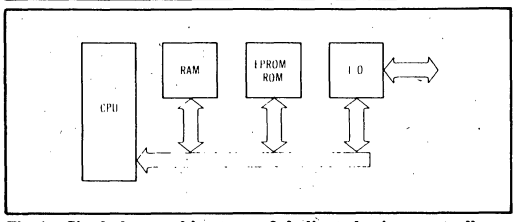


Fig 1 Single-bus architecture of dedicated microcontroller. Though inexpensive, this configuration limits available address space and requires that CPU suspend processing when controlling bus.

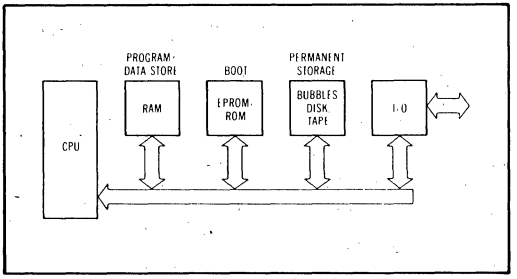


Fig 2 Improved performance results when microprocessor using RAM program storage for up to 64k bytes of data and control information is used. Disadvantages of common bus architecture are retained, however.

reside on the CPU bus. For example, throughput efficiency could be improved if it were possible to download other portions of the program into control store while executing out of control store (dual porting).

High performance in both processing power and speed is realized in distributed processing systems. In such a configuration, several processors, together with their local memories, are distributed throughout the system. These could be structured like the systems previously described; however, they have an important distinguishing element—multiple local buses with a common system or global bus. Fig 3 depicts such a system. Here, the advantages of dual porting, error checking and correction, and direct memory access all become cost effective.

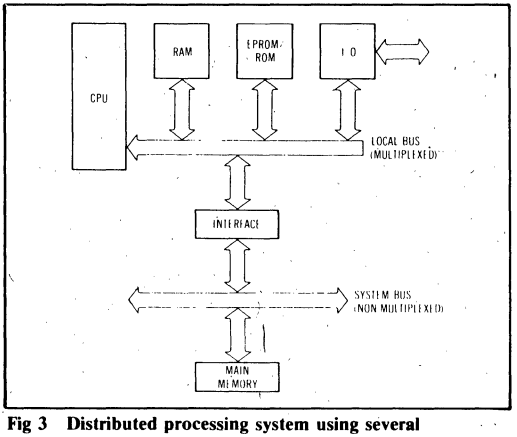


Fig 3 Distributed processing system using several processors with local memories and common (global) bus provide high performance. Each processor in system has access to large (1M-byte) global memory.

...because the global memory is so large, the RAM used must be as dense as possible to reduce the number of components.

Residing on the system bus is a global memory to which every processor has access. This memory can be very large—even greater than 1M byte. Consequently, it could be disk, tape, magnetic bubble, or RAM. If built with RAMs, the type used would be dynamic RAMs (DRAMs) for several reasons. First, because the global memory is so large, the RAM used must be as dense as possible to reduce the number of components. Lower component count reduces system cost and increases system reliability, which is inversely proportional to the number of components in the system. Second, the components should consume minimal power. Even a small amount of power per device multiplied by hundreds of devices will require a large power supply. In addition, as the power requirements increase, so do the cooling requirements, which again add to the overall system cost and operating cost.

Finally, the RAM must be low cost to be competitive and provide ample operating margins. DRAMs meet these requirements quite adequately as they provide the lowest cost per bit and also consume the lowest power per bit of RAM devices. Unfortunately, designing with DRAMs has long been considered esoteric and difficult. In fact, some designers still believe that DRAMs do not even work. The first of these beliefs was based on fact in earlier days, but the second is based on an emotional

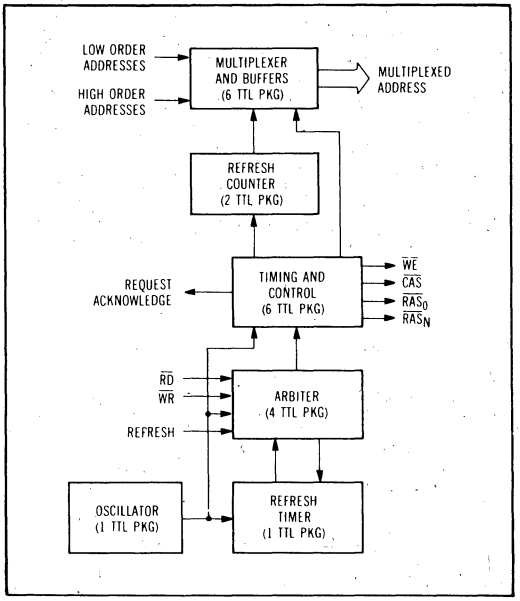


Fig 4 Typical DRAM controller. Oscillator provides timing and control logic for refresh timer.

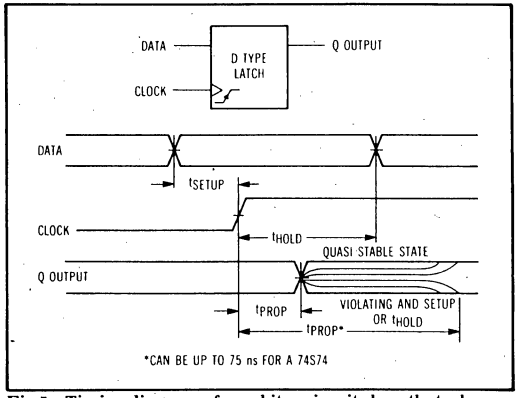


Fig 5 Timing diagrams for arbiter circuit show that when certain conditions exist, output is analog signal floating between TTL levels 1 and 0. During this 75-ns period, no decisions can be made and refresh failure occurs.

reaction to a memory that forgets unless it is periodically told to remember. DRAMs do not lose data if they are properly refreshed. This can be easily accomplished by a memory interface controller.

Designing a DRAM system

Although it is more difficult to design a DRAM system than a static RAM (SRAM) system, it is not impossible. Shown in Fig 4 is a typical DRAM controller. At the heart of the controller is an oscillator which provides timing and control logic for the refresh timer. Because DRAMs are clocked, they need signals like row address strobe (RAS), column address strobe (CAS), and write enable (WE), which come from the control logic. The refresh timer will periodically time out, typically every 15 μs, to request a refresh cycle asynchronously with respect to CPU memory requests. To decide which request (CPU or refresh) is granted first, an arbiter circuit is required. The arbiter is the most complicated controller element to design. In theory, a D type flipflop could be an arbiter (Fig 5). If refresh request is set asynchronously with respect to the system clock, a decision on the Q output can be made. If Q is true, the refresh cycle is granted; if false, the CPU is given access. Timing relationships of data and clock indicate that normal operation of the flipflop will occur if setup and hold times of data with respect to the clock are met.

If the setup or hold times are violated, however, the Q output is no longer a transistor-transistor logic (TTL) level 1 or 0. The output becomes an analog signal floating between TTL levels somewhat like a 3-state output device with the output in a high impedance state. This condition can persist for as long as 75 ns, during which it is impossible to make a decision. At the system level this appears as a refresh failure. Lastly, the controller requires multiplexers and drivers for the memory addresses. The total system is built with 20 TTL components (Fig 4).

The...DRAM controller...includes an arbiter which synchronizes the refresh and memory cycle requests to eliminate the arbitration problem....

Another consideration is design time. About four weeks are usually required for design, two weeks for worst-case analysis, six weeks for printed circuit board layout, four weeks for building and debugging, and another four weeks for redesigning to add features or correct errors. And this does not include a possible second iteration effort. In any case, the task could consume up to six man-months.

A simpler solution

Intel's dynamic RAM controller, the 8203, is contained in a single 40-pin package that incorporates the entire DRAM controller (Fig 6). It includes an arbiter which synchronizes the refresh and memory cycle requests to eliminate the arbitration problem previously described. Compatible with the 8080A, 8085A, iAPX88, and iAPX86 family of microprocessors, the device directly addresses half a megabyte of memory composed of 64k RAMs (eg, the Intel 2164). All the refresh functions are provided: timer, 8-bit address counter, and multiplexers for addresses. Because refresh is usually performed asynchronously with the CPU cycles, provision is made for performing synchronous refresh if required. At times the controller will be providing refresh when the CPU requires access. Consequently, the CPU must be placed in a WAIT mode. This is accomplished with a signal from the 8203 called SACK. In addition, the signal XACK can be used to clock data into the latches during a read cycle.

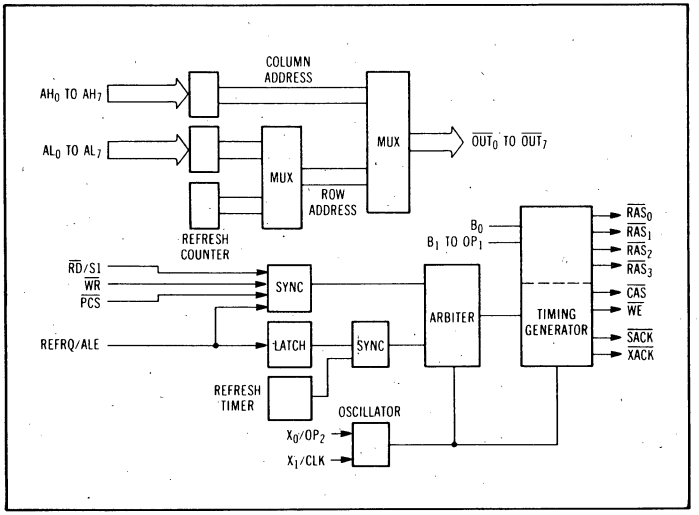


Fig 6 40-pin, 8203 DRAM controller includes arbiter that synchronizes refresh and memory cycle requests eliminating indecisive condition of Fig 5. Chip directly addresses 0.5M byte.

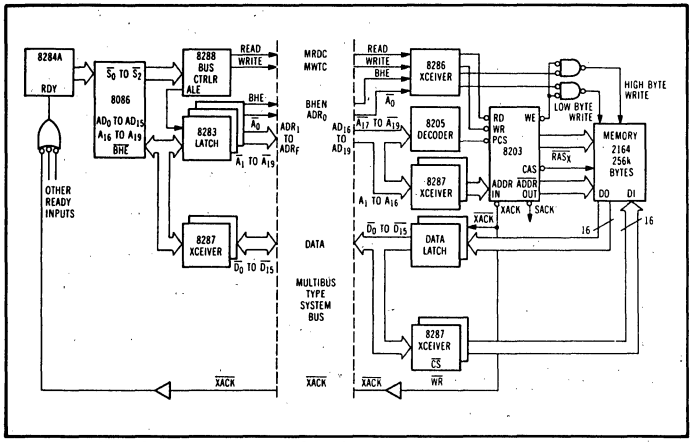


Fig 7 Typical global memory interface for microprocessor. Multiplexed address/data bus serves as local bus and demultiplexed address/data bus serves as global bus. Minimum or maximum mode operation is possible.

To illustrate the ease of interfacing global memories to the microprocessor, an iAPX86 system using an 8086 is shown. The multiplexed address/data bus is normally thought of as the local bus, and the demultiplexed address and data bus as the global bus. In much larger systems, it would be possible for the local bus to be demultiplexed immediately at the processor and for another bus that services the entire system to be the global bus. The system described works on either demultiplexed bus. The iAPX86 can be operated in either of two modes, minimum or maximum (Fig 7). In the former mode, the microprocessor generates the read and write commands directly, whereas in the latter mode, a bus controller such as the Intel 8288 is required. In this case, the iAPX86 outputs status bits S_0 to S_2 that are interpreted by the bus controller. Commands for read and write are now generated by the bus controller. Independent of the mode, the 8203 and memory interface identically to the microprocessor.

Ease of use or simplicity of design have been balanced against performance. The simple system shown (Fig 7) typically operates with one to two WAIT states required. For the minimum mode operation, the read (RD) and write (WR) commands occur too late in the memory cycle to allow the DRAM controller to generate a ready signal early enough to avoid WAIT states. Operation without WAIT states can be accomplished by transmitting advance RD or WR commands to the memory. This is a non-trivial task in the minimum mode because the iAPX86 produces the RD and WR signals in a fixed relationship after address latch enable (ALE) occurs. For maximum mode operation, the iAPX86 outputs status bit information ahead of ALE. With proper logic circuitry, these status bits can be decoded and the information used to initiate the advance RD and WR commands.

With a small amount of additional logic, it is possible to combine ease of use of the DRAM controller with high performance. As a result, the iAPX86 can operate at 5 MHz and requires no WAIT states unless the memory is

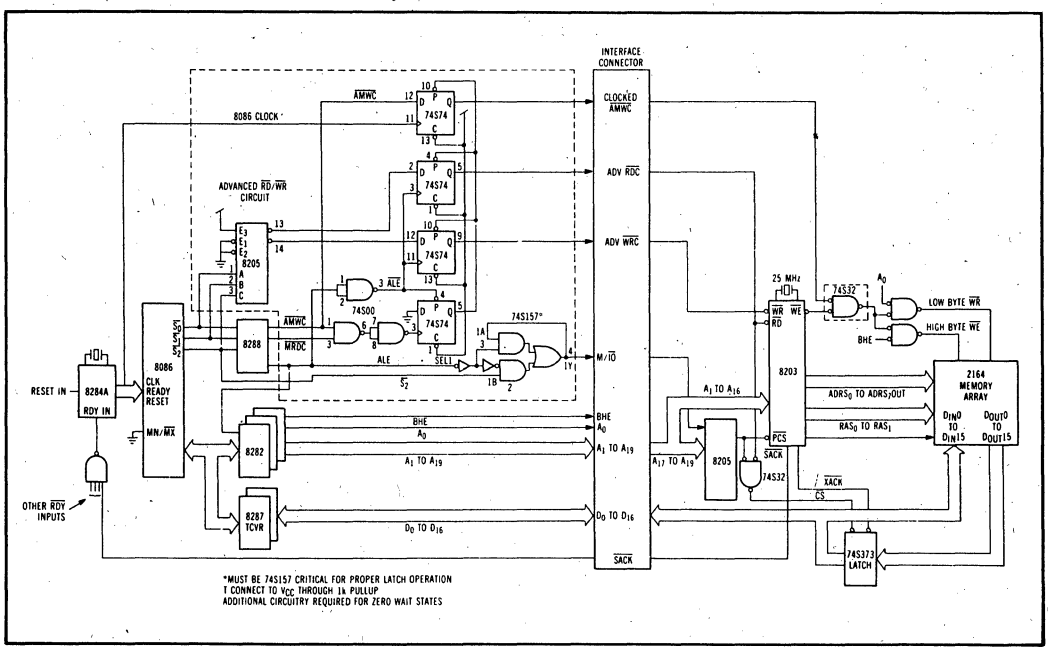


Fig 8 To achieve 5-MHz operation with no WAIT states, additional circuitry (dashed lines) must be added.

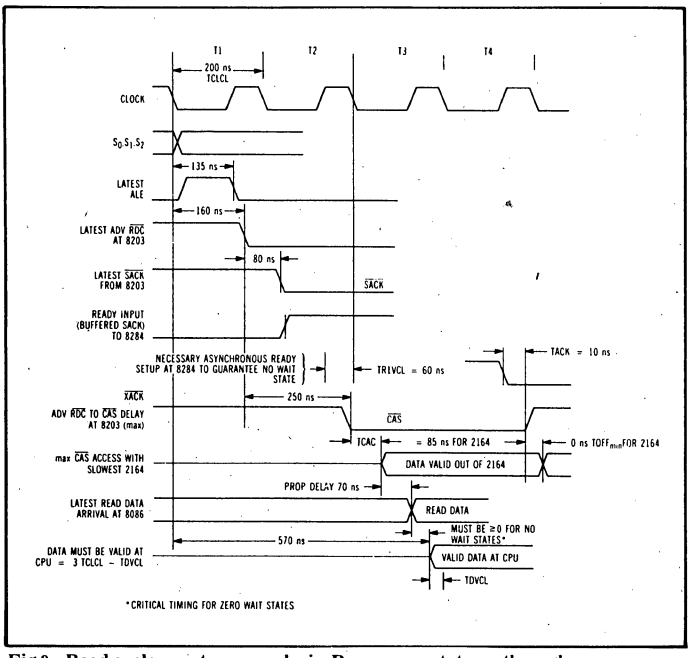


Fig 9 Read cycle worst-case analysis. Processor T states T1 through T4, are shown. For read without WAIT states, valid data must reach processor 30 ns before end of T3.

being refreshed. The circuitry added to the system is shown inside the dashed lines in Fig 8. The 8205 is a 3:8 line decoder which monitors the status lines. With the proper combination of status lines S_0, S_1, and S_2, an advanced RD command (ADV RDC) or an advanced WR command (ADV WRC) will be output on pin 13 or pin 14, respectively.

The RD or WR command, whichever is true, is latched by the corresponding 74S74 on the falling edge of ALE from the 8288 bus controller. Latch outputs at pins 5 and 9 (ADV WRC and ADV RDC) are entered into the 8203A WR and RD inputs directly. The two latches are cleared later on the trailing edge of either the memory read command (MRDC) or memory write command (MWRC) through the two 74S00 gates. System acknowledge (SACK)—used in place of (XACK) because it occurs sooner—is ANDed with protected chip select (PCS) and returned to the

Global memory can be easily built using only DRAMs and the... DRAM controller.

8284A, which provides a synchronous ready signal to the iAPX86. The S_2 status bit (memory operation) is latched by the 74S157 on the trailing edge of ALE. The 2:1 multiplexer is configured as a high speed flow-through latch by feeding the output back into the input. Propagation delay time is only 7.5 ns. The advanced memory write command (AMWC) is ANDed with WE to provide WE to the DRAMs.

Read cycle worst-case analysis (Fig 9) considers the maximum time delays. The four processor T states are indicated by T1 through T4. To read without WAIT states, valid data must reach the processor by the end of T3 minus 30 ns. The latest read data arrival at the processor does indeed fall within this time frame. The memory read cycle begins with ADV RDC (Fig 9), which is latched by the falling edge of ALE. ADV RDC reaches the 8203 at 160 ns into the cycle and begins access. Within 80 ns, SACK is valid and ANDed with PCS to be returned to the 8284A clock as READY. As a result, no WAIT states are required unless the DRAM controller is performing a refresh cycle. The system is CAS access limited, and as such the ADV RDC to CAS delay is 225 ns. The 85-ns CAS access time (t_{CAC}) must be added to this time. Finally, an additional 45-ns delay through the buffers is included for a total delay time of 510 ns. Access required is 3 T times (600 ns) minus 30 ns, or 570 ns. The system indeed requires no WAIT states for operation.

In the write cycle, the relationship between data and WE and the relationship of CAS and WE must be guaranteed. Data are written into the DRAM on the falling edge of WE. Consequently, data must be valid prior to the falling edge of WE. The skew of data from the processor and WR from the 8203 is such that it is possible for the data to be valid after the falling edge of WR. In this event, invalid data would be written into the memory as shown in Fig 10(a). In addition, DRAMs have a timing constraint, t_{CWL}, which is the overlap between CAS and WE. If CAS were early and MWTC were late, t_{CWL} would be violated as shown in Fig 10(b). Both of these requirements are satisfied by ANDing AMWC with WR.

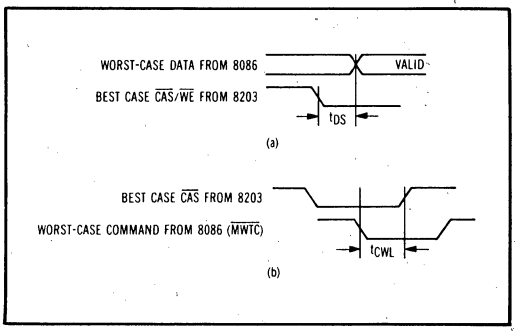


Fig 10 Required WE delay timing to memory

Fig 11 depicts the worst-case timing analysis for a write cycle, which is similar to that for the read cycle with a few exceptions. The ADV WRC is latched on the falling edge of ALE. The earliest that CAS can occur is 105 ns after ADV WRC starts the write cycle. Valid data are output from the CPU within 210 ns and reach the memory 35 ns later. By ANDing the AMWC with WR from

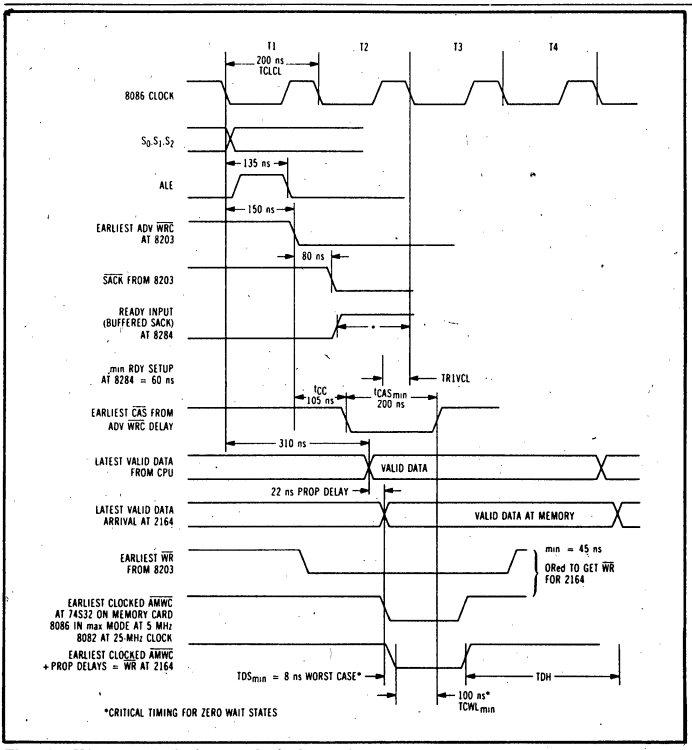


Fig 11 Worst-case timing analysis for write cycle

the 8203, WE falls a minimum of 8 ns after data are stable and valid at the memory. In addition, this ANDing guarantees a minimum t_{CWL} of 100 ns.

Overall system performance is improved by using global as well as local memories. Global memory can be easily built using only dynamic RAMs and the 8203 dynamic RAM controller. Performance, together with ease of use, is achieved by adding just five TTL components. The design of a 5-MHz system that runs without WAIT states is a good example of this approach.

Acknowledgment

The author would like to thank Bill Righter and David Chamberlin for their help in preparing this article.

intel

ARTICLE
REPRINT

AR-274

March 1983

The chip that refreshes itself

An 8k x 8 memory chip with onchip refresh and control circuitry offers designers the best aspects of both dynamic and static RAMs.

John J. Fallin
Joseph P. Altnether
William H. Righter
Intel Corporation

ORDER NUMBER: 230652-001

THE CHIP THAT REFRESHES ITSELF

An 8k x 8 memory chip with onchip refresh and control circuitry offers designers the best aspects of both dynamic and static RAMs.

by John J. Fallin, Joseph P. Altnether, and William H. Righter

The ideal microprocessor memory has three major characteristics: ease of use, flexibility, and low cost. Unfortunately, all of these characteristics are seldom available in a single device. Thus, memory system design involves compromises. Matching memory component characteristics to desired design parameters rarely results in a perfect fit.

To optimize memory design, the designer must set priorities in choosing between static random access memory (SRAM) and dynamic random access memory (DRAM). Neither choice completely supplies all the beneficial characteristics. SRAMs are easy to use, but high cost, limited density, and high power consumption limit their application to under 64k bytes. DRAMs are suitable for large memory arrays (above 64k bytes) where density and low power are important. In addition, the refresh overhead of large memory systems is spread over a large amount of memory. Between the two extremes represented by static and dynamic RAMs, however, is a performance zone where all three ideal characteristics (low cost, flexibility, and ease of use) are required.

Low cost can be achieved with a DRAM cell. A major factor determining the cost of a memory device is the physical area occupied by the memory cell. Small cell size allows small die size, which results in more die per wafer. The net result is lower cost. DRAMs achieve lower unit costs because they use a single transistor for the memory cell. Each SRAM cell, on the other hand, requires six transistors. In addition, the DRAM cell provides the benefits of low power and high density. The remaining ideal characteristics (flexibility and ease of use) demand more innovative approaches than are presently available.

A microprocessor memory system using DRAMs consists of three major elements: microprocessor, memory, and controller. To gain ease of use, the controller must be incorporated into either the microprocessor or the memory. By including the memory control in the microprocessor, a simple memory device can be built in which the cost of the control logic is distributed over the entire memory system. Sadly, this approach has inherent faults. Because refresh is derived from microprocessor timing, any operation that suspends or stretches the timing must be carefully analyzed to ensure that it does not violate system refresh timing. Examples of such time stretching operations are extended wait states, hold operations in direct memory access, or the single-step operations important in system debugging. Moreover, placing the burden of refresh on the microprocessor can reduce processor performance by suspending operation to service memory refresh.

Another technique for refresh control is to incorporate most of the refresh mechanism on the RAM chip, leaving the most difficult portion—the arbiter—to the system designer. This type of RAM is known as a

John J. Fallin is currently an applications engineer at Intel Memory Products, 2111 NE 25th, Hillsboro, OR 97123. He is responsible for iRAM applications. Mr Fallin has a BSEE from Oregon State University.

Joseph P. Altnether is the technical marketing manager at Intel Memory Products and is involved in the technical marketing of Intel iRAMs. Mr Altnether has a BSEE from St Louis University.

William H. Righter is also an applications engineer at Intel Memory Products. He is responsible for the graphics applications of dynamic RAMs. Mr Righter is currently working on a BSEE from the University of Portland.

pseudo- or quasistatic RAM. Statistical techniques or special features of a particular central processing unit (CPU) may be used to accomplish refresh, though it is not always guaranteed. The system must be thoroughly analyzed to ensure proper refresh timing.

The third memory refresh method incorporates all of the memory control logic, including arbitration logic, into the RAM chip. Thus, all DRAM cell control is contained on the same piece of silicon, creating a complete integrated memory system on a chip. This concept is called an integrated RAM (iRAM.) Combining the best features of dynamic and static RAM, the iRAM satisfies the ideal microprocessor memory requirements—a DRAM storage cell for low cost, a Joint Electron Device Engineering Council 28-pin package configuration conforming to universal flexibility standards, and ease of use because of onchip control. (See the Panel.)

Intel's 8k x 8 iRAM contains all the elements needed for complete memory control. Recognizing the need for both synchronous and asynchronous refresh, Intel offers the 2187 iRAM with synchronous refresh for special applications and the 2186 with asynchronous refresh for general purpose designs. With onchip refresh, the 2186 needs no external stimulus or control. As a result, memory is autonomous; like an SRAM it can be left alone with power applied and still retain data. Any operation that suspends or stretches the system timing has no effect on refresh operation. However, because refresh is asynchronous with the microprocessor, a memory request can occur during a refresh cycle. In this case, the iRAM signals the system microprocessor that a delay will occur in the cycle.

Device operation

The 2186/2187 performs four types of cycles: read, write, false memory, and refresh. It is important to note that chip enable ($\overline{CE}$) is an edge-triggered, not a level-triggered, input. On the 2186/2187, an access cycle is requested for every high to low transition of $\overline{CE}$. Care must be taken to ensure that a new access cycle is not requested before the previous cycle is completed. This would violate memory cycle time (TELEL) and would jeopardize data integrity. A minimum precharge ($\overline{CE}$ high time—TEHEL) must also be guaranteed. The violation of $\overline{CE}$ high time would most likely occur in systems where noise spikes can occur on chip enable.

Functioning like a clocked static RAM, iRAM addresses are latched off the external address bus on the falling edge of $\overline{CE}$. This feature is useful in many designs because it saves the designer one or two transistor-transistor logic packages. In contrast to the trailing edge write of the SRAM, the iRAM requires a leading edge write. Further, the iRAM permits three different types of access cycles. Depending on the active time of $\overline{CE}$ and its relation to output enable ($\overline{OE}$) or write enable ($\overline{WE}$), a long mode cycle, a pulsed mode cycle, or a false memory cycle (FMC) can be performed.

A long mode cycle is similar to a fully static RAM cycle in that $\overline{CE}$ remains valid throughout the entire cycle. In the pulsed mode cycle, the iRAM operates as a clocked static RAM and $\overline{CE}$ is active only at the beginning of the cycle. When $\overline{CE}$ becomes active, without an $\overline{OE}$ or $\overline{WE}$, the iRAM performs an FMC and terminates on the rising edge of $\overline{CE}$. This is useful during byte write operations of 16-bit microprocessors. In this case, a 16-bit word is selected (two iRAMS), and only one receives a write pulse to perform byte write. The other selected iRAM performs an FMC.

Assuming a refresh is not in progress when the cycle begins, an access cycle is initiated on the high to low transition of $\overline{CE}$. After activating $\overline{CE}$, addresses are latched from the external bus and data are presented to the bus. Data will remain at the bus as long as $\overline{OE}$ remains active, independent of the state of $\overline{CE}$.

Any operation that suspends or stretches the system timing has no effect on refresh operation.

If a refresh cycle is in progress at the time $\overline{CE}$ goes low, a deferred cycle occurs. In this case, ready (RDY) will respond by going low within a given time (TELRL) of $\overline{CE}$ going low. After the refresh cycle and part of the access cycle complete, RDY will return to a high state. At a specified time after this (TRHQV), valid data will become available at the data input/output (I/O) outputs.

It should be noted that $\overline{OE}$ can remain active for an unlimited period of time, and data will remain on the bus even though internal refresh cycles continue to be performed. This allows operation in systems using single-stepping hardware debug, since wait states can be inserted at will. RDY does not respond under these conditions.

For the sake of clarity, assume that a refresh cycle is not in progress at the time of $\overline{CE}$'s falling edge. A write cycle begins in the same way as a read cycle, with addresses latched on the falling edge of $\overline{CE}$. For a pulsed mode, $\overline{WE}$ must go low within a specified time of the falling edge of $\overline{CE}$ (TELWL). If this specification time is not met, an FMC occurs and thus, no write. For a long mode, TWLEH must be met to ensure a write given setup time before (TDVWL) and a given hold time (TWLDX) after the leading edge of $\overline{WE}$.

If a refresh cycle is already in progress at the onset of a write cycle, the RDY will respond at a given time (TELRL) after $\overline{CE}$'s falling edge. Data will still be latched into the device on the falling edge of $\overline{WE}$, but the actual write to the array will not occur until after the refresh cycle is completed. RDY responds during write cycles to prevent cycle timer (TELEL) violations. As in a read cycle, $\overline{CE}$ and $\overline{WE}$ can remain active for an indefinite period to accommodate single-step type operation. An FMC occurs when $\overline{CE}$ becomes active but neither $\overline{OE}$ nor $\overline{WE}$ becomes active. An FMC is a valid mode of operation and also acts like a row address strobe (RAS) only refresh, in which the row selected by the seven external row addresses is refreshed.

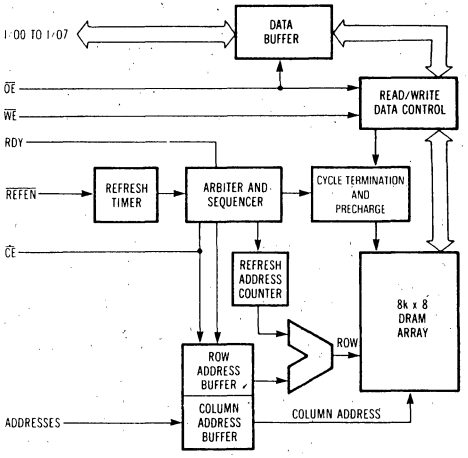
Internal vs external refresh

The 2186 internal refresh is completely automatic, requiring no external stimulus; an internal timer provides refresh requests. If an access cycle is requested during a refresh cycle, the 2186 will respond by outputting a RDY low. For applications requiring maximum performance,

the 2187 allows external generation of refresh signals. A high to low transition on the refresh enable ($\overline{\text{REFEN}}$) input of the 2187 will initiate a refresh cycle. After starting a refresh cycle, one cycle time (TELEL) must be allowed before attempting another access or refresh cycle. Deferred access cycles are not allowed on the 2187, as it has no arbitration circuitry. If $\overline{\text{REFEN}}$ is held low for at least one timer period, the internal timer will begin to time out, and the 2187 will maintain refresh with no outside intervention.

Inside the iRAM

Included in the 5-V iRAM device are a dynamic array, an arbiter, a refresh address counter, and a refresh timer along with complete control and precharge circuitry. The 2187 differs from the 2186 only in the refresh timer and arbiter control circuitry. The diagram illustrates the interrelation of these iRAM elements.



Arbiter

The arbiter that determines the priority sequence of two or more asynchronous inputs is the most significant element of the 2186 internal refresh circuitry. In the 2186, the two inputs to the arbiter are an external access cycle request and an internal refresh cycle request. When either of these requests is made, the arbiter decides whether the cycle will proceed immediately or be delayed. For example, if an access cycle is in progress at the time of an internal refresh request, the refresh cycle will be delayed until after the access cycle is completed. Conversely, if an access cycle is requested while the 2186 is performing a refresh cycle, the access cycle will be delayed. Here the 2186 will respond with a RDY low output, instructing the device that more time must be allowed. In the limit, both cycle requests can occur simultaneously. Therefore, arbitration becomes necessary along with the simple state gating as outlined.

Array

The memory array, designed with direct random access memory (DRAM) storage cells, is fabricated using an N-channel double-layer polysilicon gate process. The array features polysilicon word lines and folded metal bit lines that provide high common mode rejection.

Refresh timer

Refresh peripheral circuitry is included on the device to preserve the integrity of the data in the DRAM array. A refresh timer provides refresh cycle requests at appropriate intervals. The timer is designed to track with both process variations and temperature so as to guarantee proper refresh over all specified ranges.

Row address counter

When the internal refresh cycle is granted, refresh addresses are provided by a refresh address counter. This counter contains the 7-bit address of the next row to be refreshed and is incremented after each refresh cycle. Control logic within the peripheral circuitry handles the multiplexing of external memory addresses and refresh addresses during appropriate cycles.

Device pinout

The 2186/2187 comes in Joint Electron Device Engineering Council compatible 28-pin socket. Pin 1 (labeled "CNTRL") becomes the RDY output on the 2186, or the $\overline{\text{REFEN}}$ input on the 2187. Pin functions are described in the Table.

iRAM Pin and Signal Definitions

Pin	Description/Function
A_0 to A_{12}	Address inputs: These inputs provide the addresses needed to select one of 8192 bytes. Addresses appearing on these inputs are latched into the device on the high to low transition of $\overline{\text{CE}}$.
I/O_0 to I/O_7	Data input/output. These bidirectional pins receive data to be written to the device and output data during a read cycle.
$\overline{\text{CE}}$	Chip enable input. A high to low transition on this pin latches addresses and initiates an access cycle.
$\overline{\text{OE}}$	Output enable input . During a read cycle this input turns on the data output buffers.
$\overline{\text{WE}}$	Write enable input. Data are latched into the device on the leading (falling) edge of this signal.
↓RDY (2186 only)	This output becomes active to signify a delay in the cycle. It is open drain, allowing the wire ORing of several 2186 RDY outputs.
↓$\overline{\text{REFEN}}$ (2187 only)	This input allows for external refresh control. A high to low transition of $\overline{\text{REFEN}}$ causes a refresh cycle to be initiated. Holding $\overline{\text{REFEN}}$ low causes asynchronous timer operation.
V_{CC}	This input supplies operating voltage to the device. V_{CC} is specified at 5 V ±10%.
GND	Ground input for the device.

After $\overline{\text{REFEN}}$ returns high from this state, a minimum amount of time (TRFHEL) must be allowed before the next falling edge of $\overline{\text{CE}}$ or $\overline{\text{REFEN}}$ in order to complete any refresh cycles initiated by the timer. This mode of operation is called power-down refresh. The 2187 also supports a single-step mode of operation, which is accomplished by strobing $\overline{\text{REFEN}}$ low after an access cycle is started and then holding it low. $\overline{\text{REFEN}}$ can be kept low indefinitely and data integrity guaranteed. With $\overline{\text{REFEN}}$ held low, data remain valid on the I/O pins as long as $\overline{\text{OE}}$ remains active, even while refresh cycles are being performed.

Three requirements must be kept in mind when building microprocessor memory systems with iRAMs. The first is the need for a stable $\overline{\text{CE}}$ because the active low transition of $\overline{\text{CE}}$ latches addresses into the iRAM. Also, there is a minimum specification between transitions of $\overline{\text{CE}}$ (TEHEL) to allow for proper precharge of internal dynamic circuitry. The second requirement is the need for valid data at the memory device when the $\overline{\text{WE}}$ line is activated. This is a necessity since the iRAMs write data into the array on the leading edge of the write pulse. The third consideration is compatibility with SRAMs. In particular, the design should allow for the trailing edge write of SRAMs. This permits using SRAMs as second source chips, or allows the iRAM to replace read only memory (ROM) or erasable programmable read only memory (EPROM) during system debug stages.

Following are applications that exemplify the techniques for interfacing the asynchronous 2186 and the synchronous 2187 iRAMs to various microprocessors. Although the designs can be simpler, additional circuitry is included to provide memory site compatibility with SRAMs and EPROMs.

5-MHz 8088 processor application

The first example involves a 5-MHz 8088 microprocessor configured to a bank of 2186 iRAMs. It runs in the maximum mode without wait states for normal memory access cycles, except when RDY is activated due to internal refresh. The schematic diagram is shown in Fig 1. The iRAM chip enable circuit is a simple cross-coupled latch that provides an active low enable signal ($\overline{\text{E}}$) synchronized with address latch enable (ALE). This enable signal, along with latched status bit S2, is used to enable the 74S138 address decoder to provide stable chip enable signals ($\overline{\text{CE}}$0 to $\overline{\text{CE}}$7) to the 2186 iRAMs. The memory write control ($\overline{\text{MWTC}}$) output from the 8288 bus controller provides for both leading and trailing edge write conditions.

The basic operation of the $\overline{\text{CE}}$ circuit is as follows: early in the CPU cycle, ALE goes high, then low, clearing the cross-coupled latch and driving $\overline{\text{E}}$ high. $\overline{\text{E}}$ remains

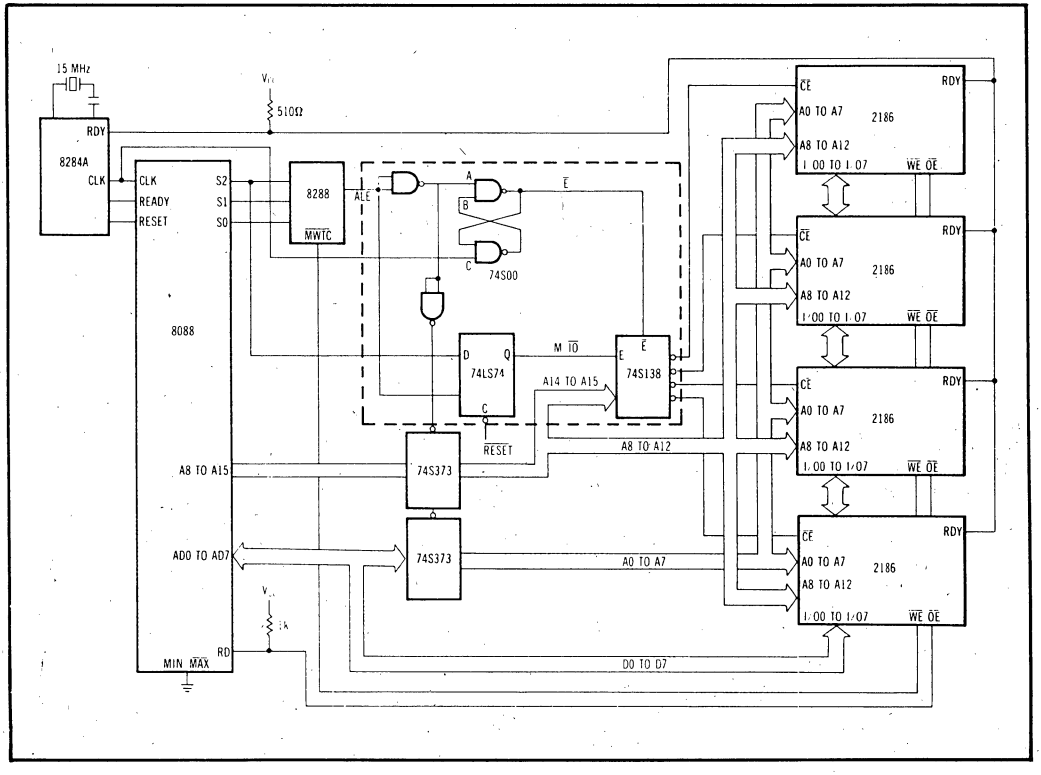


Fig 1 Schematic for iRAM use with 8088 microprocessing unit running in maximum mode. To ensure proper startup, all iRAM control inputs must be inactive for 100 μs after V_{CC} reaches spec.

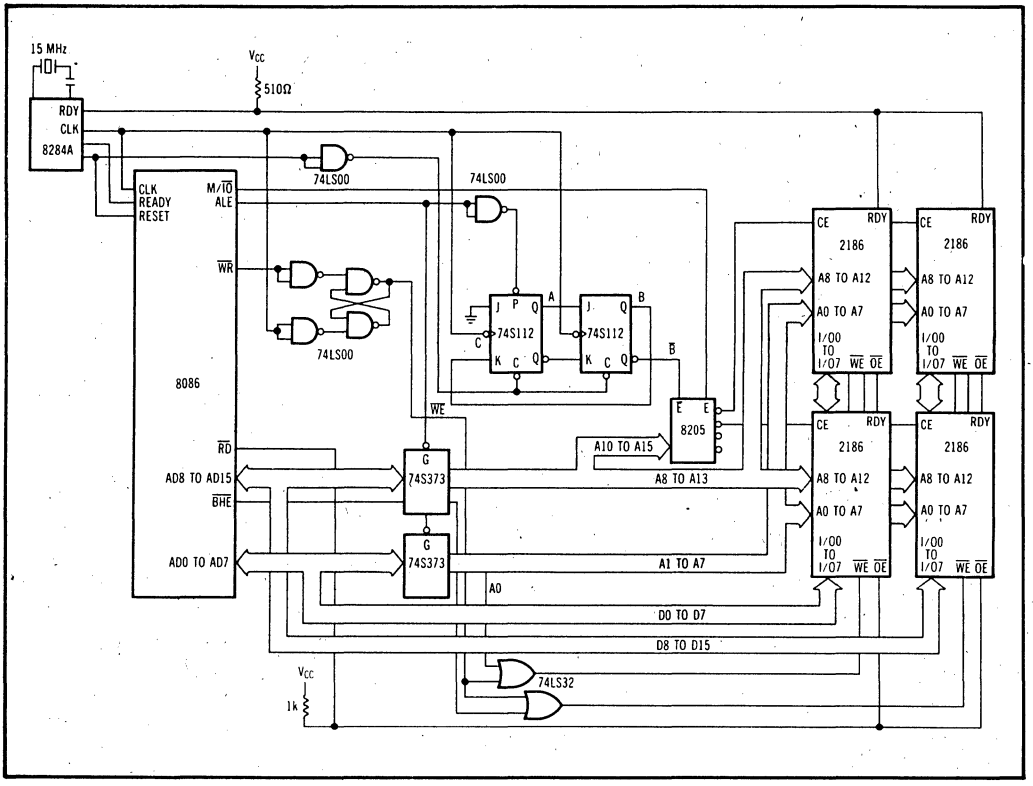


Fig 2 Schematic for iRAM use with 8086 microprocessing unit operation in minimum mode. NAND latch in processor's $\overline{WR}$ line ensures adequate delay of data to iRAM as well as supplying trailing edge WR signal required by conventional SRAMs.

high at least as long as ALE holds the clear condition of the latch. Due to the skew of the falling edge of ALE with the system clock, two possible decoder enable timing sequences exist. The first occurs when ALE returns to a low state before the falling edge of the system clock. In this case, the latch remains cleared and $\overline{E}$ remains high. On the falling edge of the system clock, the latch is set, driving $\overline{E}$ low and enabling the decoder (depending on the state of S2). The other timing sequence occurs when ALE goes low after the falling edge of the system clock. In this case, when ALE goes low, the $\overline{E}$ signal is immediately driven low as a result of the system clock (low) input on the "set" side of the latch. The memory cycle completes and early into the next cycle when ALE goes high, $\overline{E}$ returns to a high state. This clearing action occurs independently of the state of the clock.

The net result is the enabling of the 74S138 decoder after its address inputs have stabilized. Thus, a transient-free chip enable is supplied to the iRAMs. The balance of the memory system is wired in a straightforward manner. The $\overline{OE}$ signal for the memory array is connected directly to the read control ($\overline{RD}$) signal of the 8088 processor. Bidirectional data lines of the memories are connected to the processor's AD0 to AD7 lines. CPU addresses are latched by the 74S373s two gate delays after the falling edge of ALE. Although the iRAMs do not require the address latches, they are included for completeness. A typical system needs to latch the addresses from the multiplexed bus for interfacing to SRAMs, EPROMs, and so on. The address lines feed the memory array via the 74S373 flow-through latches. All of the RDY lines of the memory array connect to a RDY input of the 8284A clock generator in an OR configuration. A 510-Ω pullup resistor is used for stability.

Note that S2 is latched into a 74LS74 flipflop on the rising edge of ALE. This is important because, during certain CPU operations such as the execution of the halt instruction, the status bits are not guaranteed to remain valid until the falling edge of ALE. To guarantee proper power-up of the 2186, all control inputs must remain inactive for 100 μs after V_{CC} reaches specification. This is accomplished by tying RESET to the clear input of the M/IO flipflop. A pullup resistor on the $\overline{OE}$ is also required because the RD line on the 8088 goes into a high impedence state during RESET.

iRAMs in an 8086 based system

The interface requirements for an 8086/2186 system (Fig 2) are similar to those for an 8088. The $\overline{CE}$ generation circuitry consists of two JK flipflops, arranged as a 4-state sequencer. This sequencer makes its first transition on the rising edge of ALE, when sequencer output A is set. On the next falling edge of the system clock, sequencer output B is set, enabling the 8205 decoder if the cycle is a

memory access ($\overline{M}$/IO also gates the decoder). After one more clock cycle, output A is reset, causing output B to be reset on the fourth clock cycle. When output B is reset, the $\overline{CE}$ decoder is disabled, causing the $\overline{CE}$ to the 2186 to return to a high state. With this arrangement, $\overline{CE}$ is low for only two processor T states, allowing a $\overline{CE}$ high time of at least two T states. This relatively long deselect time is important in 16-bit systems where a 2186 could receive a $\overline{CE}$ but no $\overline{OE}$ or $\overline{WE}$. This preceding condition would cause a false memory cycle to occur, in which case an extended $\overline{CE}$ high time is required because TEHELF is greater than TEHEL.

For an 8086 operating in minimum mode, data output during a write cycle is not guaranteed to be valid at the falling edge of WR. To satisfy the leading edge write requirement of the 2186, the leading edge of $\overline{WR}$ needs to be delayed until data out is valid. This can easily be accomplished through the use of a cross-coupled NAND latch, which also provides the trailing edge write needed for SRAMs. This delayed $\overline{WE}$ is then steered to either one or both devices in the 16-bit word by ORing $\overline{WE}$ with either AO or bus high enable ($\overline{BHE}$). This allows for either word or byte writes. If wait states are needed, the RDY outputs of the 2186 can be routed back to the RDY input of the 8284A clock chip. Because the RDY outputs are open drain, a 510-Ω pullup resistor is required.

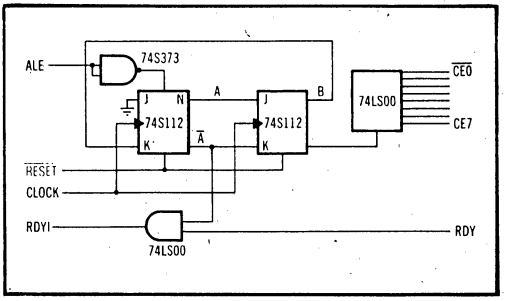


Fig 3 Additional circuitry required for 10-MHz 8086 microprocessing unit/iRAM operation with one wait state.

By adding a single AND gate to the 8086 interface (Fig 3) 2186 can be run in one wait state 10-MHz 8086 system. Upon going high, ALE causes the flipflop driving A to be set, forcing $\overline{A}$ low. After a 1-gate delay, RDYI is asserted. On the next falling edge of clock after ALE goes high, the flipflop driving A can now be reset on the next falling edge of clock, which occurs at the end of T2.

This arrangement ensures that RDYI will remain low for the time frame required to insert at least one wait state. If the 2186 responds to this access with a RDY low

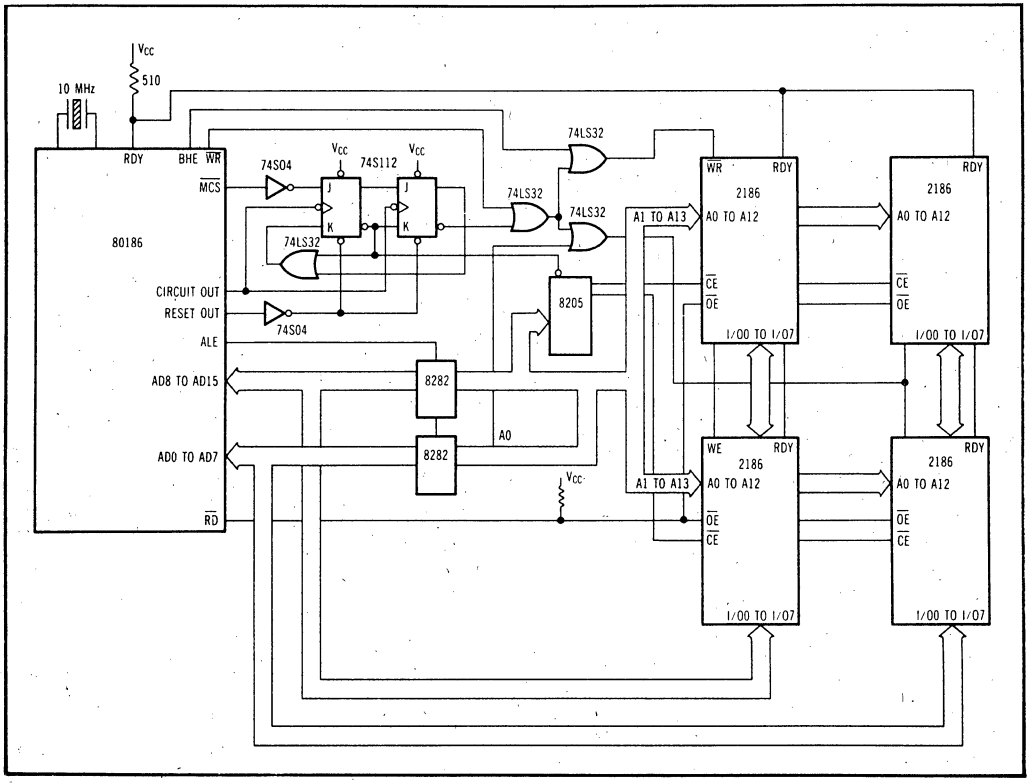


Fig 4 Interface circuitry required for iRAM operation with 5-MHz 80186 microprocessing unit. This system, based on popular iAPX 86 family of components, provides users with programmable memory chip selects for blocked memory applications.

condition, the trailing edge of RDYI will be pushed out to insert more wait states as needed. The number of wait states that can be inserted will never exceed six—assuming a maximum RDY low time of 575 ns.

Fig 4 shows the circuitry required to interface the 2186 with Intel's 80186 high integration 16-bit processor. The 80186 is an integration of 10 common iAPX 86 components, including an enhanced 8086-2 CPU, a clock generator, a local bus controller, and an interrupt controller. The clock speed of this configuration is 5 MHz. The 80186's bus structure is much like that of the 8086, but there are some useful additions, including programmable memory chip select (MCS) outputs. These outputs can be programmed to become active for a user-selected block of memory. In Fig 4, one of four available mid-range MCSs would be programmed to become active in the memory space allocated to the iRAM. This MCS signal is then used to initiate a count sequence with the two JK flipflops. The two flipflops' outputs provide both the $\overline{CE}$ and the properly positioned $\overline{WE}$ (for leading edge writes) to the iRAMs.

FMCs, in which the iRAM receives a $\overline{CE}$ but no $\overline{OE}$ or $\overline{WE}$, will occur any time a single byte write occurs. If this is the case, the $\overline{CE}$ high time requirement becomes somewhat longer than that required for a read or write cycle. The $\overline{CE}$ circuitry used was selected with this consideration in mind because of the long $\overline{CE}$ high time it guarantees.

Synchronous iRAMs in an 8088 system

One way to create a synchronous design using the 8088, operating in maximum mode with the 2187 iRAM, is to use a status decoder to generate a refresh signal during an opcode fetch cycle. The following example assumes that the 2187 iRAM is used for data store only, so that the iRAM can be sent a refresh strobe during the time the 8088 is executing an instruction fetch from some other portion of memory (ie, EPROM, ROM). It also assumes that opcode fetches occur often enough to meet the 128 refreshes/2-ms refresh specification.

In the circuit shown in Fig 5, it is evident that the 2187 and the 8088 are interconnected with a minimal amount

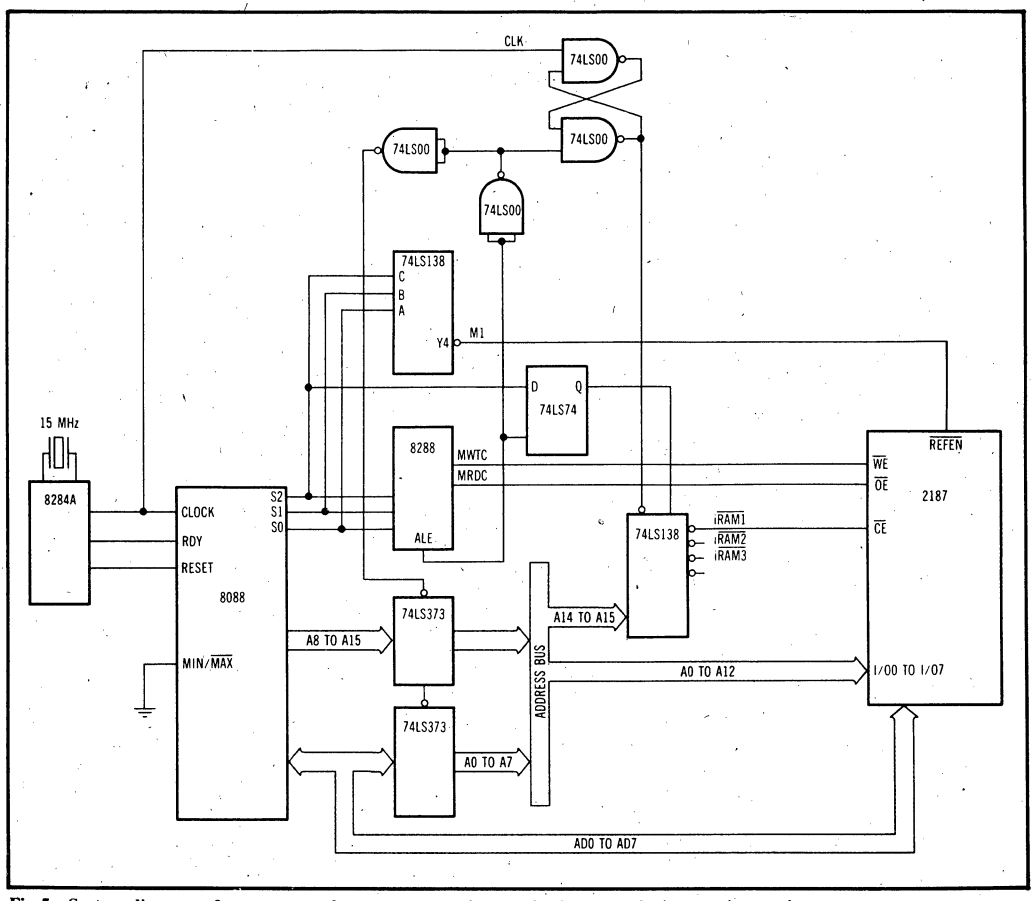


Fig 5 System diagram of 2187/8088 synchronous system. iRAM refresh occurs during 8088 instruction fetch cycle and is generated by external status decoder. Minimal transistor-transistor logic interface circuitry is required.

Combining memory and control together on a single piece of silicon... satisfies all the requirements for microprocessor memory.

of transistor-transistor logic interface circuitry. The decoded 8088 signal M1 is connected to the $\overline{\text{REFEN}}$ input of the iRAM. This connection generates a refresh strobe to the 2187 every time the 8088 performs an opcode fetch. The 8288 bus controller generates memory read commands and memory write commands that are properly timed for all memory requirements. Thus, these signals may connect directly to the $\overline{\text{WE}}$ and $\overline{\text{OE}}$ control lines of the iRAM without special conditioning circuitry. The 8-bit multiplexed address-data bus of the 8088 is connected directly to the I/O 0 to 7 lines of the iRAM. The low order addresses A0 to A7 are latched by ALE at the 74LS373 latch and , together with lines A8 to A12, form the iRAM device address.

The M/IO signal is status bit S2. It is latched by ALE and is used as one of the enable inputs to the iRAM chip enable decoder. The other decoder enable input is synchronized to ALE, providing a properly timed signal that ensures a stable $\overline{\text{CE}}$ to the iRAM. The decoder is disabled on the rising edge of ALE. However, due to the skew of the falling edge of ALE and the system clock, two slightly different enable timing sequences can occur. If ALE goes low at or after the falling edge of the clock, the $\overline{\text{E}}$ enable line is immediately activated and enables the decoder. Note that when RESET is low, REFEN will be forced low. This guarantees proper 2187 power-up. This circuit runs at 5 MHz without wait states.

Combining memory and control together on a single piece of silicon, the iRAM satisfies all the requirements for microprocessor memory. Its ease of use, flexibility, and low cost make it an attractive memory alternative. By using it in some of the applications demonstrated, engineers will find that the iRAM can help them meet the majority of both present and future memory system design needs.

2114A
1024 X 4 BIT STATIC RAM

	2114AL-1	2114AL-2	2114AL-3	2114AL-4	2114A-4	2114A-5
Max. Access Time (ns)	100	120	150	200	200	250
Max. Current (mA)	40	40	40	40	70	70

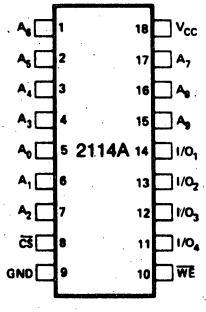
- **HMOS Technology**
- **Low Power, High Speed**
- **Identical Cycle and Access Times**
- **Single +5V Supply ±10%**
- **High Density 18 Pin Package**
- **Completely Static Memory - No Clock or Timing Strobe Required**
- **Directly TTL Compatible: All Inputs and Outputs**
- **Common Data Input and Output Using Three-State Outputs**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

The Intel® 2114A is a 4096-bit static Random Access Memory organized as 1024 words by 4-bits using HMOS, a high performance MOS technology. It uses fully DC stable (static) circuitry throughout, in both the array and the decoding, therefore it requires no clocks or refreshing to operate. Data access is particularly simple since address setup times are not required. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided.

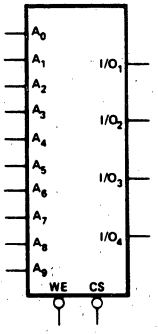
The 2114A is designed for memory applications where the high performance and high reliability of HMOS, low cost, large bit storage, and simple interfacing are important design objectives. The 2114A is placed in an 18-pin package for the highest possible density.

It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. A separate Chip Select ($\overline{CS}$) lead allows easy selection of an individual package when outputs are or-tied.

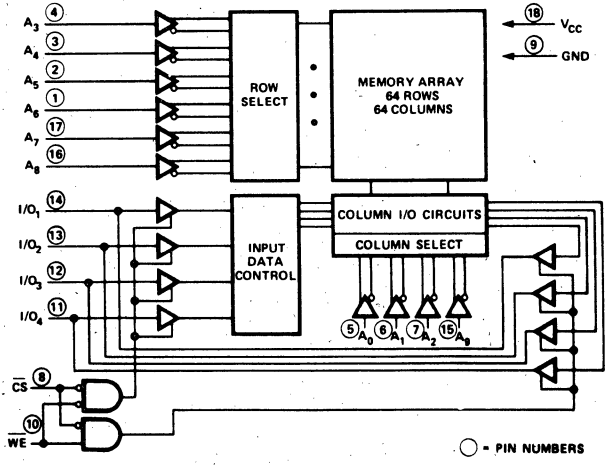
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

A_0–A_9	ADDRESS INPUTS	V_{CC}	POWER (+5V)
$\overline{WE}$	WRITE ENABLE	GND	GROUND
$\overline{CS}$	CHIP SELECT		
I/O_1–I/O_4	DATA INPUT/OUTPUT		

DECEMBER, 1979

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... -10°C to 80°C
Storage Temperature ..... -65°C to 150°C
Voltage on any Pin
With Respect to Ground ..... -3.5V to +7V
Power Dissipation 1.0W
D.C. Output Current 5mA

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, unless otherwise noted.

SYMBOL	PARAMETER	2114AL-1/L-2/L-3/L-4 Min.	Typ.[1]	Max.	2114A-4/-5 Min.	Typ.[1]	Max.	UNIT	CONDITIONS
$\|I_{LI}\|$	Input Load Current (All Input Pins)		.01	1			1	μA	V_{IN} = 0 to 5.5V
$\|I_{LO}\|$	I/O Leakage Current		.1	10			10	μA	$\overline{CS}$ = V_{IH} $V_{I/O}$ = 0 to 5.5
I_{CC}	Power Supply Current		25	40		50	70	mA	V_{CC} = max, $I_{I/O}$ = 0 mA, T_A = 0°C
V_{IL}	Input Low Voltage	-3.0		0.8	-3.0		0.8	V	
V_{IH}	Input High Voltage	2.0		6.0	2.0		6.0	V	
I_{OL}	Output Low Current	4.0	9.0		4.0	9.0		mA	V_{OL} = 0.4V
I_{OH}	Output High Current	-2.0	-2.5		-2.0	-2.5		mA	V_{OH} = 2.4V
I_{OS}[2]	Output Short Circuit Current			40			40	mA	V_{OUT} = GND

NOTE: 1. Typical values are for T_A = 25°C and V_{CC} = 5.0V.
2. Duration not to exceed 1 second.

CAPACITANCE

T_A = 25°C, f = 1.0 MHz

SYMBOL	TEST	MAX	UNIT	CONDITIONS
$C_{I/O}$	Input/Output Capacitance	5	pF	$V_{I/O}$ = OV
C_{IN}	Input Capacitance	5	pF	V_{IN} = OV

NOTE: This parameter is periodically sampled and not 100% tested.

A.C. CONDITIONS OF TEST

Input Pulse Levels ..... 0.8 Volt to 2.0 Volt
Input Rise and Fall Times ..... 10 nsec
Input and Output Timing Levels ..... 0.8 Volts to 2.0 Volts
Output Load ..... 1 TTL Gate and C_L = 100 pF

LOAD FOR T_{OTD} AND T_{OTW}

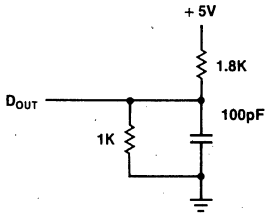


Figure 1.

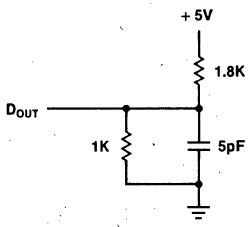


Figure 2.

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, unless otherwise noted.

READ CYCLE [1]

SYMBOL	PARAMETER	2114AL-1 Min.	2114AL-1 Max.	2114AL-2 Min.	2114AL-2 Max.	2114AL-3 Min.	2114AL-3 Max.	2114A-4/L-4 Min.	2114A-4/L-4 Max.	2114A-5 Min.	2114A-5 Max.	UNIT
t_{RC}	Read Cycle Time	100		120		150		200		250		ns
t_A	Access Time		100		120		150		200		250	ns
t_{CO}	Chip Selection to Output Valid		70		70		70		70		85	ns
t_{CX} (3)	Chip Selection to Output Active	10		10		10		10		10		ns
t_{OTD} (3)	Output 3-state from Deselection		30		35		40		50		60	ns
t_{OHA}	Output Hold from Address Change	15		15		15		15		15		ns

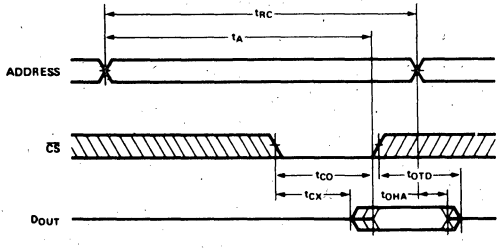
WRITE CYCLE [2]

SYMBOL	PARAMETER	2114AL-1 Min.	2114AL-1 Max.	2114AL-2 Min.	2114AL-2 Max.	2114AL-3 Min.	2114AL-3 Max.	2114A-4/L-4 Min.	2114A-4/L-4 Max.	2114A-5 Min.	2114A-5 Max.	UNIT
t_{WC}	Write Cycle Time	100		120		150		200		250		ns
t_W	Write Time	75		75		90		120		135		ns
t_{WR}	Write Release Time	0		0		0		0		0		ns
t_{OTW} (3)	Output 3-state from Write		30		35		40		50		60	ns
t_{DW}	Data to Write Time Overlap	70		70		90		120		135		ns
t_{DH}	Data Hold from Write Time	0		0		0		0		0		ns

NOTES:
1. A Read occurs during the overlap of a low $\overline{CS}$ and a high $\overline{WE}$.
2. A Write occurs during the overlap of a low $\overline{CS}$ and a low $\overline{WE}$. t_W is measured from the latter of $\overline{CS}$ or $\overline{WE}$ going low to the earlier of $\overline{CS}$ or $\overline{WE}$ going high.
3. Measured at ±500 mV with 1 TTL Gate and C_L = 500 pF.

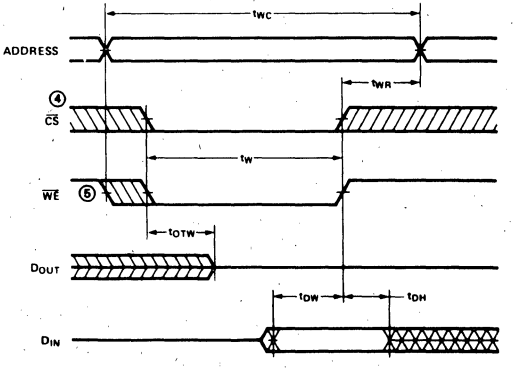
WAVEFORMS

READ CYCLE ③



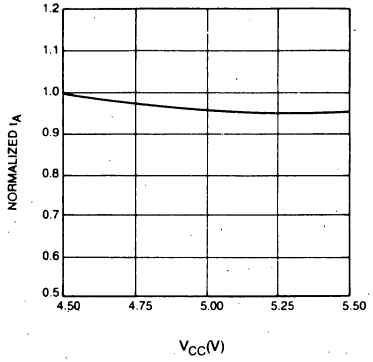
NOTES:
3. $\overline{WE}$ is high for a Read Cycle.
4. If the $\overline{CS}$ low transition occurs simultaneously with the $\overline{WE}$ low transition, the output buffers remain in a high impedance state.
5. $\overline{WE}$ must be high during all address transitions.

WRITE CYCLE

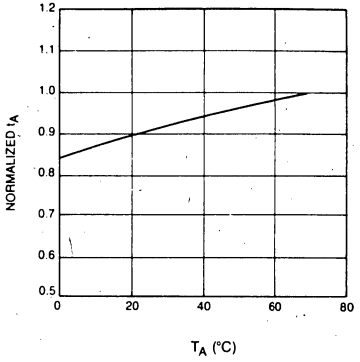


TYPICAL D.C. AND A.C. CHARACTERISTICS

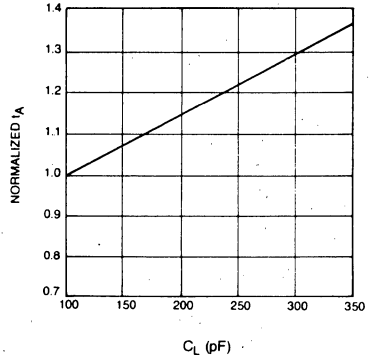
NORMALIZED ACCESS TIME VS. SUPPLY VOLTAGE



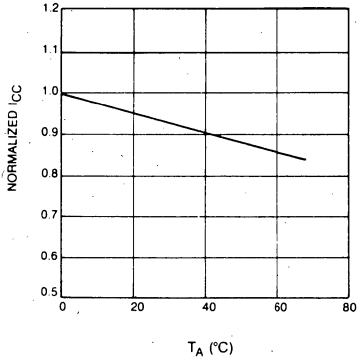
NORMALIZED ACCESS TIME VS. AMBIENT TEMPERATURE



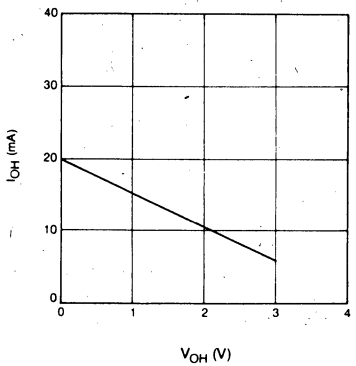
NORMALIZED ACCESS TIME VS. OUTPUT LOAD CAPACITANCE



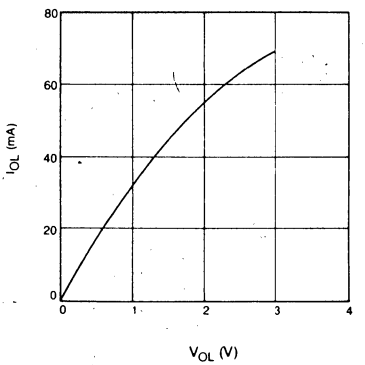
NORMALIZED POWER SUPPLY CURRENT VS. AMBIENT TEMPERATURE



OUTPUT SOURCE CURRENT VS. OUTPUT VOLTAGE



OUTPUT SINK CURRENT VS. OUTPUT VOLTAGE



2115A, 2125A FAMILY
HIGH SPEED 1K X 1 BIT STATIC RAM

	2115AL, 2125AL	2115A, 2125A	2115AL-2, 2125AL-2	2115A-2, 2125A-2
Max. T_{AA} (ns)	45	45	70	70
Max. I_{CC} (mA)	75	125	75	125

- **Pin Compatible To 93415A (2115A) And 93425A (2125A)**
- **Fan-Out Of 10 TTL (2115A Family) -- 16mA Output Sink Current**
- **Low Operating Power Dissipation --Max. 0.39mW/Bit (2115AL, 2125AL)**
- **TTL Inputs And Outputs**
- **Available in EXPRESS — Standard Temperature Range — Extended Temperature Range**
- **Uncommitted Collector (2115A) And Three-State (2125A) Output**
- **Standard 16-Pin Dual In-Line Package**

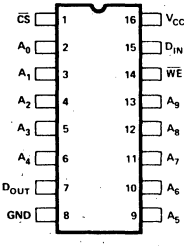
The Intel® 2115A and 2125A families are high-speed, 1024 words by 1 bit random access memories. Both open collector (2115A) and three-state output (2125A) are available. The 2115A and 2125A use fully DC stable (static) circuitry throughout — in both the array and the decoding and, therefore, require no clocks or refreshing to operate. The data is read out non-destructively and has the same polarity as the input data.

The 2115AL/2125AL at 45 ns maximum access time and the 2115AL-2/2125AL-2 at 70 ns maximum access time are fully compatible with the industry-produced 1K bipolar RAMs, yet offer a 50% reduction in power of their bipolar equivalents. The power dissipation of the 2115AL/2125AL and 2115AL-2/2125AL-2 is 394 mW maximum as compared to 814 mW maximum of their bipolar equivalents. For systems already designed for 1K bipolar RAMs, the 2115A/2125A and the 2115A-2/2125A-2 at 45 ns and 70 ns maximum access times, respectively, offer complete compatibility with a 20% reduction in maximum power dissipation.

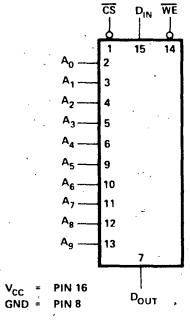
The devices are directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. A separate select ($\overline{CS}$) lead allows easy selection of an individual package when outputs are OR-tied.

The 2115A and 2125A families are fabricated with Intel's N-channel MOS Silicon Gate Technology.

PIN CONFIGURATION



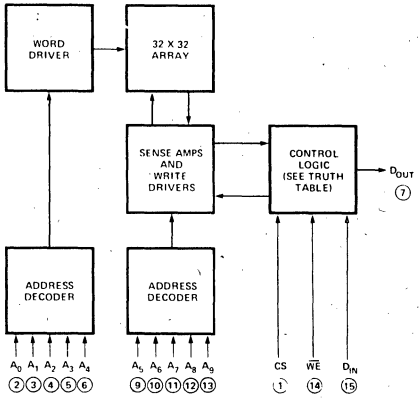
LOGIC SYMBOL



PIN NAMES

$\overline{CS}$	CHIP SELECT
A_0 TO A_9	ADDRESS INPUTS
$\overline{WE}$	WRITE ENABLE
D_{IN}	DATA INPUT
D_{OUT}	DATA OUTPUT

BLOCK DIAGRAM



TRUTH TABLE

INPUTS			OUTPUT 2115A FAMILY	OUTPUT 2125A FAMILY	MODE
$\overline{CS}$	$\overline{WE}$	D_{IN}	D_{OUT}	D_{OUT}	
H	X	X	H	HIGH Z	NOT SELECTED
L	L	L	H	HIGH Z	WRITE "0"
L	L	H	H	HIGH Z	WRITE "1"
L	H	X	D_{OUT}	D_{OUT}	READ

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . . –10°C to +85°C
Storage Temperature . . . . . –65°C to +150°C
All Output or Supply Voltages . . . . . –0.5V to +7V
All Input Voltages . . . . . –0.5V to +5.5V
D.C. Output Current . . . . . 20 mA

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS[1,2]

V_{CC} = 5V ±5%, T_A = 0°C to 75°C

Symbol	Test	Min.	Typ.	Max.	Unit	Conditions
V_{OL1}	2115A Family Output Low Voltage			0.45	V	I_{OL} = 16 mA
V_{OL2}	2125A Family Output Low Voltage			0.45	V	I_{OL} = 7 mA
V_{IH}	Input High Voltage	2.1			V	
V_{IL}	Input Low Voltage			0.8	V	
I_{IL}	Input Low Current		–0.1	–40	μA	V_{CC} = Max., V_{IN} = 0.4V
I_{IH}	Input High Current		0.1	40	μA	V_{CC} = Max., V_{IN} = 4.5V
$\|I_{CEX}\|$	2115A Family Output Leakage Current		0.1	100	μA	V_{CC} = Max., V_{OUT} = 4.5V
$\|I_{OFF}\|$	2125A Family Output Current (High Z)		0.1	50	μA	V_{CC} = Max., V_{OUT} = 0.5V/2.4V
I_{OS}[3]	2125A Family Current Short Circuit to Ground			–100	mA	V_{CC} = Max.
V_{OH}	Family Output High Voltage	2.4			V	I_{OH} = –3.2 mA
I_{CC}	Power Supply Current: I_{CC1}: 2115AL, 2115AL-2, 2125AL, 2125AL-2		60	75	mA	All Inputs Grounded, Output Open
	I_{CC2}: 2115A, 2115A-2, 2125A, 2125A-2		100	125	mA	

NOTES:

1. The operating ambient temperature ranges are guaranteed with transverse air flow exceeding 400 linear feet per minute and a two minute warm-up. Typical thermal resistance values of the package at maximum temperature are:

   θ_{JA} (@ 400 f_{PM} air flow) = 45°C/W
   θ_{JA} (still air) = 60°C/W
   θ_{JC} = 25°C/W

2. Typical limits are at V_{CC} = 5V, T_A = +25°C, and maximum loading.
3. Duration of short circuit current should not exceed 1 second.

2115A FAMILY A.C. CHARACTERISTICS[1,2] V_{CC} = 5V ±5%, T_A = 0°C to 75°C

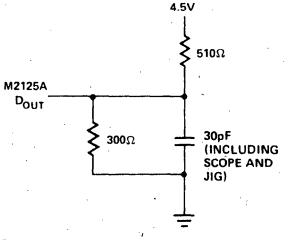
READ CYCLE

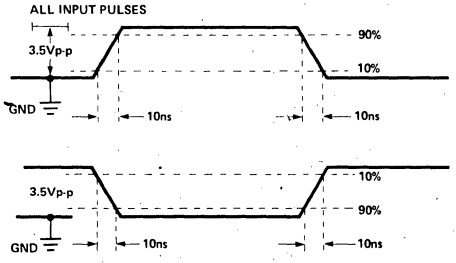
Symbol	Test	2115AL Limits Min.	Typ.	Max.	2115A Limits Min.	Typ.	Max.	2115AL-2 Limits Min.	Typ.	Max.	2115A-2 Limits Min.	Typ.	Max.	Units
t_{ACS}	Chip Select Time	5	15	30	5	15	30	5	15	30	5	15	40	ns
t_{RCS}	Chip Select Recovery Time		10	30		10	30		10	30		10	40	ns
t_{AA}	**Address Access Time**		**30**	**45**		**30**	**45**		**40**	**70**		**40**	**70**	**ns**
t_{OH}	Previous Read Data Valid After Change of Address	10			10			10			10			ns

WRITE CYCLE

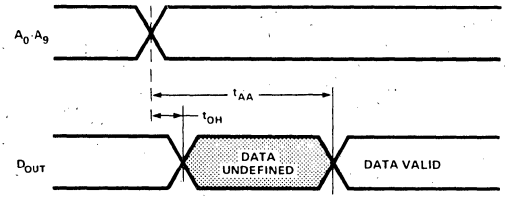
Symbol	Test	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Units
t_{WS}	Write Enable Time		10	25		10	30		10	25		10	40	
t_{WR}	Write Recovery Time	0		25	0		30	0		25	0		45	ns
t_W	Write Pulse Width	30	20		30	10		30	15		50	15		ns
t_{WSD}	Data Set-Up Time Prior to Write	0	−5		5	−5		0	−5		5	−5		ns
t_{WHD}	Data Hold Time After Write	5	0		5	0		5	0		5	0		ns
t_{WSA}	Address Set-Up Time	5	0		5	0		5	0		15	0		ns
t_{WHA}	Address Hold Time	5	0		5	0		5	0		5	0		ns
t_{WSCS}	Chip Select Set-Up Time	5	0		5	0		5	0		5	0		ns
t_{WHCS}	Chip Select Hold Time	5	0		5	0		5	0		5	0		ns

A.C. TEST CONDITIONS

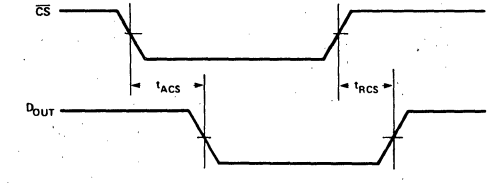




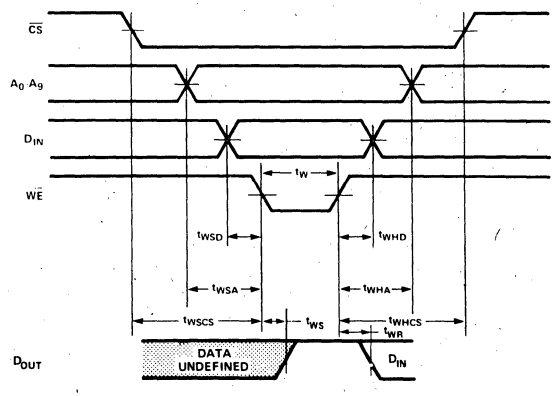
READ CYCLE



PROPAGATION DELAY FROM CHIP SELECT



WRITE CYCLE



(ALL ABOVE MEASUREMENTS REFERENCED TO 1.5V)

2125 FAMILY A.C. CHARACTERISTICS[1,2] V_{CC} = 5V ±5%, T_A = 0°C to 75°C

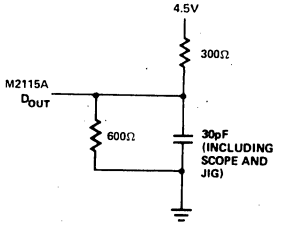
READ CYCLE

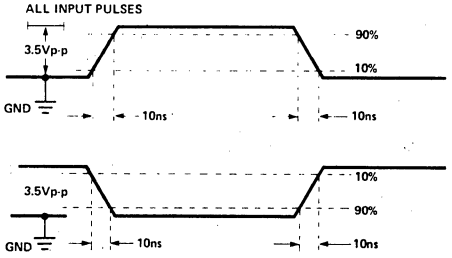
Symbol	Test	2125AL Limits Min.	Typ.	Max.	2125A Limits Min.	Typ.	Max.	2125AL-2 Limits Min.	Typ.	Max.	2125A-2 Limits Min.	Typ.	Max.	Units
t_{ACS}	Chip Select Time	5	15	30	5	15	30	5	15	30	5	15	40	ns
t_{ZRCS}	Chip Select to HIGH Z		10	30		10	30		10	30		10	40	ns
t_{AA}	**Address Access Time**		**30**	**45**		**30**	**45**		**40**	**70**		**40**	**70**	**ns**
t_{OH}	Previous Read Data Valid After Change of Address	10			10			10			10			ns

WRITE CYCLE

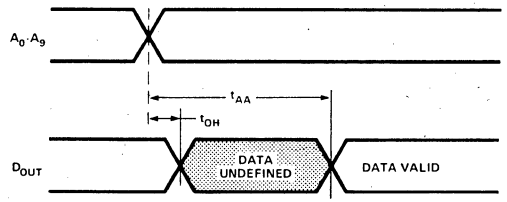
Symbol	Test	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Units
t_{ZWS}	Write Enable to HIGH Z		10	25		10	30		10	25		10	40	ns
t_{WR}	Write Recovery Time	0		25	0		30	0		25	0		45	ns
t_W	Write Pulse Width	30	20		30	10		30	10		50	15		ns
t_{WSD}	Data Set-Up Time Prior to Write	0	−5		5	−5		0	−5		5	−5		ns
t_{WHD}	Data Hold Time After Write	5	0		5	0		5	0		5	0		ns
t_{WSA}	Address Set-Up Time	5	0		5	0		5	0		15	0		ns
t_{WHA}	Address Hold Time	5	0		5	0		5	0		5	0		ns
t_{WSCS}	Chip Select Set-Up Time	5	0		5	0		5	0		5	0		ns
t_{WHCS}	Chip Select Hold Time	5	0		5	0		5	0		5	0		ns

A.C. TEST CONDITIONS

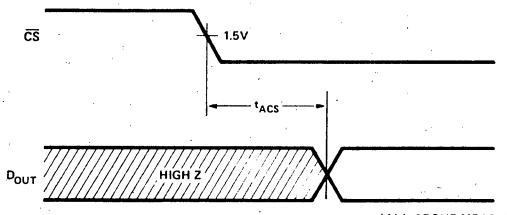




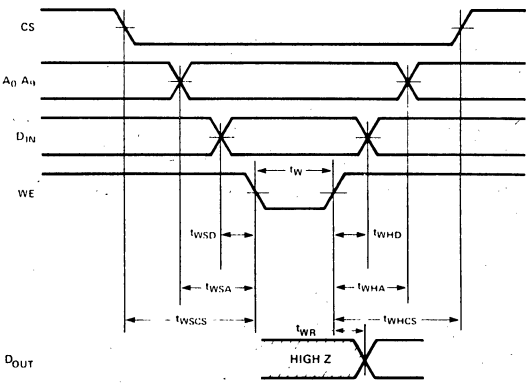
READ CYCLE



PROPAGATION DELAY FROM CHIP SELECT

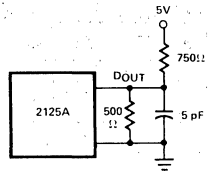


WRITE CYCLE

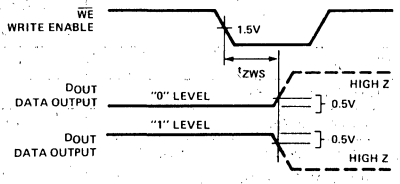


(ALL ABOVE MEASUREMENTS REFERENCED TO 1.5V)

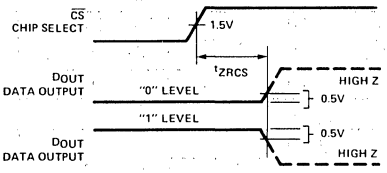
2125A FAMILY WRITE ENABLE TO HIGH Z DELAY



LOAD 1



2125A FAMILY PROPAGATION DELAY FROM CHIP SELECT TO HIGH Z



(ALL t_{ZXXX} PARAMETERS ARE MEASURED AT A DELTA OF 0.5V FROM THE LOGIC LEVEL AND USING LOAD 1.)

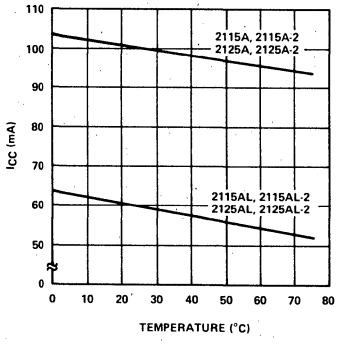
2115A/2125A FAMILY CAPACITANCE* V_{CC}= 5V, f = 1 MHz, T_A = 25°C

SYMBOL	TEST	2115A Family LIMITS		2125A Family LIMITS		UNITS	TEST CONDITIONS
		TYP.	MAX.	TYP.	MAX.		
C_I	Input Capacitance	3	5	3	5	pF	All Inputs = 0V, Output Open
C_O	Output Capacitance	5	8	5	8	pF	$\overline{CS}$ = 5V, All Other Inputs = 0V, Output Open

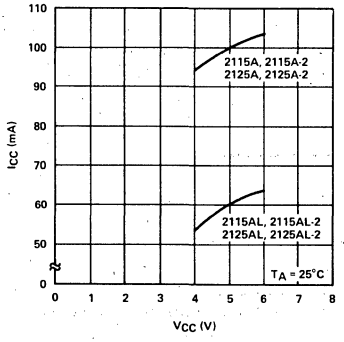
*This parameter is periodically sampled and is not 100% tested.

TYPICAL CHARACTERISTICS

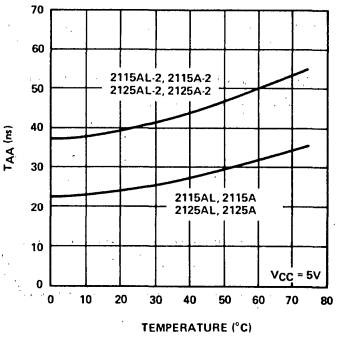
I_{CC} VS. TEMPERATURE



I_{CC} VS. V_{CC}



ACCESS TIME VS. TEMPERATURE



2115H, 2125H FAMILY
HIGH SPEED 1K X 1 BIT STATIC RAM

	2115H-2, 2125H-2	2115H-3, 2125H-3	2115H-4, 2125H-4
Max. T_{AA}(ns)	25	30	35
Max. I_{CC}(mA)	125	125	125

- **HMOS II Technology**
- **Pin Compatible to 93415A (2115H) and 93425A (2125H)**
- **16mA Output Sink Current**
- **Low Operating Power Dissipation — Max. 0.53 mW/Bit (2115H-3, 2125H-3)**
- **Standard 16-Pin Dual In-Line Package**
- **TTL Inputs and Outputs**
- **Single +5V Supply**
- **Uncommitted Collector (2115H) and Three-State (2125H) Output**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

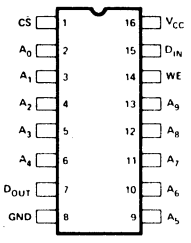
The Intel® 2115H and 2125H families are high speed, 1024 words by 1-bit random access memories fabricated with HMOS II, Intel's advanced N-channel MOS silicon gate technology. Both open collector (2115H) and three-state output (2125H) are available. The 2115H and 2125H use fully DC stable (static) circuitry throughout — in both the array and the decoding and, therefore, require no clocks or refreshing to operate. The data is read out non-destructively and has the same polarity as the input data.

HMOS II's advanced technology allows the production of the industry's fastest, low power, 1K static RAMs — offering access times as low as 25ns.

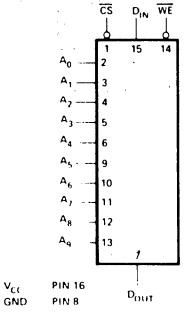
HMOS II allows the production of the 2115H/2125H families, fully compatible with the 1K Bipolar RAMs yet offering substantial reductions in power dissipation. The power dissipations of 525mW maximum and 656mW maximum compared to 814mW maximum offer reductions of 19% and 36% respectively.

The devices are directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. A separate select ($\overline{CS}$) lead allows easy selection of an individual package when outputs are OR-tied.

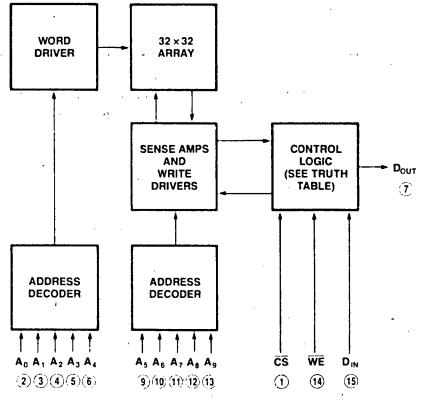
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

$\overline{CS}$	CHIP SELECT
A_0 TO A_9	ADDRESS INPUTS
$\overline{WE}$	WRITE ENABLE
D_{IN}	DATA INPUT
D_{OUT}	DATA OUTPUT

TRUTH TABLE

INPUTS			OUTPUT 2115H FAMILY	OUTPUT 2125H FAMILY	MODE
$\overline{CS}$	$\overline{WE}$	D_{IN}	D_{OUT}	D_{OUT}	
H	X	X	HIGH Z	HIGH Z	NOT SELECTED
L	L	L	HIGH Z	HIGH Z	WRITE "0"
L	L	H	HIGH Z	HIGH Z	WRITE "1"
L	H	X	D_{OUT}	D_{OUT}	READ

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . . –10°C to +85°C
Storage Temperature . . . . . –65°C to +150°C
All Output or Supply Voltages . . . . . –0.5V to +7V
All Input Voltages . . . . . –1.5V to + 7V
D.C. Output Current . . . . . 20 mA

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS [1,2]

V_{CC} = 5V ±5%, T_A = 0°C to 75°C

Symbol	Test	Min.	Typ.	Max.	Unit	Conditions
V_{OL}	2115H/25H Family Output Low Voltage			0.45	V	I_{OL} = 16 mA
V_{IH}	Input High Voltage	2.1			V	
V_{IL}	Input Low Voltage			0.8	V	
I_{IL}	Input Low Current		–0.1	–40	μA	V_{CC} = Max., V_{IN} = 0.4V
I_{IH}	Input High Current		0.1	40	μA	V_{CC} = Max., V_{IN} = 4.5V
$\|I_{CEX}\|$	2115H Family Output Leakage Current		0.1	100	μA	V_{CC} = Max., V_{OUT} = 4.5V
$\|I_{OFF}\|$	2125H Family Output Current (High Z)		0.1	50	μA	V_{CC} = Max., V_{OUT} = 0.5V/2.4V
I_{OS}	2125H Family Current Short Circuit to Ground		125	200	mA	V_{CC} = Max.
V_{OH}	Family Output High Voltage	2.4			V	I_{OH} = –5.2 mA
I_{CC}	Power Supply Current:					All Inputs Grounded, Output Open
	I_{CC2}: 2115H-2/2125H-2 2115H-4/2125H-4		80	125	mA	
	I_{CC3}: 2115H-3/2125H-3		80	125	mA	

NOTES:
1. The operating ambient temperature ranges are guaranteed with transverse air flow exceeding 400 linear feet per minute.
2. Typical limits are at V_{CC} = 5V, T_A = +25°C, and maximum loading.

2115H FAMILY A.C. CHARACTERISTICS V_{CC} = 5V ±5%, T_A = 0°C to 75°C

READ CYCLE

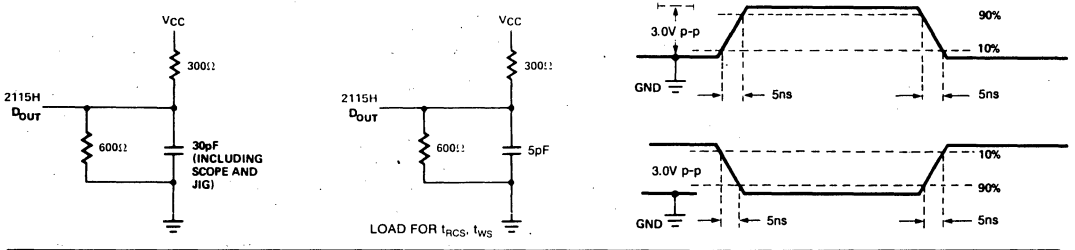
Symbol	Test	2115H-2 Limits Min.	2115H-2 Limits Max.	2115H-3 Limits Min.	2115H-3 Limits Max.	2115H-4 Limits Min.	2115H-4 Limits Max.	Units
t_{ACS}	Chip Select Time		15		20		20	ns
t_{RCS} [1]	Chip Select Recovery Time		20		20		20	ns
t_{AA}	Address Access Time		25		30		35	ns
t_{OH} [1]	Previous Read Data Valid After Change of Address	0		0		0		ns

WRITE CYCLE

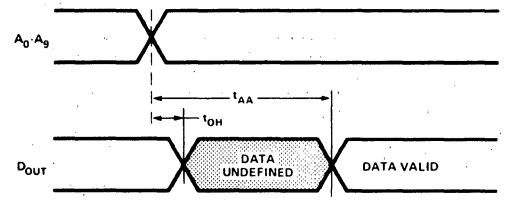
Symbol	Test	Min.	Max.	Min.	Max.	Min.	Max.	Units
t_{WS} [1]	Write Enable Time		15		20		20	ns
t_{WR}	Write Recovery Time	0	15	0	20	0	20	ns
t_W	Write Pulse Width	20		20		25		ns
t_{WSD}	Data Set-Up Time Prior to Write	0		0		0		ns
t_{WHD}	Data Hold Time After Write	0		0		0		ns
t_{WSA}	Address Set-Up Time	5		5		5		ns
t_{WHA}	Address Hold Time	0		0		0		ns
t_{WSCS}	Chip Select Set-Up Time	5		5		5		ns
t_{WHCS}	Chip Select Hold Time	0		5		5		ns

[1] These specifications are guaranteed by design and not production tested.

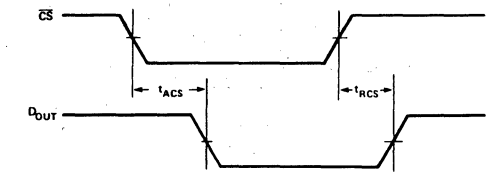
A.C. TEST CONDITIONS



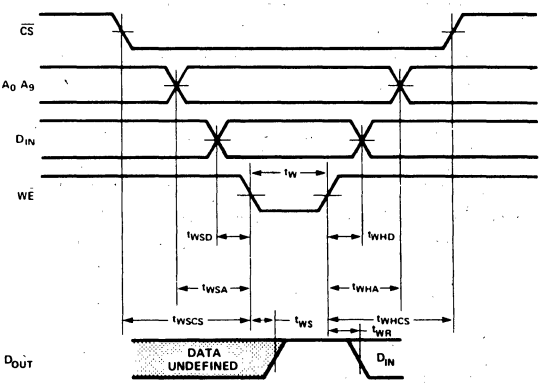
READ CYCLE



PROPAGATION DELAY FROM CHIP SELECT



WRITE CYCLE



(ALL ABOVE MEASUREMENTS REFERENCED TO 1.5V)

2125H FAMILY A.C. CHARACTERISTICS V_{CC} = 5V ±5%, T_A = 0°C to 75°C

READ CYCLE

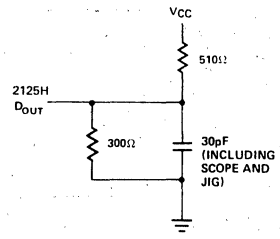
Symbol	Test	2125H-2 Limits Min.	2125H-2 Limits Max.	2125H-3 Limits Min.	2125H-3 Limits Max.	2125H-4 Limits Min.	2125H-4 Limits Max.	Units
t_{ACS}	Chip Select Time		15		20		20	ns
t_{ZRCS} [1]	Chip Select to HIGH Z		20		20		20	ns
t_{AA}	Address Access Time		25		30		35	ns
t_{OH} [1]	Previous Read Data Valid After Change of Address	0		0		0		ns

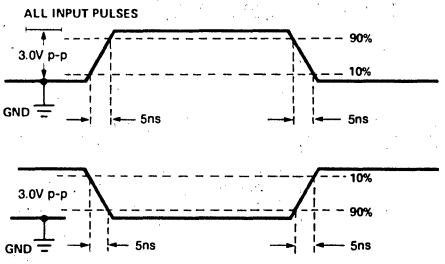
WRITE CYCLE

Symbol	Test	Min.	Max.	Min.	Max.	Min.	Max.	Units
t_{ZWS} [1]	Write Enable to HIGH Z		15		20		20	ns
t_{WR}	Write Recovery Time	0	15	0	20	0	20	ns
t_W	Write Pulse Width	20		20		25		ns
t_{WSD}	Data Set-Up Time Prior to Write	0		0		0		ns
t_{WHD}	Data Hold Time After Write	0		0		0		ns
t_{WSA}	Address Set-Up Time	5		5		5		ns
t_{WHA} [1]	Address Hold Time	0		0		0		ns
t_{WSCS}	Chip Select Set-Up Time	5		5		5		ns
t_{WHCS}	Chip Select Hold Time	0		5		5		ns

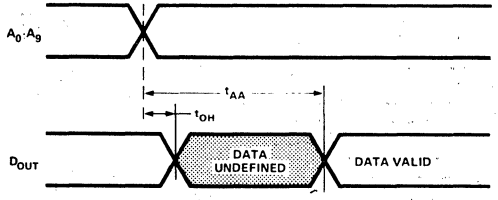
[1] These specifications are guaranteed by design and not production tested.

A.C. TEST CONDITIONS

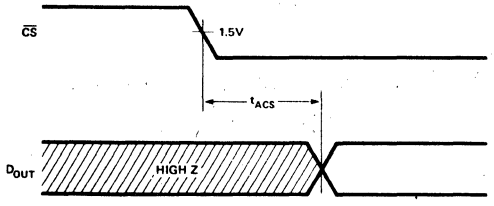




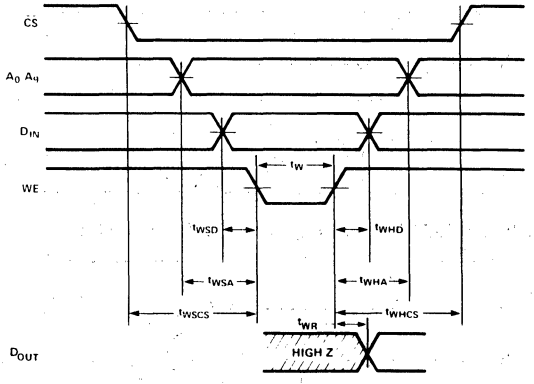
READ CYCLE



PROPAGATION DELAY FROM CHIP SELECT

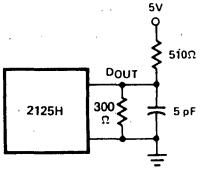


WRITE CYCLE

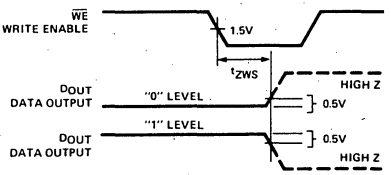


(ALL ABOVE MEASUREMENTS REFERENCED TO 1.5V)

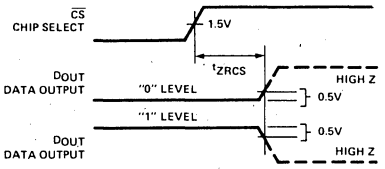
2125H FAMILY WRITE ENABLE TO HIGH Z DELAY



LOAD 1



2125H FAMILY PROPAGATION DELAY FROM CHIP SELECT TO HIGH Z



(ALL t_{ZXXX} PARAMETERS ARE MEASURED AT A DELTA OF 0.5V FROM THE LOGIC LEVEL AND USING LOAD 1.)

2115H/2125H FAMILY CAPACITANCE* V_{CC}= 5V, f = 1 MHz, T_A = 25°C

SYMBOL	TEST	2115H Family LIMITS		2125H Family LIMITS		UNITS	TEST CONDITIONS
		TYP.	MAX.	TYP.	MAX.		
C_I	Input Capacitance	3	5	3	5	pF	All Inputs = 0V, Output Open
C_O	Output Capacitance	5	8	5	8	pF	$\overline{CS}$ = 5V, All Other Inputs = 0V, Output Open

*This parameter is periodically sampled and is not 100% tested.

2147H
HIGH SPEED 4096 x 1 BIT STATIC RAM

	2147H-1	2147H-2	2147H-3	2147H	2147HL
Max. Access Time (ns)	35	45	55	70	70
Max. Active Current (mA)	180	180	180	160	140
Max. Standby Current (mA)	30	30	30	20	10

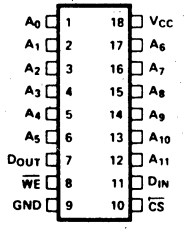
- **Pinout, Function, and Power Compatible to Industry Standard 2147**
- **HMOS II Technology**
- **Completely Static Memory—No Clock or Timing Strobe Required**
- **Equal Access and Cycle Times**
- **Single +5V Supply**
- **0.8-2.0V Output Timing Reference Levels**
- **Direct Performance Upgrade for 2147**
- **Automatic Power-Down**
- **High Density 18-Pin Package**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**
- **Separate Data Input and Output**
- **Three-State Output**

The Intel® 2147H is a 4096-bit static Random Access Memory organized as 4096 words by 1-bit using HMOS-II, Intel's next generation high-performance MOS technology. It uses a uniquely innovative design approach which provides the ease-of-use features associated with non-clocked static memories and the reduced standby power dissipation associated with clocked static memories. To the user this means low standby power dissipation without the need for clocks, address setup and hold times, nor reduced data rates due to cycle times that are longer than access times.

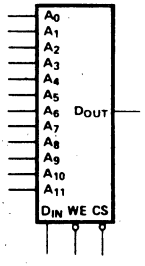
$\overline{CS}$ controls the power-down feature. In less than a cycle time after $\overline{CS}$ goes high—deselecting the 2147H—the part automatically reduces its power requirements and remains in this low power standby mode as long as $\overline{CS}$ remains high. This device feature results in system power savings as great as 85% in larger systems, where the majority of devices are deselected.

The 2147H is placed in an 18-pin package configured with the industry standard 2147 pinout. It is directly TTL compatible in all respects: inputs, output, and a single +5V supply. The data is read out nondestructively and has the same polarity as the input data. A data input and a separate three-state output are used.

PIN CONFIGURATION



LOGIC SYMBOL



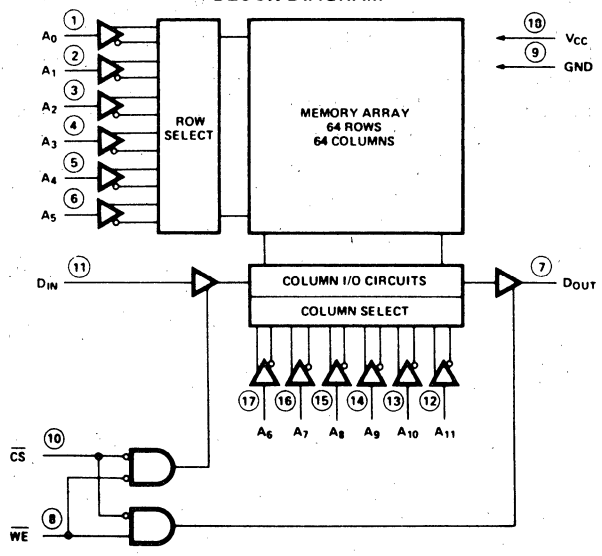
PIN NAMES

A_0–A_{11}	ADDRESS INPUTS	V_{CC}	POWER (+5V)
$\overline{WE}$	WRITE ENABLE	GND	GROUND
$\overline{CS}$	CHIP SELECT		
D_{IN}	DATA INPUT		
D_{OUT}	DATA OUTPUT		

TRUTH TABLE

$\overline{CS}$	$\overline{WE}$	MODE	OUTPUT	POWER
H	X	NOT SELECTED	HIGH Z	STANDBY
L	L	WRITE	HIGH Z	ACTIVE
L	H	READ	D_{OUT}	ACTIVE

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to 85°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
With Respect to Ground ..... −3.5V to +7V
Power Dissipation 1.2W
D.C. Output Current ..... 20 mA

COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS[1]

($T_A = 0°C$ to $70°C$, $V_{CC} = +5V \pm 10\%$, unless otherwise noted.)

Symbol	Parameter	2147H-1, 2, 3 Min.	2147H-1, 2, 3 Typ.	2147H-1, 2, 3 Max.	2147H Min.	2147H Typ.[2]	2147H Max.	2147HL Min.	2147HL Typ.[2]	2147HL Max.	Unit	Test Conditions
I_L	Input Load Current (All Input Pins)		0.01	1.0		0.01	1.0		0.01	1.0	μA	V_{CC} = Max., V_{IN} = GND to V_{CC}
I_{LO}	Output Leakage Current		0.1	50		0.1	50		0.1	50	μA	$\overline{CS} = V_{IH}$, $V_{CC} = 5.5V$, V_{OUT} = GND to 4.5V
I_{CC}	Operating Current		120	170		100	150		100	135	mA	$T_A = 25°C$; V_{CC} = Max., $\overline{CS} = V_{IL}$, Outputs Open
				180			160			140	mA	$T_A = 0°C$
I_{SB}	Standby Current		18	30		12	20		7	10	mA	V_{CC} = Min. to Max., $\overline{CS} = V_{IH}$
I_{PO}[3]	Peak Power-On Current		35	70		25	50		15	30	mA	V_{CC} = GND to V_{CC} Min., $\overline{CS}$ = Lower of V_{CC} or V_{IH} Min.
V_{IL}	Input Low Voltage	−3.0		0.8	−3.0		0.8	−3.0		0.8	V	
V_{IH}	Input High Voltage	2.0		6.0	2.0		6.0	2.0		6.0	V	
V_{OL}	Output Low Voltage			0.4			0.4			0.4	V	I_{OL} = 8 mA
V_{OH}	Output High Voltage	2.4			2.4			2.4			V	I_{OH} = −4.0 mA

NOTES:
1. The operating ambient temperature range is guaranteed with transverse air flow exceeding 400 linear feet per minute.
2. Typical limits are at $V_{CC} = 5V$, $T_A = +25°C$, and specified loading.
3. A pull-up resistor to V_{CC} on the CS input is required to keep the device deselected; otherwise, power-on current approaches I_{CC} active.

A.C. TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	5 ns
Input Timing Reference Levels	1.5V
Output Timing Reference Level (2147H-1)	1.5V
Output Timing Reference Levels (2147H, H-2, H-3, HL)	0.8–2.0V
Output Load	See Figure 1

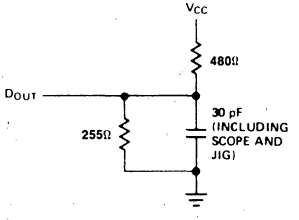


Figure 1. Output Load

CAPACITANCE[4] ($T_A = 25°C$, f = 1.0 MHz)

Symbol	Parameter	Max.	Unit	Conditions
C_{IN}	Input Capacitance	5	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	6	pF	$V_{OUT} = 0V$

NOTE:
4. This parameter is sampled and not 100% tested.

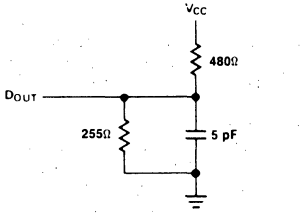


Figure 2. Output Load for t_{HZ}, t_{LZ}, t_{WZ}, t_{OW}

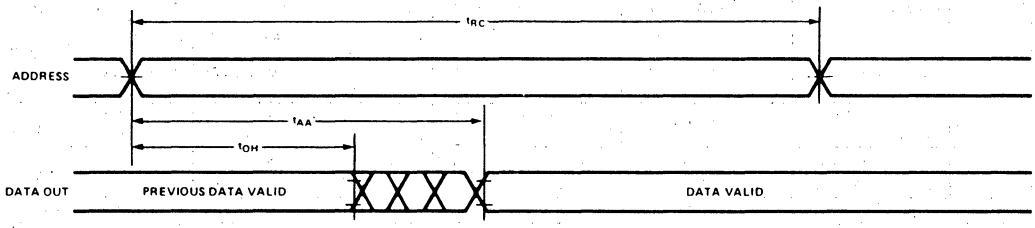
A.C. CHARACTERISTICS ($T_A = 0°C$ to $70°C$, $V_{CC} = +5V \pm 10\%$, unless otherwise noted.)

Read Cycle

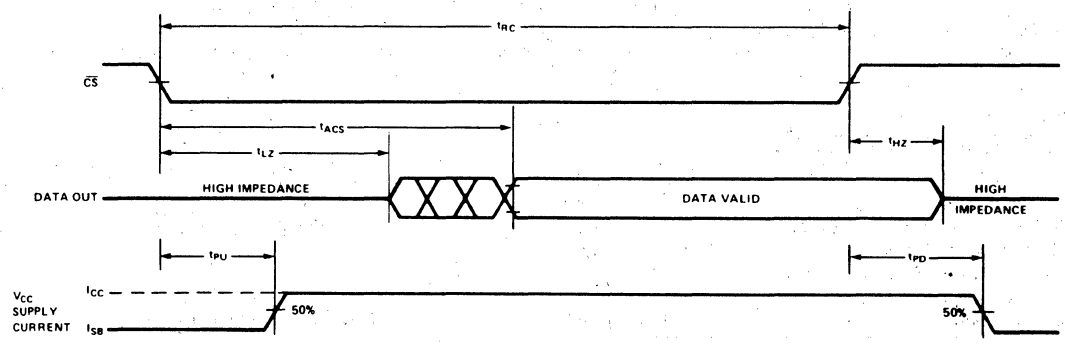
Symbol	Parameter	2147H-1 Min.	2147H-1 Max.	2147H-2 Min.	2147H-2 Max.	2147H-3 Min.	2147H-3 Max.	2147H, 2147HL Min.	2147H, 2147HL Max.	Unit
t_{RC}[1]	Read Cycle Time	35		45		55		70		ns
t_{AA}	Address Access Time		35		45		55		70	ns
t_{ACS1}[8]	Chip Select Access Time		35		45		55		70	ns
t_{ACS2}[9]	Chip Select Access Time		35		45		65		80	ns
t_{OH}	Output Hold from Address Change	5		5		5		5		ns
t_{LZ}[2,3,7]	Chip Selection to Output in Low Z	5		5		10		10		ns
t_{HZ}[2,3,7]	Chip Deselection to Output in High Z	0	30	0	30	0	30	0	40	ns
t_{PU}	Chip Selection to Power Up Time	0		0		0		0		ns
t_{PD}	Chip Deselection to Power Down Time		20		20		20		30	ns

WAVEFORMS

Read Cycle No. 1[4,5]



Read Cycle No. 2[4,6]



NOTES:

1. All Read Cycle timings are referenced from the last valid address to the first transitioning address.
2. At any given temperature and voltage condition, t_{HZ} max. is less than t_{LZ} min. both for a given device and from device to device.
3. Transition is measured ±500 mV from steady state voltage with specified loading in Figure 2.
4. $\overline{WE}$ is high for Read Cycles.
5. Device is continuously selected, $\overline{CS} = V_{IL}$.
6. Addresses valid prior to or coincident with $\overline{CS}$ transition low.
7. This parameter is sampled and not 100% tested.
8. Chip deselected for greater than 55 ns prior to selection.
9. Chip deselected for a finite time that is less than 55 ns prior to selection. If the deselect time is 0 ns, the chip is by definition selected and access occurs according to Read Cycle No. 1. Applies to 2147H, 2147HL, 2147H-3.

A.C. CHARACTERISTICS (Continued)

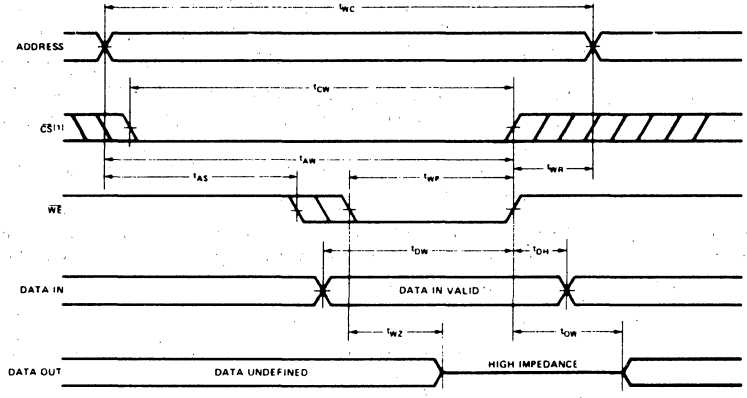
Write Cycle

Symbol	Parameter	2147H-1 Min.	2147H-1 Max.	2147H-2 Min.	2147H-2 Max.	2147H-3 Min.	2147H-3 Max.	2147H, 2147HL Min.	2147H, 2147HL Max.	Unit
t_{WC}[2]	Write Cycle Time	35		45		55		70		ns
t_{CW}	Chip Selection to End of Write	35		45		45		55		ns
t_{AW}	Address Valid to End of Write	35		45		45		55		ns
t_{AS}	Address Setup Time	0		0		0		0		ns
t_{WP}	Write Pulse Width	20		25		25		40		ns
t_{WR}	Write Recovery Time	0		0		10		15		ns
t_{DW}	Data Valid to End of Write	20		25		25		30		ns
t_{DH}	Data Hold Time	10		10		10		10		ns
t_{WZ}[3]	Write Enabled to Output in High Z	0	20	0	25	0	25	0	35	ns
t_{OW}[3]	Output Active from End of Write	0		0		0		0		ns

WAVEFORMS

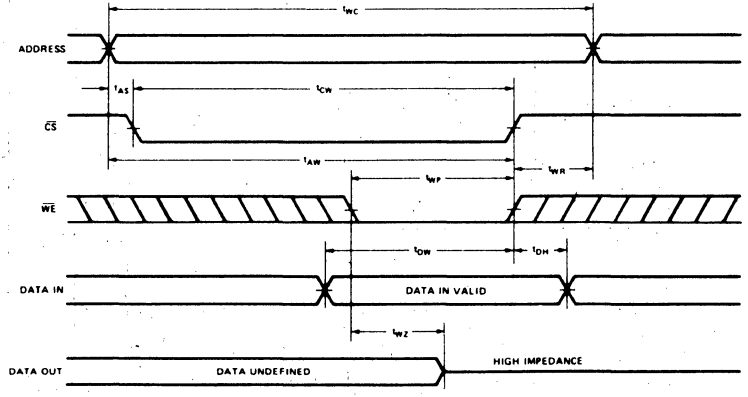
Write Cycle No. 1

($\overline{WE}$ CONTROLLED)[4]



Write Cycle No. 2

($\overline{CS}$ CONTROLLED)[4]



NOTES:

1. If $\overline{CS}$ goes high simultaneously with $\overline{WE}$ high, the output remains in a high impedance state.
2. All Write Cycle timings are referenced from the last valid address to the first transitioning address.
3. Transition is measured ±500 mV from steady state voltage with specified loading in Figure 2.
4. $\overline{CS}$ or $\overline{WE}$ must be high during address transitions.

2148H FAMILY
1024 × 4 BIT STATIC RAM

PRELIMINARY

	*2148H-2	2148H-3	2148H	*2148HL-3	2148HL
Max. Access Time (ns)	45	55	70	55	70
Max. Active Current (mA)	150	*150	*150	125	125
Max. Standby Current (mA)	30	30	30	20	20

- **Improved Performance Margins**
- **Automatic Power-Down**
- **Single +5V Supply**
- **Completely Static Memory — No Clock or Timing Strobe Required**
- **HMOS* III Technology**
- **Common Data Input and Output**
- **Three-State Output**
- **High Reliability Plastic or CERDIP Package**

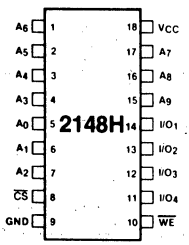
The Intel® 2148H is a 4096-bit static Random Access Memory organized as 1024 words by 4 bits using HMOS III, an ultra high-performance MOS technology. It uses a uniquely innovative design approach which provides the ease-of use features associated with non-clocked static memories and the reduced standby power dissipation associated with clocked static memories. To the user this means low standby power dissipation without the need for clocks, address setup and hold times, nor reduced data rates due to cycle times that are longer than access times.

$\overline{CS}$ controls the power-down feature. In less than a cycle time after $\overline{CS}$ goes high — disabling the 2148H — the part automatically reduces its power requirements and remains in this low power standby mode as long as $\overline{CS}$ remains high. This device feature results in system power savings as great as 85% in larger systems, where the majority of devices are disabled. A non-power-down companion, the 2149H, is available to provide a fast chip select access time for speed critical applications.

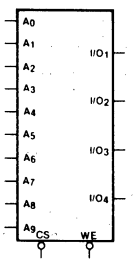
The 2148H is assembled in an 18-pin plastic package configured with the industry standard 1K × 4 pinout. It is directly TTL compatible in all respects: inputs, ouputs, and a single +5V supply. The data is read out nondestructively and has the same polarity as the input data.

* HMOS is a patent process of Intel.

PIN CONFIGURATION



LOGIC SYMBOL



PIN NAMES

A_0–A_9	ADDRESS INPUTS
$\overline{WE}$	WRITE ENABLE
$\overline{CS}$	CHIP SELECT
VO_1–VO_4	DATA INPUT/OUTPUT
V_{CC}	POWER (+5V)
GND	GROUND

TRUTH TABLE

$\overline{CS}$	$\overline{WE}$	MODE	VO	POWER
H	X	NOT SELECTED	HIGH Z	STANDBY
L	L	WRITE	D_{IN}	ACTIVE
L	H	READ	D_{OUT}	ACTIVE

Figure 1. Pin Configuration, Logic Symbol, Pin Names and Truth Table

BLOCK DIAGRAM

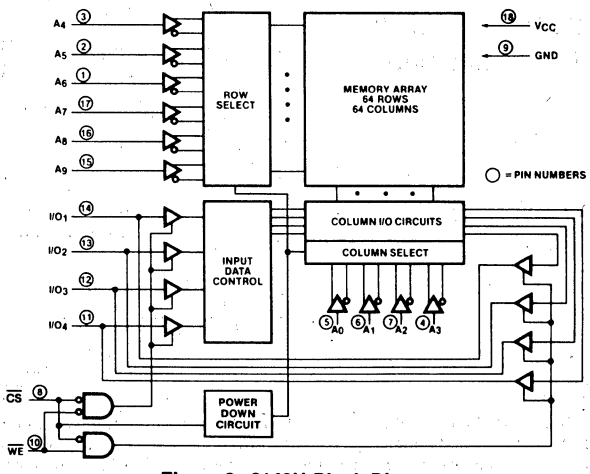


Figure 2. 2148H Block Diagram

* Improved performance margins

AUGUST, 1983
ORDER NUMBER: 230781-001

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . . −10°C to +85°C
Storage Temperature CERDIP . . −65°C to +150°C
Storage Temperature Plastic . . . . −65°C to +125°C
Voltage on Any Pin with
Respect to Ground . . . . . −3.5V to +7V
D.C. Continuous Output Current . . . . . 20 mA
Power Dissipation 1.2W

** COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. AND OPERATING CHARACTERISTICS[1]

$T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 10\%$ unless otherwise noted.

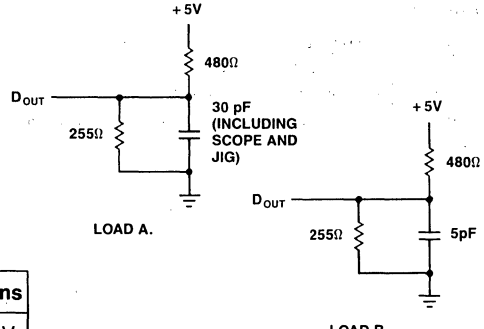
Symbol	Parameter	2148H/H-3/H-2 Min.	2148H/H-3/H-2 Typ[2]	2148H/H-3/H-2 Max.	2148HL/HL-3 Min.	2148HL/HL-3 Typ[2]	2148HL/HL-3 Max.	Unit	Test Conditions
*$\|I_{LI}\|$	Input Load Current (All Input Pins)		0.01	1.0		0.01	1.0	µA	V_{CC} = max, V_{IN} = GND to V_{CC}
*$\|I_{LO}\|$[3]	Output Leakage Current		0.1	10		0.1	10	µA	$\overline{CS}$ = V_{IH}, V_{CC} = max, V_{OUT} = GND to 4.5V
*I_{CC}	Operating Current		100	150		70	125	mA	V_{CC} = max, $\overline{CS}$ = V_{IL}, Outputs Open
I_{SB}	Standby Current		20	30		10	20	mA	V_{CC} = min to max, $\overline{CS}$ = V_{IH}
I_{PO}[4]	Peak Power-On Current		25	50		15	30	mA	V_{CC} = GND to V_{CC} min, $\overline{CS}$ = Lower of V_{CC} or V_{IH} min
V_{IL}	Input Low Voltage	−3.0		0.8	−3.0		0.8	V	
*V_{IH}	Input High Voltage	2.0		6.0	2.0		6.0	V	
V_{OL}	Output Low Voltage			0.4			0.4	V	I_{OL} = 8 mA
V_{OH}	Output High Voltage	2.4			2.4			V	I_{OH} = −4.0 mA
I_{OS}[5]	Output Short Circuit Current		±250	±275		±250	±275	mA	V_{OUT} = GND to V_{CC}

Notes:

1. The operating ambient temperature range is guaranteed with transverse air flow exceeding 400 linear feet per minute. Typical thermal resistance values of the package at maximum temperatures are: For plastic θ_{JA} (@ 400 f_{PM} air flow) = 70° C/W, θ_{JA} (still air) = 109° C/W, θ_{JC} = 42° C/W. For CERDIP θ_{JA} (@ 400 fpm air flow) = 40° C/W, θ_{JA} (still air) = 70° C/W, θ_{JC} = 25° C/W.
2. Typical limits are at V_{CC} = 5V, T_A = +25°C, and Load A.
3. For output leakage tests, data I/O pins are treated as outputs.
4. A pull-up resistor to V_{CC} on the $\overline{CS}$ input is required to keep the device deselected during power-on. Otherwise, power-on current approaches I_{CC} active.
5. Output shorted for no more than 1 second. No more than one output shorted at a time.

A.C. TEST CONDITIONS

Input Pulse Levels	GND to 3.0 Volts
Input Rise and Fall Times	5 nsec
*Output Timing Reference Levels	0.8 & 2.0 Volts
Output Load	See Load A.



CAPACITANCE[6]

T_A = 25°C, f = 1.0 MHz

Symbol	Parameter	Max.	Unit	Conditions
C_{IN}	Address/Control Capacitance	5	pF	V_{IN} = OV
C_{IO}	Input/Output Capacitance	7	pF	V_{OUT} = OV

Note 6. This parameter is sampled and not 100% tested.

* Improved performance margins.

A.C. CHARACTERISTICS

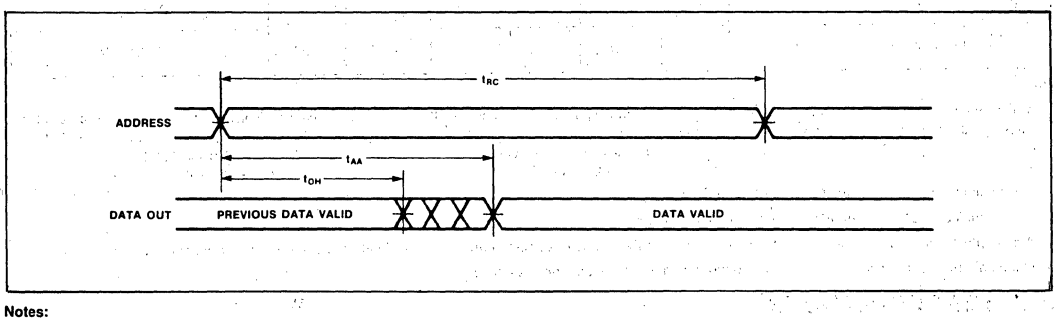
T_A = 0°C to +70°C, V_{CC} = +5V ±10% unless otherwise noted.

READ CYCLE

Symbol	Parameter	2148H-2		2148H-3/HL-3		2148H/HL		Unit	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
t_{RC}	Read Cycle Time	45		55		70		ns	
t_{AA}	Address Access Time		45		55		70	ns	
t_{ACS}	Chip Select Access Time		45		55		70	ns	
t_{OH}	Output Hold from Address Change	5		5		5		ns	
t_{LZ}	Chip Selection Output in Low Z	20		20		20		ns	Note 4
t_{HZ}	Chip Deselection to Output in High Z	0	20	0	20	0	20	ns	Note 4
t_{PU}	Chip Selection to Power Up Time	0		0		0		ns	
t_{PD}	Chip Deselection to Power Down Time		30		30		30	ns	

WAVEFORMS

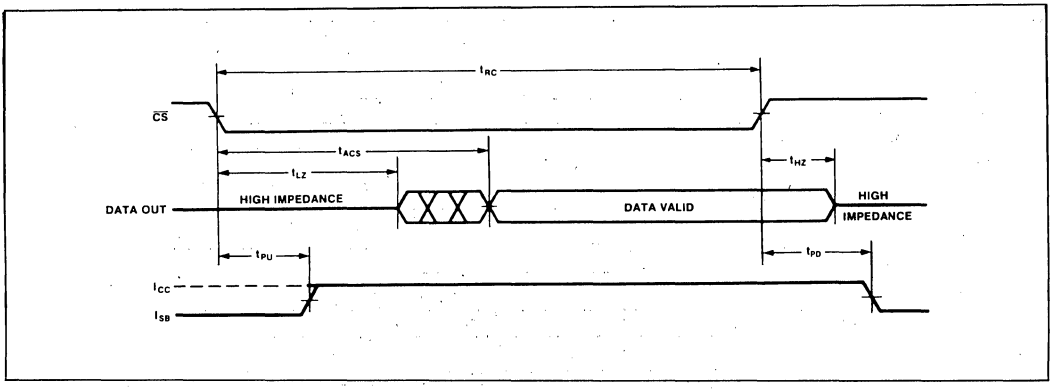
READ CYCLE No. 1[1,2]



Notes:

1. $\overline{WE}$ is high for Read Cycles.
2. Device is continuously selected, $\overline{CS}$ = V_{IL}.
3. Addresses valid prior to or coincident with $\overline{CS}$ transition low.
4. Transition is measured ±500mV from high impedance voltage with Load B.

READ CYCLE No. 2[1,3]



Notes:

1. $\overline{WE}$ is high for Read Cycles.
2. Device is continuously selected, $\overline{CS}$ = V_{IL}.
3. Addresses valid prior to or coincident with $\overline{CS}$ transition low.
4. Transition is measured ±500mV from high impedance voltage with Load B.

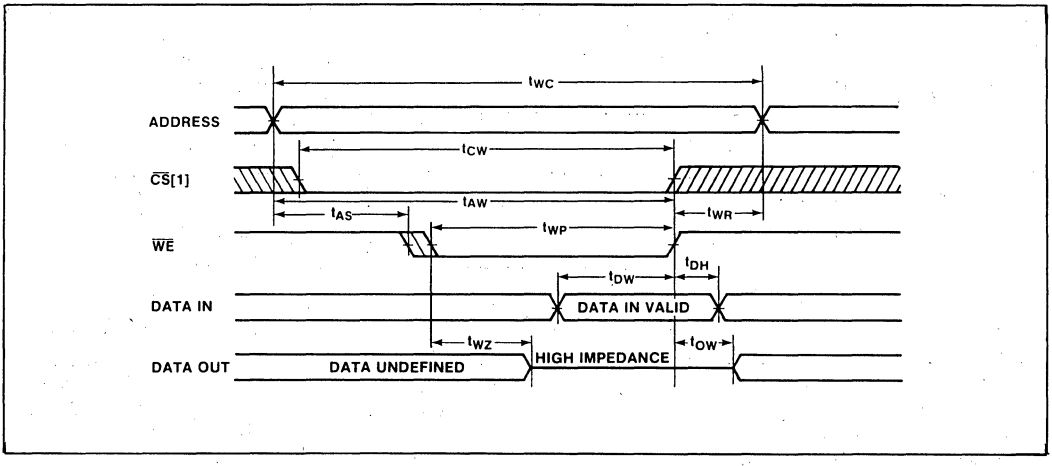
A.C. CHARACTERISTICS (continued)

WRITE CYCLE

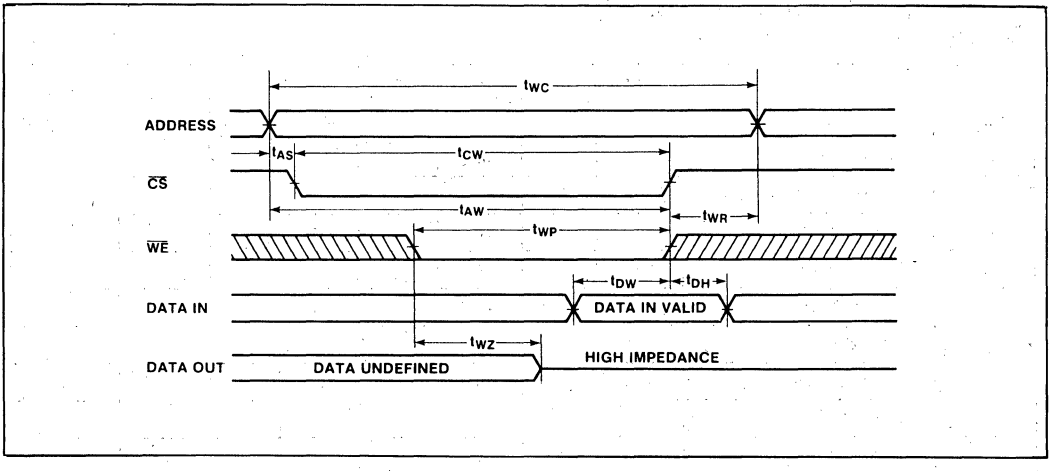
Symbol	Parameter	2148H-2		2148H-3/HL-3		2148/HL		Unit	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
t_{WC}	Write Cycle Time	45		55		70		ns	
t_{CW}	Chip Selection to End of Write	40		50		65		ns	
t_{AW}	Address Valid to End of Write	40		50		65		ns	
t_{AS}	Address Setup Time	0		0		0		ns	
t_{WP}	Write Pulse Width	35		40		50		ns	
t_{WR}	Write Recovery Time	5		5		5		ns	
t_{DW}	Data Valid to End of Write	20		20		25		ns	
t_{DH}	Data Hold Time	0		0		0		ns	
t_{WZ}	Write Enabled to Output in High Z	0	15	0	20	0	25	ns	Note 2
t_{OW}	Output Active from End of Write	0		0		0		ns	Note 2

WAVEFORMS

WRITE CYCLE No. 1 ($\overline{WE}$ CONTROLLED)



WRITE CYCLE No. 2 ($\overline{CS}$ CONTROLLED)[1]



Notes:

1. If $\overline{CS}$ goes high simultaneously with $\overline{WE}$ high, the output remains in a high impedance state.
2. Transition is measured ±500mV from high impedance voltage with Load B.

2149H
1024 x 4-BIT STATIC RAM

	2149H-2	2149H-3	2149H	2149HL
Max. Address Access Time (ns)	45	55	70	70
Max. Chip Select Access Time (ns)	20	25	30	30
Max. Active Current (mA)	180	180	180	125

- **Fast Chip Select Access Time—20ns Maximum**
- **HMOS II Technology**
- **Equal Access and Cycle Times**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**
- **High Density 18-Pin Package**
- **Common Data Input and Output**
- **Three-State Output**
- **Single +5V Supply**
- **Automatic Power-Down 2148H Available**

The Intel® 2149H is a 4096-bit static Random Access Memory organized as 1024 words by 4 bits using HMOS II, a high performance MOS technology. It provides a maximum chip select access time as low as 20 ns instead of an automatic power-down feature. This fast chip select access time feature increases system throughput. An automatic power-down companion, the 2148H, is available for power critical applications.

The 2149H is assembled in an 18-pin package configured with the industry standard 1Kx4 pinout. It is directly TTL compatible in all respects: inputs, outputs and a single +5V supply. The data is read out non-destructively and has the same polarity as the input data.

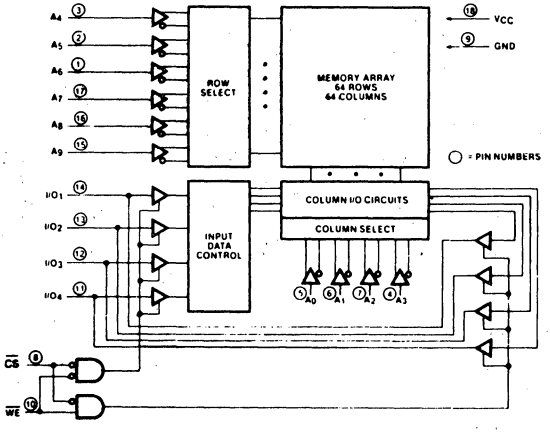
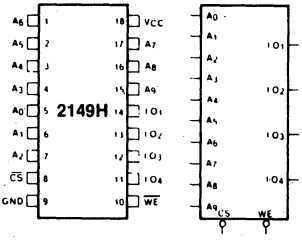


Figure 1. 2149H Block Diagram



PIN NAMES

A_0-A_9	ADDRESS INPUTS
$\overline{WE}$	WRITE ENABLE
$\overline{CS}$	CHIP SELECT
I/O_1-I/O_4	DATA INPUT/OUTPUT
V_{CC}	POWER (+5V)
GND	GROUND

TRUTH TABLE

$\overline{CS}$	$\overline{WE}$	MODE	I/O
H	X	NOT SELECTED	HIGH-Z
L	L	WRITE	D_{IN}
L	H	READ	D_{OUT}

Figure 2. 2149H Pin Diagram

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +85°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground ..... −3.5V to +7V
D.C. Continuous Output Current ..... 20 mA
Power Dissipation 1.2W

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. AND OPERATING CHARACTERISTICS[1]

$T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 10\%$ unless otherwise noted.

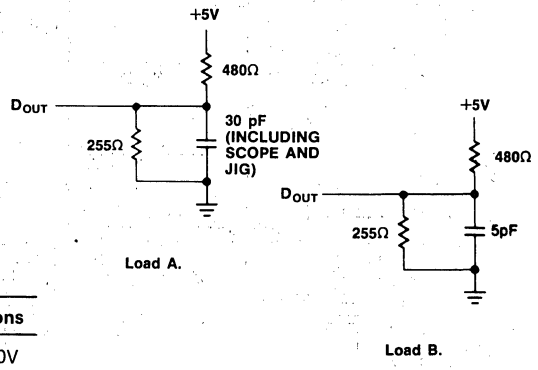
Symbol	Parameter	2149H/H-2/H-3 Min.	Typ[2]	Max.	2149HL Min.	Typ[2]	Max.	Unit	Test Conditions
$\|I_{LI}\|$	Input Load Current (All Input Pins)		0.01	1.0		0.01	1.0	μA	V_{CC} = max, V_{IN} = GND to 5.5V
$\|I_{LO}\|$	Output Leakage Current		0.1	50		0.1	50	μA	$\overline{CS} = V_{IH}$, V_{CC} = 5.5V V_{OUT} = GND to 5.5V
I_{CC}	Operating Current		120	180		90	125	mA	V_{CC} = max, $\overline{CS} = V_{IL}$, Outputs Open
V_{IL}	Input Low Voltage	−3.0		0.8	−3.0		0.8	V	
V_{IH}	Input High Voltage	2.1		6.0	2.1		6.0	V	
V_{OL}	Output Low Voltage			0.4			0.4	V	I_{OL} = 8 mA
V_{OH}	Output High Voltage	2.4			2.4			V	I_{OH} = −4.0 mA
I_{OS}[3]	Output Short Circuit Current		±150	±200		±150	±200	mA	V_{OUT} = GND to V_{CC}

Notes:

1. The operating ambient temperature range is guaranteed with transverse air flow exceeding 400 linear feet per minute. Typical thermal resistance values of the package at maximum temperatures are:
   θ_{JA} (@ 400 f_{PM} air flow) = 40° C/W
   θ_{JA} (still air) = 70° C/W
   θ_{JC} = 25° C/W
2. Typical limits are at V_{CC} = 5V, T_A = +25°C, and Load A.
3. Duration not to exceed 1 second.

A.C. TEST CONDITIONS

Input Pulse Levels	GND to 3.0 Volts
Input Rise and Fall Times	5 nsec
Input and Output Timing Reference Levels	1.5 Volts
Output Load	See Load A.



CAPACITANCE [3]

$T_A = 25°C$, f = 1.0 MHz

Symbol	Parameter	Max.	Unit	Conditions
C_{IN}	Address/Control Capacitance	5	pF	V_{IN} = 0V
C_{IO}	Input/Output Capacitance	7	pF	V_{OUT} = 0V

Note 3. This parameter is sampled and not 100% tested.

A.C. CHARACTERISTICS

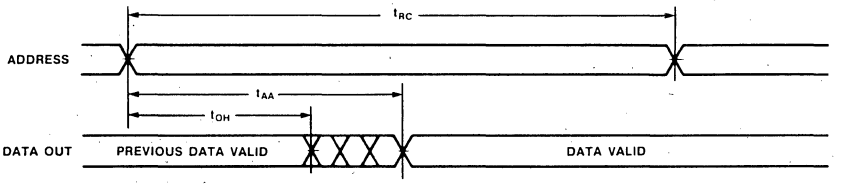
$T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 10\%$ unless otherwise noted.

READ CYCLE

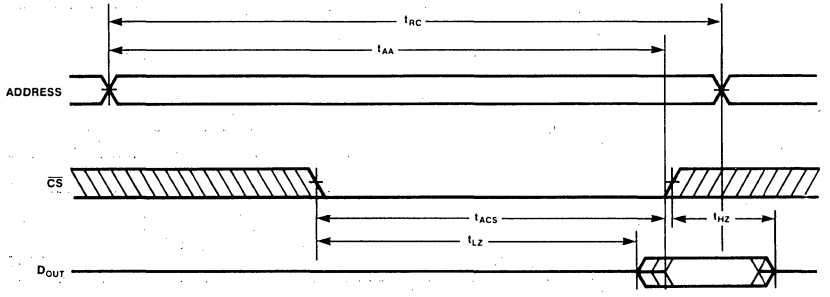
Symbol	Parameter	2149H-2 Min.	2149H-2 Max.	2149H-3 Min.	2149H-3 Max.	2149H/HL Min.	2149H/HL Max.	Unit	Test Conditions
t_{RC}	Read Cycle Time	45		55		70		ns	
t_{AA}	Address Access Time		45		55		70	ns	
t_{ACS}	Chip Select Access Time		20		25		30	ns	
t_{OH}	Output Hold from Address Change	5		5		5		ns	
t_{LZ}	Chip Selection Output in Low Z	5		5		5		ns	Note 3, 4
t_{HZ}	Chip Deselection to Output in High Z	0	15	0	15	0	15	ns	Note 3, 4

WAVEFORMS

READ CYCLE No. 1[1, 2]



READ CYCLE No. 2[3]



Notes:

1. $\overline{WE}$ is high for Read Cycles.
2. Device is continuously selected, $\overline{CS} = V_{IL}$.
3. At any given temperature and voltage condition, t_{HZ} max. is less than t_{LZ} min. both for a given device and from device to device.
4. Transition is measured ±500 mV from high impedance voltage with Load B. This parameter is sampled and not 100% tested.

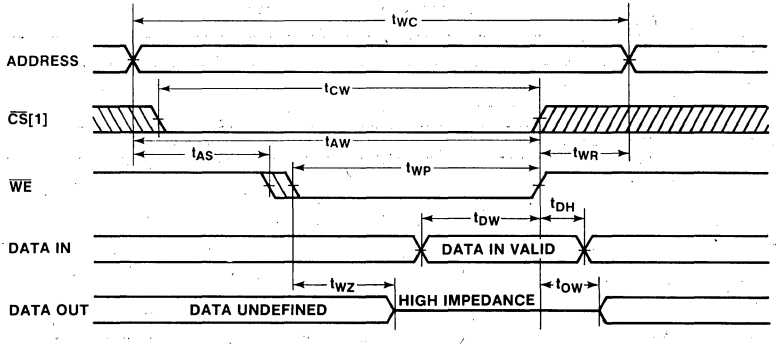
A.C. CHARACTERISTICS (continued)

WRITE CYCLE

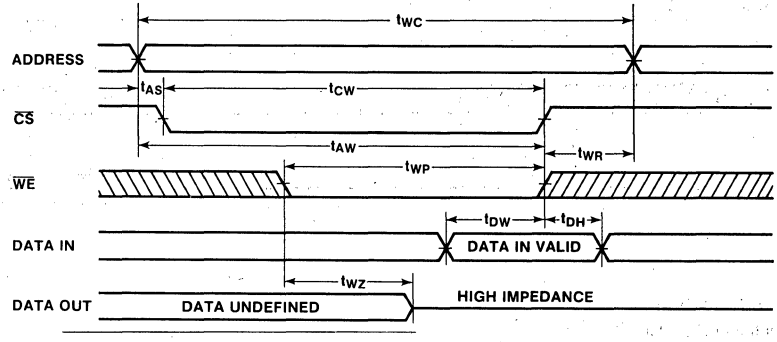
Symbol	Parameter	2149H-2 Min.	2149H-2 Max.	2149H-3 Min.	2149H-3 Max.	2149H/HL Min.	2149H/HL Max.	Unit	Test Conditions
t_{WC}	Write Cycle Time	45		55		70		ns	
t_{CW}	Chip Selection to End of Write	40		50		65		ns	
t_{AW}	Address Valid to End of Write	40		50		65		ns	
t_{AS}	Address Setup Time	0		0		0		ns	
t_{WP}	Write Pulse Width	35		40		50		ns	
t_{WR}	Write Recovery Time	5		5		5		ns	
t_{DW}	Data Valid to End of Write	20		20		25		ns	
t_{DH}	Data Hold Time	0		0		0		ns	
t_{WZ}	Write Enabled to Output in High Z	0	15	0	20	0	25	ns	Note 2
t_{OW}	Output Active from End of Write	0		0		0		ns	Note 2

WAVEFORMS

WRITE CYCLE No. 1 ($\overline{WE}$ CONTROLLED)



WRITE CYCLE No. 2 ($\overline{CS}$ CONTROLLED)



Notes:

1. If $\overline{CS}$ goes high simultaneously with $\overline{WE}$ high, the output remains in a high impedance state.
2. Transition is measured ±500 mV from high impedance voltage with Load B. This parameter is sampled and not 100% tested.

2164A FAMILY
65,536 x 1 BIT DYNAMIC RAM

	2164A-15	2164A-20
Maximum Access Time (ns)	150	200
Read, Write Cycle (ns)	260	330
Page Mode Read, Write Cycle (ns)	125	170

- **HMOS-D III technology**
- **Low capacitance, fully TTL compatible inputs and outputs**
- **Single +5V supply, ±10% tolerance**
- **128 refresh cycle/2 ms $\overline{RAS}$ only refresh**
- **Compatible with the 2118**
- **Extended page mode, read-modify-write and hidden refresh operation**
- **Inputs allow a −2.0V negative overshoot**
- **Industry standard 16-pin DIP**
- **Compatible with Intel's microprocessors and DRAM controllers**

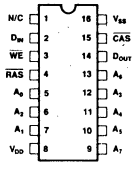
The 2164A is a 65,536 word by 1-bit N-channel MOS dynamic Random Access Memory fabricated with Intel's HMOS-D III technology for high system performance and reliability. The 2164A design incorporates high storage cell capacitance to provide wide internal device margins for reduced noise sensitivities and more reliable system operation. Moreover, high storage cell capacitance results in low soft error rates without the need for a die coat. HMOS-D III process employs the use of redundant elements.

The 2164A is optimized for high speed, high performance applications such as mainframe memory, buffer memory, microprocessor memory, peripheral storage and graphic terminals. For memory intensive microprocessor applications the 2164A is fully compatible with Intel's DRAM controllers and microprocessors to provide a complete DRAM system.

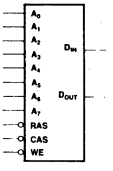
Multiplexing the 16 address bits into the 8 address input pins allows the 2164A to achieve high packing density. The 16 pin DIP provides for high system bit densities, and is compatible with widely available automated testing and insertion equipment. The two 8-bit TTL level address segments are latched into the 2164A by the two TTL clocks, Row Address Strobe ($\overline{RAS}$) and Column Address Strobe ($\overline{CAS}$). Non-critical timing requirements for the $\overline{RAS}$ and $\overline{CAS}$ clocks allow the use of the address multiplexing technique while maintaining high performance.

The non-latched, three state, TTL compatible data output is controlled by $\overline{CAS}$, independent of $\overline{RAS}$. After a valid read or read-modify-write cycle, data is held on the data output pin by holding $\overline{CAS}$ low. The data output is returned to a high impedance state, by returning $\overline{CAS}$ to a high state. Hidden refresh capability allows the device to maintain data at the output by holding $\overline{CAS}$ low while $\overline{RAS}$ is used to execute $\overline{RAS}$-only refresh cycles. Refreshing is accomplished by performing $\overline{RAS}$-only cycles, hidden refresh cycles, or normal read or write cycles on the 128 address combinations of addresses A_0 through A_6, during a 2 ms period.

PIN CONFIGURATION

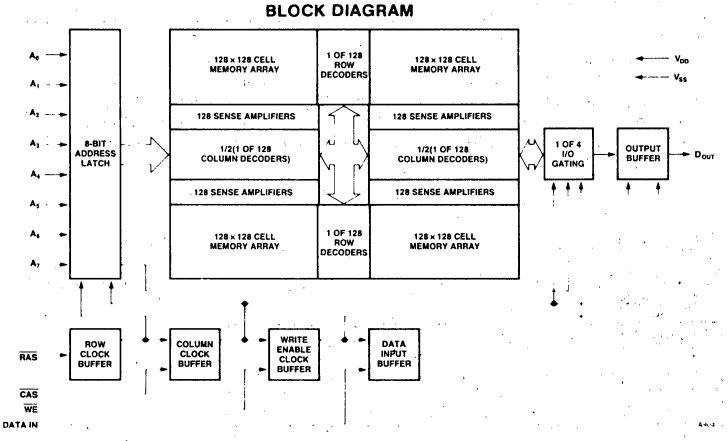


LOGIC SYMBOL



A_0-A_7	ADDRESS INPUTS
$\overline{CAS}$	COLUMN ADDRESS STROBE
D_{IN}	DATA IN
D_{OUT}	DATA OUT
$\overline{WE}$	WRITE ENABLE
$\overline{RAS}$	ROW ADDRESS STROBE
V_{DD}	POWER (+5V)
V_{SS}	GROUND

BLOCK DIAGRAM



April, 1982
Order Number: 210425-001

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature
Under Bias ..... −10°C to +80°C
Storage Temperature .. Cerdip −65°C to +150°C
Plastic −55°C to +125°C
Voltage on Any Pin except V_{DD}
Relative to V_{SS} ..... −2.0V to 7.5V
Voltage on V_{DD} Relative to V_{SS} .... −1.0V to 7.5V
Data Out Current ..... 50 mA
Power Dissipation ..... 1.0 W

*COMMENT:

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS[1]

$T_A = 0°C$ to $70°C$, $V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted.

Symbol	Parameter	Limits			Unit	Test Conditions	Notes
		Min.	Typ.[2]	Max.			
$\|I_{LI}\|$	Input Load Current (any input)			10	μA	$V_{IN} = V_{SS}$ to V_{DD}	
$\|I_{LO}\|$	Output Leakage Current for High Impedance State			10	μA	Chip Deselected: $\overline{CAS}$ at V_{IH}, $D_{OUT} = 0$ to 5.5V	
I_{DD1}	V_{DD} Supply Current, Standby		3	5	mA	$\overline{CAS}$ and $\overline{RAS}$ at V_{IH}	
I_{DD2}	V_{DD} Supply Current, Operating		42	55	mA	2164A-15, $t_{RC} = t_{RCMIN}$	3
			33	45	mA	2164A-20, $t_{RC} = t_{RCMIN}$	3
I_{DD3}	V_{DD} Supply Current, $\overline{RAS}$-Only Cycle		30	45	mA	2164A-15, $t_{RC} = t_{RCMIN}$	
			24	40	mA	2164A-20, $t_{RC} = t_{RCMIN}$	
I_{DD5}	V_{DD} Supply Current, Standby Output Enabled			6	mA	$\overline{CAS}$ at V_{IL}, $\overline{RAS}$ at V_{IH}	3
V_{IL}	Input Low Voltage (all inputs)	−1.0		0.8	V		4
V_{IH}	Input High Voltage (all inputs)	2.4		7.0	V		
V_{OL}	Output Low Voltage			0.4	V	$I_{OL} = 4.2$ mA	5
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -5$ mA	5

NOTES:
1. All voltages referenced to V_{SS}.
2. Typical values are for $T_A = 25°C$ and nominal supply voltages.
3. I_{DD} is dependent on output loading when the device output is selected. Specified $I_{DD\ MAX}$ is measured with the output open.
4. Specified $V_{IL\ MIN}$ is for steady state operation. During transitions the inputs may overshoot to −2.0V for periods not to exceed 20 ns.
5. Test conditions apply only for D.C. characteristics. A.C. parameters specified with a load equivalent to 2 TTL loads and 100 pF.

CAPACITANCE[1]

$T_A = 25°C$, $V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted.

Symbol	Parameter	Typ.	Max.	Unit
C_{11}	Address, Data In	3	5	pF
C_{12}	$\overline{WE}$, Data Out	3	6	pF
C_{13}	$\overline{RAS}$, $\overline{CAS}$	4	8	pF

NOTES:
1. Capacitance measured with Boonton Meter or effective capacitance calculated from the equation:

$$C = \frac{I\Delta t}{\Delta V}$$

with ΔV equal to 3 volts and power supplies at nominal levels.

A.C. CHARACTERISTICS [1,2,3]

$T_A = 0°C$ to $70°C$, $V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted.

READ, WRITE, READ-MODIFY-WRITE AND REFRESH CYCLES

Symbol	Parameter	2164A-15 Min.	2164A-15 Max.	2164A-20 Min.	2164A-20 Max.	Unit	Notes
t_{RAC}	Access Time From $\overline{RAS}$		150		200	ns	4,5
t_{CAC}	Access Time From $\overline{CAS}$		85		120	ns	5,6
t_{REF}	Time Between Refresh		2		2	ms	
t_{RP}	$\overline{RAS}$ Precharge Time	100		120		ns	
t_{CPN}	$\overline{CAS}$ Precharge Time (non-page cycles)	25		35		ns	
t_{CRP}	$\overline{CAS}$ to $\overline{RAS}$ Precharge Time	−20		−20		ns	
t_{RCD}	$\overline{RAS}$ to $\overline{CAS}$ Delay Time	30	65	35	80	ns	7
t_{RSH}	$\overline{RAS}$ Hold Time	85		120		ns	
t_{CSH}	$\overline{CAS}$ Hold Time	150		200		ns	
t_{ASR}	Row Address Set-Up Time	0		0		ns	
t_{RAH}	Row Address Hold Time	20		25		ns	
t_{ASC}	Column Address Set-Up Time	0		0		ns	
t_{CAH}	Column Address Hold Time	25		30		ns	
t_{AR}	Column Address Hold Time to $\overline{RAS}$	90		110		ns	
t_T	Transition time (Rise and Fall)	3	50	3	50	ns	8
t_{OFF}	Output Buffer Turn Off Delay	0	30	0	40	ns	

READ AND REFRESH CYCLES

t_{RC}	Random Read Cycle Time	260		330		ns	
t_{RAS}	$\overline{RAS}$ Pulse Width	150	10000	200	10000	ns	
t_{CAS}	$\overline{CAS}$ Pulse Width	85	10000	120	10000	ns	
t_{RCS}	Read Command Set-Up Time	0		0		ns	
t_{RCH}	Read Command Hold Time referenced to $\overline{CAS}$	5		5		ns	9
t_{RRH}	Read Command Hold Time referenced to $\overline{RAS}$	20		20		ns	9

NOTES:

1. All voltages referenced to V_{SS}.
2. An initial pause of 500 μs is required after power up followed by a minimum of eight (8) initialization cycles (any combination of cycles containing a RAS clock such as RAS-only refresh). 8 initialization cycles are required after extended periods of bias (greater than 2 ms) without clocks.
3. A.C. Characteristics assume $t_T = 5$ ns.
4. Assumes that $t_{RCD} \leq t_{RCD}(max)$. If t_{RCD} is greater than $t_{RCD}(max)$ then t_{RAC} will increase by the amount that t_{RCD} exceeds $t_{RCD}(max)$.
5. Load = 2 TTL loads and 100 pF.
6. Assumes $t_{RCD} \geq t_{RCD}(max)$.
7. $t_{RCD}(max)$ is specified as a reference point only. If t_{RCD} is less than $t_{RCD}(max)$ access time is t_{RAC}. If t_{RCD} is greater than $t_{RCD}(max)$ access time is $t_{RCD} + t_{CAC}$. $t_{RCD}(min) = t_{RAH} + t_{ASC} + t_T + t_T$ ($t_T = 5$ ns).
8. t_T is measured between $V_{IH}(min)$ and $V_{IL}(max)$.
9. Either t_{RCH} or t_{RRH} must be satisfied.

A.C. CHARACTERISTICS (con't.)

WRITE CYCLE

Symbol	Parameter	2164A-15 Min.	2164A-15 Max.	2164A-20 Min.	2164A-20 Max.	Unit	Notes
t_{RC}	Random Write Cycle Time	260		330		ns	
t_{RAS}	$\overline{RAS}$ Pulse Width	150	10000	200	10000	ns	
t_{CAS}	$\overline{CAS}$ Pulse Width	85	10000	120	10000	ns	
t_{WCS}	Write Command Set-Up Time	−10		−10		ns	10
t_{WCH}	Write Command Hold Time	30		40		ns	
t_{WCR}	Write Command Hold Time to $\overline{RAS}$	95		120		ns	
t_{WP}	Write Command Pulse Width	30		40		ns	
t_{RWL}	Write Command to $\overline{RAS}$ Lead Time	40		50		ns	
t_{CWL}	Write Command to $\overline{CAS}$ Lead Time	40		50		ns	
t_{DS}	Data-In Set-Up Time	0		0		ns	
t_{DH}	Data-In Hold Time	30		40		ns	
t_{DHR}	Data-In Hold Time to $\overline{RAS}$	95		120		ns	

READ-MODIFY-WRITE CYCLE

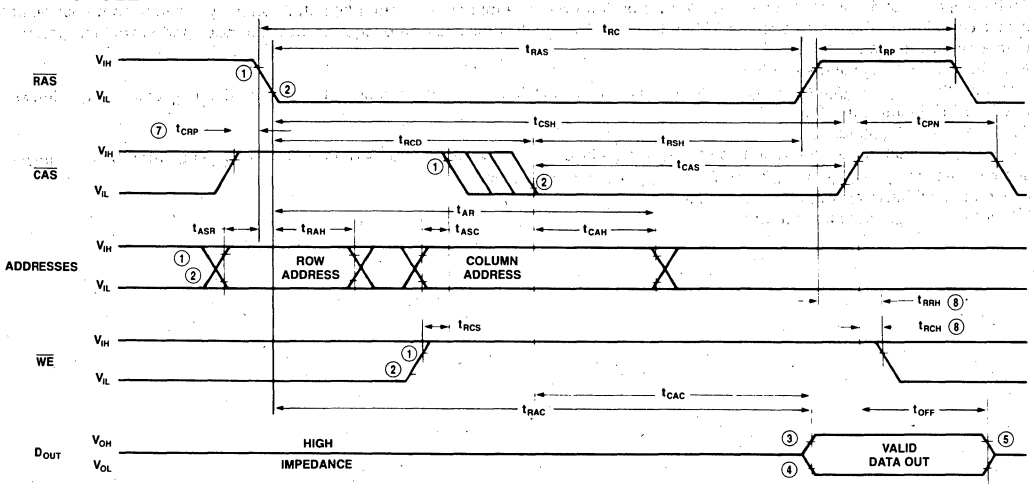
t_{RWC}	Read-Modify-Write Cycle time	280		355		ns	
t_{RRW}	RMW Cycle RAS Pulse Width	170	10000	225	10000	ns	
t_{CRW}	RMW Cycle CAS Pulse Width	105	10000	145	10000	ns	
t_{RWD}	$\overline{RAS}$ to $\overline{WE}$ Delay	125		170		ns	10
t_{CWD}	$\overline{CAS}$ to $\overline{WE}$ Delay	60		90		ns	10

NOTES:

10. t_{WCS}, t_{CWD} and t_{RWD} are specified as reference points only. If $t_{WCS} \geq t_{WCS}$(min) the cycle is an early write cycle and the data out pin will remain high impedance throughout the entire cycle. If $t_{CWD} \geq t_{CWD}$(min) and $t_{RWD} \geq t_{RWD}$(min) the cycle is a read-modify-write cycle and the data out will contain the data read from the selected address. If neither of the above conditions is satisfied, the condition of the data out is indeterminate.

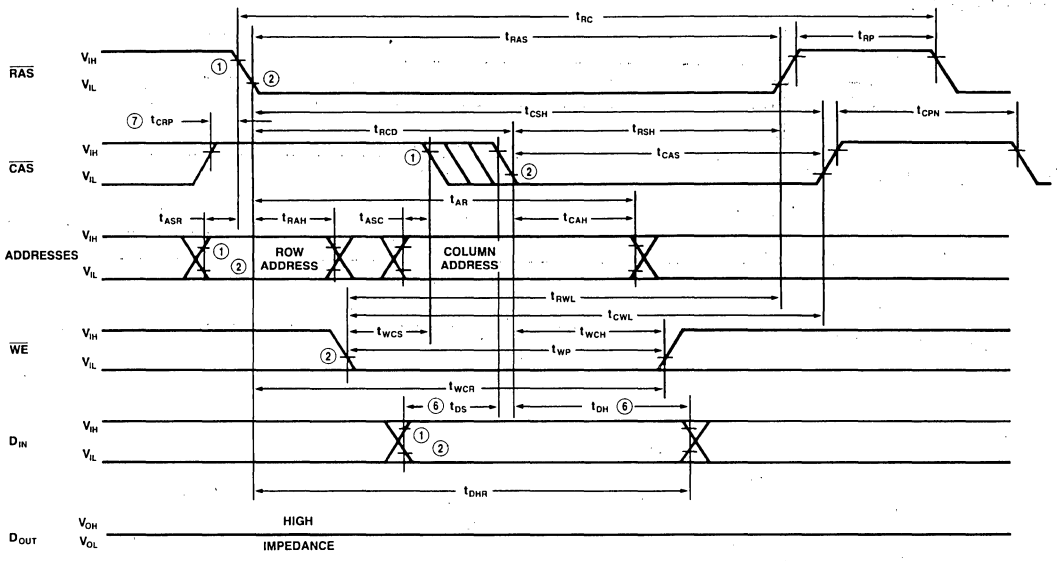
WAVEFORMS

READ CYCLE

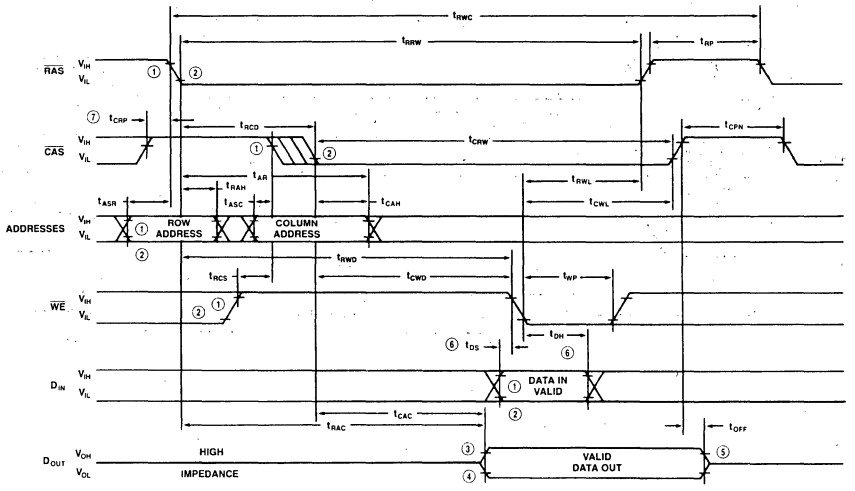


WAVEFORMS

WRITE CYCLE



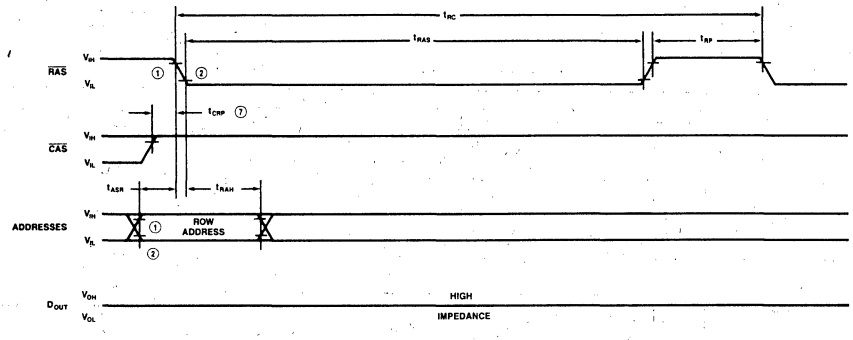
READ-MODIFY-WRITE CYCLE



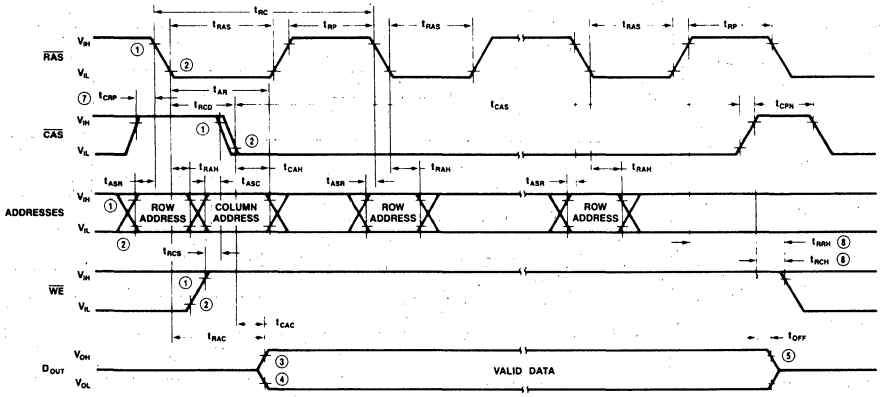
NOTES: 1,2. $V_{IH\ MIN}$ and $V_{IL\ MAX}$ are reference levels for measuring timing of input signals.
3,4. $V_{OH\ MIN}$ and $V_{OL\ MAX}$ are reference levels for measuring timing of D_{OUT}.
5. t_{OFF} is measured to $I_{OUT} \leq |I_{LO}|$.
6. t_{DS} and t_{DH} are referenced to $\overline{CAS}$ or $\overline{WE}$, whichever occurs last.
7. t_{CRP} requirement is only applicable for $\overline{RAS}/\overline{CAS}$ cycles preceeded by a $\overline{CAS}$-only cycle (i.e., for systems where $\overline{CAS}$ has not been decoded with $\overline{RAS}$).
8. Either t_{RCH} or t_{RRH} must be satisfied.

WAVEFORMS

RAS-ONLY REFRESH CYCLE



HIDDEN REFRESH CYCLE



NOTES: 1,2. $V_{IH\ MIN}$ and $V_{IL\ MAX}$ are reference levels for measuring timing of input signals.
3,4. $V_{OH\ MIN}$ and $V_{OL\ MAX}$ are reference levels for measuring timing of D_{OUT}.
5. t_{OFF} is measured to $|I_{OUT}| \leq |I_{LO}|$.
6. t_{DS} and t_{DH} are referenced to $\overline{CAS}$ or $\overline{WE}$, whichever occurs last.
7. t_{CRP} requirement is only applicable for $\overline{RAS}/\overline{CAS}$ cycles preceeded by a $\overline{CAS}$-only cycle (i.e., for systems where $\overline{CAS}$ has not been decoded with $\overline{RAS}$).
8. Either t_{RCH} or t_{RRH} must be satisfied.

D.C. AND A.C. CHARACTERISTICS, PAGE MODE [6,7,11]

$T_A = 0°C$ to $70°C$, $V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted.

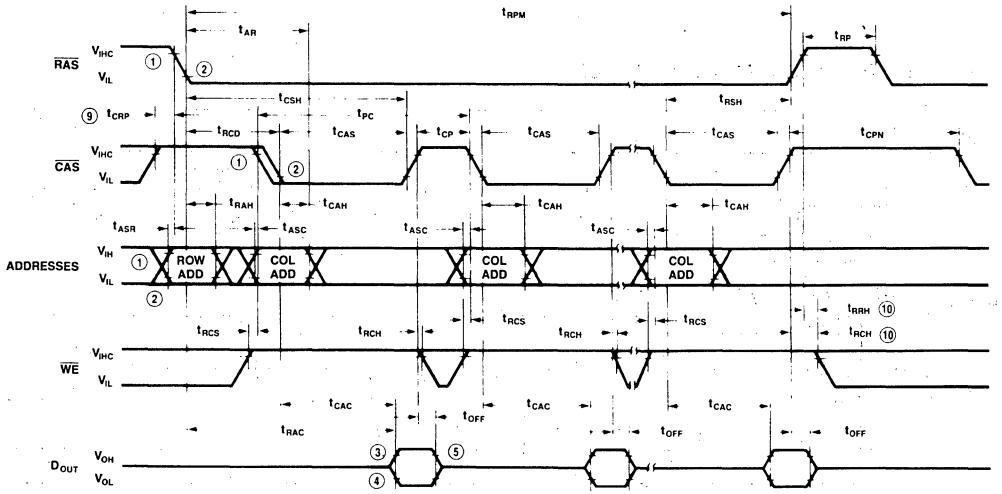
Symbol	Parameter	2164A-15 Min.	2164A-15 Max.	2164A-20 Min.	2164A-20 Max.	Unit	Notes
t_{PC}	Page Mode Read or Write Cycle	125		170		ns	
t_{PCM}	Page Mode Read Modify Write	145		195		ns	
t_{CP}	$\overline{CAS}$ Precharge Time, Page Cycle	30		40		ns	
t_{RPM1}	$\overline{RAS}$ Pulse Width, Page Mode		10000		10000	ns	
t_{CAS}	$\overline{CAS}$ Pulse Width	85	10000	120	10000	ns	
I_{DD4}	V_{DD} Supply Current Page Mode, Minimum t_{PC}, Minimum t_{CAS}		40		35	mA	8

EXTENDED PAGE MODE [11,12]

Symbol	Parameter	2164A-15 S6493 Min.	2164A-15 S6493 Max.	2164A-20 S6494 Min.	2164A-20 S6494 Max.	Unit	Notes
t_{RPM2}	$\overline{RAS}$ Pulse Width, Extended Page Mode		75000		75000	ns	

WAVEFORMS

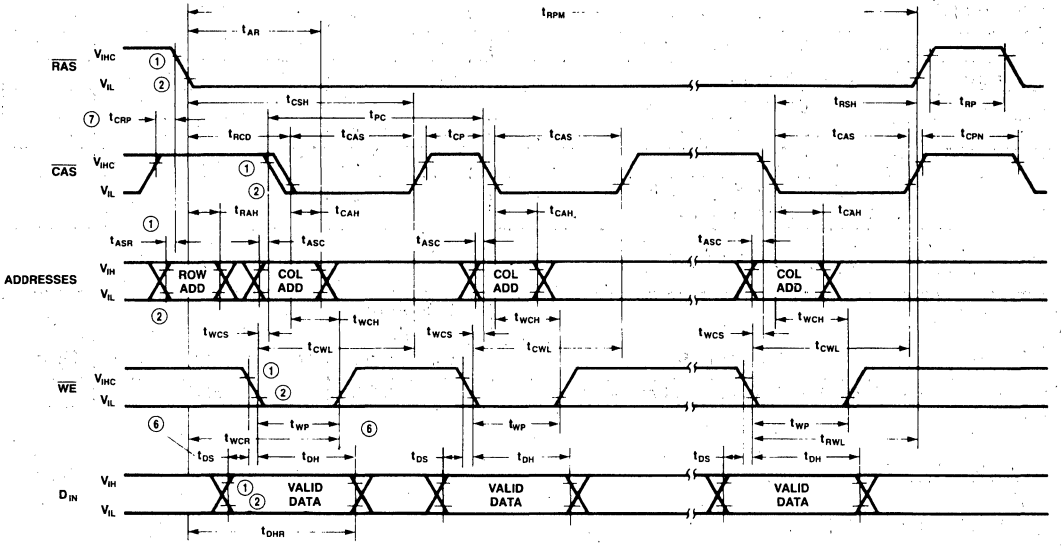
PAGE MODE READ CYCLE



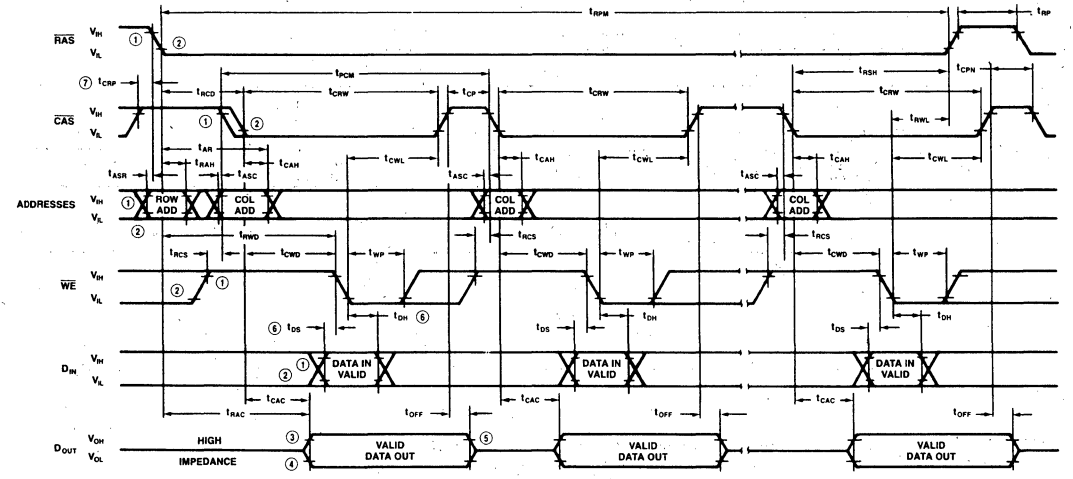
NOTES: 1,2. $V_{IH\,MIN}$ and $V_{IL\,MAX}$ are reference levels for measuring timing of input signals.
3,4. $V_{OH\,MIN}$ and $V_{OL\,MAX}$ are reference levels for measuring timing of D_{OUT}.
5. t_{OFF} is measured to $I_{OUT} \leq |I_{LO}|$.
6. All voltages referenced to V_{SS}.
7. A.C. characteristic assume $t_T = 5$ ns.
8. See the typical characteristics section for values of this parameter under alternate conditions.
9. t_{CRP} requirement is only applicable for $\overline{RAS}/\overline{CAS}$ cycles preceeded by a $\overline{CAS}$-only cycle (i.e., for systems where $\overline{CAS}$ has not been decoded with $\overline{RAS}$).
10. Either t_{RCH} or t_{RRH} must be satisfied.
11. All previously specified A.C. and D.C. characteristics are applicable.
12. For extended page mode operation, order 2164A-15 S6493, 2164A-20 S6494.

WAVEFORMS

PAGE MODE WRITE CYCLE

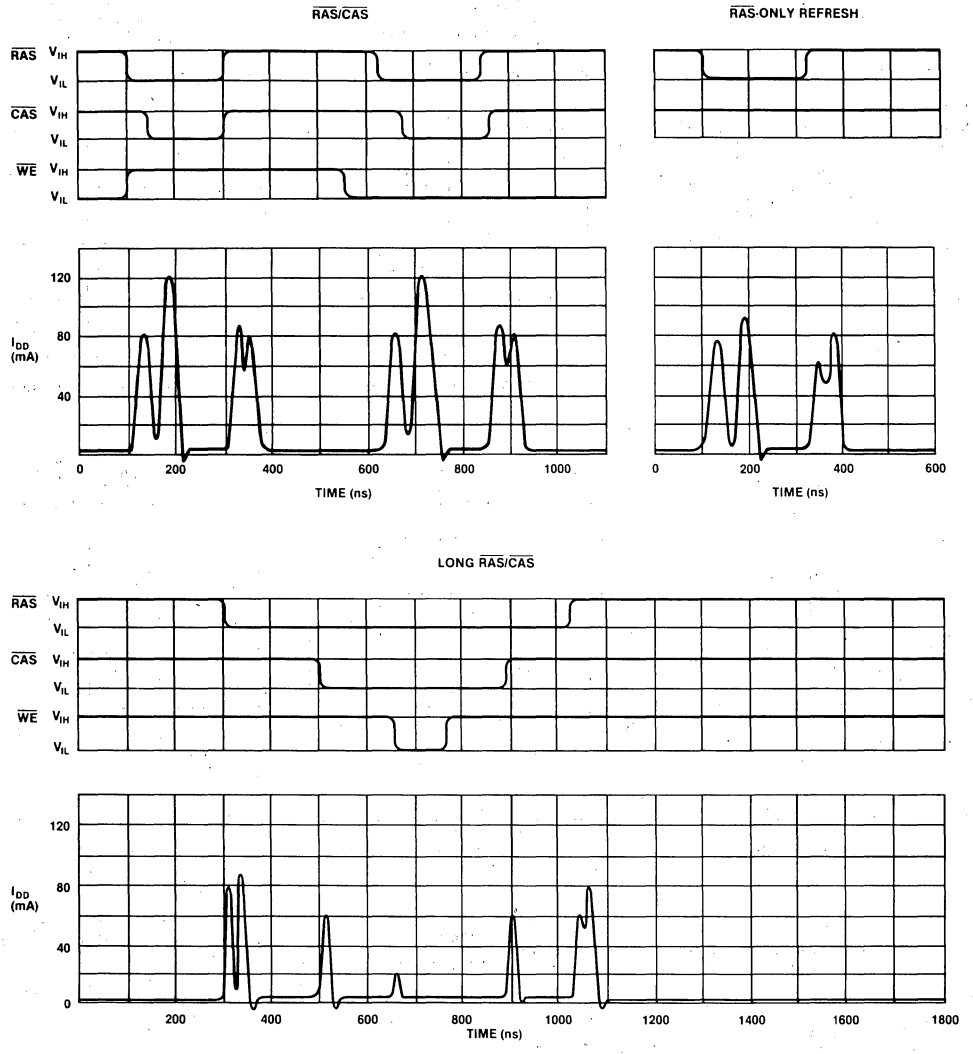


PAGE MODE READ-MODIFY-WRITE CYCLE



NOTES: 1,2. $V_{IH\ MIN}$ and $V_{IL\ MAX}$ are reference levels for measuring timing of input signals.
3,4. $V_{OH\ MIN}$ and $V_{OL\ MAX}$ are reference levels for measuring timing of D_{OUT}.
5. t_{OFF} is measured to $I_{OUT} \leq |I_{LO}|$.
6. t_{DS} and t_{DH} are referenced to $\overline{CAS}$ or $\overline{WE}$, whichever occurs last.
7. t_{CRP} requirement is only applicable for $\overline{RAS}$/$\overline{CAS}$ cycles preceeded by a $\overline{CAS}$-only cycle (i.e., for systems where $\overline{CAS}$ has not been decoded with $\overline{RAS}$).

TYPICAL SUPPLY CURRENT WAVEFORMS

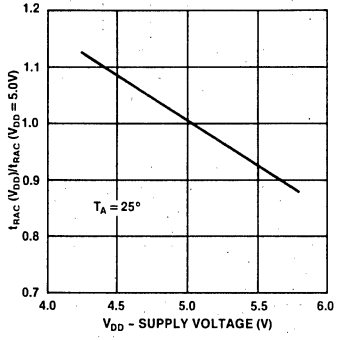


Typical power supply waveforms vs. time are shown for the $\overline{RAS}/\overline{CAS}$ timings of Read/Write, Read/Write (long $\overline{RAS}/\overline{CAS}$), and $\overline{RAS}$-only refresh cycles. I_{DD} current transients at the $\overline{RAS}$ and $\overline{CAS}$ edges require adequate decoupling of these supplies.

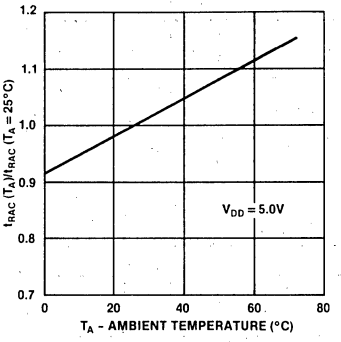
The effects of cycle time, V_{DD} supply voltage and ambient temperature on the I_{DD} current are shown in graphs included in the Typical Characteristics Section. Each family of curves for I_{DD1}, I_{DD2}, and I_{DD3} is related by a common point at $V_{DD} = 5.0V$ and $T_A = 25°C$ for $t_{RAS} = 150$ ns and $t_{RC} = 260$ ns. The typical I_{DD} current for a given condition of cycle time, V_{DD} and T_A, can be determined by combining the effects of the appropriate family of curves.

TYPICAL CHARACTERISTICS

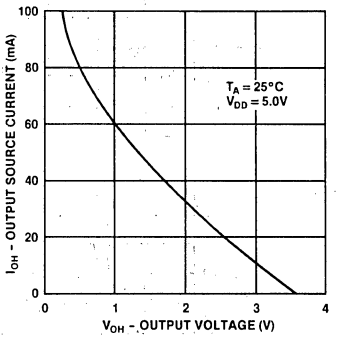
GRAPH 1
TYPICAL ACCESS TIME
t_{RAC} (NORMALIZED) vs. V_{DD}



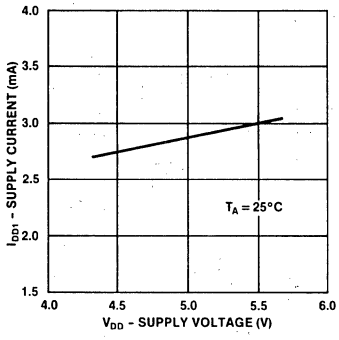
GRAPH 2
TYPICAL ACCESS TIME
t_{RAC} (NORMALIZED) vs.
AMBIENT TEMPERATURE



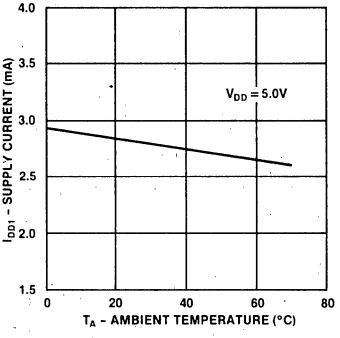
GRAPH 3
TYPICAL OUTPUT
SOURCE CURRENT
I_{OH} vs. OUTPUT VOLTAGE V_{OH}



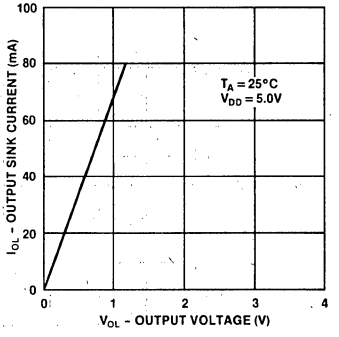
GRAPH 4
TYPICAL STANDBY CURRENT
I_{DD1} vs. V_{DD}



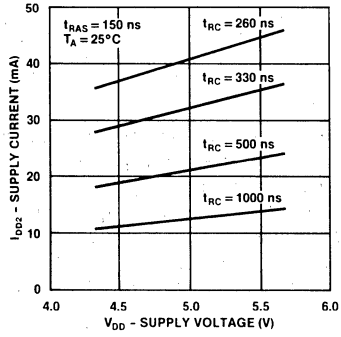
GRAPH 5
TYPICAL STANDBY CURRENT
I_{DD1} vs. AMBIENT TEMPERATURE



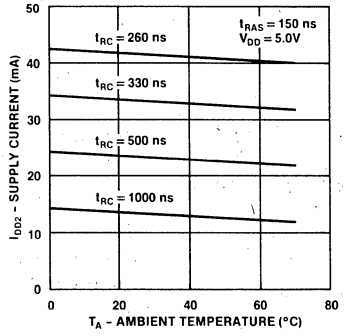
GRAPH 6
TYPICAL OUTPUT
SINK CURRENT
I_{OL} vs. OUTPUT VOLTAGE V_{OL}



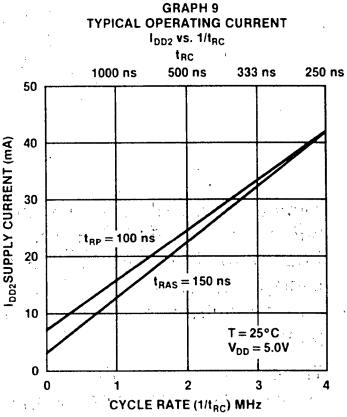
GRAPH 7
TYPICAL OPERATING CURRENT
I_{DD2} vs. V_{DD}



GRAPH 8
TYPICAL OPERATING CURRENT
I_{DD2} vs. AMBIENT TEMPERATURE

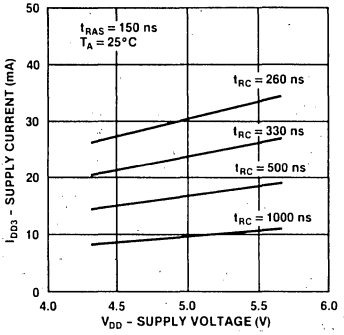


GRAPH 9
TYPICAL OPERATING CURRENT
I_{DD2} vs. 1/t_{RC}

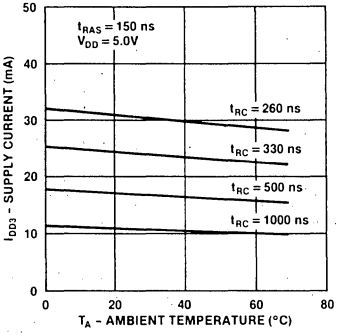


TYPICAL CHARACTERISTICS

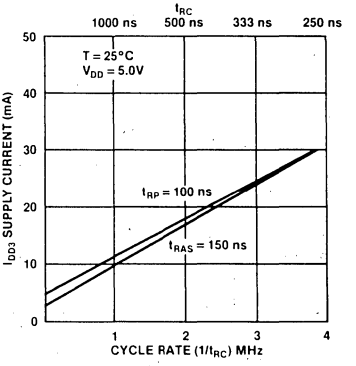
GRAPH 10
TYPICAL $\overline{RAS}$-ONLY
REFRESH CURRENT
I_{DD3} vs. V_{DD}



GRAPH 11
TYPICAL $\overline{RAS}$-ONLY
REFRESH CURRENT
I_{DD3} vs. AMBIENT TEMPERATURE



GRAPH 12
TYPICAL $\overline{RAS}$-ONLY
REFRESH CURRENT
I_{DD3} vs. $1/t_{RC}$



DEVICE DESCRIPTION

The Intel 2164A is produced with HMOS-D III, a high performance MOS technology which incorporates redundant elements. This process, combined with new circuit design concepts, allows the 2164A to operate from a single + 5V power supply, eliminating the + 12V and − 5V requirements. Pin 1 is not connected, which allows P.C.B. layout for future higher density memory generations.

The 2164A is functionally compatible with the 2118, the industry standard 5V-only 16-pin 16K dynamic RAM. This allows simple upgrade from 16K to 64K density merely by adding one additional multiplexed address line.

RAS/CAS Timing

$\overline{RAS}$ and $\overline{CAS}$ have minimum pulse widths as defined by t_{RAS} and t_{CAS} respectively. These minimum pulse widths must be maintained for proper device operation and data integrity. A cycle, once begun by bringing $\overline{RAS}$ and/or $\overline{CAS}$ low, must not be ended or aborted prior to fulfilling the minimum clock signal pulse width(s). A new cycle can not begin until the minimum precharge time, t_{RP}, has been met.

Read Cycle

A Read cycle is performed by maintaining Write Enable ($\overline{WE}$) high during a $\overline{RAS}/\overline{CAS}$ operation. The output pin of a selected device will remain in a high impedance state until valid data appears at the output at access time.

Write Cycle

A Write cycle is performed by taking $\overline{WE}$ low during a $\overline{RAS}/\overline{CAS}$ operation. Data Input (D_{IN} must be valid relative to the negative edge of $\overline{WE}$ or $\overline{CAS}$, whichever transition occurs last.

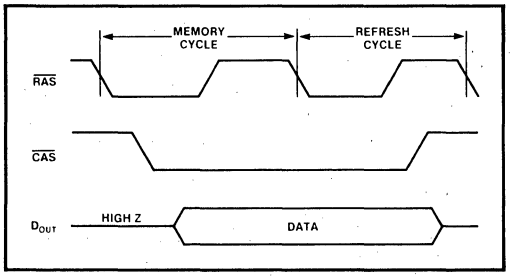
Refresh Cycles

There are 512 sense amplifiers, each controlling 128 storage cells. Thus, the 2164A is refreshed in 128 cycles. Any combination of the seven (7) low order Row Addresses RA_0 through RA_6, will select two rows of data cells (256 cells/row). Row address 7 is not critical during a refresh operation and can be either high or low. Although any cycle, Read, Write, Read-Modify-Write, or $\overline{RAS}$-only, will refresh the memory, the $\overline{RAS}$-only cycle is recommended, since it allows about 20% system power reduction over the other types of cycles.

Hidden Refresh

A standard feature of the 2164A is that refresh cycles may be performed while maintaining valid data at the output pin. This is referred to as Hidden Refresh. Hidden Refresh is performed by holding $\overline{CAS}$ at V_{IL} and taking $\overline{RAS}$ high and, after a specified precharge period (t_{RP}), executing a

"RAS-Only" refresh cycle, but with CAS held low (see figure below).



This feature allows a refresh cycle to be "hidden" among data cycles without affecting the data availability. The part will be internally refreshed at the row addressed at the time of the second RAS.

Data Output Operation

The 2164A Data Output (D_{OUT}), which has three-state capability, is controlled by CAS. During CAS high state (CAS at V_{IH}), the output is in the high impedance state. The following table summarizes the D_{OUT} state for various types of cycles.

Intel® 2164A Data Output Operation for Various Types of Cycles

Type of Cycle	D_{OUT} State
Read Cycle	Data from Addressed Memory Cell
Early Write Cycle	Hi-Z
RAS-Only Refresh Cycle	Hi-Z
CAS-Only Cycle	Hi-Z
Read-Modify-Write Cycle	Data from Addressed Memory Cell
Delayed Write Cycle	Indeterminate

Power On

An initial pause of 500 μs is required after the application of the V_{DD} supply, followed by a minimum of eight (8) initialization cycles (any combination of cycles containing a RAS clock such as RAS-only refresh). 8 initialization cycles are required after extended periods of bias (greater than 2 ms) without clocks. The V_{DD} current (I_{DD}) requirement of the 2164A during power on, is however, dependent upon the input levels of RAS and CAS and the rise time of V_{DD} shown in Figure 1.

If RAS = V_{SS} during power on, the device may go into an active cycle and I_{DD} would show spikes similar to those shown for the RAS/CAS timings. It is recommended that RAS and CAS track with V_{DD} during power on or be held at a valid V_{IH}.

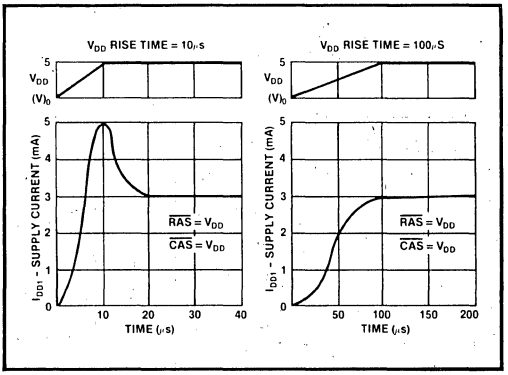


Figure 1. Typical I_{DD} vs. V_{DD} During Power Up

Page Mode Operation

Page Mode operation allows additional columns of the selected device to be accessed at the common row address set. This is done by maintaining RAS low while successive CAS cycles are performed.

Page Mode operation allows a maximum data transfer rate as Row addresses are maintained internally and do not have to be reapplied. During this operation, Read, Write and Read-Modify-Write cycles are possible. Following the entry cycle into Page Mode operation, access is t_{CAC} dependent. The Page Mode cycle is dependent upon CAS pulse width (t_{CAS}) and the CAS precharge period (t_{CP}).

Extended Page Mode Operation

An optional feature of the 2164A is extended page mode operation which allows an entire page (row) of data to be read or written during a single RAS cycle. By providing a fast t_{PC} and long RAS pulse width (t_{RPM2}), the 2164A-15 S6493 permits transfers of large blocks of data, such as required by bit-mapped graphic applications.

SYSTEM DESIGN CONSIDERATIONS

Ground and Power Gridding

Ground and power gridding can contribute to excess noise and voltage drops. An example of an unacceptable method is presented in Figure 2. This type of layout results in accumulated transient noise and voltage drops for the device located at the end of each trace (path).

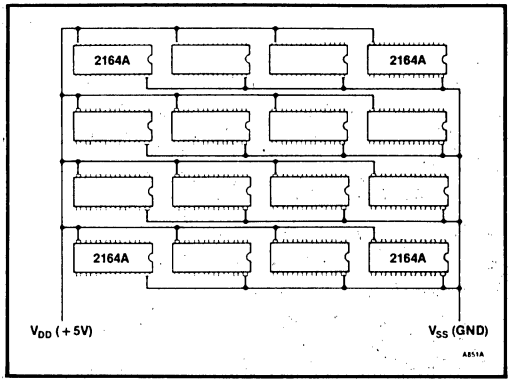


Figure 2. Unacceptable Power Distribution

Transient effects can be minimized by adding extra circuit board traces in parallel to reduce interconnection inductance (Figure 3).

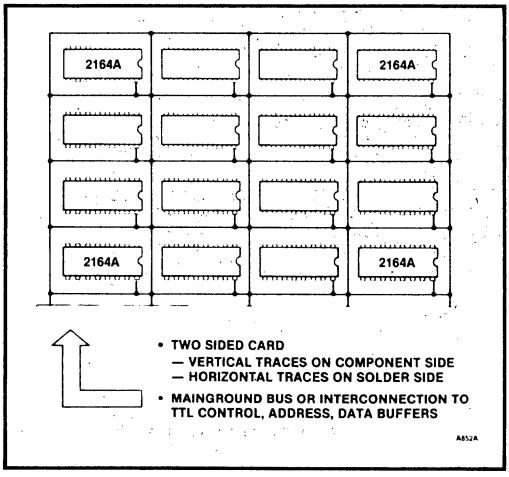


Figure 3. Recommended Power Distribution — Gridding

Power and Ground Plane

A better alternative to power and gridding is power and ground planes. Although this requires two additional inner layers to the PC board, noise and supply voltage fluctuations are greatly reduced. If power and ground planes are used, gridding is optional but typically used for increased reliability of power and ground connections and further reduction of electromagnetic noise.

It is preferable on power/ground planes to use circular voids for device pins rather than slotted voids (Figure 4). This provides maximum decoupling and minimum crosstalk between signal traces.

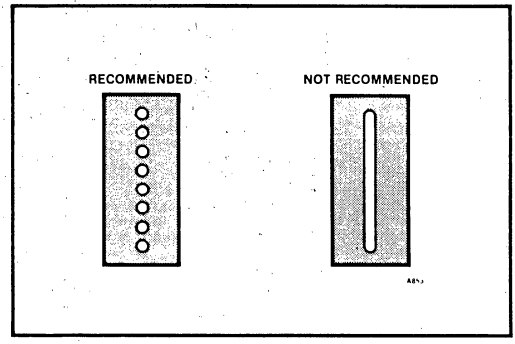


Figure 4. Recommended Voids for Multilayer PC Boards

Power Supply Decoupling

For best results, decoupling capacitors are placed on the memory array board at each memory location (Figure 5). High frequency 0.1 μF ceramic capacitors are the recommended type. Noise is minimized because of the low impedance across the circuit board traces. Typical V_{DD} noise levels for this arrangement are less than 300 mV.

A large tantalum capacitor (typically one 100 μF per 64 devices) is required at the circuit board edge connector power input pins to recharge the 0.1 μF capacitors between memory cycles.

For further details see application note (A.N.) #131, 2164A Dynamic RAM Device Description, or A.N. #133, Designing Memory Systems for Microprocessor Using the Intel 2164A and 2118 Dynamic RAMs.

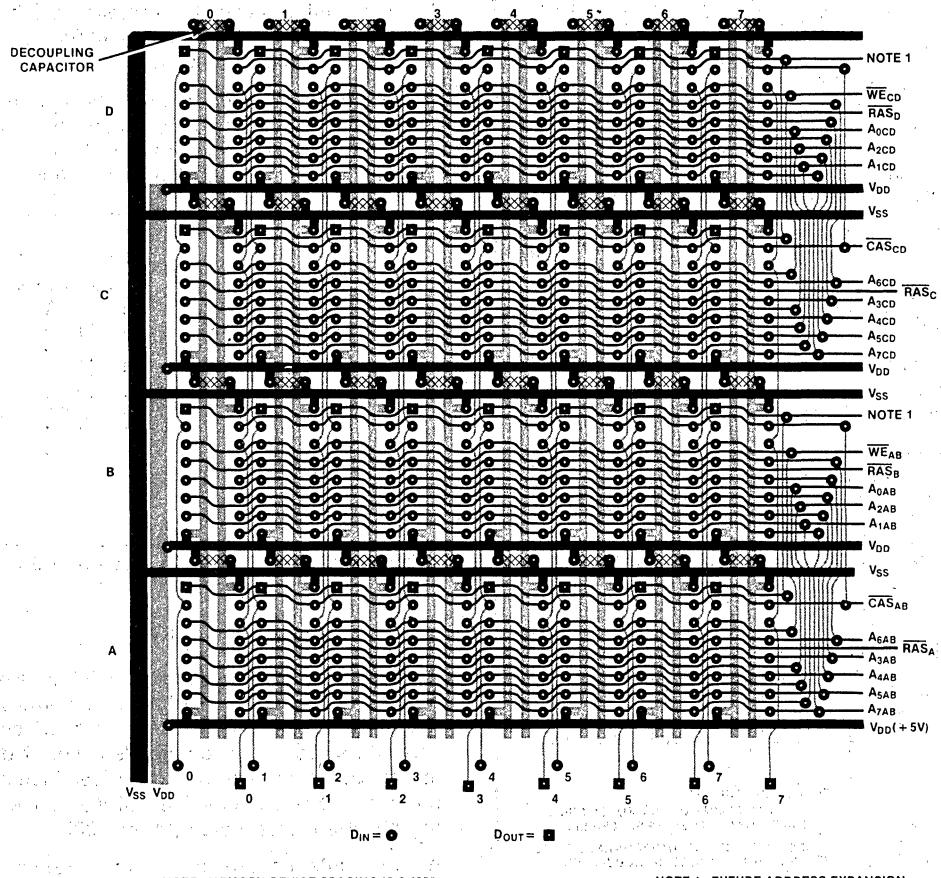


Figure 5. 2164A Memory Array PC Board Layout

PRELIMINARY

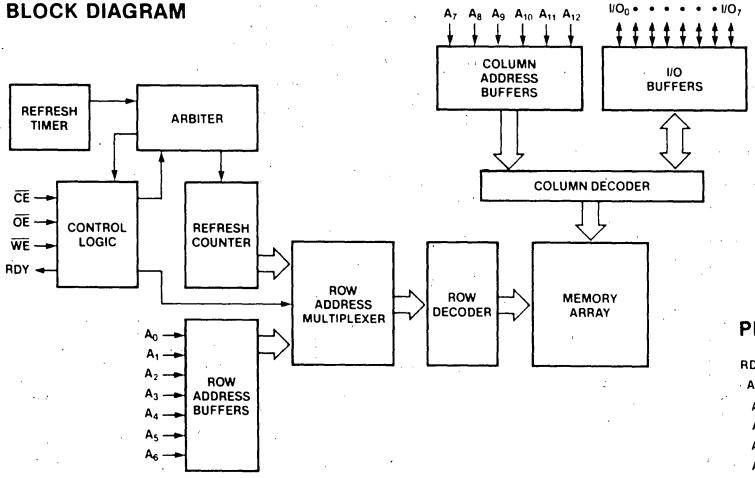
2186A FAMILY
8192 × 8 BIT INTEGRATED RAM

- Low-cost, high volume HMOS III technology
- High density one transistor cell
- Single +5V ± 10% supply
- Proven HMOS reliability
- Low active current (70 mA)
- Simple asynchronous refresh operation/ static RAM compatible
- 2764 EPROM compatible pin-out
- Two-line bus control
- JEDEC standard 28-pin site
- Low standby current (20 mA)

The Intel 2186A is a 8192 word by 8-bit integrated random access memory (iRAM) fabricated on Intel's proven HMOS dynamic RAM technology. Integrated refresh control provides static RAM characteristics at a significantly lower cost. Packaged in the industry standard 28-pin DIP, the 2186A conforms to the industry standard JEDEC 28-pin site. Designs based on 2186A timings can be made fully compatible with EPROMs and static RAMs.

The 2186A is particularly suited for microprocessor applications and incorporates many requisite system features including low power dissipation, automatic initialization, extended cycle operation and two-line bus control to eliminate bus contention.

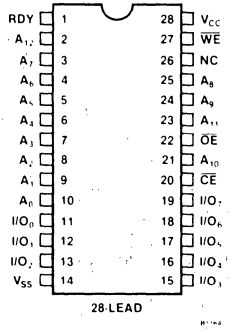
BLOCK DIAGRAM



PIN NAMES

A_0-A_{12}	ADDRESS INPUTS
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
$\overline{WE}$	WRITE ENABLE
I/O_0-I/O_7	DATA INPUT/OUTPUT
RDY	READY
V_{CC}	+5V POWER
V_{SS}	GROUND

PIN CONFIGURATION



28-LEAD

OCTOBER 1983
ORDER NO. 230867-001

ERRATA ENCLOSED PRELIMINARY

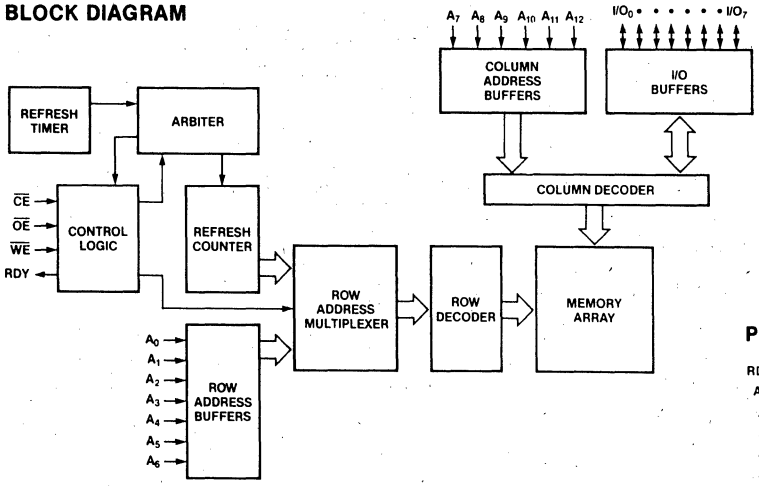
2186 S7572/3/4
8192 x 8 BIT INTEGRATED RAM

- Low-cost, high-volume HMOS technology
- High density one transistor cell
- Single +5V ± 10% supply
- Proven HMOS reliability
- Low active current (70 mA)
- Simple asynchronous refresh operation/ static RAM compatible
- 2764 EPROM compatible pin-out
- Two-line bus control
- JEDEC standard 28-pin site
- Low standby current (20 mA)

The Intel 2186 is a 8192 word by 8-bit integrated random access memory (iRAM) fabricated on Intel's proven HMOS dynamic RAM technology. Integrated refresh control provides static RAM characteristics at a significantly lower cost. Packaged in the industry standard 28-pin DIP, the 2186 conforms to the industry standard JEDEC 28-pin site. Designs based on 2186 timings can be made fully compatible with EPROMs and static RAMs.

The 2186 is particularly suited for microprocessor applications and incorporates many requisite system features including low power dissipation, automatic initialization, extended cycle operation and two-line bus control to eliminate bus contention.

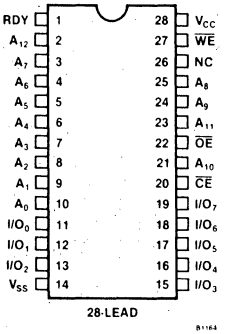
BLOCK DIAGRAM



PIN NAMES

A_0-A_{12}	ADDRESS INPUTS
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
$\overline{WE}$	WRITE ENABLE
I/O_0-I/O_7	DATA INPUT/OUTPUT
RDY	READY
V_{CC}	+5V POWER
V_{SS}	GROUND

PIN CONFIGURATION



28-LEAD

B1164

April 1983
Order Number: 210857-002

8192 x 8 BIT INTEGRATED RAM

ERRATA

CHANGE IN THE INITIALIZATION PROCEDURE

256 initialization cycles are required to guarantee initialization. These may be Read, Write, or False Memory cycles. The cycle timing for $\overline{CE}$ (and $\overline{WE}$ or $\overline{OE}$) must meet the minimum cycle time requirements shown in the A.C. characteristics. The states of the address and data lines ($A_{\emptyset}$-A_{12} and $IO_{\emptyset}$-IO_7) are not critical.

Initialization cycles may begin anytime after V_{CC} is within specification and after $\overline{CE}$, $\overline{OE}$, and $\overline{WE}$ have reached V_{IH}. Normal operation may begin anytime after the 256 initialization cycles have been completed.

This initialization procedure replaces the initialization procedure described on page 6 and applies only to S spec parts (numbers S7572, S7573, and S7574).

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with Respect to Ground ..... −1.0 V to +7 V
D.C. Continuous Current per Output ..... 10 mA
D.C. Maximum Data Out Current ..... 50 mA
D.C. Power Dissipation ..... 1.0 W

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. AND OPERATING CHARACTERISTICS[1]

TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

Symbol	Parameter	Limits Min.	Limits Max.	Unit	Test Conditions	Notes
ILI	Input Load Current (All Input Pins)		10	μA	V_{IN} = VSS to VCC	
|ILO|	Output Leakage Current		10	μA	$\overline{OE}$ = VIH	
ICC	Operating Current		70	mA	Minimum Cycle Time	2
ISB	Standby Current		20	mA	$\overline{CE}$ = VIH	
VIL	Input Low Voltage	−1.0	0.8	V		3
VIH	Input High Voltage	2.4	7.0	V		
VOL	Output Low Voltage		0.6	V	IOL = 2.1 mA	4
VOH	Output High Voltage	2.4		V	IOH = −1.0 mA	

NOTES FOR D.C. CHARACTERISTICS
1. Typical limits are VCC = +5V, TA = 25°C.
2. ICC is dependent on outputs loading when the device output is selected. Specified ICC max. is measured with the outputs open.
3. Specified VIL min. is for steady state operation. During transmisstions the inputs may overshoot to −2.0V for periods not to exceed 20 nsec.
4. IOL for RDY is 4 mA and VOL for RDY is 0.8V.

A.C. TEST CONDITIONS

Input Pulse and Timing Reference Levels ..... 0.8V to 2.4V
Input Rise and Fall Times ..... 10 nsec
Output Timing Reference Levels .. 0.8V and 2.0V
Output Load ..... See Figure 1

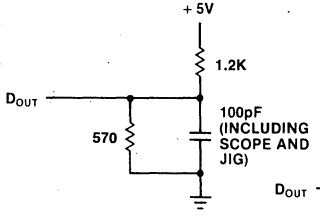


Figure 1.

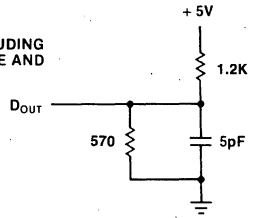


Figure 2.
(FOR HIGH IMPEDANCE MEASUREMENTS ONLY)

CAPACITANCE[5]

TA = 25°C, f = 1.0 MHz

Symbol	Parameter	Max.	Unit	Conditions
C_{ADD}	Address Capacitance	8	pF	V_{ADD} = 0V
$C_{I/O}$	I/O Capacitance	14	pF	$V_{I/O}$ = 0V
C_{IN}	Control Capacitance	14	pF	V_{IN} = 0V

NOTE: 5. This parameter is characterized and not 100% tested.

A.C. CHARACTERISTICS

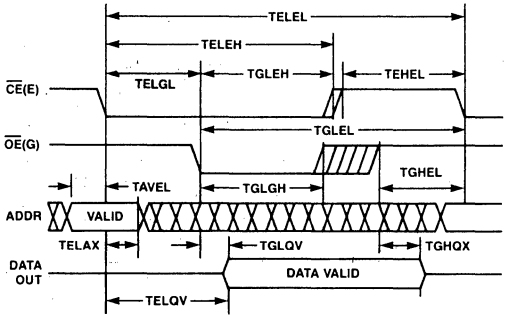
TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

READ CYCLE ($\overline{WE}$ = VIH)

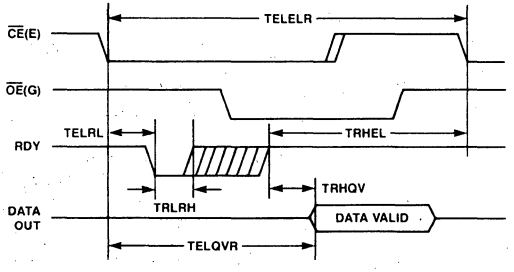
		S7572		S7573		S7574			
		2186-25		2186-30		2186-35			
Symbol	Parameter	Min.	Max.	Min.	Max.	Min.	Max.	Unit	Notes
TELEL	Cycle Time	425		500		600		ns	1
TELQV	Access Time from $\overline{CE}$		250		300		350	ns	1
TELELR	Cycle Time with Refresh	850		1000		1200		ns	2
TELQVR	Access Time from $\overline{CE}$ w/Refresh		675		800		950	ns	2
TGLQV	Access Time from $\overline{OE}$		65		70		75	ns	
TELEH	$\overline{CE}$ Pulse Width	40		40		40		ns	
TEHEL	$\overline{CE}$ High Time	40		40		40		ns	
TAVEL	Address Set-Up Time	0		0		0		ns	
TELAX	Address Hold Time	35		35		35		ns	
TGLEL	$\overline{OE}$ low to next $\overline{CE}$ low	250		275		300		ns	
TGLGH	$\overline{OE}$ Pulse Width	65		70		75		ns	
TGHEL	$\overline{OE}$ high to next $\overline{CE}$ low	40		40		40		ns	
TGHQX	$\overline{OE}$ high to Data Float	10	60	10	60	10	60	ns	3
TELGL	$\overline{CE}$ low to $\overline{OE}$ low		2000		2000		2000	ns	
TGLEH	$\overline{OE}$ low to $\overline{CE}$ high	40		40		40		ns	
TELRL	$\overline{CE}$ low to RDY low		50		60		70	ns	4
TRLRH	RDY Pulse Width	100		100		100		ns	4
TRHQV	RDY high to Data Valid		60		70		95	ns	
TRHEL	RDY high to next $\overline{CE}$ low	250		275		350		ns	

WAVEFORMS

READ CYCLE



READ CYCLE WITH REFRESH



A.C. CHARACTERISTICS

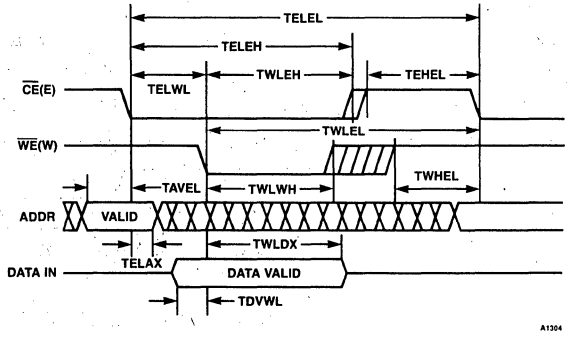
TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

WRITE CYCLE ($\overline{OE}$ = VIH)

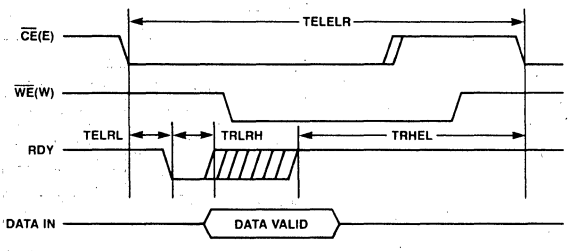
		S7572		S7573		S7574			
		2186-25		2186-30		2186-35			
Symbol	Parameter	Min.	Max.	Min.	Max.	Min.	Max.	Unit	Notes
TELEL	Cycle Time	425		500		600		ns	1
TELELR	Cycle Time with Refresh	850		1000		1200		ns	2
TELEH	$\overline{CE}$ Pulse Width	40		40		40		ns	
TEHEL	$\overline{CE}$ High Time	40		40		40		ns	
TAVEL	Address Set-Up Time	0		0		0		ns	
TELAX	Address Hold Time	35		35		35		ns	
TWLEL	$\overline{WE}$ low to next $\overline{CE}$ low	250		300		350		ns	
TWLWH	$\overline{WE}$ Pulse Width	40		40		40		ns	
TWHEL	$\overline{WE}$ high to next $\overline{CE}$ low	40		40		40		ns	
TDVWL	Data Set-Up to $\overline{WE}$ low	0		0		0		ns	
TWLDX	Data Hold from $\overline{WE}$ low	40		45		50		ns	
TELWL	$\overline{CE}$ low to $\overline{WE}$ low		2000		2000		2000	ns	
TWLEH	$\overline{WE}$ low to $\overline{CE}$ high	40		40		40		ns	
TELRL	$\overline{CE}$ low to RDY low		50		60		70	ns	4
TRLRH	RDY Pulse Width	100		100		100		ns	4
TRHEL	RDY high to next $\overline{CE}$ low	250		275		350		ns	

WAVEFORMS

WRITE CYCLE



WRITE CYCLE WITH REFRESH



A.C. CHARACTERISTICS

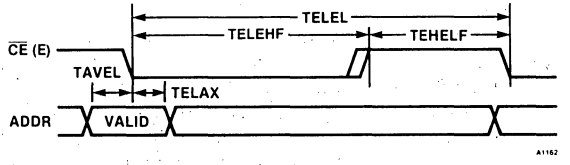
TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

FALSE MEMORY CYCLE ($\overline{OE}$ and $\overline{WE}$ = VIH)

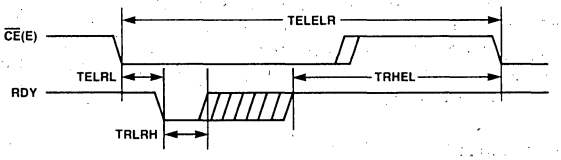
		S7572		S7573		S7574			
		2186-25		2186-30		2186-35			
Symbol	Parameter	Min.	Max.	Min.	Max.	Min.	Max.	Unit	Notes
TELEL	Cycle Time	425		500		600		ns	1
TELELR	Cycle Time with Refresh	850		1000		1200		ns	2
TELEHF	$\overline{CE}$ Pulse Width	40	2000	40	2000	40	2000	ns	5
TEHELF	$\overline{CE}$ High Time during F.M.C.	200		250		275		ns	6
TAVEL	Address Set-Up Time	0		0		0		ns	
TELAX	Address Hold Time	35		35		35		ns	
TELRL	$\overline{CE}$ low to RDY low		50		60		70	ns	4
TRLRH	RDY Pulse Width	100		100		100		ns	4
TRHEL	RDY high to next $\overline{CE}$ low	250		275		350		ns	

WAVEFORMS

FALSE MEMORY CYCLE



FALSE MEMORY CYCLE WITH REFRESH



NOTES FOR A.C. CHARACTERISTICS:

1. TELEL < TELELR and TELQV < TELQVR.
2. For reference only.
3. Transition is measured ±500 mV from steady state logic level with specified loading in Figure 2.
4. C_{RDY} < 100 pF and R_{EXT} = 1200Ω.
5. False Memory Cycles Only.
6. Note TEHELF > TEHEL.

FUNCTIONAL DESCRIPTION

The 2186 has three control pins: $\overline{CE}$ (Chip Enable), $\overline{OE}$ (Output Enable), and $\overline{WE}$ (Write Enable). An open-drain output pin called RDY indicates if refresh is occurring during an access request. *RDY will only respond when the 2186 has been selected by $\overline{CE}$ going active low during a refresh cycle.*

Cycles are initiated by latching addresses into the 2186 with the leading (falling) edge of $\overline{CE}$. When $\overline{CE}$ goes active during internal refresh, the RDY pin is pulled low signaling a delay. RDY remains low until shortly before both refresh and access (Read/Write) cycles are complete.

On-chip control circuitry tracks all operations for nearly transparent refresh. A high-speed on-chip arbitration circuit prevents conflicts from occurring between refresh and access cycles.

Access Cycles

READ CYCLE

A read cycle is initiated by $\overline{CE}$ and $\overline{OE}$ both going active low during the same cycle. $\overline{CE}$ may be either pulsed to initiate a cycle or held active low throughout the cycle. $\overline{OE}$ is a logic level; $\overline{OE}$ controls the 2186 data output bus. Access times are specified from both $\overline{OE}$ and $\overline{CE}$. Data remains on the data bus until $\overline{OE}$ returns inactive (high) *independent of $\overline{CE}$*. $\overline{WE}$ may not go active during a Read cycle.

WRITE CYCLE

A Write cycle is initiated by $\overline{CE}$ and $\overline{WE}$ going active low during the same cycle. $\overline{CE}$ may be a pulse or a logic level. *$\overline{WE}$ leading edge* latches data from the data bus into the 2186. $\overline{OE}$ may not go active during a Write cycle.

FALSE MEMORY CYCLE (FMC)

A False Memory cycle is initiated by $\overline{CE}$ going active *without* either $\overline{OE}$ or $\overline{WE}$ going active. No memory cycle will be performed. Note that address set-up and hold times must be observed for False Memory cycle operation.

Operating Modes

REFRESH OPERATION

Refresh is totally automatic and requires no external stimulus. All refresh functions are controlled internally.

A high-speed arbitration circuit will resolve any potential conflict arising between simultaneous external access and internal refresh cycle requests. The internal timer period is specified as $\geq 5\mu$sec.

The 2186 may also be refreshed by performing Read, Write, or False Memory cycles on all 128 rows (A0 through A6) within a two millisecond period.

EXTENDED CYCLE OPERATION

Extended cycle operation is defined as holding $\overline{OE}$ or $\overline{WE}$ valid (low) for indefinite periods. ($\overline{CE}$ is allowed to return high.) Data will remain valid on the bus as long as $\overline{OE}$ is valid. $\overline{WE}$ latches data on the leading (falling) edge. Automatic refreshes will continue to be performed as needed, even while $\overline{OE}$ or $\overline{WE}$ is held low; RDY will *not* respond during these extended cycle refreshes.

INITIALIZATION

To guarantee initialization, all control inputs must be inactive (high) for a 100 microsecond period after V_{CC} is within specification. No extra cycles are required before normal operation may begin.

Interfacing Considerations

The 2186 is an edge enabled RAM. Below is an illustration of a simple interface for connecting microprocessors with edge enabled memories. *A stable $\overline{CE}$ clock is necessary to avoid accidentally selecting the RAM.* Generally, stable select signals are desirable in all types of microsystem applications. Most common decoding circuits allow addresses to flow directly through the decoder (i.e. decoder permanently "enabled"). This technique may allow false decoder outputs to occur when addresses are in transition. This may result in false $\overline{CE}$ signals and potentially, invalid memory requests. A simple gating circuit will inhibit enabling the decoder until addresses are valid at the decoder inputs.

Another interfacing consideration is the relationship between $\overline{WE}$ and valid data. The 2186 performs a write operation on the leading edge of $\overline{WE}$. In a minimum mode 8088 or 8086 system, $\overline{WE}$ occurs before data is valid. The cross-coupled NAND gate configuration shown below on the $\overline{WR}$ signal will prevent this from occurring. This implementation also guarantees valid data on the rising (trailing) edge of $\overline{WE}$ to maintain compatibility with fully static RAMs. (For maximum mode 8088 or 8086 operation, the control signal $\overline{MWTC}$ directly from the 8288 bus controller serves the same function.) For a more detailed description of designing iRAM

systems, refer to Intel App. Note #132 on *"Designing Memory Systems with the 8K × 8 iRAM"*.

Layout Considerations

To ensure compatibility with other 28-pin memory devices such as EPROMs, several pins require close examination; specifically, pins number 1, 26 and 27. Following is a discussion of the system level operation and the design considerations for these pins.

PIN #1

Pin 1 on all EPROMs is reserved for the high voltage programming bias, V_{PP}. EPROMs are usually programmed external to the system. Therefore, in normal system operation, pin 1 is connected to V_{CC}.

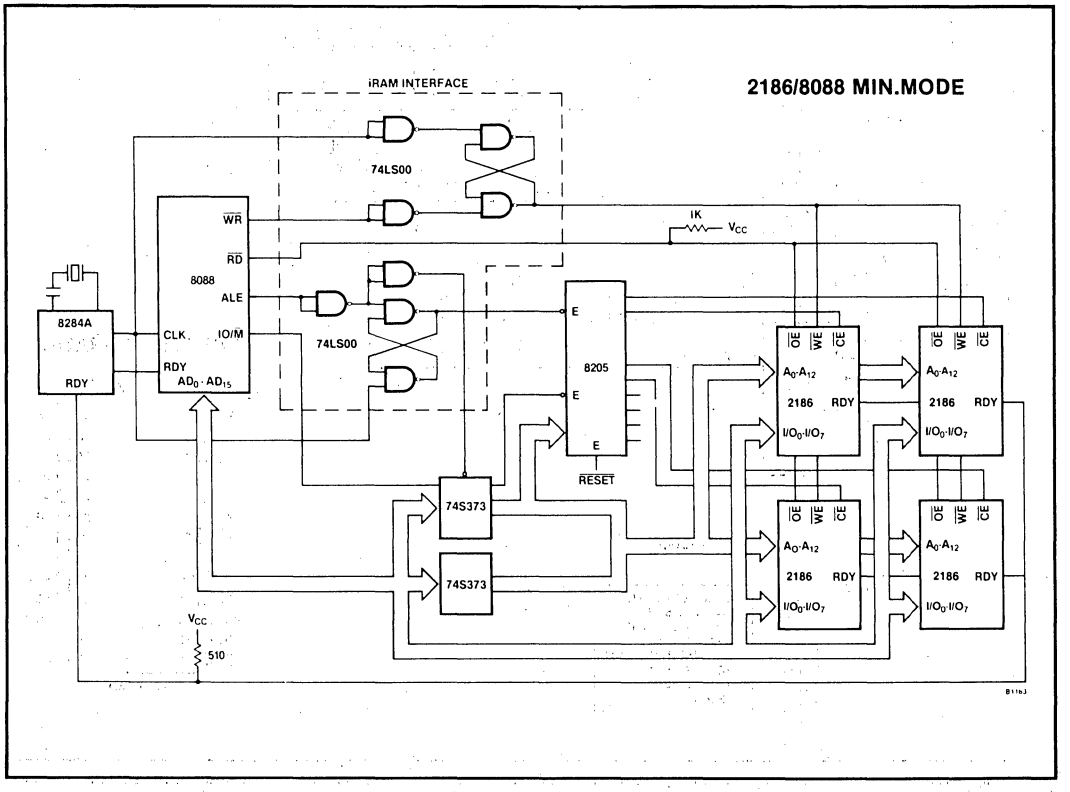
Pin 1 on the 2186 is the microprocessor handshake signal, RDY. The RDY signal may be bussed to the RDY input of either the microprocessor or clock generator. Because RDY is an open drain output, all 2186 RDY signals may be "wire OR'd" with any other RDY signals in the system. A 1200 ohm pull-up resistor is required between RDY and V_{CC}. For the 2186 application, a trace should be run from pin 1 of each socket location to the RDY input of either the microprocessor or clock generator. Also, a provision for a pull-up resistor to V_{CC} is needed. For compatibility with EPROMs, include a jumper to select V_{CC} and the RDY line on pin 1.

PIN #26

While pin 26 is a No Connect for both the 2186 and the 2764 EPROM, a trace to pin 26 from V_{CC} will guarantee compatibility between 24 pin and 28 pin EPROMs. Pin 26 will carry the additional address bit required to future higher density memories. For flexibility, provide a jumper for an address bit and/or V_{CC} on pin 26.

PIN #27

Pin 27 is labelled $\overline{WE}$ on the RAM and $\overline{PGM}$ on the EPROM. While $\overline{WE}$ is a system level control signal, $\overline{PGM}$ is only used when programming the EPROM (V_{PP} at +21V). $\overline{PGM}$ may be allowed to toggle during normal EPROM operation (V_{PP} at +5V). Therefore, $\overline{WE}$ may be bussed to every socket location with no jeopardy of illegal operation.



PRELIMINARY

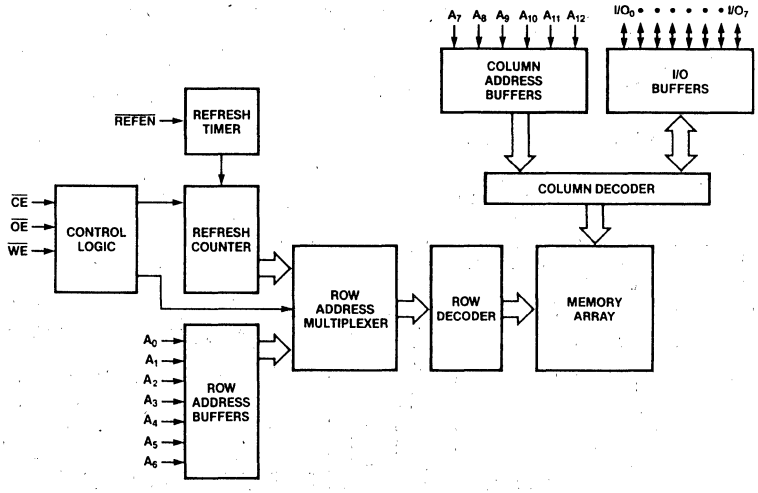
2187A FAMILY
8192 x 8 BIT INTEGRATED RAM

- Low-cost, high-volume HMOS technology
- High density one transistor cell
- Single +5V ± 10% supply
- Proven HMOS Reliability
- Low active current (70 mA)
- Simple synchronous refresh operation
- 2764 EPROM compatible pin-out
- Two-line bus control
- JEDEC standard 28-pin site
- Low standby current (20 mA)

The Intel 2187 is an 8192 word by 8-bit integrated random access memory (iRAM) fabricated on Intel's proven HMOS dynamic RAM technology. Packaged in the industry standard 28-pin DIP, the 2187 conforms to the industry standard JEDEC 28-pin site.

The 2187 is particularly suited for use in microcontroller applications, incorporating many requisite system features. These include low power dissipation, automatic initialization, extended cycle operation and two-line bus control to eliminate bus contention.

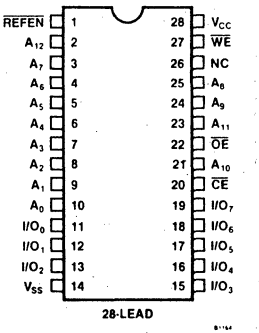
BLOCK DIAGRAM



PIN NAMES

A_0-A_{12}	ADDRESS INPUTS
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
$\overline{WE}$	WRITE ENABLE
I/O_0-I/O_7	DATA INPUT/OUTPUT
$\overline{REFEN}$	REFRESH ENABLE
V_{CC}	+5V POWER
V_{SS}	GROUND

PIN CONFIGURATION



28-LEAD

April, 1983
Order Number: 210859-001

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground ..... −1.0V to +7V
D.C. Continuous Current per Output ..... 10 mA
D.C. Maximum Data Out Current ..... 50 mA
D.C. Power Dissipation ..... 1.0 W

** COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. AND OPERATING CHARACTERISTICS[1]

TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

Symbol	Parameter	Limits Min.	Limits Max.	Unit	Test Conditions	Notes
ILI	Input Load Current (All Input Pins)		10	μA	V_{IN} = VSS to VCC	
|ILO|	Output Leakage Current		10	μA	$\overline{OE}$ = VIH	
ICC	Operating Current		70	mA	Minimum Cycle Time	2
ISB	Standby Current		20	mA	$\overline{CE}$ = VIH	
VIL	Input Low Voltage	−1.0	0.8	V		3
VIH	Input High Voltage	2.4	7.0	V		
VOL	Output Low Voltage		0.45	V	IOL = 2.1 mA	
VOH	Output High Voltage	2.4		V	IOH = −1.0 mA	

NOTES FOR D.C. CHARACTERISTICS:
1. Typical limits are VCC = +5V, TA = 25°C.
2. ICC is dependent on outputs loading when the device output is selected. Specified ICC max. is measured with the outputs open.
3. Specified VIL min. is for steady state operation. During transitions the inputs may overshoot to −2.0V for periods not to exceed 20 nsec.

A.C. TEST CONDITIONS

Input Pulse and Timing
Reference Levels ..... 0.8V to 2.4V
Input Rise and Fall Times ..... 10 nsec.
Output Timing Reference Levels .... 0.6V and 2.4V
Output Load ..... See Figure 1

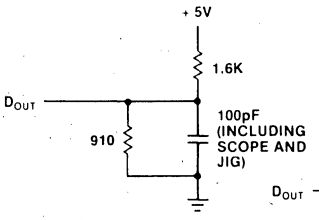


Figure 1.

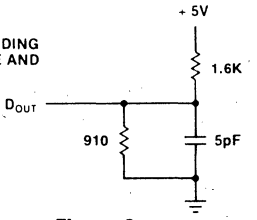


Figure 2.

(FOR HIGH IMPEDANCE MEASUREMENTS ONLY)

CAPACITANCE[4]

TA = 25°C, f = 1.0 MHz

Symbol	Parameter	Max.	Unit	Conditions
C_{ADD}	Address Capacitance	8	pF	V_{ADD} = 0V
$C_{I/O}$	I/O Capacitance	14	pF	$V_{I/O}$ = 0V
C_{IN}	Control Capacitance	14	pF	V_{IN} = 0V

NOTE: 4. This parameter is characterized and not 100% tested.

A.C. CHARACTERISTICS

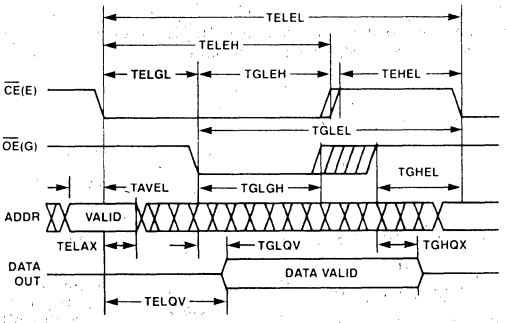
TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted

READ CYCLE ($\overline{WE}$ = VIH)

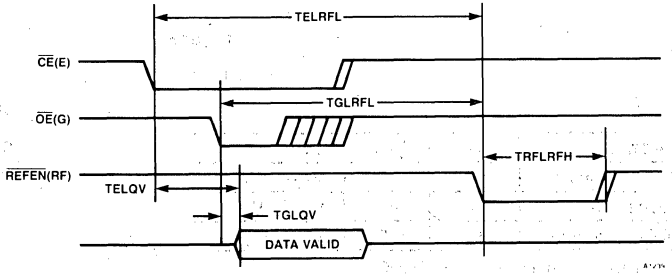
Symbol	Parameter	2187-25		2187-30		2187-35		Unit	Notes
		Min.	Max.	Min.	Max.	Min.	Max.		
TELEL	Cycle Time	425		500		600		ns	
TELQV	Access Time from $\overline{CE}$		250		300		350	ns	
TGLQV	Access Time from $\overline{OE}$		65		70		75	ns	
TELEH	$\overline{CE}$ Pulse Width	40		40		40		ns	
TEHEL	$\overline{CE}$ High Time	40		40		40		ns	
TAVEL	Address Set-Up Time	0		0		0		ns	
TELAX	Address Hold Time	35		35		35		ns	
TGLEL	$\overline{OE}$ low to next $\overline{CE}$ low	250		275		300		ns	
TGLGH	$\overline{OE}$ Pulse Width	65		70		75		ns	
TGHEL	$\overline{OE}$ high to next $\overline{CE}$ low	40		40		40		ns	
TGHQX	$\overline{OE}$ high to Data Float	0	70	0	70	0	70	ns	1
TELGL	$\overline{CE}$ low to $\overline{OE}$ low		10,000		10,000		10,000	ns	
TGLEH	$\overline{OE}$ low to $\overline{CE}$ high	40		40		40		ns	
TELRFL	$\overline{CE}$ low to $\overline{REFEN}$ low	425		500		600		ns	
TGLRFL	$\overline{OE}$ low to $\overline{REFEN}$ low	250		275		300		ns	
TRFLRFH	$\overline{REFEN}$ Pulse Width	70	9800	70	9800	70	9800	ns	5

WAVEFORMS

READ CYCLE



READ CYCLE FOLLOWED BY REFRESH



A.C. CHARACTERISTICS

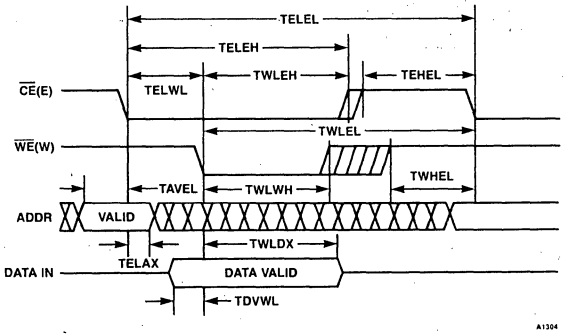
TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

WRITE CYCLE ($\overline{OE}$ = VIH)

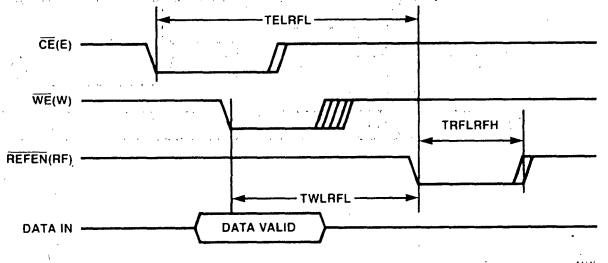
Symbol	Parameter	2187-25		2187-30		2187-35		Unit	Notes
		Min.	Max.	Min.	Max.	Min.	Max.		
TELEL	Cycle Time	425		500		600		ns	
TELEH	$\overline{CE}$ Pulse Width	40		40		40		ns	
TEHEL	$\overline{CE}$ High Time	40		40		40		ns	
TAVEL	Address Set-Up Time	0		0		0		ns	
TELAX	Address Hold Time	35		35		35		ns	
TWLEL	$\overline{WE}$ low to next $\overline{CE}$ low	250		300		350		ns	
TWLWH	$\overline{WE}$ Pulse Width	40		40		40		ns	
TWHEL	$\overline{WE}$ high to next $\overline{CE}$ low	40		40		40		ns	
TDVWL	Data Set-Up to $\overline{WE}$ low	0		0		0		ns	
TWLDX	Data Hold from $\overline{WE}$ low	40		45		50		ns	
TELWL	$\overline{CE}$ low to $\overline{WE}$ low		10,000		10,000		10,000	ns	
TWLEH	$\overline{WE}$ low to $\overline{CE}$ high	40		40		40		ns	
TELRFL	$\overline{CE}$ low to $\overline{REFEN}$ low	425		500		600		ns	
TWLRFL	$\overline{WE}$ low to $\overline{REFEN}$ low	250		300		350		ns	
TRFLRFH	$\overline{REFEN}$ Pulse Width	70	9800	70	9800	70	9800	ns	5

WAVEFORMS

WRITE CYCLE



WRITE CYCLE FOLLOWED BY REFRESH



A.C. CHARACTERISTICS

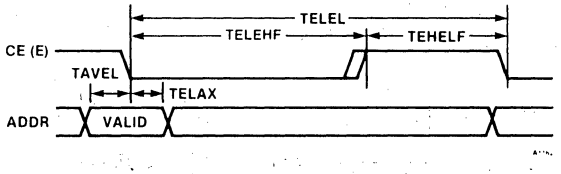
TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

FALSE MEMORY CYCLE ($\overline{OE}$ and $\overline{WE}$ = VIH)

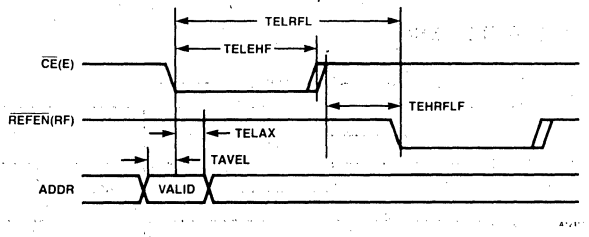
Symbol	Parameter	2187-25		2187-30		2187-35		Unit	Notes
		Min.	Max.	Min.	Max.	Min.	Max.		
TELEL	Cycle Time	425		500		600		ns	
TELEHF	$\overline{CE}$ Pulse Width	40	10,000	40	10,000	40	10,000	ns	2
TEHELF	$\overline{CE}$ High Time during F.M.C.	200		250		275		ns	3
TAVEL	Address Set-Up Time	0		0		0		ns	
TELAX	Address Hold Time	35		35		35		ns	
TELRFL	$\overline{CE}$ low to $\overline{REFEN}$ low	425		500		600		ns	
TEHRFLF	$\overline{CE}$ high to $\overline{REFEN}$ low after F.M.C.	200		250		275		ns	

WAVEFORMS

FALSE MEMORY CYCLE



FALSE MEMORY CYCLE FOLLOWED BY A REFRESH



A.C. CHARACTERISTICS

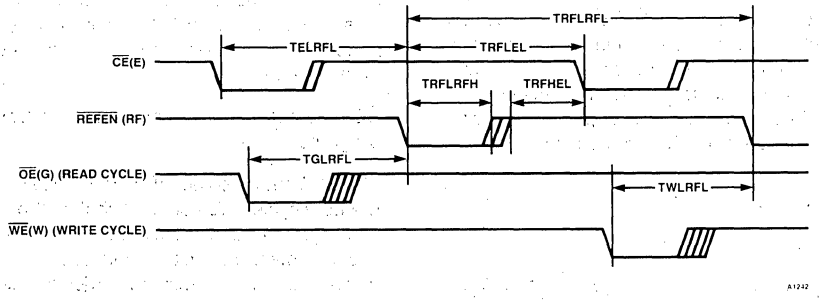
TA = 0°C to +70°C, VCC = +5V ±10% unless otherwise noted.

REFRESH CYCLE

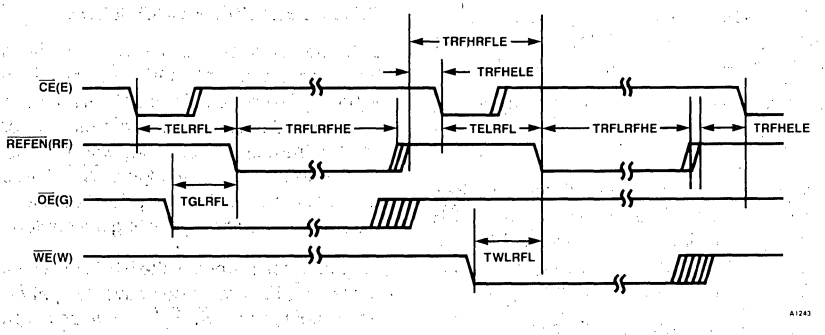
Symbol	Parameter	2187-25		2187-30		2187-35		Unit	Notes
		Min.	Max.	Min.	Max.	Min.	Max.		
TELRFL	$\overline{CE}$ low to $\overline{REFEN}$ low	425		500		600		ns	
TRFLRFH	$\overline{REFEN}$ Pulse Width	70	9800	70	9800	70	9800	ns	5
TRFHEL	$\overline{REFEN}$ high to $\overline{CE}$ low	40		40		40		ns	
TRFLRFL	$\overline{REFEN}$ cycle time to guarantee refresh	350	15600	400	15600	450	15600	ns	
TRFLEL	$\overline{REFEN}$ low to $\overline{CE}$ low	350		400		450		ns	
TRFHELE	$\overline{REFEN}$ high to $\overline{CE}$ low — extended cycle	350		400		450		ns	4
TGLRFL	$\overline{OE}$ low to $\overline{REFEN}$ low	250		275		300		ns	
TWLRFL	$\overline{WE}$ low to $\overline{REFEN}$ low	250		300		350		ns	
TRFLRFHE	$\overline{REFEN}$ Pulse Width — extended cycle	10,000		10,000		10,000		ns	4,5
TRFHRFLE	$\overline{REFEN}$ high to $\overline{REFEN}$ low — extended cycle	425		500		600		ns	

WAVEFORMS

REFRESH CYCLE



EXTENDED CYCLE REFRESH[5]



NOTES FOR A.C. CHARACTERISTICS:

1. Transition is measured ±500 mV from steady state logic level with specified loading in Figure 2.
2. Maximum applies for F.M.C. Only.
3. Note TEHELF > TEHEL.
4. TRFHELE > TRFHEL and TRFLRFHE > TRFLRFH.
5. Extended cycles occur when the $\overline{REFEN}$ pulse width is ≥10 μsec.

FUNCTIONAL DESCRIPTION

The 2187 has four control pins: $\overline{CE}$ (Chip Enable), $\overline{OE}$ (Output Enable), $\overline{WE}$ (Write Enable), and $\overline{REFEN}$ (Refresh Enable). These control lines select and control the operation of the iRAM.

$\overline{CE}$ is the general purpose chip enable line. It is an edge triggered signal that controls several internal operations. An access cycle begins with the leading (falling) edge of $\overline{CE}$. At this time, the external address is latched into the 2187 and is held throughout the current cycle. $\overline{CE}$ may be pulsed or it may remain low throughout the cycle.

$\overline{OE}$ selects a read cycle and controls the output data bus. When $\overline{OE}$ goes active (low) during a read cycle, the output drivers of the 2187 are enabled. Data remains valid on the output lines as long as $\overline{OE}$ is active — independent of the state of $\overline{CE}$.

$\overline{WE}$ is the edge triggered input that defines a write cycle. During write cycles, data is latched from the external bus into the 2187 by the leading (falling) edge of $\overline{WE}$. $\overline{WE}$ and $\overline{OE}$ may not be active during the same cycle.

$\overline{REFEN}$ initiates the internal refresh cycles of the 2187. A refresh cycle begins whenever $\overline{REFEN}$ goes active (low). An internal refresh address counter provides the refresh address. To prevent conflicts between refresh and access cycles, several timing parameters, referenced to $\overline{REFEN}$, must be met. These parameters are specified in the timing tables of the 2187 A.C. Characteristics.

Access Cycles

READ CYCLE

A read cycle is initiated when both $\overline{CE}$ and $\overline{OE}$ go active low during the same cycle. Access times are specified from both $\overline{OE}$ and $\overline{CE}$. After $\overline{OE}$ goes active (low), and the hold time for $\overline{CE}$ has been met (TGLEH), $\overline{CE}$ may go inactive. As long as $\overline{OE}$ remains active (low), data remains on the output lines — independent of the level of $\overline{CE}$.

$\overline{WE}$ may not go active (low) during the read cycle.

WRITE CYCLE

A write cycle occurs when both $\overline{CE}$ and $\overline{WE}$ go active (low) in the same cycle. Once $\overline{WE}$ has gone active (low), $\overline{CE}$ may go inactive after a minimum hold time (TWLEH). The leading (falling) edge of $\overline{WE}$ latches data from the external bus into the 2187. Data must be valid at this time.

$\overline{OE}$ must not go active during the write cycle.

FALSE MEMORY CYCLE

A false memory cycle (FMC) occurs when $\overline{CE}$ goes active (low), and $\overline{OE}$ and $\overline{WE}$ remain inactive (high). No memory cycle is performed, but address set-up and hold times must be met to guarantee the integrity of internal data. During an FMC, the 2187 performs a refresh cycle on the externally addressed row.

Note that some of the $\overline{CE}$ timing specifications for an FMC differ from those of a read or write cycle.

Other Operating Modes

REFRESH OPERATION

The 2187 supports three refresh modes. Two are controlled externally. The third mode can be enabled when the 2187 is not accessed for extended periods of time (during system stand-by or extended cycle operation). An internal refresh timer guarantees refresh to maintain data integrity.

A refresh cycle is initiated by the leading (falling) edge of $\overline{REFEN}$. Once a refresh cycle begins, cycle operation is internal and automatic. $\overline{REFEN}$ may go inactive (high) after the minimum active low pulse time has been met. Addresses are supplied by the internal refresh address counter. To guarantee internal data integrity, $\overline{REFEN}$ must be strobed at least 128 times in every 2 millisecond period. $\overline{REFEN}$ pulses may be distributed or grouped (burst mode).

$\overline{CE}$ may not go active (low) while a refresh cycle is in progress.

The 2187 may also be refreshed by read, write, or false memory cycles. To accomplish this, the user's system must cycle through each of the 128 rows in every 2 millisecond time frame. In this mode, the refresh address is comprised of the seven lowest order address bits (A_0-A_6), supplied externally.

EXTENDED CYCLE OPERATION

Extended cycle operation is useful for single-step operations and is defined by $\overline{OE}$ or $\overline{WE}$ remaining active (low) for indefinite periods of time. ($\overline{CE}$ is allowed to return high). As long as $\overline{OE}$ is active (read cycles only), data will remain valid on the output bus. During write cycles, data is latched on the leading (falling) edge of $\overline{WE}$. Throughout the remainder of the extended cycle, data may change on the external lines without affecting the contents of the iRAM.

To guarantee refresh of the iRAM array during extended cycles, $\overline{REFEN}$ must go active (low) after $\overline{CE}$ has gone active (low). While $\overline{REFEN}$ is low, an internal timer guarantees that the iRAM array is adequately refreshed, thus ensuring data integrity. Refresh cycles will occur automatically, even if $\overline{OE}$ or $\overline{WE}$ remains low. *While $\overline{REFEN}$ is low, $\overline{CE}$ may not go active (low).* Note that if $\overline{REFEN}$ is not enabled (low), proper refresh of the 2187 array cannot be guaranteed during extended cycles.

Once $\overline{\text{REFEN}}$ returns inactive (high), there are two timing specifications that must be met before the iRAM is accessed again. These are TRFLEL and TRFHELE.

The internal refresh timer may also be enabled (by holding $\overline{\text{REFEN}}$ low) when the iRAM will not be accessed for extended periods of time. It is up to the user to ensure that when $\overline{\text{REFEN}}$ returns inactive (high), no timing specifications are violated.

Initialization

Once V_{CC} is within specification, the 2187 is initialized by holding $\overline{\text{REFEN}}$ active (low) and all other control inputs inactive (high) for 100 microseconds. This may be done any time after V_{CC} meets specification. Normal operation may begin immediately after initialization.

Interfacing Considerations

The 2187 is ideally suited for use with microcontroller systems that use external memory. Figure 1 is an illustration of the simple circuitry required to interface a 2187 and an 8051 microcontroller. In this particular design, all program memory is located in the 8051's internal ROM. External data memory is mapped into the lower 32K bytes of the 8051 address space.

The 2187 is an edge enabled RAM, therefore a stable $\overline{\text{CE}}$ clock is necessary to guarantee proper iRAM operation. The system shown in Figure 1 uses ALE (Address Latch Enable) to generate $\overline{\text{CE}}$ for the 2187. Because the 8051 generates ALE for all memory cycles, internal and external, ALE must be gated with an external address line. This ensures that the 2187 is only selected during external cycles, and not during the shorter internal cycles. To guarantee data integrity, the 2187 is refreshed whenever the 8051 performs an internal cycle. In this case, the trailing (falling) edge of ALE is used to generate the $\overline{\text{REFEN}}$ signal. Once a refresh cycle begins, it will terminate automatically. This design ensures that the 2187 is properly refreshed and that it will always be ready to respond to access cycles in systems running at speeds up to 12 MHz.

Application note 132 "Designing memory systems with the 8K x 8 iRAM", contains a detailed analysis of the iRAM interface. It also describes several designs that incorporate the iRAM into microprocessor and microcontroller systems.

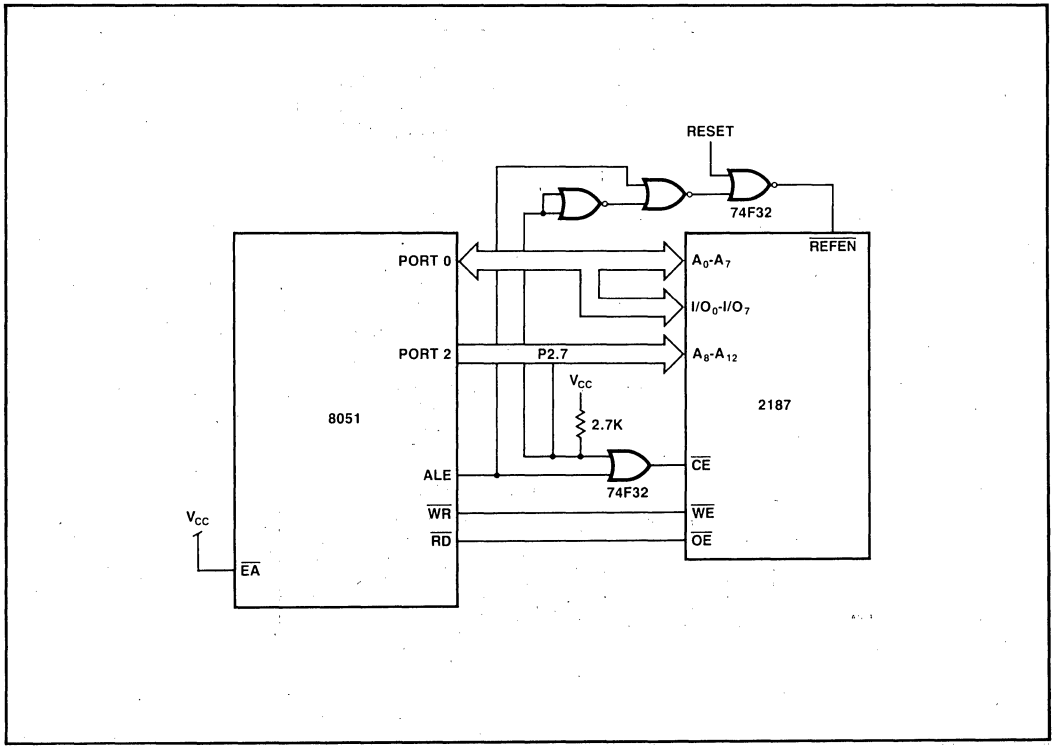


Figure 1. 8051/2187 System Interface

Layout Considerations

To ensure compatibility with other 28-pin memory devices such as EPROMs, several pins require close examination: specifically pins number 1, 26, and 27. Following is a discussion of the system level operation and the design considerations for these pins.

PIN #1

Pin 1 on all EPROMs is reserved for the high voltage programming bias V_{PP}. EPROMS are usually programmed external to the system. Therefore, in normal system operation, pin 1 is connected to V_{CC}.

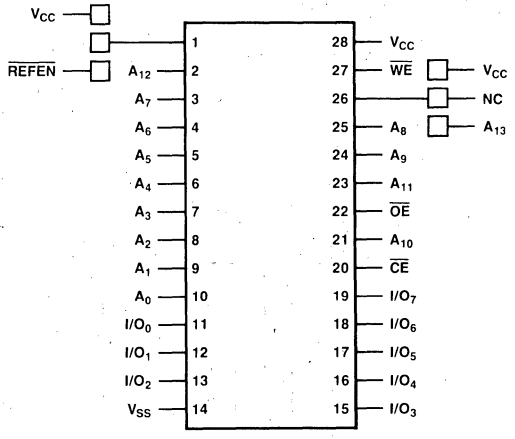
Pin 1 on the 2187 is $\overline{\text{REFEN}}$, the refresh control input. $\overline{\text{REFEN}}$ may come from a microcontroller, or a synchronous refresh timing circuit. In a system incorporating a 28 pin universal site, a trace should be run from pin 1 to the source of the refresh control signal generator, with an alternate trace running to V_{CC}. A jumper option would select $\overline{\text{REFEN}}$ when a 2187 is in the socket. V_{CC} would be selected when the site is occupied by an EPROM. (See Figure 2).

PIN #26

While pin 26 is a No Connect for both the 2186 and the 2764 EPROM, a trace to pin 26 from V_{CC} will guarantee compatibility between 24 pin and 28 pin EPROMs. Pin 26 will carry the additional address bit required to future higher density memories. For flexibility, provide a jumper for an address bit or V_{CC} on pin 26.

PIN #27

Pin 27 is labelled $\overline{\text{WE}}$ on the RAM and $\overline{\text{PGM}}$ on the EPROM. While $\overline{\text{WE}}$ is a system level control signal, $\overline{\text{PGM}}$ is only used when programming the EPROM (V_{PP} at +21V). $\overline{\text{PGM}}$ may be allowed to toggle during normal EPROM operation (V_{PP} at +5V). Therefore, $\overline{\text{WE}}$ may be bussed to every socket location with no jeopardy of illegal operation.



Jumper Matrix Connections

Pin	iRAM	EPROM		
	2187	2732	2764	27128
1	$\overline{\text{REFEN}}$	—	V_{CC}	V_{CC}
26	N.C.	V_{CC}	N.C.	A_{13}

Figure 2. Universal Site Jumper Matrix — iRAM/EPROM Interchange

PRELIMINARY

51C64 FAMILY
65,536x1 BIT CHMOS DYNAMIC RAM

	51C64-10	51C64L-10	51C64-12	51C64L-12
Maximum Access Time (ns)	100	100	120	120
Minimum Cycle Time (ns)	160	160	190	190
Maximum Column Address Access Time (ns)	50	50	60	60

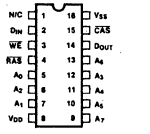
- **CHMOS-D III* Technology**
- **Low power dissipation for the 51C64**
  - **— operating current 37 mA (max.)**
  - **— standby current, TTL 3 mA (max.)**
- **Extended refresh and CHMOS standby current for the 51C64L**
  - **— refresh period, standby mode 64 mS (max.)**
  - **— standby current, CHMOS 100 μA (max.)**
- **Average soft error rate less than 10 FITs (0.001%/1000 hours)**
- **Ripplemode™ operation for a sustained data rate up to 15.3 MHz**
- **Low input/output capacitance**
- **High reliability plastic 16-pin DIP**

The Intel® 51C64 is a high speed 65,536 bit dynamic Random Access Memory. Fabricated on Intel's CHMOS-D III technology, the 51C64 offers features not provided by an NMOS technology: Ripplemode,™ fast usable speed, low power, and an average soft error rate of less than 10 Failures In Time (FITs). The 51C64 is ideally suited for applications such as graphic display terminals, battery backed-up and operated systems, and high-performance systems.

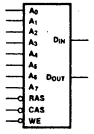
Ripplemode operation allows access of up to 256 bits at a 50 ns/bit rate with random or sequential addresses within a single row. Thus, a continuous data rate up to 15.3 million bits per second can be achieved. The 51C64 offers high performance while relaxing many critical system timing requirements for fast usable speed. In addition, the fast $\overline{RAS}$ and $\overline{CAS}$ access times are compatible with high performance microprocessors such as the 8 MHz iAPX 286 with no WAIT state operation.

The 51C64L offers a standby current of 100 μa when $\overline{RAS} \geq V_{DD} - 0.5V$. During a $\overline{RAS}$-only refresh cycle, the 51C64L extends the refresh period to 64 ms to reduce power consumption to typically 235 μW for data retention. The 51C64 comes in a 16-pin plastic dual in-line package. All inputs, outputs and control signals are TTL compatible. The input and output capacitance are significantly lowered to reduce system drive requirements.

PIN CONFIGURATION



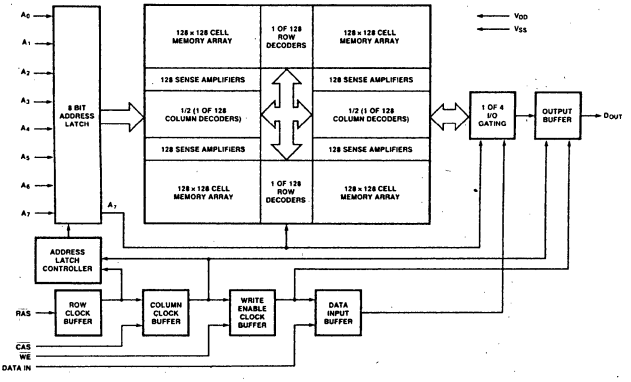
LOGIC SYMBOL



PIN NAMES

A_0-A_7	ADDRESS INPUTS
$\overline{CAS}$	COLUMN ADDRESS STROBE
D_{IN}	DATA IN
D_{OUT}	DATA OUT
$\overline{WE}$	WRITE ENABLE
$\overline{RAS}$	ROW ADDRESS STROBE
V_{DD}	POWER (+5V)
V_{SS}	GROUND

BLOCK DIAGRAM



* CHMOS-D III is a patented process of Intel Corporation.

July, 1983
Order Number: 230731-001

51C65 FAMILY
STATIC COLUMN CHMOS DYNAMIC RAM

	51C65-10	51C65L-10	51C65-12	51C65L-12
Maximum Access Time (ns)	100	100	120	120
Minimum Cycle Time (ns)	160	160	190	190
Maximum Column Address Access Time (ns)	50	50	60	60

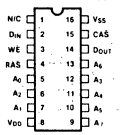
- **65,536 x 1 Organization**
- **Static Column Mode operation**
  - — data rate over 20 MHz
  - — allows easy implementation of tesselation
- **Low input/output capacitance**
- **Average soft error rate less than 10 FITs (0.001%/1000 hours)**
- **Low power dissipation for the 51C65**
  - — operating current 37 mA (max.)
  - — standby current, TTL 3 mA (max.)
- **Extended refresh and CHMOS standby current for the 51C65L**
  - — refresh period, standby mode 64 mS (max.)
  - — standby current, CHMOS 100 μA (max.)
- **High reliability plastic 16-pin DIP**

The Intel® 51C65 is a high speed 65,536 bit dynamic Random Access Memory. Fabricated on Intel's CHMOS-D III* technology, the 51C65 offers features not provided by an NMOS technology: Static Column Mode, fast usable speed, low power, and an average soft error rate of less than 10 Failures In Time (FITs). The 51C65 is ideally suited for applications such as graphic display terminals using tesselation, and high-performance systems.

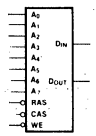
Static Column Mode operation allows access of up to 256 bits at a 50 ns/bit rate with random or sequential addresses within a single row simply by changing the column addresses. A continuous data rate over 20 million bits per second can be achieved. The 51C65 offers high performance while relaxing many critical system timing requirements for fast usable speed.

The 51C65L offers a standby current of 100 μa when RAS $\geq V_{DD}$ – 0.5V. During a RAS-only refresh cycle, the 51C65L extends the refresh period to 64 ms to reduce power consumption to typically 235 μW for data retention. The 51C65 comes in a 16-pin plastic dual in-line package. All inputs, outputs and control signals are TTL compatible. The input and output capacitance are significantly lowered to reduce system drive requirements.

PIN CONFIGURATION



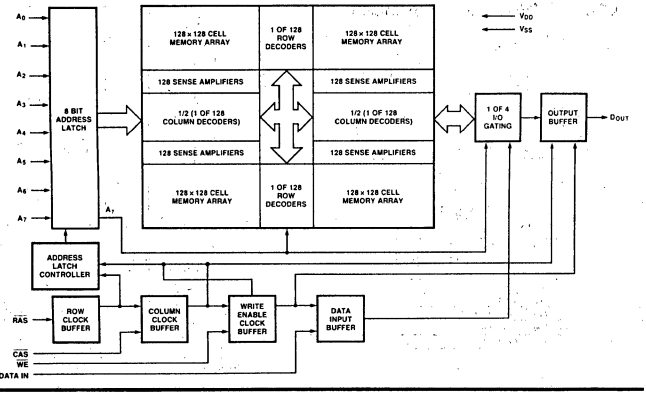
LOGIC SYMBOL



PIN NAMES

A_0-A_7	ADDRESS INPUTS
CAS	COLUMN ADDRESS STROBE
D_{IN}	DATA IN
D_{OUT}	DATA OUT
WE	WRITE ENABLE
RAS	ROW ADDRESS STROBE
V_{DD}	POWER (+5V)
V_{SS}	GROUND

BLOCK DIAGRAM



* CHMOS-D III is a patented process of Intel Corporation.

RAM FAMILY

EXPRESS

- **Standard Temperature Range**
- **Extended Temperature Range −40°C–+85°C Available**
- **168 (±8) Hour Burn-in Available**
- **Inspected to 0.1% AQL**

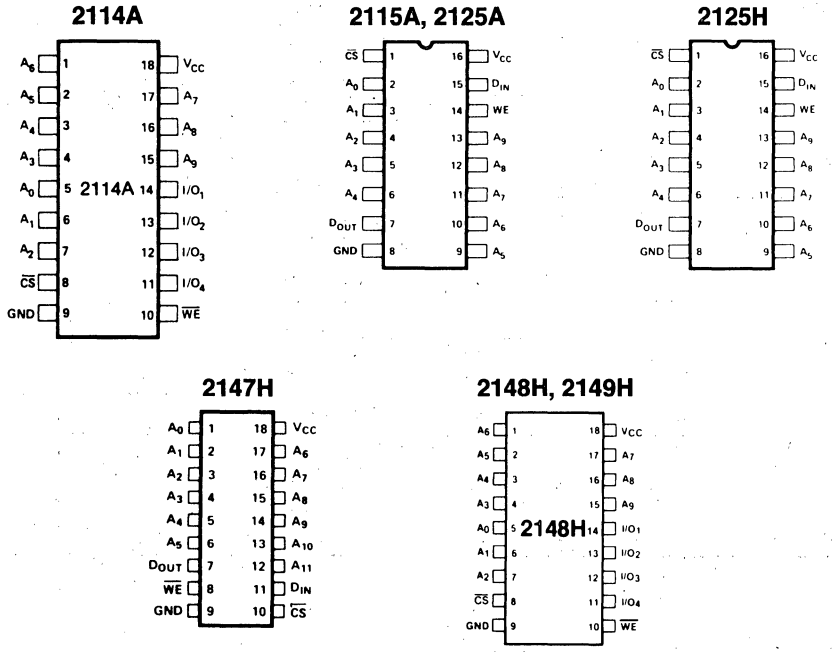
The Intel EXPRESS RAM family is a series of random-access memories which have received additional processing to enhance product operating temperature range and infant mortality. EXPRESS processing is available for several densities of RAM, allowing the choice of appropriate memory size to match system applications.

EXPRESS RAM product is available with 168(±8) hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in.

The standard EXPRESS RAM operating temperature range is 0°C to 70 or 75° C. Extended operating temperature range (−40°C to 85°C) EXPRESS product is available. EXPRESS products plus military grade RAMs (−55°C to 125°C) provide the most complete choice of standard and extended temperature range RAMs available.

Like all Intel RAMs, the EXPRESS RAM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

Detailed individual product electrical specifications are available separately in Intel's respective commercial and industrial grade product data sheets.



Pin Configuration

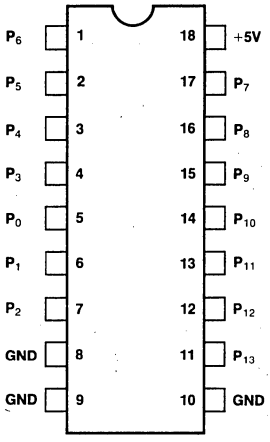
MARCH 1982
ORDER NUMBER: 210347-001

Table 1. RAM Product Family

EXPRESS

Type	Organization	Maximum Access (ns)	Power Supply	Operating Temperature (°C)	Burn-In 125°C (±8 hours)
QD 2114A-4	1K x 4	200	5V ±10%	0 to 70	168
QP 2114A-4	1K x 4	200	5V ±10%	0 to 70	168
QD 2114A-5	1K x 4	250	5V ±10%	0 to 70	168
QP 2114A-5	1K x 4	250	5V ±10%	0 to 70	168
QP 2114A-6	1K x 4	300	5V ±10%	0 to 70	168
QD 2114AL-1	1K x 4	100	5V ±10%	0 to 70	168
QP 2114AL-1	1K x 4	100	5V ±10%	0 to 70	168
QD 2114AL-2	1K x 4	120	5V ±10%	0 to 70	168
QP 2114AL-2	1K x 4	120	5V ±10%	0 to 70	168
QD 2114AL-3	1K x 4	150	5V ±10%	0 to 70	168
QP 2114AL-3	1K x 4	150	5V ±10%	0 to 70	168
QD 2114AL-4	1K x 4	200	5V ±10%	0 to 70	168
QP 2114AL-4	1K x 4	200	5V ±10%	0 to 70	168
LD 2114A-4	1K x 4	200	5V ±10%	−40 to 85	168
LD 2114A-5	1K x 4	250	5V ±10%	−40 to 85	168
LD 2114AL-3	1K x 4	150	5V ±10%	−40 to 85	168
LD 2114AL-4	1K x 4	200	5V ±10%	−40 to 85	168
TD 2114A-4	1K x 4	200	5V ±10%	−40 to 85	NONE
TD 2114A-5	1K x 4	250	5V ±10%	−40 to 85	NONE
TD 2114AL-3	1K x 4	150	5V ±10%	−40 to 85	NONE
TD 2114AL-4	1K x 4	200	5V ±10%	−40 to 85	NONE
QD 2115A	1K x 1	45	5V ±5%	0 to 75	168
QD 2115A-2	1K x 1	70	5V ±5%	0 to 75	168
QD 2115AL	1K x 1	45	5V ±5%	0 to 75	168
QD 2115AL-2	1K x 1	70	5V ±5%	0 to 75	168
QD 2125A	1K x 1	45	5V ±5%	0 to 75	168
QD 2125A-2	1K x 1	70	5V ±5%	0 to 75	168
QD 2125AL	1K x 1	45	5V ±5%	0 to 75	168
QD 2125AL-2	1K x 1	70	5V ±5%	0 to 75	168
QD 2125H-2	1K x 1	25	5V ±5%	0 to 75	168
QD 2125H-3	1K x 1	30	5V ±5%	0 to 75	168
QD 2115H-4	1K x 1	35	5V ±5%	0 to 75	168
QD 2147H	4K x 1	70	5V ±10%	0 to 70	168
QD 2147H-1	4K x 1	35	5V ±10%	0 to 70	168
QD 2147H-2	4K x 1	45	5V ±10%	0 to 70	168
QD 2147H-3	4K x 1	55	5V ±10%	0 to 70	168
QD 2147HL	4K x 1	70	5V ±10%	0 to 70	168
QD 2147HL-3	4K x 1	55	5V ±10%	0 to 70	168
QD 2148H	1K x 4	70	5V ±10%	0 to 70	168
QD 2148H-3	1K x 4	55	5V ±10%	0 to 70	168
QD 2148HL	1K x 4	70	5V ±10%	0 to 70	168
QD 2148HL-3	1K x 4	55	5V ±10%	0 to 70	168
QD2149H	1K x 4	70	5V ±10%	0 to 70	168
QD2149H-2	1K x 4	45	5V ±10%	0 to 70	168
QD2149H-3	1K x 4	55	5V ±10%	0 to 70	168
QD2149HL	1K x 4	70	5V ±10%	0 to 70	168
QD2149HL-3	1K x 4	55	5V ±10%	0 to 70	168

AFN-02163A



SUPPLY	VOLTAGES	TOLER-ANCES	PWR-UP SEQ.	CURRENT DEVICE	NOISE LEVELS
V_{CC}	+5V	±.25V	1	70 mA	±.25V
V_{IL}	0.0V	±.5V			
V_{IH}	5.0V	±1.0V			

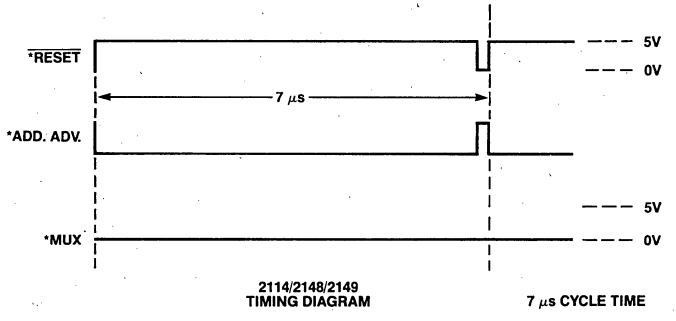
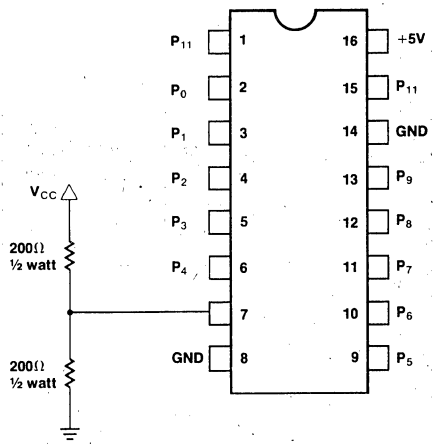
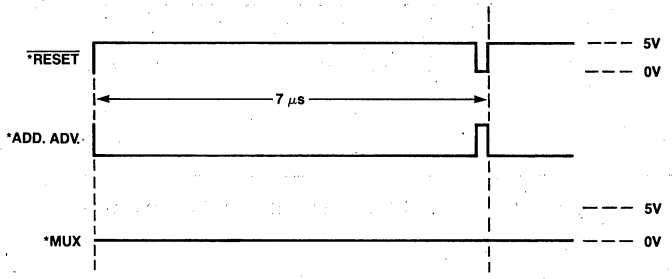


Figure 1. 2114A, 2148H, 2149H Burn-in Configuration



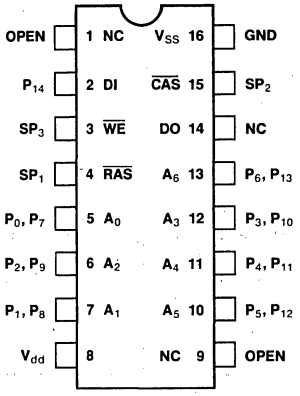
SUPPLY	VOLTAGES	TOLER-ANCES	PWR-UP SEQ.	CURRENT DEVICE	NOISE LEVELS
V_{CC}	+5V	±.25V	1st	75 mA	±.25V
V_{IL}	0.0V	±.5V			
V_{IH}	5.0V	±1.0V			



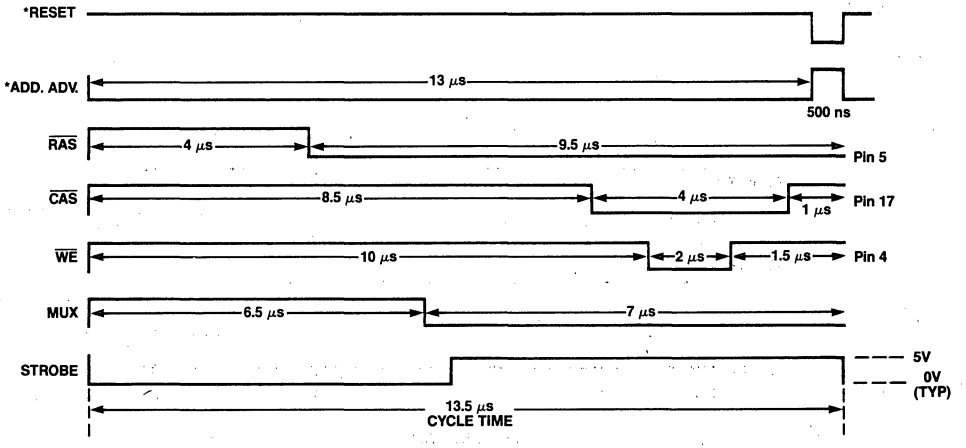
2115/2125 TIMING DIAGRAM
7 μs CYCLE TIME

Figure 2. 2115, 2125 Burn-in Configuration

AFN-02163A

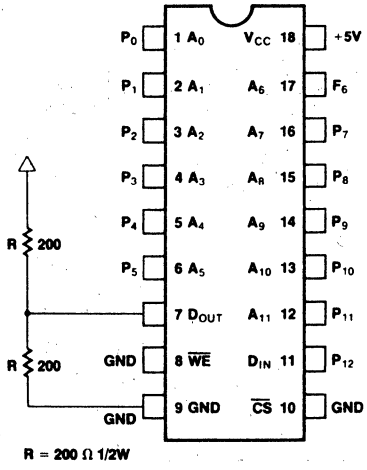


SUPPLY	VOLTAGES	TOLER-ANCES	PWR-UP SEQ.	CURRENT DEVICE	NOISE LEVELS
Pulses	+5V	4-6V	1st		
V_{dd}	+5V	±.25V	2nd	20 mA	±.25V

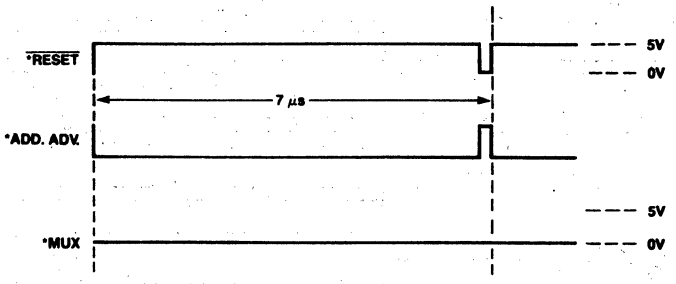


2118 TIMING DIAGRAM

Figure 3. 2118 Burn-in Configuration



SUPPLY	VOLTAGES	TOLER-ANCES	PWR-UP SEQ.	CURRENT DEVICE	NOISE LEVELS
V_{CC}	+5V	±.25V	1	125 mA	±.25V
V_{IL}	0.0V	±.5V			
V_{IH}	5.0V	±1.0V			



2147H
TIMING DIAGRAM
7 µs CYCLE TIME

Figure 4. 2147H Burn-In Configuration

8203
64K DYNAMIC RAM CONTROLLER

- **Provides All Signals Necessary to Control 64K (2164) and 16K (2117, 2118) Dynamic Memories**
- **Directly Addresses and Drives Up to 64 Devices Without External Drivers**
- **Provides Address Multiplexing and Strobes**
- **Provides a Refresh Timer and a Refresh Counter**
- **Provides Refresh/Access Arbitration**
- **Internal Clock Capability with the 8203-1 and the 8203-3**
- **Fully Compatible with Intel® 8080A, 8085A, iAPX 88, and iAPX 86 Family Microprocessors**
- **Decodes CPU Status for Advanced Read Capability in 16K mode with the 8203-1 and the 8203-3.**
- **Provides System Acknowledge and Transfer Acknowledge Signals**
- **Refresh Cycles May be Internally or Externally Requested (For Transparent Refresh)**
- **Internal Series Damping Resistors on All RAM Outputs**

The Intel® 8203 is a Dynamic Ram System Controller designed to provide all signals necessary to use 2164, 2118 or 2117 Dynamic RAMs in microcomputer systems. The 8203 provides multiplexed addresses and address strobes, refresh logic, refresh/access arbitration. Refresh cycles can be started internally or externally. The 8203-1 and the 8203-3 support an internal crystal oscillator and Advanced Read Capability. The 8203-3 is a $\pm 5\%$ V_{CC} part.

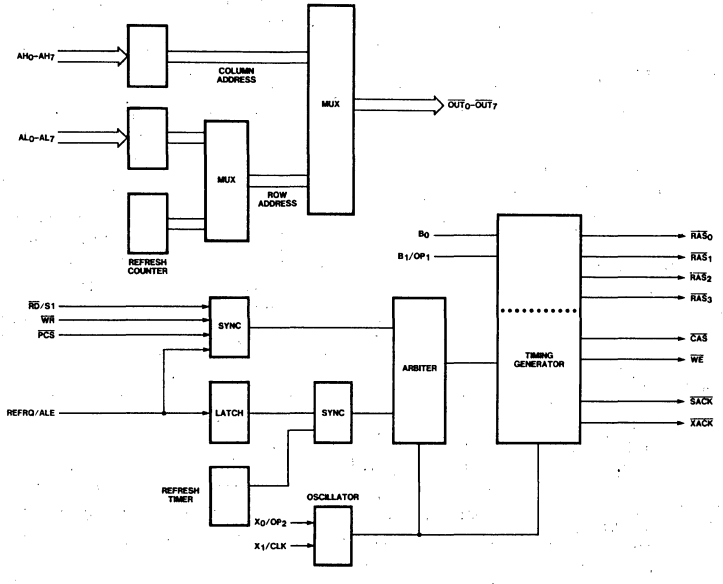


Figure 1. 8203 Block Diagram

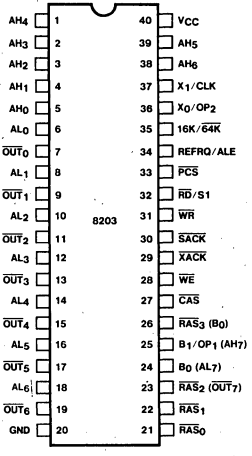


Figure 2. Pin Configuration

Table 1. Pin Descriptions

Symbol	Pin No.	Type	Name and Function
AL_0 AL_1 AL_2 AL_3 AL_4 AL_5 AL_6	6 8 10 12 14 16 18	I I I I I I I	**Address Low:** CPU address inputs used to generate memory row address.
AH_0 AH_1 AH_2 AH_3 AH_4 AH_5 AH_6	5 4 3 2 1 39 38	I I I I I I I	**Address High:** CPU address inputs used to generate memory column address.
B_0/AL_7 $B_1/OP_1/AH_7$	24 25	I I	**Bank Select Inputs:** Used to gate the appropriate RAS output for a memory cycle. B_1/OP_1 option used to select the Advanced Read Mode. (Not available in 64K mode.) See Figure 5. When in 64K RAM Mode, pins 24 and 25 operate as the AL_7 and AH_7 address inputs.
$\overline{PCS}$	33	I	**Protected Chip Select:** Used to enable the memory read and write inputs. Once a cycle is started, it will not abort even if $\overline{PCS}$ goes inactive before cycle completion.
$\overline{WR}$	31	I	**Memory Write Request.**
$\overline{RD}/S1$	32	I	**Memory Read Request:** S1 function used in Advanced Read mode selected by OP_1 (pin 25).
REFRQ/ ALE	34	I	**External Refresh Request:** ALE function used in Advanced Read mode, selected by OP_1 (pin 25).
$\overline{OUT}_0$ $\overline{OUT}_1$ $\overline{OUT}_2$ $\overline{OUT}_3$ $\overline{OUT}_4$ $\overline{OUT}_5$ $\overline{OUT}_6$	7 9 11 13 15 17 19	O O O O O O O	**Output of the Multiplexer:** These outputs are designed to drive the addresses of the Dynamic RAM array. (Note that the $\overline{OUT}_{0\text{-}7}$ pins do not require inverters or drivers for proper operation.)
$\overline{WE}$	28	O	**Write Enable:** Drives the Write Enable inputs of the Dynamic RAM array.
$\overline{CAS}$	27	O	**Column Address Strobe:** This output is used to latch the Column Address into the Dynamic RAM array.
$\overline{RAS}_0$ $\overline{RAS}_1$ $\overline{RAS}_2/\overline{OUT}_7$ $\overline{RAS}_3/B_0$	21 22 23 26	O O O I/O	**Row Address Strobe:** Used to latch the Row Address into the bank of dynamic RAMs, selected by the 8203 Bank Select pins (B_0, B_1/OP_1). In 64K mode, only $\overline{RAS}_0$ and $\overline{RAS}_1$ are available; pin 23 operates as $\overline{OUT}_7$ and pin 26 operates as the B_0 bank select input.
$\overline{XACK}$	29	O	**Transfer Acknowledge:** This output is a strobe indicating valid data during a read cycle or data written during a write cycle. $\overline{XACK}$ can be used to latch valid data from the RAM array.
$\overline{SACK}$	30	O	**System Acknowledge:** This output indicates the beginning of a memory access cycle. It can be used as an advanced transfer acknowledge to eliminate wait states. (Note: If a memory access request is made during a refresh cycle, $\overline{SACK}$ is delayed until $\overline{XACK}$ in the memory access cycle).
X_0/OP_2 X_1/CLK	36 37	I/O I/O	**Oscillator Inputs:** These inputs are designed for a quartz crystal to control the frequency of the oscillator. If X_0/OP_2 is shorted to pin 40 (V_{CC}) or if X_0/OP_2 is connected to +12V through a 1KΩ resistor then X_1/CLK becomes a TTL input for an external clock. (Note: Crystal mode for the 8203-1 and the 8203-3 only).
16K/$\overline{64K}$	35	I	**Mode Select:** This input selects 16K mode (2117, 2118) or 64K mode (2164). Pins 23-26 change function based on the mode of operation.
V_{CC}	40		**Power Supply:** +5V.
GND	20		**Ground.**

Functional Description

The 8203 provides a complete dynamic RAM controller for microprocessor systems as well as expansion memory boards. All of the necessary control signals are provided for 2164, 2118 and 2117 dynamic RAMs.

The 8203 has two modes, one for 16K dynamic RAMs and one for 64Ks, controlled by pin 35.

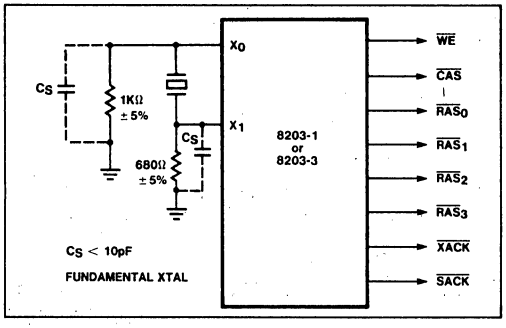


Figure 3. Crystal Operation for the 8203-1 and 8203-3

All 8203 timing is generated from a single reference clock. This clock is provided via an external oscillator or an on-chip crystal oscillator. All output signal transitions are synchronous with respect to this clock reference, except for the trailing edges of the CPU handshake signals $\overline{SACK}$ and $\overline{XACK}$.

CPU memory requests normally use the $\overline{RD}$ and $\overline{WR}$ inputs. The Advanced-Read mode allows ALE and S1 to be used in place of the $\overline{RD}$ input.

Failsafe refresh is provided via an internal timer which generates refresh requests. Refresh requests can also be generated via the REFRQ input.

An on-chip synchronizer/arbiter prevents memory and refresh requests from affecting a cycle in progress. The READ, WRITE, and external REFRESH requests may be asynchronous to the 8203 clock; on-chip logic will synchronize the requests, and the arbiter will decide if the requests should be delayed, pending completion of a cycle in progress.

16K/$\overline{64K}$ Option Selection

Pin 35 is a strap input that controls the two 8203 modes. Figure 4 shows the four pins that are multiplexed. In 16K mode (pin 35 tied to V_{CC} or left open), the 8203 has two Bank Select inputs to select one of four $\overline{RAS}$ outputs. In this mode, the 8203 is exactly compatible with the Intel 8202A Dynamic RAM Controller. In 64K mode (pin 35 tied to GND), there is only one Bank Select input (pin 26) to select the two $\overline{RAS}$ outputs. More than two banks of 64K dynamic RAM's can be used with external logic.

Other Option Selections

The 8203 has three strapping options. When OP_1 is selected (16K mode only), pin 32 changes from a $\overline{RD}$ input to an S1 input, and pin 34 changes from a REFRQ input to an ALE input. See "Refresh Cycles" and "Read Cycles" for more detail. OP_1 is selected by tying pin 25 to +12V through a 5.1K ohm resistor on the 8203-1 or 8203-3 only.

When OP_2 is selected, the internal oscillator is disabled and pin 37 changes from a crystal input (X_1) to a CLK input for an external TTL clock. OP_2 is selected by shorting pin 36 (X_0/OP_2) directly to pin 40 (V_{CC}). No current limiting resistor should be used. OP_2 may also be selected by tying pin 36 to +12V through a 1KΩ resistor.

Refresh Timer

The refresh timer is used to monitor the time since the last refresh cycle occurred. When the appropriate amount of time has elapsed, the refresh timer will request a refresh cycle. External refresh requests will reset the refresh timer.

Refresh Counter

The refresh counter is used to sequentially refresh all of the memory's rows. The 8-bit counter is incremented after every refresh cycle.

Pin #	16K Function	64K Function
23	$\overline{RAS_2}$	Address Output ($\overline{OUT_7}$)
24	Bank Select (B_0)	Address Input (AL_7)
25	Bank Select (B_1)	Address Input (AH_7)
26	$\overline{RAS_3}$	Bank Select (B_0)

Figure 4. 16K/$\overline{64K}$ Mode Selection

	Inputs		Outputs			
	B_1	B_0	$\overline{RAS_0}$	$\overline{RAS_1}$	$\overline{RAS_2}$	$\overline{RAS_3}$
16K Mode	0	0	0	1	1	1
	0	1	1	0	1	1
	1	0	1	1	0	1
	1	1	1	1	1	0
64K Mode	—	0	0	1	—	—
	—	1	1	0	—	—

Figure 5. Bank Selection

Description	Pin #	Normal Function	Option Function
B1/OP1 (16K only)/AH_7	25	Bank (RAS) Select	Advanced-Read Mode (8203-1, -3)
X_0/OP_2	36	Crystal Oscillator (8203-1 and 8203-3)	External Oscillator

Figure 6. 8203 Option Selection

Address Multiplexer

The address multiplexer takes the address inputs and the refresh counter outputs, and gates them onto the address outputs at the appropriate time. The address outputs, in conjunction with the $\overline{RAS}$ and $\overline{CAS}$ outputs, determine the address used by the dynamic RAMs for read, write, and refresh cycles. During the first part of a read or write cycle, AL_0–AL_7 are gated to $\overline{OUT}_0$–$\overline{OUT}_7$, then AH_0–AH_7 are gated to the address outputs.

During a refresh cycle, the refresh counter is gated onto the address outputs. All refresh cycles are RAS-only refresh ($\overline{CAS}$ inactive, $\overline{RAS}$ active).

To minimize buffer delay, the information on the address outputs is inverted from that on the address inputs.

$\overline{OUT}_0$–$\overline{OUT}_7$ do not need inverters or buffers unless additional drive is required.

Synchronizer/Arbiter

The 8203 has three inputs, REFRQ/ALE (pin 34), $\overline{RD}$ (pin 32) and $\overline{WR}$ (pin 31). The $\overline{RD}$ and $\overline{WR}$ inputs allow an external CPU to request a memory read or write cycle, respectively. The REFRQ/ALE input allows refresh requests to be requested external to the 8203.

All three of these inputs may be asynchronous with respect to the 8203's clock. The arbiter will resolve conflicts between refresh and memory requests, for both pending cycles and cycles in progress. Read and write requests will be given priority over refresh requests.

System Operation

The 8203 is always in one of the following states:

a) IDLE
b) TEST Cycle
c) REFRESH Cycle
d) READ Cycle
e) WRITE Cycle

The 8203 is normally in the IDLE state. Whenever one of the other cycles is requested, the 8203 will leave the IDLE state to perform the desired cycle. If no other cycles are pending, the 8203 will return to the IDLE state.

Test Cycle

The TEST Cycle is used to check operation of several 8203 internal functions. TEST cycles are requested by activating the $\overline{PCS}$, $\overline{RD}$ and $\overline{WR}$ inputs. The TEST Cycle will reset the refresh address counter and perform a WRITE Cycle. The TEST Cycle should not be used in normal system operation, since it would affect the dynamic RAM refresh.

Refresh Cycles

The 8203 has two ways of providing dynamic RAM refresh:

1) Internal (failsafe) refresh
2) External (hidden) refresh

Both types of 8203 refresh cycles activate all of the $\overline{RAS}$ outputs, while $\overline{CAS}$, $\overline{WE}$, $\overline{SACK}$, and $\overline{XACK}$ remain inactive.

Internal refresh is generated by the on-chip refresh timer. The timer uses the 8203 clock to ensure that refresh of all rows of the dynamic RAM occurs every 2 milliseconds (128 cycles) or every 4 milliseconds (256 cycles). If REFRQ is inactive, the refresh timer will request a refresh cycle every 10-16 microseconds.

External refresh is requested via the REFRQ input (pin 34). External refresh control is not available when the Advanced-Read mode is selected. External refresh requests are latched, then synchronized to the 8203 clock.

The arbiter will allow the refresh request to start a refresh cycle only if the 8203 is not in the middle of a cycle.

When the 8203 is in the idle state a simultaneous memory request and external refresh request will result in the memory request being honored first. This 8203 characteristic can be used to "hide" refresh cycles during system operation. A circuit similar to Figure 7 can be used to decode the CPU's instruction fetch status to generate an external refresh request. The refresh request is latched while the 8203 performs the instruction fetch; the refresh cycle will start immediately after the memory cycle is completed, even if the $\overline{RD}$ input has not gone inactive. If the CPU's instruction decode time is long enough, the 8203 can complete the refresh cycle before the next memory request is generated.

If the 8203 is not in the idle state then a simultaneous memory request and an external refresh request may result in the refresh request being honored first.

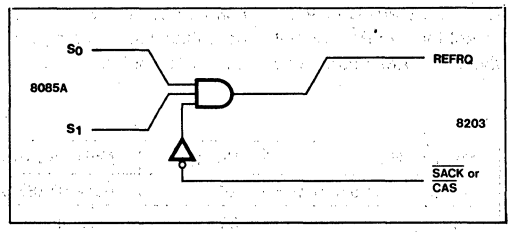


Figure 7. Hidden Refresh

Certain system configurations require complete external refresh requests. If external refresh is requested faster than the minimum internal refresh timer (t_{REF}), then, in effect, all refresh cycles will be caused by the external refresh request, and the internal refresh timer will never generate a refresh request.

Read Cycles

The 8203 can accept two different types of memory Read requests:

1) Normal Read, via the $\overline{RD}$ input
2) Advanced Read, using the S1 and ALE inputs (16K mode only)

The user can select the desired Read request configuration via the B1/OP1 hardware strapping option on pin 25.

	Normal Read	Advanced Read
Pin 25	B1 input	OP_1 (+12V)
Pin 32	RD input	S1 input
Pin 34	REFRQ input	ALE input
# RAM banks	4 ($\overline{RAS}_{0-3}$)	2 ($\overline{RAS}_{2-3}$)
Ext. Refresh	Yes	No

Figure 8. 8203 Read Options

Normal Reads are requested by activating the $\overline{RD}$ input, and keeping it active until the 8203 responds with an $\overline{XACK}$ pulse. The $\overline{RD}$ input can go inactive as soon as the command hold time (t_{CHS}) is met.

Advanced Read cycles are requested by pulsing ALE while S1 is active; if S1 is inactive (low) ALE is ignored. Advanced Read timing is similiar to Normal Read timing, except the falling edge of ALE is used as the cycle start reference.

If a Read cycle is requested while a refresh cycle is in progress, then the 8203 will set the internal delayed-SACK latch. When the Read cycle is eventually started, the 8203 will delay the active $\overline{SACK}$ transition until $\overline{XACK}$ goes active, as shown in the AC timing diagrams. This delay was designed to compensate for the CPU's READY setup and hold times. The delayed–SACK latch is cleared after every READ cycle.

Based on system requirements, either $\overline{SACK}$ or $\overline{XACK}$ can be used to generate the CPU READY signal. $\overline{XACK}$ will normally be used; if the CPU can tolerate an advanced READY, then $\overline{SACK}$ can be used, but only if the CPU can tolerate the amount of advance provided by $\overline{SACK}$. If $\overline{SACK}$ arrives too early to provide the appropriate number of WAIT states, then either $\overline{XACK}$ or a delayed form of $\overline{SACK}$ should be used.

Write Cycles

Write cycles are similiar to Normal Read cycles, except for the $\overline{WE}$ output. $\overline{WE}$ is held inactive for Read cycles, but goes active for Write cycles. All 8203 Write cycles are "early-write" cycles; $\overline{WE}$ goes active before $\overline{CAS}$ goes active by an amount of time sufficient to keep the dynamic RAM output buffers turned off.

General System Considerations

All memory requests (Normal Reads, Advanced Reads, Writes) are qualified by the $\overline{PCS}$ input. $\overline{PCS}$ should be stable, either active or inactive, prior to the leading edge of $\overline{RD}$, $\overline{WR}$, or ALE. Systems which use battery backup should pullup $\overline{PCS}$ to prevent erroneous memory requests.

In order to minimize propagation delay, the 8203 uses an inverting address multiplexer without latches. The system must provide adequate address setup and hold times to guarantee $\overline{RAS}$ and $\overline{CAS}$ setup and hold times for the RAM. The t_{AD} AC parameter should be used for this system calculation.

The B_0-B_1 inputs are similiar to the address inputs in that they are not latched. B_0 and B_1 should not be changed during a memory cycle, since they directly control which $\overline{RAS}$ output is activated.

The 8203 uses a two-stage synchronizer for the memory request inputs ($\overline{RD}$, $\overline{WR}$, ALE), and a separate two stage synchronizer for the external refresh input (REFRQ). As with any synchronizer, there is always a finite probability of metastable states inducing system errors. The 8203 synchronizer was designed to have a system error rate less than 1 memory cycle every three years based on the full operating range of the 8203.

A microprocessor system is concerned when the data is valid after $\overline{RD}$ goes low. See Figure 9. In order to calculate memory read access times, the dynamic RAM's A.C. specifications must be examined, especially the RAS-access time (t_{RAC}) and the CAS-access time (t_{CAC}). Most configurations will be CAS-access limited; i.e., the data from the RAM will be stable $t_{CC,max}$ (8203) + t_{CAC} (RAM) after a memory read cycle is started. Be sure to add any delays (due to buffers, data latches, etc.) to calculate the overall read access time.

Since the 8203 normally performs "early-write" cycles, the data must be stable at the RAM data inputs by the time $\overline{CAS}$ goes active, including the RAM's data setup time. If the system does not normally guarantee sufficient write data setup, you must either delay the $\overline{WR}$ input signal or delay the 8203 $\overline{WE}$ output.

Delaying the $\overline{WR}$ input will delay all 8203 timing, including the READY handshake signals, $\overline{SACK}$ and $\overline{XACK}$, which

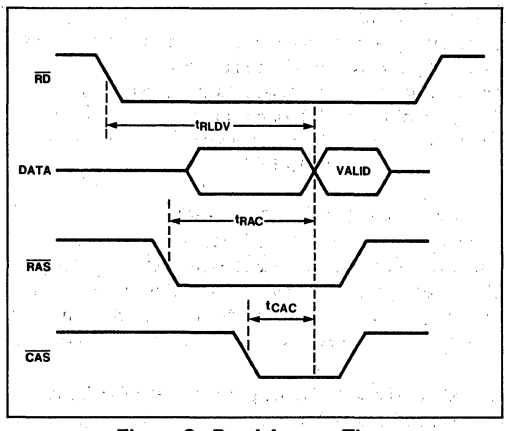


Figure 9. Read Access Time

may increase the number of WAIT states generated by the CPU.

If the $\overline{WE}$ output is externally delayed beyond the $\overline{CAS}$ active transition, then the RAM will use the falling edge of $\overline{WE}$ to strobe the write data into the RAM. This $\overline{WE}$ transition should not occur too late during the CAS active transition, or else the $\overline{WE}$ to $\overline{CAS}$ requirements of the RAM will not be met.

The $\overline{RAS}_{0-3}$, $\overline{CAS}$, $\overline{OUT}_{0-7}$, and $\overline{WE}$ outputs contain on-chip series damping resistors (typically 20Ω) to minimize overshoot.

Some dynamic RAMs require more than 2.4V V_{IH}. Noise immunity may be improved for these RAMs by adding pull-up resistors to the 8203's outputs. Intel RAMs do not require pull-up resistors.

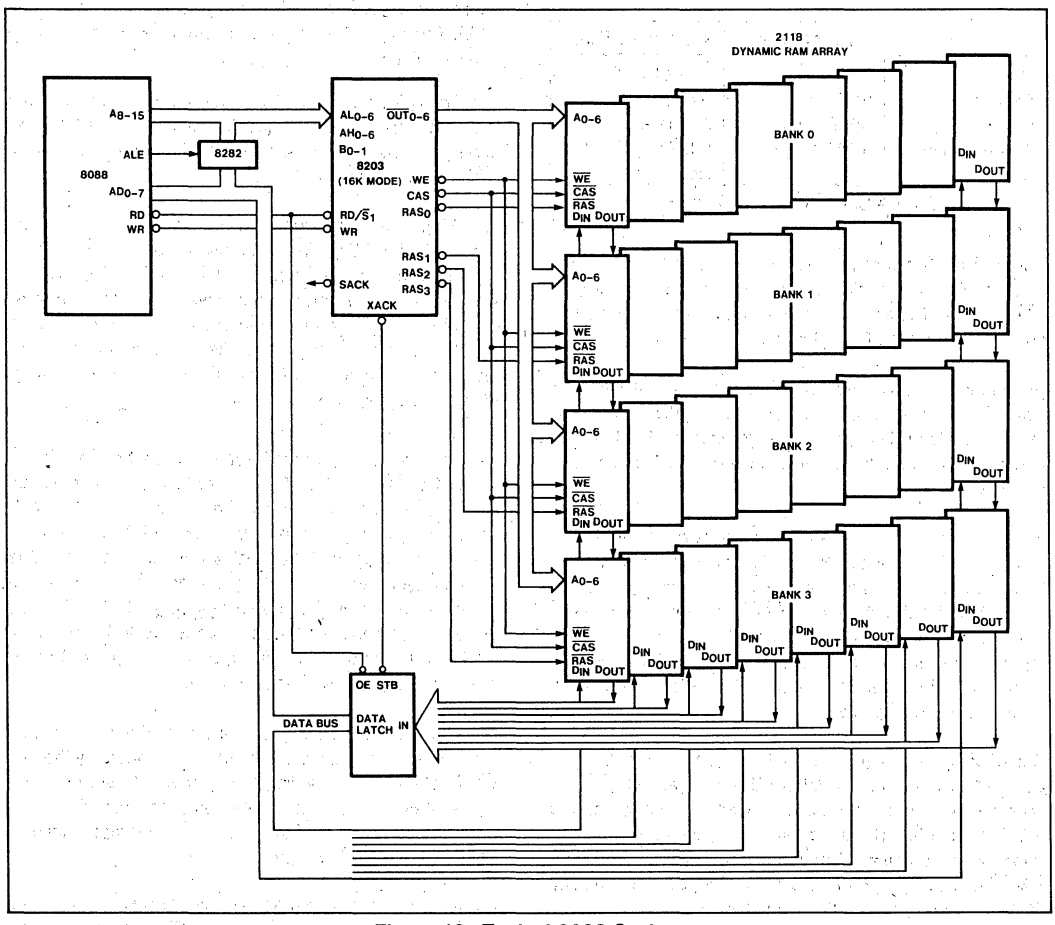


Figure 10. Typical 8088 System

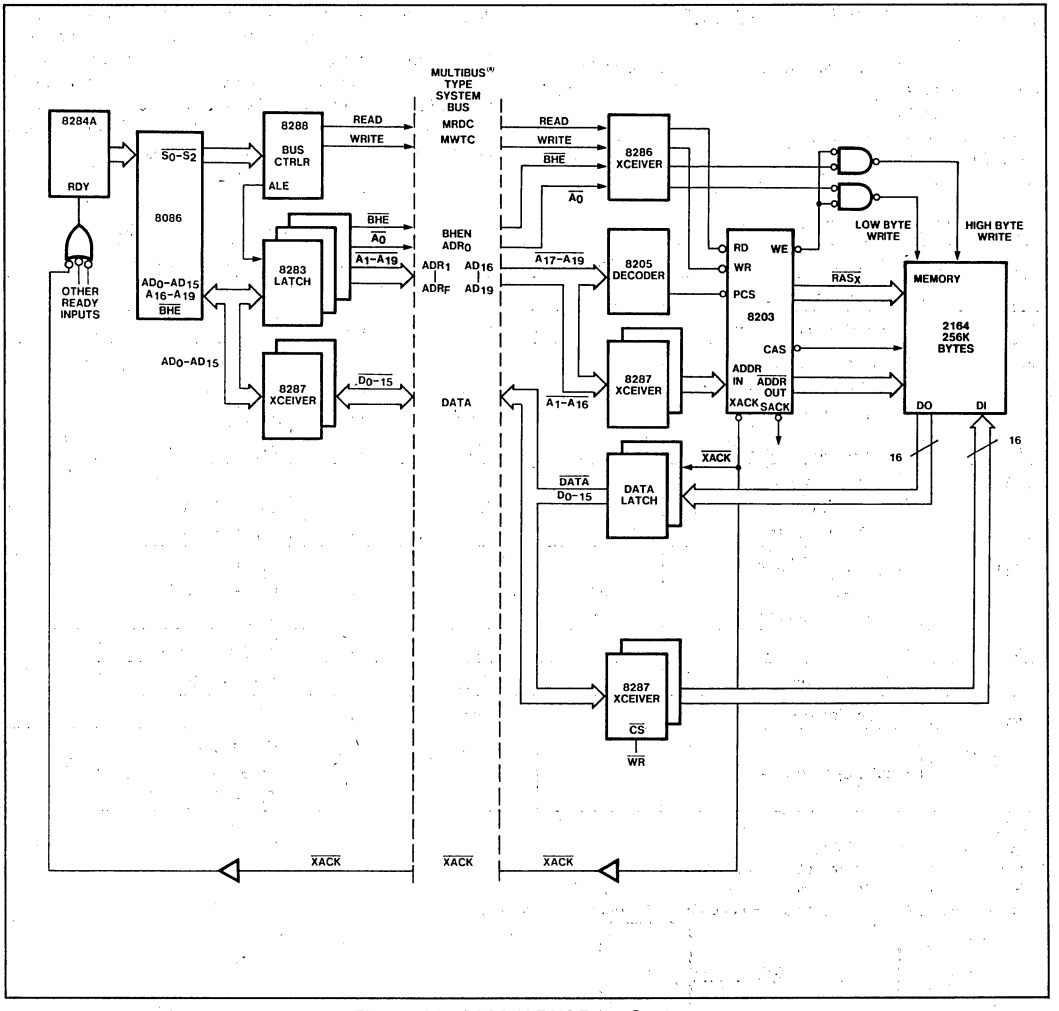


Figure 11. 8086/256K Byte System

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage On any Pin
With Respect to Ground .....−0.5V to +7V[4]
Power Dissipation .....1.6 Watts

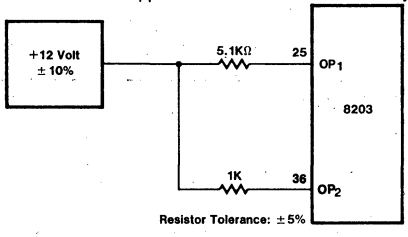
NOTE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = 5.0V ± 10% (5.0V ± 5% for 8203-3); GND = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions
V_C	Input Clamp Voltage		−1.0	V	I_C = −5 mA
I_{CC}	Power Supply Current		290	mA	
I_F	Forward Input Current				
	CLK, 64K/16K Mode select		−2.0	mA	V_F = 0.45V
	All Other Inputs[3]		−320	μA	V_F = 0.45V
I_R	Reverse Input Current[3]		40	μA	V_R = V_{CC}; Note 1
V_{OL}	Output Low Voltage				
	$\overline{SACK}$, $\overline{XACK}$		0.45	V	I_{OL} = 5 mA
	All Other Outputs		0.45	V	I_{OL} = 3 mA
V_{OH}	Output High Voltage				V_{IL} = 0.65 V
	$\overline{SACK}$, $\overline{XACK}$	2.4		V	I_{OH} = −1 mA
	All Other Outputs	2.6		V	I_{OH} = −1 mA
V_{IL}	Input Low Voltage		0.8	V	V_{CC} = 5.0V (Note 2)
V_{IH1}	Input High Voltage	2.0	V_{CC}	V	V_{CC} = 5.0V
V_{IH2}	Option Voltage		V_{CC}	V	(Note 4)
C_{IN}	Input Capacitance		30	pF	F = 1 MHz V_{BIAS} = 2.5V, V_{CC} = 5V T_A = 25°C

NOTES:
1. I_R = 200 μA for pin 37 (CLK).
2. For test mode $\overline{RD}$ & $\overline{WR}$ must be held at GND.
3. Except for pin 36 in XTAL mode.
4. 8203-1 and 8203-3 supports both OP1 and OP2, 8203 only supports OP2.



A.C. CHARACTERISTICS

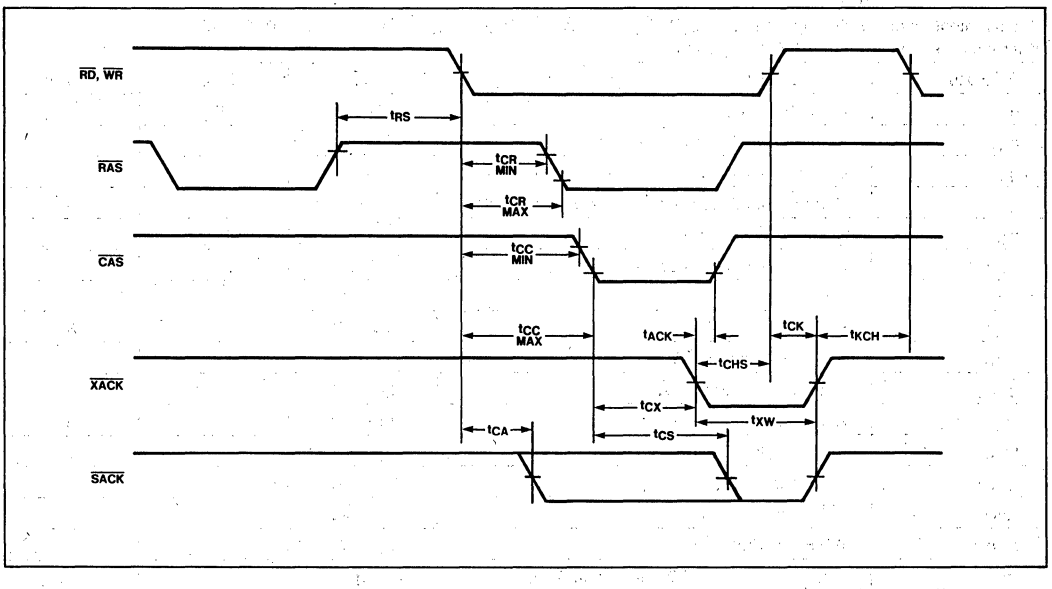
T_J = 0°C to 70°C; V_{CC} = 5V ± 10% (5.0V ± 5% for 8203-3); GND = 0V

Measurements made with respect to $\overline{RAS}_0$-$\overline{RAS}_3$, $\overline{CAS}$, $\overline{WE}$, $\overline{OUT}_0$-$\overline{OUT}_6$ are at 2.4V and 0.8V. All other pins are measured at 1.5V. All times are in nsec.

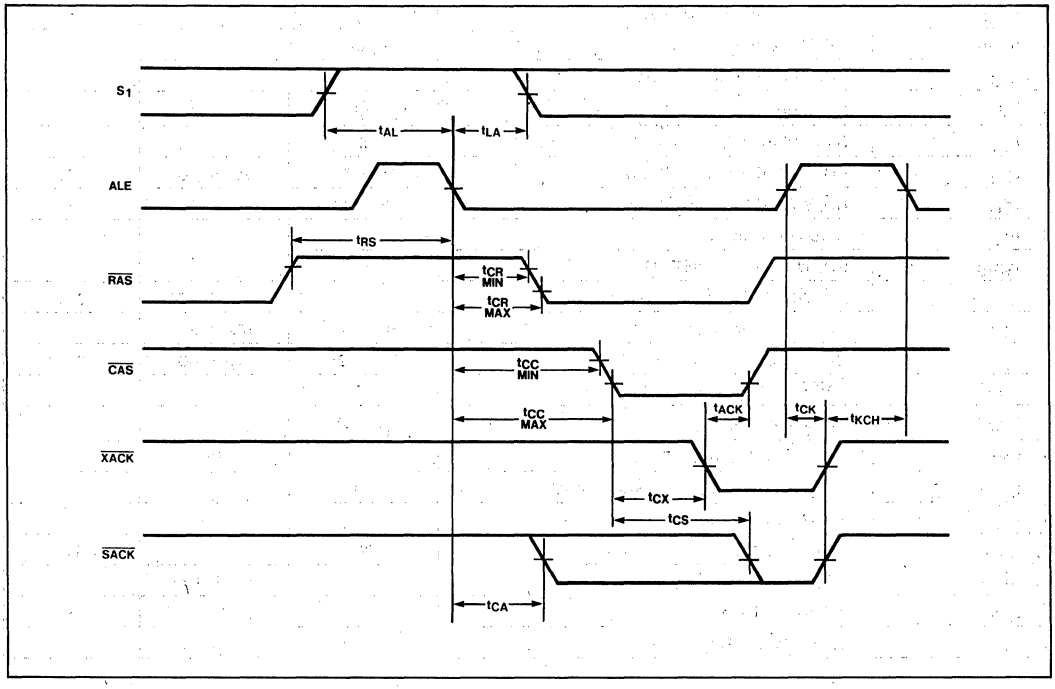
Symbol	Parameter	Min	Max	Notes
t_P	Clock Period	40	54	
t_{PH}	External Clock High Time	20		
t_{PL}	External Clock Low Time—above (>) 20 mHz	17		
t_{PL}	External Clock Low Time—below (≤) 20 mHz	20		
t_{RC}	Memory Cycle Time	10tp − 30	12tp	4, 5
t_{REF}	Refresh Time (128 cycles)	264tp	288tp	
t_{RP}	$\overline{RAS}$ Precharge Time	4tp − 30		
t_{RSH}	$\overline{RAS}$ Hold After $\overline{CAS}$	5tp − 30		3
t_{ASR}	Address Setup to $\overline{RAS}$	tp − 30		3
t_{RAH}	Address Hold From $\overline{RAS}$	tp − 10		3
t_{ASC}	Address Setup to $\overline{CAS}$	tp − 30		3
t_{CAH}	Address Hold from $\overline{CAS}$	5tp − 20		3
t_{CAS}	$\overline{CAS}$ Pulse Width	5tp − 10		
t_{WCS}	$\overline{WE}$ Setup to $\overline{CAS}$	tp − 40		
t_{WCH}	$\overline{WE}$ Hold After $\overline{CAS}$	5tp − 35		8
t_{RS}	$\overline{RD}$, $\overline{WR}$, ALE, REFRQ delay from $\overline{RAS}$	5tp		2, 6
t_{MRP}	$\overline{RD}$, $\overline{WR}$ setup to $\overline{RAS}$	0		5
t_{RMS}	REFRQ setup to $\overline{RD}$, $\overline{WR}$	2tp		6
t_{RMP}	REFRQ setup to $\overline{RAS}$	2tp		5
t_{PCS}	$\overline{PCS}$ Setup to $\overline{RD}$, $\overline{WR}$, ALE	20		
t_{AL}	S1 Setup to ALE	15		
t_{LA}	S1 Hold from ALE	30		
t_{CR}	$\overline{RD}$, $\overline{WR}$, ALE to $\overline{RAS}$ Delay	tp + 30	2tp + 70	2
t_{CC}	$\overline{RD}$, $\overline{WR}$, ALE to $\overline{CAS}$ Delay	3tp + 25	4tp + 85	2
t_{SC}	CMD Setup to Clock	15		1
t_{MRS}	$\overline{RD}$, $\overline{WR}$ setup to REFRQ	5		2
t_{CA}	$\overline{RD}$, $\overline{WR}$, ALE to $\overline{SACK}$ Delay		2tp + 47	2, 9
t_{CX}	$\overline{CAS}$ to $\overline{XACK}$ Delay	5tp − 25	5tp + 20	
t_{CS}	$\overline{CAS}$ to $\overline{SACK}$ Delay	5tp − 25	5tp + 40	2, 10
t_{ACK}	$\overline{XACK}$ to $\overline{CAS}$ Setup	10		
t_{XW}	$\overline{XACK}$ Pulse Width	tp − 25		7
t_{CK}	$\overline{SACK}$, $\overline{XACK}$ turn-off Delay		35	
t_{KCH}	CMD Inactive Hold after $\overline{SACK}$, $\overline{XACK}$	10		
t_{LL}	REFRQ Pulse Width	20		
t_{CHS}	CMD Hold Time	30		11
t_{RFR}	REFRQ to $\overline{RAS}$ Delay		4tp + 100	6
t_{WW}	$\overline{WR}$ to $\overline{WE}$ Delay	0	50	8
t_{AD}	CPU Address Delay	0	40	3

WAVEFORMS

Normal Read or Write Cycle

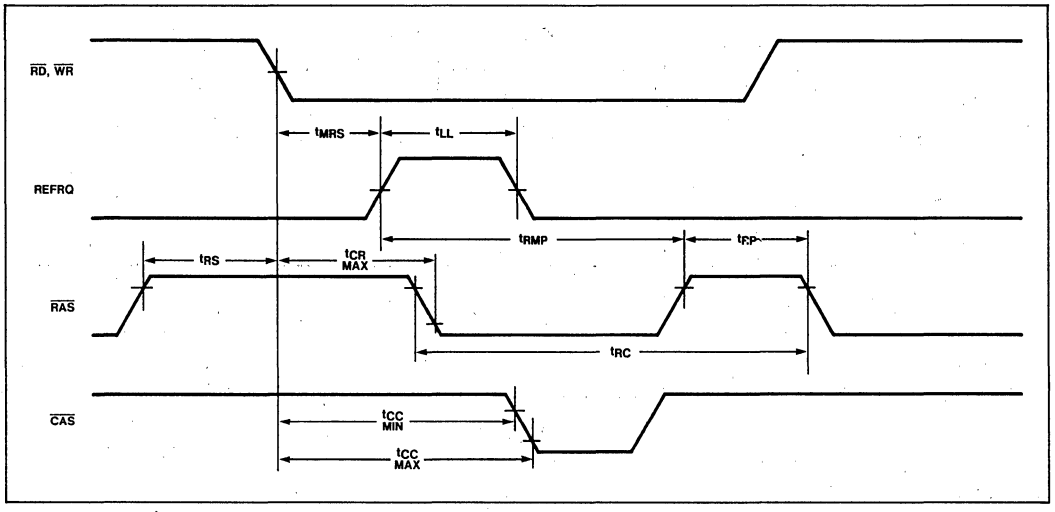


Advanced Read Mode

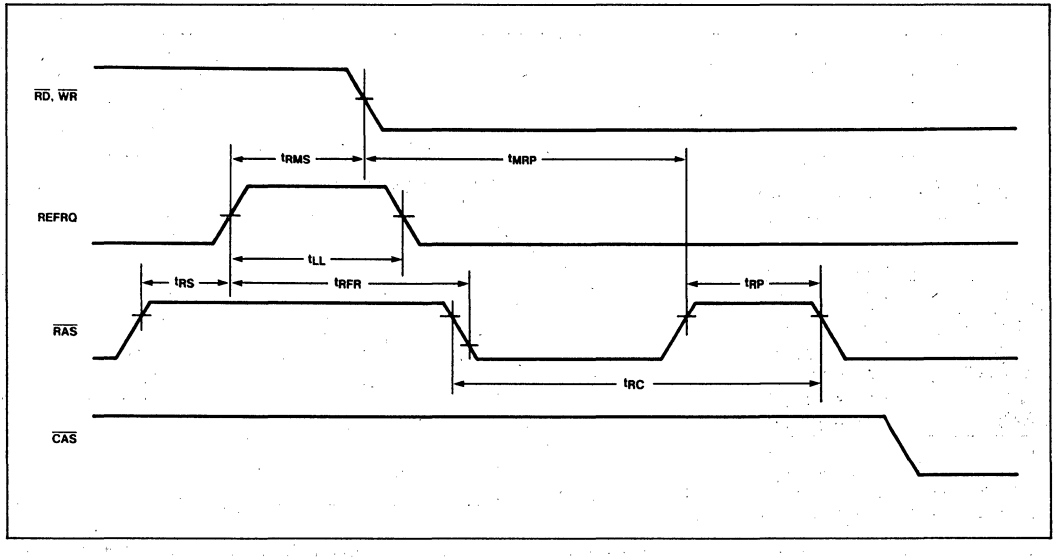


WAVEFORMS (cont'd)

Read or Write Followed By External Refresh



External Refresh Followed By Read or Write



WAVEFORMS (cont'd)

Clock And System Timing

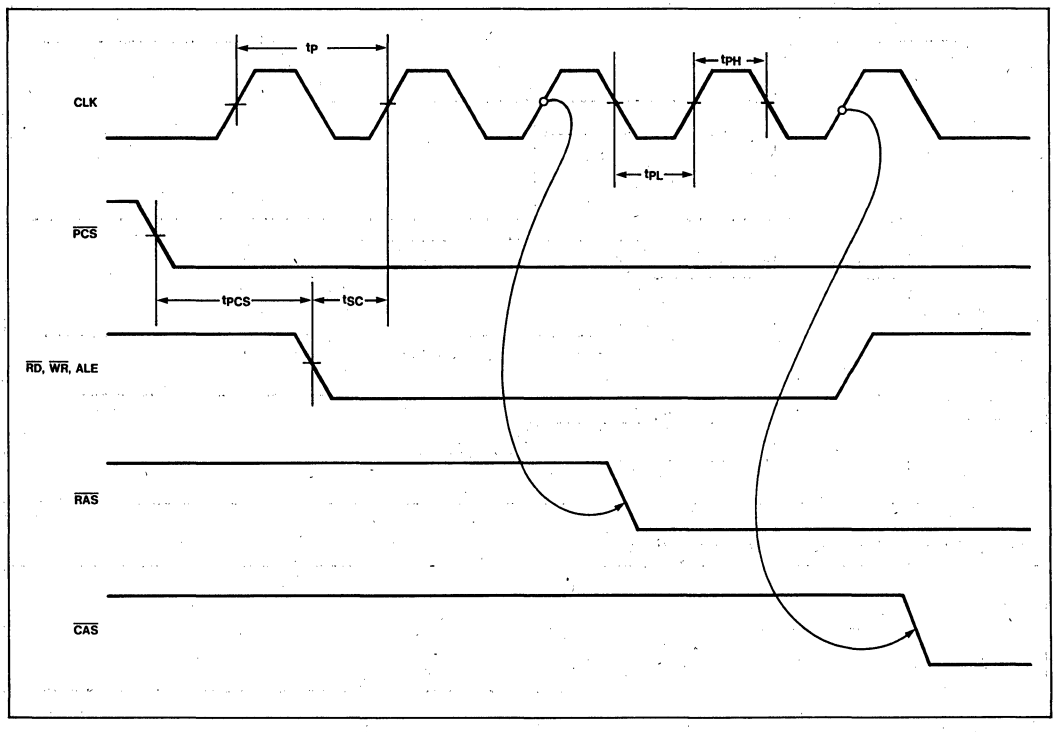


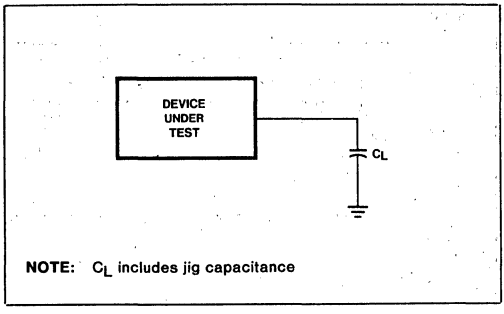
Table 2. 8203 Output Loading. All specifications are for the Test Load unless otherwise noted.

Pin	Test Load
$\overline{SACK}$, $\overline{XACK}$	C_L = 30 pF
$\overline{OUT}_0$-$\overline{OUT}_6$	C_L = 160 pF
$\overline{RAS}_0$-$\overline{RAS}_3$	C_L = 60 pF
$\overline{WE}$	C_L = 224 pF
$\overline{CAS}$	C_L = 320 pF

NOTES:

1. t_{SC} is a reference point only. ALE, $\overline{RD}$, $\overline{WR}$, and REFRQ inputs do not have to be externally synchronized to 8203 clock.
2. If t_{RS} min and t_{MRS} min are met then t_{CA}, t_{CR}, and t_{CC} are valid, otherwise t_{CS} is valid.
3. t_{ASR}, t_{RAH}, t_{ASC}, t_{CAH}, and t_{RSH} depend upon B0-B1 and CPU address remaining stable throughout the memory cycle. The address inputs are not latched by the 8203.
4. For back-to-back refresh cycles, t_{RC} max = 13tp
5. t_{RC} max is valid only if t_{RMP} min is met (READ, WRITE followed by REFRESH) or t_{MRP} min is met (REFRESH followed by READ, WRITE).
6. t_{RFR} is valid only if t_{RS} min and t_{RMS} min are met.
7. t_{XW} min applies when $\overline{RD}$, $\overline{WR}$ has already gone high. Otherwise $\overline{XACK}$ follows $\overline{RD}$, $\overline{WR}$.
8. $\overline{WE}$ goes high according to t_{WCH} or t_{WW}, whichever occurs first.

A.C. TESTING LOAD CIRCUIT

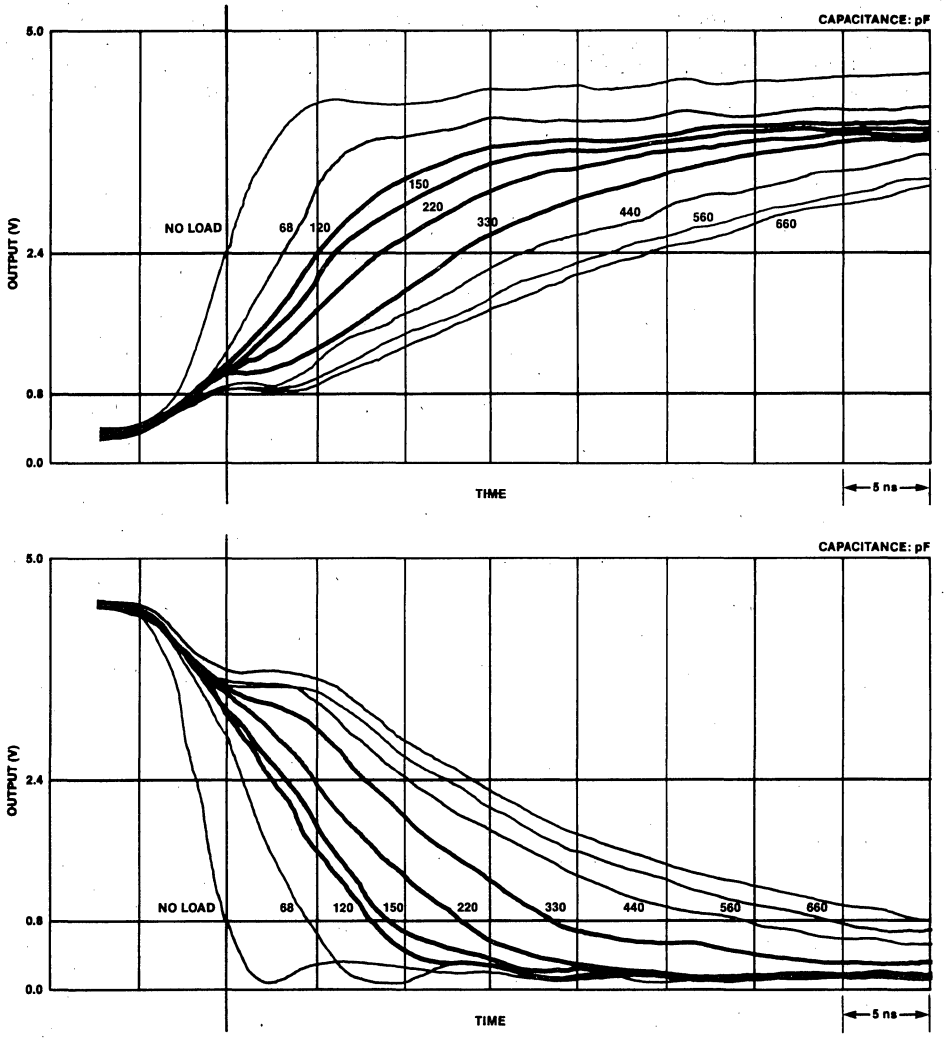


NOTE: C_L includes jig capacitance

9. t_{CA} applies only when in normal $\overline{SACK}$ mode.de.
10. t_{CS} applies only when in delayed $\overline{SACK}$ mode.
11. t_{CHS} must be be met only to ensure a $\overline{SACK}$ active pulse when in delayed $\overline{SACK}$ mode. $\overline{XACK}$ will always be activated for at least t_{XW} (t_P−25 nS). Violating t_{CHS} min does not otherwise affect device operation.

The typical rising and falling characteristic curves for the $\overline{OUT}$, $\overline{RAS}$, $\overline{CAS}$ and $\overline{WE}$ output buffers can be used to determine the effects of capacitive loading on the A.C. Timing Parameters. Using this design tool in conjunction with the timing waveforms, the designer can determine typical timing shifts based on system capacitive load.

A.C. CHARACTERISTICS FOR DIFFERENT CAPACITIVE LOADS



NOTE:
Use the Test Load as the base capacitance for estimating timing shifts for system critical timing parameters.

MEASUREMENT CONDITIONS:

$T_A = 25°C$
$V_{CC} = +5V$
$t_p = 50$ ns

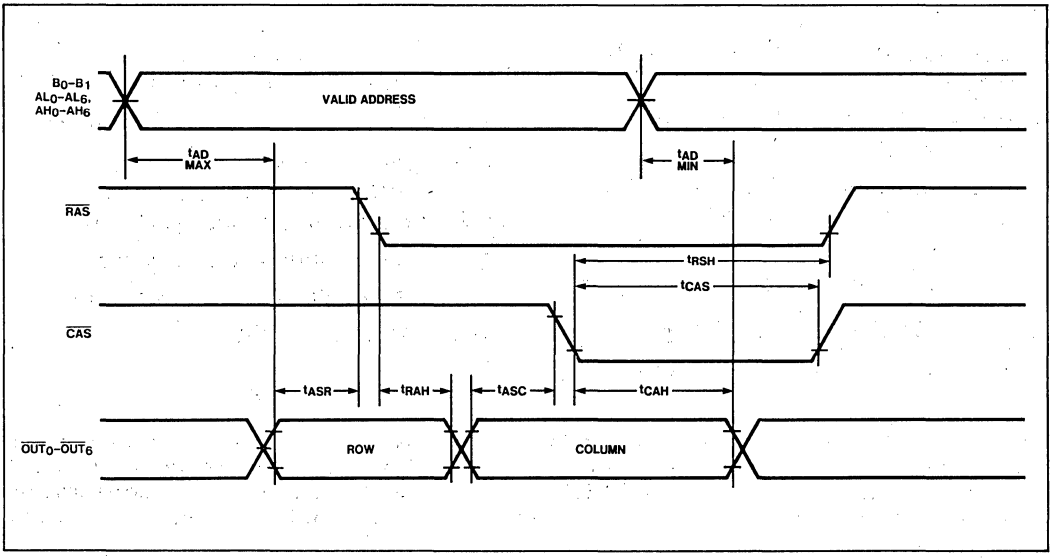
Pins not measured are loaded with the Test Load capacitance

Example: Find the effect on t_{CR} and t_{CC} using 32 2164 Dynamic RAMs configured in 2 banks.

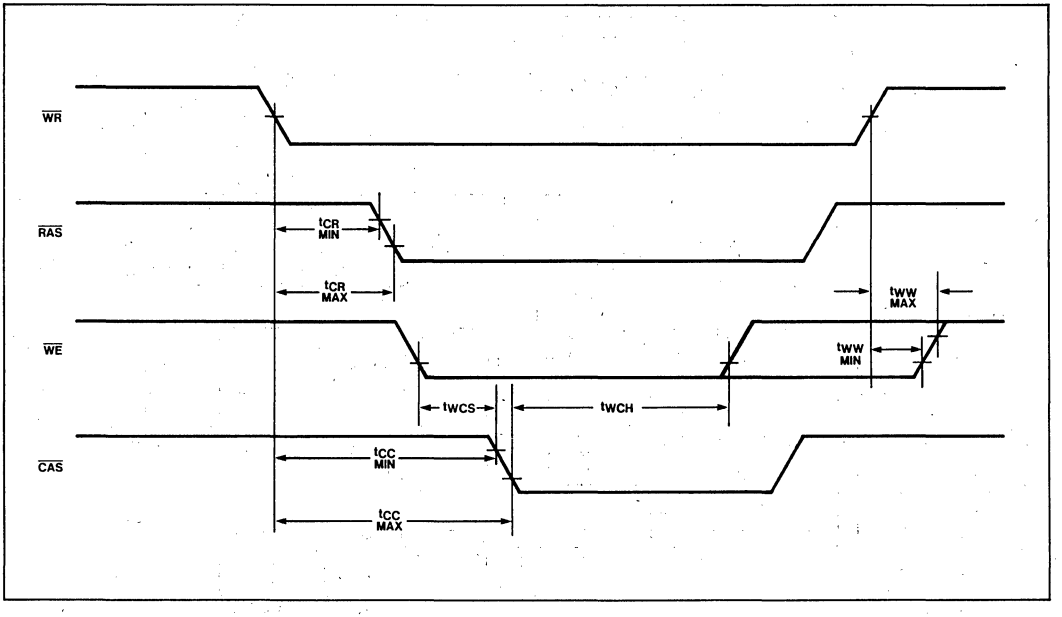
1. Determine the typical RAS and CAS capacitance:
   From the data sheet RAS = 5 pF and CAS = 5 pF.
   ∴ RAS load = 80 pF + board capacitance.
   CAS load = 160 pF + board capacitance.
   Assume 2 pF/in (trace length) for board capacitance and for this example 4 inches for RAS and 8 inches for CAS.
2. From the waveform diagrams, we determine that the falling edge timing is needed for t_{CR} and t_{CC}. Next find the curve that *best* approximates the test load; i.e., 68 pF for RAS and 330 pF for CAS.
3. If we use 88 pF for RAS loading, then t_{CR} (min.) spec should be increased by about 1 ns, and t_{CR} (max.) spec should be increased by *about* 2 ns. Similarly if we use 176 pF for CAS, then t_{CC} (min.) should decrease by 3 ns and t_{CC} (max.) should decrease by about 7 ns.

WAVEFORMS (cont'd)

Memory Compatibility Timing



Write Cycle Timing



8206/8206-2
ERROR DETECTION AND CORRECTION UNIT

- **Detects and Corrects All Single Bit Errors**
- **Detects All Double Bit and Most Multiple Bit Errors**
- **52 ns Maximum for Detection; 67 ns Maximum for Correction (16 Bit System)**
- **Syndrome Outputs for Error Logging**
- **8206-2 Timing Optimized for 8MHz iAPX 186, 188, 86, 88 and 8207-2 Systems**
- **Separate Input and Output Busses—No Timing Strobes Required**
- **Expandable to Handle 80 Bit Memories**
- **Supports Reads With and Without Correction, Writes, Partial (Byte) Writes, and Read-Modify-Writes**
- **HMOS Technology for Low Power**
- **68 Pin Leadless JEDEC Package**
- **Single +5V Supply**

The HMOS 8206 Error Detection and Correction Unit is a high-speed device that provides error detection and correction for memory systems (static and dynamic) requiring high reliability and performance. Each 8206 handles 8 or 16 data bits and up to 8 check bits. 8206's can be cascaded to provide correction and detection for up to 80 bits of data. Other 8206 features include the ability to handle byte writes, memory initialization, and error logging.

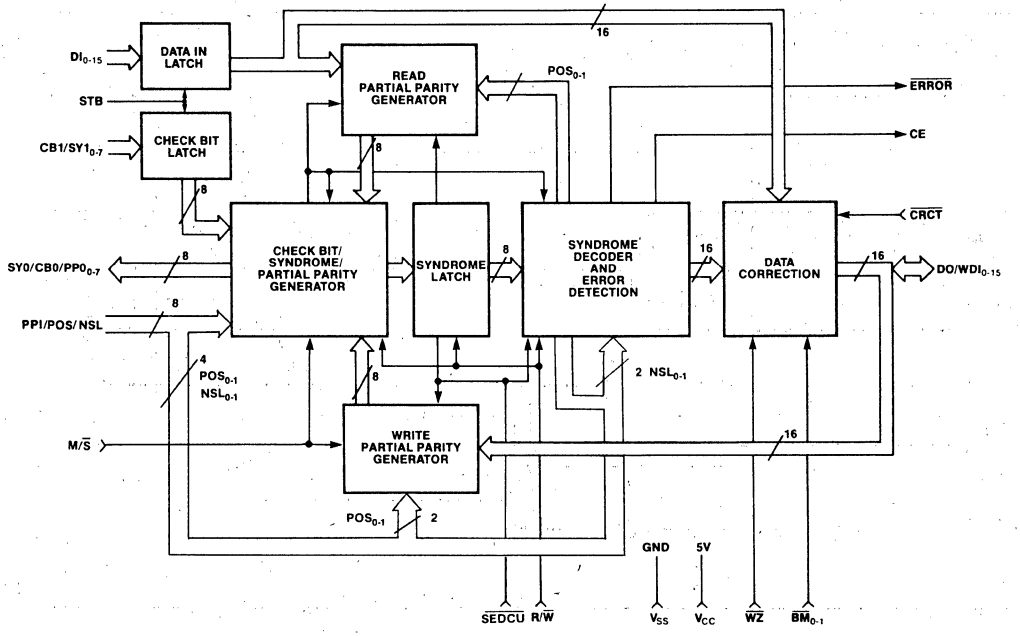


Figure 1. 8206 Block Diagram

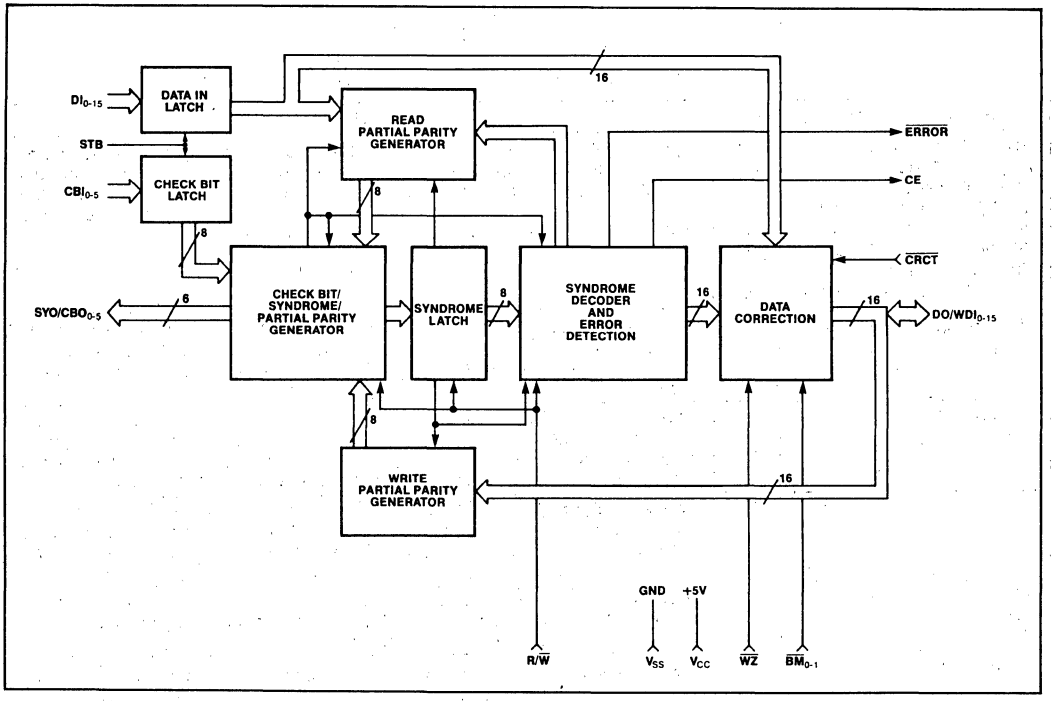


Figure 2. 8206–2 Block Diagram

Table 1. 8206 Pin Description

Symbol	Pin No.	Type	Name and Function
DI_{0-15}	1, 68-61, 59-53	I	**Data In:** These inputs accept a 16 bit data word from RAM for error detection and/or correction.
CBI/SYI_0 CBI/SYI_1 CBI/SYI_2 CBI/SYI_3 CBI/SYI_4 CBI/SYI_5 CBI/SYI_6 CBI/SYI_7	5 6 7 8 9 10 11 12	I I I I I I I I	**Check Bits In/Syndrome In:** In a single 8206 system, or in the master in a multi-8206 system, these inputs accept the check bits (5 to 8) from the RAM. In a single 8206 16 bit system, CBI_{0-5} are used. In slave 8206's these inputs accept the syndrome from the master.
DO/WDI_0 DO/WDI_1 DO/WDI_2 DO/WDI_3 DO/WDI_4 DO/WDI_5 DO/WDI_6 DO/WDI_7 DO/WDI_8 DO/WDI_9 DO/WDI_{10} DO/WDI_{11} DO/WDI_{12} DO/WDI_{13} DO/WDI_{14} DO/WDI_{15}	51 50 49 48 47 46 45 44 42 41 40 39 38 37 36 35	I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O	**Data Out/Write Data In:** In a read cycle, data accepted by DI_{0-15} appears at these outputs corrected if $\overline{CRCT}$ is low, or uncorrected if $\overline{CRCT}$ is high. The $\overline{BM}$ inputs must be high to enable the output buffers during the read cycle. In a write cycle, data to be written into the RAM is accepted by these inputs for computing the write check bits. In a partial-write cycle, the byte not to be modified appears at either DO_{0-7} if $\overline{BM}_0$ is high, or DO_{8-15} if $\overline{BM}_1$ is high, for writing to the RAM. When $\overline{WZ}$ is active, it causes the 8206 to output all zeros at DO_{0-15}, with the proper write check bits on CBO.

Table 1. 8206 Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
$SYO/CBO/PPO_0$ $SYO/CBO/PPO_1$ $SYO/CBO/PPO_2$ $SYO/CBO/PPO_3$ $SYO/CBO/PPO_4$ $SYO/CBO/PPO_5$ $SYO/CBO/PPO_6$ $SYO/CBO/PPO_7$	23 24 25 27 28 29 30 31	O O O O O O O O	**Syndrome Out/Check Bits Out/Partial Parity Out:** In a single 8206 system, or in the master in a multi-8206 system, the syndrome appears at these outputs during a read. During a write, the write check bits appear. In slave 8206's the partial parity bits used by the master appear at these outputs. The syndrome is latched (during read-modify-writes) by R/$\overline{W}$ going low.
PPI_0/POS_0 PPI_1/POS_1	13 14	I I	**Partial Parity In/Position:** In the master in a multi-8206 system, these inputs accept partial parity bits 0 and 1 from the slaves. In a slave 8206 these inputs inform it of its position within the system (1 to 4). Not used in a single 8206 system.
PPI_2/NSL_0 PPI_3/NSL_1	15 16	I I	**Partial Parity In/Number of Slaves:** In the master in a multi-8206 system, these inputs accept partial parity bits 2 and 3 from the slaves. In a multi-8206 system these inputs are used in slave number 1 to tell it the total number of slaves in the system (1 to 4). Not used in other slaves or in a single 8206 system.
PPI_4/CE	17	I/O	**Partial Parity In/Correctable Error:** In the master in a multi-8206 system this pin accepts partial parity bit 4. In slave number 1 only, or in a single 8206 system, this pin outputs the correctable error flag. CE is latched by R/$\overline{W}$ going low. Not used in other slaves.
PPI_5 PPI_6 PPI_7	18 19 20	I I I	**Partial Parity In:** In the master in a multi-8206 system these pins accept partial parity bits 5 to 7. The number of partial parity bits equals the number of check bits. Not used in single 8206 systems or in slaves.
$\overline{ERROR}$	22	O	**Error:** This pin outputs the error flag in a single 8206 system or in the master of a multi-8206 system. It is latched by R/$\overline{W}$ going low. Not used in slaves.
$\overline{CRCT}$	52	I	**Correct:** When low this pin causes data correction during a read or read-modify-write cycle. When high, it causes error correction to be disabled, although error checking is still enabled.
STB	2	I	**Strobe:** STB is an input control used to strobe data at the DI inputs and check-bits at the CBI/SYI inputs. The signal is active high to admit the inputs. The signals are latched by the high-to-low transition of STB.
$\overline{BM}_0$ $\overline{BM}_1$	33 32	I I	**Byte Marks:** When high, the Data Out pins are enabled for a read cycle. When low, the Data Out buffers are tristated for a write cycle. $\overline{BM}_0$ controls DO_{0-7}, while $\overline{BM}_1$ controls DO_{8-15}. In partial (byte) writes, the byte mark input is low for the new byte to be written.
R/$\overline{W}$	21	I	**Read/Write:** When high this pin causes the 8206 to perform detection and correction (if $\overline{CRCT}$ is low). When low, it causes the 8206 to generate check bits. On the high-to-low transition the syndrome is latched internally for read-modify-write cycles.
$\overline{WZ}$	34	I	**Write Zero:** When low this input overrides the $\overline{BM}_{0-1}$ and R/$\overline{W}$ inputs to cause the 8206 to output all zeros at DO_{0-15} with the corresponding check bits at CBO_{0-7}. Used for memory initialization.
M/$\overline{S}$	4	I	**Master/Slave:** Input tells the 8206 whether it is a master (high) or a slave (low).
$\overline{SEDCU}$	3	I	**Single EDC Unit:** Input tells the master whether it is operating as a single 8206 (low) or as the master in a multi-8206 system (high). Not used in slaves.
V_{CC}	60	I	**Power Supply:** +5V
V_{SS}	26	I	**Logic Ground**
V_{SS}	43	I	**Output Driver Ground**

Table 2. 8206-2 Pin Description Differences over the 8206.

Symbol	Pin	Type	Name and Function
CBI_{0-5}	5-10	I	**Check Bits In:** In an 8206-2 system, these inputs accept the check bits (5 to 6) from the RAM.
SYO/CBO_0 SYO/CBO_1 SYO/CBO_2 SYO/CBO_3 SYO/CBO_4 SYO/CBO_5	23 24 25 27 28 29	O O O O O O	**Syndrome Out/Check Bits Out:** In an 8206-2 system, the syndrome appears at these outputs during a read. During a write, the write check bits appear. The syndrome is latched (during read-modify-writes) by R/$\overline{W}$ going low.
CE	17	O	**Correctable Error:** In an 8206-2 system, this pin outputs the correctable error flag. CE is latched by R/$\overline{W}$ going low.
$\overline{WZ}$	34	I	**Write Zero:** When low this input overrides the BM_{0-1} and R/W inputs to cause the 8206-2 to output all zeros at DO_{0-15} with the corresponding check bits at CBO_{0-5}. Used for memory initialization.
Strap High	4	I	Must be tied High.
Strap Low	3	I	Must be tied Low.
N.C.	11-16 18-20	I	**Note:** These pins have internal pull-up resistors but if possible should be tied high or low.
N.C.	30, 31	O	**Note:** These are no connect pins and should be left open.

FUNCTIONAL DESCRIPTION

The 8206 Error Detection and Correction Unit provides greater memory system reliability through its ability to detect and correct memory errors. It is a single chip device that can detect and correct all single bit errors and detect all double bit and some higher multiple bit errors. Some other odd multiple bit errors (e.g., 5 bits in error) are interpreted as single bit errors, and the CE flag is raised. While some even multiple bit errors (e.g., 4 bits in error) are interpreted as no error, most are detected as double bit errors. This error handling is a function of the number of check bits used by the 8206 (see Figure 2) and the specific Hamming code used. Errors in check bits are not distinguished from errors in a word.

For more information on error correction codes, see Intel Application Notes AP-46 and AP-73.

A single 8206 or 8206-2 handles 8 or 16 bits of data, and up to 5 8206's can be cascaded in order to handle data paths of 80 bits. For a single 8206 8 bit system, the DI_{8-15}, DO/WDI_{8-15} and BM_1 inputs are grounded. See the Multi-Chip systems section for information on 24-80 bit systems.

The 8206 has a "flow through" architecture. It supports two kinds of error correction architecture: 1) Flow-through, or correct-always; and 2) Parallel, or check-only. There are two separate 16-pin busses, one to accept data from the RAM (DI) and the other to deliver corrected data to the system bus ($\overline{DO}$/WDI). The logic is entirely combinatorial during a read cycle. This is in contrast to an architecture with only one bus, with bidirectional bus drivers that must first read the data and then be turned around to output the corrected data. The latter architecture typically requires additional hardware (latches and/or transceivers) and may be slower in a system due to timing skews of control signals.

DATA WORD BITS	CHECK BITS
8	5
16	6
24	6
32	7
40	7
48	8
56	8
64	8
72	8
80	8

Figure 3. Number of Check Bits Used by 8206

READ CYCLE

With the R/$\overline{W}$ pin high, data is received from the RAM outputs into the DI pins where it is optionally latched by the STB signal. Check bits are generated from the data bits and compared to the check bits read from the RAM into the CBI pins. If an error is detected the $\overline{ERROR}$ flag is activated and the correctable error flag (CE) is used to inform the system whether the error was correctable or not. With the $\overline{BM}$ inputs high, the word appears corrected at the DO pins if the error was correctable, or unmodified if the error was uncorrectable.

If more than one 8206 is being used, then the check bits are read by the master. The slaves generate a partial parity output (PPO) and pass it to the master. The master 8206 then generates and returns the syndrome to the slaves (SYO) for correction of the data.

The 8206 may alternatively be used in a "check-only" mode with the $\overline{CRCT}$ pin left high. With the correction facility turned off, the propagation delay from memory outputs to 8206 outputs is significantly shortened. In this mode the 8206 issues an $\overline{ERROR}$ flag to the CPU, which can then perform one of several options: lengthen the current cycle for correction, restart the instruction, perform a diagnostic routine, etc.

A syndrome word, five to eight bits in length and containing all necessary information about the existence and location of an error, is made available to the system at the SYO_{0-7} pins. Error logging may be accomplished by latching the syndrome and the memory address of the word in error.

WRITE CYCLE

For a full write, in which an entire word is written to memory, the data is written directly to the RAM, bypassing the 8206. The same data enters the 8206 through the WDI pins where check bits are generated. The Byte Mark inputs must be low to tristate the DO drivers. The check bits, 5 to 8 in number, are then written to the RAM through the CBO pins for storage along with the data word. In a multi-chip system, the master writes the check bits using partial parity information from the slaves.

In a partial write, part of the data word is overwritten, and part is retained in memory. This is accomplished by performing a read-modify-write cycle. The complete old word is read into the 8206 and corrected, with the syndrome internally latched by R/$\overline{W}$ going low. Only that part of the word not to be modified is output onto the DO pins, as controlled by the Byte Mark inputs. That portion of the word to be overwritten is supplied by the system bus. The 8206 then calculates check bits for the new word, using the byte from the previous read and the new byte from the system bus, and writes them to the memory.

READ-MODIFY-WRITE CYCLES

Upon detection of an error the 8206 may be used to correct the bit in error in memory. This reduces the probability of getting multiple-bit errors in subsequent read cycles. This correction is handled by executing read-modify-write cycles.

The read-modify-write cycle is controlled by the R/$\overline{W}$ input. After (during) the read cycle, the system dynamic RAM controller or CPU examines the 8206 $\overline{ERROR}$ and CE outputs to determine if a correctable error occurred. If it did, the dynamic RAM controller or CPU forces R/$\overline{W}$ low, telling the 8206 to latch the generated syndrome and drive the corrected check bits onto the CBO outputs. The corrected data is available on the DO pins. The DRAM controller then writes the corrected data and corresponding check bits into memory.

The 8206 may be used to perform read-modify-writes in one or two RAM cycles. If it is done in two cycles, the 8206 latches are used to hold the data and check bits from the read cycle to be used in the following write cycle. The Intel 8207 Advanced Dynamic RAM controller allows read-modify-write cycles in one memory cycle. See the System Environment section.

INITIALIZATION

A memory system operating with ECC requires some form of initialization at system power-up in order to set valid data and check bit information in memory. The 8206 supports memory initialization by the write zero function. By activating the $\overline{WZ}$ pin, the 8206 will write a data pattern of zeros and the associated check bits in the current write cycle. By thus writing to all memory at power-up, a controller can set memory to valid data and check bits. Massive memory failure, as signified by both data and check bits all ones or zeros, will be detected as an uncorrectable error.

MULTI-CHIP SYSTEMS

A single 8206 handles 8 or 16 bits of data and 5 or 6 check bits, respectively. Up to 5 8206's can be cascaded for 80 bit memories with 8 check bits.

When cascaded, one 8206 operates as a master, and all others as slaves. As an example, during a read cycle in a 32 bit system with one master and one slave, the slave calculates parity on its portion of the word—"partial parity"—and presents it to the master through the PPO pins. The master combines the partial parity from the slave with the parity it calculated from its own portion of the word to generate the syndrome. The syndrome is then returned by the master to the slave for error correction. In systems with more than one slave the above description continues to apply, except that the partial parity outputs of the slaves must be XOR'd externally. Figure 4 shows the necessary external logic for multi-chip systems. Write and read-modify-write cycles are carried out analogously. See the System Operation section for multi-chip wiring diagrams.

There are several pins used to define whether the 8206 will operate as a master or a slave. Tables 3 and 4 illustrate how these pins are tied.

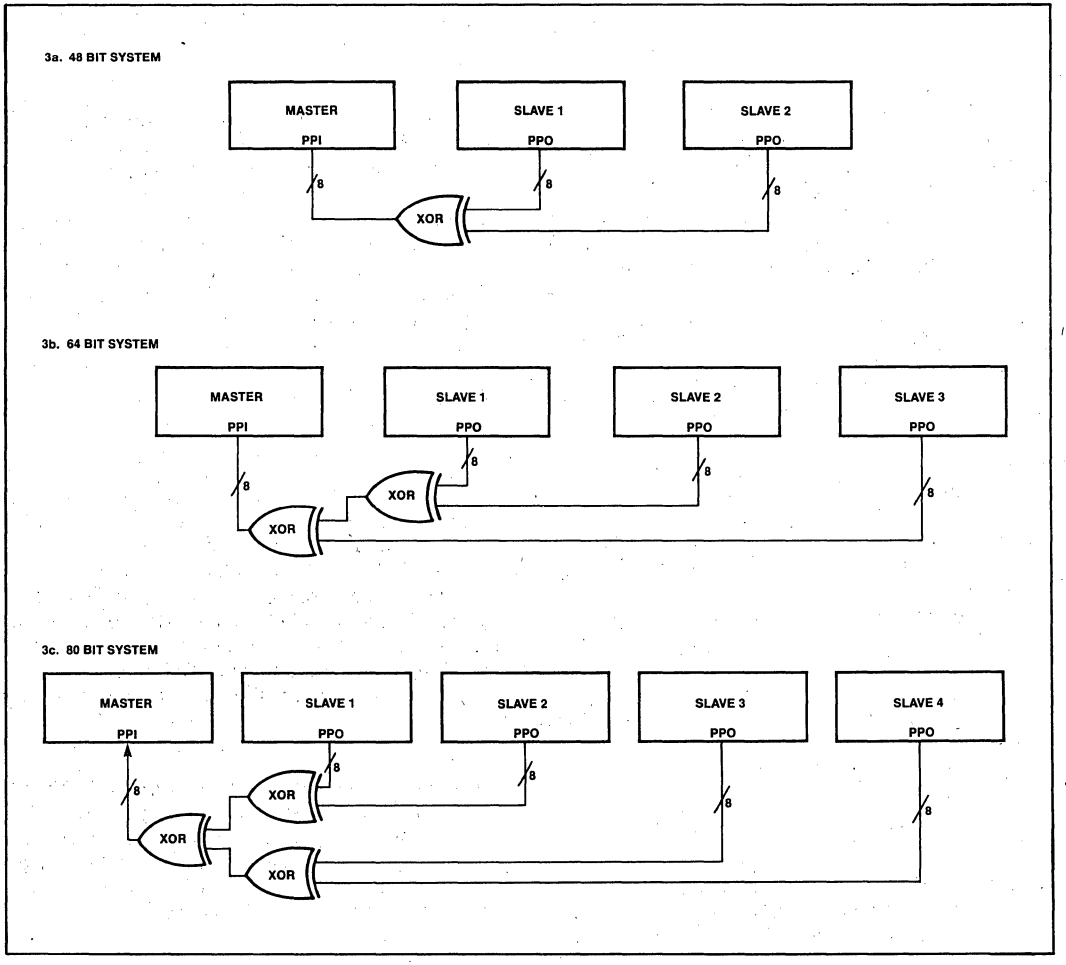


Figure 4. External Logic For Mult-Chip Systems

Table 3. Master/Slave Pin Assignments

Pin No.	Pin Name	Master	Slave 1	Slave 2	Slave 3	Slave 4
4	M/$\overline{S}$	+5V	Gnd	Gnd	Gnd	Gnd
3	$\overline{SEDCU}$	+5V	+5V	+5V	+5V	+5V
13	PPI_0/POS_0	PPI	Gnd	+5V	Gnd	+5V
14	PPI_1/POS_1	PPI	Gnd	Gnd	+5V	+5V
15	PPI_2/NSL_0	PPI	*	+5V	+5V	+5V
16	PPI_3/NSL_1	PPI	*	+5V	+5V	+5V

*See Table 3.

NOTE:
Pins 13, 14, 15, 16 have internal pull-up resistors and may be left as N.C. where specified as connecting to +5V.

Table 4. NSL Pin Assignments for Slave 1

Number of Slaves				
Pin	**1**	**2**	**3**	**4**
PPI_2/NSL_0	Gnd	+5V	Gnd	+5V
PPI_3/NSL_1	Gnd	Gnd	+5V	+5V

The timing specifications for multi-chip systems must be calculated to take account of the external XOR gating in 3, 4, and 5-chip systems. Let tXOR be the delay for a single external TTL XOR gate. Then the following equations show how to calculate the relevant timing parameters for 2-chip (n=0), 3-chip (n=1), 4-chip (n=2), and 5-chip (n=2) systems:

Data-in to corrected data-out (read cycle) =
TDVSV + TPVSV + TSVQV + ntXOR

Data-in to error flag (read cycle) =
TDVSV + TPVEV + ntXOR

Data-in to correctable error flag (read cycle) =
TDVSV + TPVSV + TSVCV + ntXOR

Write data to check-bits valid (full write cycle) =
TQVQV + TPVSV + ntXOR

Data-in to check-bits valid (read-mod-write cycle) =
TDVSV + TPVSV + TSVQV + TQVQV + TPVSV + 2ntXOR

Data-in to check-bits valid (non-correcting read-modify-write cycle) =
TDVQU + TQVQV + TPVSV + ntXOR

HAMMING CODE

The 8206 uses a modified Hamming code which was optimized for multi-chip EDCU systems. The code is such that partial parity is computed by all 8206's in parallel. No 8206 requires more time for propagation through logic levels than any other one, and hence no one device becomes a bottleneck in the parity operation. However, one or two levels of external TTL XOR gates are required in systems with three to five chips. The code appears in Table 5. The check bits are derived from the table by XORing or XNORing together the bits indicated by 'X's in each row corresponding to a check bit. For example, check bit 0 in the MASTER for data word 1000110101101011 will be "0." It should be noted that the 8206 will detect the gross-error condition of all lows or all highs.

Error correction is accomplished by identifying the bad bit and inverting it. Table 5 can also be used as an error syndrome table by replacing the 'X's with '1's. Each column then represents a different syndrome word, and by locating the column corresponding to a particular syndrome the bit to be corrected may be identified. If the syndrome cannot be located then the error cannot be corrected. For example, if the syndrome word is 00110111, the bit to be corrected is bit 5 in the slave one data word (bit 21).

The syndrome decoding is also summarized in Tables 6 and 7 which can be used for error logging. By finding the appropriate syndrome word (starting with bit zero, the least significant bit), the result is either: 1) no error; 2) an identified (correctable) single bit error; 3) a double bit error; or 4) a multi-bit uncorrectable error.

Table 5. Modified Hamming Code Check Bit Generation

Check bits are generated by XOR'ing (except for the CB0 and CB1 data bits, which are XNOR'ed in the Master) the data bits in the rows corresponding to the check bits. Note there are 6 check bits in a 16-bit system, 7 in a 32-bit system, and 8 in 48-or-more-bit systems.

BYTE NUMBER		0								1								OPERATION
BIT NUMBER		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
CHECK BITS	CB0 =	x	x	-	x	-	x	x	-	x	-	-	x	-	x	-	-	XNOR
	CB1 =	x	-	x	-	-	x	-	x	-	x	-	x	x	-	x	-	XNOR
	CB2 =	-	x	x	-	x	-	x	x	-	-	x	-	x	-	-	x	XOR
	CB3 =	x	x	x	x	x	-	-	-	x	x	x	-	-	-	-	-	XOR
	CB4 =	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	XOR
	CB5 =	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	XOR
	CB6 =	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	XOR
	CB7 =	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	XOR
DATA BITS		0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	

16 BIT OR MASTER

	2								3								OPERATION
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
	-	x	x	x	-	x	x	-	-	x	x	-	-	x	-	-	XOR
	x	x	x	-	-	x	-	x	x	x	-	-	-	-	-	x	XOR
	-	x	x	x	-	x	x	x	-	-	x	x	-	-	-	-	XOR
	x	x	-	-	x	-	x	x	x	-	-	x	x	-	-	-	XOR
	x	x	-	-	x	x	x	x	-	-	-	-	x	-	x	-	XOR
	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	XOR
	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	XOR
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	XOR
	1 6	1 7	1 8	1 9	2 0	2 1	2 2	2 3	2 4	2 5	2 6	2 7	2 8	2 9	3 0	3 1	

SLAVE #1

BYTE NUMBER		4								5								6								7								8								9								OPERATION
BIT NUMBER		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
CHECK BITS	CB0 =	x	x	-	x	-	x	x	-	x	-	-	x	-	x	-	-	x	-	x	x	-	-	x	-	x	-	x	x	-	-	x	-	-	x	x	-	x	-	x	-	-	x	-	-	x	-	-	-	XOR
	CB1 =	x	-	x	-	-	x	-	x	-	x	-	x	x	-	x	-	-	x	x	-	-	-	x	-	x	x	-	-	-	x	-	x	-	x	x	-	x	-	x	x	-	-	x	x	-	-	-	-	XOR
	CB2 =	-	x	x	-	x	-	x	x	-	-	x	-	x	-	-	x	-	x	x	-	x	-	-	x	-	x	x	-	x	-	x	x	x	-	-	x	-	x	-	x	-	-	x	x	-	x	-	x	XOR
	CB3 =	x	x	x	x	x	-	-	-	x	x	x	-	-	-	-	-	x	-	x	-	-	x	x	-	x	x	-	-	x	x	-	-	-	x	x	x	x	-	-	x	-	-	x	x	-	x	-	-	XOR
	CB4 =	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	-	x	x	-	-	-	x	x	x	x	x	-	-	-	x	-	XOR
	CB5 =	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	x	-	x	x	x	x	x	-	-	-	-	x	-	-	-	x	XOR
	CB6 =	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	-	-	x	x	x	x	-	-	-	-	x	-	x	-	XOR
	CB7 =	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	XOR
DATA BITS		3 2	3 3	3 4	3 5	3 6	3 7	3 8	3 9	4 0	4 1	4 2	4 3	4 4	4 5	4 6	4 7	4 8	4 9	5 0	5 1	5 2	5 3	5 4	5 5	5 6	5 7	5 8	5 9	6 0	6 1	6 2	6 3	6 4	6 5	6 6	6 7	6 8	6 9	7 0	7 1	7 2	7 3	7 4	7 5	7 6	7 7	7 8	7 9	

SLAVE #2 | SLAVE #3 | SLAVE #4

Table 6. 8206 Syndrome Decoding

Syndrome Bits				0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
				1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
7	6	5	4	3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0		N	CB0	CB1	D	CB2	D	D	18	CB3	D	D	0	D	1	2	D
0	0	0	1		CB4	D	D	5	D	6	7	D	D	3	16	D	4	D	D	17
0	0	1	0		CB5	D	D	11	D	19	12	D	D	8	9	D	10	D	D	67
0	0	1	1		D	13	14	D	15	D	D	21	20	D	D	66	D	22	23	D
0	1	0	0		CB6	D	D	25	D	26	49	D	D	48	24	D	27	D	D	50
0	1	0	1		D	52	55	D	51	D	D	70	28	D	D	65	D	53	54	D
0	1	1	0		D	29	31	D	64	D	D	69	68	D	D	32	D	33	34	D
0	1	1	1		30	D	D	37	D	38	39	D	D	35	71	D	36	D	D	U
1	0	0	0		CB7	D	D	43	D	77	44	D	D	40	41	D	42	D	D	U
1	0	0	1		D	45	46	D	47	D	D	74	72	D	D	U	D	73	U	D
1	0	1	0		D	59	75	D	79	D	D	58	60	D	D	56	D	U	57	D
1	0	1	1		63	D	D	62	D	U	U	D	D	U	U	D	61	D	D	U
1	1	0	0		D	U	U	D	U	D	D	U	76	D	D	U	D	U	U	D
1	1	0	1		78	D	D	U	D	U	U	D	D	U	U	D	U	D	D	U
1	1	1	0		U	D	D	U	D	U	U	D	D	U	U	D	U	D	D	U
1	1	1	1		D	U	U	D	U	D	D	U	U	D	D	U	D	U	U	D

N = No Error
CBX = Error in Check Bit X
X = Error in Data Bit X
D = Double Bit Error
U = Uncorrectable Multi-Bit Error

SYSTEM ENVIRONMENT

The 8206 interface to a typical 32 bit memory system is illustrated in Figure 5. For larger systems, the partial parity bits from slaves two to four must be

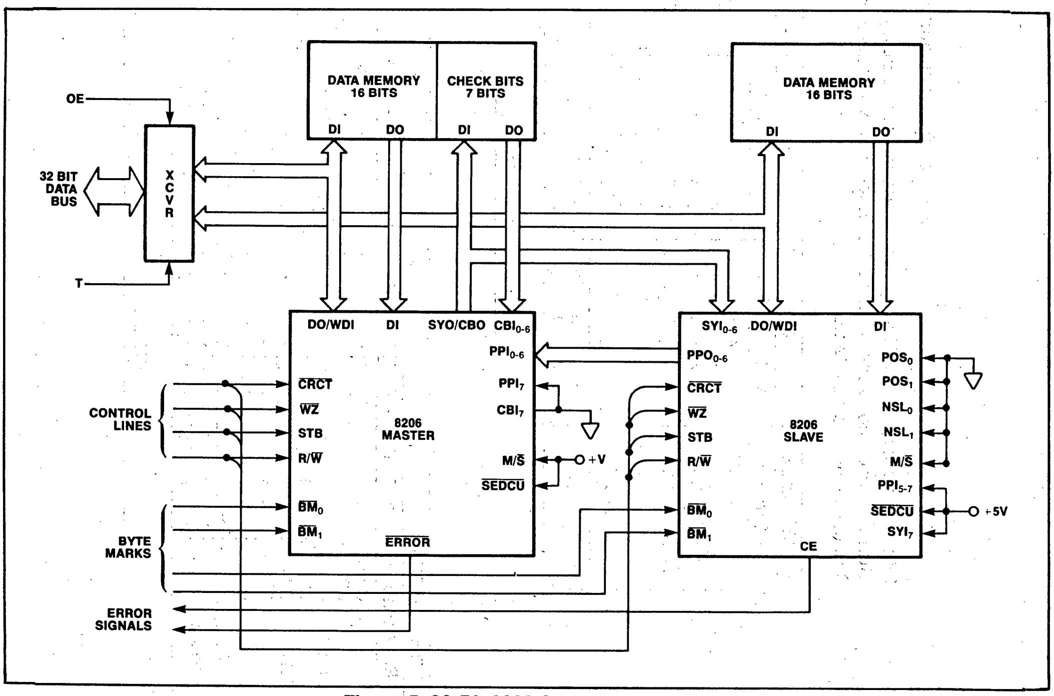


Figure 5. 32-Bit 8206 System Interface

XOR'ed externally, which calls for one level of XOR gating for three 8206's and two levels for four or five 8206's.

The 8206 is designed for direct connection to the Intel 8207 Advanced Dynamic RAM Controller. The 8207 has the ability to perform dual port memory control, and Figure 6 illustrates a highly integrated dual port RAM implementation using the 8206 and 8207. The 8206/8207 combination permits such features as automatic scrubbing (correcting errors in memory during refresh), extending RAS and CAS timings for Read-Modify-Writes in single memory cycles, and automatic memory initialization upon reset. Together these two chips provide a complete dual-port, error-corrected dynamic RAM subsystem.

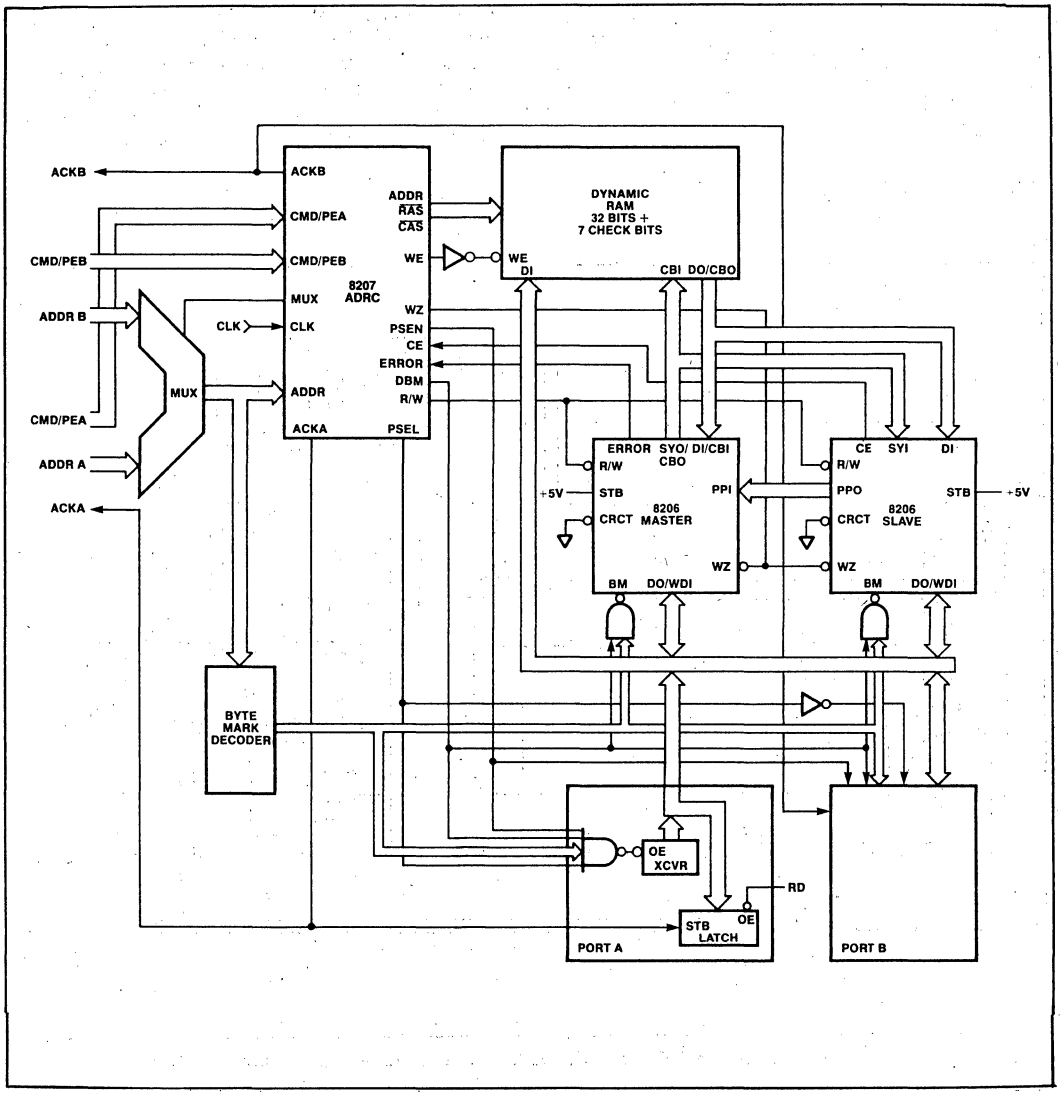


Figure 6. Dual Port RAM Subsystem with 8206/8207 (32-bit bus)

Table 7. 8206-2 Syndrome Decoding

Syndrome Bits 5	4	3	0 1 2	0 0 0	1 0 0	0 1 0	1 1 0	0 0 1	1 1 1	0 1 1	1 1 1
0	0	0		N	CB0	CB1	D	CB2	D	D	D
0	0	1		CB3	D	D	0	D	1	2	D
0	1	0		CB4	D	D	5	D	6	7	D
0	1	1		D	3	D	D	4	D	D	D
1	0	0		CB5	D	D	11	D	D	12	D
1	0	1		D	8	9	D	10	D	D	D
1	1	0		D	13	14	D	15	D	D	D
1	1	1		D	D	D	D	D	D	D	D

N = No Error
CBX = Error in Check Bit X
X = Error in Data Bit X
D = Double Bit Error

The 8206-2 handles 8 or 16 bits of data. For 8 bit 8206-2 systems, the DI_{8-15}, DO/WDI_{8-15} and $\overline{BM}_1$ inputs are grounded.

The 8206-2 is designed for direct connection to the Intel 8207-2 Advanced Dynamic RAM Controller. The 8207-2 has the ability to perform dual port memory control, and Figure 7 illustrates a highly integrated iAPX 186 RAM implementation using the 8206-2 and 8207-2. The 8206-2/8207-2 combination permits such features as automatic scrubbing (correcting errors in memory during refresh), extending RAS and CAS timings for Read-Modify-Writes in single memory cycles, and automatic memory initialization upon reset. Together these two chips provide a complete dual-port, error-corrected dynamic RAM subsystems.

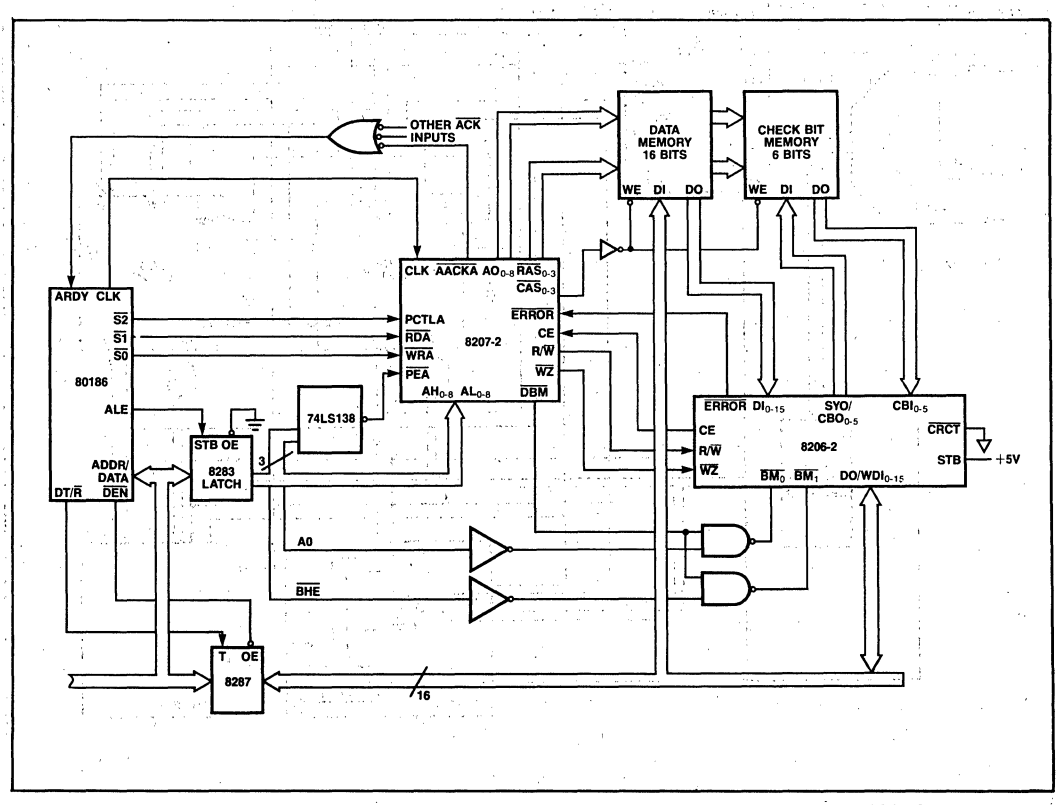


Figure 7. iAPX 186 RAM Correct Always Subsystem with the 8206-2 and the 8207-2

MEMORY BOARD TESTING

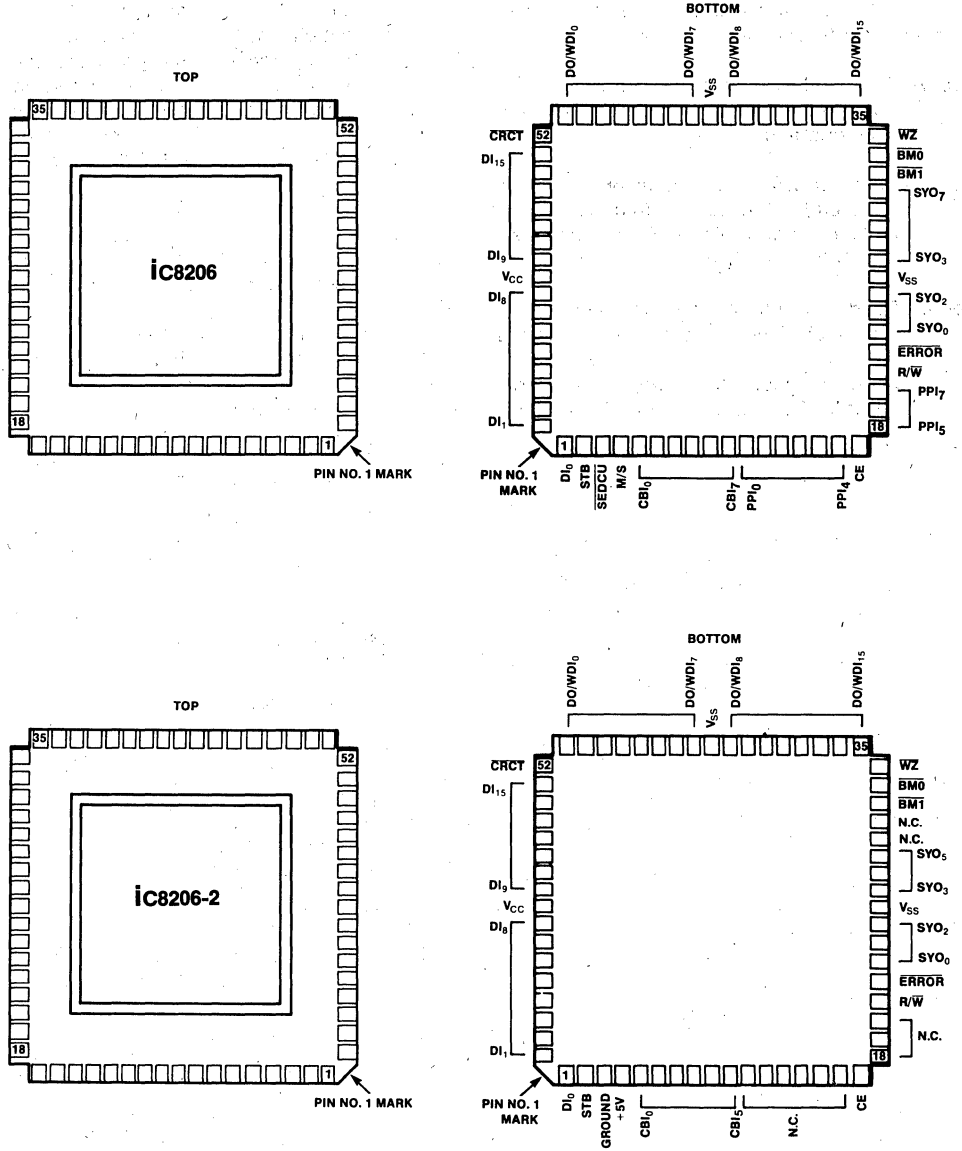
The 8206 lends itself to straightforward memory board testing with a minimum of hardware overhead. The following is a description of four common test modes and their implementation.

Mode 0—Read and write with error correction.
Implementation: This mode is the normal 8206 operating mode.

Mode 1—Read and write data with error correction disabled to allow test of data memory.
Implementation: This mode is performed with $\overline{CRCT}$ deactivated.

Mode 2—Read and write check bits with error correction disabled to allow test of check bits memory.
Implementation: Any pattern may be written into the check bits memory by judiciously choosing the proper data word to generate the desired check bits, through the use of the 8206 Hamming code. To read out the check bits it is first necessary to fill the data memory with all zeros, which may be done by activating $\overline{WZ}$ and incrementing memory addresses with $\overline{WE}$ to the check bits memory held inactive, and then performing ordinary reads. The check bits will then appear directly at the SYO outputs, with bits CB0 and CB1 inverted.

Mode 3—Write data, without altering or writing check bits, to allow the storage of bit combinations to cause error correction and detection.
Implementation: This mode is implemented by writing the desired word to memory with $\overline{WE}$ to the check bits array held inactive.



NOTE:
The 8206 and 8206-2 is packaged in a 68 pin JEDEC TYPE A hermetic chip carrier

Figure 8. 8206 and 8206-2 Pinout Diagram

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... −65°C to +150°C
Voltage On Any Pin
With Respect to Ground ..... −0.5V to +7V
Power Dissipation ..... 1.5 Watts

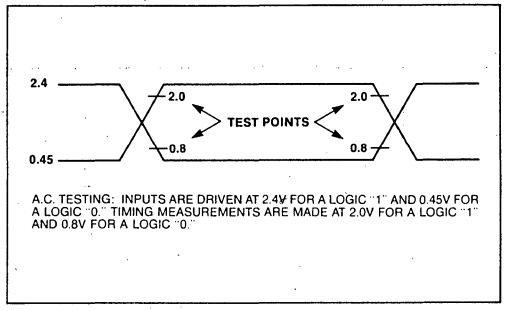
**NOTE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0V ± 10%, V_{SS} = GND)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
I_{CC}	Power Supply Current —Single 8206, 8206-2 or Slave #1 —Master in Multi-Chip or Slaves #2, 3, 4		270 230	mA mA	
V_{IL}[1]	Input Low Voltage	−0.5	0.8	V	
V_{IH}[1]	Input High Voltage	2.0	V_{CC} + 0.5V	V	
V_{OL}	Output Low Voltage —DO —All Others		0.45 0.45	V V	I_{OL} = 8mA I_{OL} = 2.0mA
V_{OH}	Output High Voltage —DO, CBO —All Other Outputs	2.6 2.4		V V	I_{OH} = −2mA I_{OH} = −0.4mA
I_{LO}	I/O Leakage Current —PPI_4/CE —DO/WDI_{0-15}		± 20 ± 10	μA μA	$0.45V \leq V_{I/O} \leq V_{CC}$
I_{LI}	Input Leakage Current —$PPI_{0-3,\ 5-7}$, CBI_{6-7}, $\overline{SEDCU}$[2] —All Other Input Only Pins		± 20 ± 10	μA μA	$0V \leq V_{IN} \leq V_{CC}$

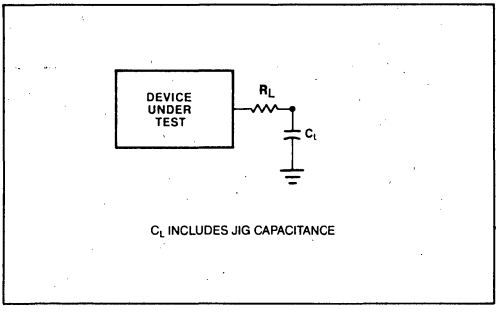
NOTES:
1. $\overline{SEDCU}$ (pin 3) and M/$\overline{S}$ (pin 4) are device strapping options and should be tied to V_{CC} or GND. V_{IH} min = V_{CC} −0.5V and V_{IL} max = 0.5V.
2. PPI_{0-7} (pins 13–20) and CBI_{6-7} (pins 11, 12) have internal pull-up resistors and if left unconnected will be pulled to V_{CC}.

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0."

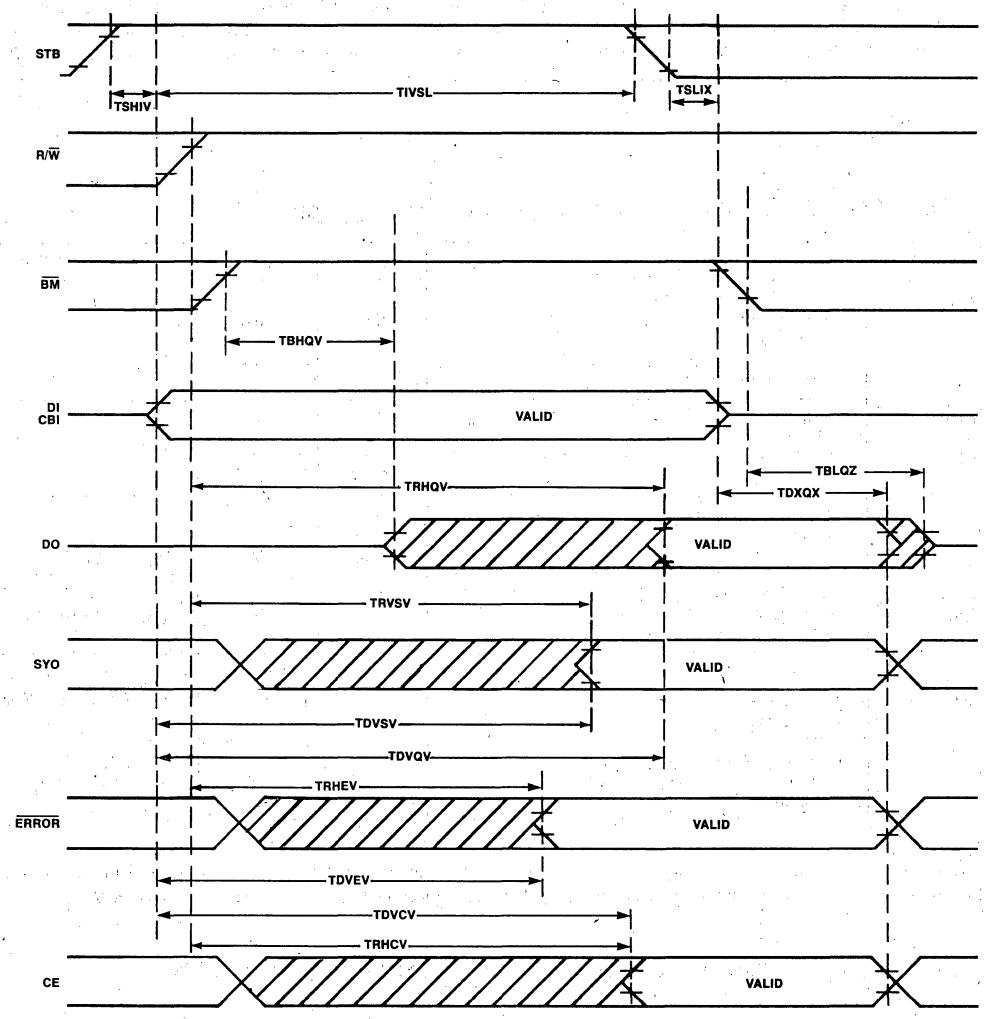
A.C. TESTING LOAD CIRCUIT



C_L INCLUDES JIG CAPACITANCE

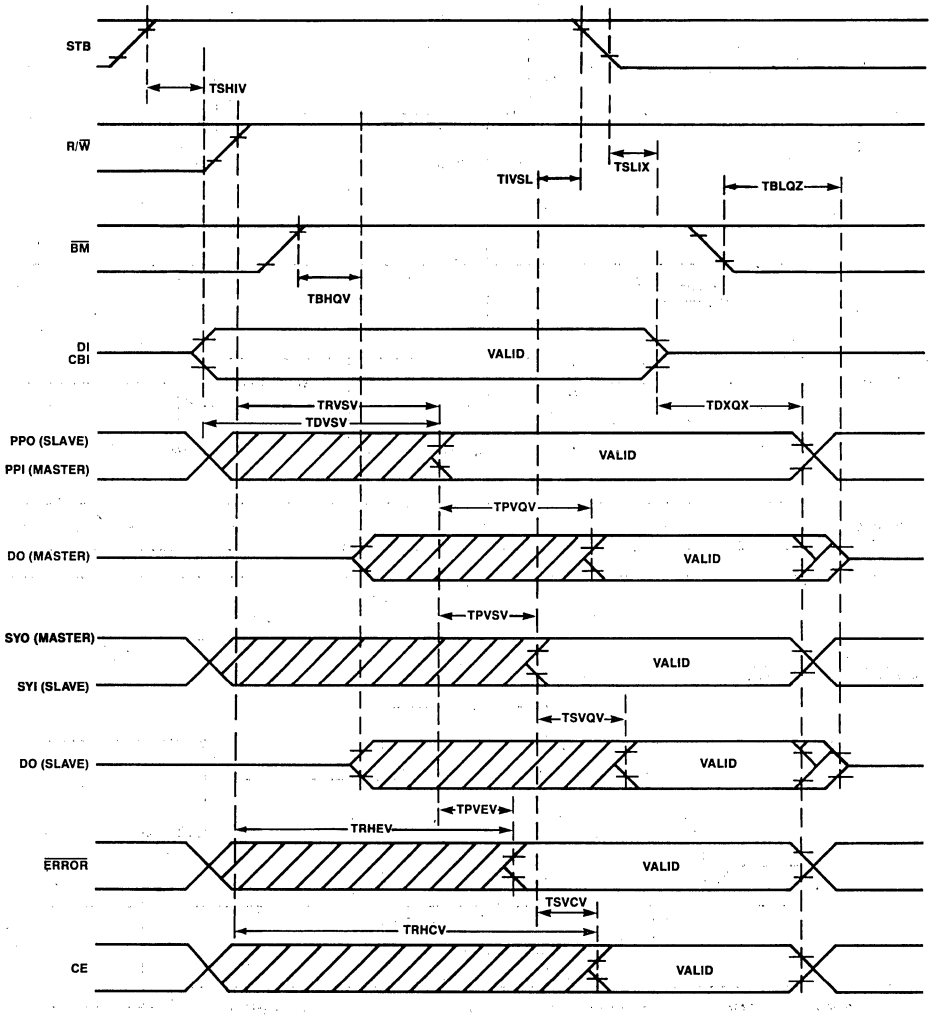
WAVEFORMS

READ



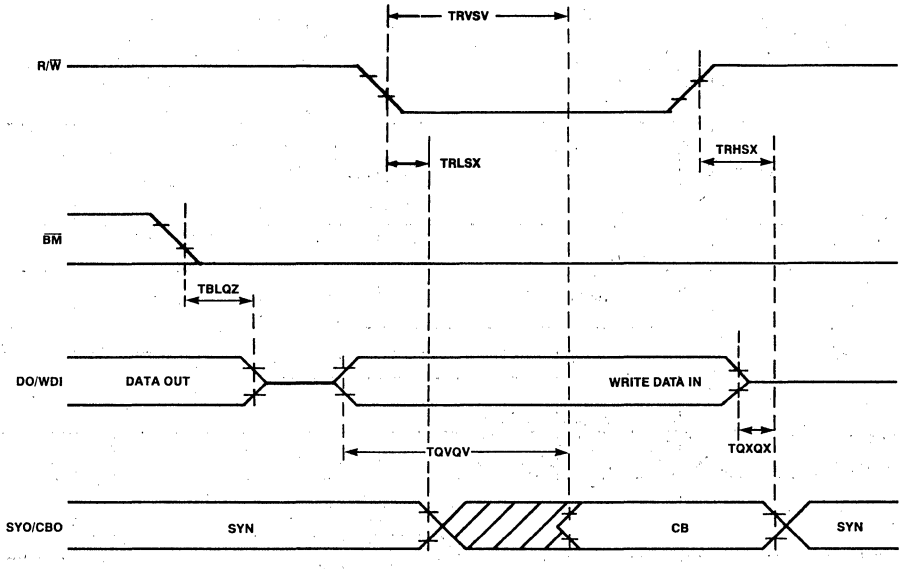
WAVEFORMS (Continued)

READ—MASTER/SLAVE

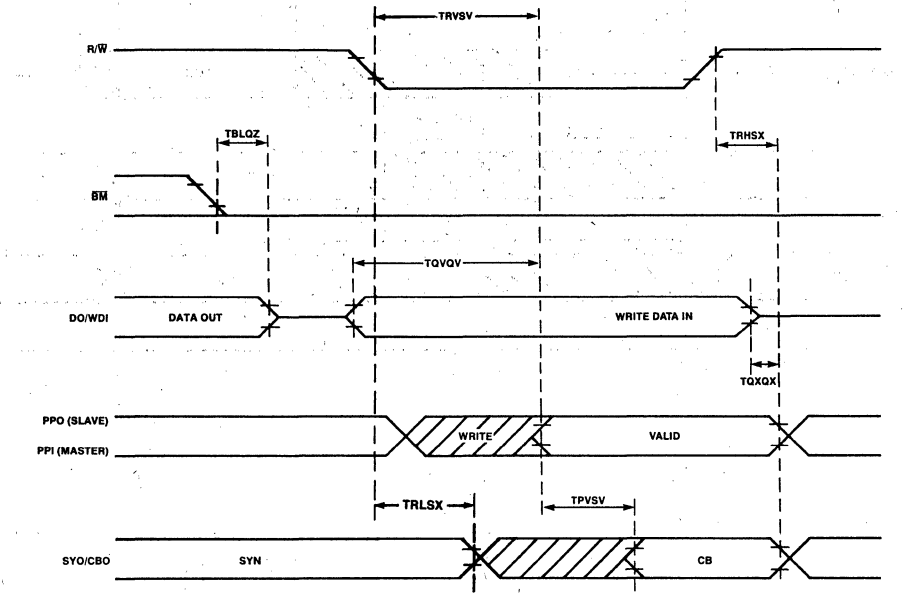


WAVEFORMS (Continued)

FULL WRITE

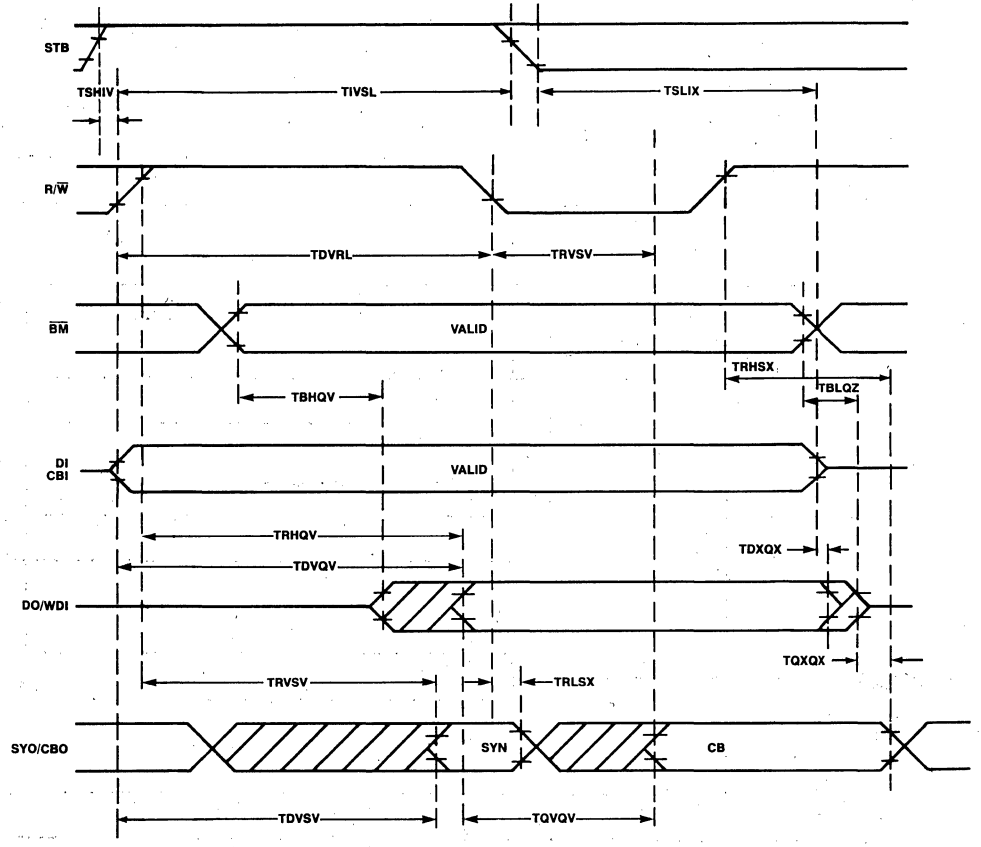


FULL WRITE—MASTER/SLAVE



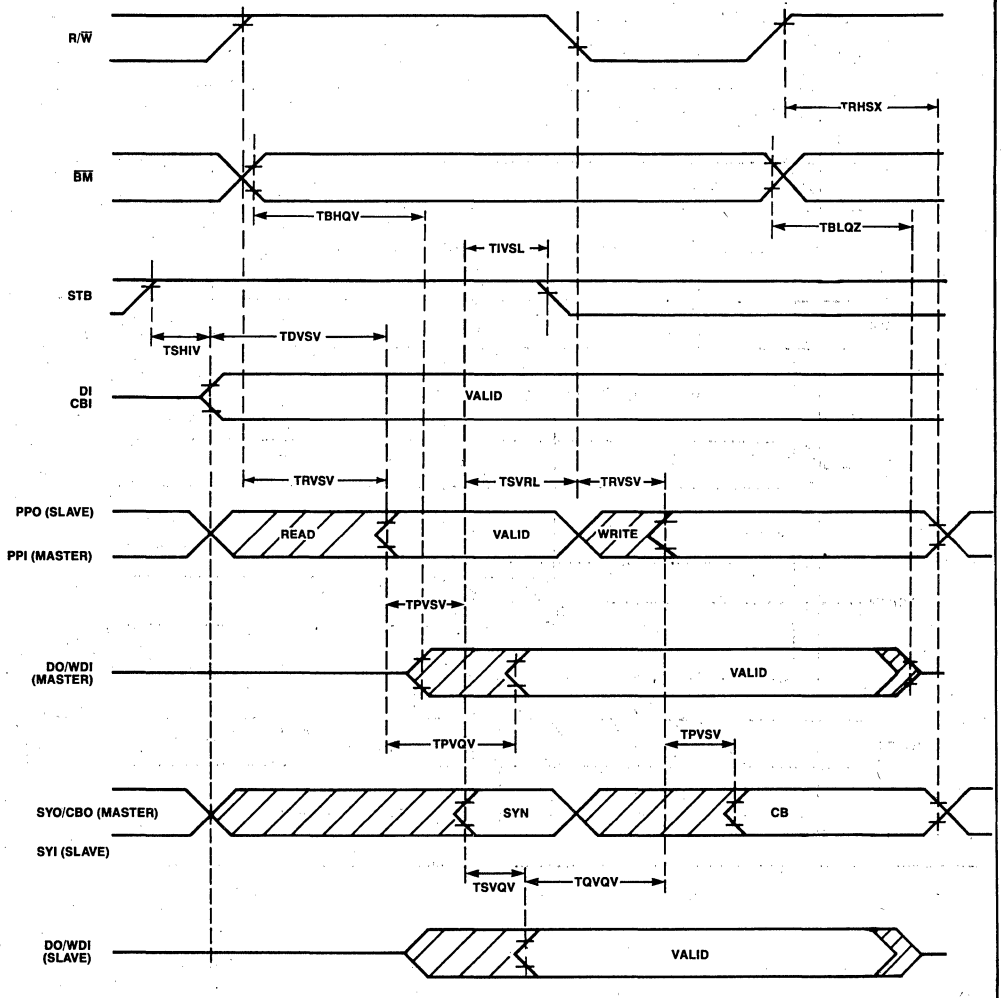
WAVEFORMS (Continued)

READ MODIFY WRITE



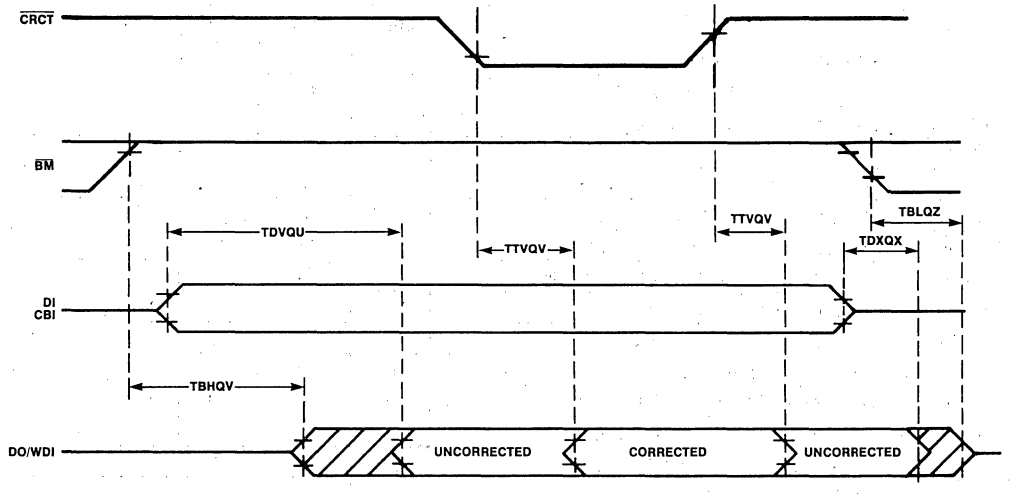
WAVEFORMS (Continued)

READ MODIFY WRITE—MASTER/SLAVE

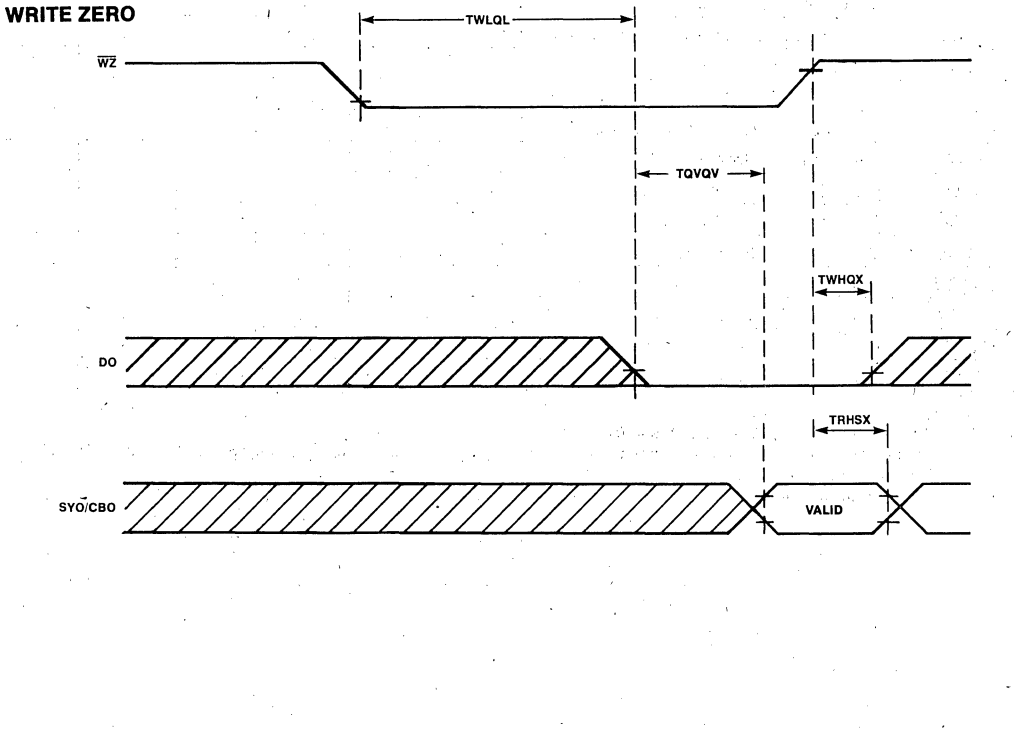


WAVEFORMS (Continued)

NON-CORRECTING READ



WRITE ZERO



A.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = +5V ± 10%, V_{SS} = 0V, R_L = 22Ω, C_L = 50 pF; all times are in nsec.)

Symbol	Parameter	8206		8206-2		Notes
		Min.	Max.	Min.	Max.	
TRHEV	$\overline{ERROR}$ Valid from R/$\overline{W}$↑		25		40	
TRHCV	CE Valid from R/$\overline{W}$↑ (Single 8206)		44		49	
TRHQV	Corrected Data Valid from R/$\overline{W}$↑		54		66	1
TRVSV	SYO/CBO/PPO Valid from R/$\overline{W}$		42		46	1
TDVEV	$\overline{ERROR}$ Valid from Data/Check Bits In		52		57	
TDVCV	CE Valid from Data/Check Bits In		70		76	
TDVQV	Corrected Data Valid from Data/Check Bits In		67		74	
TDVSV	SYO/PPO Valid from Data/Check Bits In		55		65	
TBHQV	Corrected Data Access Time		37		37	
TDXQX	Hold Time from Data/check Bits In	0		0		1
TBLQZ	Corrected Data Float Delay	0	28	0	28	1
TSHIV	STB High to Data/Check Bits In Valid	30		30		2
TIVSL	Data/Check Bits In to STB↓ Set-up	5		5		
TSLIX	Data/Check Bits In from STB↓ Hold	25		25		
TPVEV	$\overline{ERROR}$ Valid from Partial Parity In		30			3
TPVQV	Corrected Data (Master) from Partial Parity In		61			1,3
TPVSV	Syndrome/Check Bits Out from Partial Parity In		43			1,3
TSVQV	Corrected Data (Slave) Valid from Syndrome		51			3
TSVCV	CE Valid from Syndrome (Slave number 1)		48			3
TQVQV	Check Bits/Partial Parity Out from Write Data In		64		69	1
TRHSX	Check Bits/Partial Parity Out from R/W, $\overline{WZ}$ Hold	0		0		1
TRLSX	Syndrome Out from R/$\overline{W}$ Hold	0		0		
TQXQX	Hold Time from Write Data In	0		0		1
TSVRL	Syndrome Out to R/$\overline{W}$↓ Set-up	17				3
TDVRL	Data/Check Bits In to R/$\overline{W}$ Set-up	39		41		1
TDVQU	Uncorrected Data Out from Data In		32		38	
TTVQV	Corrected Data Out from $\overline{CRCT}$↓		30		33	
TWLQL	$\overline{WZ}$↓ to Zero Out		30		34	
TWHQX	Zero Out from $\overline{WZ}$↑ Hold	0		0		

NOTES:
1. A.C. Test Levels for CBO and DO are 2.4V and 0.8V.
2. T_{SHIV} is required to guarantee output delay timings: T_{DVEV}, T_{DVCV}, T_{DVQV}, T_{DVSV}. T_{SHIV} + T_{IVSL} guarantees a min STB pulse width of 35 ns (45 ns for the 8206-8).
3. Not required for 8/16 bit systems

PRELIMINARY

8207
ADVANCED DYNAMIC RAM CONTROLLER

- **Provides All Signals Necessary to Control 16K (2118), 64K (2164A) and 256K Dynamic RAMs**
- **Directly Addresses and Drives up to 2 Megabytes without External Drivers**
- **Supports Single and Dual-Port Configurations**
- **Automatic RAM Initialization in All Modes**
- **Four Programmable Refresh Modes**
- **Transparent Memory Scrubbing in ECC Mode**
- **Supports Intel iAPX 86, 88, 186, 188, and 286 Microprocessors**
- **Data Transfer and Advance Acknowledge Signals for Each Port**
- **Provides Signals to Directly Control the 8206 Error Detection and Correction Unit**
- **Supports Synchronous or Asynchronous Operation on Either Port**
- **+5 Volt Only HMOSII Technology for High Performance and Low Power**

The Intel 8207 Advanced Dynamic RAM Controller (ADRC) is a high-performance, systems-oriented, Dynamic RAM controller that is designed to easily interface 16K, 64K and 256K Dynamic RAMs to Intel and other microprocessor systems. A dual-port interface allows two different busses to independently access memory. When configured with an 8206 Error Detection and Correction Unit the 8207 supplies the necessary logic for designing large error-corrected memory arrays. This combination provides automatic memory initialization and transparent memory error scrubbing.

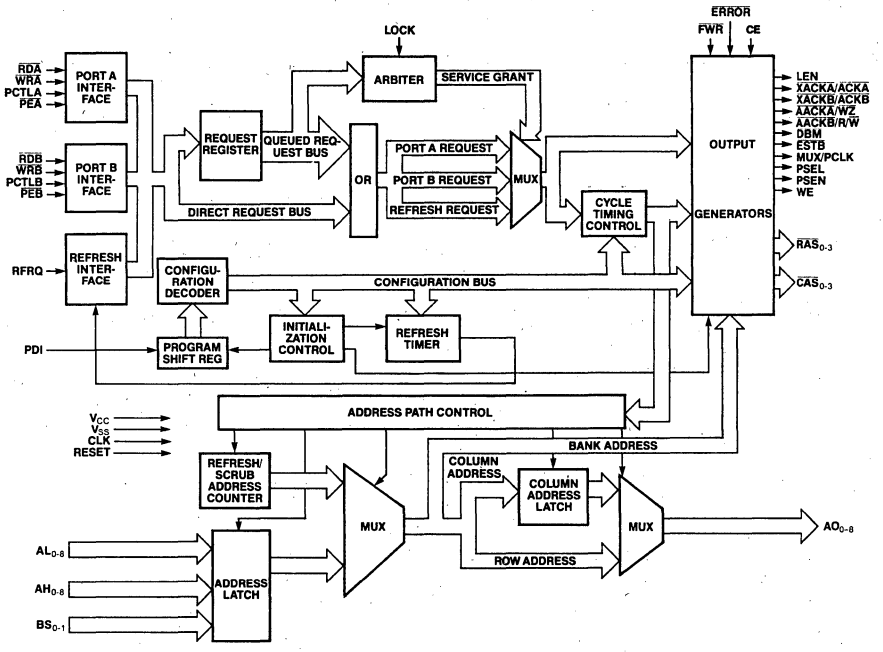


Figure 1. 8207 Block Diagram

JULY 1983
ORDER NUMBER: 210463-003

Table 1. Pin description

Symbol	Pin	Type	Name and Function
LEN	1	O	**ADDRESS LATCH ENABLE:** In two-port configurations, when Port A is running with iAPX 286 Status interface mode, this output replaces the ALE signal from the system bus controller of port A and generates an address latch enable signal which provides optimum setup and hold timing for the 8207. This signal is used in Fast Cycle operation only.
$\overline{\text{XACKA}}$/ $\overline{\text{ACKA}}$	2	O	**TRANSFER ACKNOWLEDGE PORTA/ACKNOWLEDGE PORTA:** In non-ECC mode, this pin is $\overline{\text{XACKA}}$ and inidcates that data on the bus is valid during a read cycle or that data may be removed from the bus during a write cycle for Port A. $\overline{\text{XACKA}}$ is a Multibus-compatible signal. In ECC mode, this pin is $\overline{\text{ACKA}}$ which can be configured, depending on the programming of the X program bit, as an $\overline{\text{XACK}}$ or $\overline{\text{AACK}}$ strobe. The SA programming bit determines whether the $\overline{\text{AACK}}$ will be an early $\overline{\text{EAACKA}}$ or a late $\overline{\text{LAACKA}}$ interface signal.
$\overline{\text{XACKB}}$/ $\overline{\text{ACKB}}$	3	O	**TRANSFER ACKNOWLEDGE PORT B/ACKNOWLEDGE PORT B:** In non-ECC mode, this pin is XACKB and indicates that data on the bus is valid during a read cycle or that data may be removed from the bus during a write cycle for Port B. $\overline{\text{XACKB}}$ is a Multibus-compatible signal. In ECC mode, this pin is $\overline{\text{ACKB}}$ which can be configured, depending on the programming of the X program bit, as an $\overline{\text{XACK}}$ or $\overline{\text{AACK}}$ strobe. The SB programming bit determines whether the $\overline{\text{AACK}}$ will be an early $\overline{\text{EAACKB}}$ or a late $\overline{\text{LAACKB}}$ interface signal.
$\overline{\text{AACKA}}$/ WZ	4	O	**ADVANCED ACKNOWLEDGE PORT A/WRITE ZERO:** In non-ECC mode, this pin is $\overline{\text{AACKA}}$ and indicates that the processor may continue processing and that data will be available when required. This signal is optimized for the system by programming the SA program bit for synchronous or asynchronous operation. In ECC mode, after a RESET, this signal will cause the 8206 to force the data to all zeros and generate the appropriate check bits.
$\overline{\text{AACKB}}$/ R/$\overline{\text{W}}$	5	O	**ADVANCED ACKNOWLEDGE PORT B/READ/WRITE:** In non-ECC mode, this pin is $\overline{\text{AACKB}}$ and indicates that the processor may continue processing and that data will be available when required. This signal is optimized for the system by programming the SB program bit for synchronous or asynchronous operation. In ECC mode, this signal causes the 8206 EDCU to latch the syndrome and error flags and generate check bits.
$\overline{\text{DBM}}$	6	O	**DISABLE BYTE MARKS:** This is an ECC control output signal indicating that a read or refresh cycle is occurring. This output forces the byte address decoding logic to enable all 8206 data output buffers. In ECC mode, this output is also asserted during memory initialization and the 8-cycle dynamic RAM wake-up exercise. In non-ECC systems this signal indicates that either a read, refresh or 8-cycle warm-up is in progress.
$\overline{\text{ESTB}}$	7	O	**ERROR STROBE:** In ECC mode, this strobe is activated when an error is detected and allows a negative-edge triggered flip-flop to latch the status of the 8206 EDCU CE for systems with error logging capabilities. $\overline{\text{ESTB}}$ will not be issued during refresh cycles.
LOCK	8	I	**LOCK:** This input instructs the 8207 to lock out the port not being serviced at the time LOCK was issued.
V_{CC}	9 43	I	**DRIVER POWER:** +5 Volts. Supplies V_{CC} for the output drivers. **LOGIC POWER:** +5 Volts. Supplies V_{CC} for the internal logic circuits.
CE	10	I	**CORRECTABLE ERROR:** This is an ECC input from the 8206 EDCU which instructs the 8207 whether a detected error is correctable or not. A high input indicates a correctable error. A low input inhibits the 8207 from activating WE to write the data back into RAM. This should be connected to the CE output of the 8206.
$\overline{\text{ERROR}}$	11	I	**ERROR:** This is an ECC input from the 8206 EDCU and instructs the 8207 that an error was detected. This pin should be connected to the ERROR output of the 8206.
MUX/ PCLK	12	O	**MULTIPLEXER CONTROL/PROGRAMMING CLOCK:** Immediately after a RESET this pin is used to clock serial programming data into the PDI pin. In normal two-port operation, this pin is used to select memory addresses from the appropriate port. When this signal is high, port A is selected and when it is low, port B is selected. This signal may change state before the completion of a RAM cycle, but the RAM address hold time is satisfied.
PSEL	13	O	**PORT SELECT:** This signal is used to select the appropriate port for data transfer. When this signal is high port A is selected and when it is low port B is selected.
PSEN	14	O	**PORT SELECT ENABLE:** This signal used in conjunction with PSEL provides contention-free port exchange on the data bus. When PSEN is low, port selection is allowed to change state.
WE	15	O	**WRITE ENABLE:** This signal provides the dynamic RAM array the write enable input for a write operation.

Table 1. Pin Description (Continued)

Symbol	Pin	Type	Name and Function
$\overline{FWR}$	16	I	**FULL WRITE:** This is an ECC input signal that instructs the 8207, in an ECC configuration, whether the present write cycle is normal RAM write (full write) or a RAM partial write (read-modify-write) cycle.
RESET	17	I	**RESET:** This signal causes all internal counters and state flip-flops to be reset and upon release of RESET, data appearing at the PDI pin is clocked in by the PCLK output. The states of the PDI, PCTLA, PCTLB and RFRQ pins are sampled by RESET going inactive and are used to program the 8207. An 8-cycle dynamic RAM warm-up is performed after clocking PDI bits into the 8207.
$\overline{CAS0}$ $\overline{CAS1}$ $\overline{CAS2}$ $\overline{CAS3}$	18 19 20 21	O O O O	**COLUMN ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the column address, present on the AO0-8 pins. These outputs are selected by the BS0 and BS1 as programmed by program bits RB0 and RB1. These outputs drive the dynamic RAM array directly and need no external drivers.
$\overline{RAS0}$ $\overline{RAS1}$ $\overline{RAS2}$ $\overline{RAS3}$	22 23 24 25	O O O O	**ROW ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the row address, present on the AO0-8 pins. These outputs are selected by the BS0 and BS1 as programmed by program bits RB0 and RB1. These outputs drive the dynamic RAM array directly and need no external drivers.
V_{SS}	26 60	I I	**DRIVER GROUND:** Provides a ground for the output drivers. **LOGIC GROUND:** Provides a ground for the remainder of the device.
AO0 AO1 AO2 AO3 AO4 AO5 AO6 AO7 AO8	35 34 33 32 31 30 29 28 27	O O O O O O O O O	**ADDRESS OUTPUTS:** These outputs are designed to provide the row and column addresses of the selected port to the dynamic RAM array. These outputs drive the dynamic RAM array directly and need no external drivers.
BS0 BS1	36 37	I I	**BANK SELECT:** These inputs are used to select one of four banks of the dynamic RAM array as defined by the program bits RB0 and RB1.
AL0 AL1 AL2 AL3 AL4 AL5 AL6 AL7 AL8	38 39 40 41 42 44 45 46 47	I I I I I I I I I	**ADDRESS LOW:** These lower-order address inputs are used to generate the row address for the internal address multiplexer.
AH0 AH1 AH2 AH3 AH4 AH5 AH6 AH7 AH8	48 49 50 51 52 53 54 55 56	I I I I I I I I I	**ADDRESS HIGH:** These higher-order address inputs are used to generate the column address for the internal address multiplexer.
PDI	57	I	**PROGRAM DATA INPUT:** This input programs the various user-selectable options in the 8207. The PCLK pin shifts programming data into the PDI input from optional external shift registers. This pin may be strapped high or low to a default ECC (PDI =Logic "I") or non-ECC (PDI = Logic "O") mode configuration.
RFRQ	58	I	**REFRESH REQUEST:** This input is sampled on the falling edge of RESET. If it is high at RESET, then the 8207 is programmed for internal refresh request or external refresh request with failsafe protection. If it is low at RESET, then the 8207 is programmed for external refresh without failsafe protection or burst refresh. Once programmed the RFRQ pin accepts signals to start an external refresh with failsafe protection or external refresh without failsafe protection or a burst refresh.

Table 1. Pin Description (Continued)

Symbol	Pin	Type	Name and Function
CLK	59	I	**CLOCK:** This input provides the basic timing for sequencing the internal logic.
$\overline{RDB}$	61	I	**READ FOR PORT B:** This pin is the read memory request command input for port B. This input also directly accepts the $\overline{S1}$ status line from Intel processors.
$\overline{WRB}$	62	I	**WRITE FOR PORT B:** This pin is the write memory request command input for port B. This input also directly accepts the $\overline{S0}$ status line from Intel processors.
$\overline{PEB}$	63	I	**PORT ENABLE FOR PORT B:** This pin serves to enable a RAM cycle request for port B. It is generally decoded from the port address.
PCTLB	64	I	**PORT CONTROL FOR PORT B:** This pin is sampled on the falling edge of RESET. It configures port B to accept command inputs or processor status inputs. If low after RESET, the 8207 is programmed to accept command or iAPX 286 status inputs or Multibus commands. If high after RESET, the 8207 is programmed to accept status inputs from iAPX 86 or iAPX 186 processors. The $\overline{S2}$ status line should be connected to this input if programmed to accept iAPX 86 or iAPX 186 status inputs. When programmed to accept commands or iAPX 286 status, it should be tied low or it may be used as a Multibus-compatible inhibit signal.
$\overline{RDA}$	65	I	**READ FOR PORT A:** This pin is the read memory request command input for port A. This input also directly accepts the $\overline{S1}$ status line from Intel processors.
$\overline{WRA}$	66	I	**WRITE FOR PORT A:** This pin is the write memory request command input for port A. This input also directly accepts the $\overline{S0}$ status line from Intel processors.
$\overline{PEA}$	67	I	**PORT ENABLE FOR PORT A:** This pin serves to enable a RAM cycle request for port A. It is generally decoded from the port address.
PCTLA	68	I	**PORT CONTROL FOR PORT A:** This pin is sampled on the falling edge of RESET. It configures port A to accept command inputs or processor status inputs. If low after RESET, the 8207 is programmed to accept command or iAPX 286 status inputs or Multibus commands. If high after RESET, the 8207 is programmed to accept status inputs from iAPX 86 or iAPX 186 processors. The $\overline{S2}$ status line should be connected to this input if programmed to accept iAPX 86 or iAPX 186 status inputs. When programmed to accept commands or iAPX 286 status, it should be tied low or it may be connected to INHIBIT when operating with Multibus.

GENERAL DESCRIPTION

The Intel 8207 Advanced Dynamic RAM Controller (ADRC) is a microcomputer peripheral device which provides the necessary signals to address, refresh and directly drive 16K, 64K and 256K dynamic RAMs. This controller also provides the necessary arbitration circuitry to support dual-port access of the dynamic RAM array.

The ADRC supports several microprocessor interface options including synchronous and asynchronous connection to iAPX 86, iAPX 88, iAPX 186, iAPX 188, iAPX 286 and Multibus.

This device may be used with the 8206 Error Detection and Correction Unit (EDCU). When used with the 8206, the 8207 is programmed in the Error Checking and Correction (ECC) mode. In this mode, the 8207 provides all the necessary control signals for the 8206 to perform memory initialization and transparent error scrubbing during refresh.

FUNCTIONAL DESCRIPTION

Processor Interface

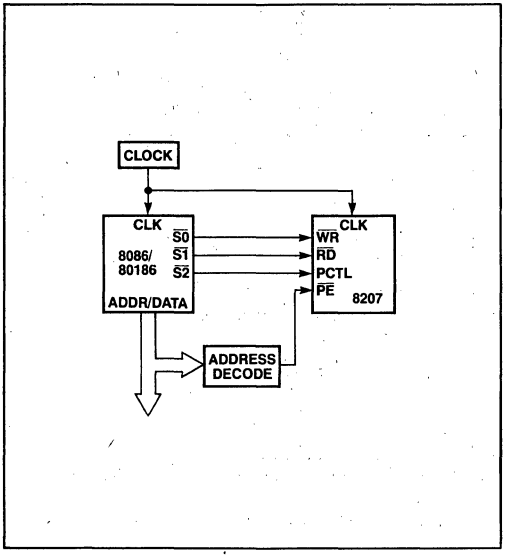
The 8207 has control circuitry for two ports each capable of supporting one of several possible bus structures. The ports are independently configurable allowing the dynamic RAM to serve as an interface between two different bus structures.

Each port of the 8207 may be programmed to run synchronous or asynchronous to the processor clock. (See Synchronous/Asynchronous Mode) The 8207 has been optimized to run synchronously with Intel's iAPX 86, iAPX 88, iAPX 186, iAPX 188 and iAPX 286. When the 8207 is programmed to run in asynchronous mode, the 8207 inserts the necessary synchronization circuitry for the $\overline{RD}$, $\overline{WR}$, $\overline{PE}$, and PCTL inputs.

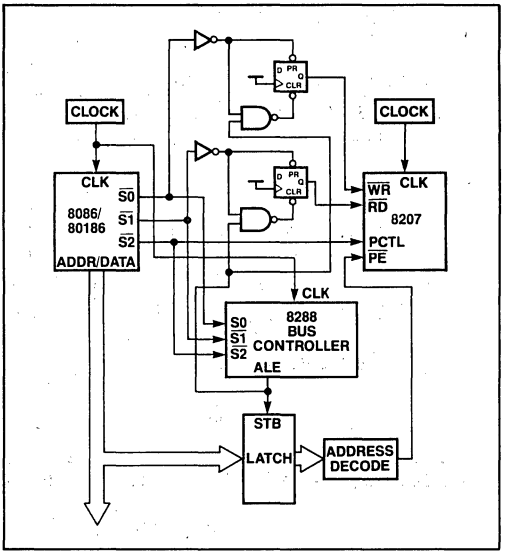
The 8207 achieves high performance (i.e. no wait states) by decoding the status lines directly from the iAPX 86, iAPX 88, iAPX 186, iAPX 188 and iAPX 286 processors. The 8207 can also be programmed to receive read or write Multibus commands or commands from a bus controller. (See Status/Command Mode)

The 8207 may be programmed to accept the clock of the iAPX 86, 88, 186, 188, or 286. The 8207 adjusts its internal timing to allow for the different clock frequencies of these microprocessors. (See Microprocessor Clock Frequency Option)

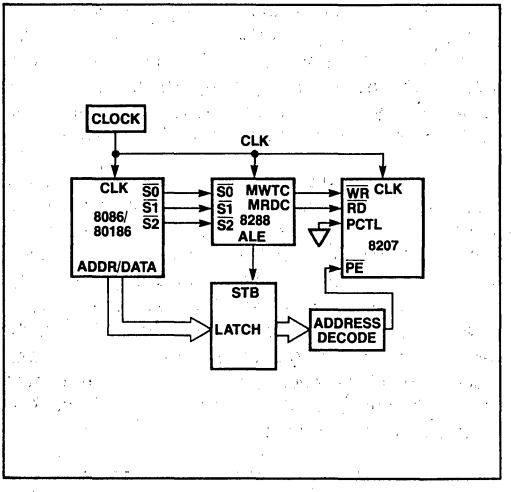
Figure 2 shows the different processor interfaces to the 8207 using the synchronous or asynchronous mode and status or command interface.



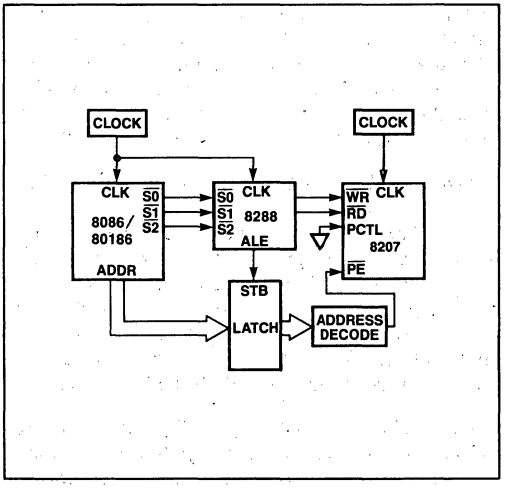
Slow-Cycle Synchronous-Status Interface



Slow-Cycle Asynchronous-Status Interface

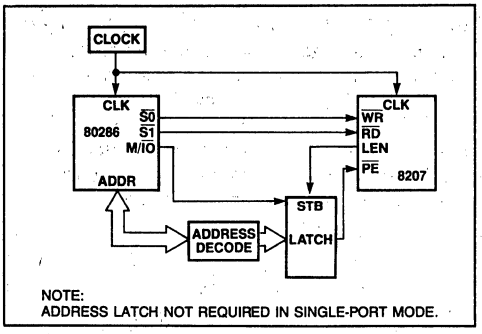


Slow-Cycle Synchronous-Command Interface

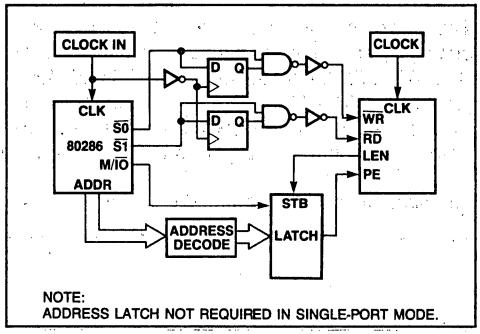


Slow-Cycle Asynchronous-Command Interface

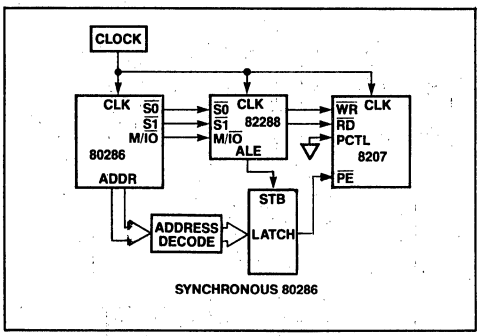
Figure 2A. Slow-cycle Port Interfaces Supported by the 8207



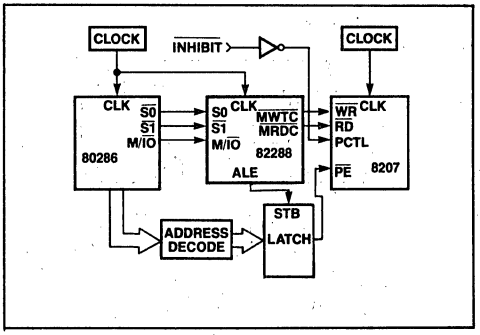
Fast-Cycle Synchronous-Status Interface



Fast-Cycle Asynchronous-Status Interface



Fast-Cycle Synchronous-Command Interface



Fast-Cycle Asynchronous-Command Interface

Figure 2B. Fast-cycle Port Interfaces Supported by the 8207

Single-Port Operation

The use of an address latch with the iAPX 286 status interface is not needed since the 8207 can internally latch the addresses with an internal signal similar in behavior to the LEN output. This operation is active only in single-port applications when the processor is interfaced to port A.

Dual-Port Operation

The 8207 provides for two-port operation. Two independent processors may access memory controlled by the 8207. The 8207 arbitrates between each of the processor requests and directs data to or from the appropriate port. Selection is done on a priority concept that reassigns priorities based upon past history. Processor requests are internally queued.

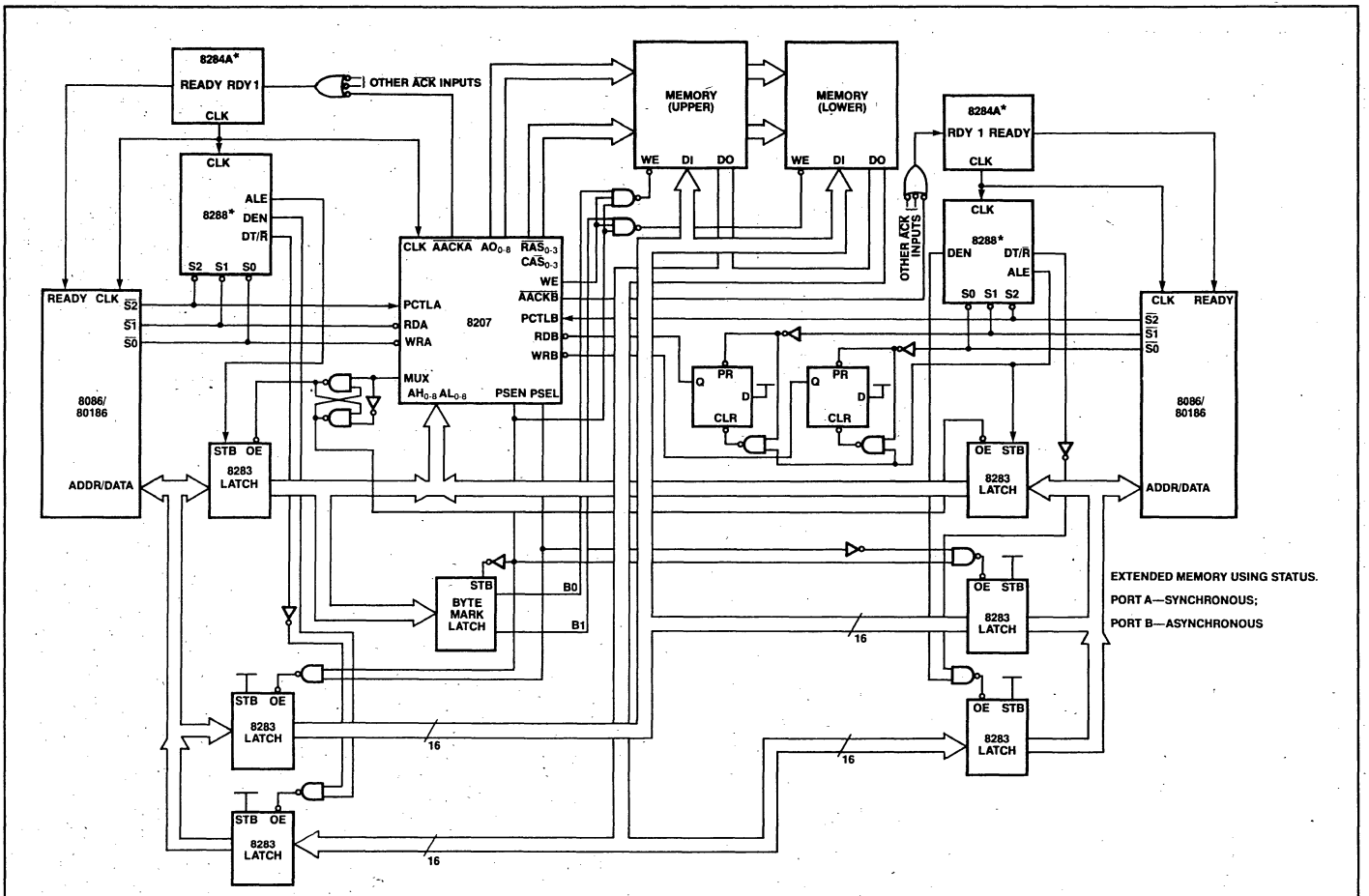
Figure 3 shows a dual-port configuration with two iAPX 86 systems interfacing to dynamic RAM. One of the processor systems is interfaced synchronously using the status interface and the other is interfaced asynchronously also using the status interface.

Dynamic RAM Interface

The 8207 is capable of addressing 16K, 64K and 256K dynamic RAMs. Figure 4 shows the connection of the processor address bus to the 8207 using the different RAMs. The 8207 directly supports the 2118 RAM family or any RAM with similar timing requirements and responses including the Intel 2164A RAM.

The 8207 divides memory into as many as four banks, each bank having its own Row ($\overline{RAS}$) and Column ($\overline{CAS}$) Address Strobe pair. This organization permits RAM cycle interleaving and permits error scrubbing during ECC refresh cycles. RAM cycle interleaving overlaps the start of the next RAM cycle with the RAM Precharge period of the previous cycle. Hiding the precharge period of one RAM cycle behind the data access period of the next RAM cycle optimizes memory bandwidth and is effective as long as successive RAM cycles occur in alternate banks.

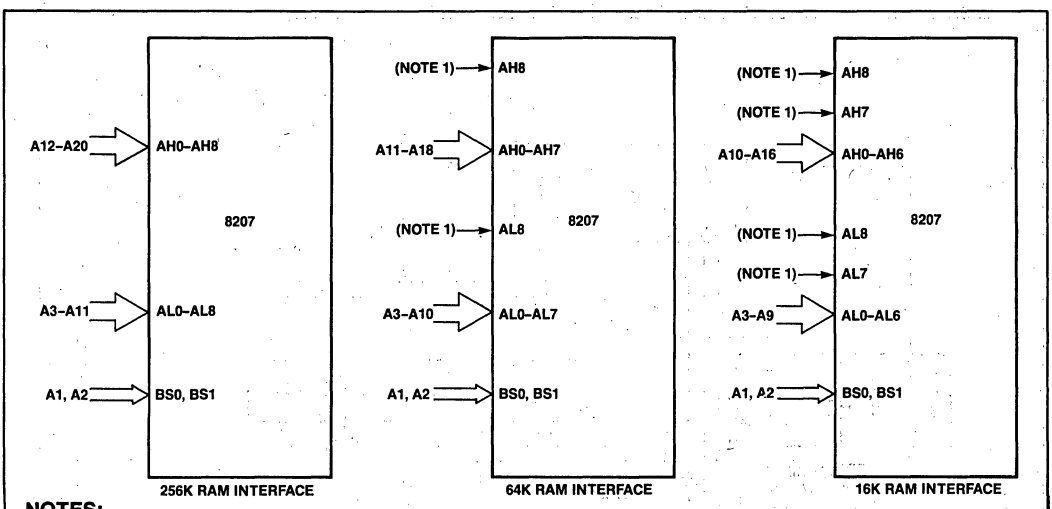
Successive data access to the same bank will cause the 8207 to wait for the precharge time of the previous RAM cycle.



NOTE:
*These components are not necessary when using the 80186 components. These functions are provided directly by the 80186.

Figure 3. 8086/80186 Dual Port System

210463-003



NOTES:
(1) Unassigned address input pins should be strapped high or low.
(2) A0 along with BHE are used to select a byte within a processor word.
(3) Low order address bits are used as bank select inputs so that consecutive memory access requests are to alternate banks allowing bank interleaving of memory cycles.

Figure 4. Processor Address Interface to the 8207 Using 16K, 64K, and 256K RAMS

If not all RAM banks are occupied, the 8207 reassigns the $\overline{RAS}$ and $\overline{CAS}$ strobes to allow using wider data words without increasing the loading on the $\overline{RAS}$ and $\overline{CAS}$ drivers. Table 2 shows the bank selection decoding and the word expansion, including $\overline{RAS}$ and $\overline{CAS}$ assignments. For example, if only two RAM banks are occupied, then two $\overline{RAS}$ and two $\overline{CAS}$ strobes are activated per bank. Program bits RB1 and RB0 are not used to check the bank select inputs BS1 and BS0. The system design must protect from accesses to "illegal", non-existent banks of memory, by deactivating the PEA, PEB inputs when addressing an illegal bank.

The 8207 can interface to fast (e.g., 2118-10) or slow (e.g., 2118-15) RAMs. The 8207 adjusts and optimizes internal timings for either the fast or slow RAMs as programmed. (See RAM Speed Option).

Memory Initialization

After programming, the 8207 performs eight RAM "warm-up" cycles to prepare the dynamic RAM for proper device operation. During "warm-up" some RAM parameters, such as tRAH, tASC, may not be met. This causes no harm to the dynamic RAM array. If configured for operation with error correction, the 8207 and 8206 EDCU will proceed to initialize all of memory (memory is written with zeros with corresponding check bits).

Table 2.
Bank Selection Decoding and Word Expansion

Program Bits		Bank Input		
RB1	RB0	BS1	BSØ	RAS/CAS Pair Allocation
0	0	0	0	$RAS_{0\text{-}3}$, $CAS_{0\text{-}3}$ to Bank 0
0	0	0	1	Illegal
0	0	1	0	Illegal
0	0	1	1	Illegal
0	1	0	0	$RAS_{0,1}$, $CAS_{0,1}$ to Bank 0
0	1	0	1	$RAS_{2,3}$, $CAS_{2,3}$ to Bank 1
0	1	1	0	Illegal
0	1	1	1	Illegal
1	0	0	0	RAS_0, CAS_0 to Bank 0
1	0	0	1	RAS_1, CAS_1 to Bank 1
1	0	1	0	RAS_2, CAS_2 to Bank 2
1	0	1	1	Illegal
1	1	0	0	RAS_0, CAS_0 to Bank 0
1	1	0	1	RAS_1, CAS_1 to Bank 1
1	1	1	0	RAS_2, CAS_2 to Bank 2
1	1	1	1	RAS_3, CAS_3 to Bank 3

Because the time to initialize memory is fairly long, the 8207 may be programmed to skip initialization in ECC mode. The time required to initialize all of memory is dependent on the clock cycle time to the 8207 and can be calculated by the following equation:

eq.1 $T_{INIT} = (2^{23})\ T_{CLCL}$

if $T_{CLCL} = 125$ ns then $T_{INIT} \approx 1$ sec.

8206 ECC Interface

For operation with Error Checking and Correction (ECC), the 8207 adjusts its internal timing and changes some pin functions to optimize performance and provide a clean dual-port memory interface between the 8206 EDCU and memory. The 8207 directly supports a master-only (16-bit word plus 6 check bits) system. Under extended operation and reduced clock frequency, the 8207 will support any ECC master-slave configuration up to 80 data bits, which is the maximum set by the 8206 EDCU. (See Extend Option)

Correctable errors detected during memory read cycles are corrected immediately and then written back into memory.

In a synchronous bus environment, ECC system performance has been optimized to enhance processor throughput, while in an asynchronous bus environment (the Multibus), ECC performance has been optimized to get valid data onto the bus as quickly as possible. Performance optimization, processor throughput or quick data access may be selected via the Transfer Acknowledge Option.

The main difference between the two ECC implementations is that, when optimized for processor throughput, RAM data is always corrected and an advanced transfer acknowledge is issued at a point when, by knowing the processor characteristics, data is guaranteed to be valid by the time the processor needs it.

When optimized for quick data access, (valid for Multibus) the 8206 is configured in the uncorrecting mode where the delay associated with error correction circuitry is transparent, and a transfer acknowledge is issued as soon as valid data is known to exist. If the $\overline{\text{ERROR}}$ flag is activated, then the transfer acknowledge is delayed until after the 8207 has instructed the 8206 to correct the data and the corrected data becomes available on the bus. Figure 5 illustrates a dual-port ECC system.

Figure 6 illustrates the interface required to drive the $\overline{\text{CRCT}}$ pin of the 8206, in the case that one port (PORT A) receives an advanced acknowledge (not Multibus-compatible), while the other port (PORT B) receives $\overline{\text{XACK}}$ (which is Multibus-compatible).

Error Scrubbing

The 8207/8206 performs error correction during refresh cycles (error scrubbing). Since the 8207 must refresh RAM, performing error scrubbing during refresh allows it to be accomplished without additional performance penalties.

Upon detection of a correctable error during refresh, the RAM refresh cycle is lengthened slightly to permit the 8206 to correct the error and for the corrected word to be rewritten into memory. Uncorrectable errors detected during scrubbing are ignored.

Refresh

The 8207 provides an internal refresh interval counter and a refresh address counter to allow the 8207 to refresh memory. The 8207 will refresh 128 rows every 2 milliseconds or 256 rows every 4 milliseconds, which allows all RAM refresh options to be supported. In addition, there exists the ability to refresh 256 row address locations every 2 milliseconds via the Refresh Period programming option.

The 8207 may be programmed for any of four different refresh options: Internal refresh only, External refresh with failsafe protection, External refresh without failsafe protection, Burst Refresh mode, or no refresh. (See Refresh Options)

It is possible to decrease the refresh time interval by 10%, 20% or 30%. This option allows the 8207 to compensate for reduced clock frequencies. Note that an additional 5% interval shortening is built-in in all refresh interval options to compensate for clock variations and non-immediate response to the internally generated refresh request. (See Refresh Period Options)

External Refresh Requests after RESET

External refresh requests are not recognized by the 8207 until after it is finished programming and preparing memory for access. Memory preparation includes 8 RAM cycles to prepare and ensure proper

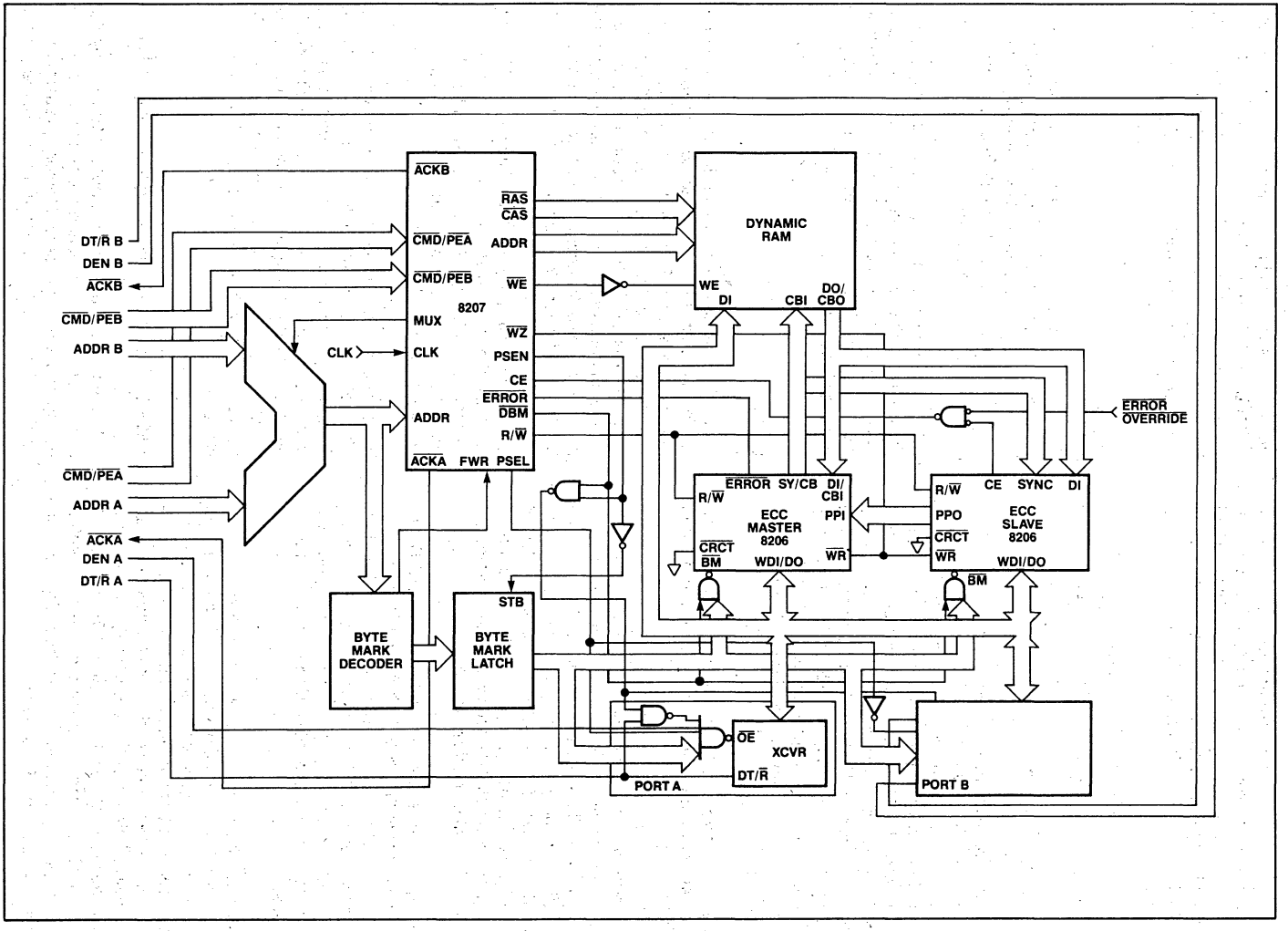


Figure 5. Two-Port ECC Implementation Using the 8207 and the 8206

210463-003

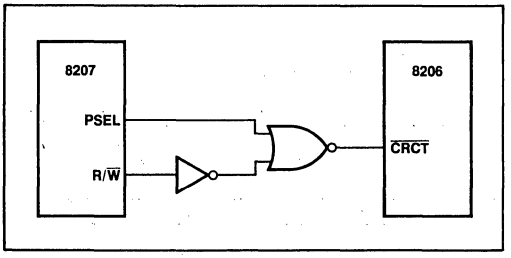


Figure 6. Interface to 8206 $\overline{\text{CRCT}}$ Input When Port A Receives $\overline{\text{AACK}}$ and Port B Receives $\overline{\text{XACK}}$

dynamic RAM operation, and memory initialization if error correction is used. Many dynamic RAMs require this warm-up period for proper operation. The time it takes for the 8207 to recognize a request is shown below.

eq. 2 Non-ECC Systems: $T_{RESP} = T_{PROG} + T_{PREP}$

eq. 3 where: $T_{PROG} = (66)(T_{CLCL})$ which is programming time

eq. 4 $T_{PREP} = (8)(32)(T_{CLCL})$ which is the RAM warm-up time

if $T_{CLCL} = 125$ ns then $T_{RESP} \approx 41$ us

eq. 5 ECC Systems: $T_{RESP} = T_{PROG} + T_{PREP} + T_{INIT}$

if $T_{CLCL} = 125$ ns then $T_{RESP} \approx 1$ sec

RESET

RESET is an asynchronous input, the falling edge of which is used by the 8207 to directly sample to logic levels of the PCTLA, PCTLB, RFRQ, and PDI inputs. The internally synchronized falling edge of RESET is used to begin programming operations (shifting in the contents of the external shift register into the PDI input).

Until programming is complete the 8207 registers but does not respond to command or status inputs. A simple means of preventing commands or status from occurring during this period is to differentiate the system reset pulse to obtain a smaller reset pulse for the 8207. The total time of the reset pulse and the 8207 programming time must be less than the time before the first command in systems that alter the default port synchronization programming bits (default is Port A synchronous, Port B asynchronous). Differentiated reset is unnecessary when the default port synchronization programming is used.

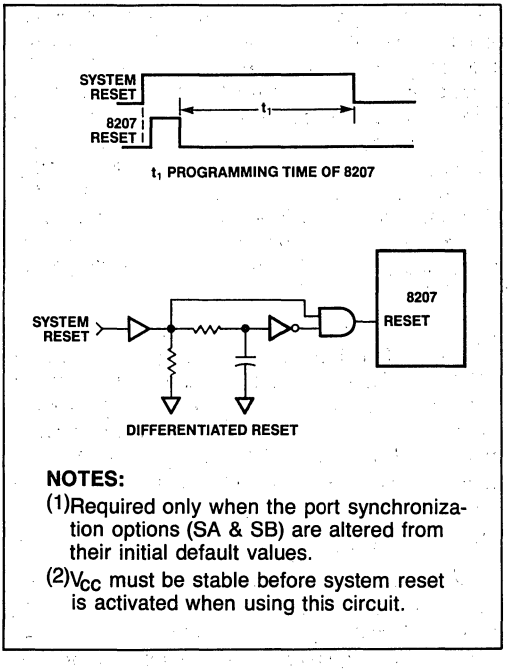
The differentiated reset pulse would be shorter than the system reset pulse by at least the programming period required by the 8207. The differentiated reset pulse first resets the 8207, and system reset would reset the rest of the system. While the rest of the system is still in reset, the 8207 completes its programming. Figure 7 illustrates a circuit to accomplish this task.

Within four clocks after RESET goes active, all the 8207 outputs will go high, except for PSEN, WE, and AO0-2, which will go low.

OPERATIONAL DESCRIPTION

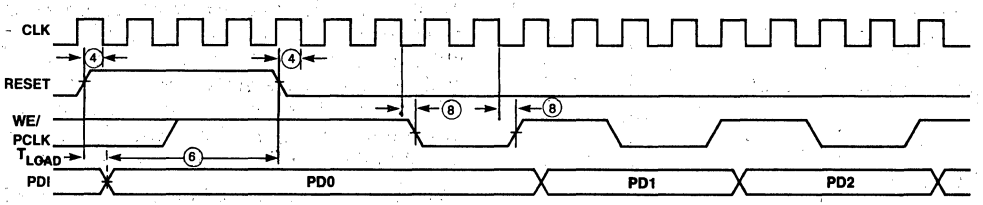
Programming the 8207

The 8207 is programmed after reset. On the falling edge of RESET, the logic states of several input pins are latched internally. The falling edge of RESET actually performs the latching, which means that the logic levels on these inputs must be stable prior to that time. The inputs whose logic levels are latched at the end of reset are the PCTLA, PCTLB, REFRQ, and PDI pins. Figure 8 shows the necessary timing for programming the 8207.



NOTES:
(1) Required only when the port synchronization options (SA & SB) are altered from their initial default values.
(2) V_{CC} must be stable before system reset is activated when using this circuit.

Figure 7. 8207 Differentiated Reset Circuit



NOTES:
TRTVCL — Reset is an asynchronous input, if reset occurs before T_1, then it is guaranteed to be recognized.
TPGVCL — Minimum PDI valid time prior to reset going low.
TCLPC — MUX/PCLK delay.
TLOAD — Asynchronous load data propagation delay.

Figure 8. Timing Illustrating External Shift Register Requirements for Programming the 8207

Status/Command Mode

The two processor ports of the 8207 are configured by the states of the PCTLA and PCTLB pins. Which interface is selected depends on the state of the individual port's PCTL pin at the end of reset. If PCTL is high at the end of the reset, the 8086 Status interface is selected; if it is low, then the Command interface is selected.

The status lines of the 80286 are similar in code and timing to the Multibus command lines, while the status code and timing of the 8076 and 8088 are identical to those of the 80186 and 80188 (ignoring the differences in clock duty cycle). Thus there exists two interface configurations, one for the 80286 status or Multibus memory commands, which is called the Command interface, and one for 8086, 8088, 80186 or 80188 status, called the 8086 Status interface. The Command interface can also directly interface to the command lines of the bus controllers for the 8086, 8088, 80186 and the 80286.

The 8086 Status interface allows direct decoding of the status of the iAPX 86, iAPX 88, iAPX 186 and the iAPX 188. Table 3 shows how the status lines are decoded. While in the Command mode the iAPX 286 status can be directly decoded. Microprocessor bus controller read or write commands or Multibus commands can also be directed to the 8207 when in Command mode.

Refresh Options

Immediately after system reset, the state of the REFRQ input pin is examined. If REFRQ is high, the 8207 provides the user with the choice between self-refresh or user-generated refresh with failsafe protection. Failsafe protection guarantees that if the

Table 3A. Status Coding of 8086, 80186 and 80286

Status Code			Function	
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	8086/80186	80286
0	0	0	INTERRUPT	INTERRUPT
0	0	1	I/O READ	I/O READ
0	1	0	I/O WRITE	I/O WRITE
0	1	1	HALT	IDLE
1	0	0	INSTRUCTION FETCH	HALT
1	0	1	MEMORY READ	MEMORY READ
1	1	0	MEMORY WRITE	MEMORY WRITE
1	1	1	IDLE	IDLE

Table 3B. 8207 Response

8207 Command			Function	
PCTL	$\overline{RD}$	$\overline{WR}$	8086/80186 Status Interface	80286/Status or Command Interface
0	0	0	IGNORE	IGNORE
0	0	1	IGNORE	READ
0	1	0	IGNORE	WRITE
0	1	1	IGNORE	IGNORE
1	0	0	READ	IGNORE
1	0	1	READ	INHIBIT
1	1	0	WRITE	INHIBIT
1	1	1	IGNORE	IGNORE

user does not come back with another refresh request before the internal refresh interval counter times out, a refresh request will be automatically generated. If the REFRQ pin is low immediately after a reset, then the user has the choice of a single external refresh cycle without failsafe, burst refresh or no refresh.

Internal Refresh Only

For the 8207 to generate internal refresh requests, it is necessary only to strap the REFRQ input pin high.

External Refresh with Failsafe

To allow user-generated refresh requests with failsafe protection, it is necessary to hold the REFRQ input high until after reset. Thereafter, a low-to-high transition on this input causes a refresh request to be generated and the internal refresh interval counter to be reset. A high-to-low transition has no effect on the 8207. A refresh request is not recognized until a previous request has been serviced.

External Refresh without Failsafe

To generate single external refresh requests without failsafe protection, it is necessary to hold REFRQ low until after reset. Thereafter, bringing REFRQ high for one clock period causes a refresh request to be generated. A refresh request is not recognized until a previous request has been serviced.

Burst Refresh

Burst refresh is implemented through the same procedure as a single external refresh without failsafe (i.e., REFRQ is kept low until after reset). Thereafter, bringing REFRQ high for at least two clock periods causes a burst of up to 128 row address locations to be refreshed.

In ECC-configured systems, 128 locations are scrubbed. Any refresh request is not recognized until a previous request has been serviced (i.e., burst completed).

No Refresh

It is necessary to hold REFRQ low until after reset. This is the same as programming External Refresh without Failsafe. No refresh is accomplished by keeping REFRQ low.

Option Program Data Word

The program data word consists of 16 program data bits, PD0—PD15. If the first program data bit PD0 is set to logic 1, the 8207 is configured to support ECC. If it is logic 0, the 8207 is configured to support a non-ECC system. The remaining bits, PD1—PD15, may then be programmed to optimize a selected configuration. Figures 9 and 10 show the Program words for non-ECC and ECC operation.

Using an External Shift Register

The 8207 may be configured to use an external shift register with asynchronous load capability such as a 74LS165. The reset pulse serves to parallel load the shift register and the 8207 supplies the clocking signal to shift the data in. Figure 11 shows a sample circuit diagram of an external shift register circuit.

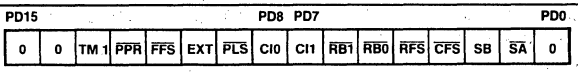
Serial data is shifted into the 8207 via the PDI pin (57), and clock is provided by the MUX/PCLK pin (12), which generates a total of 16 clock pulses. After programming is complete, data appearing at the input of the PDI pin is ignored. MUX/PCLK is a dual-function pin. During programming, it serves to clock the external shift register, and after programming is completed, it reverts to a MUX control pin. As the pin changes state to select different port addresses, it continues to clock the shift register. This does not present a problem because data at the PDI pin is ignored after programming. Figure 8 illustrates the timing requirements of the shift register circuitry.

ECC Mode (ECC Program Bit)

The state of PDI (Program Data In) pin at reset determines whether the system is an ECC or non-ECC configuration. It is used internally by the 8207 to begin configuring timing circuits, even before programming is completely finished. The 8207 then begins programming the rest of the options.

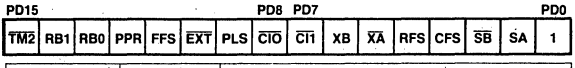
Default Programming Options

After reset, the 8207 serially shifts in a program data word via the PDI pin. This pin may be strapped either high or low, or connected to an external shift register. Strapping PDI high causes the 8207 to default to a particular system configuration with error correction, and strapping it low causes the 8207 to default to a particular system configuration without error correction. Table 4 shows the default configurations.



PROGRAM DATA BIT	NAME	POLARITY/FUNCTION
PD0	ECC	ECC=0 FOR NON-ECC MODE
PD1	$\overline{SA}$	$\overline{SA}$=0 PORT A IS SYNCHRONOUS $\overline{SA}$=1 PORT A IS ASYNCHRONOUS
PD2	SB	SB=0 PORT B IS ASYNCHRONOUS SB=1 PORT B IS SYNCHRONOUS
PD3	$\overline{CFS}$	$\overline{CFS}$=0 FAST-CYCLE iAPX 286 MODE $\overline{CFS}$=1 SLOW-CYCLE iAPX 86 MODE
PD4	$\overline{RFS}$	$\overline{RFS}$=0 FAST RAM $\overline{RFS}$=1 SLOW RAM
PD5 PD6	$\overline{RB0}$ $\overline{RB1}$	RAM BANK OCCUPANCY SEE TABLE 2
PD7 PD8	CI1 CI0	COUNT INTERVAL BIT 1; SEE TABLE 6 COUNT INTERVAL BIT 0; SEE TABLE 6
PD9	$\overline{PLS}$	$\overline{PLS}$=0 LONG REFRESH PERIOD $\overline{PLS}$=1 SHORT REFRESH PERIOD
PD10	EXT	EXT=0 NOT EXTENDED EXT=1 EXTENDED
PD11	$\overline{FFS}$	$\overline{FFS}$=0 FAST CPU FREQUENCY $\overline{FFS}$=1 SLOW CPU FREQUENCY
PD12	$\overline{PPR}$	$\overline{PPR}$=0 MOST RECENTLY USED PORT PRIORITY $\overline{PPR}$=1 PORT A PREFERRED PRIORITY
PD13	TM1	TM1=0 TEST MODE 1 OFF TM1=1 TEST MODE 1 ENABLED
PD14	0	RESERVED MUST BE ZERO
PD15	0	RESERVED MUST BE ZERO

Figure 9. Non-ECC Mode Program Data Word



PROGRAM DATA BIT	NAME	POLARITY/FUNCTION
PD0	ECC	ECC=1 ECC MODE
PD1	SA	SA=0 PORT A ASYNCHRONOUS SA=1 PORT A SYNCHRONOUS
PD2	$\overline{SB}$	$\overline{SB}$=0 PORT B SYNCHRONOUS $\overline{SB}$=1 PORT B ASYNCHRONOUS
PD3	CFS	CFS=0 SLOW-CYCLE iAPX 86 MODE CFS=1 FAST-CYCLE iAPX 286 MODE
PD4	RFS	RFS=0 SLOW RAM RFS=1 FAST RAM
PD5	$\overline{XA}$	$\overline{XA}$=0 MULTIBUS-COMPATIBLE ACKA $\overline{XA}$=1 ADVANCED ACKA NOT MULTIBUS-COMPATIBLE
PD6	XB	XB=0 ADVANCED ACKB NOT MULTIBUS COMPATIBLE XB=1 MULTIBUS-COMPATIBLE ACKB
PD7	$\overline{CI1}$	COUNT INTERVAL BIT 1; SEE TABLE 6
PD8	$\overline{CI0}$	COUNT INTERVAL BIT 0; SEE TABLE 6
PD9	PLS	PLS=0 SHORT REFRESH PERIOD PLS=1 LONG REFRESH PERIOD
PD10	$\overline{EXT}$	$\overline{EXT}$=0 MASTER AND SLAVE EDCU $\overline{EXT}$=1 MASTER EDCU ONLY
PD11	FFS	FFS=0 SLOW CPU FREQUENCY FFS=1 FAST CPU FREQUENCY
PD12	PPR	PPR=0 PORT A PREFERRED PRIORITY PPR=1 MOST RECENTLY USED PORT PRIORITY
PD13 PD14	RB0 RB1	RAM BANK OCCUPANCY SEE TABLE 2
PD15	$\overline{TM2}$	$\overline{TM2}$=0 TEST MODE 2 ENABLED $\overline{TM2}$=1 TEST MODE 2 OFF

Figure 10. ECC Mode Program Data Word

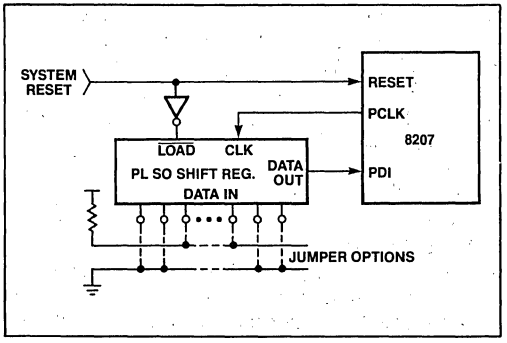


Figure 11. External Shift Register Interface

Table 4A.
Default Non-ECC Programming, PDI Pin (57) Tied to Ground.

Port A is Synchronous ($\overline{EAACKA}$ and $\overline{XACKA}$)
Port B is Asynchronous ($\overline{LAACKB}$ and $\overline{XACKB}$)
Fast-cycle Processor Interface (10 or 16 MHz)
Fast RAM
Refresh Interval uses 236 clocks
128 Row refresh in 2 ms; 256 Row refresh in 4 ms
Fast Processor Clock Frequency (16 MHz)
"Most Recently Used" Priority Scheme
4 RAM banks occupied

Table 4B.
Default ECC Programming, PDI Pin (57) Tied to V_{CC}.

Port A is Synchronous
Port B is Asynchronous
Fast-cycle Processor Interface (10 or 16 MHz)
Fast RAM
Port A has $\overline{EAACKA}$ strobe (non-multibus)
Port B has $\overline{XACKB}$ strobe (multibus)
Refresh interval uses 236 clocks
128 Row refresh in 2 ms; 256 Row refresh in 4 ms
Master EDCU only (16-bit system)
Fast Processor Clock Frequency (16 MHz)
"Most Recently Used" Priority Scheme
4 RAM banks occupied

If further system flexibility is needed, one or two external shift registers can be used to tailor the 8207 to its operating environment.

Synchronous/Asynchronous Mode (SA and SB Program Bits)

Each port of the 8207 may be independently configured to accept synchronous or asynchronous port commands ($\overline{RD}$, $\overline{WR}$, PCTL) and Port Enable ($\overline{PE}$) via the program bits SA and SB. The state of the SA and SB programming bits determine whether their associated ports are synchronous or asynchronous.

While a port may be configured with either the Status or Command interface in the synchronous mode, certain restrictions exist in the asynchronous mode. An asynchronous Command interface using the control lines of the Multibus is supported, and an asynchronous 8086 interface using the control lines of the 8086 is supported, with the use of TTL gates as illustrated in Figure 2. In the 8086 case, the TTL gates are needed to guarantee that status does not appear at the 8207's inputs too much before address, so that a cycle would start before address was valid.

Microprocessor Clock Frequency Option (CFS and FFS Program Bits)

The 8207 can be programmed to interface with slow-cycle microprocessors like the 8086, 8088, 80188 and 80186 or fast-cycle microprocessors like the 80286. The CFS bit configures the microprocessor interface to accept slow or fast cycle signals from either microprocessor group.

This option is used to select the speed of the microprocessor clock. Table 5 shows the various microprocessor clock frequency options that can be programmed.

Table 5.
Microprocessor Clock Frequency Options

Program Bits		Processor	Clock Frequency
CFS	FFS		
0	0	iAPX 86, 88, 186, 188	5 MHz
0	1	iAPX 86, 88, 186, 188	8 MHz
1	0	iAPX 286	10 MHz
1	1	iAPX 286	16 MHz

The external clock frequency must be programmed so that the failsafe refresh repetition circuitry can adjust its internal timing accordingly to produce a refresh request as programmed.

RAM Speed Option (RFS Program Bit)

The RAM Speed programming option determines whether RAM timing will be optimized for a fast or slow RAM. Whether a RAM is fast or slow is measured relative to the 2118-10 (Fast) or the 2118-15 (Slow) RAM specifications.

Refresh Period Options (CI0, CI1, and PLS Program Bits)

The 8207 refreshes with either 128 rows every 2 milliseconds or 256 rows every 4 milliseconds. This translates to one refresh cycle being executed approximately once every 15.6 microseconds. This rate can be changed to 256 rows every 2 milliseconds or a refresh approximately once every 7.8 microseconds via the Period Long/Short, program bit PLS, programming option. The 7.8 microsecond refresh request rate is intended for those RAMs, 64K and above, which may require a faster refresh rate.

In addition to PLS program option, two other programming bits for refresh exist: Count Interval 0 (CI0) and Count Interval 1 (CI1). These two programming bits allow the rate at which refresh requests are generated to be increased in order to permit refresh requests to be generated close to the same 15.6 or 7.8 microsecond period when the 8207 is operating at reduced frequencies. The interval between refreshes is decreased by 0%, 10%, 20%, or 30% as a function of how the count interval bits are programmed. A 5% guardband is built-in to allow for any clock frequency variations. Table 6 shows the refresh period options available.

The numbers tabulated under Count Interval represent the number of clock periods between internal refresh requests. The percentages in parentheses represent the decrease in the interval between refresh requests. Note that all intervals have a built-in 5% (approximately) safety factor to compensate for minor clock frequency deviations and non-immediate response to internal refresh requests.

Extend Option (EXT Program Bit)

The Extend option lengthens the memory cycle to allow longer access time which may be required by the system. Extend alters the RAM timing to compensate for increased loading on the Row and Column Address Strobes, and in the multiplexed Address Out lines.

Port Priority Option and Arbitration (PPR Program Bit)

The 8207 has to internally arbitrate among three ports: Port A, Port B and Port C—the refresh port. Port C is an internal port dedicated to servicing refresh requests, whether they are generated internally by the refresh inverval counter, or externally by the user. Two arbitration approaches are available via

Table 6. Refresh Count Interval Table

Freq. (MHz)	Ref. Period (μS)	CFS	PLS	FFS	Count Interval CI1, CI0 (8207 Clock Periods)			
					00 (0%)	01 (10%)	10 (20%)	11 (30%)
16	15.6	1	1	1	236	212	188	164
	7.8	1	0	1	118	106	94	82
10	15.6	1	1	0	148	132	116	100
	7.8	1	0	0	74	66	58	50
8	15.6	0	1	1	118	106	94	82
	7.8	0	0	1	59	53	47	41
5	15.6	0	1	0	74	66	58	50
	7.8	0	0	0	37	33	29	25

the Port Priority programming option, program bit PPR. PPR determines whether the most recently used port will remain selected (PPR = 1) or whether Port A will be favored or preferred over Port B (PPR = 0).

A port is selected if the arbiter has given the selected port direct access to the timing generators. The front-end logic, which includes the arbiter, is designed to operate in parallel with the selected port. Thus a request on the selected port is serviced immediately. In contrast, an unselected port only has access to the timing generators through the front-end logic. Before a RAM cycle can start for an unselected port, that port must first become selected (i.e., the MUX output now gates that port's address into the 8207 in the case of Port A or B). Also, in order to allow its address to stabilize, a newly selected port's first RAM cycle is started by the front-end logic. Therefore, the selected port has direct access to the timing generators. What all this means is that a request on a selected port is started immediately, while a request on an unselected port is started two to three clock periods after the request, assuming that the other two ports are idle. Under normal operating conditions, this arbitration time is hidden behind the RAM cycle of the selected port so that as soon as the present cycle is over a new cycle is started. Table 7 lists the arbitration rules for both options.

Port LOCK Function

The LOCK function provides each port with the ability to obtain uninterrupted access to a critical region of memory and, thereby, to guarantee that the opposite port cannot "sneak in" and read from or write to the critical region prematurely.

Only one LOCK pin is present and is multiplexed between the two ports as follows: when MUX is high, the 8207 treats the LOCK input as originating at PORT A, while when MUX is low, the 8207 treats LOCK as originating at PORT B. When the 8207 recognizes a LOCK, the MUX output will remain pointed to the locking port until LOCK is deactivated. Refresh is not affected by LOCK and can occur during a locked memory cycle.

Table 7. The Arbitration Rules for the Most Recently Used Port Priority and for Port A Priority Options Are As follows:

1.	If only one port requests service, then that port—if not already selected—becomes selected.
2a.	When no service requests are pending, the last selected processor port (Port A or B) will remain selected. (Most Recently Used Port Priority Option)
2b.	When no service requests are pending, Port A is selected whether it requests service or not. (Port A Priority Option)
3.	During reset initialization only Port C, the refresh port, is selected.
4.	If no processor requests are pending after reset initialization, Port A will be selected.
5a.	If Ports A and B simultaneously(*) request service while Port C is being serviced, then the next port to be selected is the one which was not selected prior to servicing Port C. (Most Recently Used Port Priority Option)
5b.	If Ports A and B simultaneously(*) request service while Port C is selected, then the next port to be selected is Port A. (Port A Priority Option)
6.	If a port simultaneously requests service with the currently selected port, service is granted to the selected port.
7.	The MUX output remains in its last state whenever Port C is selected.
8.	If Port C and either Port A or Port B (or both) simultaneously request service, then service is granted to the requester whose port is already selected. If the selected port is not requesting service, then service is granted to Port C.
9.	If during the servicing of one port, the other port requests service before or simultaneously with the refresh port, the refresh port is selected. A new port is not selected before the presently selected port is deactivated.
10.	Activating LOCK will mask off service requests from Port B if the MUX output is high, or from Port A if the MUX output is low.
* By "simultaneous" it is meant that two or more requests are valid at the clock edge at which the internal arbiter samples them.	

Dual-Port Considerations

For both ports to be operated synchronously, several conditions must be met. The processors must be the same type (Fast or Slow Cycle) as defined by Table 8 and they must have synchronized clocks. Also when processor types are mixed, even though the clocks may be in phase, one frequency may be twice that of the other. So to run both ports synchronous using the status interface, the processors must have related timings (both phase and frequency). If these conditions cannot be met, then one port must run synchronous and the other asynchronous.

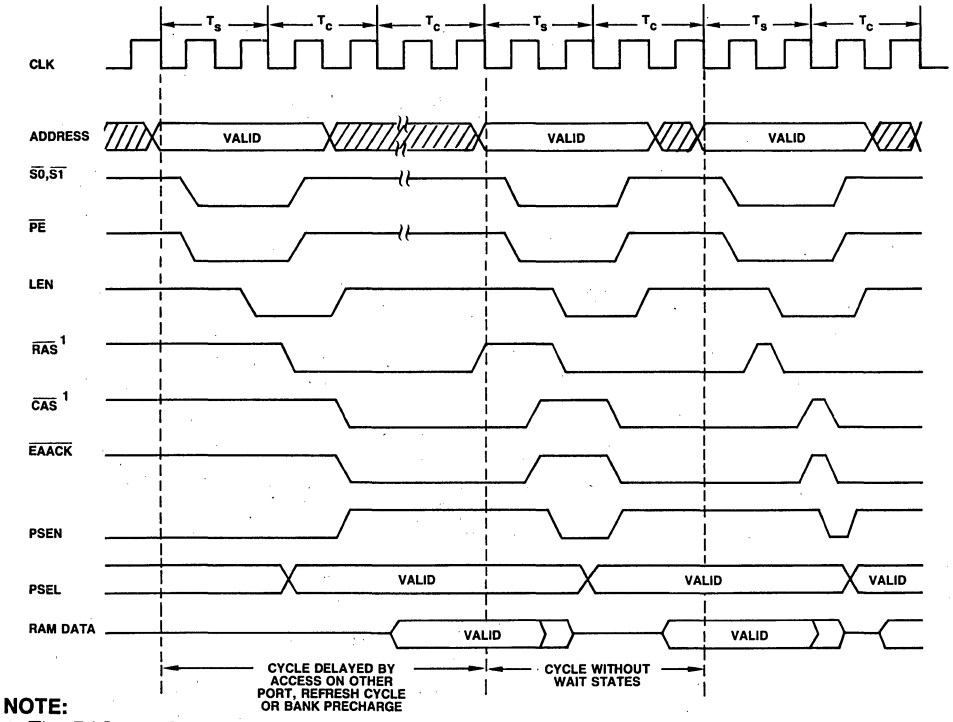
Figure 3 illustrates an example of dual-port operation using the processors in the slow cycle group. Note the use of cross-coupled NAND gates at the MUX output for minimizing contention between the two latches, and the use of flip flops on the status lines of the asynchronous processor for delaying the status and thereby guaranteeing RAS will not be issued, even in the worst case, until address is valid.

Processor Timing

In order to run without wait states, $\overline{AACK}$ must be used and connected to the $\overline{SRDY}$ input of the appropriate bus controller. $\overline{AACK}$ is issued relative to a point within the RAM cycle and has no fixed relationship to the processor's request The timing is such, however, that the processor will run without wait states, barring refresh cycles, bank precharge, and RAM accesses from the other port. In non-ECC fast cycle, fast RAM, non-extended configurations (80286), $\overline{AACK}$ is issued on the next falling edge of the clock after the edge that issues RAS. In non-ECC, slow cycle, non-extended, or extended with fast RAM cycle configurations (8086, 80188, 80186), $\overline{AACK}$ is issued on the same clock cycle that issues RAS. Figure 14 illustrates the timing relationship between $\overline{AACK}$, the RAM cycle, and the processor cycle for several different situations.

Port Enable ($\overline{PE}$ setup time requirements depend on whether the associated port is configured for synchronous or asynchronous fast or slow cycle operation. In a synchronous fast cycle configuration, PE is required to be setup to the same clock edge as the status or commands. If PE is true (low), a RAM cycle is started; if not, the cycle is aborted. The memory cycle will only begin when both valid signals ($\overline{PE}$ and $\overline{RD}$ or WR) are recognized at a particular clock edge. In aynchronous operation, PE is required to be setup to the same clock edge as the internally synchronized status or commands. Externally, this allows the internal synchronization delay to be added to the status (or command)-to-$\overline{PE}$ delay time, thus allowing for more external decode time that is available in synchronous operation.

The minimum synchronization delay is the additional amount that $\overline{PE}$ must be held valid. If $\overline{PE}$ is not held valid for the maximum synchronization delay time, it is possible that $\overline{PE}$ will go invalid prior to the status or command being synchronized. In such a case the 8207 aborts the cycle. If a memory cycle intended for the 8207 is aborted, then no acknowledge ($\overline{AACK}$ or $\overline{XACK}$) is issued and the processor locks up in endless wait states. Figure 15 illustrates the status (command) timing requirements for synchronous and asynchronous systems. Figures 16 and 17 show a more detailed hook-up of the 8207 to the 8086 and the 80286, respectively.



NOTE:
1. The RAS and CAS shown in figure are different banks being accessed.

Figure 14. iAPX 286/8207 Synchronous-Status Timing Programmed in non-ECC Mode, C0 Configuration (Read Cycle)

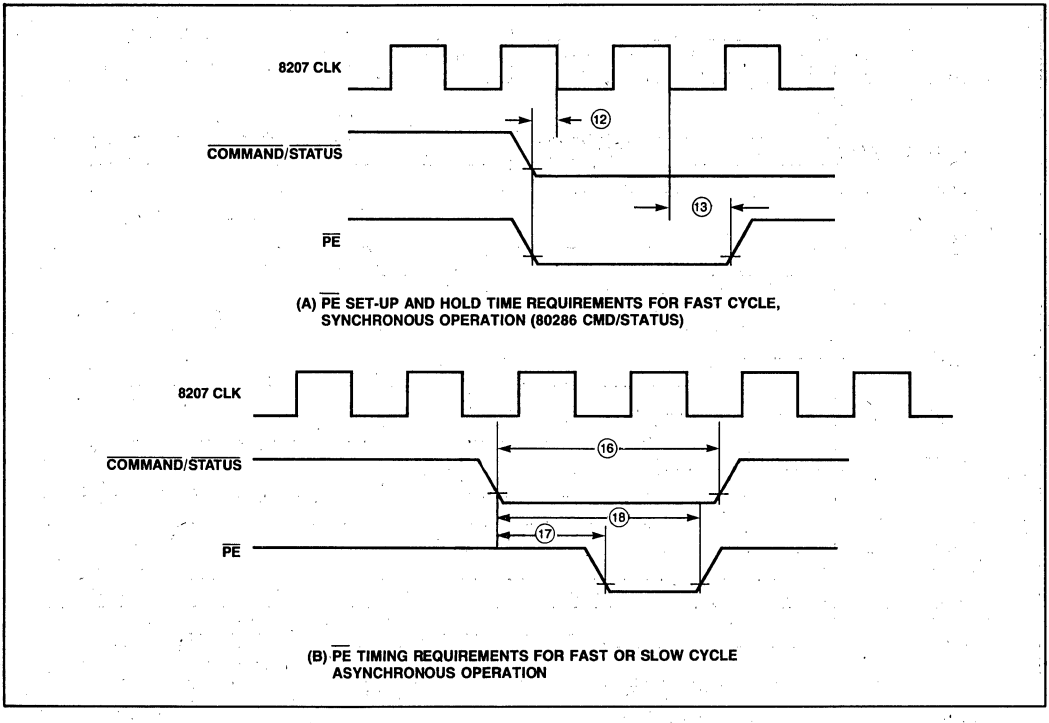


Figure 15.

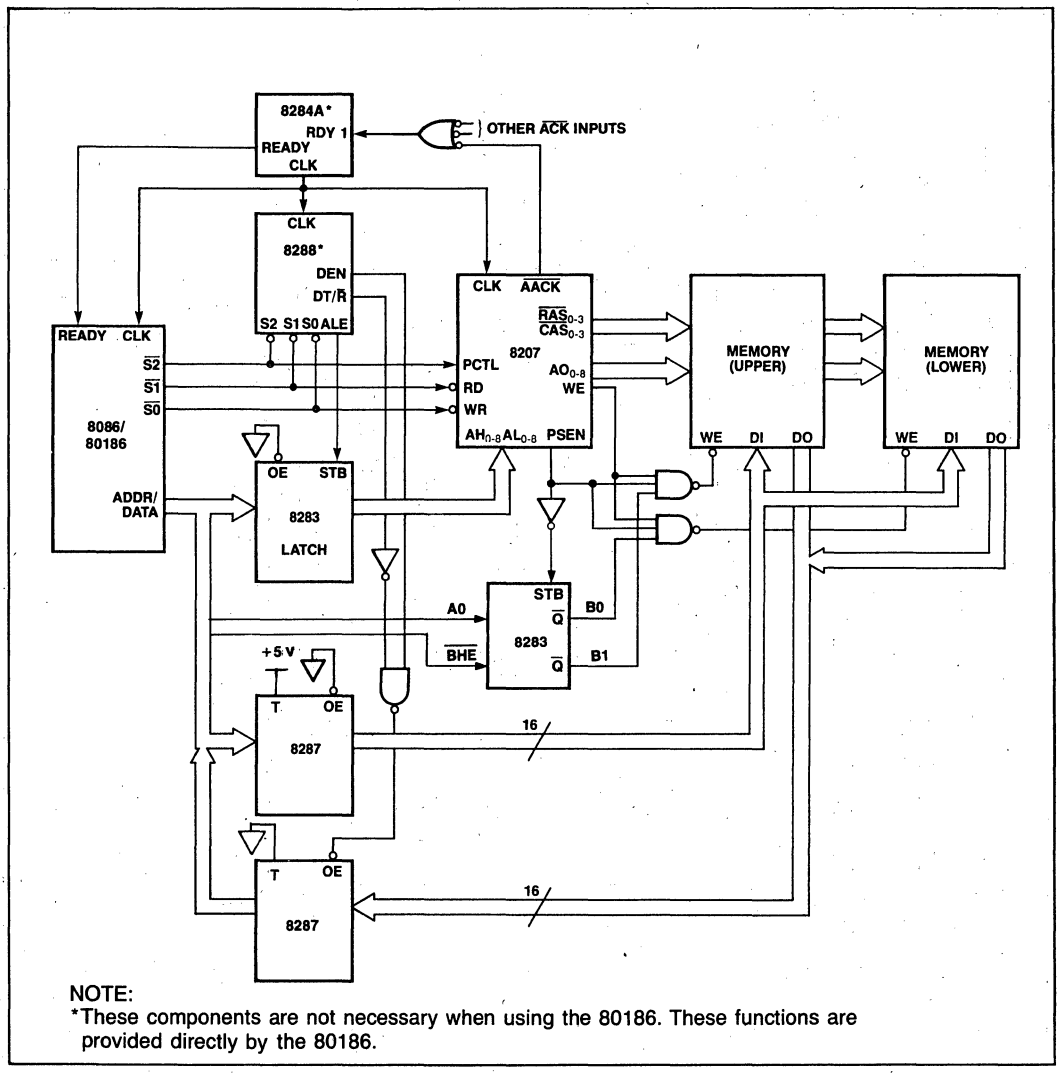
Memory Acknowledge (AACK, XACK)

In system configurations without error correction, two memory acknowledge signals per port are supplied by the 8207. They are the Advanced Acknowledge strobe ($\overline{AACK}$) and the Transfer Acknowledge strobe ($\overline{XACK}$). The CFS programming bit determines for which processor $\overline{AACKA}$ and $\overline{AACKB}$ are optimized, either 80286 (CFS = 1) or 8086/186 (CFS = 0), while the SA and SB programming bits optimize $\overline{AACK}$ for synchronous operation ("early" $\overline{AACK}$) or asynchronous operation ("late" $\overline{AACK}$).

Both the early and late $\overline{AACK}$ strobes are three clocks long for CFS = 1 and two clocks long for CFS = 0. The $\overline{XACK}$ strobe is asserted when data is valid (for reads) or when data may be removed (for writes) and meets the Multibus requirements. $\overline{XACK}$ is removed asynchronously by the command going inactive. Since in asynchronous operation the 8207 removes read data before late $\overline{AACK}$ or $\overline{XACK}$ is recognized by the CPU, the user must provide for data latching in the system until the CPU reads the data. In synchronous operation, data latching is unnecessary since the 8207 will not remove data until the CPU has read it.

In ECC-based systems there is one memory acknowledge ($\overline{XACK}$ or $\overline{AACK}$) per port and a programming bit associated with each acknowledge. If the X programming bit is high, the strobe is configured as $\overline{XACK}$, while if the bit is low, the strobe is configured as $\overline{AACK}$. As in non-ECC, the SA and SB programming bits determine whether the $\overline{AACK}$ strobe is early or late ($\overline{EAACK}$ or $\overline{LAACK}$).

Data will always be valid a fixed time after the occurrence of the advanced acknowledge. Table 9 summarizes the various transfer acknowledge options.



NOTE:
*These components are not necessary when using the 80186. These functions are provided directly by the 80186.

Figure 16. 8086/80186, 8207 Single Port Non-ECC Synchronous Systems

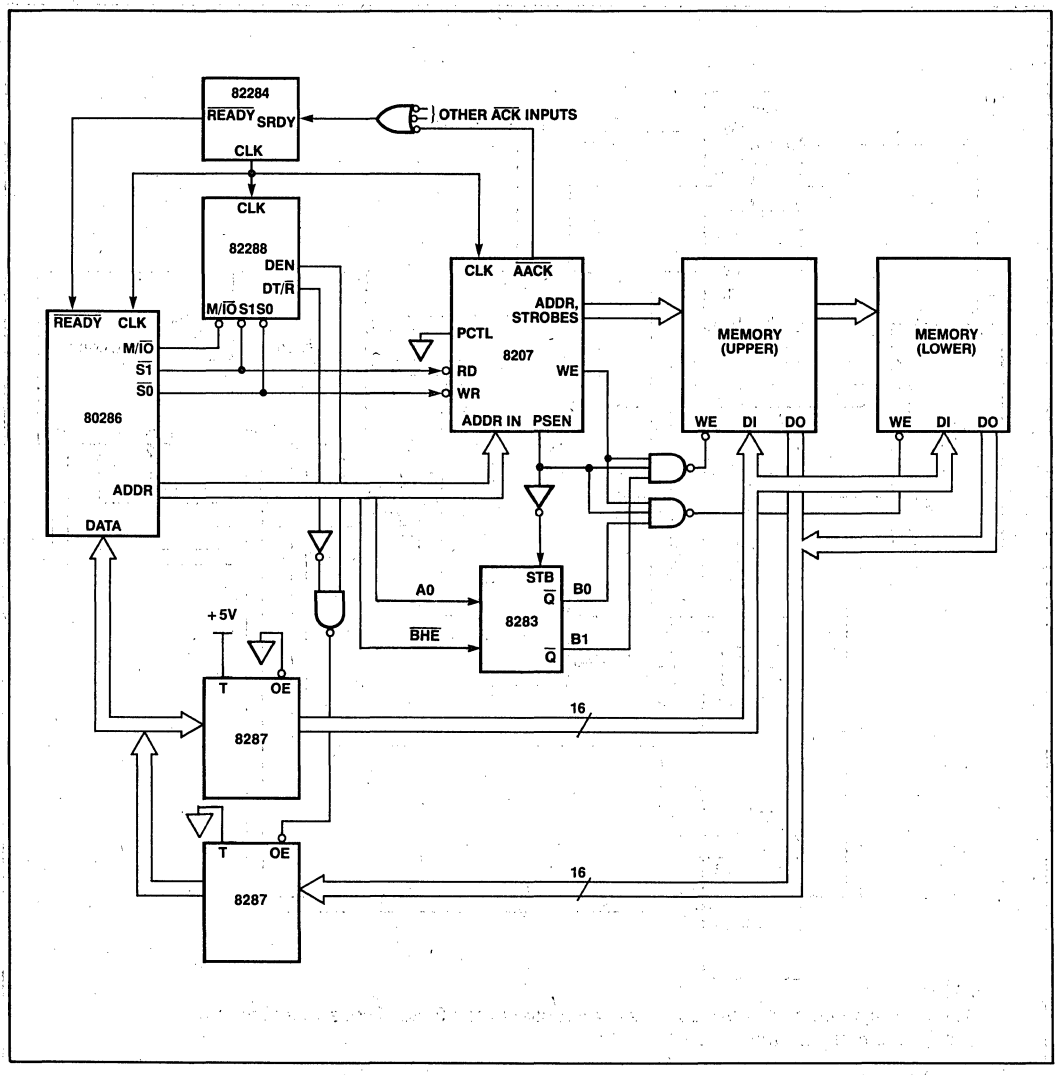


Figure 17. 80286 Hook-up to 8207 Non-ECC Synchronous System–Single Port.

Table 8. Processor Interface/Acknowledge Summary

CYCLE	PROCESSOR	REQUEST TYPE	SYNC/ASYNC INTERFACE	ACKNOWLEDGE TYPE
FAST CYCLE CFS=1	80286	STATUS	SYNC	EAACK
	80286	STATUS	ASYNC	LAACK
	80286	COMMAND	SYNC	EAACK
	80286	COMMAND	ASYNC	LAACK
	8086/80186	STATUS	ASYNC	LAACK
	8086/80186	COMMAND	ASYNC	LAACK
	MULTIBUS	COMMAND	ASYNC	XACK
SLOW CYCLE CFS=0	8086/80186	STATUS	SYNC	EAACK
	8086/80186	STATUS	ASYNC	LAACK
	8086/80186	COMMAND	SYNC	EAACK
	8086/80186	COMMAND	ASYNC	LAACK
	MULTIBUS	COMMAND	ASYNC	XACK

Table 9. Memory Acknowledge Option Summary

	Synchronous	Asynchronous	XACK
Fast Cycle	AACK Optimized for Local 80286	AACK Optimized for Remote 80286	Multibus Compatible
Slow Cycle	AACK Optimized for Local 8086/186	AACK Optimized for Remote 8086/186	Multibus Compatible

Test Modes

Two special test modes exist in the 8207 to facilitate testing. Test Mode 1 (non-ECC mode) splits the refresh address counter into two separate counters and Test Mode 2 (ECC mode) presets the refresh address counter to a value slightly less than rollover.

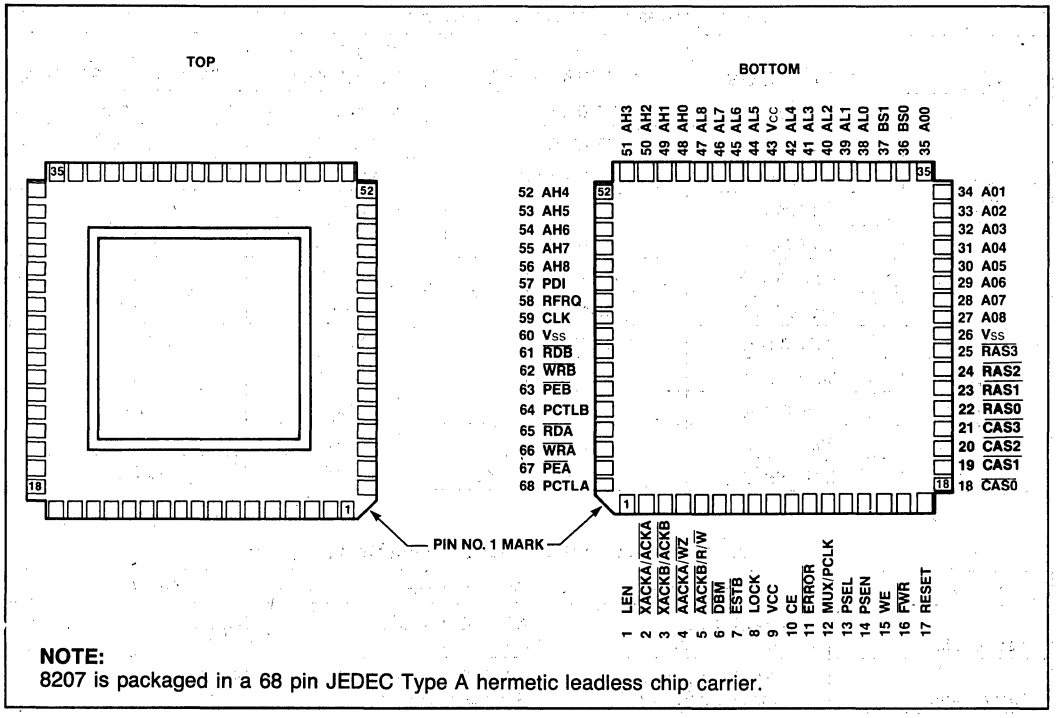
Test Mode 1 splits the address counter into two, and increments both counters simultaneously with each refresh address update. By generating external refresh requests, the tester is able to check for proper operation of both counters. Once proper individual counter operation has been established, the 8207 must be returned to normal mode and a second test performed to check that the carry from the first counter increments the second counter. The outputs of the counters are presented-on the address out bus with the same timing as the row and column addresses of a normal scrubbing operation. During Test Mode 1, memory initialization is inhibited, since the 8207, by definition, is in non-ECC mode.

Test Mode 2 sets the internal refresh counter to a value slightly less than rollover. During functional testing other than that covered in Test Mode 1, the 8207 will normally be set in Test Mode 2. Test Mode 2 eliminates memory initialization in ECC mode. This allows quick examination of the circuitry which brings the 8207 out of memory initialization and into normal operation.

General System Considerations

The RAS_{0-3}, CAS_{0-3}, AO_{0-8}, output buffers were designed to directly drive the heavy capacitive loads associated with dynamic RAM arrays. To keep the RAM driver outputs from ringing excessively in the system environment and causing noise in other output pins it is necessary to match the output impedance of the RAM output buffers with the RAM array by using series resistors and to add series resistors to other control outputs for noise reduction if necessary. Each application may have different impedance characteristics and may require different series resistance values. The series resistance values should be determined for each application. In non-ECC systems unused ECC input pins should be tied high or low to improve noise immunity.

The 8207 is packaged in a 68-pin, leadless JEDEC type A hermetic chip carrier.



NOTE:
8207 is packaged in a 68 pin JEDEC Type A hermetic leadless chip carrier.

Figure 19. 8207 Pinout Diagram

ABSOLUTE MAXIMUM RATINGS

Ambient Temperature
Under Bias ..... −0° C to +70° C
Storage Temperature ..... −65°C to +150°C
Voltage On Any Pin With
Respect to Ground ..... −.5V to +7V
Power Dissipation ..... 2.5 Watts

NOTICE: Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS

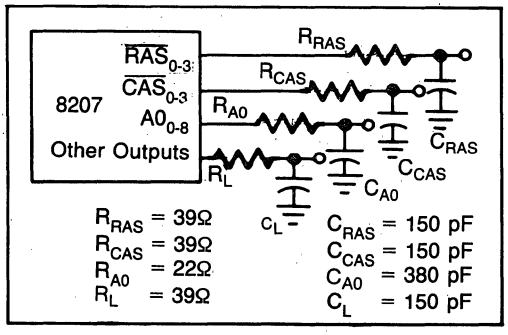
(T_A = 0°C to +70°C, V_{CC} = 5.0V ± 10% for 8207; ± 5% for 8207-2 and 8207-5, V_{SS} = GND)

Symbol	Parameter	Min.	Max.	Units	Comments
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	Note 1
V_{OH}	Output High Voltage	2.4		V	Note 1
V_{ROL}	RAM Output Low Voltage		0.45	V	Note 1
V_{ROH}	RAM Output High Voltage	2.6		V	Note 1
I_{CC}	Supply Current		400	mA	T_A = 25°C
I_{LI}	Input Leakage Current		+10	μA	0V ≤ V_{IN} ≤ V_{CC}
V_{CL}	Clock Input Low Voltage	−0.5	+0.6	V	
V_{CH}	Clock Input High Voltage	3.8	V_{CC} + 0.5	V	
C_{IN}	Input Capacitance		20	pF	fc = 1 MHz

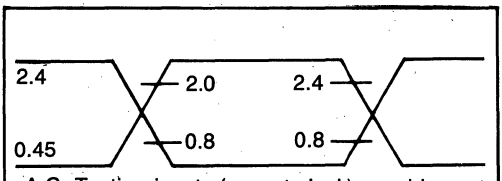
NOTES:

I_{OL} = 8 mA and I_{OH} = -0.2 mA (Typically I_{OL} = 10 mA and I_{OH} = -0.88 mA)

A.C. Testing Load Circuit



A.C. Testing Input, Output Waveform



A.C. Testing inputs (except clock) are driven at 2.4V for a logic "1" and 0.45V for a logic "0" (clock is driven at 4.0V and 0.45V for logic "1" and "0" respectively). Timing measurements are made at 2.0V, 2.4V for logic "1" and 0.8 V for logic "0".

A.C. CHARACTERISTICS

(T_A = 0°C to 70°C; V_{CC} = +5V ± 10% for 8207; ± 5% for 8207-2, 8207-5; V_{SS} = 0V)
Measurements made with respect to RAS_{0-3}, CAS_{0-3}, AO_{0-8}, are at 2.4V and 0.8V. All other pins are measured at 2.0V and 0.8V. All times in nsec unless otherwise indicated. Testing done with specified test load.

CLOCK AND PROGRAMMING

Ref.	Symbol	Parameter		8207 & 8207-2 Min.	8207 & 8207-2 Max.	8207-5 Min.	8207-5 Max.	Units	Notes
—	tF	Clock Fall Time			10		10	ns	35
—	tR	Clock Rise Time			10		10	ns	35
1	TCLCL	Clock Period	8207	62.5	250			ns	1
			8207	125	500			ns	2
			8207-2	125	500	200	500	ns	3
2	TCL	Clock Low Time	8207	15	230			ns	1
			8207	TCLCL/2-12				ns	2
			8207-2	TCLCL/2-12		TCLCL/2-12		ns	3
3	TCH	Clock High Time	8207	20	235			ns	1
			8207	TCLCL/3-3				ns	2
			8207-2	TCLCL/3-3		TCLCL/3-3		ns	3
4	TRTVCL	Reset to CLK↓ Setup		40		65		ns	4
5	TRTH	Reset Pulse Width		4 TCLCL		4 TCLCL		ns	
6	TPGVRTL	PCTL, PDI, RFRQ to RESET↓ Setup		125		200		ns	5
7	TRTLPGX	PCTL, RFRQ to RESET↓ Hold		10		10		ns	
8	TCLPC	PCLK from CLK↓ Delay			45		65	ns	
9	TPDVCL	PDIn to CLK↓ Setup		60		100		ns	
10	TCLPDX	PDIn to CLK↓ Hold		40		65		ns	6

A.C. CHARACTERISTICS (Continued)

RAM WARM-UP AND INITIALIZATION

64	TCLWZL	$\overline{WZ}$ from CLK↓ Delay		40		65	ns	7

SYNCHRONOUS μP PORT INTERFACE

11	TPEVCL	$\overline{PE}$ to CLK↓ Setup	30		50			2
12	TKVCL	$\overline{RD}$, $\overline{WR}$, $\overline{PE}$, PCTL to CLK↓ Setup	20		30		ns	1
13	TCLKX	$\overline{RD}$, $\overline{WR}$, $\overline{PE}$, PCTL to CLK↓ Hold	0		0		ns	
14	TKVCH	$\overline{RD}$, $\overline{WR}$, PCTL to CLK↑ Setup	20		30		ns	2

ASYNCHRONOUS μP PORT INTERFACE

15	TRWVCL	$\overline{RD}$, $\overline{WR}$ to CLK↓ Setup	30		30		ns	8,9
16	TRWL	$\overline{RD}$, $\overline{WR}$ Pulse Width	2TCLCL+30		2TCLCL+50		ns	
17	TRWLPEV	$\overline{PE}$ from $\overline{RD}$, $\overline{WR}$↓ Delay 8207		TCLCL-20			ns	1
		8207		TCLCL-30			ns	2
		8207-2		TCLCL-30		TCLCL-50	ns	3
18	TRWLPEX	$\overline{PE}$ to $\overline{RD}$, $\overline{WR}$↓ Hold	2TCLCL+30		2TCLCL+50		ns	
19	TRWLPTV	PCTL from $\overline{RD}$, $\overline{WR}$↓ Delay		TCLCL-30		TCLCL-50	ns	2
20	TRWLPTX	PCTL to $\overline{RD}$, $\overline{WR}$↓ Hold	2TCLCL+30		2TCLCL+50		ns	2
21	TRWLPTV	PCTL from $\overline{RD}$, $\overline{WR}$↓ Delay		2TCLCL-20			ns	1
22	TRWLPTX	PCTL to $\overline{RD}$, $\overline{WR}$↓ Hold	3TCLCL+30				ns	1

A.C. CHARACTERISTICS (Continued)

RAM INTERFACE

Ref.	Symbol	Parameter	8207 & 8207-2 Min.	8207 & 8207-2 Max.	8207-5 Min.	8207-5 Max.	Units	Notes
23	TAVCL	AL, AH, BS to CLK↓ Setup	35 + tASR		55 + tASR		ns	10
24	TCLAX	AL, AH, BS TO CLK↓ Hold	0		0		ns	
25	TCLLN	LEN from CLK↓ Delay		35		55	ns	
26	TCLRSL	$\overline{RAS}$↓ from CLK↓ Delay		35		55	ns	
28	TCLRSH	$\overline{RAS}$↑ from CLK↓ Delay		50		70	ns	
27	tRCD	$\overline{RAS}$ to $\overline{CAS}$ Delay	TCLCL/2-25 75 TCLCL-25		60		ns ns ns	11,13,14 12,13,14 1,13,14
29	tRAH	Row AO to $\overline{RAS}$↓ Hold	TCLCL/4-10 40 TCLCL/2-10 90		30		ns ns ns ns	11,13,15 12,13,15 1,13,15,16 13,15,17
30	tASR	Row AO to $\overline{RAS}$↓ Setup						10,18
31	tASC	Column AO to $\overline{CAS}$↓ Setup	5 5 TCLCL/2-26		5		ns ns ns	11,13,19,20 12,13,19 1,13,19
32	tCAH	Column AO to $\overline{CAS}$ Hold	(See DRAM Interface Tables)					21
33	TCLCSL	$\overline{CAS}$↓ from CLK↓ Delay	TCLCL/4+30	TCLCL/1.8+53	TCLCL/4+30	TCLCL/1.8+78	ns ns	11 12
34	TCLCSL	$\overline{CAS}$↓ from CLK↓ Delay		35			ns	1
35	TCLCSH	$\overline{CAS}$↑ from CLK↓ Delay		50		70	ns	
36	TCLW	WE from CLK↓ Delay		35		55	ns	
37	TCLTKL	$\overline{XACK}$↓ from CLK↓ Delay		35		55	ns	
38	TRWLTKH	$\overline{XACK}$↑ from $\overline{RD}$↑, $\overline{WR}$↑ Delay		50		60	ns	
39	TCLAKL	$\overline{AACK}$↓ from CLK↓ Delay		35		55	ns	
40	TCLAKH	$\overline{AACK}$↑ from CLK↓ Delay		50		70	ns	
41	TCLDL	$\overline{DBM}$ from CLK↓ Delay		35		55	ns	

ECC INTERFACE

42	TWRLFV	$\overline{FWR}$ from $\overline{WR}$↓ Delay 8207 8207-2		2TCLCL-40 TCLCL+ TCL-40		TCLCL+ TCL-65	ns ns	1,22 2,22
43	TFVCL	$\overline{FWR}$ to CLK↓ Setup	40		65		ns	23
44	TCLFX	$\overline{FWR}$ to CLK↓ Hold	0		0		ns	24
45	TEVCL	$\overline{ERROR}$ to CLK↓ Setup	20		30		ns	25,26
46	TCLEX	$\overline{ERROR}$ to CLK↓ Hold	0		0		ns	
47	TCLRL	R/$\overline{W}$ from CLK↓ Delay		40		55	ns	
48	TCLRH	R/$\overline{W}$↑ from CLK↓ Delay		50		70	ns	
49	TCEVCL	CE to CLK↓ Setup	20		30		ns	25,27
50	TCLCEX	CE to CLK↓ Hold	0		0		ns	
51	TCLES	$\overline{ESTB}$ from CLK↓ Delay		35		55	ns	

A.C. CHARACTERISTICS (Continued)

PORT SWITCHING AND LOCK

Ref.	Symbol	Parameter	8207 & 8207-2 Min.	8207 & 8207-2 Max.	8207-5 Min.	8207-5 Max.	Units	Notes
52	TCLMV	MUX from CLK↓ Delay		45		65	ns	
53	TCLPNV	PSEN from CLK↓ Delay	TCL TCL	60 TCL+35	TCL TCL	60 TCL+35	ns ns	28 29
54	TCLPSV	PSEL from CLK↓		35		55	ns	
55	TLKVCL	LOCK to CLK↓ Setup	30		50		ns	30,31
56	TCLLKX	LOCK to CLK↓ Hold	10		10		ns	30,31
57	TRWLLKV	LOCK from RD↓, WR↓ Delay		2TCLCL-30		2TCLCL-50	ns	31,32
58	TRWHLKX	LOCK to RD↑, WR↑ Hold	3TCLCL+30		3TCLCL+50		ns	31,32

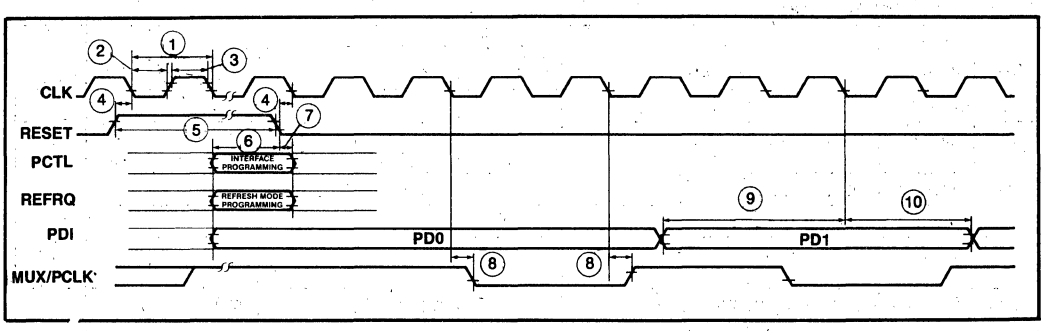
REFRESH REQUEST

59	TRFVCL	RFRQ to CLK↓ Setup	20		30		ns	
60	TCLRFX	RFRQ to CLK↓ Hold	10		10		ns	
61	TFRFH	Failsafe RFRQ Pulse Width	TCLCL+30		TCLCL+50		ns	33
62	TRFXCL	Single RFRQ Inactive to CLK↓ Setup	20		30		ns	34
63	TBRFH	Burst RFRQ Pulse Width	2TCLCL+30		2TCLCL+50		ns	33

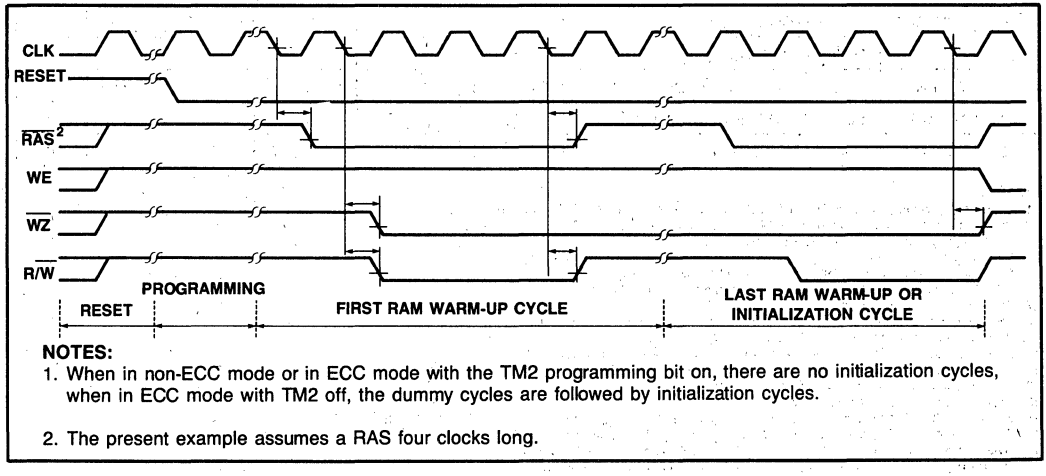
NOTES:
1. Specification when programmed in the Fast Cycle processor mode (iAPX 286 mode).
2. Specification when programmed in the Slow Cycle processor mode (iAPX 186 mode).
3. Must be programmed in Slow Cycle processor mode.
4. RESET is internally synchronized to CLK. Hence a set-up time is required only to guarantee its recognition at a particular clock edge.
5. The first programming bit (PD0) is also sampled by RESET going low.
6. TCLPDX is guaranteed if programming data is shifted using PCLK.
7. WZ is issued only in ECC mode.
8. TRWVCL is not required for an asynchronous command except to guarantee its recognition at a particular clock edge.
9. Valid when programmed in either Fast or Slow Cycle mode.
10. tASR is a user specified parameter and its value should be added accordingly to TAVCL.
11. When programmed in Slow Cycle mode and 125 ns ⩽ TCLCL < 200 ns.
12. When programmed in Slow Cycle mode and 200 ns ⩽ TCLCL.
13. Specification for Test Load conditions.
14. tRCD (actual) = tRCD (specification) +0.06 (ΔC_{RAS}) - 0.06 (ΔC_{CAS}) where ΔC = C (test load) - C (actual) in pF. (These are first order approximations.)
15. tRAH (actual) = tRAH (specification) + 0.06 (ΔC_{RAS}) - 0.022 (ΔC_{A0}) where ΔC = C (test load) - C (actual) in pF. (These are first order approximations.)
16. When programmed in Fast Cycle mode (8207 only) and 62.5 ns ⩽ TCLCL < 200 ns.
17. When programmed in Fast Cycle mode (8207 only) and 200 ns ⩽ TCLCL.
18. tASR (actual) = tASR (specification) +0.06 (ΔC_{A0}) - 0.025 (ΔC_{RAS}) where ΔC = C (test load) - C (actual) in pF. (These are first order approximations.)
19. tASC (actual) = tASC (specification) +0.06 (ΔC_{A0}) - 0.025 (ΔC_{CAS}) where ΔC = C (test load) - C (actual) in pF. (These are first order approximations.)
20. tASC is a function of clock frequency and thus varies with changes in frequency. A mimimum value is specified.
21. See 8207 DRAM Interface Tables 14 - 18.
22. TWRLFV is defined for both synchronous and asynchronous $\overline{FWR}$. In systems in which $\overline{FWR}$ is decoded directly from the address inputs to the 8207. TCLFV is automatically guaranteed by TCLAV.
23. TFVCL is defined for synchronous $\overline{FWR}$.
24. TCLFV is defined for both synchronous and asynchronous $\overline{FWR}$. In systems in which $\overline{FWR}$ is decoded directly from the address inputs to the 8207. TCLFV is automatically guaranteed by TCLAV.
25. $\overline{ERROR}$ and CE are set-up to CLK↓ in fast cycle mode and CLK↑ in slow cycle mode.
26. $\overline{ERROR}$ is set-up to the same edge as R/$\overline{W}$ is referenced to, in RMW cycles.
27. CE is set-up to the same edge as WE is referenced to in RMW cycles.
28. Specification when TCL < 25 ns.
29. Specification when TCL ⩾ 25 ns.
30. Synchronous operation only. Must arrive by the second clock falling edge after the clock edge which recognizes the command in order to be effective.
31. LOCK must be held active for the entire period the opposite port must be locked out. One clock after the release of LOCK the opposite port will be able to obtain access to memory.
32. Asynchronous mode only. In this mode a synchronizer stage is used internally in the 8207 to synchronize up LOCK. TRWLLKV and TRWHLKX are only required for guaranteeing that LOCK will be recognized for the requesting port, but these parameters are not required for correct 8207 operation.
33. TFRFH and TBRFH pertain to asynchronous operation only.
34. Single RFRQ cannot by supplied asynchronously.
35. tR and tF are referenced from the 3.5V and 1.0V levels.

WAVEFORMS

Clock and Programming Timings



RAM Warm-up and Memory Initialization Cycles



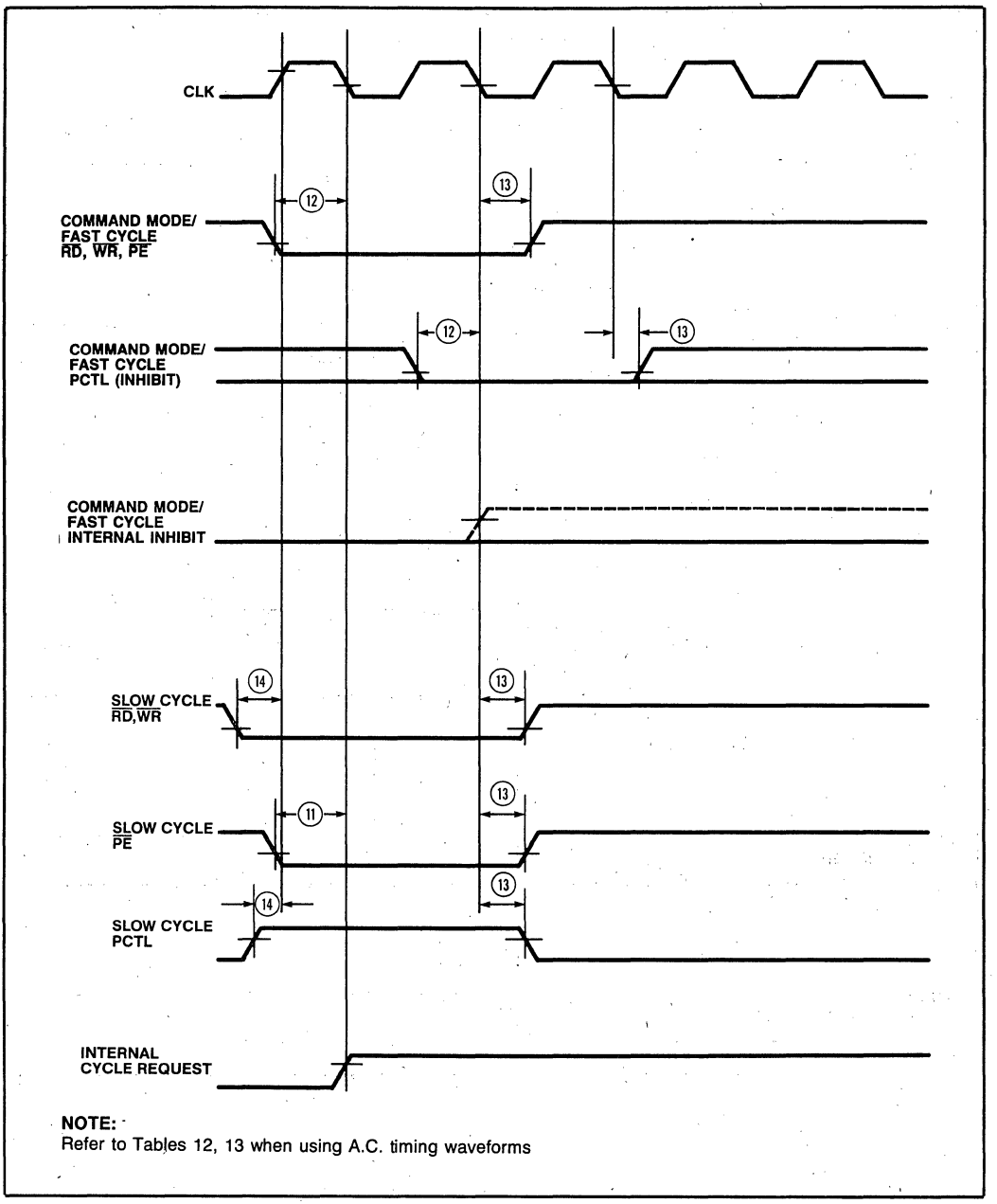
NOTES:

1. When in non-ECC mode or in ECC mode with the TM2 programming bit on, there are no initialization cycles, when in ECC mode with TM2 off, the dummy cycles are followed by initialization cycles.

2. The present example assumes a RAS four clocks long.

WAVEFORMS (Continued)

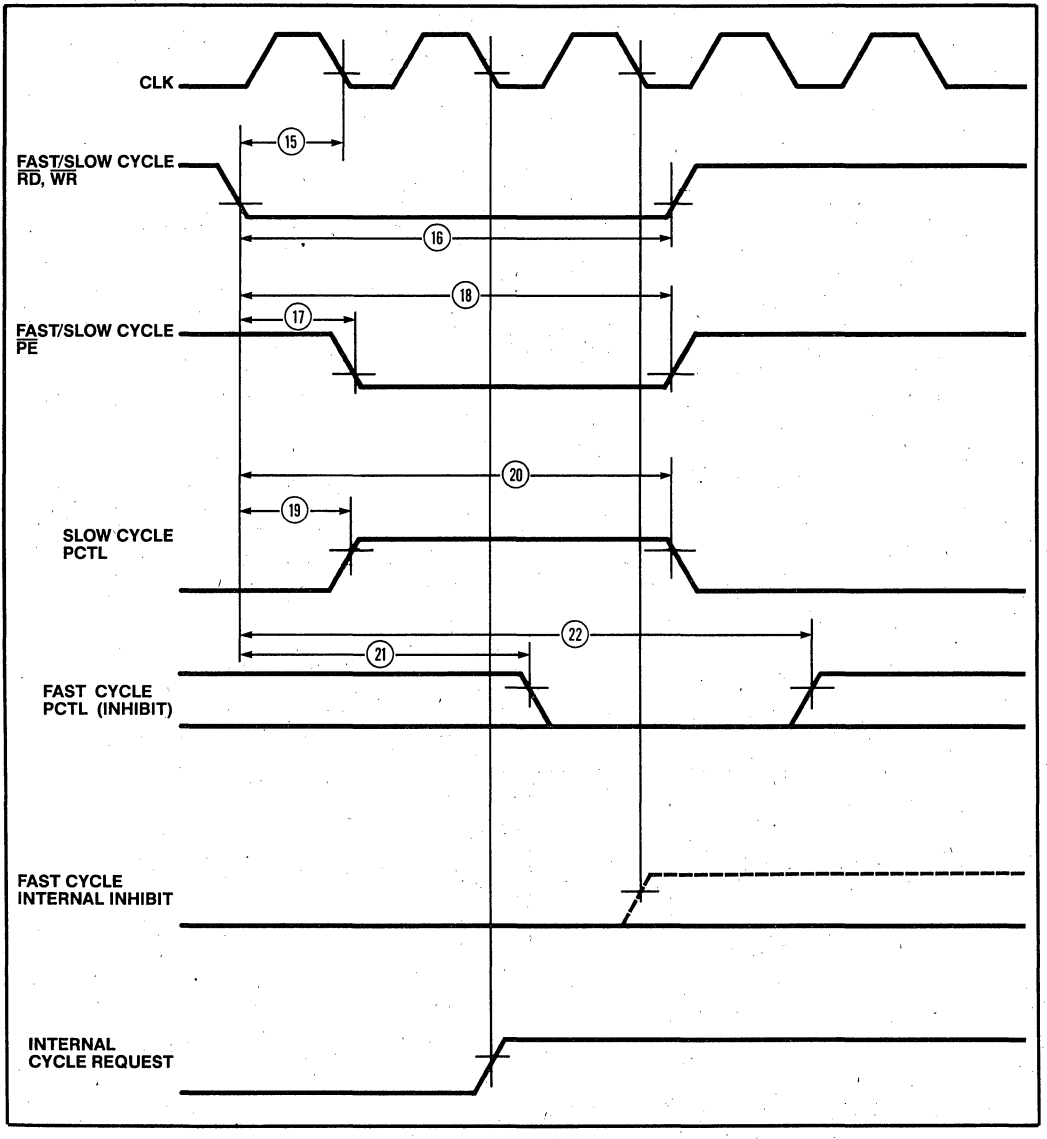
Synchronous Port Interface



NOTE:
Refer to Tables 12, 13 when using A.C. timing waveforms

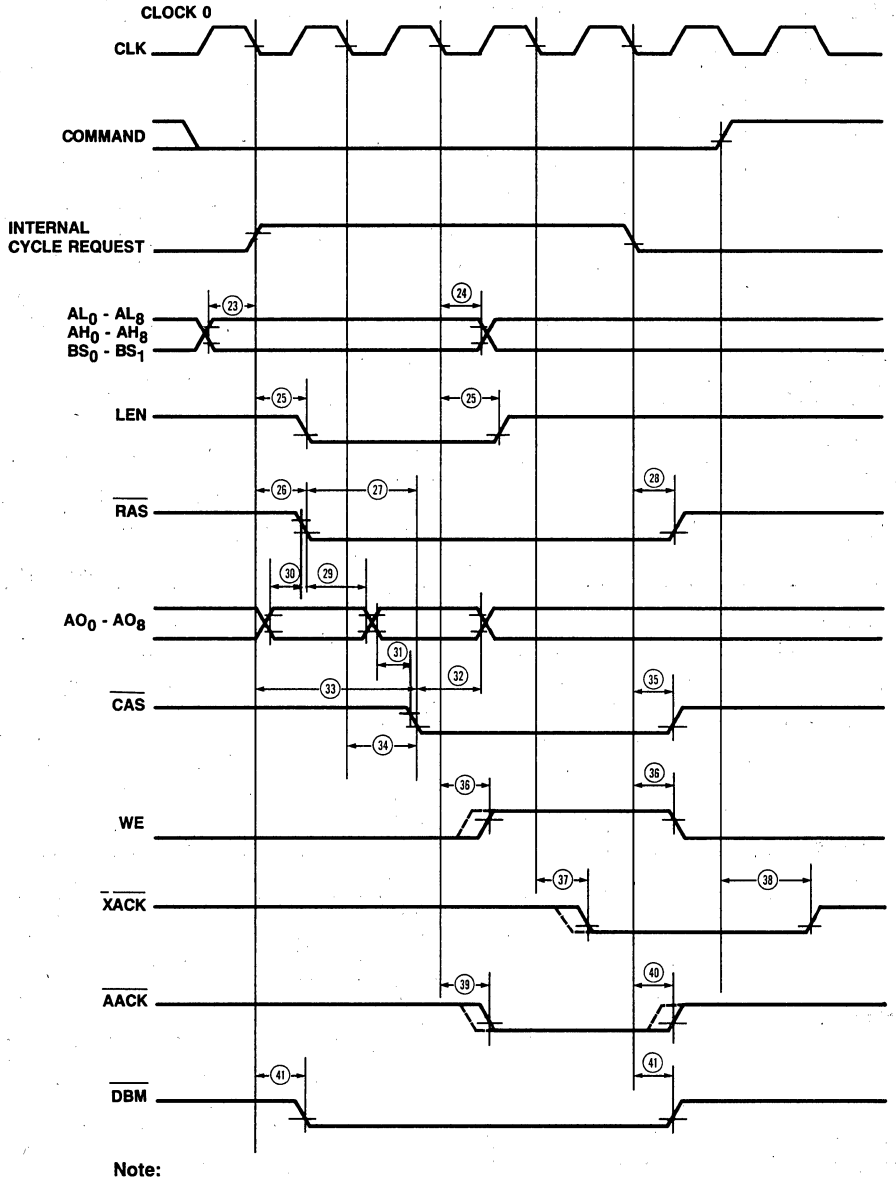
WAVEFORMS (Continued)

Asynchronous Port Interface



WAVEFORMS (Continued)

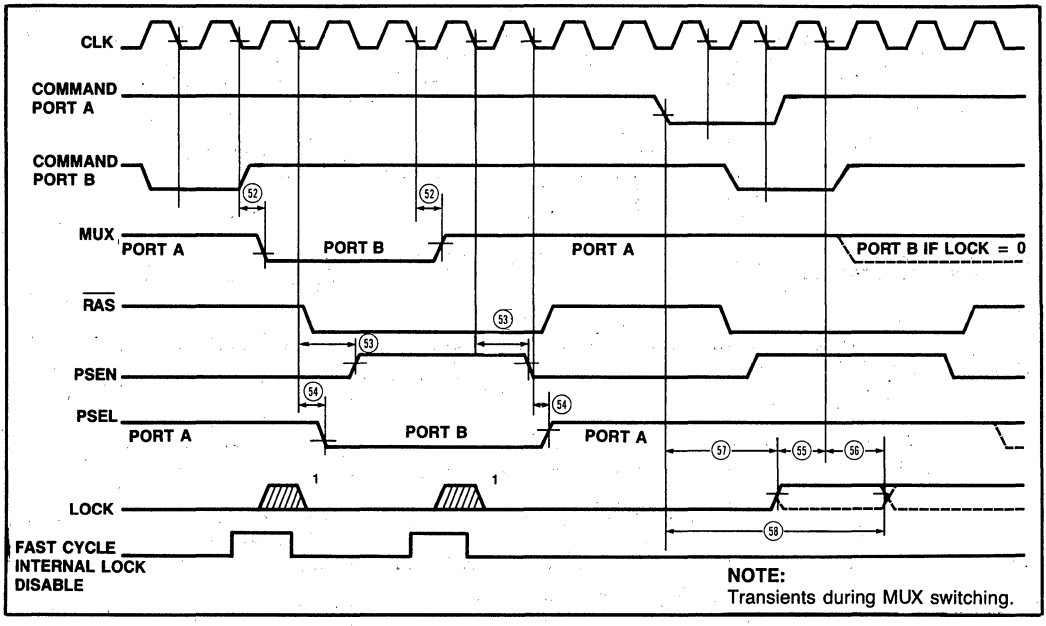
RAM Interface Timing
ECC and Non-ECC Mode



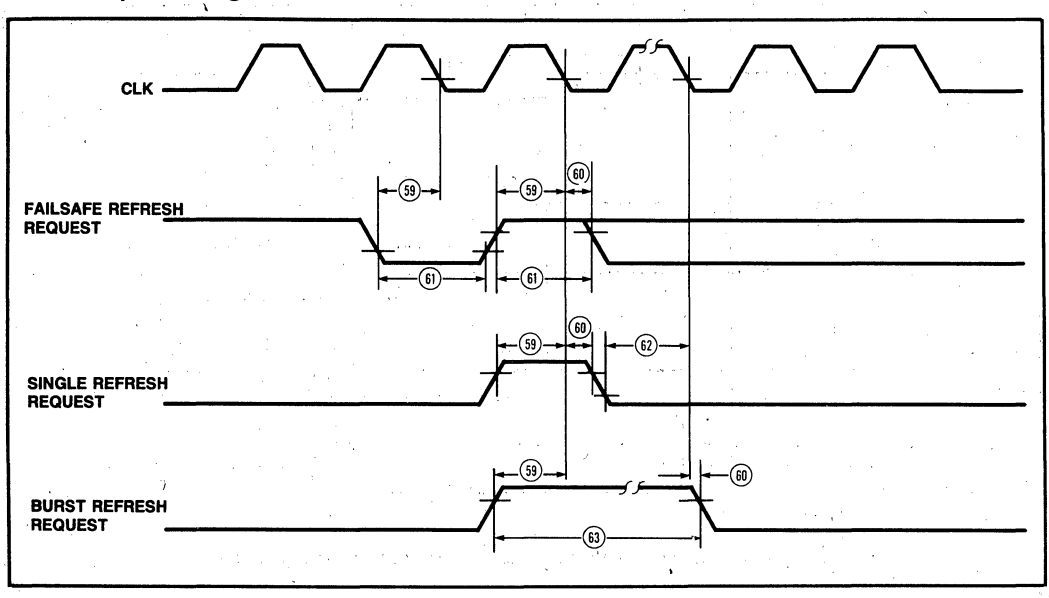
Note:
Dashed waveform indicates that either clock edge may cause the signal transition

WAVEFORMS (Continued)

Port Switching and Lock Timing

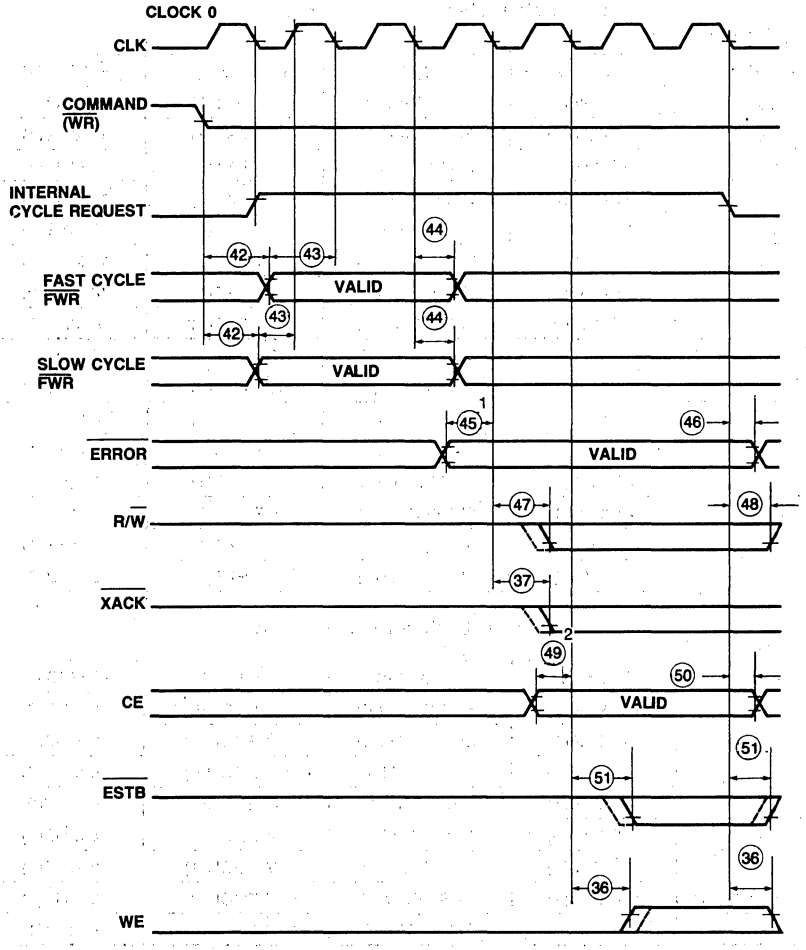


Refresh Request Timing



WAVEFORMS (Continued)

ECC Interface Timing



NOTE:

1. This parameter is set-up to the falling edge of clock, as shown, for fast cycle configurations. It is set-up to the rising edge of clock if in slow cycle configurations. Table 13A shows which clock and clock edge these signals are set-up in the R/$\overline{W}$ L column.

2. CE is set-up to the same edge as WE is referenced to in RMW cycles.

CONFIGURATION TIMING CHARTS

The timing charts that follow are based on 8 basic system configurations where the 8207 operates.

Tables 10 and 11 give a description of non-ECC and ECC system configurations based on the 8207's PD0, PD3, PD4, PD10 and PD11 programming bits.

Table 10. Non-ECC System Configurations

Non-ECC Mode: PD0=0

Timing Conf.	$\overline{CFS}$(PD3)	$\overline{RFS}$(PD4)	EXT(PD10)	$\overline{FFS}$(PD11)
C_0	iAPX286(0)	FAST RAM(0)	NOT EXT(0)	10 MHZ(1)
C_0	iAPX286(0)	FAST RAM(0)	EXT(1)	10 MHZ(1)
C_0	iAPX286(0)	SLOW RAM(1)	NOT EXT(0)	10 MHZ(1)
C_0	iAPX286(0)	SLOW RAM(1)	EXT(1)	10 MHZ(1)
C_0	iAPX286(0)	FAST RAM(0)	NOT EXT(0)	16 MHZ(0)
C_1	iAPX286(0)	SLOW RAM(1)	NOT EXT(0)	16 MHZ(0)
C_1	iAPX286(0)	FAST RAM(0)	EXT(1)	16 MHZ(0)
C_2	iAPX286(0)	SLOW RAM(1)	EXT(1)	16 MHZ(0)
C_3	iAPX186(1)	FAST RAM(0)	NOT EXT(0)	8 MHZ(0)
C_3	iAPX186(1)	SLOW RAM(1)	NOT EXT(0)	8 MHZ(0)
C_3	iAPX186(1)	FAST RAM(0)	EXT(1)	8 MHZ(0)
C_3	iAPX186(1)	FAST RAM(0)	NOT EXT(0)	5 MHZ(1)
C_3	iAPX186(1)	FAST RAM(0)	EXT(1)	5 MHZ(1)
C_3	iAPX186(1)	SLOW RAM(1)	NOT EXT(0)	5 MHZ(1)
C_3	iAPX186(1)	SLOW RAM(1)	EXT(1)	5 MHZ(1)
C_4	iAPX186(1)	SLOW RAM(1)	EXT(1)	8 MHZ(0)

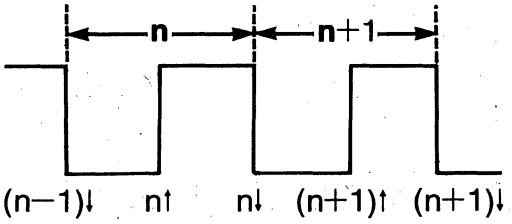
Table 11. ECC System Configurations

ECC Mode: PD0=1

Timing Conf.	CFS(PD3)	RFS(PD4)	$\overline{EXT}$(PD10)	FFS(PD11)
C_0	iAPX286(1)	SLOW RAM(0)	M/S EDCU(0)	10 MHZ(0)
C_0	iAPX286(1)	SLOW RAM(0)	M EDCU(1)	10 MHZ(0)
C_0	iAPX286(1)	FAST RAM(1)	M/S EDCU(0)	10 MHZ(0)
C_0	iAPX286(1)	FAST RAM(1)	M EDCU(1)	10 MHZ(0)
C_0	iAPX286(1)	FAST RAM(1)	M EDCU(1)	16 MHZ(1)
C_1	iAPX286(1)	SLOW RAM(0)	M EDCU(1)	16 MHZ(1)
C_2	iAPX286(1)	FAST RAM(1)	M/S EDCU(0)	16 MHZ(1)
C_3	iAPX286(1)	SLOW RAM(0)	M/S EDCU(0)	16 MHZ(1)
C_4	iAPX186(0)	SLOW RAM(0)	M/S EDCU(0)	5 MHZ(0)
C_4	iAPX186(0)	FAST RAM(1)	M/S EDCU(0)	5 MHZ(0)
C_4	iAPX186(0)	SLOW RAM(0)	M EDCU(1)	8 MHZ(1)
C_4	iAPX186(0)	FAST RAM(1)	M EDCU(1)	8 MHZ(1)
C_5	iAPX186(0)	SLOW RAM(0)	M/S EDCU(0)	8 MHZ(1)
C_5	iAPX186(0)	FAST RAM(1)	M/S EDCU(0)	8 MHZ(1)
C_6	iAPX186(0)	SLOW RAM(0)	M EDCU(1)	5 MHZ(0)
C_6	iAPX186(0)	FAST RAM(1)	M EDCU(1)	5 MHZ(0)

Using the Timing Charts

The notation used to indicate which clock edge triggers an output transition is "n↑" or "n↓", where "n" is the number of clock periods that have passed since clock 0, the reference clock, and "↑" refers to rising edge and "↓ to falling edge. A clock period is defined as the interval from a clock falling edge to the following falling edge. Clock edges are defined as shown below.



The clock edges which trigger transitions on each 8207 output are tabulated in Table 12 for non-ECC mode, and Table 13 for ECC mode. "H" refers to the high-going transition, and "L" to low-going transition; "V" refers to valid, and "$\overline{V}$" to non-valid.

Clock 0 is defined as the clock in which the 8207 begins a memory cycle, either as a result of a port request which has just arrived, or of a port request which was stored previously but could not be serviced at the time of its arrival because the 8207 was performing another memory cycle. Clock 0 may be identified externally by the leading edge of RAS. which is always triggered on 0↓.

Notes for interpreting the timing charts.

1. **PSEL - valid** is given as the latest time it can occur. It is entirely possible for PSEL to become valid before the time given. In a refresh cycle, PSEL can switch as defined in the chart, but it has no bearing on the refresh cycle itself, but only on a subsequent cycle for one of the external ports.

2. **LEN - low** is given as the latest time it can occur. LEN is only activated by port A configured in Fast Cycle iAPX286 mode, and thus it is not activated by a refresh cycle, although it may be activated by port A during a refresh cycle.

3. **ADDRESS - col** is the time column address becomes valid.

4. In non-ECC mode the $\overline{CAS}$, $\overline{EAACK}$, $\overline{LAACK}$ and $\overline{XACK}$ outputs are not issued during refresh.

5. In ECC mode there are really seven types of cycles: Read without error, read with error, full write, partial write without error, partial write with error, refresh without error, and refresh with error. These cycles may be derived from the timing chart as follows:

   A. Read without error: Use row marked 'RD, RF'.

   B. Read with error: Use row marked 'RMW', except for $\overline{EAACK}$ and $\overline{LAACK}$, which should be taken from 'RD, RF'. If the error is uncorrectable, WE will not be issued.

   C. Full write: Use row marked 'WR'.

   D. Partial write without error: Use row marked 'RMW', except that $\overline{DBM}$ and $\overline{ESTB}$ will not be issued.

   E. Partial write with error: Use row marked 'RMW', except that $\overline{DBM}$ will not be issued. If the error is uncorrectable, WE will not be issued.

   F. Refresh without error: Use row marked 'RD, RF', except that $\overline{ESTB}$, $\overline{EAACK}$, $\overline{LAACK}$, and $\overline{XACK}$ will not be issued.

   G. Refresh with error: Use row marked 'RMW' except that $\overline{EAACK}$, $\overline{LAACK}$, $\overline{ESTB}$, and $\overline{XACK}$ will not be issued. If the error is uncorrectable WE will not be issued.

6. **XACK - high** is reset asynchronously by command going inactive and not by a clock edge.

7. **MUX - valid** is given as the latest time it can occur.

Table 13 A. Timing Chart — ECC Mode

		PSEN		PSEL		DBM		LEN		RAS		CAS		R/W		WE	
C_n	CYCLE	H	L	V	$\overline{V}$	L	H	L	H	L	H	L	H	L	H	H	L
C_0	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓				
	WR	0↓	5↓	0↓	6↓			0↓	2↓	0↓	6↓	1↓	6↓	1↓	6↓	3↓	6↓
	RMW	0↓	8↓	0↓	9↓	0↓	9↓	0↓	2↓	0↓	9↓	1↓	9↓	4↓	9↓	6↓	9↓
C_1	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓				
	WR	0↓	5↓	0↓	6↓			0↓	2↓	0↓	6↓	1↓	6↓	1↓	6↓	3↓	6↓
	RMW	0↓	8↓	0↓	9↓	0↓	9↓	0↓	2↓	0↓	9↓	1↓	9↓	4↓	9↓	6↓	9↓
C_2	RD, RF	0↓	6↓	0↓	7↓	0↓	7↓	0↓	2↓	0↓	5↓	1↓	7↓				
	WR	0↓	6↓	0↓	7↓			0↓	2↓	0↓	7↓	1↓	7↓	1↓	7↓	4↓	7↓
	RMW	0↓	10↓	0↓	11↓	0↓	11↓	0↓	2↓	0↓	11↓	1↓	11↓	5↓	11↓	8↓	11↓
C_3	RD, RF	0↓	6↓	0↓	7↓	0↓	7↓	0↓	2↓	0↓	5↓	1↓	7↓				
	WR	0↓	6↓	0↓	7↓			0↓	2↓	0↓	7↓	1↓	7↓	1↓	7↓	4↓	7↓
	RMW	0↓	10↓	0↓	11↓	0↓	11↓	0↓	2↓	0↓	11↓	1↓	11↓	5↓	11↓	8↓	11↓
C_4	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓	0↓	2↓	0↓	3↓	0↓	4↓				
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	0↓	5↓	1↑	5↓	3↑	5↓
	RMW	0↓	6↓	0↓	7↓	0↓	7↓	0↓	2↓	0↓	7↓	0↓	7↓	3↑	7↓	5↑	7↓
C_5	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓	0↓	2↓	0↓	3↓	0↓	4↓				
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	0↓	5↓	1↑	5↓	3↑	5↓
	RMW	0↓	6↓	0↓	7↓	0↓	7↓	0↓	2↓	0↓	7↓	0↓	7↓	3↑	7↓	5↑	7↓
C_6	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓	0↓	2↓	0↓	3↓	0↓	4↓				
	WR	0↓	3↓	0↓	4↓			0↓	2↓	0↓	4↓	0↓	4↓	1↑	4↓	2↑	4↓
	RMW	0↓	4↓	0↓	5↓	0↓	5↓	0↓	2↓	0↓	5↓	0↓	5↓	2↑	5↓	3↑	5↓

Table 13 B. Timing Chart — ECC Mode

		COL ADDR		$\overline{ESTB}$		$\overline{EAACK}$		$\overline{LAACK}$		$\overline{XACK}$		MUX	
C_n	CYCLE	V	V	L	H	L	H	L	H	L	H	V	V
C_0	RD, RF	0↓	2↓			2↓	5↓	3↓	6↓	4↓	RD	−2↓	2↓
	WR	0↓	2↓			2↓	5↓	2↓	5↓	4↓	WR	−2↓	2↓
	RMW	0↓	2↓	6↓	8↓	5↓	8↓	5↓	8↓	7↓	WR	−2↓	2↓
C_1	RD, RF	0↓	3↓			3↓	6↓	3↓	6↓	4↓	RD	−2↓	2↓
	WR	0↓	3↓			2↓	5↓	2↓	5↓	4↓	WR	−2↓	2↓
	RMW	0↓	3↓	6↓	8↓	5↓	8↓	5↓	8↓	7↓	WR	−2↓	2↓
C_2	RD, RF	0↓	3↓			4↓	7↓	4↓	7↓	5↓	RD	−2↓	2↓
	WR	0↓	3↓			3↓	6↓	3↓	6↓	5↓	WR	−2↓	2↓
	RMW	0↓	3↓	8↓	10↓	7↓	10↓	7↓	10↓	9↓	WR	−2↓	2↓
C_3	RD, RF	0↓	3↓			4↓	7↓	5↓	8↓	5↓	RD	−2↓	2↓
	WR	0↓	3↓			3↓	6↓	3↓	6↓	5↓	WR	−2↓	2↓
	RMW	0↓	3↓	8↓	10↓	7↓	10↓	7↓	10↓	9↓	WR	−2↓	2↓
C_4	RD, RF	0↓	2↓			1↓	3↓	2↑	4↑	3↑	RD	−1↓	2↓
	WR	0↓	2↓			1↓	3↓	2↑	4↑	3↓	WR	−1↓	2↓
	RMW	0↓	2↓	5↑	6↑	3↓	5↓	4↑	6↑	5↓	WR	−1↓	2↓
C_5	RD, RF	0↓	2↓			2↓	4↓	3↑	5↑	3↑	RD	−1↓	2↓
	WR	0↓	2↓			1↓	3↓	2↑	4↑	3↓	WR	−1↓	2↓
	RMW	0↓	2↓	5↑	6↑	3↓	5↓	4↑	6↑	5↓	WR	−1↓	2↓
C_6	RD, RF	0↓	2↓			1↓	3↓	1↑	3↑	2↑	RD	−1↓	2↓
	WR	0↓	2↓			1↓	3↓	1↑	3↑	2↓	WR	−1↓	2↓
	RMW	0↓	2↓	3↑	4↑	1↓	3↓	2↑	4↑	3↓	WR	−1↓	2↓

8207 — DRAM Interface Parameter Equations

Several DRAM parameters, but not all, are a direct function of 8207 timings, and the equations for these parameters are given in the following tables. The following is a list of those DRAM parameters which have NOT been included in the following tables, with an explanation for their exclusion.

READ, WRITE, READ-MODIFY-WRITE & REFRESH CYCLES

tRAC: response parameter.
tCAC: response parameter.
tREF: See "Refresh Period Options"
tCRP: must be met only if CAS-only cycles, which do not occur with 8207, exist.
tRAH: See "A.C. Characteristics"
tRCD: See "A.C. Characteristics"
tASC: See "A.C. Characteristics"
tASR: See "A.C. Characteristics"
tOFF: response parameter.

READ & REFRESH CYCLES

tRCH: WE always goes active after $\overline{CAS}$ goes active, hence tRCH is guaranteed by tCPN.

WRITE CYCLE

tRC: guaranteed by tRWC.
tRAS: guaranteed by tRRW.
tCAS: guaranteed by tCRW.
tWCS: WE always activated after $\overline{CAS}$ is activated, except in memory initialization, hence tWCS is always negative (this is important for RMW only) except in memory initialization; in memory initialization tWCS is positive and has several clocks of margin.
tDS: system-dependent parameter.
tDH: system-dependent parameter.
tDHR: system-dependent parameter.

READ-MODIFY-WRITE CYCLE

tRWD: don't care in 8207 write cycles, but tabulated for 8207 RMW cycles.
tCWD: don't care in 8207 write cycles, but tabulated for 8207 RMW cycles.

Table 14. Non-ECC Mode - RD, RF Cycles

Parameter	Fast Cycle Configurations			Slow Cycle Configurations		Notes
	C_0	C_1	C_2	C_3	C_4	
tRP	3TCLCL–T26	4TCLCL–T26	4TCLCL–T26	2TCLCL–T26	2TCLCL–T26	1
tCPN	3TCLCL–T35	3TCLCL–T35	3TCLCL–T35	2.5TCLCL–T35	2.5TCLCL–T35	1
tRSH	2TCLCL–T34	3TCLCL–T34	3TCLCL–T34	3TCLCL–T34	4TCLCL–T34	1
tCSH	4TCLCL–T26	6TCLCL–T26	6TCLCL–T26	3TCLCL–T26	4TCLCL–T26	1
tCAH	TCLCL–T34	2TCLCL–T34	2TCLCL–T34	2TCLCL–T34	2TCLCL–T34	1
tAR	2TCLCL–T26	3TCLCL–T26	3TCLCL–T26	2TCLCL–T26	2TCLCL–T26	1
tT	3/30	3/30	3/30	3/30	3/30	2
tRC	6TCLCL	8TCLCL	8TCLCL	5TCLCL	6TCLCL	1
tRAS	3TCLCL–T26	4TCLCL–T26	4TCLCL–T26	3TCLCL–T26	4TCLCL–T26	1
tCAS	3TCLCL–T34	5TCLCL–T34	5TCLCL–T34	3TCLCL–T34	4TCLCL–T34	1
tRCS	2TCLCL–TCL –T36–TBUF	2TCLCL–TCL –T36–TBUF	2TCLCL–TCL –T36–TBUF	1.5TCLCL–TCL –T36–TBUF	1.5TCLCL–TCL –T36–TBUF	1

Table 12 A. Timing Chart — Non-ECC Mode

		PSEN		PSEL		$\overline{DBM}$		LEN		$\overline{RAS}$		$\overline{CAS}$		WE	
C_n	CYCLE	H	L	V	$\overline{V}$	L	H	L	H	L	H	L	H	H	L
C_0	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓	0↓	2↓	0↓	3↓	1↓	4↓		
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	1↓	5↓	2↓	5↓
C_1	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓		
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	1↓	5↓	2↓	5↓
C_2	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓		
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	1↓	5↓	2↓	5↓
C_3	RD, RF	0↓	2↓	0↓	3↓	0↓	3↓	0↓	2↓	0↓	3↓	0↓	3↓		
	WR	0↓	3↓	0↓	4↓			0↓	2↓	0↓	4↓	0↓	4↓	2↑	4↓
C_4	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓	0↓	2↓	0↓	4↓	0↓	4↓		
	WR	0↓	3↓	0↓	4↓			0↓	2↓	0↓	4↓	0↓	4↓	2↑	4↓

Table 12 B. Timing Chart — Non-ECC Mode

		COL ADDR		$\overline{EAACK}$		$\overline{LAACK}$		$\overline{XACK}$		MUX	
C_n	CYCLE	V	$\overline{V}$	L	H	L	H	L	H	V	$\overline{V}$
C_0	RD, RF	0↓	2↓	1↓	4↓	2↓	5↓	3↓	$\overline{RD}$	−2↓	2↓
	WR	0↓	2↓	1↓	4↓	1↓	4↓	3↓	$\overline{WR}$	−2↓	2↓
C_1	RD, RF	0↓	3↓	2↓	5↓	2↓	5↓	4↓	$\overline{RD}$	−2↓	2↓
	WR	0↓	3↓	1↓	4↓	1↓	4↓	3↓	$\overline{WR}$	−2↓	2↓
C_2	RD, RF	0↓	3↓	2↓	5↓	3↓	6↓	4↓	$\overline{RD}$	−2↓	2↓
	WR	0↓	3↓	1↓	4↓	1↓	4↓	3↓	$\overline{WR}$	−2↓	2↓
C_3	RD, RF	0↓	2↓	0↓	2↓	1↓	3↓	2↓	$\overline{RD}$	−1↓	2↓
	WR	0↓	2↓	0↓	2↓	1↑	3↑	2↓	$\overline{WR}$	−1↓	2↓
C_4	RD, RF	0↓	2↓	1↓	3↓	1↓	3↓	3↑	$\overline{RD}$	−1↓	2↓
	WR	0↓	2↓	0↓	2↓	1↑	3↑	2↓	$\overline{WR}$	−1↓	2↓

Table 15. Non-ECC Mode - WR Cycle

Parameter	Fast Cycle Configurations			Slow Cycle Configurations		Notes
	C_0	C_1	C_2	C_3	C_4	
tRP	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	2.5TCLCL−T35	2.5TCLCL−T35	1
tRSH	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tCSH	5TCLCL−T26	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	3/30	2
tRWC	8TCLCL	8TCLCL	8TCLCL	6TCLCL	6TCLCL	1
tRRW	5TCLCL−T26	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCRW	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tWCH	3TCLCL+TCL −T34	3TCLCL+TCL −T34	3TCLCL+TCL −T34	3TCLCL+TCL −T34	3TCLCL+TCL −T34	1, 3
tWCR	4TCLCL+TCL −T26	4TCLCL+TCL −T26	4TCLCL+TCL −T26	3TCLCL+TCL −T26	3TCLCL+TCL −T26	1, 3
tWP	2TCLCL+TCL −T36−TBUF	2TCLCL+TCL −T36−TBUF	2TCLCL+TCL −T36−TBUF	2TCLCL−T36 −TBUF	2TCLCL−T36 −TBUF	1
tRWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1
tCWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1

Table 16 A. ECC Mode — RD, RF Cycles

Parameter	Fast Cycle Mode				
	C_0	C_1	C_2	C_3	Notes
tRP	4TCLCL−T26	4TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCPN	3TCLCL−T35	3TCLCL−T35	3TCLCL−T35	3TCLCL−T35	1
tRSH	3TCLCL−T34	3TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tCSH	6TCLCL−T26	6TCLCL−T26	7TCLCL−T26	7TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	2
tRC	8TCLCL	8TCLCL	9TCLCL	9TCLCL	1
tRAS	4TCLCL−T26	4TCLCL−T26	5TCLCL−T26	5TCLCL−T26	1
tCAS	5TCLCL−T34	5TCLCL−T34	6TCLCL−T34	6TCLCL−T34	1
tRCS	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	1

Table 16 B. ECC Mode — RD, RF Cycles

Parameter	Slow Cycle Mode			
	C_4	C_5	C_6	Notes
tRP	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	1.5TCLCL−T35	1.5TCLCL−T35	1.5TCLCL−T35	1
tRSH	3TCLCL−T34	3TCLCL−T34	3TCLCL−T34	1
tCSH	4TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCAH	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	2
tRC	5TCLCL	5TCLCL	5TCLCL	1
tRAS	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tCAS	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tRCS	0.5TCLCL−T36 −TBUF	0.5TCLCL−T36 −TBUF	0.5TCLCL−T36 −TBUF	1

Table 17 A. ECC Mode — WR Cycle

Parameters	Fast Cycle Mode				Notes
	C_0	C_1	C_2	C_3	
tRP	3TCLCL–T26	3TCLCL–T26	3TCLCL–T26	3TCLCL–T26	1
tCPN	4TCLCL–T35	4TCLCL–T35	4TCLCL–T35	4TCLCL–T35	1
tRSH	5TCLCL–T34	5TCLCL–T34	6TCLCL–T34	6TCLCL–T34	1
tCSH	6TCLCL–T26	6TCLCL–T26	7TCLCL–T26	7TCLCL–T26	1
tCAH	TCLCL–T34	2TCLCL–T34	2TCLCL–T34	2TCLCL–T34	1
tAR	2TCLCL–T26	3TCLCL–T26	3TCLCL–T26	3TCLCL–T26	1
tT	3/30	3/30	3/30	3/30	2
tRWC	9TCLCL	9TCLCL	10TCLCL	10TCLCL	1
tRRW	6TCLCL–T26	6TCLCL–T26	7TCLCL–T26	7TCLCL–T26	1
tCRW	5TCLCL–T34	5TCLCL–T34	6TCLCL–T34	6TCLCL–T34	1
tWCH	5TCLCL–T34	5TCLCL–T34	6TCLCL–T34	6TCLCL–T34	1, 4
tWCR	6TCLCL–T26	6TCLCL–T26	7TCLCL–T26	7TCLCL–T26	1, 4
tWP	3TCLCL–T36 –TBUF	3TCLCL–T36 –TBUF	3TCLCL–T36 –TBUF	3TCLCL–T36 –TBUF	1
tRWL	3TCLCL–T36 –TBUF	3TCLCL–T36 –TBUF	3TCLCL–T36 –TBUF	3TCLCL–T36 –TBUF	1
tCWL	3TCLCL–T36 –TBUF	3TCLCL–T36 –TBUF	3TCLCL–T36 –TBUF	3TCLCL–T36 –TBUF	1

Table 17 B. ECC Mode — WR Cycle

Parameters	Slow Cycle Mode			
	C_4	C_5	C_6	Notes
tRP	2TCLCL–T26	2TCLCL–T26	2TCLCL–T26	1
tCPN	2.5TCLCL–T35	2.5TCLCL–T35	2.5TCLCL–T35	1
tRSH	5TCLCL–T34	5TCLCL–T34	4TCLCL–T34	1
tCSH	5TCLCL–T26	5TCLCL–T26	4TCLCL–T26	1
tCAH	2TCLCL–T34	2TCLCL–T34	2TCLCL–T34	1
tAR	2TCLCL–T26	2TCLCL–T26	2TCLCL–T26	1
tT	3/30	3/30	3/30	2
tRWC	7TCLCL	7TCLCL	6TCLCL	1
tRRW	5TCLCL–T26	5TCLCL–T26	4TCLCL–T26	1
tCRW	5TCLCL–T34	5TCLCL–T34	4TCLCL–T34	1
tWCH	5TCLCL–T34	5TCLCL–T34	4TCLCL–T34	1, 4
tWCR	5TCLCL–T26	5TCLCL–T26	4TCLCL–T26	1, 4
tWP	3TCLCL–TCL –T36–TBUF	3TCLCL–TCL –T36–TBUF	3TCLCL–TCL –T36–TBUF	1
tRWL	3TCLCL–TCL –T36–TBUF	3TCLCL–TCL –T36–TBUF	3TCLCL–TCL –T36–TBUF	1
tCWL	3TCLCL–TCL –T36–TBUF	3TCLCL–TCL –T36–TBUF	3TCLCL–TCL –T36–TBUF	1

Table 18 A. ECC Mode — RMW

Parameters	Fast Cycle Mode				Notes
	C_0	C_1	C_2	C_3	
tRP	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tCPN	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	1
tRSH	8TCLCL−T34	8TCLCL−T34	10TCLCL−T34	10TCLCL−T34	1
tCSH	9TCLCL−T26	9TCLCL−T26	11TCLCL−T26	11TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	2
tRWC	12TCLCL	12TCLCL	14TCLCL	14TCLCL	1
tRRW	9TCLCL−T26	9TCLCL−T26	11TCLCL−T26	11TCLCL−T26	1
tCRW	8TCLCL−T34	8TCLCL−T34	10TCLCL−T34	10TCLCL−T34	1
tRCS	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	1
tRWD	6TCLCL−T26	6TCLCL−T26	8TCLCL−T26	8TCLCL−T26	1
tCWD	5TCLCL−T34	5TCLCL−T34	7TCLCL−T34	7TCLCL−T34	1
tWP	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1
tRWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1
tCWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1

intel

December 1982

The Designer's Guide to iRAMs

Order Number: 210860-001

THE DESIGNER'S GUIDE TO iRAMs

The iRAM is the first of a new generation of VLSI memories; a complete dynamic RAM system on a chip. It combines the advantages of the simple static RAM interface with the high density and low power dissipation of the more economical dynamic RAM. While extraordinarily complex internally (more than 150,000 active elements), the external interface of the iRAM is just slightly different from the interface of the 2K × 8 static RAMs that you have used before. The iRAM can sit in a 28-pin Universal Site. Designs based on the 2186 are fully compatible with EPROMs, EEPROMs, and 8K × 8 static RAMs. This is an important consideration when designing for compatibility with multiple vendors. There are three differences between the iRAM and SRAM interface. These are described below.

This guide is intended to show you how to use the iRAM. It contains a functional summary of the iRAM, and several simple microprocessor and microcontroller application examples. The enclosed access time matrix matches the access speeds of the iRAM to the operating speed of any Intel microprocessor. For more detailed design information, please consult Intel Application Note 132 "Designing Memory Systems with the 8K × 8 iRAM".

FUNCTIONAL DESCRIPTION

Just like a static RAM (or EPROM), access to the iRAM is initiated by activating the Chip Enable control signal ($\overline{CE}$). The bar over $\overline{CE}$ indicates that it is an active low signal. Activating $\overline{CE}$ latches the valid external addresses from the system bus into the RAM. In simple systems, the iRAM's address latches can eliminate the need to demultiplex the address/data bus from the processor.

Because the leading (falling) edge of $\overline{CE}$ starts the internal sequencing of a memory cycle, a transition on the $\overline{CE}$ control signal must be glitch-free. Any spurious transitions of $\overline{CE}$ may inadvertantly select the iRAM at the wrong time, resulting in a loss of data or worse. Figure 1 shows a simple circuit that ensures a clean transition for $\overline{CE}$.

Once $\overline{CE}$ has been activated, the user may select one of three different cycles; a Read cycle, a Write cycle or a False Memory cycle. For a Read cycle, the read control line, called $\overline{OE}$ (Output Enable), is activated. Data will remain valid

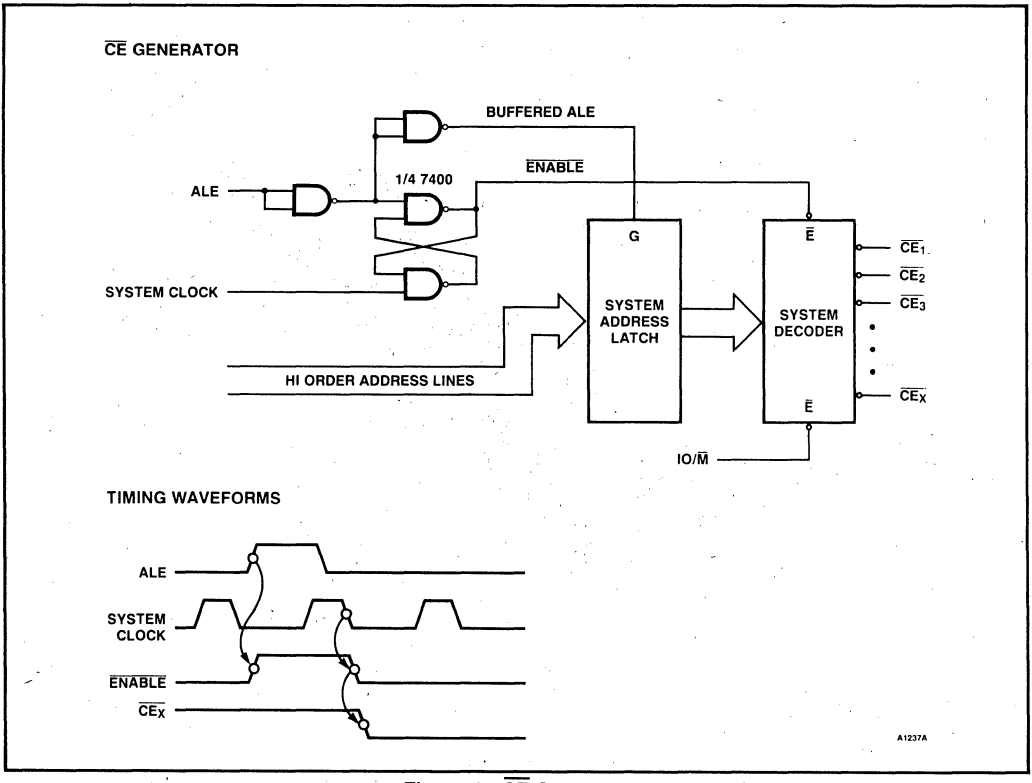


Figure 1. $\overline{CE}$ Generator

at the RAM outputs as long as $\overline{OE}$ is held active — regardless of the state of $\overline{CE}$. $\overline{OE}$ must return to the inactive state prior to the next occurence of $\overline{CE}$.

For a Write cycle, the write control line, called $\overline{WE}$ (Write Enable), is activated after $\overline{CE}$ is enabled. The only requirement for a successful Write cycle is that data be valid at the data inputs prior to $\overline{WE}$ going active. Some older processors, and the iAPX 86/88 in minimum mode, require the addition of a single flip-flop to delay $\overline{WE}$ going active until data is valid. Figure 2 illustrates an example of this circuit and the associated timing waveforms. Finally, $\overline{WE}$ must return to the inactive state prior to the next occurence of $\overline{CE}$. Note that $\overline{OE}$ and $\overline{WE}$ may not both go active during the same cycle.

A False Memory cycle (FMC) occurs whenever $\overline{CE}$ is activated and neither $\overline{OE}$ nor $\overline{WE}$ go active. Addresses must be valid at the iRAM prior to $\overline{CE}$ going active for an FMC. The designer should be aware that some unique timing requirements exist for FMCs.

Because internal refresh may be occurring at any time, a simple handshake procedure is used to notify the processor of a delay in the cycle. If $\overline{CE}$ arrives while the iRAM is in the midst of a refresh cycle, the ready output line (RDY) will go inactive low. RDY is usually used to generate WAIT states, and several RDY lines can be "OR'd" together at the system level. When both the internal refresh cycle and the requested external access cycle are complete, RDY is released and the processor finishes the transfer.

An alternate version of the iRAM gives the designer complete control over access and refresh cycles. On the 2187 iRAM, the RDY output is replaced by the refresh enable input ($\overline{REFEN}$). To initiate a refresh cycle, $\overline{REFEN}$ is activated. An internal refresh row address counter provides the refresh address. Note that $\overline{CE}$ and $\overline{REFEN}$ may not both go active during the same cycle. This synchronous iRAM is especially suited for use with microcontrollers that can not accept a ready handshake line.

In summary, there are three key interface requirements that the iRAM designer must address:

- $\overline{CE}$ input must be glitch free
- $\overline{WE}$ input may have to be delayed until data is valid
- RDY output is used to request a WAIT state during refresh/access overlap.

Some examples of how to use the iRAM are shown on the following pages. A microcontroller application and two different microprocessor systems are outlined, complete with their major interface elements. The timing charts match the operating speed of your processor with the appropriate iRAM access time.

A list of the iRAM literature that is available from Intel is included in this guide. You can get this literature, and answers to any questions about the availability and pricing of the iRAM from your local Intel sales office.

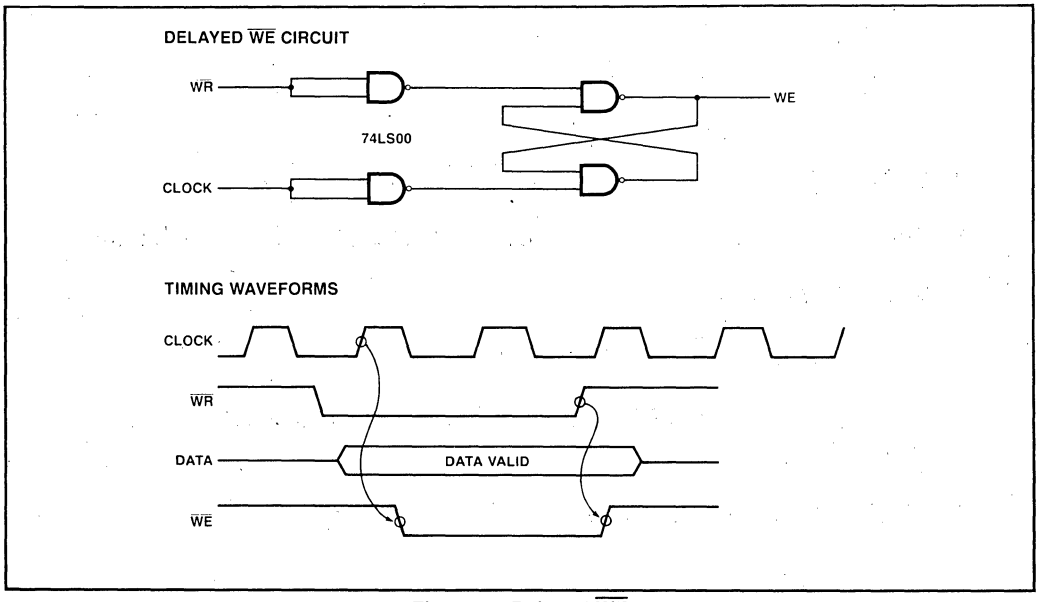
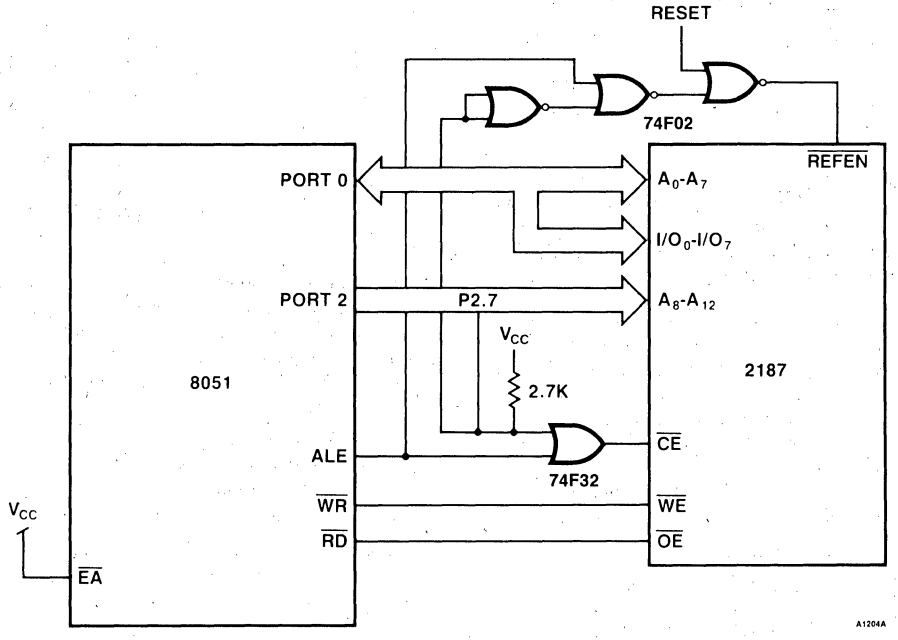


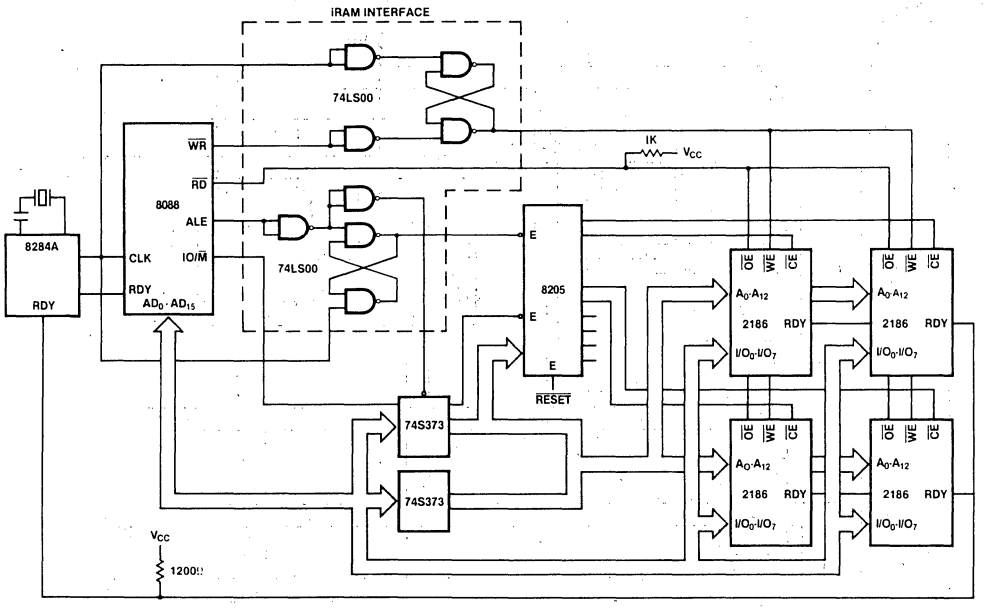
Figure 2. Delayed $\overline{WE}$

8051 MICROCONTROLLER SYSTEM



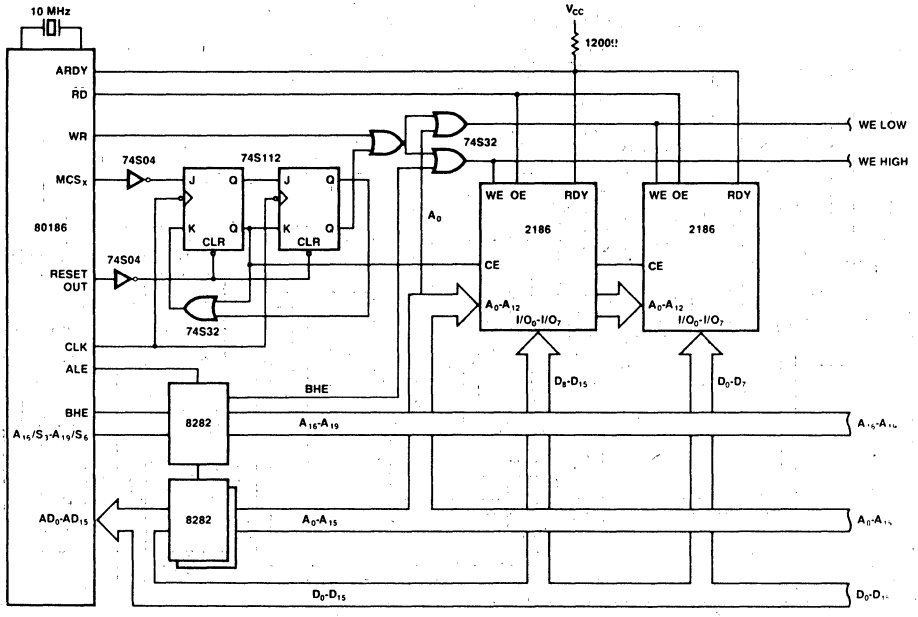
- USES THE 2187 SYNCHRONOUS REFRESH iRAM
- THE 2187 PROVIDES 8K BYTES OF EXTERNAL DATA MEMORY
- VERY SIMPLE INTERFACE
- NO ADDRESS LATCHES REQUIRED
- REFRESH OCCURS DURING INTERNAL MICROCONTROLLER OPCODE FETCHES
- SYSTEM RUNS AT 12 MHz

iAPX 88 MINIMUM MODE SYSTEM



- SIMPLE MINIMUM MODE SYSTEM
- FOUR 2186 iRAMs PROVIDE 32K BYTES OF LOCAL STORAGE
- CLEAN $\overline{CE}$ GENERATED BY ONE TTL PACKAGE (74LS00)
- ONE TTL PACKAGE (74LS00) DELAYS $\overline{WE}$ UNTIL DATA IS VALID
- THE iRAMs ARE IN UNIVERAL SITE SOCKETS — COMPATIBLE WITH SRAMs AND EPROMS
- SYSTEM RUNS AT 5 MHz WITH NO WAIT STATES

iAPX 186 SYSTEM



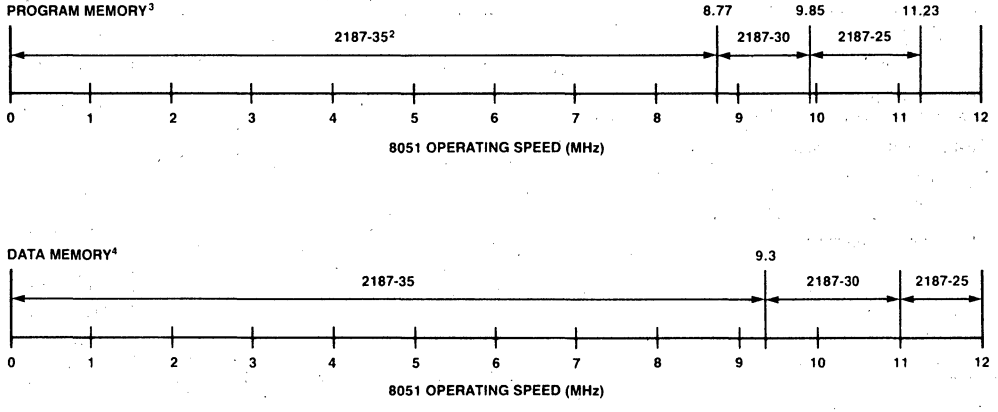
- TWO 2186 iRAMs PROVIDE 8K WORDS OF LOCAL STORAGE
- SIMPLE CIRCUITRY GENERATES A CLEAN $\overline{CE}$
- DELAYED $\overline{WE}$ GATED TO SELECT ONE OR BOTH iRAMs FOR 8 OR 16 BIT DATA TRANSFERS
- READY HANDSHAKE LINE (RDY) CONNECTS DIRECTLY TO THE PROCESSOR
- THE iRAMs UNIVERSAL SITE SOCKETS ARE COMPATIBLE WITH SRAMs AND EPROMs

Microprocessor Operating Speed Vs. iRAM Access Time[1]

Microprocessor Speed		0 Wait States	1 Wait State	2 Wait States	3 Wait States
8085AH	3 MHz	note 2	2186-35	2186-35	2186-35
	5 MHz		2186-35	2186-35	2186-35
	6 MHz		2186-30	2186-35	2186-35
8086[3]	5 Mhz	2186-30	2186-35	2186-35	2186-35
	8 MHz		2186-30	2186-35	2186-35
	10 MHz		2186-25	2186-35	2186-35
80186[4]	8 MHz		2186-30	2186-35	2186-35
	10 MHz			2186-30	2186-35
	15 MHz				2186-25
80286[4]	8 MHz			2186-30	2186-35
	10 MHz			2186-25	2186-35
	15 MHz				

NOTES:
1. Buffer delays not included.
2. Due to its RDY response requirements, the 8085 cannot run without wait states.
3. Timing also applicable for the 8088 microprocessor.
4. Specifications for higher clock speeds not available. Memory requirements for 10 MHz and 15 MHz are extrapolated from 8 MHz specifications.

8051 iRAM REQUIREMENTS[1]



NOTES:
1. Buffers delays not included.
2. Runs with no wait states over this speed range.
3. The iRAM is used to store both program code and data.
4. Program code resides in the 8051's internal ROM or in external ROM.

2186/2187 Literature

Title	Description	For Whom
Data Sheets	2186S, 2187	All iRAM users
Designer's Guide to the iRAM	General introduction to iRAM system design, includes system block diagrams and timing matrix: system speed vs. iRAM access time.	All iRAM users
Designing memory systems with the 8K x 8 iRAM	Application Note 132 — describes the design of iRAM memory systems, includes a description of internal iRAM functions.	All iRAM users
The iRAM in microcontroller systems	How to design microcontroller memory systems using the synchronous 2187 iRAM. Available February 83.	All 2187 users
Smart Memories	Article Reprint 235 — overview of latest trends in smart memory technologies	All iRAM users
Memory Components Handbook	Data sheets and Ap Notes for all Intel Memory Components — includes 2186S, 2187 data sheets and AP 132	All iRAM users
iRAM System Reliability	A summary of the system testing performed on the 2186. Includes reliability data and test descriptions.	iRAM users who require extensive qualification
iRAM Arbiter Reliability	Report on the reliability of the iRAM arbitration circuit.	iRAM users who require extensive qualification

2186/2187 Literature Guide by Topic

	Topic				
Title	iRAM Tech. Design	2186 System Design	2187 System Design	iRAM Reliability	Order Number
2186S Data Sheet	★	★			210857-001
2187 Data Sheet	★		★		210859-001
AP-132 Designing Systems with iRAMs	★	★	★		210443-001
AR-235 Smart Memories	★				210784-001
Designer's Guide to iRAMs		★	★		210860
iRAM System Reliability	★			★	Available on Request
iRAM Arbiter Reliability	★			★	Available on Request
The iRAM in microcontroller systems	★		★		Available Feb. 83

EPROMS (Erasable Programmable Ready Only Memories)

4

intel

APPLICATION NOTE

AP-154

April 1983

Programming Intel's 27256 EPROM

Robert Davis
Applications Engineering
Non-Volatile Memory Division
Intel Corporation

APRIL 1983
ORDER NUMBER: 210970-002

INTRODUCTION

With the introduction of the 27256 32K byte EPROM, a new generation of high density componentized software is possible. Intel's process and product technology advances have increased EPROM memory storage capabilities from 2K bits to 256K bits. This non-volatile memory can allow the designer a more reliable, user-friendly system. Since it is produced in the 28 pin JEDEC-approved Intel universal site, the 27256 can easily be designed into existing printed circuit boards.

Figure 1 shows the evolution of Intel's EPROM family. The new generation 27256 brings with it improved performance and state-of-the-art reliability. The absence of $\overline{PGM}$, replaced by A14, and a lower programming voltage (12.5V) highlight the additions accompanying the 27256. Advanced technology from the 27256 will soon bring enhanced performance to lower density EPROMs, specifically the 2764A and 27128A. This document concerns programming characteristics of the 27256, concentrating on those factors which will be new to the EPROM memory designer.

THE int$_e$ligent Programming™ ALGORITHM

The int$_e$ligent Programming™ Algorithm was developed as an improved alternative to the 50 msec per byte programming techniques for Intel's 2764 and 27128 EPROMs. By taking advantage of the variable programming times required by the cells in an EPROM array, programming speed increases of 5 or 6 times have been achieved. The success of this algorithm, coupled with the long programming times which would be inherent in programming the 27256 with a 50 msec per byte algorithm, has prompted Intel to develop a new 27256 int$_e$ligent Programming™ Algorithm specifically tailored to the requirements of the customer.

The 27256 programming algorithm is similar to the int$_e$ligent Programming Algorithms used for Intel's 2764 and 27128 EPROMs. It is now available as a standard feature in many PROM programmers. This new programming algorithm is fast and guarantees that each cell has been programmed reliably.

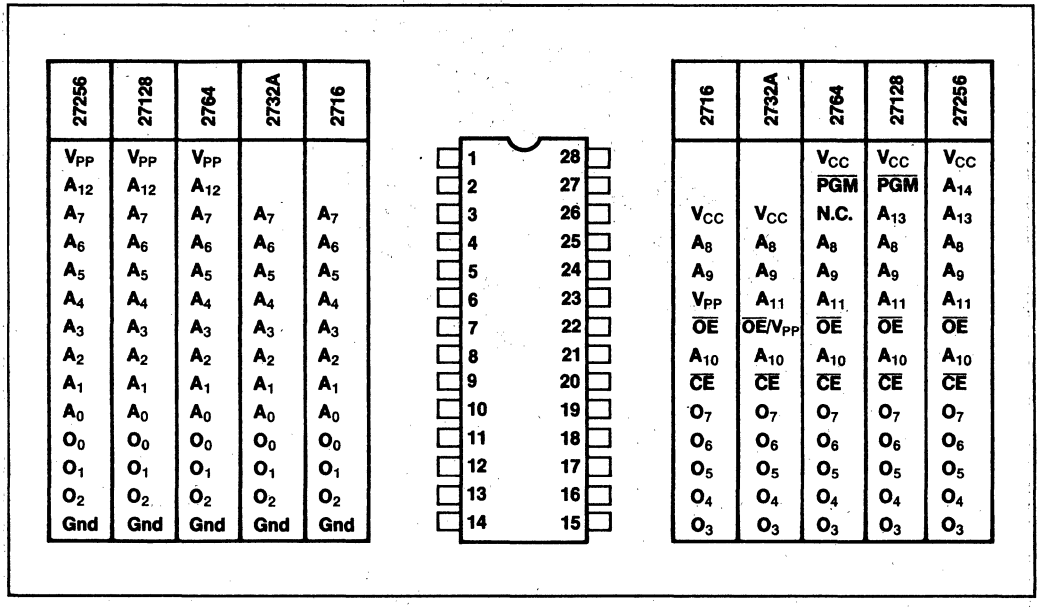


Figure 1. Intel Universal Memory Site for EPROMs

The 27256 int$_e$ligent Programming Algorithm shown in figure 2 is a feedback control loop. In examining this diagram, three distinct characteristics can be seen. First, the programming voltage has been reduced from 21 ± 0.5 volts, as was required on earlier generation 2764 and 27128 EPROMs, to 12.5 ± 0.5 volts. In all cases the V_{PP} voltage should never exceed 14 volts, and a 0.1 microfarad capacitor should be placed between V_{PP} and ground to insure proper decoupling. The technology advances implemented in the 27256 allow the programming voltage to be reduced while taking advantage of the performance of the programmed cell. The second characteristic that should be noted is the maximum number of 1ms pulses which are applied throughout the closed loop programming algorithm. As shown, 25 iterations of the loop is the maximum number allowed. Intel's reliability data indicate that 25 iterations is sufficient to reliably program each of the EPROM bits in the array. The last characteristic which should be mentioned is the insertion of a byte verify after the iteration count has been maximized. This will save time by failing any device which may not verify correctly after 25 one msec pulses.

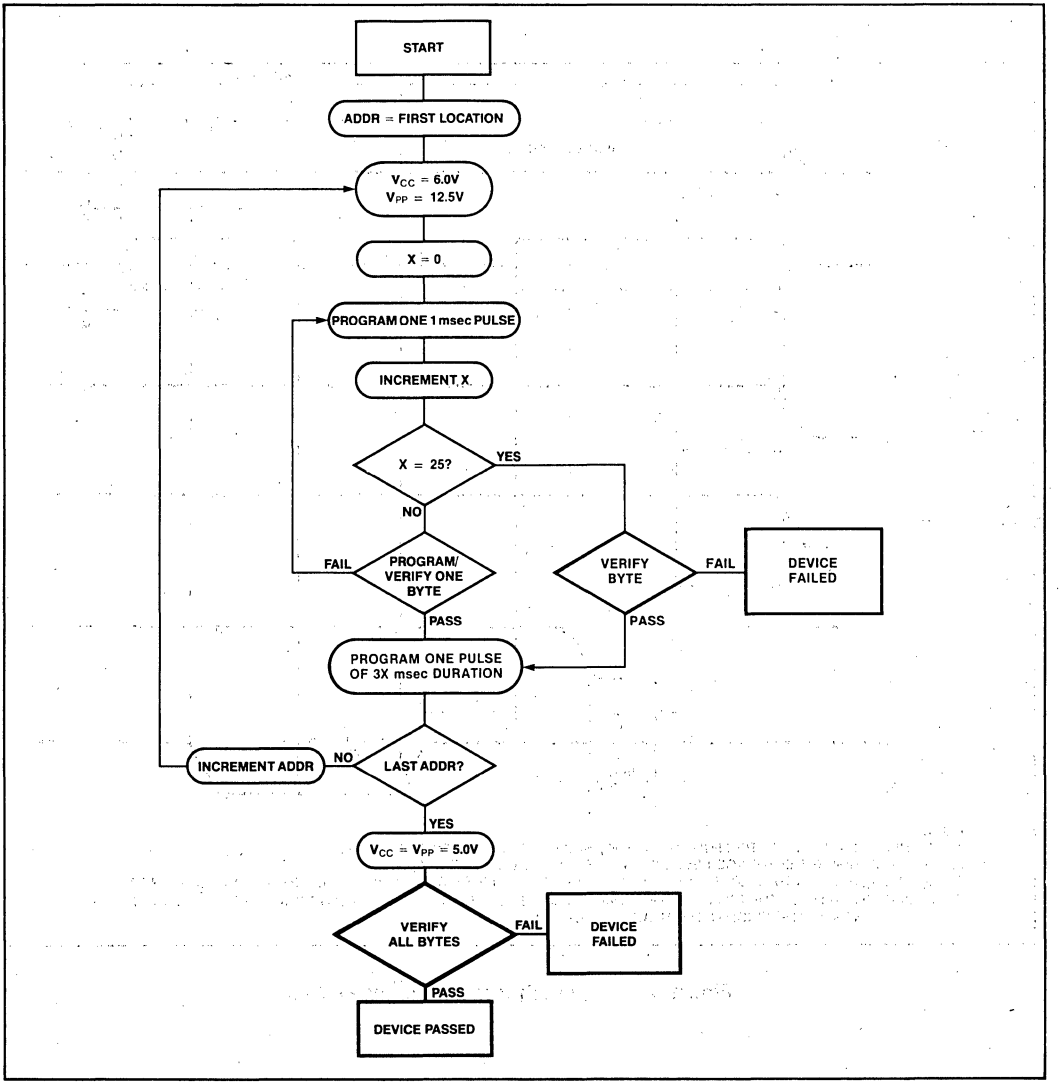
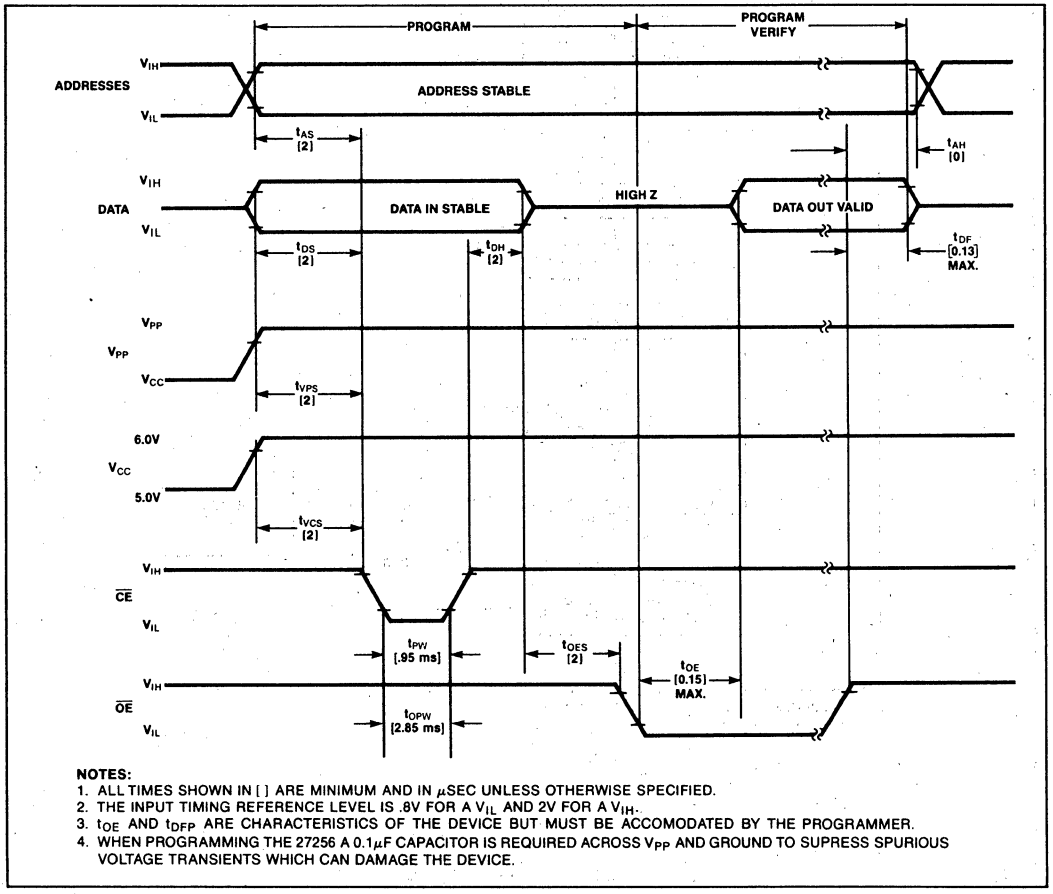


Figure 2. int$_e$ligent Programming™ Algorithm

27256 PROGRAMMING WAVEFORMS

With the replacement of the $\overline{PGM}$ signal by A14 on Pin 27, changes are required in the manner in which the 27256 is programmed. Figure 3 shows the waveforms required to program the 27256. As illustrated, the 27256 will be placed in the program mode when V_{PP} is set to 12.5 volts, and $\overline{CE}$ is pulsed to V_{IL}. The verify operation will then be selected with $\overline{CE}$ at V_{IH}, and $\overline{OE}$ held at V_{IL}. This results from multiplexing both the chip enable function and the programming enable function onto Pin 20. The V_{PP} level, either 5 volts or 12.5 volts, gates which function is selected. Table 1 indicates the various modes of operation of a 27256.

In on-board gang programming applications, care must be taken not to enable the outputs of all resident 27256 devices while in the verify mode. A common $\overline{OE}$ signal could cause bus contention. AR-294 will discuss this issue in detail.



NOTES:
1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR A V_{IL} AND 2V FOR A V_{IH}.
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 27256 A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

Figure 3. int$_e$ligent Programming™ Waveforms

Table 1. 27256 Operational Modes

MODE \ PINS	CE (20)	OE (22)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	OUTPUTS (11-13, 15-19)
Read	V_{IL}	V_{IL}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	V_{CC}	V_{CC}	High Z
intelligent Programming	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{OUT}
Optional Verify	V_{IL}	V_{IL}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	V_{IH}	X	V_{PP}	V_{CC}	High Z
intelligent Identifier	V_{IL}	V_{IL}	V_H	V_{CC}	V_{CC}	Code

NOTES:
1. X can be V_{IH} or V_{IL}.
2. V_H = 12.0V ± 0.5V

SUMMARY

The introduction of the 27256 continues Intel's leadership in high density EPROM storage capability. With this density increase comes the ability to design in system firmware, creating a more reliable, user-friendly environment for the computer operator. The intelligent Programming Algorithm developed for Intel's 2764 and 27128 EPROMs, provides many benefits, including lower costs, which result from higher system manufacturing throughput. In line with this strategy, the 27256 intelligent Programming Algorithm was designed to meet the technology requirements of the new device. With this algorithm, the 27256 may be programmed according to the waveforms shown in figure 3, ensuring programming reliability and efficiency.

intel

ARTICLE
REPRINT

AR-260

March 1983

E-PROMs Graduate To 256-K Density With Scaled N-Channel Process

M. Van Buskirk, M. Holler, G. Korsh,
B. Lee, S. Lee, D. Tang, G. Teng,
S. Fouts, P. Dang, and W. Fisher

MARCH 1983
ORDER NUMBER: 210988-001

Technical articles

E-PROMs graduate to 256-K density with scaled n-channel process

With 32-K bytes per chip, erasable programmable read-only memory can carry application software for business and personal computers

by M. Van Buskirk, M. Holler, G. Korsh, B. Lee, S. Lee,
D. Tang, G. Teng, S. Fouts, P. Dang, and W. Fisher, *Intel Corp., Santa Clara, Calif.*

☐ Since the introduction of the 2-K 1702 in 1971, the erasable programmable read-only memory has shaken off its reputation as a mere prototyping tool and emerged as a major commodity, worth some $240 million last year in the U. S. alone. Central to that growth, a heady pace of process and circuit innovations has doubled E-PROM densities every one to two years.

The advent of the 256-K chip—exemplified by the 27256 from Intel—signals the crossing of key technical hurdles and with it an open path to accelerated development of megabit and larger arrays. Though clearly the child of earlier generations of E-PROMs, the part has been thoroughly scaled down to achieve thinner oxides and minimum features of 1 micrometer. New sensing and decoding circuitry are incorporated as well.

Along the path to even larger arrays, the price per bit of E-PROMs will continue to drop, closing in on that of ROM, the least expensive semiconductor storage. Indeed, the 256-K ROM has only a year's jump on the 27256. The cost of E-PROM is projected at just 50% more than that of ROM in 1985: 6 versus 4 millicents per bit.

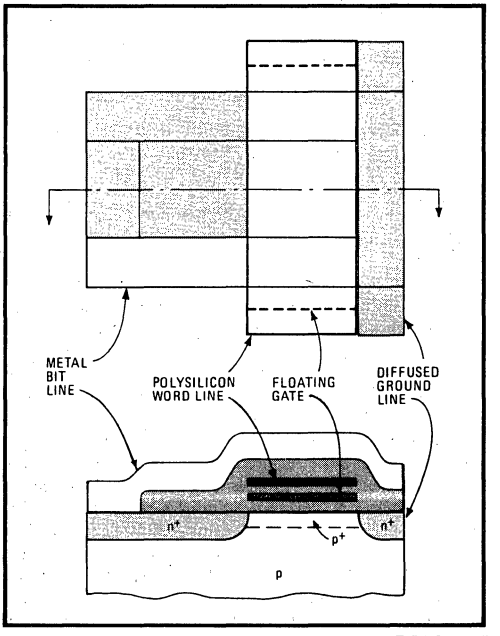
Achieving 256-K density in an E-PROM calls for a host of advances working in concert, dramatic scaling down of device dimensions being only the most obvious one. Processing changes also accompany the smaller geometries and thinner layers. New designs for the decoding and sensing circuitry cope with the worsening problem of statistical variations in device parameters among the more than quarter million bits in the array.

The flexibility of alterable nonvolatile storage coupled with the economy of high chip density promises intriguing possibilities for the architect of microsystems, particularly portable computers for the mass market. As the table on page 93 suggests, significant system and application software fits in a very few 256-K arrays. For example, two or three chips can carry a Pascal compiler and a sophisticated word-processing program.

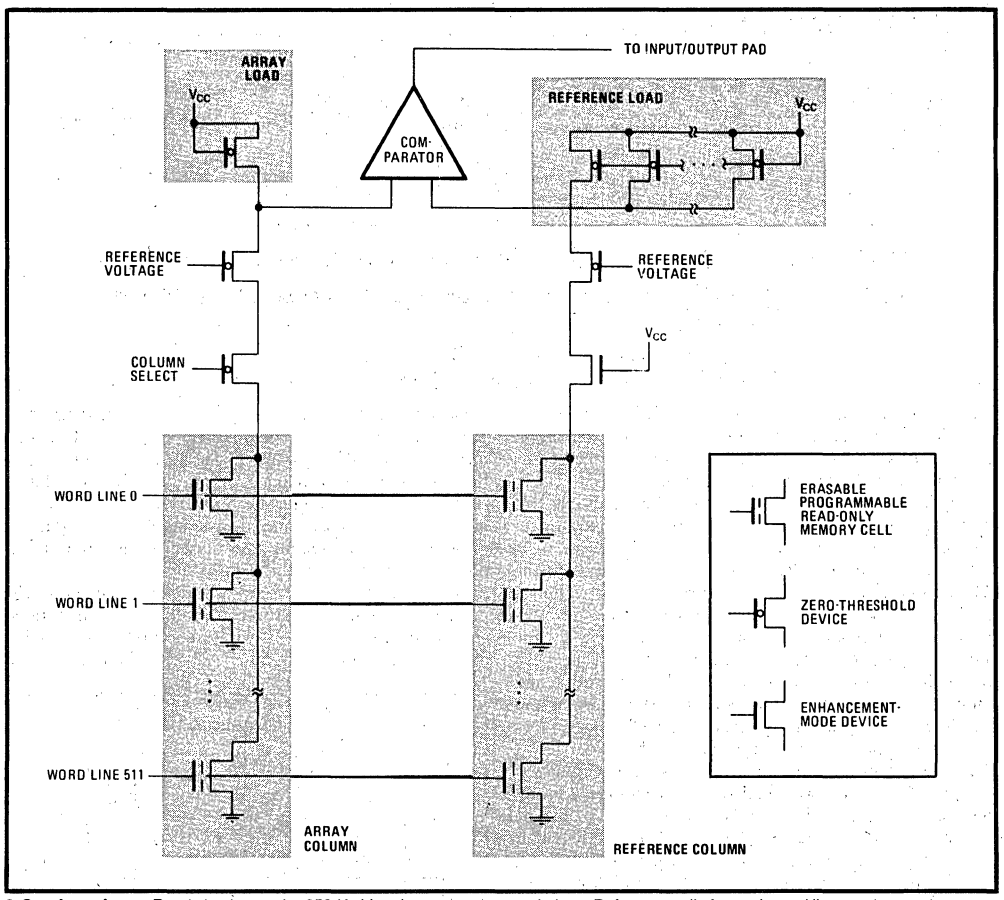
To the user of a machine incorporating such firmware, the friendliness and convenience of fast, 200-nanosecond memory accessed with the push of a button contrasts sharply with the fuss and delay of floppy disks. Further, for the end user and equipment maker alike, E-PROM continues to offer compelling advantages over ROM, advantages that will come at an ever lower premium as E-PROMs succeed in catching up with ROMs in density.

Speeding a product to market, for example, often means last-minute software changes, a costly and impractical requirement with mask-programmable chips, but a simple matter for E-PROMs. Similarly, the E-PROM can hold down the costs and delays involved in updating programs or in correcting those last few bugs that somehow made it out into the field.

Scaling chip geometries down in size has been the primary instrument of progress in all types of integrated circuits in the last 6 to 10 years. The first major scaling down of E-PROMs occurred in 1980 with the introduction



1. Scaled. Two-micrometer design rules squeeze the E-PROM cell down to 6 by 6 μm. The active channel area, beneath the floating polysilicon gate, is just 1 by 1.2 μm. The n^+ regions are 0.5 μm deep.



2. Sensing scheme. Read circuitry on the 256-K chip tolerates process variations. Reference cells for each word line monitor read currents; identical geometry for the array and reference load transistors ensures a constant current ratio for sensing.

of the 2764 64-K part with a 159-square-micrometer cell and H-MOS (high-performance MOS) peripheral circuits. The major technology advance prior to that was the conversion in 1977 to n-channel MOS technology with depletion-mode devices for the 16-K 2716, a move that made operation possible from a single 5-volt power supply.

A completely redesigned process and second major scaling in 1982 have resulted in another leap in density and performance for E-PROMs, producing a 256-K array with a 6-by-6-μm cell, for a chip size of just 28,500 square mils. The floating-gate storage transistor, with a channel effectively 1 μm long by 1.2 μm wide, employs a first oxide 325 angstroms thick and a second of 400 Å (Fig. 1). The peripheral circuitry, with a 0.5-picojoule speed-power product, is equivalent to chips built in H-MOS II, a second-generation high-performance MOS technology that uses 2-m channel lengths and 400-Å gate oxides for minimum gate delays of 0.4 ns.

The practical constraint of maintaining the same 5-V power supply used in previous technologies while scaling transistor sizes stresses the materials constituting the devices: both substrate and gate dielectrics are exposed to much higher electric fields. The fabrication process must build a margin of safety into the chip to prevent unwanted effects like time-dependent oxide failure, pn-junction breakdown, and parasitic MOS-transistor action between adjacent diffused regions.

In E-PROMs, the high voltage required for programming the cells further aggravates the situation. In fact, in the storage transistor, the oxide surrounding the floating gate must be flawless, or else electrons leak off, losing the stored data. For that reason, the programming voltage in the 27256 is scaled down to 12.5v (see "Scaling down the E-PROM cell," opposite). With that scaling, the 400-A oxides in the transistors of the peripheral circuitry that route the programming voltage to the array experience an electric field of 3.25 megavolts per centimeter, rough-

Scaling down the E-PROM cell

A two-step scaling-down process starting from the 64-K erasable programmable read-only memory (the 2764) has resulted in the 256-K chip designated the 27256 by Intel. The original 64-K E-PROM represented a modest scaling of the 16-K, using positive photoresists and projection-printer lithography. The first step beyond the 64-K chip called for wafer-stepping lithography for levels such as the first polysilicon and contact openings, where significant area could be saved without redesigning the devices or changing the other process steps. Those moves substantially reduced the size of the 64-K chip and ushered in a 128-K version.

At the same time, however, a program was under way to scale down all the design rules and use the stepper lithography to full advantage. That complete redesign of the process not only produced the 256-K array, but suggests that, from a device design perspective, 512-K and 1-megabit arrays will be feasible in the next two to three years. However, substantially better control over dimensions will have to accompany that continued scaling. Several variations were made upon the so-called classical scaling theory and can be understood with reference to the operation of the E-PROM cell.

The E-PROM cell is a simple modification of a conventional n-channel MOS enhancement-mode transistor whose drain connects to a bit line and whose source is grounded. The transistor's gate floats and is controlled by capacitive coupling to a polysilicon word line overlying the gate. Current conducted through the transistor is read as a logical 1, and the absence of current as a 0.

The cell is programmed to preserve a nonconducting state by applying high voltages to the word and bit lines simultaneously. Under those conditions, hot electrons are injected from the channel to the floating gate, charging it to a negative voltage. With the gate charged by the trapped electrons, the word line cannot couple enough voltage to it to turn on the transistor during a subsequent read operation. Exposing the cell to ultraviolet light elevates the trapped electrons' energy to the level of the conduction band of the surrounding oxide, and their mutual repulsion then causes them to flow off the gate.

The cell for the 256-K chip was derived from that of its 64-K predecessor using a scaling factor of 2. As shown in the table, the constant-field scaling theory was followed almost exactly for the device dimensions, oxide thicknesses, and programming voltage. Significant departures from the formulas were made in the read voltage, which was not scaled at all, and the channel-doping concentration, which was more than doubled (in fact almost quintupled). The doping was boosted this much to increase the electric fields in the channel and thereby improve the programming efficiency. Such an increase was feasible because the read voltage was not scaled: the threshold voltage is set as a fraction of the read voltage to ensure the optimal logic threshold, and it therefore could also remain unchanged.

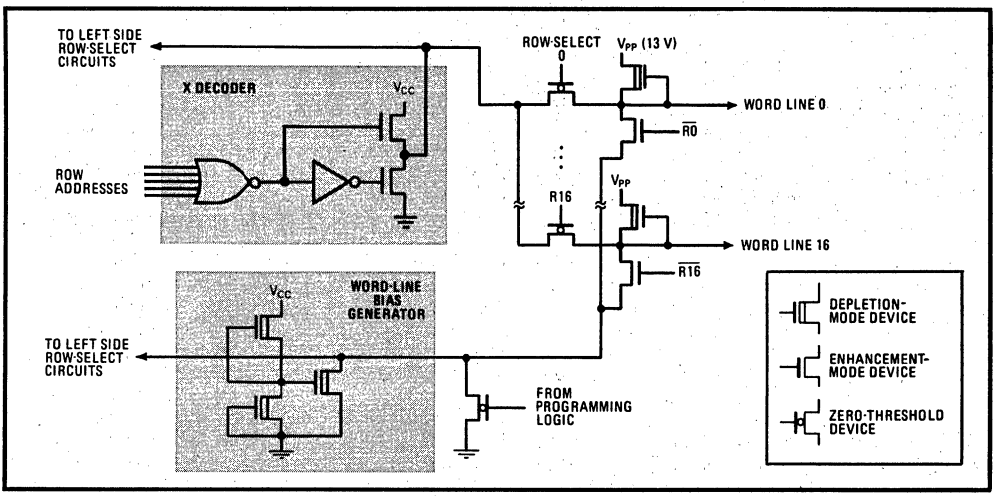
In the scaling theory, the read voltage would be scaled to prevent an increase in the electric field—and hence the stress—across the oxides. In an E-PROM, however, special treatment is required for the oxides, which see a worst-case field due to the high programming voltage. Thus, with the programming voltage scaled and oxides with the requisite integrity fabricated, the read voltage presents no problem. In fact, leaving the read voltage at 5 volts instead of scaling it to 2.5 V substantially increases the read current and speeds up access to the array.

Because of the high channel fields caused by the high dopant concentration, a 5-V drain voltage could suffice to inject hot electrons to the gate during a read, an effect appropriately known as parasitic programming. Therefore, the drain voltage was scaled down well below 5 V.

The devices in the peripheral circuitry were scaled from an effective length of 3.4 micrometers to 1.7 μm, and their oxides were thinned from 650 angstroms to 400 Å, which is equivalent to H-MOS II transistors. Those two changes increased the current of a minimum-length transistor by a factor of only 2.3: the increase in channel doping of the enhancement-mode devices degrades the electron mobility, limiting the current increase. Because the supply voltage was not scaled, the speed-power product of the technology is greater than it otherwise might have been, decreasing only by half to 0.5 picojoule.

SCALING DATA FOR THE 256-K ERASABLE PROGRAMMABLE READ-ONLY MEMORY

Parameter	64-K chip	Constant-field scaling formula	Theoretical 256-K chip	Actual scaling formula	Actual 256-K chip
Cell area	159 μm^2	K^{-2}	40 μm^2	$\approx K^{-2}$	36 μm^2
Floating-gate oxide	725 Å	K^{-1}	363 Å	$\approx K^{-1}$	325 Å
Control-gate oxide	900 Å	K^{-1}	450 Å	K^{-1}	450 Å
Channel doping	2×10^{16} cm^{-3}	K	4×10^{16} cm^{-3}	$K^{1.7}$	9×10^{16} cm^{-3}
Threshold voltage	1.6 V	K^{-1}	0.8 V	1	1.6 V
Cell current	55 μA	K^{-1}	28 μA	$K^{+0.6}$	80 μA
Read voltage	5 V	K^{-1}	2.5 V	1	5 V
Program voltage	21 V	K^{-1}	11 V	$\sim K^{-0.7}$	12.5 V



3. Biased decoder. With a 0.5-volt bias applied to deselected word lines during a read operation, depletion-mode pass transistors can be used in the decoder. They increase the selected word line's voltage, speeding access and extending the allowed power-supply range.

ly equal to the electric field found in previous parts.

The thin oxide between the substrate and floating gate illustrates the engineering that went into each of the process steps. Early in the development, that 325-Å-thick oxide showed defect densities of around 100/cm². (The oxide integrity is judged with a sensitive measurement of the dc current flowing through a large-area capacitor biased to 15 V.) Based on the total active area, A, of the channels of the 256-K chip's array transistors, a simple model for the yield, such as e^{-AD}, predicts the loss due to that defect density, D, at 54%, an unacceptably high figure for a single process step.

With careful adjustment of process parameters, the thin-oxide defect density was reduced below 5/cm², raising the yield of that step to an estimated 96%. Similarly, each new step could be fine-tuned independently of any other procedures, simplifying the process debugging. Despite the many new procedures, the overall structure and the sequence of steps do not depart radically from previous E-PROMs, a help in the retraining of manufacturing personnel.

In particular, the shared drain contact, common source diffusion, and self-aligned floating polysilicon gate of the E-PROM cell have not changed conceptually since the 2716 of five years ago. What has changed dramatically is the minimum feature size: the evolution from projection printing and wet etching to wafer-stepping lithography and anisotropic plasma etching on critical levels has shrunk the 27256's cell to the size of a contact hole on the old 2716.

Most elements of the E-PROM process are affected by that radical shrink. The field oxide that isolates adjacent diffusions is thinned to 0.6 μm, a move that cuts the length of the bird's beak in half. (The bird's beak, or transition region, between the oxide and the active device is wasted area.) As a result, the minimum spacing on the mask between adjacent field oxide regions is only 2 μm. The thinner oxide is possible thanks to the lower programming voltage, which reduces the chance of parasitic transistor action beneath the oxide.

The dose of boron implanted in the transistor channels to set their threshold voltage is increased, compensating for the effects of the shorter, narrower channels. In turn, the gate oxides are thinned to boost the transconductance of the more highly doped channels. The arsenic for the source and drain regions is implanted 0.5 μm or shallower to control the channel length and forestall punchthrough. To prevent the metalization from penetrating those shallow junctions, an aluminum-silicon alloy substitutes for the usual aluminum.

Although process development goes far to ensure a reliable part with good margins, new circuits were also required for the E-PROM's jump to 256-K density. For one, some statistical variation in cell characteristics is inevitable, and the more bits per chip, the greater the likelihood that one or more bits will fall outside the acceptable limits. Thus, circuits that compensate for these process-induced variations in effect raise yield. Sensing circuitry offers one example.

The 27256 uses two entire columns of reference cells, associating two cells with each of the 512 word lines rather than one with each of the eight comparators in the output section (Fig. 2). The tolerance to variations in read currents quintuples that in the 2764. Undoubtedly, future E-PROMs will extend this concept again, incorporating multiple reference columns.

Before the 2764, E-PROMs employed single-ended sense amplifiers, which are comparators that switch when the input current exceeds some threshold, called the trip current, which is not a function of any cell parameters. To make the trip current dependent on the read current itself, the 2764 used a differential comparator, which compares the read current with a fixed fraction of the current from a reference cell—a replica of

the array cells. If a cell's read current is lower than expected, then the reference cell's current will probably be low as well, adjusting the trip current accordingly. Such a circuit not only continues to operate under process variations, but also allows the design of a comparator optimized for speed.

Unfortunately, carrying that scheme even further on the 27256 and increasing the number of reference cells naturally leads to a wider distribution of their currents, and the conventional differential comparator could introduce an error because of that variation. To compensate for that error, a constant-current–ratio differential comparator was introduced. In it, the lower impedance reference load is obtained by connecting several transistors in parallel, each of which is an identical copy of the array load device (see Fig. 2 again). The parallel connection then gives the correct impedance without changing the transistor operating point, permitting correct operation over a much wider range of read current.

Extending a scheme

In the usual comparator, the dimensions of the load device on the reference cell or column are adjusted to give a lower impedance than that of the load device on the array column. The voltage drops across the different impedances are the differential input to the comparator. However, the different load-device dimensions change the devices' threshold voltages, putting the reference load at a different operating point. That difference means that the ratio of the array current to the reference current changes with variations in the reference current.

Another innovation, in the row-decoder circuits, helps speed access time and extend the power-supply variations the chip will tolerate. The scheme devised for the 27256 uses a negative-threshold pass transistor; however, it also incorporates a bias-voltage generator to hold the deselected word lines at about 0.5 V, rather than 0 V, during a read operation (Fig. 3). The bias current for the generator is supplied by pull-up transistors connected to the programming voltage supply. The 0.5-V shift suffices to reduce the leakage current without turning on deselected cells during a read operation.

However, during programming, when the high voltage applied to selected columns can couple to the floating gates, the 0.5-V bias would be enough to partially turn on deselected cells. Thus, a shunt transistor is included in the bias generator to pull the deselected word lines all the way to ground during programming.

In a conventional decoder, if a positive-threshold device is used, the word-line voltage rises very slowly above the level of one threshold below the supply voltage, slowing access to the cells. It also ups the minimum power-supply voltage by about 1 V. (The minimum supply voltage is the cell's threshold voltage, plus that voltage required to generate a detectable current difference between programmed and erased cells, typically about 3 V.)

Using a negative-threshold device overcomes both those effects, but introduces its own problems if not accompanied by a bias-voltage generator. With a depletion-mode pass transistor, the deselected devices still pass a significant leakage current. That current pulls down the decoder output below the supply voltage so that the selected word line still receives a reduced voltage.

TYPICAL PROGRAM SIZES FOR 256-K MEMORIES

Program	Bytes	Number of 256-K erasable programmable read-only memories
Basic interpreter	20-K to 32-K	5/8 to 1
Pascal compiler	32-K to 40-K	1 to 1 1/4
Asteroids	8-K	1/4
Screen-oriented editor	10-K	1/3
Word processor	32-K to 48-K	1 to 1 1/2
Spelling dictionary	12-K	3/8
Relational data base	32-K to 128-K	1 to 4

The 27256 also illustrates the utility of two circuit features trademarked as the intelligent Identifier and the intelligent Programming Algorithm. A persistent problem for the user of scaled-down E-PROM technology is the changing programming requirements. Each new generation forces customers to convert to lower voltages and altered algorithms. The identifier, an on-chip, unalterable, 2-byte code, specifies the chip's manufacturer, programming algorithm, voltage, and pulse width.

Saving progamming time

With ever denser E-PROM arrays, the programming algorithm proves its worth in saving programming time. This adaptive, closed-loop algorithm varies the width and the number of program pulses to reach an adequate margin in the minimum time, typically 4 milliseconds per byte for the 27256.

The old programming algorithm, developed for the 2716, used a fixed 50-ms pulse width. That width is determined by the worst-case programming time, that is, the longest time any bit in the array will need for complete programming. As with other device characteristics, process variations lead to a distribution of programming times; the fixed-width pulse must be long enough to program the slowest bit in the array. At the 256-K level, chances are good that at least one bit would need a very long programming pulse.

The closed-loop programming algorithm requires a method of determining the programmed cell "margin," or retention characteristics. Margin has generally been represented as the maximum supply voltage at which both programmed and erased cells can be read. A programmed cell is one in which the threshold voltage has been forced to a higher voltage. Increasing the supply voltage boosts the voltage applied to a cell's gate; eventually, the higher threshold is reached, turning on a cell that is intended to remain off.

Thus, the strategy in the programming algorithm is to verify that a cell is programmed at an elevated supply voltage of 6 V. The procedure begins by applying a 1-ms pulse to the first byte. At the end of the pulse, the chip tries to read the programmed cells. If they are programmed, an additional 3-ms pulse is applied, boosting the margin about 1.5 V. If the cell is not programmed, 1-ms pulses continue to be applied until programming is verified. When it is, a pulse three times as long as the sum of the 1-ms pulses already applied adds the required margin. To program a 27256 takes about 3 minutes. □

intel

ARTICLE REPRINT

AR-265

March 1983

Versatile Algorithm, Equipment Cut EPROM Programming Time

Don Knowlton
EPROM Product Line Manager
Intel Corp.

ORDER NUMBER: 230614-001

Versatile algorithm, equipment cut EPROM programming time

Programming high-density EPROMs for large-volume applications can entail significant time and expense. To minimize both, use programming software and hardware that recognize the different characteristics of individual EPROM cells.

Don Knowlton, Intel Corp

Using the capabilities of μP-based EPROM programming equipment and employing a new algorithm that recognizes differences among EPROM cells can dramatically reduce programming time for the newest high-density devices. The typical factor-of-six reduction this system can achieve results in considerable cost savings for large-volume applications. The time required to program an 8k-byte 2764, for example, drops from 7 min to an average of 1.25 min with the procedure. As a bonus, the technique helps ensure that the device receives adequate programming—in terms of memory-cell charge—to maintain long-term reliability.

Reducing programming time and costs for EPROMs has become increasingly important because the chips have become a cost-effective, easy-to-use alternative to masked ROM in high-volume applications requiring code flexibility or simplified inventory—a major switch from EPROMs' original small-volume prototyping applications. And volume usage makes EPROM programming a significant manufacturing consideration, subject to minimization efforts.

Higher densities increase programming time

Despite dramatic (eightfold) increases in EPROM storage capacity, early technology improvements (from p-channel MOS through 3-power-supply NMOS to single-power-supply NMOS) held EPROM programming times to less than 2 min (**Table 1**). Since the appearance of the 16k-bit EPROM, however, the minimum time for reliably programming each EPROM cell has remained constant at 45 msec, doubling the total device programming time with each doubling of density. The result is an increase in the per-unit cost of programming—involving either additional time (labor) or extra equipment.

The programming procedure currently in use for most EPROMs isn't yet a major burden on most users, though; it uses a nominal 50-msec pulse per EPROM byte, resulting in a total programming time of approximately 1.5 min for a 16k-bit chip. With the introduction of the 2764 (64k bits) and devices with even higher density, however, programming times have increased. A 256k-bit EPROM, for example, would require 24 min for programming by this conventional method.

TABLE 1— EPROM PROGRAMMING-TIME EVOLUTION

DEVICE	BYTES	PROGRAMMING TIME (MIN)
1702A	256	2.0
2708	1024	1.5
2716	2048	1.75
2732	4096	3.5
2764	8192	7.0
2764*	8192	1.25
27128*	16384	2.5

*USING ENHANCED PROGRAMMING ALGORITHM DESCRIBED IN THE TEXT.

Most EPROM cells program in much less than 45 msec, however. In fact, empirical data (**Fig 1**) shows that very few cells require longer than 8 msec for programming. Therefore, a procedure that takes into account the characteristics of individual EPROM cells can significantly reduce a device's programming time.

Arbitrarily reducing programming time is risky, though, because a cell's ability to achieve and maintain its programmed state is a function of this time (see **box,** "EPROM programming"). What's needed, therefore, is a way to verify the level to which individual cells have been programmed.

Fortunately, such techniques exist. By determining the charge stored in a cell relative to the minimum charge needed to program the cell to a detectable level, you can check for a program margin that assures reliable EPROM operation.

One way to check program margin involves varying the select-gate voltage, V_{CC}. And although you wouldn't

Reduce EPROM programming time with an intelligent algorithm

necessarily use this procedure in the actual programming of an EPROM, it serves to illustrate some EPROM characteristics and thus merits discussion.

An erased cell (having a ONE output) with no charge on its floating gate has a characteristic similar to an ordinary NMOS I-V curve (**Fig 2**). As you begin programming the cell (to a ZERO), the cell threshold as a function of select-gate voltage begins to increase. Thus, if you externally increase V_{CC}, connected to the select gate, you can determine cell threshold by observing the V_{CC} value at which a ZERO-to-ONE transition occurs on the output. If a transition occurs within V_{CC}'s normal operating range of 4.75 to 5.25V, the EPROM cell is inadequately programmed.

Some examples illustrate the process. First, consider a cell programmed for time t_1-t_0, storing a charge Q_1 on the floating gate:

$$Q_1 = \int_{t_0}^{t_1} i(t)\, dt.$$

If the charge isn't adequate to fully program the cell, the EPROM output exhibits a ZERO-to-ONE transition as V_{CC} increases through a level less than the 5.25V max operating voltage.

Now consider a second cell programmed for time t_2-t_0, storing a charge Q_2 on the floating gate. If increasing V_{CC} causes a ZERO-to-ONE transition at exactly 5.25V, the cell has exactly the amount of charge required for programming. However, the cell has no margin for tolerating charge loss or small variations in V_{CC} beyond the specification maximum, and such a device might not be reliable in long-term operation.

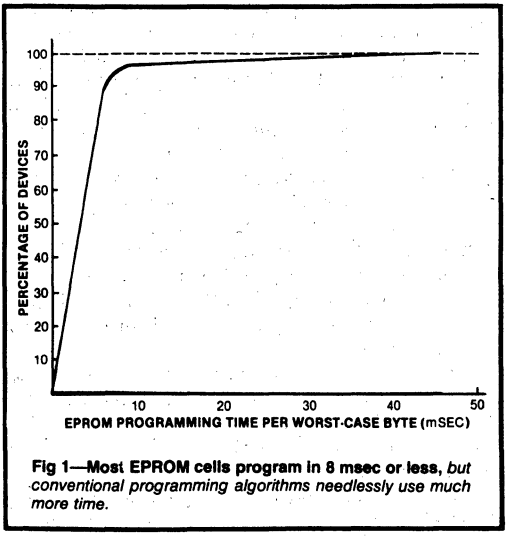


Fig 1—Most EPROM cells program in 8 msec or less, *but conventional programming algorithms needlessly use much more time.*

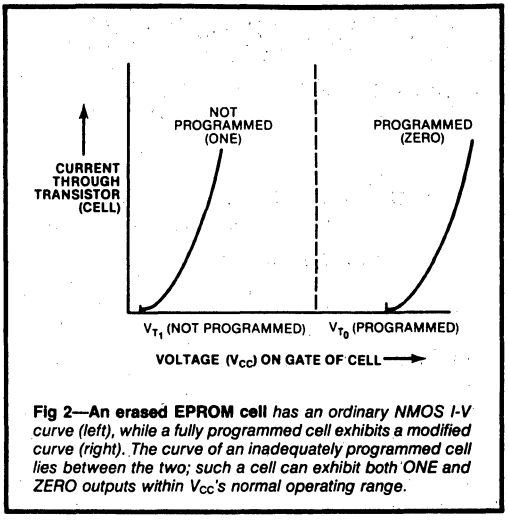


Fig 2—An erased EPROM cell *has an ordinary NMOS I-V curve (left), while a fully programmed cell exhibits a modified curve (right). The curve of an inadequately programmed cell lies between the two; such a cell can exhibit both ONE and ZERO outputs within V_{CC}'s normal operating range.*

Finally, consider a device programmed for time t_2+T-t_0. In this case, charge added by the additional programming time T increases programming margin. The ZERO-to-ONE transition detected while increasing V_{CC} thus occurs when V_{CC} is greater than 5.25V, providing assurance of reliability.

Margin checking doesn't occur in conventional EPROM programming, however. Instead, each EPROM cell receives a 45- to 55-msec write pulse, and manufacturers ensure program margin by screening out devices having bytes that don't program within 45 msec. This programming procedure is thus an open loop—no actual verification of margin occurs.

Improved algorithm uses closed-loop technique

In contrast, the trademarked inteligent Programming algorithm, a procedure devised by Intel for programming high-density EPROMs, guarantees reliability through the closed-loop technique of margin checking. It ensures that an EPROM cell's intended ZERO output won't become a ONE within V_{CC}'s normal operating range.

The algorithm begins by setting V_{CC} to 6V (higher than required for normal operation but necessary to provide programming margin), setting the programming voltage V_{PP} to 21V and then iteratively supplying 1-msec LOW-going programming pulses to the EPROM's program-disable pin $\overline{PGM}$. After each pulse, the algorithm checks the EPROM's output for the desired programmed value. If the output is incorrect, the algorithm repeats the pulse-and-check operation; incorrect output after 15 pulses causes rejection of the EPROM device.

If the EPROM is fully functional, however, one of the pulses results in proper EPROM output. At that point, the algorithm supplies still another programming pulse—this one four times longer than the combined length of the previously applied 1-msec pulses. This longer pulse helps ensure that the EPROM cell has adequate programming margin for reliable operation. Although the pulse can be as long as 60 msec, very few EPROM bytes require this much programming time. In fact, most EPROM bytes program with only one or two 1-msec pulses, so a typical total programming time per byte is 5 to 10 msec.

TABLE 2—
EPROM PROGRAMMER DESIGN RANGES

PARAMETER	MINIMUM	MAXIMUM
V_{CC}	4.5V	7.0V
V_{PP}	10.0V	25.0V
t_{PW}*	0.1 mSEC	60 mSEC

*PROGRAMMING-PULSE WIDTH

Use commercial programmers or design your own

Most commercial EPROM programming equipment can accommodate this programming algorithm with minor changes to hardware and software. For example, Data I/O Models 120A and 121A programmers require that firm's Revision D software; the company's Unipak, Unipak II and Mospak programmers require Revisions 004, 001 and 003, respectively. You need Revision B software from Pro-Log to use the algorithm on that firm's M980 control unit; you also need the PM9080 module and PA28-80 socket adapter.

If you design your own equipment to implement the algorithm, you must consider several application factors in weighing performance and flexibility against cost. For instance, if your application is dedicated to programming only the 2764, you can construct a system totally in hardware to minimize costs. You need only three fixed-voltage power supplies (5, 6 and 21V), and you can generate pulses with simple devices such as one-shots. Logic arrays or a μP can perform the counting and algorithm sequencing.

Systems that program several types of EPROMs require a more general approach; they're best implemented with a μP and programmable power supplies. Look-up tables in the program-store memory can access algorithms for various EPROMs in this type of system, and tables can contain pulse widths and voltage parameters. Most parameters are common to many EPROMs, so programming the various types is easy.

Moreover, you can employ software in such systems to count the 1-msec pulses and construct various other pulse widths to satisfy the programming algorithm's timing requirements.

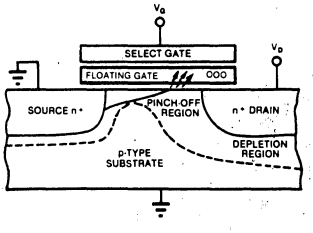
Supply-voltage generation, however, proves more difficult. An elegant and flexible solution involves using software-controlled programmable-output power supplies to deliver V_{PP} and V_{CC}, thus providing compatibility with future designs.

As an alternative, you could employ selectable supplies. Unfortunately, this less costly approach might

EPROM programming

Information storage in an EPROM results from electrically charging a floating gate in a stacked-gate MOS transistor. To charge the gate, you raise both the select-gate voltage V_G and the drain voltage V_D to a high positive level while holding source and substrate at ground potential **(figure);** suitable drain and gate voltages cause the transistor to operate in saturation.

The acceleration of electrons in the transistor's pinch-off region results from the high electric field. Some of these electrons acquire enough energy to enter the conduction band of the silicon dioxide (which electrically isolates the floating gate under normal operating voltages) and get attracted to the positive potential of the floating gate.



Information storage in an EPROM cell *results from the transfer of charge to an MOS transistor's floating gate during the programming operation.*

The basic equation describing the behavior of cell programming is:

$$i = C_{FG}\frac{dV}{dt},$$

where C_{FG} is the floating-gate capacitance. The charge accumulated on the floating gate is therefore:

$$Q = \int_{t_0}^{t_1} i(t)\,dt = C_{FG}V.$$

Thus, the amount of charge stored in the cell and the resultant level of programming are functions of the cell programming time $t_1 - t_0$.

Get a factor-of-six speedup with individual-cell programming

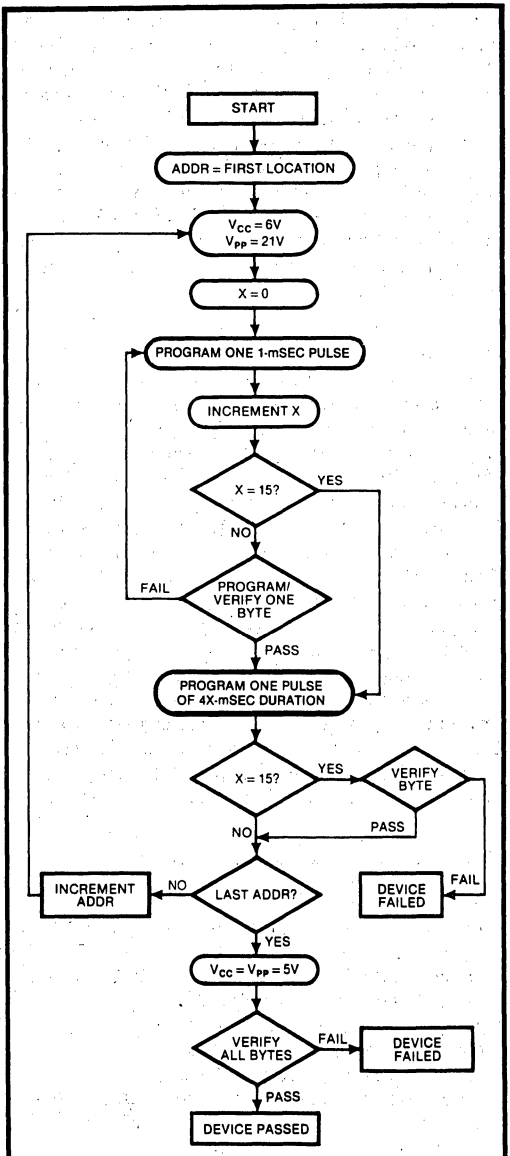
leave the equipment lacking the voltage required for programming future devices. **Table 2,** however, shows suggested voltage and timing ranges for equipment design that should meet anticipated requirements for Intel MOS EPROMs.

Equipment designed to program several EPROMs in parallel requires special design considerations. Devices that fail to program slow down the entire programming operation if each address is allowed to program for the maximum time (pulse counter=15). A ganged unit, however, should detect programming failures immediately and disable faulty devices to minimize programming time for the other EPROMs. This step is particularly critical in high-volume programming.

Good design practices help

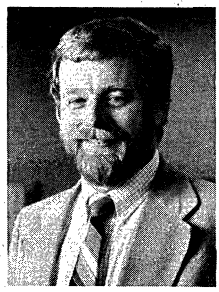
Furthermore, following many of the design rules used in dynamic-RAM applications can help you improve the reliability of high-density MOS-EPROM programmers. For example,

- Provide well-regulated power supplies at the EPROM. Short, low-inductance traces, high-frequency capacitive decoupling and ground-plane routing all reduce the effects of current surges and are necessary for high speed.
- Make sure that all signals are clean. Minimize undershoot, overshoot and glitches. Use an oscilloscope with at least a 150-MHz bandwidth to find potential problems.
- Observe absolute maximum specs on V_{PP}. Because V_{PP} is the highest voltage applied to the chip, it's usually the closest voltage to an EPROM's physical limits. Even small glitches exceeding absolute maximum ratings can result in chip destruction, so use a high-bandwidth oscilloscope to look for overshoot when setting V_{PP}. EDN



Intelligent programming algorithms, *like the one for Intel's 27128 described by this flowchart, cut programming time by more than 80%.*

Author's biography



Don Knowlton is EPROM product-line manager at Intel's Nonvolatile Memory Div (Santa Clara, CA), where he's responsible for EPROM strategic marketing. Before joining Intel, he worked at Advanced Micro Devices. Don holds BSEE and MSEE degrees, both from Purdue University. In his spare time, he enjoys playing 5-string banjo.

2732A
32K (4K x 8) UV ERASABLE PROM

- **200 ns (2732A-2) Maximum Access Time . . . HMOS*-E Technology**
- **Compatible with High-Speed 8mHz iAPX 186...Zero WAIT State**
- **Two Line Control**
- **Compatible with 12 MHz 8051 Family**
- **Industry Standard Pinout . . . JEDEC Approved**
- **Low Standby Current...30 mA Maximum**
- **$\pm 10\%$ V_{CC} Tolerance Available**
- **int$_e$ligent Identifier™ Mode**
- **TTL Compatible**

The Intel 2732A is a 5V only, 32,768 bit ultraviolet erasable and electrically programmable read-only-memory (EPROM). The standard 2732A access time is 250 ns with speed selection (2732A-2) available at 200 ns. The access time is compatible with high performance microprocessors such as the 8 MHz iAPX 186. In these systems, the 2732A allows the microprocessor to operate without the addition of WAIT states.

An important 2732A feature is the separate output control, Output Enable ($\overline{OE}$), from the Chip Enable control ($\overline{CE}$). The $\overline{OE}$ control eliminates bus contention in microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's EPROMs. AP-72 is available from Intel's Literature Department.

The 2732A has a standby mode which reduces power consumption without increasing access time. The maximum active current is 125 mA, while the maximum standby current is only 35 mA, a 70% saving. The standby mode is selected by applying the TTL-high signal to the $\overline{CE}$ input.

The 2732A is fabricated with HMOS*-E technology, Intel's high-speed N-channel MOS Silicon Gate Technology.

*HMOS is a patented process of Intel Corporation.

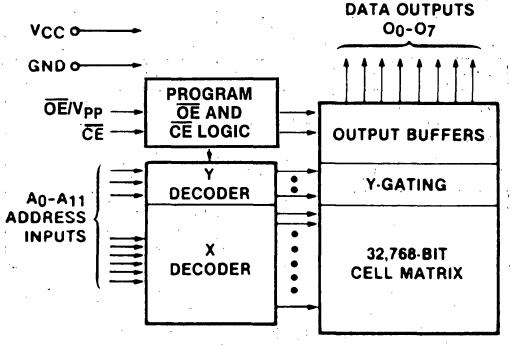


Figure 1. Block Diagram

PIN NAMES

A_0-A_{11}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}/V_{PP}$	OUTPUT ENABLE/ V_{PP}
O_0-O_7	OUTPUTS

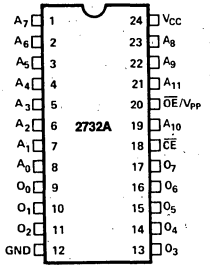


Figure 2. Pin Configuration

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +125°C
All Input or Output Voltages with Respect to Ground ..... +6V to −0.3V
Voltage on Pin 22 with Respect to Ground ..... +13.5V to −0.3V
V_{PP} Supply Voltage with Respect to Ground During Programming ..... +22V to −0.3V

NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	2732A/A-2/A-3/A-4	2732A-20/A-25/A-30
Operating Temperature Range	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ± 5%	5V ± 10%

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits Min.	Limits Typ.[1]	Limits Max.	Units	Conditions
I_{IL}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{CC1}[2]	V_{CC} Current (Standby)			35	mA	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC2}[2]	V_{CC} Current (Active)			100	mA	$\overline{OE} = \overline{CE} = V_{IL}$
V_{IL}	Input Low Voltage	−0.1		0.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	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}[2]	V_{PP} Read Voltage	3.8		V_{CC}	V	$V_{CC} = 5.0V \pm 0.25V$

A.C. CHARACTERISTICS

Symbol	Parameter	2732A-2 2732A-20 Min.	2732A-2 2732A-20 Max.	2732A 2732A-25 Min.	2732A 2732A-25 Max.	2732A-3 2732A-30 Min.	2732A-3 2732A-30 Max.	2732A-4 Min.	2732A-4 Max.	Units	Test Conditions†
t_{ACC}	Address to Output Delay		200		250		300		450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		200		250		300		450	ns	$\overline{OE} = V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		70		100		150		150	ns	$\overline{CE} = V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Not Driven	0	60	0	60	0	130	0	130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		ns	$\overline{CE} = \overline{OE} = V_{IL}$

†A.C. TEST CONDITIONS

Output Load ..... 1 TTL gate and $C_L = 100$ pF
Input Rise and Fall Times ..... ≤ 20 ns
Input Pulse Levels ..... 0.45V to 2.4V

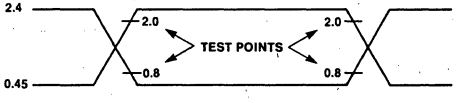
Timing Measurement Reference Level:
Inputs ..... 0.8 and 2.0V
Outputs ..... 0.8 and 2.0V

NOTES: 1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for $t_A = 25°C$ and nominal supply voltages.
4. 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

CAPACITANCE[2] (T_A = 25°C, f = 1 MHz)

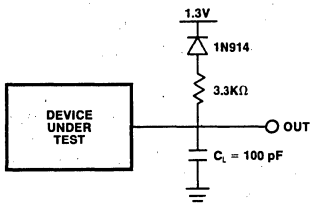
Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN1}	Input Capacitance Except $\overline{OE}/V_{PP}$	4	6	pF	V_{IN} = 0V
C_{IN2}	$\overline{OE}/V_{PP}$ Input Capacitance		20	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

A.C. TESTING INPUT/OUTPUT WAVEFORM



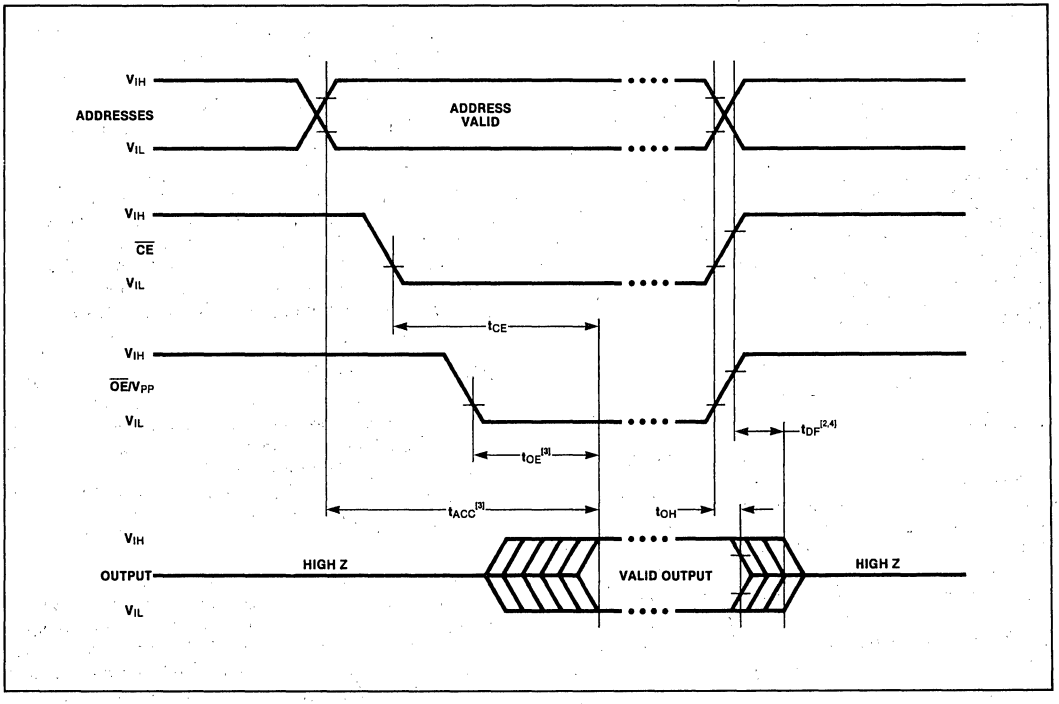
A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0."

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



ERASURE CHARACTERISTICS

The erasure characteristics of the 2732A are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2732A in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2732A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 2732A window to prevent unintentional erasure.

The recommended erasure procedure for the 2732A is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000μW/cm^2 power rating. The 2732A should be placed within 1 inch of the lamp tubes during erasure.

DEVICE OPERATION

The six modes of operation of the 2732A are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for $\overline{OE}/V_{PP}$ during programming and 12V on A$_9$ for the int$_e$ligent Identifier™ mode. In the program mode the $\overline{OE}/V_{PP}$ input is pulsed from a TTL level to 21V.

Table 1. Mode Selection

MODE \ PINS	$\overline{CE}$ (18)	$\overline{OE}/V_{PP}$ (20)	A_9 (22)	V_{CC} (24)	OUTPUTS (9-11,13-17)
Read	V_{IL}	V_{IL}	X	+5	D_{OUT}
Output Disable	V_{IL}	V_{IH}	X	+5	High Z
Standby	V_{IH}	X	X	+5	High Z
Program	V_{IL}	V_{PP}	X	+5	D_{IN}
Program Inhibit	V_{IH}	V_{PP}	X	+5	High Z
Int$_e$ligent Identifier	V_{IL}	V_{IL}	V_H	+5	Code

Notes: 1. X can be V_{IH} or V_{IL}
2. V_H = 12.0 ±0.5V

Read Mode

The 2732A has two control functions, both of which must be logically active in order 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. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

The 2732A has a standby mode which reduces the maximum active current from 125 mA to 35 mA. The 2732A is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays, Intel has provided a 2 line control function that accommodates this use of multiple memory connection. The two line control function allows for:

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

To use these two control lines most efficiently, $\overline{CE}$ (pin 18) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

PROGRAMMING

CAUTION: Exceeding 22V on Pin 20 ($\overline{OE}/V_{PP}$) will permanently damage the 2732A.

Initially, and after each erasure, all bits of the 2732A are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2732A is in the programming mode when the $\overline{OE}/V_{PP}$ input is at 21V. It is required that a 0.1 μF capacitor be placed across $\overline{OE}/V_{PP}$ and ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec, active low, TTL program pulse is applied to the $\overline{CE}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time —either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2732A must not be programmed with a DC signal applied to the $\overline{CE}$ input.

Programming of multiple 2732As in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2732As may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{CE}$ input programs the paralleled 2732As.

Program Inhibit

Programming of multiple 2732As in parallel with different data is also easily accomplished. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 2732As may be common. A TTL level program pulse applied to a 2732A's $\overline{CE}$ input with $\overline{OE}/V_{PP}$ at 21V will program that 2732A. A high level $\overline{CE}$ input inhibits the other 2732As from being programmed.

Verify

A verify (Read) should be performed on the programmed bits to determine that they were correctly programmed. The verify is accomplished with $\overline{OE}/V_{PP}$ and $\overline{CE}$ at V_{IL}. Data should be verified t_{DV} after the falling edge of $\overline{CE}$.

int$_e$ligent Identifier™ Mode

The int$_e$ligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ±5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 22) of the 2732A. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 8) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during int$_e$ligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 2732A, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

Intel began manufacturing 2732As during 1982 that contained the int$_e$ligent Identifier feature. Earlier generation devices do not contain identifier information, and if erased, will respond with a "one" (V_{OH}) on each data line when operated in this mode. Programmed, preidentifier mode 2732As will respond with the current data contained in locations 0 and 1 when subjected to the int$_e$ligent Identifier operation.

System Consideration

The power switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note, AP-72, and by properly selected decoupling capacitors. It is recommended that a 0.1μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effects of PC board-traces.

Table 2. 2732A int$_e$ligent Identifier™ Bytes

Pins / Identifier	A_0 (8)	O_7 (17)	O_6 (16)	O_5 (15)	O_4 (14)	O_3 (13)	O_2 (11)	O_1 (10)	O_0 (9)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	0	0	0	0	0	0	0	1	01

PROGRAMMING[4]

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.	Max.		
I_{LI}	Input Current (All Inputs)			10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{OL}	Output Low Voltage During Verify			0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage During Verify	2.4			V	$I_{OH} = -400$ μA
I_{CC}	V_{CC} Supply Current		85	100	mA	
V_{IL}	Input Low Level (All Inputs)	−0.1		0.8	V	
V_{IH}	Input High Level (All Inputs Except $\overline{OE}/V_{PP}$)	2.0		V_{CC}	V	
I_{PP}	V_{PP} Supply Current			30	mA	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{PP}$
V_{ID}	Ag int$_e$ligent Identifier Voltage	11.5		12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits			Units	Test Conditions†
		Min.	Typ.	Max.		
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_{OEH}	$\overline{OE}$ Hold Time	2			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}	$\overline{OE}$ High to Output Not Driven	0		130	ns	
t_{DV}	Data Valid from $\overline{CE}$			1	μs	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{IL}$
t_{PW}	$\overline{CE}$ Pulse Width During Programming	20	50	55	ms	
t_{PRT}	$\overline{OE}$ Pulse Rise Time During Programming	50			ns	
t_{VR}	V_{PP} Recovery Time	2			μs	

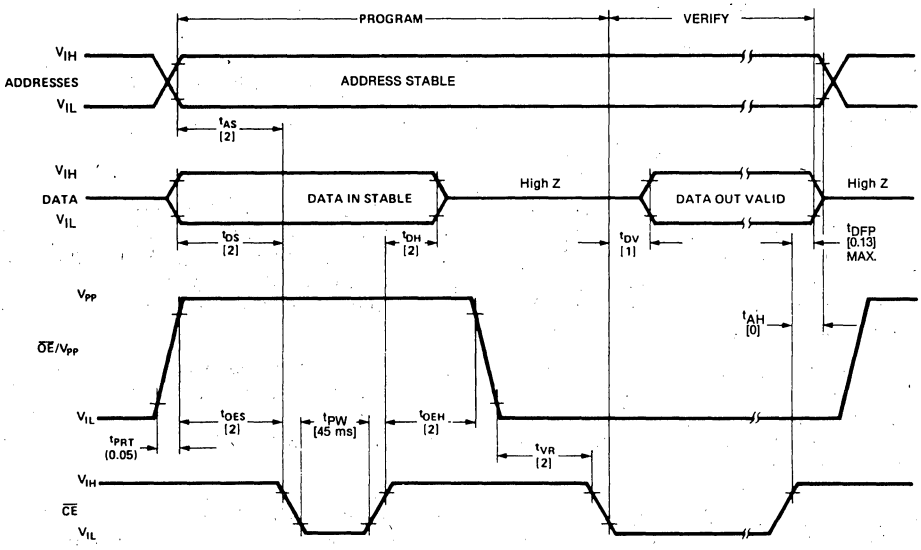
†A.C. TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ..... ≤20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level .....0.8V and 2.0V

NOTES:

1. Typical values are for $T_A = 25°C$ and nominal supply voltages.
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. OE may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of CE without impacting t_{ACC}.
4. When programming the 2732A, a 0.1μF capacitor **is required** across OE/V_{PP} and ground to suppress spurious voltage transients which may damage the device.

PROGRAMMING WAVEFORMS



NOTES:
1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS 0.8V FOR A V_{IL} AND 2V FOR A V_{IH}.

P2732A
32K (4K x 8) PRODUCTION EPROM

- **200 ns (P2732A-2) Maximum Access Time . . . HMOS*-E Technology**
- **Compatible with High-Speed 8 MHz iAPX 186 . . . Zero WAIT State**
- **Two Line Control**
- **Compatible with 12 MHz 8051 Family**
- **Industry Standard Pinout . . . JEDEC Approved**
- **Low Active Current . . . 100 mA Max.**
- **int$_e$ligent Identifier™ Mode**
- **Fast 20 ms Programming Time**
- **TTL Compatible**

The Intel P2732A is a 5V-only, Electrically Programmable Read-Only Memory in a plastic package. One time programmable, it has been designed for high volume production environments where a programmable memory is required for flexibility. The standard P2732A access time is 250 ns with speed selection (P2732A-2) available at 200 ns. The access time is compatible with high performance microprocessors such as the 8 MHz iAPX 186. In these systems, the P2732A allows the microprocessor to operate without the addition of WAIT states.

The P2732A is ideal for volume production environments where inventory and lead time risks occur for program codes. Inventoried in the unprogrammed state, the P2732A is programmed quickly and efficiently when the need to change code arises. Costs incurred for new ROM masks or obsoleted ROM inventories are avoided. The tight package dimensional controls, inherent non-erasability, and high reliability of the P2732A make it the ideal component for these production applications.

Using Intel's HMOS*-E technology, low power consumption combined with high speed data access are achieved. The maximum P2732A active current is 100 mA, while standby is only 35mA. The standby mode is selected by applying a TTL-high signal to the $\overline{CE}$ input.

PIN NAMES

A_0-A_{11}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}/V_{PP}$	OUTPUT ENABLE/V_{PP}
O_0-O_7	OUTPUTS

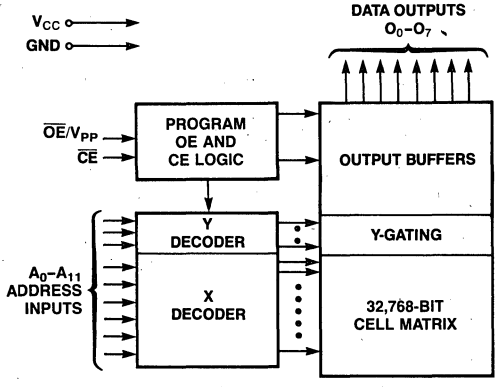


Figure 1. Block Diagram

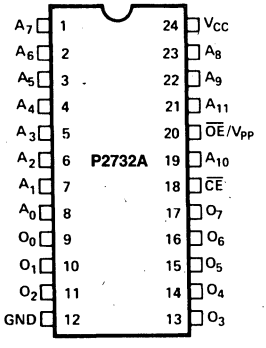


Figure 2. Pin Configuration

*HMOS is a patented process of Intel Corporation.

OCTOBER 1983
ORDER NUMBER: 230724-002

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +125°C
All Input or Output Voltages with Respect to Ground ..... +6V to −0.3V
Voltage on Pin 22 with Respect to Ground ..... +13.5V to −0.3V
V_{PP} Supply Voltage with Respect to Ground During Programming ..... +22V to −0.3V

NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	P2732A/A-2/A-3/A-4
Operating Temperature Range	0°C–70°C
V_{CC} Power Supply[1,2]	5V ± 5%

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Conditions
		Min.	Typ.[1]	Max.		
I_{IL}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{CC1}[2]	V_{CC} Current (Standby)		15	35	mA	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC2}[2]	V_{CC} Current (Active)		60	100	mA	$\overline{OE} = \overline{CE} = V_{IL}$
V_{IL}	Input Low Voltage	−0.1		0.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400\ \mu A$
V_{PP}[2]	V_{PP} Read Voltage	3.8		V_{CC}	V	$V_{CC} = 5.0V \pm 0.25V$

A.C. CHARACTERISTICS

Symbol	Parameter	P2732A-2		P2732A		P2732A-3		P2732A-4		Units	Test Conditions†
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.		
t_{ACC}	Address to Output Delay		200		250		300		450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		200		250		300		450	ns	$\overline{OE} = V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		70		100		150		150	ns	$\overline{CE} = V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Float	0	60	0	60	0	105	0	130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		ns	$\overline{CE} = \overline{OE} = V_{IL}$

†A.C. TEST CONDITIONS

Output Load ..... 1 TTL gate and $C_L = 100$ pF
Input Rise and Fall Times ..... ≤20 ns
Input Pulse Levels ..... 0.45V to 2.4V

Timing Measurement Reference Level:
Inputs ..... 0.8 and 2.0V
Outputs ..... 0.8 and 2.0V

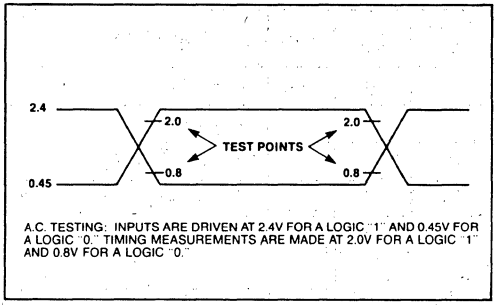
NOTES:

1. Typical values are for $T_A = 25°C$ and nominal supply voltages.
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 on page 4-18
3. $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impacting t_{ACC}.
4. When programming the 2732A, a 0.1μF capacitor **is required** across $\overline{OE}/V_{PP}$ and ground to suppress spurious voltage transients which may damage the device.

CAPACITANCE[2] (T_A = 25°C, F = 1 MHz)

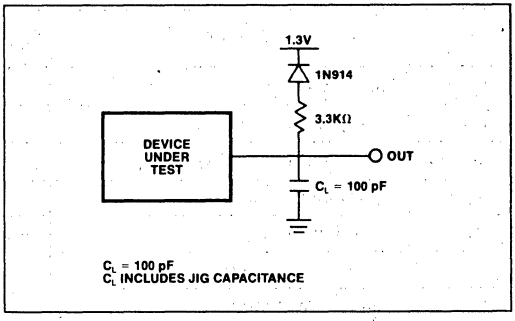
Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN1}	Input Capacitance Except $\overline{OE}/V_{PP}$	4	6	pF	V_{IN} =0V
C_{IN2}	$\overline{OE}/V_{PP}$ Input Capacitance		20	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

A.C. TESTING INPUT/OUTPUT WAVEFORM



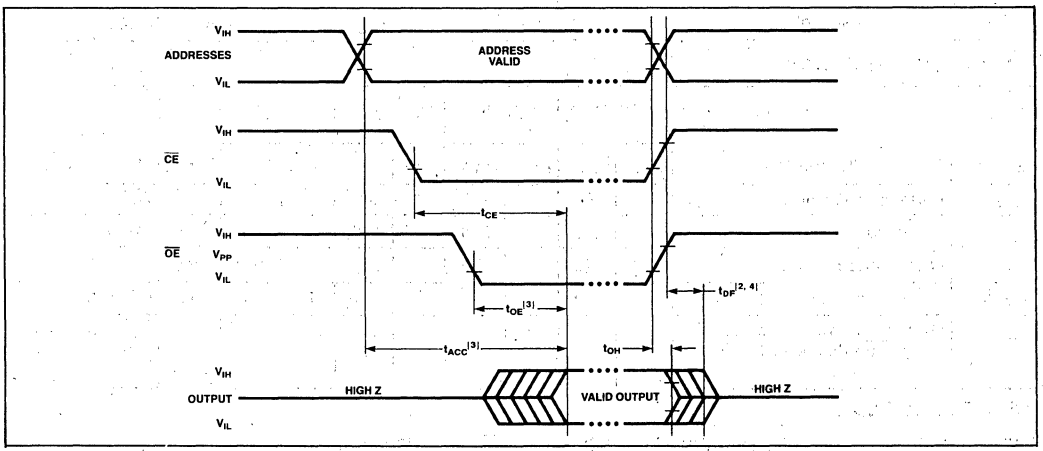
A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0". TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0".

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



NOTES:
1. Typical values are for T_A = 25°C and nominal supply voltages.
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 on this page.
3. $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impacting t_{ACC}.
4. When programming the P2732A, a 0.1μF capacitor is required across $\overline{OE}/V_{PP}$ and ground to suppress spurious voltage transients which may damage the device.

APPLICATIONS INFORMATION

Intel's P2732A is the result of a multi-year effort to make EPROMs more cost effective for production applications. The benefits of a plastic package enable the P2732A to be used for high volume production with lower profile boards and easier production assembly (no cover over UV transparent windows).

The reliability of plastic EPROMs is equivalent to traditional CERDIP packaging. The plastic is rugged and durable making it optimal for auto insertion and auto handling equipment. Design and testing ensures device programmability, data integrity, and impermeability to moisture.

Intel's Plastic EPROMs are designed for total compatibility with their CERDIP packaged predecessors. This encompasses quality, reliability, and programming. All Intel Plastic EPROMs have passed Intel's strict process and product reliability qualifications.

DEVICE OPERATION

The six modes of operation of the P2732A are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for $\overline{OE}/V_{PP}$ during programming and 12V on A_9 for the int$_e$ligent Identifier™ mode. In the program mode the $\overline{OE}/V_{PP}$ input is pulsed from a TTL level to 21V.

Table 1. Mode Selection

MODE \ PINS	$\overline{CE}$ (18)	$\overline{OE}/V_{PP}$ (20)	A_9 (22)	V_{CC} (24)	OUTPUTS (9-11,13-17)
Read	V_{IL}	V_{IL}	X	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	X	V_{CC}	High Z
Standby	V_{IH}	X	X	V_{CC}	High Z
Program	V_{IL}	V_{PP}	X	V_{CC}	D_{IN}
Program Inhibit	V_{IH}	V_{PP}	X	V_{CC}	High Z
Int$_e$ligent Identifier	V_{IL}	V_{IL}	V_H	V_{CC}	Code

NOTES:
1. X can be V_{IH} or V_{IL}
2. V_H = 12V ± 0.5V

Read Mode

The P2732A has two control functions, both of which must be logically active in order 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 gather data from the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

Standby Mode

The P2732A has a standby mode which reduces the maximum active current from 100 mA to 35 mA. The P2732A is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

TWO LINE OUTPUT CONTROL

Because EPROMs are usually used in larger memory arrays, Intel has provided two control lines which accommodate this multiple memory connection. The two control lines allow for:

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

To use these two control lines most efficiently, $\overline{CE}$ (pin 18) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

System Consideration

The power switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's two line control and by use of properly selected decoupling capacitors. It is recommended that a 0.1μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effects of PC board traces.

Application Note AP-72, which is available from Intel's Literature Department, describes microprocessor system implementation of the two-line control function on Intel's EPROMs.

PROGRAMMING

CAUTION: Exceeding 22V on Pin 20 ($\overline{OE}/V_{PP}$) will permanently damage the P2732A.

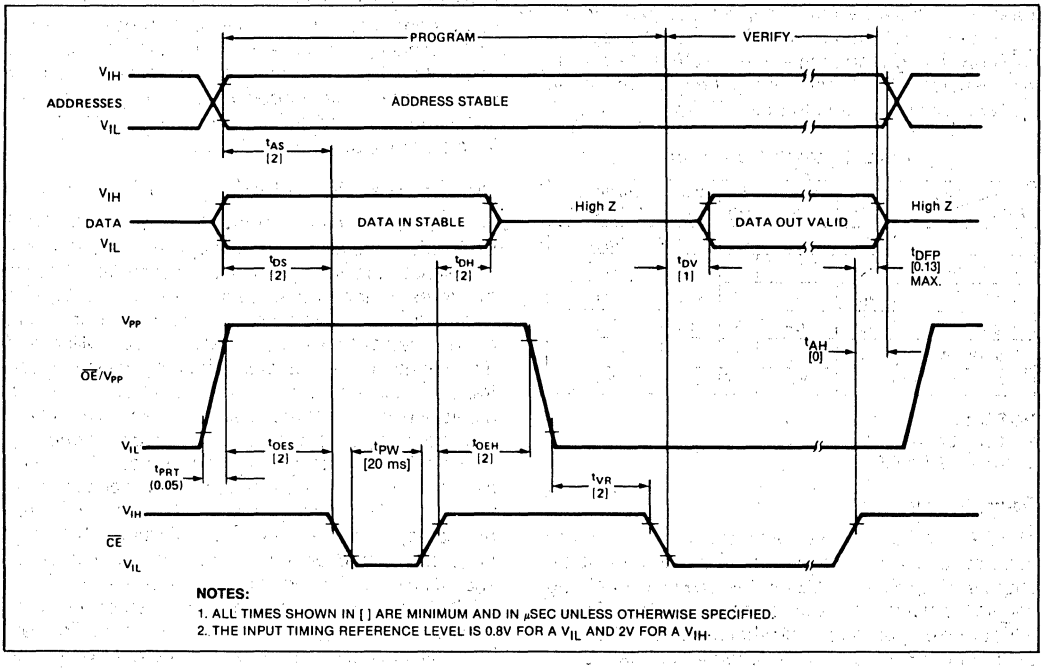
Initially all bits of the P2732A are in the "1" state. Data is introduced by selectively programming "0's" into the bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be present in the data word.

The P2732A is in the programming mode when the $\overline{OE}/V_{PP}$ input is at 21V. It is required that a 0.1μF capacitor be placed across $\overline{OE}/V_{PP}$ and ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 20ms active low, TTL program pulse is applied to the $\overline{CE}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time—either individually, sequentially, or at random. The program pulse has a maximum width of 55ms. The P2732A must not be programmed with a DC signal applied to the $\overline{CE}$ input.

Programming of multiple P2732As in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled P2732As may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{CE}$ input programs the paralleled P2732A.

PROGRAMMING WAVEFORMS



NOTES:

1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS 0.8V FOR A V_{IL} AND 2V FOR A V_{IH}.

PROGRAMMING[1]

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.	Max.		
I_{LI}	Input Current (All Inputs)			10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{OL}	Output Low Voltage During Verify			0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage During Verify	2.4			V	$I_{OH} = -400$ μA
I_{CC}	V_{CC} Supply Current		85	100	mA	
V_{IL}	Input Low Level (All Inputs)	−0.1		0.8	V	
V_{IH}	Input High Level (All Inputs Except OE/V_{PP})	2.0		$V_{CC}+1$	V	
I_{PP}	V_{PP} Supply Current			30	mA	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{PP}$
V_{ID}	Ag int$_e$ligent Identifier Voltage	11.5		12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits			Units	Test Conditions†
		Min.	Typ.	Max.		
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_{OEH}	$\overline{OE}$ Hold Time	2			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}	$\overline{OE}$ High to Output Not Driven	0		130	ns	
t_{DV}	Data Valid from $\overline{CE}$			1	μs	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{IL}$
t_{PW}	$\overline{CE}$ Pulse Width During Programming	20	50	55	ms	
t_{PRT}	$\overline{OE}$ Pulse Rise Time During Programming	50			ns	
t_{VR}	V_{PP} Recovery Time	2			μs	

A.C. TEST CONDITIONS

Input Rise and Fall Time (10% to 90%) ... ≤ 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTE:
1. When programming the P2732A, a 0.1μF capacitor is required across $\overline{OE}/V_{PP}$ and ground to suppress spurious voltage transients which may damage the device.

Program Inhibit

Programming of multiple P2732As in parallel with different data is also easily accomplished. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel P2732As may be common. A TTL low level pulse applied to a P2732A's $\overline{CE}$ input with $\overline{OE}/V_{PP}$ at 21V will program that P2732A. A high level $\overline{CE}$ input inhibits the other P2732As from being programmed.

Verify

A verify (Read) should be performed on the programmed bits to determine that they were correctly programmed. The verify is accomplished with $\overline{OE}/V_{PP}$ and $\overline{CE}$ at V_{IL}. Data should be verified t_{DV} after the falling edge of $\overline{CE}$.

inteligent Identifier™ Mode

The inteligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ±5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 22) of the P2732A. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 8) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during inteligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1(A0 = V_{IH}) the device identifier code. For the Intel P2732A, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

Table 2: P2732A inteligent Identifier™ Bytes

Identifier \ Pine	A_8 (8)	O_7 (17)	O_6 (16)	O_5 (15)	O_4 (14)	O_3 (13)	O_2 (11)	O_1 (10)	O_0 (9)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	0	0	0	0	0	0	0	1	01

2764
64K (8K x 8) UV ERASABLE PROM

- **200 ns (2764-2) Maximum Access Time . . . HMOS*-E Technology**
- **Compatible with High-Speed 8mHz iAPX 186...Zero WAIT State**
- **Two Line Control**
- **Pin Compatible to 27128 EPROM**
- **int$_e$ligent Programming™ Algorithm**
- **Industry Standard Pinout . . . JEDEC Approved**
- **Low Active Current...100mA Max.**
- **TTL Compatible**

The Intel 2764 is a 5V only, 65,536-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The standard 2764 access time is 250 ns with speed selection available at 200 ns. The access time is compatible with high-performance microprocessors such as Intel's 8 mHz iAPX 186. In these systems, the 2764 allows the microprocessor to operate without the addition of WAIT states. The 2764 is also compatible with the 12 MHz 8051 family.

An important 2764 feature is the separate output control, Output Enable ($\overline{OE}$) from the Chip Enable control ($\overline{CE}$). The $\overline{OE}$ control eliminates bus contention in microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's EPROMs. AP-72 is available from Intel's Literature Department.

The 2764 has a standby mode which reduces power consumption without increasing access time. The maximum active current is 100 mA, while the maximum standby current is only 40 mA. The standby mode is selected by applying a TTL-high signal to the $\overline{CE}$ input.

$\pm 10\%$ V_{CC} tolerance is available as an alternative to the standard $\pm 5\%$ V_{CC} tolerance for the 2764. This can allow the system designer more leeway with regard to his power supply requirements and other system parameters.

The 2764 is fabricated with HMOS*-E technology, Intel's high-speed N-channel MOS Silicon Gate Technology.

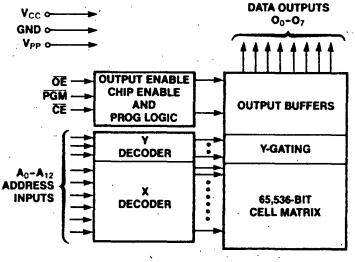
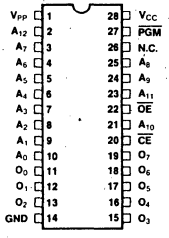


Figure 1. Block Diagram

27256	27128	2732A	2716
V_{PP}	V_{PP}		
A_{12}	A_{12}		
A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2
Gnd	Gnd	Gnd	Gnd

2764



2716	2732A	27128	27256
		V_{CC}	V_{CC}
		$\overline{PGM}$	A_{14}
V_{CC}	V_{CC}	A_{13}	A_{13}
A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$
A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3

NOTE: INTEL "UNIVERSAL SITE"-COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE 2764 PINS

Figure 2. Pin Configurations

MODE SELECTION

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13, 15-19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
int$_e$ligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
int$_e$ligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ±0.5V

PIN NAMES

A_0-A_{12}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
$\overline{PGM}$	PROGRAM
N.C.	NO CONNECT

*HMOS is a patented process of Intel Corporation.

OCTOBER 1983
ORDER NUMBER: 210570-003

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .....−10°C to +80°C
Storage Temperature .....−65°C to +125°C
All Input or Output Voltages with Respect to Ground ..... +7.0V to −0.6V
Voltage on Pin 28 with Respect to Ground .....+13.5V to −0.6V
V_{PP} Supply Voltage with Respect to Ground During Programming .....+22V to −0.6V

NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	2764-2	2764	2764-3	2764-4	2764-25	2764-30	2764-45
Operating Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 10%	5V ± 10%	5V ± 10%
V_{PP} Voltage[2]	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits				Conditions
		Min	Typ[3]	Max	Unit	
I_{LI}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{PP1}[2]	V_{PP} Current Read			5	mA	$V_{PP} = 5.5V$
I_{CC1}[2]	V_{CC} Current Standby			40	mA	$\overline{CE} = V_{IH}$
I_{CC2}[2]	V_{CC} Current Active		70	100	mA	$\overline{CE} = \overline{OE} = V_{IL}$
V_{IL}	Input Low Voltage	−.1		+.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400$ μA

A.C. CHARACTERISTICS

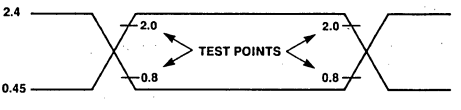
Symbol	Parameter	2764-2 Limits		2764-25 & 2764 Limits		2764-30 & 2764-3 Limits		2764-45 & 2764-4 Limits		Unit	Test Conditions
		Min	Max	Min	Max	Min	Max	Min	Max		
t_{ACC}	Address to Output Delay		200		250		300		450	ns	$\overline{CE}=\overline{OE}=V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		200		250		300		450	ns	$\overline{OE}=V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		75		100		120		150	ns	$\overline{CE}=V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Float	0	60	0	60	0	105	0	130	ns	$\overline{CE}=V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		ns	$\overline{CE}=\overline{OE}=V_{IL}$

NOTES: 1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for $t_A = 25°C$ and nominal supply voltages.
4. 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

CAPACITANCE ($T_A = 25°C$, $f = 1MHz$)

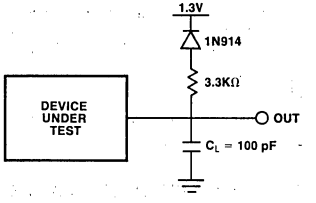
Symbol	Parameter	Typ.[1]	Max.	Unit	Conditions
C_{IN}[2]	Input Capacitance	4	6	pF	$V_{IN}=0V$
C_{OUT}	Output Capacitance	8	12	pF	$V_{OUT}=0V$

A.C. TESTING INPUT/OUTPUT WAVEFORM



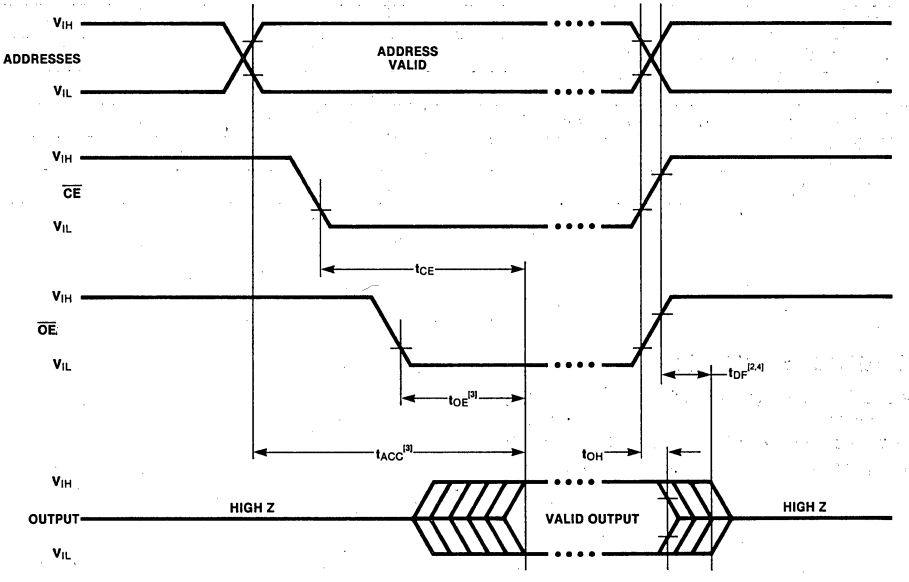
A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0."

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



NOTES:
1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.
3. $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{ACC}.
4. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first. Output float is defined as the point where data is no longer driven.

STANDARD PROGRAMMING

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

Symbol	Parameter	Limits			Test Conditions
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input Hight Level	2.0	$V_{CC}+1$	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage During Verify	2.4		V	$I_{OH} = -400\ \mu A$
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL} = \overline{PGM}$
V_{ID}	Ag for int$_e$ligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

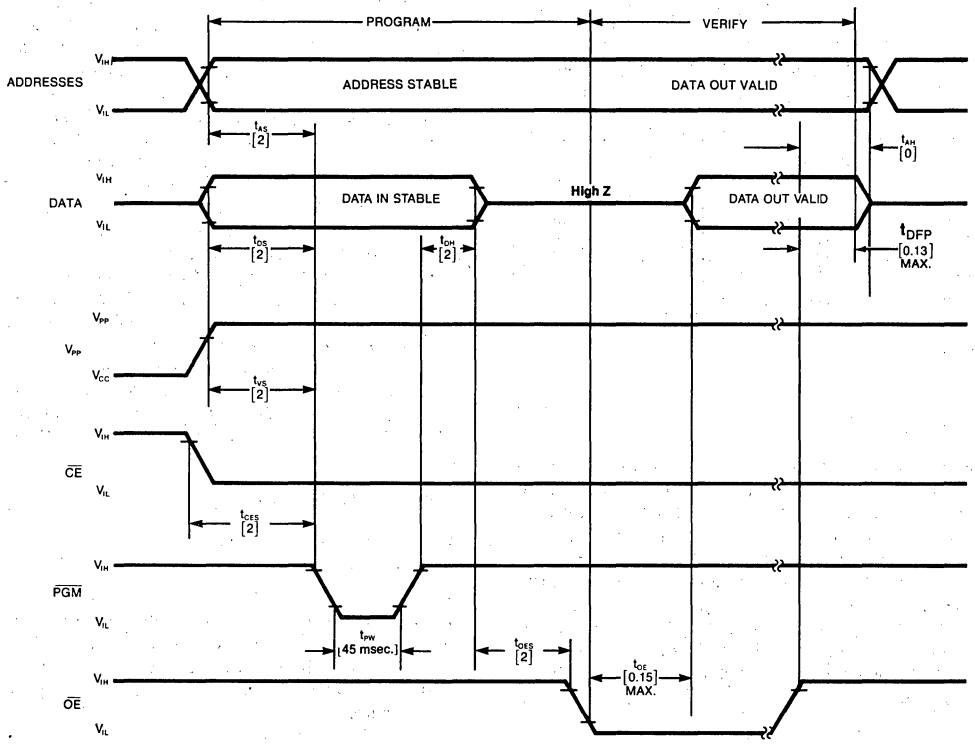
Symbol	Parameter	Limits				Test Conditions*
		Min.	Typ.	Max.	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}[2]	Output Enable to Output Float Delay	0		130	ns	
t_{VS}	V_{PP} Setup Time	2			μs	
t_{PW}	$\overline{PGM}$ Pulse Width During Programming	45	50	55	ms	
t_{CES}	$\overline{CE}$ Setup Time	2			μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

*A.C. CONDITIONS OF TEST

Input Rise and Fall Times (10% to 90%).....20ns
Input Pulse Levels.....0.45V to 2.4V
Input Timing Reference Level.....0.8V and 2.0V
Output Timing Reference Level.....0.8V and 2.0V

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

STANDARD PROGRAMMING WAVEFORMS



1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR V_{IL} AND 2V FOR A V_{IH}.
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 2764, A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

ERASURE CHARACTERISTICS

The erasure characteristics of the 2764 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (A). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000 Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2764 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2764 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 2764 window to prevent unintentional erasure.

The recommended erasure procedure for the 2764 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 W-sec/cm². The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm² power rating. The 2764 should be placed within 1 inch of the lamp tubes during erasure. The maximum integrated dose a 2764 can be exposed to without damage is 7258 Wsec/cm² (1 week @ 12000 μW/cm²). Exposure of the 2764 to high intensity UV light for long periods may cause permanent damage.

DEVICE OPERATION

The eight modes of operation of the 2764 are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for V_{PP} and 12V on A9 for int$_e$ligent Identifier mode.

Table 1. MODE SELECTION

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11–13, 15–19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
inteligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
inteligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

NOTES:
1. X can be V_{IH} or V_{IL}
2. $V_H = 12.0V \pm 0.5V$

READ MODE

The 2764 has two control functions, both of which must be logically active in order 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. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

STANDBY MODE

The 2764 has standby mode which reduces the maximum active current from 100 mA to 40 mA. The 2764 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays. Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

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

To use these two control lines most efficiently. $\overline{CE}$ (pin 20) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

System Considerations

The power switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer — the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitance loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note, AP-72, and by properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effect of PC board-traces.

PROGRAMMING MODES

Caution: Exeeding 22V on pin 1 (V_{PP}) will permanently damage the 2764.

"1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2764 is in the programming mode when V_{PP} input is at 21V and $\overline{CE}$ and $\overline{PGM}$ are both at TTL low. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

Standard Programming

For programming, $\overline{CE}$ should be kept TTL-low at all times while V_{PP} is kept at 21V. When the address and data are stable, a 50 msec, active-low, TTL program pulse is applied to the $\overline{PGM}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time—either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec.

Programming of multiple 2764s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2764s may be connected together when they are programmed with the same data. A low-level TTL pulse applied to the $\overline{PGM}$ input programs the paralleled 2764s.

Program Inhibit

Programming of multiple 2764s in parallel with different data is also easily accomplished by using the Program inhibit

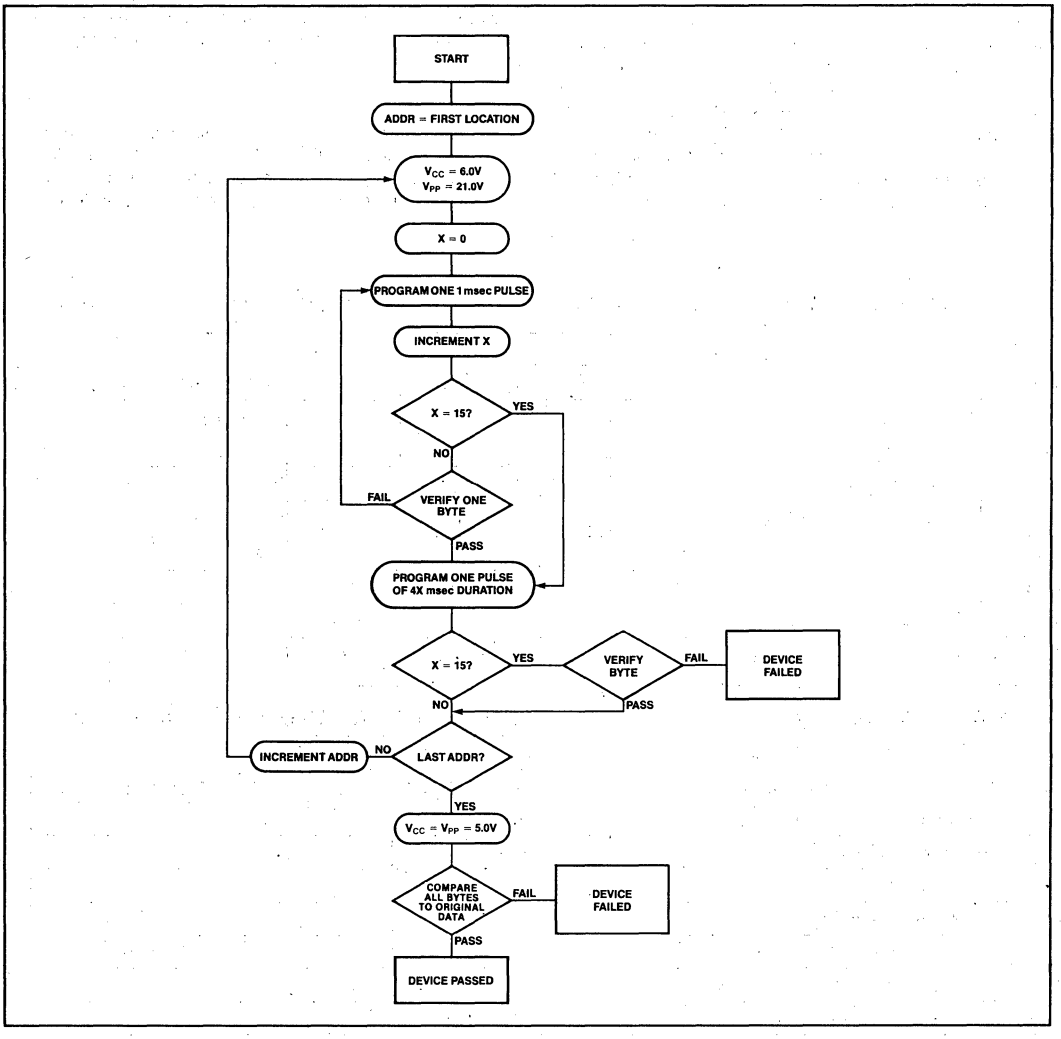


Figure 3. 2764 int$_e$ligent Programming™ Flowchart

mode. A high-level $\overline{CE}$ or $\overline{PGM}$ input inhibits the other 2764s from being programmed. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 2764s may be common. A TTL low-level pulse applied to a 2764 $\overline{CE}$ and $\overline{PGM}$ input with V_{PP} at 21V will program that 2764.

Verify

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify is performed with $\overline{CE}$ and $\overline{OE}$ at V_{IL}, $\overline{PGM}$ at V_{IH} and V_{PP} at 21V.

int$_e$ligent Programming™ Algorithm

The 2764 int$_e$ligent Programming Algorithm is the preferred programming method since it allows Intel 2764s to be programmed in a significantly faster time than the standard 50 msec per-byte programming routine. Typical programming times for 2764s are on the order of a minute and a half, which is a five-fold reduction in programming time from the standard method. This fast algorithm results in improved reliability characteristics compared to the standard 50 msec algorithm. A flowchart of the int$_e$ligent Programming Algorithm is shown in Figure 3. This is compatible with the 27128 int$_e$ligent Programming Algorithm.

This fast algorithm results in high reliability characteristics through the "closed loop" technique of margin checking. To ensure reliable program margin the int$_e$ligent Programming Algorithm utilizes two different pulse types: initial and over-program. The duration of the initial $\overline{PGM}$ pulse(s) is one millisecond, which will then be followed by a longer over-program pulse of length 4X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular 2764 location, before a correct verify occurs. Up to 15 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{PP} = 21.0V. When the int$_e$ligent Programming cycle has been completed, all bytes should be compared to the original data with $V_{CC} = V_{PP} = 5.0V$.

int$_e$ligent Programming™ Algorithm

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

Symbol	Parameter	Limits			Test Conditions
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input High Level	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	I_{OH} = −400 μA
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL} = \overline{PGM}$
V_{ID}	Ag for int$_e$ligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

Symbol	Parameter	Limits				Test 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	(see Note 4)
t_{DFP}	$\overline{OE}$ High to Output Float Delay	0		130	ns	
t_{VPS}	V_{PP} Setup Time	2			μs	
t_{VCS}	V_{CC} Setup Time	2			μs	
t_{PW}	$\overline{PGM}$ Initial Program Pulse Width	0.95	1.0	1.05	ms	(see Note 3)
t_{OPW}	$\overline{PGM}$ Overprogram Pulse Width	3.8		63	ms	(see Note 2)
t_{CES}	$\overline{CE}$ Setup Time	2			μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

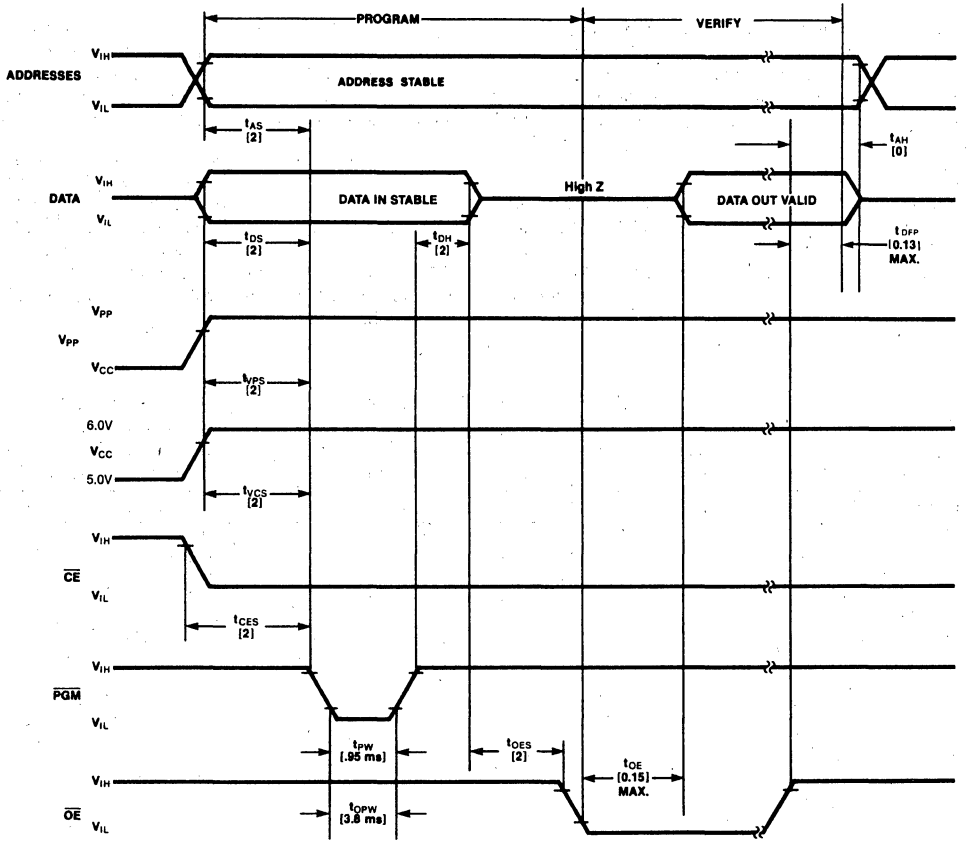
***A.C. CONDITIONS OF TEST**

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTES:

1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The length of the overprogram pulse will vary from 3.8 msec to 63 msec as a function of the iteration counter value X.
3. Initial Program Pulse width tolerance is 1 msec ±5%.
4. 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

inteligent Programming™ WAVEFORMS



1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR V_{IL} AND 2V FOR A V_{IH}.
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE R2764 A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

int$_e$ligent Identifier™ Mode

The int$_e$ligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ±5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 24) of the 2764. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 10) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during int$_e$ligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 2764, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

During 1982, Intel will begin manufacturing 2764s that will contain the int$_e$ligent Identifier feature. Earlier generation devices will not contain identifier information, and if erased, will respond with a "one" (V_{OH}) on each data line when operated in this mode. Programmed, pre-identifier mode 2764s will respond with the current data contained in locations 0 to 1 when subjected to the int$_e$ligent Identifier operation.

Table 2. 2764 int$_e$ligent Identifier™ Bytes

Pins / Identifier	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	0	0	0	0	0	0	1	0	02

2764A
ADVANCED 64K (8Kx8) UV ERASABLE PROM

- **200 nsec Standard Access Time —HMOS II*-E Technology**
- **Low Power —60 mA Maximum Active —20 mA Maximum Standby**
- **Two Line Control**
- **inteligent Programming™ Algorithm —Fastest EPROM Programming**
- **inteligent Identifier™ Mode —Automated Programming Operations**
- **Compatible with 2764, 27128, 27256**
- **±10% V_{CC} Tolerance Available**

The Intel 2764A is a 5V only, 65,536-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2764A is an advanced version of the 2764 and is fabricated with Intel's HMOSII-E technology which significantly reduces die size and greatly improves the device's performance, power consumption, reliability and producibility.

The typical 2764A is the 2746A-2 with an access time of 200 ns. This is an improvement over the 2764 standard time of 250 ns. This is compatible with high-performance microprocessors, such as Intel's 8 MHz iAPX 186 allowing full speed operation without the addition of WAIT states. The 2764A is also directly compatible with the 12 MHz 8051 family.

Several advanced features have been designed into the 2764A that allow fast and reliable programming—the inteligent Programming Algorithm and the inteligent Identifier Mode. Programming equipment that takes advantage of these innovations will electronically identify the 2764A and then rapidly program it using an efficient programming method.

The 2764A also offers reduced power consumption compared to the 2764. The maximum active current on faster speed parts is 60 mA while the maximum standby current is only 20 mA. The standby mode lowers power consumption without increasing access time.

Two-line control and JEDEC-approved, 28 pin packaging are standard features of all Intel higher density EPROMs. This ensures easy microprocessor interfacing and minimum design efforts when upgrading, adding or choosing between non-volatile memory alternatives.

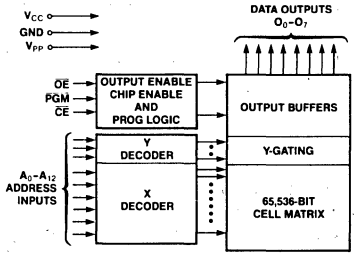


Figure 1. Block Diagram

27256	27128	2732A	2716
V_{PP}	V_{PP}		
A_{12}	A_{12}		
A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2
Gnd	Gnd	Gnd	Gnd

2764A

V_{PP}	1	28	V_{CC}
A_{12}	2	27	$\overline{PGM}$
A_7	3	26	N.C.
A_6	4	25	A_8
A_5	5	24	A_9
A_4	6	23	A_{11}
A_3	7	22	$\overline{OE}$
A_2	8	21	A_{10}
A_1	9	20	$\overline{CE}$
A_0	10	19	O_7
O_0	11	18	O_6
O_1	12	17	O_5
O_2	13	16	O_4
GND	14	15	O_3

2716	2732A	27128	27256
		V_{CC}	V_{CC}
		$\overline{PGM}$	A_{14}
V_{CC}	V_{CC}	A_{13}	A_{13}
A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$
A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3

NOTE: INTEL "UNIVERSAL SITE"-COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE 2764A PINS

Figure 2. Pin Configurations

MODE SELECTION

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13, 15-19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
inteligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
inteligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ±0.5V

PIN NAMES

A_0-A_{12}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
$\overline{PGM}$	PROGRAM
N.C.	NO CONNECT

*HMOS is a patented process of Intel Corporation

OCTOBER 1983
ORDER NUMBER: 230702-002

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .....−10°C to +80°C
Storage Temperature .....−65°C to +125°C
All Input or Output Voltages with Respect to Ground .....+6.25V to −0.6V
Voltage on Pin 24 with Respect to Ground .....+13.5V to −0.6V
V_{PP} Supply Voltage with Respect to Ground During Programming .....+14V to −0.6V

NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	2764A-1, 2764A-2, 2764A 2764A-3, 2764A-4	2764A-20, 2764A-25 2764A-30, 2764A-45
Operating Temperature Range	0°–70°C	0°–70°C
V_{CC} Power Supply[1,2]	5V ±5%	5V ±10%
V_{PP} Voltage[2]	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits Min	Typ[3]	Max	Unit	Conditions
I_{LI}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{PP1}[2]	V_{PP} Current Read			5	mA	$V_{PP} = 5.5V$
I_{CC1}[2]	V_{CC} Current Standby			20/35[5]	mA	$\overline{CE} = V_{IH}$
I_{CC2}[2]	V_{CC} Current Active			60/75[5]	mA	$\overline{CE} = \overline{OE} = V_{IL}$
V_{IL}	Input Low Voltage	−.1		+.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400$ μA
V_{PP}[2]	V_{PP} Read Voltage	3.8		V_{CC}	V	$V_{CC} = 5.0V \pm 0.25V$

A.C. CHARACTERISTICS

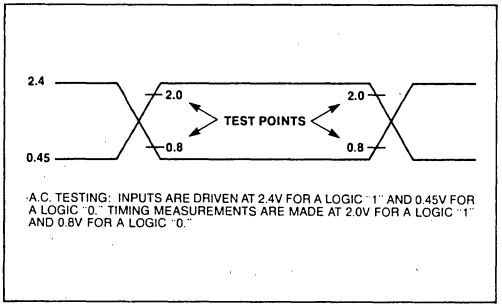
Symbol	Parameter	2764A-1 Limits Min	Max	2764A-20 & 2764A-2 Limits Min	Max	2764A-25 & 2764A Limits Min	Max	2764A-30 & 2764A-3 Limits Min	Max	2764A-45 & 2764A-4 Limits Min	Max	Unit	Test Conditions
t_{ACC}	Address to Output Delay		180		200		250		300		450	ns	$\overline{CE}=\overline{OE}=V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		180		200		250		300		450	ns	$\overline{OE}=V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		65		75		100		120		150	ns	$\overline{CE}=V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Float	0	55	0	55	0	60	0	105	0	130	ns	$\overline{CE}=V_{IL}$
t_{OH}	Output Hold from Addresses $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		0		ns	$\overline{CE}=\overline{OE}=V_{IL}$

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for $t_A = 25°C$ and nominal supply voltages.
4. 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 on the following page.
5. Max I_{CC} rating differs with access time. Rating of 60 mA active and 20 mA standby are for 2764As at 200 msec and 180 msec access time only.

CAPACITANCE ($T_A = 25°C$, $f = 1MHz$)

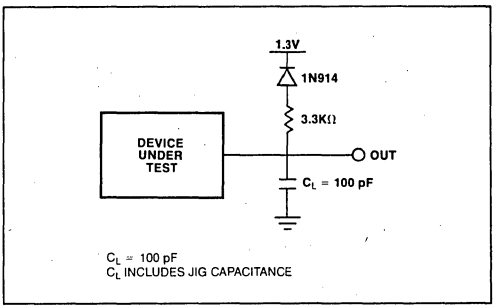
Symbol	Parameter	Typ.[1]	Max.	Unit	Conditions
C_{IN}[2]	Input Capacitance	4	6	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	8	12	pF	$V_{OUT} = 0V$

A.C. TESTING INPUT/OUTPUT WAVEFORM



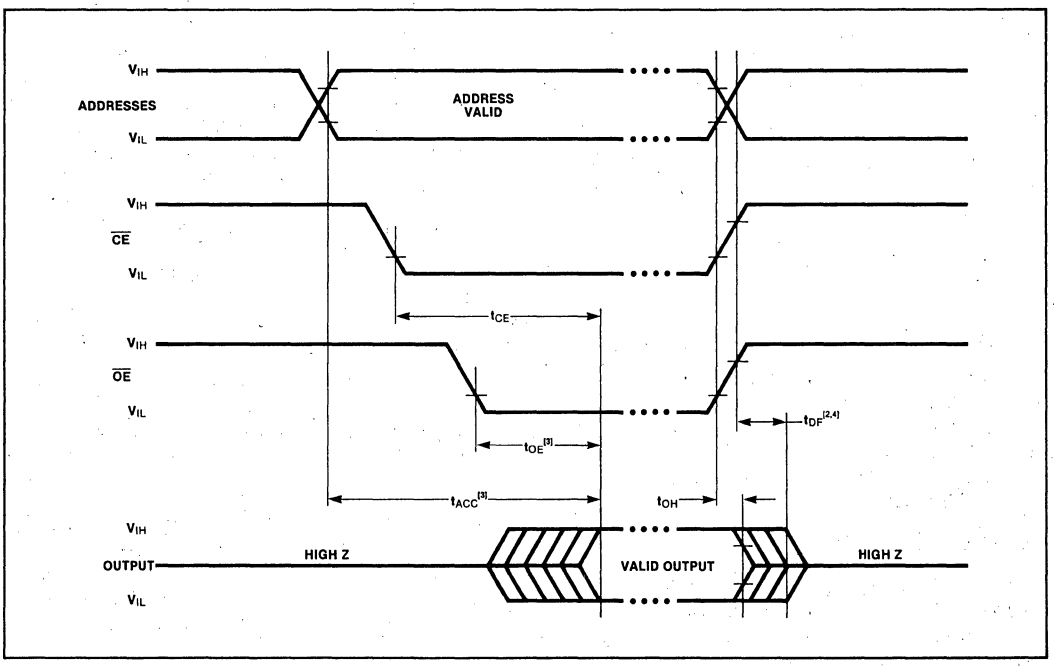
A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0."

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



NOTES:
1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.
3. $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{ACC}.
4. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first.

DEVICE OPERATION

The seven modes of operation of the 2764A are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for V_{PP} and 12V on A9 for inteligent identifier mode.

Table 1. MODE SELECTION

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11–13, 15–19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
inteligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
inteligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

NOTES:
1. X can be V_{IH} or V_{IL}
2. $V_H = 12.0V \pm 0.5V$

READ MODE

The 2764A has two control functions, both of which must be logically active in order 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. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

STANDBY MODE

The 2764A has standby mode which reduces the maximum current from 75 mA to 35 mA. The 2764A is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays, Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

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

To use these two control lines most efficiently, $\overline{CE}$ (pin 20) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

System Considerations

The power switching characteristics of HMOSII-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note AP-72, Order Number 8566, and by properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effect of PC board-traces.

PROGRAMMING MODES

Caution: Exceeding 14V on pin 1 (V_{PP}) will permanently damage the 2764A.

Initially, and after each erasure, all bits of the 2764A are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2764A is in the programming mode when V_{PP} input is at 12.5V and $\overline{CE}$ and $\overline{PGM}$ are both at TTL low. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

inteligent Programming™ Algorithm

The 2764A inteligent Programming Algorithm rapidly programs Intel 2764A EPROMs using an efficient and reliable method particularly suited to the production programming environment. Typical programming time for individual devices is on the order of one and a half minutes. Programming reliability is also ensured as the incremental program margin of each byte is continually monitored to determine when it has been successfully programmed. A flowchart of the 2764A inteligent Programming Algorithm is shown in Figure 3.

The inteligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial $\overline{CE}$ pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 3X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular 2764A location, before a correct verify occurs. Up to 25 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{PP} = 12.5V. When the inteligent Programming cycle has been completed, all bytes should be compared to the original data with $V_{CC} = V_{PP} = 5.0V$.

Program Inhibit

Programming of multiple 2764As in parallel with different data is easily accomplished by using the Program Inhibit mode. A high-level $\overline{CE}$ or $\overline{PGM}$ input inhibits the other 2764As from being programmed.

Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 2764As may be common. A TTL low-level pulse applied to the $\overline{CE}$ input with V_{PP} at 12.5V will program the selected 2764A.

Verify

A verify should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with $\overline{OE}$ at V_{IL}, $\overline{CE}$ at V_{IL}, $\overline{PGM}$ at V_{IH} and V_{PP} at 12.5V.

inteligent Identifier™ Mode

The inteligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ±5°C ambient temperature range that is required when programming the 2764A.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 24) of the 2764A. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 10) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during inteligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 2764A, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

Table 2. 2764A inteligent Identifier™ Bytes

Identifier \ Pins	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	0	0	0	0	1	0	0	0	08

NOTES:
1. $A_9 = 12.0V \pm 0.5V$
2. A_1–A_6, A_{10}–A_{13}, $\overline{CE}$, $\overline{OE} = V_{IL}$
3. $A_{14} = V_{IH}$ or V_{IL}

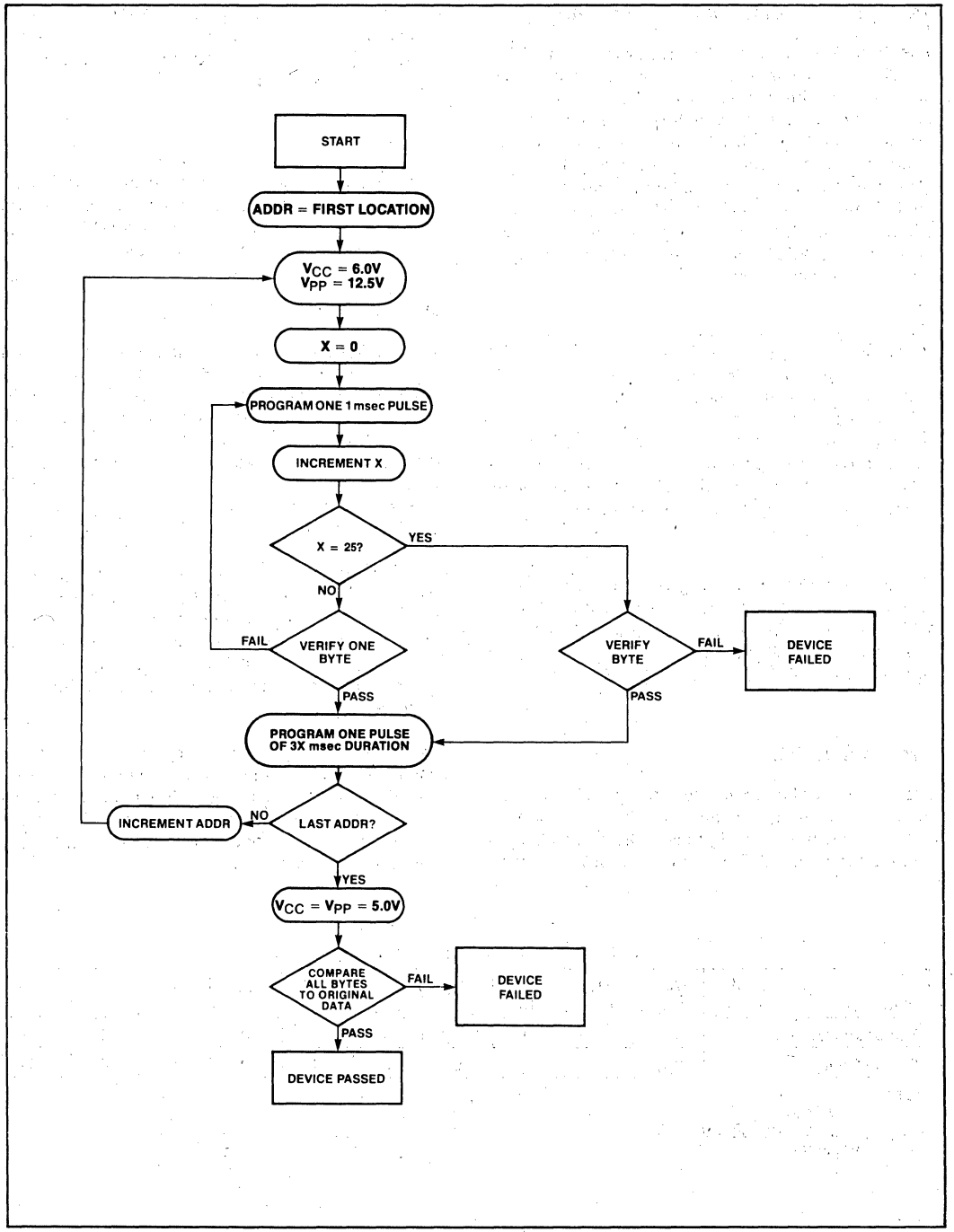


Figure 3. 2764A int$_e$ligent Programming™ Flowchart

ERASURE CHARACTERISTICS

The erasure characteristics of the 2764A are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room level fluorescent lighting could erase that typical 2764A in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2764A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 2764A window to prevent unintentional erasure.

The recommended erasure procedure for the 2764A is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 Wsec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The 2764A should be placed within 1 inch of the lamp tubes during erasure. The maximum integrated dose a 2764A can be exposed to without damage is 7258 Wsec/cm^2 (1 week @ 12000 μW/cm^2). Exposure of the 2764A to high intensity UV light for long periods may cause permanent damage.

RELEVANT INTEL LITERATURE

AR-265 Versatile Algorithm, Equipment Cut Programming Time
RR-35B EPROM Reliability Data Summary

int$_e$ligent Programming™ Algorithm

D.C. PROGRAMMING CHARACTERISTICS:

T_A = 25 ± 5°C, V_{CC} = 6.0V ± 0.25V, V_{PP} = 12.5V ± 0.5V

Symbol	Parameter	Limits			Test Conditions (see Note 1)
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input High Level	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	I_{OH} = −400 μA
I_{CC2}	V_{CC} Supply Current (Program & Verify)		75	mA	
I_{PP2}	V_{PP} Supply Current (Program)		50	mA	$\overline{CE} = V_{IL}$
V_{ID}	A_9 int$_e$ligent Identifier Voltage	11.5	12.5	V	

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.

A.C. PROGRAMMING CHARACTERISTICS:

$T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 12.5V \pm 0.5V$

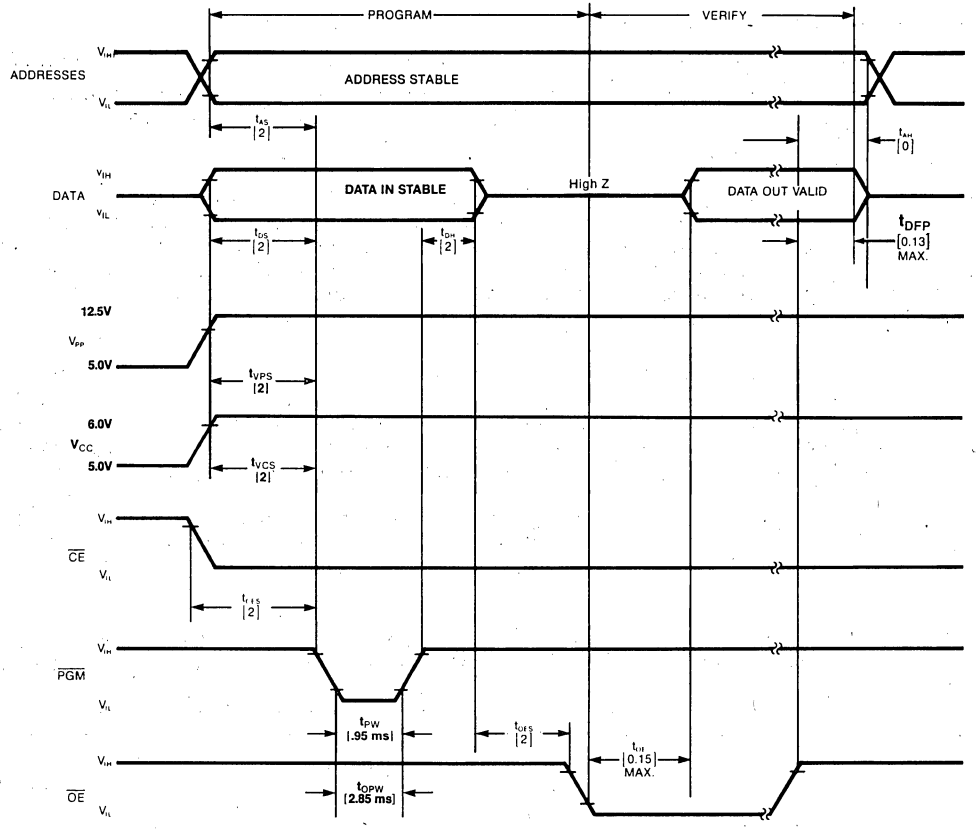
Symbol	Parameter	Limits				Test Conditions* (see Note 1)
		Min.	Typ.	Max.	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}[4]	$\overline{OE}$ High to Output Float Delay	0		130	ns	
t_{VPS}	V_{PP} Setup Time	2			μs	
t_{VCS}	V_{CC} Setup Time	2			μs	
t_{PW}	$\overline{PGM}$ Initial Program Pulse Width	0.95	1.0	1.05	ms	(see Note 3)
t_{OPW}	$\overline{PGM}$ Overprogram Pulse Width	2.85		78.75	ms	(see Note 2)
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

*A.C. CONDITIONS OF TEST

Input Rise and Fall Times (10% to 90%) ... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ... 0.8V and 2.0V

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The length of the overprogram pulse may vary from 2.85 msec to 78.75 msec as a function of the iteration counter value X.
3. Initial Program Pulse width tolerance is 1 msec ±5%.
4. 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

int$_e$ligent Programming™ WAVEFORMS



1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR V_{IL} AND 2V FOR A V_{IH}.
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 2764A, A 0.1 μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

P2764
64K (8K x 8) PRODUCTION EPROM

- **Guaranteed 2 Minutes Maximum Programming Time**
- **Compatible with High-Speed 8 MHz iAPX 186 . . . Zero WAIT State**
- **Two-Line Control**
- **Pin Compatible to 27128 EPROM**
- **Industry-Standard Pinout . . . JEDEC Approved**
- **Low Active Current . . . 100 mA Max.**
- **intelligent Programming™ Algorithm—for Reliable Programming**
- **TTL Compatible**

The Intel P2764 is a 5V-only, Electrically Programmable Read-Only Memory in a plastic package. One time programmable, it has been designed for high volume production environments where a programmable memory is required for flexibility. The standard P2764 access time is 250ns making it compatible with high performance microprocessors, such as the Intel 8MHz iAPX 186. In these systems, the P2764 allows the microprocessor to operate without the addition of WAIT states. The P2764 is also compatible with the 12 MHz 8051 family.

The P2764 is ideal for volume production environments where inventory and lead time risks occur for program codes. Inventoried in the unprogrammed state, the P2764 is programmed quickly and efficiently using the intelligent Programming Algorithm for new codes. Costs incurred for new ROM masks or obsoleted ROM inventories are avoided. The tight package dimensional controls, inherent non-erasability, and legible marking of the P2764 make it the ideal component for these production applications.

Additional features of the P2764 include high reliability and fast, reliable programming. The P2764 uses Intel's intelligent Programming™ Algorithm which typically enables the entire 8,192 bytes to program in under 2 minutes. This algorithm also ensures that the P2764 will reliably retain the programmed data throughout its effective life.

Using Intel's HMOS*-E technology, low power consumption combined with high speed data access are achieved. The typical P2764 active current is 70mA, while standby is only 25mA. The standby mode is selected by applying a TTL-high signal to the $\overline{CE}$ input.

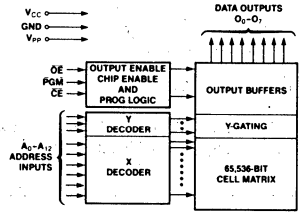


Figure 1. Block Diagram

27256	27128	2732A	2716	P2764	2716	2732A	27128	27256
V_{PP}	V_{PP}			V_{PP} 1 28 V_{CC}			V_{CC}	V_{CC}
A_{12}	A_{12}			A_{12} 2 27 $\overline{PGM}$			$\overline{PGM}$	A_{14}
A_7	A_7	A_7	A_7	A_7 3 26 N.C.	V_{CC}	V_{CC}	A_{13}	A_{13}
A_6	A_6	A_6	A_6	A_6 4 25 A_8	A_8	A_8	A_8	A_8
A_5	A_5	A_5	A_5	A_5 5 24 A_9	A_9	A_9	A_9	A_9
A_4	A_4	A_4	A_4	A_4 6 23 A_{11}	V_{PP}	A_{11}	A_{11}	A_{11}
A_3	A_3	A_3	A_3	A_3 7 22 $\overline{OE}$	$\overline{OE}$	$\overline{OE}$ V_{PP}	$\overline{OE}$	$\overline{OE}$
A_2	A_2	A_2	A_2	A_2 8 21 A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
A_1	A_1	A_1	A_1	A_1 9 20 $\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
A_0	A_0	A_0	A_0	A_0 10 19 O_7	O_7	O_7	O_7	O_7
O_0	O_0	O_0	O_0	O_0 11 18 O_6	O_6	O_6	O_6	O_6
O_1	O_1	O_1	O_1	O_1 12 17 O_5	O_5	O_5	O_5	O_5
O_2	O_2	O_2	O_2	O_2 13 16 O_4	O_4	O_4	O_4	O_4
Gnd	Gnd	Gnd	Gnd	GND 14 15 O_3	O_3	O_3	O_3	O_3

NOTE: INTEL "UNIVERSAL SITE"-COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE P2764 PINS

Figure 2. Pin Configurations

MODE SELECTION

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13, 15 -19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
intelligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
intelligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code

1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ±0.5V

PIN NAMES

A_0-A_{12}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
$\overline{PGM}$	PROGRAM
N.C.	NO CONNECT

*HMOS is a patented process of Intel Corporation

OCTOBER 1983
ORDER NUMBER: 230629-002

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +125°C
All Input or Output Voltages with Respect to Ground ..... +7.0V to −0.6V
Voltage on Pin 24 with Respect to Ground ..... +13.5V to −0.6V
V_{PP} Supply Voltage with Respect to Ground During Programming ..... +22V to −0.6V

*NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	P2764	P2764-3	P2764-4
Operating Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ± 5%	5V ± 5%	5V ± 5%

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits				Conditions
		Min	Typ[3]	Max	Unit	
I_{LI}	Input Load Current			10	μA	V_{IN} = 5.5V
I_{LO}	Output Leakage Current			10	μA	V_{OUT} = 5.5V
I_{PP1}[2]	V_{PP} Current Read			5	mA	V_{PP} = 5.5V
I_{CC1}[2]	V_{CC} Current Standby		25	40	mA	$\overline{CE}$ = V_{IH}
I_{CC2}[2]	V_{CC} Current Active		70	100	mA	$\overline{CE}$ = $\overline{OE}$ = V_{IL}
V_{IL}	Input Low Voltage	−.1		+.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+1	V	
V_{OL}	Output Low Voltage			.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −400 μA
V_{PP}[2]	V_{PP} Read Voltage	3.8		V_{CC}	V	V_{CC} = 5.0V ± 0.25V

A.C. CHARACTERISTICS

Symbol	Parameter	P2764 Limits		P2764-3 Limits		P2764-4 Limits		Unit	Test Conditions
		Min	Max	Min	Max	Min	Max		
t_{ACC}	Address to Output Delay		250		300		450	ns	$\overline{CE}=\overline{OE}=V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		250		300		450	ns	$\overline{OE}=V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		100		120		150	ns	$\overline{CE}=V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Float	0	60	0	105	0	130	ns	$\overline{CE}=V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		ns	$\overline{CE}=\overline{OE}=V_{IL}$

NOTES:

1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for t_A = 25°C and nominal supply voltages.
4. 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

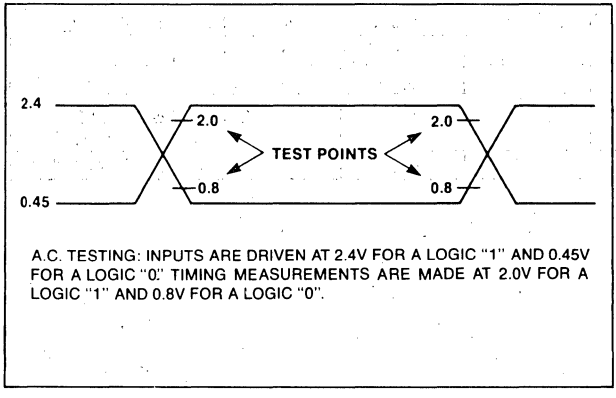
CAPACITANCE (T_A = 25°C, f = 1MHz)

Symbol	Parameter	Typ.[1]	Max.	Unit	Conditions
C_{IN}[2]	Input Capacitance	4	6	pF	V_{IN}=0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT}=0V

NOTES:

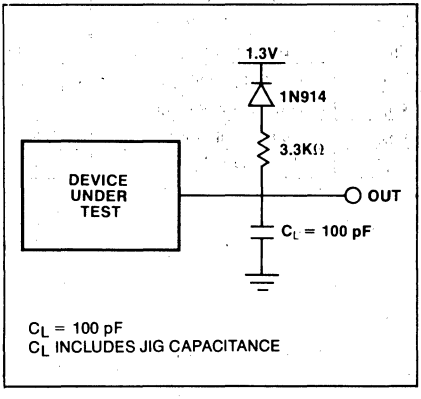
1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.

A.C. TESTING INPUT/OUTPUT WAVEFORM



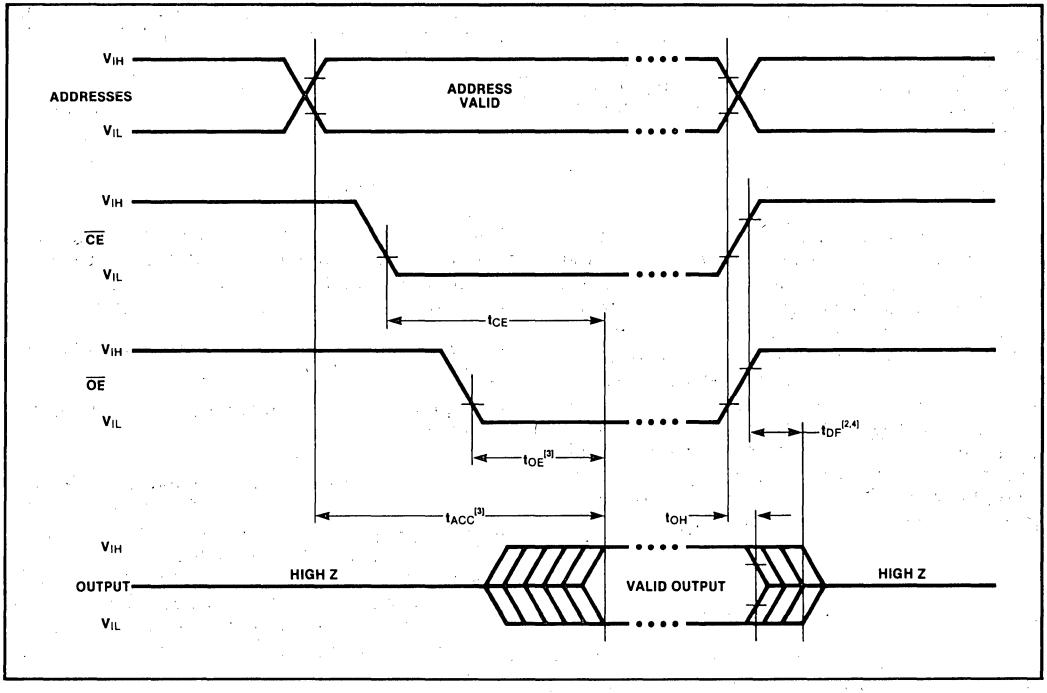
A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0".

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



NOTES:

1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.
3. $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{ACC}.
4. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first. Output float is defined as the point where data is no longer driven.

APPLICATIONS INFORMATION

Intel's P2764 is the result of a multi-year effort to make EPROMs more cost effective for production applications. The benefits of a plastic package enable the P2764 to be used for high volume production with lower profile boards and easier production assembly (no cover over UV transparent windows).

The reliability of plastic EPROMs is equivalent to traditional CERDIP packaging. The plastic is rugged and durable making it optimal for auto insertion and auto handling equipment. Design and testing ensures device programmability, data integrity, and impermeability to moisture.

Intel's Plastic EPROMs are designed for total compatibility with their CERDIP packaged predecessors. This encompasses quality, reliability, and programming. All Intel Plastic EPROMs have passed Intel's strict process and product reliability qualifications.

DEVICE OPERATION

The seven modes of operation of the P2764 are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for V_{PP} and 12V on A9 for int$_e$ligent Identifier mode. The int$_e$ligent Programming Algorithm requires 6V on V_{CC}.

READ MODE

The P2764 has two control functions, both of which must be logically active in order 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. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least t_{ACC}-t_{OE}.

Table 1. MODE SELECTION

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13, 15-19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
int$_e$ligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
int$_e$ligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

NOTES:
1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ±0.5V

STANDBY MODE

The P2764 has a standby mode which reduces the maximum active current from 100 mA to 40 mA. The P2764 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays, Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

a) the lowest possible memory power dissipation, and

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

To use these two control lines most efficiently, $\overline{CE}$ (pin 20) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

System Considerations

The power-switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer — the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these tran-

sient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note, AP-72, and by properly selected decoupling capacitors. It is recommended that a 0.1μF cermic capacitor be used on every device between V_{CC} and GND. This should be a high-frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effect of PC board-traces.

PROGRAMMING MODES

CAUTION: The P2764 must be programmed using the inteligent Programming Algorithm. Exceeding 22V on pin 1 (V_{PP}) will permanently damage the P2764.

Initially, all bits of the P2764 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word.

The P2764 is in the programming mode when the V_{PP} Input is at 21V and $\overline{CE}$ and $\overline{PGM}$ are both at TTL-low. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

inteligent Programming™ Algorithm

The inteligent Programming Algorithm allows the P2764 to be programmed in a significantly faster time than standard 2764 EPROMs. Typical inteligent programming times are on the order of a minute and a half. This is a five-fold reduction in programming time from a standard 50 msec method. A flowchart of the inteligent Programming Algorithm is shown in Fig. 3.

Execution of the inteligent Programming Algorithm at maximum speed will result in a guaranteed maximum programming time of two minutes for the P2764 when programmed individually. This excludes the time for initial erase check, final verify, and programmer overhead.

This fast algorithm results in high reliability characteristics through the "closed loop" technique of margin checking. The P2764 is designed to retain data for at least 20 years.

To ensure reliable program margin, the inteligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial $\overline{PGM}$ pulse (s) is one millisecond, which will then be followed by a longer overprogram pulse of length 4X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular P2764 location, before a correct verify occurs. Up to 15 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{PP} = 21.0V. When the inteligent Programming cycle has been completed, all bytes should be compared to the original data with $V_{CC} = V_{PP} = 5.0V$.

Program Inhibit

Programming of multiple P2764s in parallel with different data is also easily accomplished by using the Program inhibit mode. A high-level $\overline{CE}$ or $\overline{PGM}$ input inhibits the other P2764s from being programmed. Except for $\overline{CE}$ (or $\overline{PGM}$), all like inputs (including $\overline{OE}$) of the parallel F2764s may be common. A TTL low-level pulse applied to a P2764 $\overline{CE}$ and $\overline{PGM}$ input with V_{PP} at 21V will program that P2764.

Program Verify

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify is accomplished with $\overline{CE}$ and $\overline{OE}$ at V_{IL}. However, $\overline{PGM}$ is at V_{IH}.

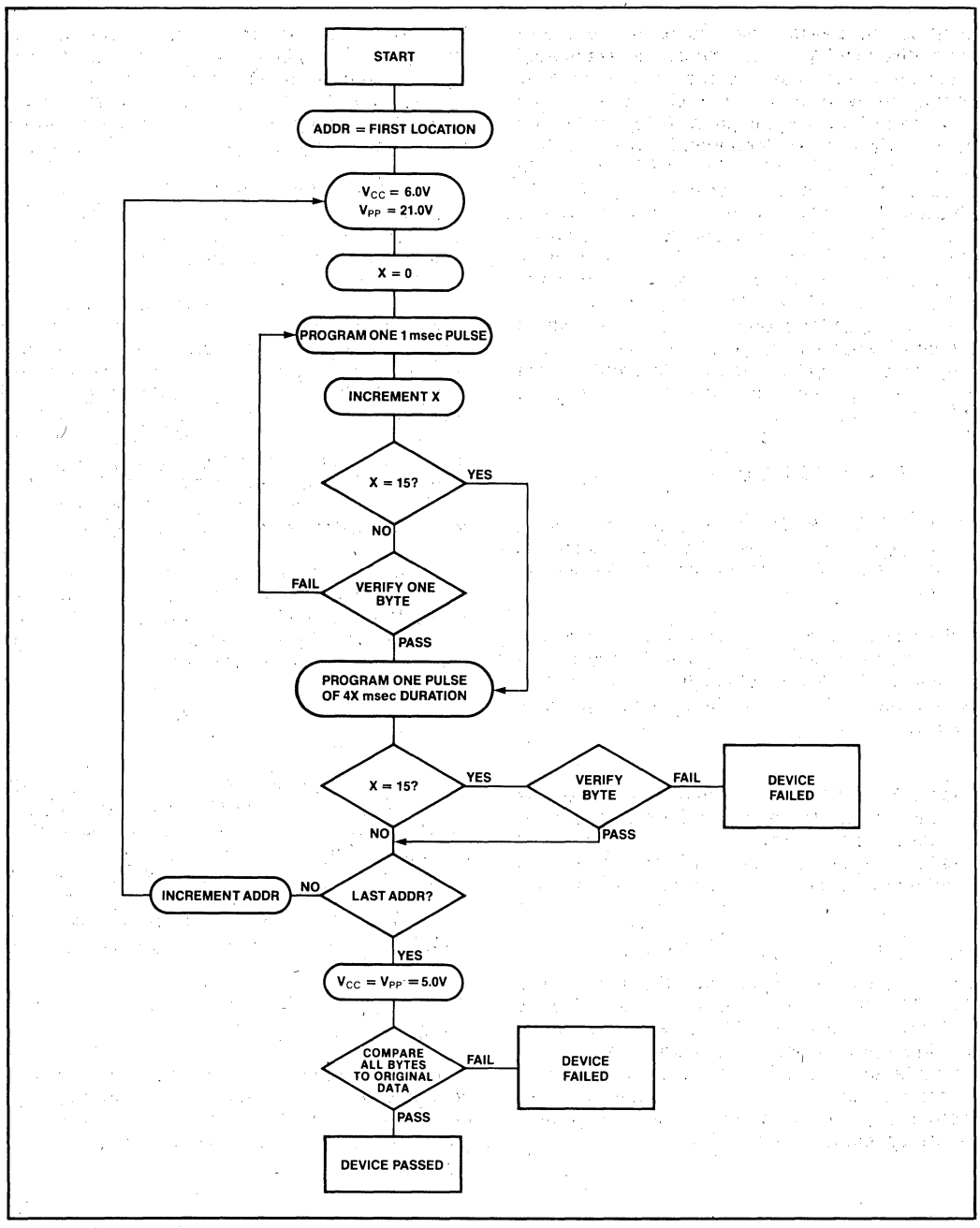


Figure 3. P2764 int$_e$ligent Programming™ Flowchart

inteligent Programming™ Algorithm

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

Symbol	Parameter	Limits			Test Conditions
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input High Level	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	$I_{OH} = -400\ \mu A$
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL} = \overline{PGM}$
V_{ID}	A9 inteligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

Symbol	Parameter	Limits				Test 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}	$\overline{OE}$ High to Output Not Driven	0		130	ns	
t_{VPS}	V_{PP} Setup Time	2			μs	
t_{VCS}	V_{CC} Setup Time	2			μs	
t_{PW}	$\overline{PGM}$ Initial Program Pulse Width	0.95	1.0	1.05	ms	(see Note 3)
t_{OPW}	$\overline{PGM}$ Overprogram Pulse Width	3.8		63	ms	(see Note 2)
t_{CES}	$\overline{CE}$ Setup Time	2			μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

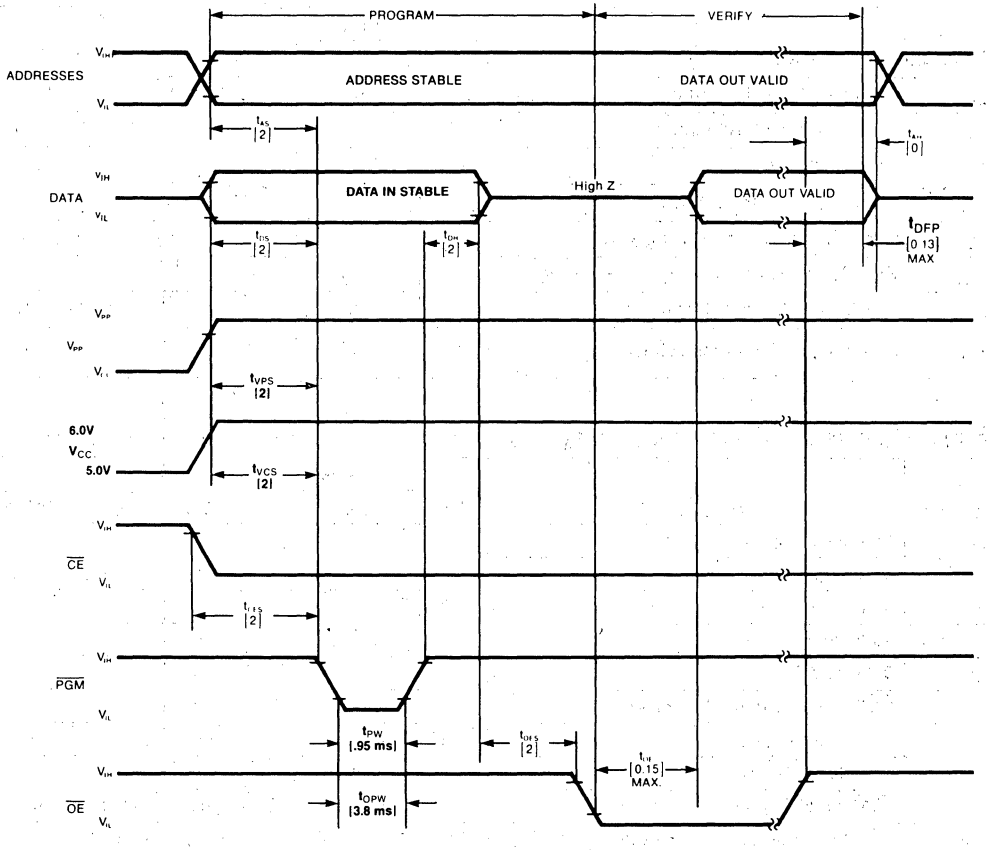
***A.C. CONDITIONS OF TEST**

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTES:

1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The length of the overprogram pulse will vary from 3.8 msec to 63 msec as a function of the iteration counter value X.
3. Initial Program Pulse width tolerance is 1 msec ± 5%.
4. 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 on the following page.

inteligent Programming™ WAVEFORMS



1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR V_{IL} AND 2V FOR A V_{IH}.
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE P2764, A 0.1μ CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

int$_e$ligent Identifier™ Mode

The int$_e$ligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ± 5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 24) of the P2764. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 10) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during the int$_e$ligent Identifier mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel P2764, these two identifier bytes are given in Table 2. These are also the same identifier bytes as the Intel 2764 EPROM. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

Note: In programming equipment that employs the int$_e$ligent Identifier function, a provision must be made to ensure that the identifier bytes select only the int$_e$ligent Programming™ Algorithm to program the P2764, or that the function can be manually bypassed to do so.

Table 2. P2764 int$_e$ligent Identifer™ Bytes

Identifier \ Pins	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	0	0	0	0	0	0	1	0	02

P2764A
ADVANCED 64K (8K x 8) PRODUCTION EPROM

- **int$_e$ligent Programming™ Algorithm — Fast, Reliable Programming**
- **Compatible with High Speed 8 MHz iAPX 186...Zero Wait States**
- **Two-Line Control**
- **TTL Compatible**
- **int$_e$ligent Identifier™ — Error-Free Programming**
- **Low Power**
  - **— 60 mA Maximum Active**
  - **— 20 mA Maximum Standby**
- **Compatible with P2764 and 2764**
- **JEDEC Approved Pinout**

The Intel P2764A is a 5V-only. Electrically Programmable Read-Only Memory in a plastic package. One-time programmable, it is designed for high volume production environments where programmable memory is required for flexibility. The P2764A is an advanced version of the P2764 and is fabricated with Intel's HMOS*II-E technology which significantly reduces die size and greatly improves the device's performance, power consumption, and reliability.

The P2764A is ideal for high volume production environments where inventory and lead times are important factors. Costs incurred for new ROM masks or obsolete ROM inventory are avoided. With the int$_e$ligent Programming™ Algorithm, the P2764A provides quick and efficient programming throughput. The rugged package of the P2764A makes it the ideal component for auto-insertable production applications.

Standard access time is 250 nsec, making it compatible with high performance microprocessors such as the Intel 8 MHz iAPX 186. In these systems, the P2764A allows the microprocessor to operate without the addition of WAIT states.

The P2764A also offers reduced power consumption compared to the P2764. The maximum active current on the 200 ns version is 60 mA while the maximum standby current is only 20 mA. The standby mode lowers power consumption without increasing access time.

Two-Line control and JEDEC-approved, 28 pin packaging are standard features of all Intel high density EPROMs. This ensures easy microprocessor interfacing and minimum design efforts when upgrading, adding or choosing among non-volatile memory alternatives.

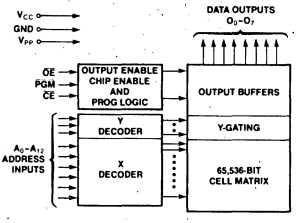
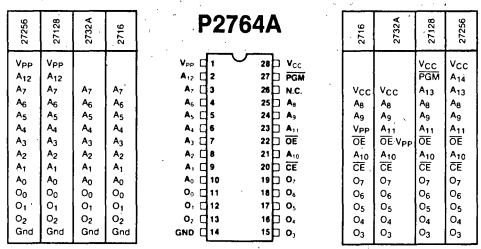


Figure 1. Block Diagram

MODE SELECTION

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13, 15-19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	x	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
int$_e$ligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
int$_e$ligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ±0.5V



NOTE: INTEL "UNIVERSAL SITE"-COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE P2764A PINS.

Figure 2. Pin Configurations

PIN NAMES

A_0-A_{12}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
$\overline{PGM}$	PROGRAM
N.C.	NO CONNECT

*HMOS is a patented process of Intel Corporation.

OCTOBER 1984
ORDER NO. 230864-001

27128
128K (16K x 8) UV ERASABLE PROM

- **250 ns Maximum Access Time . . . HMOS*-E Technology**
- **Compatible with High-Speed 8 MHz iAPX 186...Zero WAIT State**
- **Two-Line Control**
- **Pin Compatible to 2764 EPROM**
- **Industry Standard Pinout . . . JEDEC Approved**
- **$\pm 10\%$ V_{CC} Tolerance Available**
- **Low Active Current . . . 100 mA Max.**
- **int$_e$ligent Programming™ Algorithm**

The Intel 27128 is a 5V only, 131,072-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The standard 27128 access time is 250 ns which is compatible with high-performance microprocessors such as Intel's 8 MHz iAPX 186. In these systems the 27128 allows the microprocessor to operate without the addition of WAIT states. The 27128 is also compatible with the 12 MHz 8051 family.

An important 27128 feature is the separate output control, Output Enable ($\overline{OE}$) from the Chip Enable control ($\overline{CE}$). The $\overline{OE}$ control eliminates bus contention in microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's EPROMs. AP-72 is available from Intel's Literature Department.

The 27128 has standby mode which reduces the power consumption without increasing access time. The maximum active current is 100 mA, while the maximum standby current is only 40 mA. The standby mode is selected by applying a TTL-high signal to the $\overline{CE}$ input.

$\pm 10\%$ V_{CC} tolerance is available as an alternative to the standard $\pm 5\%$ V_{CC} tolerance for the 27128. This can allow the system designer more leeway with regard to his power supply requirements and other system parameters.

The 27128 is fabricated with HMOS*-E technology, Intel's high-speed N-channel MOS Silicon Gate Technology.

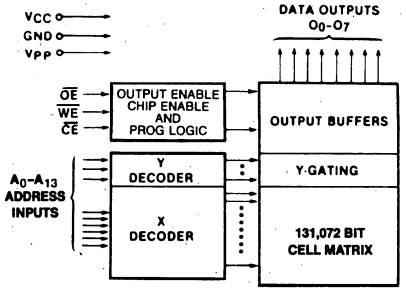


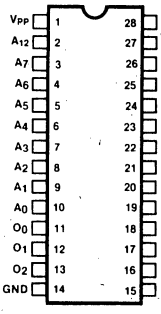
Figure 1. Block Diagram

MODE SELECTION

Mode \ Pins	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13,15-19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
int$_e$ligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
int$_e$ligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

NOTES:
1. X can be V_{IH} or V_{IL}
2. $V_H = 12.0V \pm 0.5V$

27256	2764	2732A	2716
V_{PP}	V_{PP}		
A_{12}	A_{12}		
A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2
Gnd	Gnd	Gnd	Gnd



2716	2732A	2764	27256
		V_{CC}	V_{CC}
		$\overline{PGM}$	A_{14}
V_{CC}	V_{CC}	N.C.	A_{13}
A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$
A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3

NOTE: INTEL "UNIVERSAL SITE" COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE 27128 PINS

Figure 2. Pin Configurations

PIN NAMES

A_0-A_{13}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
$\overline{PGM}$	PROGRAM
N.C.	NO CONNECT

*HMOS is a patented process of Intel Corporation.

OCTOBER 1983
ORDER NUMBER: 210224-005

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .....−10°C to +80°C
Storage Temperature .....−65°C to +125°C
All Input or Output Voltages with Respect to Ground .....+7.0V to −0.6V
Voltage on Pin 24 with Respect to Ground .....+13.5V to −0.6V
V_{PP} Supply Voltage with Respect to Ground During Programming .....+22V to −0.6V

NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	27128	27128-3	27128-4	27128-25	27128-30	27128-45
Operating Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 10%	5V ± 10%	5V ± 10%
V_{PP} Voltage[2]	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[3]	Max.		
I_{LI}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{PP1}[2]	V_{PP} Current Read/Standby			5	mA	$V_{PP} = 5.5V$
I_{CC1}[2]	V_{CC} Current Standby		15	40	mA	$\overline{CE} = V_{IH}$
I_{CC2}[2]	V_{CC} Current Active		60	100	mA	$\overline{CE} = \overline{OE} = V_{IL}$
V_{IL}	Input Low Voltage	−.1		+.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400\ \mu A$
V_{PP}[2]	V_{PP} Read Voltage	3.8		V_{CC}	V	$V_{CC} = 5.0V \pm 0.25V$

A.C. CHARACTERISTICS

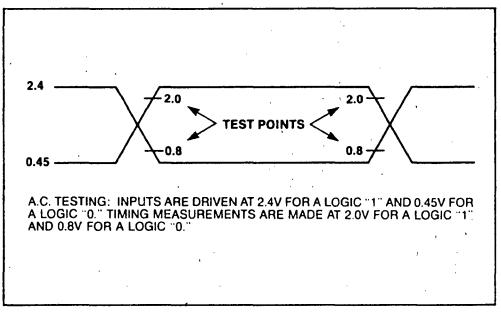
Symbol	Parameter	27128-25 & 27128 Limits		27128-30 27128-3 Limits		27128-45 & 27128-4 Limits		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
t_{ACC}	Address to Output Delay		250		300		450	ns	$\overline{CE}=\overline{OE}=V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		250		300		450	ns	$\overline{OE}=V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		100		120		150	ns	$\overline{CE}=V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Float	0	60	0	105	0	130	ns	$\overline{CE}=V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		ns	$\overline{CE}=\overline{OE}=V_{IL}$

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for $t_A = 25°C$ and nominal supply voltages.
4. 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

CAPACITANCE (T_A = 25°C, f = 1 MHz)

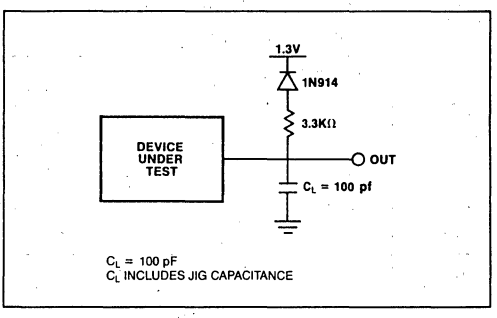
Symbol	Parameter	Typ.[1]	Max.	Unit	Conditions
C_{IN}[2]	Input Capacitance	4	6	pF	V_{IN}=0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT}=0V

A.C. TESTING INPUT/OUTPUT WAVEFORM



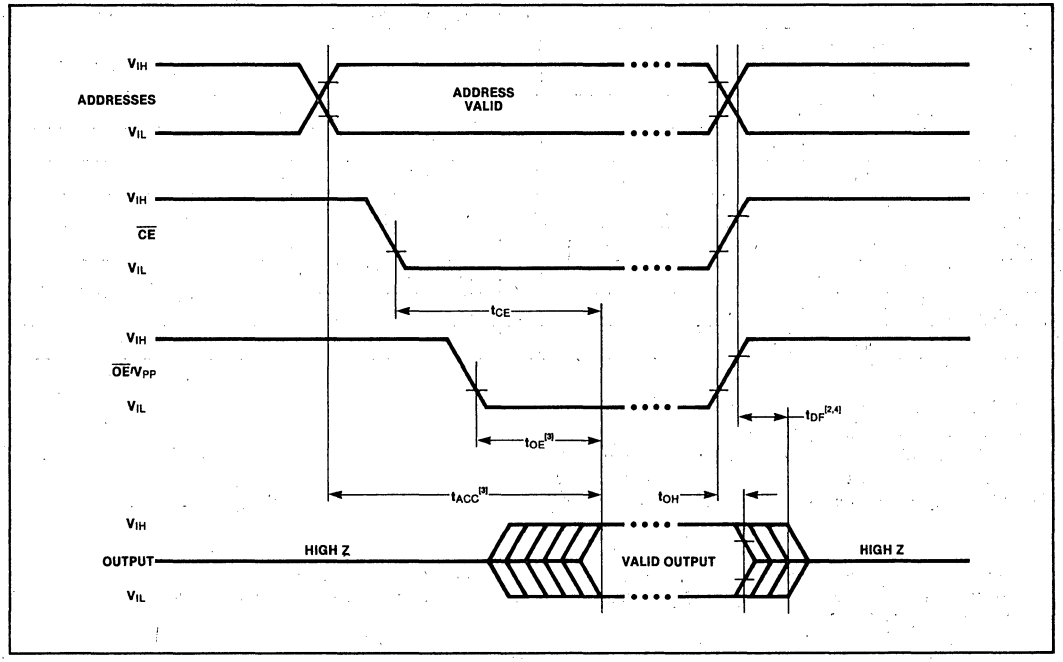
A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0."

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



NOTES:
1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.
3. $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{ACC}.
4. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first.

STANDARD PROGRAMMING

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

Symbol	Parameter	Limits			Test Conditions
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{OL}	Output Low Voltage During Verify		0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage During Verify	2.4		V	$I_{OH} = -400$ μA
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input High Level	2.0	$V_{CC} + 1$	V	
I_{CC1}	V_{CC} Supply Current (Program Inhibit)		40	mA	$\overline{CE} = V_{IH}$
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL} = \overline{PGM}$
I_{PP3}	V_{PP} Supply Current (Verify)		5	mA	$\overline{CE} = V_{IL}$ $\overline{PGM} = V_{IH}$
I_{PP4}	V_{PP} Supply Current (Program Inhibit)		5	mA	$\overline{CE} = V_{IH}$
V_{ID}	A9 Intelligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

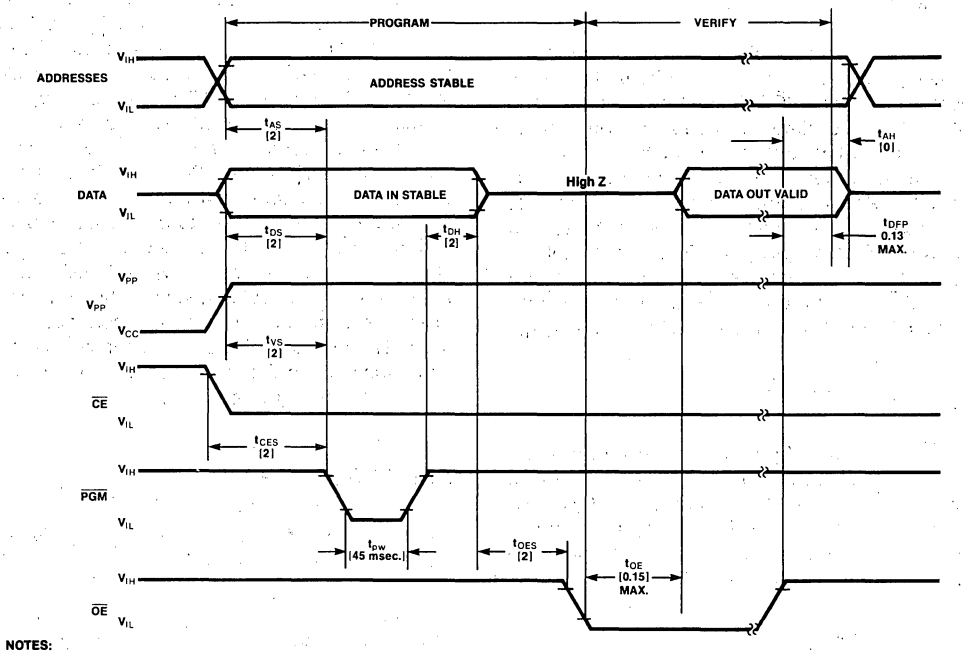
Symbol	Parameter	Limits				Test Conditions*
		Min.	Typ.	Max.	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}[2]	Output Enable to Output Float Delay	0		130	ns	
t_{VS}	V_{PP} Setup Time	2			μs	
t_{PW}	$\overline{PGM}$ Pulse Width During Programming	45	50	55	ms	
t_{CES}	$\overline{CE}$ Setup Time	2			μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

*A.C. CONDITIONS OF TEST

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

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

STANDARD PROGRAMMING WAVEFORMS



NOTES:
1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR A V_{IL} AND 2V FOR A V_{IH}.
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 87128 A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

ERASURE CHARACTERISTICS

The erasure characteristics of the 27128 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000 Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 27128 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 27128 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 27128 window to prevent unintentional erasure.

The recommended erasure procedure for the 27128 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 Wsec/cm². The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm² power rating. The 27128 should be placed within 1 inch of the lamp tubes during erasure. The maximum integrated dose a 27128 can be exposed to without damage is 7258 Wsec/cm² (1 week @ 12000 μW/cm²). Exposure of the 27128 to high intensity UV light for long periods may cause permanent damage.

DEVICE OPERATION

The eight modes of operation of the 27128 are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for V_{PP} and 12V on A9 for int$_e$ligent Identifier mode.

Table 1. Mode Selection

Mode \ Pins	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11–13, 15–19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
int$_e$ligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
int$_e$ligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

NOTES:
1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ±0.5V

READ MODE

The 27128 has two control functions, both of which must be logically active in order 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. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

STANDBY MODE

The 27128 has standby mode which reduces the maximum active current from 100 mA to 40 mA. The 27128 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays, Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

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

To use these two control lines most efficiently, $\overline{CE}$ (pin 20) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

System Considerations

The power switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note, AP-72, and by properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effects of PC board traces.

PROGRAMMING MODES

Caution: Exceeding 22V on pin 1 (V_{PP}) will permanently damage the 27128.

Initially, and after each erasure, all bits of the 27128 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 27128 is in the programming mode when V_{PP} input is at 21V and $\overline{CE}$ and $\overline{PGM}$ are both at TTL low. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

Standard Programming

For programming, $\overline{CE}$ should be kept TTL-low at all times while V_{PP} is kept at 21V. When the address and data are stable, a 50 msec, active-low, TTL program pulse is applied to the $\overline{PGM}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time —either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec.

Programming of multiple 27128s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 27128s may be connected together when they are programmed with the same data. A low-level TTL pulse applied to the $\overline{PGM}$ input programs the paralleled 27128s.

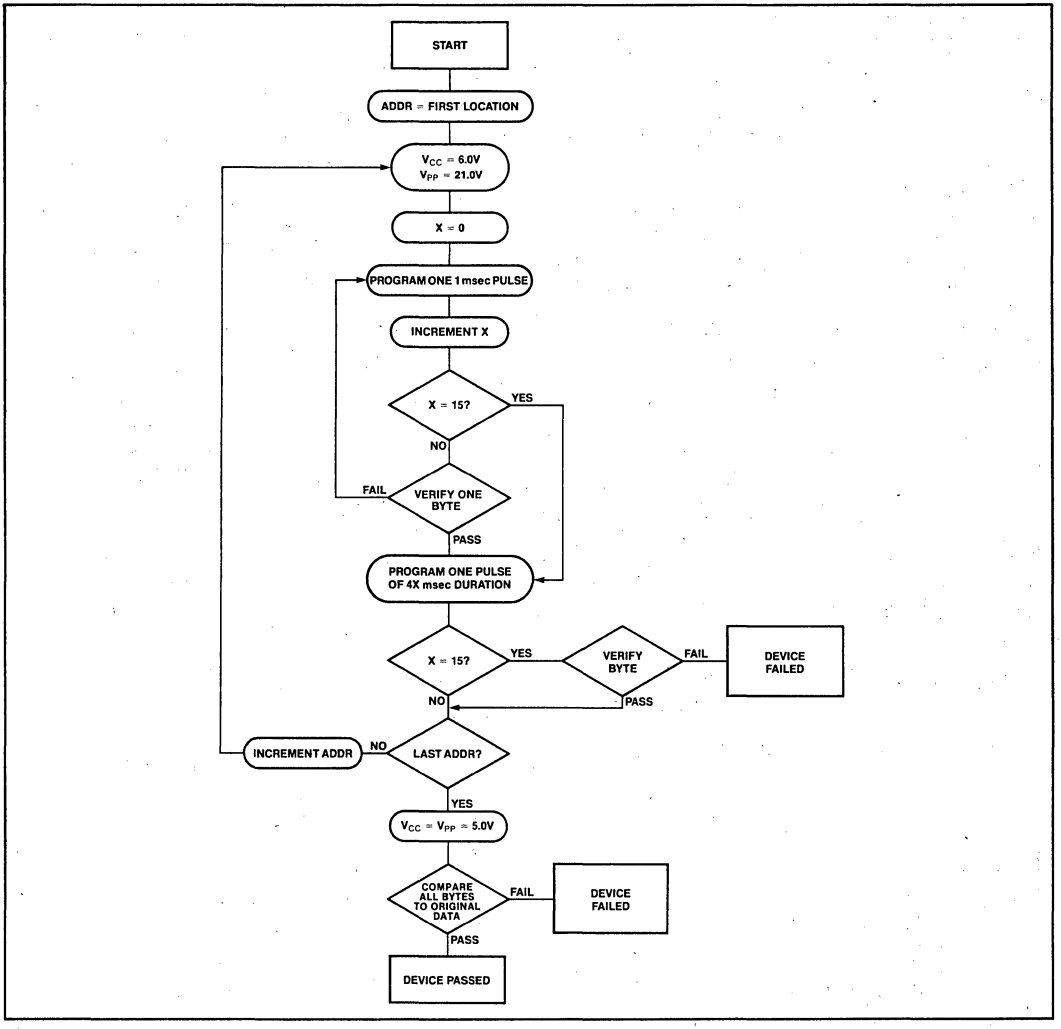


Figure 3. 27128 int$_e$ligent Programming™ Flowchart

Program Inhibit

Programming of multiple 27128s in parallel with different data is easily accomplished by using the Program Inhibit mode. A high-level $\overline{CE}$ or $\overline{PGM}$ input inhibits the other 27128s from being programmed. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 27128s may be common. A TTL low-level pulse applied to the $\overline{CE}$ and $\overline{PGM}$ inputs with V_{PP} at 21V will program the selected 27128.

Verify

A verify should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with $\overline{CE}$ and $\overline{OE}$ at V_{IL}, $\overline{PGM}$ at V_{IH} and V_{PP} at 21V.

int$_e$ligent Programming™ Algorithm

The 27128 int$_e$ligent Programming Algorithm is the preferred programming method since it allows Intel 27128s to be programmed in a significantly faster time than the standard 50 msec per byte programming routine. Typical programming times for 27128s are on the order of two minutes, which is a six-fold reduction in programming time from the standard method. This fast algorithm results in improved reliability characteristics over the standard 50 msec

algorithm. A flowchart of the 27128 inteligent Programming Algorithm is shown in Figure 3. This is compatible with the 2764 inteligent Programming Algorithm.

This fast algorithm assures reliable programming through the "closed loop" technique of margin checking. To ensure reliable program margin the inteligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial $\overline{PGM}$ pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 4X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular 27128 location, before a correct verify occurs. Up to 15 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{PP} = 21.0V. When the inteligent Programming cycle has been completed, all bytes should be compared to the original data with $V_{CC} = V_{PP} = 5.0V$.

inteligent Programming™ Algorithm

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits			Test Conditions (see Note 1)
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input High Level	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	I_{OH} = −400 μA
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL} = \overline{PGM}/\overline{WE}$
V_{ID}	A_9 inteligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits				Test Conditions* (see Note 1)
		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}[4]	$\overline{OE}$ High to Output Float Delay	0		130	ns	
t_{VPS}	V_{PP} Setup Time	2			μs	
t_{VCS}	V_{CC} Setup Time	2			μs	
t_{PW}	$\overline{PGM}/\overline{WE}$ Initial Program Pulse Width	0.95	1.0	1.05	ms	(see Note 3)
t_{OPW}	$\overline{PGM}/\overline{WE}$ Overprogram Pulse Width	3.8		63	ms	(see Note 2)
t_{CES}	$\overline{CE}$ Setup Time	2			μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

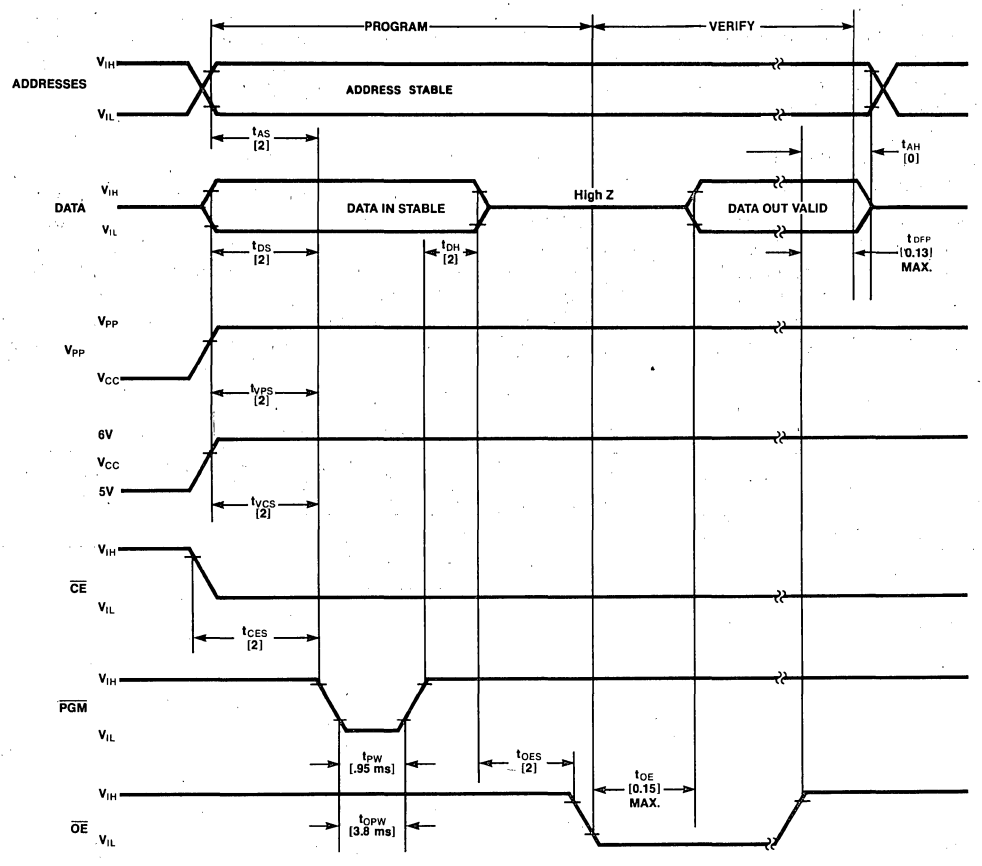
***A.C. CONDITIONS OF TEST**

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTES:

1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The length of the overprogram pulse will vary from 3.8 msec to 63 msec as a function of the iteration counter value X.
3. Initial Program Pulse width tolerance is 1 msec ± 5%.
4. This parameter is only sampled as is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram on the following page.

inteligent Programming™ WAVEFORMS



NOTES:
1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR A V_{IL} AND 2V FOR A V_{IH}.
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 27128, A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

inteligent Identifier™ Mode

The inteligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ± 5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 24) of the 27128. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 10) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during inteligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 27128, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

Intel will begin manufacturing 27128s during 1982 that will contain the inteligent Identifier feature. Earlier generation devices will not contain identifier information, and if erased, will respond with a "one" (V_{OH}) on each data line when operated in this mode. Programmed, pre-identifier mode 27128s will respond with the current data contained in locations 0 and 1 when subjected to the inteligent Identifier operation.

Table 2. 27128 inteligent Identifier Bytes

Identifier \ Pins	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	1	0	0	0	0	0	1	1	83

27128A
ADVANCED 128K (16Kx8) UV ERASABLE PROM

- **200 nsec Typical Access Time —HMOS II*-E Technology**
- **Low Power —100 mA Maximum Active —40 mA Maximum Standby**
- **Two Line Control**
- **inteligent Programming™ Algorithm —Fastest EPROM Programming**
- **inteligent Programming™ Mode —Automated Programming Operations**
- **Compatible with 2764A, 27128, 27256**
- **± 10% V_{CC} Tolerance Available**

The Intel 27128 is a 5V only, 131,072-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 27128A is an advanced version of the 27128 and is fabricated with Intel's HMOS II-E technology which significantly reduces die size and greatly improves the device's performance, power consumption, reliability and producibility.

The typical 27128A access time is 200 ns which is an improvement over the 27128 standard time of 250 ns. This is compatible with high-performance microprocessors, such as Intel's 8 MHz iAPX 186 allowing full speed operation without the addition of WAIT states. The 27128A is also directly compatible with the 12 MHz 8051 family.

Several advanced features have been designed into the 27128A that allow fast and reliable programming—the inteligent Programming Algorithm and the inteligent Identifier Mode. Programming equipment that takes advantage of these innovations will electronically identify the 27128 and then rapidly program it using an efficient programming method.

The 27128 also offers reduced power consumption compared to the 27128. The maximum active current on faster speed parts is 60 mA while the maximum standby current is only 20 mA. The standby mode lowers power consumption without increasing access time.

Two-line control and JEDEC-approved, 28 pin packaging are standard features of all Intel higher density EPROMs. This ensures easy microprocessor interfacing and minimum design efforts when upgrading, adding or choosing between non-volatile memory alternatives.

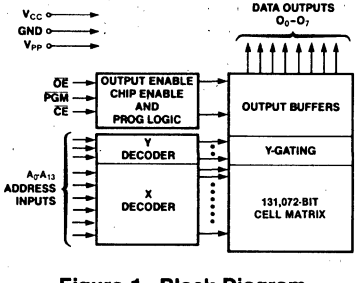
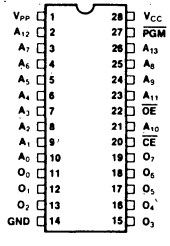


Figure 1. Block Diagram

27256	2764A	2732A	2716
VPP	VPP		
A12	A12		
A7	A7	A7	A7
A6	A6	A6	A6
A5	A5	A5	A5
A4	A4	A4	A4
A3	A3	A3	A3
A2	A2	A2	A2
A1	A1	A1	A1
A0	A0	A0	A0
O0	O0	O0	O0
O1	O1	O1	O1
O2	O2	O2	O2
Gnd	Gnd	Gnd	Gnd

27128A



2716	2732A	2764A	27256
		VCC	VCC
		PGM	A14
VCC	VCC	N.C.	A13
A8	A8	A8	A8
A9	A9	A9	A9
VPP	A11	A11	A11
OE	OE/VPP	OE	OE
A10	A10	A10	A10
CE	CE	CE	CE
O7	O7	O7	O7
O6	O6	O6	O6
O5	O5	O5	O5
O4	O4	O4	O4
O3	O3	O3	O3

NOTE: INTEL "UNIVERSAL SITE"-COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE 27128A PINS.

Figure 2. Pin Configurations

MODE SELECTION

MODE \ PINS	CE (20)	OE (22)	PGM (27)	A9 (24)	VPP (1)	VCC (28)	Outputs (11-13, 15-19)
Read	VIL	VIL	VIH	X	VCC	VCC	DOUT
Output Disable	VIL	VIH	VIH	X	VCC	VCC	High Z
Standby	VIH	X	X	X	VCC	VCC	High Z
Program	VIL	VIH	VIL	X	VPP	VCC	DIN
Verify	VIL	VIL	VIH	X	VPP	VCC	DOUT
Program Inhibit	VIH	X	X	X	VPP	VCC	High Z
inteligent Identifier	VIL	VIL	VIH	VH	VCC	VCC	Code
inteligent Programming	VIL	VIH	VIL	X	VPP	VCC	DIN

1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ± 0.5V

PIN NAMES

A_0-A_{13}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
$\overline{PGM}$	PROGRAM
N.C.	NO CONNECT

*HMOS is a patented process of Intel Corporation

OCTOBER 1983
ORDER NUMBER: 230849-001

PRELIMINARY

27256
256K (32K x 8) UV ERASABLE PROM

- **Software Carrier Capability**
- **250 ns Maximum Access Time**
- **Two-Line Control**
- **int$_e$ligent Identifier™ Mode**
  —Automated Programming Operations
- **TTL Compatible**
- **Industry Standard Pinout . . . JEDEC Approved**
- **Low Power**
  —100 mA max. Active
  — 40 mA max. Standby
- **int$_e$ligent Programming™ Algorithm**
  —Fastest EPROM Programming

The Intel 27256 is a 5V only, 262,144-bit ultraviolet Erasable and Electrically Programmable Read Only Memory (EPROM). Organized as 32K words by 8 bits, individual bytes are accessed in under 250ns. This is compatible with high performance microprocessors, such as the Intel 8MHz iAPX 186, allowing full speed operation without the addition of performance-degrading WAIT states. The 27256 is also directly compatible with Intel's 8051 family of microcontrollers.

The 27256 enables implementation of new, advanced systems with firmware intensive architectures. The combination of the 27256's high density, cost effective EPROM storage, and new advanced microprocessors having megabit addressing capability provides designers with opportunities to engineer user-friendly, high reliability, high-performance systems.

The 27256's large storage capability of 32K bytes enables it to function as a high density software carrier. Entire operating systems, diagnostics, high-level language programs and specialized application software can reside in a 27256 EPROM directly on a system's memory bus. This permits immediate microprocessor access and execution of software and eliminates the need for time consuming disk accesses and downloads.

Several advanced features have been designed into the 27256 that allow for fast and reliable programming —the int$_e$ligent identifier™ mode and the int$_e$ligent Programming™ Algorithm. Programming equipment that takes advantage of these innovations will electronically identify the 27256 and then rapidly program it using an efficient programming method.

Two-line control and JEDEC-approved, 28-pin packaging are standard features of all Intel high-density EPROMs. This assures easy microprocessor interfacing and minimum design efforts when upgrading, adding, or choosing between nonvolatile memory alternatives.

The 27256 is manufactured using Intel's advanced HMOS*II-E technology.

*HMOS is a patented process of Intel Corporation.

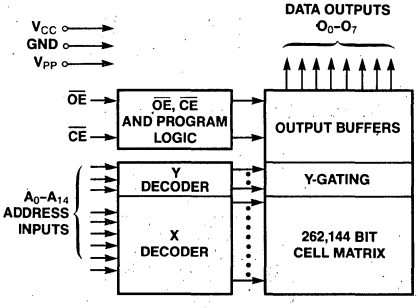
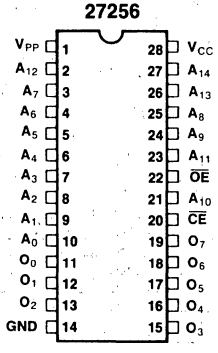


Figure 1. Block Diagram



PIN NAMES	
A_0-A_{14}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS

Figure 2. Pin Configuration

OCTOBER 1983
ORDER NO. 210827-006

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +125°C
All Input or Output Voltages with
Respect to Ground .....+6.25 V to −0.6V
Voltage on Pin 24 with
Respect to Ground .....+13.5V to −0.6V
V_{PP} Supply Voltage with Respect
to Ground .....+14.0 V to −0.6V

**NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	27256	27256-3	27256-4	27256-25	27256-30	27256-45
Operating Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 10%	5V ± 10%	5V ± 10%

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[3]	Max.		
I_{LI}	Input Load Current			10	µA	V_{IN} = 5.5V
I_{LO}	Output Leakage Current			10	µA	V_{OUT} = 5.5V
I_{PP1}[2]	V_{PP} Current Read/Standby			5	mA	V_{PP} = 5.5V
I_{CC1}[2]	V_{CC} Current Standby		20	40	mA	$\overline{CE}$ =V_{IH}
I_{CC2}[2]	V_{CC} Current Active		45	100	mA	$\overline{CE}$ =$\overline{OE}$ =V_{IL} V_{PP} =V_{CC}
V_{IL}	Input Low Voltage	−.1		+.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+1	V	
V_{OL}	Output Low Voltage			.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −400 µA
V_{PP}[2]	V_{PP} Read Voltage	3.8		V_{CC}	V	V_{CC}= 5.0V ± 0.25V

READ OPERATION

A.C. CHARACTERISTICS

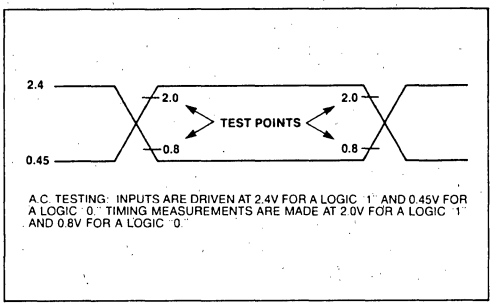
Symbol	Parameter	27256-25 & 27256 Limits		27256-30 & 27256-3 Limits		27256-45 & 27256-4 Limits		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
t_{ACC}	Address to Output Delay		250		300		450	ns	$\overline{CE}=\overline{OE}=V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		250		300		450	ns	$\overline{OE}=V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		100		120		150	ns	$\overline{CE}=V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Float	0	60	0	105	0	130	ns	$\overline{CE}=V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		ns	$\overline{CE}=\overline{OE}=V_{IL}$

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for t_A =25°C and nominal supply voltages.
4. 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 on the following page.

CAPACITANCE (T_A = 25°C, f = 1 MHz)

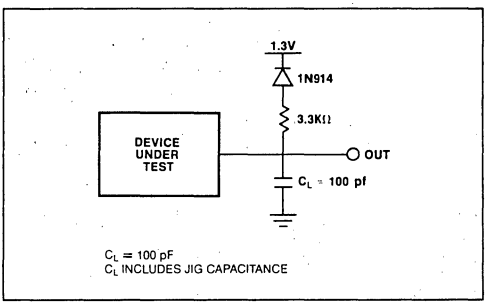
Symbol	Parameter	Typ.[1]	Max.	Unit	Conditions
C_{IN}[2]	Input Capacitance	4	6	pF	V_{IN}=0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT}=0V

A.C. TESTING INPUT/OUTPUT WAVEFORM



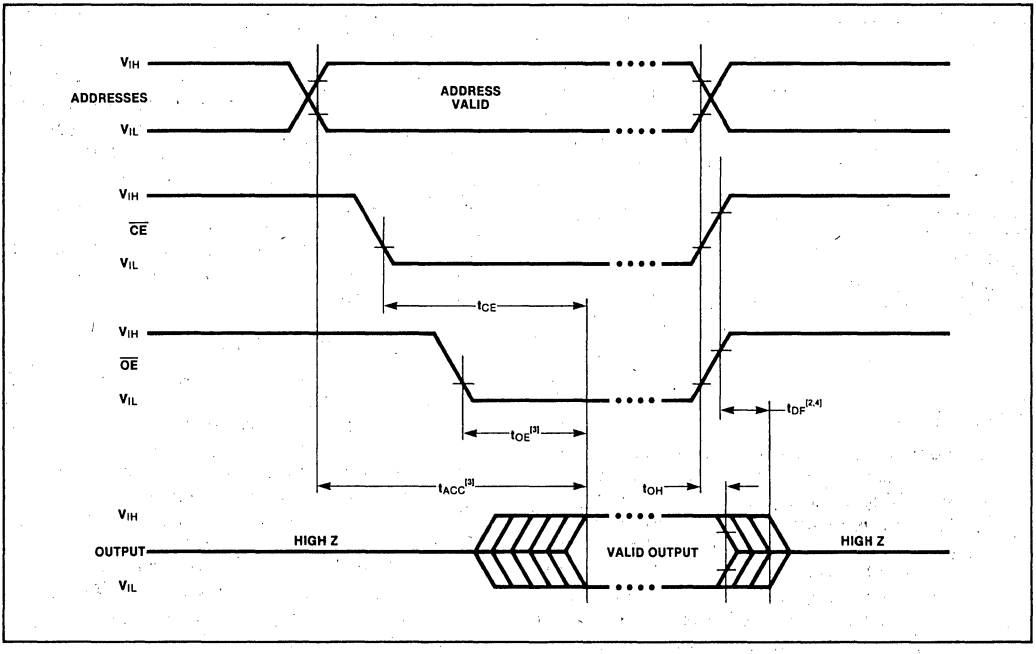
A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0". TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0".

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



NOTES:
1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.
3. $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{ACC}.
4. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first.

DEVICE OPERATION

The eight modes of operation of the 27256 are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for V_{PP} and 12V on A9 for int$_e$ligent identifier mode.

Table 1. Operating Modes

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	OUTPUTS (11-13, 15-19)
Read	V_{IL}	V_{IL}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	V_{CC}	V_{CC}	High Z
int$_e$ligent Programming	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{OUT}
Optional Verify	V_{IL}	V_{IL}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	V_{IH}	X	V_{PP}	V_{CC}	High Z
int$_e$ligent Identifier	V_{IL}	V_{IL}	V_H	V_{CC}	V_{CC}	Code

NOTES:
1. X can be V_{IH} or V_{IL}
2. $V_H = 12.0V \pm 0.5V$

READ MODE

The 27256 has two control functions, both of which must be logically active in order 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. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

STANDBY MODE

The 27256 has a standby mode which reduces the maximum active current from 100 mA to 40 mA. The 27256 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Two Line Output Control

Because EPROMs are usually used in larger memory arrays, Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

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

To use these two control lines most efficiently, $\overline{CE}$ (pin 20) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

System Considerations

The power switching characteristics of HMOS II-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer —the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive and inductive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control and by

properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effects of PC board traces.

PROGRAMMING

Caution: Exceeding 14V on pin 1 (V_{PP}) will permanently damage the 27256.

Initially, and after each erasure, all bits of the 27256 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 27256 is in the programming mode when the V_{PP} input is at 12.5V and $\overline{CE}$ is at TTL-low. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

inteligent Programming™ Algorithm

The 27256 inteligent Programming Algorithm rapidly programs Intel 27256 EPROMS using an efficient and reliable method particularly suited to the production programming environment. Typical programming times for individual devices are on the order of five minutes. Programming reliability is also ensured as the incremental program margin of each byte is continually monitored to determine when it has been successfully programmed. A flowchart of the 27256 inteligent Programming Algorithm is shown in Figure 3.

The inteligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial $\overline{CE}$ pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 3X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular 27256 location, before a correct verify occurs. Up to 25 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0 V and V_{PP} = 12.5V. When the inteligent Programming cycle has been completed, all bytes should be compared to the original data with V_{CC} = V_{PP} = 5.0V.

Program Inhibit

Programming of multiple 27256s in parallel with different data is easily accomplished by using the Program Inhibit mode. A high-level $\overline{CE}$ input inhibits the other 27256s from being programmed.

Except for $\overline{CE}$ and $\overline{OE}$, all like inputs of the parallel 27256s may be common. A TTL low-level pulse applied to the $\overline{CE}$ input with V_{PP} at 12.5V will program the selected 27256.

Verify

A verify should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with $\overline{OE}$ at V_{IL}, $\overline{CE}$ at V_{IH} and V_{PP} at 12.5V.

Optional Verify

The optional verify may be performed in place of the verify mode. It is performed with $\overline{OE}$ at V_{IL}, $\overline{CE}$ at V_{IL} (as opposed to the standard verify which has $\overline{CE}$ at V_{IH}), and V_{PP} at 12.5V. The outputs will tri-state according to the signal presented to $\overline{OE}$. Therefore, all devices with V_{PP}=12.5V and $\overline{OE}$=V_{IL} will present data on the bus independent of the $\overline{CE}$ state. When parallel programming several devices which share a common bus, V_{PP} should be lowered to V_{CC} (=6.0V) and the normal read mode used to execute a program verify.

inteligent Identifier™ Mode

The inteligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ± 5°C ambient temperature range that is required when programming the 27256.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 24) of the 27256. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 10) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during inteligent Identifier Mode.

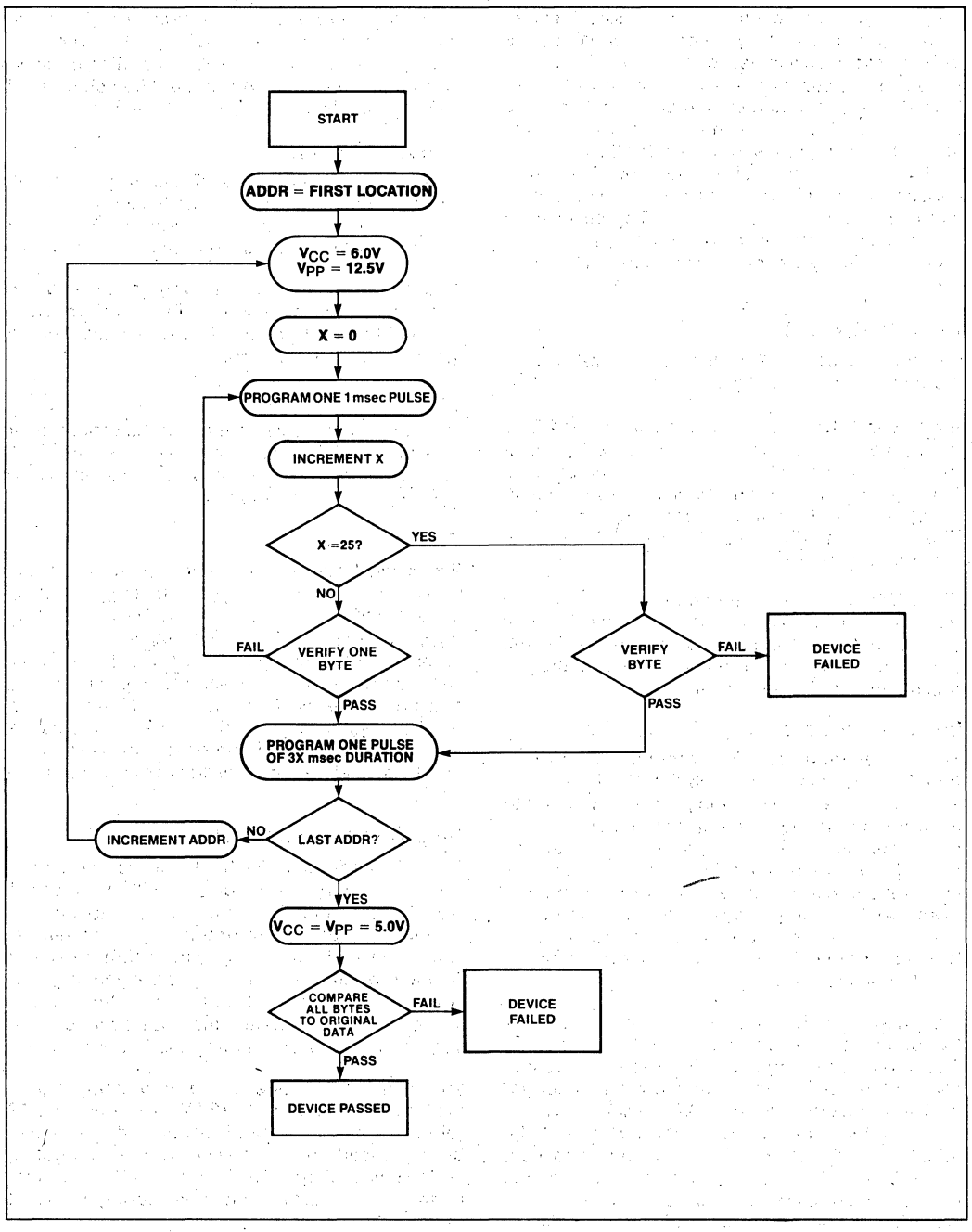


Figure 3. 27256 int$_e$ligent Programming™ Flowchart

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 27256, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

ERASURE CHARACTERISTICS

The erasure characteristics of the 27256 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000 Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 27256 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 27256 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 27256 window to prevent unintentional erasure.

The recommended erasure procedure for the 27256 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 Wsec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The 27256 should be placed within 1 inch of the lamp tubes during erasure. The maximum integrated dose a 27256 can be exposed to without damage is 7258 Wsec/cm^2 (1 week @ 12000 μW/cm^2). Exposure of the 27256 to high intensity UV light for long periods may cause permanent damage.

Table 2. 27256 inteligent Identifier™ Bytes

Identifier \ Pins	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	VIH	0	0	0	0	0	1	0	0	04

NOTES:
1. A_9 = 12.0V ± 0.5V
2. A_1–A_8, A_{10}–A_{13}, $\overline{CE}$, $\overline{OE}$ = V_{IL}
3. A_{14} = V_{IH} or V_{IL}

inteligent Programming™ Algorithm

D.C. PROGRAMMING CHARACTERISTICS:

T_A = 25 ± 5°C, V_{CC} = 6.0V ± 0.25V, V_{PP} = 12.5V ± 0.5V

Symbol	Parameter	Limits			Test Conditions (see Note 1)
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	V_{IN} = V_{IL} or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input High Level	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	I_{OH} = −400 μA
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		50	mA	$\overline{CE}$ = V_{IL}
V_{ID}	A_9 inteligent Identifier Voltage	11.5	12.5	V	

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.

A.C. PROGRAMMING CHARACTERISTICS:
$T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 12.5V \pm 0.3V$

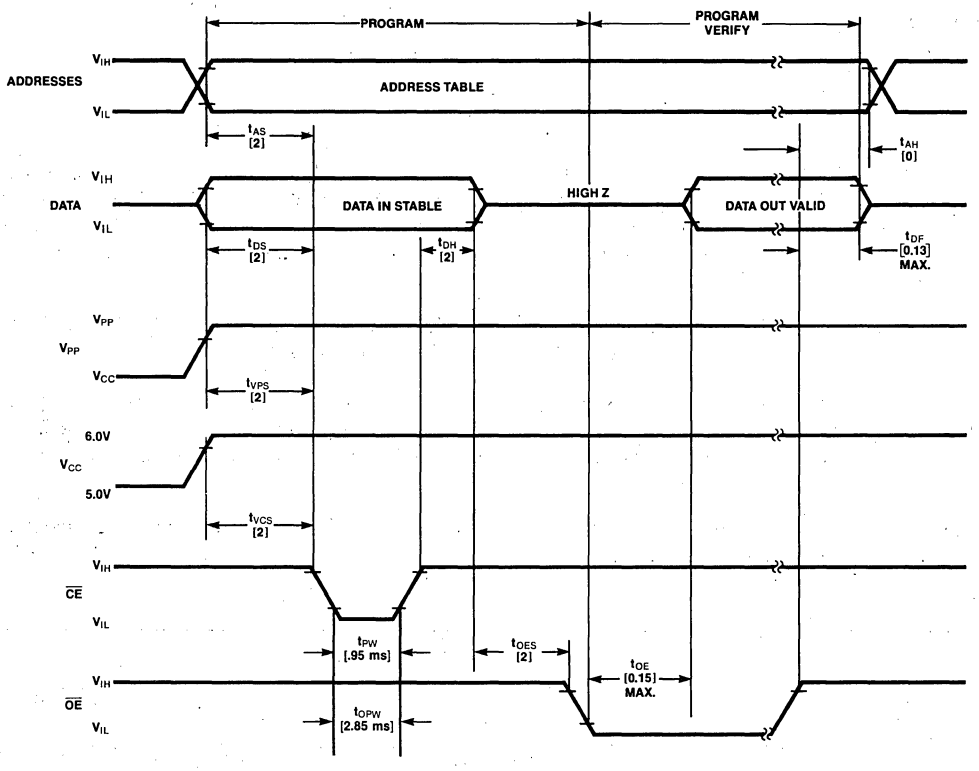
Symbol	Parameter	Limits				Test Conditions* (see Note 1)
		Min.	Typ.	Max.	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}[4]	$\overline{OE}$ High to Output Float Delay	0		130	ns	
t_{VPS}	V_{PP} Setup Time	2			μs	
t_{VCS}	V_{CC} Setup Time	2			μs	
t_{PW}	$\overline{CE}$ Initial Program Pulse Width	0.95	1.0	1.05	ms	(see Note 3)
t_{OPW}	$\overline{CE}$ Overprogram Pulse Width	2.85		78.75	ms	(see Note 2)
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

***A.C. CONDITIONS OF TEST**

Input Rise and Fall Times (10% to 90%) ... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ... 0.8V and 2.0V

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The length of the overprogram pulse may vary from 2.85 msec to 78.75 msec as a function of the iteration counter value X.
3. Initial Program Pulse width tolerance is 1 msec ±5%.
4. 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 on the following page.

int$_e$ligent Programming™ WAVEFORMS



NOTES:
1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR A V_{IL} AND 2V FOR A V_{IH}.
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 27256 A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

UV ERASABLE PROM FAMILY

EXPRESS

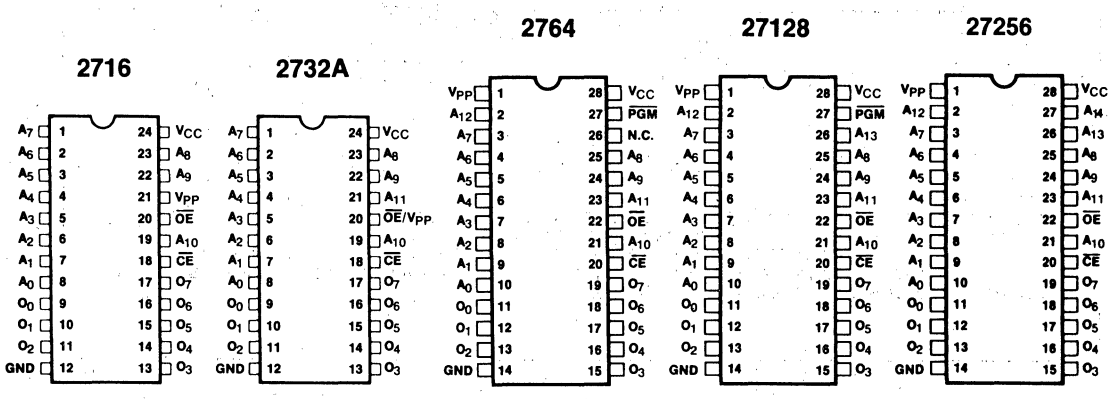
- **0–70°C Temperature Range Standard**
- **Extended Temperature Range −40°C – +85°C Available**
- **Two Line Control**
- **168±8 Hour Burn-in Available**
- **Industry Standard Pinout . . . JEDEC Approved**
- **Inspected To 0.1% AQL**

The Intel EXPRESS EPROM family is a series of ultraviolet erasable and electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. Intel's JEDEC approved 28 pin Universal Memory Socket provides the industry standard upgrade path to higher density EPROMs.

EXPRESS EPROM products are available with 168±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in.

The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to 85°C) EXPRESS products are available. EXPRESS products plus military grade EPROMs (−55°C to 125°C) provide the most complete choice of standard and extended temperature range EPROMs available.

Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.



PIN CONFIGURATION

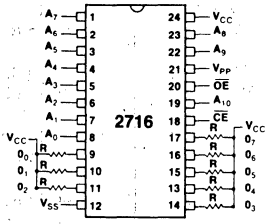
OCTOBER 1983
ORDER NUMBER: 210322-002

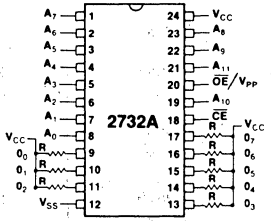
EXPRESS
EPROM Product Family

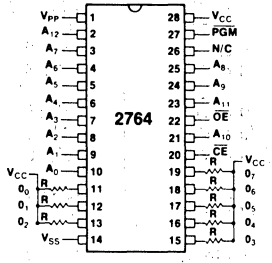
Type	Organization	Maximum Access (ns)	Power Supply	Operating Temperature (°C)	Burn-in 125°C (hr)
QD2716-1	2048x8	350	5V ± 10%	0 to 70	168±8
QD2716-2	2048x8	390	5V ± 5%	0 to 70	168±8
QD2716	2048x8	450	5V ± 5%	0 to 70	168±8
LD2716	2048x8	450	5V ± 5%	-40 to 85	168±8
TD2716	2048x8	450	5V ± 5%	-40 to 85	NONE
QD2732A-2	4096x8	200	5V ± 5%	0 to 70	168±8
QD2732A	4096x8	250	5V ± 5%	0 to 70	168±8
QD2732A-3	4096x8	300	5V ± 5%	0 to 70	168±8
QD2732A-4	4096x8	450	5V ± 5%	0 to 70	168±8
QD2732A-20	4096x8	200	5V ± 10%	0 to 70	168±8
QD2732A-25	4096x8	250	5V ± 10%	0 to 70	168±8
QD2732A-30	4096x8	300	5V ± 10%	0 to 70	168±8
LD2732A	4096x8	250	5V ± 5%	-40 to 85	168±8
LD2732A-4	4096x8	450	5V ± 5%	-40 to 85	168±8
LD2732A-25	4096x8	250	5V ± 10%	-40 to 85	168±8
LD2732A-45	4096x8	450	5V ± 10%	-40 to 85	168±8
TD2732A	4096x8	250	5V ± 5%	-40 to 85	NONE
TD2732A-4	4096x8	450	5V ± 5%	-40 to 85	NONE
TD2732A-25	4096x8	250	5V ± 10%	-40 to 85	NONE
TD2732A-45	4096x8	450	5V ± 10%	-40 to 85	NONE
QD2764-2	8192x8	200	5V ± 5%	0 to 70	168±8
QD2764	8192x8	250	5V ± 5%	0 to 70	168±8
QD2764-3	8192x8	300	5V ± 5%	0 to 70	168±8
QD2764-4	8192x8	450	5V ± 5%	0 to 70	168±8
QD2764-25	8192x8	250	5V ± 10%	0 to 70	168±8
QD2764-30	8192x8	300	5V ± 10%	0 to 70	168±8
QD2764-45	8192x8	450	5V ± 10%	0 to 70	168±8
LD2764	8192x8	250	5V ± 5%	-40 to 85	168±8
LD2764-4	8192x8	450	5V ± 5%	-40 to 85	168±8
LD2764-25	8192x8	250	5V ± 10%	-40 to 85	168±8
LD2764-45	8192x8	450	5V ± 10%	-40 to 85	168±8
TD2764	8192x8	250	5V ± 5%	-40 to 85	NONE
TD2764-4	8192x8	450	5V ± 5%	-40 to 85	NONE
TD2764-25	8192x8	250	5V ± 10%	-40 to 85	NONE
TD2764-45	8192x8	450	5V ± 10%	-40 to 85	NONE
QD27128	16384x8	250	5V ± 5%	0 to 70	168±8
QD27128-3	16384x8	300	5V ± 5%	0 to 70	168±8

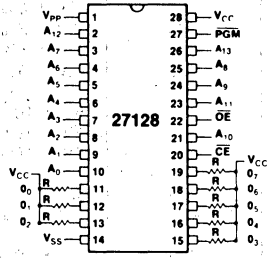
EXPRESS
EPROM Product Family
(Cont.)

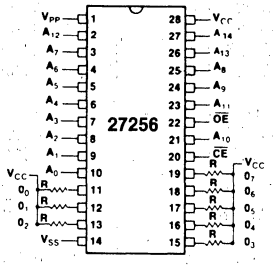
Type	Organization	Maximum Access (ns)	Power Supply	Operating Temperature (°C)	Burn-in 125°C (hr)
QD27128-4	16384x8	450	5V ± 5%	0 to 70	168 ± 8
QD27128-25	16384x8	250	5V ± 10%	0 to 70	168 ± 8
QD27128-45	16384x8	450	5V ± 10%	0 to 70	168 ± 8
LD27128	16384x8	250	5V ± 5%	– 40 to 85	168 ± 8
LD27128-4	16384x8	450	5V ± 5%	– 40 to 85	168 ± 8
LD27128-45	16384x8	450	5V ± 10%	– 40 to 85	168 ± 8
TD27128	16384x8	250	5V ± 5%	– 40 to 85	NONE
TD27128-4	16384x8	450	5V ± 5%	– 40 to 85	NONE
TD27128-45	16384x8	450	5V ± 10%	– 40 to 85	NONE
QD27256	32768x8	250	5V ± 5%	0 to 70	168 ± 8
QD27256-25	32768x8	250	5V ± 10%	0 to 70	168 ± 8
LD27256	32768x8	250	5V ± 10%	– 40 to 85	168 ± 8
LD27256-25	32768x8	250	5V ± 5%	– 40 to 85	168 ± 8
TD27256	32768x8	250	5V ± 5%	– 40 to 85	None
TD27256-25	32768x8	250	5V ± 10%	– 40 to 85	None

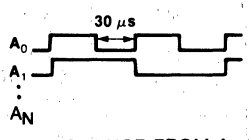












BINARY SEQUENCE FROM A_0 to A_N

Figure 1. Burn-In Bias and Timing Diagrams

READ OPERATION

D.C. AND A.C. CHARACTERISTICS

Electrical Parameters of EXPRESS EPROM products are identical to standard data sheet parameters except for:

Symbol	Parameter	Limits										Test Conditions
		TD2716 LD2716		TD2732A LD2732A		TD2764 LD2764		TD27128 LD27128		TD27256 LD27256		
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
T_{OE}	Output Enable to Output Delay (ns)		150									$\overline{CE} = V_{IL}$
t_{DF}	Output Enable to Output Float (ns)	0	130									$\overline{CE} = V_{IL}$
I_{CC1}	V_{CC} Standby Current (mA)				45		50		50		50	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC2}	V_{CC} Active Current (mA)				150		125		125		125	$\overline{OE} = \overline{CE} = V_{IL}$

E^2PROMs & NVRAMs (In Circuit Programmable Writable Memories)

5

December 1982

Reliability Aspects of a Floating Gate E^2PROM

Bruce Euzent, Nick Boruta,
Jimmy Lee and Ching Jenq
Intel Corporation

Based on presentation at 1981 International Reliability Physics Symposium, Orlando, Florida, April 7, 1981.

INTRODUCTION

Electrically Erasable Programmable Read Only Memories (E^2PROMs) that can be electrically erased and written one byte at a time are new components being used in computer systems. The E^2PROM is particularly attractive in applications requiring field update of program store memory or non-volatile data capture. It is only recently that E^2PROMs which operate via Fowler-Nordheim tunneling to a floating polysilicon gate have become available. The E^2PROM has the data retention requirements of earlier generations of PROMs, but also must maintain its field-programmable characteristics over its device life.

In this paper we shall first review the basic operation of the Intel 2816 E^2PROM cell. Intrinsic failure mechanisms which limit the applications of E^2PROMs will be examined, and then defect mechanisms will be discussed. Finally lifetest data will be presented to predict operating failure rates.

Device Operation

The Intel 2816 uses the FLOTOX structure, which has been discussed in detail in previous literature[1]. Basically, it utilizes an oxide of less than 200Å thick between the floating polysilicon gate and the N+ region as shown in Figure 1.

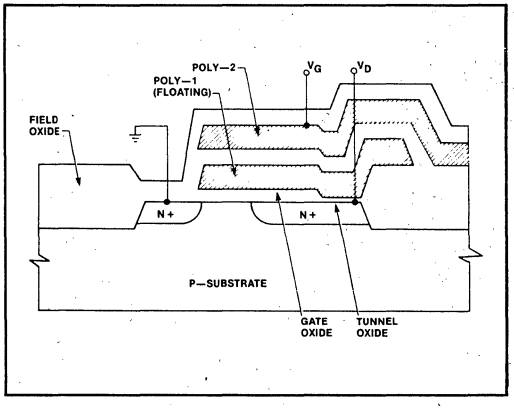


Figure 1. FLOTOX Device Structure Cross Section

Both erase and write are accomplished by tunneling the electrons through thin oxide using the Fowler-Nordheim mechanism[2]. The I-V characteristic of Fowler-Nordheim tunneling is shown in Figure 2, where the current is approximately exponentially dependent on the electric field applied to the oxide.

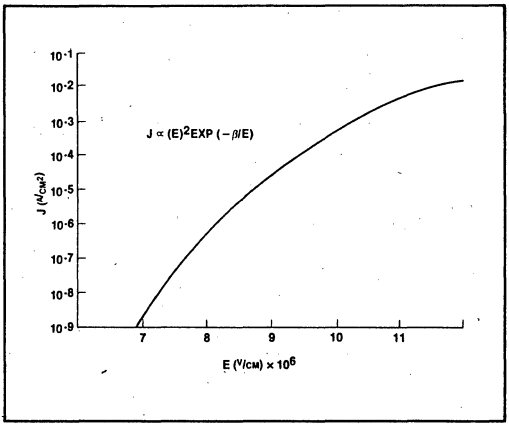


Figure 2. Fowler-Nordheim Tunneling I-V Characteristic

During the erase operation, approximately 20V is applied to the top gate of each cell in the byte while the drain is kept at ground potential. The electrical field in the thin oxide region is directed from the floating gate to the N+ region such that electrons tunnel through the oxide and are stored on the floating gate. This shifts the cell threshold in the positive direction causing the cell to shut off current flow and present a logical "1" at its output (as seen in Figure 3a).

On the other hand, when the cell is written to logic "0", the top gate is pulled down to ground potential and a high voltage is applied to the drain (with the source end floating). Electrons are depleted from the floating gate

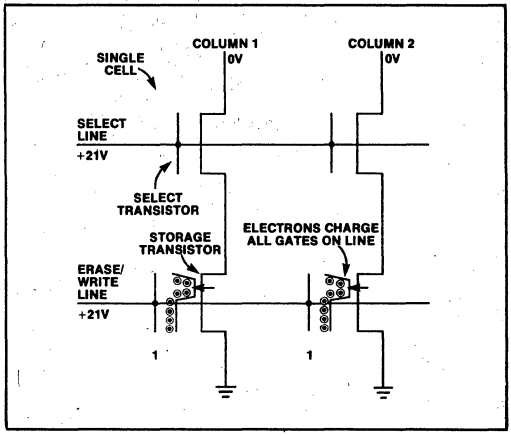


Figure 3a. Schematic of Memory Cell Operation During Erase

as seen in Figure 3b, and the cell is left with a negative threshold. Since the interpoly oxide is much thicker than the "tunnel oxide" and the electric field across the interpoly oxide is much smaller, the erase and write operations are predominantly controlled by the thin oxide region.

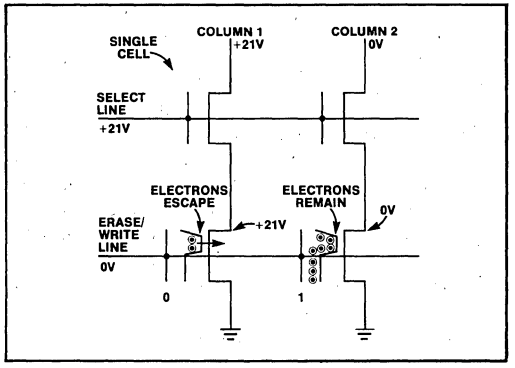


Figure 3b. Schematic of Memory Cell Operation During Write

Read Retention

The floating gate structure is known for its excellent charge retention properties. The reliability of this structure in the case of the EPROM device has been reported before[3]. The only remaining concern of the data retentivity of the 2816 is possible charge gain or loss through the tunnel oxide due to Fowler-Nordheim tunneling. The maximum electric field is built up across the tunnel oxide for a written cell, one that has a net positive charge on the floating gate. In this state the positive top gate voltage creates an electric field which adds to the field created by the positive charge on the floating gate, and there exists the probability that electrons may tunnel to the floating gate and shift the cell threshold. The band diagram of this condition is shown in Figure 4. However, the amount of current which may pass through the thin oxide during read or deselect is kept low by biasing the top gate of the memory cell at an internally generated voltage less than V_{CC}. The effect on the threshold shift of the cell can only be observed after long-term stress. Under this condition, the accelerated voltage test can be very useful.

If we assume Fowler-Nordheim tunneling is the predominant mechanism governing the movement of electrons, the threshold shift of the cell will be dependent solely on the voltage between the top gate and the N+ region. This has been proven to be true in both simulations and experiments, where we found that

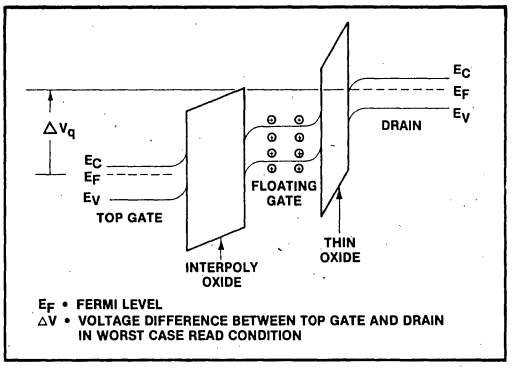


Figure 4. Band Diagram During Read of Written Cell

there is a one-to-one relationship between the V_T and the stress voltage. In other words, we can stress the device by applying a higher voltage to the top gate such that the change of the threshold voltage can be measured. The data then will be used to predict the same characteristics at the much lower normal read voltage. In Figure 5, the aforementioned simulation and experimental data are shown. The cell was biased at a voltage 4V higher than the normal read condition and the threshold voltage of the cell was monitored over a period of a week. A simulation was also generated to compare with the observed threshold shift and to demonstrate the technique we use to predict whether the data retention of the cell is accurate. As can be seen in the Figure 4, even under the accelerated voltage test the cell V_T still will not cross above the sense level after more than 10 years. Similar data has also been taken by writing the cell to a more negative initial threshold. In this case, the shift of the threshold can be observed at a stress of normal read voltage. Clearly, a 1V/1V relationship holds and an extrapolation can be made that the correct data will be retained for more than 10 years of continuous read.

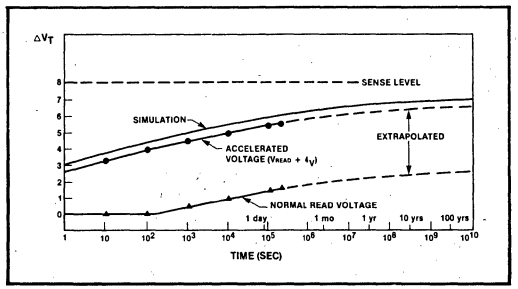


Figure 5. Single Cell Threshold Voltage Shift vs. Log Time During Read of a Written Cell

Intrinsic Charge Trapping

An ideal feature of a tunneling dielectric is that it should never remember the number of electrons that passed through it or the voltage that was previously applied across the film. Unfortunately, for thermally grown SiO_2 there always exists a certain number of electron and hole traps[4–9]. When these traps are occupied the net charging state of the tunnel oxide will be changed and thus cause the tunneling current across the film to vary if the applied voltage has remained the same.

Figure 6 plots the threshold voltage of a 2816 cell in erase (charged) and write (discharged) states as a function of erase/write cycles. The solid line is for a single cell, while the dashed line is for a typical 2816 array. It is seen that the threshold window, defined as the difference between the erase and write threshold, is slightly increased in the first few E/W cycles and then saturates and remains almost constant until 10^4 cycles. From that point, the window begins to narrow gradually until around 10^6 cycles where the window is collapsed.

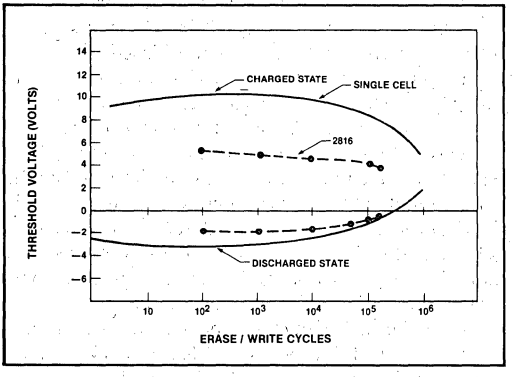


Figure 6. Typical Cell and Device Window vs. Log Cycles

Our study shows that the behavior of the widening and narrowing of the threshold window can be explained by charge trapping in tunnel oxide. The window widening effect is found to be caused by the following mechanism:

Assume a cell is to be erased following a write cycle. During the preceding write cycle, the floating gate is biased negatively relative to the substrate. A layer of positive charge will be formed, either through the tunneling of holes from Si into SiO_2 or electrons in the reverse direction. These positive charges are in general at 20–30Å away from the SiO_2/Si interface, as in Figure 7a. At the beginning of the erase step, the positive charges will cause an increase in electric field at the injection interface, i.e., SiO_2/Si interface, as shown in Figure 7b. This will in turn increase the tunneling current to the floating gate, where the amount of stored electrons is thus increased, causing the erase threshold to increase. During the erase cycle, however, the polarity of bias voltage across the tunnel oxide will cause the positive charge at SiO_2/Si interface to be neutralized through the reverse tunneling mechanism that forms these charges. At the same time a new layer of positive charges is formed near the anode[11, 12], i.e., poly/SiO_2 interface, as shown in Figure 7c. These charges will then cause the write threshold to increase through the same mechanism as that discussed for the erase threshold. In addition to positive charge trapping, our study also shows that there is a uniform distribution of electron traps throughout the oxide[11, 12]. When the cell is erased or written, electrons are injected through the oxide and some of them will be captured by these traps,

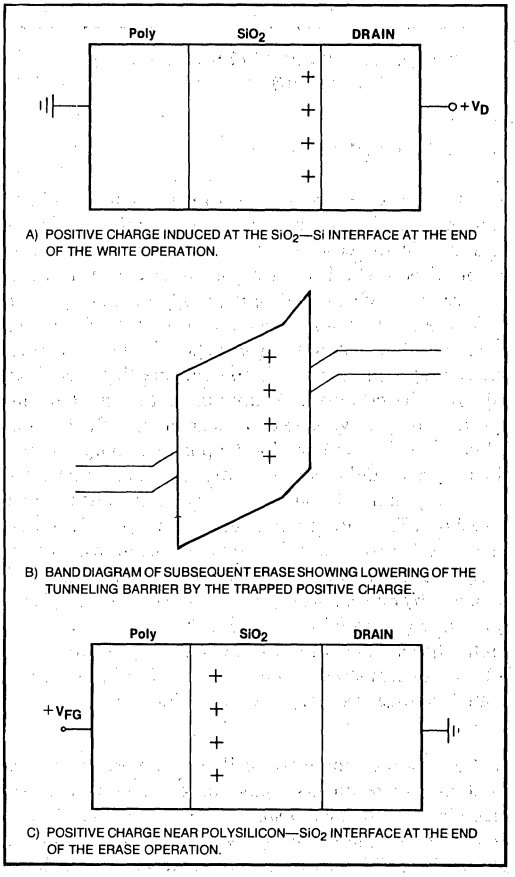


Figure 7. Threshold Window Widening

causing the build-up of negative charges in the oxide, as shown in Figure 8. The negative charges will reduce the electric field at the injection interface, thus decreasing the tunneling current and causing the threshold window to narrow. It has been found that the electron traps are not only preexisting in the oxide but also generated during the E/W cycles[9-12] because of the high field stress and the accompanying high current flow. The non-saturated build-up of negative charges, because of the continuous generation of electrons traps will finally cause the threshold window to collapse.

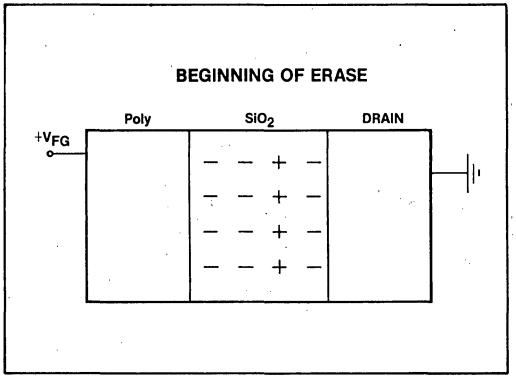


Figure 8. Negative Charges Trapped Uniformly Across Tunnel Oxide

Defect Charge Loss

EPROMs have been shown to have excellent data retention[3]. In this section we will discuss data retention studies that have been performed on the Intel 2816 E^2PROM. Since in E^2PROMs the number of Erase/Write cycles during the device lifetime is 3 to 4 orders of magnitude greater than in the EPROM, we will also need to address the effects of cycling on data retention.

As in the case of EPROMs the charge loss from the floating gate can be described as either intrinsic or defect-related. We will discuss the defect-related charge loss since the intrinsic charge loss on a typical device is identical to the EPROM and has been described before[3].

Analysis of cells exhibiting defect-related charge loss shows that the leakage current has an exponential dependence on the potential of the floating gate. This is different from the EPROM where defect leakage current exhibits a linear (ohmic) dependence on voltage.[3] The exponential dependence is indicative of electron tunneling. The effect of the defect, then, is the lowering or narrowing of the thin oxide barrier, allowing tunneling to occur at voltage differences between the floating gate and the drain that would ordinarily be insufficient to support tunneling.

Erase/write cycling effects on data retention were studied by comparing 250°C retention before cycling to that after 10,000 cycles. Figure 9 shows a plot of the cumulative % data retention failure during 500 hours 250°C retention bake. Data from the Intel 2716 EPROM is included as a comparison. From this data it is clear that the retention failure rate closely resembles that of the Intel 2716 EPROM.

Since the defect charge loss failure mechanism is temperature activated it is simple to construct screens on a production basis for these types of failures similar to those used on EPROMs.

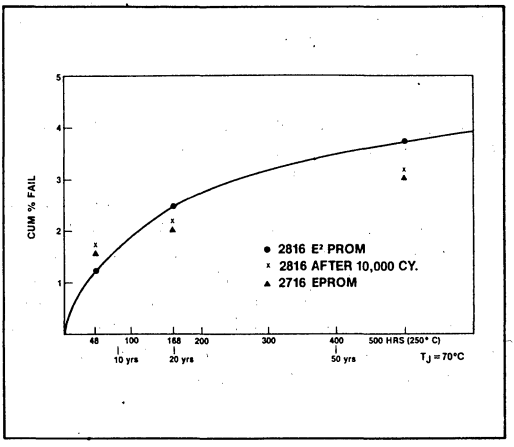


Figure 9. Intel 2816 Data Retention at 250°C, Percent Fail vs. Time

Accelerated Test Results

An E^2PROM has an additional reliability requirement over standard PROMs. Besides the integrity of data retention, an E^2PROM must withstand up to 10,000 erase and write programming pulses per byte. Besides the previously discussed window closing phenomenon there are reliability considerations due to high voltage operation. Dielectric breakdown[13] is a common MOS failure mechanism, which has been shown to be highly voltage accelerated. The reliability of the Intel 2816 during erase/write cycles was measured by performing the full number of erase/write cycles on each byte. Erase/write cycling was done at 70°C and 25°C with no difference in observed failure rate between these temperatures.

The results of erase/write cycling are shown in Figure 10A. The devices under test are completely tested after 2,000, 5,000 and 10,000 total cycles on each byte. The devices are programmed to several data patterns and tested to data sheet specifications. In addition, the devices are tested for high temperature data retention. As can be seen from Figure 10A, the failure rate per 1000 cycles decreases as a function of the number of cycles, which is typical for defect mechanisms such as dielectric breakdown.[13] From the time the initial data was gathered in 1981, recent data (Figure 10A) has shown the failure rate to have been reduced by a factor of 2.

Two major types of failures were found: Tunnel oxide breakdown and oxide breakdown in the row select circuitry. These failures were minimized by using standard screening techniques for oxide breakdown. Figure 10B shows the failure mode distribution found during erase/write cycling of 549 devices.

Tunneling oxide breakdown failures are cells which fail either to program or to retain data following programming due to conduction through the thin oxide at low electric fields. In the case of the programming failures, the breakdown extends all the way through the oxide layer. The data retention failures exhibit characteristics similar to those of the defect charge loss failures discussed in the previous section and are probably due to a partially broken down oxide layer. Further cycling of this type of retention failure has been found to result in it becoming a programming failure.

Table I shows expected failure rates in %/1000 hours at a 60% upper confidence level based on expected device life and the average number of cycles per byte. In a typical system it is expected that some bytes will be written more often than others, so these failure rates serve as a guideline.

As can be seen in Table II, acceptable failure rates are achieved for the design goal of 10,000 erase/write cycles per byte. To achieve 10,000 cycles per byte in ten (10) years, each byte must be altered approximately three times per day.

As a final verification of device reliability a standard high temperature lifetest at 125°C was performed on devices programmed with a checkerboard data pattern. The lifetest was performed on devices with no additional cycles and devices with 10,000 cycles on each

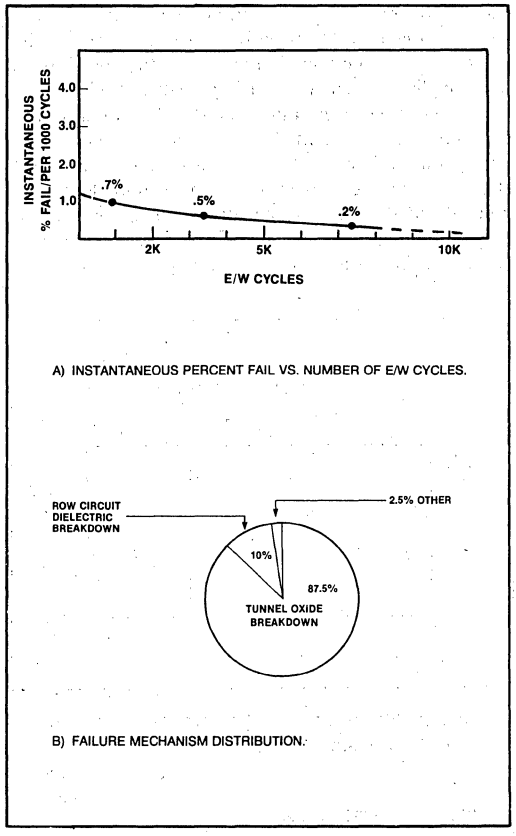


Figure 10. Erase/Write Cycling Results

Table I. Erase/Write Cycling Failure Rate (per 1000 hours at a 60% UCL)

Device Life	No. of Cycles		
	2000	5000	10,000
5 years	.035	.06	.092
10 years	.017	.029	.047
20 years	.009	.017	.023

Table II. 125°C Lifetest Results

Cycles	48 Hrs	168 Hrs	500 Hrs	1000 Hrs	2000 Hrs
0	0/1422	1/1422[a]	1/443[b]	0/429	0/270
10,000	0/336	0/336	0/336	0/150	—
Total	0/1758	1/1758	1/779	0/579	0/270

Failure Analysis:
a) = Non-repeatable charge gain, contamination, lev.
b) = Input leakage, contamination, lev.

byte. As can be seen from the data in Table II standard MOS failure mechanisms were observed. This data is significant in that it shows no additional defect mechanisms related to data retention or erase/write cycling of the Intel 2816 E^2PROM.

Failure rate predictions are made in Table III at a 60% upper confidence level for both 55°C and 70°C operation. The .013%/1000 hrs. failure rate at 55°C shows good reliability comparable to other semiconductor memories.

Table III. Failure Rate Predictions at a 60% U.C.L.

125°C Device Hrs.	Activation Energy	Equivalent Hours		Lifetest Failures	Failure Rate % per 1000 Hrs.	
		55°	70°		55°C	70°
$3.2x10^6$	0.3 eV	$2.1x10^7$	$1.3x10^7$	0	.004	.007
$3.2x10^6$	0.6 eV	$1.3x10^8$	$5.3x10^7$	0	.001	.002
$3.2x10^6$	1.0 eV	$1.6x10^9$	$3.4x10^8$	2	.000	.001
				Combined	.005	.010

SUMMARY

This paper has discussed a number of E^2PROM failure mechanisms for both erase/write cycling and data retention. It has been shown that Fowler-Nordheim tunneling used for programming does not affect data retention. Erase/write cycling has been shown to degrade device margins by only a small amount and is easily guardbanded. Erase/write cycling does contribute to a significant portion of the observed failure rate due to oxide breakdown under high field operation. Finally, it has been shown that E^2PROMs can perform reliably in applications requiring up to 10,000 erase/write cycles per byte.

REFERENCES

1. W. S. Johnson, et al, "16-K EE-PROM Relies on Tunneling for Byte-Eraseable Program Storage," *Electronics*, February 28, 1980, p. 113–117.
2. R. Williams, *Phys. Rev.*, Vol. 140, p. 569, 1965.
3. R. E. Shiner, J. M. Caywood, B. L. Euzent, "Data Retention in EPROMs," 1980 Proceedings of the 18th Annual Reliability Physics Symposium, p. 238–243.
4. E. H. Nicollian, C. N. Berglund, P. F. Schmidt, I. M. Andrews, *J. Appl. Phys.*, Vol. 42, p. 5654, 1971.
5. M. H. Woods and R. Williams, *J. Appl. Phys.*, p 47, 1082, 1976.
6. W. C. Johnson, *IEEE Trans. Nucl. Sci.*, p. NS–22, 2144, 1975.
7. D. J. DiMaria, Proceeding of the International Topical Conference on the Physics of SiO_2 and its Interfaces, p. 160, 3/78.
8. E. Harari, *Appl. Phys. Letter*, p. 30, 601, 1977.
9. C. S. Jenq, "High Field Generation of Interface States and Electron Traps in MOS Capacitors," Ph.D. dissertation, Princeton Univ., 12/77.
10. C. S. Jenq, W. C. Johnson, "High Field Generation of Electron Traps in MOS Capacitors," presented in Semiconductor Interface Specialist Conference, 12/77.
11. C. S. Jenq, T. R. Rangarath, C. H. Huang, "Charge Trapping in Floating Gate Tunnel Oxide" presented in 1980 Non-Volatile Semiconductor Memory workshop to be published.
12. D. R. Young, "Electron Trapping in SiO_2", presented in 1980 Non-Volatile Semiconductor Memory workshop.
13. D. Crook, "Method of Determining Reliability Screen for Time Dependent Dielectric Breakdown," 1979 17th Annual Reliability Physics Symposium Proceedings, p. 1-5.

October 1983

Designing with the System Friendly 2817A 5V-Only E^2PROM

Justin Orion
Applications Engineer
Intel Corporation

ORDER NUMBER: 230714-001

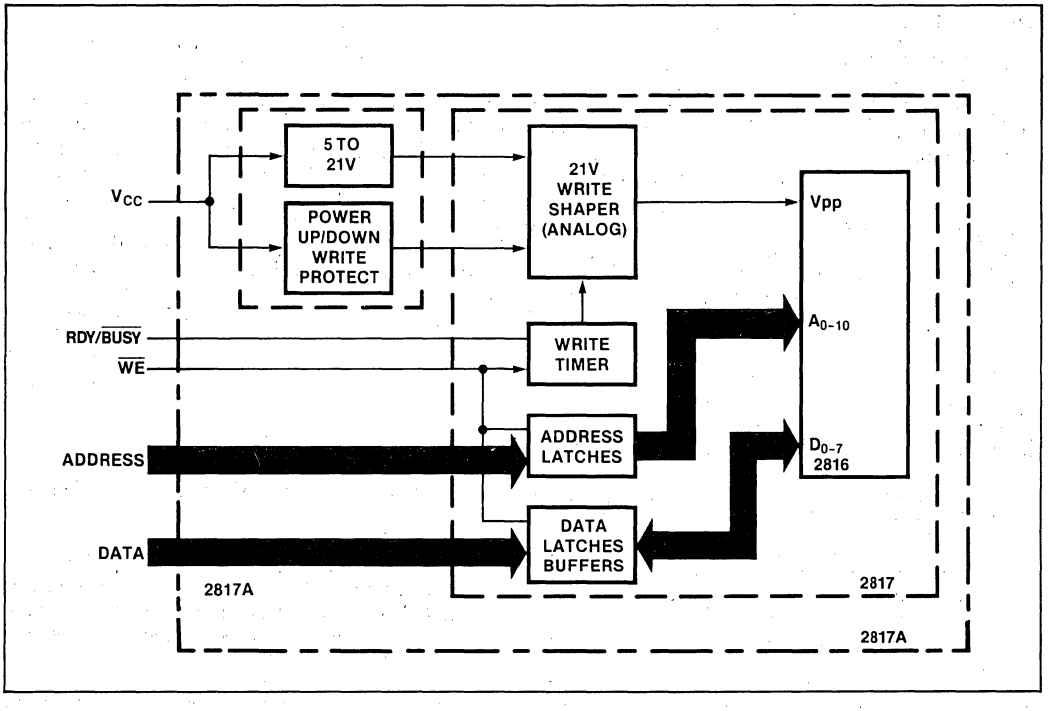


Figure 1. E²PROM Evolution: Increasing Intelligence On-Chip

Advantages of an Intelligent E²PROM

The 2817A has on-chip latches and timing which allow it to connect directly to the microprocessor data bus. Writing a byte to the 2817A is accomplished by sending a write pulse to the part. The timing for this pulse is the same as the timing for a static RAM write cycle. Upon receiving this write pulse the 2817A starts a write operation. During a write operation the 2817A's data lines go to a high impedance state, allowing normal processing to continue on the microprocessor data bus. After the write operation has been completed, the system CPU is notified via the 2817A's RDY/$\overline{\text{BUSY}}$ output. The RDY/$\overline{\text{BUSY}}$ signal frees the designer from having to time the E²PROM write operation by either hardware timing logic or software time-out loops.

Note that while performing a write operation the 2817A will not respond to read or write requests. The 2817A's data bus will remain in a high impedance state, regardless of the input control signals, until the write operation has been completed.

APPLICATIONS FOR E²PROMS

Application Area #1: Firmware Remote Downloading

Remote downloading is important in any system that uses firmware to store program code. Storing software in ultraviolet erasable EPROMs has the advantages of being non-volatile and allowing zero wait-state execution with high-speed processors such as the 8 MHZ 8086-2. A limitation of this method of firmware storage is that system software usually has bugs in the first year or so and is frequently improved or upgraded. To make a firmware change in a system with UV EPROMs a serviceperson must go to each customers site, partially disassemble each system, replace the EPROMs, and re-assemble the system. Now, by storing some or all of the system software code in E²PROMs, software changes can be made by telephone, without sending a serviceperson to the customer site. To make software changes in E²PROMs, the serviceperson calls the customer site, connects the system to the service center computer via telephone, and transfers the new code to the customer's system. In less than a minute the system can be up and running with the new software at a minimal cost to the manufacturer or customer.

Firmware stored in E²PROM can be changed/upgraded remotely over any type of communications link.

All E²PROM or a Combination of EPROM/E²PROM?

System firmware can be completely stored in E²PROM to allow remote alterability of all code, or a combination of E²PROM and UV EPROM can be used.

If the firmware is completely stored in E²PROM, any or all of the firmware can be replaced electrically. This type of system is shown in Figure 2. When the system is prepared for shipping, the operating system software is downloaded directly into the E²PROM memory after the E²PROMs have been installed in the system. When the software is changed, upgraded, or expanded the entire new software package can be loaded remotely over the telephone.

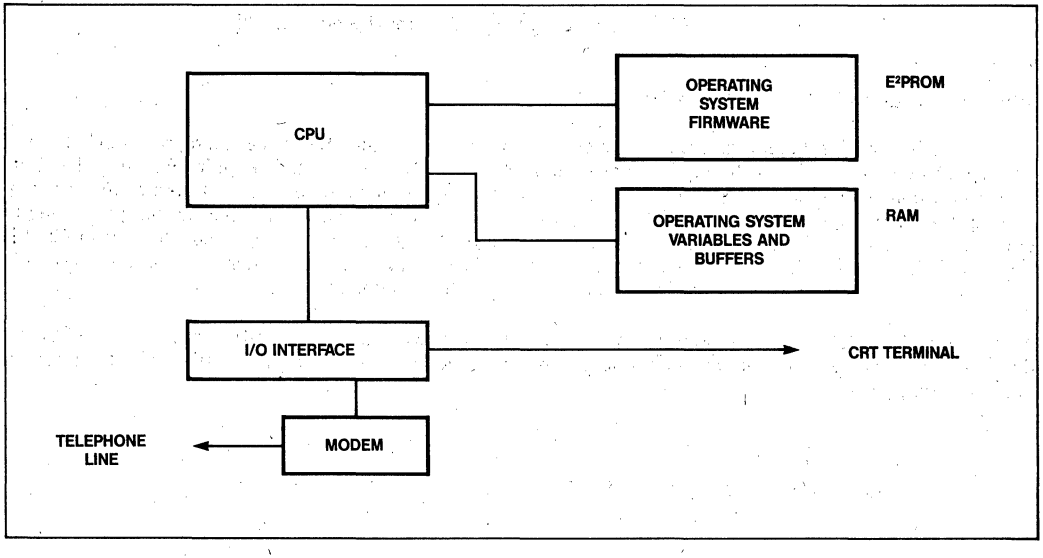


Figure 2. Total Firmware Storage in E²PROM

In a system with a large amount of program code, it may be desirable to use EPROM for storing most of the firmware and use E²PROM for the corrected portions of the code. (see Figure 3) This method, sometimes called "firmware patching," allows a system to use high density EPROM to store a firmware package that is segmented into a number of sections or routines. A small part of the system's E²PROM is used to store a short program that calls each routine as it is needed (see Figure 4). When an error is found in one of the firmware routines, a corrected version is loaded into the extra E²PROM space. The new corrected routine is then called instead of the erroneous routine by changing the calling program (see Figure 5). The corrected routine and new call command can be sent directly from a service center to the system at the customer site over any telephone line.

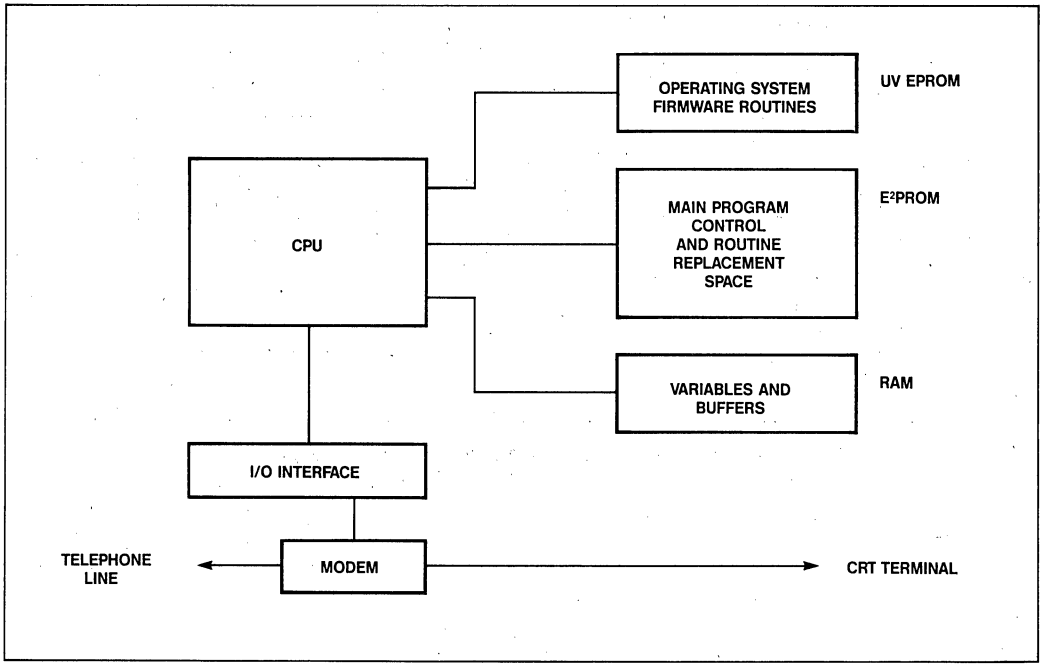


Figure 3. Partial Firmware Storage in E²PROM

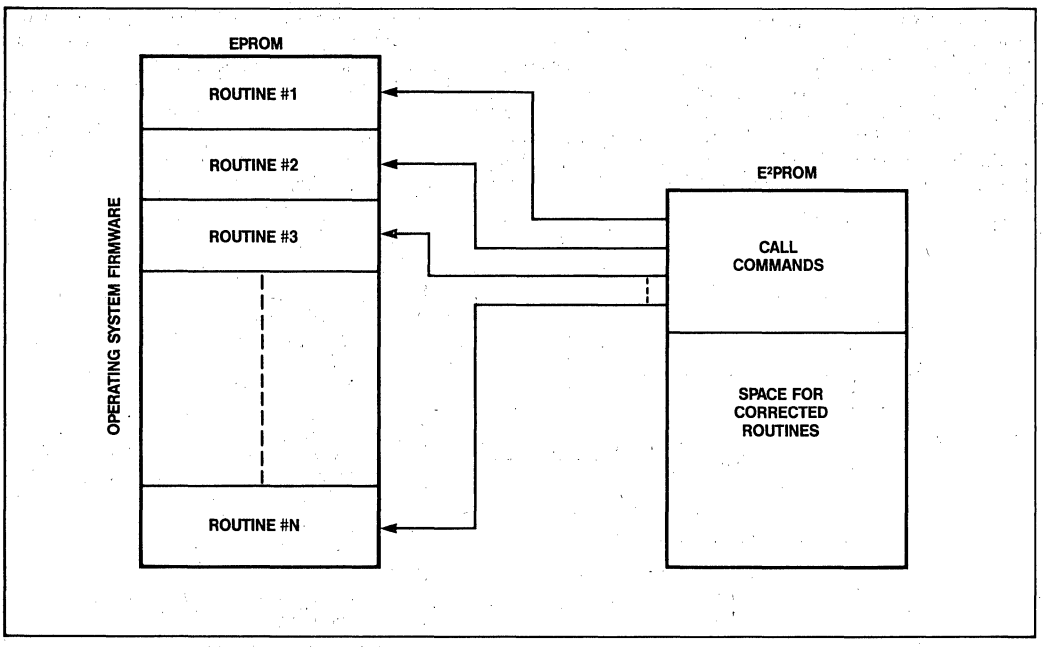


Figure 4. Firmware Patching: Operating System Memory Map

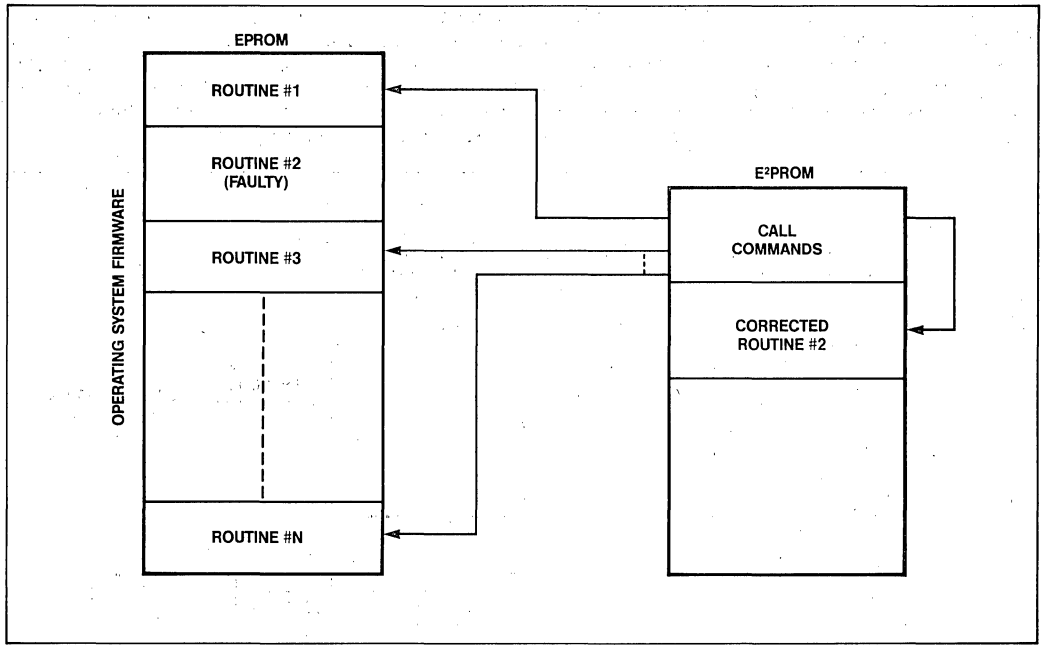


Figure 5. Firmware Patching Example: Replacement of Faulty Routine with Corrected Routine in E²PROM

Easy Software Upgrades

Software corrections, upgrades, or the addition of software options are easily done in a system using E²PROMs for complete or partial firmware storage. The following is a typical example.

The OEM's service center calls the customer on the telephone and tells him that the center wishes to upgrade the customer's system firmware. The center instructs the system operator to set a "download" switch on the system, connect the phone to the system, and depress the system RESET switch. The new software is transferred from the center to the customer system, where it is stored in E²PROM. The new software package then tests itself and the system. The system's CRT display keeps the operator informed of each step in the process. When the system test is done, the CRT display instructs the operator to disconnect the phone, turn off the "download" switch, and reset the system. The system is now up and running with the new software after only 40 seconds of down time. (see section on Fast Array Programming on page 29.)

Hardware Implementation of Remote Downloading

As the new code is being received over the phone from the service center it is loaded into RAM. The code is sent in blocks and a checksum is sent with each block. If a data error occurs due to a poor phone connection, the block is re-transmitted. Once the entire program has been correctly received, it is transferred to E²PROM.

In a system using both EPROM/ROM and E²PROM the downloading software routine is executed out of the EPROM/ROM. In a system with only E²PROM, the downloading routine can be executed either out of the system RAM or out of E²PROM. With the first method the downloading routine is transferred from E²PROM to RAM. (see Figure 6) The routine is then executed out of RAM and the new code is loaded remotely over a communications link into E²PROM. A copy of the downloading routine is included in the new code in the E²PROM for future remote upgrades. At the end of the transfer operation, control is passed back to the new program in E²PROM.

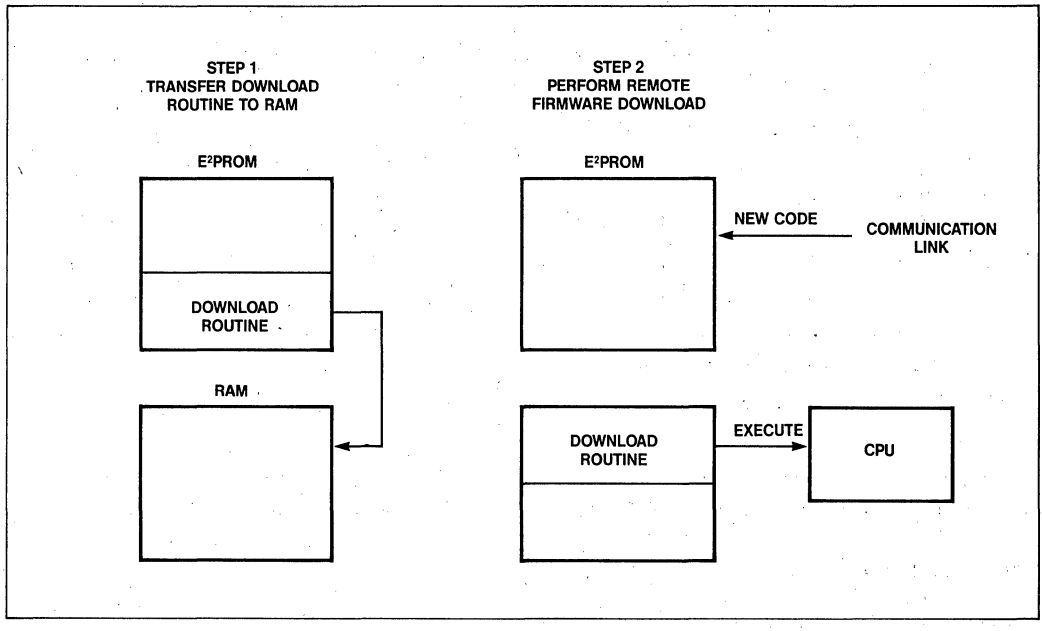


Figure 6. Remote Downloading to an E²PROM-only System with Download Routine in RAM

In systems where there are two or more E²PROMs the downloading routine can be initially executed out of the first E²PROM. (see Figure 7) After the new program code has been loaded into the rest of the E²PROMs in the system, execution control is passed to a copy of the download routine in E²PROM #2, as shown in Figure 7. The remaining new code is then loaded into E²PROM #1.

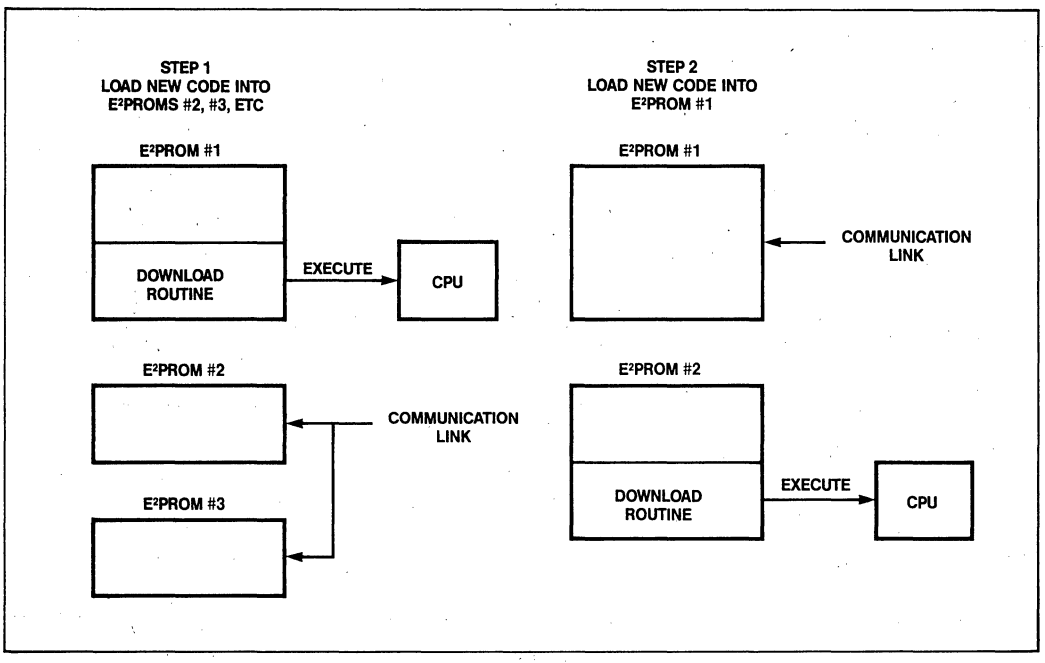


Figure 7. Remote Downloading to an E²PROM-only System with Downloading Routine in E²PROM

Demo Circuit

The 2817A/8088 stand-alone system design shown in this applications note can send data over a serial RS232 cable, as shown in the block diagram in Figure 8. This design can also send over any telephone line using an acoustic coupler. The hookup for transmission over a phone line is shown in Figure 9. The software on the applications demo board is set up to send text messages between boards when they are connected as shown in Figure 8 or 9. The messages are then stored in E²PROM memory where they remain unchanged until it is desired to delete a message or clear the message array. It can be seen that program code rather than text messages can just as easily be transferred between the two systems and stored in E²PROM memory. The downloading routine would be transferred to and executed from the system RAM. The new program code would then be executed directly out of E²PROM.

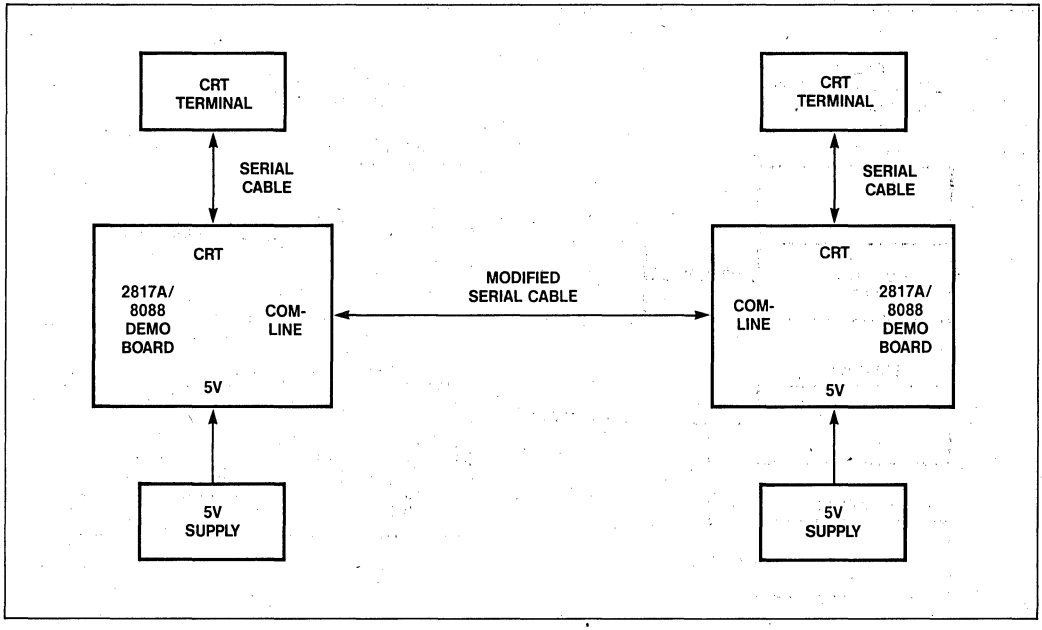


Figure 8. Remote Download Demo Hookup

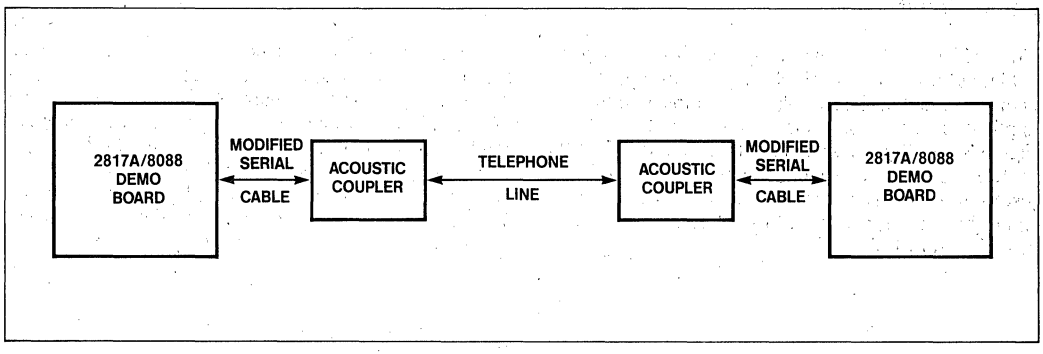


Figure 9. Remote Downloading over a Telephone Line using the 2817A/8088 Demo Board

The "modified serial cable" shown in Figure 8 is an RS232 serial cable with the send and receive lines switched as shown in Figure 10.

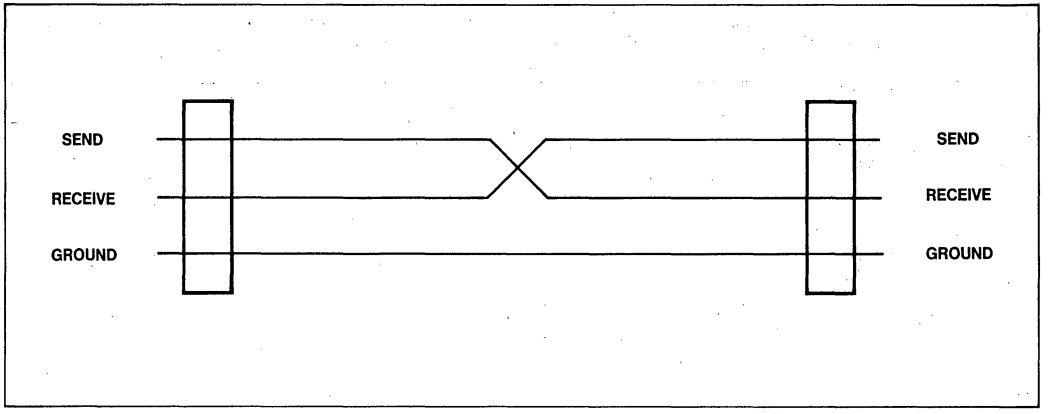


Figure 10. Modified RS232 Serial Cable

The 8088 software for performing a data block transfer between systems is shown on pages 42-44 in Appendix A. Figures 11 and 12 show the flow charts for the data transfer software routines.

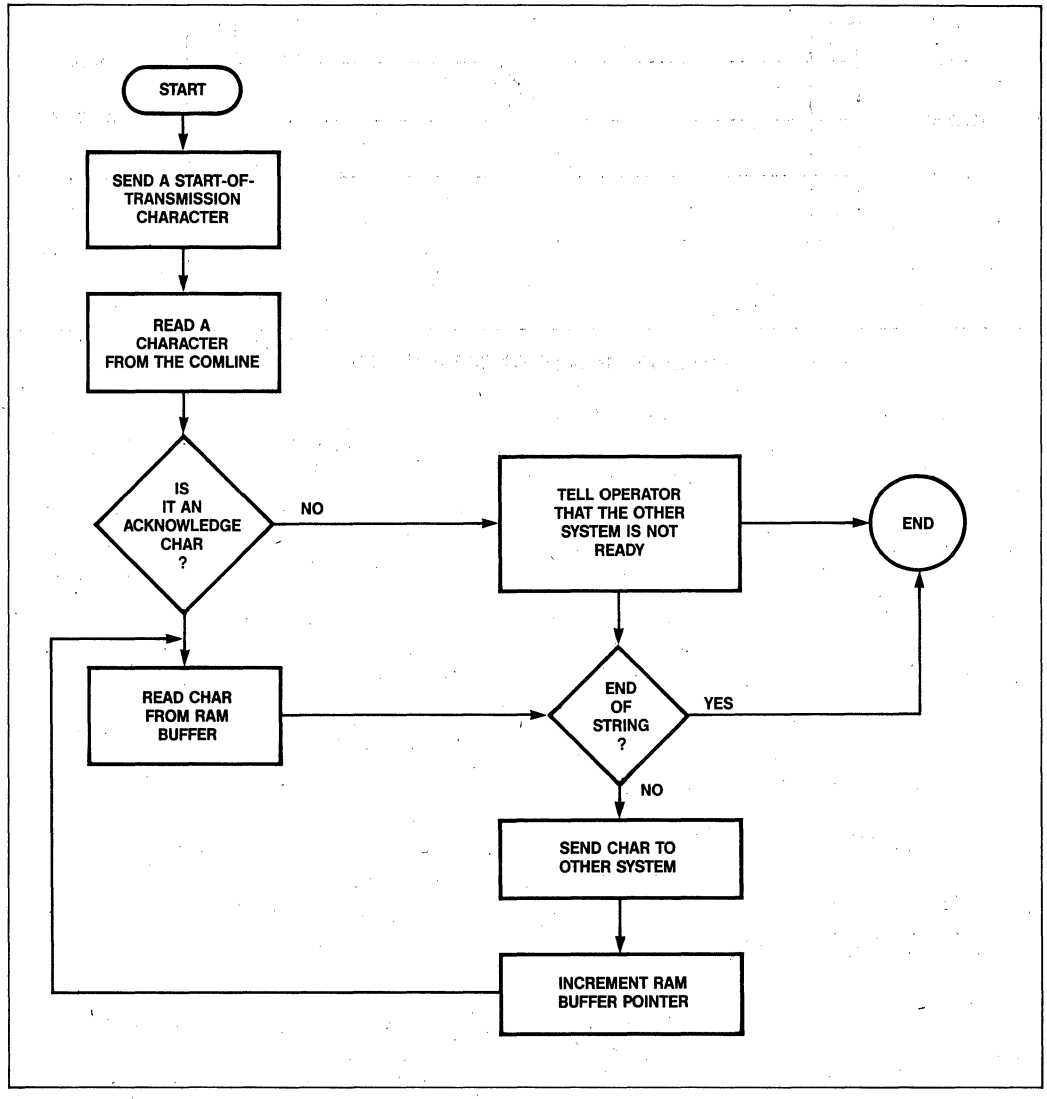


Figure 11. Data Transfer Software: Transmitting System

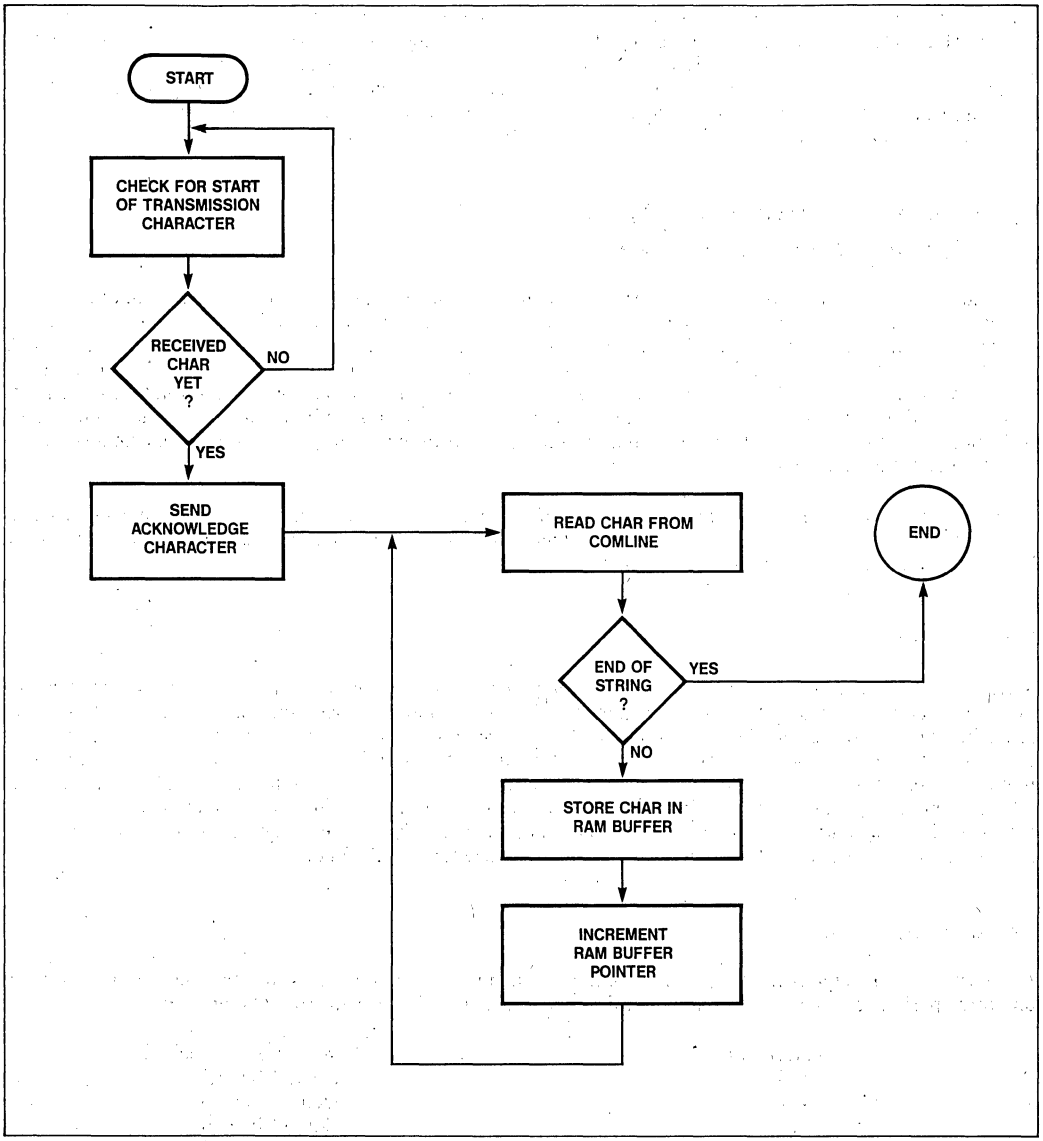


Figure 12. Data Transfer Software: Receiving System

Since serial channel drivers in this design do not provide full spec RS232 levels, the serial cables should be kept under 10 feet in length. The drivers are less than full spec to reduce the complexity of this system for demonstration purposes. These drivers require only a 5V supply to operate rather than the +12V and −12V required for standard RS232 interfaces. The circuit description for the RS232 serial channel drivers under the section, "2817A/8088 Applications Demo Board Circuit Operation"gives more information on the circuitry and also shows a design example for a full spec RS232 interface.

Application Area #2: System Reconfiguration

Small Systems—User Friendly Operation

In small systems such as modems, point of sale terminals, and data terminals, hardware switches have been used to set operation parameters such as baud rates, synchronous/asynchronous selection, and data constants (tax rates, display attributes, and many others). It is usually necessary to have printed tables and operation manuals in order to understand how and when to set these switches. Wouldn't it be much easier if the CRT terminal asked you what baud rate you wanted or drew a pricing table on the screen and asked you to fill it in? With E²PROM this is easily done. Parameters entered by the operator can be stored in specific blocks, called "look-up tables," in E²PROM. Each time the system is powered up, the CPU sets all parameters based on the information stored in the E²PROM look-up table.

E²PROM may also be used to re-define the function of any given key of a terminal keyboard. Intelligent terminals and graphics terminals with these "soft keys" are highly flexible in that each user can define the functions he needs for the most efficient use of the terminal.

Large Systems

Medium and large sized computer systems and computer system networks often have hardware switches and jumpers to control peripheral channel assignments, data rates, and even user accessibility. E²PROMs are now replacing these switches and allowing the parameters or assignments to be changed via software. Special access codes in firmware allow only restricted access to critical function controls. There is no need to have manuals on hand to make simple peripheral channel changes. The firmware or software displays a "menu" on the terminal CRT screen and asks the user to fill in the desired numbers or parameter values. These numbers and values are then stored permanently in E²PROM look-up tables, and the system's parameters and operational modes are set according to the values stored in E²PROM each time the system is powered up.

Demo Board Circuit

For the purpose of demonstrating system re-configuration, imagine a computer room with three computer systems and three peripherals. The systems are identified as #1, #2, and #3. The three peripherals are:

Peripheral	Indicated on demo board as:
Hard Disk	Demo Channel A
Printer	Demo Channel B
Mag Tape Drive	Demo Channel C

A peripheral control unit is used to determine which system is connected to which peripheral, as shown in Figure 13. More than one peripheral can be allocated to a given system. An E²PROM memory in the peripheral control unit acts as a software switch to permanently retain the present peripheral configuration until changed by an operator. As the data storage requirements of each computer system changes, the various peripherals can be re-assigned as needed. For more efficient operation a routine could be written which would automatically re-configure the peripheral channels based on the data storage requirements/requests of the systems. The peripheral assignments on the 2817A/8088 demo board are changed by a software command from the operator. The new assignments are stored in a look-up table in E²PROM as shown in Figure 14. Figure 15 shows the location of the look-up table in E²PROM. In this manner the present system configuration remains in effect through system power-down and power-up.

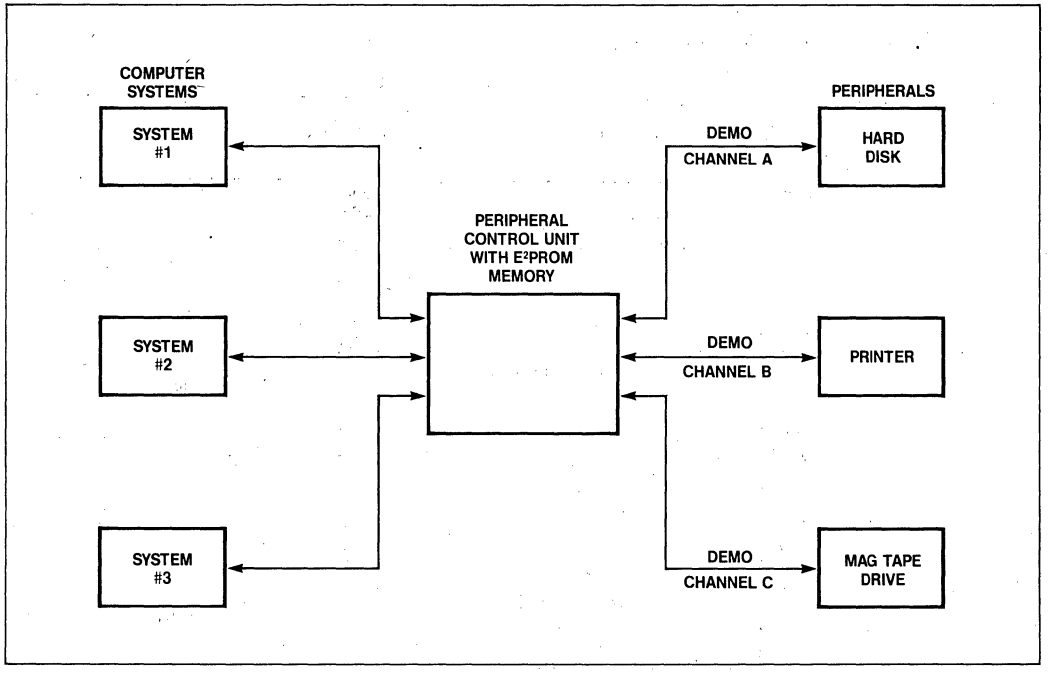


Figure 13. System Re-configuration Demo: Computer Room

		PERIPHERAL
BYTE #1	SYSTEM #	(DEMO CHANNEL A) HARD DISK
BYTE #2	SYSTEM #	(DEMO CHANNEL B) PRINTER
BYTE #3	SYSTEM #	(DEMO CHANNEL C) MAG TAPE DRIVE

The value "System #" Determines Which System has access to the Peripheral.

Figure 14. System Re-configuration Look-up Table

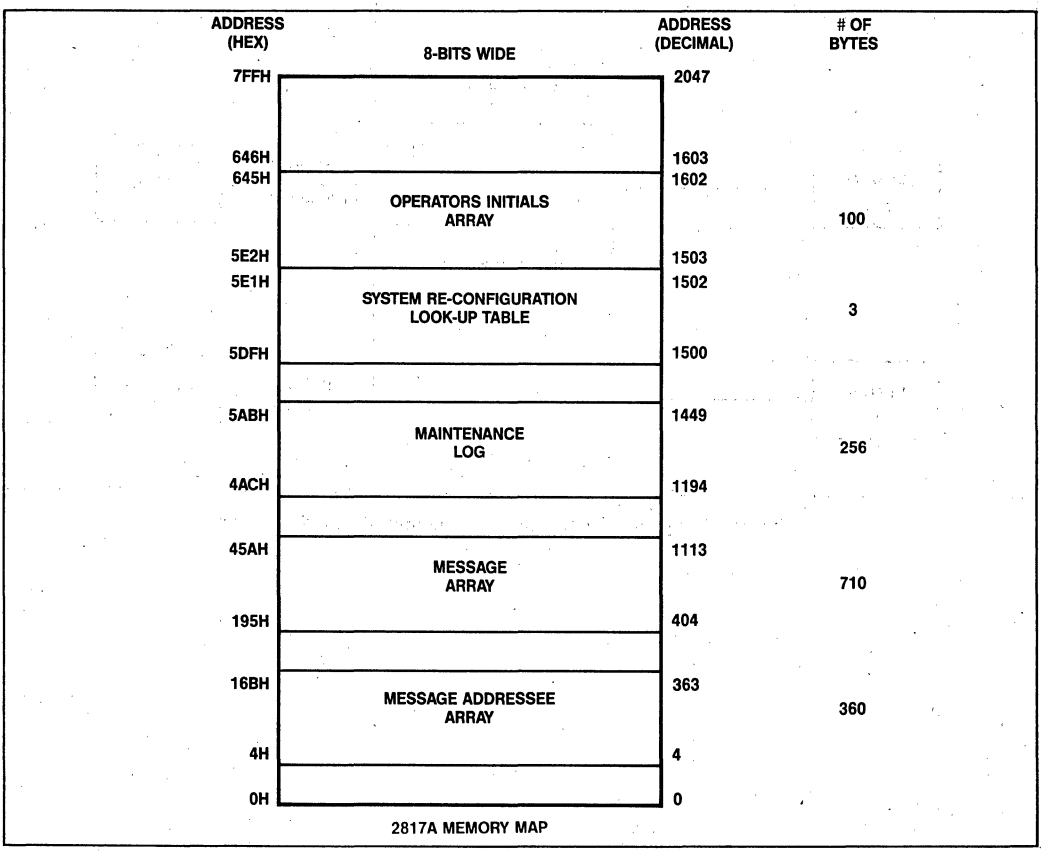


Figure 15. 2817A/8088 Applications Demo

Further detailed information on the hardware operation of the System Reconfiguration Demo is given in the section "2817A/8088 Applications Demo Board Circuit Operation" under "Demo Channel Display LED Ports."

Application Area #3: Maintenance Log

When equipment is repaired or upgraded it is important to record the nature of each repair problem and to document the revision levels. The information, usually filed at the user's site, is often lost. E²PROM can be used to store maintenance record information and revision level numbers in the system itself, thus eliminating possible confusion with other similiar systems. With an E²PROM maintenance log, the information is always readily accessible through the software. A few simple software commands can be used to enter or display maintenance log information. Complete information about the system's optional equipment and even the serial number can be stored in the system's E²PROM log. This information may also be read over a telephone line by the manufacturer's service center computer, eliminating the possibility of operator error in relaying the information.

Demo Board Circuit

The demo board has an E²PROM maintenance log. The memory space allocated for this log in the 2817A is shown in Figure 15. Entries are made from the keyboard, and the contents of the log can be displayed on the CRT at any time. The entries are automatically numbered as they are entered.

Application Area #4: Electronic Message Storage

E²PROM can be used to store messages in a small stand-alone system. The low cost and ease of use of the 2817A makes it very desirable for small microprocessor-based designs. Visualize one of the numerical control systems that supervises the various phases of production in an industrial plant. To keep shift operators informed of any process changes, messages can be displayed on the CRT display. These messages, stored in the system's E²PROM by a previous operator or a supervisor, inform future operators of temporary or permanent operation modifications, or warn them of any problems that should be closely monitored. No floppy disk needs to be left in the terminal to store the message: all that is needed is the in-system E²PROM. This kind of message storage capability is also a useful feature on personal computers, personal development systems, and intelligent modems.

Another application for E²PROM message storage is re-programmable restricted access keys to computer systems. In a system using E²PROM for security control, assess to critical data can be restricted based on specific text strings, or "keys", that are entered by the operator. In this type of system the "keys" are stored in E²PROM and are changed as needed by an authorized company officer. For example, a data processing system using this technique allows access to sensitive product cost tables only upon receipt of a specific text string that is determined by an authorized company officer. The text string can be as simple as "code table" or can be related to a product line name such as "Intel Personal Development System". The text key can also be a completely unrelated phrase such as "Donald Duck" or anything else that is easy for an authorized operator to remember, yet prevents general access. Such a security system is highly reliable because it does not depend upon electro-mechanical storage device—it uses reliable, easy-to-design-in E²PROMs.

Demo Board Circuit

The demo board can accept a message destined for a given addressee. The message and the name are kept in separate arrrays as shown in Figure 15. Whenever the board is powered up, the names in the addressee array are listed on the CRT, thus giving notice of the names of the people who have messages. The messages for any specific addressee can be viewed on command. Messages for any specific addressee can later be deleted, or the entire message array can be cleared. The list of addressees can also be viewed at any time on command.

MICROPROCESSOR INTERFACING

General Concepts

The 2817A is designed to be easily interfaced to a microprocessor. The control signals, $\overline{CE}$, $\overline{OE}$, and $\overline{WE}$ allow the simple basic design shown in Figure 16 to be used. By using the two-line control concept incorporated in all Intel 28-pin memories, the possibility of bus contention with other memory devices on the same data bus is eliminated. This is accomplished by connecting the $\overline{CE}$ input to the output of the system memory address decoder and by connecting the CPU's read and write command signals to the $\overline{OE}$ and $\overline{WE}$ inputs, respectively, as shown in Figure 16. In single line control the data bus is driven as a function of the addresses. This causes overlap on the $\overline{CE}$ select lines which can result in data bus contention as shown in Figure 17. Through two-line control, the data bus is driven only when both the address select ($\overline{CE}$) and the memory command ($\overline{RD}$) are active (see Figure 18).

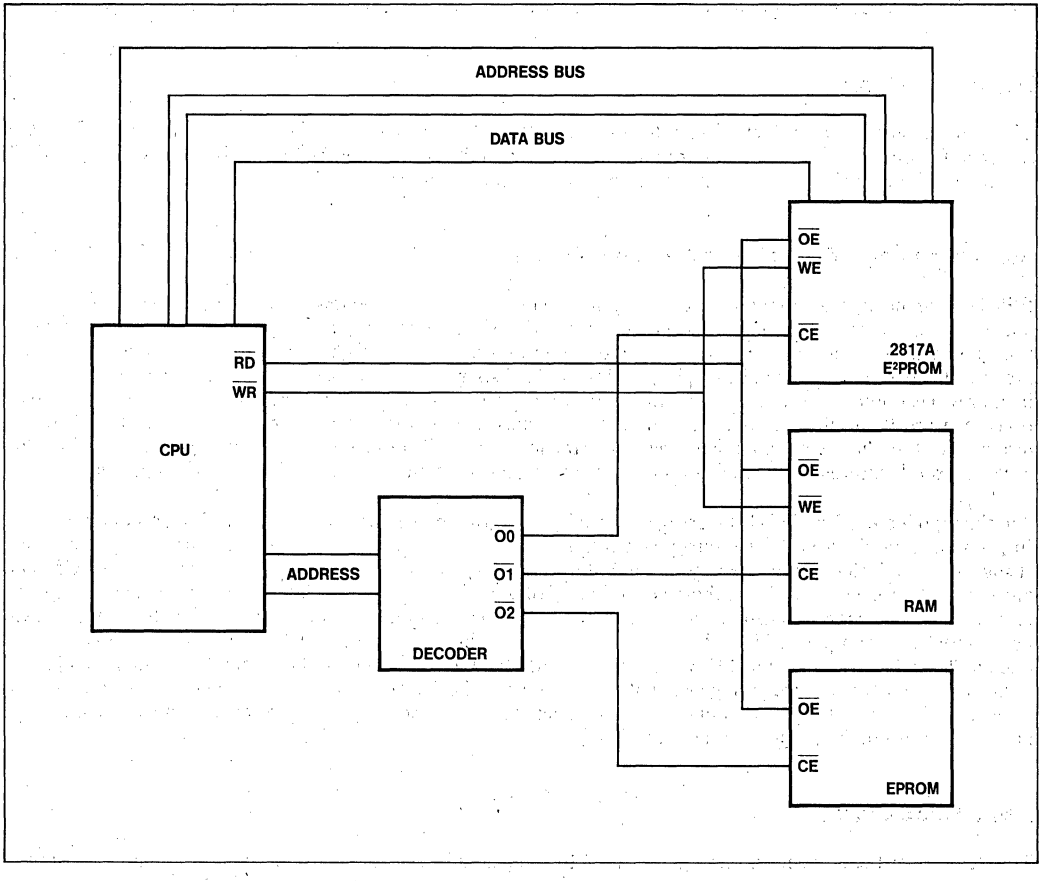


Figure 16. Basic Microprocessor Interface

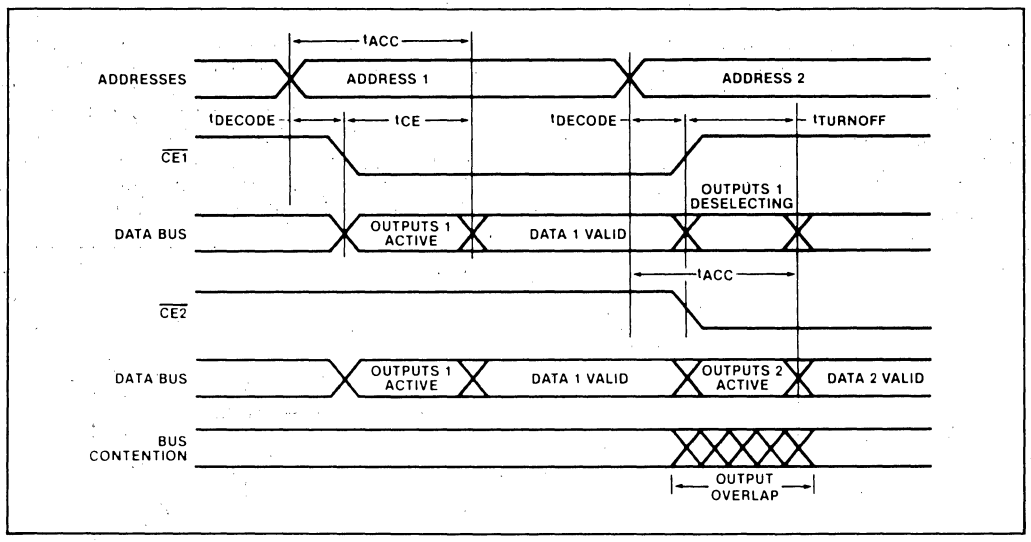


Figure 17. Single-Line Control and Bus Contention

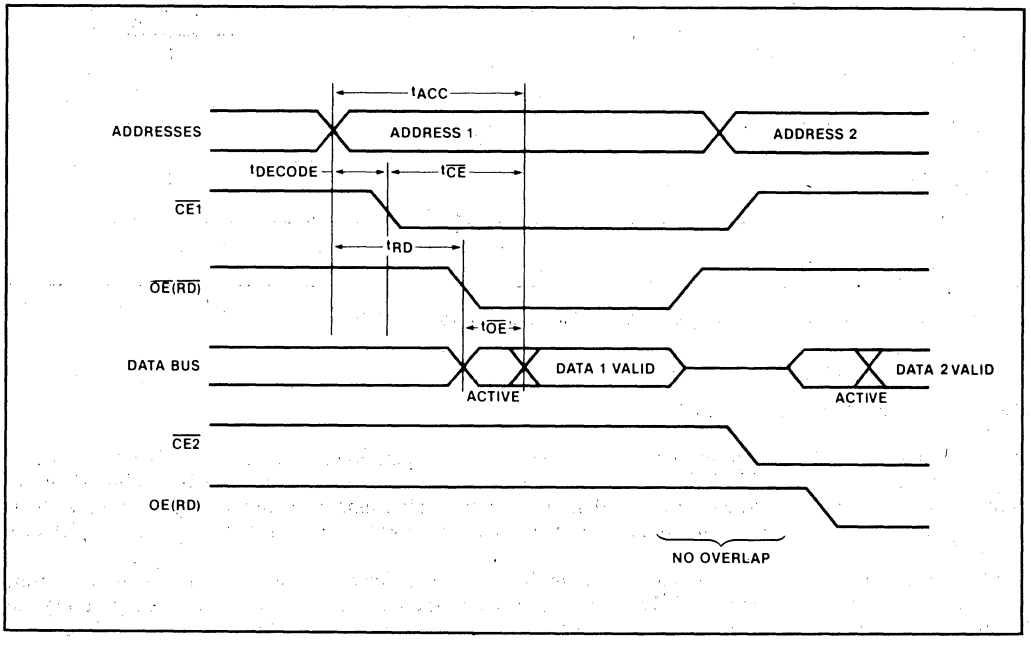


Figure 18. Two-Line Control Architecture

Interfacing to the 8088

Interfacing the 2817A to the 8088 8-bit processor is a simple task. The 8088 is a very powerful processor having a one megabyte direct addressing capability and the 8086 instruction set. Figure 19 shows the interface used on the 2817A/8088 applications demo board. The 8288 bus controller is used to generate various memory, I/O, and control signals. One of the signals is a delayed write pulse command, $\overline{MWTC}$, which is used to write to the 2186 Integrated RAM. The 8288 also acts as a buffer for all the memory, I/O, and control signals. As shown in Figure 19, two-line control is used in connecting the 8088 commands and address selection lines to the 2817A. The status of the $\overline{RDY}/\overline{BUSY}$ line is polled via the 8088's $\overline{TEST}$ input by using the WAIT instruction. See "How to use the $\overline{RDY}/\overline{BUSY}$ output" (below) for detailed operational information about the $\overline{RDY}/\overline{BUSY}$ line.

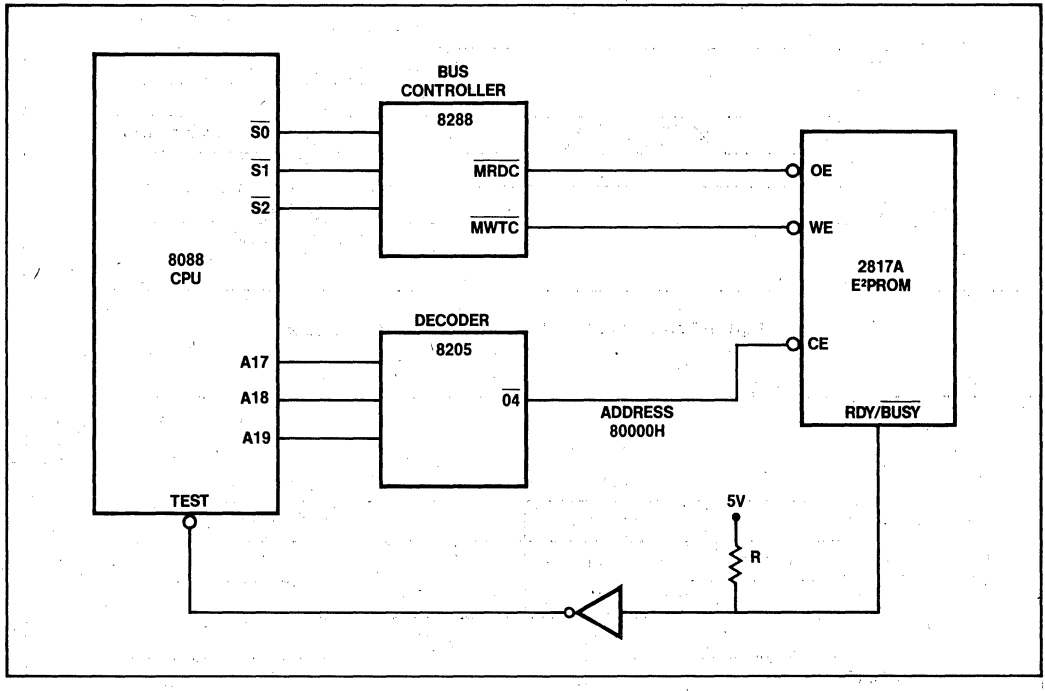


Figure 19. 2817A Interface to the 8088

DESIGN CONSIDERATIONS

How to use the $\overline{RDY}/\overline{BUSY}$ Output

The 2817A has a $\overline{RDY}/\overline{BUSY}$ output that indicates when a write operation is in progress and when the 2817A is ready for access. the $\overline{RDY}/\overline{BUSY}$ line goes low when a write operation starts and goes back high when the operation is completed. The system CPU can poll this output to determine when another byte can be written, or can use the output as an interrupt to notify the CPU when the current write operation has been completed.

Polled mode is usually the easiest interface to design. It is a good choice as long as the CPU can afford to wait while a byte is being written into the 2817A. In systems where the CPU must continue processing during the 20 milliseconds it takes to complete a byte write operation, an interrupt mode should be used.

Polled Methods

A polled mode can be used whenever it is not necessary for the CPU to be constantly processing while an E²PROM byte write operation is taking place. An example is the remote upgrade of a system's software over the telephone. The service center calls a customer and informs him that a new software package is available for the customer's system and is ready for transmission over the telephone. The customer connects the system to the phone line, either via an acoustic coupler or a direct connection, and waits about 40 seconds (see section on Fast Array Programming) while the new software package is being sent. During this time the only task the CPU is likely to have is the supervision of the downloading operation, so it can afford to wait while data is being written into E²PROM. The user's system now has the latest firmware package, received quickly and inexpensively over the phone, and good for another year or so until the next upgrade.

Polled mode is usually done by connecting the RDY/$\overline{\text{BUSY}}$ output to the CPU data bus via a three-state buffer as shown in Figure 20. (If desired, the rising edge of the RDY/$\overline{\text{BUSY}}$ signal could be used to set a positive edge-triggered flip-flop.) By polling the E2 RDY port the CPU can determine when the 2817A is ready for a read cycle or another write cycle. If any of the Intel I/O port devices (8155, 8255, 8355, or 8755A) are being used in the system, the RDY/$\overline{\text{BUSY}}$ output can be connected to one of the input ports of one of these devices.

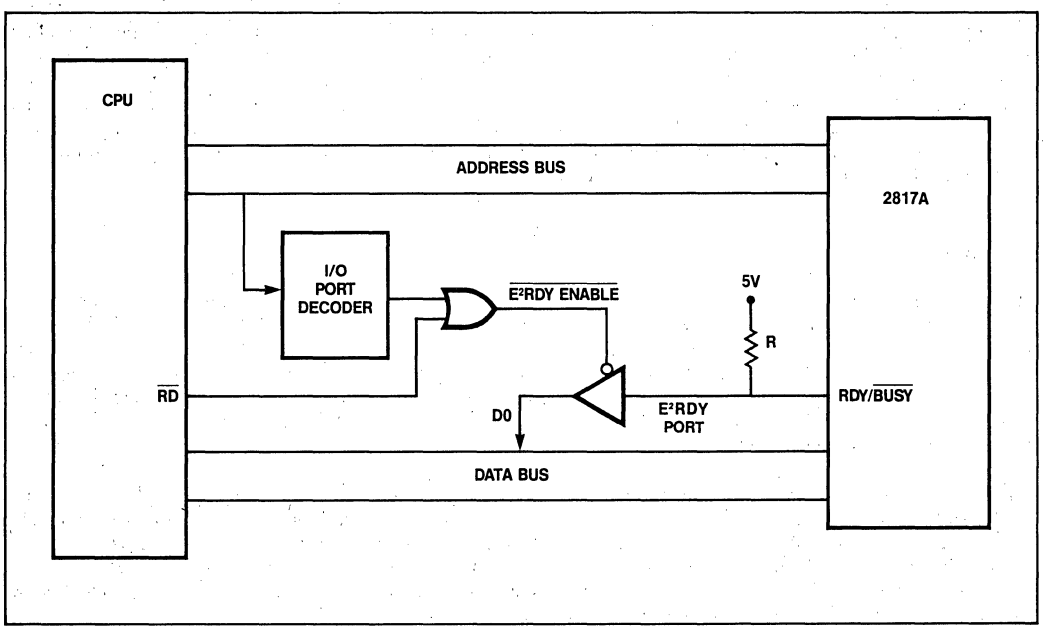


Figure 20. Polled Mode: Using an I/O Input Port to Poll the RDY/$\overline{\text{BUSY}}$ Output

Polled mode for the 8086/8088 microprocessor family is particularly easy using the $\overline{\text{TEST}}$ input and the WAIT instruction. This method is used on the 2817A/8088 demo board. The hardware hookup is shown in Figure 19.

In the 2817A/8088 demo board design a polled method was chosen for its hardware and software simplicity. The hardware consists of an inverter between the RDY/$\overline{\text{BUSY}}$ output and the $\overline{\text{TEST}}$ input of the 8088. In the software, writing to the 2817A simply requires a MOV instruction, and a NOP and a WAIT instruction, as follows:

```
;
;
        MOV     E2PROM, AL          ;THE AL REG HAS THE DATA BYTE TO
                                    :BE STORED
;
        NOP                         ;THIS ALLOWS ENOUGH TIME FOR THE
                                    :RDY/BUSY LINE TO BE RECOGNIZED
                                    :BY THE "WAIT" INSTRUCTION
;
        WAIT                        ;WAIT UNTIL THE WRITE OPERATION IS
                                    ;COMPLETED
;
;
```

The MOV instruction writes the data byte contained in the AL register to the 2817A. The 2817A latches the byte's address on the falling edge of the $\overline{\text{WE}}$ signal and latches the data on the rising edge of $\overline{\text{WE}}$. Inside the 2817A the write operation begins, while externally the RDY/$\overline{\text{BUSY}}$ output goes low. The inverted value to the 8088 $\overline{\text{TEST}}$ input is a "high". When the 8088 starts to execute the $\overline{\text{WAIT}}$ instruction the CPU will poll the $\overline{\text{TEST}}$ input. The short delay caused by the NOP instruction insures that the $\overline{\text{TEST}}$ input will be high when the wait instruction is executed. Otherwise, the $\overline{\text{TEST}}$ input would be sampled immediately after the write command ($\overline{\text{MWTC}}$) goes inactive high. As long as the $\overline{\text{TEST}}$ input is high the 8088 will remain in wait state. When the write operation in the 2817A has been completed, the RDY/$\overline{\text{BUSY}}$ output will go low, the $\overline{\text{TEST}}$ input to the 8088 will go high, and program execution will continue with the next instruction after WAIT.

Interrupt Methods

Interrupt mode is desirable when data or parameters need to be stored at irregular and relatively frequent intervals, and the CPU must be constantly processing. ("Relatively" meaning not so frequent as to exceed the write endurance of the E²PROM). With interrupt mode, as implemented in this application note, not only is the CPU's processing time not significantly affected by the E²PROM write cycles, but the user's main program does not have to worry about checking on the E²PROM to see if a given write cycle is done, nor do any status flags have to be monitored. The user program need only write the data to be stored in E²PROM to a RAM buffer table. The status and interrupt subroutines do the rest.

For the 8088, an 8259A Programmable Interrupt Controller is used to handle interrupt signals. The 8259A can itself handle up to 8 interrupt lines: each line is separately maskable and priority handling is dynamically programmable. The 2817A's RDY/$\overline{\text{BUSY}}$ output can be directly connected to any one of the 8259A's interrupt inputs. The edge-triggered mode is used to trigger the interrrupt input to the 8259A when the RDY/$\overline{\text{BUSY}}$ output returns high upon completion of the write operation. The schematic diagrams in Figure 21 and Appendix B show the electrical connections for using an 8259A with 8088 CPU.

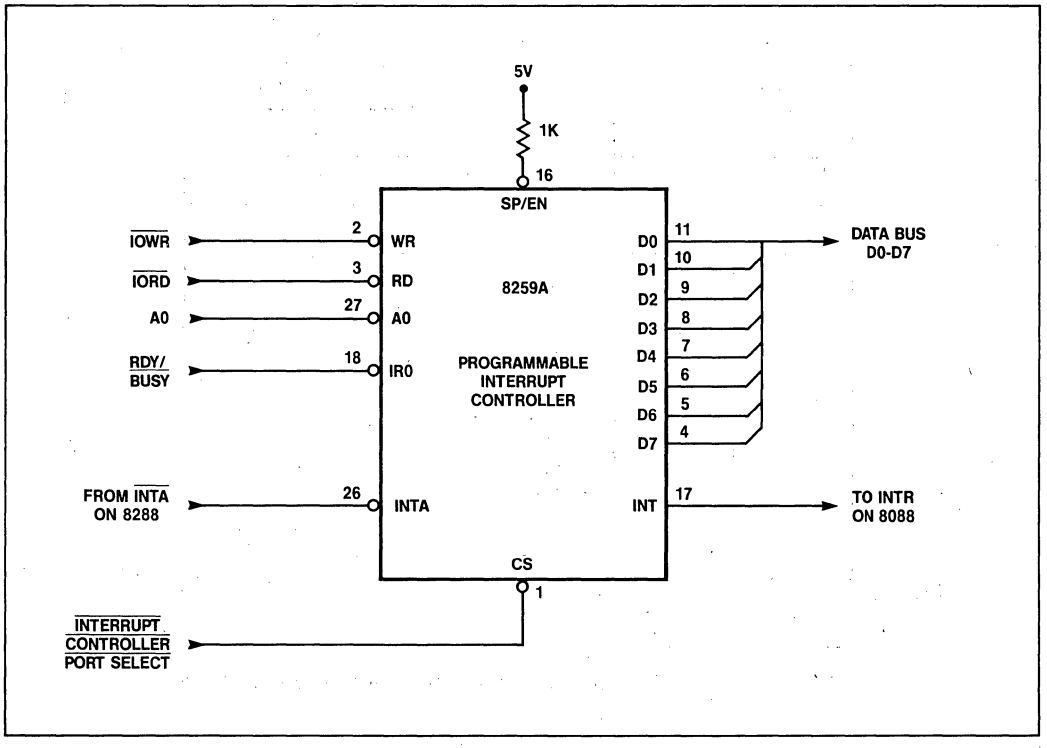


Figure 21. Interrupt Mode: 8259A in an 8088 System

Interrupt Subroutine

The following is an example of how to implement an interrupt-driven system in software.

A section of the system RAM is organized exactly like the look-up table in the 2817A E²PROM. In this example the table consists of ten bytes. (see Figure 22)

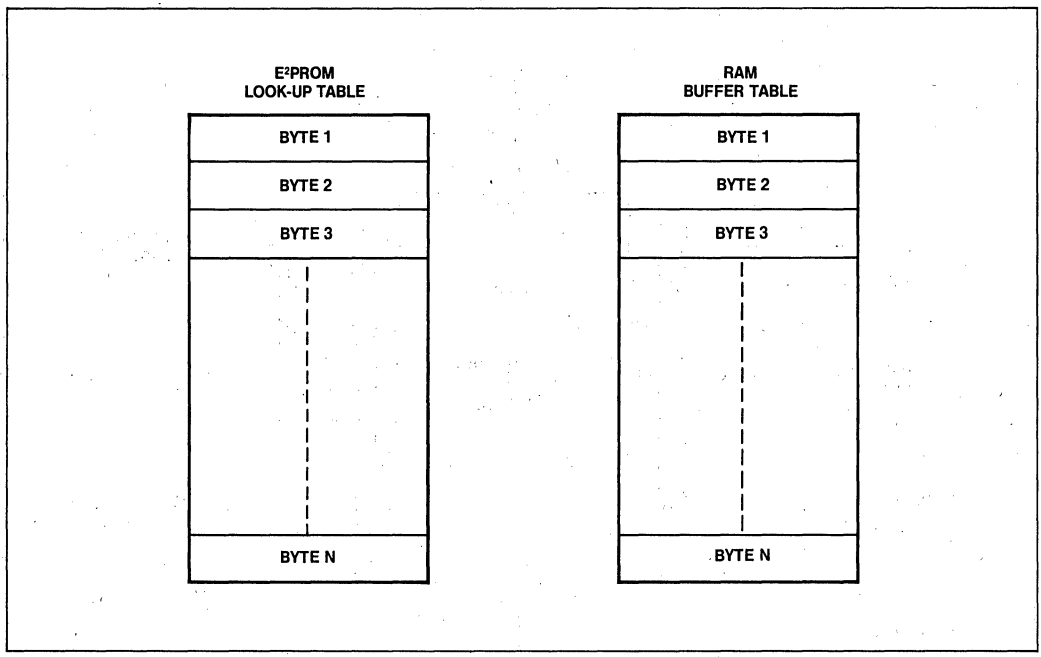


Figure 22. E²PROM Look-up Table and RAM Buffer for Interrupt Mode

Each time the user program receives or produces data to be stored in E²PROM, all the user program has to do is write the desired data into the corresponding locations in the RAM buffer table, then call a short status subroutine. The transfer of data to the E²PROM and the monitoring of the write operation are done by the status and interrupt subroutines.

The status subroutine checks if an E²PROM write operation is in progress, which would mean that the interrupt subroutine is already in the process of trasnsferring data from RAM to E²PROM. If so, then nothing else needs to be done. If no transfer is in progress, then the status routine will start an E²PROM write operation by looking for the first byte of data in the RAM buffer table that is not equal to its corresponding byte of data in the E²PROM table. When that write operation is complete, the interrupt subroutine is called, which checks for any more data to be written to E²PROM. The interrupt subroutine is called after each write operation is completed until all the new data in the RAM buffer table has been transferred to E²PROM.

The flowcharts for these subroutines are shown in Figures 23 and 24, and the software implementation for the 2817A/8088 demo board is on pages 44 through 47. The code to initialize the 8259A Programmable Interrupt Controller is also given in Appendix A.

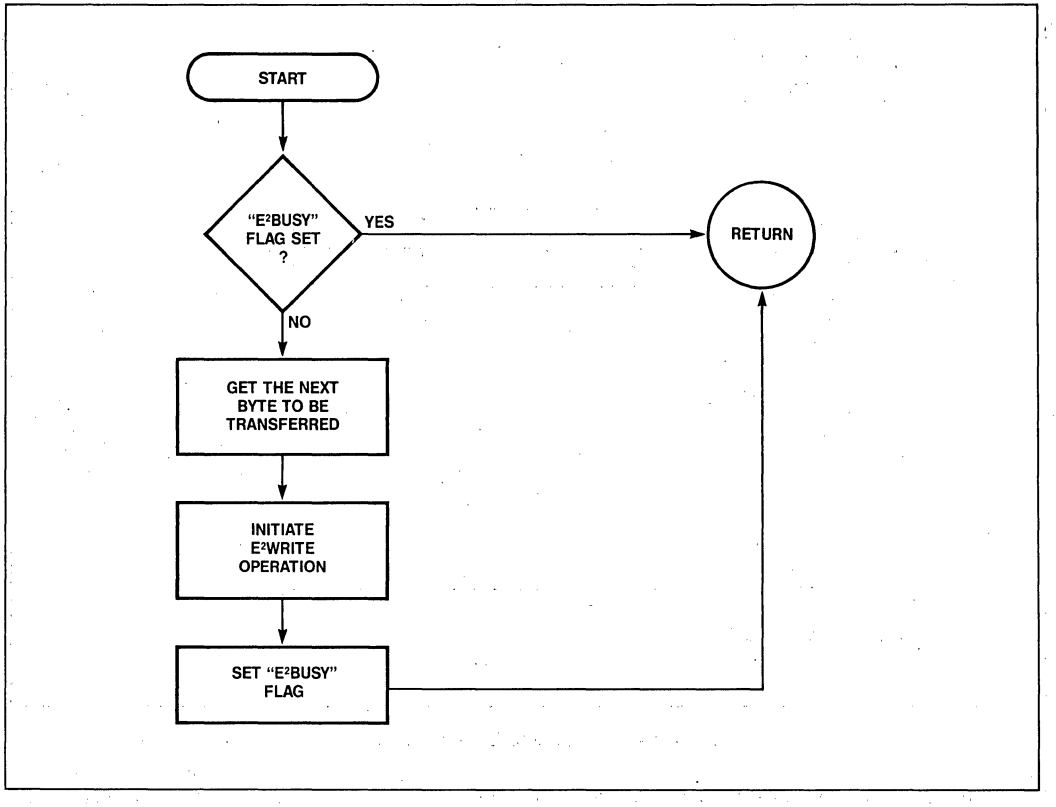


Figure 23. Status Subroutine

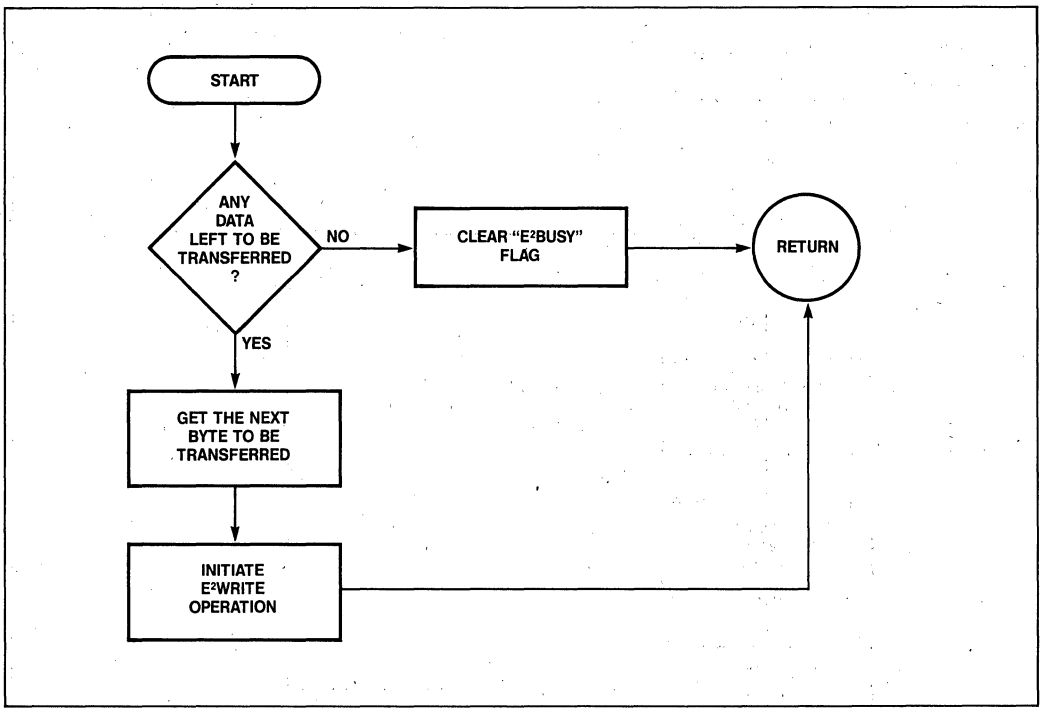


Figure 24. Interrupt Subroutine

When first initializing this look-up table, each byte in the E²PROM table is set equal to its corresponding byte in the RAM buffer table. Any subsequent changes to the RAM buffer table are easily detected by the status and interrupt subroutines by comparing each pair of corresponding locations in the two tables.

Combining Multiple RDY/$\overline{\text{BUSY}}$ Outputs in a 2817A Array

The 2817A's RDY/$\overline{\text{BUSY}}$ pin is an open-drain output, so multiple RDY/$\overline{\text{BUSY}}$ outputs can be or-tied together. The value of the pull-up resistor for the RDY/$\overline{\text{BUSY}}$ output can be calculated by the following formula:

$$R(\text{pullup}) = \frac{4.6V}{2.1ma - I_{IL}}$$

where I_{IL} (or I_{LI}) is the total V_{IL} input current of all device inputs connected to RDY/$\overline{\text{BUSY}}$.

There is no limit to the number of RDY/$\overline{\text{BUSY}}$ outputs that can be or-tied together.

Write Protection during Power Transitions

The 2817A has an on-chip write protection circuit that prevents a write operation from occurring when Vcc is below 4V.

All E²PROMs need inadvertent write protection, either internally or externally, when system power is being turned on or off. Since an E²PROM can be programmed in-circuit the possibility exists that it can be programmed accidentally when system power is removed. This is because the same control signals, $\overline{CE}$ and $\overline{WE}$, that can normally initiate a write operation could also trigger the write mode as system power is failing. As the 5V supply to the system falls, the system components will become unpredictable, and $\overline{CE}$ and $\overline{WE}$ may be spuriously driven active low as shown in Figure 25. To prevent a write operation from accidentally occurring when system power is dropping, an on-chip write protection circuit is needed to hold inactive at least one of the control signals. The same protection is needed when Vcc is rising from 0V to 5V as shown in Figure 25.

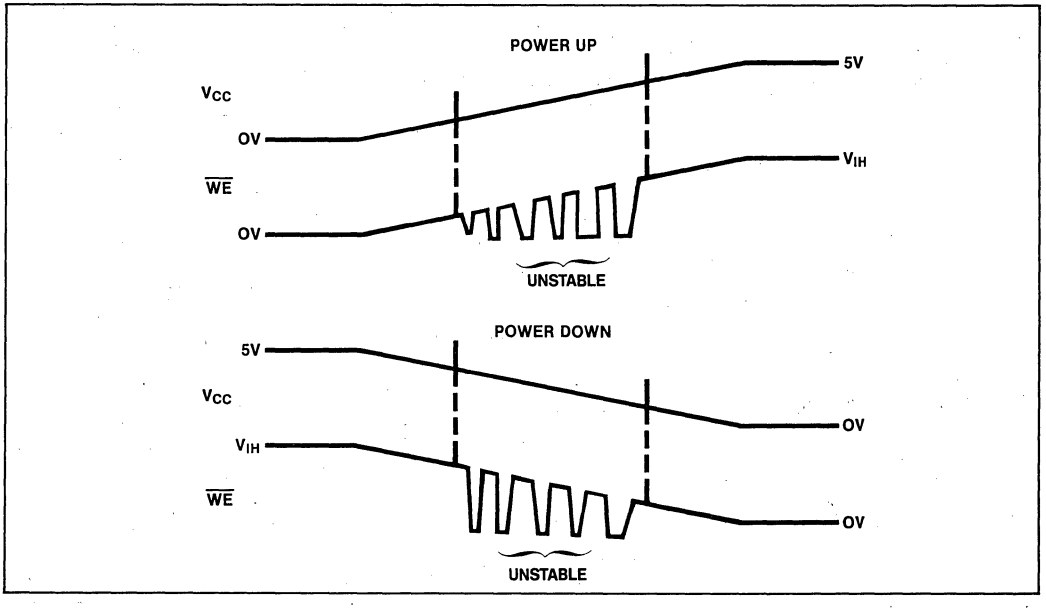


Figure 25. Typical TTL Driver Instability during V_{CC} Power Transitions

Alternative RDY/$\overline{BUSY}$ Interfacing Methods

There are many ways to interface the 2817A to a microprocessor using the RDY/$\overline{BUSY}$ output. The polled and interrupt-driven modes discussed in this application note are the easiest and most commonly used methods. These methods are compatible with the concept of an "intelligent" E²PROM which allows write operations to be done without holding up the microprocessor.

When first looking at the 2817A it might seem as if the RDY/$\overline{\text{BUSY}}$ line should be connected directly to a microprocessor's "ready" input since the signals have similiar names. There are certain timing and operational considerations, however, that make this method more difficult to design into a system than the interrupt or polled modes described earlier. These considerations are discussed below.

It is possible to use the RDY/$\overline{\text{BUSY}}$ output to hold up the microprocessor until the write operation is completed. This involves the insertion of wait states into the microprocessor's instruction cycle. For the 8085A and the 8088/8086, wait states occur when the READY input is brought inactive low. This type of interface would be implemented by connecting the 2817A's RDY/$\overline{\text{BUSY}}$ output to the microprocessor's READY input. At first this method might seem even simpler than polled mode, but a closer examination shows that this is not true.

The first consideration is the discrepancy between the timing of the 2817A's RDY/$\overline{\text{BUSY}}$ signal and the "ready" input of a microprocessor. Taking the 8088 as an example, the READY input is required to be low (if wait states are desired) about the same time that the 8088's write signal goes low. The top three signals in Figure 26 show this requirement. (The READY input shown is that of the 8284A clock generator. The output of the 8284A is then normally connected to the READY pin on the 8088. The $\overline{\text{AMWC}}$ write signal is the output of the 8288 bus controller which is used with the 8088 in MAX mode.) As shown by the bottom signal in Figure 26, the 2817A's RDY/$\overline{\text{BUSY}}$ output does not go low until after the $\overline{\text{WE}}$ input signal goes back high.

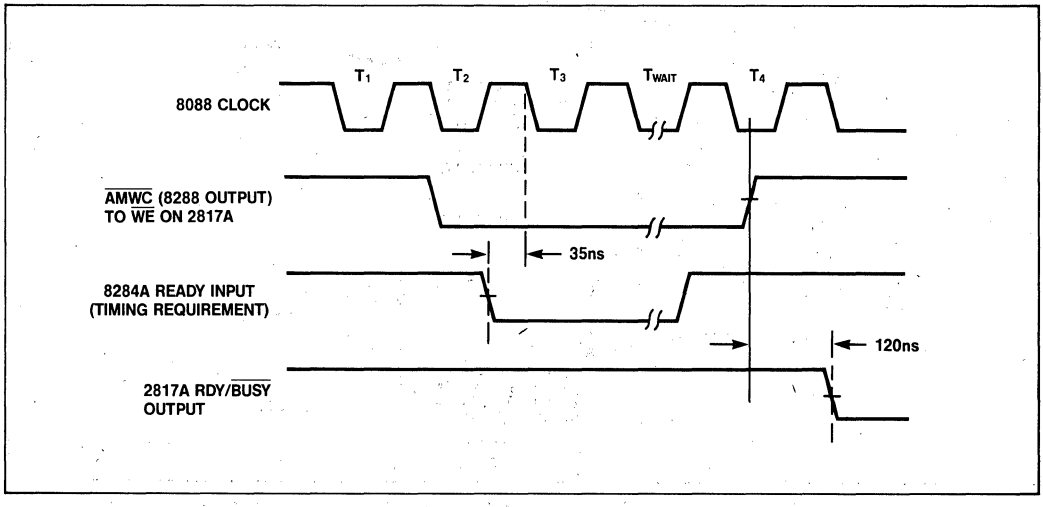


Figure 26. 8088 Ready Input Timing Requirement

Although this timing discrepancy is not critical, the implications for any given system (and the microprocessor used) must be examined carefully before this method of interfacing is used.

Another consideration that must be taken into account is the possibility of two or more sequential write cycles being done by the microprocessor. An 8088 string operation, for example, would allow the CPU to do one bus write cycle after another without any other intervening cycles. The 2817A would be able to respond successfully to the first write cycle, but not the second. The following would happen:

Upon seeing the rising and falling edges of the write pulse during the first write cycle, the 2817A would start an E²PROM write operation. The RDY/$\overline{\text{BUSY}}$ output would go low 75 ns after the rising edge of the write pulse as shown in Figure 27. The RDY/$\overline{\text{BUSY}}$ signal would not hold up the CPU until the following write cycle. While the first byte write operation is in progress the 2817A is not affected by any of the input signals and the data bus is in a high impedance state. When this write operation is completed and the RDY/$\overline{\text{BUSY}}$ signal returns high, the CPU will complete its second write cycle. The 2817A will not have recognized the second write cycle, however, because it did not see the $\overline{\text{WE}}$ signal fall.

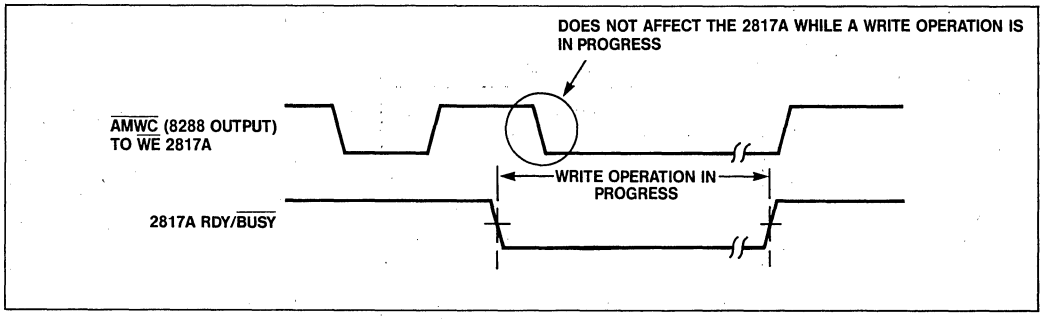


Figure 27. Limitation if 2817A RDY/$\overline{\text{BUSY}}$ is Connected to 8088 Ready Input: Loss of 2nd Byte During Two Consecutive Write Cycles

The direct connection of the 2817A's RDY/$\overline{\text{BUSY}}$ to the 8088's (or any other microprocessor's) READY input would thus require that no software be written that might cause two or more consecutive write cycles to occur.

The design of the 2817A's on-chip intelligence gives the 2817A maximum flexibility in interfacing to microprocessor-based systems. While the interrupt-driven and polled interface modes are two relatively simple (and widely used) approaches, many other methods are possible. In each case the timing specifications should be studied to insure that the timing requirements of all devices involved are met.

E²PROM Mapping Techniques

Inherent in the design of all E²PROM's is a 10,000 write cycle per byte limitation. As an example of the allowable write cycle frequency, all 2048 bytes in one 2817A could be written three times each day, 365 days a year, for 10 years before exceeding guaranteed limits. The E²PROM is not designed to be written to as often as RAM memory. Mapping techniques such as the ones shown here, however, allow higher write cycle frequencies than normal to be used.

Electronic Message Storage Applications

A memory usage mapping technique such as the one used in the Electronic Message Storage Demo described in this applications note can more evenly distribute the number of write cycles to each byte in the E²PROM array, thus stretching the usable life of the E²PROM. In message storage, if a pointer is used to indicate the location of the next empty message block, that pointer has to be re-written into an E²PROM memory location each time a message is stored in the E²PROM memory. That particular pointer storage location could reliably be used 10,000 times and still be guaranteed to remain within specification. If a system using this method had one 2817A for message storage, it could store 3 messages per day, 365 days a year for 10 years.

In the Electronic Message Storage Demo, no pointer is stored each time a message is stored. Instead, the 2817A is partitioned into a number of fixed-length message block spaces. Empty blocks have the byte "FF" in the 1st location, as shown in Figure 28. Thus, whenever a message is to be stored, the 2817A is searched for a block with an "FF" in the 1st location. Using this method, if the 2817A in a message storage system is divided into 20 100-byte blocks (2817A = 2K bytes), up to 60 messages can be stored each day, 365 days a year, for 10 years. In this example, for every additional 2817A in the system, 60 more messages of this size could be stored each day.

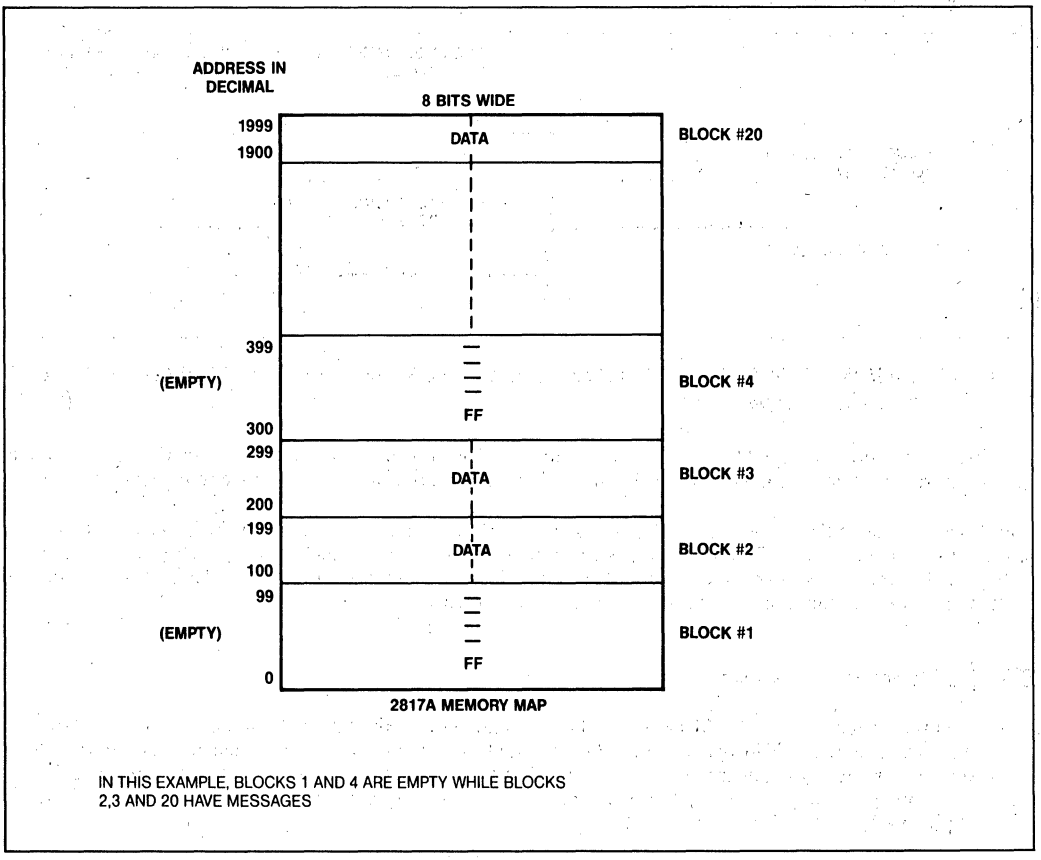


Figure 28. Message Storage Technique Example

Fast Array Programming

In large memory storage applications where fast write times are important, fast array programming can provide a solution. The total write time of any block of data to the array is reduced by a factor equal to the number of 2817As in the array. For example, an 8K byte block of data can be stored in an array of sixteen 2817As (32K x 8 of E²PROM memory) in 1/16 the normal time, that is, 8192 x 20 milliseconds x 1/16 = 10 seconds. This is particularly useful in designs that require a large amount of non-volatile memory to be used in harsh environments. No additional hardware or software is required: the memory address decoder is simply connected according to the schematic diagram in Figure 29. The result is a decoding of the least significant address lines rather than the most significant address lines.

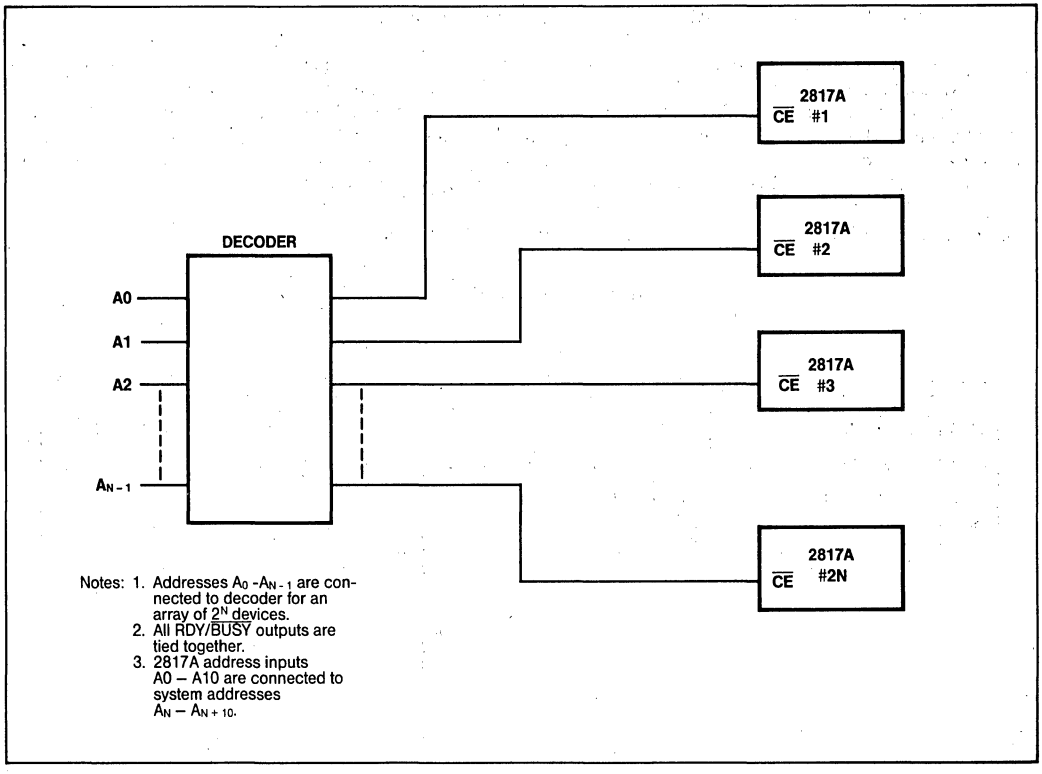


Figure 29. Fast Array Programming

To transfer data to the E²PROM memory, a series of high-speed, system write cycles are performed, one immediately after the other, to the E²PROM memory array. Each 2817A will internally perform a write operation using one of the successive bytes. For example, if the array in Figure 29 had 64 2817As, then 64 bytes would be written in each burst of high-speed write cycles. The total programming time for each 64-byte block would be the same as for one 2817A: 20 milliseconds.

Effectively, the total time needed to write a block of data into the array shown in Figure 29 is equal to:

Total Write Time =

$$\frac{\text{write time (20 ms) x \# of bytes to be stored}}{\text{\# of 2817A devices in the array}}$$

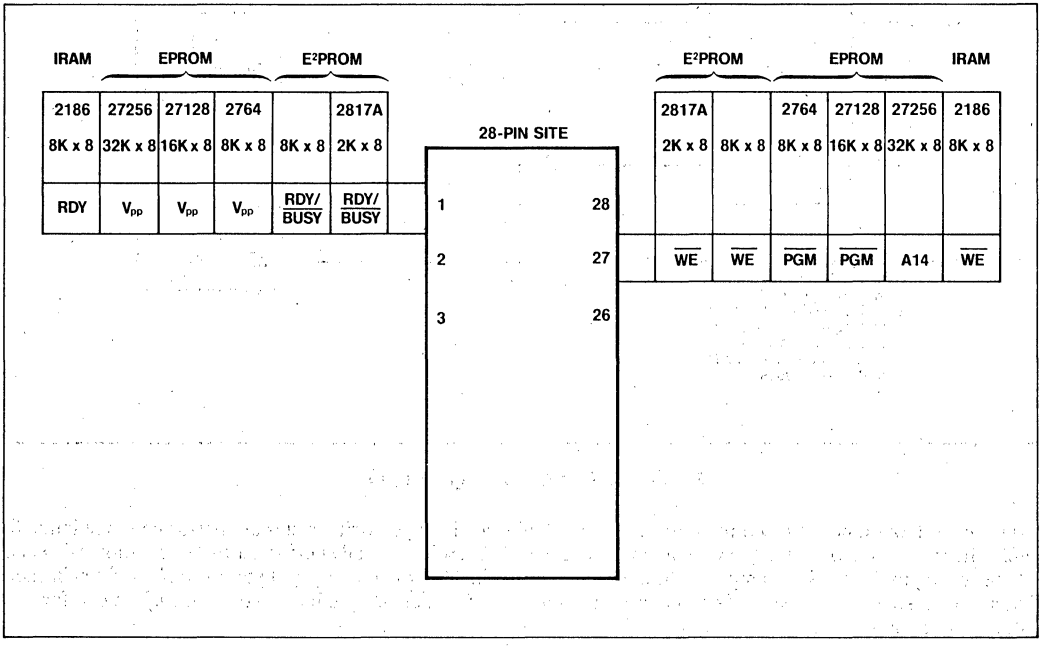
The more 2817As there are in the array, the shorter the write time for a given block of data. For example, writing a block of 16K bytes to an array of 32 2817As (64K bytes of E²PROM) would take 10 seconds (12K bits per second). A more significant application would be a ¼ megabyte E²PROM array (128 2817As). Writing 100K bytes to such an array would take only 15 seconds, which is 51K bits/second.

MULTI-DEVICE CONFIGURATION ON THE 28-PIN SITE

Intel's line of 28-pin memory devices is designed to be easily interchangeable on a single 28-pin site. The hardware and software designer need no longer bear the risk of board re-design due to inaccurate estimations of the amount and types of memory required. E²PROMs, EPROMs, and RAMs can be mixed and matched in the same 28-pin site. A small number of jumpers can be used to accommodate the different signals that use pins 1 and 27. This section has examples of how to lay out a printed circuit board memory array to accommodate any combination of 28-pin memories, including the multi-device sites on the 2817A/8088 applications demo.

The chart in Table 1 shows the different signals for pins 1 and 27 for Intel's 28-pin memories.

Table 1. 28-Pin Memory Device Compatibility



2817A/8088 Applications Demo Multi-Device Sites

Figure 30 shows a pin matrix of nine pins that can be used to allow different types of 28-pin memory devices to be plugged into socket U 18 on the 2817A/8088 applications demo board. A 27128 UV EPROM is normally plugged into socket U18 in Figure 30. A 2817A E²PROM is plugged into socket U17, and a 2186 iRAM resides in socket U19. Figure 31 shows which jumper pairs to connect for any one of the following devices in socket U18:

2817A	2K x 8 E²PROM
	8K x 8 E²PROM
2764	8K x 8 UV EPROM
27128	16K x 8 UV EPROM
27256	32K x 8 UV EPROM
2186	8K x 8 iRAM

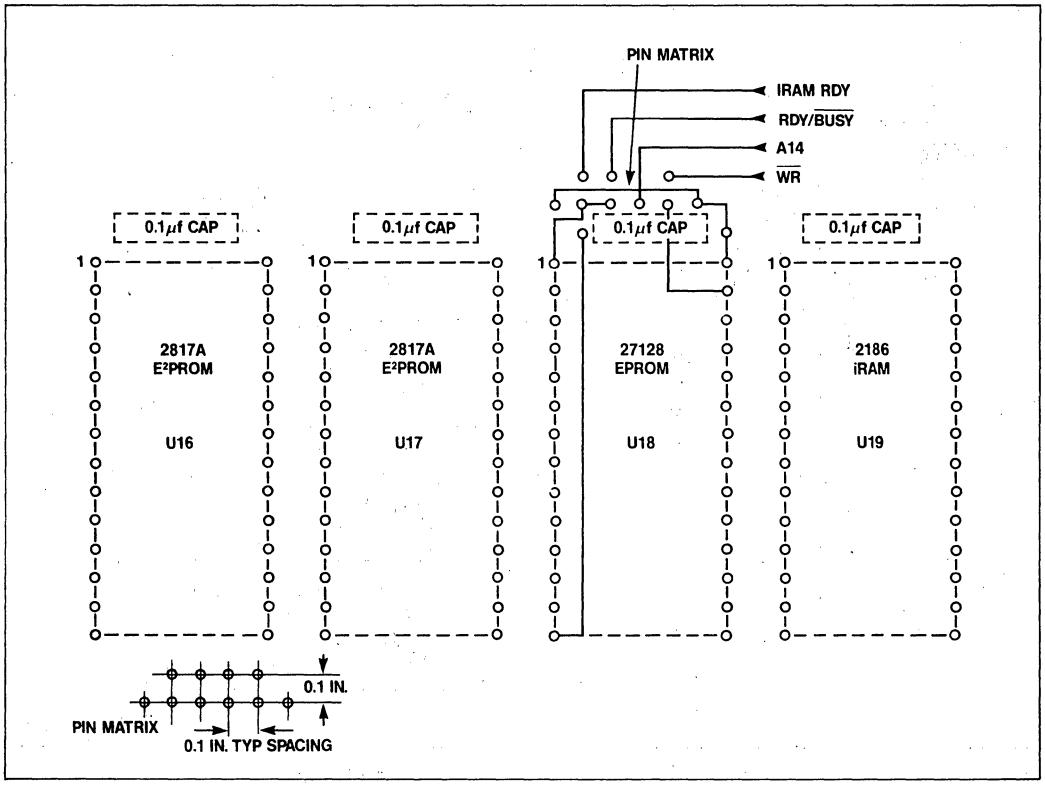


Figure 30. 2817A/8088 Applications Demo 28-Pin Site Array

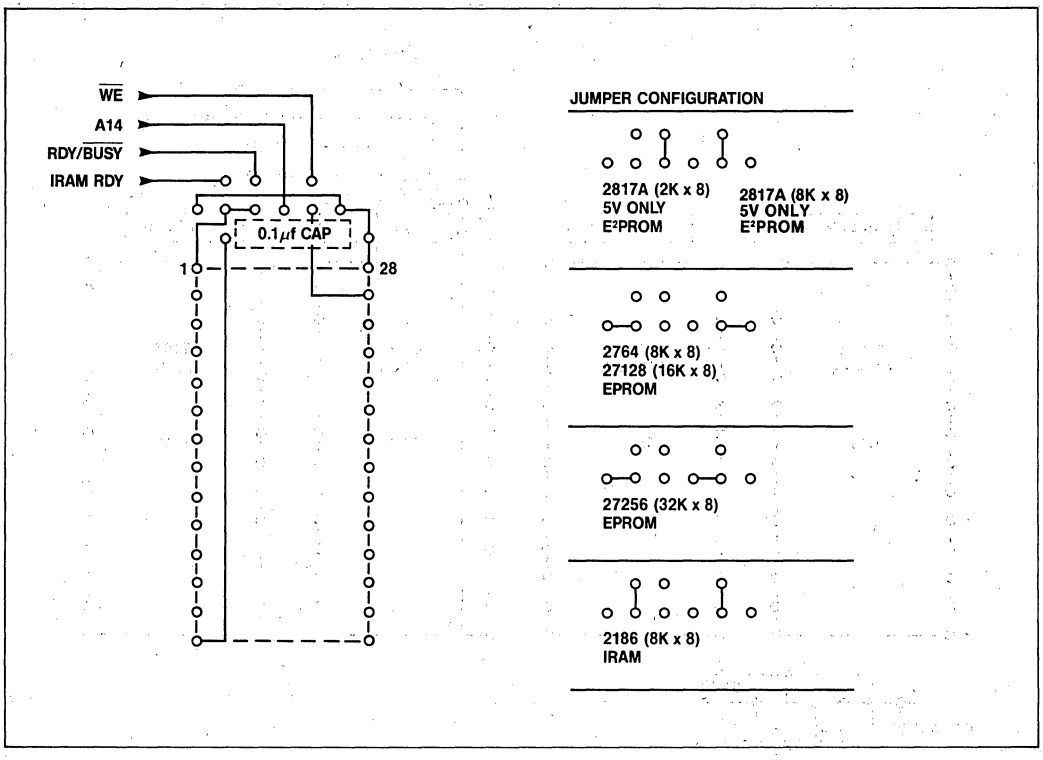


Figure 31. 2817A and 28-Pin EPROM/iRAM Site

If three 2817A's are plugged into the board in sockets U16, U17, and U18, a new firmware package can be remotely downloaded via the comline interface. The program code for performing the downloading operation would reside in the 2817A in socket U18. The routine in this E²PROM would perform most of the downloading operation. After the 2817As in sockets U16 and U17 have been loaded, program control would be transferred to one of the E²PROMs in those sockets in order to load the 2817A in socket U18 (see also "Application: Firmware Remote Downloading" on page 3).

By designing with the JEDEC-compatible 28-pin site and using a small number of jumbers as shown in the examples, boards and systems can be designed to be highly flexible and to have a long product life.

2817A/8088 APPLICATIONS DEMO BOARD CIRCUIT OPERATION

The memory map in Figure 32 shows the organization of the memory space in this design. The input/output map in Figure 33 shows the addresses of the I/O ports. Figure 34 shows a simple block diagram of the 2817A/8088 demo board. Figure 35 contains a more detailed block diagram. The complete schematic for the board is in Appendix B.

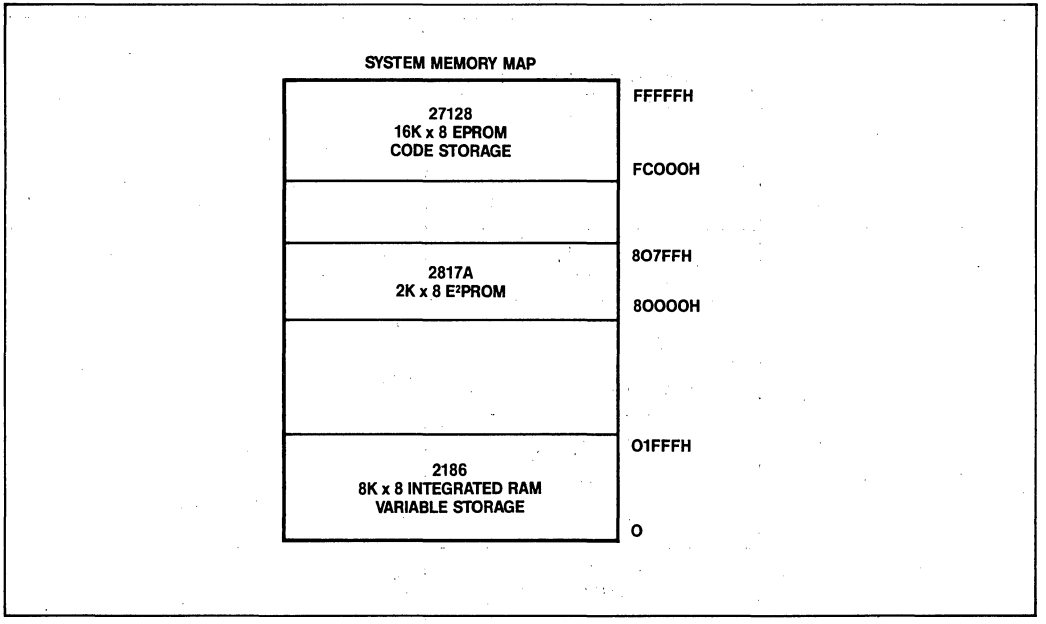


Figure 32. 2817A/8088 Applications Demo Board

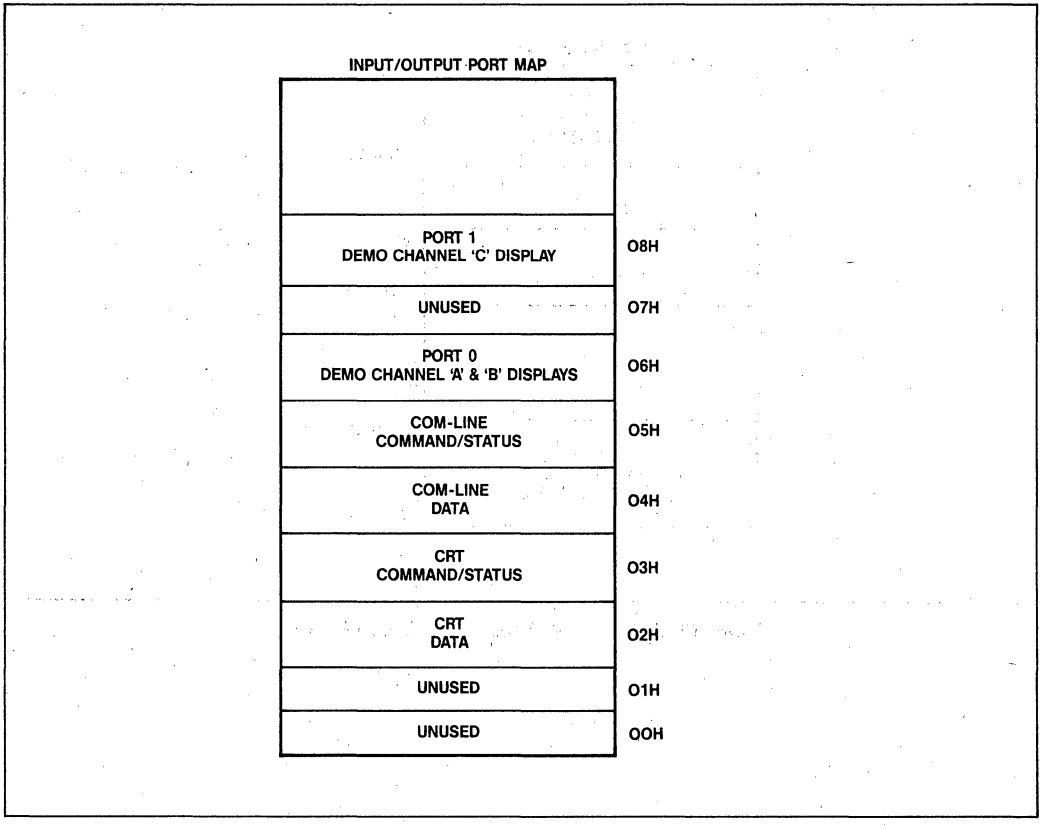


Figure 33. 2817A/8088 Applications Demo Board

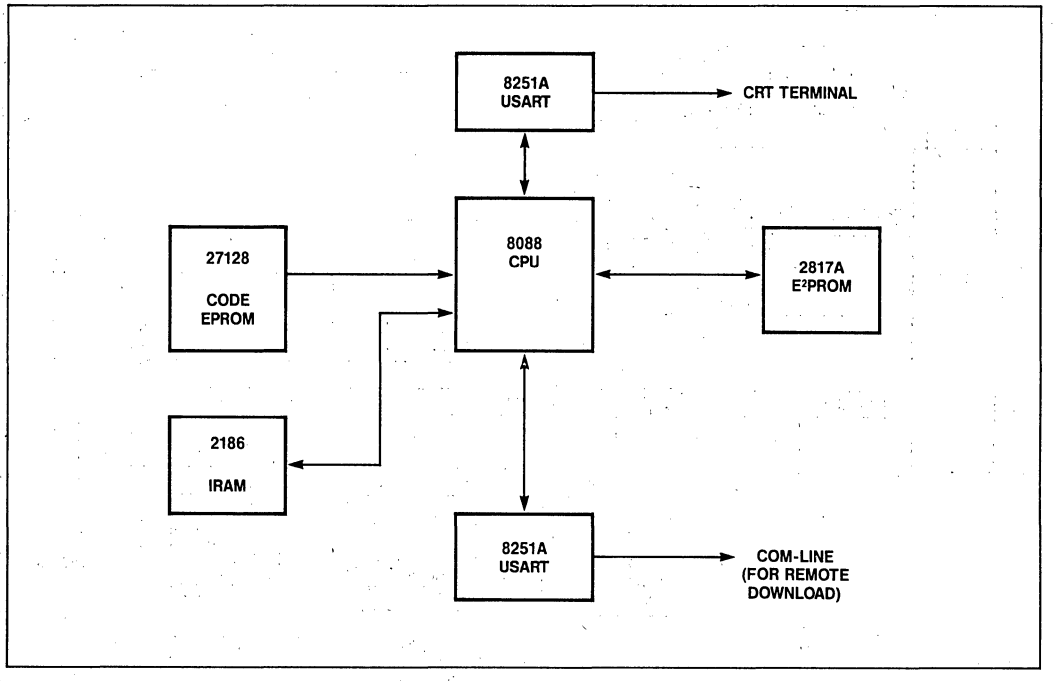


Figure 34. 2817A/8088 Applications Demo Board Simplified Block Diagram

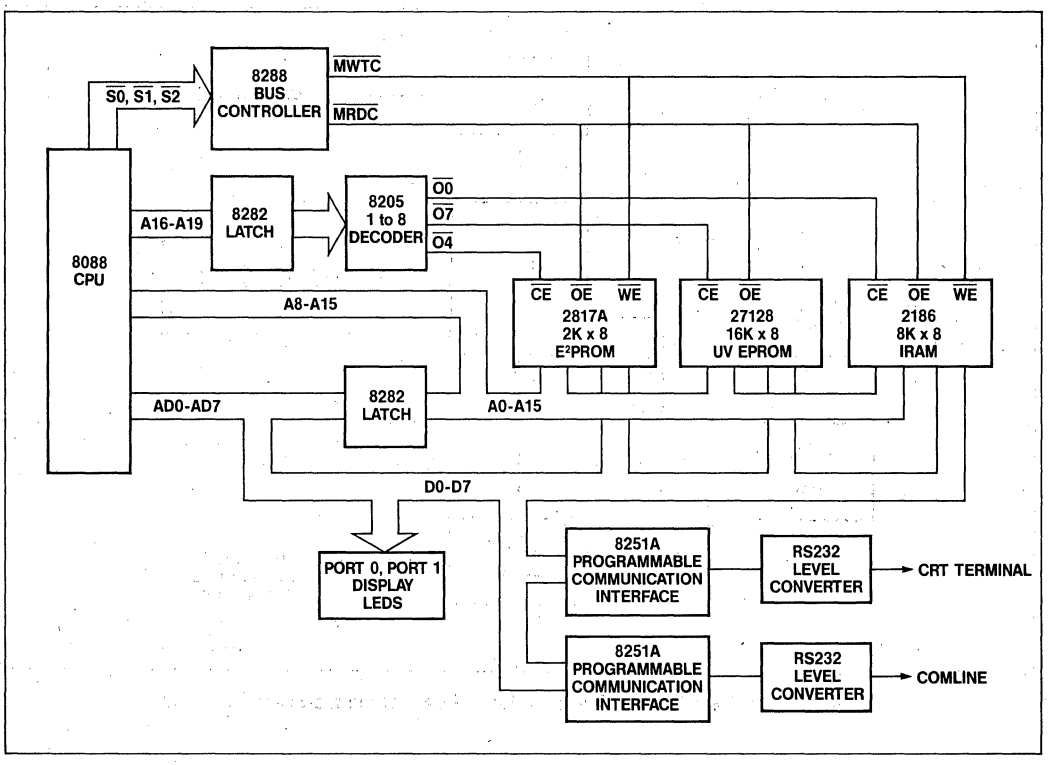


Figure 35. 2817A/8088 Applications Demo Board: Full Block Diagram

2817A Interface Circuitry: The RDY/$\overline{\text{BUSY}}$ Output (Figures 37 and 38)

The 2817A's RDY/$\overline{\text{BUSY}}$ output is connected to the $\overline{\text{TEST}}$ input of the 8088 after being inverted. This allows the 8088 CPU to go into a wait mode via the WAIT instruction while 2817A write operations are in progress. For more detailed information, see "How to Use the RDY/$\overline{\text{BUSY}}$ Output" on page 19.

CPU: The 8088 8-bit Microprocessor (Figure 37)

The 8088 microprocessor has an 8-bit wide external data bus and a one megabyte direct addressing capability. It also has an address/data multiplexed bus, on which the 8-bit data bus is multiplexed with the lower 8 bits of the address bus. The 8088's internal data/address paths are all sixteen bits wide. It has a full range of arithmetic capabilities including multiplication and division, all operating directly on sixteen-bit quantities. All of the arithmetic operations can be performed on any one of the eight 16-bit general purpose registers. The 8088 also has string operation commands which allow high-speed processing of large data blocks.

MEMORY

2817A-2K x 8 E²PROM (Figure 38)

The 2817A is used to store user messages entered from the keyboard and sent over the comline channel from other demo boards. The 2817A also stores the maintenance log information, operator's initials entered from the keyboard, and the system re-configuration demo status bytes.

27128-16K x 8 UV EPROM (Figure 40)

The 16K byte demo program, which consists of 5K bytes of 8088 code and 11K bytes of ASCII text data, is stored in the 27128. This 16K byte EPROM is located at the top of the 8088's memory space as shown in Figure 32.

2186-8K x 8 Integrated RAM (Figure 40)

This device has a dynamic RAM memory array with on-chip refresh controller circuitry. The stack, buffer space, and variables for the 8088's program code are stored in the 2186. The only external circuitry required is the $\overline{CE}$ pulse forming circuit shown in Figure 40. This circuit generates a single, glitchless $\overline{CE}$ pulse each time the iRAM memory is to be accessed. The operation of the $\overline{CE}$ pulse circuit is as follows: the iRAM $\overline{CE}$ output will not go active low until both the memory space of the iRAM is selected and the ALE pulse occurs. The iRAM $\overline{CE}$ signal will then stay low until the iRAM's READY output goes back high and the 74S112 flip-flops are cleared on the falling edge of the CLK signal.

Demo Channel Display LED Ports (Figure 42)

Two 74LS374 8-bit D flip-flop devices are used to store the system re-configuration demo status bits for display with LEDs. Output ports 0 and 1 are used to control nine light-emitting diodes. The output states of these bits are determined by the system re-configuration look-up table in E²PROM (see Figure 14). When the demo board is powered up, the firmware reads the look-up table to determine what states the port bits should be in. These port bits could easily control peripheral channelling between systems and peripheral devices in a computer room. The port bits could also control system parameters such as communications modes, baud rates, display formats, and "soft key" definitions.

8251A Programmable Communications Interface

Hardware (Figure 41)

Baud Rate

The first counter in the 74LS393 is used to generate the 4800 baud rate for the CRT terminal comm(communications)-interface. The baud rate is produced by dividing the PCLK 2.5 MHZ clock by 8 to obtain a 312.5KHZ clock. The 2nd 4-bit counter in the 74LS393 is used to genrate the 300 baud rate for the Comline comm-interface. This is done by dividing the 312.5KHZ clock by 16 down to 19.5KHZ.

RS232 Level Interface Circuit

A standard RS232 serial line driver and receiver circuit requires ±12V power supplies to generate the required ±12V signal levels. To reduce the complexity of operating this demo the RS232 level shifting circuits were designed so that they need only a +5V power supply. These circuits allow the demo board to communicate with any system that has standard RS232 level inputs and outputs. The level shifter circuit shown with the CRT interface, however, will not actively assert a valid logic "0" level when connected to another level shifter exactly like it. To allow two demo boards to communicate with each other the level shifter circuit at the top of Figure 41 has to be modified. For this reason the Comline level shifter at the bottom of Figure 41 is slightly different than the CRT interface level shifter at the top of the Figure. This allows the Comline channels of two boards to be connected together so that data can be transferred between demo boards to be ultimately stored in E²PROM.

An example of a circuit that will provide the full RS232 spec + and -12V signal levels is shown in Figure 36.

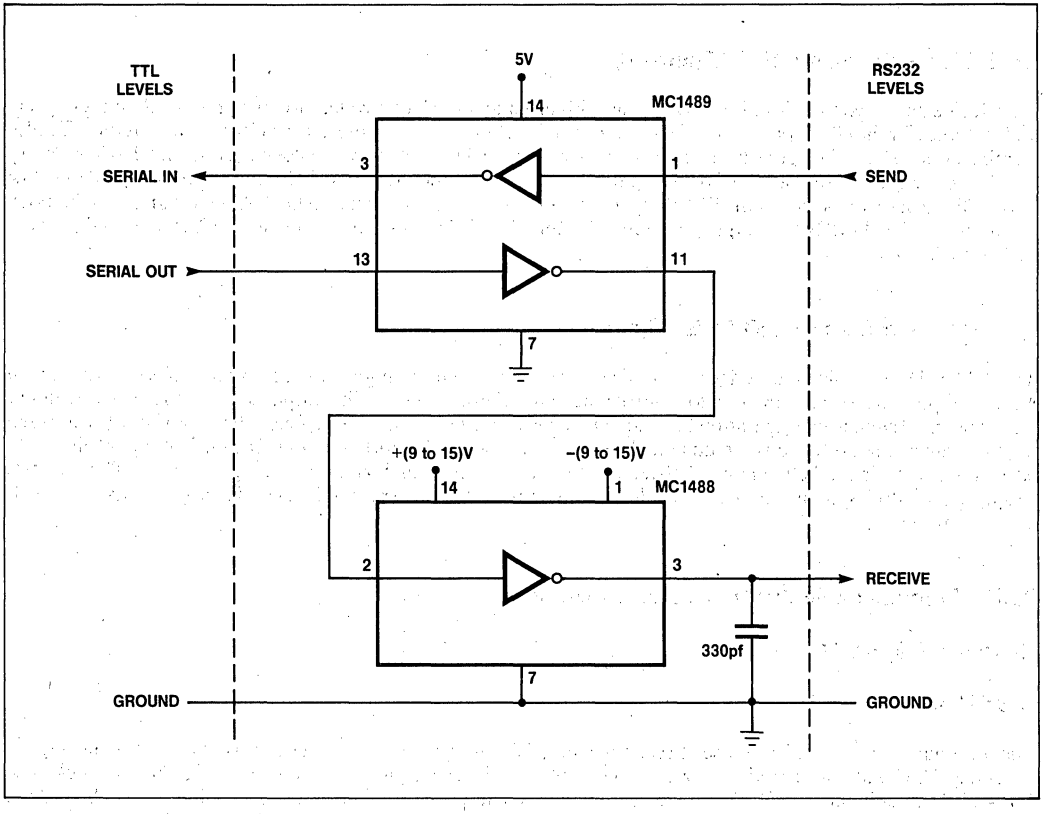


Figure 36. Full Spec TTL to RS232 Level Converter Circuit

8251A Software Initialization

The following software listing shows the procedure used on the 2817A/8088 applications demo to initialize the 8251A Programmable Communication Interface devices.

```
;
;
;
;
;            EQUATES
;
;
MODE_INSTR        EQU         11001111B         ;ASYNCHRONOUS MODE
                                                ;64X BAUD RATE
                                                ;8-BIT CHARACTER LENGTH
                                                ;PARITY DISABLED
                                                ;2 STOP BITS
;
COMMAND_INSTR     EQU         00100101B         ;TRANSMIT ON
                                                ;DTR/ = 1
                                                ;RECEIVE ENABLE


                                                ;NORMAL OP - NO BREAK CHAR
                                                ;NO ERROR RESET
                                                ;RTS/ = 0


;
CRT_PCI_CMD             EQU     03H     ;COMMAND MODE I/O PORT FOR
                                        ;THE CRT TERMINAL PROG'BLE COMM
                                        ;INTERFACE
;
COMLINE_PCI_CMD         EQU     05H     ;COMMAND MODE I/O PORT FOR
                                        ;THE COMLINE USART
;
;


;
;       CODE TO INITIALIZE THE 8251A'S
;
;
        MOV         DX,CRT_PCI_CMD
        CALL        INIT_PCI            ;INITIALIZE THE CRT COMM-INTERFACE
;
        MOV         DX,COMLINE_PCI_CMD
        CALL        INIT_PCI            ;INITIALIZE THE COMLINE PCI
;
;
;
;
```

```
;
;*************************************************************************
;
;               SUBROUTINES
;
;
;
INIT_PCI:
;
                                                ;INITIALIZE THE COMM-INTERFACE
                                                ;DEVICE SPECIFIED
                                                ;IN THE 'DX' REGISTER.

;
;
        MOV     AL,MODE_INSTR
        OUT     DX,AL
;
        MOV     AL,COMMAND_INSTR
        OUT     DX,AL
;
        RET
;
;
```

Support Devices

8284A-The Clock Generator (page 50)

This device produces a clean, stable clock for the 8088 with the correct 33% duty cycle.The 8284A also interfaces the system's asynchronous iRAM READY signal with the 8088, as well as generating the RESET signal.

8288-Bus Controller (page 52)

This device generates the memory access signals (memory read, memory write, ALE, etc.) for the system.

APPENDIX A
SOFTWARE LISTINGS

Routines for Data Transfer Between 2817A/8088 Demo Boards over the 'COMLINE' Communications Interface Channel

This is the routine for the system sending the data block.

```
;
;
COMLINE_PCI_DATA        EQU     4H      ;8251A PROGRAMMABLE COMMUNICATIONS
                                        ;INTERFACE DATA PORT
;
COMLINE_PCI_CMD         EQU     5H      ;PROG'BLE COMM-INTERFACE COMMAND/
                                        ;STATUS PORT
;
;
;
;
        MOV     AH,SOT
        CALL    SEND_CHAR               ;SEND A 'START OF TRANSMISSION' CHARACTER
                                        ;(THE OTHER SYSTEM WILL RECEIVE THE 'SOT'
                                        ;CHARACTER AND RETURN AN ACKNOWLEDGEMENT
                                        ; CHARACTER)
;
        CALL    REC_CHAR                ;GET A CHARACTER FROM THE OTHER SYSTEM
        CMP     AH,ACK                  ;IS IT AN 'ACKNOWLEDGE' CHARACTER?
        JNE     NOT_READY               ;IF NOT, INFORM OPERATOR THAT THE
                                        ;OTHER SYSTEM IS NOT READY.
;
        CALL    DELAY                   ;SMALL DELAY TO LET THE OTHER GET READY
;
        LEA     BP,DATA_BLOCK           ;LOAD POINTER TO DATA BLOCK
SEND_LOOP:
        MOV     AH,[BP]                 ;FETCH A BYTE
        CALL    SEND_CHAR               ;SEND IT
        CMP     AH,EOS                  ;END OF DATA STRING?
        JNE     SEND_LOOP               ;IF NOT, CONTINUE
;OTHERWISE, FINISH UP
```

Next, the software for the system receiving the data block:

```
;
        CALL    REC_CHAR                ;GET CHARACTER FROM THE COMLINE
        CMP     AH,SOT                  ;START OF TRANSMISSION CHAR?
;
        JE      CONT_1                  ;IF SO, CONTINUE
        JMP     MAIN_LOOP               ;IF NOT, RETURN TO MAIN LOOP
;
CONT_1:
        MOV     AH,ACK
        CALL    SEND_CHAR               ;SEND AN ACKNOWLEDGEMENT CHARACTER
                                        ; BACK TO THE OTHER UNIT
```

```
;
        LEA     BP,DATA_BLK_BUFFER          ;SET POINTER TO DATA BLOCK
                                            ;BUFFER
;
REC_BLK_LOOP:
        CALL    REC_CHAR            ;GET BYTE FROM THE COMLINE
        CMP     AH,EOS              ;END OF DATA BLOCK?
        JE      FINISH_UP           ;IF SO, FINISH UP
        MOV     [BP],AH             ;IF NOT, STORE BYTE AND GET ANOTHER
        JMP     REC_BLK_LOOP
;
FINISH_UP:
;
```

The following are the subroutines used for the software for data transfer between 2817A/8088 Demo boards.

```
;
;               SUBROUTINES
;
;
;
;
SEND_CHAR:                          ;TAKES THE BYTE IN THE 'AH' REG. AND SENDS
                                    ; IT OVER THE COM-LINE TO THE OTHER SYSTEM.
;
;
RDY_SEND_POLL:
        IN      AL,COMLINE_PCI_CMD
        TEST    AL,1                        ;READY TO SEND YET?
        JE      RDY_SEND_POLL               ;IF NOT, CHECK AGAIN
;
        MOV     AL,AH
        OUT     COMLINE_PCI_DATA            ;IF READY, SEND CHARACTER
        RET
;
;
;
;
REC_CHAR:                           ;GETS A BYTE FROM THE OTHER SYSTEM OVER
                                    ;THE COM-LINE(VIA THE COM-LINE COMM-
                                    ;INTERFACE) AND
                                    ;PASSES IT BACK TO THE CALLING ROUTINE IN
                                    ;THE 'AH' REG.
```

```
;
;
RDY_REC_POLL:
        IN      AL,COMLINE_PCI_CMD
        TEST    AL,2                    ;HAS A CHARACTER BEEN RECEIVED YET?
        JE      RDY_REC_POLL            ;IF NOT, TRY AGAIN
;
        IN      AL,COMLINE_PCI_DATA     ;IF READY, GET CHARACTER
        MOV     AH,AL                   ;SAVE IT
        AND     AH,7FH                  ;STRIP OFF THE PARITY BIT
        RET
;
;
```

Interrupt Mode Software

Status Subroutine

```
;
;
XFER_STATUS_SUBR:                       ;DETERMINES IF AN E2PROM TRANSFER IS
                                        ;ALREADY IN PROGRESS.  IF NOT, THE
                                        ;NEXT BYTE TO BE TRANSFERRED IS
                                        ;WRITTEN TO E2PROM.
;
;
        CMP     E2_BUSY,1               ;E2 WRITE IN PROGRESS?
        JNE     DO_TRANSFER             ;IF NOT, PROCEED
        RET                             ;IF SO, RETURN, SINCE THE NEW DATA
                                        ;IN THE RAM BUFFER TABLE WILL BE
                                        ;DETECTED BY THE INTERRUPT SUBROUTINE
                                        ;AND EVENTUALLY TRANSFERRED TO
                                        ;E2PROM.
;
DO_TRANSFER:
        CALLWRITE_E2_DATA               ;FIND NEXT BYTE TO BE TRANSFERRED AND
                                        ;WRITE IT TO THE E2PROM.
;
;
        RET
;
;
```

Interrupt subroutine

```
;
;
E2_INTERRUPT_SUBROUTINE:                ;EACH TIME AN E2PROM WRITE OPERATION
                                        ;ENDS, CHECK IF THERE ARE ANY MORE
                                        ;BYTES TO BE TRANSFERRED.  IF SO,
                                        ;FETCH THE BYTE AND WRITE IT TO
                                        ;THE E2PROM.
;
        PUSH    DX
        PUSH    AX
        MOV     AL,OCW2
        MOV     DX,PIC_A0_LOW           ;INFORM THE PROGRAMMABLE
        OUT     DX,AL                   ;INTERRUPT CONTROLLER THAT THE
        POP     AX                      ;INTERRUPT HAS BEEN SERVICED
        POP     DX
;
        CALL    WRITE_E2_DATA
        IRET
;
```

This subroutine determines if there is any data left to be transferred from the RAM Buffer table to the E²PROM table, and if there is, transfers the next byte.

```
;
;
WRITE_E2_DATA:                          ;CHECK FOR ANY MORE DATA TO BE
                                        ;TRANSFERRED.  IF NONE, CLEAR THE
                                        ;"E2_BUSY" FLAG.  IF ANOTHER BYTE
                                        ;IS TRANSFERRED, SET THE "E2_BUSY"
                                        ;FLAG.
;
;
        PUSH    SI
        PUSH    DI
;
        LEA     SI,RAM_BUFFER_TABLE     ;POINT THE SI REG. TO THE RAM
                                        ;BUFFER TABLE
;
        LEA     DI,E2PROM_TABLE         ;POINT THE DI REG. TO THE E2PROM
                                        ;LOOK-UP TABLE
;
        MOV     CX,10                   ;DEFAULT COUNT = SIZE OF LOOK-UP
                                        ;TABLE
;
REPE    CMPSB                           ;SEARCH FOR ANY NEW DATA BYTES
                                        ;IN THE RAM BUFFER TABLE.
;
        JZ      NO_NEW_DATA             ;IF NO NEW DATA, FINISH UP
;
        ;OTHERWISE, WRITE THE NEW DATA BYTE TO E2PROM
        DEC     SI
        DEC     DI                      ;BACK UP TO THE NEW, UNEQUAL BYTE
;
        MOVSB                           ;WRITE THE NEW BYTE TO THE 2817A
                                        ; (THE 2817A WILL NOW START A WRITE
                                        ;  OPERATION, INTERNALLY)
```

```
;
        MOV     E2_BUSY,1               ;SET THE FLAG
;
        JMP     E2_XFER_FINISH



NO_NEW_DATA:
        MOV     E2_BUSY,0               ;CLEAR THE FLAG
E2_XFER_FINISH:
        POP     DI
        POP     SI
        RET
;
;
```

This is how a section of the stack segment is allocated for the RAM Buffer table.

```
;
;
STACK     SEGMENT
;
;
;
RAM_BUFFER_TABLE    DB      ?       ;BYTE 1
                    DB      ?       ;BYTE 2
                    DB      ?       ;BYTE 3
                    DB      ?       ;BYTE 4
                    DB      ?       ;BYTE 5
                    DB      ?       ;BYTE 6
                    DB      ?       ;BYTE 7
                    DB      ?       ;BYTE 8
                    DB      ?       ;BYTE 9
                    DB      ?       ;BYTE 10
```

This is the space allocation for the actual E²PROM Look-up table in the 2817A.

```
;
;
E2PROM          SEGMENT
;
;
E2PROM_TABLE        DB      ?       ;BYTE 1
                    DB      ?       ;BYTE 2
                    DB      ?       ;BYTE 3
                    DB      ?       ;BYTE 4
                    DB      ?       ;BYTE 5
                    DB      ?       ;BYTE 6
                    DB      ?       ;BYTE 7
                    DB      ?       ;BYTE 8
                    DB      ?       ;BYTE 9
                    DB      ?       ;BYTE 10
```

The 'DI' register is used in the "WRITE-E2-DATA" subroutine above to reference the E²PROM. The following ASSUME directive and MOV instructions would be needed to assign the proper addresses to the segment registers:

```
;
      ASSUME        ES:E2PROM,SS:STACK
;
        MOV         AX,0
        MOV         SS,AX
        MOV         AX,8000H
        MOV         ES,AX
;
;
```

8259A Initialization

The following is the code to initialize the 8259A Programmable Interrupt Controller for use in the 2817A/8088 Demo board.

```
;                       EQUATES
;
;
;
;
INT_CNTRLR_A0_LOW       EQU     0H
INT_CNTRLR_A0_HIGH      EQU     1H
;
ICW1            EQU         00010111B       ;EDGE TRIGGERED MODE
                                            ;ADDR INTERVAL OF 4
                                            ;SINGLE MODE
                                            ;ICW4 NEEDED
```

```
;
ICW2        EQU     00100000B       ;TYPE 20H, (VECTOR OF 20H
                                    ;WILL COME FROM 8259A: IT'S
                                    ;VALUE IS MULTIPLIED BY 4,
                                    ;SO THAT AN INTERRUPT 0 TO
                                    ;THE 8259A WILL RESULT IN THE
                                    ;VECTOR BEING FETCHED FROM
                                    ;LOCATION 80H)
;
ICW4        EQU     00000001B       ;NOT SPECIAL FULLY NESTED
                                    ;MODE
                                    ;NON-BUFFERED MODE
                                    ;NORMAL EOI
                                    ;8086 MODE
;
OCW1        EQU     11111110B       ;MASK OUT ALL BUT INTERRUPT 0
;
OCW2        EQU     00100000B       ;END OF INTERRUPT COMMAND
                                    ;NON-SPECIFIC EOI COMMAND
;
;


;
;           RAM MEMORY SPACE
;
;
;
STACK       SEGMENT AT 0                ;RAM AREA LOCATED FROM 0 TO 1FFFH
            DB      128 DUP(?)          ;RESERVED LOC'S


;
INTR_VECT   DW      ?                   ;E2PROM READY INTERRUPT SERVICE
                                        ;SUBROUTINE VECTOR - OFFSET.
;
            DW      ?                   ;SEGMENT VALUE
;
;
E2_BUSY     DB      ?                   ;SOFTWARE FLAG TO INDICATE THAT
                                        ;AN E2PROM BLOCK TRANSFER IS IN
                                        ;PROGRESS
;
;
            ORG     1FFFH
TOP_OF_STACK DW     ?
;
STACK       ENDS
;
;
;
;
```

```
;
;           CODE TO INITIALIZE THE 8259A
;
;
;
        OUT     INT_CNTRLR_A0_LOW, ICW1     ;1ST INITIALIZATION
                                            ;CONTROL WORD
;
        OUT     INT_CNTRLR_A0_HIGH, ICW2    ;2ND INITIALIZATION CONTROL
                                            ;WORD
;
        OUT     INT_CNTRLR_A0_HIGH, ICW4    ;LAST INIT. CONTROL WORD


        OUT     INT_CNTRLR_A0_HIGH, 0CW1    ;MASK OUT UNUSED INTR LINES
;
        MOV     AX, OFFSET INTR_SERV_ROUT
        MOV     INTR_VECT, AX               ;LOAD THE OFFSET VALUE OF THE
                                            ;INTERRUPT SERVICE ROUTINE INTO
                                            ;THE 1ST TWO BYTES OF THE
                                            ;INTERRUPT VECTOR DOUBLEWORD
;
        MOV     AX, CS
        MOV     INTR_VECT+2, AX             ;LOAD THE SEGMENT VALUE INTO THE
                                            ;2ND TWO BYTES OF THE INTERRUPT
                                            ;VECTOR DOUBLEWORD
```

APPENDIX B
2817A/8088 APPLICATIONS DEMO

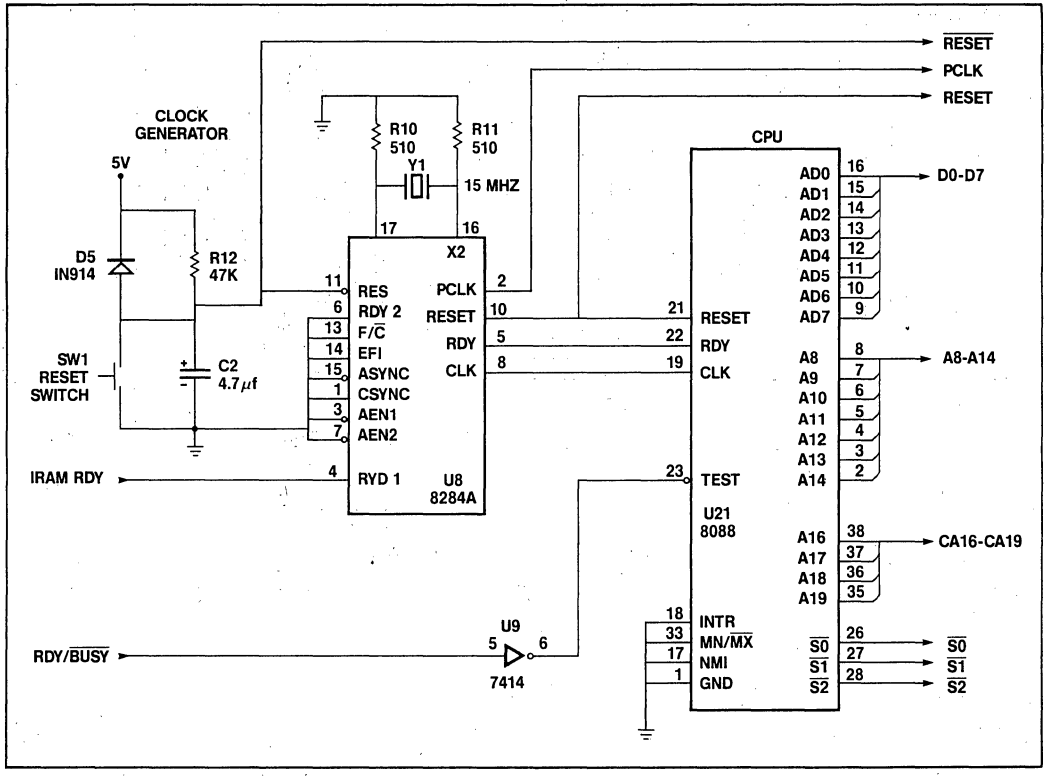


Figure 37. 2817A/8088 Application Demo

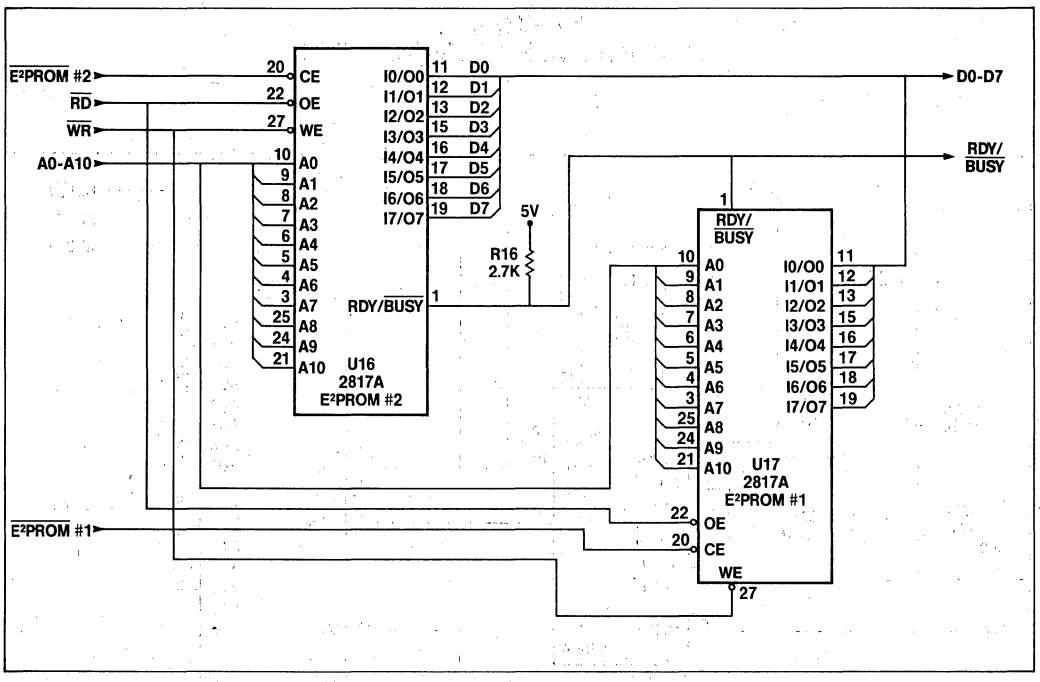


Figure 38. 2817A E²PROM Sockets

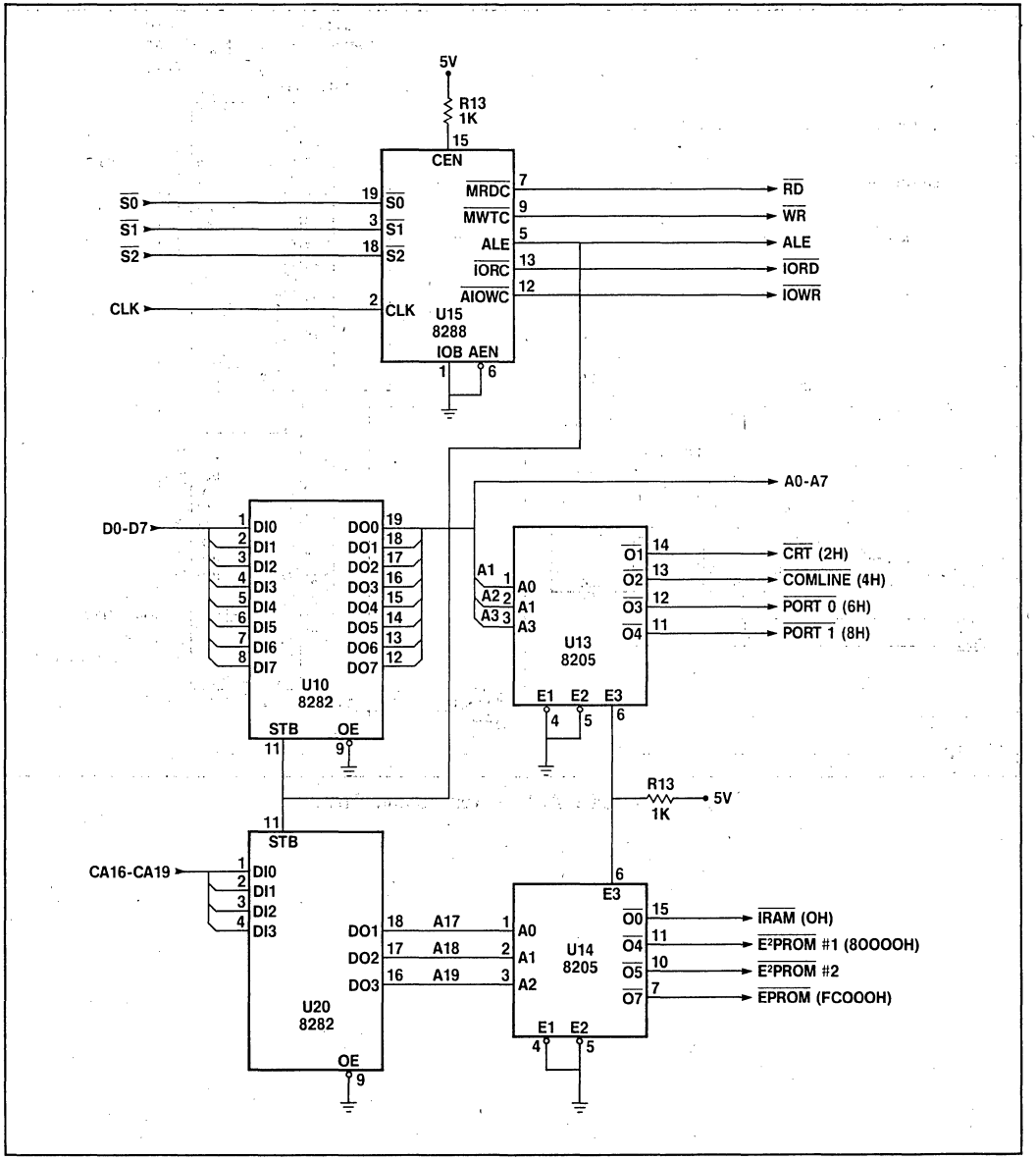


Figure 39. 2817A/8088 Applications Demo

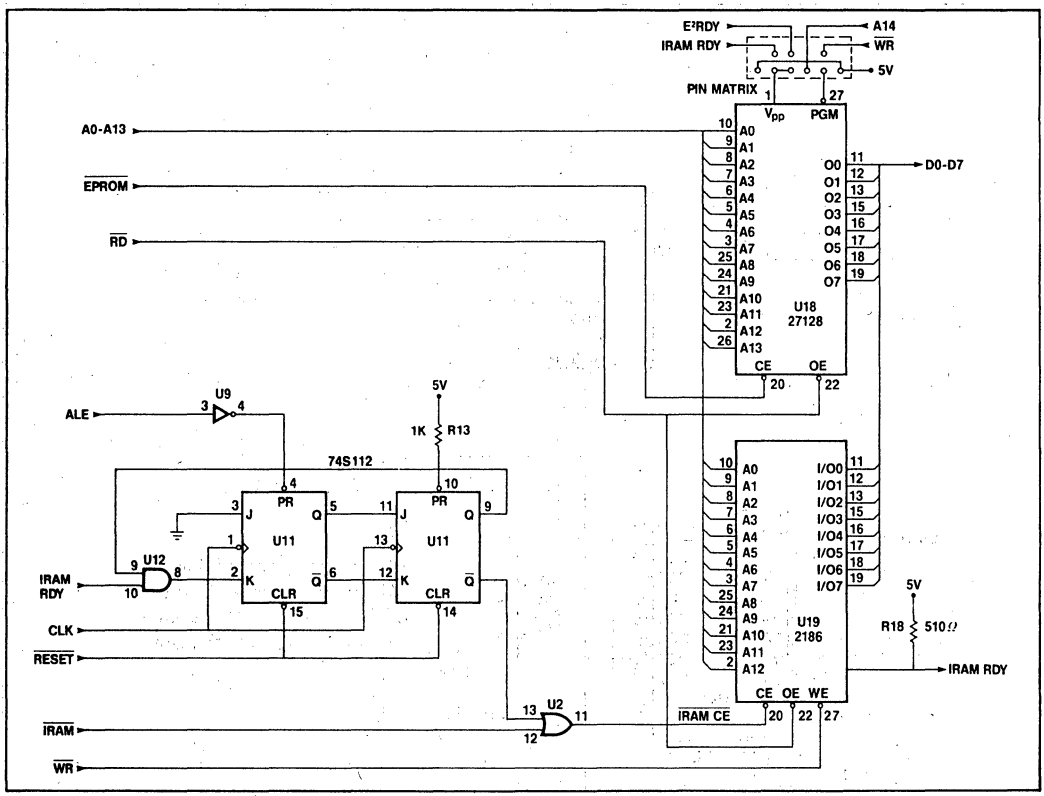


Figure 40. 2817A/8088 Applications Demo

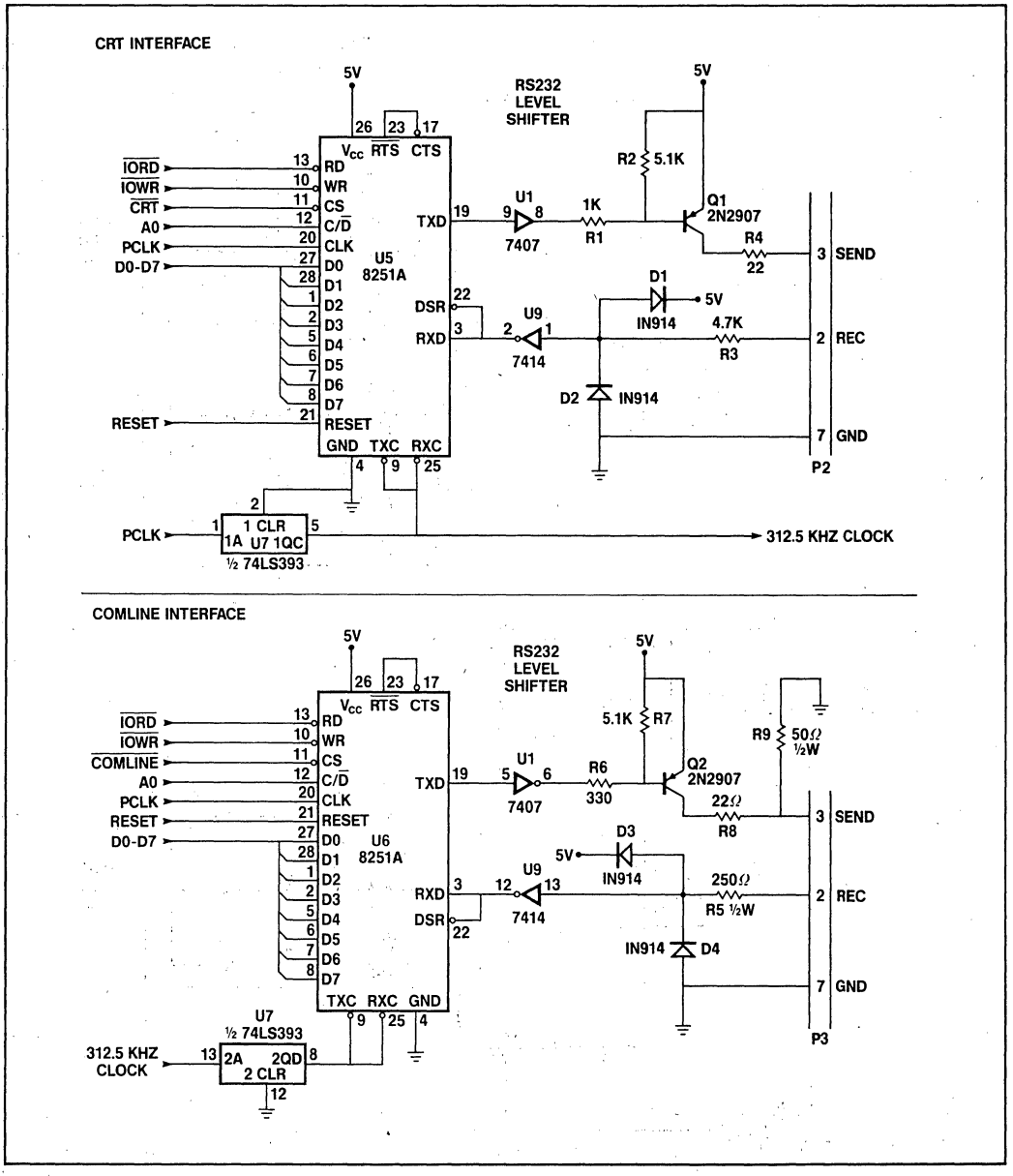


Figure 41. 2817A/8088 Applications Demo

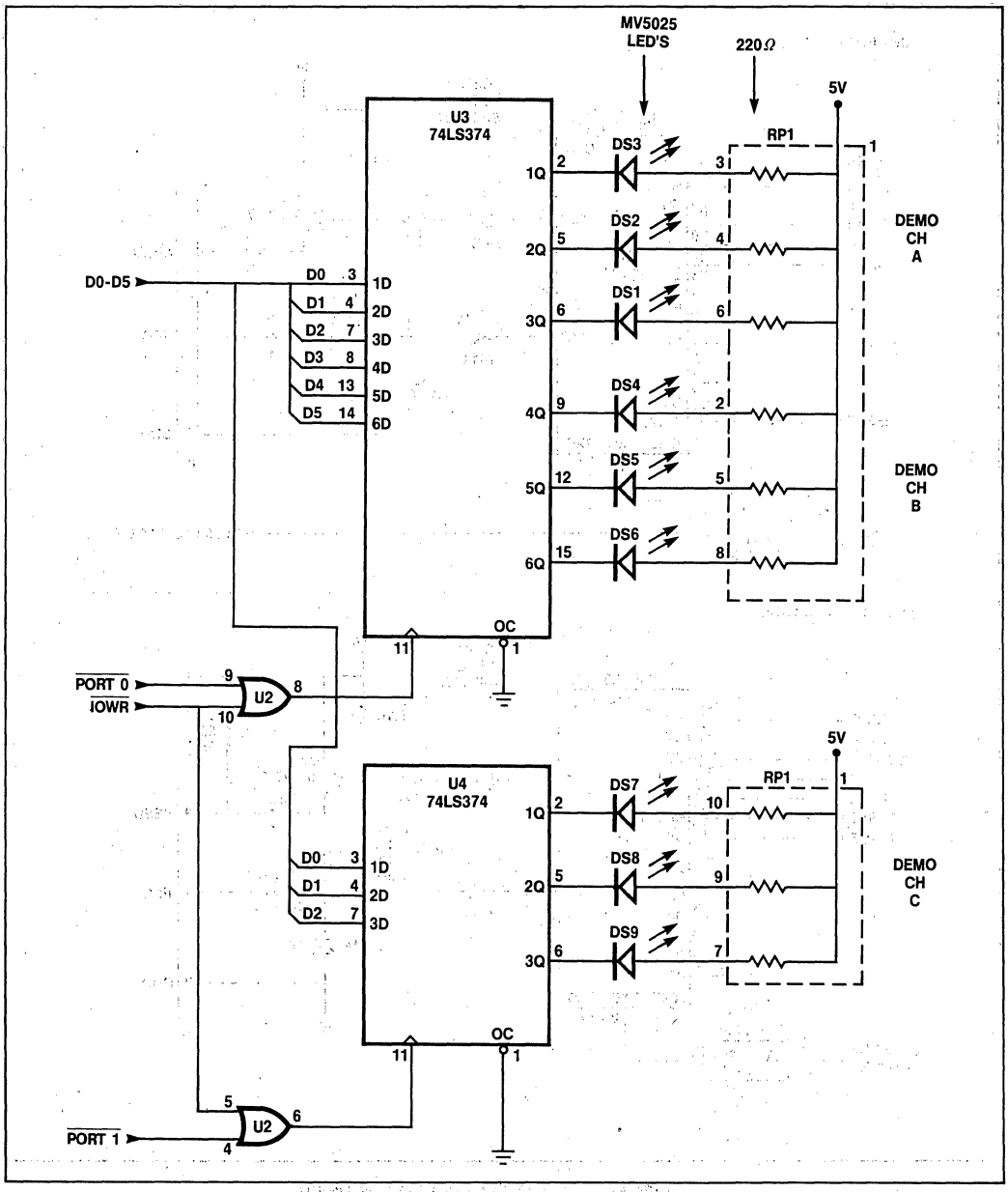
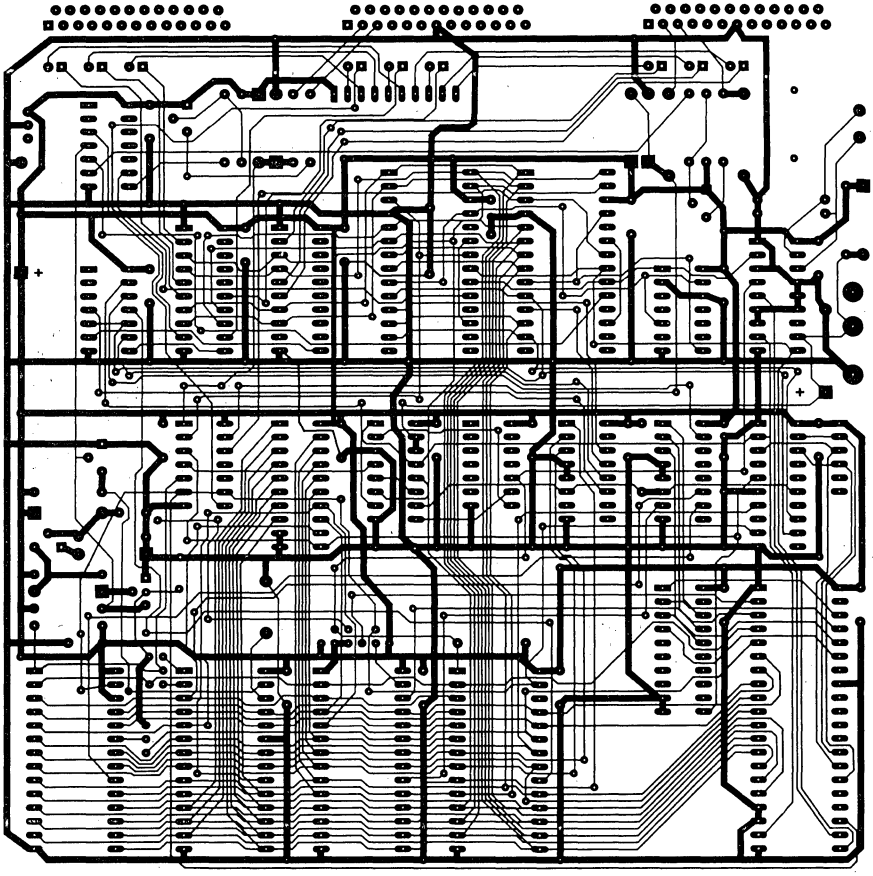


Figure 42. 2817A/8088 Applications Demo, Demo Channels A, B, C Display LED's

APPENDIX C
PRINTED CIRCUIT BOARD LAYOUT



APPENDIX D
PARTS LIST

2817A/8088 Applications Demo Parts List

IC's

Quantity	Description
1	D8088
1	D8284A
2	D8282
1	C2186-30
1	D27128-3
2	D3205
2	D8251A
1	74LS393
1	7407
1	74LS14
1	D8288
2	74LS374
1	74LS32
1	74S112
1	74LS08
1	2817A

Diodes

5	1N914
9	MV-5025

Capacitors

1	4.7 Microfarad
22	0.1 Microfarad Ceramic
1	25 Microfarad, 25V

Transistors

2	2N2907

Switches

1	Micro Switch #21SM284

Crystal

1	15 Megahertz

Sockets

4	28 Pin Low Profile Soldertail

Connectors

3	25 Pin RS232 Connector ITTJO-DBP-25SCA 8040
1	Molex Connector for Power Supply Housing #22-01-2037 2695 Series Peg #15-04-9210 Wafer #22-05-3031.7478 Series Terminals #08-56-0110 2759 Series

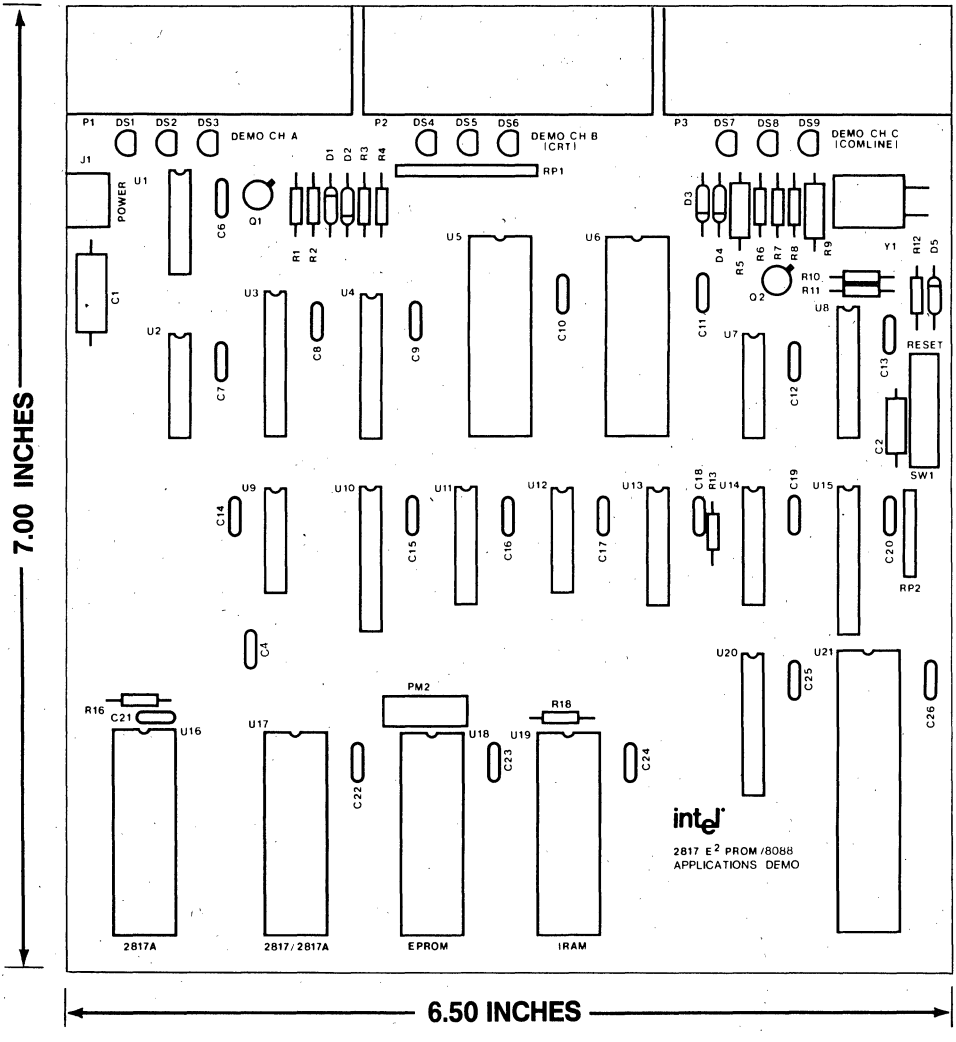
Headers

1	6 Pair Unit

Resistors

1	2.7K Ohm
1	4.7K Ohm
2	1K
3	510 Ohm
1	47K
2	5.1K
2	22 Ohm
1	330 Ohm
1	250 Ohm ½W
1	50 Ohm ½W
1	220 Ohm R-Pak 9 Pin SIP

APPENDIX E
FABRICATION/LAYOUT DRAWING



intel®

ARTICLE REPRINT

AR-119

March 1980

16-K EE-PROM Relies On Tunneling For Byte-Erasable Program Storage

W. S. Johnson, G. L. Kuhn, A. L. Renninger, and G. Perlegos

Electronics, February 28, 1980

The electrically erasable programmable read-only memory, or EE-PROM, will one day be the standard form of program storage in microprocessor-based systems. It will follow in the steps of the ultraviolet-light–erasable PROM, for it, too, will become available in increasingly larger byte-wide arrays and will in time share silicon with single-chip microcomputers.

As with the E-PROM, the success of the EE-PROM described in this article hinges upon the mastery of a difficult process. The floating-gate avalanche cell, also pioneered by Intel, is a tricky construction that still eludes many a memory maker. Likewise, the widespread availability of large EE-PROMs is still years off.

The EE-PROM process will be perfected, though, because the rewards go beyond the elimination of the expensive quartz window on the E-PROM package. The electrically erasable memory will usher in systems previously not practical. The microprocessor system whose programs can be altered remotely, as by phone, is one example. Another is the system that is immune to power outages, as it protects its contents in ROM. Perhaps most important, systems will be able to adjust their own program memory to environmental changes.

To be sure, there is more than one way to build an EE-PROM. The metal-nitride-oxide-semiconductor (MNOS) structure has served for years in modest-sized arrays for TV tuning applications, for example. In fact, a year ago Hitachi Ltd. announced a 2-K-by-8-bit MNOS replacement for the 2716 E-PROM. Compatibility with the 2716 is the impetus behind the device described in the following article, but it uses only silicon and its derivatives, plus metal. Also, in place of avalanche injection, which can injure a cell, electrons tunnel to and from a floating gate.

-John G. Posa

16-K EE-PROM relies on tunneling for byte-erasable program storage

Thin oxide is key to floating-gate tunnel-oxide (Flotox) process used in 2,048-by-8-bit replacement for UV-light–erasable 2716 E-PROM

by W. S. Johnson, G. L. Kuhn, A. L. Renninger, and G. Perlegos, *Intel Corp., Santa Clara, Calif.*

☐ The erasable programmable read-only memory, or E-PROM, is the workhorse program memory for microprocessor-based systems. It is able to retain data for years, and it can be reprogrammed, but to clear out its contents for new data, ultraviolet light must be made to stream through its quartz window. This works well for many applications, but the technique foregoes single-byte—in favor of bulk—erasure and in-circuit self-modification schemes.

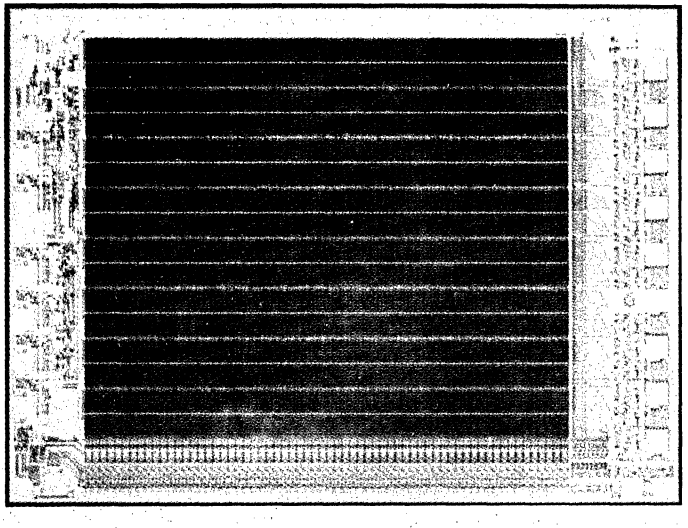
Electrical erasability is clearly the next step for such memories, but like ultraviolet erasure a few years back, it is hard to achieve. In fact, the design of an electrically erasable read-only memory is paradoxical. In each cell, charge must somehow be injected into a storage node in a matter of milliseconds. Once trapped, however, this charge may have to stay put for years while still allowing the cell to be read millions of times. Although these criteria are easily met individually, the combination makes for a design with conflicting requirements.

These demands are more than met in a new EE-PROM, which is a fully static, 2-K-by-8-bit, byte- or chip-erasable nonvolatile memory. At 16,384 bits, this new design not only meets the goal of high density, but also has long-term retention, high performance, and no refreshing requirement, in addition to functional simplicity unmatched by present nonvolatile memories. The device need not be removed from a board for alterations, and performance is consistent with the latest generation of 16-bit microprocessors such as the 8086.

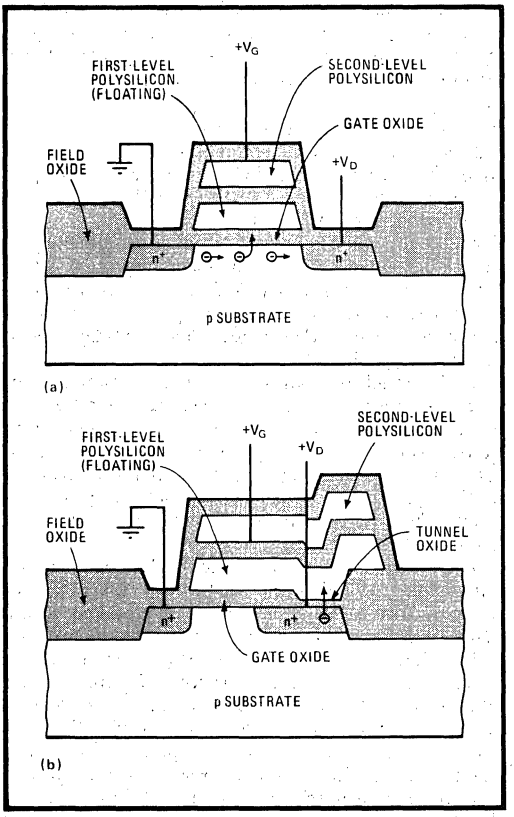
This achievement required the development of a new nonvolatile process technology, HMOS-E, as well as a new cell structure, Flotox, for floating-gate tunnel oxide.

Conflicting requirements

Nonvolatile semiconductor memories generally store information in the form of electron charge. At cell sizes achievable today, this charge is represented by a few million electrons. To store that many electrons in a 10-millisecond program cycle requires an average current on the order of 10^{-10} amperes. On the other hand, if it is essential that less than 10% of this charge leaks away in 10 years, then a leakage current on the order of



The next memory. The 16-K electrically erasable programmable read-only memory is eminently suitable for microprocessor program storage. Organized as 2,048 by 8 bits, the EE-PROM allows full-chip or individual-byte erasure using the same supply (V_{pp}) as for programming.



1. First Famos, now Flotox. The Famos cell (a) found in all E-PROMs stores charge on the floating gate by avalanche means. Flotox cell (b), the heart of the EE-PROM, relies on electron tunneling through thin oxide to charge and discharge the floating gate.

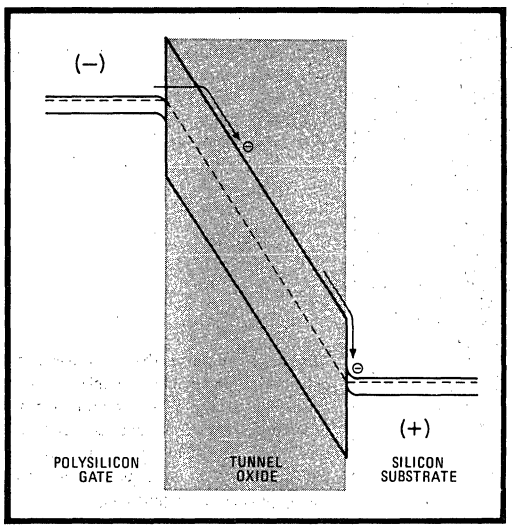
10^{-21} A or less must be guaranteed during read or storage operations. The ratio of these currents, $1:10^{11}$, represents a difficult design problem. Few charge-injecting mechanisms are known that can be turned off reliably during nonprogram periods for such a ratio.

One structure that has proven capable of meeting such stringent reliability requirements has done so for many millions of devices over the last nine years. This is the floating-gate avalanche-injection MOS (Famos) device used in the 1702, 2708, 2716, and 2732 E-PROM families. In the Famos structure, shown in Fig. 1a, a polysilicon gate is completely surrounded by silicon dioxide, one of the best insulators around. This ensures the low leakage and long-term data retention.

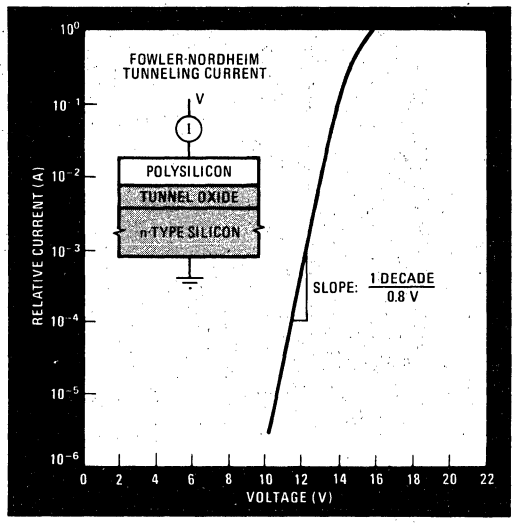
To charge the floating gate, electrons in the underlying MOS device are excited by high electric fields in the channel, enabling them to jump the silicon/silicon-dioxide energy barrier between the substrate and the thin gate dielectric. Once they penetrate the gate oxide, the electrons flow easily toward the floating gate as it was previously capacitively coupled with a positive bias to attract them.

Because of Famos' proven reliability, the floating-gate approach was favored for the EE-PROM. The problem, of course, was to find a way to discharge the floating gate electrically. In an E-PROM, this discharge is effected by exposing the device to ultraviolet light. Electrons absorb photons from the UV radiation and gain enough energy to jump the silicon/silicon-dioxide energy barrier in the reverse direction as they return to the substrate. This suffices for off-board program rewriting, but the object of the EE-PROM is to satisfy new applications that demand numerous alterations of the stored data without removing the memory from its system environment. What evolved was the new cell structure called Flotox (Fig. 1b).

In the quest for electrical erasability, many methods were considered, and several potentially viable solutions were pursued experimentally. One initially attractive



2. Tunneling. For a thin enough oxide, as shown here, under a field strength of 10^7 V/cm, Fowler-Nordheim tunneling predicts that a certain number of electrons will acquire enough energy to jump the forbidden gap and make it from the gate to the substrate.



3. Current characteristic. In Fowler-Nordheim tunneling, current flow depends strongly on voltage across the oxide, rising an order of magnitude for every 0.8 V. Charge retention is adequate so long as the difference between programming and reading is at least 8.8 V.

approach attempts to harness a parasitic charge-loss mechanism discovered in the earliest E-PROMs. Referring again to Fig. 1a, the polysilicon grains on the top surface of the floating gate tend, under certain processing conditions, to form sharp points called asperities. The sharpness of the asperities creates a very high local electric field between the polysilicon layers, shoving electrons from the floating gate toward the second level of polysilicon. This effect is purposely subdued in today's E-PROMs by controlling oxide growth on top of the floating gate because this parasitic electron-injection mechanism would otherwise interfere with proper E-PROM programming.

It was first thought that asperity injection could be used to erase the chip. In fact, fully functional, electrically erasable test devices were produced; but the phenomenon proved unreproducible and the devices tended to wear out quickly after repeated program and erase cycling. After over a year's effort, that approach was abandoned.

Tunneling solution

The solution turned out to be the one that initially seemed impossible. After investigating many methods of producing energetic electrons, it was decided to approach the problem from a different direction: to pass low-energy electrons through the oxide. This could be accomplished through Fowler-Nordheim tunneling, a well-known mechanism, depicted by the band diagram in Fig. 2. Basically, when the electric field applied across an insulator exceeds approximately 10^7 volts per centimeter, electrons from the negative electrode (the polysilicon in Fig. 2) can pass a short distance through the forbidden gap of the insulator and enter the conduction band. Upon their arrival there, the electrons flow freely toward the positive electrode.

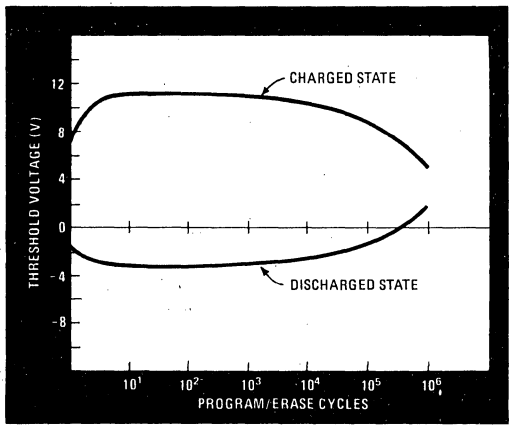
This posed two fundamental problems. First, it was commonly believed that silicon dioxide breaks down catastrophically at about 10^7 V/cm, and MOS FETs are normally operated at field strengths 10 times below this. Second, to induce Fowler-Nordheim tunneling at reasonable voltages (20 V), the oxide must be less than 200 angstroms thick. Oxide thickness below about 500 Å had rarely even been attempted experimentally, and it was feared that defect densities might prove prohibitively high.

To be weighed against these risks, however, were several advantages. Tunneling in general is a low-energy, efficient process that eliminates power dissipation. Fowler-Nordheim tunneling in particular is bilateral and can be used for charging the gate as well as discharging it. Finally, the tunnel oxide area could be made very small, which is of course consistent with the needs of high-density processing.

With these motivating factors, development was initiated to grow reliable, low-defect oxides less than 200 Å thick. The success of this effort resulted in the realization of a working cell structure called Flotox.

The Flotox device cross section is pictured in Fig. 1b. It resembles the Famos structure except for the additional tunnel-oxide region over the drain. With a voltage V_g applied to the top gate and with the drain voltage V_d at 0 V, the floating gate is capacitively coupled to a positive potential. Electrons are attracted through the tunnel oxide to charge the floating gate. On the other hand, applying a positive potential to the drain and grounding the gate reverses the process to discharge the floating gate.

Flotox, then, provides a simple, reproducible means for both programming and erasing a memory cell. But



4. Good endurance. The endurance of the EE-PROM depends on the threshold-voltage difference between the charged and discharged states. Though repeated cycling degrades thresholds, the chip should stay within tolerable limits for 10^4 to 10^6 cycles.

what about charge retention and refresh considerations with such a thin oxide? The key to avoiding such problems is given in Fig. 3, which shows the exceedingly strong dependence of the tunnel current on the voltage across the oxide. This is characteristic of Fowler-Nordheim tunneling.

The current in Fig. 3 rises one order of magnitude for every 0.8-v change in applied voltage. If the 11 orders of magnitude requirement is recalled, it is apparent that the difference between the voltage across the tunnel oxide during programming and that during read or storage operations must be in excess of 8.8 v.

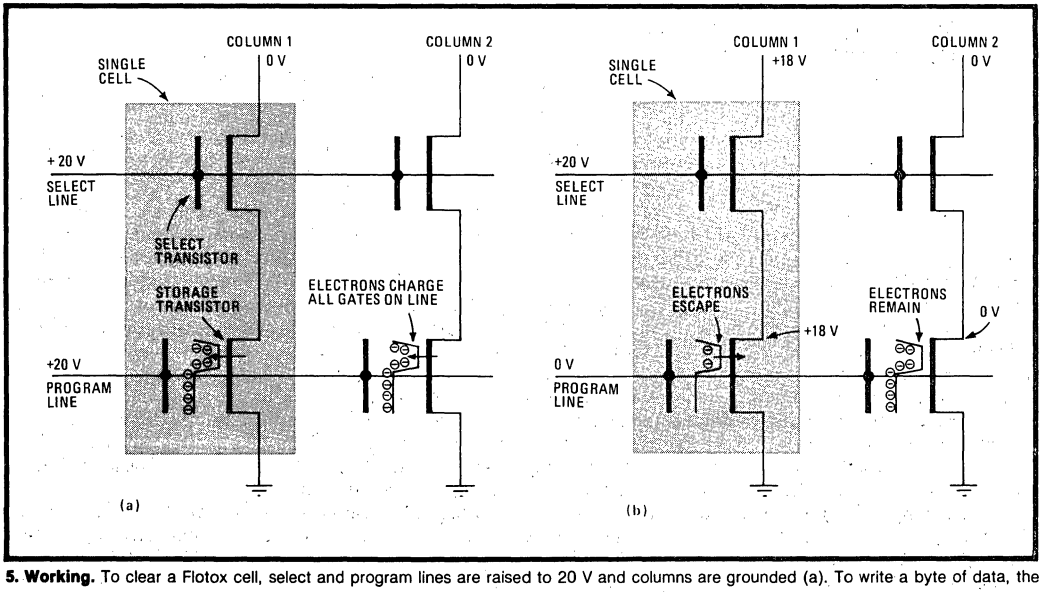
This value, including margins for processing variations, is reasonable. Furthermore, data is not disrupted during reading or storage so that no refreshing is required under normal operating or storage conditions. Extensive experimental testing has verified that data retention exceeding 10 years at a temperature of 125°C is possible.

Another important consideration is the behavior of the electrically erasable memory cell under repeated program erase cycling. This is commonly referred to as endurance. The threshold voltage of a typical Flotox cell, in both the charged and discharged states, is shown in Fig. 4 as a function of the number of programming or erasing cycles. There is some variation in the threshold voltages with repeated cycling but this remains within tolerable limits out to very high numbers of cycles—somewhere between 10^4 and 10^6 cycles.

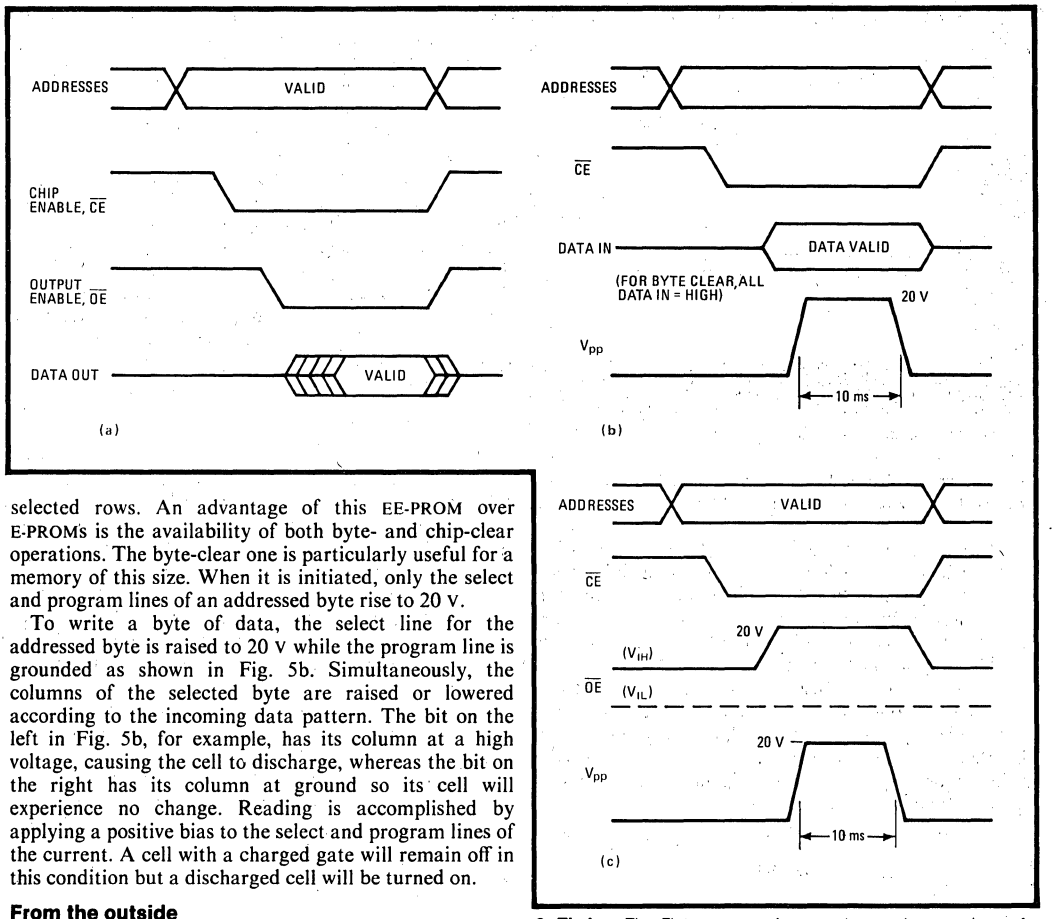
Putting Flotox to work

The Flotox cell is assembled into a memory array using two transistors per cell as shown in Fig. 5. The Flotox device is the actual storage device, whereas the upper device, called the select transistor, serves two purposes. First, when discharged, the Flotox device exhibits a negative threshold. Without the select transistor, this could result in sneak paths for current flow through nonselected memory cells. Secondly, the select transistor prevents Flotox devices on nonselected rows from discharging when a column is raised high.

The array must be cleared before information is entered. This returns all cells to a charged state as shown schematically in Fig. 5a. To clear the memory all the select lines and program lines are raised to 20 V while all the columns are grounded. This forces electrons through the tunnel oxide to charge the floating gates on all of the



5. Working. To clear a Flotox cell, select and program lines are raised to 20 V and columns are grounded (a). To write a byte of data, the program line is grounded and the columns of the selected byte are raised or lowered according to the data pattern (b).



6. Timing. The Flotox memory's operating modes are shown for reading (a), writing or clearing of bytes (b), and chip clearing (c). Both writing and erasing require a 10-ms program-voltage pulse. The read mode is functionally identical to that of a 2716 E-PROM.

selected rows. An advantage of this EE-PROM over E-PROMs is the availability of both byte- and chip-clear operations. The byte-clear one is particularly useful for a memory of this size. When it is initiated, only the select and program lines of an addressed byte rise to 20 V.

To write a byte of data, the select line for the addressed byte is raised to 20 V while the program line is grounded as shown in Fig. 5b. Simultaneously, the columns of the selected byte are raised or lowered according to the incoming data pattern. The bit on the left in Fig. 5b, for example, has its column at a high voltage, causing the cell to discharge, whereas the bit on the right has its column at ground so its cell will experience no change. Reading is accomplished by applying a positive bias to the select and program lines of the current. A cell with a charged gate will remain off in this condition but a discharged cell will be turned on.

From the outside

In terms of its pinout and control functions, the EE-PROM has evolved from the 2716 E-PROM. Both are housed in 24-pin dual in-line packages, for instance, and both offer a power-down standby mode. In addition, both utilize the same powerful two-line control architecture for optimal compatibility with high-performance microprocessor systems. Referring to Fig. 6a, it is seen that both control lines, chip enable ($\overline{CE}$) and output enable ($\overline{OE}$), are taken low to initiate a read operation. The purpose of chip enable is to bring the memory out of standby to prepare it for addressing and sensing. Until the output-enable pin is brought low, however, the outputs remain in the high-impedance state to avoid system bus contention. In its read mode, the EE-PROM is functionally identical to the 2716.

A single +5-V supply is all that is needed for carrying out a read. For the clear and write functions, an additional supply (V_{PP}) of 20 V is necessary. The timing for writing a byte is shown in Fig. 6b. The chip is powered up by bringing $\overline{CE}$ low. With address and data applied, the write operation is initiated with a single 10-ms, 20-V pulse applied to the V_{PP} pin. During the write operation, $\overline{OE}$ is not needed and is held high.

A byte clear is really no more than a write operation. As indicated in Fig. 6b, a byte is cleared merely by being written with all 1s (high). Thus altering a byte requires nothing more than two writes to the addressed byte, first with the data set to all 1s and then with the desired data. This alteration of a single byte takes only 20 ms. In other nonvolatile memories, changing a single byte requires that the entire contents be read out into an auxiliary memory. Then the entire memory is rewritten. This process not only requires auxiliary memory; for a 2-kilobyte device it takes about one thousand times as long (20 ms vs 20 seconds).

Chip clear timing is shown in Fig. 6c. The only difference between byte clear and chip clear is that $\overline{OE}$ is raised to 20 V during chip clear. The entire 2 kilobytes are cleared with a single 10-ms pulse. Addresses and data are not all involved in a chip-clear operation. □

2004
4K (512 x 8) NON-VOLATILE RANDOM ACCESS MEMORY

- **5 Volt Only Operation**
- **Fast Static RAM Read/Write Cycles**
  2004-2, 200ns Max.
  2004, 250ns Max.
  2004-3, 300ns Max.
- **Single Line STORE & RECALL**
- **10ms Self-Timed STORE Cycles for 2004-2 and 2004 (20ms for 2004-3)**
- **Automatic Recall on Power Up**
- **Write Protect Circuit to Preserve Data On Power-Up and Power-Down**
- **Lower Power Standby Mode**
- **10-Year Data Retention for each STORE**
- **Minimum 10,000 Non-Volatile STORE Cycle Endurance**
- **Unlimited Endurance for Read, Write, and RECALL Cycles**
- **HMOS*-E FLOTOX Cell Design**
- **Conforms to JEDEC Byte-Wide Universal Site**

The Intel 2004 Non-Volatile Random Access Memory (NVRAM) is a 4K device with 512 × 8 architecture. It provides the real-time read/write functions of a static RAM together with the reliable non-volatile storage capability of an E²PROM array to preserve its memory contents when power is removed.

Internally, the 2004 NVRAM consists of a high speed static RAM array backed up, bit-for-bit, by an E²PROM array for non-volatile storage. The transfer of memory data between the static RAM and the E²PROM array occurs in parallel for fast storage and recall as well as minimal system support.

Two functions are provided to transfer data between the volatile RAM and its non-volatile E²PROM counterpart. The STORE function transfers RAM data into the E²PROM while the RECALL function fetches E²PROM data and places it in the RAM array. Both functions are controlled by a single $\overline{NE}$ signal which can easily be activated with traditional circuitry in memory mapped space, through an I/O port, or from the output of a power-fail detector.

The RAM operating characteristics of the 2004 NVRAM provides high speed microprocessor performance with unlimited endurance. In the non-volatile storage mode, data retention is specified at over 10 years for each STORE operation. Over 10,000 STORE operations can be performed reliably.

The 2004 NVRAM is furnished in a 28-pin byte wide package with its address, data and control lines configured according to the standard JEDEC universal 28-pin site.

*HMOS is patent process of Intel Corporation.

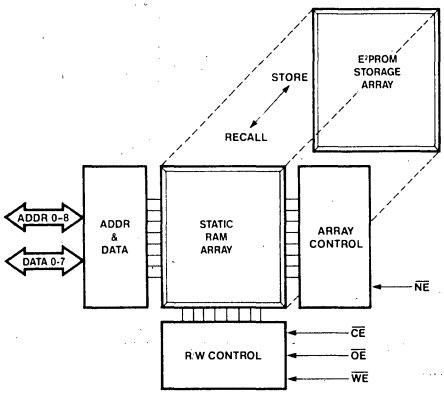


Figure 1. 2004 Functional Diagram

2004 PIN NAMES

A_0-A_8	ADDRESSES
I/O	DATA INPUT/OUTPUTS
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE (READ)
$\overline{NE}$	NON-VOLATILE ENABLE (STORE/RECALL CONTROL)
$\overline{WE}$	WRITE ENABLE
N. C.	NO CONNECT

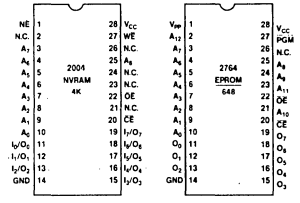


Figure 2. 2004 Pin Configuration

OCTOBER 1983
ORDER NUMBER: 230665-004

NVRAM APPLICATIONS

The non-volatile RAM is designed for operation in systems where important variable data needs to be retained during periods when power is not applied to the system. It combines the flexibility of a static RAM with the reliable non-volatility of E²PROM.

The NVRAM provides a more flexible alternative than the E²PROM. E²PROMs are best suited for non-volatile storage of program code or system parameters which are occasionally altered. The NVRAM is designed for fast non-volatile storage of multiple data bytes which are in transit in the system or are being altered at microprocessor speeds.

Both NVRAMs and E²PROMs provide important, but separate, functions in the same system. An example is in communication equipment. The NVRAM is used for buffer storage for data being received or transmitted over a communications link. If power fails, a single microprocessor instruction causes the NVRAM to store all the buffered data. The E²PROM array in this system is used to store the microprocessor program code as well as look-up tables for the various operating parameters. The E²PROM program code can be updated via the communications link, and the operating parameters can be modified either remotely over the link or directly from the keyboard.

Non-volatile RAMs offer a silicon alternative to memory designs using battery-back CMOS RAM. The simple operation, single chip solution, and superior reliability of the NVRAM is a preferable alternative to the chemical battery.

The 2004 non-volatile RAM features 2-line control. By requiring a chip enable ($\overline{CE} = 0$) whenever a device function is to be activated (determined by $\overline{OE}$, $\overline{WE}$, and $\overline{NE}$), data bus contention is eliminated, system noise is reduced, and system design is simplified.

SYSTEM CONSIDERATIONS

Figure 3 illustrates a typical hardware system's block diagram. A power fail detection circuit is used to notify the system CPU of the loss of AC power via an interrupt line. The microprocessor decodes the interrupt, enters a servicing routine which writes important system data into the NVRAM, then performs a STORE initiation cycle. The STORE operation is completed in 10ms during which the power supply has held V_{CC} at 5V. As power is finally lost in the system, an on-chip V_{CC} STORE protection circuit prevents any subsequent unwanted STORE operations from being started due to system power-down noise. Data is retained in the non-volatile storage array in the NVRAM.

Figure 4 shows how to generate the $\overline{NE}$ signal with an I/O port bit for an array of 2004s.

It is a good design practice to decouple the power supply pin on any memory device. Therefore, it is recommended that a 0.1μf ceramic capacitor be used on every device between V_{CC} and GND.

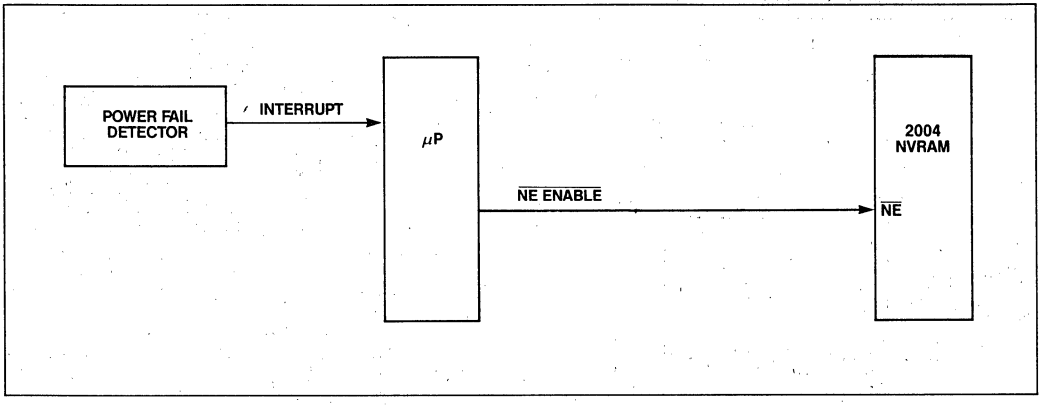


Figure 3. NVRAM System Block Diagram

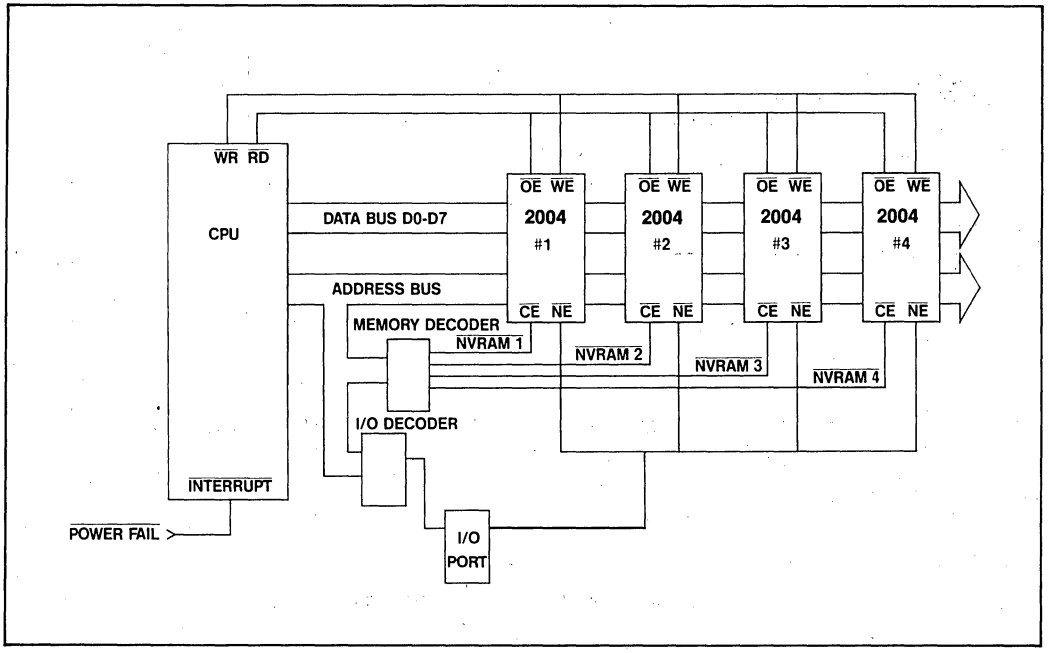


Figure 4. 2004 Array: Generating $\overline{NE}$ by I/O Output Port Bit

OPERATIONAL OVERVIEW

The non-volatile RAM uses standard control lines in accordance with established microprocessor interfaces and pinout standards. An additional control pin, non-volatile enable ($\overline{NE}$), is used to control the STORE and RECALL functions to the non-volatile storage array. For RAM access cycles the $\overline{NE}$ input is held high on the NVRAM. Addressing of each of the individual 512 RAM array bytes is accomplished using address lines 0 through 8 (A0-A8) and enabling the chip ($\overline{CE}$). Data is then transferred a byte (8 bits) at a time to and from the RAM array in the NVRAM. Data is written by lowering Write Enable ($\overline{WE}$) and holding Output Enable ($\overline{OE}$) at a TTL high level. Data is read by lowering the $\overline{OE}$ input while holding the $\overline{WE}$ input high. These functions are generally accomplished using a hardware design as shown in Figure 5.

The non-volatile functions of the NVRAM are implemented by putting a TTL low on the $\overline{NE}$ pin. A RECALL operation is initiated by bringing $\overline{OE}$ low while $\overline{NE}$ = 0. This causes the data from the NVRAM's internal non-volatile storage array to be transferred to the static RAM array, from which it can be externally accessed. The RECALL function can be activated by the CPU at any time, as often as desired without affecting the integrity of the data in the non-volatile storage array. The RECALL function is also automatically performed when the NVRAM is powered up.

A STORE operation is started by bringing $\overline{WE}$ low while holding $\overline{NE}$ low. This causes the data in the static RAM array to be transferred to the non-volatile storage array. The STORE function is typically used to save data during power failure periods and is executed

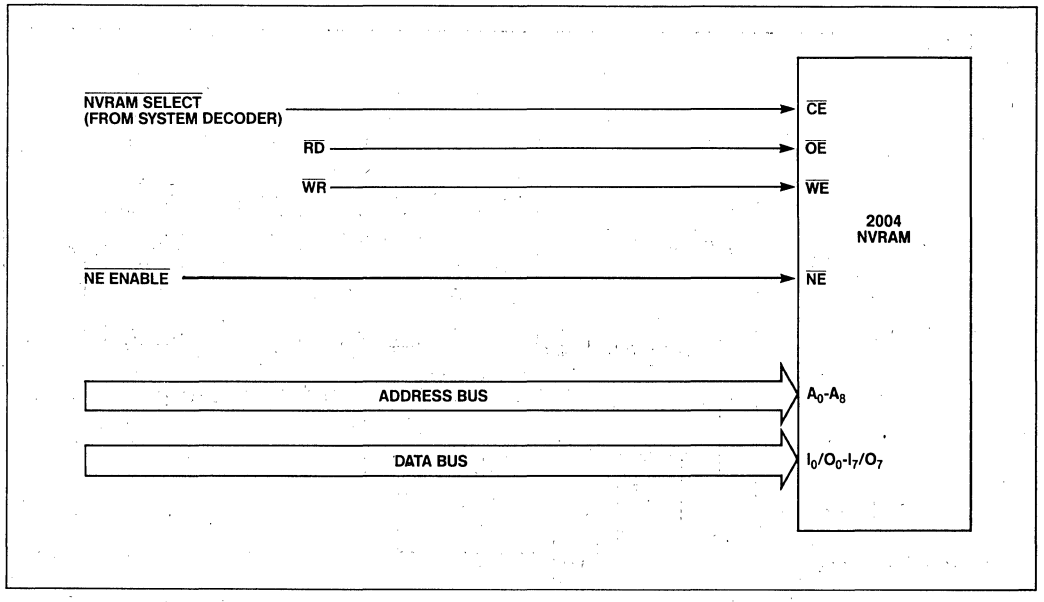


Figure 5. NVRAM System Interface

when a power fail signal is received by the system from the power supply. Since the STORE cycle takes 10ms for reliable storage, the NVRAM's Vcc supply must be maintained for this duration.

All operations of the NVRAM are gated by the chip enable ($\overline{CE}$) input. When $\overline{CE}$ is held at a TTL high level, the NVRAM is in the standby mode and consumes almost 50% less power. In this mode the data outputs are in a high impedance state. A summary of operational modes is listed in Table 1.

INADVERTENT STORE PROTECTION DURING POWER FAIL

When system power is falling the TTL devices which generate the $\overline{CE}$, $\overline{OE}$, $\overline{WE}$ and $\overline{NE}$ signals will no longer be stable. There is a possibility that the noise on these lines could initiate a STORE operation as V_{CC} is falling. An on-chip circuit is needed to prevent an inadvertent STORE operation. The 2004 has an on-chip inadvertent write protection circuit. When V_{CC} is between 4.0 and 4.4V the device's write mode is disabled. The lockout voltage on all products will fall in this range. Individual units will lockout at one specific voltage.

STORE AND RECALL CONSIDERATIONS

Data retention for data that has been written into the 2004's non-volatile array by a STORE operation is greater than 10 years. The STORE endurance is a minimum of 10^4 cycles, that is, at least 10,000 STORE operations can be reliably performed.

Table 1. 2004 Operational Modes Vcc = 5V

	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	$\overline{NE}$	Outputs
Standby	V_{IH}	X	X	X	Hi–Z
READ	V_{IL}	V_{IL}	V_{IH}	V_{IH}	Data Out
WRITE	V_{IL}	X	V_{IL}	V_{IH}	Data In
RECALL–Power Up	X	X	X	V_{IH}	Hi–Z
RECALL–Standard	V_{IL}	V_{IL}	V_{IH}	V_{IL}	Hi–Z
STORE	V_{IL}	V_{IH}	V_{IL}	V_{IL}	Hi–Z

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . . -10°C to +80°C
Storage Temperature . . . . . -65°C to +100C
All Input or Output Voltages with
Respect to Ground . . . . . +6V to -0.3V

*NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS

Operating Conditions: $T_A = 0°C$ to 70°C
VCC = 5 VDC ± 5%

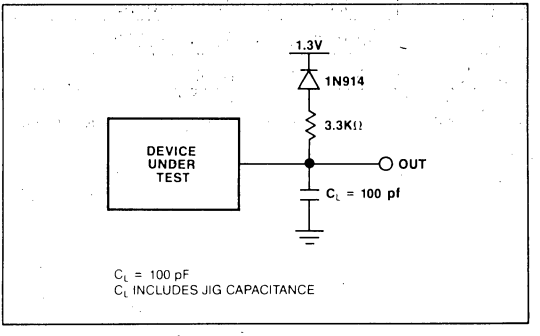
Symbol	Parameter	Min	Max	Units	Conditions
I_{LI}	Input leakage current		10	μA	VCC = max, VIN = GND to VCC
I_{LO}	Output leakage current		10	μA	VCC = max, $\overline{CE}$ = VIH, VOUT = GND to VCC
I_{CC1}	VCC current (Standby)		55	mA	VCC = max, $\overline{CE}$ = VIH
I_{CC2}	VCC current (Active)		100	mA	VCC = max, Mode = READ or WRITE
I_{CC3}	VCC current (STORE)		100	mA	VCC = max, Mode = STORE
I_{CC4}	VCC current (RECALL)		100	mA	VCC = max, Mode = RECALL
V_{IL}	Input low voltage	-0.1[1]	0.8	V	
V_{IH}	Input high voltage	2.0	VCC + 1	V	
V_{OL}	Output low voltage		0.4	V	I_{OL} = 2.1ma
V_{OH}	Output high voltage	2.4		V	I_{OH} = -400μa
V_{RCL}	V_{cc} level at which automatic RECALL begins during Power-Up	4.0	4.4	V	

Note 1. -1.0V spikes less than 20ns in duration are allowed

A.C. TEST CONDITIONS

Input pulse levels: 0.45V and 2.4V
Input rise and fall times: 20 nsec (10% and 90% levels)
Output timing reference levels: 0.8V and 2.0V

A.C. TESTING LOAD CIRCUIT



A.C. CHARACTERISTICS

READ CYCLE

Symbol	Parameter	2004-2		2004		2004-3		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{RC}	Read Cycle Time	200		250		300		ns
t_{ACC}	Address Access Time		200		250		300	ns
t_{CE}	Chip Select Access		200		250		300	ns
t_{OE}	$\overline{OE}$ Access Time		70		100		150	ns
t_{CLZ}	$\overline{CE}$ Selection to Active Output	10		10		10		ns
t_{CHZ}	$\overline{CE}$ Deselection to Output Not Driven		60		60		130	ns
t_{OLZ}	$\overline{OE}$ Selection to Active Output	10		10		10		ns
t_{OHZ}	$\overline{OE}$ Deselection to Output Not Driven		60		60		130	ns
t_{PU}	$\overline{CE}$ Selection to Power Up Time[1]	10		10		10		ns
t_{PD}	$\overline{CE}$ Deselection to Power Down Time[1]		90		120		160	ns
t_{OH}	Output Held from Addresses, $\overline{CE}$, or $\overline{OE}$, (whichever occurs first)	0		0		0		ns

WRITE CYCLE

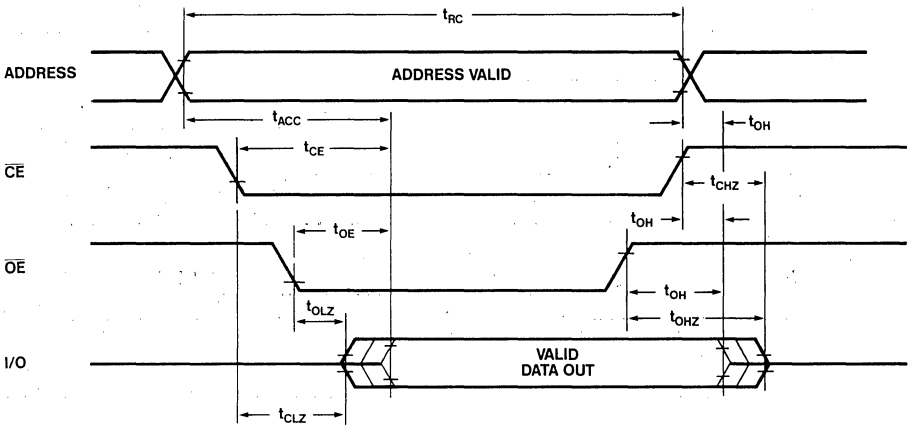
Symbol	Parameter	2004-2		2004		2004-3		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{WC}	Write Cycle Time	200		250		300		ns
t_{CW}	Chip Selection to End of Write	200		250		300		ns
t_{AW}	Address Valid to End of Write	200		250		300		ns
t_{AS}	Address Setup Time	0		0		0		ns
t_{WP}	Write Pulse Width	120		150		200		ns
t_{DW}	Data Valid to End of Write	120		150		200		ns
t_{DH}	Data Hold Time	0		0		0		ns
t_{WLZ}	$\overline{WE}$ Disabled to Active Output	10		10		10		ns
t_{WHZ}	$\overline{WE}$ Enabled to Output Not Driven		60		90		130	ns
t_{WR}	Write Recovery Time	0		0		0		ns

Note 1. This refers to the powering up and down of the device's internal circuitry.

WAVEFORMS

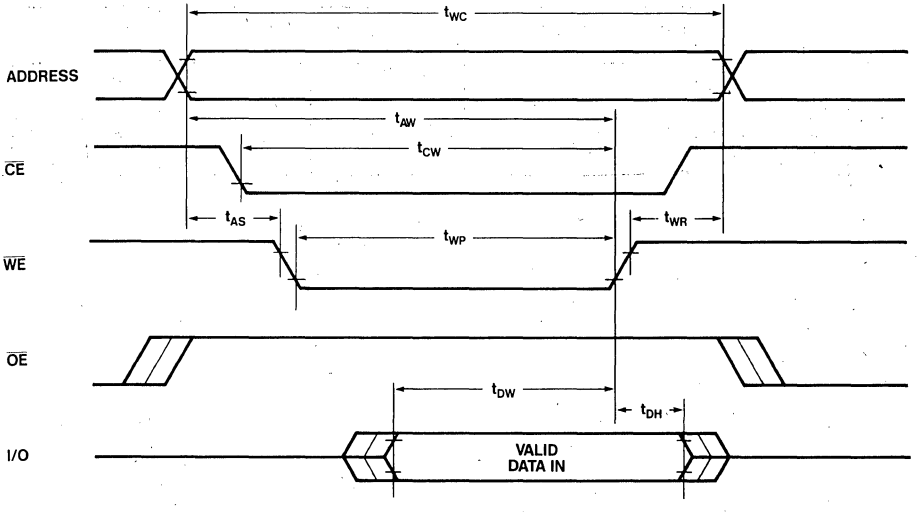
READ

CONDITIONS: $\overline{NE} = V_{IH}$, $\overline{WE} = V_{IH}$



WRITE

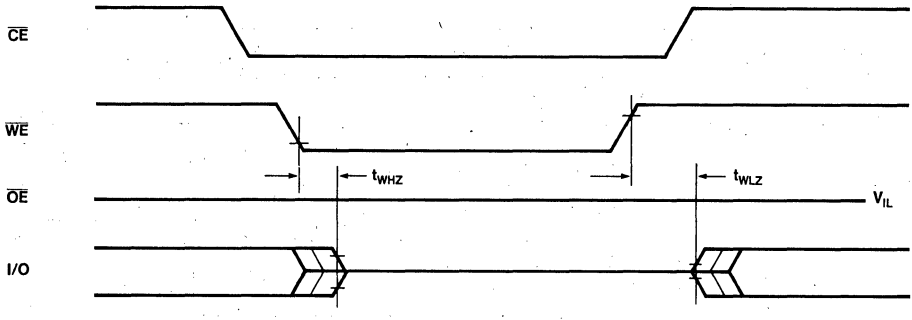
FOR SYSTEMS WITH 2-LINE CONTROL (INTEL MICROPROCESSORS)
CONDITION: $\overline{NE} = V_{IH}$



WAVEFORMS

WRITE

WITH SINGLE LINE CONTROL
CONDITION: $\overline{NE} = V_{IH}$ (OUTPUT ENABLE AND DISABLE TIMES SHOWN ONLY)

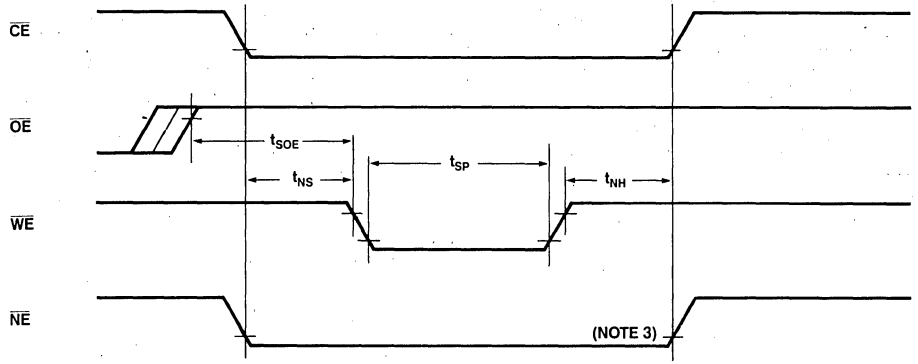


STORE Operation[1, 2]

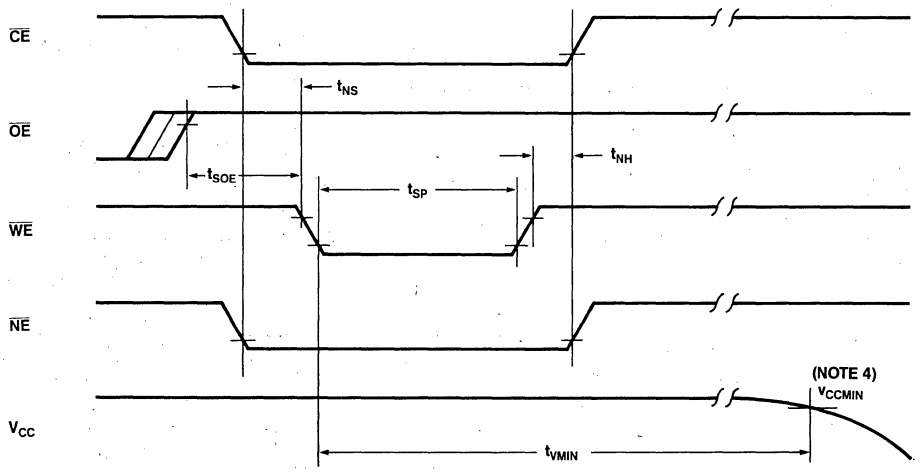
Symbol	Parameter	2004-2		2004		2004-3		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{SOE}	$\overline{OE}$ Disable to STORE Function	200		200		200		ns
t_{SP}	STORE Pulse Width	120		150		200		ns
t_{STR}	STORE Cycle Time		10		10		20	ms
t_{VMIN}	Vcc Above Vccmin after STORE Operation Initiated	10		10		20		ms
t_{NS}	Setup Time to $\overline{WE}$ for STORE Initiation Cycle	0		0		0		ns
t_{NH}	Hold Time after $\overline{WE}$ for STORE Initiation Cycle	0		0		0		ns

WAVEFORMS (continued)

STORE INITIATION CYCLE UNDER NORMAL CONDITIONS



STORE INITIATION CYCLE DURING POWER FAIL



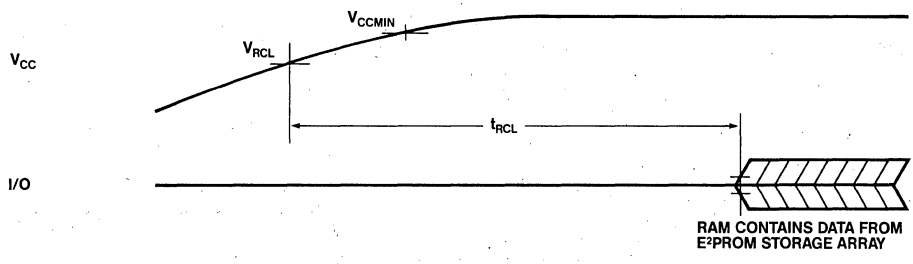
NOTES:

1. During a STORE initiation cycle, addresses and I/O inputs = DON'T CARE.
2. For a STORE cycle, $\overline{WE}$ must not go low before $\overline{NE}$. Otherwise, a RAM write cycle will result.
3. A lockout feature on the $\overline{NE}$ input prevents subsequent STORE cycles from occurring until $\overline{NE}$ is brought back high. The $\overline{NE}$ input should therefore be brought back high after a non power-fail STORE operation is initiated to allow normal read/write access after completion of the STORE operation.
4. Vccmin = 4.75V for 5% Vcc spec
5. Once a STORE operation has begun, all inputs are ignored and the outputs are in a high impedance state.

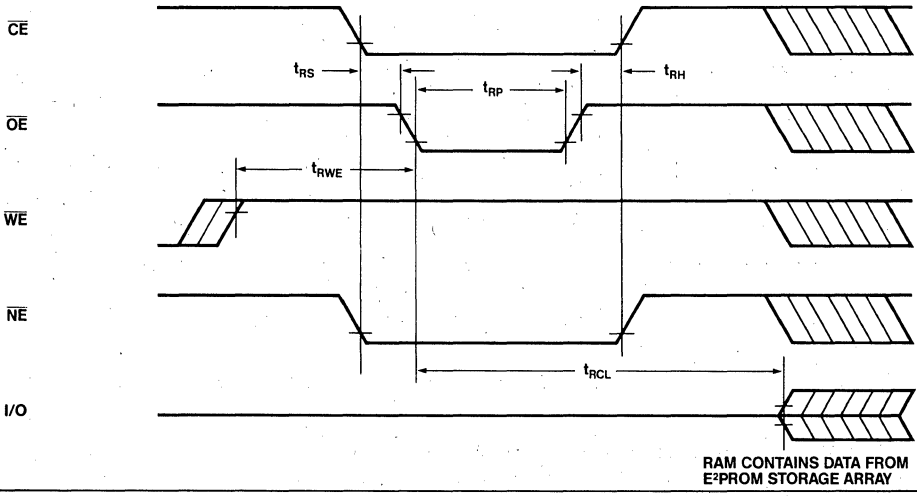
RECALL Operation[1]

Symbol	Parameter	2004-2		2004		2004-3		Units
		Min	Max	Min	Max	Min	Max	
t_{RWE}	$\overline{WE}$ Disable to RECALL Command	200		200		200		ns
t_{RP}	RECALL Pulse Width	120		150		200		ns
t_{RCL}	RECALL Cycle Time		10		10		10	μs
t_{RS}	Setup Time to RECALL Command	0		0		0		ns
t_{RH}	Hold Time after RECALL Command	0		0		0		ns

AUTOMATIC RECALL DURING POWER UP[2]



RECALL INITIATION CYCLE UNDER NORMAL CONDITIONS (NOTE 3)



NOTES:

1. During a RECALL Initiation Cycle, the address and data inputs = DON'T CARE.
2. During Automatic Power-Up RECALL, $\overline{CE} = \overline{WE} = \overline{OE} = \overline{NE}$ = DON'T CARE.
3. Once a RECALL operation has begun, all inputs are ignored and the outputs are in a high impedance state.
4. Vccmin = 4.75V for 5% Vcc spec

PRELIMINARY

2816A
16K (2K x 8) ELECTRICALLY ERASABLE PROM

- 5 Volt Only Operation
- Fast Read Access Time:
  - —2816A-2, 200ns Max
  - —2816A, 250ns Max
  - —2816A-3, 350ns Max
  - —2816A-4, 450ns Max
- Byte Erase/Write with TTL Level $\overline{WE}$ Signal
- 9ms Byte Erase/Write Time
- 9ms Chip Erase Time
- HMOS*-E Flotox Cell Design
- Minimum Endurance of 10,000 Erase/Write Cycles per Byte
- Unlimited Number of Read Cycles
- Conforms to JEDEC Universal Site
- Erase/Write Specifications Guaranteed 0-70°C
- Write Protect Circuit to Preserve Data on Power Up and Power Down

The Intel 2816A is a 16,384-bit electrically erasable programmable read-only memory (E^2PROM). The 2816A can be easily erased and reprogrammed on a byte basis with a TTL-low level signal on $\overline{WE}$. The 2816A operates from a single 5 Volt supply. External programming voltage and write pulse shaping are not required because they are generated by on-chip circuitry.

The Intel 2816A is compatible with the Intel 2816 E^2PROM. Dual voltage detection logic allows the 2816A to use an existing, externally supplied high voltage programming pulse required with the 2816 to write to the Intel 2816A. No hardware changes are required when substituting a 2816A in an existing 2816 socket. System upgrades to 5 volt only operation can be implemented, however, by removing the 21V and write shaping circuitry. The 2816A, like the 2816, has fast read access speeds allowing zero wait state read cycles with high performance microprocessors such as the iAPX286.

The electrical erase/write capability of the 2816A makes it ideal for a wide variety of applications requiring in-system, non-volatile erase and write. Any byte can be erased in 9ms without affecting the data in any other byte. Alternatively, the entire memory can be erased in 9ms allowing the total time to rewrite all 2k bytes to be cut by 50%. The 2816A is part of the Intel E^2PROM family that provides a significant increase in flexibility allowing new applications (remote firmware update of program code, dynamic parameter storage) never before possible.

The 2816 E^2PROM possesses Intel's 2-line control architecture to eliminate bus contention in a system environment. A power down mode is also featured; in standby mode, power consumption is reduced by over 50% without increasing access time. The standby mode is achieved by applying a TTL-high signal to the $\overline{CE}$ input.

*HMOS is a patented process of Intel Corporation

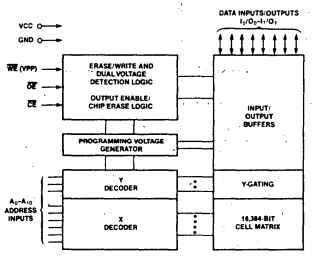


Figure 1. 2816A Functional Block Diagram

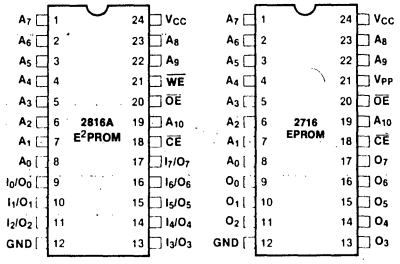


Figure 2. Pin Configuration

PIN NAMES	
A_0-A_{10}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	DATA OUTPUTS
I_0-I_7	DATA INPUTS
$\overline{WE}$ (VPP)	WRITE ENABLE (VPP OPTION)

OCTOBER 1983
ORDER NUMBER: 210823-004

DEVICE OPERATION

The 2816A has six modes of operation, listed in Table 1. All operational modes are designed to provide maximum microprocessor compatibility and system consistency.

Table 1. Mode Selection

MODE \ PIN	$\overline{CE}$ (18)	$\overline{OE}$ (20)	$\overline{WE}$ (21)	INPUTS/ OUTPUTS
Read	V_{IL}	V_{IL}	V_{IH}	D_{OUT}
Standby	V_{IH}	Don't Care	Don't Care	High Z
Byte Erase	V_{IL}	V_{IH}	V_{IL}	$D_{IN}=V_{IH}$
Byte Write	V_{IL}	V_{IH}	V_{IL}	D_{IN}
Chip Erase	V_{IL}	+10V to +15V	V_{IL}	$D_{IN}=V_{IH}$
No Operation	V_{IL}	V_{IH}	V_{IH}	High Z
E/W Inhibit	V_{IH}	V_{IH}	V_{IL}	High Z

All control inputs are TTL-compatible with the exception of chip erase.

READ MODE

Optimal system efficiency depends to a great extent on a tightly coupled microprocessor/memory interface. The E^2PROM device should respond rapidly with data to allow the highest possible CPU performance. The 2816A satisfies this high performance requirement because of read access times typically less than 250ns. Program execution directly out of electrically erasable memory has never before been possible; the 2816A opens this new, powerful applications segment.

The 2816A uses Intel's proven 2-line control architecture for read operation. The 2-line control function removes bus contention from the system environment and allows low power dissipation (by deselecting unused devices).

Figure 3 shows the timing disadvantages of a single-line control architecture. 2-line control, shown in Figure 4, has been developed by Intel to solve this bus contention and the associated system reliability problems. Both $\overline{CE}$ and $\overline{OE}$ must be at logic low levels to obtain information from the device. Chip enable ($\overline{CE}$) is the power control pin and should be used for device selection. The output enable ($\overline{OE}$) pin serves to gate internal data to the output pins. Assuming that the address inputs are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a time delay of t_{OE}, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Figure 5 shows a typical system interconnection. Here the 2816A contains program information that the 8086 requires for system function. $\overline{CE}$ (pin 18) is decoded from addresses as the primary device selection function. $\overline{OE}$ (pin 20) should be made a common connection to all devices in system, and connected

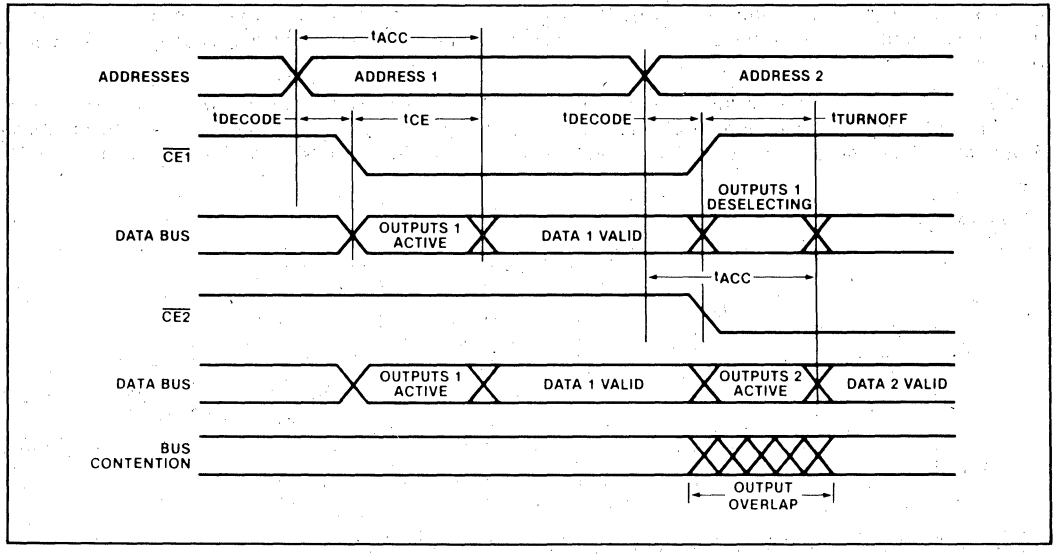


Figure 3. Single-Line Control and Bus Contention

AFN-00865A

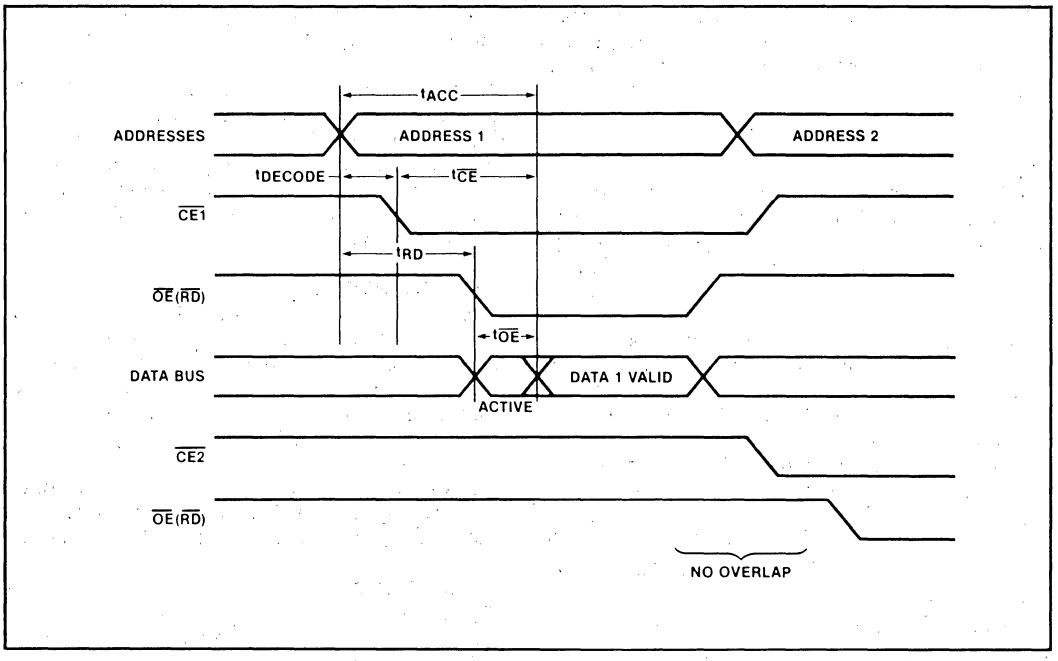


Figure 4. Two-Line Control Architecture

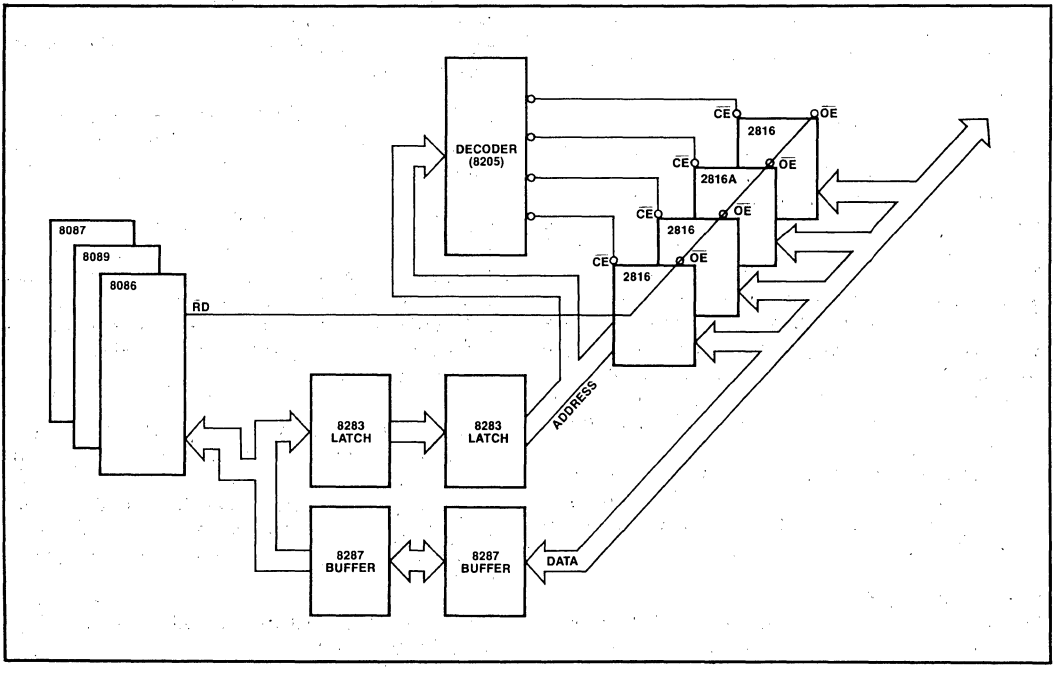


Figure 5. iAPX 86/2816A Read Architecture

AFN-00865A

to the $\overline{RD}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are only active when data is desired from a particular memory device.

WRITE MODE

The 2816A is erased and reprogrammed electrically as opposed to EPROMs which require ultraviolet light for erasure. The device offers dynamic flexibility because both byte (single location) and chip erase are possible.

A close examination of the broad application spectrum for the E^2 device reveals an inherent need for single location erase capability. Program store applications can be classified in several ways. Figure 6 lists various storage modes and the required erase function. In greater than 80% of all cases, a byte erase feature is necessary.

APPLICATION TYPE	IDEAL ERASE MODE
• STRICT PROGRAM STORE	CHIP
• RELOCATABLE PROGRAM STRUCTURE	BYTE
• PROGRAM STORE EXTENSION	BYTE
• PROGRAM EXECUTION CONSTANTS	BYTE
• PROGRAM DEPENDENT DATA STORE	BYTE
• DATA STORE APPLICATIONS	BYTE

Figure 6. Microprocessor Storage Types

To write a particular location, that byte must be erased prior to a data write. Erasing is accomplished by placing the byte address at the address input pins, applying logic 1 (TTL-high) to all eight data input pins, and lowering $\overline{CE}$, $\overline{WE}$ to V_{IL}. The $\overline{OE}$ pin must be held at V_{IH} during byte erase and write operations. The $\overline{WE}$ pulse width must be a minimum of 9ms, and a maximum of 15ms. Once the location has been erased, the same operation is repeated for a data write. The data input pins in this case reflect the byte that is to be stored. The data to be programmed, address and control signals must be presented to the 2816A throughout the required programming time.

Because the device is designed to be written in system, all data sheet specifications (including write and erase operations) hold over the full operating temperature range (0–70°C).

CHIP ERASE MODE

Should one wish to erase the entire 2816A array at once, the device offers a chip erase function. When the chip erase function is performed, all 2K bytes are returned to a logic 1 (FF) state.

The 2816A's chip erase function is engaged when the output enable ($\overline{OE}$) pin is raised above 9 volts. When $\overline{OE}$ is greater that 9 volts and $\overline{CE}$ and $\overline{WE}$ are in the normal write mode, the entire array is erased. This chip erase function takes approximately 10ms. The data input pins must be held to a TTL-high level during this time.

STANDBY MODE

The 2816A has a standby mode which reduces active power dissipation by 50% from 100ma to 50ma. The 2816A is placed in standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in the standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ and $\overline{WE}$ input.

The standby mode for the 2816A is a superset of the standby mode for the 2816. The standby mode for the 2816A includes the E/W Inhibit mode of the 2816.

NO OPERATION MODE

This mode is frequently entered while in a read or write cycle. In the READ cycle, $\overline{CE}$ may go low before $\overline{OE}$ goes low because t_{CE} ($\overline{CE}$ to Output Delay) is longer than t_{OE} ($\overline{OE}$ to Output Delay). While $\overline{CE}$ is low with $\overline{OE}$ and $\overline{WE}$ at TTL-high, no READ, ERASE, or WRITE operation will occur. The read operation would begin in the READ cycle when $\overline{OE}$ input also falls to TTL-low.

The No Operation mode differs from Standby Mode in that active power is drawn by the 2816A in this mode.

WRITE TIME CHARACTERISTICS

The 2816A write time specification is 9ms (min.) and 15ms (max.). If the write pulse width applied to the $\overline{WE}$ input is 9ms, the programmed byte is guaranteed to be correctly and reliably programmed to any location in the 2816A.

The maximum pulse width to the $\overline{WE}$ input of 15ms limits the duration of the pulse width. Exceeding this specification may overstress the E^2PROM cells and affect long term device reliability. Any write pulse width between 9ms and 15ms will also guarantee byte programming and chip erase over the full temperature range.

Programmed data will be retained by the Intel 2816A for over 10 years.

Although the number of read cycles is unlimited, a characteristic of all E^2PROMs is that the total number of erase/write cycles is not unlimited. The 2816A has been designed and manufactured to meet applications requiring up to 1×10^4 erase/write cycles per byte. The erase/write cycling characteristic is completely byte independent. Adjacent bytes are not affected during erase/write cycling.

On-Chip Write Protection on V_{CC} Power Up and Power Down

An erase/write of a byte in the 2816A is accomplished with input signals $\overline{CE}$, $\overline{WE} = V_{IL}$. During system (Vcc) power up and power down, this condition may be present as Vcc ramps up to or down from its steady state value of 5 volts. To prevent the possibility of an inadvertent byte write during this power transition period, an on-chip sensing circuit disables the internal programming circuit if Vcc falls below 4 volts (V_{LKO}).

VPP OPTION

Although the Intel® 2816A requires only 5 volts for programming, it was designed to be totally upward compatible with the Intel 2816. No hardware changes are required when substituting a 2816A into a 2816 socket. Shaping of the Vpp programming pulse is also not required with the 2816A. Table 2 lists the Vpp Option Modes.

Table 2. VPP Option Modes

PIN / MODE	$\overline{CE}$ (18)	$\overline{OE}$ (20)	VPP (21)	INPUTS/ OUTPUTS
Byte Erase	V_{IL}	V_{IH}	20–22V	$D_{IN} = V_{IH}$
Byte Write	V_{IL}	V_{IH}	20–22V	D_{IN}
Chip Erase	V_{IL}	+10V to +15V	20–22V	$D_{IN} = V_{IH}$

APPLICATIONS

The 2816A E^2PROM is a new and powerful addition to the Intel non-volatile family. Like other Intel E^2PROMs, it offers a high degree of flexibility through in-circuit alterability while retaining the non-volatile characteristics of ROM.

Because of these device parameters, the device is ideal in many applications. Some of the potential uses are listed below:

1. Calibration constants storage (continuous calibration)
2. Software alterable central stores (dynamic reconfiguration)
3. Remote communication programming
4. PC and NC Industrial Applications
5. CRT Terminal configuration and custom graphic and font sets
6. Military replacement for core memory and fuse-link PROMs
7. Point of sale terminals
8. Remote alterable look-up tables
9. Printer communications, controllers
10. Remote data or error logging
11. Replacing CMOS RAM and battery backup
12. Remote firmware update of program code

AVAILABLE LITERATURE

To give the system designer an opportunity to more thoroughly understand the device attributes and uses, a library of E^2PROM information is available in the Memory Component Handbook. Some of the E^2PROM application information included is listed below.

AR 119 — 16-K EE-PROM Relies On Tunneling for Byte-Erasable Program Storage

AP 100 — Reliability Aspects of a Floating Gate E^2PROM

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +100°C
All Input or Output Voltages with
Respect to Ground..... +6V to −0.3V
Maximum Duration of Erase/Write Cycle ... 20ms
V_{OE} with Respect to Ground..... +17V to −0.1V

**NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. AND A.C. OPERATING CONDITIONS DURING READ AND WRITE

Temperature Range	0°C-70°C
V_{CC} Power Supply	5V ± 5%

D.C. CHARACTERISTICS

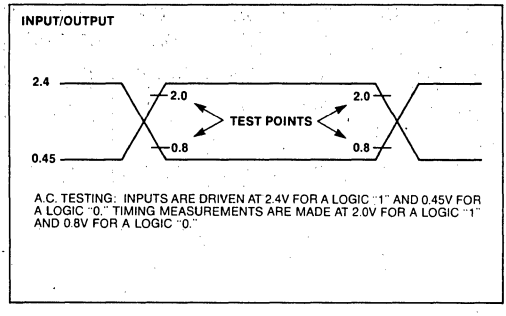
Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[1]	Max.		
I_{LI}	Input Leakage Current			10	μA	$V_{IN}=5.25V$[2]
I_{LO}	Output Leakage Current			10	μA	$V_{OUT}=5.25V$
I_{CCA}	V_{CC} Current (Active)		50	100	mA	$OE=CE=V_{IL}$
I_{CCS}	V_{CC} Current (Standby)		25	50	mA	$CE=V_{IH}$
I_{CCW}	V_{CC} Current (Byte Erase/Write)			130	mA	$WE=CE=V_{IL}$
$V_{IL(D.C.)}$	Input Low Voltage (D.C.)	−0.1		0.8	V	
$V_{IL(A.C.)}$	Input Low Voltage (A.C.)	−0.4			V	Time = 10 ns
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			0.45	V	$I_{OL}=2.1mA$
V_{OH}	Output High Voltage	2.4			V	$I_{OH}=-400\mu A$
V_{OE}	$\overline{OE}$ Voltage for Chip Erase	10		15	V	$I_{OE}=10\mu A$
V_{LKO}	V_{CC} Lockout Level for Data Protection	4.0		4.4	V	

A.C. TEST CONDITIONS

Output Load ..... 1TTL gate and C_L = 100pF
Input Rise and Fall Times (10% to 90%) 20ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level..... 0.8V and 2.0V
Output Timing Reference Level .... 0.8V and 2.0V

Notes:
1. This parameter is only sampled and not 100% tested.
2. $I_{LI(Max)}$ is less than 10μa when V_{IN} is less than 5.25V.

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0."

CAPACITANCE [1] $T_A = 25°C$, f = 1 MHz

Symbol	Parameter	Typ.	Max.	Units	Test Conditions
C_{IN}	Input Capacitance	5	10	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance		10	pF	$V_{OUT} = 0V$
C_{VCC}	V_{CC} Capacitance		500	pF	$\overline{OE} = \overline{CE} = V_{IH}$
$C_{WE\ (VPP)}$	WE Input Capacitance		50	pF	$\overline{OE} = \overline{CE} = V_{IH}$

A.C. CHARACTERISTICS

READ

Symbol	Parameter	2816A-2 Limits			2816A Limits			2816A-3 Limits			2816A-4 Limits			Units	Test Conditions
		Min.	Typ.[1]	Max.	Min.	Typ.[1]	Max.	Min.	Typ.[1]	Max.	Min.	Typ.[1]	Max.		
t_{ACC}	Address to Output Delay		150	200		200	250		300	350		400	450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		150	200		200	250		300	350		400	450	ns	$\overline{OE} = V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay	10		75	10		100	10		120	10		150	ns	$\overline{CE} = V_{IL}$
t_{DF}[2]	$\overline{OE}$ High to Output Float	0		80	0		80	0		100	0		130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0			0			0			0			ns	$\overline{CE} = \overline{OE} = V_{IL}$

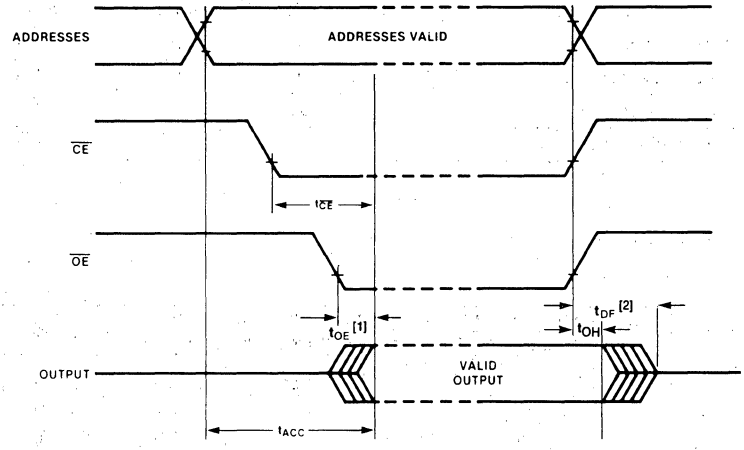
WRITE

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[1]	Max.		
t_{AS}	Address to Write Set-Up Time	150			ns	
t_{CS}	$\overline{CE}$ to Write Set-Up Time	150			ns	
t_{DS}[3]	Data to Write Set-Up Time	0			ns	
t_{DH}[3]	Data Hold Time	100			ns	
t_{WP}[4]	Write Pulse Width for 2816A	9		15	ms	
t_{WR}	Write Recovery Time	100			ns	
t_{OS}	Chip Erase Set-Up Time	0			ns	V_{OE} = 10V
t_{OH}	Chip Erase Hold Time	0			ns	V_{OE} = 10V
t_{BOS}	Byte Erase/Write Set-Up Time	0			ns	
t_{BOH}	Byte Erase/Write Hold Time	0			ns	
t_{WWR}[5]	Cycle Delay Time Following Write Cycle	40			μs	

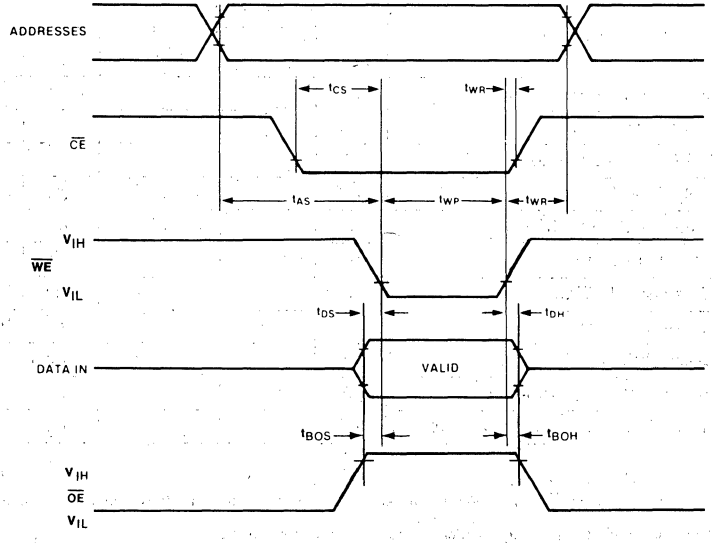
NOTES:
1. This parameter is only sampled and not 100% tested.
2. t_{DF} is measured from the point when $\overline{CE}$ or $\overline{OE}$ returns high (whichever occurs first) to the time when the outputs are no longer driven. This parameter is not 100% tested.
3. The data is set up and hold times for chip erase are identical to those specified for byte erase.
4. Adherence to t_{WP} specification is important to device reliability.
5. Adherence to t_{WWR} is important for device reliability.

WAVEFORMS

READ



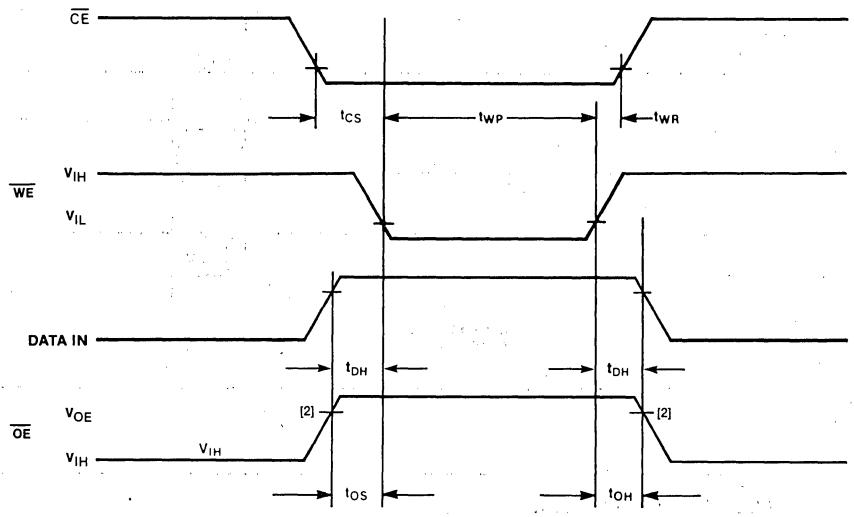
BYTE ERASE OR WRITE[3]



NOTES:
1. $\overline{OE}$ may be delayed up to 150 ns after falling edge of $\overline{CE}$ without impact on t_{CE} for 2816A.
2. t_{DF} is measured from the point when $\overline{CE}$ or $\overline{OE}$ returns high (whichever occurs first) to the time when the outputs are no longer driven. This parameter is not 100% tested.
3. Prior to a data write, an erase operation must be performed. For a byte erase, data in = V_{IH}.

WAVEFORMS (Continued)

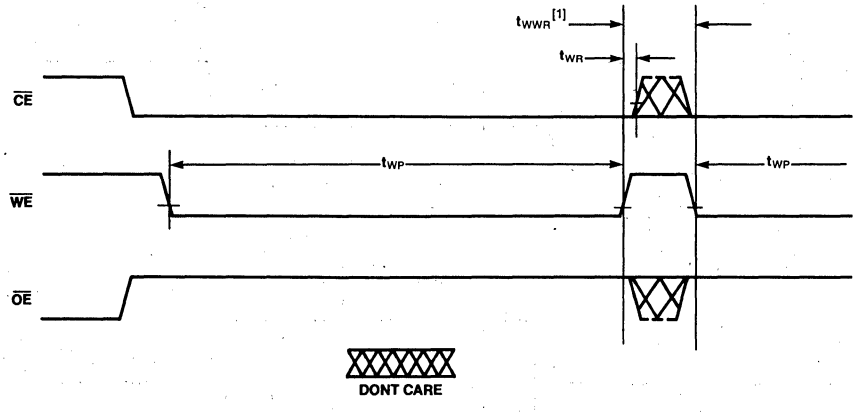
CHIP ERASE[1]



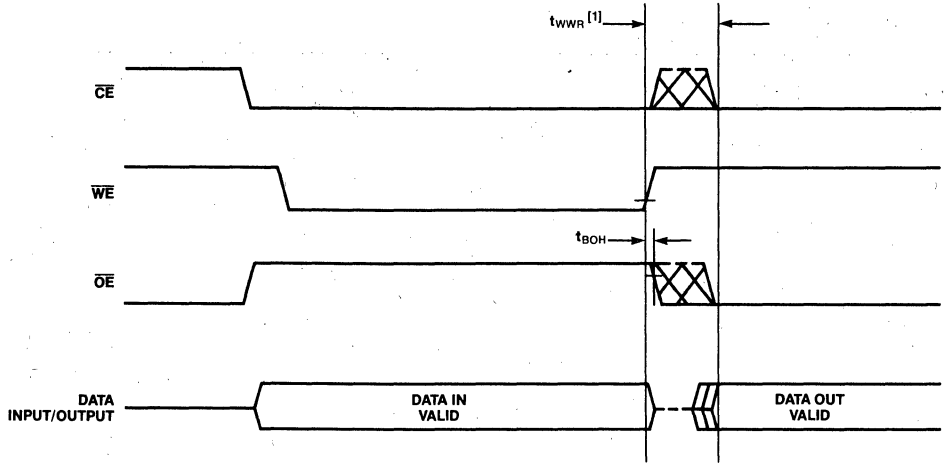
NOTES:
1. In the chip erase mode $D_{IN} = V_{IH}$.
2. Timing reference for $\overline{OE}$ during Chip Erase is 90% of V_{OE}.

WAVEFORMS (Continued)

CONSECUTIVE BYTE ERASE OR WRITE CYCLES



BYTE WRITE FOLLOWED BY READ CYCLE



NOTE:
1. Adherence to t_{WWR} is important for device reliability.

VPP OPTION SPECIFICATIONS[1]

ABSOLUTE MAXIMUM RATINGS

Vpp Supply Voltage with Respect to Ground
During Write/Erase ..... +22.5V to −0.1V
Maximum Duration of Vpp Supply at 22V
During Erase/Write Inhibit ..... 24 Hrs.
Maximum Duration of Vpp Supply at 22V
During Write/Erase ..... 20 ms

**NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[2]	Max.		
$I_{PP(R)}$	V_{PP} Current (Read)			10	μa	$\overline{CE}=V_{IL}$, $V_{PP}=4$ to 6
V_{PP}	Read Voltage	4		6	V	
$I_{PP(W)}$	V_{PP} Current (Byte Erase/Write)			10	μa	$\overline{CE}=V_{IL}$
V_{PP}	Write/Erase Voltage	20	21	22	V	
$I_{PP(C)}$	V_{PP} Current (Chip Erase)			10	μa	

CAPACITANCE[2] $T_A=25°C$, f = 1 MHz

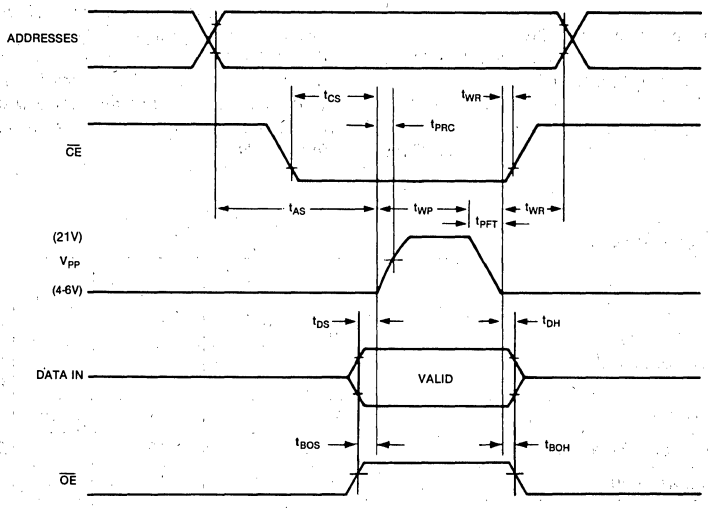
Symbol	Parameter	Typ.	Max.	Units	Test Conditions
C_{VPP}	V_{PP} Capacitance		50	pf	$\overline{OE}=\overline{CE}=V_{IH}$

A.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[2]	Max.		
t_{PRC}	V_{PP} RC Time Constant			750	μs	
t_{PFT}	V_{PP} Fall Time			100	μs	

NOTES:
1. Only specifications unique to V_{PP} option operation are shown. All other specifications under 5 volt only operation apply to V_{PP} option operation as well, except where noted.
2. This parameter is only sampled and not 100% tested.

BYTE ERASE OR WRITE WITH VPP OPTION[1,2]



NOTES:

1. Only specifications unique to V_{PP} option operation are shown. All other specifications under 5 volt only operation apply to V_{PP} option operation as well, except where noted.
2. Prior to a data write, an erase operation must be performed. For a byte erase, data in = V_{IH}.

PRELIMINARY

2817A
16K (2K x 8) ELECTRICALLY ERASABLE PROM

- 5 Volt Only Operation
- On-Chip Latches for Direct Microprocessor Interface
- Automatic Byte-Erase-before-Write
- Self Timed Byte Write
- Fast Read Access Time:
  - —2817A-1 200ns max
  - —2817A-2 200ns max
  - —2817A 250ns max
  - —2817A-3 350ns max
  - —2817A-4 450ns max
- Write Protect Circuit to Preserve Data on Power Up and Power Down
- 10,000 Erase/Write Cycles per Byte
- Reliable Intel HMOS*-E FLOTOX Cell Design Technology
- READY/$\overline{\text{BUSY}}$ Line for End-of-Write Signal
- 10 Year Data Retention For Each Write

The Intel 2817A is a 16,384 bit Electrically Erasable Programmable Read Only Memory. Like the Intel 2816A it has completely Non-Volatile Data Storage. In addition, it offers a high degree of integrated functionality which enables in-circuit byte writes to be performed with minimal hardware and software overhead. The Intel 2817A is a product of Intel's advanced E^2PROM technology and uses the powerful HMOS*-E process for reliable, non-volatile data storage.

*HMOS is a patented process of Intel Corporation.

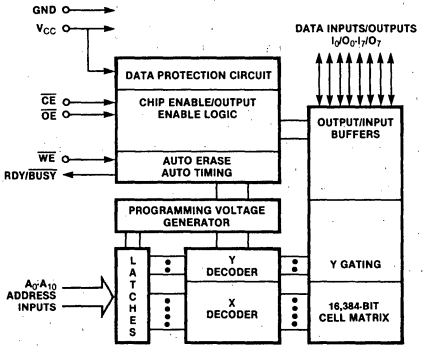


Figure 1. 2817A Functional Block Diagram

PIN NAMES

A_0-A_{10}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	DATA OUTPUTS
I_0-I_7	DATA INPUTS
RDY/$\overline{BUSY}$	DEVICE READY/BUSY
N.C.	NO CONNECT
$\overline{WE}$	WRITE ENABLE

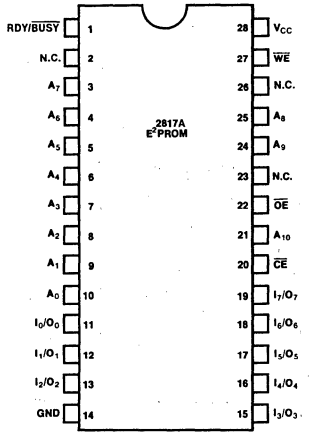


Figure 2. 2817A Pin Configuration

OCTOBER 1983
ORDER NO. 230641-002

The Intel 2817A incorporates all the interfacing hardware logic and the secondary voltage supply required to perform data writes. The device has complete self-timing which leaves the processor free to perform other tasks until the 2817A signals 'Ready'. With a transparent erase before write, the user benefits by saving an erase command which contributes to efficient usage of system processing time. On chip latching further enhances system performance.

The Intel 2817A's exceptionally fast read access time makes it compatible with high performance microprocessor applications. It uses Intel's proven 2-line control architecture which eliminates bus contention in a system environment. Combining these features with the 2817A's open drain 'Ready' signal makes the device an extremely powerful, yet simple to use, E^2 memory — available to the designer today.

The density, and level of integrated control, makes the Intel 2817A suitable for users requiring low hardware overhead, high system performance, minimal board space and design ease. Designing with, and using the 2817A, is extremely cost effective since all of the required programming voltage and interfacing hardware required for other E^2PROM devices has been eliminated. See Figures 1, 2, and 3 for the Intel 2817A's block diagram, pinout, and simple interface requirements.

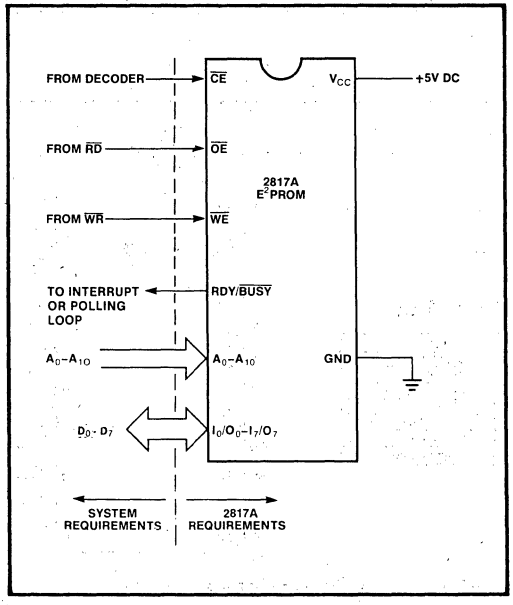


Figure 3. Simple 2817A Interface Requirements

DEVICE OPERATION

The Intel 2817A has 3 basic modes of user operation which are detailed in Table 1. All modes are designed to enhance the 2817A's functionality to the user and provide total Intel E^2PROM microprocessor compatibility.

Table 1. V_{CC} = +5V

Mode \ Pin	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	I_0/O_0-I_7/O_7	RDY/$\overline{BUSY}$
Read	V_{IL}	V_{IL}	V_{IH}	D_{OUT}	Hi-Z
Standby	V_{IH}	X	X	High Z	Hi-Z
Byte Write	V_{IL}	V_{IH}	V_{IL}	D_{IN}	V_{OL}
Byte Erase	Automatic before each 'Write'				

NOTE: RDY/$\overline{BUSY}$ is an open-drain output with I_{OL} = 2.1 ma.

The Write Mode

The 2817A is programmed electrically in-circuit, yet it provides non-volatile storage without the constraint of ultraviolet erasure with EPROMs or of batteries with CMOS RAMs. Writing to non-volatile memory has never been easier as no external latching, erasing or timing are needed. When commanded to byte write, the 2817A automatically latches the address, data, and control signals, and starts the write. While the write operation is in progress, the RDY/$\overline{BUSY}$ output is low. The data bus is not used by the 2817A during the write operation, allowing the processor to perform other tasks.

After the write cycle is initiated with a $\overline{WE}$ strobe pulse, the 2817A completes the cycle off-line from the processor in two transparent steps. First, the existing data at the addressed location is automatically erased. At the beginning of this step the inputs are locked out, the data lines are brought to a high impedance state and the RDY/$\overline{BUSY}$ signal is lowered to V_{OL}. Second, the new data byte is written into the device. At the

end of this process, the 2817A raises its RDY/$\overline{BUSY}$ signal to V_{OH} to notify the processor that the write cycle is complete and that the device is ready for read or write access.

The write endurance is 10^4 cycles, that is, up to 10,000 write cycles can be reliably performed on each byte.

The Read Mode

One aspect of the 2817A's high performance is its fast read access time—typically less than 200 ns. Its read cycle is similar to that of EPROMs and static RAMs. The Intel 2817A can be selected using decoded system address lines to $\overline{CE}$ and then the device can be read, within the device selection time, using the processor's $\overline{RD}$ signal connected to $\overline{OE}$.

Read retention for data written into the 2817A is greater than 10 years.

The Standby Mode

The 2817A has a standby mode in which power consumption is reduced by 60%. This offers the user power supply cost benefits when designing a system with Intel 2817A's. This mode occurs when the device is deselected ($\overline{CE} = V_{IH}$). The data pins are put into the high impedance state regardless of the signals applied to $\overline{OE}$ and $\overline{WE}$ concurrent with the reading and writing of other devices.

On-Chip Data Protection on V_{CC} Power Up and Power Down

An erase/write of a byte in the 2817A is accomplished with input signals $\overline{CE}$, $\overline{WE} = V_{IL}$. During system (V_{CC}) power up and power down, this condition may be present as V_{CC} ramps up to or down from its steady state value of 5 volts. To prevent the possibility of an inadvertent byte write during this power transition period, an on-chip sensing circuit disables the internal programming circuit if V_{CC} falls below 4 volts (V_{LKO}).

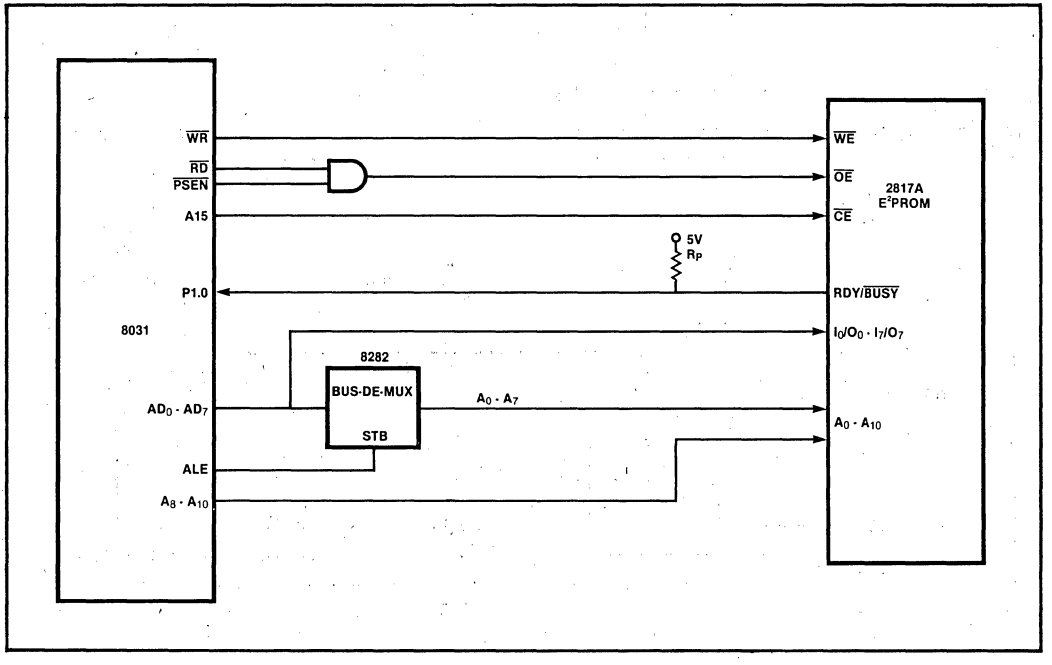


Figure 4. 2817A E^2PROM as Remotely-Alterable Non-Volatile Program Code Memory in an MCS® -51 System.

System Implementation

The 2817A is compatible with Intel MCS® -51 Microcomputer and the iAPX 86/88, 186/188, and 286 microprocessor families. The 2817A requires no interface circuitry. Figure 4 shows an example of the 2817A used as program storage memory in an 8031 single-chip microcomputer system. The 8031 program code is modified or upgraded remotely without having to remove the E^2PROM from the system.

The Intel 8282 8-bit latch shown in Figure 4 is used to de-multiplex addresses A0-A7. (The latch is typically required in any 8031 system.) The ANDing of signals $\overline{PSEN}$ and $\overline{RD}$ allows both program code and external data for the 8031 to be fetched/read from the 2817A E^2PROM. The 8031 determines when a given write cycle has been completed by reading port bit P1.0 which is connected to the 2817A's RDY/$\overline{BUSY}$ output.

Application Note AP-158 shows a complete schematic diagram of an 8088-based system design example. Included in AP-158 are various hardware design options for different applications and a number of 8088 code software routines for remote downloading applications,

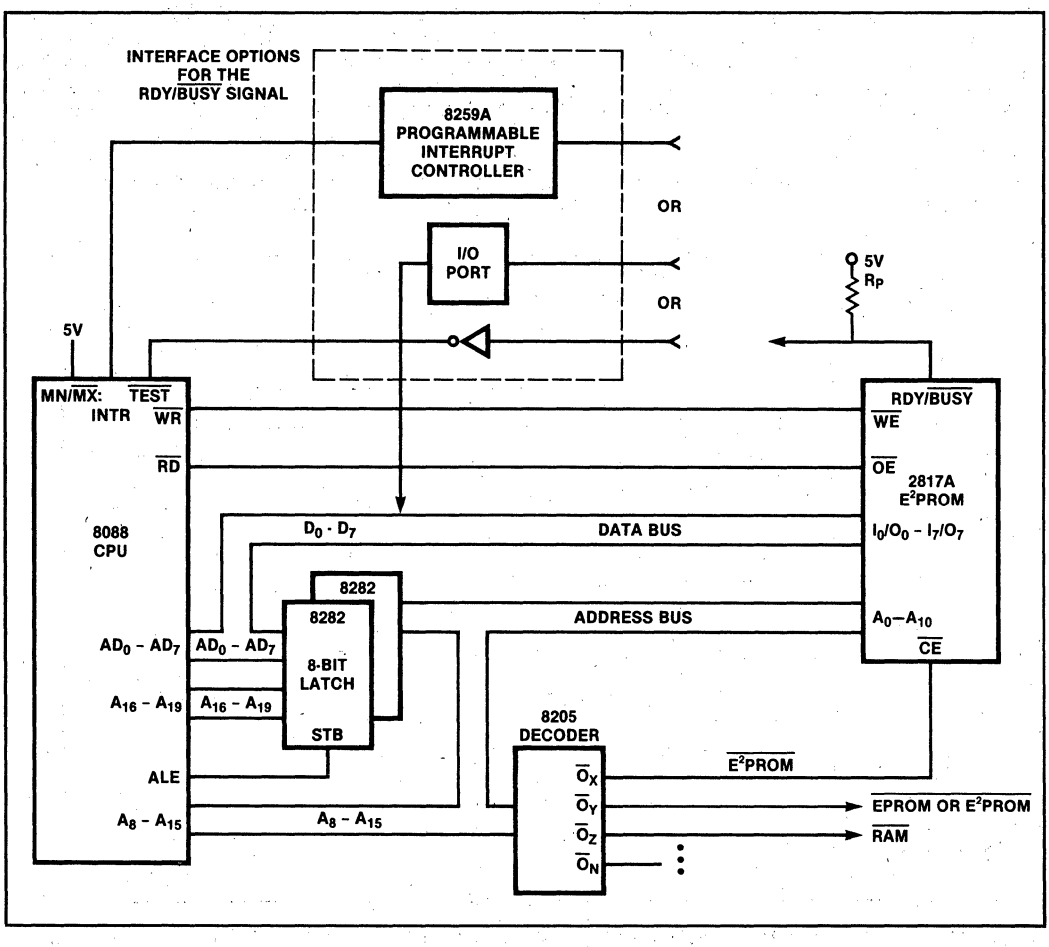


Figure 5. 2817A/8088 Interface Example

and for using the 8259A Programmable Interrupt Controller and the 8251A Programmable Communications Interface.

Interfacing to the Intel 8088 is similar to the 8031 interface. The difference lies in the use of the 8259A (Programmable Interrupt Controller). The Ready line can be connected to any of the interrupt request pins in order to interrupt the processor. (See Figure 5). Alternatively, it can be polled through an I/O port or connected through an inverter to $\overline{TEST}$ on the Intel 8088.

Using RDY/$\overline{BUSY}$

The RDY/$\overline{BUSY}$ pin is an open-drain output which allows two or more RDY/$\overline{BUSY}$ signals to be OR-tied together. To calculate the value of the pull-up resistor for the RDY/$\overline{BUSY}$ output, the following formula can be used:

$$R_P = \frac{4.6V}{2.1ma - I_{IL}}$$

where I_{IL} = the total V_{IL} input current of all devices connected to the RDY/$\overline{BUSY}$ line.

A typical pull-up resistor value for the RDY/$\overline{BUSY}$ output is 3.0K ohms, assuming that the input sink current (I_{IL}) of the input(s) being driven is less than 0.5 ma.

Applications

The Intel 2817A is ideal for non-volatile memory requirements in applications requiring storage of user defined functions, calibration constants, configuration parameters and accumulated totals. Soft key configuration in a character-oriented or graphics terminal is an example where user defined functions, such as protocol, color, screen attributes and character fonts can be keyed in by the user. Calibration constants can be stored in the 2817A with a smart interface for a robot's axis of movement. Movement constants, compensation algorithms and learned axis characteristics would be included. In programmable controllers and data loggers, configuration parameters for polling time, sequence and location, can be stored in the 2817A. Additional applications include accumulated totals for dollars, energy consumption, volume and even the logging of service performed on computer boards or systems for documentation purposes.

The Intel 2817A is cost effective for lower density E^2PROM applications and can therefore be used to provide a lower system cost to the user. The 2817A user will find that tangible cost savings per system include: board space and component reductions, reduced assembly costs, savings in inventory costs, handling costs and Quality Assurance. System designers will find the 2817A reduces design time by a sizeable factor due to the integration of timing, logic and latching.

The 2817A will also open up new applications in environments where flexible parameters/data storage could not be implemented before. Applications with board space constraints are ideal for the 2817A due to the on-chip integration of all functions required.

In Application Note AP-158 a comprehensive discussion of four major application areas for E^2PROMs is given:

1. Remote Firmware Downloading
2. System Re-Configuration (system parameter storage)
3. Maintenance Logging
4. Electronic Message Storage

Write Time Characteristics

The 2817A's internal write cycle contains an automatic erase feature. Furthermore, the 2817A automatically determines through its self-timer when a byte has been written and signals the completion via the Ready/$\overline{Busy}$ signal. The Ready/$\overline{Busy}$ signal is an open-drain output. The 2817A's internal cycle consists of an automatic 10 ms erase followed by a write. The total cycle is the time that Ready is held low by the device. The 2817A maximum specification is 20 ms for a combined Erase/Write cycle. The 2817A-1 is ideal for users desiring a faster erase/write cycle time. The total cycle time for this part is 10msec.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature..... −65°C to +100°C
All Input or Output Voltages with
Respect to Ground ..... +6V to −.3V

**NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. AND A.C. OPERATING CONDITIONS DURING READ AND WRITE

Temperature Range	0°C - 70°C
V_{CC} Power Supply	5V ± 5%

D.C. CHARACTERISTICS

Symbol	Parameter	Min.	Typ.(1)	Max.	Units	Test Conditions
I_{LI}	Input Leakage Current			10	μA	$V_{In} = V_{CC}$ Max.(2)
I_{LO}	Output Leakage Current			10	μA	$V_{Out} = V_{CC}$ Max.
I_{CCA}	V_{CC} Current (Active)		60	120	mA	$\overline{OE} = \overline{CE} = V_{IL}$
I_{CCS}	V_{CC} Current (Standby)			55	mA	$\overline{CE} = V_{IH}$, $V_{CC} = V_{CC}$ Max.
I_{CCW}	V_{CC} Current (Write)			150	mA	$\overline{WE}$ = ⊓, $\overline{CE} = V_{IL}$
V_{IL} (D.C.)	Input Low Voltage (D.C.)	−0.1		.8	V	
V_{IL} (A.C.)	Input Low Voltage (A.C.)	−0.4			V	Time = 10ns
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			.45	V	$I_{OH} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400\mu A$
V_{LKO}	V_{CC} Level for Write Lockout	4.0		4.4	V	

NOTE:
1. This parameter only sampled and not 100% tested.
2. $I_{IL(max)}$ is less than 10μa when V_{IN} is less than 5.25V.

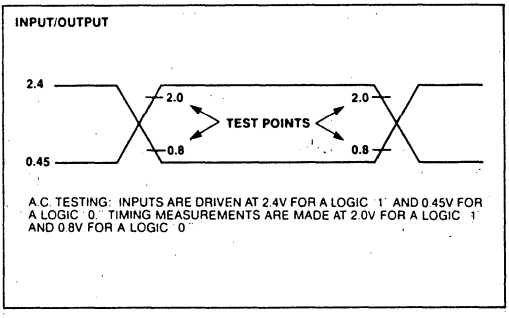
CAPACITANCE (T_A = 25°C, f = 1 MHz)

Symbol	Parameter	Typ.(1)	Max.	Units	Test Conditions
C_{IN}	Input Capacitance	5	10	pF	V_{IN} = OV
C_{OUT}	Output Capacitance		10	pF	$\overline{OE} = \overline{CE} = V_{IH}$

A.C. TEST CONDITIONS

Output Load . . . . . 1 TTL gate + C_L = 100pF
Input Rise and Fall Times(10% to 90%) 20ns
Input Pulse Levels . . . . . 0.45V to 2.4V
Input Timing Reference Level . . . . 0.8V and 2.0V
Output Timing Reference Level . . . 0.8V and 2.0V

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. CHARACTERISTICS

READ

Symbol	Parameter	2817A-1 2817A-2 Limits			2817A Limits			2817A-3 Limits			2817A-4 Limits			Units	Test Conditions
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.		
t_{ACC}	Address to Output delay.		150	200		200	250		300	350		400	450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay.		250	200		200	250		300	350		400	450	ns	
t_{OE}	$\overline{OE}$ to Output Delay.			75			100			120			150	ns	
t_{DF}(2)	$\overline{OE}$ High to Output not Driven	0		60	0		60	0		80	0		100	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0			0			0			0			ns	$\overline{CE}$, $\overline{OE} = V_{IL}$

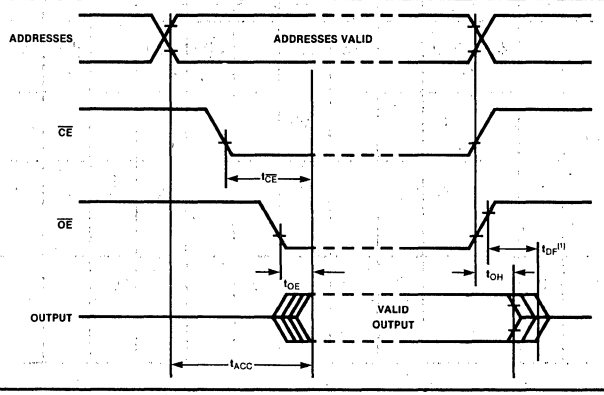
NOTES:
1. This parameter only sampled and not 100% tested.
2. t_{DF} is measured from the point when $\overline{CE}$ or $\overline{OE}$ returns high (whichever occurs first) to the time when the outputs are no longer driven. This parameter is not 100% tested.

A.C. CHARACTERISTICS (Cont.)
WRITE

Symbol	Parameter	2817A-1			2817A-2			2817A 2817A-3			2817A-4			Units
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
t_{AS}	Address to write set-up time	20			20			20			60			ns
t_{CS}	$\overline{CE}$ to write set-up time	20			20			20			20			ns
t_{WP}	Write pulse width	100			100			100			200			ns
t_{AH}	Address hold time	50			50			50			110			ns
t_{DS}	Data set-up time	50			50			50			70			ns
t_{DH}	Data hold time	20			20			20			20			ns
t_{CH}	$\overline{CE}$ hold time	0			0			0			0			ns
t_{DB}	Time to Device Busy			120			120			120			120	ns
t_{WR}	Bytes Write Cycle Time		9	10		17	20		17	20		25	30	mS
	Number of Writes per byte	10^4			10^4			10^4			10^4			

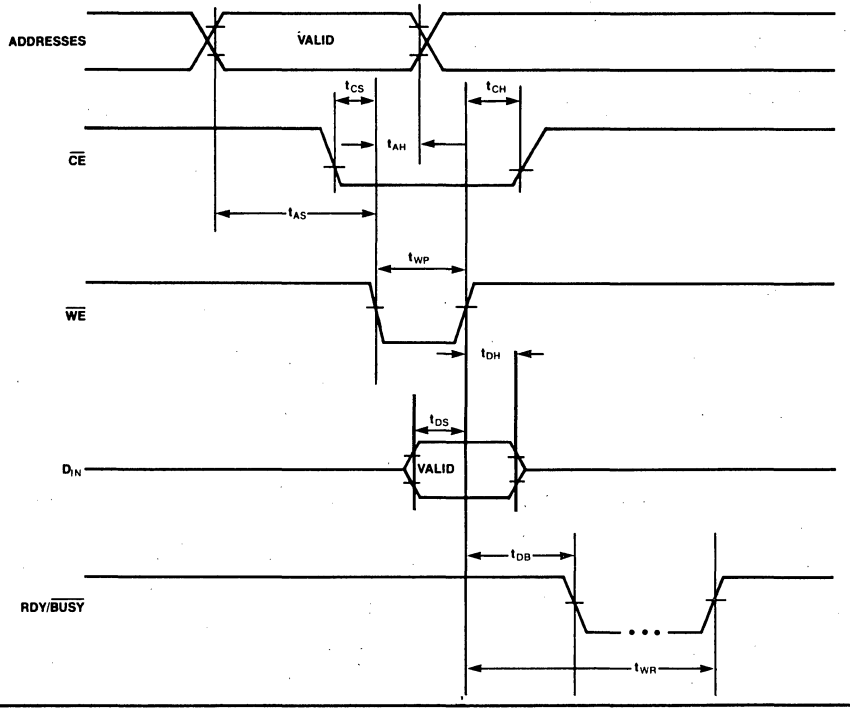
WAVEFORMS

READ



NOTE:
1. t_{DF} is measured from the point when $\overline{CE}$ or $\overline{OE}$ returns high (whichever occurs first) to the time when the outputs are no longer driven. This parameter is not 100% tested.

WRITE



AVAILABLE LITERATURE

The Intel E^2PROM family of devices, the 2816A and 2817A, is supported by many Application Notes. Topics covered range from Intel E^2PROM Technology and Reliability to Design considerations and Applications support for Designers implementing large arrays of E^2PROMs.

These notes and more are available in the Memory Components Handbook (Order No. 210830) or the Intel Literature Department, 3065 Bowers Ave., Santa Clara, CA 95051. To obtain this book contact your local Field Sales office. Your Field Applications Engineer is available to discuss all aspects of the Intel E^2 product line with you.

Bubble Memory

6

A PRIMER ON MAGNETIC BUBBLE MEMORY

Magnetic bubble memory is a solid-state technology with high reliability, ruggedness, small size, light weight, and limited power dissipation. It has applications in telecommunications, data acquisition, industrial control, terminals, and small business computers. Yet many potential users remain unsure of the nature of a bubble memory. This primer is intended to introduce these users to the technology.

What a Magnetic Bubble Memory Is

A magnetic bubble memory stores data in the form of cylindrical magnetic domains in a thin film of magnetic material. The presence of a domain (a bubble) is interpreted as a binary 1, and absence of a domain is a 0. Bubbles are created from electrical signals by a bubble generator within the memory, and reconverted to electrical signals by an internal detector. Externally the memory is TTL-compatible.

An external rotating magnetic field propels these cylindrical domain bubbles through the film. Metallic patterns or chevrons deposited on the film steer the domains in the desired directions. Transfer rates, once started, are in the tens of thousands of bits per second, but because the data circulates past a pickup point at which it becomes available to the outside world, there is a latency averaging tens of milliseconds before data transfer can begin. In these respects, magnetic bubble memories are serial high-density storage devices like electromechanical disk memories. However, in a disk, the stored bits are stationary on a moving medium, whereas in the magnetic bubble memory the medium is stationary and the bits move.

Advantages of Magnetic Bubble Memories

The principal advantage of magnetic bubble memories are their non-volatility—that is, if power fails, the stored data is retained. Products incorporating bubble memories therefore do not require battery backups. Magnetic bubble memories share this feature with read only memories (ROMs), erasable PROMs (EPROMS), and electrically erasable PROMs (E^2PROMS). However, unlike any of these technologies, magnetic bubble memories can have data written into them at any time, at speeds comparable to those at which data is read. Furthermore, unlike disk memories, bubble memories are quiet and very reliable, because they have no moving parts. They are compact, and they dissipate very little power. Their support circuits are compatible with microprocessor systems. With a million or more bits per device, a bubble memory can store 16 to 64 times the amount of data of alternative semiconductor memories, providing very high storage capability in a compact space. Bubble memory has a wide variety of applications, some of which are listed in Table 1.

Table 1. Bubble Memory Applications

Numerical control	Robotics
Process control	Oil exploration
Aircraft navigation	Data acquisition
Cable television	Portable instruments
Telecommunications terminals	Avionics
Point-of-sale terminals	Gasoline pumps
Private branch telephone exchanges	Personal computers
Word processors	Office equipment
Flight-line test equipment	Automatic test equipment
Data encryption	

How Bubbles are Formed

Magnetic domains are found in all kinds of magnetic materials—iron bars, the coating on magnetic tape, ferrite toroids (the most common form of computer memory in the 1960s). Each domain is a group of atoms with parallel magnetic orientations. When the material in bulk is unmagnetized, the domains are oriented at random in three dimensions. When the material is magnetically saturated, most of the domains have the same orientation. Magnetization to a level less than saturation orients some of the domains to a common direction, but leaves many of them randomly oriented. When a domain orientation changes, usually by imposing an external magnetic field, the domain itself does not physically move, but boundaries between domains that have different orientations move or disappear altogether.

In an extremely thin film, less than 0.001 inch thick, the domain orientations may be constrained to two dimensions. In some kinds of material (orthoferrites and garnets), with proper crystallographic orientation, the domain orientations are always perpendicular to the film. When these materials are not in a magnetic field, some domains are oriented upward and some downward (north magnetic poles of some domains are on top of the film, and those of other domains are on the bottom). In these materials, the magnetic domains tend to be long and snakelike in the absence of an external field (Figure 1). When a weak magnetic field is applied perpendicular to the film, the domains that are oriented opposite to the applied field become substantially narrower. As the applied field, called a *bias field*, is made stronger, the length continues to decrease, until it becomes approximately the same as the width. Each domain is now cylindrical, magnetized oppositely to the applied field, and immersed in a much larger domain that is magnetized in the same direction as the field.

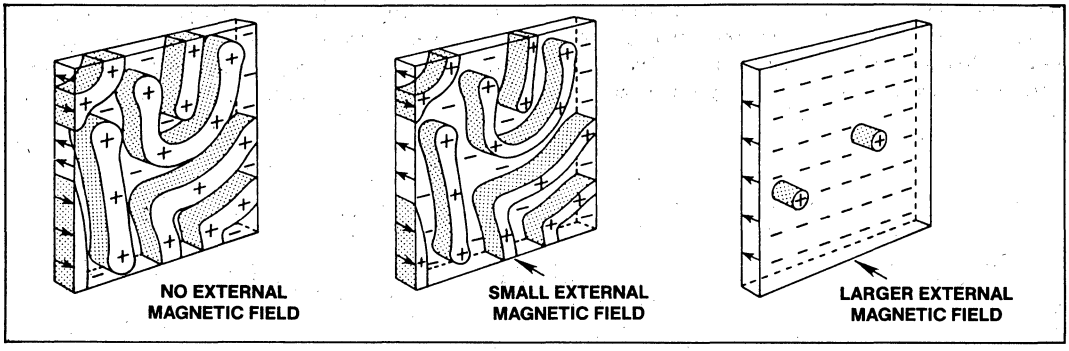


Figure 1. Magnetic Domains in Thin Film Under Increasing Magnetic Bias Field.

These small domains are the bubbles, generally less than 3 micrometers (1/ 10,000 inch) in diameter (Figure 2). When they are viewed from above, only the round shape is apparent, giving the domains the appearance of bubbles. If the bias field were to be made still stronger, all the bubbles would shrink and then disappear altogether; the entire film would be magnetized in the same direction as the bias field. The effect is reversible—that is, if the bias is removed, the domains return to a snakelike form.

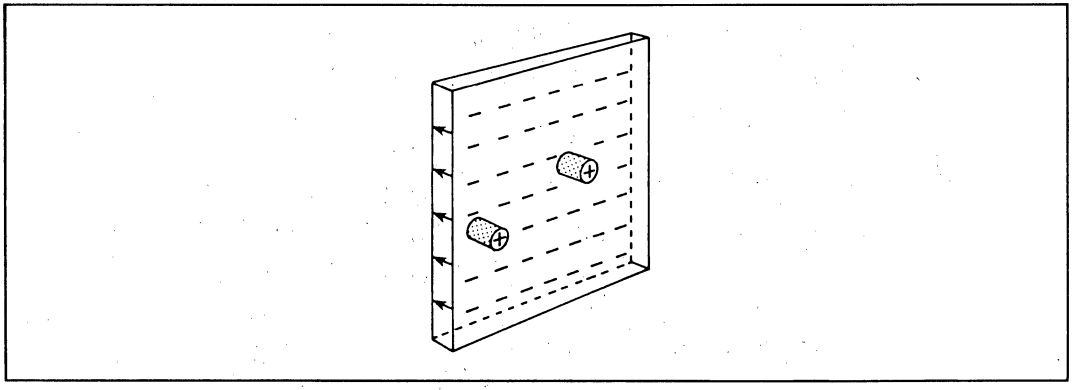


Figure 2. Magnetic Bubbles in a Thin Film

Why a Bubble Moves

Magnetic bubbles will move if they are in a magnetic field gradient. For instance, it will move from a region of lesser magnetic field strength to a region of greater strength. This is similar to the way a nail is pulled to the end of a bar magnet when it gets close the magnet.

In a bubble memory a magnetic film pattern is overlaid on the layer of bubbles. When this layer is magnetized it pulls the bubbles to the points of greatest field strength (or poles) as in Figure 3. The bubbles could then be moved if the pattern elements were moved.

A more easily controlled magnetic field is generated by two coils wrapped around the bubble layer and magnetic film pattern. With appropriate specification of the current in two coils positioned at right angles, the direction of the poles on the stationary elements can be changed in a controlled manner.

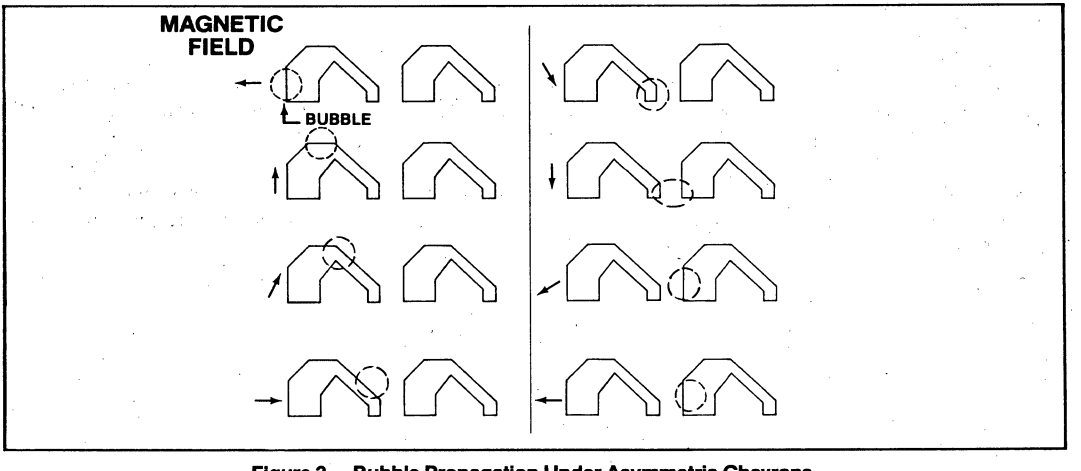


Figure 3. Bubble Propagation Under Asymmetric Chevrons

Various shapes for these metallic patterns have been used by different manufacturers to control the movement of the bubbles. At Intel asymmetric chevrons are used (Figure 3).

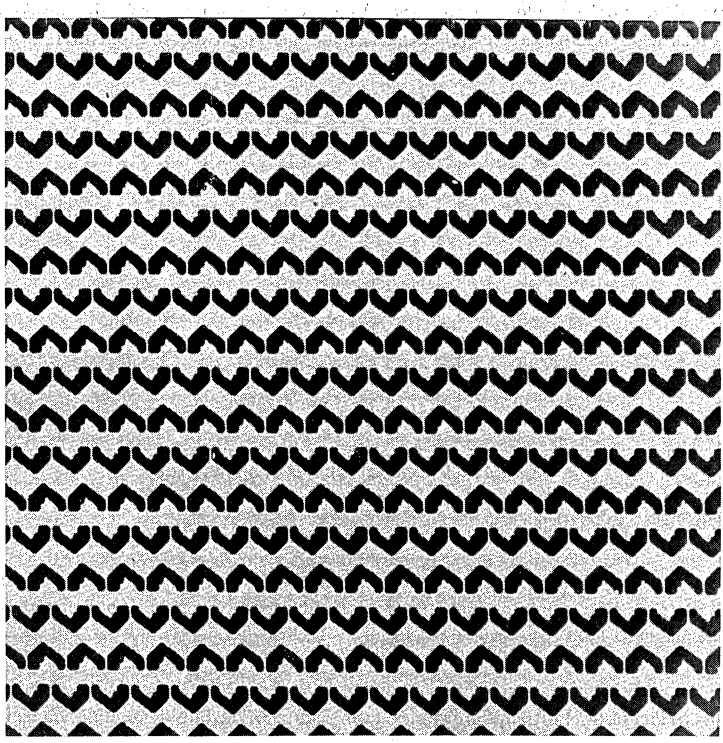


Photo 1. Asymmetric Chevrons Deposited on a Thin Film

Why Magnetic Bubbles are Non-Volatile

In a magnetic bubble memory system, the bias field in which the bubbles exist is generated by a pair of *permanent* magnets. The substrate bearing the thin film and its bubbles is mounted between these magnets and is therefore continuously subject to the bias field.

The rotating field that propels the bubbles through the film is generated by currents in two coils wrapped around the substrate at right angles to each other. These currents are generated by electronic circuits that are part of the magnetic bubble memory system. No mechanical motion is involved.

If power fails, the circuits stop operating, the rotating field disappears, and the bubbles stop moving. But the bias field, generated by the permanent magnet, is not affected. Therefore the bubbles and the data that they represent are maintained in the film. When power is restored the data is again accesible.

BUBBLE MEMORY MANUFACTURING TECHNOLOGY

Bubble memories are produced in a process that resembles semiconductor manufacturing in many ways (Figure 4). Manufacturing begins with a nonmagnetic garnet wafer on which a magnetic film is deposited, using conventional techniques. An ion implantation process alters the magnetization of the top surface of the film, discouraging the formation of abnormal bubbles with undesirable dynamic properties. Then nonmagnetic conductors, bubble-steering patterns of magnetic metal, insulation, passivation, and bonding pads are deposited in much the same way as successive layers on semiconductor integrated circuits. Patterns in each layer are defined photolithographically, just as with semiconductors.

Magnetic bubble technology differs from semiconductor technology in the materials used and in the complexity of the process. Semiconductor circuits use eight or more layers of silicon doped with various materials that affect its electrical characteristics, compared to about three layers of essentially pure metallic and insulating material in bubble technology. These materials are chosen for their magnetic rather than their electrical properties.

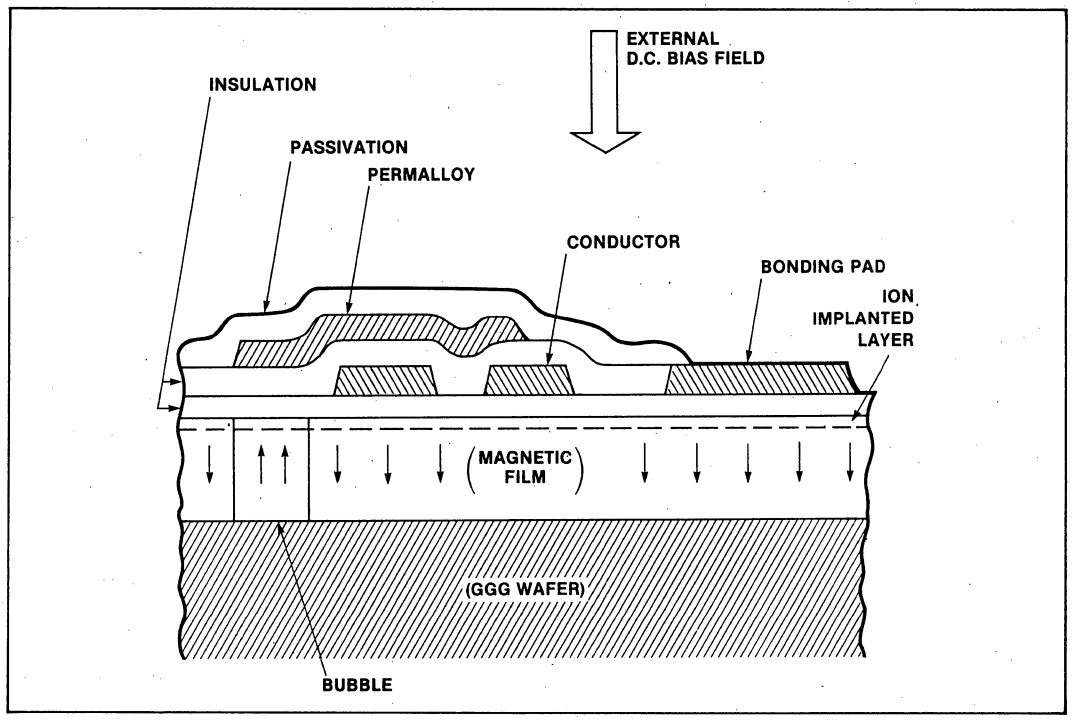


Figure 4. Magnetic Bubble Chip Cross Section

Bubble Memory Functional Description

The Intel 7110 magnetic bubble memory unit contains the bubble chip, the coils that generate the rotating field, two permanent magnets for the bias field, and a magnetic shield that prevents disturbances by external fields and forms a return path for the bias field around the bubble chip (Figure 5).

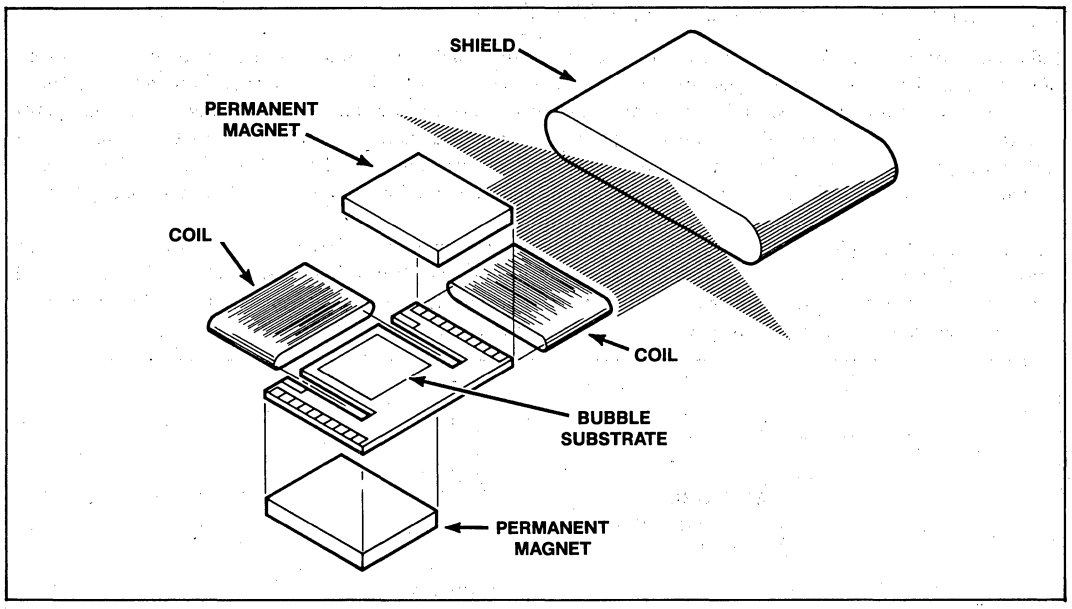


Figure 5. Magnetic Bubble Unit Assembly—Exploded View

Bubble Memory Architecture

Data is stored in the bubble memory unit with a block-replicate architecture (Figure 6). This architecture consists of a number of endless storage loops around which corresponding bits of successive pages continuously circulate, and two tracks, designated input and output, through which the controller writes and reads data in the storage loops. Exchange or replication of data between the tracks and the loops occurs in all loops simultaneously—the key idea in this architecture.

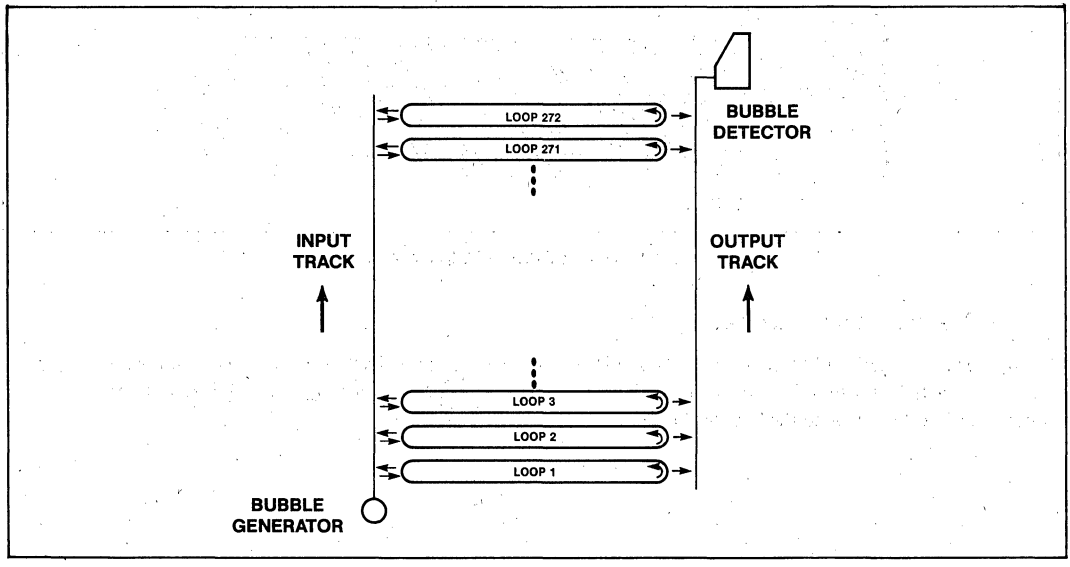


Figure 6. Block-Replicate Architecture

WRITING DATA INTO THE BUBBLE MEMORY

Seed Bubble

The seed bubble, at the beginning of the input track, is generated by an electric current pulse in a hairpin-shaped loop of conductive material. The pulse is strong enough to reverse the bias field locally and thus allow a bubble domain to be created. Once having been created, the seed bubble remains in existence as long as the external bias field is maintained.

The seed circulates under a permalloy patch, driven by the rotating field that propagates bubbles elsewhere in the memory. This bubble is constrained to a kidney shape by interaction of the bias and rotating field with the metal patch (Figure 7). The seed is split in two by a current pulse in the hairpin-shaped conductor. One of them remains under the patch as the seed, quickly regaining its original size; the other one, driven by the rotating field, is transferred to the input track section of the chip. The current pulse that splits the seed is generated to store a binary 1 in the memory; to store a 0, the pulse is omitted, and no bubble is generated.

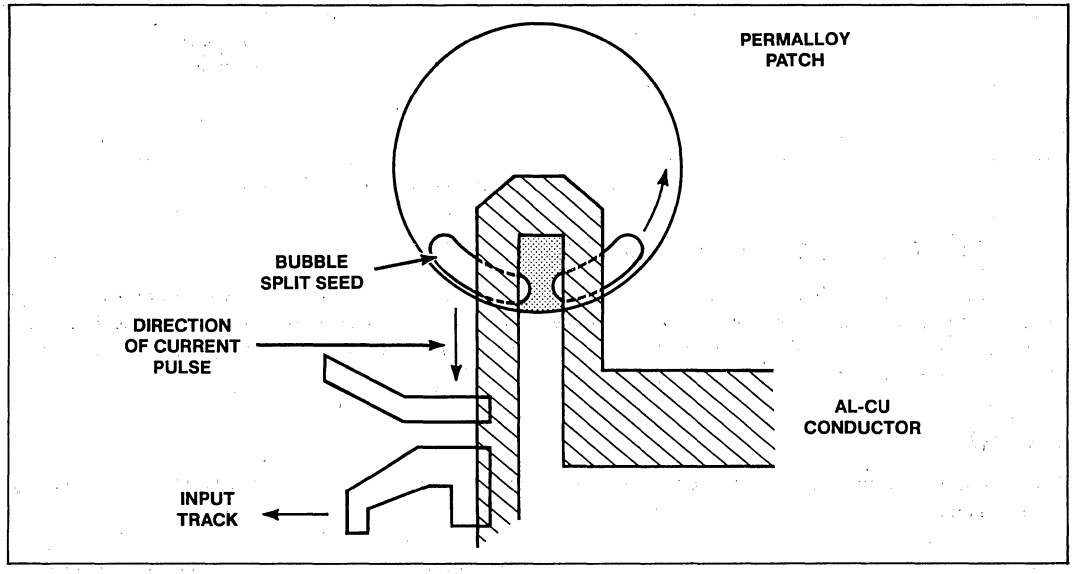


Figure 7. Seed Bubble and Bubble Generation

A seed bubble is maintained at one end of the input track. Bubbles corresponding to binary 1's in the input word are split from the seed and propagate along the input track. When the input track contains exactly one page (64 bytes) then the bubbles exchange places with old bubbles previously circulating in the loops. This is accomplished by an operation called swapping. Thereafter the new bubbles circulate, while the old bubbles now in the track propagate to the end and are destroyed.

Swapping

Transfer of data from the input track to a storage loop involves a swap, bringing the old data onto the input track for destruction at the end of the line, while the new data takes its place in the loop. This is done when a current pulse in an associated conductor under the chevrons causes a bubble to jump from the input track to the storage loop and vice versa. The swap pulse is essentially rectangular, preserving the bubble without cutting it in two.

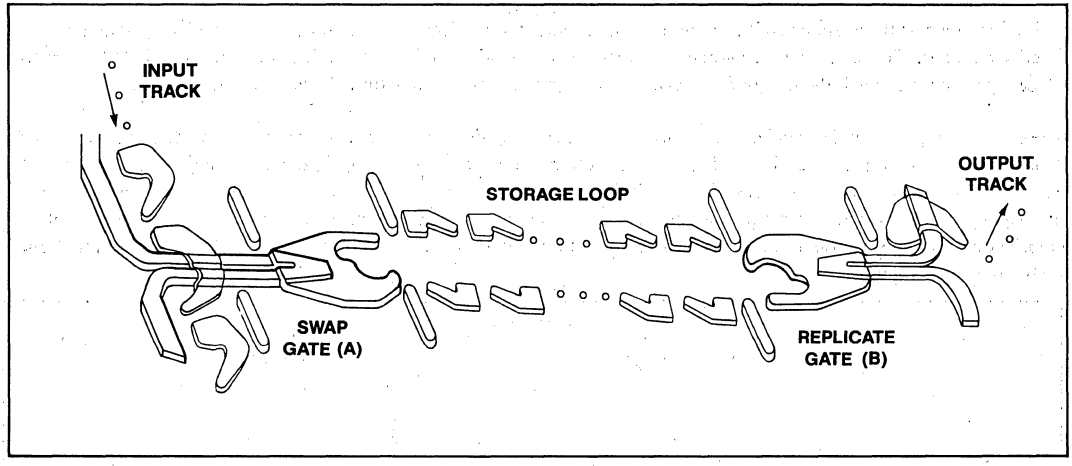


Figure 8. Swapping and Replication Configuration in Bubble Memory

READING DATA STORED IN THE BUBBLE MEMORY

To read the stored data, the circulating bubbles are replicated, one bubble or one unoccupied bubble site from each loop, onto the output track, after which they propagate to a bubble detector at its far end. After detection, these output bubbles are also destroyed. Meanwhile, the data in the loops continues to circulate, permitting a particular page to be read out repeatedly without regeneration, and protecting the stored data if power fails.

Replication

Data is transferred from the storage loop to the output track by replication, continuing to circulate in the loop after having been read out.

For replication, the bubble is propagated under a large element where it is stretched out. As it passes under a hairpin shaped conductor loop it is cut by a current pulse just as in bubble generation.
The replicating current pulse waveshape has a high, narrow leading spike for cutting the original bubble in two, and a lower and wider trailing portion during which the new bubble moves under the output track. The entire pulse lasts about one-quarter of a cycle of the rotating field. In this manner the data in the storage loops is replicated onto the output track, and yet retained in the storage loops in case of a sudden power failure.

Near the end of the output track is a bubble detector—essentially a magnetoresistive bridge formed by interconnecting the permalloy chevrons to make a continuous electrical path of maximum length (Figure 9). As bubbles pass under the bridge, the resistance changes slightly, modulating the currents through the bridge and creating an output voltage of several millivolts. Bubbles are stretched at right angles to the direction of propagation by adding parallel rows of chevrons; these stretched bubbles generate larger output signals at the detector. Beyond the detector, the output track runs the bubbles into the guard rail and destroys them.

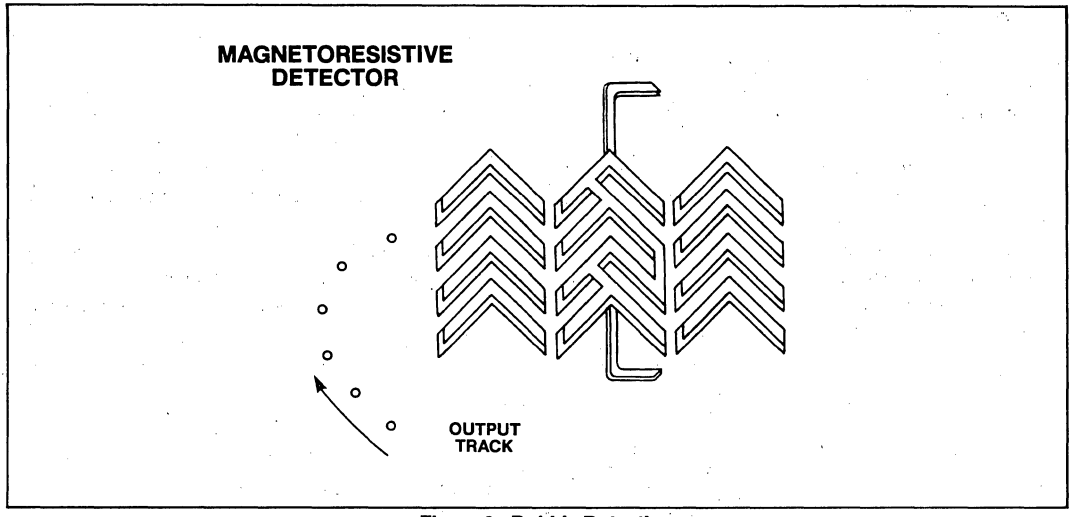


Figure 9. Bubble Detection

Redundancy

The Intel magnetic bubble memory unit physically stores data in 320 storage loops, with capacities of 4,096 bits each. Of the 320 loops, 272 are actually used (active) and 48 are spares (inactive); the boot loop records which loops are used.

Boot Loop

Some of the loops of an individual memory are set aside as spares. The decision as to which loops are to be used (active) and which are not to be used (inactive) is made after the memory unit has been assembled and is undergoing tests at the factory. The outcome of this decision is stored in an extra loop included in each memory chip, in the form of a 12 bit code for each "active" and "inactive" loop.

Whenever power is turned on in the memory system, the system must be initialized before it can be used. Part of the initialization process includes reading the contents of this extra loop, called the boot loop, and placing this information in a bootloop register in the formatter/sense amplifier. From then on, as long as power is on, this register identifies the "active" loops for both reading and writing; "inactive" loops are ignored. The formatter does not attempt to store data in "inactive" loops, and the sense amplifier ignores any data that appears from these loops.

Data Storage—External Appearance

Data is stored logically as 2,048 pages of 512 data bits each. 256 data bits plus 14 error-correction check bits and 2 unused bits are stored in each half of the bubble chip. If automatic error correction is not used, these 16 bits are available for data storage.

Error Correction

Error detection and correction can be performed in the formatter/sense amplifier, which includes a 14-bit cyclic redundancy code that corrects a single burst error of up to five bits in each 270-bit block including the code itself. These code bits are appended to the end of each 256-bit data block when writing into the cell, and checked when the block is read. The error correction feature can be used or not at the user's discretion, by properly setting a register in the bubble memory controller chip. If it is not used, the loops occupied by the code bits become available for additional data.

Access Time and Data Rate

Bubbles circulate at a rate of 50 kilohertz (the rotating field makes 50,000 complete revolutions per second). Average access time to the first bit of the first page is about 41 milliseconds—half the length of time required for a bubble to make one complete circuit of the loop, plus the time to shift a bubble along the length of the output track.

The 320 active and spare loops are actually in four "quads" of 80 loops each (Figure 10). This arrangement shortens the input and output tracks and thus reduces the read and write cycle times. The quads are separately addressable in pairs; in each pair the quads store odd-numbered and even-numbered bits of a word respectively. There are four seed bubbles and four input tracks, and four output tracks. The four output tracks share two detector bridges in such a way that there can never be bubbles from two tracks in a single detector simultaneously. By this means the four streams of output bubbles are interleaved into two bit streams that are stored in two registers in the sense amplifier. The data in these registers is interleaved again into a single stream transmitted serially to the controller.

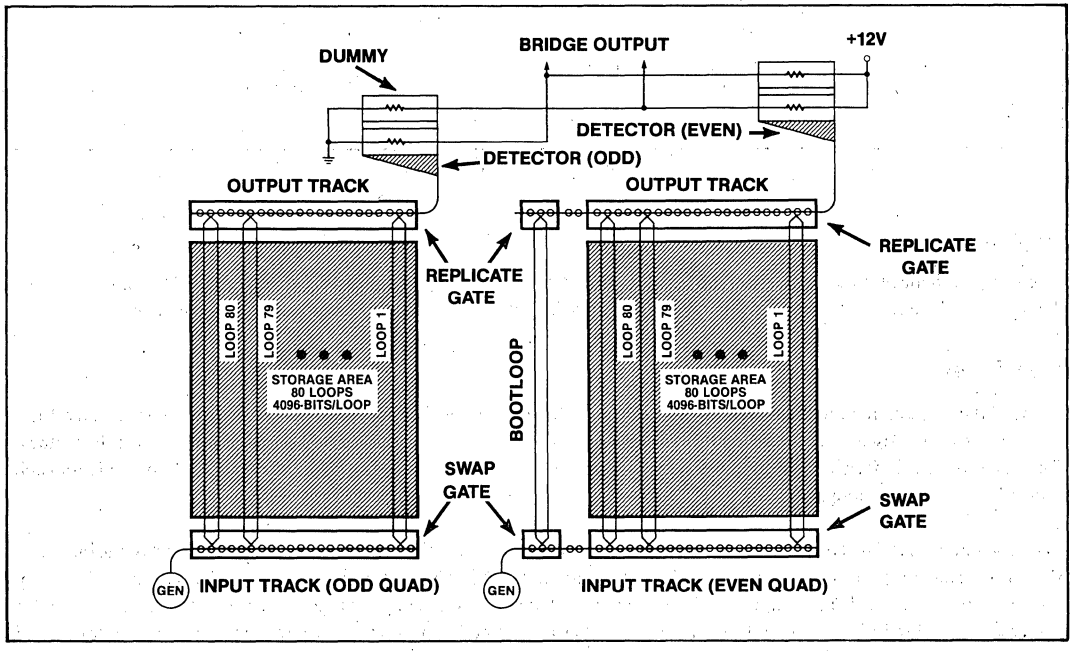


Figure 10. Organization of Bubble Memory (One-Half Chip)

SPECIFIC STRUCTURES OF A MAGNETIC BUBBLE MEMORY

A magnetic bubble memory system consists of a controller and up to eight 1-megabit magnetic bubble subsystems. A minimum system has a controller and one subsystem. The subsystem comprises one magnetic bubble unit in which the data is actually stored, and the peripheral units listed in Table 2 and diagrammed in Figure 11. These circuits are described later in this primer.

Table 2. Components of Intel Bubble Memory System

CONTROLLER	SUBSYSTEM
7220 Bubble Memory Controller (for 1 to 8 subsystems)	Memory 7110 Magnetic Bubble Unit Peripheral Units 7242 Formatter/Sense Amplifier 7230 Current Pulse Generator 7250 Coil Predriver 7254 Drive Transistor Assembly (2 required per subsystem)

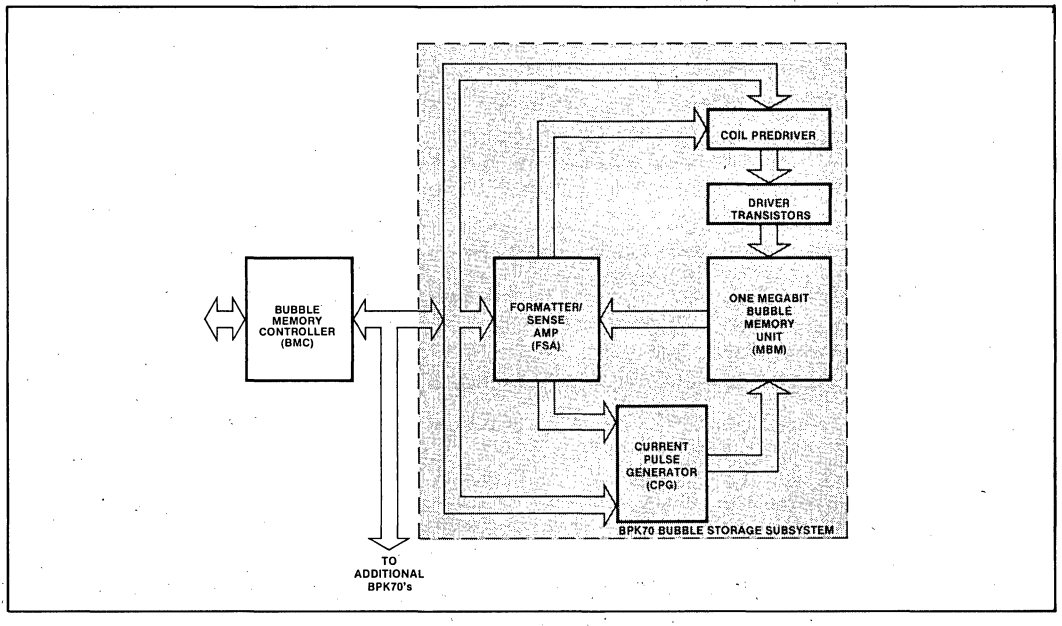


Figure 11. Minimum Magnetic Bubble Memory System, Shaded Portion is Bubble Subsystem

SUPPORT CHIPS

Five semiconductor integrated circuits are necessary to support each bubble chip. These components are described in some detail in the following paragraphs. In addition, each bubble memory system requires a controller, a separate integrated circuit described later.

Formatter/Sense Amplifier (FSA)

Serial data to be stored in or read from the bubble memory passes through the FSA. The FSA keeps track of which loops in the bubble memory are spares, executes the error correction coding and decoding if it is implemented, and shifts data to the bubble memory input tracks or from the output tracks, amplifying the output signals from the memory.

The FSA has a chip-select input, which is normally grounded (permanently enabled). However, each FSA drives the chip-select input of other circuits associated with the same bubble chip, so they are all enabled at the proper time.

Current Pulse Generator (CPG)

All signals except those that control the rotating field originate in the CPG. This device is the source of a current pulse that cuts a new bubble from the seed bubble whenever the FSA has a binary 1 to be stored. Later, when this bubble reaches the loop in which it is to reside, the CPG issues the signal that swaps it with the bubble or non-bubble previously stored in that location of the loop. When data is to be read, the bubble is replicated on the output track by still another signal from the CPG.

Coil Predriver (CPD)

Four digital signals (positive and negative versions of both X and Y waveforms) are sent to the CPD from the controller with appropriate durations and phases to control the rotating field that moves the bubbles in the memory. The CPD combines and inverts these to form eight pulsed outputs that are amplified in a separate transistor package to drive the coils surrounding the bubble chip with a triangular current waveform.

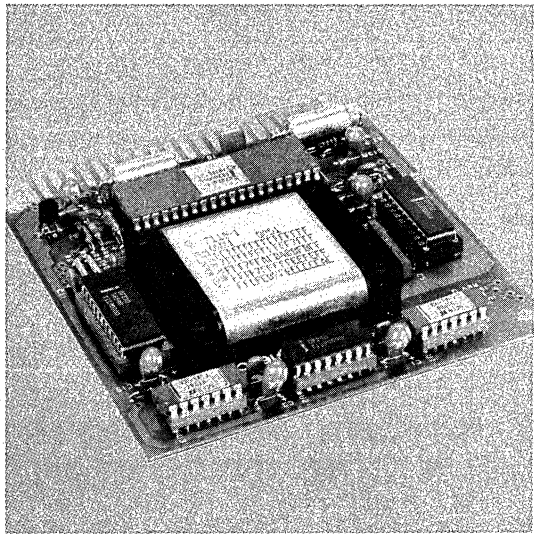


Photo 2. The Minimum Magnetic Bubble Memory System Including Controller

CONTROLLER

The bubble memory controller is the interface between the memory system and the equipment it serves. It converts serial data to parallel and parallel data to serial, and generates all timing signals required by the other support circuits in the bubble memory system. It can control up to eight bubble subsystems, for a total of a megabyte of memory.

Internal storage on the controller includes a first-in-first-out buffer with a capacity of 40 bytes. This buffer stores data to be sent serially to the FSA or just received from the FSA on one side, and data to or from the parallel bus served by the bubble memory on the other. It also serves as a speed matching device between the user at the parallel bus and the FSA which must transfer data to and from the bubble device at exactly the rotating field ratio in each channnel.

GLOSSARY

Bias field—a magnetic field perpendicular to a magnetic thin film that maintains conditions necessary to support formation of magnetic bubbles in the film.

Boot loop—in a magnetic bubble memory with serial/parallel/serial architecture and redundant loops, a special loop containing information that identifies which loops are active and which are inactive, as determined by factory test. This loop also contains the information necessary to synchronize the bubble memory page locations with the controller after power up.

Bubble, magnetic—a cylindrical magnetic domain in a thin film of orthoferrite or garnet. When viewed from above, the cylindrical shape appears spherical, hence the name "bubble." A bubble represents a binary 1 in most magnetic bubble memories.

Chevron—one of many possible shapes for a magnetic pattern deposited on a thin film to steer bubbles in a desired direction. Asymmetric chevrons are used in Intel memories.

Detector—a means of distinguishing bubbles from non-bubbles (1s from 0s) when a word is read from the bubble memory.

Domain, magnetic—a small region of a ferromagnetic substance that contains many similarly oriented atoms, so that the region as a whole is magnetized in that direction.

E²PROM—an acronym for electrically erasable programmable read-only memory, which is a memory component that, though nominally read-only, can accept changes to any work stored in it by electrical means, but at substantially slower speed than that at which stored words are read.

EPROM—an acronym for erasable programmable read-only memory, which is a memory component that, though nominally read-only, can be completely erased, usually by exposure to ultraviolet light, and then reloaded with new information, but at substantially slower speed than that at which stored words are read.

Ferrite—any of several compounds of iron, oxygen, and another metal, with magnetic properties that are useful in certain microwave applications and in computer memories.

Field, magnetic—a region of space in which a magnetic force exists and can be measured.

Garnet—a naturally occurring silicate mineral sometimes used in jewelry. Synthetic garnets with the same crystal structure can be made of oxides of iron and yttrium or one of the rare earths. Garnet is the preferred material for the thin magnetic film in a bubble memory.

Input track—a series of magnetic metal patterns that control the movement of bubbles in a thin film, and thereby lead them from a bubble generator toward one or more storage patterns.

Ion implantation—a process involving accelerators, similar to the machines used by nuclear physicists, for depositing dopants on and just below the surface of an electronic component; used to alter the physical properties of the material.

Latency—a delay between a request to read or write data in a memory and the actual beginning of the operation, imposed by a requirement for the address to move physically (but not necessarily mechanically) to a point where the data transfer can take place.

Magnetization vector—an expression of the magnitude and direction of a magnetic field at a point in space.

Magnetoresistance—a change in electrical resistance due to the presence of a magnetic field.

Major loop—in a magnetic bubble memory, an endless loop containing a bubble generator, a bubble detector, and/or a bubble annihilator, through which data is read or written, and which transfers bubbles to or from one or more minor loops (q.v.) in which they are stored. In some designs the major loop is not endless, and all bubbles not transferred out of it collapse when they reach the end. In these cases the major loop becomes an input or output track (q.v.).

Minor loop—in a magnetic bubble memory, an endless loop in which bubbles are stored, having been transferred into it from a major loop or input track (q.v.) and accessible by transfer into a major loop or output track (q.v.).

Non-Volatility—a property of some memory technologies that retains the integrity of stored data when power is turned off.

Orthoferrite—one of several oxides of iron and either yttrium or a rare earth. The molecular structure is simpler than that of garnet (q.v.). Orthoferrites were the first materials used for the thin magnetic film in experimental bubble memories, but have yielded to garnets, which have more desirable properties—notably ease of preparation as thin films with the necessary magnetic characteristics.

Output track—a series of magnetic metal patterns that control the movement of bubbles in a thin film, and thereby lead them from one or more storage patterns toward a bubble detector.

Permalloy—an easily magnetized and demagnetized alloy of nickel and iron.

PROM—acronym for programmable read-only memory—a read-only memory whose content is loaded by the user after delivery, as opposed to read-only memories whose content is fixed during manufacture. Once loaded, the data in a PROM is not alterable.

Pseudo-random access—a property of some memory technologies in which the time of access to blocks of stored data is largely (but not necessarily entirely) independent of the position of the block in the storage medium, but in which the time of access to bits, words or other entities depends on the position of that entity within the block.

Random access—a property of some memory technologies in which the time of access to any stored bit, word, or other entity is wholly independent of that entity's position in the storage medium.

Saturation—a state of magnetization of a material by a field such that, if the field is increased, the magnetization of the material does not increase and the magnetic flux density increases in proportion to the field (having increased much more rapidly in weaker fields).

Seed—a permanent bubble in a magnetic bubble memory, from which other bubbles are cut to represent stored binary 1s.

Serial access—a property of some memory technologies in which the time of access to any stored bit, word, or other entity depends strongly on that entity's position in the storage medium.

Thin film—any film of material deposited on a suitable substrate to take advantage of the material's special properties when dispersed as a film. Thickness ranges usually from about 10-9 to 10-6 meter, and occasionally to 10-5 meter or more, as in bubble memories.

T-I bar—one of several possible shapes for a magnetic pattern deposited on a thin film to steer bubbles in a desired direction, consisting of shapes like the letter T and the letter I alternately along a track. This pattern was used extensively in early bubble memory designs, but is no longer generally employed.

intel

APPLICATION NOTE

AP-119

December 1982

Microprocessor Interface for the BPK 72

Paul Wells
Application Engineer

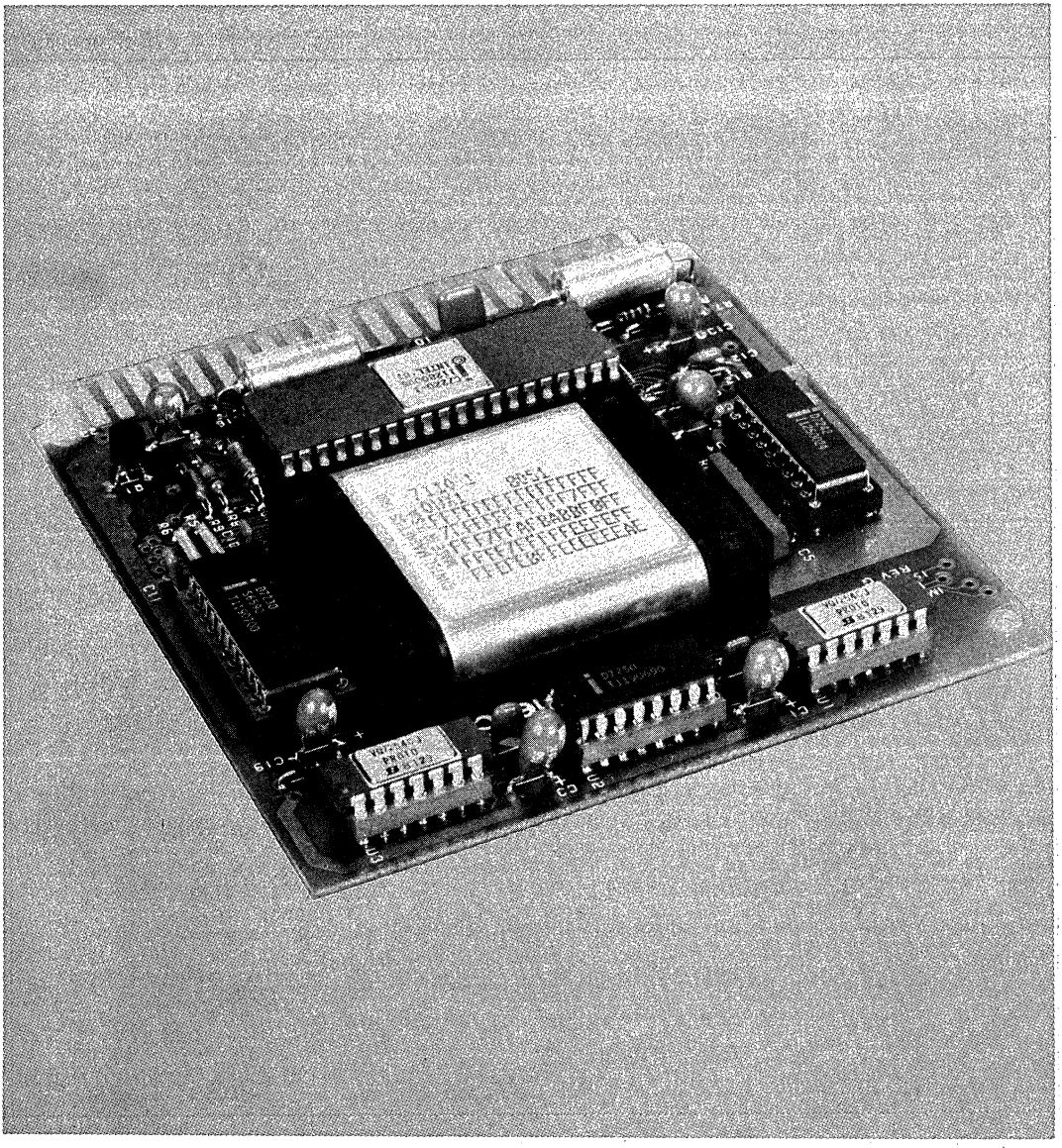
ORDER NUMBER: 210367-002

INTRODUCTION

To date, a major obstacle in the implementation of bubble memories in systems has been the inherently complex control requirements imposed by the bubble memory devices themselves. With the advent of Intel's BPK 72 bubble memory prototype kit, a design engineer can immediately realize the benefits of non-volatility, form factor, density and reliability without the complex control concerns. This application note provides additional background on the operating characteristics of the BPK 72 and is intended to further ease the design effort required in the implementation of bubble memory systems.

OVERVIEW

This application note provides an example of Bubble Memory system implementation using the BPK 72 and an Intel 8086 microprocessor. Before looking at this example, some explanation is necessary as to how this implementation was attained and how a user can take advantage of the principles involved.



As an introduction, the basic architecture of the BPK 72 is reviewed followed by an explanation of the operating characteristics of the BPK 72 kit as a whole and of the 7220 Bubble Memory Controller. Once the building blocks are in place, a detailed account of the implementation of a bubble memory kit is offered. The final section, which involves the actual implementation of the BPK 72 and an SDK-86, completes the application note.

BUBBLE SYSTEM OVERVIEW

A block diagram of the Intel Magnetics 128K-byte system is shown in Figure 1. The support circuitry used with one 7110 magnetic bubble memory (MBM) in the BPK 72 kit consists of the following integrated circuit components: one 7250 Coil Predriver, two 7254 Quad VMOS Drive Transistor packs, one 7230 Current Pulse Generator, and one 7242 Formatter/Sense Amplifier. The 7220 Bubble Memory Controller (BMC) completes the basic system.

The 7250 and the two 7254s supply the drive currents for the in-plane rotating magnetic field (X and Y coils) that move the magnetic bubbles within the MBM. The 7230 supplies the current pulses that generate the magnetic bubbles and transfer the bubbles into and out of the storage loops of the MBM.

The 7242 accepts signals from the bubble detectors in the MBM during read operations, buffers the signals and performs data formatting tasks that include the transparent handling of bootloop information. During write operations, the 7242 enables the current pulses of the 7230 that cause the bubbles to be generated in the 7110 MBM. Automatic error detection and correction of the data can be performed by the 7242.

The 7220 provides the user interface, performs serial-to-parallel and parallel-to-serial data conversions, and generates all timing signals necessary for the proper operation of the MBM support circuitry.

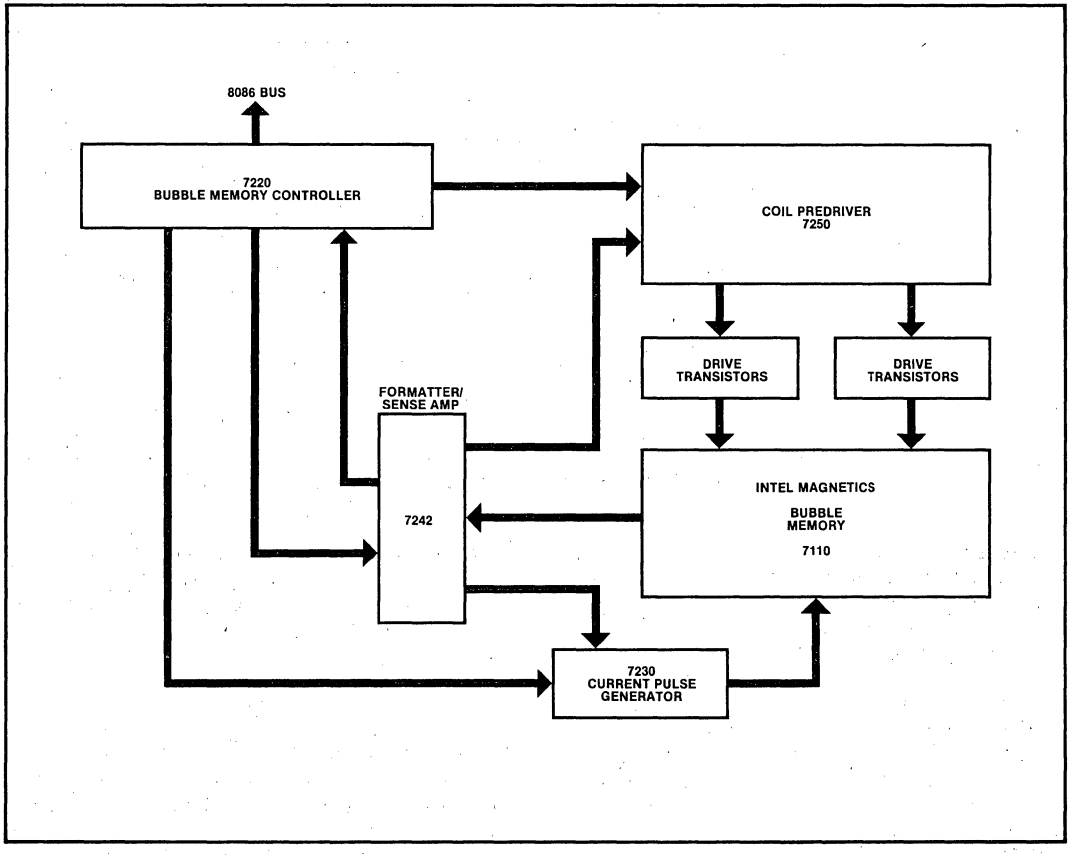


Figure 1. Block Diagram of the 128K Byte Magnetic Bubble Memory System

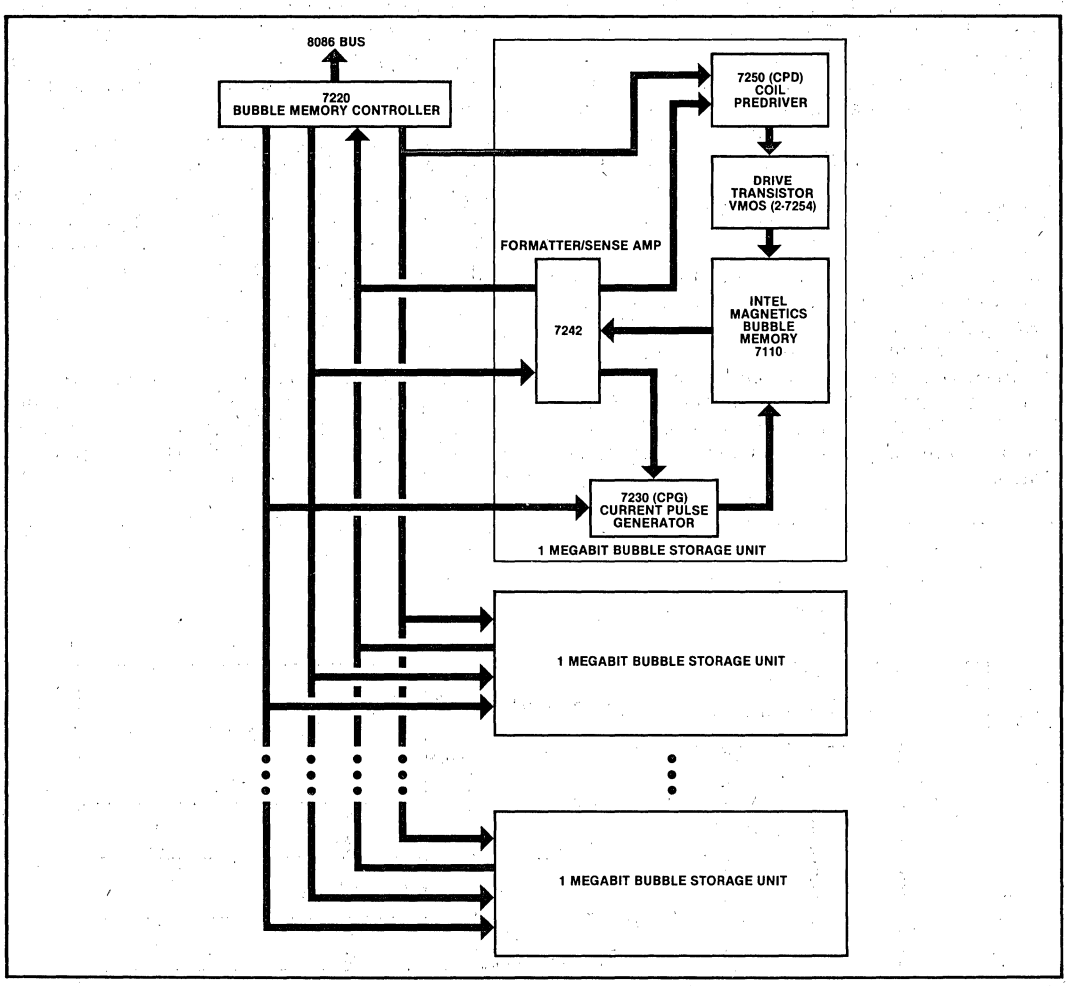


Figure 2. Bubble Memory System Expansion up to One Megabyte

Figure 2 shows how larger systems can be built from the basic components. A Bubble Storage Unit consists of one 128K-byte MBM and the five support chips shown. The components needed for one MBM cell are available as the BPK 70 kit. Larger systems can be constructed from the components supplied with one BPK 72 kit (which includes the 7220 controller) and one or more BPK 70 kits. For example, a one megabyte system can be assembled from one BPK 72 kit and seven BPK 70 kits. No additional TTL parts are required when building multibubble systems with up to eight MBMs.

One 7220 is capable of controlling up to eight Bubble Storage Units simultaneously. Larger systems can be configured with multiple 7220's and additional Bubble Storage Units.

Functional Organization of the 7110 Bubble Memory

The Intel Magnetics 7110 Bubble Memory utilizes a "major track/minor loop" architecture. With this architecture, if a binary 1 is to be written, a "seed bubble," always present in the 7110, is split in two. One bubble remains at the generator as the

seed, and the other is propagated down the input (major) track. If a 0 is to be written, the seed bubble is not duplicated. The data generated is sent down the input track, in serial, until it is aligned with the "swap" gates at the minor loops of the device. The new data is then swapped into the minor loops in parallel at the same time the old data is swapped out to the major track.

To read data from the 7110, data is rotated in the minor loops until it is positioned at the "replicate" gates opposite the output track. On receipt of a replicate signal, the data in the minor loops is duplicated by splitting the bubbles. The original data remains in the minor loops, and the duplicate data is clocked down the output track where the detector elements of the bubble memory operate to transform the presence or absence of a bubble into small electrical signals that are converted into digital '1' and '0' signals in the 7242 FSA.

With the 7110, the process of reading data from the minor loops by simultaneously splitting all of the bubbles in a page is known as "block replicate." The advantage of the block replicate architecture is that the data currently stored in the minor loops is not compromised during a read operation; the data to be read never leaves the minor loops. This architecture can be contrasted with earlier architectures that required the data to leave the minor loops, be detected and then returned to the minor loops. In the event of a power failure, bubble systems not utilizing the block replicate architecture could suffer a loss of data during a read operation; the data being sensed would not be returned from the major loop to the minor loops.

With the 7110 MBM, there are 2048 positions for the data within a minor loop. To move the bubbles in the MBM, a magnetic field is induced and rotated in the plane of the 7110. As the field is rotated 360 degrees, every bubble is moved ahead one position, and all of the bubbles maintain the same position relative to one another. All of the bubbles in similar positions in the loops are referred to as a "page."

By way of illustration, suppose the bubble is made of five minor loops (a,b,c,d,e) capable of holding nine pages of data (Table 1). During four 360 degree "rotations" of the in-plane magnetic field, the nine pages of data shift four positions (1.1, 1.2, 1.3, 1.4).

Table 1. 7110 Loop Operation

abcde	abcde	abcde	abcde
00000	00011	00000	00000
00011	00000	00000	11111
00000	00000	11111	00000
00000	11111	00000	00000
11111	00000	00000	00000
00000	00000	00000	10110*
00000	00000	10110*	00000
00000	10110*	00000	00011
10110*	00000	00011	00000
1.1	1.2	1.3	1.4

* = page zero

The 7110 MBM actually contains 320 minor loops, of which 272 must be good. The additional 48 loops provide 15% redundancy. This redundancy factor allows some of the loops in the 7110 to be bad while maintaining a completely functional one megabit device. A map of the good and bad loops is placed on the label of the 7110 and is also

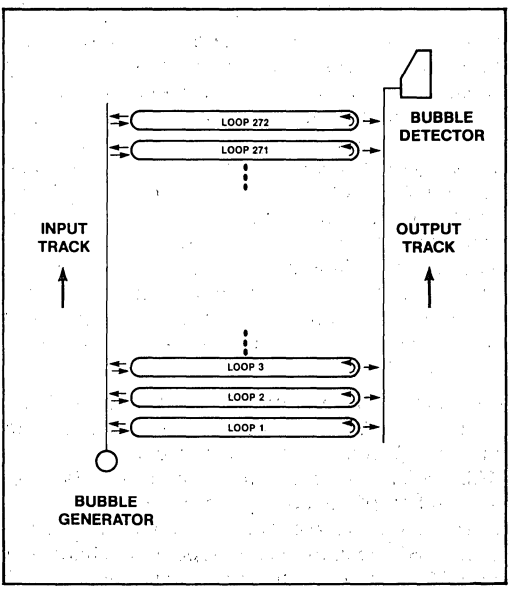


Figure 3. Functional Organization of the 7110

encoded and placed in the boot loop of the device as it is tested. This map, the bootloop, consists of forty bytes of data. Each good loop in the 7110 is represented by a one, each bad loop by a zero. When the system is initialized, the 7220 BMC reads the bootloop from the 7110 and decodes it. The bootloop is then automatically placed in the bootloop register of the 7242. The bootloop register serves as a working 'map' of the 7110 for read and write operations.

With the pages of data rotating around the minor loops, there must be a mechanism to orient the device and to assign a starting address to a page. The mechanism used to identify page zero involves the bootloop that resides on the 7110. Page zero (or address zero) is defined as the position of the 7110 after the bootloop has been read by the 7220 controller. Thus, each time the host CPU sends an "initialize" command, the bootloop is read by the 7220, and the 7110 is queued at page zero. From this point, any desired page in the bubble can be obtained by the controller.

Data Flow Within the Bubble Memory System

To better understand the relationship between the 7110 MBM and its support circuitry, the data flow within the bubble system during a read operation is examined. During the read operation, bubbles from the storage loops are replicated onto an output track and then moved to a detector within the MBM. All movements within the MBM occur under the influence of a rotating magnetic field; the number of rotations and the rotation timing are under the control of the 7220 BMC. The detector outputs a differential voltage according to whether a bubble is present or absent in the detector at any given time. This voltage is fed to the detector input of the 7242 Formatter/Sense Amplifier (FSA).

The data path between the 7110 MBM and the 7242 FSA consists of two channels (channel A and channel B) connected to the two halves of the MBM. When data is written, the bit stream is divided with half of the data going to each side of the MBM. During a read operation, data from each half of the MBM goes to the corresponding channel of the FSA. In the FSA, the sense amplifier performs a sample-and-hold function on the detector input data, and produces a digital 0 or 1. The resulting data bit is then paired with the corresponding bit in the FSA bootloop register.

If an incoming data bit is found to be from a good loop (a corresponding "1" in the FSA bootloop register), it is stored in the FSA FIFO; otherwise, it is ignored. This process continues until both FSA FIFOs (channels A and B) are filled with 256 bits. Error detection and correction, if enabled by the user, is applied to each block of 256 bits at this point. If error correction is not enabled, 272 bits of data can be buffered in each FIFO.

As data leaves the 7242 FSA, the bit patterns buffered in each of the FSA FIFOs is interleaved and sent to the 7220 BMC in the form of a serial bit stream via a one-line bidirectional data bus (DIO line). In the 7220 BMC, the data undergoes a serial-to-parallel conversion and is assembled into bytes that are buffered in the 7220 FIFO. It is from this FIFO that the data is written onto the user interface.

COMMUNICATING WITH THE 7220

The CPU views the 7220 BMC as two input/output ports on the bus. When the least-significant bit of the address line is active (A0 = 1), the command/status port is selected. When the least-significant bit of the address line is inactive (A0 = 0), the bidirectional data port is selected. In order to define the operations on these ports, it is necessary to understand something of the internal organization of the 7220 Bubble Memory Controller.

For simplicity, the user need only view the 7220 as containing a 40-byte FIFO and a collection of 8-bit registers. The FIFO is a buffer through which data passes on its way from the 7242 Formatter/Sense Amplifier (FSA) to the user, or from the user to the FSAs. The primary purpose of the FIFO is to reconcile differences in timing requirements between the user interface to the 7220 controller and the controller interface to the FSAs.

The six 8-bit registers internal to the 7220 are loaded by the user prior to any operation of the bubble system and contain information regarding the operating mode of the 7220. Loading the 7220 registers before any commands are sent is similar to passing parameters to a subroutine prior to invocation, hence, the registers are often referred to as "parametric registers."

Data transferred between the CPU and the 7220 FIFO and parametric registers takes place over an 8-bit data port. The choice as to whether the data is destined for the FIFO or the parametric registers, however, is made through the command/status port. In one case, the actual commands that cause some operation to take place, such as a read or write, consist of a 4-bit code sent by the CPU to select one of 16 possible commands. This 4-bit code occupies the low-order nibble (bits 0, 1, 2, and 3) of the command byte. The command byte must also have bit 4 set to indicate to the 7220 that a command is being sent. In the

second case, another 4-bit code on the command port (bits 0, 1, 2, and 3) is used to select either one of the parametric registers or the 7220 FIFO. As shown in Table 2, if bit 4 of the command byte is set to zero, the value of the low-order nibble is taken to be a pointer value that specifies a parametric register or the 7220 FIFO. This pointer is referred to as the "Register Address Counter" (RAC).

Table 2. Command Port Function

FUNCTION	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Command	0	0	0	1	C	C	C	C
RAC	0	0	0	0	R	R	R	R

RAC values that may be sent out on the command port and the corresponding register names are illustrated in Table 3. The RAC points to, or selects, six unique registers and the 7220 FIFO. Once a RAC value is sent by the CPU to the 7220 via the command port, the next read or write operation to the data port transmits data to or receives data from the register addressed. Notice that the six registers have values that are in ascending order starting at 0AH and that the FIFO has a value of 0.

The reason for this ordering is due to the auto-incrementing feature of the RAC; once the first register is selected, each subsequent byte of data on the data port causes the RAC to be automatically incremented and to point to the next register in the sequence. Once the most-significant byte of the Address Register has been loaded, the RAC value automatically rolls over from 0FH to 0 and points to the 7220 FIFO. The system is now in position to transfer data to or from the FIFO without the user code explicitly pointing to the FIFO.

Table 3. Register Address Counter Assignments

Register Name	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Read/ Write
Utility Register	0	0	0	0	1	0	1	0	R/W
Block Length Register (LSB)	0	0	0	0	1	0	1	1	W
Block Length Register (MSB)	0	0	0	0	1	1	0	0	W
Enable Register	0	0	0	0	1	1	0	1	W
Address Register (LSB)	0	0	0	0	1	1	1	0	R/W
Address Register (MSB)	0	0	0	0	1	1	1	1	R/W
7220 FIFO	0	0	0	0	0	0	0	0	R/W

Once the FIFO has been selected, the RAC stops incrementing and continues to point to the FIFO until changed by the user software. This sequence minimizes the number of instructions necessary for a given transaction and aids in establishing a protocol to ensure that all of the necessary information is sent to the controller. The user, however, is not bound to follow this automatic sequence. Each parametric register may be selected and loaded in any order; specific registers may be updated where needed, but in each case, the host software must explicitly name the register to be loaded. Until a user is familiar with the bubble system, it is recommended that the auto-incrementing feature be used.

It is important to remember that once a command has been given to the 7220 BMC, the parametric registers must not be updated until the Status byte indicates that the operation is complete. The parametric registers are, in effect, working registers for the controller during the execution of a command. For example, during a Read or Write operation, the Block Length Register, which contains the terminal page count for the operation, is decremented by the 7220. Similarly, the Starting Address Register, which initially contains the starting page for an operation, is incremented by the controller as each page is transferred. Attempting to modify these registers during the operation of a command causes the block count and address to be incorrect.

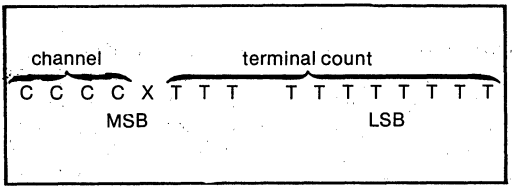
Addressing the Bubble Memory System

One of the interesting aspects of the Intel Bubble Memory System is its inherent addressing flexibility. The user may treat a 7220 BMC with eight

bubbles as a collection of 16K pages of 64 bytes each (addressing each bubble in turn) or as collection of 2K pages of 512 bytes each (addressing eight bubbles in parallel). Of course, there are a variety of configurations in between these two extremes, each dictated by the user's need for speed, power consumption, address space, and cost. Control over the configuration is achieved at run time via two of the parametric registers: the Block Length Register and the Starting Address Register.

The Block Length Register (BLR) is a 16-bit value divided into two fields: the "terminal count" field and the "channel" field. The bit configuration for the BLR is as follows:

Table 4. Block Length Register



The "terminal count" field ranges over eleven bits and defines the total number of pages requested for a read or write operation. With eleven bits in the field, a user may request from one to 2048 pages be transferred (eleven bits of zero indicate a 2048-page transfer). The width of the page is effectively defined in the "channel" field. This field specifies the number of FSA channels that are to be addressed. Recalling that each 7242 FSA has two channels to communicate with one 7110 bubble memory, the legal combinations in this field address one channel (one half of a 7110), two, four, eight, or 16 channels. These combinations translate into page sizes of 32, 64, 128, 256, or 512 bytes, respectively. (The one-channel mode of operation is usually reserved for diagnostic purposes, and examples of its use will be illustrated later.)

Table 5 shows the relationship between the "channel" field and the number of FSA channels selected. Notice that the channel field bits are encoded. A value of "0001" binary selects two FSA channels: 0 and 1.

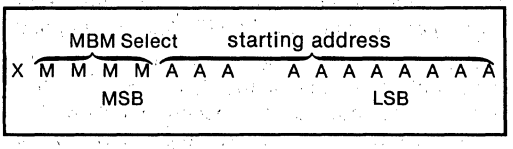
Table 5. FSA Channel Select

	Channel field (BLR MSB bits 7, 6, 5,4)				
	0000	0001	0010	0100	1000
Number of channels selected:	0	0,1	0,1,2,3	0 to 7	0 to F

Thus, a BLR value of "0001" in the high-order four bits selects one bubble through channels 0 and 1 Similarly, a BLR value of "0010" selects two bubbles in parallel with a page size of 128 bytes. This, however, is not the complete story. For example, a value of "0100" in the BLR selects four bubbles in parallel through channels 0 to 7. Suppose, that there are eight bubbles in the system and that the user desires to arrange the eight bubbles as two sets of four. The mechanism to communicate through channels 0 to 7 and channels 8 to F resides with the Address Register (AR).

The Address Register contains a 16-bit value divided into two fields: a "starting address" field of eleven bits and a "magnetic bubble memory (MBM) select" field of four bits.

Table 6. Starting Address Register



The eleven bits in the starting address field of the AR are set by the user to indicate to the 7220 BMC on which page of a bubble's 2048 pages the transfer is to start. For example, if a read operation is to start at page 1125 and is to continue for 16 pages, the starting address field contains 1125, and a value of 16 is placed in the terminal count field of the BLR. After each page is transferred, the starting address field is incremented and the terminal count is decremented by the controller.

Continuing with the example of two banks of four bubbles, notice in Table 7 that the MBM select field is needed to switch between the two banks. A value of "0000" in bits 3, 4, 5, and 6 of the high-order byte of the address register selects bank 0 or FSA channels 0 through 7; a value of "0001" selects bank 1 or FSA channels 8 through F. Each bank contains 2048 pages of 256 bytes.

To operate eight bubbles serially, a user needs only to specify a value of "0001" once in the channel field of the BLR and to begin with a value of "0000" in the MBM select field. As page 2048 is written in the first bubble, the AR, managed by the 7220 controller, rolls over to 0 and updates the MBM select field with no additional bit manipulation. In this case, the bubble system appears as 16K pages of 64 bytes each. Power consumption is one-eighth of that consumed by operating eight bubbles in parallel. However, the data rate is limited to the data rate of one bubble.

Table 7. FSA Channel Select/MBM Select

MBM SELECT AR MSB BITS (6, 5, 4, 3)	"CHANNEL FIELD" (BLR MSB bits 7, 6, 5, 4)				
	0000	0001	0010	0100	1000
0 0 0 0	0	0,1	0,1,2,3	0 to 7	0 to F
0 0 0 1	1	2,3	4,5,6,7	8 to F	
0 0 1 0	2	4,5	8,9,A,B		
0 0 1 1	3	6,7	C,D,E,F		
0 1 0 0	4	8,9			
0 1 0 1	5	A,B			
0 1 1 0	6	C,D			
0 1 1 1	7	E,F			
1 0 0 0	8				
1 0 0 1	9				
1 0 1 0	A				
1 0 1 1	B				
1 1 0 0	C				
1 1 0 1	D				
1 1 1 0	E				
1 1 1 1	F				

The Enable Register

The Enable register is the parametric register that defines the various modes of operation of the 7220 controller. The data transfer mode (polled, interrupt driven, or DMA operation) is selected by setting the appropriate bit in this register. Likewise, the type of error correction to be applied to the data is selected, based on the bits selected in this register.

While the function of each of the enable register fields is described in the BPK 72 manual, some of the finer points and implications are detailed here.

Note that it is possible to completely change the operating characteristics of the bubble system through software control. A system can go from the DMA mode with error correction enabled to a system operating in polled I/O with no error correction enabled by altering the value of the Enable register. Though most implementations will not take advantage of this degree of flexibility, there are cases where the Enable register is modified during system operation. For example, the normal interrupt and MFBTR bits can be modified between operations to change interrupt and read data rates, respectively. (If the error correction mode is changed, the CPU must issue an Initialize command to the 7220 controller).

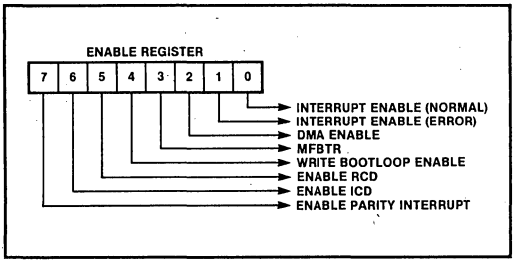


Figure 4. Enable Register Definition

The interrupt capabilties of the 7220 are reflected in the NORMAL, PARITY and ERROR INTERRUPT bits of the ENABLE register byte. The 7220 controller is capable of issuing interrupts to a CPU at the normal completion of an operation, if a parity error is encountered between the 7220 controller and the CPU, or if a data transfer error is found by the 7242 FSA. Any (or all) of these conditions are selected via the Enable register byte, and any resultant interrupts are sent to the CPU via a single INT line. At this point, the software must examine the status register to determine the cause of the interrupt. (An additional interrupt, the FIFO half-full interrupt, is issued on the DRQ pin and is not controlled by the Enable byte).

One of the more difficult aspects of the ENABLE register byte to understand is the operation of the ERROR INTERRUPT bit (bit 2). This bit normally is not used alone, but in conjunction with the ENABLE RCD and ENABLE ICD bits of this register. These three bits form combinations that gate selected 7242 error conditions to the CPU. For example, if, while operating under error correction, a user does not wish to be bothered by an interrupt that indicates an error has been corrected automatically by the system, a specific pattern of these three bits would be selected (100 or 010 from Table 8). If the user wishes to be notified of all errors, another pattern would be selected (011 or 101).

Table 8. Error Correction Combinations

Enable ICD	Enable RCD	Interrupt Enable (ERROR)	Interrupt Action
0	0	0	No interrupts due to errors
0	0	1	Interrupt on TE only
0	1	0	Interrupt on UCE or TE
0	1	1	Interrupt on UCE, CE or TE
1	0	0	Interrupt on UCE or TE
1	0	1	Interrupt on UCE, CE or TE
1	1	0	Not used
1	1	1	Not used

The purpose of the ERROR INTERRUPT bit is not to enable or disable error interrupts, but rather to aid in selecting the type of error interrupt received by the CPU. If any type of error correction is selected, interrupts are enabled automatically.

The ENABLE RCD (read corrected data) bit causes the error correction algorithm to be applied to the data being transferred from the 7110 MBM in an almost transparent manner. The RCD bit allows the 7220 controller to send its own commands to the 7242 FSA. These commands cause the FSA to automatically correct and transfer to the controller, any data that is found to be in error and that is considered correctable.

With only the RCD bit on, no interrupt is generated if a correctable error is found. However, the user is informed that a correctable error was encountered and corrected during the data transfer via the 7220 status byte at the end of the operation. Uncorrectable and timing errors cause an interrupt to which the CPU must respond. With both the RCD bit and ERROR INTERRUPT bit on, the CPU is notified via an interrupt whenever a correctable, uncorrectable or timing error is encountered.

The RCD mode of operation is suitable for transfers where a GO/NO GO termination is sufficient. For example, when loading executable code from the bubble to RAM, it is necessary to know that the transfer was good (with errors corrected) or aborted due to an uncorrectable error.

A retry of an uncorrectable page of data is accomplished by sending another Read command without modifying the parametric registers. It may be the case that the errors encountered were soft (read) errors that may not be present on a retry. Thus, what may have been detected as an uncorrectable error, may become a correctable error (or simply vanish) on a subsequent read of the offending page. In this case, the error correction ability of the system corrects the errors automatically without additional user intervention.

The advantage of the RCD mode of operation is that error correction can be applied transparently to the CPU except for uncorrectable conditions. The disadvantage is that a page of uncorrectable data is passed to the controller before the interrupt is sent. The software must have the ability to clear the 7220 FIFO prior to rereading the offending page from the bubble.

If a given page continues to show up as having a correctable error after a number of retrys, it is up to the user's protocol to determine the action to be taken. One protocol suitable for handling errors involves "scrubbing" the data. Suppose a page appears with an error and, on retry, the error is still present. If the error is correctable, the data should be corrected and written back to the bubble and then read back into RAM. The probability of encountering an uncorrectable error after the first retry is 1 in 10^{16}. Data scrubbing after one retry maintains this level of reliability.

The ENABLE ICD (internally correct data) bit also enables the error correction capability of the bubble system, but allows a slightly different interaction between the 7220 controller and the 7242 FSA than defined for the RCD mode. Error interrupt conditions are the same as defined for RCD operation. With the ICD bit on, correctable errors are handled automatically, but the operation halts for uncorrectable or timing errors. With both the ICD and ERROR INTERRUPT bits on, the operation halts for correctable, uncorrectable or timing errors. The ICD mode differs from the RCD mode in that when an operation halts due to an error, the offending page is held in the 7242 FSA and is not automatically transferred to the 7220 FIFO. Though the difference is subtle, the ICD mode of operation allows more flexibility in error logging and recovery. With data held in the 7242, the number of the bad page can be read for logging purposes, and the data can be recycled through the error correction network or reread from the bubble repeatedly. When the CPU is interrupted due to an error in the ICD mode, the user must look at the 7220 status byte to determine the type of error encountered. If the error is correctable, the user's software sends a Read Corrected Data command (0CH) to the controller. This command causes the controller to issue it's own commands to the 7242 to correct the error and to transfer the data to the 7220 FIFO. (Recall this action is done automatically when the RCD mode is selected; uncorrectable errors can be handled as described above).

As an example of how the ICD mode can be utilized, suppose that during a data transfer in the RCD mode, a correctable error consistently occurs. The

error, of course, is automatically handled by the 7242, and the only indication that an error had been corrected is through the status byte at the end of the transfer. There is no information as to how many or in what page the error or errors appear. One way to diagnose the problem is to reread the entire data block in the ICD mode with the ERROR INTERRUPT bit on. The transfer stops at the appearance of any error, and the data remains in the 7242. The page number of the error can be found by reading the Address Register since this register is incremented automatically after each page is read if no error is detected.

The user should then issue an RCD command to the 7220 to allow the page to be corrected and transferred to the 7220. Once the transfer is complete, the enable register again is changed to disable all error correction, and the 7220 is reinitialized. The entire block is read again and compared with the corrected version. (Error correction bits are appended to the data and can be ignored.) If a bad loop is suspected, the bad loop location could be calculated and the bootloop modified.

It is unlikely that repeated correctable errors are sufficient motivation to modify the bootloop. Repeated uncorrectable errors, however, at the same location, might be sufficient reason. Note that modifying the bootloop is an extreme measure and should only be performed as a last resort and only if justified by test data.

The Status Register

The 7220's 8-bit Status register is accessed by reading the Command port (A0 = 1). This register provides information regarding error conditions, the termination of commands, and the readiness of the controller to transfer data or accept new commands.

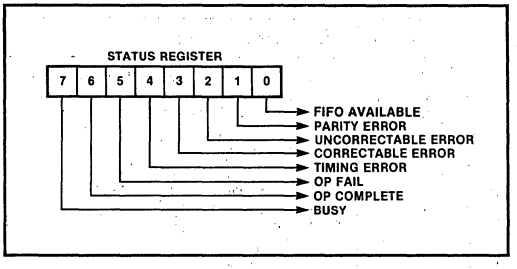


Figure 5. Status Register Definition

Values for the Uncorrectable Error and Correctable Error fields are generated when error correction is utilized as previously defined. The PARITY ERROR bit is set when a parity error is encountered on data sent to the controller on the D_0-D_7 lines. The TIMING ERROR bit is set for a number of conditions. The most frequent cause of a timing error is when the CPU fails to keep up with the rate at which the controller is filling or emptying the FIFO (an overflow or underflow condition). With one bubble in the system and the MFBTR bit of the Enable byte set to one, the controller moves data to or from the FIFO at a rate of about one byte every 80 microseconds. With eight bubbles operating in parallel, the rate is about one byte every 10 microseconds. (With the MFBTR bit set to 0, the data rate on a one page transfer or the last page of a multipage transfer is four times these rates.) Once a Read or Write command is issued, if the CPU cannot meet these transfer requirements, a timing error results.

Another way in which a timing error occurs is when the proper number of bits is not set in the bootloop register of the 7242 FSA. The 7242 must have 272 loops active to operate properly (270 with error correction enabled). If a mistake is made either when the bootloop of the 7110 is written or if the bootloop register is loaded incorrectly from RAM by the user, a timing error results. A timing error also occurs if the Write Bootloop command is issued to the 7220 controller and the WRITE BOOTLOOP ENABLE bit of the Enable byte is not on. Finally, a timing error is generated if the bootloop synch code is not found when a Read Bootloop or Initialize command is issued.

The OP FAIL and OP COMPLETE bits of the status register simply indicate the state of an operation after a command is executed. If an operation fails (OP FAIL = 1), the cause can be determined by looking at the other error bits of the status byte. When an operation (command) terminates successfully, the OP COMPLETE bit is set, and the status register shows a 40H.

The FIFO AVAILABLE bit of the status byte is more complex than the other bits since its meaning can change depending on the type of operation being performed as outlined below.

From an operational point of view, the FIFO AVAILABLE bit acts as a gate for the FIFO handling software. During a write operation, if the FIFO bit is set (1), there is room for more data; if the FIFO bit is clear (0), the FIFO is full. During a read operation, if the FIFO bit is set, data has been placed in the FIFO by the controller; if it is clear, the FIFO is empty.

Table 9. FIFO Available Bit Semantics

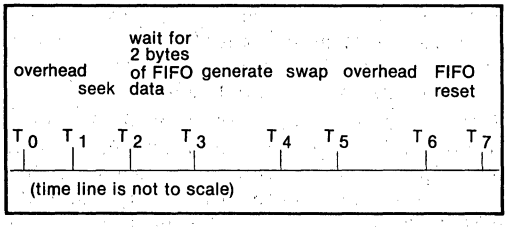
FIFO AVAIL BIT	BUSY = 1 & writing	BUSY = 1 & reading	BUSY = 0 & reading
1	room for data	data avail.	data avail.
0	no room for data	no data	no data

Note that it is possible to complete an operation with data still remaining in the FIFO (indicated by a 41H status value). This condition is quite legal; it is up to the software to remove the data or to issue a FIFO RESET command.

The BUSY bit indicates when the controller is in the process of executing a command. When a command is sent, the BUSY bit goes active within a few microseconds after the command is received and remains active until the operation either completes or fails. It is important to note that the BUSY bit remains active until all other bits in the status byte have been set. Thus it is possible to see logically-exclusive conditions such as BUSY and OP COMPLETE at the same time. The key to interpreting the status byte is to consider the status byte valid only after the BUSY bit returns to an inactive level. The single exception to this rule is the FIFO AVAILABLE bit.

The action of the controller during a write operation is one of the more complex sequences and serves as a good illustration of the behavior of the BUSY and FIFO AVAILABLE bits. Suppose a Write command is sent to transfer an arbitrary number of pages. Table 10 shows the activity of the controller at various steps in the sequence.

Table 10. Stages of a Write Command



Before the Write command is sent, the FIFO is in a general-purpose mode and remains in this mode until T_2. When the command is sent at T_0, the BUSY bit is low and, in fact, the BUSY bit must be low in order for the controller to accept a new command (except Abort). Sometime between T_0 and T_1, the BUSY bit goes high. Thus, between T_1 and T_2, the status byte will be 80H.

At T_2, the FIFO is internally placed in the "write mode," and FIFO AVAILABLE changes meaning from "FIFO has data" to "FIFO has room". For proper operation, the FIFO must be empty prior to issuing the WRITE command. This condition can be guaranteed by using the FIFO Reset command. Assuming the FIFO is empty, at T_2 the status byte changes from 80H to 81H. The status byte remains at 81H until T_6 (unless the CPU is able to fill the FIFO in which case, the FIFO AVAILABLE bit toggles between 0 and 1).

At T_7 (the completion of the command), the status byte should be 40H if the CPU did not load data between T_6 and T_7. If data was loaded during this interval, the status value is 41H.

Notice that if the FIFO contains data when the Write command is sent, the CPU can, by mistake, overflow the FIFO during the "seek" portion of the command. This condition results from the FIFO AVAILABLE bit being a "1" due to data present in the FIFO, not because there is room in the FIFO. While the following diagnostic routines take advantage of the "preloading" ability of the FIFO, the examples of operational software at the end of this application note do not preload the FIFO.

7220 Commands

The 7220 command set consists of 16 commands identified by a 4-bit command code. The function of most of the commands is obvious from the command name (e.g., Initialize, Abort, Read, Write). These commands are adequately described in the BPK 72 manual. There are, however, some commands and protocols that merit additional discussion (specific examples are covered later in this document).

Table 11. 7220 Commands

D3	D2	D2	D1	Command Name
0	0	0	0	Write Bootloop Register Masked
0	0	0	1	Initialize
0	0	1	0	Read Bubble Data
0	0	1	1	Write Bubble Data
0	1	0	0	Read Seek
0	1	0	1	Read Bootloop Register
0	1	1	0	Write Bootloop Register
0	1	1	1	Write Bootloop
1	0	0	0	Read FSA Status
1	0	0	1	Abort
1	0	1	0	Write Seek
1	0	1	1	Read Bootloop
1	1	0	0	Read Corrected Data
1	1	0	1	Reset FIFO
1	1	1	0	MBM Purge
1	1	1	1	Software Reset

In general, all commands sent to the 7220 controller must be preceded by the setting of the parametric registers. While there are some exceptions as with the Abort command, it is usually necessary to supply operating information to the controller via the parametric registers prior to issuing any command. Since many initial problems stem from failing to load the registers prior to issuing commands, the user software should never assume that the regsiters contain valid data.

After the bubble system has been powered up, the 7220 controller inhibits (or ignores) all commands except an Initialize or Abort command. One of these commands must be sent prior to issuing any other command. Normally, the first command issued after loading the parametric registers is the Initialize command. This complex command reads and decodes the bootloop information from each bubble in the system and places this information in the bootloop register of the corresponding 7242 FSA. Pointers internal to the 7220 automatically are prepared for normal operation. As described later, the combination of the Abort, MBM Purge and Write Bootloop Register commands is functionally similar to the Initialize command. (The only time the MBM Purge command is used is in conjunction with the Abort command).

Once the system has been initialized, the remainder of the command set can be selected. Assuming, for example, that a Read command is to be executed, the user selects the page number and length of the transfer via the parametric regisiters and then issues the Read command. If the system uses the polled mode, the CPU reads the status register and waits for the BUSY bit to go active and then for the FIFO READY bit to indicate that data is being sent to the FIFO. Data can be taken from the FIFO until the FIFO READY bit goes inactive.

If the page selected for the read operation is not in position to be read (i.e., the page is not at the replicate gates), additional time is required to execute the Read command as the proper page is rotated into position. In systems where faster response is desired, the Read Seek command can be used to place the page into position in order to free the CPU to perform other tasks. Once the page is in position, approximately eight milliseconds are required before the data is available to the CPU. This latency only occurs on the first page of a multipage transfer. Similarly, when a page is not in a position to be written, Write Seek can be used to position the page at the swap gates.

If there is any doubt regarding the state of the FIFO prior to a read or write operation, the user should issue a FIFO Reset command in order to clear the 7220's FIFO counter before initiating the data transfer. If a prior transfer is stopped with data remaining in the FIFO or if the FIFO is partially filled, the 7220's internal FIFO counter is not zero, and there is a danger that the subsequent transfer count may be incorrect. If the FIFO is reset properly, execution of a FIFO Reset command is redundant.

Although the 7220 FIFO may be treated as a 40-byte RAM buffer, the temptation to "pre-load" the FIFO with 40 bytes of data and then to issue a Write command should be avoided due to the danger of overflowing the FIFO. Prior to issuing a Write command, a FIFO Reset command should be sent, and the parametric registers should be loaded. Following the Write command, the CPU should monitor the status byte and wait for the BUSY and FIFO AVAILABLE bits to go active. When this status condition occurs, the user software should then send the proper number of bytes to the 7220. The FIFO AVAILABLE bit of the status byte should be polled prior to sending each byte.

An exception to not preloading the FIFO is when a Write Bootloop, Write Bootloop Register, or Write Bootloop Register Masked command is used. Prior to issuing any of these commands, a FIFO Reset command must be sent before preloading the bootloop data into the FIFO. When one of the bootloop-related commands is issued, the 7220 controller immediately begins taking data from the FIFO. If the FIFO is not preloaded, incorrect data may be transferred. The operation of the normal Write command differs from the bootloop-related commands in that, after a Write command is issued, the 7220 waits for at least two bytes to be present in the FIFO before beginning to transfer data to the bubble.

If the FSA encounters an error condition during a read or write operation, the status of the FSA is reflected in the 7220 status byte. If the user system decodes the error and decides to continue, the error flags in the 7220 controller and FSA first must be cleared. To clear the status bytes, the software can issue an Initialize command. However, this command resets all of the current operating parameters in the 7220 controller. To continue processing without resetting the system, the software can use the Software Reset command. This command resets any error flags and clears the FIFO, but does not affect the parametric register fields that define the system configuration (e.g., number of FSA channels selected).

INSTALLING THE BPK 72 BUBBLE MEMORY KIT

This section examines the individual components of the Bubble Memory System and how each component can be analyzed. All elements of the bubble system need not be working before any meaningful diagnostics can be effected. In general, a user first establishes communication between the host CPU and the 7220 controller. Next, communication with the 7242 formatter/sense-amplifier is verified via the 7220 controller. Finally, the operation of the 7110 Bubble Memory is checked. The software that exercises each of these phases of implementation should be small, well-defined device drivers that can be controlled through a system monitor.

The procedures that follow are applicable to most startup problems. The procedures are organized in chronological fashion and address each step of the installation process as it would normally occur. Software drivers in 8086 assembly language are provided to illustrate the basic functions supported by the device drivers.

Powering Up for the First Time

With power removed from the IMB-72 board, insert all of the supporting integrated circuits with the exception of the 7110 Bubble Memory Module. Insert the "dummy module" included in the BPK 72 kit in place of the 7110. The dummy module is electrically equivalent to the 7110 module and allows the circuits of the BPK 72 kit to be tested without the possibility of damaging the bubble. With both the +5V and +12V power supplies turned off, insert the IMB 72 with the dummy module into the edge connector. As power is applied to the system, monitor the RESET.OUT/pin of the 7220 controller and verify that the signal goes from low to high after power is applied. The low-to-high transition indicates that the power-up sequence has been completed successfully.

Communicating With the 7220 Bubble Memory Controller

The first step in communicating with the 7220 is to write initial values to the parametric registers using the code sequence in Table 15. When the registers have been set, the code shown in Table 12 can be used to examine the 7220 status byte.

The status value returned in Table 12 should be 40H. The user should not continue until the proper status value can be obtained repeatedly after performing the power-up sequence. Reading back the correct status indicates that the host CPU and the 7220 are communicating and that the power-up sequence is being performed by the 7220.

Table 12. Reading 7220 Controller Status

```
RDSTAT:
  ; THIS PROGRAM READS THE 7220
  ; STATUS BYTE
  ; TO READ STATUS, THE HOST CPU MUST
  ; READ FROM THE 7220 WITH A0 = 1.
    IN      AL, 49H       ; COMMANDS/STATUS
                          ; PORT ADDRESS OF
                          ; 7220
    MOV     STATUS, AL    ; MOVE AL REGISTER
                          ; TO STATUS
    RET
```

Once the power-up sequence is complete and the 7220 status register has been read, the 7220 FIFO can be accessed. The software drivers that write and read the FIFO are shown in Tables 13 and 14. Notice that these code sequences do not send commmands to the 7220; only data is transferred to and from the controller. The purpose here is to test the bus interface and timing between the CPU and the 7220 controller. In this case, the 7220 FIFO is used as a general purpose RAM. Any data can be written to the FIFO, but it is best to use an easily indentifiable sequence (e.g., an incrementing pattern) for easy recognition.

Table 13. Writing the 7220 FIFO

```
WTFIFO:
  ; THIS PROGRAM WRITES 40 BYTES FOR
  ; MEMORY TO THE 7220 FIFO.
  ; DATA IS ASSUMED TO BE ATBUFADR.
    MOVE    SI,BUFADR     ; LOAD BUFFER
                          ; POINTER
    MOV     CX,40         ; LOAD COUNT
WRT1:
    LODSB                 ; PUT BYTE AT SI
                          ; INTO AL, AUTO INCR
                          ; SI
    OUT     48H, AL       ; OUTPUT BYTE TO
                          ; DATA PORT
    LOOP    WRT1          ; DECREMENT COUNT,
                          ; LOOP IF NOT 0
    RET
```

Once forty bytes have been written to the FIFO, the 7220 status byte should be read. The status value should be "41H" (indicating that data is in the FIFO). Other status values such as "parity error" can be ignored. While status values give some indication of the CPU-7220 interaction, the integrity of the data is more important here. If the data read back is not the same as the data sent, a fundamental timing and/or interface problem between the CPU and the 7220 is indicated.

To verify that data is being transmitted to the 7220, the code sequence shown in Table 14 can be used to read back the FIFO data into user RAM space for direct comparison with the original pattern.

Table 14. Reading the 7220 FIFO

```
RDFIFO:
  ; THE PROGRAM READS 40 BYTES FROM
  ; THE 7220 FIFO INTO MEMORY.
    MOV    DI, BUFADR  ; LOAD BUFFER AD-
                       ; DRESS INTO DI
    MOV    CX,40       ; LOAD COUNT INTO
                       ; CX
RD1:
    IN     AL,48H      ; INPUT FROM DATA
                       ; PORT
    STOSB              ; STORE AL AT ADDR
                       ; IN DI, AUTO INCR. DI
    LOOP   RD1         ; DECREMENT COUNT
                       ; IN CX, LOOP IF NOT 0
    RET
```

After reading the FIFO, the status byte should be read (a value of "40H" or "42H," indicating that the FIFO has no data, should be obtained). The user should not proceed until the FIFO can be written and read correctly and until the FIFO status indicates the amount of data in the FIFO (not empty or empty). These steps verify that the CPU can communicate with the 7220. Note that no data has been transferred to or from the 7242 Formatter/Sense Amplifier or the 7110 bubble device (or dummy module).

Communicating With the 7242 Formatter/Sense Amplifier

The next step in verifying the BPK 72 is to ensure that the 7220 is driving the 7242 Formatter/Sense Amplifier properly by first setting up the 7220 for interaction with the 7242 and then sending commands to the 7220 to exercise the 7242 functions that can be verified easily.

Under normal operating conditions an Initialize command is the second command sent to the system. However, the Initialize command assumes that the 7110 Bubble Memory is installed and attempts to read bootloop information. Since the dummy module is installed at this time, timing errors result from the attempted Initialize command. Although no harm results from using the Initialize command, an Abort command followed by an MBM-Purge command can be used in place of the Initialize command to eliminate timing errors. The Abort command is sent by executing the code sequence at label "CMND9" in Table 16. When Abort command execution is complete, the user should read the status byte and check for an op-complete indication (40H).

Table 15. Write Register Sequence for Two FSA Channels

```
WTREG2:;                        WRITE REGISTERS
; 2 FSA CHANNELS SELECTED.
; THIS IS USED FOR DEBUG TO WRITE/READ THE
; BOOTLOOP REGISTERS AND CHECK FOR MISSING SEEDS, ETC.
; THE FOLLOWING VALUES INTO THE 7220 REGISTERS
;       B = 01H     : 1 PAGE TRANSFER
;       C = 10H     : SELECT 2 CHANNELS (WHOLE BUBBLE)
;       D = 08H     : STANDARD TRANSFER RATE
;       E = 00H     : PAGE 0
;       F = 00H     : FIRST BUBBLE

    MOV     AL, 0BH         ; SELECT B REGISTER
    OUT     49H, AL
    MOV     AL, 01H         ; ONE PAGE TRANSFERS
    OUT     48H, AL
    MOV     AL, 10H         ; WHOLE BUBBLE (2 FSA CHANNELS)
    OUT     48H, AL
    MOV     AL, 08H         ; LOW FREQ
    OUT     48H, AL
    MOV     AL, 00H         ; START ADDRESS = 0000H
    OUT     48H, AL
    MOV     AL, 00H         ; FIRST BUBBLE
    OUT     48H, AL
    RET
```

Once the op-complete status is received, the MBM-Purge command is issued by executing the routine labeled "CMNDE" in Table 16. This command, as described in the BPK 72 manual, clears all of the controller registers, counters and address RAM (except the block length register), the NFC bits, the FSA present counter and the high-order four bits of the address register. After the command is complete, the user again should receive an operation complete indication on reading the status byte.

After the Abort and MBM-Purge commands are executed and is status verified, additional commands may be sent to the 7220 BMC. Since the purpose of this section is to verify the interaction of the 7242 and 7220, manually loading and reading the 7242 bootloop registers can be used for the verification. Two additional commands are required to load and read the bootloop registers: the Write Bootloop Register command and the Read Bootloop Register command. These commands transfer data between the 7242 bootloop registers and the 7220 FIFO. Since the ability to transfer data between user RAM and the 7220

Table 16. 7220 Controller Commands

```
CMNDS:          ; 7220 COMMANDS
    ; THESE 16 ROUTINES EACH SEND A SINGLE COMMAND TO THE 7220.
    ; FOR EXAMPLE, THE "INITIALIZE COMMAND" WILL WRITE 11H
    ; TO THE 7220 WITH A0 = 1. THESE ARE THE 7220 COMMANDS LISTED
    ; IN THE BPK-72 USERS MANUAL.
CMND0:
    MOV         AL, 10H         ; WRITE BOOTLOOP REGISTER MASKED COMMAND
    OUT         49H, AL
    RET
CMND1:
    MOV         AL, 11H         ; INITIALIZE COMMAND
    OUT         49H, AL
    RET
CMND2:
    MOV         AL, 12H         ; READ COMMAND
    OUT         49H, AL
    RET
CMND3:
    MOV         AL, 13H         ; WRITE COMMAND
    OUT         49H, AL
    RET
CMND4:
    MOV         AL, 14H         ; READ SEEK COMMAND.
    OUT         49H, AL
    RET
CMND5:
    MOV         AL, 15H         ; READ BOOTLOOP REGISTER COMMAND
    OUT         49H, AL
    RET
CMND6:
    MOV         AL, 16H         ; WRITE BOOTLOOP REGISTER COMMAND
    OUT         49H, AL
    RET
CMND7:
    MOV         AL, 17H         ; WRITE BOOTLOOP COMMAND
    OUT         49H, AL
    RET
CMND8:
    MOV         AL, 18H         ; READ FSA STATUS COMMAND
    OUT         49H, AL
    RET
CMND9:
    MOV         AL, 19H         ; ABORT COMMAND
    OUT         49H, AL
    RET
CMNDA:
    MOV         AL, 1AH         ; WRITE SEEK COMMAND.
    OUT         49H, AL
    RET
CMNDB:
```

Table 16. 7220 Controller Commands (cont.)

```
        MOV     AL, 1BH         ; READ BOOTLOOP COMMAND
        OUT     49H, AL
        RET
CMNDC:
        MOV     AL, 1CH         ; READ CORRECTED DATA COMMAND
        OUT     49H, AL
        RET
CMNDD:
        MOV     AL, 1DH         ; FIFO RESET COMMAND
        OUT     49H, AL
        RET
CMNDE:
        MOV     AL, 1EH         ; MBM PURGE COMMAND
        OUT     49H, AL
        RET
CMNDF:
        MOV     AL, 1FH         ; SOFTWARE RESET COMMAND
        OUT     49H, AL
        RET
```

FIFO has been verified previously, these two additional commands verify the system's ability to transfer between user RAM and the 7242 FSA.

The 7220 parametric registers must be loaded prior to sending the Write Bootloop Register command. The sequence of operations is important; loading the parametric registers destroys the first byte of data in the 7220 FIFO. If valid bootloop information is placed in the FIFO before the parametric registers are loaded, the first byte of bootloop register information is invalid. Accordingly, the sequence of operations must be as follows:

(1) load the 7220 parametric registers
(2) load bootloop data into the 7220 FIFO
(3) send the Write Bootloop Register command.

As a point of interest, if a user wishes to maintain the system bootloop in EPROM rather than to allow automatic handling by the system, the Initialize command would not be used and would be replaced by a sequence similar to the one described.

After the 7220 parametric registers are loaded, the CPU next must load the 7220 FIFO with 40 bytes of bootloop register data using the "write FIFO" sequence from Table 13. This sequence then is followed by the code sequence to issue the Write Bootloop Register command. The data pattern written to the bootloop register should be an easily identified sequence of bytes such as an incrementing pattern. Under operational conditions, the data written to the bootloop registers represents "loop map" information that is written on the label of the 7110 device. Under these test conditions, it only is necessary to ensure that the 40 bytes sent out are the same 40 bytes read back.

Once the Write Bootloop Register command has been sent, the status byte is read (when the BUSY bit goes low) and an operation-complete status is verified. Any parity error indication may be ignored. Valid status at this point indicates that communication with the 7242 has been established. To verify that the data has been transferred properly, the contents of the bootloop register are read into the 7220's FIFO. The CPU then must transfer the data to user RAM in order to compare the data with the original pattern. To read the bootloop register, it only is necessary to issue the Read Bootloop Register command. This command places the contents of the 7242's bootloop register into the 7220's FIFO. The user then must execute the "read FIFO" sequence from Table 14 in order to transfer the data from the 7220 FIFO to RAM. Comparing the loop map written into the bootloop register and the loop map read from the bootloop register should show the loop maps to be equal.

Installing the 7110 MBM

Reading and writing the 7110 bubble memory requires the application of specific control signals at the appropriate times within the read or write cycles. These control signals originate from the 7254 and 7230 integrated circuits and are generated under the control of the 7220 BMC. Prior to installing the 7110, the presence of the control signals should be verified. While it is unlikely that the 7110 can be seriously damaged, it is possible for the "seeds" and bootloop established at the factory to be lost if there are problems with the 7254 or 7330 control signals and, if lost, would require additional steps on the part of the user to regenerate the seeds and bootloop data. With the dummy module installed, the required control signals can be verified directly on the bubble socket, and the possibility of damaging the bubble can be avoided.

The first control signal waveform to check is the coil drive on pins 9, 10, 11, and 12 of the 7110 socket. The drive current can be verified by ensuring that the voltage waveform on these pins (or on pins 1 and 7 of the 7254) conforms to Figure 6A when the drive field is being rotated. To rotate the drive field, the following code sequence can be used:

1. Write the parametric registers.
2. Send the Read command.

Next, the "cut and transfer" pulses generated during a read operation should be checked. The waveforms on pins 2 and 3 of the 7110 socket (REPLICATE.A and REPLICATE.B), should appear as shown in Figure 6B.

The cut and transfer pulses that occur during a write operation should now be verified. The waveforms on pins 7 and 8 of the 7110 socket (GENERATE. A and GENERATE. B) should appear as shown in Figure 6C. Since a write operation is required, a new code sequence must be used for this test:

1. Write the parametric registers.
2. Write data (any patten) to the FIFO.
3. Send the Write command.

bootloop register of the 7242 first must be loaded to allow data to be written. A Write Bootloop Register Masked command can be used to write a bootloop register pattern of all ones; it is only necessary to write the bootloop register once.

Finally, the SWAP pin is tested for proper operation during a write operation. The waveforms on pins 13 and 14 of the 7110 (SWAP.A and SWAP.B) should appear as shown in Figure 6D. The code sequence described for a write operation may be used.

One additional check of the system should be made prior to installing the 7110 device to determine if valid status values are received after a Read or Write command is issued to the 7220 BMC. Since the bubble is not yet installed, no data actually is transferred; the system should, however, execute the Read or Write command, and valid status should be received. Since a new command cannot be issued to the 7220 while a command is in progress, an Abort command is sent to cancel any command that may be pending from the last test performed. Next, a FIFO Reset command is sent to clear any data remaining in the FIFO. The status byte received should indicate an OP-COMPLETE and FIFO AVAILABLE status condition. The 7220 now is ready to execute a Read or Write command.

First, the 7220 parametric registers are loaded using the modified "diagnostic" driver shown in Table 17. This routine selects one FSA channel (half of a bubble) and, with ECC disabled, requires the loading of only 34 bytes in the 7220 FIFO. By limiting the FIFO to less than 40 bytes, FIFO underflow/overflow conditions are eliminated, and timing errors are avoided in the status byte. After, the 7220 FIFO is preloaded with 34 bytes of data (any pattern), a Write command is issued to the 7220 BMC. The 7220 status value received following command execution should reflect OP-COMPLETE since the 7220 transferred the data from its FIFO to the 7242 and executed the Write command as though the bubble were in place.

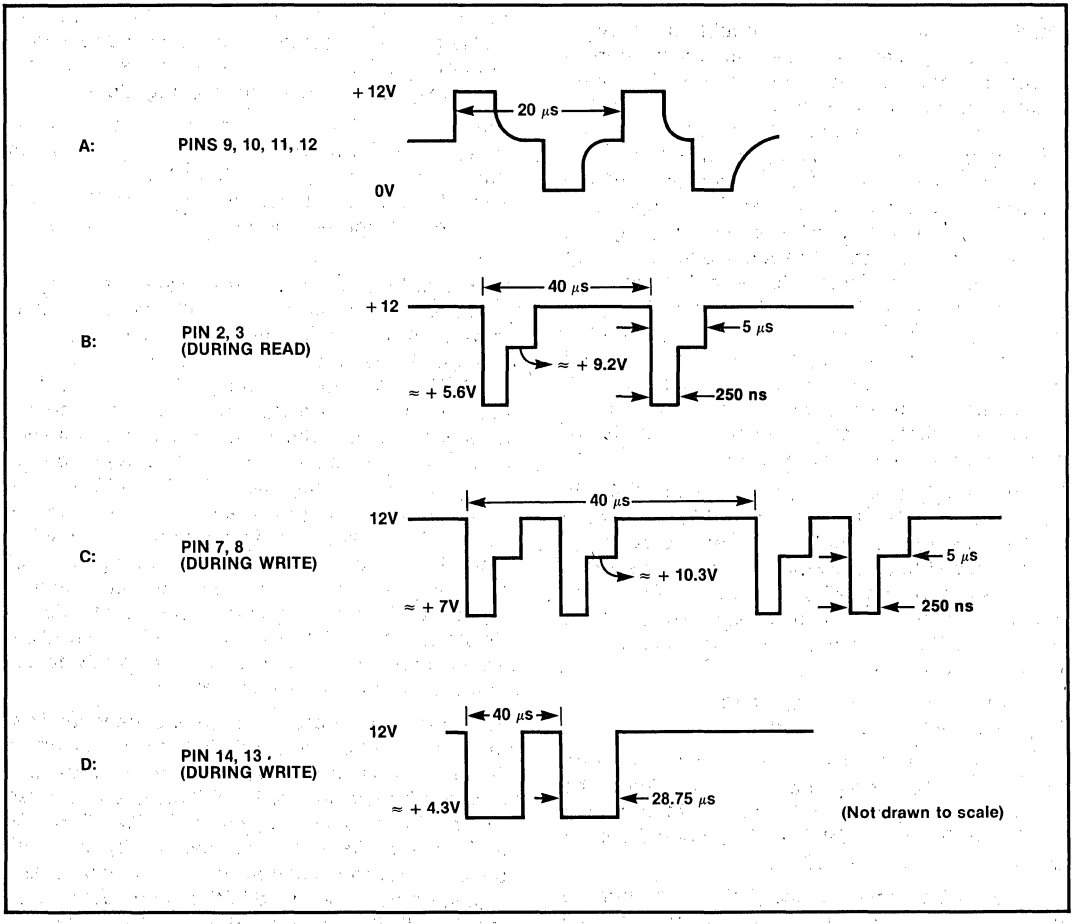


Figure 6. Control Signal Waveforms

To test the system in the read mode, the 7220 parametric registers are reloaded and a Read command is issued to the 7220. The user software must now read 34 bytes of "data" from the 7220's FIFO. Note that the data read will consist of all zeroes since no bubble is in place.

When the system completes all of the previous tests successfully, the 7110 bubble memory device may be inserted. Before proceeding, REMOVE POWER FROM THE SYSTEM.

Installing the 7110 is no different from installing any other device. Remove the dummy module in the 7110 socket and insert the 7110 Bubble Memory. Note that the 7110 is keyed to prevent the device from being inserted incorrectly. When power is applied, the system should execute its power-up sequence as described for the dummy module, and the 7220 status byte should return OP-COMPLETE after the parametric registers have been loaded.

Table 17. Write Register Sequence for One FSA Channel

```
WTREG1:;            WRITE REGISTERS (ONE HALF BUBBLE)
    ; THIS PROGRAM WRITES THE 7220 REGISTERS "B" THROUGH "F".
    ; DIAGNOSTIC ROUTINE WITH ONE FSA CHANNEL SELECTED
    ; THE FOLLOWING VALUES ARE WRITTEN TO THE 7220 REGISTERS.
    ;                B = 01H  :        1 PAGE TRANSFER
    ;                C = 00H  :        SELECT 1 CHANNEL (HALF BUBBLE)
    ;                D = 08H  :        LOW FREQ
    ;                E = 00H  :        PAGE 0
    ;                F = 00H  :        FIRST BUBBLE
      MOV     AL, 0BH       ; SET REGISTER ADDRESS COUNTER (RAC) TO B REGISTER
      OUT     49H, AL       ; PROT ADDRESS OF 7220 WITH A0 = 1
      MOV     AL, 01H       ; SET B REGISTER TO 01H (ONE PAGE TRANSFER)
      OUT     48H, AL       ; PORT ADDRESS OF 7220 WITH A0 = 0
      MOV     AL, 0H        ; SELECT HALF BUBBLE (1 FSA CHANNEL)
      OUT     48H, AL
      MOV     AL, 08H       ; SELECT LOW FREQ (NO ERROR CORRECTION)
      OUT     48H, AL
      MOV     AL, 0H        ; START ADDRESS = 000H
      OUT     48H, AL
      MOV     AL, 0H        ; SELECT THE FIRST BUBBLE
      OUT     48H, AL
      RET
```

Normal Read and Write Operations

Under normal operating conditions, a user sends an Initialize command and then proceeds to access the bubble. The Initialize command automatically purges the RAM area of the 7220, reads and decodes the bootloop on the 7110, fills the 7242 bootloop registers, and places the 7110 at page 0. This very important command is the next command to be tested before reading and writing data.

To verify the Initialize command, load the 7220 parametric registers to select both FSA channels for one bubble and then send the Initialize command. Status following execution of this command should be 40H, OP-COMPLETE. Once the 7220 is initialized, data can be transferred to and from the bubble. For a first attempt, it is recommeded that the operations be kept simple. That is, avoid error correction, DMA, or interrupts and only attempt single page transactions until reasonably familiar with the basic operations.

Prior to issuing the Write command, a FIFO Reset command is sent and then the parametric registers are loaded to select the page address and number of FSA channels. After the Write command is sent, the data should be output to the 7220 FIFO. When the proper number of bytes have been transferred, the 7220 status byte should reflect OP-COMPLETE and FIFO AVAILABLE to indicate that the data has been written into the 7110 bubble memory and can now be read. To read back the data written, issue a FIFO Reset command and reload the parametric registers to select the same page address in which the data was written. Issue the Read command to move the data from the 7110 to the 7220 FIFO and then use the "read FIFO" routine to transfer the data to user RAM. As always, the 7220 status byte should be checked after the operation.

AN IMPLEMENTATION EXAMPLE

To illustrate the ease with which Intel's bubble memory solution may be implemented, an MCS®-86 System Design Kit (SDK-86) is used as a vehicle to control a single BPK 72 bubble memory kit.

The bus interface between the 8086 CPU and the 7220 bubble memory controller requires seven integrated circuits and consists of four sections: address decode, data bus decode and buffering, a clock circuit, and miscellaneous control logic. The system requires power supply voltages of + 12V, + 5V, and, if a CRT is used, − 12V.

The 8086 bus is expanded through two 50-pin, wirewrap connectors, and the BPK 72 is connected to the SDK-86 by a flat cable into a 40-pin connector located on the SDK-86. The following interface diagram shows how the signals required by the bubble system are derived from the 8086. Detailed diagrams of the address, data, clock and control logic are in the appendix.

Either the SDK-86's Keypad or Serial monitor may be used to write and debug the necessary software drivers to control the BPK 72. There is, however, an EPROM-based monitor (BMDSDK) explicity designed for the BPK 72 and is available from the Intel Insite Library. Some of the bubble-specific portions of this monitor are discussed in the following text.

Monitor Software

The BMDSDK Bubble Monitor is a highly-modular program that is written in 8086 assembly language and that resides in two 2716 EPROMs. This monitor implements, at the console level, most of the standard SDK-86 monitor functions (display/change memory, etc.) and all of the 7220 commands. The current version of the monitor utilizes only polled I/O protocol; implementing an interrupt-driven system on the SDK-86 is possible using the principles outlined in this application note. The DMA mode of operation is not available with the hardware described.

The BPK 72 driver routines are confined to one module; a listing of this module is included in the appendix. To provide some feeling for the elements of "operational" software as opposed to the test drivers discussed earlier, the write function implemented in BMDSDK monitor is examined. The flow chart in Figure 9 shows how the routine is constructed on a functional basis. Note that the subroutine reflects a very "safe" approach in that the FIFO Reset command always is sent prior to issuing the Write command. While the FIFO Reset command is not mandatory, if there is any a doubt regarding the state of the FIFO prior to a read or write operation, resetting the FIFO is a good idea. Note also that a running byte count is maintained and that the routine exits when the count goes to zero. Such a counter is not actually necessary; the FIFO AVAILABLE bit alone can be used to gate the data to the 7220.

The calling program supplies the BMWRIT routine with the total number of bytes to be transferred in the CX register. The total number of bytes written is sent to the console at the end of the operation as a monitor function. BMWRIT also returns the value of the status byte to the calling program.

Note that at label WRIT01, the routine does not progress after the Write command is sent unless both the BUSY and FIFO AVAILABLE bits are set by the controller. Once these values are set, the code issues a byte of data to the controller only if the FIFO AVAILABLE bit indicates there is room. The remainder of the code in BMWRIT is concerned with processing special write requests for the bootloop and bootloop register commands.

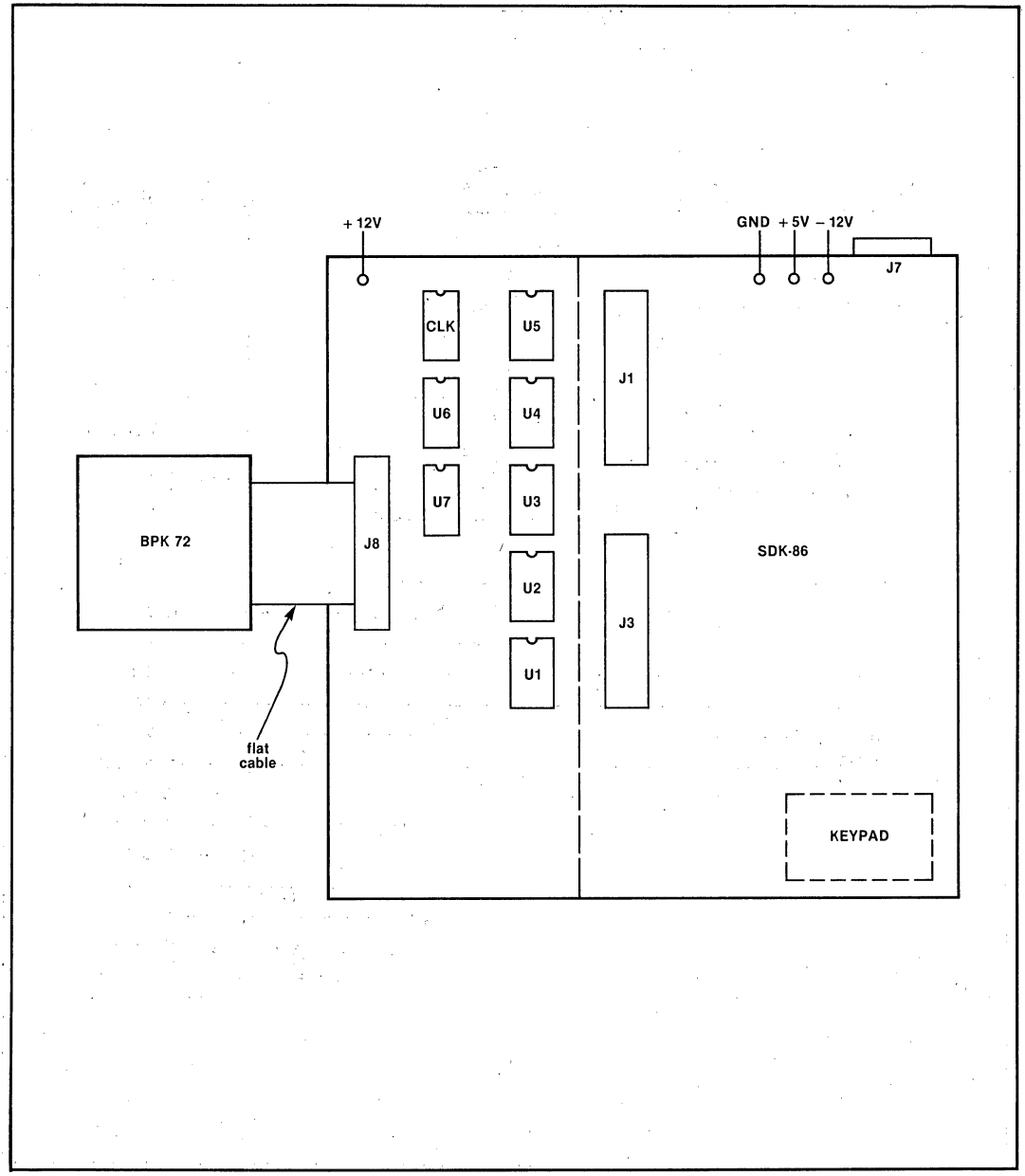


Figure 7. SDK-86/BPK 72 Implementation

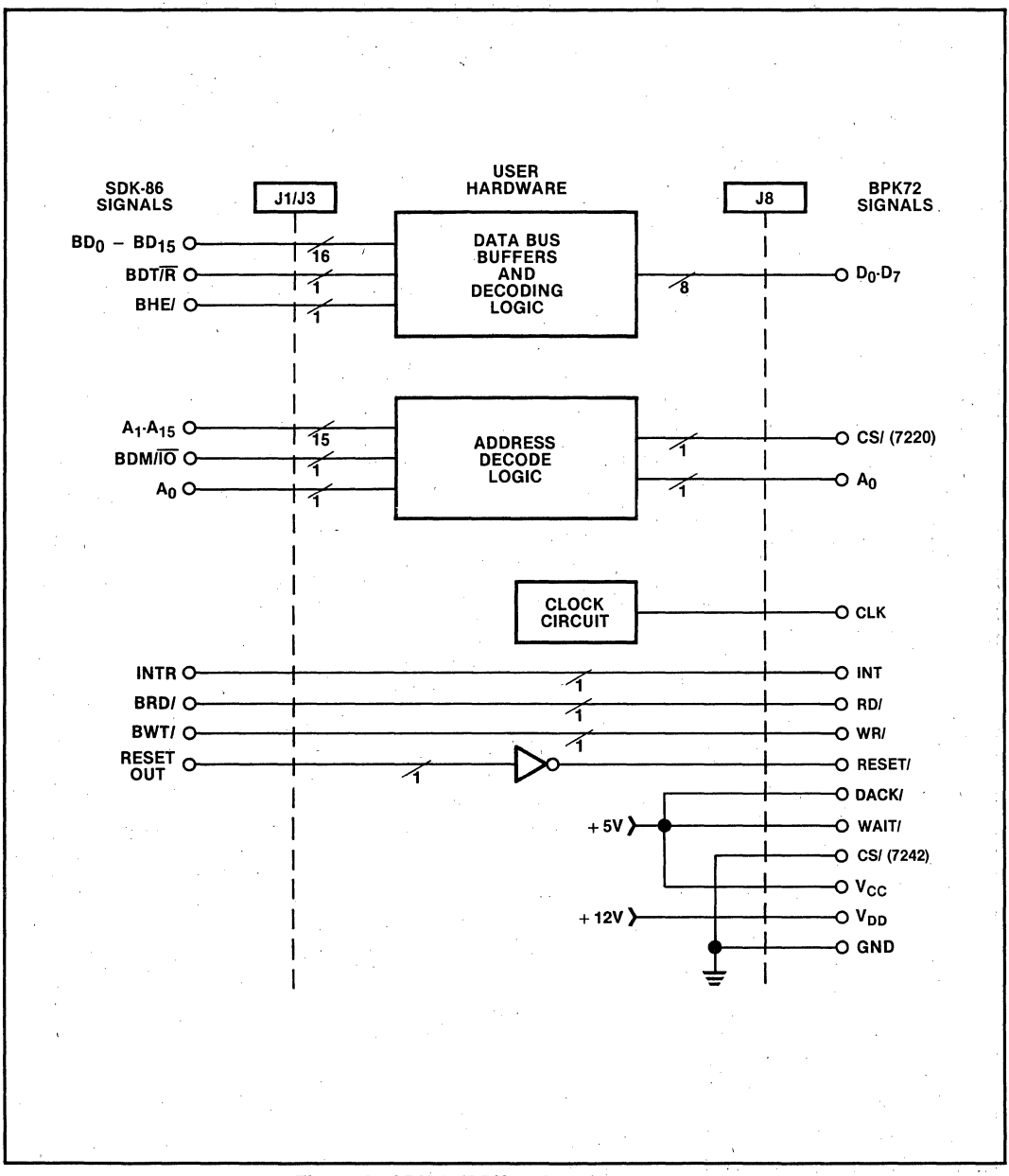


Figure 8. SDK-86/BPK 72 Interface Diagram

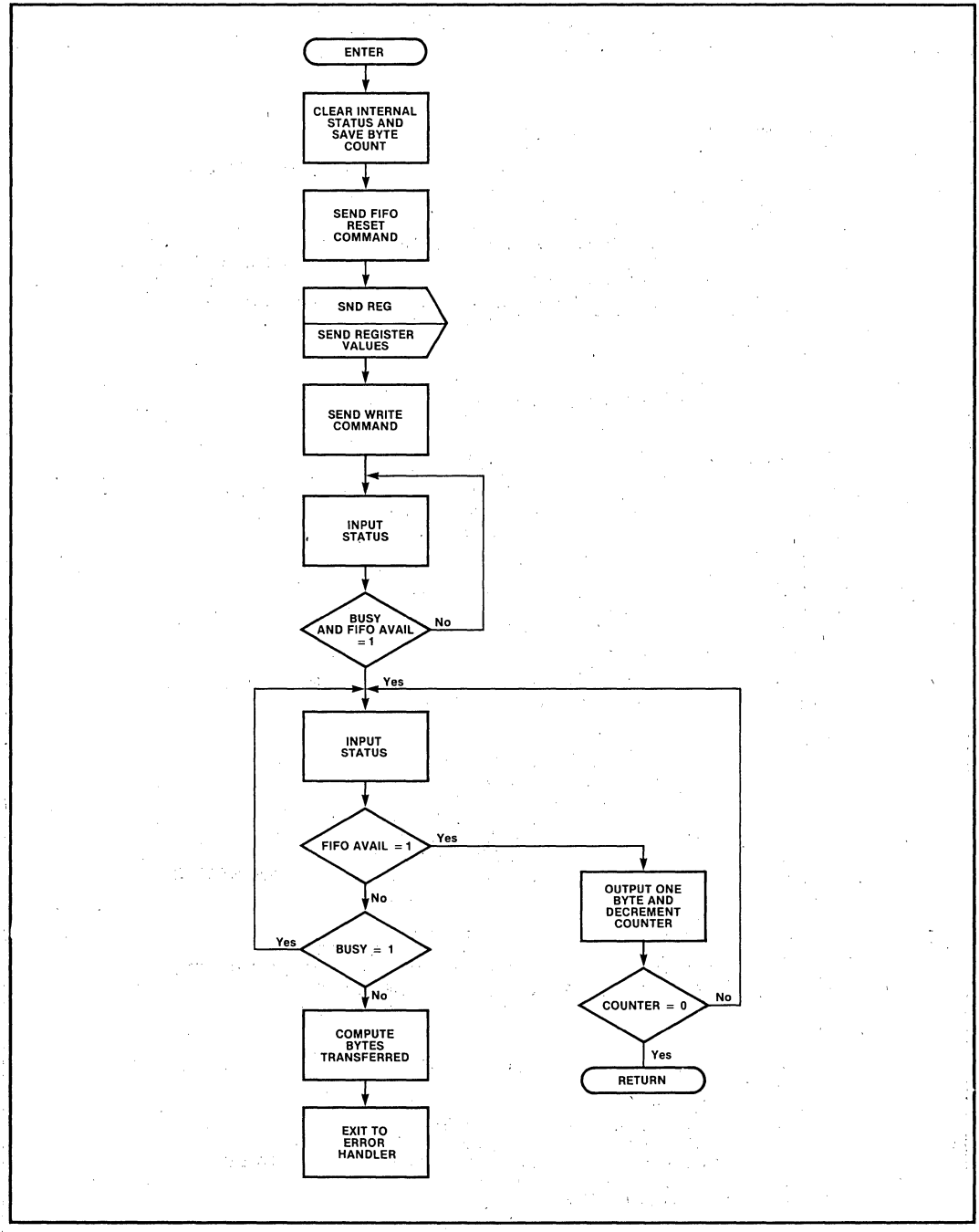


Figure 9. BMWRIT Flowchart

Table 18. BMWRIT Procedure for the SDK-86

```
;
;  FUNCTION: BMWRIT - WRITE BUBBLE MEMORY DATA.
;  INPUTS: CX = # OF BYTES TO WRITE.
;  OUTPUTS: A = STATUS: F/F(C = 1: ERROR OCCURED) BX = # OF BYTES WRITTEN.
;  CALLS: SNDREG, BMWAIT.
;  DESTROYS: ALL.
;  DESCRIPTION:   THIS PROCEDURE PERFORMS A BUBBLE MEMORY WRITE OPERATION.
;                 AN ERROR WILL OCCUR IF THE NUMBER OF BYTES GIVEN FOR THE
;                 WRITE OPERATION EXCEED THE NUMBER THAT THE BMC EXPECTS
;                 (DERIVED FROM COMMAND, BLOCK LENGTH AND NUMBER OF FSA
;                  CHANNELS), OR IF THE NUMBER OF BYTES IS LESS THAN THAT
;                  WHICH THE BMC EXPECTS.
;
BMWRIT:
        XOR     AL, AL              ; A = 0
        MOV     STATUS, AL          ; CLEAR STATUS
        MOV     BX, CX
        MOV     AL, CFR
        OUT     BMSTAT, AL          ; FIFO RESET
        CALL    SNDREG              ; SEND REGISTERS TO BMC.
        MOV     SI, BUFADR          ; SET UP SRC BFR PTR (IN DATA SEG)
        MOV     AL, BMCMD           ; GET COMMAND
        OUT     BMSTAT, AL          ; ISSUE IT.
WRIT01:
        IN      AL, BMSTAT
        TEST    AL, BUSYBT          ; WAIT FOR BUSY. . .
        JZ      WRIT01
        TEST    AL, FIFOBT          ; AND FIFO READY
        JZ      WRIT01
;
;  KEEP STUFFING DATA INTO FIFO UNTIL DONE OR AN ERROR OCCURS.
;  (NOTE: BMC GOING NOT BUSY IS AN ERROR).
;
WRIT03:
        IN      AL, BMSTAT          ; GET STATUS
        TEST    AL, FIFOBT          ; FIFO READY?
        JZ      WRIT04              ; NO, WAIT FOR IT
        LODSB                       ; YES, GET DATA FOR IT
        OUT     BMDATA, AL          ; GIVE IT TO BMC
        LOOP    WRIT03              ; LOOP UNTIL DONE.
        JMP     BMWAIT              ; XFER DONE, WAIT FOR A GOOD STATUS
WRIT04:
        TEST    AL, BUSYBT
        JNZ     WRIT03              ; OK IF STILL BUSY
        SUB     BX, CX              ; BX:# OF BYTES XFERED
        JMP     CTRL99              ; ERROR IF NOT BUSY AND CX NOT ZERO
;  SPECIAL WRITE FOR BOOTLOOP AND BOOTLOOP REG CMNDS
;
BMWRTB:
        XOR     AL, AL              ; A = 0
        MOV     STATUS, AL          ; CLEAR STATUS
        MOV     BX, CX
        MOV     AL, CFR
        OUT     BMSTAT, AL          ; FIFO RESET
        CALL    SNDREG              ; SEND REGISTERS TO BMC.
        MOV     SI, BUFADR          ; SET UP SRC BFR PTR (IN DATA SEG)

  ; FILL FIFO WITH 20/40/41 BYTES
```

Table 19. BMWRIT Procedure for the SDK-86 (cont.)

```
;
;
;
WRTB01:
        LODSB
        OUT     BMDATA, AL      ; STICK IN FIFO.
        LOOP    WRTB01          ; LOOP UNTIL FILL COUNT = 0.
        IN      AL, BMSTAT      ; GET BMC STATUS
        TEST    AL, BUSYBT      ; CHECK BUSY BIT.
        JZ      SHORT WAITEX    ; NOT BUSY, ALREADY DONE.
        MOV     CX, OFFFFH      ; JUST IN CASE. . .
WAITPO:                         ; POLLED WAIT MODE
        IN      AL, BMSTAT      ; GET STATUS
        TEST    AL, BUSYBT      ; CHECK BUSY BIT
        LOOPNZ  WAITPO          ; LOOP IF STILL BUSY
        JCXZ    CTRL99          ; PROBABLY AN ERROR IF CX = 0
WAITE:
        MOV     STATUS, AL      ; A = STATUS
        RET
```

SUMMARY

The purpose of this application note is to provide a more clear understanding of the functions and characteristics of the BPK 72 one-megabit bubble memory kit. This kit has been designed specifically to relieve the user of the design effort that historically is associated with implementing a bubble memory system, and to provide a simple interface that is compatible with a broad range of microprocessor systems.

The BPK 72 is a subsystem in itself that should be viewed as simply one more component on the system bus. This component-level approach, plus the inherent flexibility of the kit, provides the user with maximum utility and functionality. By understanding how each of the subsystem parts fits together and by approaching the implementation of the kit in a methodical fashion as described in this note, the development of a working system is facilitated.

APPENDIX A

SDK-86/BPK 72
HARDWARE INTERFACE

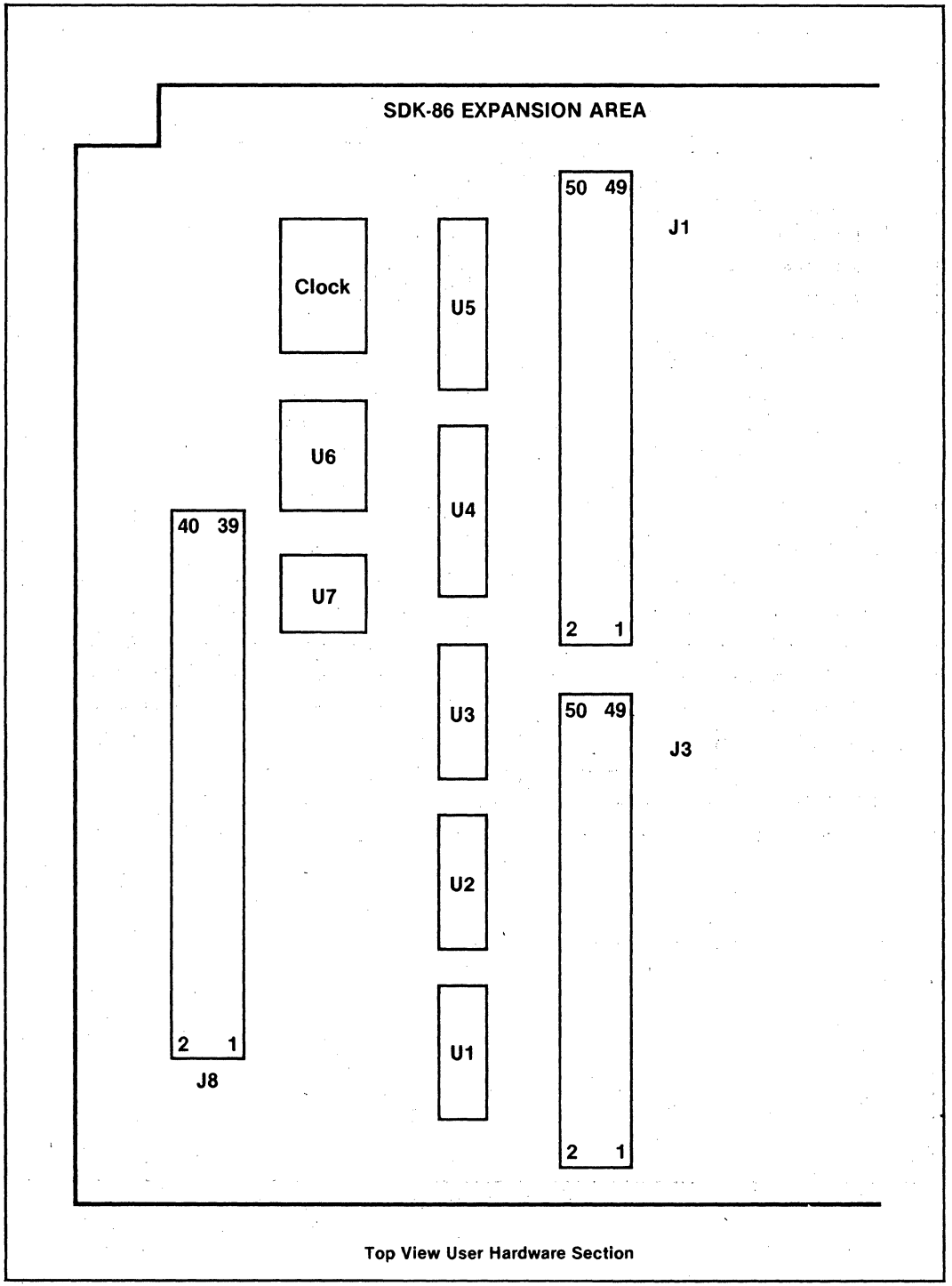
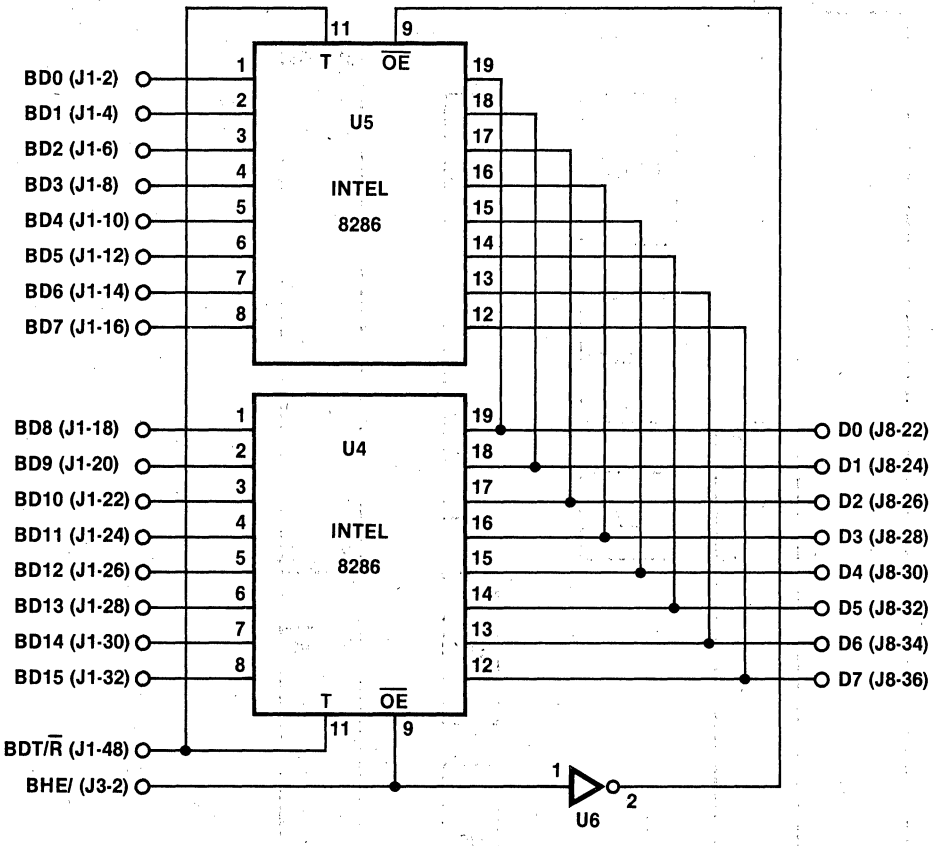


Figure 10. Parts Layout



8286 – VCC = 20
Gnd = 10

Figure 11. Data Bus Buffer and Decoding Logic

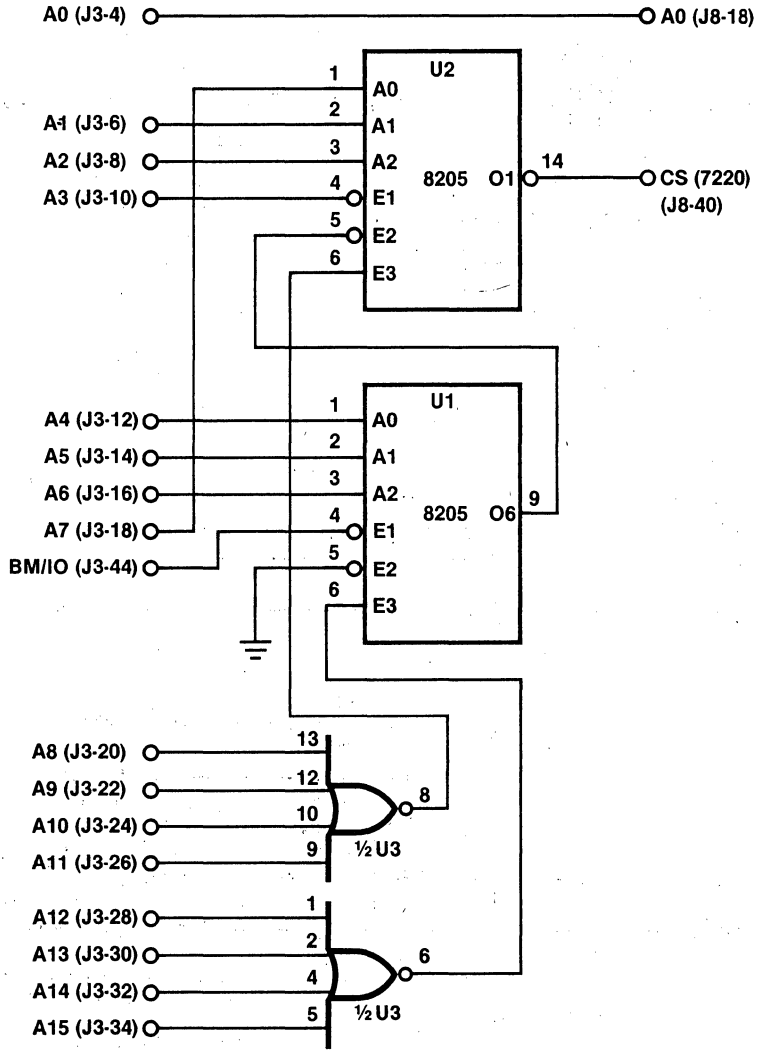


Figure 12. Address Decode Logic

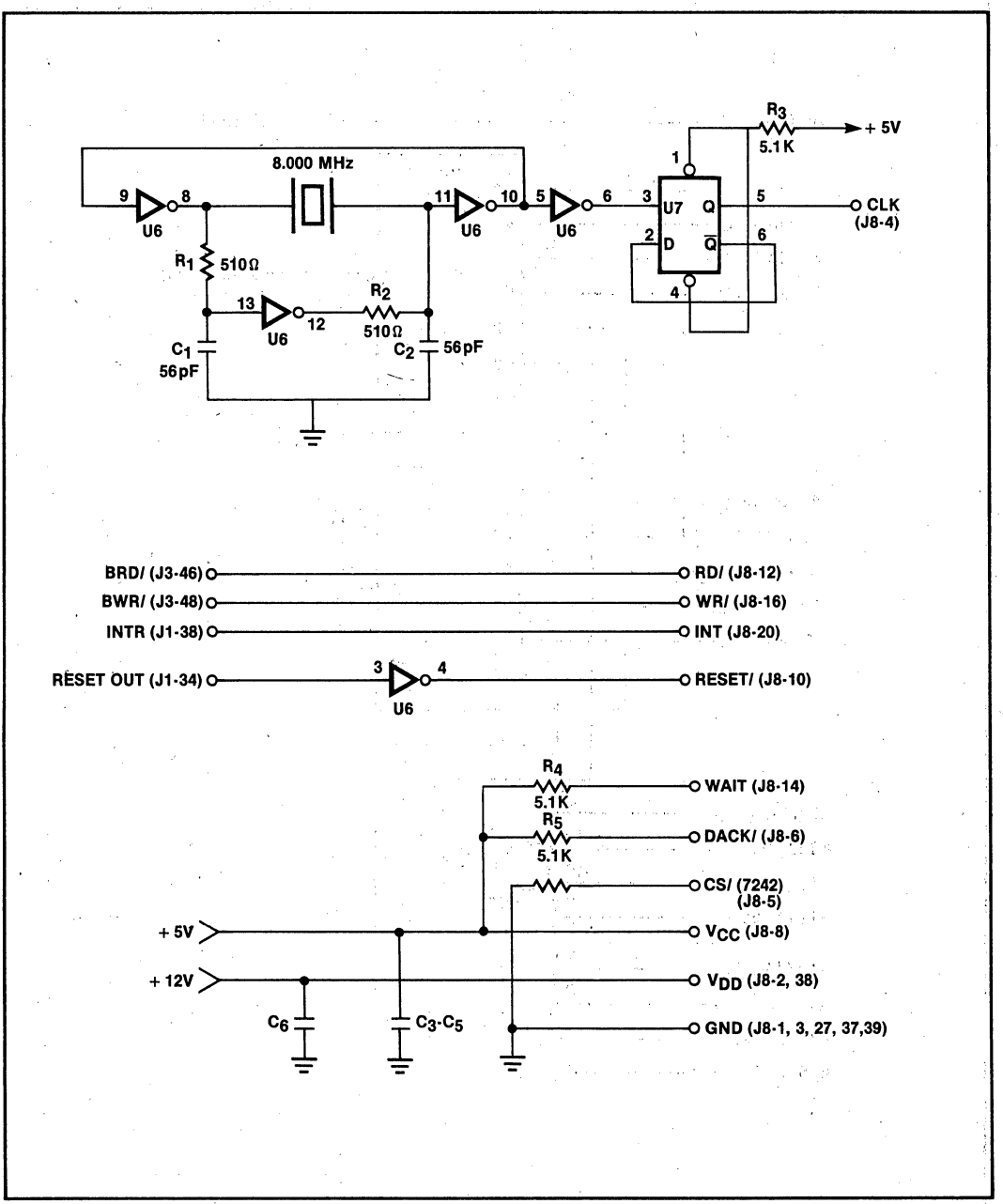


Figure 13. Clock Circuit and Control Signals

Table 20. SDK-86 Pinout

Pin	J1/J2	J3/J4	J5	J6
2	BD0	BHE/	P2C1	—
4	BD1	A0	P2C2	P1B3
6	BD2	A1	P2C3	P1B4
8	BD3	A2	P2B7	P1B2
10	BD4	A3	P2B0	P1B5
12	BD5	A4	P2B6	P1B1
14	BD6	A5	P2B3	P1B6
16	BD7	A6	P2B4	P1B0
18	BD8	A7	P2B2	P1B7
20	BD9	A8	P2B5	P1C3
22	BD10	A9	P2B1	P1C2
24	BD11	A10	P2C0	P1C1
26	BD12	A11	P2C4	P1C0
28	BD13	A12	P2C5	P1C4
30	BD14	A13	P2C6	P1C5
32	BD15	A14	P2C7	P1C6
34	RESET OUT	A15	P2A0	P1C7
36	PCLK/	A16	P2A7	P1A0
38	INTR	A17	P2A1	P1A7
40	TEST	A18	P2A6	P1A1
42	HOLD	A19	P2A2	P1A6
44	BHLDA	BM/IO/	P2A5	P1A2
46	BDEN/	BRD/	P2A3	P1A5
48	BDT/R/	BWR/	P2A4	P1A3
50	BALE	BINTA/	—	P1A4

All Odd Pins are Ground except as follows:

	J2
41	CSX/ (FD000-FDFFF)
43	CSY/ (FC000-FCFFF)
45	BS3
47	BS4
49	BS5

Table 21. SDK-86/BPK 72 Cable Wiring

Signal	J8	P1
+ 12v	2, 38	B, X
+ 5v	8	F
Ground	1, 3, 27, 37, 39	1, A, P, 22, Z
D0	22	11
D1	24	12
D2	26	13
D3	28	14
D4	30	15
D5	32	16
D6	34	17
D7	36	18
CS/ (7220)	40	Y
A0	18	10
RD/	12	J
WR/	16	K
INT	20	N
RESET/	10	H
CS/ (7242)	5	E
WAIT/	14	8
CLK	4	4
DACK/	6	L

Cable is standard 40 conductor Flat Cable.
All Odd Conductors are grounded at J8.

Table 22. SDK-86/BPK 72 Parts List

Item	Description	QT	Ref	
1	IC-8205 - Bindry Decoder	2	U1, U2	Intel (TI-74LS13)
2	IC-8286 - Octal Bus Tranciever	2	U4, U5	Intel
3	IC-746525 - Dual 4 Input M	1	U3	Any
4	IC-74H04 - Inverter	1	U6	Any
5	Resistor 510Q 1/4w	2	R1, R2	Any
6	Capacitor, 56pF 25V	2	C1,C2	Any
7	Capacitor, .1pF 25V	4	C3-C6	Any
8	Crystal, 8.000MHz Serie Res.	1	Y1	Any
9	Connector, 50 pin wirewrap	2	J1, J3	3M # 3433
10	Connector, 40 pin wirewrap	1	J8 (M)	3M # 3432
11	Connector, 40 pin	1	J8 (F)	3M # 3417
12	Connector, 44 pin Edge w/w	1	P1	Any
13	IC Socket, 20 pin w/w	2		Any (Augat)
14	IC Socket, 16 pin w/w	3		Any
15	IC Socket, 14 pin w/w	3		Any
16	Adapter Plug Assembly, 16 pin	1		Augat#616-CE1
17	Flat Cable, 40 Conductor, 1 Ft.	1		3M # 3365
18	IC-74LS74 - Dual D Flip-Flop	1	07	Any
19	Resistor 5.1K 1/4W ±5%	3	R3, R4, R5 R5	Any
20	IC-74LS32 - OR Gate	1	U8	Any

APPENDIX B

SDK-86/BPK 72
SOFTWARE DRIVER

```
ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE DRIVER
OBJECT MODULE PLACED IN :F1:DRIVER.OBJ
ASSEMBLER INVOKED BY:  asm86 :f1:DRIVER.a86 xref print(:f1:DRIVER.lst) debug WORKFILES(:F0:,:F0:)

LOC  OBJ                  LINE     SOURCE

                             1     $TITLE(                        BPK-72 DRIVER ROUTINES.)
                             2             NAME    DRIVER
                             3 +1  $INCLUDE(:F1:RAMDEF.EXT)
                         =1  4     ;
                         =1  5     ;       publics from module RAMDEF, file RAMDEF.A86
                         =1  6     ;
----                     =1  7     STACK   SEGMENT STACK
                         =1  8             EXTRN   BMSTAK:NEAR
----                     =1  9     STACK   ENDS
                         =1 10     ;
----                     =1 11     DATA    SEGMENT PUBLIC
                         =1 12             EXTRN   RAM:BYTE,SCRBUF:BYTE,MYBUF:BYTE
                         =1 13             EXTRN   DEFADR:WORD,DEFBUB:BYTE,DEFNFC:BYTE,DEFENA:BYTE
                         =1 14             EXTRN   DEFMOD:BYTE,DEFPAG:WORD,DEFBLK:WORD
                         =1 15             EXTRN   BUFADR:WORD,BLKLEN:WORD,ENABLE:BYTE,PAGENO:WORD
                         =1 16             EXTRN   BBLNUM:BYTE,NFC:BYTE,MODE:BYTE,STATUS:BYTE,BMCMD:BYTE
                         =1 17             EXTRN   INBUF:BYTE,INBUFP:WORD,INBUFC:BYTE
                         =1 18             EXTRN   INBUFA:WORD,INBUFL:BYTE
                         =1 19             EXTRN   OUTBUF:BYTE,OUTBFP:WORD,OUTBFC:BYTE
                         =1 20             EXTRN   OUTBFA:WORD,OUTBFL:BYTE
                         =1 21             EXTRN   RDLEN:WORD,WRLEN:WORD
                         =1 22             EXTRN   PROMPT:BYTE,LEVMSK:BYTE
                         =1 23             EXTRN   BPADR:WORD,USERRG:WORD
                         =1 24             EXTRN   POPREGS:WORD,PUSHREGS:WORD
                         =1 25             EXTRN   USERBX:WORD,USERDS:WORD,USERBP:WORD,USERSS:WORD
                         =1 26             EXTRN   USERSP:WORD,USERIP:WORD,USERCS:WORD,USERFL:WORD
                         =1 27             EXTRN   USERPC:WORD
----                     =1 28     DATA    ENDS
                         =1 29     ;
                            30 +1  $INCLUDE(:F1:BMC.EQU)
                         =1 31     ;
                         =1 32     ; THESE ARE THE COMMAND EQUATES FOR BMDS
                         =1 33     ;
  0010                   =1 34     CWBRM   EQU     10H                     ; WRITE BOOTLOOP WITH MASK.
  0011                   =1 35     CIZ     EQU     11H                     ;INITIALIZE
  0012                   =1 36     CRD     EQU     12H                     ;READ
  0013                   =1 37     CWD     EQU     13H                     ;WRITE
  0014                   =1 38     CRS     EQU     14H                     ;READ SEEK
  0015                   =1 39     CRBR    EQU     15H                     ;READ BOOTLOOP REGISTER
  0016                   =1 40     CWBR    EQU     16H                     ;WRITE BOOTLOOP REGISTER
  0017                   =1 41     CWB     EQU     17H                     ;WRITE BOOTLOOP
  0018                   =1 42     CRFS    EQU     18H                     ;READ FIFO STATUS
  0019                   =1 43     CAB     EQU     19H                     ;ABORT
  001A                   =1 44     CWRS    EQU     1AH                     ; WRITE SEEK.
  001B                   =1 45     CRB     EQU     1BH                     ;READ BOOTLOOP
  001C                   =1 46     CRCD    EQU     1CH                     ;READ CORRECTED DATA
  001D                   =1 47     CFR     EQU     1DH                     ;FIFO RESET
  001E                   =1 48     CPURG   EQU     1EH                     ; MBM PURGE COMMAND.
  001F                   =1 49     CSR     EQU     1FH                     ;SOFTWARE RESET
                         =1 50     ;
```

```
LOC  OBJ                  LINE     SOURCE

                    =1      51     ; I/O PORT ADDRESSES.
                    =1      52     ;
  00E1              =1      53     BMSTAT  EQU     0E1H            ; BUBBLE MEMORY DEVICE STATUS PORT.
  00E0              =1      54     BMDATA  EQU     0E0H            ; BUBBLE MEMORY DEVICE DATA PORT.
                    =1      55     ;
                    =1      56     ; STATUS WORD BITS
                    =1      57     ;
  0001              =1      58     FIFOBT  EQU     01H             ; FIRST BIT IS FIFO STATUS
  0002              =1      59     PARERR  EQU     02H             ; SECOND BIT IS PARITY ERROR.
  0004              =1      60     UNCERR  EQU     04H             ; THIRD BIT IS UNCORRECTABLE ERROR BIT.
  0008              =1      61     CORERR  EQU     08H             ; FOURTH BIT IS CORRECTABLE ERROR BIT.
  0010              =1      62     TIMERR  EQU     10H             ; FIFTH BIT IS TIMING ERROR BIT.
  0020              =1      63     OPFAIL  EQU     20H             ; OPERATION FAIL BIT.
  0040              =1      64     OPDONE  EQU     40H             ; OPERATION COMPLETE BIT.
  0080              =1      65     BUSYBT  EQU     80H             ; BUSY BIT.
                    =1      66     ;
                    =1      67     ; ENABLE REG BITS
                    =1      68     ;
  0001              =1      69     INTENA  EQU     01H             ; INTERRUPT NORMAL
  0002              =1      70     IERENA  EQU     02H             ; INTERRUPT ERROR
  0004              =1      71     DMAENA  EQU     04H             ; DMA
  0008              =1      72     RSVD1   EQU     08H
  0010              =1      73     WBLENA  EQU     10H             ; WRITE BOOTLOOP
  0020              =1      74     RCDENA  EQU     20H             ; READ CORRECTED DATA
  0040              =1      75     ICDENA  EQU     40H             ; INTERNALLY CORRECTED DATA
  0080              =1      76     RSVD2   EQU     80H
                            77 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                            78     CODE    SEGMENT PUBLIC
                            79             ASSUME  DS:DATA,CS:CODE,SS:STACK
                            80     ;*************************************************************
                            81     ;                                                           *
                            82     ;                 BPK72 DRIVER routines                     *
                            83     ;                 ===== ====== ========                     *
                            84     ;                                                           *
                            85     ;  The routines in this module constitute the routines      *
                            86     ;  needed to directly drive the BPK72 bubble memory         *
                            87     ;  development board.  This module is designed to be self   *
                            88     ;  contained, and may be called by ANY user procedures.     *
                            89     ;                                                           *
                            90     ;          The procedures in this module are                *
                            91     ;                                                           *
                            92     ;  BMCTRL - Perform non-data transfer BMC operations.      *
                            93     ;  BMREAD - Perform data read BMC operations.              *
                            94     ;  BMWRIT - Perform data write BMC operations.             *
                            95     ;                                                           *
                            96     ;  ZAPREG - Set internal registers to an acceptable value  *
                            97     ;                                                           *
                            98     ;          Parameter passing                                *
                            99     ;          ========= =======                                *
                           100     ;                                                           *
                           101     ;  All parameters are passed to the BMC driver routines via *
                           102     ;  common (PUBLIC) variables.  These variables are        *
                           103     ;                                                           *
                           104     ;  BUFADR - The memory address of the input/output buffer  *
                           105     ;            to be used for data transfer operations.       *
                           106     ;  ENABLE - The enable byte to be passed to the BMC before *
                           107     ;            every operation.                               *
                           108     ;  PAGENO - The starting block number to be passed to the  *
                           109     ;            BMC before every operation.  (NOTE: This field *
                           110     ;            has no meaning for control operations).        *
                           111     ;  BLKLEN - The number of pages to be transfered by the BMC.*
                           112     ;            (NOTE: This field has no meaning for control   *
                           113     ;            operations).                                   *
                           114     ;  BBLNUM - The bubble select to be transfered to the BMC  *
                           115     ;            before every operation.(NOTE: This field has   *
                           116     ;            no meaning for SOME control operations).       *
                           117     ;  NFC    - The number of FSA channels passed to the BMC    *
                           118     ;            before every operation. (NOTE: This field has  *
                           119     ;            no meaning for SOME of the control operatioNs).*
                           120     ;                                                           *
                           121     ;  For a detailed definition of the ENABLE,PAGENO.BLKLEN,  *
                           122     ;   BBLNUM. and NFC fields, refer to the BPK-72 USER MANUAL *
                           123     ;   or the Bubble Memory Design Handbook.                  *
                           124     ;                                                           *
                           125     ;*************************************************************
                           126 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           127     ;*******************************
                           128     ;
                           129     ; ENTRY POINTS
                           130     ;
                           131             PUBLIC  ZAPREG,BMCTRL,BMWAIT,BMREAD,BMWRIT,BMWRTB
                           132     ;
                           133     ;***********************
                           134     ;
                           135     ; MISC EQUATES
                           136     ;
  000B                     137     REG1    EQU     0BH                     ; FIRST BMC REGISTER TO USE IS BLOCK LENGTH
  003C                     138     STATER  EQU     3CH                     ; STATUS WORD ERROR MASK
                           139                                             ; IGNORE PARITY ERR, REV D OF BMC
                           140 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           141     ;*******************************
                           142     ;
                           143     ;    MODE BYTE DEFINITION
                           144     ;    ==== ==== ==========
                           145     ;
                           146     ; The bits in the MODE BYTE specify the type of the data transmission
                           147     ; TO USE, AND WHETHER TO PRINT STATUS AFTER EACH OPERATION.
                           148     ; If interrupts are enabled in the MODE BYTE, they must also be selected
                           149     ; in the ENABLE BYTE for desired operation to occur.
                           150     ;
  0001                     151     INTMOD  EQU     01H             ; FIRST BIT IN MODE WORD IT INTERRUPT SELECT.
  0002                     152     DMAMOD  EQU     02H             ; SECOND BIT IN MODE WORD IS DMA SELECT.
  0080                     153     DBGMOD  EQU     80H             ; DEBUG BIT OF MODE WORD
                           154 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                          155      ;*******************************
                          156      ;
                          157      ; FUNCTION: BMCTRL - PERFORM BMC CONTROL OPERATIONS (NON-DATA TRANSFER).
                          158      ; INPUTS: NONE
                          159      ; OUTPUTS: A=STATUS;F/F(C=1: AN ERROR OCCURED).
                          160      ; CALLS: SNDREG,BMWAIT
                          161      ; DESTROYS: ALL
                          162      ; DESCRIPTION: THIS PROCEDURE IS USED TO PERFORM NON-DATA TRANSFER
                          163      ;              BMC OPERATIONS.
                          164      ;
0000                      165      BMCTRL:
0000 E8D700               166              CALL    SNDREG                  ; LOAD BMC REGISTERS.
0003 A00000         E     167              MOV     AL,BMCMD                ; GET COMMAND.
0006 E6E1                 168              OUT     BMSTAT,AL               ; INITIATE COMMAND.
0008 E80E00               169              CALL    BMWAIT                  ; WAIT FOR COMPLETION.
000B 243C                 170              AND     AL,STATER               ; DO WE HAVE AN ERROR?
000D A00000         E     171              MOV     AL,STATUS               ; LOAD STATUS INTO 'A' FOR EXIT
0010 7502                 172              JNZ     SHORT CTRL99            ; ERROR, RETURN WITH FLAG SET.
0012 F8                   173              CLC                             ; CLEAR CARRY(ERROR FLAG)
0013 C3                   174              RET                             ; AND RETURN
                          175      ;
                          176      ; WE HAD AN ERROR, RETURN WITH ERROR FLAG(CARRY FLAG) SET.
                          177      ;          THIS IS THE GENERAL ERROR EXIT
                          178      ;
0014                      179      CTRL99:
0014 A20000         E     180              MOV     STATUS,AL
0017 F9                   181              STC                             ; SET ERROR FLAG (CARRY FLAG)
0018 C3                   182              RET                             ; AND RETURN.
                          183      ;*******************************
                          184      ;
                          185      ; FUNCTION: BMWAIT
                          186      ; INPUTS: NONE
                          187      ; OUTPUTS: STATUS IN A
                          188      ; CALLS: NOTHING
                          189      ; DESTROYS: A,F/F
                          190      ; DESCRIPTION: THIS PROCEDURE WILL WAIT UNTIL THE CURRENT BMC
                          191      ;              OPERATION COMPLETES.
                          192      ;
0019                      193      BMWAIT:
                          194      ;
                          195      ; CHECK CURRENT STATUS (GOOD ONLY IF RAC=0 AND BSY=0)
                          196      ;
0019 E4E1                 197              IN      AL,BMSTAT               ; GET BMC STATUS
001B A880                 198              TEST    AL,BUSYBT               ; CHECK BUSY BIT.
001D 740B                 199              JZ      SHORT WAITEX            ; NOT BUSY, ALREADY DONE.
001F B9FFFF               200              MOV     CX,0FFFFH               ; JUST IN CASE...
0022                      201      WAITPO:                                 ; POLLED WAIT MODE
0022 E4E1                 202              IN      AL,BMSTAT               ; GET STATUS
0024 A880                 203              TEST    AL,BUSYBT               ; CHECK BUSY BIT
0026 E0FA                 204              LOOPNZ  WAITPO                  ; LOOP IF STILL BUSY
0028 E3EA                 205              JCXZ    CTRL99                  ; PROBABLY AN ERROR IF CX=0
002A                      206      WAITEX:                                 ; CORRECT STATUS AND RETURN.
002A A20000         E     207              MOV     STATUS,AL               ; A = STATUS
002D C3                   208              RET
                          209 +1   $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           210     ;*****************************
                           211     ;
                           212     ; FUNCTION: BMREAD
                           213     ; INPUTS: CX = NUMBER OF BYTES TO READ, ES SET TO DS
                           214     ; OUTPUTS: A = STATUS; F/F(C=1: ERROR OCCURED)
                           215     ;          BX = NUMBER OF BYTES READ
                           216     ; CALLS: SNDREG
                           217     ; DESTROYS: ALL
                           218     ; DESCRIPTION: ALL PARAMETERS ARE PASSED THROUGH COMMON(PUBLIC)
                           219     ;              VARIABLES( SEE MODULE HEADER).
                           220     ;
002E                       221     BMREAD:
002E 32C0                  222             XOR     AL,AL                   ; A = 0
0030 A20000          E     223             MOV     STATUS,AL               ; CLEAR STATUS.
0033 8BD9                  224             MOV     BX,CX                   ; SAVE BYTE COUNT FOR LOOP
0035 E8A200                225             CALL    SNDREG                  ; SEND REGISTERS TO BMC.
0038 8B3E0000        E     226             MOV     DI,BUFADR               ; SET UP DEST BFR PTR (IN EXTRA SEG)
003C 8CD8                  227             MOV     AX,DS
003E 8EC0                  228             MOV     ES,AX                   ; SET EXTRA SEG FOR BYTE MOVE DEST
0040 A00000          E     229             MOV     AL,BMCMD                ; GET COMMAND
0043 E6E1                  230             OUT     BMSTAT,AL               ; ISSUE IT.
                           231     ;
0045 B9FFFF                232             MOV     CX,0FFFFH
0048                       233     BMRD1:
0048 E4E1                  234             IN      AL,BMSTAT
004A A880                  235             TEST    AL,BUSYBT
004C E1FA                  236             LOOPZ   BMRD1                   ; WAIT FOR BUSY, BUT NOT FOREVER
004E E3C4                  237             JCXZ    CTRL99                  ; CX=0 PROBABLY AN ERROR
0050 8BCB                  238             MOV     CX,BX
                           239     ;
                           240     ;        READ LOOP
                           241     ;        ==== ====
                           242     ;
0052                       243     BMRD2:
0052 E4E1                  244             IN      AL,BMSTAT               ; GET STATUS
0054 A801                  245             TEST    AL,FIFOBT               ; FIFO EMPTY?
0056 7407                  246             JZ      SHORT BMRD3             ; YEP, GO CHECK FOR BUSY.
0058 E4E0                  247             IN      AL,BMDATA               ; NOPE, GET DATA
005A AA                    248             STOSB                           ; STORE IT
005B E2F5                  249             LOOP    BMRD2                   ; AND GO FOR MORE.
005D EBBA                  250             JMP     BMWAIT                  ; XFER DONE, WAIT FOR A GOOD STATUS
005F                       251     BMRD3:                                  ; NOTHING IN FIFO, IS OP COMPLETE?
005F A880                  252             TEST    AL,BUSYBT               ; CHECK BUSY BIT
0061 75EF                  253             JNZ     BMRD2                   ; STILL BUSY, WAIT.
0063 2BD9                  254             SUB     BX,CX                   ; BX <- # OF BYTES XFERED
0065 EBAD                  255             JMP     CTRL99
                           256 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                          257      ;*****************************
                          258      ;
                          259      ; FUNCTION: BMWRIT - WRITE BUBBLE MEMORY DATA.
                          260      ; INPUTS: CX = # OF BYTES TO WRITE.
                          261      ; OUTPUTS: A = STATUS; F/F(C=1:ERROR OCCURED), BX=# OF BYTES WRITTEN.
                          262      ; CALLS: SNDREG,BMWAIT.
                          263      ; DESTROYS: ALL.
                          264      ; DESCRIPTION: THIS PROCEDURE PERFORMS A BUBBLE MEMORY WRITE OPERATION.
                          265      ;              AN ERROR WILL OCCUR IF THE NUMBER OF BYTES GIVEN FOR THE
                          266      ;              WRITE OPERATION EXCEED THE NUMBER THAT THE BMC EXPECTS
                          267      ;              (DERIVED FROM COMMAND, BLOCK LENGTH AND NUMBER OF FSA
                          268      ;               CHANNELS), OR IF THE NUMBER OF BYTES IS LESS THAN THAT
                          269      ;              WHICH THE BMC EXPECTS.
                          270      ;
0067                      271      BMWRIT:
0067 32C0                 272              XOR     AL,AL                   ; A = 0
0069 A20000          E    273              MOV     STATUS,AL               ; CLEAR STATUS
006C 8BD9                 274              MOV     BX,CX
006E B01D                 275              MOV     AL,CFR
0070 E6E1                 276              OUT     BMSTAT.AL               ; FIFO RESET
0072 E86500               277              CALL    SNDREG                  ; SEND REGISTERS TO BMC.
0075 8B360000        E    278              MOV     SI.BUFADR               ; SET UP SRC BFR PTR (IN DATA SEG)
0079 A00000          E    279              MOV     AL,BMCMD                ; GET COMMAND
007C E6E1                 280              OUT     BMSTAT,AL               ; ISSUE IT.
007E                      281      WRIT01:
007E E4E1                 282              IN      AL,BMSTAT
0080 A880                 283              TEST    AL,BUSYBT               ; WAIT FOR BUSY...
0082 74FA                 284              JZ      WRIT01
0084 A801                 285              TEST    AL,FIFOBT               ;   AND FIFO READY
0086 74F6                 286              JZ      WRIT01
                          287      ;
                          288      ; KEEP STUFFING DATA INTO FIFO UNTIL DONE OR AN ERROR OCCURS.
                          289      ; (NOTE: BMC GOING NOT BUSY IS AN ERROR).
                          290      ;
0088                      291      WRIT03:
0088 E4E1                 292              IN      AL,BMSTAT               ; GET STATUS
008A A801                 293              TEST    AL,FIFOBT               ; FIFO READY?
008C 7407                 294              JZ      WRIT04                  ; NO. WAIT FOR IT
008E AC                   295              LODSB                           ; YES, GET DATA FOR IT
008F E6E0                 296              OUT     BMDATA,AL               ; GIVE IT TO BMC
0091 E2F5                 297              LOOP    WRIT03                  ; LOOP UNTIL DONE.
0093 EB84                 298              JMP     BMWAIT                  ; XFER DONE, WAIT FOR A GOOD STATUS
0095                      299      WRIT04:
0095 A880                 300              TEST    AL,BUSYBT
0097 75EF                 301              JNZ     WRIT03                  ; OK IF STILL BUSY
0099 2BD9                 302              SUB     BX,CX                   ; BX <- # OF BYTES XFERED
009B E976FF               303              JMP     CTRL99                  ; ERROR IF NOT BUSY AND CX NOT ZERO
                          304      ;
                          305      ; SPECIAL WRITE FOR BOOTLOOP AND BOOTLOOP REG CMNDS
                          306      ;
009E                      307      BMWRTB:
009E 32C0                 308              XOR     AL,AL                   ; A = 0
00A0 A20000          E    309              MOV     STATUS,AL               ; CLEAR STATUS
00A3 8BD9                 310              MOV     BX,CX
00A5 B01D                 311              MOV     AL,CFR
```

```
LOC  OBJ                  LINE     SOURCE

00A7 E6E1                  312              OUT     BMSTAT.AL               ; FIFO RESET
00A9 E82E00                313              CALL    SNDREG                  ; SEND REGISTERS TO BMC.
00AC 8B360000        E     314              MOV     SI.BUFADR               ; SET UP SRC BFR PTR (IN DATA SEG)
                           315      ;
                           316      ; FILL FIFO WITH 20/40/41 BYTES
                           317      ;
00B0                       318      WRTB01:
00B0 AC                    319              LODSB
00B1 E6E0                  320              OUT     BMDATA,AL               ; STICK IN FIFO.
00B3 E2FB                  321              LOOP    WRTB01                  ; LOOP UNTIL FILL COUNT=0.
00B5 A00000          E     322              MOV     AL,BMCMD
00B8 E6E1                  323              OUT     BMSTAT,AL               ; SEND CMND
00BA E95CFF                324              JMP     BMWAIT
                           325 +1   $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                          326      ;*******************************
                          327      ;
                          328      ; FUNCTION: ZAPREG - ZAP ALL INTERNAL REGISTERS.
                          329      ; INPUTS: NONE
                          330      ; OUTPUTS: NONE
                          331      ; CALLS: NOTHING
                          332      ; DESTROYS: NOTHING.
                          333      ; DESCRIPTION: SET ALL INTERNAL REGISTERS EXCEPT 'ENABLE' TO AN
                          334      ;              ACCEPTABLE VALUE.  NOTE:  AN ACCEPTABLE VALUE MAY
                          335      ;              OR MAY NOT BE THE ONE DESIRED AS A DEFAULT.
                          336      ;
00BD                      337      ZAPREG:
00BD 9C                   338              PUSHF                           ; SAVE FLAGS
00BE 50                   339              PUSH    AX                      ; SAVE REGISTERS
00BF 53                   340              PUSH    BX
00C0 BB0000               341              MOV     BX,0
00C3 891E0000         E   342              MOV     PAGENO,BX               ; STARTING PAGE NUMBER = 0
00C7 43                   343              INC     BX
00C8 891E0000         E   344              MOV     BLKLEN,BX               ; BLOCK LENGTH = 1
00CC 32C0                 345              XOR     AL,AL
00CE A20000           E   346              MOV     BBLNUM.AL               ; BUBBLE NUMBER = 0
00D1 FEC0                 347              INC     AL
00D3 A20000           E   348              MOV     NFC.AL                  ; # OF FSA CHANNELS = 1 (2 CHANNELS)
00D6 5B                   349              POP     BX                      ; RESTORE REGISTERS.
00D7 58                   350              POP     AX
00D8 9D                   351              POPF
00D9 C3                   352              RET
                          353 +1   $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           354     ;*******************************
                           355     ;
                           356     ; FUNCTION: SNDREG - FORMAT AND SEND INTERNAL REGISTERS TO BMC.
                           357     ; INPUTS: NONE
                           358     ; OUTPUTS: NONE
                           359     ; DESTROYS: NOTHING.
                           360     ; DESCRIPTION: FORMAT AND SEND ALL INTERNAL REGISTERS TO THE BMC.
                           361     ;
00DA                       362     SNDREG:
00DA 9C                    363             PUSHF
00DB 50                    364             PUSH    AX                      ; SAVE REGISTERS
00DC 53                    365             PUSH    BX
00DD 51                    366             PUSH    CX
00DE B00B                  367             MOV     AL,REG1                 ; GET FIRST REGISTER ADDRESS.
00E0 E6E1                  368             OUT     BMSTAT,AL               ; SELECT IT.
                           369     ;
                           370     ; CONSTRUCT AND SEND BLOCK LENGTH.
                           371     ;
00E2 8B1E0000        E     372             MOV     BX,BLKLEN               ; HL = BLOCK LENGTH
00E6 8AC3                  373             MOV     AL,BL                   ; A  = BLOCK LENGTH LSB
00E8 E6E0                  374             OUT     BMDATA,AL               ; GIVE IT TO BMC.
00EA A00000          E     375             MOV     AL,NFC                  ; A  = NUMBER OF FSA CHANNELS.
00ED B104                  376             MOV     CL,4
00EF D2E0                  377             SHL     AL,CL
00F1 0AC7                  378             OR      AL,BH                   ; MERGE INTO BLOCK MSB
00F3 E6E0                  379             OUT     BMDATA,AL               ; GIVE IT TO BMC.
                           380     ;
                           381     ; SEND ENABLE BYTE.
                           382     ;
00F5 A00000          E     383             MOV     AL,ENABLE               ; GET ENABLE BYTE
00F8 E6E0                  384             OUT     BMDATA,AL               ; GIVE IT TO BMC
                           385     ;
                           386     ; CONSTRUCT AND SEND ADDRESS REGISTER.
                           387     ;
00FA 8B1E0000        E     388             MOV     BX,PAGENO               ; HL = STARTING PAGE NUMBER
00FE 8AC3                  389             MOV     AL,BL                   ; A  = ADDRESS REGISTER LSB
0100 E6E0                  390             OUT     BMDATA,AL               ; GIVE IT TO BMC.
0102 A00000          E     391             MOV     AL,BBLNUM               ; A  = BUBBLE NUMBER
0105 B103                  392             MOV     CL,3
0107 D2E0                  393             SHL     AL.CL
0109 0AC7                  394             OR      AL,BH                   ; MERGE INTO PAGE NUMBER MSB.
010B E6E0                  395             OUT     BMDATA,AL               ; GIVE IT TO BMC.
                           396     ;
                           397     ; RESTORE REGISTERS AND RETURN.
                           398     ;
010D 59                    399             POP     CX
010E 5B                    400             POP     BX
010F 58                    401             POP     AX
0110 9D                    402             POPF
0111 C3                    403             RET
                           404 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           405     CODE    ENDS
                           406             END
```

XREF SYMBOL TABLE LISTING
---- ------ ----- -------

```
NAME      TYPE     VALUE  ATTRIBUTES, XREFS

??SEG . . SEGMENT        SIZE=0000H PARA PUBLIC
BBLNUM. . V BYTE   0000H  EXTRN  16# 346 391
BLKLEN. . V WORD   0000H  EXTRN  15# 344 372
BMCMD . . V BYTE   0000H  EXTRN  16# 167 229 279 322
BMCTRL. . L NEAR   0000H  CODE PUBLIC  131 165#
BMDATA. . NUMBER   00E0H    54# 247 296 320 374 379 384 390 395
BMRD1 . . L NEAR   0048H  CODE  233# 236
BMRD2 . . L NEAR   0052H  CODE  243# 249 253
BMRD3 . . L NEAR   005FH  CODE  246 251#
BMREAD. . L NEAR   002EH  CODE PUBLIC  131 221#
BMSTAK. . L NEAR   0000H  EXTRN  8#
BMSTAT. . NUMBER   00E1H    53# 168 197 202 230 234 244 276 280 282 292 312 323 368
BMWAIT. . L NEAR   0019H  CODE PUBLIC  131 169 193# 250 298 324
BMWRIT. . L NEAR   0067H  CODE PUBLIC  131 271#
BMWRTB. . L NEAR   009EH  CODE PUBLIC  131 307#
BPADR . . V WORD   0000H  EXTRN  23#
BUFADR. . V WORD   0000H  EXTRN  15# 226 278 314
BUSYBT. . NUMBER   0080H    65# 198 203 235 252 283 300
CAB . . . NUMBER   0019H    43#
CFR . . . NUMBER   001DH    47# 275 311
CIZ . . . NUMBER   0011H    35#
CODE. . . SEGMENT        SIZE=0112H PARA PUBLIC   78# 79 405
CORERR. . NUMBER   0008H    61#
CPURG . . NUMBER   001EH    48#
CRB . . . NUMBER   001BH    45#
CRBR. . . NUMBER   0015H    39#
CRCD. . . NUMBER   001CH    46#
CRD . . . NUMBER   0012H    36#
CRFS. . . NUMBER   0018H    42#
CRS . . . NUMBER   0014H    38#
CSR . . . NUMBER   001FH    49#
CTRL99. . L NEAR   0014H  CODE  172 179# 205 237 255 303
CWB . . . NUMBER   0017H    41#
CWBR. . . NUMBER   0016H    40#
CWBRM . . NUMBER   0010H    34#
CWD_. . . NUMBER   0013H    37#
CWRS. . . NUMBER   001AH    44#
DATA. . . SEGMENT        SIZE=0000H PARA PUBLIC   11# 28 79
DBGMOD. . NUMBER   0080H    153#
DEFADR. . V WORD   0000H  EXTRN  13#
DEFBLK. . V WORD   0000H  EXTRN  14#
DEFBUB. . V BYTE   0000H  EXTRN  13#
DEFENA. . V BYTE   0000H  EXTRN  13#
DEFMOD. . V BYTE   0000H  EXTRN  14#
DEFNFC. . V BYTE   0000H  EXTRN  13#
DEFPAG. . V WORD   0000H  EXTRN  14#
DMAENA. . NUMBER   0004H    71#
DMAMOD. . NUMBER   0002H    152#
ENABLE. . V BYTE   0000H  EXTRN  15# 383
FIFOBT. . NUMBER   0001H    58# 245 285 293
ICDENA. . NUMBER   0040H    75#
```

```
NAME      TYPE     VALUE  ATTRIBUTES, XREFS

IERENA. . NUMBER   0002H    70#
INBUF . . V BYTE   0000H  EXTRN  17#
INBUFA. . V WORD   0000H  EXTRN  18#
INBUFC. . V BYTE   0000H  EXTRN  17#
INBUFL. . V BYTE   0000H  EXTRN  18#
INBUFP. . V WORD   0000H  EXTRN  17#
INTENA. . NUMBER   0001H    69#
INTMOD. . NUMBER   0001H   151#
LEVMSK. . V BYTE   0000H  EXTRN  22#
MODE. . . V BYTE   0000H  EXTRN  16#
MYBUF . . V BYTE   0000H  EXTRN  12#
NFC . . . V BYTE   0000H  EXTRN  16# 348 375
OPDONE. . NUMBER   0040H    64#
OPFAIL. . NUMBER   0020H    63#
OUTBFA. . V WORD   0000H  EXTRN  20#
OUTBFC. . V BYTE   0000H  EXTRN  19#
OUTBFL. . V BYTE   0000H  EXTRN  20#
OUTBFP. . V WORD   0000H  EXTRN  19#
OUTBUF. . V BYTE   0000H  EXTRN  19#
PAGENO. . V WORD   0000H  EXTRN  15# 342 388
PARERR. . NUMBER   0002H    59#
POPREGS . V WORD   0000H  EXTRN  24#
PROMPT. . V BYTE   0000H  EXTRN  22#
PUSHREGS. V WORD   0000H  EXTRN  24#
RAM . . . V BYTE   0000H  EXTRN  12#
RCDENA. . NUMBER   0020H    74#
RDLEN . . V WORD   0000H  EXTRN  21#
REG1. . . NUMBER   000BH   137# 367
RSVD1 . . NUMBER   0008H    72#
RSVD2 . . NUMBER   0080H    76#
SCRBUF. . V BYTE   0000H  EXTRN  12#
SNDREG. . L NEAR   00DAH  CODE  166 225 277 313 362#
STACK . . SEGMENT         SIZE=0000H PARA STACK
STATER. . NUMBER   003CH   138# 170
STATUS. . V BYTE   0000H  EXTRN  16# 171 180 207 223 273 309
TIMERR. . NUMBER   0010H    62#
UNCERR. . NUMBER   0004H    60#
USERBP. . V WORD   0000H  EXTRN  25#
USERBX. . V WORD   0000H  EXTRN  25#
USERCS. . V WORD   0000H  EXTRN  26#
USERDS. . V WORD   0000H  EXTRN  25#
USERFL. . V WORD   0000H  EXTRN  26#
USERIP. . V WORD   0000H  EXTRN  26#
USERPC. . V WORD   0000H  EXTRN  27#
USERRG. . V WORD   0000H  EXTRN  23#
USERSP. . V WORD   0000H  EXTRN  26#
USERSS. . V WORD   0000H  EXTRN  25#
WAITEX. . L NEAR   002AH  CODE  199 206#
WAITPO. . L NEAR   0022H  CODE  201# 204
WBLENA. . NUMBER   0010H    73#
WRIT01. . L NEAR   007EH  CODE  281# 284 286
WRIT03. . L NEAR   0088H  CODE  291# 297 301
WRIT04. . L NEAR   0095H  CODE  294 299#
WRLEN . . V WORD   0000H  EXTRN  21#
WRTB01. . L NEAR   00B0H  CODE  318# 321
```

```
NAME        TYPE    VALUE   ATTRIBUTES, XREFS
ZAPREG. . . L NEAR  00BDH   CODE PUBLIC  131 337#

ASSEMBLY COMPLETE. NO ERRORS FOUND
```

December 1982

Powerfail Considerations for Magnetic Bubble Memories

Dick Pierce
Applications Engineer
Intel Corporation

December 1982
ORDER NUMBER: 210300-001

INTRODUCTION

Intel has developed a new, comprehensive power-fail circuit that is incorporated into all Intel Bubble Board Memory products: BPK 72 Bubble Memory Prototype Kit, iSBX™ 251 MULTIMODULE™ board, and the iSBC® 254 MULTIBUS® compatible board. The use of this circuit also is recommended for all customer-designed bubble memory boards. The overall performance enhancements offered by this circuit include improved noise immunity and a factor-of-four reduction in the time required to shut down the bubble system.

Scope and Organization

In an effort to focus on implementation details, this application note is organized so that a reader can obtain sufficient information to implement a bubble design without an intimate working knowledge of the powerfail circuitry. However, for those interested, a complete detailed explanation of the integrated powerfail circuitry and the additional external circuitry is included. Appendix A contains a technical discussion of the effects of power loss on a Magnetic Bubble Chip. In addition, the previous circuit versions (Revision 0 and Revision 1), along with the present circuit, are completely documented and compared in Appendix B.

Bubble Memory Operation and the Powerfail Function

The power-fail circuitry is partially integrated into two of the five MBM support components, and additional required circuitry is provided by external components. Historically, several evolutionary improvements have been made in the external circuitry (see Table 1) to further reduce the risk of data loss following an abrupt power failure.

An essential feature of the bubble memory (MBM) is non-volatile data storage. This non-volatility results from two permanent magnets within the bubble device that produce a magnetic field (bias field) that maintain the magnetic domains, or bubbles (representing data) in the chip even when power is removed. The bubbles remain stationary in fixed positions until the data is accessed. To move the bubbles, an in-plane rotating magnetic field is induced by pulsing two mutually-perpendicular coils surrounding the bubble chip. Special conductor lines on the bubble chips provide all the current related functions for reading and writing to the bubble device. A special support IC produces current pulses (swap, relicate, and generate) to perform these functions. A complete set of support circuits provides the necessary timing and waveforms to precisely maneuver the bubbles to their desired positions. To prevent bubbles from moving to undesired positions, certain precautions must be observed.

As power is applied or removed, the system must prevent any current transients in the coils or bubble function conductors. If power is removed with the coils operating, the system must ensure that the coil currents are shut down in an orderly fashion to guarantee that the magnetic bubbles come to rest in stable, known positions. The powerfail reset circuit ensures that the system is powered up in an orderly manner and serves to alert the system should power fail. Both the power-up and

Table 1. Powerfail Reset Circuit Product History

Product	Powerfail Circuit Revision Level		
	0	1	2
BPK 72	July 1979 thru August 1982 Rev. A thru Rev. G	N/A	September 1982
iSBX™-251 Board	N/A	September 1981 thru October 1982	November 1982
iSBX-251C Board	N/A	N/A	July 1982
iSBC®-254 Board	December 1980 thru July 1982	July 1982 thru November 1982	November 1982

power-down sequences require a finite period of time to complete their functions until the sequence is complete. To allow proper execution of a power down sequence, the system voltages (+5V DC, +12V DC) must not decay to a level that prevents operation of the powerfail circuitry and critical bubble memory functions. In most power supply designs, adequate energy storage is available to provide enough "hold time" to complete an orderly shutdown. However, if dc power decays too rapidly sufficient time may not exist for a proper shutdown and may cause data to be lost within the MBM.

System Description

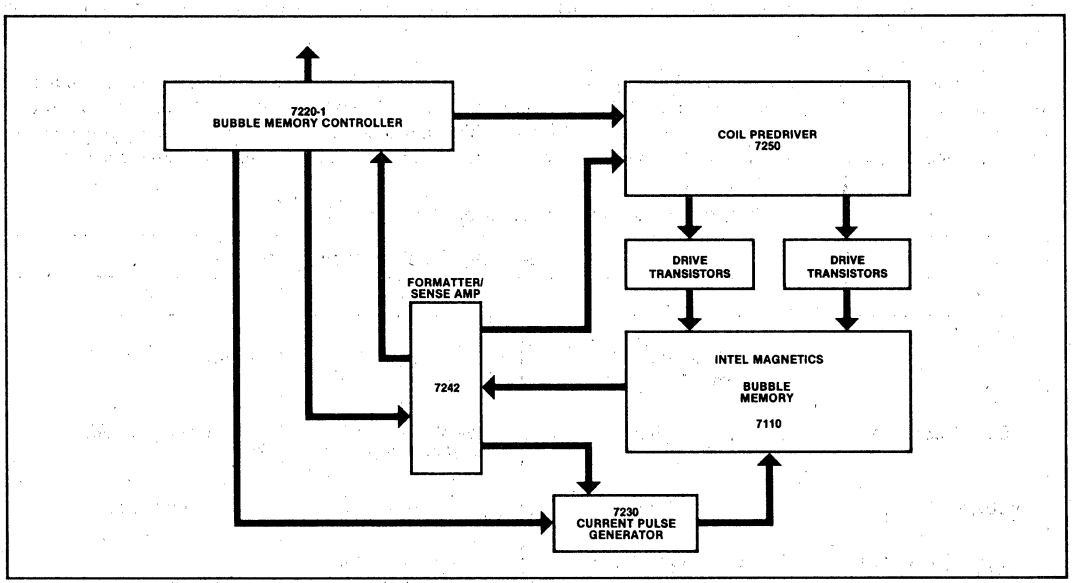
The basic Intel Bubble Memory system consists of one 7110 magnetic bubble memory and five integrated support components: a 7220-1 Bubble Memory Controller (BMC), a 7230 Current Pulse Generator (CPG), a 7242 Formatter-Sense Amplifier (FSA), a 7250 Coil Predriver (CPD), and two 7254 quad drive transistor packages. These support circuits are interfaced to the MBM as shown in Figure 1 to form the basic one megabit (128K byte) system. The support components provide all of the functions necessary for the storage and retrieval of data within the MBM. In addition, two of the support components, the 7220-1 BMC and the 7230 CPG, contain the integrated powerfail circuitry that facilitates proper power-up and power-down operations.

OVERVIEW — POWER UP/DOWN OPERATION

A block diagram of the power fail circuitry for the bubble memory system is shown in Figure 2. The following paragraphs provide an operational overview of the integrated powerfail circuit and the external circuit requirements.

During a power up sequence, the 7230 holds PWR.FAIL/* active (low) until both supplies are above the minimum required level. The 7230 contains power supply monitors (+5V and +12V) that determine when either supply falls below threshold level and activate PWR.FAIL/ signal accordingly. On power-up, the PWR.FAIL/ signal is delayed an additional 2 msec by an external RC network (time delay 1) to allow the 7220-1 substrate bias generator to fully charge. Following this delay, the positive-going transition on the 7220-1 PWR.FAIL/ input initiates a 7220-1 power-up sequence.

The RESET.OUT/ signal was designed to remain active during the power-up sequence and then to go inactive at the conclusion of the 50 μs power-up sequence. However, the RESET.OUT/ signal is indeterminate during execution of the 7220-1 power-up sequence. A second external RC network (time delay 2) derived from PWR.FAIL/ ensures that RESET.OUT/ is



*"/" denotes an inactive signal.

Figure 1. System Block Diagram

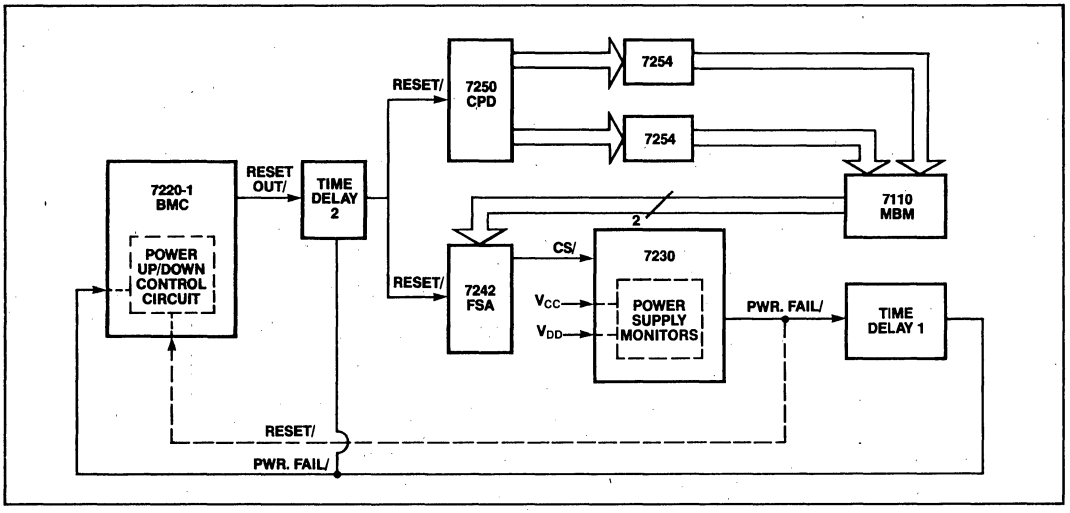


Figure 2. Block Diagram of Powerfail

held active ($\leq$0.8V) during this time. The RESET.OUT/ signal occassionally will remain in its active state following a power-up sequence; accordingly the first command issued to the BMC during an initialization sequence must be an Abort command to ensure that RESET.OUT/ is deactivated.

The power-up sequence is designed to power the system up in an orderly fashion and to prevent any current transients from reaching the bubble device. The power-down sequence ensures that the coil drivers are shut down in the proper phase and that the support circuits are reset. When power fails, the 7230 notifies the 7220-1 by asserting the PWR.FAIL/ signal. The 7220-1 responds to a negative transition on either the PWR.FAIL/ input or the RESET/ input (external circuit revision level dependent) and initiates a power-down sequence. If the coils are active (i.e., bubbles propagating), the 7220-1 first terminates the coil drive control signals during the appropriate phase and then resets the support circuits by asserting the RESET.OUT/ signal. The two system supply voltages must not decay faster than the specified rates to ensure the RESET/ input to all the support circuits (excluding the 7220-1) reaches an active level (less than 0.8 volts).

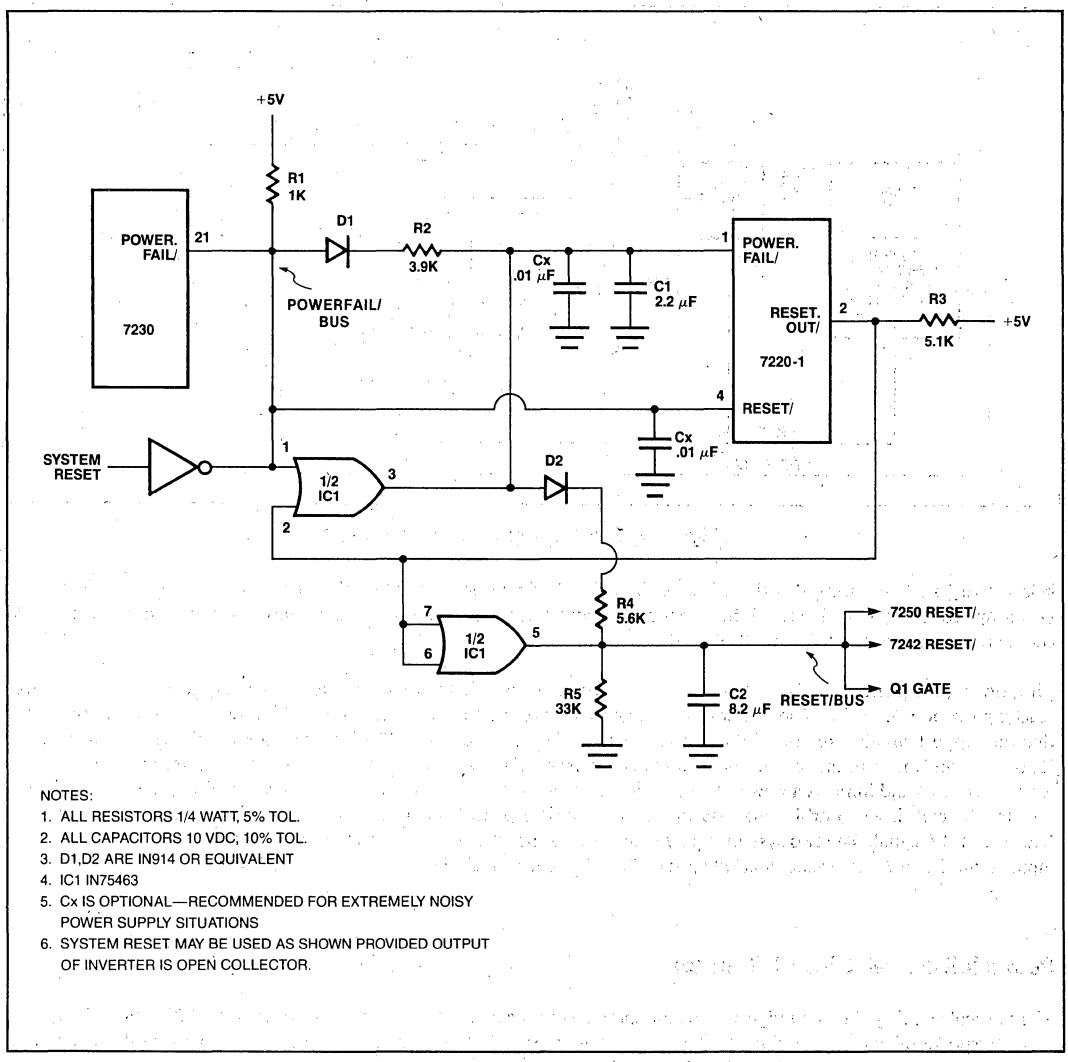
Powerfail Reset Circuit Solution

The external circuitry shown in Figure 3, in conjunction with the integrated circuitry contained in the 7230 and 7220-1, comprises the powerfail circuit (revision 2). This design contains six additional components compared to previous powerfail circuits and includes an 8-pin DIP IC (TI 75463).

This revised circuit has been fully developed and tested by Intel and currently is incorporated in many bubble products. Operational details are not required for the user to implement a custom design using the circuit in Figure 3. However, for any bubble memory designs that cannot conform to the recommended powerfail circuit, a reader must understand the system characteristics and requirements prior to choosing an alternative design.

The software implementation details to ensure correct powerfail circuit operation are shown in Figure 4. This routine should be implemented as a routine for cold start operation (application of power) and warm start operation (a RESET/ pulse applied to the 7220-1 BMC). The voltage decay rates shown in Table 2 also cannot be exceeded.

The power-up routine is based on the typical power-up timing shown in Figure 5. This timing does not assume that a system reset has been incorporated into the powerfail circuit. If the hardware reset line is used, the user must ensure that the 7220-1 RESET/ input is inactive before issuing the first Abort command. In addition, user software always must issue an Abort command every time the system is reset.



NOTES:
1. ALL RESISTORS 1/4 WATT, 5% TOL.
2. ALL CAPACITORS 10 VDC, 10% TOL.
3. D1,D2 ARE IN914 OR EQUIVALENT
4. IC1 IN75463
5. Cx IS OPTIONAL—RECOMMENDED FOR EXTREMELY NOISY POWER SUPPLY SITUATIONS
6. SYSTEM RESET MAY BE USED AS SHOWN PROVIDED OUTPUT OF INVERTER IS OPEN COLLECTOR.

Figure 3. External Powerfail Circuit Solution

Table 2. Power Supply Decay Rate Specifications During Power-down or Power Failure

Power Down/Powerfail Decay Rate			
V_{CC} (volts/msec)		V_{DD} (volts/msec)	
Min.	Max.	Min.	Max.
None	0.45	None	1.1

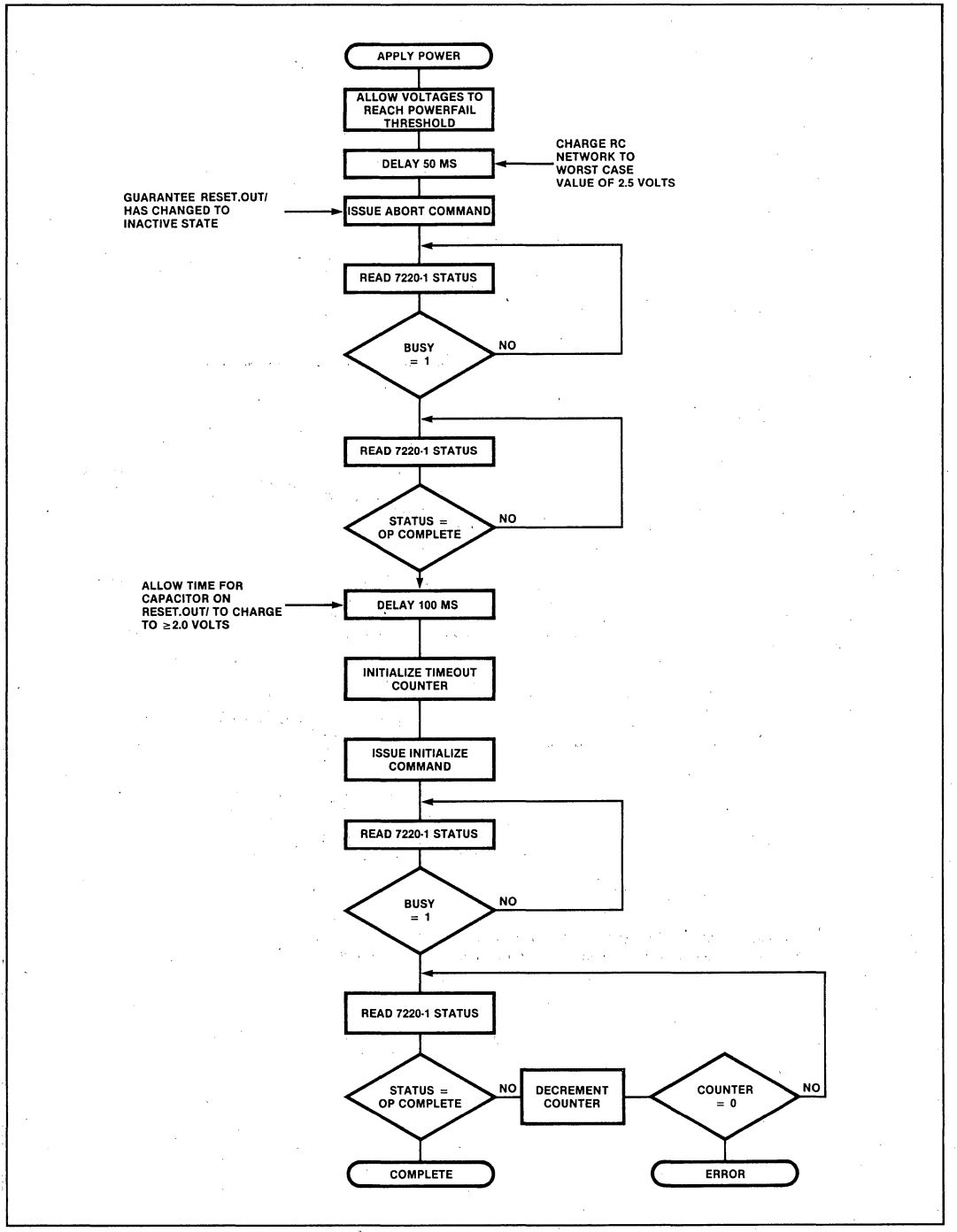
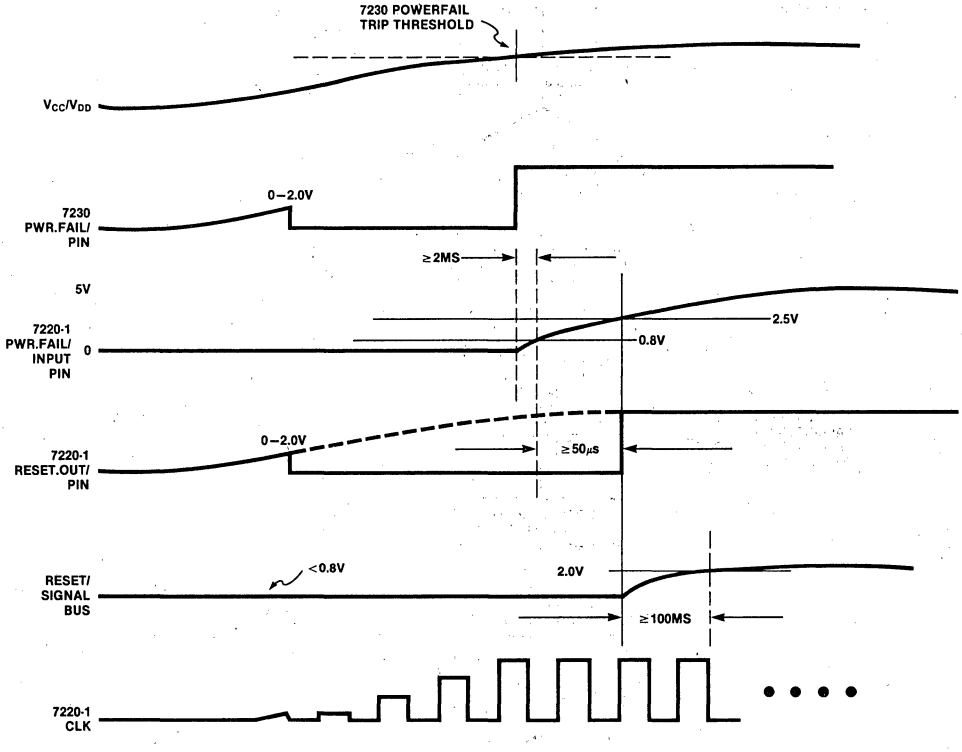


Figure 4. Power-up Flowchart



NOTES:
1. 7220-1 INPUTS SHOULD NOT LEAD V_{CC} BY GREATER THAN 0.5 VOLTS.
2. THE ABOVE SEQUENCE APPLIES ONLY IF THE SYSTEM RESET INPUT IS INACTIVE.

Figure 5. Power-up Timing for Powerfail Reset Circuit (Revision 2)

The worst case power-down timing sequence is also included in Figure 6. The total system power-down time varies according to whether the coils are active (i.e., rotating magnetic field is on) or inactive. The worst case power-down sequence is guaranteed to be completed provided that the above voltage decay rates are met.

INTEGRATED POWERFAIL CIRCUIT CHARACTERISTICS

Introduction

The following section provides an in-depth look at the input and output characteristics of the support circuits that contain the integrated powerfail circuitry. A complete understanding of these characteristics establishes the groundwork necessary for the detailed description of the overall powerfail circuit operation that follows.

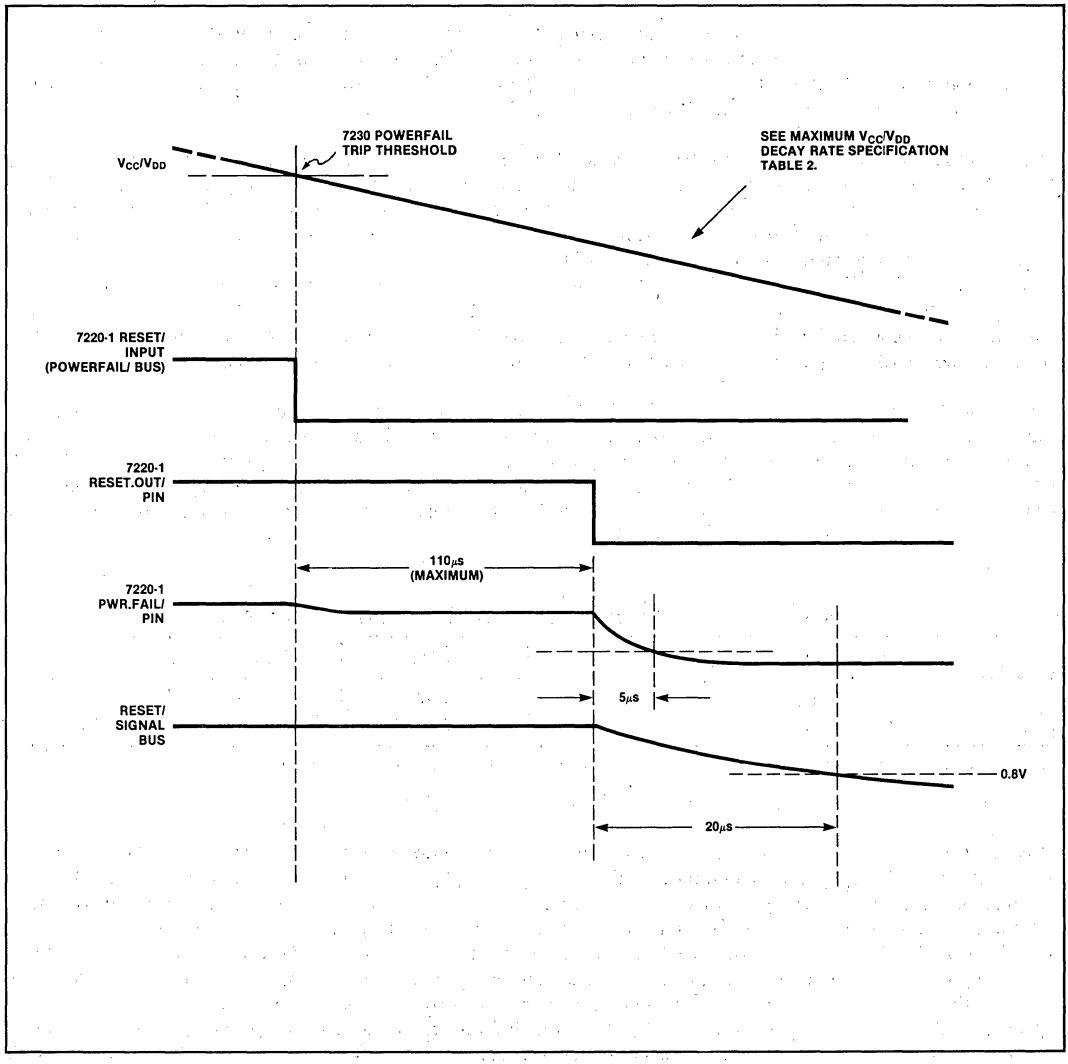


Figure 6. Power-down Timing for Powerfail Reset Circuit (Revision 2)

7230 PWR.FAIL/ OUTPUT

The 7230 Current Pulse Generator PWR.FAIL/ output is responsible for indicating when the system supply voltages (+5V, +12V) reach correct operating levels. During power up, normal operation, and power down, an internal zener reference comparator circuit within the 7230 senses both V_{CC} and V_{DD} and indicates when both levels are above approximately 92 percent of their nominal values. An active state on PWR.FAIL/ indicates one or both dc voltages are below this threshold. The PWR.FAIL/ output is an active-low, open-collector output requiring an external pullup resistor.

The PWR.FAIL/ output is asserted (active low) as power is applied until the +5V and +12V supplies both reach approximately their 92 percent levels at which point the 7230 output transistor switches off to allow the PWR.FAIL/ signal to rise to an inactive level governed by an external RC network. The RC networks on the PWR.FAIL/ line must hold the PWR.FAIL/ signal at an active level for at least 2.0 milliseconds to guarantee adequate time for the BMC to power up. The 7230 PWR.FAIL/ output then will remain inactive until one or both system voltages fall below the threshold.

The PWR.FAIL/ output is not an internally latched signal. In other words, the output responds immediately to any transition through the threshold (trip point). The disadvantage to this excellent response capability is that the output will toggle on transitions through the threshold. Systems should be designed to avoid an extremely noisy power supply or temporary power loss that could cause the PWR.FAIL/ signal to pulse for a very short duration.

During temporary power loss in Revision 0 and Revision 1 circuits, the PWR.FAIL/ input to the 7220-1 could pulse below V_{IH} (2.5 volts) and initiate a power down sequence. The 7220-1 PWR.FAIL/ input should remain active until the entire power down sequence is completed (maximum 110 μsec). As detailed later in the 7220-1 PWR.FAIL/ input description, if the 7220-1 PWR.FAIL/ input goes inactive during execution of a power down sequence, the sequence is immediately terminated. This type of termination can stop the drive field in the wrong phase and compromise bubble data. The solution is to use the 7220-1 RESET/ input to initiate a power down sequence rather than the 7220-1 PWR.FAIL/ input.

Two important considerations in properly designing a powerfail circuit are 1) the accuracy of threshold trip point of the 7230 PWR.FAIL/ output and 2) the behavior of this output at low voltages (below 2 volts).

The worst case threshold level that the 7230 PWR.FAIL/ output will trip must be above the worst case operating limits of the support circuits with an additional margin to allow for an adequate period of time to complete a power down sequence (worst case 110 microseconds for revision level 1 and 2 powerfail reset circuits). In the case of the 7230 CPG and the 7110 MBM, which both have a ±5% voltage specification for V_{CC} and/or V_{DD}, special powerfail characteristics are applicable. As shown below, (Table 3) only critical bubble memory functions are guaranteed at these supply values and not full memory operation.

Table 3. Powerfail Characteristics for 7230 Threshold Trip Point*

Symbol	Min.	Typ.	Max.
V_{CC} TH	4.43V	4.60V	4.70V
V_{DD} TH	10.75V	11.10V	11.28V

*Powerfail characteristics apply to 7110 bubble memory data integrity only and not to full memory operation.

Second, the 7230 PWR.FAIL/ output cannot be guaranteed active (low) until V_{CC} reaches about 2.0 volts since the output transistor is not operational until that point. As V_{CC} is applied, the output is not active and will track (follow within a few tenths of a volt) V_{CC} until V_{CC} reaches approximately 2.0 volts. At this point, the output transistor turns on and the output goes active (low) and remains low until the system voltages both reach the threshold trip point as described earlier. A similar response occurs as power is removed. The output transistor turns on and pulls the output active (low) at the threshold point and remains turned on until V_{CC} reaches approximately 2.0 volts where the output goes inactive (transistor not operating). This operation must be controlled on power-up and depends on the rate of rise of system voltages. This is because the PWR.FAIL/ output is indirectly connected to the RESET/ input of the support circuits (7250 and 7242 and Q1 reference current switch) through two RC networks in Rev. 0 and Rev. 1 power-fail circuits. These inputs can rise to as much as 1.5V before the 7230 PWR.FAIL/ output turns on, which is above V_{IL} max-

imum (0.8V) thus potentially enabling these circuits. This could result in current transients reaching the drive coils or bubble function conductors and move bubbles from their rest position resulting in data loss. Observing the rate of rise specifications protects against this possibility . The revision 2 powerfail circuit eliminates this problem and has no rate of rise limitation.

7220-1 PWR.FAIL/ INPUT

The 7220-1 PWR.FAIL/ input serves a dual function; a positive transition initiates a power-up sequence while a negative transition initiates a power-down sequence of the bubble memory system. In order for the 7220-1 to become fully functional an on chip back-bias generator must fully charge the 7220-1 substrate. Therefore, before any sequence can be executed, including the power-up sequence a time delay is required. An external RC delay on the PWR.FAIL/ input ensures this input is held low (<0.8V) at least 2.0 milliseconds after V_{CC} has reached the 7220-1 voltage specification range.

The power-up sequence is initiated once the RC network charges to a point where the 7220-1 recognizes a positive transition on the PWR.FAIL/ input. From a cold start (application of power), a positive transition must occur or the controller will not power-up correctly. Once the power-up sequence is completed, the RESET.OUT/ is designed to be released, however, two possible exceptions exist. First, if the 7220-1 RESET/ is held low during power-up, the 7220-1 internal power-up sequence will be completed however RESET.OUT/ will not be released until RESET/ is inactive. Second, the 7220-1's internal RESET.OUT/ output transistor may remain turned on dependent upon the power-up status of certain internal 7220-1 flip-flops. Because of this an ABORT command is always necessary to internally reset these flip-flops, in turn ensuring release of the RESET.OUT/ output.

If the 7220-1 BMC does not receive a positive transition on PWR.FAIL/ during power-up, a power-up sequence is not initiated. This leaves the controller in an unknown state. In this unknown state the controller cannot communicate properly with the data and control inputs. This can only occur as a result of:

1. **"Brown out"** — short duration of power failure in which power drops below specified levels.

2. **Power-up circuit failure** — The PWR.FAIL/ pin never reaches V_{IH} (minimum) of 2.5 volts.

The above conditions are resolved by ensuring a positive transition occurs on the PWR.FAIL/ input during power-up and after brownout. It is necessary to execute a power-up sequence even though power to the system is only interrupted momentarily in order to restore the 7220-1 to the required internal state.

Once the PWR.FAIL/ positive transition has occured, this input should remain in the inactive state (V_{IH} > 2.5V) as long as power is applied to the system. If power is removed, it is the negative transition of this input which intitiates the second function, power down. The function can also be initiated with the RESET/ input of the 7220-1.

An important consideration is how the 7220-1 PWR.FAIL/ input distinguishes between positive and negative transitions. On power up (positive transition), crossing the input threshold (typically 1.6V to 1.9V) a pulse is generated internally which resets the 7220-1 to a known state and initiates a power-up sequence. On power down (negative transition), crossing the input threshold (typically 1.35V to 1.6V with the designed- in hysteresis) the signal initiates a power-down sequence. If a power-down sequence has been initiated, a positive transition must not inadvertently occur on the 7220-1 PWR.FAIL/ input prior to the power-down sequence completion. A positive transition internally generates a reset pulse (to halt any current BMC activity) and initiates a power-up sequence effectively terminating a power down sequence. The result is a possibility of shutting the coil drives down in the improper phase resulting in data loss in the MBM.

The PWR.FAIL/ input has built in hysteresis to reduce the susceptibility to multiple threshold crossings or glitching. However, the values of hysteresis range from 50 mV to 400 mV. To improve noise and power fluctuation immunity, the use of PWR.FAIL/ input for initiating a power down sequence was abandoned in Revision 1 and Revision 2 circuit designs. The 7220-1 RESET/ input is used instead to initiate power down (see next section.)

7220-1 RESET/ INPUT

The 7220-1 RESET/ input, when asserted, will terminate any current BMC activity and initiate a RESET sequence (identical to the sequence initiated by the PWR.FAIL/ input going active). After the sequence is concluded, the RESET.OUT/ is activated to reset the MBM support circuitry. RESET.OUT/ will remain active until RESET/ is inactive.

The RESET/ input is a level sensitive latched input. This is a distinct advantage over the PWR.FAIL/ input; where any fluctuations of the input once the signal was recognized could possibly terminate the power down sequence. The RESET/ input is latched on the negative edge of the BMC clock and must be active low (<.8V) for at least one clock period (250ns) to guarantee recognition.

7220-1 RESET.OUT/

The RESET.OUT/ output has two functions: 1) to guarantee the bubble memory system is disabled during power-up and after power down of the bubble memory system and 2) to provide a pulse (reset) to the support circuits during normal operation. Since the RESET.OUT/ output is an active low open drain, it requires an external pullup resistor to V_{CC}.

The support circuits controlled by RESET.OUT/ are the 7250 Coil Predriver, the 7242 Formatter Sense Amplifier, and a VMOS transistor switch which enables a reference current for the 7230. These circuits must be disabled during the entire power-up sequence and immediately following the conclusion of a power-down sequence to prevent any current transients or extraneous enable pulses. Data loss is a possible consequence should the support circuits not remain disabled during power cycling.

During power up the RESET.OUT/ signal can not be guaranteed active (low) until the 7220-1 power-up sequence has executed. Therefore, external circuitry must assure RESET.OUT/ does not rise above V_{IL} maximum (.8V) until 50 μs after initiation of the power-up sequence. By ensuring the RESET.OUT/ is active during power-up it guarantees the support circuits are reset to a known state. The 7220-1 BMC is designed with the capability to reset the support circuits during normal operations by pulsing the RESET.OUT/ 750 μs (3 clock periods). This pulse can occur as the result of two user issued commands to the BMC: an INITIALIZE command and an MBM PURGE command.

The external RC network on the RESET.OUT/ signal prevents the RESET.OUT/ pulse from going active during its 750 μs duration. In spite of an inability to reset the support circuits by issuing the proper command, correct operation is guaranteed since the support circuits only require a one time reset signal at power-on.

Additional Bubble Memory Controller Inputs

The 7220-1 has several additional inputs that could indirectly affect power up operation. It is important that the user exercise caution and adhere to all requirements to ensure proper power-up operations. The following outlines those requirements.

CLK (CLOCK)

The CLK input of the 7220-1 must be present when the positive power up transition occurs at the 7220-1 PWR.FAIL/ input. This requirement allows the BMC to properly execute a power-up sequence. The input requirements are a precise 4MHz (±.1%) with a 50 percent duty cycle (±5%).

DACK/ (DATA ACKNOWLEDGE)

The DACK/ input is normally used in conjunction with an INTEL DMA controller chip (8257 or 8237) which automatically provides drive for this input. However, if DMA is not used a 5.1K pullup resistor to V_{CC} is required. This requirement prevents erratic BMC operation.

WAIT/

The WAIT/ input must also be guaranteed inactive through an external 5.1K pullup resistor. It is designed to be used in parallel controller applications to maintain synchronization between controllers should an error be detected in one during a data transfer.

CS/, RD/, WR/, A0, D0-D8

These inputs require no special considerations other than to observe the V_{IH} minimum specification. This specification prevents an incorrect power-up sequence execution.

ENERGY STORAGE REQUIREMENTS

The data integrity and non-volatility of the MBM during power down operations is guaranteed by design provided the voltage decay rates specifications for both V_{CC} and V_{DD} are observed. Most commercially available power supplies provide enough energy storage to fulfill these requirements. However, some applications may exist where the bubble memory could suddenly become disconnected from the dc supply; a case where the power supply energy storage is not of value. In these special applications, the local onboard capacitance must meet the hold up time requirement.

The worst case onboard capacitance values can be determined according to the following equation:

$$C = \frac{Q\ max}{V\ min} = \frac{I\ max\ \Delta T\ max}{\Delta V\ min}$$

A worst case calculation must include the following considerations: 1) If any additional circuitry exists on the pc board that uses the same power supplies, the additional current drain must be accounted for and 2) the worst case (minimum) threshold trip point of the 7230 is used.

The capacitance required on a pc board containing one / megabit bubble memory system is calculated as follows:

$$C_{5V} = \frac{366 \times 10^{-3}\ amp \times (110 \times 10^{-6}\ sec)}{0.01 \quad x\ 5\ volts} = 805\ \mu F$$

$$C_{12V} = \frac{381 \times 10^{-3}\ amp \times (110 \times 10^{-6}\ sec)}{0.01 \quad x\ 12\ volts} = 350\ \mu F$$

Supplemental Powerfail Sensing

In many systems, additional signals are available that provide advanced warning of an imminent power failure or the existence of an abnormal condition prior to actual loss of dc power (e.g., AC powerfail sensing, AC or DC overvoltage, ambient over/under temperature). These signals are easily incorporated into the powerfail circuit design via an open-collector gate or inverter connected to the PWR.FAIL/ signal bus.

The advantage of utilizing these signals is the bubble memory system can complete a power down sequence prior to losing dc power. However, local dc powerfail sensing is always required due to the possibility of local dc power loss without the loss of AC power.

Noise Effects of Powerfail Circuit Operation

The 7230's powerfail voltage monitoring function is implemented internally with two independent, logically-OR'ed voltage comparators. The comparators respond quickly to a sudden loss of V_{CC} or V_{DD} and therefore can respond to noise transients on the power supply lines that cross the comparator switching threshold. As much as 100 mV of noise

from coil drive switching is not uncommon. Note that the operating power supply tolerance for all INTEL Bubble Memory products is ±5% including up to 50 mV of noise on the power supply lines. This tolerance should not be confused with the operation of the powerfail circuit beyond the normal operating range during power-down operation.

To minimize "nuisance" activation of the PWR.FAIL/ signal bus, ample high frequency decoupling on the 7230's V_{CC} and V_{DD} pins should be provided. Typically, 0.01 μF to 0.1 μF ceramic disk or mica capacitors are sufficient. Another source of unwanted powerfail circuit activation is noise that is coupled directly onto the PWR.FAIL/ signal bus. This noise is minimized through good printed circuit layout practices and, if required, by the inclusion of a small capacitor directly on the PWR.FAIL/ bus. This capacitor slightly increases the power-down time and should be kept as small as possible (0.01 μF maximum).

APPENDIX A

TECHNICAL DISCUSSION OF POWER LOSS EFFECT ON 7110

The effects of power loss on an MBM are best understood by describing the way in which the device functions and the way in which it can lose data.

A magnetic bubble memory device (See Figure 7) consists of a bubble memory chip, two mutually-perpendicular coils, two permanent magnets, and a shield to provide protection from interference by external magnetic fields. The two permanent magnets produce an external magnetic field (bias field) that maintains the magnetic domains, or bubbles, in the chip even when power is removed. To move the bubbles, an in-plane rotating magnetic field is induced by pulsing the two mutually-perpendicular coils.

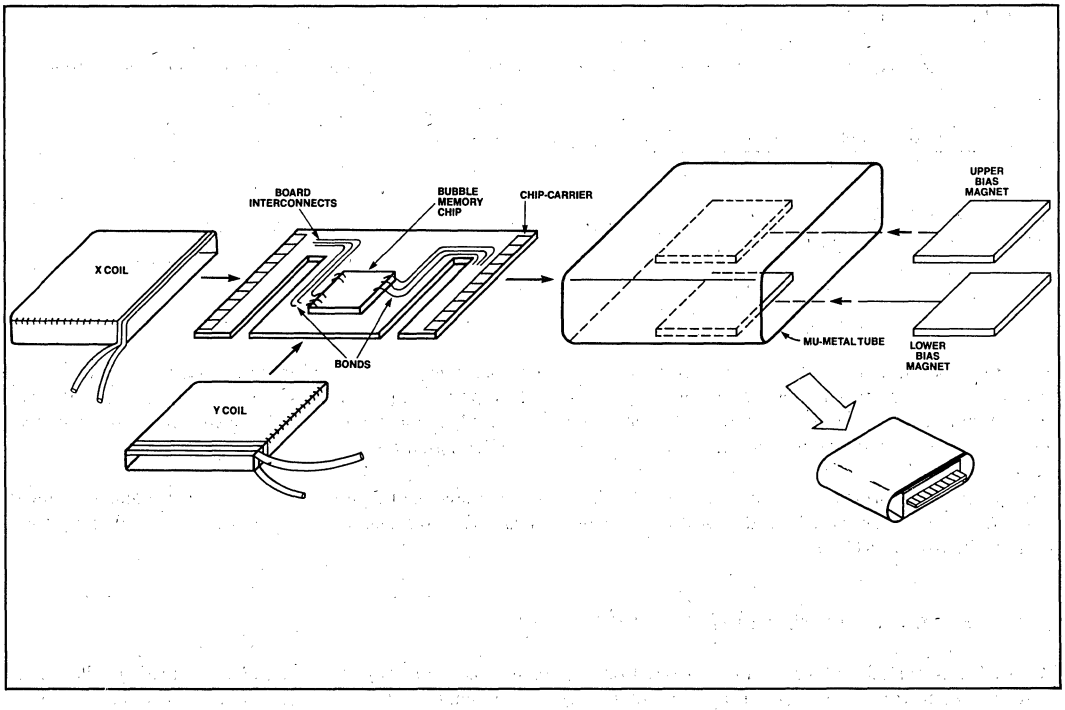


Figure 7. Device Break-down

The bubble memory chip itself consists of a thin magnetic garnet crystal film grown on a non-magnetic gadolinium-gallium-garnet substrate. This thin film possesses a property that magnetic moments associated with each atom in the single crystal structure have only two possible directions: an upward or downward direction perpendicular to the plane of the film. This constraint in direction results in only two conditions of magnetization (see Figure 8). These magnetic moments tend to group themselves together into magnetic domains. The size and shape of the domains are determined primarily by a balancing of several forces that minimize the sum of magnetic energy.

Without an external field, the film surface area of upward domains is equal to that of downward domains and there is no net magnetic field within the plane of film. Application of an external magnetic field perpendicular to the film causes domains to line up in the direction of the field. As the external field is increased, the downward domains enlarge while the opposing (upward) domains shrink until they finally are reduced to a cylindrical shape. This microscopic magnetized cylinder opposing the externally applied field is a magnetic bubble. Within the magnetic film, the presence of a magnetic bubble represents a binary one and the absence of a magnetic bubble represents a binary zero.

The memory function is provided by the bubble. However, an organized means is needed to propagate the bubbles along certain paths and to provide storage sites. A soft ferromagnetic material (permalloy) is deposited on the thin garnet film in C-shaped patterns. These patterns are arranged to form shift-register like loops that provide the means to store and move bubbles. Each pattern is magnetized according to the rotating magnetic field, and the polarity of each pattern changes instantaneously as the rotating magnetic field vector changes. The rotating field is generated by driving the X and Y coils with triangular-waveform currents, one lagging the other by 90° in phase. A magnetic bubble propagates from one storage site (permalloy pattern) to the next for every 360° of rotation of the rotating field. Each storage site has a preferred position (home) for the bubble to reside corresponding to zero degrees of the rotating magnetic field. All bubbles start, stop and are stored in this position.

In the event of power failure, it is important that the rotating magnetic field is shut down in the proper phase (i.e., 0°). If an orderly shut down is not complete, the rotating field may be shut down in an improper phase that causes bubbles to stop in an unstable position within the storage loops. When this type of stoppage occurs, the bubbles either will come to rest in another, but incorrect, stable position or will leave their original storage loop (possibly contaminating valid data in another storage loop).

As power is applied, it also is important that the rotating magnetic field does not move (i.e., current transients must be prevented from reaching the coils). This function also is provided via the powerfail circuitry. Thus, the purpose of the powerfail circuitry is twofold 1) to prevent any current transients from reaching the X-Y coils or bubble function generators and 2) to halt the coils in proper phase should power fail.

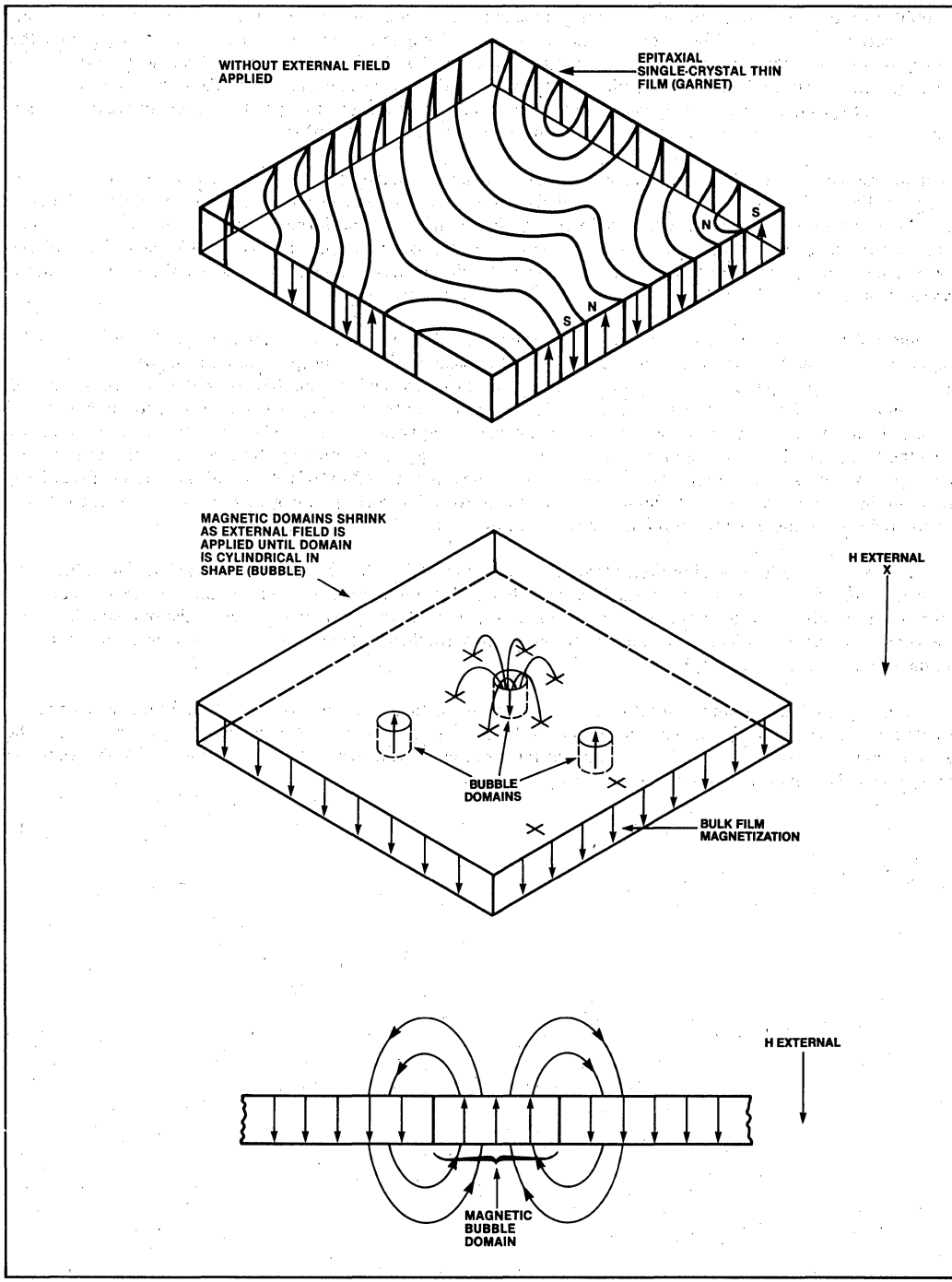


Figure 8. Device Magnetization

APPENDIX B

DETAIL POWER CIRCUIT DESCRIPTION

As discussed in the Introduction, the powerfail reset circuit actually consists of two portions — an integrated section and several additional external components. The degree to which external disturbances (noise, power fluctuations) influence system performance depends heavily on the system environment and configuration. Consequently, the reliable analysis of their effect on system performance is difficult and generally is best accomplished by measurement. In this Appendix, each revision level of the powerfail reset circuit is detailed. Several timing diagrams based on measurement and computer simulation also are included.

Powerfail Reset Circuit — Revision 0

Summary

The overall performance of the powerfail reset circuit (revision 0) is adequate provided that a specific set of conditions is observed. The requirements are summarized below (Table 4). Noise is also a concern. System generated noise is typically low level and can usually be neglected in portions of the circuit where the signal levels are high. Often, however, bubble systems generate significant levels of noise in a system where signal levels are low. Even low-level noise can degrade overall bubble memory system performance.

Table 4. Power Supply Requirements for Powerfail Reset Circuit (Revision 0)

	V_{CC} (volts/msec)		V_{DD} (volts/msec)	
	Min.	**Max.**	**Min.**	**Max.**
Power-Up Voltage Rate of Rise	0.11	None	None	None
Power-Down/Power Failure Decay Rate	None	0.70	None	.15

Noise, power fluctuations, and a rapid decay of voltage are the primary contributors to the incorrect operation of the first powerfail reset circuit (revision level 0). Since noise and power fluctuations are unavoidable in most practical systems, techniques for minimizing these effects were developed for subsequent circuits. Note that no bubble memory is immune to extremely abrupt removal of dc power. All bubble memory systems require a minimal amount of time to effect an orderly shutdown in order to maintain data integrity.

Subsequent circuit designs have been implemented to minimize system requirements by reducing the overhead required to power-down the bubble system.

The most serious fault of any powerfail reset circuit is where bubble memory data integrity is jeopardized. The first powerfail reset circuit design (revision 0) could not prevent data loss when:

1) Power was removed too rapidly for the system to ensure proper power-down.

2) Power was applied too slowly.

3) Multiple threshold crossings or "glitches" occured on the 7220-1 PWR.FAIL/ input while the coils were active.

The first two conditions can be easily prevented by following the requirements shown in Table 4. The third condition was difficult to reliably prevent and was the motivation for the revision of the circuit.

Power-up

When power initially is applied to the system (Figure 9), the PWR.FAIL/ signal is designed to be asserted by the 7230 CPG until both V_{CC} and V_{DD} reach approximately 92 percent of their nominal values. Referring to Figures 9 and 10, the 7230 internal PWR.FAIL/ output transistor cannot be guaranteed operational until V_{CC} reaches approximately 2.0 volts. During this indeterminate state of the output transistor, the floating output lags V_{CC} by approximately 0.7 volts. Therefore, the RC networks on the PWR.FAIL/ signal line (R1/C1 and R2/C2) begin charging immediately after power is applied. They continue to charge until the 7230 PWR.FAIL/ output transistor turns on. The 7230 PWR.FAIL/ output goes inactive (transistor off) when both supplies have reached the power-fail trip point. Since the RESET/ input of the 7242 FSA and the 7250 CPD are tied via the R1C1/R2C2 network to 7230 PWR.FAIL/ output, these support circuits potentially could be enabled if the 7230 PWR.FAIL/ output were allowed to rise above V_{IL} (0.8 volts). A current transient then could activate the MBM coils or bubble function conductors and cause bubbles to move to an unstable position. Note that a slow power-on ramp would be the only condition that could prematurely enable the support circuits.

Once V_{CC} reaches approximately 2.0 volts, the PWR.FAIL/ output transistor turns on to pull the PWR.FAIL/ signal low until both V_{CC} and V_{DD} reach the powerfail trip point. When the trip point is reached, the output transistor is turned-off and the PWR.FAIL/ signal is allowed to rise to the inactive level. The RC networks continue to hold the PWR.FAIL/ signal at an active level for at least 2.0 milliseconds after V_{CC} and V_{DD} have reached the trip point level. The RC delay ensures adequate time for the 7220-1 BMC's substrate bias generator to become fully operational and fully charge the 7220-1 substrate to its operational bias voltage. At some time before the PWR.FAIL/ signal reaches the 7220-1 V_{IH} (maximum) of 2.5 volts, the 7220-1 power-on initialization sequence starts. Up to this point, the 7220-1 is in an indeterminate state and the RESET.OUT/ signal, which is derived from the PWR.FAIL/ signal should be active. The behavior of the RESET.OUT/ signal, however, is similar to the 7230 PWR.FAIL/ output at low V_{CC} (below approximately 2.0 volts). As V_{CC} is slowly applied to the system, the RESET.OUT/ output transistor initially is inactive and the pullup resistor forces this output to follow 7220-1 PWR.FAIL input. Once V_{CC} reaches approximately 1.8 volts, the output transistor should turn on (RESET.OUT/ active) and remain active until completion of the power up sequence. During the inactive period, the RESET.OUT/ signal is capable of reaching the inactive level and potentially enabling the support circuits prematurely.

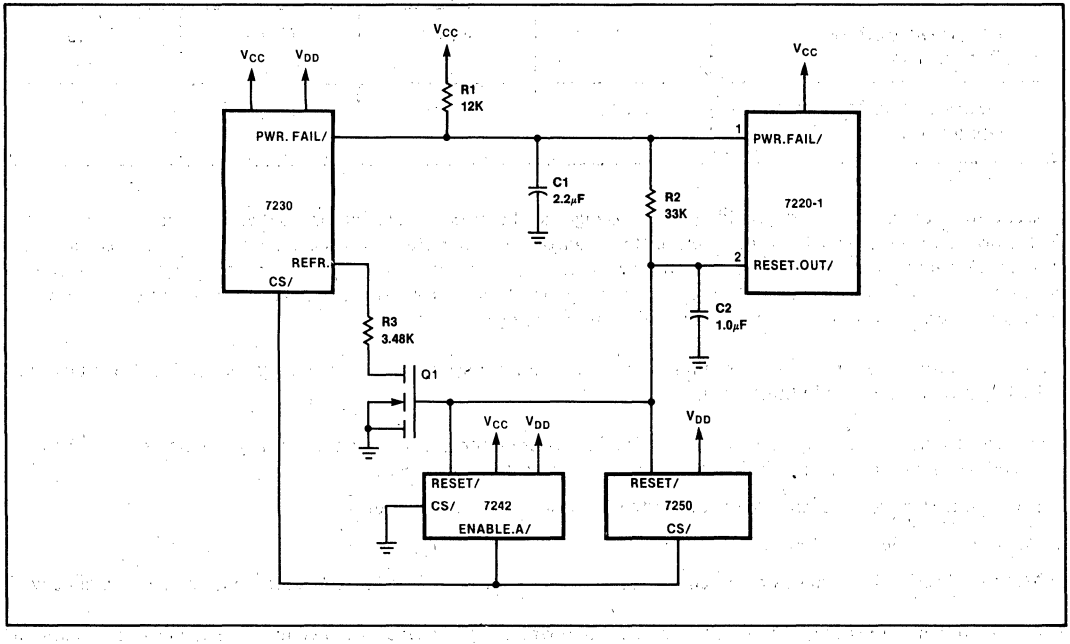
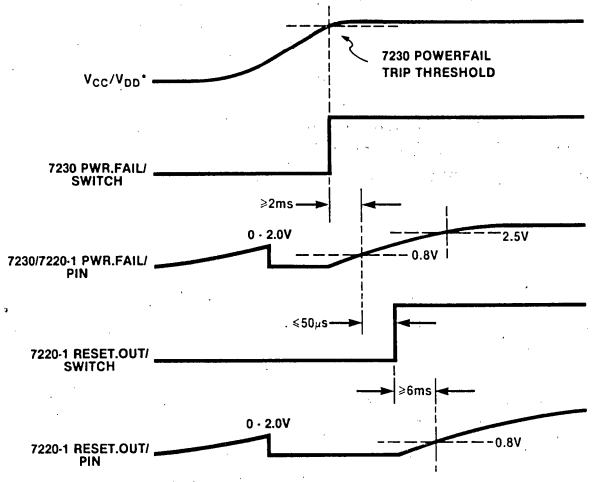
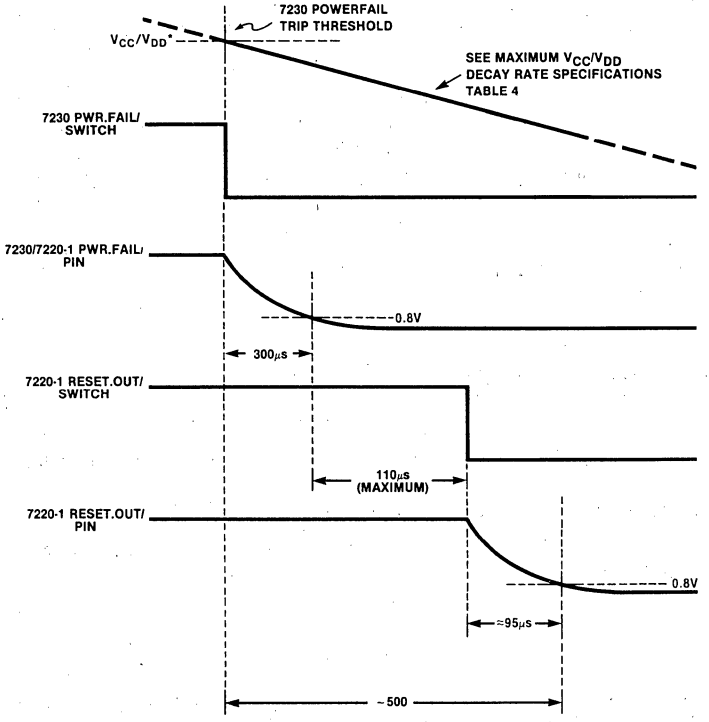


Figure 9. Revision 0 Circuit



Power-up Timing

*V_{CC} OR V_{DD}, WHICHEVER IS LAST TO CROSS POWERFAIL TRIP THRESHOLD.



Power-down Timing

*V_{CC} OR V_{DD}, WHICHEVER IS FIRST TO CROSS POWERFAIL TRIP THRESHOLD.

Figure 10. Power-up/Power-down Timing (Revision 0)

At the completion of the power-on initialization sequence, the 7220-1's internal RESET.OUT/ output transistor should be allowed to turn off. However, depending on the power-up state of certain internal 7220-1 flip-flops, this output may remain active. An Abort command is capable of internally resetting these flip-flops and releasing the RESET.OUT/ output to allow it to rise to the inactive level as determined by the R2/C2 delay network. When RESET.OUT/ reaches its inactive level, the 7242 FSA and 7250 CPD RESET/ lines are deactivated and 7230 current reference switch Q1 is turned on. The 7242 ENABLE.A/ line, which is controlled by the 7220-1, may now be activated; when active, this line enables the 7230 CS/ and 7250 CS/ (chip select) lines. The system now is fully operational and ready to execute an Initialize command (provided the Abort command had been issued).

Power-down Operation

If either V_{CC} or V_{DD} falls below the 7230 powerfail trip level, the internal PWR.FAIL/ signal in the 7230 is asserted immediately. However, due to the charge on capacitor C1 in the power-up delay network, the PWR.FAIL/ signal is prevented from reaching the active low level until C1 discharges to V_{IL} (maximum 0.8V).

When the PWR.FAIL/ signal level reaches the logic low-level threshold of the 7220-1's PWR.FAIL/ input, an internal power-down sequence is initiated within the 7220-1. As discussed earlier in the 7220-1 PWR.FAIL/ input description, the 7220-1 PWR.FAIL/ input cannot tolerate any positive threshold crossings during the power-down sequence. If a positive transition should occur; a power-up sequence will be initiated taking precedence over the power-down sequence currently in progress, and this unorderly shutdown could result in the loss of data.

The execution time of 7220-1 power-down sequence varies according to whether the coils are active (i.e., rotating magnetic field is on) or inactive. If the rotating field is off, the power down sequence is completed in approximately 10 microseconds. If the rotating field is on and a swap operation has not been initiated, the worst-case power-down time is increased to 26 microseconds; if a swap operation has been initiated, the power-down time sequence requires a maximum of 110 microseconds. The power-down time is shown in Figure 10. Note that the total system power-down time, since the operation is not complete until the RESET.OUT/ signal line is asserted, is the sum of the 7220-1's internal power-down sequence time and the discharge times for capacitors C1 and C2. To ensure proper operation of the bubble system for data integrity during power-down operations, the power supply maximum decay rates must be observed.

Powerfail Reset Circuit — Revision 1

Summary

The powerfail reset circuit (revision 1) was designed to reduce the requirements placed on the revision 0 powerfail reset circuit and to further reduce the risk of data loss during power-up/down operation. Specifically, the improvements realized were:

1. The possibility of data loss was eliminated provided that the circuit was operated within voltage decay rates specifications.
2. Power-down time was shortened to reduce the energy storage requirements.

The power supply requirements (shown in Table 5) were relaxed with this implementation, which reduces the system requirements and the possibility of data loss.

Power-up

The power-up operation of the circuit shown in Figure 11 is unchanged from the power-up operation of the revision 0 circuit. The characteristics associated with the operation of the powerfail reset circuit below approximately 2.0 volts were not resolved with this circuit solution. If the voltage rise time specifications were not observed, the support circuits could have been enabled prematurely and would allow current transients to reach the drive coils or bubble function conductors (resulting in data loss).

Table 5. Power Supply Requirements for Powerfail Reset Circuit (Revision 1)

	V_{CC} (volts/msec)		V_{DD} (volts/msec)	
	Min.	**Max.**	**Min.**	**Max.**
Power-Up Voltage Rate of Rise	0.12	None	None	None
Power-Down/Power Failure Decay Rate	None	0.45	None	1.1

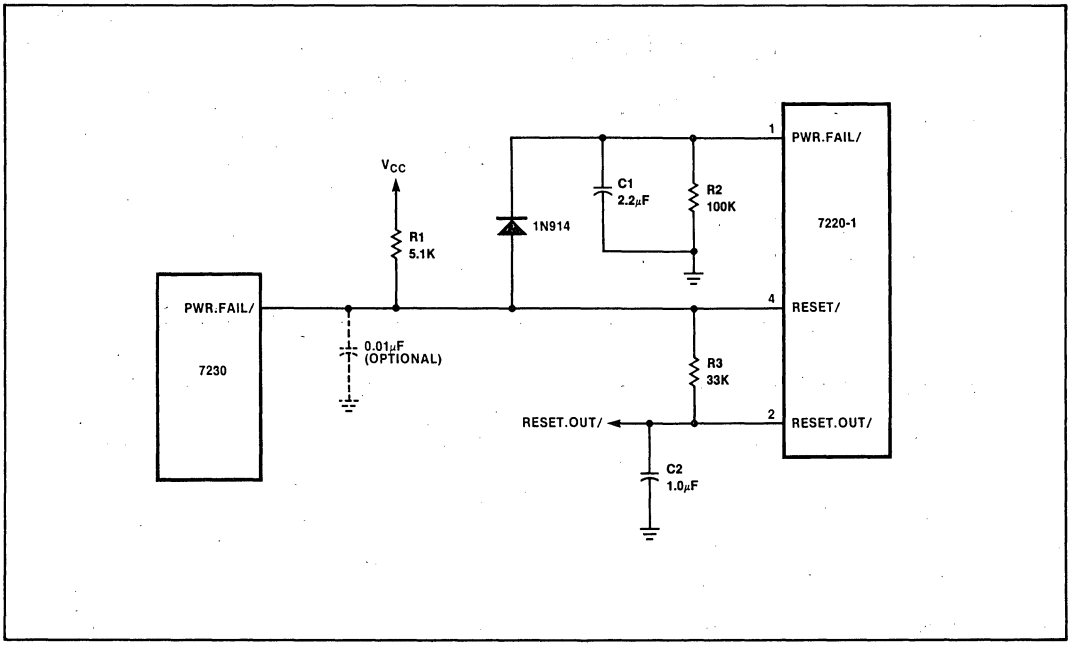


Figure 11. Revision 1 Circuit

Power-down

The simple modifications implemented in the external powerfail circuit (revision 1) greatly reduced the overall power-down operation timing (See Figure 12). This modification made use of the 7220-1 RESET/ input to initiate a power-down sequence instead of the 7220-1 PWR.FAIL/ input by effectively isolating the 7230 PWR.FAIL/ signal from delay capacitor C1 during power-down operations (eliminating an initial capacitor discharge delay). The 7220-1 BMC initiates an internal power-down sequence whenever its RESET/ input goes active, identical to the negative transition of the 7220-1 PWR.FAIL/ input. The difference between these two 7220-1 input signals is that the RESET/ input is latched and does not recognize a low-to-high transition and power-up therefore must be initiated by the positive transition of the 7220-1 PWR.FAIL/ input. With this circuit, the power-up operation timing was unaltered, and the power-down operation timing was reduced from approximately 500 microseconds in the revision 0 powerfail circuit to approximately 200 microseconds in the revision 1 powerfail circuit.

The primary reason for further refining this approach was the increased possibility for a "communication lockout" by the 7220-1. "Communication lockout" resulted when power was temporarily lost from the system. Specifically, the following two conditions were responsible for the "communication lockout":

1) The 7220-1 RESET/ input was activated low due to power loss (minimum pulse width must be 250 nanoseconds to ensure that it is latched) and initiated a power-down sequence.

2) The 7220-1 PWR.FAIL/ discharged but not below the inactive state (0.8 to 2.5 volts, typically 1.5 volts), before power was restored. A power-up sequence could not be initiated to reset the BMC to a known state and communication is "locked out."

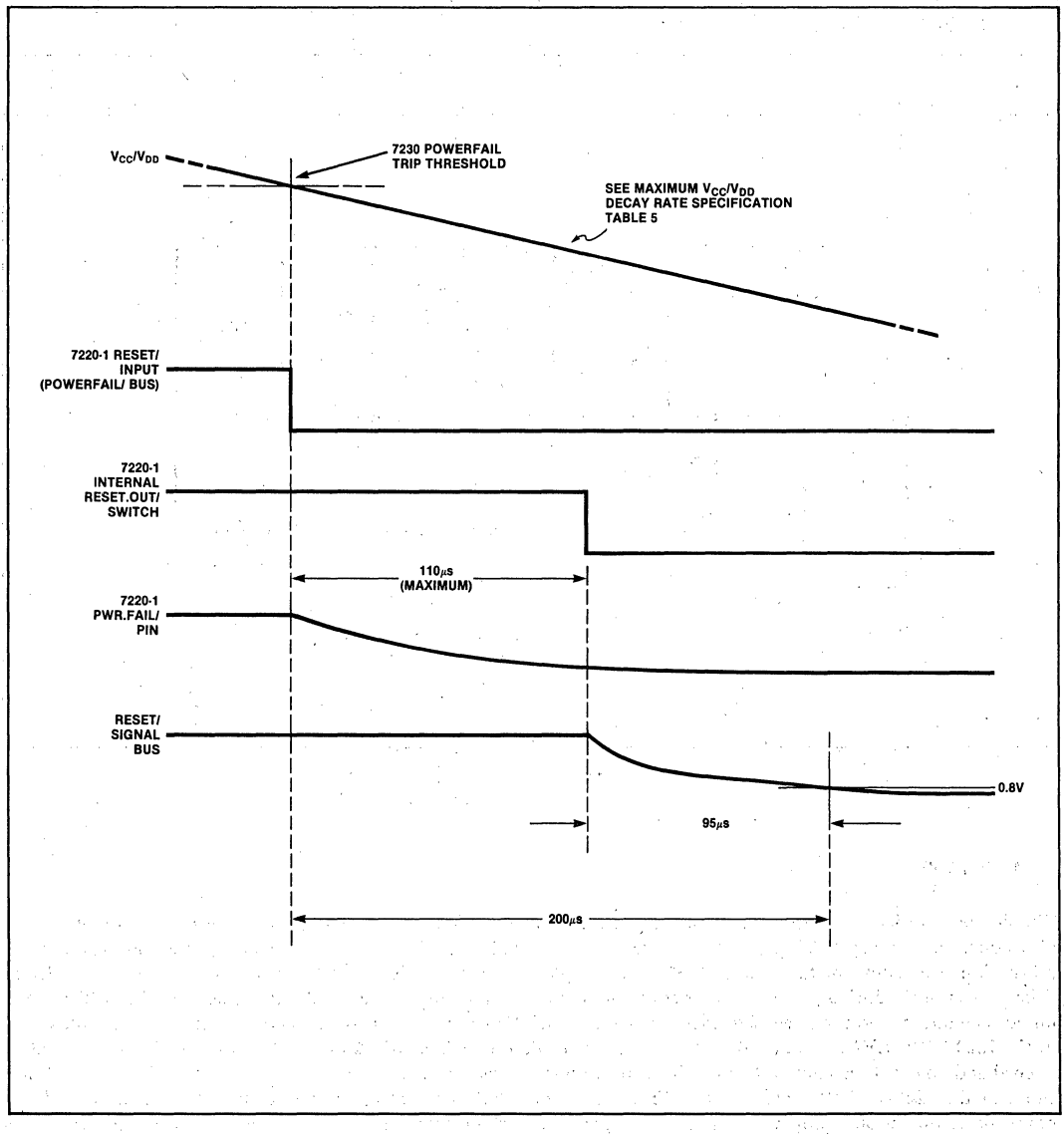


Figure 12. Power-down Timing (Revision 1)

Even if the circuit is operated within the voltage decay rate specifications, this inconvenience is still possible; the only solution is to pulse the 7220-1 PWR.FAIL/ input long enough to discharge C1 to a worst case value of 0.8 volt either by power cycling or external control. This user inconvenience and special system requirement led to the development of the next powerfail reset circuit.

Powerfail Reset Circuit — Revision 2

The powerfail reset circuit (revision 2) was developed to eliminate the possibility of data loss during power-up and power-down operation provided the power supply requirements are observed. The following paragraphs describe the principals of operation of the powerfail reset circuit. As power is applied or removed, several different signal value combinations are possible which complicate the analysis of this circuit. For the sake of simplicity, a general overview of a typical case is included rather than a detailed representation of each case. Throughout this discussion it is helpful for the reader to refer to the schematic diagram (Figure 3) and the timing diagrams (Figure 5 and 6).

Power-up

The overall circuit operation is complicated by the additional component, IC1. The power-up operation of the revision 2 circuit is very similar to previous circuits, however, the possibility of prematurely enabling the support components is eliminated. Diodes D1, D2 and resistor R5 serve to prevent capacitor C2 from charging beyond a level (0.8V) that could potentially deactivate the RESET/ signal bus to the 7242 FSA, the 7250 CPD and the VMOS transistor switch. Resistor R5 is chosen so that as V_{CC} is applied, diodes D1 and D2 will be forward biased and provide sufficient voltage drop to prevent capacitor C2 from charging above 0.8V.

Once the 7220-1 power-up sequence is complete or the first Abort command is received, the 7220-1 RESET.OUT/ is deactivated and capacitor C2 is allowed to fully charge. When the RESET/ signal bus reaches an inactive state the power-up sequence is complete and the system is prepared to accept an Initialize command (provided the Abort command has been issued).

Power-down

The power-down operation of the external powerfail reset circuit (revision 2) is very similar to revision 1. The fundamental difference is the ability to maintain a charge on capacitor C1 throughout the 7220-1 power-down sequence. This eliminates any glitch sensitivity or incorrect circuit operation during momentary power loss. The 7220-1 BMC initiates an internal power-down sequence whenever its RESET/ input goes active. The 7220-1 RESET.OUT/ signal is gated through IC1 and remains inactive during this time period preventing capacitor C1 from discharging. At the completion of the 7220-1 power-down sequence RESET.OUT/ signal is pulled low which causes both of the IC1 OR gate outputs to go low. The current sinking capability of these outputs act to quickly discharge capacitors C1 and C2 and complete the power-down sequence.

October 1983

8085 To BPK 72 Interface

Ulmont Smith Jr.
Applications Engineer

OCTOBER 1983
ORDER NUMBER: 210849-002

8085 TO BPK 72 INTERFACE

INTRODUCTION

Bubble Memory is quickly emerging as the preferred high density storage medium for a variety of microprocessor applications. Considering their size and reliability, Bubble Memory allows the designer to utilize the advantages of microprocessors in environments that were not possible using other high density peripheral storage technologies. Aside from portable or rugged environmental applications, bubbles also open up new design possibilities for desk-top terminal applications. Some of the benefits that can be realized from the implementation of Bubble Storage are increased flexibility, reduced maintenance, and non-volatility.

In addition to a one megabit Bubble Memory, Intel magnetics also manufactures a complete family of integrated-support circuits that simplify the task of designing with Bubble Memory. The family of support circuits provides an easy-to-use microprocessor interface via a single VLSI component, the Bubble Memory Controller (BMC). The remaining support circuits are controlled by the Bubble Memory Controller allowing the designer total freedom from the control signals associated with Bubble Memory technology.

At the component level, the BPK 72 (Bubble Memory Prototype Kit) provides the best opportunity to discover the potential of bubble storage. The BPK 72 comes complete with all the hardware and documentation necessary to prototype a one megabit (128K-bytes) Bubble Memory System. The BPK 72 is completely assembled and tested leaving the designer with the simple task of interfacing to a host processor.

This application note demonstrates how little effort is required to interface a BPK 72 with an 8085 microprocessor. The first four sections, "Introduction, BPK 72 Overview, Constructing the Hardware Interface, Implementing the 8085/BPK 72 Software Driver," and Appendix A (software listing) provide all the information necessary to interface a BPK 72 with an 8085 microprocessor based system. The remaining chapters describe in detail the hardware and software considerations involved with designing and implementing a Bubble Memory Interface.

A set of generalized flowcharts describing the software driver may also be found in Appendix A to facilitate the task of interfacing with other microprocessors.

BPK 72 OVERVIEW

The BPK 72 consists of a completely assembled and tested 10cm x 10cm printed circuit board containing a one megabit Bubble Memory and the complete family of integrated support circuits.

A block diagram of the BPK 72 is presented in Figure 1. It illustrates the key components in a one megabit, 128K-byte Bubble Memory System.

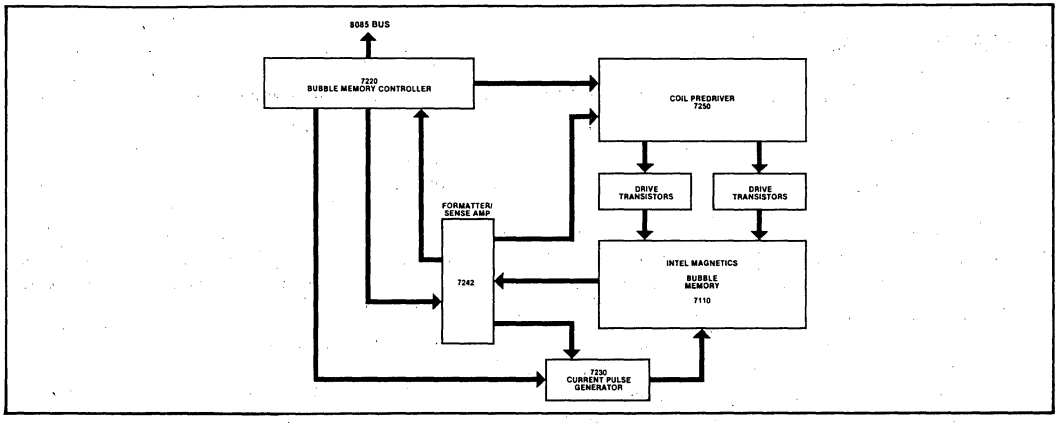


Figure 1. Block Diagram of the BPK 72

The 7110 Bubble Memory Module is supported by the following integrated circuits:

7220-1 Bubble Memory Controller (BMC)
The 7220-1 provides a convenient microprocessor interface and generates the timing signals necessary for the proper operation of the remaining support circuitry.

7242 Formatter Sense Amplifier (FSA)
The 7242 is responsible for detecting and enabling the generation of magnetic bubbles within the 7110. The 7242 also performs data formatting tasks and the option of automatic error detection and correction.

7250 Coil Predriver and 7254 Drive Transistors
The 7250 and two 7254s supply the drive currents for the rotating magnetic field that move the magnetic bubbles within the 7110 Bubble Memory Module.

7230 Current Pulse Generator (CPG)
The 7230 generates a set of waveforms necessary to input and output data from the 7110.

CONSTRUCTING THE HARDWARE INTERFACE

The hardware necessary to interface a BPK 72 with an 8085 microprocessor consists of a few simple connections to the system bus and the addition of only three integrated circuits; 7406—hex inverter (open collector), 7430—eight input nand gate, and an 8284A—Intel clock generator.

A schematic is presented in Figure 2 of the interface logic between a BPK 72 and the demultiplexed bus from an 8085 microprocessor.

The interface uses the eight input nand gate to enable chip-select on the BPK 72 when an I/O instruction is executed at ports 0FEH ("H" designates hexadecimal notation) or 0FFH. The address line A8 from the microprocessor bus is connected to A0 on the BPK 72 to select one of two internal ports. If the ports 0FEH and 0FFH are not available, simply connect A8 to the input of the nand gate and move a higher order address line (A9–A15) to A0 on the BPK 72. In the event that the I/O addresses are changed, the user must enter the new port locations into the software driver (see Appendix A). The I/O port locations are initialized as equates at the beginning of the program. All system dependent variables have been parameterized whenever possible.

The designer has the option of memory mapping the BPK 72 or utilizing 2 of the 256 I/O ports available on the 8085. The I/O ports were chosen for this interface to simplify the address decoding and to provide easy access to existing systems.

POWER SUPPLY REQUIREMENTS

The BPK 72 operates on standard +5V and +12V DC power within a 5% tolerance. The worst case power consumption is a follows:

+5VDC = 2 watts maximum
+12VDC = 5 watts maximum

When power is applied or removed from a Bubble Memory System, the rotating magnetic field within the 7110 Bubble Memory is held in the proper phase to insure non-volatility. This is accomplished through the use of a power fail reset circuit. The following power supply specifications must be observed to effectively support the power fail circuitry:

A. VDD = +12V, ±5% tolerance
   Power off/power fail voltage decay rate—less than 1.1 volts/millisecond
B. VCC = +5V, ±5% tolerance
   Power off/power fail voltage decay rate—less than 0.45 volts/millisecond
C. Voltage sequencing—no restrictions
D. Power on voltage rate of rise—no restrictions

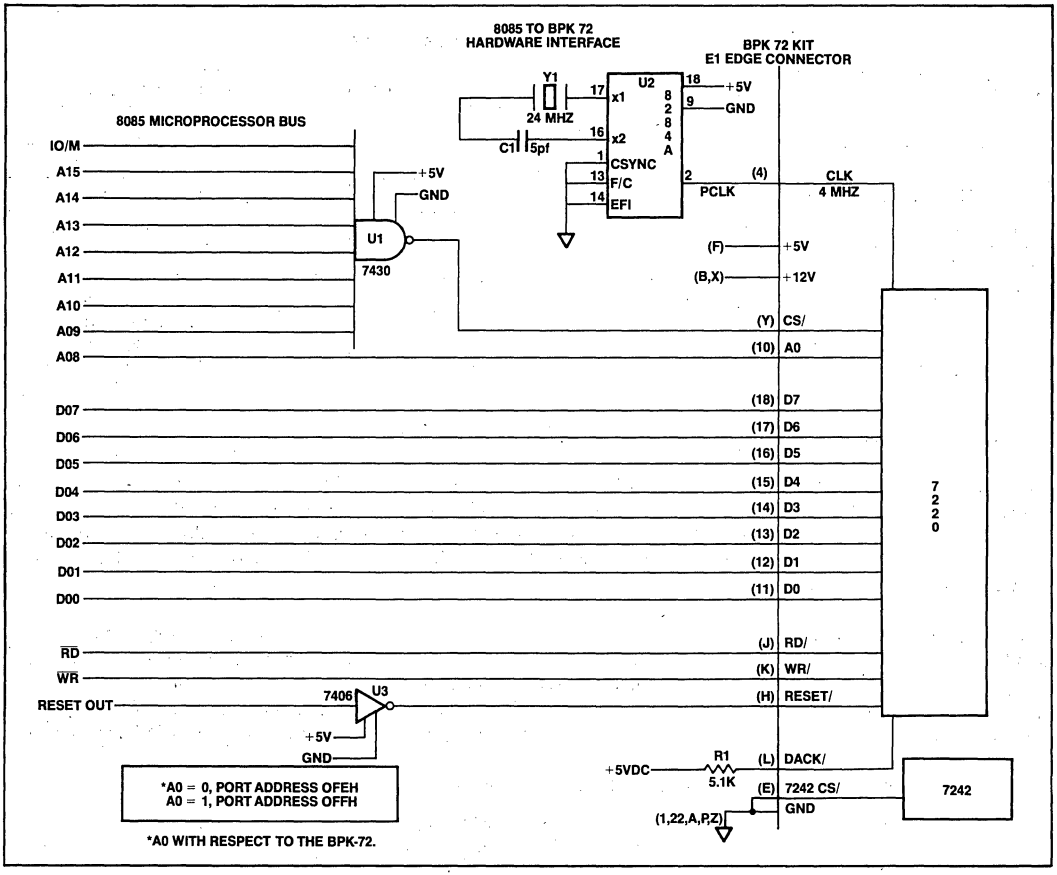


Figure 2. Hardware Interface

The interface designer should verify that the system power supply decay rates meets the specifications previously listed. To simulate worst case conditions, connect a 2 watt load on the +5 volt supply and a 5 watt load on the +12 volt supply. The power supply decay rates can be easily measured during the removal of power with a standard oscilloscope. No attempt should be made to use the BPK 72 until the power supply decay rates have been verified.

Table 1. 8085/BPK 72 Interface Parts List

Item	Description	Quantity	Reference	Manufacturer
1	IC-7430—8 input nand gate	1	U1	any
2	IC-8284A—clock generator	1	U2	Intel
3	IC-7406—hex inverter open collector	1	U3	any
4	Crystal—24.0000MHz fundamental mode, series resonant	1	Y1	any
5	Resistor—5.1Kohm, 1/4W, 5%	1	R1	any
6	Mica Capacitor—5pf, 100VDC, 5%	1	C1	any
7	Edge connector, 44 pin	1	E1	TRW, CINCH #50-44B-10

IMPLEMENTING THE 8085/BPK 72 SOFTWARE DRIVER

An 8085 to BPK 72 software driver program listing is presented in Appendix A. The driver consists of a set of subroutines that can be called to perform commonly used Bubble Memory commands. A detailed description and flowchart of each subroutine is provided with the program listing. The software driver is relocatable and may be linked with other programs. The name of the program is "BPK72." It begins at 0800H and requires less than 1K bytes of memory allocation.

The software driver is written in 8085 assembly language. It can be easily incorporated into existing systems as part of a utility program to transfer data between the BPK 72 and the 8085's addressable memory. The subroutines have been designed to eliminate the need for any further software development concerning the operation of the BPK 72. Assembly was chosen over higher level languages to provide the most efficient and portable code. With only minor modifications to the parameterized variables, the program, "BPK72," will run on almost any 8085 based system.

The following subroutines in the program "BPK72" will now be discussed:

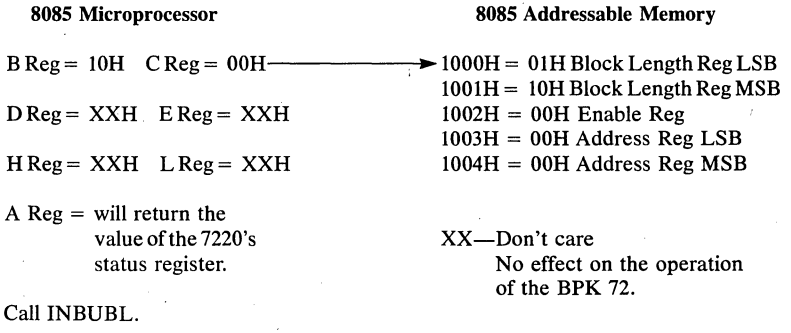
INBUBL—Initialize Bubble Memory
WRBUBL—Write Bubble Memory data
RDBUBL—Read Bubble Memory data
ABORT—Abort present command, reset BPK 72

INITIALIZING THE BUBBLE

After powering up, the BPK 72 must be initialized before any data transfers can begin. Initialization is needed to synchronize the 7220 Bubble Memory Controller with the data in the 7110 Bubble Memory storage loops and also because the 7110 employs redundancy. The 7110 Bubble Memory contains 320 storage loops. However, only 272 of the 320 loops are necessary for a 100% functional one megabit part. The additional 48 loops provide a 15% redundancy. Redundancy is used to significantly increase the yield of Bubble Memory modules during manufacture.

A map of the active and inactive loops is placed on a label attached to the case of the 7110. The same map is also placed in the 7110 during final test. When the system is initialized, the 7220 reads the map (boot loop) from the 7110 and decodes it. The boot loop is transferred from the 7220 into a pair of boot loop registers in the 7242 formatter sense amplifier. The boot loop registers are used to format data to insure that only functional loops are enabled during read or write operations.

Only one call to the initialization subroutine, INBUBL, is necessary to initialize a BPK 72. The following is an example of how to call INBUBL:



The example shown above demonstrates how to set up the B–C registers prior to calling the initialization subroutine, INBUBL. The B–C register pair must contain the address of the first of five consecutive locations within the 8085's addressable memory. In this example, the B–C registers are pointing to the first of five memory locations starting at 1000H. The data contained in 1000H through 1004H is a memory image of the parametric registers within the Bubble Memory Controller. The parametric registers contain a set of flags and parameters that determine exactly how the 7220 will respond to a software command.

Note the values used for the block length and address registers. These values must always be used during the initialization process with a one megabit Bubble Memory System. The enable register is shown with a 00H indicating the absence of error detection and correction. The 7220 and 7242 provide an optional error detection and correction feature to enhance data integrity. It is recommended that first time users begin without the use of error correction. Later on if error correction is desired, a 20H should be placed in the memory location designated as the enable register. A discussion concerning the use of error correction may be found in the section titled, "Communicating with the 7220."

Figure 3 illustrates the sequence of program flow necessary to initialize a Bubble Memory System using the subroutine INBUBL. Note that Figure 3 includes a test of the Bubble Memory Controller's status register. The status register is separate from the parametric registers and contains information about error conditions, completion or termination of commands, and the 7220's readiness to transfer data. To simplify the task of verifying a successful initialization, INBUBL returns the value of the 7220's status register to the calling routine through the 8085's "A" register. A successful initialization will return a 40H status. All other values indicate a BPK 72 system failure. Consult Appendix B in the unlikely event that the subroutine INBUBL fails to return a successful status.

READING AND WRITING

Only one call to the subroutine RDBUBL or WRBUBL is necessary to transfer data between the BPK 72 and the 8085's addressable memory.

Like many high density peripheral storage devices, Bubble Memory data is organized into pages rather than bytes. The 7220 Bubble Memory Controller partitions the one megabit Bubble Memory into 2048 pages of either 64 or 68 bytes in length. The page length is dependent upon the use of automatic error detection and correction—64 bytes with error correction and 68 bytes without. Data transfers are specified in terms of whole pages. Therefore the minimum amount of data that can be transferred from one read or write command is 64 or 68 bytes.

The parametric registers are used to communicate to the controller which page or pages will be transferred during a read or write command. The address register LSB and the first three bits of the address register MSB define the starting page address for read or write commands. The block length register determines how many pages will be transferred starting at the location defined by the address register. Theoretically, data transfers can range from 1 to 2048 pages in length. However, this application limits the maximum data transfer between the BPK 72 and the 8085's

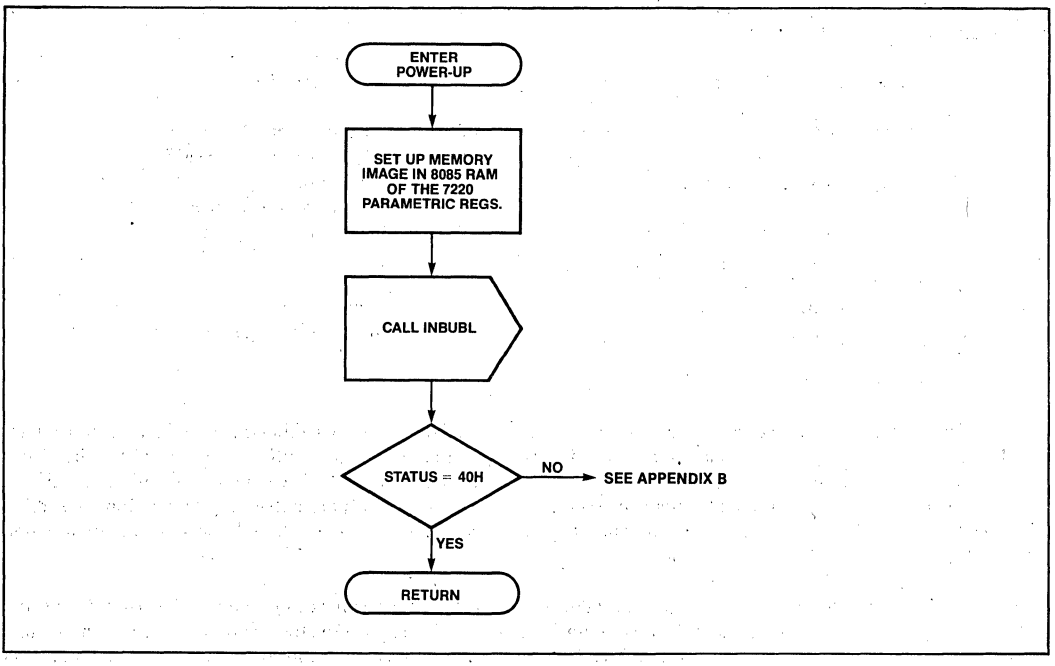


Figure 3. Initializing the BPK 72

memory to no more than 255 contiguous pages. This limitation results from the need to prevent data transfers that could exceed the addressable memory space of the 8085. The block length register LSB may be assigned any value between 1 and 255 depending on the size of the transfer. A detailed description of the parametric registers may be found in the section titled, "Communicating with the 7220."

The following is an example of how to use the Read Bubble Memory subroutine, RDBUBL, to transfer the first 16 pages (00H–0FH) of data from the BPK 72 to the 8085's addressable memory, starting at location 2000H:

8085 Microprocessor	8085 Addressable Memory
B Reg = 10H C Reg = 00H →	1000H = 10H Block Length Reg LSB
	1001H = 10H Block Length Reg MSB
D Reg = 20H E Reg = 00H →	1002H = 00H Enable Reg*
	1003H = 00H Address Reg LSB
H Reg = XXH L Reg = XXH	1004H = 00H Address Reg MSB
A Reg = will return the value of the 7220's status register.	2000H = start data transfer 243FH = last data transfer (1088 byte transfer)
	XX—Don't care. No effect on the operation of the BPK 72.
Call RDBUBL.	*—Assumes that the BPK 72 was initialized without error correction.

The Write Bubble Memory subroutine, WRBUBL, can be substituted for the call to RDBUBL to transfer data from the 8085's addressable memory to the first 16 pages in the BPK 72.

The example shown above demonstrates how to set up the B–C and D–E registers prior to calling a read or write subroutine. Just as in the case of initialization, the B–C registers contain the address of the first of five consecutive memory locations within the 8085's addressable memory. The data contained in the memory addressed by the B–C registers is used to load the 7220's parametric registers. The D–E register pair contains the address of the first byte of data to be transferred to or from the 8085's addressable memory.

Figure 4 illustrates how the read and write subroutines, RDBUBL and WRBUBL, should be called from another routine. The flowchart includes a program path to handle errors in the unlikely event that the read or write subroutines fail to return a successful status. First time users can omit the additional program flow for preliminary evaluation. The next section, "Checking the Status," describes the appropriate status values necessary to verify a successful data transfer.

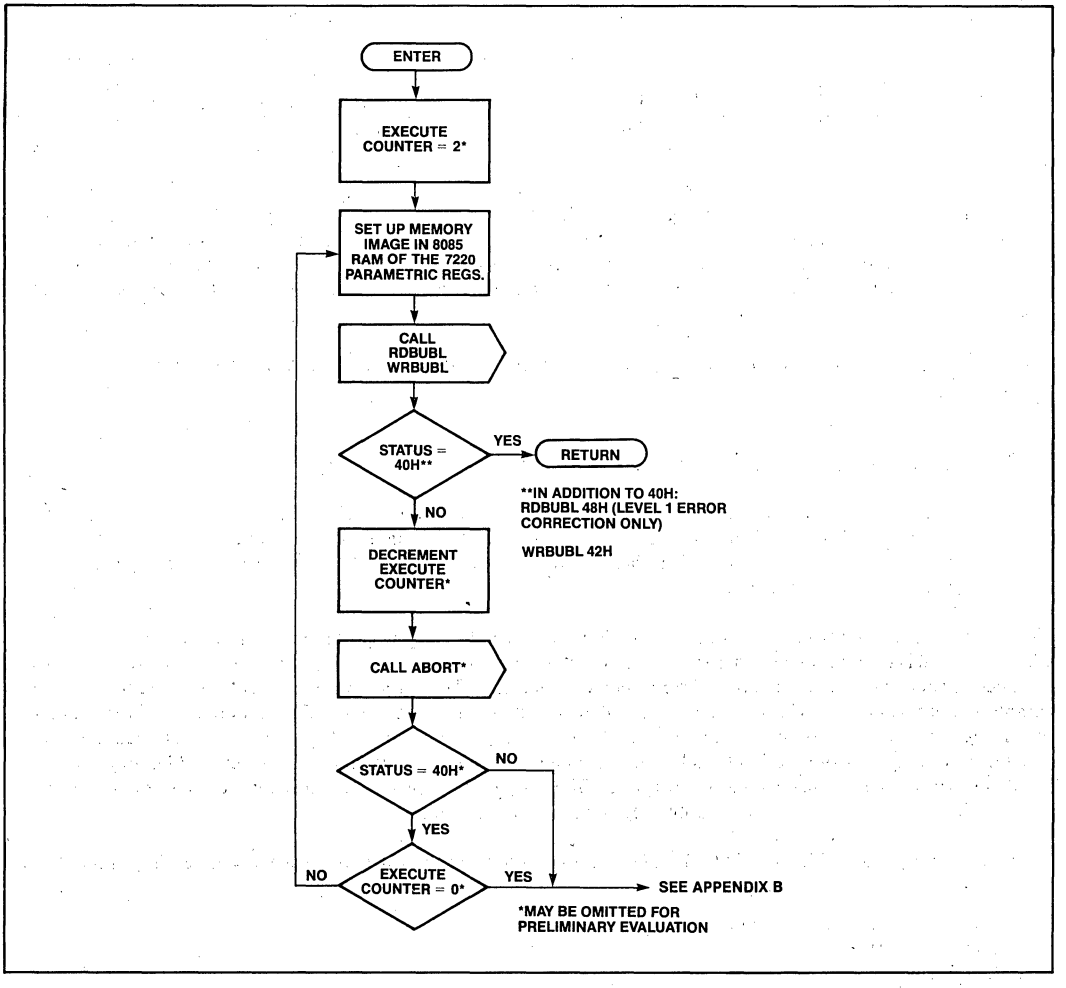


Figure 4. Reading and Writing to the BPK 72

CHECKING THE STATUS

After calling a subroutine to initialize, read, or write Bubble Memory data, the 7220's status register should be read to verify that the command was successfully executed. Note that flowcharts 1 and 2 include a test of the status register to detect for any errors. In order to facilitate the task of verification, each of the commonly used subroutines in the program "BPK72" return the contents of the 7220's status register to the calling routine through the 8085's "A" register. It is the responsibility of the calling routine to verify the success of each subroutine. A list of acceptable status register values for each of the subroutines in the program "BPK72" is presented in Table 2.

Table 2. Acceptable Status Register Values

Subroutine	Acceptable Status Register Value(s)	Comments
INBUBL	40H	OP-complete
WRBUBL	40H	OP-complete
	42H	OP-complete, parity error
RDBUBL	40H	OP-complete
	48H	OP-complete, correctable error*
ABORT	40H	OP-complete

*Level 1 error correction only

If any read errors are encountered during the transfer of data, they will almost always result from external noise interfering with the signal path between the 7110 Bubble Memory and the 7242 formatter sense amplifier. Since the data within the Bubble Memory is usually correct, a second attempt to transfer data should be successful. Figure 4 illustrates the use of the ABORT command to reset the Bubble Memory Controller before making another attempt to read or write Bubble Memory data.

Service information is presented in Appendix B in the unlikely event that any of the subroutines in Table 2 do not function properly.

7220 MICROPROCESSOR INTERFACE OVERVIEW

The key to any interface incorporating a BPK 72 is the Bubble Memory Controller. The controller provides a complete interface to a TTL level microprocessor bus that allows the designer total freedom from the intricate timing and waveforms necessary to support a Bubble Memory System. A block diagram of the 7220 Bubble Memory Controller is presented in Figure 5.

The 7220 interface circuitry consists of one 8-bit bidirectional port. The port provides access to internal registers. The address line A0 is used to select either the command/status or parametric/data registers. A command register is used to issue instructions such as read or write Bubble Memory data. The status register provides information about the completion or termination of commands and the 7220's readiness to transfer data. The parametric registers contain a set of flags and parameters that determine exactly how the 7220 will respond to a software command. The data register is actually a forty byte FIFO to buffer the timing differences between the 7110 Bubble Memory and a host processor. In order to transfer data to (from) the BPK 72, the host processor must load the parametric registers followed by issuing a read or write Bubble Memory data command.

To maintain design flexibility, the 7220 Bubble Memory Controller provides the user with three different modes of data transfer:

1. DMA, direct memory access
2. Interrupt-driven
3. Polled I/O

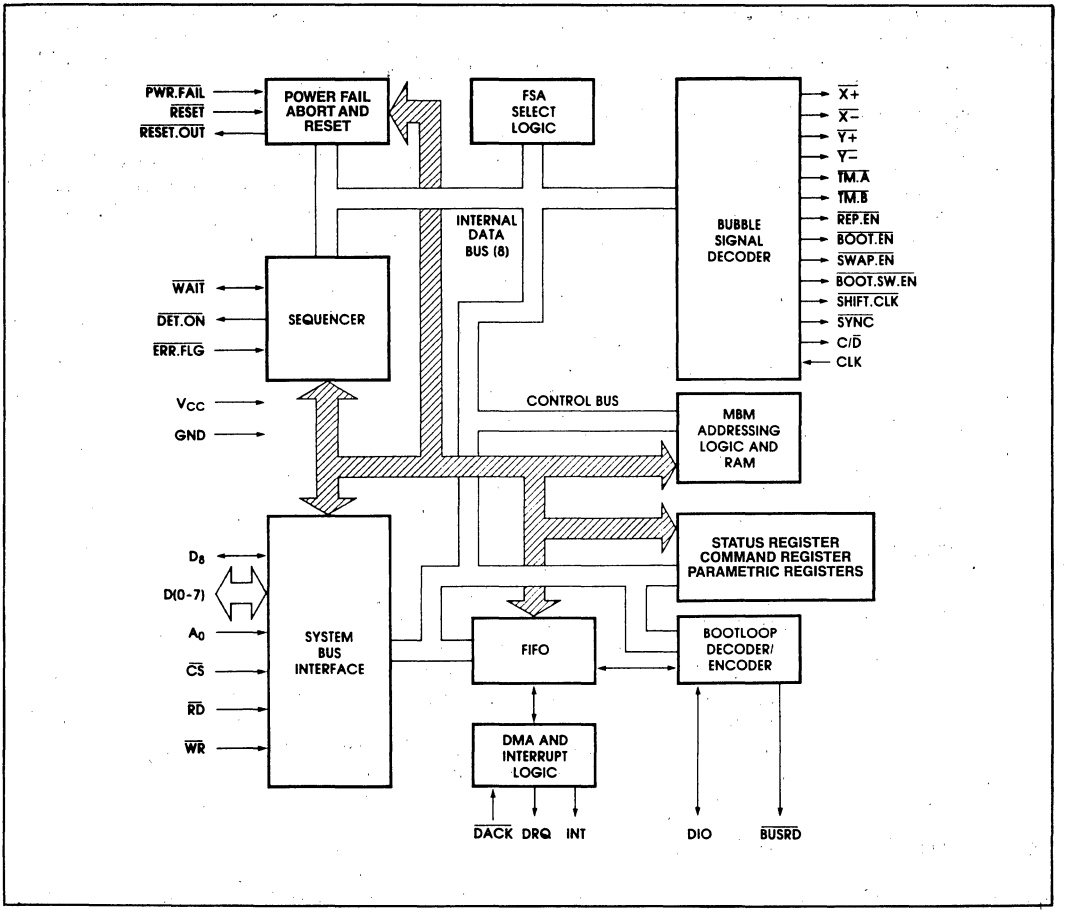


Figure 5. Block Diagram of the 7220 Bubble Memory Controller

In the DMA data transfer mode, the 7220 operates in conjunction with a DMA controller (such as Intel's 8257) using the DRQ (data request) and DACK (data acknowledge) lines for handshaking. With the help of a DMA controller, the 7220 transfers the data to (from) the host processor's memory. Once the data transfer begins, program intervention is not required until the entire data transfer has been completed.

In the interrupt mode, the 7220 along with an interrupt controller (such as Intel's 8259) uses the DRQ (data request) line to initiate a data transfer. The DRQ line becomes active when the 7220 is ready to send or receive a burst of data. A typical data burst is 22 contiguous bytes for an interrupt-driven interface. A set of software drivers are also necessary to service the interrupts to coordinate the transfer of data between the 7220 and the memory associated with a host processor. One advantage to the interrupt mode is multitasking. Since the host processor is only servicing the 7220 during data transfers, dead time between data transfers can be utilized for other processor tasks.

A polled mode interface reads the 7220 status register to determine when to transfer one byte of data. Of all the interface modes, polled I/O is the simplest configuration to implement. No special hardware or external controllers are necessary to interface the 7220 with a microprocessor. The major portion of a polled mode design is the software. Just as in the interrupt mode, a set of software drivers are required to read and write data to the 7220.

This application uses a polled mode configuration. The polled I/O data transfer mode was selected over DMA and interrupt-driven to simplify the interface design. A polled mode interface does not require the use of a DMA or interrupt controller. Furthermore, the polled mode interface provides the most flexibility for incorporating a BPK 72 into existing 8085 systems. Since the majority of a polled mode design consists of software, simple program modifications to accommodate existing systems can be easily entered into the software driver provided in Appendix A.

In terms of performance, the polled I/O transfer mode is the lowest compared to DMA or interrupt-driven. The DMA and interrupt modes offer the advantage of multitasking. However, the average access time and data transfer rate remain the same for each data transfer mode. The following formulas and examples demonstrate how to calculate the transfer time for a one megabit Bubble Memory System:

READ N-page transfer:
Transfer time = seek time + 8.7 ms + 7.5 ms (N−1)

WRITE N-page transfer:
Transfer time = seek time +7.5 ms (N)

Average seek time = 41 ms
Worst case seek time = 82 ms
Average data rate = 8.5 K-bytes/sec

For Example:

A. Time to read 1 page (assuming avg seek time):
Transfer time = 41 ms + 8.7 ms = 49.7 ms
B. Time to write 1 page (assuming avg seek time):
Transfer time = 41 ms + 7.5 ms = 48.5 ms
C. Time to read 10 contiguous pages (assuming avg seek time):
Transfer time = 41 ms + 8.7 ms + 7.5 ms (10–1) = 117.2 ms
D. Time to write 10 contiguous pages (assuming avg seek time):
Transfer time = 41 ms + 7.5 ms (10) = 116.0 ms

HARDWARE INTERFACE DESCRIPTION

To simplify the task of interfacing a BPK 72 with a microprocessor, the 7220 Bubble Memory Controller provides a convenient set of TTL signals that may be directly connected to a system bus. The interface signals on the BPK 72 necessary to implement a polled mode configuration are presented in Table 3.

PARITY BETWEEN THE 8085 AND BPK 72

The 7220 has the capability of generating and detecting odd parity using the bidirectional data line D8. The parity bit may be used to increase the reliability of the data path between the 7220 and a host processor. During data transfers, odd parity is generated for read operations and tested for write operations. The host processor may read the 7220 status register to determine if a parity error occurred during a write operation. Parity is typically implemented when a long transmission path exists between the host processor and the 7220. Since most systems utilize a simple edge connector backplane and a short transmission path (less than 18 inches), parity is not necessary. Parity is not implemented in this application to minimize the hardware complexity.

The parity bit, D8, is not stored within the 7110 Bubble Memory module. A separate and more effective error detection and correction feature is available as an option to increase the data integrity within the 7110. See the section titled, "Communicating with the 7220" for further details about the option of automatic error detection and correction.

Table 3. BPK 72 Polled Mode Interface Signals

Signal	Function
A0	Address line A0 = 0 Selects the FIFO data buffer or the parametric registers. A0 = 1 Selects command/status registers.
D0–D7	8 bit bidirectional data bus.
D8	Optional odd parity bit, not used in this application.
CS/	Chip select input. A logic high will tri-state the 7220 interface signals. (Slash, "/" designates a low active signal, system ground)
RD/	Read 7220 registers or data FIFO.
WR/	Write 7220 registers or data FIFO.
DACK/	DMA acknowledge. If DMA is not used, DACK/ requires an external pullup resistor to VCC (5.1 Kohm).
CLK	4 MHz TTL level clock. Clock period = 250 ns, 0.25 ns tolerance. Duty cycle = 50%, 5% tolerance.
RESET/	A low on this pin forces the interruption of any 7220 activity, performs a controlled shut-down, and initiates a reset sequence. The next instruction following RESET/ must be an abort command.
7242 CS/	7242 chip select signal is used to select banks of 7242s. 7242 CS/ must be tied low (system ground) for a single bank configuration.

4 MHZ CLOCK

The BPK 72 requires an external 4 MHz (may be asynchronous with respect to a host processor) TTL level clock. The specifications for the period and duty cycle are presented in Table 3. The 7220 uses the external clock to generate the timing signals that control the rotating magnetic field within the 7110 Bubble Memory. For reliable operation, the clock tolerances must be observed to assure that the rotating field is stable and accurate.

An Intel integrated circuit, 8284A clock driver, is used to generate the 4 MHz external clock. The 8284A along with a 24MHz series resonant crystal (fundamental mode) will provide a precise and accurate clock for any interface incorporating a BPK 72. The circuit configuration for the 8284A is illustrated in Figure 2. Other techniques of clock generation are acceptable as long as the duty cycle and period are within the specifications listed in Table 3.

SOFTWARE INTERFACE DESCRIPTION

The software driver presented in Appendix A contains the following subroutines that may be called from another routine:

* INBUBL —Initialize the BPK 72.
* RDBUBL —Read Bubble Memory data.
* WRBUBL —Write Bubble Memory data.
ABORT —Abort present command, reset BPK 72.
FIFORS —Reset 7220 FIFO data buffer.
WRFIFO —Write 7220 FIFO data buffer.
RDFIFO —Read 7220 FIFO data buffer.
WRBLRS —Write 7242 boot loop registers.
RDBLRS —Read 7242 boot loop registers.
MBMPRG —Bubble Memory purge command.
** RDBOOT —Read Bubble Memory boot loop.
** BOOTUP —Write Bubble Memory boot loop.

* Most commonly used commands.
** Diagnostic routines (see Appendix B).

Each of the subroutines listed above is described in further detail in Appendix A. Along with each subroutine is a generalized flowchart displaying the program flow. The user is encouraged to read the software driver to better understand the software interaction necessary to interface a BPK 72 with an 8085 microprocessor.

COMMUNICATING WITH THE 7220

Some additional background is necessary to understand the operation of the 7220 Bubble Memory Controller. Figure 6 illustrates the user-accessible registers that control and format the flow of data between the 7110 Bubble Memory and a host processor.

The address assignments for the user-accessible registers within the 7220 are presented in Table 4. The registers are listed in two groups. The first group (status, command, register address counter) consists of those registers that are selected and accessed in one operation. The second group contains the FIFO data buffer and the parametric registers (utility, block length, enable, address), they are selected according to the contents of the register address counter (RAC).

Table 4. Address Assignments for the User-Accessible Registers

A0	D7	D6	D5	D4	D3	D2	D1	D0	Symbol	Name of Register	Read/Write
1	0	0	0	1	C	C	C	C	CMDR	Command Register	Write Only
1	0	0	0	0	B	B	B	B	RAC	Register Address Counter	Write Only
1	S	S	S	S	S	S	S	S	STR	Status Register	Read Only

NOTES:
SSSSSSSS = 8-bit status information returned to the user from the STR
CCCC = 4-bit command code sent to the CMDR by the user.
BBBB = 4-bit register address sent to the RAC by the user.
B3B2B1B0 = 4-bit contents of RAC at the time the user makes a read or write request with A0 = 0.
LSB = Least Significant Byte
MSB = Most Significant Byte

Table 5. Parametric Registers and FIFO Data Buffer

	RAC						
A0	B3	B2	B1	B0	Symbol	Name of Register	Read/Write
0	1	0	1	0	UR	Utility Register	Read or Write
0	1	0	1	1	BLR LSB	Block Length Register LSB	Write Only
0	1	1	0	0	BLR MSB	Block Length Register MSB	Write Only
0	1	1	0	1	ER	Enable Register	Write Only
0	1	1	1	0	AR LSB	Address Register LSB	Read or Write
0	1	1	1	1	AR MSB	Address Register MSB	Read or Write
0	0	0	0	0	FIFO	FIFO Data Buffer	Read or Write

To successfully implement the hardware and software presented in this application, certain restrictions are placed on the contents of the user-accessible registers. Each of the user-accessible registers and any necessary restrictions will now be discussed in further detail.

COMMAND REGISTER

The 7220 command set consists of 16 commands identified by a 4 bit command code. A list of the commands is presented in Table 6.

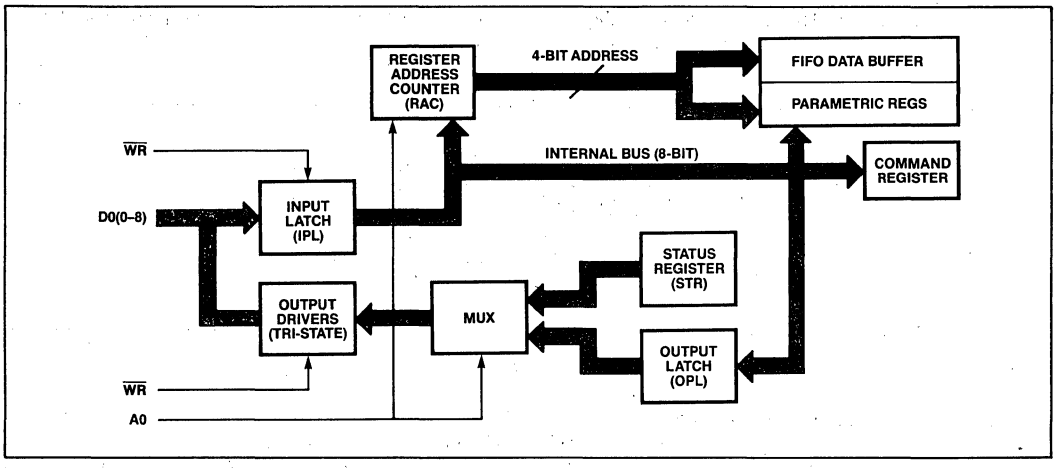


Figure 6. 7220 User Accessible Registers

Table 6. 7220 Commands

D3	D2	D2	D1	Command Name
0	0	0	0	Write Bootloop Register Masked
0	0	0	1	Initialize
0	0	1	0	Read Bubble Data
0	0	1	1	Write Bubble Data
0	1	0	0	Read Seek
0	1	0	1	Read Bootloop Register
0	1	1	0	Write Bootloop Register
0	1	1	1	Write Bootloop
1	0	0	0	Read FSA Status
1	0	0	1	Abort
1	0	1	0	Write Seek
1	0	1	1	Read Bootloop
1	1	0	0	Read Corrected Data
1	1	0	1	Reset FIFO
1	1	1	0	MBM Purge
1	1	1	1	Software Reset

The commands listed in Table 6 are provided for reference purposes only. The software driver in Appendix A consists of a series of subroutines that automatically issue the appropriate commands to perform a data transfer.

The function of each command is usually apparent from the command name (e.g., initialize, read bubble data, write bubble data). Additional detail concerning the function of each command may be found in the BPK 72 user's manual.

REGISTER ADDRESS COUNTER

The register address counter consists of a 4 bit address that points to one of the six parametric registers:

Utility register (UT)—The utility register is a general purpose register available to the user in connection with Bubble Memory System operations. It has no direct effect on the operation of the 7220. It is provided as a convenience to the user.

Block length register (BLR)—The contents of the block length register determine the system page size and the number of pages to be transferred in response to a single bubble read or write command. The bit configuration is as follows:

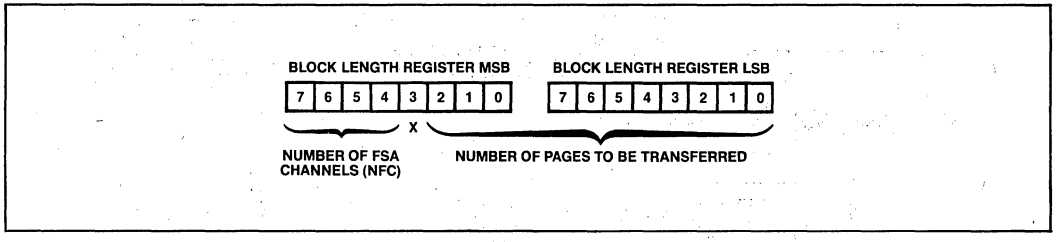


Figure 7. Block Length Registers

The 7220 has the capability of supporting up to eight 7110 Bubble Memory modules. Each 7110 contains two channels that are sensed by a 7242 formatter sense amplifier (FSA). In multiple Bubble Memory configurations, the BLR allows the user to select the page size. Since the BPK 72 consists of only one Bubble Memory module, the field specifying the number of FSA channels in the BLR MSB must contain −0001B ("B" designates a binary notation). After the FSA field is set, the page size is dependent upon the use of error detection and correction. Error correction will be discussed in the next section describing the function of the enable register.

The BLR LSB and the first 3 bits of the BLR MSB determine the number of pages to be transferred during a single read or write command. This application restricts the user to no more than 255 contiguous pages to prevent data transfers that could exceed the addressable memory space of the 8085.

For This Application

BLR MSB—10H at all times.
("H" designates a hexadecimal notation)

BLR LSB—Selectable from 01H to FFH (1 to 255 pages).

CAUTION: 00H in the BLR LSB will enable a 2048 page transfer resulting in a timing error.

Enable Register (ER)—The user sets the bits in the enable register to enable or disable various functions within the 7220. The individual bit descriptions are as follows:

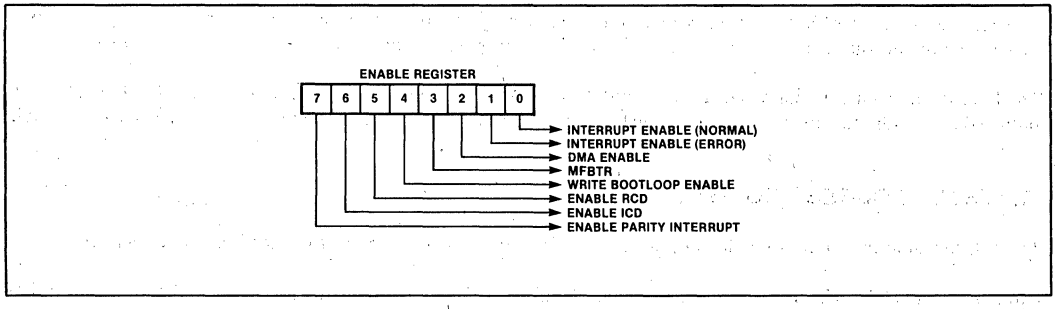


Figure 8. Enable Register

One of the most important functions concerning the enable register is the option of automatic error detection and correction. If error correction is enabled during a write operation, the 7242 formatter sense amplifier appends each 256 bit block of data with a 14 bit fire code. Both the data and the fire code are stored within the 7110 Bubble Memory module. During a read operation, the 7242 compares the data with the fire code to check for any errors. With respect to the FSA, errors are either correctable (the FSA is able to reconstruct the data using an error correction algorithm before transferring the data to the 7220) or uncorrectable. Additional information about the fire code is available in the BPK 72 user's manual.

The enable register offers three levels of error correction. All three levels utilize the same error correction algorithm but differ in their interaction with a host processor. Table 6 defines the relevant register bits for the various levels of error correction.

Table 6. Error Correction Levels

Error Correction Level	Bit 6 (ICD)	Enable Register Bit 5 (RCD)	Bit 1 (Int Enable)
Level 0	0	0	0
Level 1	0	1	0
Level 2	1	0	0
Level 3	1	0	1

Level 0 does not enable the error detection and correction algorithm. In this mode, the 7220 partitions one megabit systems into 2048 pages consisting of 68 bytes per page.

Level 1 is the most popular level of error correction. If an error is detected during a read operation, the 7242 automatically cycles the data through its error correction algorithm and transfers the data to the 7220. If the error was correctable, the 7220 will continue to function normally i.e., correctable errors in Level 1 are transparent to the host processor. If the error was uncorrectable, the 7220 will stop reading at the end of the page wherein the error was encountered. In the unlikely event that the 7220 stops because of an uncorrectable error, the host processor should try at least one more attempt to read the data. In most cases, errors result from random noise that can interfere with the signal path between the 7110 and 7242. Since the data is usually correct within the 7110, another attempt to read the data should yield a successful status.

Level 2 and Level 3 differ from Level 1 in that page-specific logging of uncorrectable errors is possible and the transfer of erroneous data can be prevented. Level 3 differs from Level 2 in that Level 3 also allows the logging of correctable errors.

Neither Level 2 nor Level 3 is supported by this application because the probability of an uncorrectable error is typically one in 10^{16} bits read. An error rate of this magnitude will produce few if any uncorrectable errors throughout the useful life of a Bubble Memory System.

It is recommended that Level 1 error correction be utilized to improve the integrity of the data within the 7110. In Level 1, the 7220 assigns 64 bytes to a page in one megabit Bubble Memory Systems.

Aside from error correction, the enable register performs many other functions.

Enable Parity Interrupt—If this bit is set, any parity errors between the host and the 7220 during write operations will generate an interrupt. Since parity and the interrupt mode are not used in this application, the enable parity interrupt bit should be reset to a logical zero.

Write Bootloop Enable—This bit must be reset to prevent accidental erasure of the boot loop within the 7110.

MFBTR—The MFBTR bit should always be reset to maximize the data transfer rate between the 7220 and 7242 during read operations.

DMA Enable—If this bit is set, the 7220 will attempt to transfer data in the DMA mode. Since this application utilizes a polled mode interface, this bit must be reset to a logical zero.

Interrupt Enable (Normal)—If this bit is set, an interrupt is sent to the host processor after the successful completion of a Bubble Memory command. Since this application uses a polled mode interface, this bit should be reset to a logical zero.

For This Application

Enable Reg—00H. No error correction.
—20H. Level 1 error correction.

Address Register (AR)—The contents of the address register determine which starting address locations will be used during a read or write command. For systems with a multiple Bubble Memory configuration, an additional magnetic Bubble Memory (MBM) select field is used to specify which Bubble Memory(s) will be selected. The bit configuration is as follows:

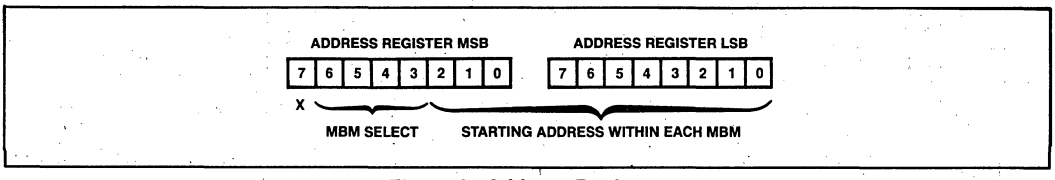


Figure 9. Address Registers

Since the BPK 72 consists of only one 7110 Bubble Memory module, the MBM select field must contain −0000B ("B" designates a binary notation).

For This Application

AR MSB—00000XXX

AR LSB—XXXXXXXX, X = user selectable page address
from 0 to 2047.

STATUS REGISTER

In a polled data transfer mode, the status register provides information about error conditions, completion or termination of commands, and the 7220's readiness to transfer data or accept new commands. The bit configuration for the status register is as follows:

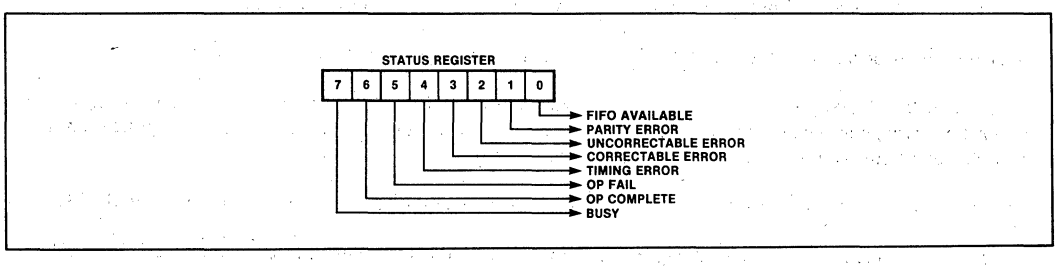


Figure 10. Status Register

Busy—When active (Logic 1), the Busy bit indicates that the 7220 is in the process of executing a command. Bits 1 through 6 of the status register are valid only when the busy bit is not active (Logic 0).

OP Complete—When active (Logic 1), the OP Complete bit indicates the successful completion of a command.

OP Fail—When active (Logic 1), the OP Fail bit indicates that the 7220 was unable to successfully complete the current command.

Timing Error—When active (Logic 1), the Timing Error bit indicates that an FSA has reported a timing error to the 7220, or that the host system has failed to keep up with the required data rate during a read or write operation.

Correctable Error—When active (Logic 1), the Correctable Error bit indicates that an FSA has detected a correctable error in the last block of data read from the 7110.

Uncorrectable Error—When active (Logic 1), the Uncorrectable Error bit indicates that an FSA has detected an uncorrectable error in the last block of data read from the 7110.

Parity Error—When active (Logic 1), the Parity Error bit indicates that a parity error was detected between the 7220 and the host processor. Parity errors are only detected by the 7220 during write operations. Since parity is not used in this application, ignore all parity errors.

FIFO Ready—When the 7220 is busy, an active FIFO Ready bit (Logic 1) indicates that the FIFO has data for reading or space for writing. When the 7220 is not busy, the FIFO Ready bit (Logic 0) indicates that the 40 byte FIFO and the input and output latches are completely empty.

SUMMARY

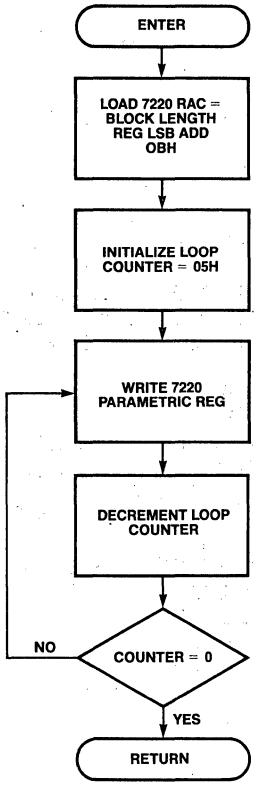
This application note is intended to eliminate almost all of the development effort necessary to interface an 8085 microprocessor with a BPK 72. With the addition of only a few IC's and the software driver presented in Appendix A, the designer is well on the way to incorporating the benefits of improved reliability, reduced maintenance, and non-volatility into any 8085 microprocessor based system.

APPENDIX A
8085 TO BPK-72 INTERFACE SOFTWARE DRIVER LISTING AND FLOWCHARTS

```
ASM80 :F1:BPKHDR


ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    1


 LOC  OBJ        LINE       SOURCE STATEMENT

                   1 ;*******************************************************************************
                   2 ;
                   3 ;
                   4 ;                        PROGRAM: 8085 TO BPK72 SOFTWARE DRIVER V1.0
                   5 ;                        ULMONT S. SMITH JR.
                   6 ;                        INTEL CORPORATION
                   7 ;                        3065 BOWERS AVENUE
                   8 ;                        SANTA CLARA, CALIFORNIA 95051
                   9 ;
                  10 ;
                  11 ;*******************************************************************************
                  12 ;
                  13 ;
                  14 ;
                  15 ; ABSTRACT:
                  16 ;
                  17 ;       THIS PROGRAM  CONSISTS OF A SET OF  BUBBLE MEMORY SOFTWARE DRIVERS
                  18 ;       THAT SUPPORT A POLLED MODE INTERFACE BETWEEN A BPK72, 1MBIT BUBBLE
                  19 ;       MEMORY PROTOTYPE KIT,  AND  A STANDARD  8085  MICROPROCESSOR.  THE
                  20 ;       PROGRAM  UTILIZES A SET  OF PUBLIC  DIRECTIVES THAT  CAN BE CALLED
                  21 ;       TO PERFORM A BUBBLE MEMORY INITIALIZATION, READ, WRITE,  AND OTHER
                  22 ;       COMMONLY USED COMMANDS.  IN THE UNLIKELY EVENT THAT THE 7110 BUBBLE
                  23 ;       MEMORY BOOT LOOP IS LOST, TWO ROUTINES ARE PROVIDED TO EXAMINE AND
                  24 ;       REWRITE THE BOOT LOOP CODE.
                  25 ;
                  26 ;
                  27 ;
                  28 ; PROGRAM ORGANIZATION:
                  29 ;
                  30 ;       FUNCTIONS:
                  31 ;                        INTPAR
                  32 ;                        FIFORS
                  33 ;                        BYTCNT
                  34 ;                        WRITE
                  35 ;                        READ
                  36 ;                        ABORT
                  37 ;                        WRBUBL
                  38 ;                        RDBUBL
                  39 ;                        INBUBL
                  40 ;                        BOOTUP
                  41 ;                        RDBOOT
                  42 ;                        WRFIFO
                  43 ;                        RDFIFO
                  44 ;                        WRBLRS
                  45 ;                        RDBLRS
                  46 ;                        MBMPRG
                  47 ;
                  48 ;
                  49 ; EXTERNAL DECLARATIONS: NONE
                  50 ;
                  51 ;
                  52 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    2


 LOC  OBJ        LINE       SOURCE STATEMENT

                   53 ;
                   54 ; PUBLIC SYMBOLS:
                   55 ;
                   56 ;              FIFORS - RESET 7220 FIFO DATA BUFFER
                   57 ;              ABORT  - ABORT PRESENT COMMAND, RESET BPK72
                   58 ;              WRBUBL - WRITE BUBBLE MEMORY DATA
                   59 ;              RDBUBL - READ BUBBLE MEMORY DATA
                   60 ;              INBUBL - INITIALIZE THE BPK72
                   61 ;              BOOTUP - WRITE BUBBLE MEMORY BOOT LOOP
                   62 ;              RDBOOT - READ BUBBLE MEMORY BOOT LOOP
                   63 ;              WRFIFO - WRITE 7220 FIFO DATA BUFFER
                   64 ;              RDFIFO - READ 7220 FIFO DATA BUFFER
                   65 ;              WRBLRS - WRITE 7242 BOOT LOOP REGISTERS
                   66 ;              RDBLRS - READ 7242 BOOT LOOP REGISTERS
                   67 ;              MBMPRG - BUBBLE MEMORY PURGE COMMAND
                   68 ;
                   69 ;
                   70 ;*******************************************************************************
                   71 ;
                   72                          NAME BPK72
                   73 ;
                   74 ;*******************************************************************************
                   75 ;
                   76 ;
                   77 ;*******************************************************************************
                   78 ;
 0800              79                          ORG     0800H
                   80 ;
                   81 ;*******************************************************************************
                   82 ;
                   83 ;
                   84 ;*******************************************************************************
                   85 ;
                   86 ;                        PROGRAM EQUATES
                   87 ;
                   88 ;*******************************************************************************
                   89 ;
                   90 ;
 00FE              91 PRTA00  EQU     0FEH    ; A POLLED  MODE  INTERFACE  REQUIRES  ONLY  TWO  I/O
 00FF              92 PRTA01  EQU     0FFH    ; PORTS DESIGNATED BY THE A0 LINE ON THE BPK72 BOARD.
                   93                         ; THIS APPLICATION USES:
                   94                         ;
                   95                         ;       0FEH - A0=0 FOR PRTA00 (PORT A0= 0)
                   96                         ;              RD/WR BUBBLE MEMORY DATA AND REGS
                   97                         ;
                   98                         ;       0FFH - A0=1 FOR PRTA01 (PORT A0= 1)
                   99                         ;              RD STATUS REG
                  100                         ;              WR BUBBLE MEMORY COMMANDS
                  101 ;
                  102 $EJECT
```

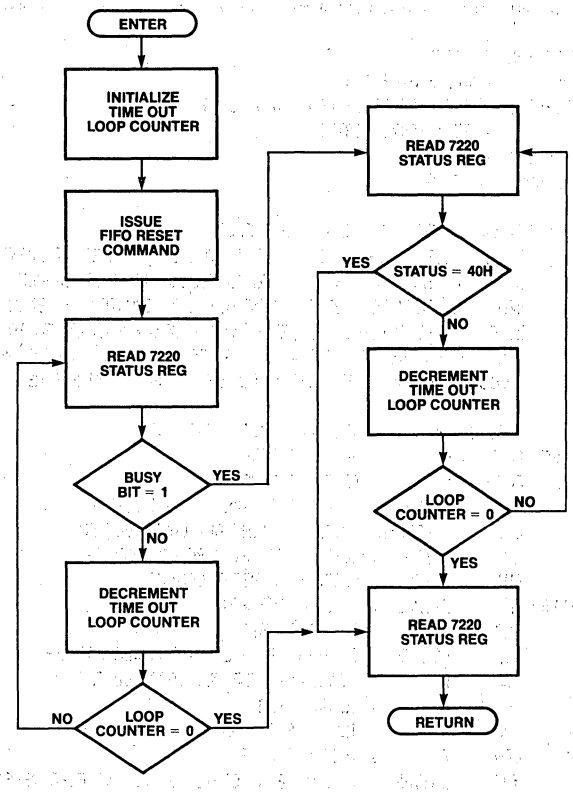
```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    3


 LOC  OBJ         LINE        SOURCE STATEMENT

                   103 ;*******************************************************************************
                   104 ;
                   105 ; FUNCTION: INTPAR
                   106 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                   107 ; OUTPUTS:  7220 PARAMETRIC REGS
                   108 ; CALLS:    NONE
                   109 ; DESTROYS: A, F/FS
                   110 ;
                   111 ; DESCRIPTION: LOAD THE 7220 PARAMETRIC REGS
                   112 ;       THE B-C REGS  CONTAIN THE ADDRESS TO THE FIRST OF FIVE CONTIGUOUS
                   113 ;       MEMORY LOCATIONS IN RAM.  THE DATA  ADDRESSED BY THE  B-C REGS IS
                   114 ;       USED TO LOAD  THE PARAMETRIC  REGISTERS IN THE 7220 BUBBLE MEMORY
                   115 ;       CONTROLLER.  INTPAR COPIES THE DATA IN RAM TO THE PARAMETRIC REGS.
                   116 ;
 0800 C5           117 INTPAR: PUSH    B       ; SAVE B-C REGS
 0801 D5           118         PUSH    D       ; SAVE D-E REGS
 0802 3E0B         119         MVI     A,0BH   ; LOAD A REG WITH BLR LSB ADDRESS
 0804 D3FF         120         OUT     PRTA01  ; LOAD 7220 RAC WITH BLR LSB ADDRESS
 0806 1E05         121         MVI     E,05H   ; INITIALIZE LOOP COUNTER
 0808 0A           122 LOAD:   LDAX    B       ; LOAD A REG FROM B-C REG ADDRESS
 0809 D3FE         123         OUT     PRTA00  ; WRITE PARAMETRIC REG
 080B 03           124         INX     B       ; INCREMENT B-C REGS TO THE NEXT ADDRESS IN RAM
 080C 1D           125         DCR     E       ; DECREMENT LOOP COUNTER
 080D C20808       126         JNZ     LOAD    ; IF NOT ZERO, JMP LOAD
 0810 D1           127         POP     D       ; RESTORE D-E REGS
 0811 C1           128         POP     B       ; RESTORE B-C REGS
 0812 C9           129         RET             ; RETURN TO CALL
                   130 ;
                   131 ;
                   132 ;
                   133 $EJECT
```



COMMENTS: THE UTILITY REGISTER IS NOT USED. THE RAC IS AUTOMATICALLY INCREMENTED AFTER EACH WRITE (WR/) IS EXECUTED. THE RAC WILL NOT INCREMENT BEYOND 00H.

Figure 11. INTPAR

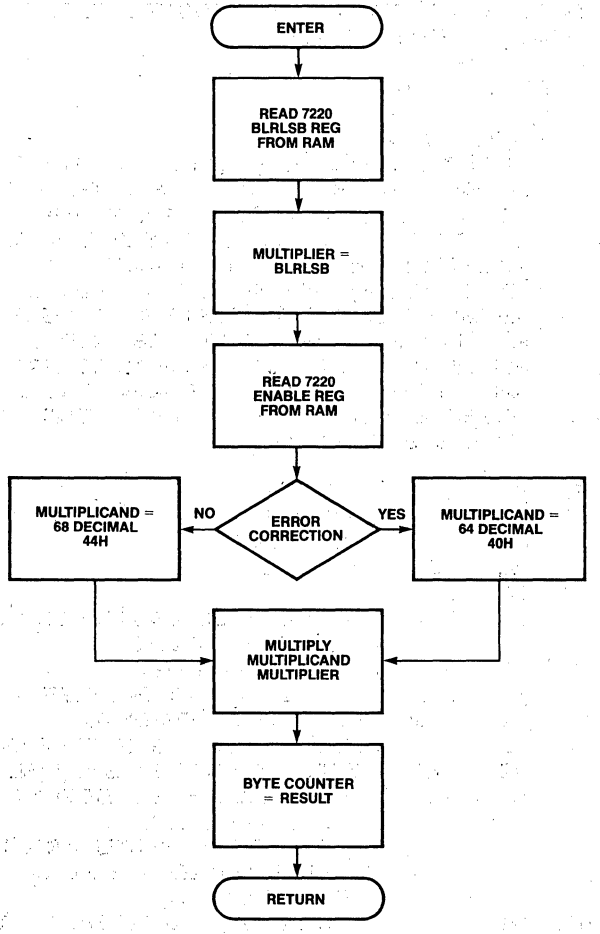
```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    4


 LOC  OBJ        LINE       SOURCE STATEMENT

                  134 ;*******************************************************************************
                  135 ;
                  136 ; FUNCTION: FIFORS
                  137 ; INPUTS:   BPK72 STATUS REG
                  138 ; OUTPUTS:  ISSUE FIFO RESET COMMAND TO BPK72
                  139 ;           A REG= BPK72 STATUS REG
                  140 ; CALLS:    NONE
                  141 ; DESTROYS: A, F/FS
                  142 ;
                  143 ; DESCRIPTION: RESET 7220 FIFO DATA BUFFER
                  144 ;       A FIFO RESET COMMAND IS ISSUED TO THE BPK72.  AFTER ISSUING THE
                  145 ;       COMMAND,  THE BPK72 STATUS REG IS POLLED UNTIL AN OP-COMPLETE,
                  146 ;       40H,  HAS BEEN READ OR THE TIME OUT LOOP COUNTER DECREMENTS TO
                  147 ;       ZERO.  FIFORS RETURNS THE VALUE OF THE BPK72 STATUS REG TO  THE
                  148 ;       CALLING ROUTINE  VIA THE 8085'S A REG.  ONLY  A  STATUS OF 40H
                  149 ;       INDICATES A SUCCESSFUL EXECUTION OF THE FUNCTION FIFORS.
                  150 ;
                  151         PUBLIC  FIFORS  ; DECLARE PUBLIC FUNCTION
 0813 D5          152 FIFORS: PUSH    D       ; SAVE D-E REGS
 0814 C5          153         PUSH    B       ; SAVE B-C REGS
 0815 0640        154         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 0817 11FFFF      155         LXI     D,0FFFFH; INTIALIZE TIME OUT LOOP COUNTER
 081A 3E1D        156         MVI     A,1DH   ; LOAD A REG= FIFO RESET COMMAND
 081C D3FF        157         OUT     PRTA01  ; WRITE FIFO RESET COMMAND
 081E DBFF        158 BUSYFR: IN      PRTA01  ; READ STATUS REG
 0820 07          159         RLC             ; TEST BUSY BIT= 1
 0821 DA2E08      160         JC      POLLFR  ; IF BUSY= 1, POLL STATUS REG FOR 40H
 0824 1B          161         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0825 AF          162         XRA     A       ; CLEAR A REG
 0826 B2          163         ORA     D       ; TEST D REG= 00H
 0827 B3          164         ORA     E       ; TEST E REG= 00H
 0828 C21E08      165         JNZ     BUSYFR  ; IF NOT ZERO, CONTINUE POLLING FIFO RESET COMMAND
 082B C33B08      166         JMP     RETFR   ; TIME OUT ERROR, RETURN
 082E DBFF        167 POLLFR: IN      PRTA01  ; READ STATUS REG
 0830 A8          168         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
 0831 CA3B08      169         JZ      RETFR   ; IF OP-COMPLETE, JMP RETFR
 0834 1B          170         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0835 AF          171         XRA     A       ; CLEAR A REG
 0836 B2          172         ORA     D       ; TEST D REG= 00H
 0837 B3          173         ORA     E       ; TEST E REG= 00H
 0838 C22E08      174         JNZ     POLLFR  ; IF NOT ZERO, CONTINUE POLLING FIFO RESET COMMAND
 083B C1          175 RETFR:  POP     B       ; RESTORE B-C REGS
 083C D1          176         POP     D       ; RESTORE D-E REGS
 083D DBFF        177         IN      PRTA01  ; READ STATUS REG
 083F C9          178         RET             ; RETURN TO CALL
                  179 ;
                  180 ;
                  181 ;
                  182 $EJECT
```



COMMENT: MINIMUM TIME OUT LOOP COUNT = 50μS.

Figure 12. FIFORS

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    5


 LOC  OBJ        LINE       SOURCE STATEMENT

                  183 ;*********************************************************************************
                  184 ; FUNCTION: BYTCNT
                  185 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                  186 ; OUTPUTS:  H-L REGS= BYTE COUNTER
                  187 ; CALLS:    NONE
                  188 ; DESTROYS: A, H, L, F/FS
                  189 ;
                  190 ; DESCRIPTION: BYTE COUNTER
                  191 ;       THE B-C REGS CONTAIN THE ADDRESS TO THE FIRST OF FIVE CONTIGUOUS MEMORY
                  192 ;       LOCATIONS IN RAM.  THE DATA  ADDRESSED BY  THE B-C REGS IS USED TO LOAD
                  193 ;       THE PARAMETRIC REGS IN  THE 7220  BUBBLE MEMORY CONTROLLER.  THE ENABLE
                  194 ;       REG IS READ FROM RAM TO DETERMINE IF ERROR CORRECTION HAS BEEN ENABLED.
                  195 ;       THE USE OF ERROR CORRECTION REQUIRES A 64 BYTE TRANSFER/PAGE  - 68 BYTE
                  196 ;       TRANSFER/PAGE WITHOUT ERROR CORRECTION.  THE  BLOCK  LENGTH  REG LSB IS
                  197 ;       ALSO READ FROM RAM TO  DETERMINE THE NUMBER  OF PAGES TO BE TRANSFERRED
                  198 ;       DURING THE NEXT READ OR WRITE COMMAND.  THE  NUMBER  OF  BYTES PER PAGE
                  199 ;       MULTIPLIED BY THE NUMBER OF PAGES IS COMPUTED AND PASSED TO THE CALLING
                  200 ;       ROUTINE VIA THE 8085'S H-L REGS.  DATA TRANSFERS ARE LIMITED TO  16,320
                  201 ;       BYTES WITH ERROR CORRECTION AND  17,340 BYTES WITHOUT.  ONLY THE BLRLSB
                  202 ;       IS USED TO GENERATE THE BYTE COUNTER.
                  203 ;
  0840 C5         204 BYTCNT: PUSH    B       ; SAVE B-C REGS
  0841 D5         205         PUSH    D       ; SAVE D-E REGS
  0842 0A         206         LDAX    B       ; LOAD A REG WITH BLRLSB
  0843 6F         207         MOV     L,A     ; MOVE BLRLSB TO L REG
  0844 03         208         INX     B       ;
  0845 03         209         INX     B       ; INCREMENT B-C REGS TO ADDRESS THE ENABLE REG IN RAM
  0846 0A         210         LDAX    B       ; LOAD A REG WITH ENABLE REG
  0847 67         211         MOV     H,A     ; MOVE ENABLE REG TO H REG
  0848 1640       212         MVI     D,40H   ; INITIALIZE D REG 64 BYTES/PAGE XFER, 40H
  084A 3E60       213         MVI     A,60H   ; ERROR CORRECTION DETECTION MASK
  084C A4         214         ANA     H       ; LOGICAL AND MASK WITH H REG, TEST FOR ERROR CORRECTION
  084D C25208     215         JNZ     MULT    ; IF ZERO, ERROR CORRECTION IS NOT ENABLED
  0850 1644       216         MVI     D,44H   ; NO ERROR CORRECTION, 68 BYTES/PAGE XFER, 44H
                  217                         ; MULTIPLY (D REG) X (L REG)
                  218                         ; 64 OR 68 BYTES X NO. OF PAGES IN BLRLSB
                  219                         ; RESULT WILL BE PLACED IN THE H-L REGS
                  220                         ; BEGIN MULTIPLY ROUTINE
  0852 2600       221 MULT:   MVI     H,0H    ; INITIALIZE MOST SIGNIFICANT BYTE OF RESULT
  0854 1E09       222         MVI     E,09H   ; INITIALIZE BIT COUNTER
  0856 7D         223 MULT0:  MOV     A,L     ; MOVE LOW ORDER BYTE INTO A REG
  0857 1F         224         RAR             ; ROTATE LEAST SIGNIFICANT BIT OF MULTIPLIER
  0858 6F         225         MOV     L,A     ; MOVE LOW ORDER BYTE OF RESULT INTO L REG
  0859 1D         226         DCR     E       ; DECREMENT BIT COUNTER
  085A CA6708     227         JZ      DONE    ; EXIT IF COMPLETE
  085D 7C         228         MOV     A,H     ; MOVE HIGH ORDER BYTE INTO A REG
  085E D26208     229         JNC     MULT1   ; IF CARRY= 0, JMP MULTI
  0861 82         230         ADD     D       ; ADD D REG TO A REG
  0862 1F         231 MULT1:  RAR             ; CARRY= 0, SHIFT HIGH ORDER BYTE OF RESULT
  0863 67         232         MOV     H,A     ; MOVE HIGH ORDER RESULT INTO H REG
  0864 C35608     233         JMP     MULT0   ; CONTINUE LOOPING
  0867 D1         234 DONE:   POP     D       ; RESTORE D-E REGS
  0868 C1         235         POP     B       ; RESTORE B-C REGS
  0869 C9         236         RET             ; RETURN TO CALL
                  237 ;
```



COMMENTS: THE PARAMETRIC REGS—BLRLSB AND THE ENABLE REG CAN NOT BE READ FROM THE 7220. THEY MUST BE READ FROM A MEMORY IMAGE IN RAM. SINCE ONLY THE BLRLSB IS USED TO COMPUTE THE BYTE COUNTER, DATA TRANSFERS ARE LIMITED TO 255 PAGES.

Figure 13. BYTCNT

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    6


 LOC  OBJ         LINE       SOURCE STATEMENT

                   238 ;*******************************************************************************
                   239 ;
                   240 ; FUNCTION: WRITE
                   241 ; INPUTS:   D-E REGS, STARTING ADDRESS OF DATA IN RAM
                   242 ;           H-L REGS, BYTE COUNTER
                   243 ;           BPK72 STATUS REG
                   244 ; OUTPUTS:  WRITE DATA TO BUBBLE MEMORY
                   245 ; CALLS:    NONE
                   246 ; DESTROYS: A, H, L, F/FS
                   247 ;
                   248 ; DESCRIPTION: TRANSFER DATA FROM RAM TO BUBBLE MEMORY
                   249 ;       THE  D-E REGS  CONTAIN THE STARTING  ADDRESS  IN  RAM OF  DATA
                   250 ;       TO BE WRITTEN  INTO  THE  BUBBLE  MEMORY.  THE  H-L  REGS MUST
                   251 ;       CONTAIN  A BYTE COUNTER  INDICATING  THE  NUMBER OF DATA BYTES
                   252 ;       TO BE TRANSFERRED.  THIS FUNCTION  BEGINS BY ISSUING THE WRITE
                   253 ;       BUBBLE MEMORY  DATA COMMAND FOLLOWED BY POLLING THE STATUS REG
                   254 ;       TO  DETERMINE IF THE 7220 FIFO DATA BUFFER IS READY TO RECEIVE
                   255 ;       DATA.  DATA IS  TRANSFERRED  UNTIL  THE  BYTE  COUNTER OR TIME
                   256 ;       OUT LOOP COUNTER DECREMENTS TO ZERO.  THE PARAMETRIC REGISTERS
                   257 ;       MUST BE LOADED  WITH THE DESIRED VALUES  PRIOR TO CALLING THIS
                   258 ;       FUNCTION.
                   259 ;
 086A D5           260 WRITE:  PUSH    D       ; SAVE D-E REGS
 086B C5           261         PUSH    B       ; SAVE B-C REGS
 086C 01FFFF       262         LXI     B,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 086F 3E13         263         MVI     A,13H   ; LOAD A REG= WRITE BUBBLE MEMORY DATA COMMAND
 0871 D3FF         264         OUT     PRTA01  ; WRITE, WRITE BUBBLE MEMORY DATA COMMAND
 0873 0B           265 BUSYWR: DCX     B       ; DECREMENT TIME OUT LOOP COUNTER
 0874 AF           266         XRA     A       ; CLEAR A REG
 0875 B0           267         ORA     B       ; TEST B REG= 00H
 0876 B1           268         ORA     C       ; TEST C REG= 00H
 0877 CA8108       269         JZ      FINSHW  ; IF ZERO, TIME OUT ERROR, JMP FINSHW
 087A DBFF         270         IN      PRTA01  ; READ STATUS REG
 087C 07           271         RLC             ; TEST BUSY BIT= 1
 087D D27308       272         JNC     BUSYWR  ; IF ZERO, CONTINUE POLLING BUSY BIT
                   273                         ; CONTINUED ON NEXT PAGE
                   274 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    7


 LOC  OBJ        LINE        SOURCE STATEMENT

 0880 DBFF        275 POLLWR: IN      PRTA01  ; READ STATUS REG
 0882 0F          276         RRC             ; TEST FIFO READY BIT= 1
 0883 DA9608      277         JC      WFIFO   ; IF FIFO READY= 1, JMP WFIFO
 0886 DBFF        278         IN      PRTA01  ; READ STATUS REG
 0888 07          279         RLC             ; TEST BUSY BIT= 1
 0889 D2A108      280         JNC     FINSHW  ; IF ZERO, ERROR, JMP FINSHW
 088C 0B          281         DCX     B       ; DECREMENT TIME OUT LOOP COUNTER
 088D AF          282         XRA     A       ; CLEAR A REG
 088E B0          283         ORA     B       ; TEST B REG= 00H
 088F B1          284         ORA     C       ; TEST C REG= 00H
 0890 CAA108      285         JZ      FINSHW  ; IF ZERO, TIME OUT ERROR, JMP FINSHW
 0893 C38008      286         JMP     POLLWR  ; CONTINUE POLLING FIFO READY BIT
 0896 1A          287 WFIFO:  LDAX    D       ; LOAD A REG FROM D-E REG ADDRESS
 0897 D3FE        288         OUT     PRTA00  ; WRITE A REG TO 7220 FIFO DATA BUFFER
 0899 13          289         INX     D       ; INCREMENT D-E REGS TO NEXT ADDRESS IN RAM
 089A 2B          290         DCX     H       ; DECREMENT BYTE COUNTER
 089B AF          291         XRA     A       ; CLEAR A REG
 089C B4          292         ORA     H       ; TEST H REG= 00H
 089D B5          293         ORA     L       ; TEST L REG= 00H
 089E C28008      294         JNZ     POLLWR  ; IF BYTE COUNTER NOT ZERO, JMP POLLWR
 08A1 C1          295 FINSHW: POP     B       ; RESTORE B-C REGS
 08A2 D1          296         POP     D       ; RESTORE D-E REGS
 08A3 C9          297         RET             ; RETURN TO CALL
                  298 ;
                  299 ;
                  300 ;
                  301 $EJECT
```

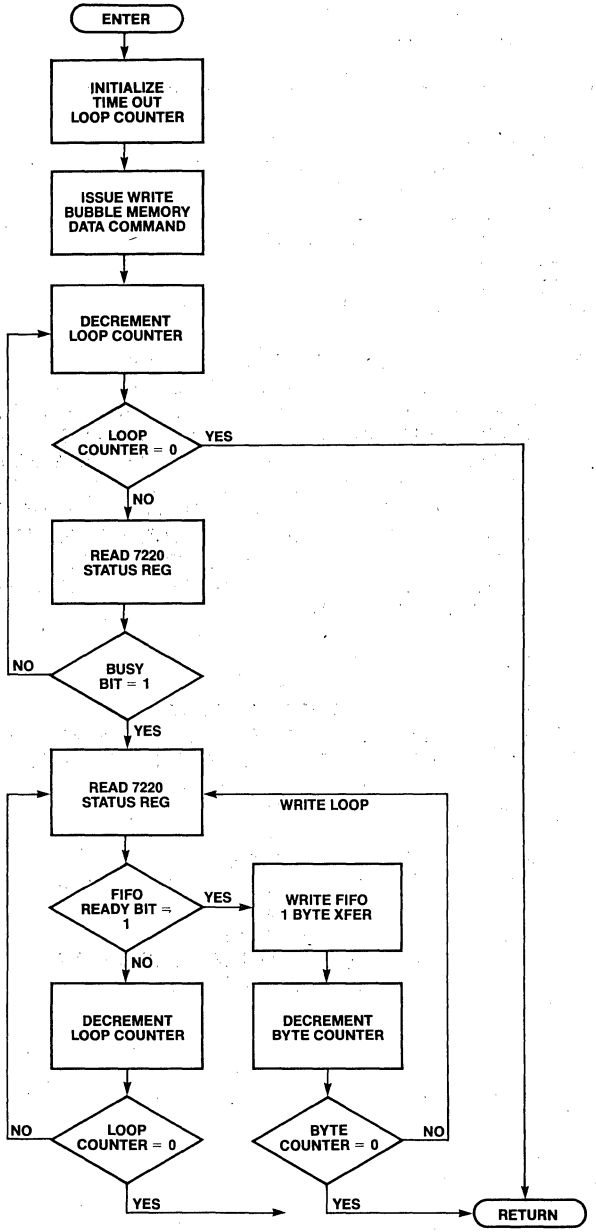


Figure 14. WRITE

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    8

 LOC  OBJ        LINE       SOURCE STATEMENT

                  302 ;*******************************************************************************
                  303 ;
                  304 ; FUNCTION:'READ
                  305 ; INPUTS:   D-E REGS, STARTING ADDRESS IN RAM
                  306 ;           H-L REGS, BYTE COUNTER
                  307 ;           BPK72 STATUS REG
                  308 ;           READ DATA FROM BUBBLE MEMORY
                  309 ; OUTPUTS:  WRITE DATA TO RAM
                  310 ; CALLS:    NONE
                  311 ; DESTROYS: A, H, L, F/FS
                  312 ;
                  313 ; DESCRIPTION: TRANSFER DATA FROM BUBBLE MEMORY TO RAM
                  314 ;       THE D-E REGS CONTAIN THE STARTING ADDRESS IN RAM USED TO STORE
                  315 ;       DATA READ FROM THE  BUBBLE MEMORY.  THE H-L  REGS MUST CONTAIN
                  316 ;       A BYTE  COUNTER  INDICATING  THE  NUMBER  OF DATA  BYTES TO BE
                  317 ;       TRANSFERRED.  THIS FUNCTION BEGINS  BY ISSUING THE READ BUBBLE
                  318 ;       MEMORY  DATA  COMMAND  FOLLOWED  BY  POLLING  THE  STATUS  REG
                  319 ;       TO DETERMINE  IF  THE  7220 FIFO  DATA  BUFFER  CONTAINS  DATA
                  320 ;       AVAILABLE FOR READING.  DATA  IS  TRANSFERRED  UNTIL  THE BYTE
                  321 ;       COUNTER  OR  TIME  OUT  LOOP COUNTER  DECREMENTS TO ZERO.  THE
                  322 ;       PARAMETRIC REGS MUST BE LOADED  WITH THE  DESIRED VALUES PRIOR
                  323 ;       TO CALLING THIS FUNCTION.
                  324 ;
 08A4 D5          325 READ:   PUSH    D       ; SAVE D-E REGS
 08A5 C5          326         PUSH    B       ; SAVE B-C REGS
 08A6 01FFFF      327         LXI     B,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 08A9 3E12        328         MVI     A,12H   ; LOAD A REG= READ BUBBLE MEMORY DATA COMMAND
 08AB D3FF        329         OUT     PRTA01  ; WRITE, READ BUBBLE MEMORY DATA COMMAND
 08AD 0B          330 BUSYRD: DCX     B       ; DECREMENT TIME OUT LOOP COUNTER
 08AE AF          331         XRA     A       ; CLEAR A REG
 08AF B0          332         ORA     B       ; TEST B REG= 00H
 08B0 B1          333         ORA     C       ; TEST C REG= 00H
 08B1 CADB08      334         JZ      FINSHR  ; IF ZERO, TIME OUT ERROR, JMP FINSHR
 08B4 DBFF        335         IN      PRTA01  ; READ STATUS REG
 08B6 07          336         RLC             ; TEST BUSY BIT= 1
 08B7 D2AD08      337         JNC     BUSYRD  ; IF ZERO, CONTINUE POLLING BUSY BIT
                  338                         ; CONTINUED ON NEXT PAGE
                  339 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    9


 LOC  OBJ        LINE        SOURCE STATEMENT

 08BA DBFF        340 POLLRD: IN      PRTA01  ; READ STATUS REG
 08BC 0F          341         RRC             ; TEST FIFO READY BIT= 1
 08BD DAD008      342         JC      RFIFO   ; IF FIFO READY= 1, JMP RFIFO
 08C0 DBFF        343         IN      PRTA01  ; READ STATUS REG
 08C2 07          344         RLC             ; TEST BUSY BIT= 1
 08C3 D2DB08      345         JNC     FINSHR  ; IF ZERO, ERROR, JMP FINSHR
 08C6 0B          346         DCX     B       ; DECREMENT TIME OUT LOOP COUNTER
 08C7 AF          347         XRA     A       ; CLEAR A REG
 08C8 B0          348         ORA     B       ; TEST B REG= 00H
 08C9 B1          349         ORA     C       ; TEST C REG= 00H
 08CA CADB08      350         JZ      FINSHR  ; IF ZERO, TIME OUT ERROR, JMP FINSHR
 08CD C3BA08      351         JMP     POLLRD  ; CONTINUE POLLING FIFO READY BIT
 08D0 DBFE        352 RFIFO:  IN      PRTA00  ; LOAD A REG WITH ONE BYTE FROM FIFO DATA BUFFER
 08D2 12          353         STAX    D       ; STORE A REG IN REG D-E ADDRESS
 08D3 13          354         INX     D       ; INCREMENT D-E REGS TO NEXT ADDRESS IN RAM
 08D4 2B          355         DCX     H       ; DECREMENT BYTE COUNTER
 08D5 AF          356         XRA     A       ; CLEAR A REG
 08D6 B4          357         ORA     H       ; TEST H REG= 00H
 08D7 B5          358         ORA     L       ; TEST L REG= 00H
 08D8 C2BA08      359         JNZ     POLLRD  ; IF BYTE COUNTER NOT ZERO, JMP POLLRD
 08DB C1          360 FINSHR: POP     B       ; RESTORE B-C REGS
 08DC D1          361         POP     D       ; RESTORE D-E REGS
 08DD C9          362         RET             ; RETURN TO CALL
                  363 ;
                  364 ;
                  365 ;
                  366 $EJECT
```

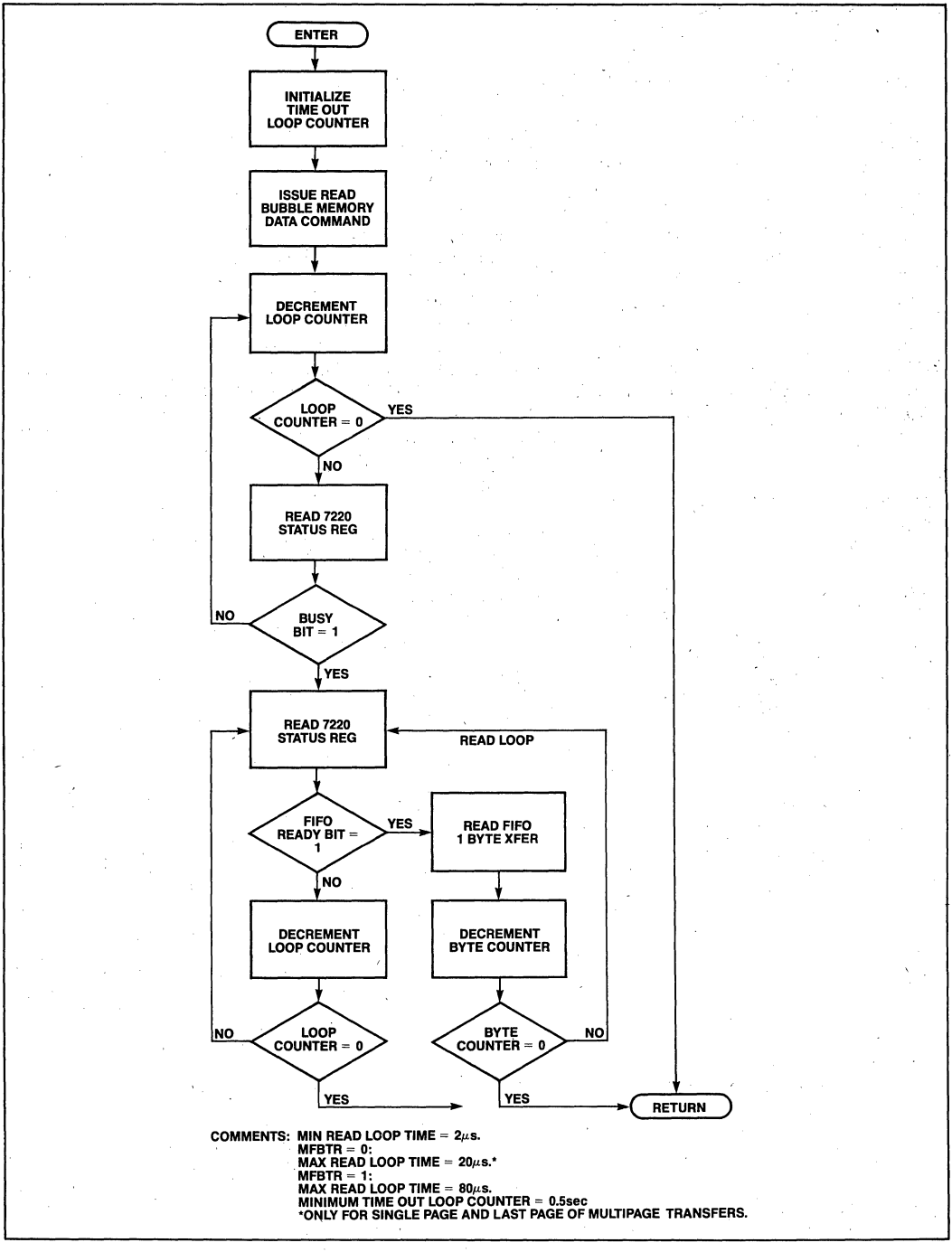


Figure 15. READ

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   10


 LOC  OBJ        LINE       SOURCE STATEMENT

                  367 ;*******************************************************************************
                  368 ;
                  369 ; FUNCTION: ABORT
                  370 ; INPUTS:   BPK72 STATUS REG
                  371 ; OUTPUTS:  ISSUE ABORT COMMAND TO BPK72
                  372 ;           A REG= BPK72 STATUS REG
                  373 ; CALLS:    NONE
                  374 ; DESTROYS: A, F/FS
                  375 ;
                  376 ; DESCRIPTION: ABORT PRESENT COMMAND, RESET BPK72
                  377 ;       AN ABORT  COMMAND IS  ISSUED  TO THE BPK72.  AFTER  ISSUING THE
                  378 ;       COMMAND,  THE  BPK72 STATUS REG IS POLLED UNTIL AN OP-COMPLETE,
                  379 ;       40H,  HAS BEEN  READ  OR THE  TIME OUT  LOOP COUNTER DECREMENTS
                  380 ;       TO ZERO.  THE ABORT FUNCTION  RETURNS  THE  VALUE  OF THE BPK72
                  381 ;       STATUS REG TO THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY A
                  382 ;       STATUS  OF 40H INDICATES A  SUCCESSFUL  EXECUTION  OF THE ABORT
                  383 ;       FUNCTION.
                  384 ;
                  385         PUBLIC  ABORT   ; DECLARE PUBLIC FUNCTION
  08DE D5         386 ABORT:  PUSH    D       ; SAVE D-E REGS
  08DF C5         387         PUSH    B       ; SAVE B-C REGS
  08E0 11FFFF     388         LXI     D,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
  08E3 0640       389         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
  08E5 3E19       390         MVI     A,19H   ; LOAD A REG= ABORT COMMAND
  08E7 D3FF       391         OUT     PRTA01  ; WRITE ABORT COMMAND
  08E9 DBFF       392 BUSYA:  IN      PRTA01  ; READ STATUS REG
  08EB 07         393         RLC             ; TEST BUSY BIT= 1
  08EC DAF908     394         JC      POLLA   ; IF BUSY= 1, POLL STATUS REG FOR 40H
  08EF 1B         395         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
  08F0 AF         396         XRA     A       ; CLEAR A REG
  08F1 B2         397         ORA     D       ; TEST D REG= 00H
  08F2 B3         398         ORA     E       ; TEST E REG= 00H
  08F3 C2E908     399         JNZ     BUSYA   ; IF NOT ZERO, CONTINUE POLLING ABORT COMMAND
  08F6 C30609     400         JMP     RETA    ; TIME OUT ERROR, RETURN
  08F9 DBFF       401 POLLA:  IN      PRTA01  ; READ STATUS REG
  08FB A8         402         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
  08FC CA0609     403         JZ      RETA    ; IF OP-COMPLETE, JMP RETA
  08FF 1B         404         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
  0900 AF         405         XRA     A       ; CLEAR A REG
  0901 B2         406         ORA     D       ; TEST D REG= 00H
  0902 B3         407         ORA     E       ; TEST E REG= 00H
  0903 C2F908     408         JNZ     POLLA   ; IF NOT ZERO, CONTINUE POLLING ABORT COMMAND
  0906 C1         409 RETA:   POP     B       ; RESTORE B-C REGS
  0907 D1         410         POP     D       ; RESTORE D-E REGS
  0908 DBFF       411         IN      PRTA01  ; READ STATUS REG
  090A C9         412         RET             ; RETURN TO CALL
                  413 ;
                  414 ;
                  415 ;
                  416 $EJECT
```

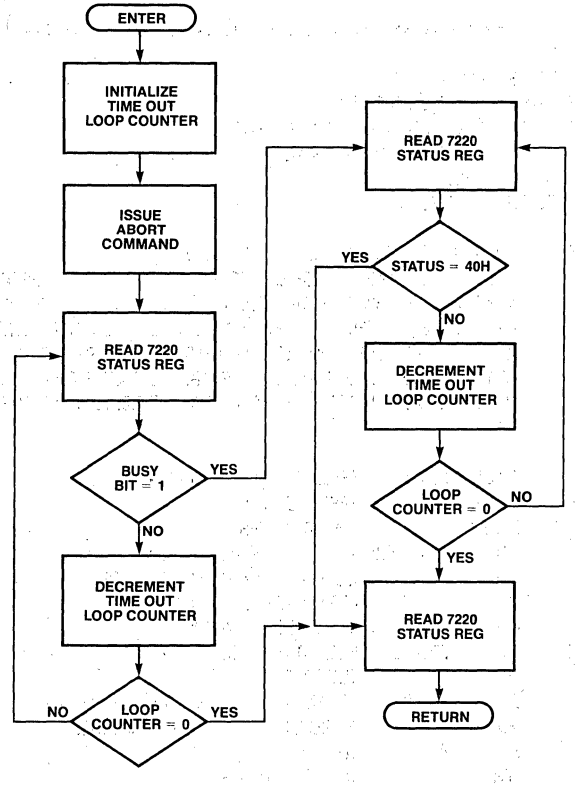


Figure 16. ABORT

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   11

 LOC  OBJ        LINE       SOURCE STATEMENT

                  417 ;*******************************************************************************
                  418 ;
                  419 ; FUNCTION: WRBUBL
                  420 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                  421 ;           D-E REGS, STARTING ADDRESS OF DATA IN RAM
                  422 ;           BPK72 STATUS REG
                  423 ; OUTPUTS:  WRITE DATA TO BUBBLE MEMORY
                  424 ;           A REG= BPK72 STATUS REG
                  425 ; CALLS:    FIFORS
                  426 ;           INTPAR
                  427 ;           BYTCNT
                  428 ;           WRITE
                  429 ; DESTROYS: A, F/FS
                  430 ;
                  431 ; DESCRIPTION: WRITE BUBBLE MEMORY DATA
                  432 ;       THE B-C  REGS CONTAIN THE ADDRESS  TO THE  FIRST  OF FIVE
                  433 ;       CONTIGUOUS  MEMORY  LOCATIONS IN RAM.  THE DATA ADDRESSED
                  434 ;       BY THE  B-C  REGS IS USED  TO LOAD THE  PARAMETRIC  REGS.
                  435 ;       THE D-E  REGS  CONTAIN  THE STARTING ADDRESS  IN  RAM  OF
                  436 ;       DATA TO BE WRITTEN INTO THE BUBBLE MEMORY. GIVEN THE DATA
                  437 ;       IN RAM  USED TO  LOAD THE PARAMETRIC REGS,  THIS FUNCTION
                  438 ;       WILL  RESET  THE 7220 FIFO,  LOAD  THE  PARAMETRIC  REGS,
                  439 ;       COMPUTE THE BYTE COUNTER, AND COPY THE DATA FROM RAM INTO
                  440 ;       THE BUBBLE MEMORY.  WRBUBL RETURNS THE VALUE OF THE BPK72
                  441 ;       STATUS  REG TO THE CALLING ROUTINE VIA  THE 8085'S A REG.
                  442 ;       ONLY  A  STATUS  OF  40H OR 42H  INDICATES  A  SUCCESSFUL
                  443 ;       EXECUTION OF WRBUBL.
                  444 ;
                  445         PUBLIC  WRBUBL  ; DECLARE PUBLIC FUNCTION
090B E5           446 WRBUBL: PUSH    H       ; SAVE H-L REGS
090C C5           447         PUSH    B       ; SAVE B-C REGS
090D 0640         448         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
090F CD1308       449         CALL    FIFORS  ; CALL FIFORS, WRITE FIFO RESET COMMAND
0912 A8           450         XRA     B       ; TEST FOR STATUS= 40H, OP-COMPLETE
0913 C23109       451         JNZ     RETWR   ; IF NOT ZERO, FIFO ERROR, JMP RETWR
0916 C1           452         POP     B       ; RESTORE B-C REGS
0917 CD0008       453         CALL    INTPAR  ; CALL INTPAR, LOAD PARAMETRIC REGS
091A CD4008       454         CALL    BYTCNT  ; CALL BYTCNT, COMPUTE BYTE COUNTER
091D CD6A08       455         CALL    WRITE   ; CALL WRITE,  WRITE BUBBLE DATA
0920 C5           456         PUSH    B       ; SAVE B-C REGS
                  457                         ; CONTINUED ON NEXT PAGE
                  458 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   12


 LOC  OBJ        LINE        SOURCE STATEMENT

 0921 21FFFF      459         LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 0924 DBFF        460 LOOPWR: IN      PRTA01  ; READ STATUS REG
 0926 07          461         RLC             ; TEST FOR BUSY BIT= 1
 0927 D23109      462         JNC     RETWR   ; IF ZERO, NOT BUSY, JMP RETWR
 092A 2B          463         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 092B AF          464         XRA     A       ; CLEAR A REG
 092C B4          465         ORA     H       ; TEST H REG= 00H
 092D B5          466         ORA     L       ; TEST L REG= 00H
 092E C22409      467         JNZ     LOOPWR  ; CONTINUE POLLING STATUS REG
 0931 C1          468 RETWR:  POP     B       ; RESTORE B-C REGS
 0932 E1          469         POP     H       ; RESTORE H-L REGS
 0933 DBFF        470         IN      PRTA01  ; READ STATUS REG
 0935 C9          471         RET             ; RETURN TO CALL
                  472 ;
                  473 ;
                  474 ;
                  475 $EJECT
```

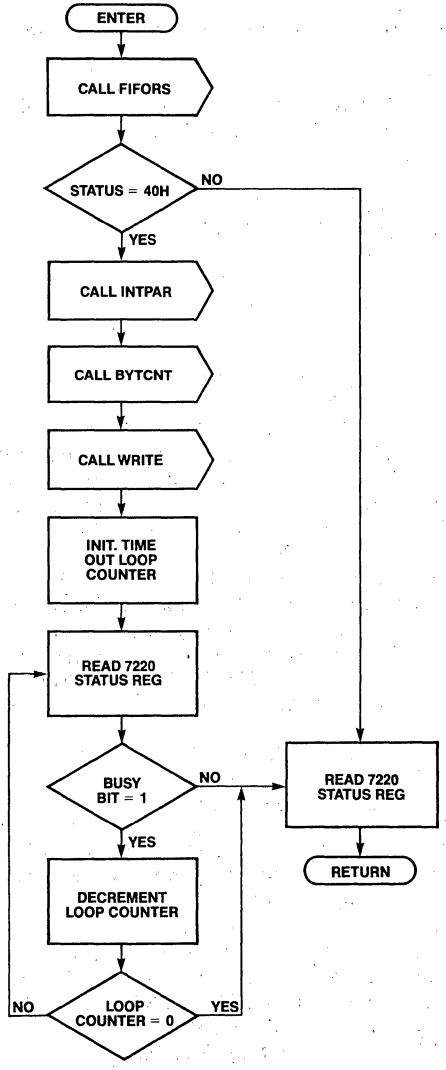


Figure 17. WRBUBL

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   13


 LOC  OBJ         LINE       SOURCE STATEMENT

                   476 ;*******************************************************************************
                   477 ;
                   478 ; FUNCTION: RDBUBL
                   479 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                   480 ;           D-E REGS, STARTING ADDRESS IN RAM
                   481 ;           BPK72 STATUS REG
                   482 ;           READ DATA FROM BUBBLE MEMORY
                   483 ; OUTPUTS:  WRITE DATA TO RAM
                   484 ;           A REG= BPK72 STATUS REG
                   485 ; CALLS:    FIFORS
                   486 ;           INTPAR
                   487 ;           BYTCNT
                   488 ;           READ
                   489 ; DESTROYS: A, F/FS
                   490 ;
                   491 ; DESCRIPTION: READ BUBBLE MEMORY DATA
                   492 ;       THE B-C REGS CONTAIN THE ADDRESS TO THE FIRST OF FIVE
                   493 ;       CONTIGUOUS MEMORY LOCATIONS IN RAM. THE DATA ADDRESSED
                   494 ;       BY THE B-C REGS IS USED TO LOAD THE PARAMETRIC REGS. THE D-E
                   495 ;       REGS CONTAIN THE STARTING ADDRESS IN RAM USED TO STORE
                   496 ;       DATA READ FROM THE BUBBLE MEMORY. GIVEN THE DATA IN RAM
                   497 ;       USED TO LOAD THE PARAMETRIC REGS, THIS FUNCTION WILL RESET
                   498 ;       THE 7220 FIFO, LOAD THE PARAMETRIC REGS, COMPUTE THE
                   499 ;       BYTE COUNTER, AND COPY THE DATA FROM THE BUBBLE MEMORY INTO
                   500 ;       RAM. RDBUBL RETURNS THE VALUE OF THE BPK72 STATUS REGISTER
                   501 ;       TO THE CALLING ROUTINE VIA THE 8085'S A REG. ONLY A STATUS
                   502 ;       OF 40H OR 48H WITH ERROR CORRECTION INDICATES A SUCCESSFUL
                   503 ;       EXECUTION OF RDBUBL.
                   504 ;
                   505         PUBLIC  RDBUBL  ; DECLARE PUBLIC FUNCTION
 0936 E5           506 RDBUBL: PUSH    H       ; SAVE H-L REGS
 0937 C5           507         PUSH    B       ; SAVE B-C REGS
 0938 0640         508         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 093A CD1308       509         CALL    FIFORS  ; CALL FIFORS, WRITE FIFO RESET COMMAND
 093D A8           510         XRA     B       ; TEST FOR STATUS= 40H, OP-COMPLETE
 093E C25C09       511         JNZ     RETRD   ; IF NOT ZERO, FIFO ERROR, JMP RETRD
 0941 C1           512         POP     B       ; RESTORE B-C REGS
 0942 CD0008       513         CALL    INTPAR  ; CALL INTPAR, LOAD PARAMETRIC REGS
 0945 CD4008       514         CALL    BYTCNT  ; CALL BYTCNT, COMPUTE BYTE COUNTER
 0948 CDA408       515         CALL    READ    ; CALL READ, READ BUBBLE DATA
 094B C5           516         PUSH    B       ; SAVE B-C REGS
                   517                         ; CONTINUED ON NEXT PAGE
                   518 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   14


 LOC  OBJ        LINE        SOURCE STATEMENT

 094C 21FFFF      519         LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 094F DBFF        520 LOOPRD: IN      PRTA01  ; READ STATUS REG
 0951 07          521         RLC             ; TEST FOR BUSY BIT= 1
 0952 D25C09      522         JNC     RETRD   ; IF ZERO, NOT BUSY, JMP RETRD
 0955 2B          523         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0956 AF          524         XRA     A       ; CLEAR A REG
 0957 B4          525         ORA     H       ; TEST H REG= 00H
 0958 B5          526         ORA     L       ; TEST L REG= 00H
 0959 C24F09      527         JNZ     LOOPRD  ; CONTINUE POLLING STATUS REG
 095C C1          528 RETRD:  POP     B       ; RESTORE B-C REGS
 095D E1          529         POP     H       ; RESTORE H-L REGS
 095E DBFF        530         IN      PRTA01  ; READ STATUS REG
 0960 C9          531         RET             ; RETURN TO CALL
                  532 ;
                  533 ;
                  534 ;
                  535 $EJECT
```

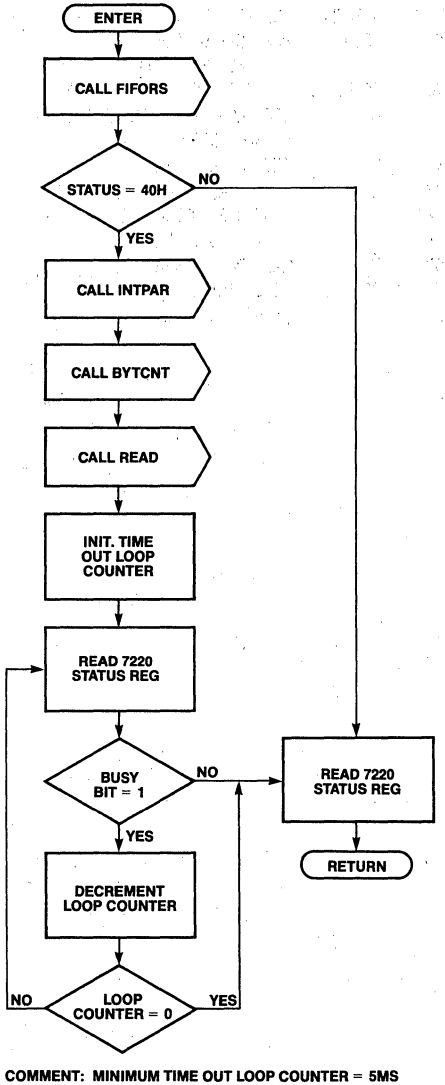


Figure 18. RDBUBL

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   15


 LOC  OBJ        LINE       SOURCE STATEMENT

                  536 ;*******************************************************************************
                  537 ;
                  538 ; FUNCTION: INBUBL
                  539 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                  540 ;           BPK72 STATUS REG
                  541 ; OUTPUTS:  A REG= BPK72 STATUS REG
                  542 ; CALLS:    ABORT
                  543 ;           INTPAR
                  544 ; DESTROYS: A, F/FS
                  545 ;
                  546 ; DESCRIPTION: INITIALIZE THE BPK72
                  547 ;       THE B-C REGS CONTAIN THE ADDRESS TO THE FIRST OF  FIVE CONTIGUOUS
                  548 ;       MEMORY LOCATIONS IN RAM.  THE  DATA ADDRESSED  BY  THE  B-C  REGS
                  549 ;       IS  USED TO  LOAD THE  PARAMETRIC REGS.  THIS FUNCTION WILL WRITE
                  550 ;       THE  PARAMETRIC  REGS FOLLOWED  BY  ISSUING  A  BUBBLE  MEMORY
                  551 ;       INITIALIZATION  COMMAND.  AFTER  ISSUING  THE COMMAND,  THE BPK72
                  552 ;       STATUS REG IS POLLED UNTIL AN OP-COMPLETE,  40H,  IS  READ OR THE
                  553 ;       TIME  OUT  LOOP  COUNTER  DECREMENTS TO ZERO.  THIS  COMMAND MUST
                  554 ;       PRECEED ALL  OTHER  COMMANDS AFTER POWERING UP THE BPK72.  INBUBL
                  555 ;       RETURNS THE VALUE OF THE BPK72 STATUS  REG TO THE CALLING ROUTINE
                  556 ;       VIA THE 8085'S A REG.  ONLY A STATUS OF 40H INDICATES A SUCCESSFUL
                  557 ;       EXECUTION OF INBUBL.
                  558 ;
                  559         PUBLIC  INBUBL  ; DECLARE PUBLIC FUNCTION
0961 D5           560 INBUBL: PUSH    D       ; SAVE D-E REGS
0962 C5           561         PUSH    B       ; SAVE B-C REGS
0963 0640         562         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
0965 CDDE08       563         CALL    ABORT   ; CALL ABORT COMMAND
0968 A8           564         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
0969 C29709       565         JNZ     RETIN   ; IF ZERO, OP-COMPLETE, CONTINUE
096C C1           566         POP     B       ; PARAMETRIC REGS STARTING ADDRESS IN REG B
096D CD0008       567         CALL    INTPAR  ; CALL INTPAR, LOAD PARAMETRIC REGS
0970 C5           568         PUSH    B       ; SAVE B-C REGS
0971 0640         569         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
0973 11FFFF       570         LXI     D,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
0976 3E11         571         MVI     A,11H   ; LOAD A REG= INITIALIZE COMMAND
0978 D3FF         572         OUT     PRTA01  ; WRITE INITIALIZE COMMAND
                  573                         ; CONTINUED ON NEXT PAGE
                  574 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   16

 LOC  OBJ        LINE        SOURCE STATEMENT

 097A DBFF        575 BUSYIN: IN      PRTA01  ; READ STATUS REG
 097C 07          576         RLC             ; TEST BUSY BIT= 1
 097D DA8A09      577         JC      POLLIN  ; IF BUSY= 1, POLL STATUS REG FOR 40H
 0980 1B          578         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0981 AF          579         XRA     A       ; CLEAR A REG
 0982 B2          580         ORA     D       ; TEST D REG= 00H
 0983 B3          581         ORA     E       ; TEST E REG= 00H
 0984 C27A09      582         JNZ     BUSYIN  ; IF NOT ZERO, CONTINUE POLLING THE INITIALIZE COMMAND
 0987 C39709      583         JMP     RETIN   ; TIME OUT ERROR, RETURN
 098A DBFF        584 POLLIN: IN      PRTA01  ; READ STATUS REG
 098C A8          585         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
 098D CA9709      586         JZ      RETIN   ; IF OP-COMPLETE, JMP RETIN
 0990 1B          587         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
 0991 AF          588         XRA     A       ; CLEAR A REG
 0992 B2          589         ORA     D       ; TEST D REG= 00H
 0993 B3          590         ORA     E       ; TEST E REG= 00H
 0994 C28A09      591         JNZ     POLLIN  ; IF NOT ZERO, CONTINUE POLLING INITIALIZE COMMAND
 0997 C1          592 RETIN:  POP     B       ; RESTORE B-C REGS
 0998 D1          593         POP     D       ; RESTORE D-E REGS
 0999 DBFF        594         IN      PRTA01  ; READ STATUS REG
 099B C9          595         RET             ; RETURN TO CALL
                  596 ;
                  597 ;
                  598 $EJECT
```

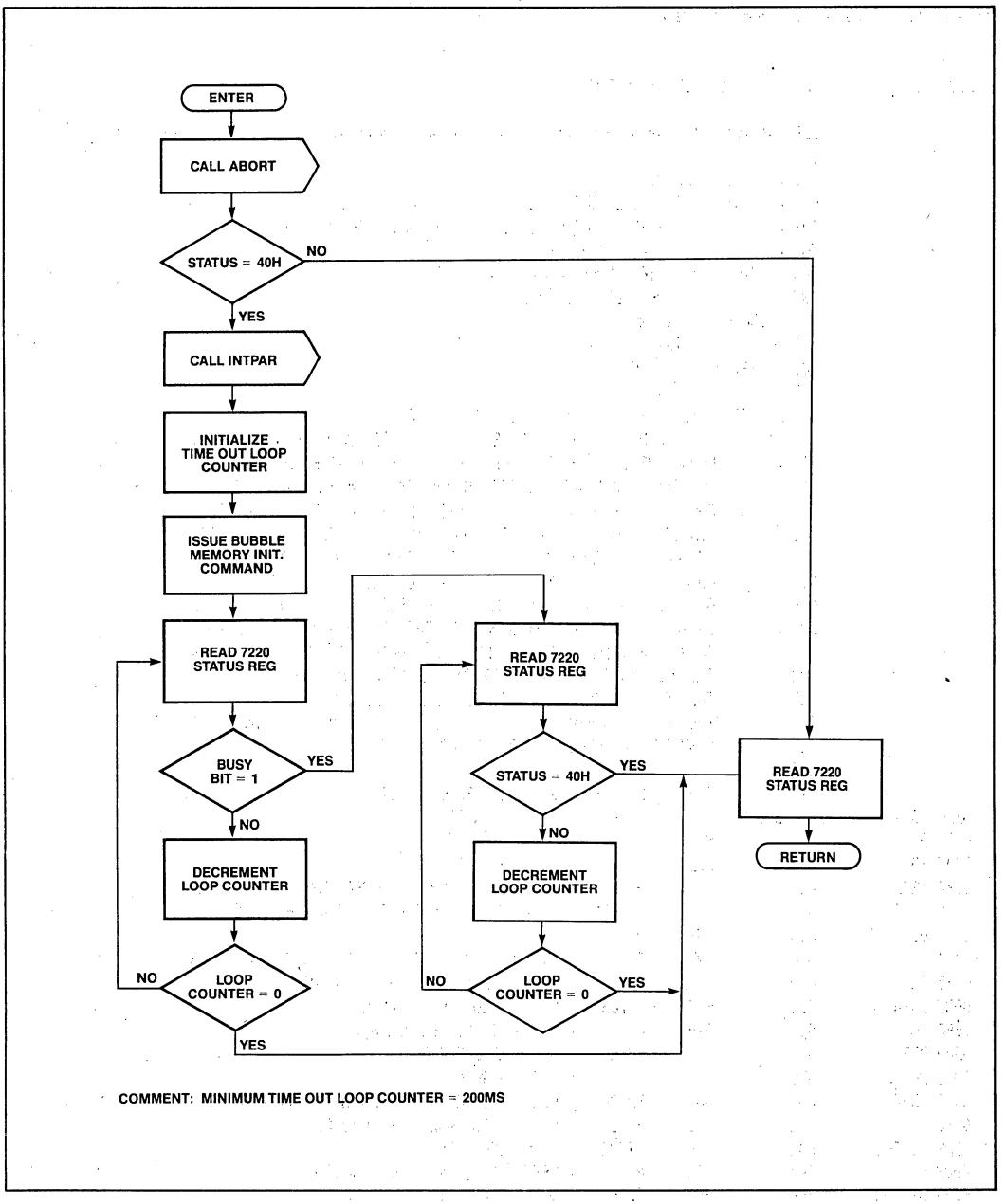


Figure 19. INBUBL

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   17


 LOC  OBJ         LINE       SOURCE STATEMENT

                   599 ;*******************************************************************************
                   600 ;
                   601 ; FUNCTION: BOOTUP
                   602 ; INPUTS:   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                   603 ;           D-E REGS, STARTING ADDRESS OF BOOT LOOP CODE IN RAM
                   604 ;           BPK72 STATUS REG
                   605 ; OUTPUTS:  WRITE BUBBLE MEMORY BOOT LOOP
                   606 ;           A REG= BPK72 STATUS REG
                   607 ; CALLS:    FIFORS
                   608 ;           INTPAR
                   609 ; DESTROYS: A, F/FS
                   610 ;
                   611 ; DESCRIPTION: WRITE BUBBLE MEMORY BOOT LOOP
                   612 ;       THIS  FUNCTION WILL  WRITE THE  BOOT LOOP CODE  INTO THE 7110
                   613 ;       BUBBLE MEMORY.  THE D-E  REGS PROVIDE THE ADDRESS TO THE FIRST
                   614 ;       OF FORTY  CONTIGUOUS BYTES  IN RAM THAT CONTAIN THE BOOT LOOP
                   615 ;       CODE.  THE B-C REGS  CONTAIN  THE ADDRESS TO THE FIRST OF FIVE
                   616 ;       CONTIGUOUS MEMORY  LOCATIONS ALSO IN RAM.  THE DATA ADDRESSED
                   617 ;       BY THE  B-C  REGS  IS  USED  TO  LOAD  THE  PARAMETRIC  REGS.
                   618 ;       NOTE   THAT  THE  PARAMETRIC  ENABLE  REG  WRITE   BOOT  LOOP
                   619 ;       BIT  IS  AUTOMATICALLY  SET  AND A  FORTY-FIRST BYTE  OF ZERO
                   620 ;       IS WRITTEN  TO THE  FIFO DATA BUFFER TO AVOID A TIMING ERROR.
                   621 ;       BEFORE  A RETURN IS EXECUTED,  THE PARAMETRIC  ENABLE  REG IS
                   622 ;       RESTORED TO ITS VALUE PRIOR TO CALLING BOOTUP.  BOOTUP RETURNS
                   623 ;       THE VALUE OF THE BPK72  STATUS REG TO THE CALLING ROUTINE VIA
                   624 ;       THE 8085'S  A REG.  ONLY  A STATUS  OF  40H OR 42H  INDICATES
                   625 ;       A SUCCESSFUL EXECUTION OF BOOTUP.
                   626 ;
                   627         PUBLIC  BOOTUP  ; DECLARE PUBLIC FUNCTION
 099C C5           628 BOOTUP: PUSH    B       ; SAVE B-C REGS
 099D D5           629         PUSH    D       ; SAVE D-E REGS
 099E E5           630         PUSH    H       ; SAVE H-L REGS
 099F 3E0D         631         MVI     A,0DH   ; LOAD A REG= 0DH, 7220 RAC ENABLE REG ADDRESS
 09A1 D3FF         632         OUT     PRTA01  ; WRITE 7220 RAC WITH ENABLE REG ADDRESS
 09A3 03           633         INX     B       ;
 09A4 03           634         INX     B       ; INCREMENT B-C REGS TO ENABLE REG RAM ADDRESS
 09A5 0A           635         LDAX    B       ; LOAD A REG= ENABLE REG FROM RAM
 09A6 0610         636         MVI     B,10H   ; LOAD B REG= BOOT LOOP ENABLE MASK
 09A8 B0           637         ORA     B       ; SET BOOT LOOP ENABLE BIT
 09A9 D3FE         638         OUT     PRTA00  ; WRITE ENABLE REG
 09AB AF           639         XRA     A       ; CLEAR A REG
 09AC D3FF         640         OUT     PRTA01  ; LOAD 7220 RAC WITH FIFO DATA BUFFER ADDRESS
 09AE 0640         641         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 09B0 CD1308       642         CALL    FIFORS  ; CALL FIFO RESET
 09B3 A8           643         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
 09B4 C2230A       644         JNZ     RETBT   ; IF NOT ZERO, ERROR, JMP RETBT
                   645                         ; CONTINUED ON NEXT PAGE
                   646 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   18

 LOC  OBJ        LINE        SOURCE STATEMENT

 09B7 0E28        647         MVI     C,28H   ; LOAD C REG= 28H, BYTE COUNTER= 40 DECIMAL
 09B9 3EFF        648         MVI     A,0FFH  ; LOAD A REG= FFH
 09BB D3FE        649 ALLFFS: OUT     PRTA00  ; WRITE A REG INTO FIFO DATA BUFFER
 09BD 0D          650         DCR     C       ; DECREMENT BYTE COUNTER
 09BE C2BB09      651         JNZ     ALLFFS  ; IF BYTE COUNTER= ZERO, CONTINUE
 09C1 21FFFF      652         LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 09C4 3E16        653         MVI     A,16H   ; LOAD A REG= WRITE BOOT LOOP REG COMMAND
 09C6 D3FF        654         OUT     PRTA01  ; WRITE, WRITE BOOT LOOP REG COMMAND
 09C8 DBFF        655 BUSYB:  IN      PRTA01  ; READ STATUS REG
 09CA 07          656         RLC             ; TEST BUSY BIT= 1
 09CB DAD809      657         JC      POLLBR  ; IF BUSY= 1, POLL STATUS REG FOR 40H
 09CE 2B          658         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 09CF AF          659         XRA     A       ; CLEAR A REG
 09D0 B4          660         ORA     H       ; TEST H REG= 00H
 09D1 B5          661         ORA     L       ; TEST L REG= 00H
 09D2 C2C809      662         JNZ     BUSYB   ; IF NOT ZERO, CONTINUE POLLING WRBLRS COMMAND
 09D5 C3230A      663         JMP     RETBT   ; TIME OUT ERROR, RETURN
 09D8 DBFF        664 POLLBR: IN      PRTA01  ; READ STATUS REG
 09DA A8          665         XRA     B       ; TEST STATUS= 40H
 09DB CAE809      666         JZ      CONT    ; IF ZERO, CONTINUE, OP-COMPLETE
 09DE 2B          667         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 09DF AF          668         XRA     A       ; CLEAR A REG
 09E0 B4          669         ORA     H       ; TEST H REG= 00H
 09E1 B5          670         ORA     L       ; TEST L REG= 00H
 09E2 CA230A      671         JZ      RETBT   ; IF ZERO, TIME OUT, ERROR
 09E5 C3D809      672         JMP     POLLBR  ; CONTINUE POLLING WRBLRS COMMAND
                  673                         ; CONTINUED ON NEXT PAGE
                  674 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   19


 LOC  OBJ        LINE        SOURCE STATEMENT

 09E8 CD1308      675 CONT:   CALL    FIFORS  ; CALL FIFO RESET
 09EB A8          676         XRA     B       ; TEST STATUS= 40H
 09EC C2230A      677         JNZ     RETBT   ; IF NOT ZERO, ERROR, JMP RETBT
 09EF 0E28        678         MVI     C,28H   ; LOAD C REG= 28H, BYTE COUNTER= 40 DECIMAL
 09F1 1A          679 BLCODE: LDAX    D       ; LOAD A REG FROM D REG ADDRESS
 09F2 13          680         INX     D       ; INCREMENT D REG TO THE NEXT ADDRESS
 09F3 D3FE        681         OUT     PRTA00  ; WRITE BOOT LOOP CODE INTO FIFO DATA BUFFER
 09F5 0D          682         DCR     C       ; DECREMENT BYTE COUNTER
 09F6 C2F109      683         JNZ     BLCODE  ; IF NOT ZERO, JMP BLCODE
 09F9 AF          684         XRA     A       ; CLEAR A REG
 09FA D3FE        685         OUT     PRTA00  ; WRITE 41ST BYTE OF ZERO INTO FIFO DATA BUFFER
 09FC 21FFFF      686         LXI     H,0FFFFH; LOAD TIME OUT LOOP COUNTER
 09FF 0EFD        687         MVI     C,0FDH  ; MASK, MASK OUT PARITY BIT
 0A01 3E17        688         MVI     A,17H   ; LOAD A REG= WRITE BOOT LOOP COMMAND
 0A03 D3FF        689         OUT     PRTA01  ; WRITE, WRITE BOOT LOOP COMMAND
 0A05 DBFF        690 BUSYBL: IN      PRTA01  ; READ STATUS REG
 0A07 07          691         RLC             ; TEST BUSY BIT= 1
 0A08 DA150A      692         JC      POLLBL  ; IF BUSY=1, POLL STATUS REG FOR OP-COMPLETE
 0A0B 2B          693         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0A0C AF          694         XRA     A       ; CLEAR A REG
 0A0D B4          695         ORA     H       ; TEST H REG= 00H
 0A0E B5          696         ORA     L       ; TEST L REG= 00H
 0A0F C2050A      697         JNZ     BUSYBL  ; IF NOT ZERO, CONTINUE POLLING THE WRBL COMMAND
 0A12 C3230A      698         JMP     RETBT   ; TIME OUT ERROR, RETURN
 0A15 DBFF        699 POLLBL: IN      PRTA01  ; READ STATUS REG
 0A17 A1          700         ANA     C       ; RESET BIT 1, PARITY BIT
 0A18 A8          701         XRA     B       ; TEST STATUS= 40H OR 42H, OP-COMPLETE
 0A19 CA230A      702         JZ      RETBT   ; IF ZERO, CONTINUE, OP-COMPLETE
 0A1C 2B          703         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0A1D AF          704         XRA     A       ; CLEAR A REG
 0A1E B4          705         ORA     H       ; TEST H REG= 00H
 0A1F B5          706         ORA     L       ; TEST L REG= 00H
 0A20 C2150A      707         JNZ     POLLBL  ; CONTINUE POLLING WRITE BOOT LOOP COMMAND
 0A23 E1          708 RETBT:  POP     H       ; RESTORE H-L REGS
 0A24 D1          709         POP     D       ; RESTORE D-E REGS
 0A25 C1          710         POP     B       ; RESTORE B-C REGS
 0A26 CD0008      711         CALL    INTPAR  ; CALL INTPAR, LOAD THE PARAMETRIC REGS
 0A29 DBFF        712         IN      PRTA01  ; READ STATUS REG
 0A2B C9          713         RET             ;
                  714 ;
                  715 ;
                  716 ;
                  717 $EJECT
```

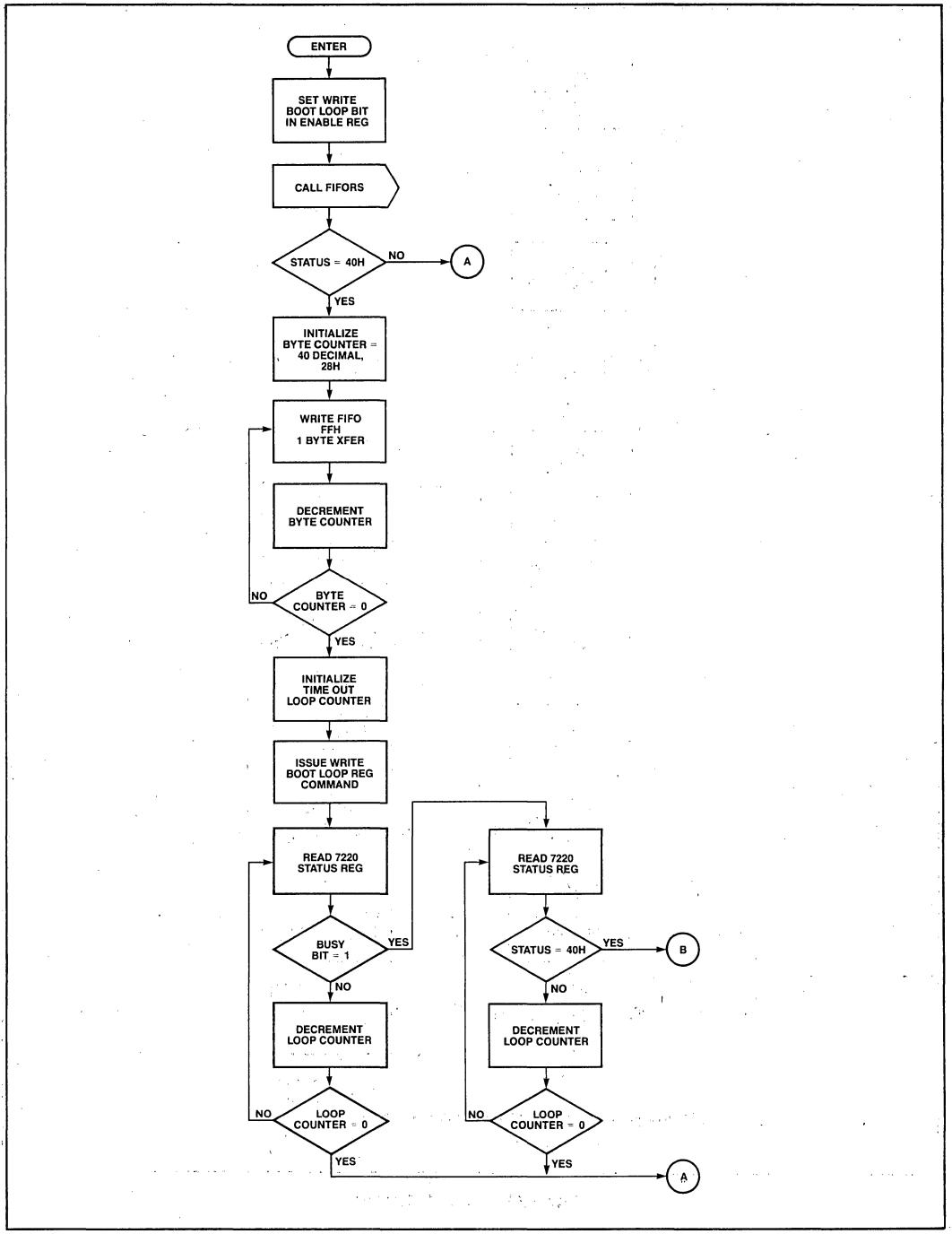


Figure 20. BOOTUP

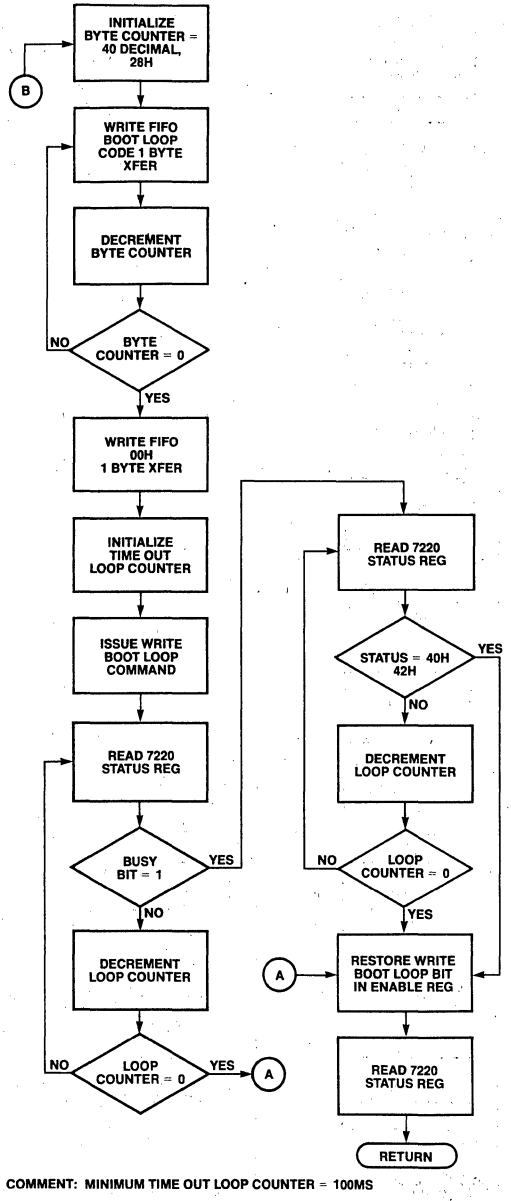


Figure 20 (Con't). BOOTUP

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   20


 LOC  OBJ        LINE       SOURCE STATEMENT

                  718 ;*******************************************************************************
                  719 ;
                  720 ; FUNCTION: RDBOOT
                  721 ; INPUTS:   D-E REGS, STARTING ADDRESS IN RAM
                  722 ;           BPK72 STATUS REG
                  723 ;           READ BUBBLE MEMORY BOOT LOOP
                  724 ; OUTPUTS:  COPY BUBBLE MEMORY BOOT LOOP TO RAM
                  725 ;           A REG= BPK72 STATUS REG
                  726 ; CALLS:    FIFORS
                  727 ; DESTROYS: A, F/FS
                  728 ;
                  729 ; DESCRIPTION: READ BUBBLE MEMORY BOOT LOOP
                  730 ;       THE  D-E REGS CONTAIN THE STARTING  ADDRESS  TO THE  FIRST OF 40
                  731 ;       CONTIGUOUS  MEMORY  LOCATIONS  IN  RAM  THAT WILL BE LOADED WITH
                  732 ;       A COPY OF THE BOOT LOOP CODE.  RDBOOT  RETURNS  THE VALUE OF THE
                  733 ;       BPK72  STATUS  REG TO  THE CALLING ROUTINE VIA THE 8085'S A REG.
                  734 ;       ONLY A STATUS OF 40H INDICATES A SUCCESSFUL EXECUTION OF RDBOOT.
                  735 ;
                  736         PUBLIC  RDBOOT  ; DECLARE PUBLIC FUNCTION
 0A2C C5          737 RDBOOT: PUSH    B       ; SAVE B-C REGS
 0A2D D5          738         PUSH    D       ; SAVE D-E REGS
 0A2E E5          739         PUSH    H       ; SAVE H-L REGS
 0A2F 0640        740         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 0A31 0E28        741         MVI     C,28H   ; LOAD C REG= 28H, BYTE COUNTER= 40 DECIMAL
 0A33 CD1308      742         CALL    FIFORS  ; CALL FIFO RESET
 0A36 A8          743         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
 0A37 C26A0A      744         JNZ     RETRDB  ; IF NOT ZERO, ERROR, JMP RETRDB
 0A3A 04          745         INR     B       ; B REG= 41H, OP-COMPLETE, FIFO FULL
 0A3B 21FFFF      746         LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
 0A3E 3E1B        747         MVI     A,1BH   ; LOAD A REG= READ BOOT LOOP COMMAND
 0A40 D3FF        748         OUT     PRTA01  ; WRITE, READ BOOT LOOP COMMAND
 0A42 DBFF        749 BUSYRB: IN      PRTA01  ; READ STATUS REG
 0A44 07          750         RLC             ; TEST BUSY BIT= 1
 0A45 DA520A      751         JC      BTLPRD  ; IF BUSY= 1, POLL STATUS REG FOR 41H
 0A48 2B          752         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0A49 AF          753         XRA     A       ; CLEAR A REG
 0A4A B4          754         ORA     H       ; TEST H REG= 00H
 0A4B B5          755         ORA     L       ; TEST L REG= 00H
 0A4C C2420A      756         JNZ     BUSYRB  ; IF NOT ZERO, CONTINUE POLLING RDBL COMMAND
 0A4F C36A0A      757         JMP     RETRDB  ; TIME OUT ERROR, RETURN
                  758                         ; CONTINUED ON NEXT PAGE
                  759 $EJECT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72   PAGE  21


 LOC  OBJ        LINE       SOURCE STATEMENT

 0A52 DBFF        760 BTLPRD: IN      PRTA01  ; READ STATUS REG
 0A54 A8          761         XRA     B       ; TEST STATUS= 41H, OP-COMPLETE, FIFO FULL
 0A55 CA620A      762         JZ      FIFORD  ; IF ZERO, JMP TO FIFO READ
 0A58 2B          763         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
 0A59 AF          764         XRA     A       ; CLEAR A REG
 0A5A B4          765         ORA     H       ; TEST H REG= 00H
 0A5B B5          766         ORA     L       ; TEST L REG= 00H
 0A5C CA6A0A      767         JZ      RETRDB  ; IF ZERO, TIME OUT, ERROR
 0A5F C3520A      768         JMP     BTLPRD  ; CONTINUE POLLING RDBL COMMAND
 0A62 DBFE        769 FIFORD: IN      PRTA00  ; READ FIFO DATA BUFFER
 0A64 12          770         STAX    D       ; WRITE RAM AT ADDRESS IN D REG
 0A65 13          771         INX     D       ; INCREMENT D REG TO NEXT RAM ADDRESS
 0A66 0D          772         DCR     C       ; DECREMENT BYTE COUNTER
 0A67 C2620A      773         JNZ     FIFORD  ; IF NOT ZERO, JMP FIFO READ
 0A6A DBFF        774 RETRDB: IN      PRTA01  ; READ STATUS REG
 0A6C E1          775         POP     H       ; RESTORE H-L REGS
 0A6D D1          776         POP     D       ; RESTORE D-E REGS
 0A6E C1          777         POP     B       ; RESTORE B-C REGS
 0A6F C9          778         RET             ; RETURN TO CALL
                  779 ;
                  780 $EJECT
```

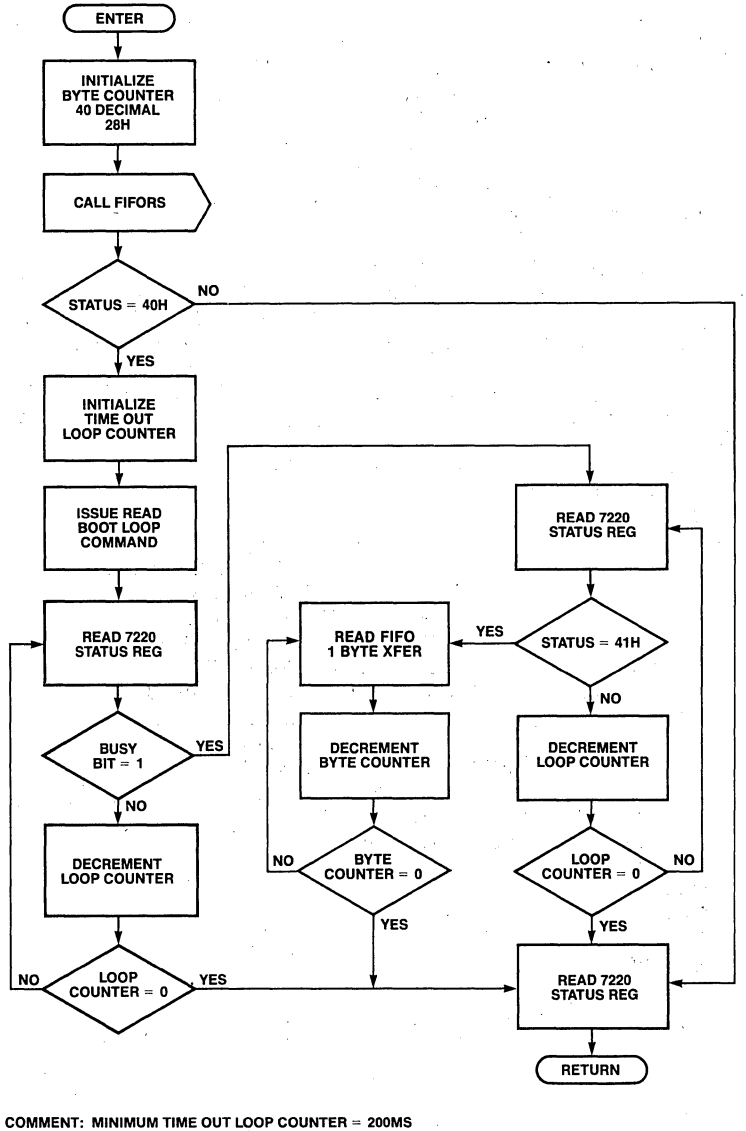


Figure 21. RDBOOT

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   22


 LOC  OBJ        LINE       SOURCE STATEMENT

                  781 ;*************************************************************************
                  782 ;
                  783 ; FUNCTION: WRFIFO
                  784 ; INPUTS:   D-E REGS, STARTING ADDRESS OF DATA IN RAM
                  785 ;           BPK72 STATUS REG
                  786 ; OUTPUTS:  WRITE 40 BYTES IN THE BPK72 FIFO DATA BUFFER
                  787 ;           A REG= BPK72 STATUS REG
                  788 ; CALLS:    FIFORS
                  789 ; DESTROYS: A, F/FS
                  790 ;
                  791 ; DESCRIPTION: WRITE 7220 FIFO DATA BUFFER
                  792 ;       THE  D-E REGS PROVIDE THE ADDRESS TO THE FIRST OF 40 CONTIGUOUS
                  793 ;       BYTES IN RAM THAT CONTAIN DATA TO BE LOADED INTO THE BPK72 FIFO
                  794 ;       DATA BUFFER.  WRFIFO WILL TRANSFER THE DATA FROM RAM TO THE FIFO
                  795 ;       DATA BUFFER.  WRFIFO RETURNS  THE VALUE OF THE BPK72 STATUS REG
                  796 ;       TO THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY  A STATUS OF
                  797 ;       41H OR 43H INDICATES A SUCCESSFUL EXECUTION OF WRFIFO.
                  798 ;
                  799         PUBLIC  WRFIFO  ; DECLARE PUBLIC FUNCTION
 0A70 C5          800 WRFIFO: PUSH    B       ; SAVE B-C REGS
 0A71 D5          801         PUSH    D       ; SAVE D-E REGS
 0A72 0640        802         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
 0A74 0E28        803         MVI     C,28H   ; LOAD C REG= 28H, INITIALIZE LOOP COUNTER
 0A76 CD1308      804         CALL    FIFORS  ; CALL FIFORS, WRITE FIFO RESET COMMAND
 0A79 A8          805         XRA     B       ; TEST FOR STATUS REG= 40H, OP-COMPLETE
 0A7A C2850A      806         JNZ     RETWF   ; IF NOT ZERO, FIFO ERROR, JMP RETWF
 0A7D 1A          807 INFIFO: LDAX    D       ; LOAD A REG FROM D-E REG ADDRESS
 0A7E D3FE        808         OUT     PRTA00  ; WRITE A REG TO 7220 FIFO DATA BUFFER
 0A80 13          809         INX     D       ; INCREMENT D-E REGS TO NEXT ADDRESS IN RAM
 0A81 0D          810         DCR     C       ; DECREMENT LOOP COUNTER
 0A82 C27D0A      811         JNZ     INFIFO  ; IF LOOP COUNTER NOT ZERO, JMP INFIFO
 0A85 D1          812 RETWF:  POP     D       ; RESTORE D-E REGS
 0A86 C1          813         POP     B       ; RESTORE B-C REGS
 0A87 DBFF        814         IN      PRTA01  ; READ STATUS REG
 0A89 C9          815         RET             ; RETURN TO CALL
                  816 ;
                  817 ;
                  818 ;
                  819 $EJECT
```

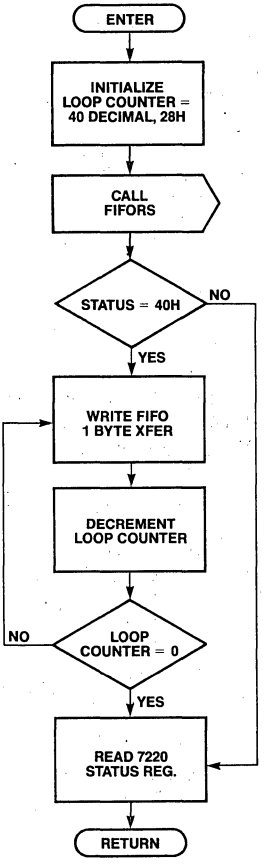


Figure 22. WRFIFO

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72   PAGE   23


 LOC  OBJ        LINE       SOURCE STATEMENT

                  820 ;*************************************************************************
                  821 ;
                  822 ; FUNCTION: RDFIFO
                  823 ; INPUTS:   D-E REGS STARTING ADDRESS IN RAM
                  824 ;           BPK72 STATUS REG
                  825 ;           READ 40 BYTES OF DATA FROM BPK72 FIFO DATA BUFFER
                  826 ; OUTPUTS:  TRANSFER FIFO DATA BUFFER TO RAM
                  827 ;           A REG= BPK72 STATUS REG
                  828 ; CALLS:    NONE
                  829 ; DESTROYS: A, F/FS
                  830 ;
                  831 ; DESCRIPTION: READ 7220 FIFO DATA BUFFER
                  832 ;       THE  D-E  REGS  CONTAIN THE ADDRESS TO THE  FIRST OF 40 CONTIGUOUS
                  833 ;       BYTES IN RAM THAT  WILL BE LOADED  WITH THE CONTENTS OF THE  BPK72
                  834 ;       FIFO DATA BUFFER. RDFIFO WILL TRANSFER THE DATA FROM THE FIFO DATA
                  835 ;       BUFFER TO RAM.  RDFIFO RETURNS THE VALUE OF  THE BPK72 STATUS  REG
                  836 ;       TO THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY A STATUS OF 40H
                  837 ;       OR 42H INDICATES A SUCCESSFUL EXECUTION OF RDFIFO.
                  838 ;
                  839         PUBLIC  RDFIFO  ; DECLARE PUBLIC FUNCTION
  0A8A C5         840 RDFIFO: PUSH    B       ; SAVE B-C REGS
  0A8B D5         841         PUSH    D       ; SAVE D-E REGS
  0A8C 0E28       842         MVI     C,28H   ; LOAD C REG= 28H, INITIALIZE LOOP COUNTER
  0A8E DBFE       843 OUTFIF: IN      PRTA00  ; LOAD A REG WITH ONE BYTE FROM FIFO DATA BUFFER
  0A90 12         844         STAX    D       ; LOAD A REG IN D-E REG ADDRESS
  0A91 13         845         INX     D       ; INCREMENT D-E REGS TO NEXT ADDRESS
  0A92 0D         846         DCR     C       ; DECREMENT LOOP COUNTER
  0A93 C28E0A     847         JNZ     OUTFIF  ; IF LOOP COUNTER NOT ZERO, JMP OUTFIF
  0A96 D1         848         POP     D       ; RESTORE D-E REGS
  0A97 C1         849         POP     B       ; RESTORE B-C REGS
  0A98 DBFF       850         IN      PRTA01  ; READ STATUS REG
  0A9A C9         851         RET             ; RETURN TO CALL
                  852 ;
                  853 ;
                  854 $EJECT
```

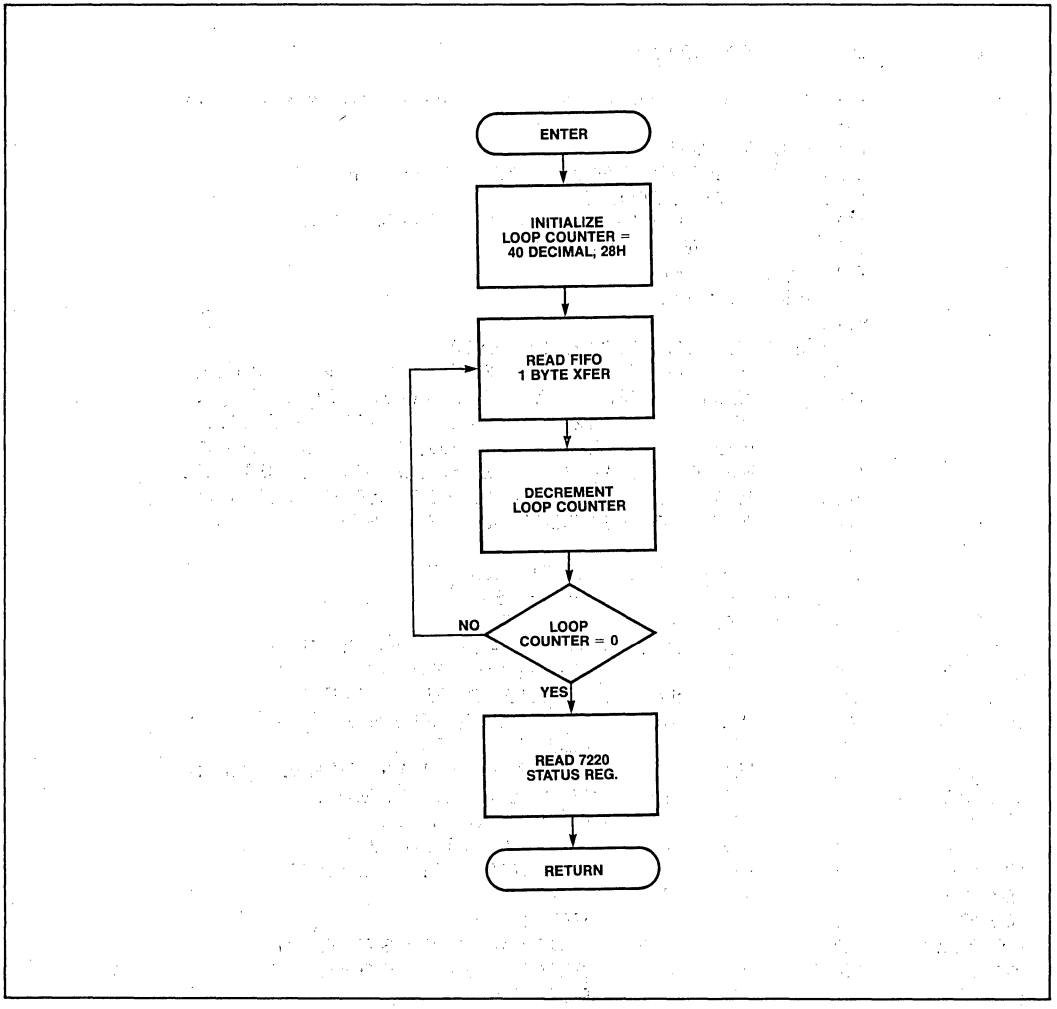
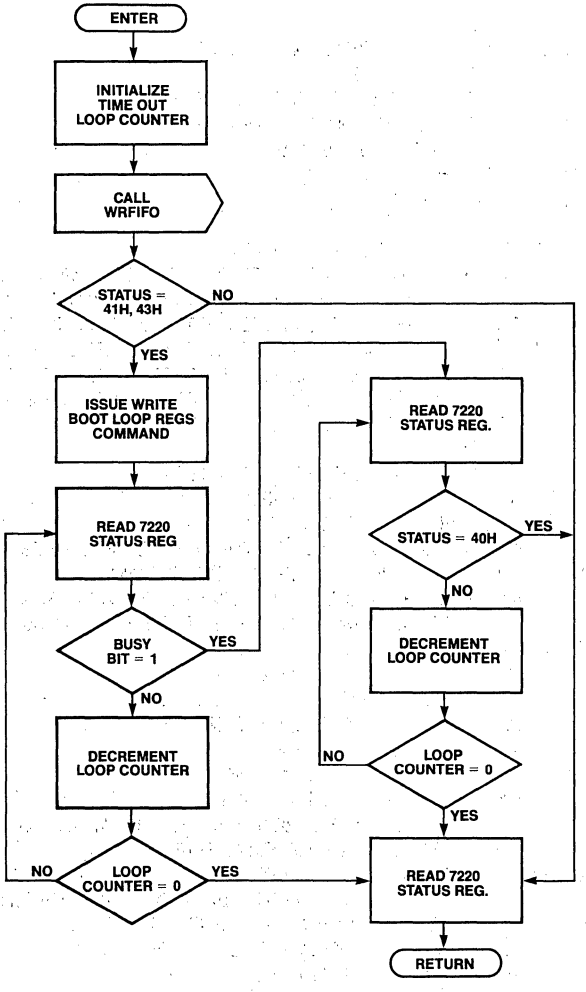


Figure 23. RDFIFO

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   24


 LOC  OBJ        LINE       SOURCE STATEMENT

                  855 ;*************************************************************************
                  856 ;
                  857 ; FUNCTION: WRBLRS
                  858 ; INPUTS:   D-E REGS, STARTING ADDRESS OF DATA IN RAM
                  859 ;           BPK72 STATUS REG
                  860 ; OUTPUTS:  WRITE BUBBLE MEMORY BOOT LOOP REGISTERS COMMAND
                  861 ;           A REG= BPK72 STATUS REG
                  862 ; CALLS:    WRFIFO
                  863 ; DESTROYS: A, F/FS
                  864 ;
                  865 ; DESCRIPTION: WRITE 7242 BOOT LOOP REGISTERS
                  866 ;       THE D-E REGS PROVIDE THE ADDRESS TO THE FIRST OF 40 CONTIGUOUS
                  867 ;       MEMORY  LOCATIONS  IN RAM THAT  CONTAIN DATA TO BE LOADED INTO
                  868 ;       THE 7242,  FORMATTER  SENSE  AMPLIFIER,  BOOT  LOOP REGISTERS.
                  869 ;       WRBLRS WILL  TRANSFER  THE DATA  FROM  RAM  TO  THE  BOOT LOOP
                  870 ;       REGISTERS.  WRBLRS  RETURNS THE VALUE OF THE BPK72  STATUS REG
                  871 ;       TO THE CALLING  ROUTINE VIA THE 8085'S A REG.  ONLY A STATUS OF
                  872 ;       40H INDICATES A SUCCESSFUL EXECUTION OF WRBLRS.
                  873 ;
                  874         PUBLIC  WRBLRS  ; DECLARE PUBLIC FUNCTION
  0A9B C5         875 WRBLRS: PUSH    B       ; SAVE B-C REGS
  0A9C E5         876         PUSH    H       ; SAVE H-L REGS
  0A9D 0641       877         MVI     B,41H   ; LOAD B REG= 41H, OP-COMPLETE, FIFO FULL
  0A9F 0EFD       878         MVI     C,0FDH  ; MASK, MASK OUT PARITY BIT
  0AA1 21FFFF     879         LXI     H,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
  0AA4 CD700A     880         CALL    WRFIFO  ; CALL WRITE FIFO DATA BUFFER
  0AA7 A1         881         ANA     C       ; RESET BIT 1, PARITY BIT
  0AA8 A8         882         XRA     B       ; TEST STATUS= 41H OR 43H, OP-COMPLETE, FIFO FULL
  0AA9 C2CE0A     883         JNZ     RETWBL  ; IF NOT ZERO, ERROR, JMP RETWBL
  0AAC 05         884         DCR     B       ; B REG= 40H, OP-COMPLETE
  0AAD 3E16       885         MVI     A,16H   ; LOAD A REG= WRITE BOOT LOOP REG COMMAND
  0AAF D3FF       886         OUT     PRTA01  ; WRITE, WRITE BOOT LOOP REG COMMAND
  0AB1 DBFF       887 BSYWBL: IN      PRTA01  ; READ STATUS REG
  0AB3 07         888         RLC             ; TEST BUSY BIT= 1
  0AB4 DAC10A     889         JC      POLWBL  ; IF BUSY= 1, POLL STATUS REG FOR 40H
  0AB7 2B         890         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
  0AB8 AF         891         XRA     A       ; CLEAR A REG
  0AB9 B4         892         ORA     H       ; TEST H REG= 00H
  0ABA B5         893         ORA     L       ; TEST L REG= 00H
  0ABB C2B10A     894         JNZ     BSYWBL  ; IF NOT ZERO, CONTINUE POLLING WRBLR COMMAND
  0ABE C3CE0A     895         JMP     RETWBL  ; TIME OUT ERROR, RETURN
  0AC1 DBFF       896 POLWBL: IN      PRTA01  ; READ STATUS REG
  0AC3 A8         897         XRA     B       ; TEST STATUS REG= 40H, OP-COMPLETE
  0AC4 CACE0A     898         JZ      RETWBL  ; IF ZERO, OP-COMPLETE, JMP RETWBL
  0AC7 2B         899         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
  0AC8 AF         900         XRA     A       ; CLEAR A REG
  0AC9 B4         901         ORA     H       ; TEST H REG= 00H
  0ACA B5         902         ORA     L       ; TEST L REG= 00H
  0ACB C2C10A     903         JNZ     POLWBL  ; IF NOT ZERO, CONTINUE POLLING WRBLR COMMAND
  0ACE E1         904 RETWBL: POP     H       ; RESTORE H-L REGS
  0ACF C1         905         POP     B       ; RESTORE B-C REGS
  0AD0 DBFF       906         IN      PRTA01  ; READ STATUS REG
  0AD2 C9         907         RET             ; RETURN TO CALL
                  908 $EJECT
```



COMMENT: MIN TIME OUT LOOP COUNTER = 1MS

Figure 24. WRBLRS

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   25


 LOC  OBJ        LINE       SOURCE STATEMENT

                  909 ;*****************************************************************************
                  910 ;
                  911 ; FUNCTION: RDBLRS
                  912 ; INPUTS:   D-E REGS, STARTING ADDRESS IN RAM
                  913 ;           BPK72 STATUS REG
                  914 ;           READ DATA FROM 7242 BOOT LOOP REGISTERS
                  915 ; OUTPUTS:  TRANSFER BOOT LOOP REGISTER DATA TO RAM
                  916 ;           A REG= BPK72 STATUS REG
                  917 ; CALLS:    RDFIFO
                  918 ; DESTROYS: A, F/FS
                  919 ;
                  920 ; DESCRIPTION: READ 7242 BOOT LOOP REGISTERS
                  921 ;       THE D-E REGS CONTAIN THE ADDRESS TO THE FIRST OF 40 CONTIGUOUS
                  922 ;       MEMORY LOCATIONS IN RAM TO BE  LOADED WITH THE CONTENTS OF THE
                  923 ;       7242, FORMATTER  SENSE AMPLIFIER, BOOT LOOP REGISTERS.  RDBLRS
                  924 ;       WILL COPY  THE CONTENTS  OF THE  BOOT  LOOP  REGISTERS TO RAM.
                  925 ;       RDBLRS  RETURNS THE  VALUE OF  THE  BPK72  STATUS  REG  TO THE
                  926 ;       CALLING ROUTINE  VIA THE 8085'S  A REG.  ONLY A  STATUS OF 40H
                  927 ;       INDICATES A SUCCESSFUL EXECUTION OF RDBLRS.
                  928 ;
                  929         PUBLIC  RDBLRS  ; DECLARE PUBLIC FUNCTION
  0AD3 C5         930 RDBLRS: PUSH    B       ; SAVE B-C REGS
  0AD4 E5         931         PUSH    H       ; SAVE H-L REGS
  0AD5 06C1       932         MVI     B,0C1H  ; LOAD B REG= C1H, OP-COMPLETE, FIFO FULL >22 BYTES (BUSY BIT=1)
  0AD7 21FFFF     933         LXI     H,0FFFFH; INTIALIZE TIME OUT LOOP COUNTER
  0ADA 3E15       934         MVI     A,15H   ; LOAD A REG= READ BOOT LOOP REGS COMMAND
  0ADC D3FF       935         OUT     PRTA01  ; WRITE THE READ BOOT LOOP REGS COMMAND
  0ADE DBFF       936 BSYRBL: IN      PRTA01  ; READ STATUS REG
  0AE0 07         937         RLC             ; TEST BUSY BIT= 1
  0AE1 DAEE0A     938         JC      POLRBL  ; IF BUSY= 1, POLL STATUS REG FOR C1H
  0AE4 2B         939         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
  0AE5 AF         940         XRA     A       ; CLEAR A REG
  0AE6 B4         941         ORA     H       ; TEST H REG= 00H
  0AE7 B5         942         ORA     L       ; TEST L REG= 00H
  0AE8 C2DE0A     943         JNZ     BSYRBL  ; IF NOT ZERO, CONTINUE POLLING READ BOOT LOOP REG COMMAND
  0AEB C3010B     944         JMP     RETRBL  ; TIME OUT ERROR, RETURN
  0AEE DBFF       945 POLRBL: IN      PRTA01  ; READ STATUS REG
  0AF0 A8         946         XRA     B       ; TEST STATUS= C1H, OP-COMPLETE, FIFO FULL
  0AF1 CAFE0A     947         JZ      CALLRD  ; IF ZERO, OP-COMPLETE, JMP CALLRD
  0AF4 2B         948         DCX     H       ; DECREMENT TIME OUT LOOP COUNTER
  0AF5 AF         949         XRA     A       ; CLEAR A REG
  0AF6 B4         950         ORA     H       ; TEST H REG= 00H
  0AF7 B5         951         ORA     L       ; TEST L REG= 00H
  0AF8 CA010B     952         JZ      RETRBL  ; IF ZERO, ERROR, JMP RETRBL
  0AFB C3EE0A     953         JMP     POLRBL  ; CONTINUE POLLING READ BOOT LOOP REG COMMAND
  0AFE CD8A0A     954 CALLRD: CALL    RDFIFO  ; CALL READ FIFO
  0B01 E1         955 RETRBL: POP     H       ; RESTORE H-L REGS
  0B02 C1         956         POP     B       ; RESTORE B-C REGS
  0B03 DBFF       957         IN      PRTA01  ; READ STATUS REG
  0B05 C9         958         RET             ; RETURN TO CALL
                  959 $EJECT
```

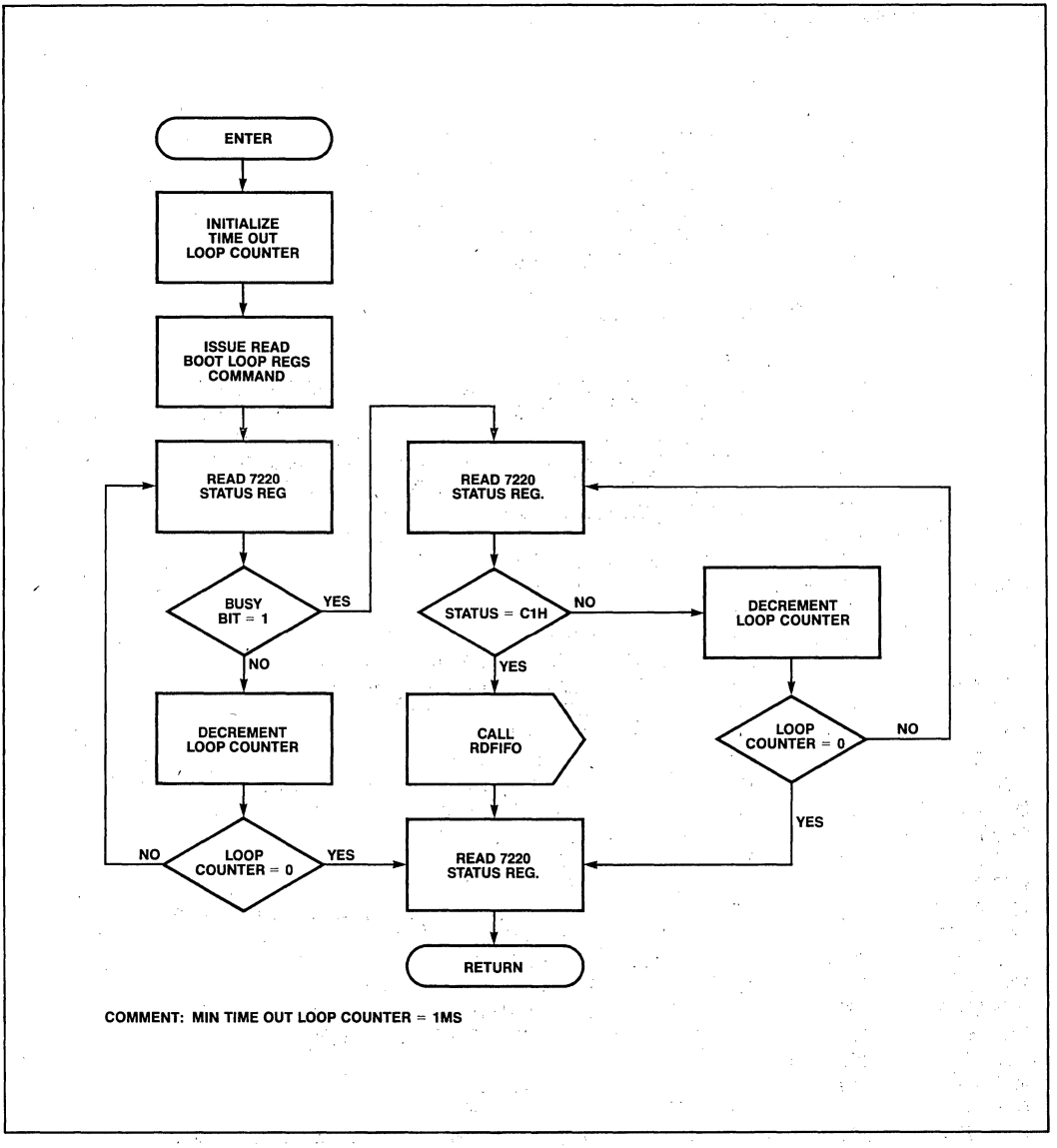
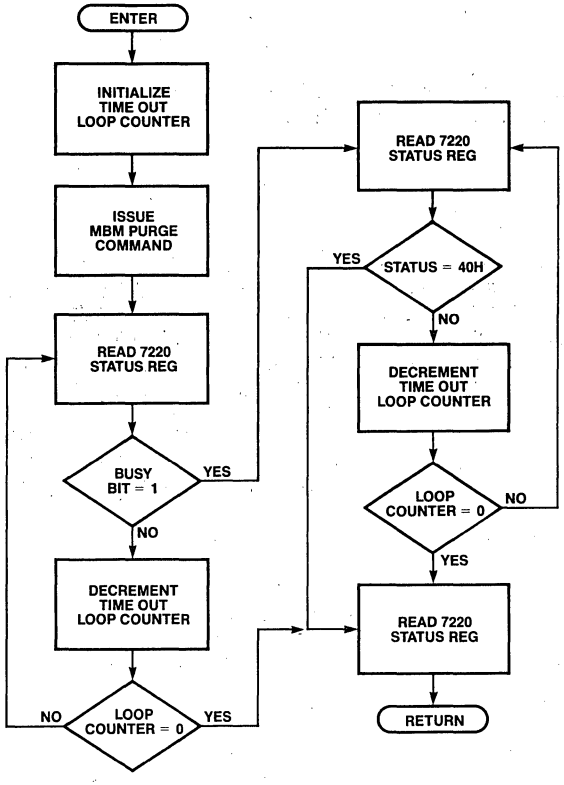


Figure 25. RDBLRS

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   26


 LOC  OBJ        LINE       SOURCE STATEMENT

                  960 ;*************************************************************************
                  961 ;
                  962 ; FUNCTION: MBMPRG
                  963 ; INPUTS:   BPK72 STATUS REG
                  964 ; OUTPUTS:  ISSUE MBM PURGE COMMAND
                  965 ;          A REG= BPK72 STATUS REG
                  966 ; CALLS:    NONE
                  967 ; DESTROYS: A, F/FS
                  968 ;
                  969 ; DESCRIPTION: MBM PURGE COMMAND
                  970 ;       AN MBM PURGE COMMAND IS ISSUED TO THE BPK72.  AFTER ISSUING THE
                  971 ;       COMMAND,  THE BPK72  STATUS REG IS POLLED UNTIL AN OP-COMPLETE
                  972 ;       40H,  HAS BEEN  READ OR THE  TIME OUT LOOP  COUNTER DECREMENTS
                  973 ;       TO ZERO.  MBMPRG  RETURNS THE VALUE OF THE BPK72 STATUS REG TO
                  974 ;       THE CALLING ROUTINE VIA THE 8085'S A REG.  ONLY A STATUS OF 40H
                  975 ;       INDICATES A SUCCESSFUL EXECUTION OF MBMPRG.
                  976 ;
                  977         PUBLIC  MBMPRG  ; DECLARE PUBLIC FUNCTION
  0B06 D5         978 MBMPRG: PUSH    D       ; SAVE D-E REGS
  0B07 C5         979         PUSH    B       ; SAVE B-C REGS
  0B08 0640       980         MVI     B,40H   ; LOAD B REG= 40H, OP-COMPLETE
  0B0A 11FFFF     981         LXI     D,0FFFFH; INITIALIZE TIME OUT LOOP COUNTER
  0B0D 3E1E       982         MVI     A,1EH   ; LOAD A REG= MBM PURGE COMMAND
  0B0F D3FF       983         OUT     PRTA01  ; WRITE MBM PURGE COMMAND
  0B11 DBFF       984 BSYMBM: IN      PRTA01  ; READ STATUS REG
  0B13 07         985         RLC             ; TEST BUSY BIT= 1
  0B14 DA210B     986         JC      POLMBM  ; IF BUSY= 1, POLL STATUS REG FOR 40H
  0B17 1B         987         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
  0B18 AF         988         XRA     A       ; CLEAR A REG
  0B19 B2         989         ORA     D       ; TEST D REG= 00H
  0B1A B3         990         ORA     E       ; TEST E REG= 00H
  0B1B C2110B     991         JNZ     BSYMBM  ; IF NOT ZERO, CONTINUE POLLING THE MBMPRG COMMAND
  0B1E C32E0B     992         JMP     RETMBM  ; TIME OUT ERROR, RETURN
  0B21 DBFF       993 POLMBM: IN      PRTA01  ; READ STATUS REG
  0B23 A8         994         XRA     B       ; TEST STATUS= 40H, OP-COMPLETE
  0B24 CA2E0B     995         JZ      RETMBM  ; IF OP-COMPLETE, JMP RETMBM
  0B27 1B         996         DCX     D       ; DECREMENT TIME OUT LOOP COUNTER
  0B28 AF         997         XRA     A       ; CLEAR A REG
  0B29 B2         998         ORA     D       ; TEST D REG= 00H
  0B2A B3         999         ORA     E       ; TEST E REG= 00H
  0B2B C2210B    1000         JNZ     POLMBM  ; IF NOT ZERO, CONTINUE POLLING MBM PURGE COMMAND
  0B2E C1        1001 RETMBM: POP     B       ; RESTORE B-C REGS
  0B2F D1        1002         POP     D       ; RESTORE D-E REGS
  0B30 DBFF      1003         IN      PRTA01  ; READ STATUS REG
  0B32 C9        1004         RET             ; RETURN TO CALL
                 1005 $EJECT
```



COMMENT: MINIMUM TIME OUT LOOP COUNTER = 150μS.

Figure 26. MBMPRG

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE   27


  LOC  OBJ         LINE        SOURCE STATEMENT

                   1006 ;
                   1007         END


PUBLIC SYMBOLS
ABORT  A 08DE    BOOTUP A 099C    FIFORS A 0813    INBUBL A 0961    MBMPRG A 0B06    RDBLRS A 0AD3    RDBOOT A 0A2C
RDBUBL A 0936    RDFIFO A 0A8A    WRBLRS A 0A9B    WRBUBL A 090B    WRFIFO A 0A70

EXTERNAL SYMBOLS


USER SYMBOLS
ABORT  A 08DE    ALLFFS A 09BB    BLCODE A 09F1    BOOTUP A 099C    BSYMBM A 0B11    BSYRBL A 0ADE    BSYWBL A 0AB1
BTLPRD A 0A52    BUSYA  A 08E9    BUSYB  A 09C8    BUSYBL A 0A05    BUSYFR A 081E    BUSYIN A 097A    BUSYRB A 0A42
BUSYRD A 08AD    BUSYWR A 0873    BYTCNT A 0840    CALLRD A 0AFE    CONT   A 09E8    DONE   A 0867    FIFORD A 0A62
FIFORS A 0813    FINSHR A 08DB    FINSHW A 08A1    INBUBL A 0961    INFIFO A 0A7D    INTPAR A 0800    LOAD   A 0808
LOOPRD A 094F    LOOPWR A 0924    MBMPRG A 0B06    MULT   A 0852    MULT1  A 0862    MULTO  A 0856    OUTFIF A 0A8E
POLLA  A 08F9    POLLBL A 0A15    POLLBR A 09D8    POLLFR A 082E    POLLIN A 098A    POLLRD A 08BA    POLLWR A 0880
POLMBM A 0B21    POLRBL A 0AEE    POLWBL A 0AC1    PRTA00 A 00FE    PRTA01 A 00FF    RDBLRS A 0AD3    RDBOOT A 0A2C
RDBUBL A 0936    RDFIFO A 0A8A    READ   A 08A4    RETA   A 0906    RETBT  A 0A23    RETFR  A 083B    RETIN  A 0997
RETMBM A 0B2E    RETRBL A 0B01    RETRD  A 095C    RETRDB A 0A6A    RETWBL A 0ACE    RETWF  A 0A85    RETWR  A 0931
RFIFO  A 08D0    WFIFO  A 0896    WRBLRS A 0A9B    WRBUBL A 090B    WRFIFO A 0A70    WRITE  A 086A

ASSEMBLY COMPLETE,  NO ERRORS
```

APPENDIX B
SERVICE INFORMATION

SERVICE INFORMATION

Typically, a Bubble Memory System will never require any special service throughout its useful life. The sequence of program flow presented in Appendix B is not required for normal read/write operation. However, power supply failure, socket contact problems, or component failures may inadvertently produce a BPK 72 system failure.

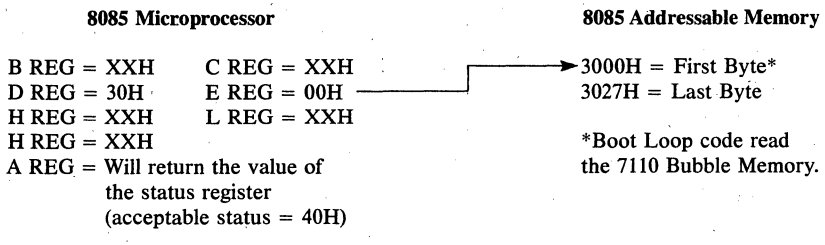
Note: Power supply failure is defined as any violation of the power supply specifications listed in the section titled, "Power Supply Requirements."

A figure titled, "BPK 72 Failure Recovery" is included in Appendix C to illustrate the sequence of events necessary to remedy a Bubble Memory System failure. The flowchart is intended as a guide for handling a Bubble Memory System failure. A system failure is defined as continued attempts that fail to read and write data correctly. Upon detection of a BPK 72 system failure, the first course of action is to verify the existence of the seeds within the 7110 Bubble Memory module. Four replicating Bubble Memory generators reside in the 7110. Each generator requires one seed from which all other bubbles are created. Under extreme circumstances such as power supply failure, one or all of the seeds can be destroyed making it impossible to write data into the 7110's storage loops. The "BPK 72 Failure Recovery" flowchart requests a call to the "seed verification procedure." The "seed verification procedure" should be followed closely to determine if any of the seeds are missing.

In the unlikely event that some or all of the seeds are lost, the "BPK 72 Failure Recovery" figure instructs the reader to perform the "procedure to reseed a 7110 Bubble Memory." The seed replacement procedure will create a seed in each of the four generators. After completing the seed replacement procedure, the "seed verification procedure" should be performed again to confirm that all four seeds are present in the 7110.

The next step in diagnosing a BPK 72 system failure is to verify the accuracy of the boot loop code within the 7110. The boot loop is a map containing information about the active and inactive storage loops. The 7110 is designed with a 15% storage loop redundancy to improve the product yield during manufacture. A diagnostic subroutine named RDBOOT can be called to read the boot loop from the 7110. It is the responsibility of the calling routine to verify that the boot loop code read from the 7110 matches byte for byte with the code found on the label attached to the case of the Bubble Memory module.

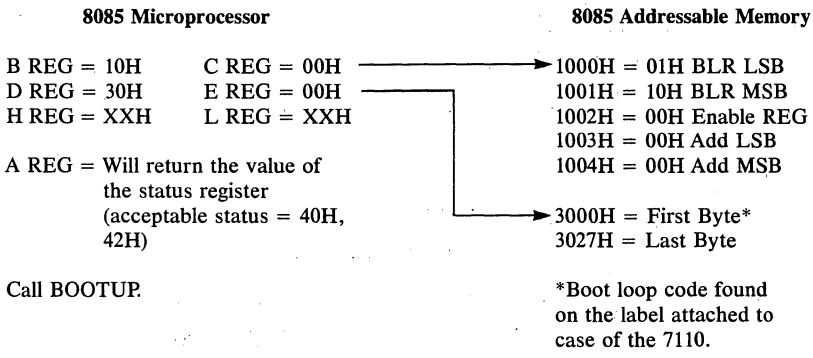
The following is an example of how to use the read Bubble Memory boot loop subroutine, RDBOOT:



Additional detail regarding the use of the read Bubble Memory boot loop subroutine, RDBOOT, may be found in the software listing presented in Appendix A.

If the boot loop is incorrect, a subroutine called BOOTUP is provided for writing the boot loop into the 7110.

The following is an example of how to use BOOTUP to write the boot loop code into the 7110:



Additional detail regarding the use of the write Bubble Memory boot loop subroutine, BOOTUP, may also be found in the software listing presented in Appendix A.

After the seeds and boot loop have been examined and replaced as necessary, the remaining step is to call the initialization subroutine, INBUBL. See the section titled, "Initalizing the Bubble" for a description of how to call the initialization subroutine. If the initialization subroutine returns a status of 40H, the BPK 72 is ready to be put back into service.

Contact the local Intel field sales office in the unlikely event that the BPK 72 system failure guidelines do not eliminate the problem.

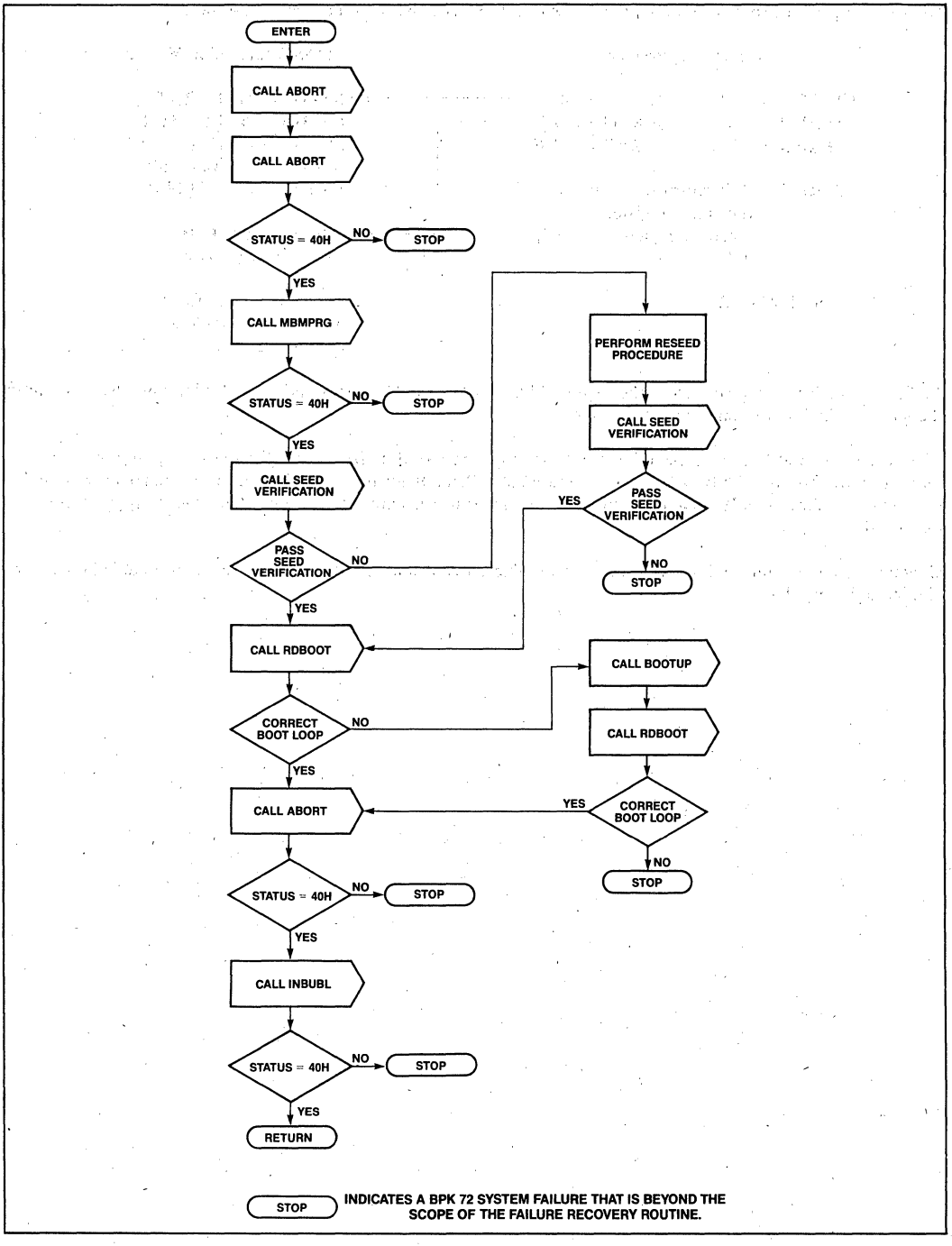
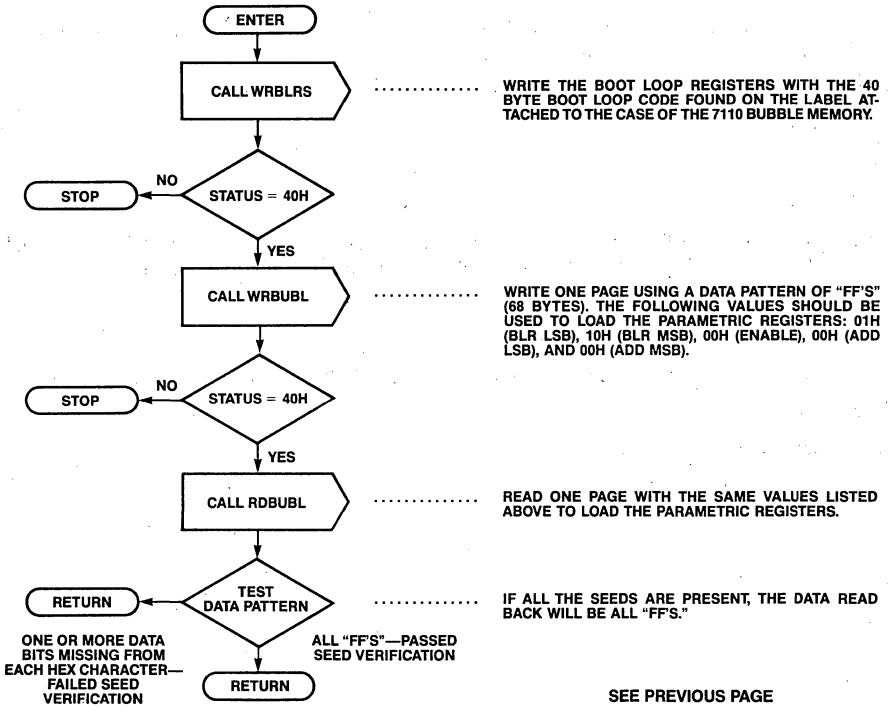


Figure 28. BPK 72 Failure Recovery



If one or more seeds are missing, the data read back will be a pattern with one or more bits missing from each hex character. One example of several possible patterns is shown below. Each pattern will typically contain a dominant pair of hex characters (i.e., "88's" or "AA's"). In any case, if seeds are missing no "FF's" will be read using the subroutine, RDBUBL.

88	88	88	88	88	88	88	88	88	00	08	88	88	88	88	88
88	88	88	88	88	08	80	88	88	88	88	88	88	88	88	88
88	A8	88	80	08	88	88	88	88	80	88	88	A8	88	8A	88
A8	88	8A	88	88	88	88	A8	88	AA	88	88	88	8A	88	88

Do not attempt to use the seed verification procedure without first performing the program sequence described in Figure 28, "BPK 72 Failure Recovery."

Figure 29. Seed Verification Procedure

PROCEDURE TO RESEED A 7110 BUBBLE MEMORY

1. Remove power from circuit.
2. Remove the 7230 current pulse generator from its socket, and install the 7230 in the socket provided on the seed module. Be careful to note the orientation of Pin 1.
3. Install the seed module (with the 7230 installed) in the 7230 socket.
4. Apply power to the circuit.
5. Call ABORT.
6. Call MBMPRG.
7. Call WRBUBL (1 page transfer, any location, data pattern is not important). Parametric register values; 01H (BLR LSB), 10H (BLR MSB), 00H (ENABLE), 00H (add LSB), and 00H (add MSB).
8. Remove power from circuit.
9. Remove the seed module from the 7230 socket.
10. Remove the 7230 from the seed module and reinstall the 7230 in its socket on the IMB-72 board.
11. Apply power to the circuit.
12. Reseed procedure is now complete.

intel

APPLICATION NOTE

AP-157

October 1983

Software Design and Implementation Details for Bubble Memory Systems

Richard Pierce
Applications Engineer
Intel Corporation

ORDER NUMBER: 230707-002

INTRODUCTION

The 7220-1 is a single-chip LSI Bubble Memory Controller (BMC) that implements a bubble memory storage subsystem (with up to eight bubble storage units (BSUs) per BMC). Each bubble storage unit consists of five support circuits in addition to the bubble memory chip and provides one megabit (128 kbytes) of non-volatile read/write memory. This application note examines the programmatic interface to the 7220-1 BMC and how communications between a host processor and the bubble subsystem (i.e., 7220-1) govern all bubble system operations.

The BMC provides all the control and timing signals for the bubble and support circuits. All data synchronization and error checking is automatically performed by the bubble subsystem to ensure reliable data storage. The BMC easily interfaces to microprocessor systems and communicates via a set of high-level commands. This application note explains the operations and functions performed by these instructions and provides the basic programmatic interface descriptions and program guidelines to allow you to design and implement a program module or "driver" to control all bubble system operations. In addition, several possible software interface levels are defined. While the design guidelines presented are not targeted for integration with any particular operating system, they do, however, provide conceptual information on design requirements and serve as a foundation on which to develop a modular and flexible software driver aligned with your specific application requirements.

Product Line Overview

Intel offers a complete line of bubble memory components, development kits and assembled boards.

The BPK 72 (Bubble Memory Prototype Kit) serves primarily as a means to evaluate the potential of bubble storage. The BPK 72 comes complete with all the hardware and documentation necessary to prototype a one-megabit bubble memory system. After the kit is assembled, the designer is left with the simple task of interfacing to a host processor.

The BPK 70 one-megabit bubble storage subsystem is a fully interchangable component bubble memory system. Each kit contains all the components in a Bubble Storage Unit (BSU). A single 7220-1 Bubble Memory Controller (purchased separately) can operate up to eight BPK 70 subsystems at one time. The BPK 70 is available for the production of custom systems where high volume is required.

The iSBX™ 251 Magnetic Bubble MULTIMODULE™ board is a one-megabit bubble memory mounted on a standard dual width Intel MULTIMODULE memory expansion card. Completely assembled and tested, the iSBX-251 board is fully plug compatible with all Intel iSBC Single Board Computers that have iSBX connectors.

The iSBC® 254 Bubble Memory Board is a completely assembled Intel MULTIBUS® memory board. The board can be configured with one bubble memory (128 kbytes), two bubble memories (256 kbytes), or four bubble memories (512 kbytes).

The 7220-1 BMC acts as the interface to the host processor in each of the aforementioned products, thus simplifying system programming. The basic programming techniques discussed in this application note will provide the system programmer with a complete understanding of programming requirements and shorten the software development time.

SOFTWARE INTERFACE

Basic Driver Operation

As will become evident, a basic compromise or "tradeoff" exists between the design of your application software and the capabilities of the bubble memory controller. While you will be responsible for the development of a driver to integrate the bubble into your system, the level of driver interaction and degree of flexibility will vary according to the needs of your application. If an application program is small and simple, a basic bubble driver simply may be called from the main program. At the next level of driver sophistication, the bubble system is viewed by the application program as a logical device. At this level, the key to driver design is the mapping of the "logical bubble interface" (as viewed by the application program) into the "physical bubble interface" (as implemented by the bubble drivers). This logical-to-physical mapping serves to isolate application programs and system software from the idiosyncracies of the bubble memory controller. At the

highest level of driver sophistication, the application program treats the bubble system as a collection of named data areas or files similar to the way in which data is stored and retrieved in disk operating systems. At the file system level, an application program can ignore the mechanics of bubble storage and access and merely present a "file name" to the driver to open, read or write, and then close the desired "bubble file."

At the subsystem level (i.e., "physical bubble interface"), the bubble driver is responsible for all system interaction with the bubble and is intrinsic to the efficient and reliable operation of the bubble system. The driver accepts bubble memory commands and command execution parameters from the application program, controls and monitors command execution, and returns operational status information to the application program at command completion. To perform all of these operations, the bubble driver must support the bit/byte level of the bubble memory controller's command and status registers and the "parametric" registers that define the operating mode, system configuration, and extent of the transfer. Depending on the interface level of the application software, the driver itself may be made up of a set of subroutines that are called individually by the host to perform specific bubble system operations.

SOFTWARE-SELECTIVE CONFIGURATIONS

Before you can begin to design a software driver, you must consider all of the selective configurations available within the bubble system and which of these configurations you want to support. As will be explained, the type of data transfer (i.e., direct memory access, interrupt driven, or polled), the data transfer rate, and the selective implementation of the bubble system's error correction feature all are software controlled. The complexity of the driver design depends on the system flexibility desired. For example, a driver may be designed to support only one transfer mode and may not make use of error correction. Conversely, a more generalized driver can be designed to support various transfer modes, data rates, and levels of error correction. The ensuing paragraphs define the driver responsibilities associated with the available software configurations and will aid you in designing a driver that satisfies your specific application.

Data Organization

Probably the most important aspect of the bubble memory system interface is the organization of data. From a software viewpoint, data logically is organized into blocks of bytes called "pages." During data transfer operations, one or more of these pages are transferred between the bubble(s) and the host microprocessor. A page is the smallest increment of data that can be transferred; single bytes cannot be transferred. Conceptually, the data organization within a bubble memory is analogous to a disk system. Just as disk sector sizes are fixed when a disk is formatted, bubble page sizes are established, under software control, when the bubble system is initialized. For a single bubble system, the page size is fixed at either 64 bytes when error correction is implemented or 68 bytes without error correction, and the total number of pages available is 2048. In systems with multiple bubbles, page size can vary from 64 bytes (68 bytes without error correction) to 512 bytes (544 bytes without error correction) depending on the number of bubble devices in the system. Page size is directly proportional to the system data rate and also determines the total number of available pages (address field size). As an example, consider a system consisting of two bubbles (using error correction). With two bubbles, there are two possible ways to configure the system; paralleling the two bubbles for a page size of 128 bytes and a total number of 2048 pages or treating the bubbles serially for a page size of 64 bytes and a total number of 4096 pages. The average data rate for the 128-byte page is 17.0 kbytes per second, and the average data rate for the 64-byte page is 18.5 kbytes per second.

The selection of the appropriate page size depends primarily on the data rate supported by the system. For file system implementation, an additional consideration in page size selection is bubble transfer efficiency versus bubble storage efficiency. Essentially, the following system factors must be weighed:

- Data Rate. The higher the data rate, the faster the microprocessor must respond to the demands of the bubble memory controller. Depending on the data transfer mode selected, some data rates may exceed the data transfer rate of the host microprocessor.
- Bubble Transfer Efficiency. A file consisting of a few large pages can be transferred more efficiently (faster) than a file consisting of a number of small pages.
- Bubble Storage Efficiency. For a typical file system, space must be allocated within each page to link the pages of each file together (individual pages of a file may not necessarily be contiguous). Too large or too small a page size can waste

bubble storage space. If most files are comprised of many small pages (e.g., 64 bytes), a large percentage of bubble storage would be required for establishing the fore/back pointer linkage. Conversely, if most files are smaller than a single page, a large amount of space would remain unused at the end of each page.

Buffering

Buffer operation is an extremely important factor in the reliable transfer of data between the bubble and system memory and is a major consideration in software driver design. A buffer, ideally speaking, is a memory storage area that contains the same amount of data storage as the data block to be transferred. The bubble system's bubble memory controller includes a first-in, first-out (FIFO) 40 byte data buffer that reconciles timing differences between the parallel data transfer to or from the host microprocessor and the serial data transfer to or from the bubble memory. Accordingly, when an application program requests data from a bubble, the software driver is responsible for keeping up with the FIFO for the duration of the data transfer in order to prevent the FIFO from overflowing or underflowing. The specific software driver requirements are dependent on the method of data transfer selected.

Data Transfer Interface Modes

Three distinct software interface techniques can be used to interface host system memory with the bubble system for page data transfer: DMA: interrupt driven, and polled.

In the DMA transfer mode, the BMC operates in conjunction with a DMA controller (e.g., Intel's 8257 or 8237) and uses the DRQ (data request) and DACK/ (data acknowledge) signal lines for establishing the handshake protocol. Assisted by the DMA controller, the BMC transfers data to or from the host system memory. Once the transfer begins, further program intervention is not required until the entire transfer has been completed.

In the interrupt-driven data transfer mode, the DRQ line is connected to an interrupt controller (e.g., Intel's 8259A). During non-DMA data transfers the DRQ line indicates when the BMC's FIFO is half-full (bubble read operations) or half-empty (bubble write operations). This method of interrupting results in a data block transfer arrangement in which the software is responsible for performing the appropriate transfer of data (typically 22 bytes) to or from the FIFO when the interrupt occurs. Using this technique, the software driver only processes the FIFO buffer as needed, and program waits during I/O transfers (polled I/O) are eliminated.

The polled I/O mode is the most simple to implement since no special or external hardware is required to perform data transfers. In the polled I/O mode, the software must determine when to transfer data to or from the FIFO by continually polling a status bit in the BMC's Status Register. This status bit indicates the presence or absence of data in the FIFO on a byte-by-byte basis. The polled I/O mode places significant demand on the host system's processing time since the software continuously must monitor the Status Register to ensure that FIFO overflow or underflow does not occur.

ECC Highlights

The last software controlled option to be considered in the design of your bubble driver is if the built-in error correction circuitry (ECC) within the 7242 Formatter/Sense Amplifier is to be implemented and, if so, what level of error correction is best suited to your application.

Although the inherent data integrity of your bubble memory is extremely high, the incorporation of error correction improves the overall integrity of your system by several orders of magnitude. While boosting the data integrity, the implementation of error correction adds increased overhead to both driver design and host interaction. Since the associated error handling routines can range from simple to complex, you must carefully weigh the software requirements before considering the level of error correction to be supported. In balancing driver and host responsibilities, you must understand the following factors pertaining to ECC:

- The type and nature of errors associated with bubble memories.
- The way in which ECC operates.
- The various levels of error correction available.

All of these factors are explained in detail in a later section.

COMMUNICATING WITH THE BMC

All communications between the host and the bubble memory actually are peformed through the 7220-1 BMC. The BMC has two input/output (I/O) ports; an 8-bit bidirectional data port and an 8-bit command/status port (Figure 1). The port addressed is determined by the least-significant bit of the port address byte. Conceptually, the BMC can be thought of as a disk system controller in that data in the bubble memory is organized into blocks called "pages" that are similar to disk sectors. Information such as starting page location, direction of transfer, and the number of pages to be transferred is passed to the BMC before the desired read or write operation is initiated.

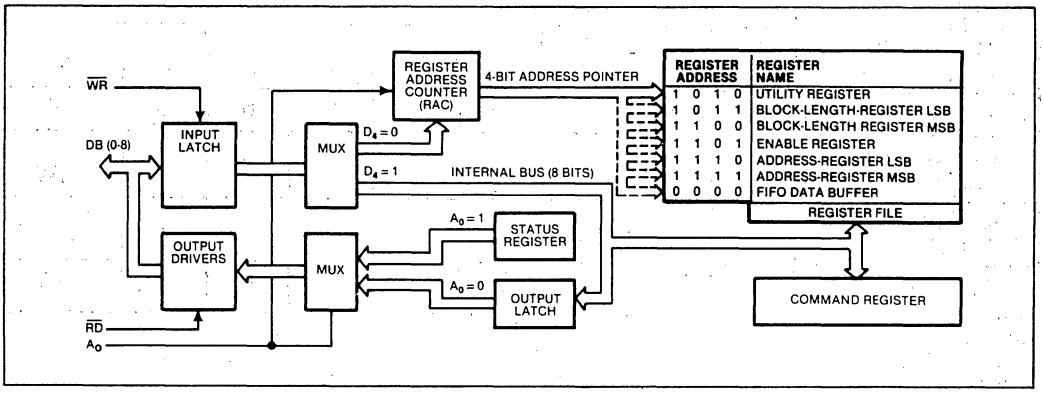


Figure 1. BMC Block Diagram

For simplicity, you can think of the BMC as a 40-byte FIFO (first-in first-out) buffer and a series of six user-accessible 8-bit registers. The FIFO passes data between the outside world and the bubble system's 7242 Formatter/Sense Amplifier and compensates for speed variations. The six registers are loaded prior to most operations and contain information regarding the upcoming transfer and the operating mode of the BMC. Since these registers always are loaded before a command is sent similar to passing parameters to a subroutine before it is invoked, this set of registers is referred to as the "parametric registers." The transfer of data between the host and the BMC's FIFO and parametric registers takes place over the 8-bit data port. The destination (FIFO or parametric register) of the data port transfer is determined by the value in another register called the RAC. As shown in Table 1, bit 4 of the command/status port byte is used to distinguish between a BMC command and either the address of one of the parametric registers or the FIFO.

Table 1. Command Port Function

Function	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Command	0	0	0	1	C	C	C	C
RAC	0	0	0	0	R	R	R	R

When bit 4 is a "one," the low-order four bits are decoded as a BMC command, and when bit 4 is a "zero," the low-order four bits are interpreted as a pointer to the parametric registers/FIFO. This 4-bit register pointer is referred to as the "Register Address Counter" or simply the "RAC."

RAC values that may be written to the command/status port and the registers selected are outlined in Table 2. The RAC points to, or selects, one of the six registers or the FIFO. Once a RAC value is written to the command status port, the next data port read or write operation transfers data between the host interface and the register addressed.

Table 2. Register Address Counter Assignments*

Register Name	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Read/ Write
Utility Register	0	0	0	0	1	0	1	0	R/W
Block Length Register (LSB)	0	0	0	0	1	0	1	1	W
Block Length Register (MSB)	0	0	0	0	1	1	0	0	W
Enable Register	0	0	0	0	1	1	0	1	W
Address Register (LSB)	0	0	0	0	1	1	1	0	R/W
Address Register (MSB)	0**	0	0	0	1	1	1	1	R/W
7220 FIFO	0	0	0	0	0	0	0	0	R/W

NOTE(S):
* With A0 = 1
** Write-only bit

Referring now to the table, notice that the register addresses are in hexadecimal order from "A" to "F" and that the FIFO has an address of zero. This arrangement of addresses is due to the RAC's auto-incrementing feature. Once a register is selected, each subsequent data port I/O read or write causes the RAC to advance and to point to the next register in the sequence. After the most significant byte of the Address Register is addressed, the RAC advances from F to 0 to point to the FIFO. When it reaches this point it no longer increments. The system now is ready to transfer data into or out of the FIFO without further instructions from the host.

After the FIFO is selected, the RAC stops incrementing and continues to point to the FIFO until the RAC again is accessed through the command/status port. The auto-incrementing feature minimizes the number of instructions required for a given command sequence and ensures that all of the required parametric information is sent to the BMC.

As a user, you are not required to utilize the auto-incrementing feature; each parametric register can be selected and loaded in any order, and specific registers may be updated as required. When individual registers are not accessed in order, each register must be specifically addressed and loaded. Until you become more familiar with the bubble system, the auto-incrementing feature is recommended.

A point to remember is that once a command has been issued to the BMC, the parametric registers must not be updated until the operation is complete. The parametric registers essentially are working registers for the BMC during command execution. When a bubble read or write operation is in progress, the Block Length Register, as explained later in this chapter, contains the terminal page count and is decremented with each page transferred. Attempting to modify this register during command execution would cause the final page count to be incorrect.

The Parametric Registers

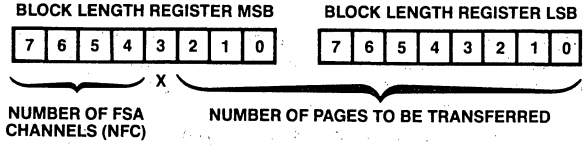
Now that you have been introduced to the Register Address Counter and its operation, let's look at the individual parametric registers addressed by the RAC in more detail.

Utility Register

The Utility Register is a user-defined register that can be both written and read. This register is not incremented or decremented by the BMC. Since the Utility Register is the first register of the RAC sequence, if the register is not used, the least-significant byte of the Block Length Register initially can be addressed to eliminate the Utility Register from the sequence. Note that the 4 Mbit bubble memory controller (7224) does not contain a utility register.

Block Length Register

The Block Length Register is made up of two 8-bit registers, a low-order byte register and a high-order byte register. The contents of the block length register determine the system page size and also the number of pages to be transferred in response to a single bubble data read or write command. The Block Length Register or "BLR" is a write-only register that is divided into a terminal count field and a channel field as follows:



The terminal count field is eleven bits in length and is loaded with the total number of pages to be transferred in the ensuing bubble read or write operation. With a field length of eleven bits, from 1 to 2048 pages can be transferred (all zeroes in the field indicates a 2048-page transfer).

The page width (size) is defined by the 4-bit channel field. This field actually specifies the number of formatter/sense amplifier channels available. Note that each 7242 formatter/sense amplifier has two channels to communicate with each bubble memory, therefore the acceptable values in this field select one channel (one half of a bubble memory), two, four, eight, or 16 channels. These field values correspond to page sizes of 32, 64, 128, 256, and 512 bytes (assuming error correction), respectively, when the bubble memories are operated in parallel. (The one-channel mode usually is reserved for diagnostic operations.) Table 3 shows the relationship among page size, channel field value, and formatter/sense amplifier channel selection for parallel bubble operation.

As shown in the table, the channel field bits are encoded and only one bit ever is set in the field.

For example, a channel field value of "0001" selects one bubble memory through channels 0 and 1.

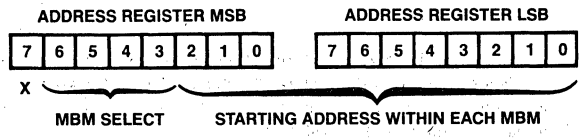
Table 3. FSA Channel Select/MBM Select

MBM Select AP, MSB Bits (6, 5, 4, 3)	"Channel Field" (BLR MSB Bits 7, 6, 5, 4)				
	0000*	0001	0010	0100	1000
0 0 0 0	0	0, 1	0, 1, 2, 3	0 to 7	0 to F
0 0 0 1	1	2, 3	4, 5, 6, 7	8 to F	
0 0 1 0	2	4, 5	8, 9, A, B		
0 0 1 1	3	6, 7	C, D, E, F		
0 1 0 0	4	8, 9			
0 1 0 1	5	A, B			
0 1 1 0	6	C, D			
0 1 1 1	7	E, F			
1 0 0 0	8				
1 0 0 1	9				
1 0 1 0	A				
1 0 1 1	B				
1 1 0 0	C				
1 1 0 1	D				
1 1 1 0	E				
1 1 1 1	F				

NOTE(S):
*Normally reserved for diagnostic operations.

Address Register

The Address Register, like the Block Length Register, is made up of two 8-bit registers; a low-order byte register and a high-order byte register. The Address Register is divided into a starting address field and an MBM (Magnetic Bubble Memory) select field show as follows:



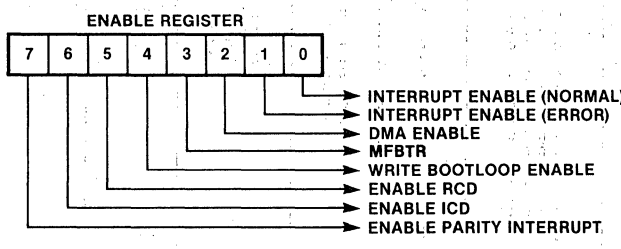
The Address Register's starting address field is eleven bits in length and is used to define on which page of a bubble's 2048 pages that the transfer is to start. The starting address field is incremented with each page transferred during multipage transfers, automatically selecting the next sequential page.

The Address Register's MBM select field is used in conjunction with the Block Length Register's channel field to control the serial selection of bubble memories or groups of bubble memories operated in parallel. To better understand the function of the MBM select field, consider a system consisting of four bubble memories operated as two banks of two bubble memories each.

Referring back to table 3, the channel field in the Block Length Register would be set to "0010" to select two bubbles in parallel and a corresponding page size of 128 bytes. To select between the two banks, the Address Register's MBM select field would be set to "0000" to select the first bank (FSA channels 0 through 3). As page 2048 is transferred to or from the first bank, the Address Register's starting address field rolls over to "0000" and increments the MBM select field to "0001" to select the second bank (FSA channels 4 through 7).

Enable Register

While the Address and Block Length Registers define the system configuration and data transfer, the Enable Register defines the various modes of operation under which the data transfer is performed and defines the conditions under which interrupts can be generated. Several of the Enable Register bits are used individually while other bits are used in combination. Figure 3 shows the individual Enable Register bit definitions.



Interrupt Enable (Normal), when set ("1"), enables the BMC to interrupt the host processor on the successful completion of a command. Conversely, if this bit is not set, an interrupt is not generated on command completion and the host processor must poll the BMC's Status Register to determine when command execution is complete.

Interrupt Enable (Error) is used in conjunction with the Enable RCD and Enable ICD bits to select various error conditions under which the BMC will terminate command execution and interrupt the host processor. The following table outlines the bits combination and corresponding error conditions recognized.

Table 4. Error Correction Options

Enable ICD (bit 6)	Enable RCD (bit 5)	Interrupt Enable (Error) (bit 1)	Interrupt Condition
0	0	0	No interrupt on error
0	0	1	Interrupt only on timing error
0	1	0	Interrupt on uncorrectable or timing error
0	1	1	*Interrupt on uncorrectable, correctable or timing error
1	0	0	Interrupt on uncorrectable or timing error
1	0	1	Interrupt on uncorrectable, correctable or timing error
1	1	0	Illegal
1	1	1	Illegal

NOTE(S):
*Normally not used.

DMA Enable, when set, enables the BMC to operate in the DMA data transfer mode. In the DMA mode, a DMA controller is interfaced to the BMC and DRQ-DACK/protocol is used to perform byte transfers. Note that in the DMA mode, the BMC activates its DRQ output *each time* it places a byte in the FIFO (bubble read operation) or each time there is room for *at least one byte* in the FIFO (bubble write operation). When the DMA Enable bit is not set, the BMC operates in the interrupt driven or polled mode. In either of these modes, the BMC's DRQ output goes active when 22 or more bytes are present in the FIFO during a bubble read operation or when there is space for 22 bytes during a bubble write operation. Note that if the DRQ is not used (i.e., polled mode), the host processor must examine the Status Register's FIFO Ready bit to determine when data should be taken from or written to the FIFO.

MFBTR, (Maximum FSA to BMC Transfer Rate), determines the maximum burst transfer rate from the FSA(s) to the BMC FIFO. This bit only applies to bubble read operations and is effective only during single-page transfers or during the transfer of the last page of a multipage transfer. Table 3 shows the effect of MFTBR bit on the transfer rate for parallel bubble operation.

Write Bootloop Enable, when set, enables the bootloop to be rewritten. Conversely, if this bit is not set and a Write Bootloop command is received, the command is aborted immediately and the Timing Error is set in the Status Register. Since writing the bootloop only is performed as a diagnostic test or to recover from a system failure, this bit provides protection against accidental rewrite of the bootloop data.

Enable RCD (Read Corrected Data), when set, causes the BMC to issue a Read Corrected Data instruction to all the FSAs in the system in response to one or more FSAs reporting an error. On receipt of the instruction, each FSA cycles the erroneous page through its error correction logic and then transfers the page to the BMC. For any FSA not reporting an error, the data harmlessly cycles through the error correction logic. When the page transfer is complete, the BMC interrogates the FSA to determine if the error was correctable (if the error was uncorrectable, erroneous data would have been transferred to the BMC).

Enable ICD (Internally Correct Data), when set, causes the BMC to issue an Internally Correct Data instruction. Any FSA reporting an error causes the instruction to be issued to all FSAs in the system.

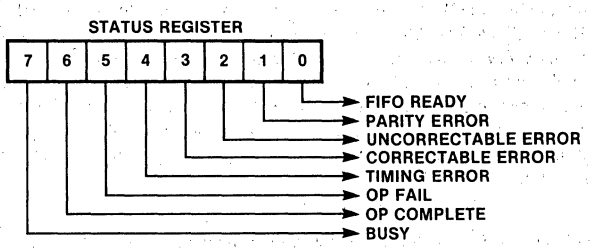
On receipt of the instruction, each FSA cycles the data through its error correction logic (regardless of whether it contained an error), but does not transfer the page to the BMC. After cycling the page, each FSA reports its error Status (correctable or uncorrectable) to the BMC. The FSA not reporting an error harmlessly cycles through the error correction logic and reports a correctable error to the BMC. Note that both the Enable RCD and Enable ICD bits cannot be set at the same time.

Enable Parity Interrupt, when set, enables the BMC to interrupt the host processor when it detects a parity error on a data byte sent from the host processor (the BMC automatically checks for odd parity on each data byte received from the host processor and implements odd parity on each byte sent to the host processor). When the Enable Parity Interrupt bit is not set and parity error is detected, no interrupt is generated (following the transfer, the Parity Error bit in the Status Register will be set).

Status Register

As stated at the beginning of this chapter, the host processor executes an I/O read instruction with the BMC's A0 address input equal to "1" to access the Status Register. As will become evident in the individual Status Register bit descriptions, the Status Register is read in response to interrupts from the BMC and, during polled data transfers, is read continually to determine when data is to be written to or read from the BMC's FIFO.

The Status Register also provides information regarding error conditions, the completion or termination of commands, and the BMC's readiness to accept new commands. The individual Status Register bits are shown as follows.



Note that bits 1 through 6 are valid only when the Busy bit (bit 7) is not set and that these bits are cleared whenever a new command is received. The Status Register also can be cleared by writing to the register address counter (RAC) with bit 5 set to "1" and any valid parametric register address (0AH through 00H).

Busy. Bit 7, the Busy bit, is set ("1") when the BMC is in the process of executing a command. The BMC sets its Busy bit shortly after a command is received (the Busy bit must be clear in order for the BMC to accept any command other than an Abort command) and keeps Busy set until the operation is complete (or until an operation is halted because an error has occurred.)

It is important to note that the Busy bit remains set until all other status bits have been updated and that it therefore is possible to see illogical bit combinations such as Busy and Op Complete at the same time. With the exception of the FIFO Available bit, all other Status Register bits should be considered valid only after the Busy bit returns to an inactive ("0") level.

OP Complete, Bit 6, Op Complete, is set when the BMC successfully completes the execution of a command.

OP Fail, Bit 5, Op Fail, is set when the BMC is unable to complete execution of a command. In general, this bit is valid only after the Busy bit returns to an inactive level; other error bits in the Status Register will be set to indicate why the operation failed.

Timing Error, Bit 4, Timing Error, is set for several error conditions. The most frequent cause of timing errors is when the host processor cannot keep up with the rate at which the BMC fills or empties its FIFO (referred to as an overflow or underflow condition). Timing errors also occur if the correct number of bits is not set in an FSA's bootloop register (to function properly, an FSA must have either 272 loops active when error correction is not implemented or 270 loops active when error correction is implemented). This condition occurs during a Read command if a mistake is made either when the bubble's bootloop is written or if the bootloop register is loaded incorrectly from the user's system. Another source of timing error is if no bootloop sync code is found during an Initialize or Read Bootloop command. The last source of a timing error is when a Write Bootloop command is received by the BMC and the Write Bootloop Enable bit is not set in the Enable Register. Of the preceding sources, regular or periodic occurrences of timing errors usually indicate an inherent inability of the host processor to meet the bubble's data transfer requirements and will be most apparent during bubble read operations, especially if the MFBTR bit is set (i.e., MFBTR=0 in the Enable Register).

Correctable Error, Bit 3, Correctable Error, is set when an FSA reports that a correctable error was detected during the last bubble read operation. Depending on the level of error correction selected, the setting of this bit indicates a correctable error in the current page held by the FSA, in the last page read, or in one of the pages read during a multipage transfer.

Uncorrectable Error, Bit 2, Uncorrectable Error, is set when an FSA reports an uncorrectable error during the last bubble read operation. Like the Correctable Error bit, the setting of this bit depends on the level of error correction selected and indicates an uncorrectable error in either the last page transferred or in the page currently held by the FSA.

Parity Error, Bit 1, Parity Error, is set when the BMC detects a parity error on the data byte sent from the host during a bubble write operation. If the Enable Parity Interrupt bit is set in the Enable Register, the BMC will terminate the write operation and interrupt the host processor when the parity error is detected.

FIFO Available, Bit 0, FIFO Available, is unique in that it is the only bit the Status Register that is valid when the Busy bit is set. During data transfer operations (i.e., when the Busy bit is set), the FIFO Available bit acts as a gate for the host microprocessor's data handling software. During bubble write operations, if the FIFO Available bit is set there is room for more data in the FIFO and the host microprocessor can proceed with the transfer. Conversely, if the FIFO Available bit is not set, the FIFO is full and the host must wait for the BMC to "catch up" before sending more data. During bubble read operations, if the FIFO Available bit is set, data has been placed in the FIFO by the BMC and can be taken by the host microprocessor; if the FIFO Available bit is not set, the data is not yet available.

FIFO

The BMC's first-in first-out (FIFO) buffer passes data between the user interface and the FSA. The primary purpose for the FIFO is to reconcile timing differences between both the user interface and the BMC and between the BMC and the FSAs.

The FIFO itself is 40 bytes wide and, including the FIFO's input and output latches and the BMC's input latch, can store up to 43 bytes. The FIFO is "dual ported" in that data can be written into one port while simultaneously being read from the other port.

When the BMC is busy executing a command, the FIFO functions as a data buffer, and when the BMC is not busy, the FIFO is available for use as a general-purpose FIFO. For this reason, the BMC is said to be in the general-purpose FIFO mode when it is not busy. During execution of commands that involve the transfer of data between the user interface and the FSAs, the data passes through the FIFO, and the status of the FIFO is indicated by the FIFO Ready bit in the BMC's Status Register. When the FIFO Ready bit is set ("one") during a write operation, there is space in the FIFO for more data, and when this bit is set during a read operation, data is present in the FIFO.

The BMC's DRQ output also indicates FIFO status. In the DMA data transfer mode, DRQ is used in conjunction with the DACK/ input from a DMA controller (e.g., in Intel 8257 or 8237) on the user interface to provide a standard DMA data transfer protocol. In the non-DMA transfer mode (polled or interrupt-driven data transfers), the DRQ output indicates that the FIFO is either half full or half empty according to the direction of the data transfer; during a write operation, DRQ is set when there is space for 22 more bytes, and during read operation, DRQ is set when there are 22 bytes present in the FIFO. Accordingly, when performing non-DMA data transfers, blocks of 22 bytes should be transferred to or from the FIFO so that the host processor does not spend a disproportionate amount of time servicing the DRQ-initiated interrupt or polling the BMC's Status Register.

The FIFO automatically is addressed after the last parametric register is written due to the self-incrementing nature of the Register Address Counter. Alternatively, the FIFO can be addressed explicitly by writing to address 0 of the Register Address Counter. Note that since byte 0 of the FIFO is cleared whenever the Register Address Counter is addressed, writing the parametric registers (or to the FIFO itself) must be avoided while the FIFO contains valid data. Also, when using the FIFO in the general-purpose mode, the host system is responsible for resetting the FIFO prior to any data transfer.

During read and write operations, the host system is responsible for keeping up with the data transfer in order to prevent a FIFO overflow or underflow condition. If the FIFO overflows or underflows during the data transfer, the operation is aborted and the Op Fail and Timing Error bits are set in the BMC's Status Register.

The FIFO AVAILABLE bit is also set whenever the RAC is not pointing to the BMC FIFO (RAC address = 00H). When writing the parametric registers, for example, if the user reads the status between say the Enable Register and the Address Register, the status byte would indicate FIFO AVAILABLE. This status value is forced by internal BMC logic.

COMMANDS

The BMC's command set consists of 16 commands that are selected by a 4-bit command code. As previously described, commands are passed to the BMC by writing to the BMC's command port with bit 4 of the command byte set to "1" (the Register Address Counter is addressed when bit 4 is "0"). Table 5 lists the 4-bit command codes and the corresponding commands.

For most commands sent to the BMC, the parametric registers must be loaded with operating information before the command is sent. Also, with the exception of the Abort command, commands cannot be issued to the BMC while another command is being executed (i.e., when the Busy bit in the Status Register is set). The remainder of this section offers brief descriptions of the BMC's command set; descriptions of command usage are presented in succeeding sections. Table 6 of this section presents a summary of command execution times.

The 16 commands can be grouped into categories according to their frequency of use. The most commonly used commands are:

- Abort
- Initialize
- Read Bubble Data
- Write Bubble Data

Other commands used during normal bubble system operation include:

- Read Seek
- Write Seek
- MBM Purge
- Read Corrected Data
- Reset FIFO
- Software Reset
- Read FSA Status
- Read Bootloop

The remaining commands are related to bootloop operation and are used only for diagnostic purposes:

- Read Bootloop Register
- Write Bootloop Register
- Write Bootloop Register Masked
- Write Bootloop

Abort

The Abort command (unlike any other command), when issued while the BMC is busy executing another command, causes the termination of the command being executed. On receipt of this command, the MBMs are stopped in an orderly manner to prevent the loss of bubble data. Since the Abort command is the only command recognized by the BMC while it is busy, this command is issued following power-up or whenever the BMC is in an unknown state. The Abort command requires no prior loading of the parametric registers.

When an Abort command is issued while the BMC is not busy, the command functions as an FIFO Reset to clear any data present in the FIFO. An Abort command issued when the MBM drive coils are active (i.e., data transfer command is executing indicated by the Busy bit in the Status Register) must be followed by an Initialize command.

Table 5. BMC Command Set

D3	D2	D2	D1	Command Name
0	0	0	0	Write Bootloop Register Masked
0	0	0	1	Initialize
0	0	1	0	Read Bubble Data
0	0	1	1	Write Bubble Data
0	1	0	0	Read Seek
0	1	0	1	Read Bootloop Register
0	1	1	0	Write Bootloop Register
0	1	1	1	Write Bootloop
1	0	0	0	Read FSA Status
1	0	0	1	Abort
1	0	1	0	Write Seek
1	0	1	1	Read Bootloop
1	1	0	0	Read Corrected Data
1	1	0	1	Reset FIFO
1	1	1	0	MBM Purge
1	1	1	1	Software Reset

Table 6. 7220-1 Command Execution Times

Four-Bit Command Code (Hex)	Command	Description	Performance
0	LRBLRMSK	1 FSA channel selected 2 FSA channel selected	900 μs 900 μs
1	Initialize	Best case (N=#MBM) Worst case	350+(85,200) N μs 350+(164,740) N μs
2	Read	Single page (MFBTR=0) Single page (MFBTR=1) N page transfer (MFBTR=0) N page transfer (MFBTR=1)	t_{SEEK}+8690 μs t_{SEEK}+12,770 μs t_{SEEK}+8690+(7500) (N−1) μs t_{SEEK}+12,770+(7500) (N−1) μs
3	Write	1 page transfer N page transfer	t_{SEEK}+7450 μs t_{SEEK}+7450+(7500) (N−1) μs
4	Read Seek	Best case Worst case	7350 μs 89,250 μs
5	Read BLR	Any number of FSA channels	900 μs
6	Write BLR	Any number of FSA channels	900 μs
7	Write BL	Single bubble selected	82,850 μs
8	Read FSA Status	1 bubble in system N bubbles in system	75 μs 75+40 (N−1) μs
9	Abort	1 bubble in system N bubbles in system	100 μs 100+40 (N−1) μs
A	Write Seek	Best case Worst case	7350 μs 89,250 μs
B	Read BL	Best case Worst case	86,000 μs 165,000 μs
C	Read Corrected Data	Any number of FSAs selected	1400 μs
D	FIFO Reset	N/A	50 μs
E	MBM Purge	N/A	150 μs
F	Software Reset	N/A	50 μs

Initialize

The Initialize command prepares the bubble system for subsequent operations and is used when the bubble system is powered up (following the Abort command) or whenever the system's error correction implementation is changed. The Initialize command effectively performs the following BMC commands:

- Abort
- MBM Purge
- FIFO Reset
- Read Bootloop
- Write Bootloop Register Masked

When an Initialize command is received, all internal registers within the BMC are cleared and the FIFO is reset. The BMC then reads the bootloop from each bubble and writes the corresponding bootloop information into the bootloop registers of each FSA. The bubble is left positioned at page zero (logically) corresponding to the values set in the BMC page address counters. Before an Initialize command can be issued, the following information must be loaded into the parametric registers:

- The channel field in the Block Length Register must be set to "0001" to arrange all bubbles in the system in a serial configuration to allow the individual bootloops to be read from each bubble and subsequently written to the bootloop registers of the corresponding FSA channels.
- The MBM select field in the Address Register must select the last (highest numbered) bubble in the system to inform the BMC of the number of bubbles present within the system (for a one bubble system the value would be "0000").
- The bits selecting error correction in the Enable Register must be set according to how subsequent read and write operations are to be performed (with or without error correction) since the number of 1's written to the FSA bootloop registers is not the same with error correction implemented as when error correction is not implemented. Note that simply switching between ECC modes (level 1, 2, or 3) does not require re-initialization.

Read Bubble Data

The Read Bubble Data command causes data to be read from the MBMs and into the BMC's FIFO. Immediately before the Read Bubble Data command is issued, the host computer must load, with the exception of the Utility Register, all of the parametric registers. Specifically, the following parametric information must be loaded prior to command execution:

- The channel and "number of pages to be transferred" in the Block Length Register must be set to define the page size (number of FSA channels) and number of pages to be transferred.
- The appropriate bits must be set in the Enable Register to select the transfer mode (DMA or non-DMA) and interrupt sources; if error correction is to be used (i.e., if the FSA bootloop registers have been initialized for error correction), the level of error correction must be selected.
- The MBM select and starting address fields in the Address Register must be set to define the (first) MBM and page within the MBM where the transfer is to occur.

Write Bubble Data

The Write Bubble Data command causes data read from the BMC's FIFO to be written into the MBMs. Immediately prior to issuing the Write Bubble Data command, the host computer must load the Block Length, Enable, and Address Registers as described for the Read Bubble Data command. Data should not be loaded into the BMC FIFO until after the command has been issued. Note that a write data transfer does not begin until at least two bytes of data have been loaded into the BMC FIFO.

Read Seek

The Read Seek command is used to reduce system access time of a subsequent Read Bubble Data command by positioning the page to be read at the selected bubble's output track. Note that although the Read Seek command does not cause any data to be transferred, the following information must be loaded into the parametric registers before the command is issued:

- The channel field in the Block Length Register must specify the page size (number of FSA channels).
- The error correction bits in the Enable Register must be set identical to the values used when the system was initialized.
- The MBM select field in the Address Register must be set to select the bubble containing the page to be read and the address field must specify a page number that is one less than the (first) page to be read by the subsequent Read Bubble Data command.

Write Seek

The Write Seek command is used to reduce the system access time for a subsequent Write Bubble command by positioning the page to be written at the selected bubble's input track. Note that like the Read Seek command, the Write Seek command does not cause any data to be transferred. Similarly, the following information is issued:

- The channel field in the Block Length Register must specify the page size (number of FSA channels).
- The error correction bits in the Enable Register must be set identical to the values when the system was initialized.
- The MBM select field in the Address Register must be set to select the bubble containing the page to be written and the address field must specify a page number that is one less than the (first) page to be written by the subsequent Write Bubble Data command.

MBM Purge

The MBM Purge command is used in place of the Initialize command when the bootloop register is to be loaded from an external source (once a bootloop has been identified, the bootloop register pattern can be maintained by the host and loaded directly from external memory with a Write Bootloop Register or Write Bootloop Register Masked command to conserve power and increase speed during an initialization sequence). The MBM Purge command clears the BMC's internal registers and the address field of the Address Register. The MBM select field in the Address Register and the other parametric registers are not cleared. Like the Abort command, the MBM Purge command does not require the loading of the parametric registers prior to command execution.

Read Corrected Data

The Read Corrected Data command is used only when error correction is implemented and only is applicable when error correction level 2 or 3 is selected (the Read Corrected Data command is issued automatically by the BMC when error correction level 1 is selected). When an FSA reports a correctable error, the Read Corrected Data command is issued to cause the corrected page in the FSA(s) to be read into the BMC's FIFO. Note that since the parametric registers previously were loaded for the read operation in which the error was detected and since the address field is not incremented (i.e., the address of the page in error is in the address field), it is not necessary to load the parametric registers prior to issuing the Read Corrected Data command.

Reset FIFO

The Reset FIFO command causes the BMC's FIFO (and its input and output latches) to be cleared. Normally, this command is issued prior to issuing the Write Bootloop, Write Bootloop Register, or Write Bootloop Register Masked commands to ensure that the FIFO will accept 40 bytes; the Reset FIFO command does not require loading of the parametric registers.

Software Reset

The Software Reset command clears the BMC's internal registers, and the terminal count field in the Block Length Register and starting address field in the Address Register. Additionally, the Software Reset command causes the BMC to clear the status register of every FSA channel in the system. Since the Software Reset command does not clear the FSA channel and MBM select fields in the Block Length and Address Registers or any of the Enable Register bits, it is not necessary to reinitialize the parametric registers following a Software Reset command, and no loading of the parametric registers is required prior to command execution.

Read FSA Status

The Read FSA Status command causes the BMC to read the 8-bit status register of each FSA channel in the system (the number of FSA channels specified in the channel field of the Block Length Register). This command is issued to determine the number of FSAs in a system. Following command execution, the individual status register bytes will be in the BMC's FIFO in ascending order; one byte per FSA channel. All values returned to the host will be 20H or 28H if error correction is enabled. This occurs because the FSA status is cleared when read. The BMC always reads the FSA status prior to interrupting (or informing) the host of an error. Therefore, the Read FSA status command can only read the "cleared" FSA status values (20H or 28H). No loading of the parametric registers is necessary prior to command execution.

Read Bootloop Register

The Read Bootloop Register command causes the BMC to read the bootloop register of the selected FSA channel into its FIFO. This command is used initially to ensure that the bubble system is communicating properly (bubble-to-FSA and FSA-to-BMC communication established) and is used to transfer bootloop information to the host system for subsequent bootloop initialization from an external source (e.g., user EPROM). Prior to command execution, the channel field in the Block Length Register and the MBM select field in the Address Register must be loaded with the number of FSA channels (normally two) and the corresponding bubble (FSA channel pair) to be selected. Note that since each individual FSA channel's bootloop register contains 20 bytes, reading a pair of FSA channels fills the BMC's 40-byte FIFO; reading the bootloop registers of more than two channels is possible but not recommended since data must be taken from the FIFO to avoid an overflow condition. Also, since the bootloop register data from each FSA channel pair actually is interleaved on a bit-by-bit basis before it is assembled into bytes, reading the bootloop register for a single channel likewise is not recommended. Remember that a unique BMC status value (C1H or C3H) is expected with this command (see "Considerations for Polled Command Execution").

Write Bootloop Register

The Write Bootloop Register command causes the BMC to write the contents of its FIFO into the bootloop register of the selected FSA channel(s). This command (and the Write Bootloop Register Masked command) is used during bubble system initialization when the bootloop is written from an external source rather than from the bubble itself. In order to use the Write Bootloop Register command, the bootloop register data must be loaded into the FIFO prior to command execution and the channel and MBM select fields of the Block Length and Address Registers must be loaded with the number of FSA channels (normally two) and the corresponding bubble (FSA channel pair) to be selected. Recalling that each individual FSA channel's bootloop register contains 20 bytes, 40 bytes normally are written into the FIFO to initialize the two FSA bootloop registers associated with each bubble. However, a 41st byte of zeroes *must* be written to ensure a successful command execution status. Note that the parametric registers should be loaded prior to loading the FIFO.

Proper operation of the bubble system requires that each FSA bootloop register contains either 135 (error correction selected) or 136 (error correction disabled) logic "1" bits that correspond to the 135 or 136 valid data storage loops.

Write Bootloop Register Masked

The Write Bootloop Register Masked command is identical to the Write Bootloop Register command previously described with the exception that the number of "1" bits written automatically is stopped when the required 135 or 136 logic "1" bits have been written for each channel. The parametric registers are loaded as described for the Write Bootloop Register command; the number of logic "1" bits written (135 or 136) is determined by the setting of the error correction bits in the Enable Register.

Write Bootloop

The Write Bootloop command causes the existing contents of the selected bubble's bootloop to be replaced by the 40 bytes of information currently contained in the BMC's FIFO. This command is used only after it has been determined that the existing bootloop is invalid (i.e., a storage loop previously identified as "good" has become defective) and typically is not required for the life of the bubble system. Remember that the parametric registers must be loaded prior to pre-loading the FIFO with the 40 bytes of information to be written followed by a 41st byte of zeroes.

If it should become necessary to use the Write Bootloop command, the channel field in the Block Length Register and the MBM select field in the Address Register must be loaded with the number of FSA channels and the corresponding bubble to be selected. As an additional safeguard, the Write Bootloop Enable bit in the Enable Register additionally must be set (if this bit is not set, command execution immediately will be aborted and the Timing Error bit will be set in the Status Register).

Read Bootloop

The Read Bootloop command causes the BMC to read the bootloop of the selected bubble into its FIFO. This command is used to determine if the existing bootloop is valid by comparing the bubble bootloop information to that on the label of the device. Remember that the parametric registers must be loaded prior to command execution. Also note that a BMC status value of C1H or C3H is expected (see "Considerations for Polled Command Execution").

ERROR CORRECTION

As mentioned earlier in this application note, several factors pertaining to error correction must be understood and weighed according to your specific application before implementing error correction. From a software perspective, the selection of the appropriate level of error correction is based on the host system's requirements relative to error logging and data recovery. Before describing the individual levels of error correction and the associated software requirements, the types of errors that can occur within bubble memories are described as a preface to error correction.

Bubble Errors

To understand the function and implementation of error correction, the differences between bubble errors and semiconductor device errors must be distinguished. In conventional semiconductor memory, errors are classified as either hard (non-recoverable) or soft (recoverable). Usually, hard errors are the result of physical or irreversible damage within the device itself (e.g., oxide breakdown or junction burnout), and soft errors are the result of transient conditions and generally do not reappear when the data is rewritten.

Unlike semiconductor devices, bubble memory devices have no active elements and, as such, rarely experience hard (non-recoverable) errors. Accordingly, all bubble memory errors can be considered as soft errors (for information on irreversible failure mechanisms in bubbles, refer to the Intel 7110 Bubble Memory Reliability Report, RR-36 Order Number 210632). In order to further define the nature of soft errors in bubbles, errors are classified either as "data" errors or "read" errors. A data error occurs when a data inversion occurs within a storage loop; the data is lost, but since no physical damage has occurred, the data can be rewritten and the bubble can remain operational. Data errors typically occur when the bubble is operated beyond its safe operating region and, as such, are seldom encountered during normal operation. The most common type of soft error is a "read" error that occurs during a bubble read operation, usually as a result of noise in the detection/sense circuitry. Since the data in the storage loop is unaltered during a bubble read, the data can be recovered by simply repeating the read operation.

Error Detection/Correction Capability

In respect to the Formatter/Sense Amplifier (FSA), errors either are correctable (the FSA is able to reconstruct the data using an error correction algorithm before the data is transferred to the BMC) or uncorrectable, irrespective of the type of error (data error or read error). The error correction code used by the FSA is a 14-bit Fire code that is appended to each 256-bit block of data. This code is capable of correcting all single error bursts up to, and including, five bits in length and has proven to be well suited to the error model for the 7110 MBM. Table 7 outlines the FSA's error correction capability and probability of page errors; the bit error rate can be obtained by dividing the "probability of page in error" by the number of bits per page.

ECC Options

The FSA's error correction circuitry (ECC), in conjunction with the BMC, provides three levels of error correction. Each level places unique demands on the host system that range from simple data recovery to the logging of specific pages in which the error occurs. The desired level of error correction is selected by the setting of the appropriate bit or bits within the BMC's Enable Register. Table 8 defines the relevant Enable Register bits for the three levels of error correction available.

Table 7. FSA Error Detection/Correction Capability

Type of Error	Approximate Probability of Page in Error	% Correction	% Detection
Single Read Error	10^{-6}	100	—
Single Data Error	10^{-7}	100	—
Random Double Read Error	10^{-12}	—	100
Read Error Burst Length 2	—	100	—
Data Error Burst Length 2	—	100	—
Read Error Burst Length 3/4	10^{9}	100	—
Random Double Data Error	10^{14}	—	100
Read Error Burst Length 5	—	100	—
Random Triple Read Error	10^{17}	—	100
Single Soft + Read Burst 2	—	—	100
Undetected/Uncorrected Error Escape Rate	10^{13}	—	—

NOTES:

1. Read errors are recoverable by retry or error correction.
2. Data errors are recoverable by error correction methods only.

Table 8. Error Correction Level Selection

Error Correction Level	Enable Register Bit		
	Bit 6 ICD	Bit 5 RCD	Bit 1 Interrupt Enable (Error)
Level 1	0	1	0
Level 2	1	0	0
Level 3	1	0	1

In typical bubble memory systems, the prevention of the erroneous transfer of data to the host is primary concern, and the actual location of the error (error logging by page) is secondary. This fact is especially true if the error is correctable. Both the prevention of transferring erroneous data and error logging of the page in error, however, can be satisfied by error correction, and you must select the error correction level that best fits your specific application.

ECC Operation

When error correction is implemented, each page of data written to the MBM contains additional bits for the ECC code. For example, in a single-bubble system like the BPK 72, the page size decreases from 68 bytes to 64 bytes per page when error correction is implemented. As each page of data is read from the MBM and assembled into the FSA, the FSA's error correction circuitry checks the integrity of the data. When an error is detected (in a completely assembled page), the FSA notifies the BMC (by activating its ERR.FLG/ line) and sets the appropriate bits within its internal status register according to whether or not the error is correctable. The BMC, depending on the level of error correction selected and the nature of the error (correctable or uncorrectable), either issues a command to the FSA to continue the transfer (transparent to the host) or interrupts the host and waits for additional instructions before proceeding.

The point at which the data is transferred between the FSA and the BMC's FIFO is the key to understanding the difference among the three levels of error correction. Once this timing is understood, it will become easier to see how the levels differ in terms of the interrupts generated and the intervention required by the host (software driver) during the transfer. Note that while the BMC generates an interrupt when an error is detected, it is not mandatory for the host to support interrupts; the host can read the BMC's Status Register at the completion of command execution to determine the nature of the error condition.

During a bubble read operation with error correction enabled, the FSA senses and formats the data and places the data in its two 270-bit serial FIFOs (one FIFO for each channel). The data is read in the fully buffered mode where a full page (512 bits) plus the additional 28 ECC bits are read into the two FIFOs before any data is transferred to the BMC's FIFO. By fully buffering the data, the FSA can detect an error and notify the BMC before any data is transferred. If no error is detected by the FSA, the contents of the FSA's FIFOs are written to the BMC's FIFO while the next page from the MBM is being read into the FSA (note that the 28-bits of ECC code are never transferred to the BMC's FIFO). In order to correct an error once it has been detected, the FSA cycles the data through its error correction network. Once the data is cycled, the "corrected" data is either automatically transferred to the BMC (transparent to the host) or the data is held in the FSA FIFOs awaiting further instruction from the host. As an alternative to using the "corrected" data (error correction level dependent), the page in error could simply be reread when an error was reported since a majority of the errors encountered are "read" errors. The following paragraphs describe ECC operation during each of the three levels of error correction.

Level 1

Level 1 is the minimum level of error correction and is used only when the host system is concerned with maintaining bubble data integrity. As an example of this type of application, consider a bubble-based operating system that is downloaded into user RAM for execution. With this application, the primary concern is that the entire operating system is transferred correctly rather than where the error occurred. As will be seen, Level 1 is well suited to applications where go/no-go types of data transfers are required.

If an error is detected during Level 1 operation, the FSA activates its ERR.FLG/ line to the BMC. The BMC, in response, automatically issues a Read Corrected Data (RCD) command to the FSA which causes the FSA to cycle the data through its ECC network and update its status register, and then to transfer the data to the BMC. If the soft error was correctable, valid data would have been transferred to the BMC; the BMC would increment its Address Register to the next page to allow the operation to continue, and an interrupt would not be generated (i.e., the entire operation would be transparent to the host system). However, if the soft error was beyond the capability of the FSA's error correction circuitry (i.e., an uncorrectable error), invalid data would have been transferred to the BMC. The BMC, after reading the "uncorrectable error" status from the FSA, stops command execution and interrupts the host, and does not increment its Address Register. Since the host is aware that an uncorrectable error has occurred, the proper response is to repeat the entire read operation (by reloading the parametric register and reissuing the read command) since the erroneous page already has been transferred to the host.

Note that since the BMC does not increment its Address Register after reading the "uncorrectable error" status from the FSA, it is possible to perform page-specific logging of uncorrectable errors.

Level 2

Level 2 differs from Level 1 in that the transfer of erroneous (uncorrectable) data to the BMC is prevented and successive retries of the page in error are possible since the Address Register is not incremented. As with Level 1, correctable errors are transparent to the host system.

When a soft error is detected during Level 2 operation, the BMC automatically issues an Internally Corrected Data (ICD) command to the FSA.

In response to this command, the FSA cycles the data through its error correction network and updates its status register to indicate if the error is correctable or uncorrectable, but does not transfer the data to the BMC. The BMC, in turn, reads the FSA's status register and updates its own Status Register. If the error is correctable, the BMC automatically issues an RCD command to the FSA (to transfer the corrected data) and increments its Address Register to the next page address to allow the read operation to continue. Like Level 1 operation, the transfer of the corrected page is transparent to the host; the subtle difference between Level 1 and Level 2 is the time required to execute since the data is cycled through the error correction network twice (first by the ICD command and again by the RCD command). While the second correction cycle does not change the data, it does, however add approximately 350 milliseconds to the transfer operation.

If the soft error is uncorrectable, command execution is halted on the page in error and the BMC interrupts the host system. When interrupted, the host system can read the BMC's Address Register to determine the page in error and can issue a subsequent Read command (without reloading the parametric register) to retry the page. If, when the page is reread, the error does not recur (i.e., if the uncorrectable error is a "read" error), command execution continues with the next page. If successive retrys are unsuccessful (i.e., if the uncorrectable error is a "data" error), the page most likely will have to be rewritten. The host system, however, can examine the erroneous page by issuing an RCD command to the BMC to transfer the uncorrectable data from the FSA. Following the transfer of the erroneous page, the BMC again will interrupt the host and will not increment its Address Register. Note that the uncorrectable data transferred to the BMC will not necessarily match the erroneous data originally read from the MBM since the FSA's error correction circuitry attempts to correct the data.

Level 3

Level 3 offers the most complete means of error handling and, at the same time, is the most software intensive level since the host system is interrupted when either a correctable or uncorrectable error is encountered. Accordingly, error logging may be performed on pages containing correctable as well as uncorrectable errors, and an unlimited number of retrys can be attempted on the erroneous page. As with Level 2, the transfer of erroneous data to the BMC can be prevented.

When a soft error is detected during Level 3 operation, the BMC automatically issues an ICD command to the FSA. Like Level 2 operation, the FSA cycles the data through its error correction network and updates its status register, but does not transfer the data to the BMC. The BMC, in turn, reads the FSA status register, updates its own Status Register, and interrupts the host system even if the error is correctable. When interrupted, the host system must examine the BMC's Status Register to determine if the error is correctable or uncorrectable and can log the address of the page in error by reading the BMC's address register (the Address Register is not incremented). Note that it is not necessary for the host to support interrupts as the host can continuously poll the BMC's Status Register for an error interrupt.

If the error is correctable, the host system can either issue an RCD command (to transfer the corrected data from the FSA to the BMC) or can retry the page by issuing subsequent Read commands. Note that by retrying pages with correctable errors, the host system can distinguish between "read" and "data" errors since consistently correctable errors on a page indicate a "data" error (a "read" error would not recur when the page was reread).

If the error is uncorrectable, the host system only would retry the erroneous page by issuing additional Read commands and, if successive retrys were unsuccessful, would have to rewrite the page. As with Level 2 operation, the host system can issue an RCD command to transfer the uncorrectable data from the FSA.

Status Register

When using error correction, the host system can read the BMC's Status Register at the time of the interrupt or at the completion of command execution to ascertain additional information relating to the error condition. The relevant bits of the STR are bits 6, 5, 3 and 2; the remaining bits are irrelevant to error correction.

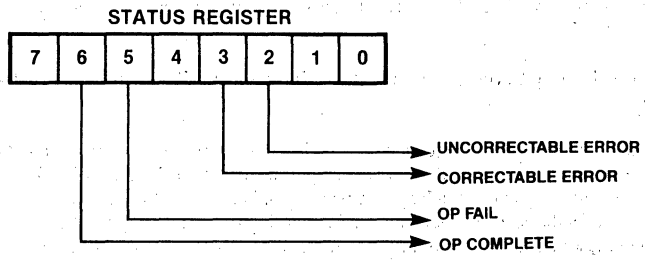


Table 9 lists the possible status indications that could occur when using error correction.

Table 9. Error Correction Status Indications

STR Bit Set	Error Correction Level	Indication
3	3	Correctable error detected, address of page containing the error is in the AR.
5 and 2	All	Uncorrectable error detected, address of page containing the error is in the AR.
6	All	Operation complete, no errors detected.
6 and 3	1 and 2	Operation complete, one or more correctable errors detected and corrected.
6 and 2	1	Operation complete, uncorrectable error detected on last page of multipage transfer.
5, 3 and 2*	1 and 2**	Very Rare. During a multipage transfer, one or more correctable errors detected and corrected on one or more pages, and a subsequent uncorrectable error detected on a page other than the last page of the specified transfer.
6, 3 and 2*	1	Extremely rare. During a multipage transfer, one or more correctable errors detected and corrected on one or more pages, and a subsequent uncorrectable error detected on the last page of the specified transfer.

*These status indications may occur on single-page transfers in Multi-MBM systems when one MBM has a correctable error and another MBM has an uncorrectable error.

**This status indication may occur in level 3 in Multi-MBM systems when more than one MBM has an error on the same page and at least one of the errors is correctable and one of the errors is uncorrectable.

DRIVER DESIGN

This section contains specific software interface examples that outline important considerations associated with the various ways in which a bubble system can be operated. As will be explained, command execution can be performed either in an interrupt driven mode or in a polled mode irrespective of the data transfer mode (polled, interrupt-driven, or DMA) selected to effectively provide the six unique operating modes shown in Figure 2. Since both polled and interrupt driven command execution are common to all three data transfer modes, details concerning command execution are discussed prior to examining specific examples of each data transfer mode. Before you begin to design your software driver, it is recommended that you thoroughly understand the information presented in this section.

Command Execution

To better understand the software driver's responsibilities, it is helpful to separate command execution into two phases; a command phase and a result phase. During the command phase, the driver generally (command dependent) loads the parametric registers, issues the desired command, then verifies that the command has been accepted; during the result phase, the driver determines the success or failure of the command issued either by polling the BMC's Status Register (polled mode) or through an interrupt service routine (interrupt-driven).

- Polled Mode. After loading the parametric registers (if required), the software driver issues the desired command, checks to see if the command was accepted, and then continuously polls the BMC's Status Register for an Op Complete indication.
- Interrupt-Driven Mode. After loading the parametric registers (if required), the software issues the desired command, checks to see if the command was accepted, and then waits for the interrupt to occur at successful command completion. Note that the interrupt Enable (Normal) bit must be set in the Enable Register to allow the BMC to generate the interrupt at successful command completion. Also, since an interrupt is not generated if the intended operation fails, additional provisions must be included to avoid "hanging" the system while waiting for the command completion interrupt. Error interrupts are reported according to the settings of the three "error correction" bits in the Enable Register.

Another important fact to note is that the BMC does not support the typical two-way handshaking common to most interrupt-driven peripheral devices and that interrupts from the BMC are cleared (or acknowledged) either when a subsequent command is issued or when the Register Address Counter (RAC) is written with the modifier bit (bit D5) set to "one" and with bits D3 through D0 set to "zero"; interrupts are not cleared by reading the BMC's Status Register.

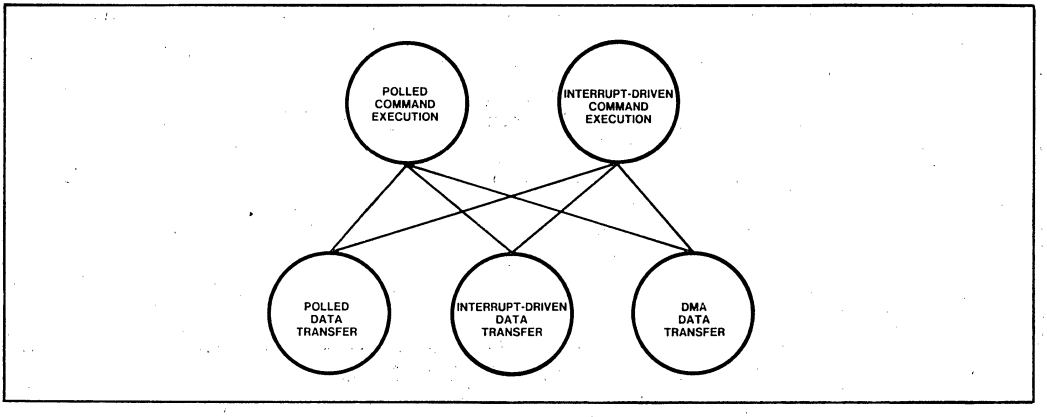


Figure 2. Bubble System Operating Modes

Considerations For Polled Command Execution

The operation of the BMC (and in particular, the BMC's Status Register) during polled command execution is the key to software driver development. Figure 3 shows the activity of the BMC at various stages during a typical command execution sequence. The disussion of the BMC's Status Register is centered around the status bits associated with command execution (i.e., the BUSY, OP COMPLETE, and OP FAIL status bits) although as will be explained, additional Status Register bits can have an effect on the state of the BUSY bit.

Before the command is sent at time T0, the BUSY bit is clear and, in fact, must be clear in order for the BMC to accept a new command (other than an Abort command). Sometime between T0 and T1, the BUSY bit is set to indicate that the command has been accepted and that command execution has begun. Note that all software examples in the remainder of this chapter include a check to ensure that the BUSY bit has been set after a command has been issued.

If the Status Register is examined between T1 and T2, the status byte value will be 80H (BUSY bit set). Depending on the command issued, the FIFO AVAILABLE bit may set at T2 if it is necessary for the host to initiate the transfer of data (see "polled data transfers" later in this section). Whether or not the FIFO AVAILABLE bit is set, the only normal status byte values between T2 and Tn-1 are 80H or 81H (BUSY and FIFO AVAILABLE).

At time Tn (completion of the command), the Status Register indicates the success or failure of the command. Generally, the BUSY bit is cleared at this time (indicating that command execution has terminated); an exception occurs if the FIFO still contains 22 or more bytes of data at command completion (i.e., when the BMC completes its internal microcode routine). During execution of read commands, it is possible for the host to leave data within the FIFO, and if 22 or more bytes are remaining, the BUSY bit will remain set even though command execution has been completed.

If the command is executed successfully and if the FIFO is empty, the status byte value will be either 40H (OP COMPLETE) or 42H (OP COMPLETE and PARITY ERROR). If data was loaded into the FIFO during the interval between T2 and Tn-2, the possible status byte values will be:

41H or 43H if the FIFO contains less than 22 bytes
C1H or C3H if the FIFO contains 22 or more bytes

In the case of unsuccessful command execution, several additional status byte values again are possible depending on the command being executed. Generally, the OP FAIL bit is set in the Status Register along with additional bits that indicate the nature of the failure (e.g., TIMING ERROR or PARITY ERROR). As with successful command execution, the BUSY bit is cleared when command execution is completed unless the FIFO still contains 22 or more bytes.

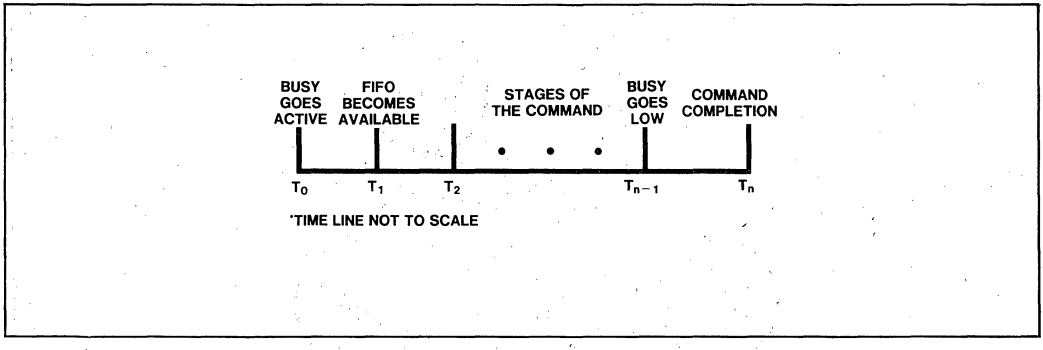


Figure 3. Command Execution Stages

Considerations For Interrupt-Driven Command Execution

As previously mentioned, when the Interrupt Enable (Normal) bit is set, an interrupt occurs only when a command is executed successfully as indicated by the setting of the Op Complete bit in the Status Register. If the Interrupt Enable (Error) bit is not set in the Enable Register and an error occurs that causes the operation to fail, an interrupt will not be generated. The user system must take this fact into account and provide either a fail-safe timer in hardware or a timeout counter in system software to guard against the possibility of "hanging" the system while waiting for an interrupt.

The following possible conditions are applicable to interrupt-driven operation:

1. After issuing a command, the host processor halts and waits for the interrupt to occur. Under this condition, the system will hang if the expected interrupt never occurs. A fail-safe timer connected to a low-level input of an interrupt controller (e.g., an Intel 8259A) would eliminate this problem.
2. In a "true" interrupt driven system where the host processor performs other tasks while the bubble command is being executed, the fail-safe timer in hardware again would be used although it is possible to use a software timeout counter. When operating in the non-DMA data transfer mode (polled or interrupt-driven data transfers), an interrupt service routine would be used to complete the data transfer and then either would wait for the command interrupt or would continuously poll the Status Register until command execution was successfully completed or until a software counter timed out (indicating unsuccessful command execution). However, devoting an excessive amount of time to polling by the host defeats the intention of an interrupt-driven system and reaffirms the need to provide a "watchdog" timer in hardware. In the DMA data transfer mode, the logical approach to detecting when execution of a command has failed is to provide a fail-safe timer in hardware that generates a timeout interrupt to the host.

An additional consideration common to all three data transfer modes is the fact that when an interrupt occurs in response to the successful execution of a command, data still may be present in the BMC's FIFO. If the interrupt service routine immediately acknowledges receipt of the command completion interrupt at the source (i.e., at the BMC), any data remaining in the BMC's FIFO when the interrupt is cleared would be destroyed. To eliminate this potential problem, an intermediate level of interrupt control would be required (again an Intel 8259A) to allow the interrupt routine to complete the transfer. The details described in the "Interrupt-Driven Data Transfer Mode" section will help to clarify the interrupt service routine requirements necessary to avoid leaving any data in the FIFO at command completion.

Data Transfer Modes

As mentioned at the beginning of this section, both polled and interrupt driven command execution can support any of the three data transfer modes. Regardless of the data transfer mode used, the basic operation of the BMC is the same for each data transfer mode. During all data transfers (i.e., during execution of a Read Bubble Data or Write Bubble Data command), the BMC continues to transfer pages of data until the page count in the Block Length Register is satisfied. Since the BMC's FIFO cannot hold a complete page of data, the host processor is required to "keep up" with the BMC as it continues to read data from, or to write data to, the MBM until all data has been transferred. At the beginning of read/write command execution, a "seek" operation is performed to locate the designated page. Once the addressed page is located within the MBM, the read data transfer begins (a write data transfer does not begin until at least two bytes of data have been loaded into the FIFO).

Polled Data Transfers

The polled data transfer mode is the most time-consuming in terms of processor overhead while at the same time is the easiest mode to implement. To support the polled data transfer mode, it is essential that the Status Register bit definitions be clearly understood and, in particular, the function of the FIFO AVAILABLE bit, since the interpretation of the bit changes according to the direction of the transfer.

From an operational viewpoint, the FIFO AVAILABLE bit acts as a gate for the FIFO-handling software. During a bubble write operation, if the FIFO AVAILABLE bit is set, there is room for additional data, and if the FIFO AVAILABLE bit is clear, the FIFO is full. During a bubble read operation, if the FIFO AVAILABLE bit is set, data has been placed in the FIFO by the BMC, and if the bit is clear, the FIFO is empty.

The BUSY bit indicates when the controller is in the process of executing a command. As previously described, the BUSY bit is set within a few microseconds following receipt of the command and remains set until the operation is completed (successfully or unsuccessfully). Remember that since the BUSY bit is internally gated with the BMC's DRQ output signal, it is possible during a read operation to obtain such logically exclusive status indications as BUSY and OP COMPLETE if the host fails to empty the FIFO following comand execution (during non-DMA data transfers, the DRQ signal acts as a "half full/half empty" flag for the BMC's FIFO).

Later in this section, a basic polled data transfer mode read/write routine is flowcharted. Note that to allow the status byte to be considered valid only after the BUSY bit is cleared and to avoid concurrent setting of the BUSY and OP COMPLETE bits at command completion, a running byte counter is implemented in software to ensure that all bytes have been removed from the FIFO. While the FIFO AVAILABLE bit alone can be used to gate the data to or from the FIFO, a byte counter is required to determine when the specified number of bytes has been transferred. Also note that a FIFO Reset command is sent prior to issuing the Write Bubble Data command as a "safety measure." Although issuing the FIFO Reset command is not mandatory, resetting the FIFO is recommended if there is any doubt regarding the state (emptiness) of the FIFO prior to initiating command execution.

Interrupt-Driven Data Transfers

The interrupt-driven data transfer mode requires less processor overhead than the polled data transfer mode (the polling wait time is eliminated) and also allows the data to be transferred in blocks rather than in individual bytes. The actual data transfer rate is fixed; the benefits of this mode only can be realized if the interrupt service routine is efficient (fast) enough to allow additional time for processing other tasks and is based on the host processor's execution speed. Accordingly, if the interrupt-driven data transfer mode is selected, make sure that the additional hardware and software required provide the desired performance increase.

The interrupt-driven data transfer mode is based on the fact that the BMC's DRQ line doubles as a "FIFO half full/FIFO half empty" indicator when the BMC is not in the DMA Mode. Physically, the DRQ line is connected either directly to the host's interrupt input or to an interrupt controller as the interrupt source for the data transfer. When an interrupt occurs, the host processor transfers a block of data to or from the FIFO in a single burst.

The recommended size of the data block transferred is 22 bytes. This block size was selected because transferring 22 bytes, and not 20 bytes (half of the FIFO), accounts for the two additional bytes held in the BMC's internal FIFO input and output latches while also optimizing the buffering capability of the BMC's FIFO. Since 22-byte block transfers rarely are exact multiples of the total number of bytes transferred, an efficient method must be devised to transfer the last remaining bytes of the transfer. While several methods are possible, the following method ensures that an interrupt is issued to transfer all of the "remainder" bytes (i.e., less than 22 bytes never will be "left" in the FIFO to prevent the DRQ line from going active and initiating the last block transfer).

As an example, assume that a two-page (128 byte) read or write data transfer is to be performed. The initial DRQ interrupt indicates that at least 22 bytes should be transferred. However, the first block transferred would be 18 bytes that correspond to the "remainder" bytes (128 mod 22 is 18). By first transferring 18 bytes, five subsequent interrupts occur (each initiating a 22-byte block transfer) to guarantee that no additional polling of the Status Register is required to transfer any "leftover" bytes. The calculation of the remainder count used for the first execution of the interrupt service routine is the responsibility of user software.

DMA Data Transfer Mode

In terms of a bubble system, direct memory access or DMA is the transfer of data between system memory and the BMC without host processor assistance or intervention. Since host processor involvement with the actual data transfer is not required, the overhead associated with software controlled (non-DMA) data transfers is eliminated.

Overall system performance is the primary factor for considering the DMA data transfer mode. That is, if the overhead associated with the available software-controlled data transfer modes degrades system performance to an unacceptable level, the implementation of DMA may prove to be the ideal solution. However, to support the DMA data transfer mode, additional hardware is required (e.g., a programmable DMA controller).

From a system perspective, the DMA hardware definition will dictate the software requirements. In a fixed system, discrete devices may be configured to perform a majority of the data transfer operations, including memory addressing, transfer direction, and terminal count (i.e., the number of bytes to be transferred).

The implementation of a programmable DMA controller such as an Intel 8257 or 8237 provides additional flexibility by allowing DMA to be performed under program control. Since it is not possible to describe all possible configurations, the BMC's operation during DMA transfers is described followed by a specific software example using an Intel 8257 DMA Controller.

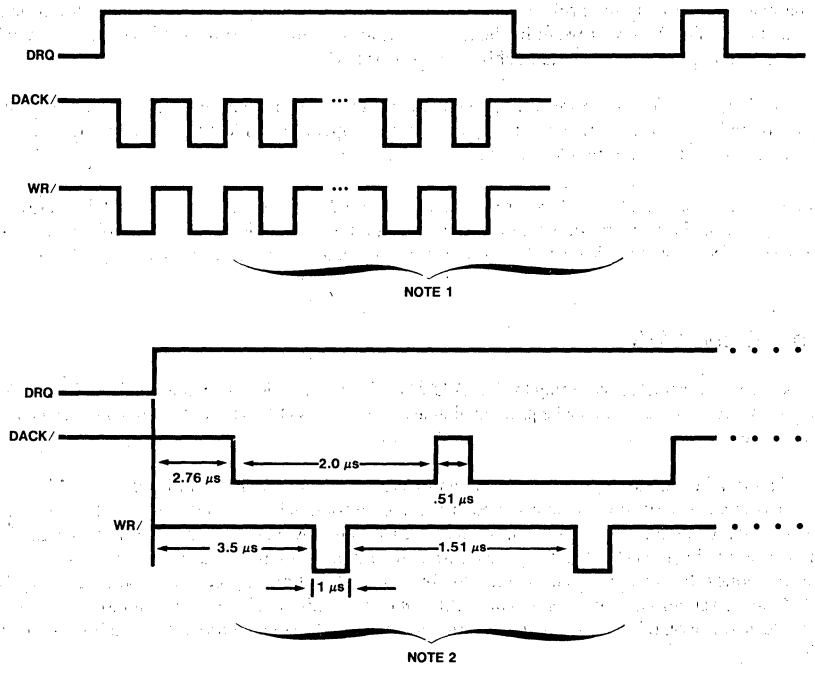
BMC Operation During DMA

The DMA data transfer mode is selected by setting the DMA ENABLE bit in the BMC's Enable Register. In the DMA mode, the BMC supports a standard two-way handshake protocol that uses the BMC's DRQ (Data Request) output signal to interact with the DMA hardware.

To initiate a DMA data transfer (assuming that the DMA hardware has been properly initialized), the BMC's parametric registers are loaded and a read or write command is issued to start the transfer. When the BMC is ready to transfer a data byte to or from system memory, it activates its DRQ output signal. The DMA hardware responds to DRQ by first gaining control of the system bus, then activating the DACK/ signal to the BMC, and finally making a read or write request to the BMC. The BMC responds to the request by placing a data byte on the bus (read operation) or accepting the byte on the bus (write operation). The DRQ signal remains active as long as the BMC either has additional bytes to transfer to system memory (and the FIFO contains at least one byte of data) or expects additional bytes from system memory (and the FIFO contains at least one empty location).

Referring to figure 4, when a Write Bubble Data command is issued, the BMC continuously requests (holds DRQ active) bytes from system memory until the FIFO is filled (i.e., data initially is transferred in a "burst" mode). Once the FIFO is filled, the BMC issues subsequent DRQ requests on a byte-by-byte basis until the last bytes have been transferred. When command execution is complete, the BUSY bit is cleared and the OP COMPLETE bit is set in the Status Register. If the INTERRUPT ENABLE (NORMAL) bit is set in the Enable Register, an interrupt is generated.

When a Read Bubble Data command is issued, the BMC activates DRQ only after the first data byte has been assembled and placed in the FIFO. While the DMA hardware is responding to the DRQ, additional bytes enter the FIFO at a minimum rate of 80 microseconds per byte (the single-MBM system transfer rate; the transfer rate increases according to the number of MBMs operated in parallel). Typical DMA response times usually are fast enough to take each byte before the next byte arrives, and data is transferred on a byte-to-byte basis.



NOTES:
1. THE DATA TRANSFER SEQUENCE INITIALLY OPERATES IN A "BURST" MODE (TYPICALLY 44 TO 46 BYTES ARE TRANSFERRED).
2. THE TIMING SHOWN IS ONE EXAMPLE OF VALUES OBTAINED DURING NORMAL OPERATION. THE VALUES MAY VARY IN YOUR SYSTEM PROVIDED THEY ARE WITHIN SPECIFICATIONS.

Figure 4. DMA Handshake Timing During Read/Write Command Execution

START-UP PROCEDURES

Whenever power is applied to the bubble memory system, a powerfail reset circuit is activated to satisfy specific power/timing sequences required by the BMC. All bubble system designs must include a powerfail reset circuit to ensure proper initialization of the bubble system.

The primary function of the powerfail reset circuit is to ensure that the bubble memory system powers up correctly. A built-in hardware time delay allows the BMC to complete its internal power-up sequence prior to enabling the additional support components; a software time delay is needed before any commands can be issued to the BMC. Following the delay, the first user communication with the BMC must be an Abort command whether the power-up is a cold start (application of power) or a warm restart routine (a RESET/ pulse applied to the 7220-1 BMC). Note that the status register should not be considered valid until the Abort command has been issued. A complete power-up flowchart, including initialization, is shown in figure 5.

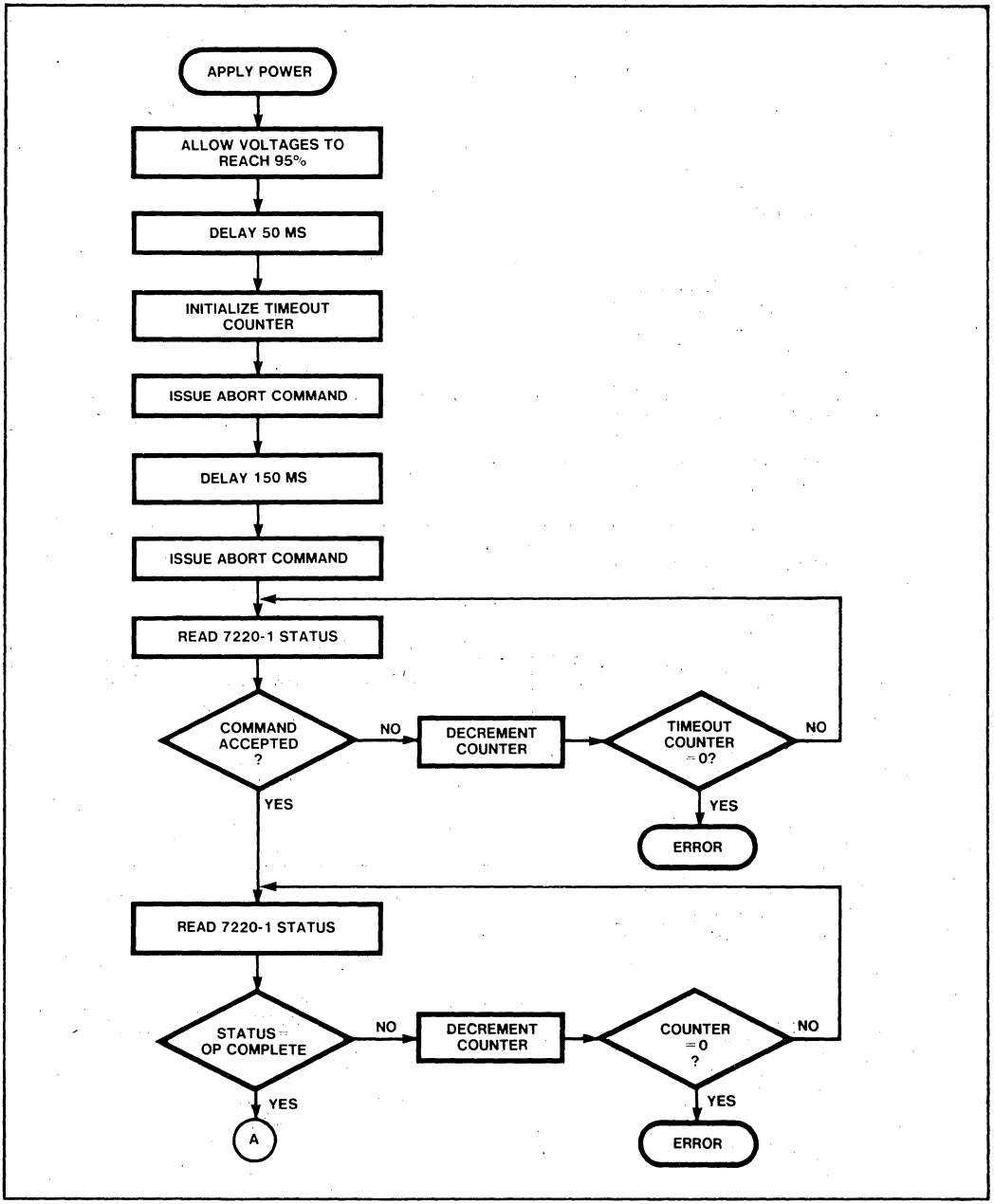


Figure 5. Power-Up Sequence (Polled Command Execution)

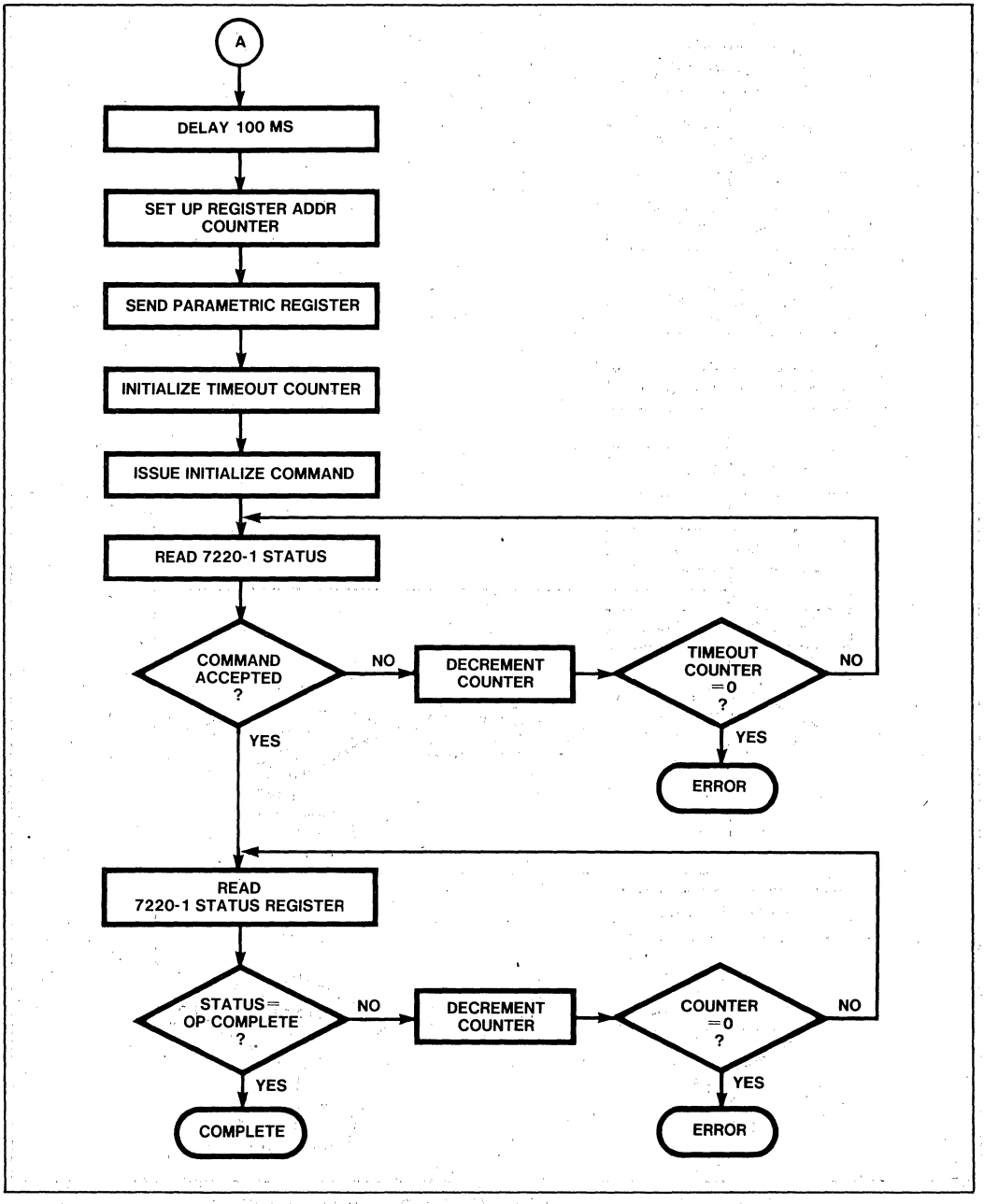


Figure 5. Power-Up Sequence (Polled Command Execution) (Cont.)

Initialization

Following successful execution of an Abort command, the user must initialize the bubble memory system using the Initialize command. The Initialize command requires that the parametric registers first are loaded with specific values. The software flowchart (figure 5) shows one example for polled command execution without error correction for a one-bubble system (i.e., the BPK 72 or iSBX 251). The Block Length Register always must be loaded with the values shown while the MBM Group Select bits in the Address Register always must select the last MBM in the system. For your particular application, set the appropriate bits in the Enable Register prior to issuing the Initialize command. Note that error execution mode changes require re-initialization of the bubble system.

The flowchart example polls the Status Register to see whether or not the initialization was successful. The OP COMPLETE bit indicates success. If an initialization problem occurs, the TIMING ERROR and OP FAIL bits will be set in the Status Register.

In an interrupt-driven system, the interpretation of the INT (interrupt) line depends on the setting of specific bits in the Enable Register. If INTERRUPT ENABLE (NORMAL) is set in the register, on the *successful* completion of command execution, the INT line will activate on the user interface. If INTERRUPT ENABLE (ERROR) is set in the register, the INT line will activate when an error condition occurs (in this case a TIMING ERROR). The user should devise interrupt service routines that can differentiate between the two interrupt sources by polling the Status Register. It is apparent from this discussion that a system that relies solely on interrupts may find it necessary to incorporate a watchdog timer (timeout counter interrupt) depending on the selection of the interrupt enable bits. A timeout counter avoids the possibility of never receiving an interrupt if a command is *unsuccessful* (this condition is possible if only the INTERRUPT ENABLE (NORMAL) bit is set).

If a system uses the BMC's DRQ signal for data transfers (DMA or interrupt mode), special precautions are necessary during Intialization. First, it is recommended that the DMA Enable bit in the Enable Register be disabled (i.e., set to 0) for the Initialize command. Once the Initialize command is successfully completed, the DMA Enable bit is enabled for the subsequent DMA data transfer.

When the BMC is in the non-DMA mode, the DRQ pin acts as a half full/half empty flag for the BMC's FIFO. Since the FIFO is used to temporarily store bootloop data during execution of the Initialize command, the DRQ pin is toggled. User-written software should therefore take appropriate precautions with respect to the DRQ pin during initialization (i.e., mask all interrupts, disable DMA controllers, etc.).

The Initialize command establishes a "working subset" of the bootloop information from the bubble and places this information into the bootloop registers of the FSA. A masking algorithm within the BMC reduces the total of 320 possible good storage loops to a working subset of 270 or 272 good loops (error correction dependent).

If the bootloop information in the MBM is incorrect, the system can inadvertently appear to be initialized correctly. As a one-time check to ensure that the system has been initialized correctly, simply read the bootloop registers from the FSA and count the number of "1's."

The count should yield:

- 270 "1's" out of 320 with error correction implemented.
- 272 "1's" out of 320 without error correction.

NORMAL OPERATION EXAMPLES

The following software flowcharts outline the important considerations and provide examples of each mode of operation previously discussed. First, figures 6 and 7 show the two possible command execution techniques. Note that in each case, the appropriate additional software must be inserted according to the data transfer mode selected. Figures 8 through 10 detail each data transfer mode. For any commands that do not involve the transfer of data (i.e., Abort, FIFO Reset, etc.), no additional software is required to be inserted in the command execution examples.

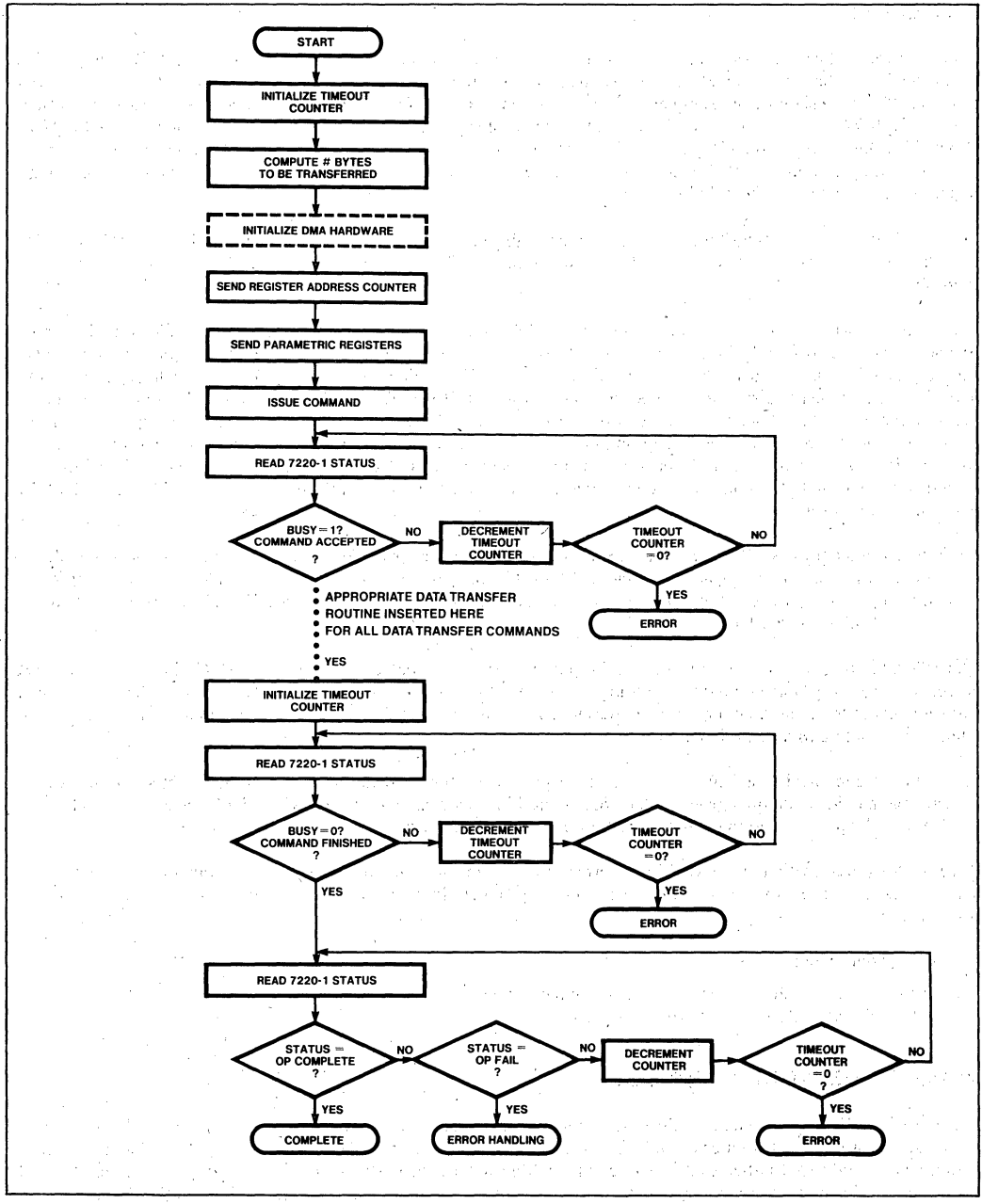


Figure 6. Polled Command Execution Routine

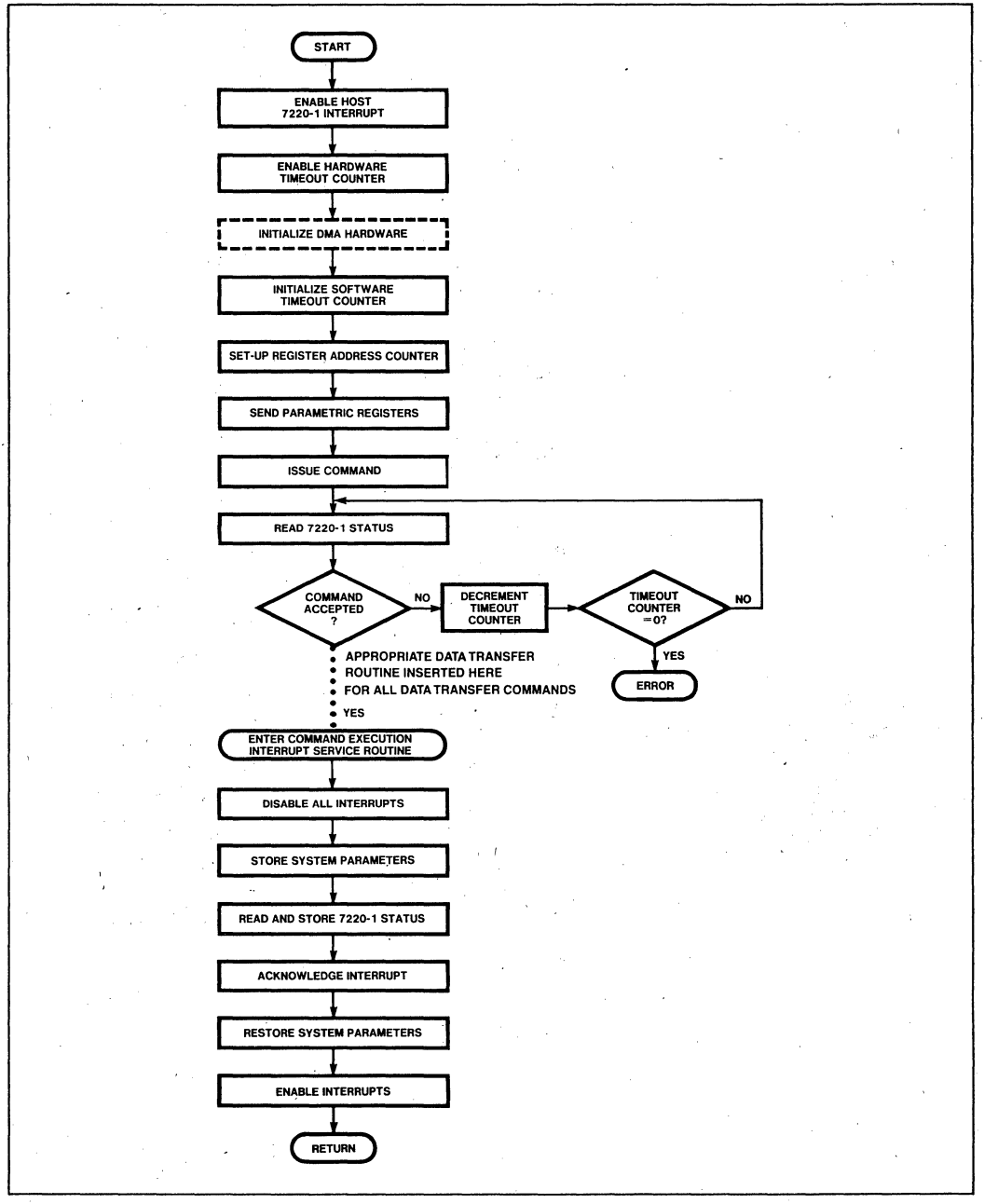


Figure 7. Interrupt-Driven Command Execution Routine

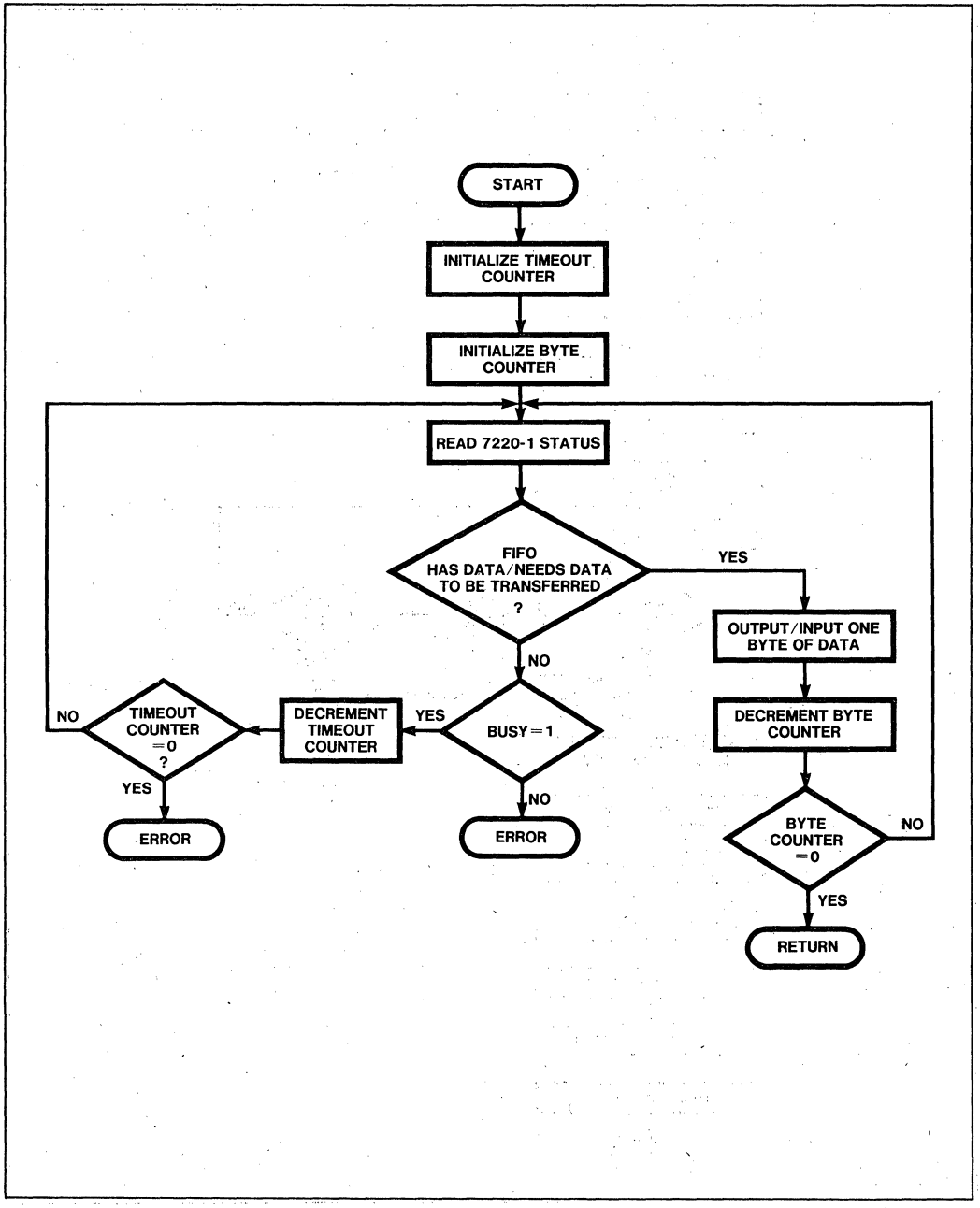


Figure 8. Polled Data Transfer Routine

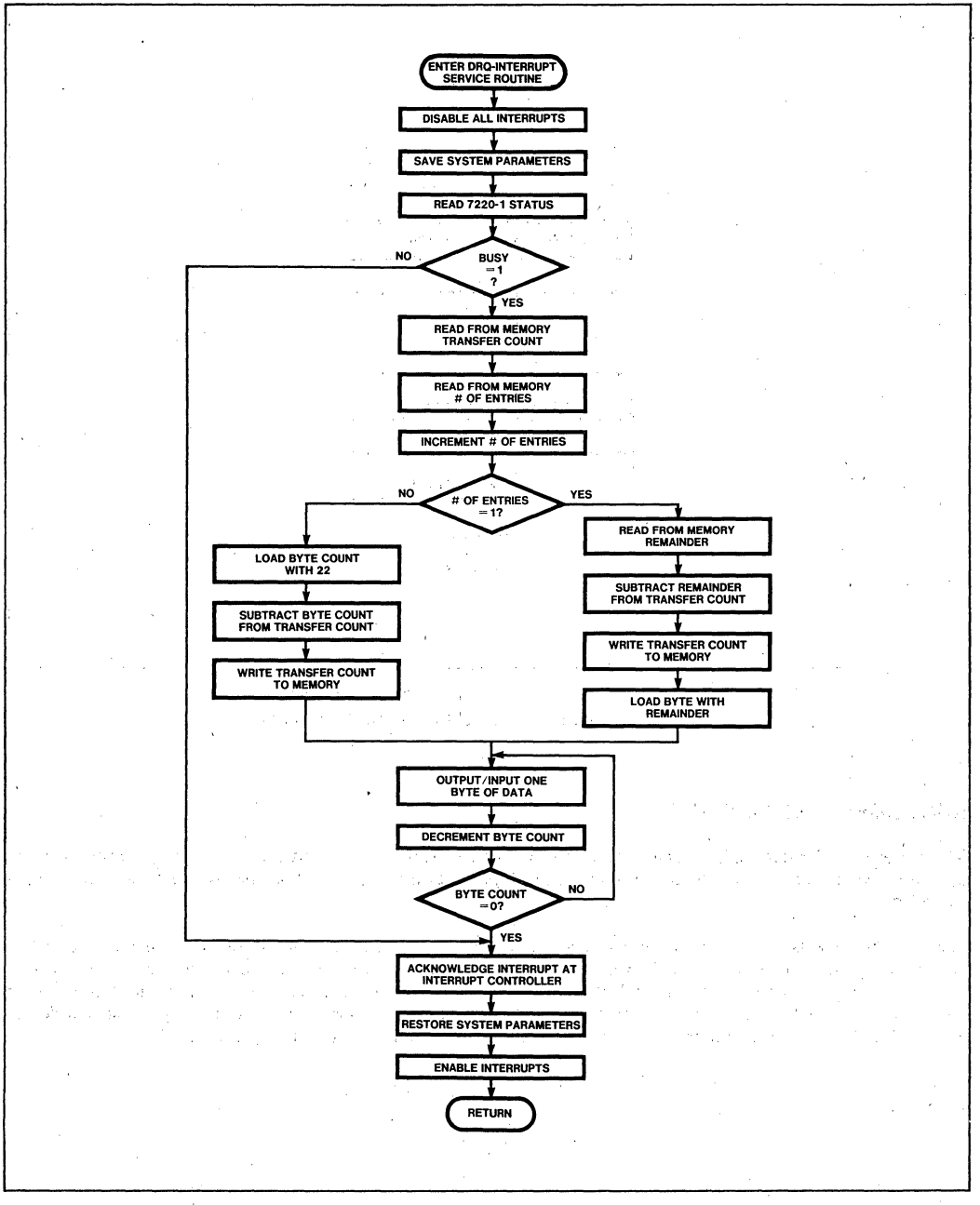


Figure 9. Interrupt-Driven Data Transfer Routine

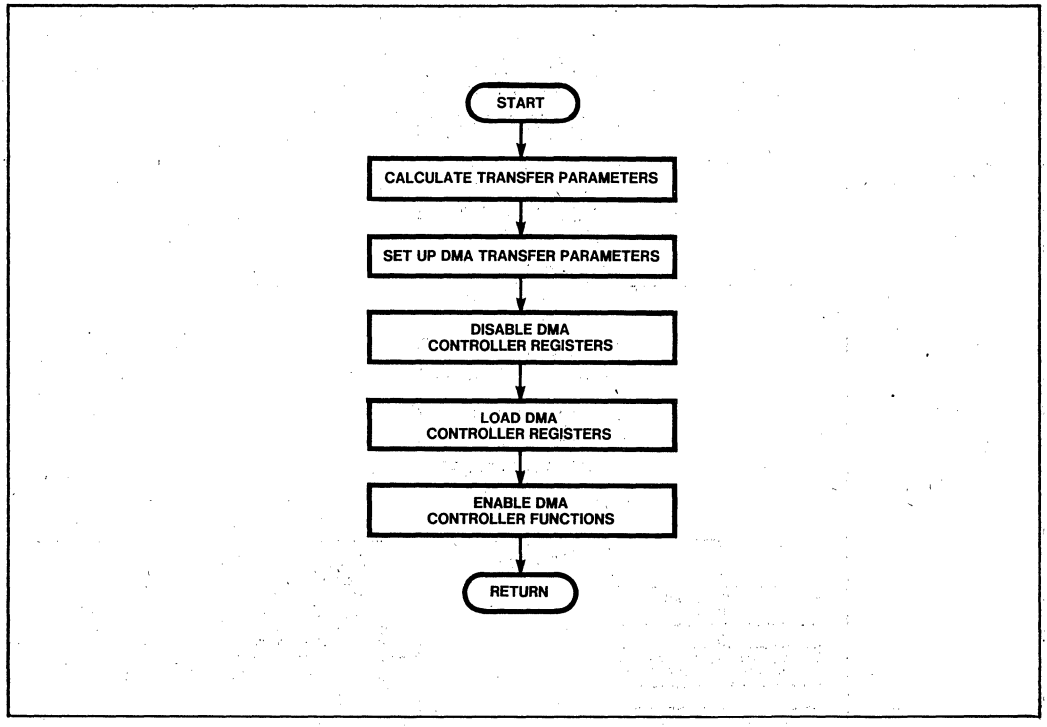


Figure 10. DMA Hardware Initialization Routine (Assumes an Intel 8257 DMA Controller)

SHUT-DOWN PROCEDURES

The power down procedure is the same regardless of whether the user voluntarily powers down the bubble memory system or power is inadvertently lost. The only special precaution is to ensure that the voltage decay rates are not exceeded.

The 7230 Current Pulse Generator contains a special powerfail detection circuit. The purpose of this circuit is to detect when the power supplies fall below threshold levels. Such an event automatically initiates an orderly shutdown of the rotating magnetic field and control signals for the 7110 function generators, in such a manner that no MBM data will be lost or invalidated, provided the voltage decay rates are met. When power is restored, the software implementation details described in the Start-Up procedures should be observed to ensure correct powerfail circuit operation.

intel

APPLICATION NOTE

AP-164

November 1983

Using CMOS Minimizes Bubble Memory Power Consumption

Peggy M. Lammer
Ulmont Smith, Jr.
Applications Engineers
Intel Corporation

ORDER NUMBER: 230826-001

INTRODUCTION

More and more microprocessor systems are becoming portable. Quite frequently, portable systems run on a limited power budget with a battery power supply. Others need to limit how much power they dissipate because their small enclosures dissipate only a limited amount of heat. When designing portables or any piece of processor controlled equipment, consider using bubble memories for mass storage. Bubble memories are solid state, rugged, reliable, and very small; a complete 128 kbyte bubble memory system will occupy less than a 10 cm x 10 cm area of board space. In addition, bubble memories are non-volatile; the memories still maintain their data integrity even if they are powered down to save energy when the processor is not accessing them.

To minimize a low power project's design time, part of the application note is a completely designed CMOS controlled switch that will power-down a bubble system when it is not being accessed. The switch is inexpensive and can reduce your standby power dissipation up to 99%.

OVERVIEW

This application note provides information on low power techniques for bubble memories. All the techniques can be incorporated into your existing bubble memory system with very little effort or expense. A large part of the note focuses on power switching because it is easy to add the extra hardware, and power switching offers the greatest amount of average power savings. For the moment you should know that power switching is supplying the bubble with power only when it is interacting with the host processor and removing the bubble system's power supply when the bubble is not in use.

There are two main parts to the application note. The first will explain why average power dissipation will vary with the frequency of bubble activity. Reading this section will give you an idea of how much power can be saved compared to your present operation.

The second part starts by covering some hardware considerations for power switching bubble memories. It then describes a power switching circuit designed to these considerations. In addition, this section discusses some software techniques and a secondary hardware technique called detector switching that will further reduce average power dissipation.

MEMORY ACTIVITY LEVELS AND POWER DISSIPATION

The bubble memory already reduces its power dissipation by only creating magnetic fields to move the bubbles when data is being accessed. When data is not being transferred, the magnetic fields need not be present, i.e., the coil drive currents which create the fields are removed and less power is dissipated. Table 1 is a break down of power dissipation by component for the Intel one bubble memory system in Figure 1. A system consists of a 7220 bubble memory controller and up to eight BPK-70 memory cells; a cell is a bubble memory and its support ICs.

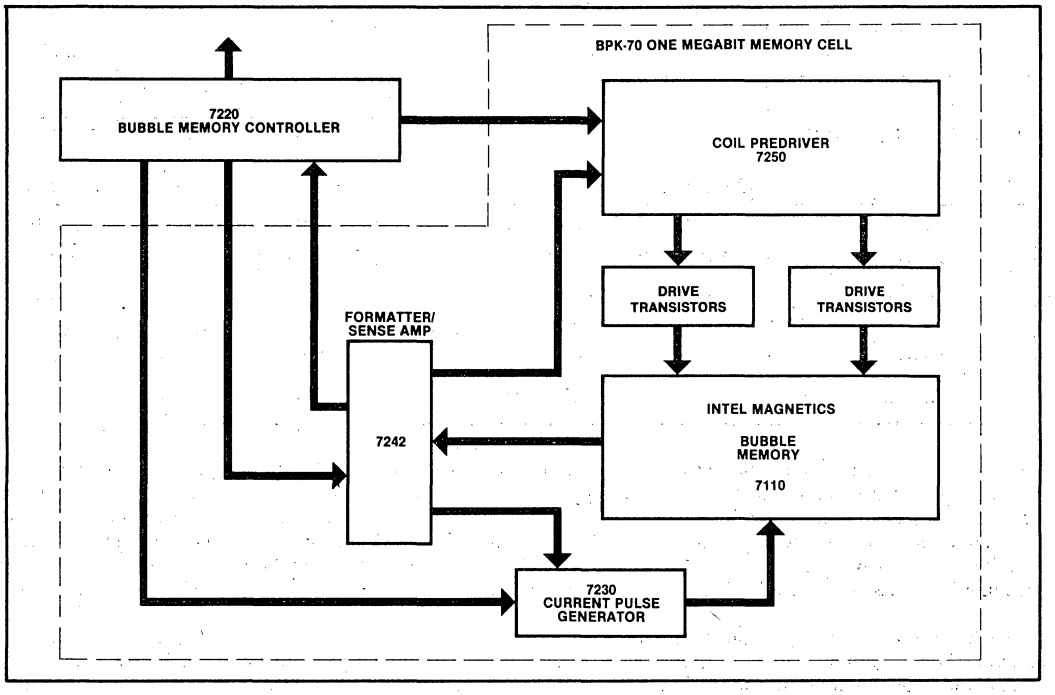


Figure 1. Bubble Memory System Block Diagram

Table 1. Bubble Memory Power Requirements

Configuration		Power (Watts)					
BPK70	Capacity (Bytes)	+5V (Maximum)	+12V (Maximum)	Total Active (Maximum)	Total Active (Typical)	Total Standby (Maximum)	Total Standby (Typical)
1	128K	1.92	4.80	6.72	3.90	3.03	1.55
2	256K	2.79	9.60	12.39	7.30	4.57	2.60
Breakdown by Device		**Power (Watts)**					
		+5V (Maximum)	+12V (Maximum)	Total Active (Maximum)	Total Active (Typical)	Total Standby (Maximum)	Total Standby (Typical)
7110		0	1.740	1.740	1.480	0.440	0.290
7220		1.050	0	1.050	0.500	1.050	0.500
7230		0.235	0.440	0.675	0.390	0.475	0.225
7242		0.630	0.375	1.005	0.500	1.005	0.500
7250		0	0.945	0.945	0.480	0.060	0.030
7254(2)		0	1.300	1.300	0.550	0	0

Without a power switch, a bubble system not being accessed (in standby mode) will dissipate an average of 1.55 watts, 3.0 W worst case. When the host processor does access the memory, (i.e. the coils have a current running through them) the power dissipation increases to 3.9 W typically, 6.7 W worst case. Obviously, since the bubble system does nothing but dissipate 1.55 W in standby mode, it would be ideal to shut it off and dissipate zero watts. That is the purpose of a power switch. For example, the CMOS controlled switch in this note reduces standby power consumption to less than 25 mW. That is a 99% decrease in standby power consumption compared to worst case.

Figure 2 is a graph showing memory activity levels versus average power dissipation. On the graph, the average amount of time the bubble is active is represented as a percentage of duty cycle. Duty cycle is a ratio of the amount of time the bubble is active to the total time spent in standby and active modes. There are three types of systems plotted on the graph. Case 1 is worst case bubble memory systems. Case 2 represents typical systems and case 3 is typical power switched systems.

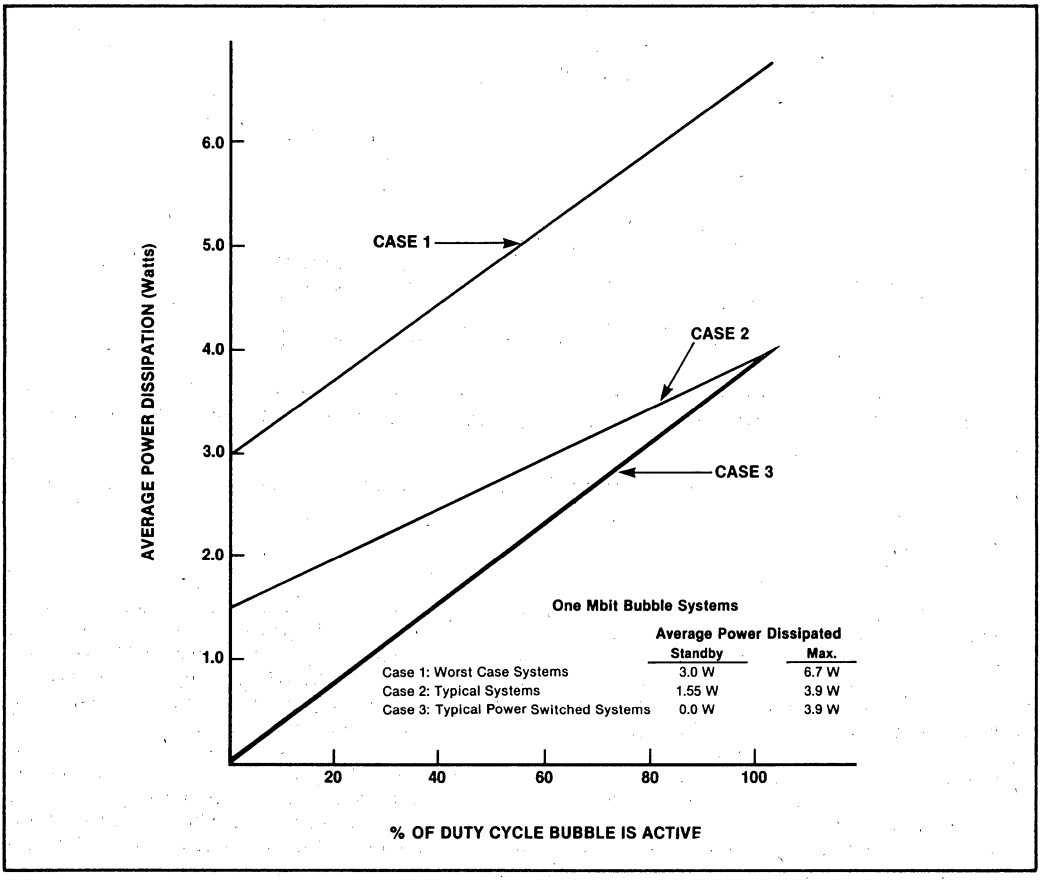


Figure 2. A Comparison of Bubble Memory Activity to the Amount of Average Power Dissipated by the Bubble Memory System

The lowest value on each plot is the power that system dissipates in standby mode. The maximum point is the power that would be dissipated if the processor was dedicated to accessing the bubble memory continuously. To give an example of an application that falls somewhere between the extremes, consider a processor monitoring vintage wine cellars. It runs a program which collects data about the cellars' environmental conditions and stores the information in the bubble memory. Figure 3 shows how power is dissipated by a typical bubble system in two minutes assuming that the program accesses the memory for an average of six seconds when it stores data.

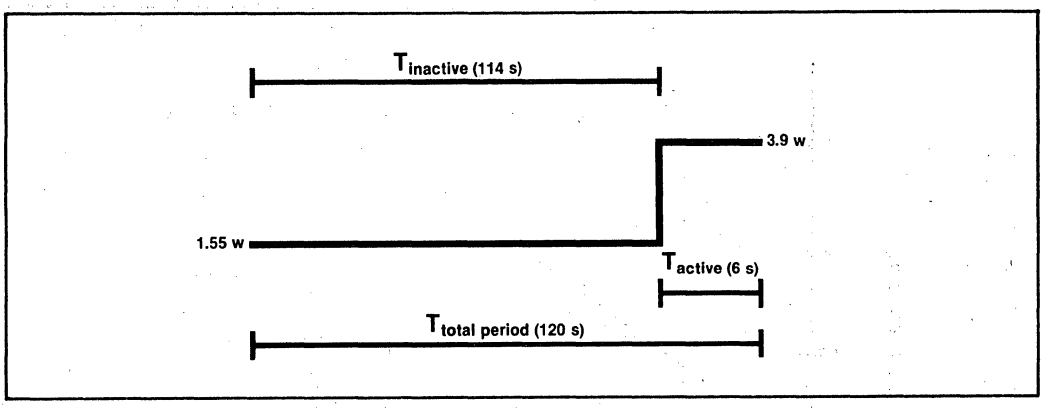


Figure 3. A Duty Cycle Period for a Bubble System with Typical Power Dissipation (Not drawn to scale)

There are two ways to figure how much average power is dissipated in the monitor example. By calculating the average,

(6s (3.90W) + 114s (1.55W)) / 120s = 1.66W

or by using the graph of Figure 2. To use the graph, the duty cycle must be calculated; six seconds is 5% of 120 seconds. After finding 5% on the % duty cycle axis, the amount of power dissipated can be read off the typical system plot. It is also interesting to note how much power would have been dissipated had the bubble system been worst case or power switched. Those values are 3.15 W and 0.35 W respectively; switching time overhead was neglected. If the monitor was to run on a battery for long periods of time, the graph indicates that it would be very worthwhile to add a power switch, about 1.40 W saved per minute over worst case design.

Generally, your duty cycle will be well below 20%. Data acquisition, portable terminal and portable PC applications have long term duty cycles of less than 10%. Even in a high transaction rate application the duty cycle is frequently small. For instance, the processor in a grocery store's point of sale terminal accesses a bubble system for information on items as often as every second. If the information can be read out in 256 byte blocks, then it will take the bubble approximately 63.5 ms to access the data, yet the duty cycle is still only about 6%, (63.5 ms/ 1 s) x 100% ~ 6%.

If your data access and standby times vary, you will want to estimate your average duty cycle so you can use the graph to determine how much power could be saved by switching the bubble. A bubble will take about 41 ms to locate specific memory locations and then it can transfer consecutive pages of 64 bytes approximately every 7.5 ms. Estimate your active unswitched time as follows,

.041s + .0075s (NUMBER OF PAGES TO TRANSFER) = ACTIVE TIME (seconds)

A switched system will have a slightly greater active time due to switching overhead,

Power-up Time + ACTIVE TIME + Power-down Time = ACTIVE TIME (switched)

This incremental difference is not neglible if ACTIVE TIME is very small. For example, the switch documented in the note takes 160 ms to power-up and 48 ms to power-down. If the grocery store's point of sale terminal used that power switch then its active time would change from 6% to 27%. From the graph, the difference in average power dissipation is .76 W, (1.8 W — 1.04 W). Switching is not recommended for systems with average duty cycle periods of less than one second.

LOW POWER DESIGN TECHNIQUES

The most efficient way to reduce your bubble system's average power consumption is to turn off the bubble when it is not being used by the processor. Designing in the simple switch documented in this section will do this quickly. All the hardware considerations necessary to successfully power switch a bubble system are included.

Two other topics will be discussed besides power switching, a secondary hardware switch for the bubble memory's detector and some software considerations including the fastest way to initialize a power switched bubble system and some software considerations such as the fastest way to initialize a power switched bubble system and some energy efficient software techniques for bubble memories in general.

HARDWARE CONSIDERATIONS FOR POWER SWITCHES

This is a basic outline of what is involved in power switching a bubble system. It is not a very complicated process. The CMOS controlled switch or one of your own design can be added in with very little system modification. Figure 4 is a block diagram of how the interface between your bubble memory and host processor will look with the sample switch in place. A table of typical values measured when the switch was validated appears in the appendix.

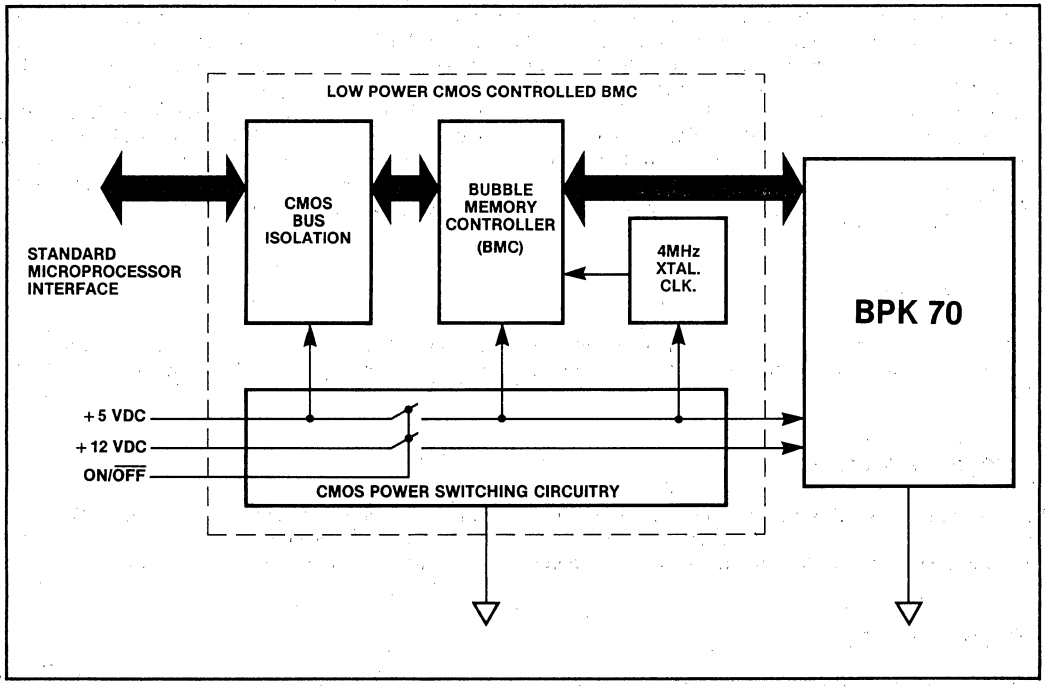


Figure 4. Block Diagram of a Low Power Bubble Memory System

Integrating A Power Switch To A Bubble Memory

There are two power supplies to the bubble system that will need to be disconnected when the bubble is in standby mode. They are +5 Vdc and +12 Vdc respectively. In addition to supplying these voltages within a ±5% voltage tolerance, the switch must also comply with the power-down specifications for Intel bubble systems. The power-down decay rates are;

Power on the 5 Vdc line must not decay at a rate exceeding 0.45 V/ms.

Power on the 12 Vdc line must not decay at a rate exceeding 1.10 V/ms.

These rates must be maintained for 120 μs after the 7220 bubble memory controller (BMC) recognizes that a power-down has occurred; for more information on power-downs, see Application Note 127, Powerfail Considerations for Magnetic Bubble Memories.

In any case, the power switch circuit described in this note maintains a 150 μs delay for itself on either a controlled power-down or on complete system power loss. The incorporated delay allows the switch to be swiftly integrated into any existing memory system whose supplies already comply with the decay rates.

Aside from the switch interface, the BMC will do its communications directly with the host processor. During power-ups and downs, these two technically sophisticated devices need to be isolated from each other so that neither one sends destructive noise to the other (the BMC in particular should not see any inputs greater than V_{CC} + 0.5 Vdc at anytime). This isolation can be accomplished with bus transcievers which maintain a large impedance between the devices. The transcievers will also add a round trip delay to the bus signals, and that delay should be added to the read and write strobes, RD/ and WR/. For example, the transcievers used in the sample switch have a round trip delay of 100 ns worst case so the two strobes should be increased from 200 ns to 300 ns. In low power systems, 300 ns strobes are not uncommon and system performance should not be compromised. A typical case, the Intel 8088 Microprocessor running at 5 MHz will still have a read strobe in excess of 300 ns.

Bubble System Clock

The BMC needs a 4 MHz, TTL level clock to do self-contained timing. The clock generator will have to be switched with the bubble memory system. Again, this is to prevent signals larger than V_{CC} +0.5 Vdc from being sent to the BMC.

Selecting Components

Selected parts should dissipate a minimum amount of power or the switch will defeat its own purpose. When the sample switch was prototyped, several CMOS components were picked because of their low power CMOS characteristics. In particular, the transcievers are compatible with either a 5 Vdc CMOS or TTL processor bus.

Switch Selection

A bubble system power switch should be as fast and reliable as the bubble system, have a small 'on' resistance, and be able to operate using the existing power supplies. There are two choices for switches, relays and FETs. FETs were chosen over relays for the prototype switch because they are faster and more reliable. Frequently, mechanical relays do not have lifespans comparable to bubble memory systems.

An N-channel FET was used on the 5 Vdc supply line. NFETS have low 'on' resistances and are for the most part inexpensive. By connecting a 5 Vdc supply to the NFET drain, a nominal 5 Vdc can be supplied to the bubble by the

source terminal when Vgs is large enough to turn on the NFET. The switch is turned on and off by changing the gate voltage, Vg, from 12 Vdc to 0.0 Vdc. When Vg is 12 Vdc, Vgs is 7 Vdc and the NFET is on; zero volts at Vg does not create a large enough Vgs to turn the NFET on.

In a similar way, a PFET on the 12 Vdc line supplies the bubble system with a nominal 12 Vdc supply. An NFET was not used because it would require 19 Vdc at the gate to set Vgs to 7 Vdc and have the FET turn on. Instead Vg switches from 0.0 Vdc to 12 Vdc.

Although the voltage drops across the FETs will be very small, they will add incrementally to the amount the bubble supplies vary from their specified voltages. For example, if the switch has a −1.5% voltage drop across it and your power supplies vary by ±3.5%, then the total operating range for the bubble's supplies is −5% to +2% nominal which is still within specifications. However, if your supplies varied by ±4%, then the bubble supplies would be out of specification by 0.5% on the low margin. To always stay within specification, operate your power supplies slightly off-set into their high margins. This procedure is not necessary, but it does add some operating margin to the system.

For example, in the sample switch the NFET for the 5 Vdc line has a worst case 'on' resistance of 0.16 ohms (@ 75 °C). The 12 Vdc line's PFET has 0.40 ohms (@ 75 °C) of 'on' resistance.

Using Table 1 the maximum power supply currents can be calculated,

Vd = +5 Vdc; 384 mA
Vs = +12 Vdc; 400 mA

and the worst case voltage drops across the FETs are then,

(+5 Vdc) Vds = (384 mA) 4) (0.16 ohms) = 62 mV
(+12 Vdc) Vds = (400 mA) (0.40 ohms) = −160 mV

Finally, the operating margins will change by,

(+5 Vdc) (62mV/5V) 100% = 1.24%;
+6.24% to −3.76% = operating margin
(+12 Vdc) (160 mV/12V) 100% = 1.33%;
+6.33% to −3.67% = operating margin

Doing a similar analysis for two bubble system with only one bubble active at anytime yields these operating margins,

(+5 Vdc) +6.79% to −3.21%
(+12 Vdc) +6.75% to −3.25%

A POWER SWITCH

Some characteristics of the CMOS controlled power switch shown in Figure 5 have already been described. To reiterate, the switch is easy to assemble and should require little if any system modifications. A user will have only one bus signal to indicate power on or off to the switches, (there is also an optional interrupt line to signal the processor that the supplies are operational, i.e., above powerfail threshold levels). The circuit itself is low power (25 mW standby max.) and has reliability compatible with the bubble memory system. Finally, unlike disks whose mechanical latency make them difficult to switch, the switch and the bubble system are all solid state so frequent switching can be accomplished rapidly.

A TTL or CMOS processor using DMA or polled data transfer modes with a bubble system (two bubble memories maximum) can handle this switch.

Driving The FET Switches

The one bus signal from the processor to the switch in Figure 5 is called POWER-ON. To supply power to the bubble system, POWER-ON is set active high; a logic low on POWER-ON signals a power-down.

POWER-ON is fed to the positive inputs of two comparators with open collector outputs; both negative inputs are set to 1.5 Vdc, i.e., POWER-ON signals greater than 1.5 Vdc are considered logic highs. The output of one comparator is sent to the BMC's RESET/ input. RESET/ is the BMC's hard reset signal. When it goes active low, it is an indication to the BMC that a power-down is occurring.

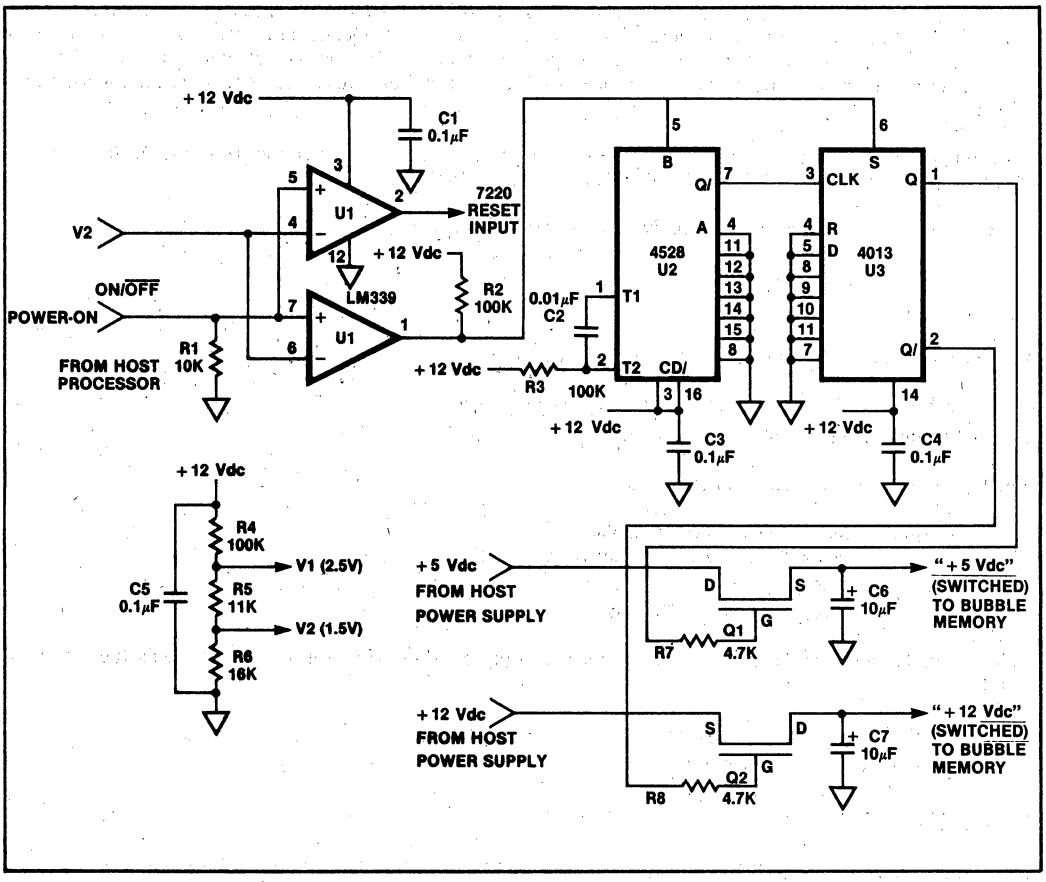


Figure 5. Bubble Memory Power Supply Switching Circuitry

Table 2. Parts List

Part	Purpose
	ALL RESISTORS ARE 0.25 W. 5% TOLERANCE.
R1 = 10 kohms	Sets gate input to low in absence of a drive signal.
R2 = 100 kohms	CMOS pull-up resistors.
R3 = 100 kohms	Sets 150 μs or greater time cosntant delay for one-shot.
R4 = 100 kohms	Voltage divider (Establishes V1 and V2 references).
R5 = 11 kohms	Voltage divider (Establishes V1 and V2 references).
R6 = 16 kohms	Voltage divider (Establishes V1 and V2 references).
R7, R8 = 4.7 kohms	Controls transition rate of Vs and Vd.
R9, R10 = 5.1 kohms	CMOS pull-up resistors.
C1, C3, C4, C5, C8, 0.1μF, 50 VDC	Power supply decoupling capacitors.
C2 = .01μF (typical) 50 VDC	Sets 150 μs or greater time constant delay for one-shot.
C6, C7 5 10 μF, 25 VDC	Power supply decoupling capacitors.
U1	LM339 low power quad comparator.
U2	4528 CMOS dual one-shot.
U3	4013 CMOS dual D flip flop.
U4, U5	74SC245 CMOS octal transcievers.
U6	74LS08 AND gate.
Q1	Siemens N-channel Econofet, BUZ71A
Q2	International Rectifier P-channel FET, IRF9531
Q3	N-channel FET, 2N6659 (Optional 7110 Bubble Memory Detector Switching)

The other comparator's output, herein referred to as Vo, is tied high to 12 Vdc through 100 kohms. It controls two CMOS components, a D flip flop through its SET input and a one-shot through its trigger input. While Vo remains high, the flip flop's outputs will remain set (Q equals +12 Vdc and Q/ equals 0.0 Vdc) since SET, equal to Vo, will be high, 12 Vdc. The outputs, Q and Q/, are connected to the FET gates through 4.7 kohm resistors; Q drives the NFET's gate on the 5 Vdc line and Q/ drives the PFET on the 12 Vdc line. The resistors are in series with the FETs' internal capacitances and are used to control the rate at which the supplies power-up and down.

When POWER-ON goes to logic low, the output connected to RESET/ will cause the BMC to start its internal power-down routine. It will now take the BMC 120 μs (worst case) to execute an orderly shut down. During this time, the power-down decay rate specified in the hardware considerations section MUST BE SUPPORTED or data integrity may be compromised.

The power switch keeps the supply lines open by keeping the FETs on until 150 μs have elapsed after POWER-ON goes low. Vo will follow POWER-ON as it goes from logic high to logic low. The one-shot will trigger off Vo's fall-

ing edge. After waiting 150 μs, the output from the one-shot will clock the D flip flop and Q and Q/ will be reset. This resets the NFET's gate voltage to 0.0 V and the PFET's to 12 Vdc. Now both FETs are off, i.e., so are the bubble system's power supplies.

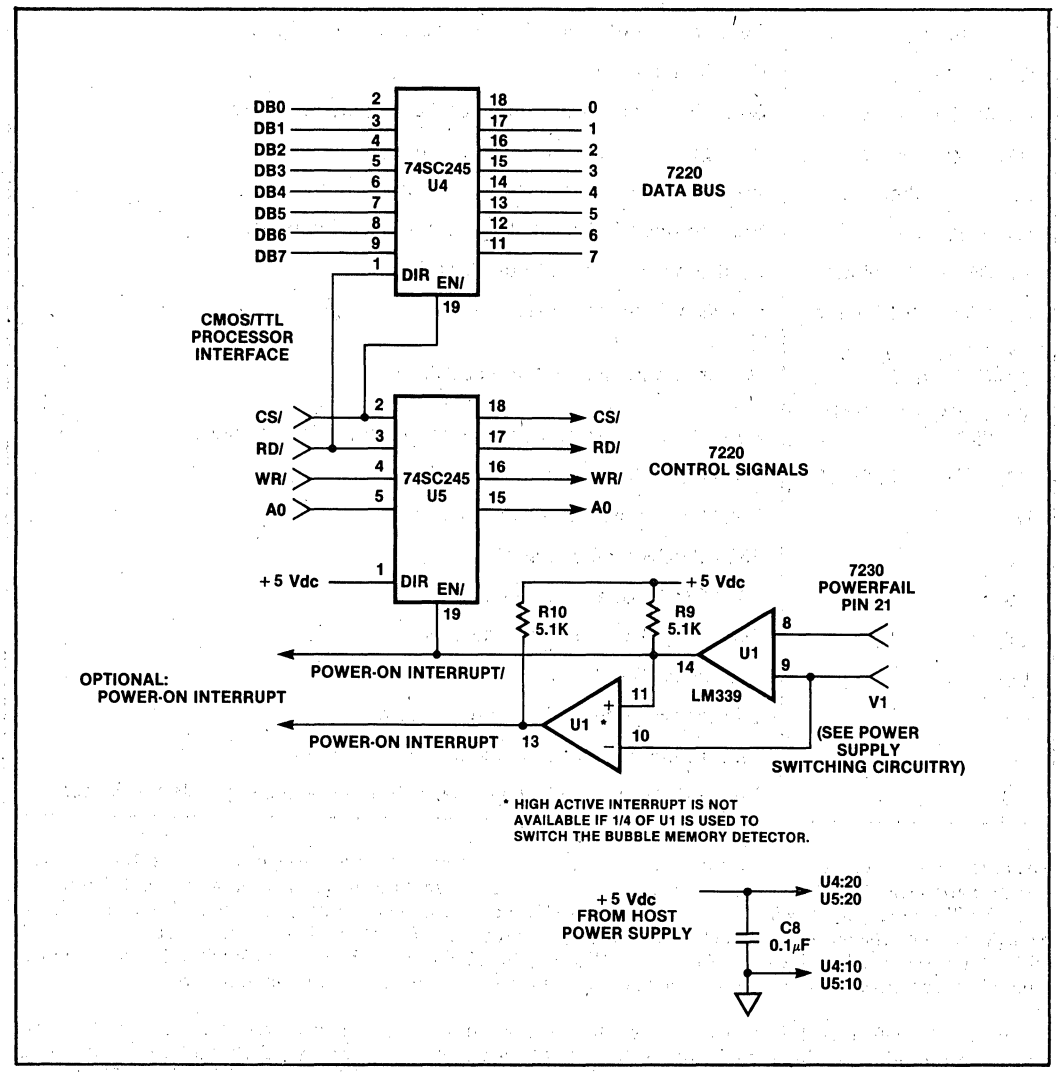


Figure 6. Bubble Memory Power Switch CMOS Bus Isolation Circuitry Polled Mode Only

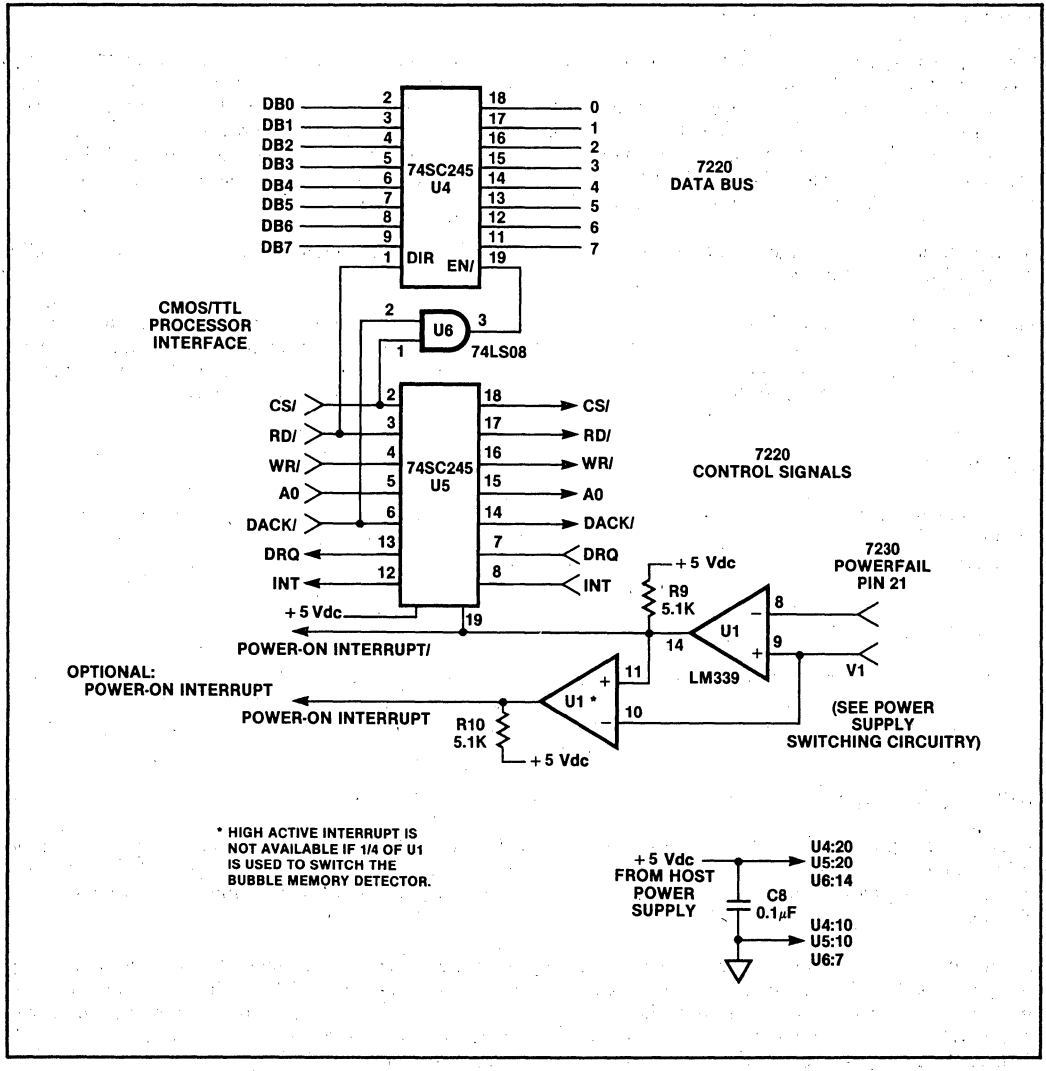


Figure 7. Bubble Memory Power Switch CMOS Bus Isolation Circuitry Polled/DMA Mode

Enabling The Interface Bus

Once the power supplies are operational, the switching circuit will enable the interface bus. Figures 6 and 7 are two possible designs for the bus transcievers; Figure 6's circuit supports polled data transfers only, Figure 7 is a modified version of Figure 6 which supports either polled or DMA. Both designs use the open collector output of a comparator to enable the BMC control signals; the output is tied high to 5 Vdc via 5.1 kohm resistors. The inputs to the comparator are 2.5 Vdc on the positive input and the 7230 current pulse generator's powerfail output, PWR.FAIL/ (pin 21), is sent to the negative input. When PWR.FAIL/ is high, greater than 2.5V, the bus is enabled, and an active low PWR.FAIL/ signal would disable the bus.

The data bus lines are not enabled until the processor selects the bubble system to do a data transfer by setting CS/ (or DACK/ for DMA) active low.

As an option, bus the output signal of the PWR.FAIL/ comparator to the host processor for an interrupt to detect when the power supplies are operational. Otherwise the processor will have to delay interacting with the bubble system until the supplies can be guaranteed operational. Invert the interrupt line with the unused fourth comparator if your processor supports active high interrupts; PWR.FAIL/ is active low on a power-down.

SOFTWARE CONSIDERATIONS

All the software information needed to successfully power switch a bubble system is presented here. Application Note 157, Software Design and Implementation Details for Bubble Memory Systems, is a useful reference if you are unfamiliar with the fundamentals of bubble memory software.

Data Transfer Modes

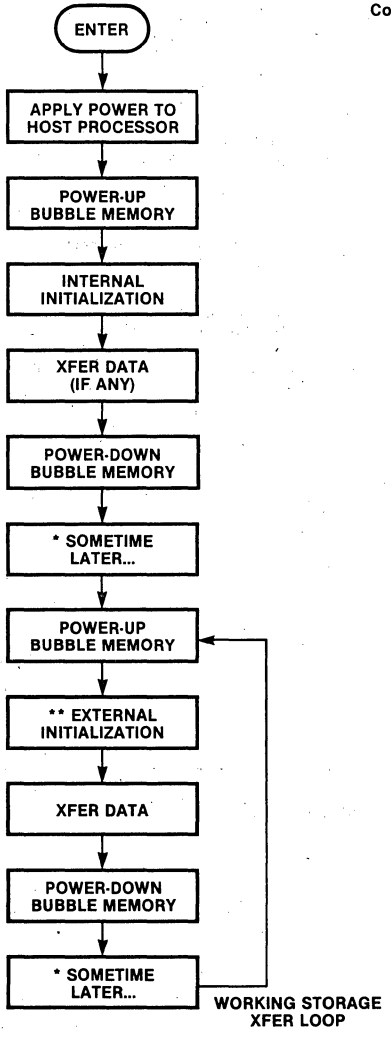
Two data transfer modes compatible with the switching circuit's bus interface are polled and DMA. Polled mode is easy and consumes the least amount of power and board space. DMA requires a DMA controller and the BMC's DRQ, DACK/ and INT signals are added to the bus interface.

Initializing The Bubble Memory

An initialization procedure must be followed after every power-up to place the BMC in a known state, to load the bootloop code into the bootloop registers and to synchronize the bubble memory to its first logical page of 64 bytes. In power switched systems, power-ups will occur before each memory access and a fast initialization routine is very desirable.

There are two ways to initialize a bubble memory. One is an internally generated command sequence executed by sending the INITIALIZE command to the BMC. The other method emulates INITIALIZE by sending the command sequence and bootloop code from the host processor to the BMC, but does not synchronize the bubble memory. This external initialization does have the advantage of being faster; worst case execution of an INITIALIZE command is 170 ms versus 5 ms for an external initialization.

Both types of initialization should be used in a power switched configuration. Send an INITIALIZE command after every cold start to synchronize the bubble memory. At the completion of the data transfer, have the BMC execute a WRITE SEEK to location (page) 395H or a READ SEEK to location 9BH. This will synchronize the bubble as the INITIALIZE command did. The bubble is non-volatile so this synchronization will not be lost even when power is removed. On the next and subsequent power-ups, use the external initialization to quickly put the BMC into a known state and place the bootloop code into the bootloop registers. Then continue to do a seek operation on each power-down to keep synchronization. Figure 8 flowcharts the operating sequence just described for low power bubble memory systems. Figures 9 and 10 are the power-up and down sequences. Flowcharts for the internal and external initialization routines are in the appendix.



Comments: The sequence of events outlined in the flowchart must be executed each time the host processor is powered-up.

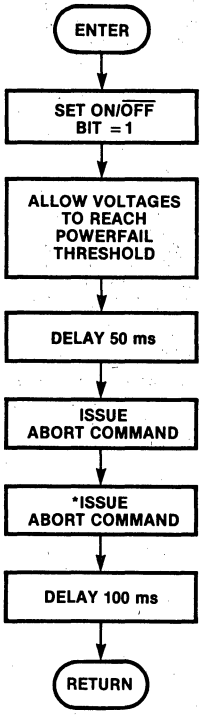
* The typical time between removing the power from a Bubble Memory to powering back up should be approximately one second or greater to derive the maximum benefits from power switching.

** An external initialization is used to decrease the power-up overhead time from 235 ms with an internal initialization to less than 160 ms. However, in all cases, an internal intitialization can be substituted for an external intitialization to accommodate existing software.

Additional detail can be found in the following flowcharts:

A. Powering-Up A Bubble Memory, Fig. 9
B. Powering-Down A Bubble Memory, Fig. 10
C. Internal Initialization, Fig. 13
D. External Initialization, Fig. 14

Figure 8. Low Power Bubble Memory System Flowchart



Comments: An optional Power-On Interrupt is available from the Bubble Memory power supply switching circuitry to detect when V_S and Vd have reached their respective threshold voltage. The interrupt can be omitted in place of an additional delay. The amount of delay is system and power supply dependent but is usually less than 5 ms.

Further detail concerning the successful execution of the Abort command can be found in Application Note 157, "Software Design and Implementation Details for Bubble Memory Systems."

* In all cases, a second ABORT command guarantees that the BMC resets.

Total execution time is typically less than 155 ms.

Figure 9. Power-up Flowchart for a Low Power Bubble Memory System

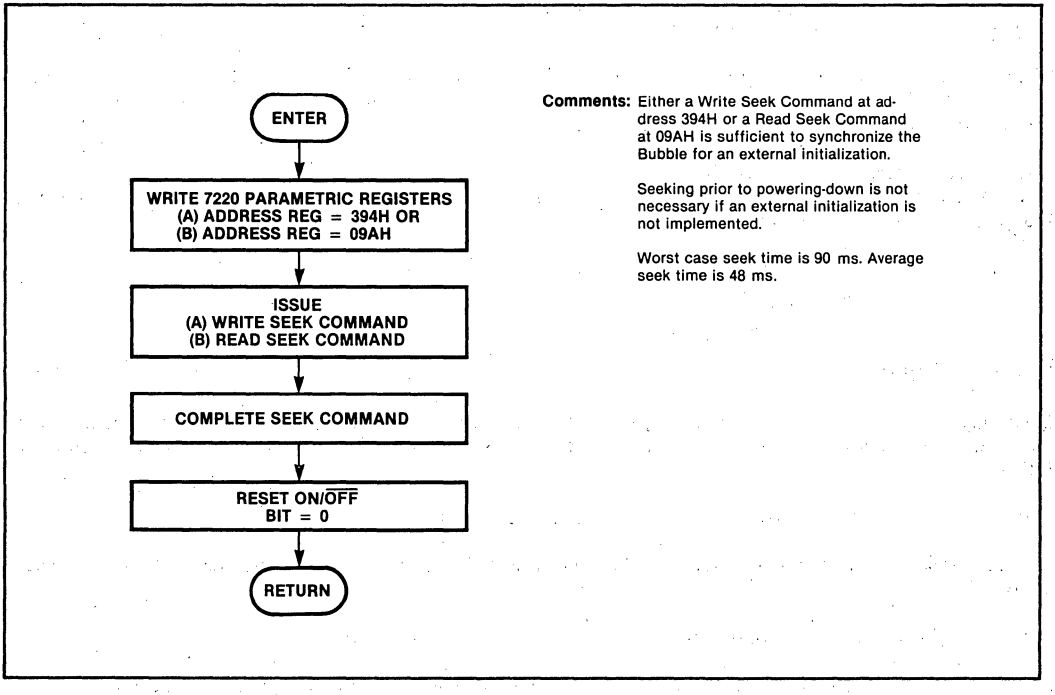


Figure 10. Power-down Flowchart for a Low Power Bubble Memory System

An easy way of keeping the bootloop code outside the bubble is to read it out of the bootloop registers, after each cold start, into the host system's RAM. Then retrieve the code from RAM for each external initialization.

The seek commands expect their operands to be the number of the page one previous to the page you wish to seek. For example, the page that should appear as the operand in the WRITE SEEK command for the initialization routine above is 394H and the READ SEEK operand should be 9AH.

If speed is not a factor, you can use the INITIALIZE command after every power-up, but the total amount of time before a data transfer can begin will still be the time it takes for the power supplies to become operational, typically 155 ms plus the time to initialize the bubble system,

Power-up + Internal Initialization = 325 ms (worst case)

Power-up + External Initialization = 160 ms (worst case)

Efficient Software

Every operation run on your system sets its own bubble memory needs. How efficiently your system responds to these needs determines how much power is dissipated. Some suggestions for energy efficient drivers and programs follow.

If information is called in a fixed sequence, store it in that fixed sequence.

Instead of repeatedly accessing the bubble for the same information, transfer the data into system RAM and retrieve it from there.

Transferring many pages of data is more efficient than doing many small transfers.

Intrinsically, running multibubble systems in serial will use less power than running parallel memories since in the former case only the coil drives of one bubble memory will be active at any one time. For example, a system running two bubbles in serial will have one active bubble memory and one bubble in standby mode any time the system is accessed. This means 5.45 W, 3.9 W + 1.55 W, will typically be dissipated during the access. Run in parallel, these same two bubbles will typically dissipate 3.9 W each or 7.8 W total during a data transfer.

DETECTOR SWITCHING

Although power switching is the primary hardware technique, the BMC has an output signal that indicates when the detector stacks are active. One advantage detector switching has over power switching is that the bubble system does not need to be reinitialized when power is reapplied.

Since the stacks are only used to sense the bubbles during read operations, the DETECTOR ON/ signal can be used to switch power to the detectors. Standby power consumption can be reduced by 20% per bubble memory if the detectors are switched off and the incremental amount you gain by leaving them off during write operations depends on the frequency and duration of your read and write operations. For example, a jet airplane's bubble memory flight recorder (using eight one mega-bit bubbles) has data written out to it on every flight. It will only be read if there is a problem during a flight. By switching off the detectors, four watts are saved on each flight (0.5 W/bubble).

Figure 11 is the circuit diagram for a detector switch. DETECTOR ON/ is inverted with a comparator; DETECTOR ON/ is the negative input, 1.5 Vdc is the positive input and the output is tied high to 5 Vdc through 100 kohms. The inverted signal is used to change an NFET's gate voltage which turns the NFET on and off. That in turn switches the detector supply on and off. Placement of the NFET is critical; lay it out as close to the bubble memory as possible.

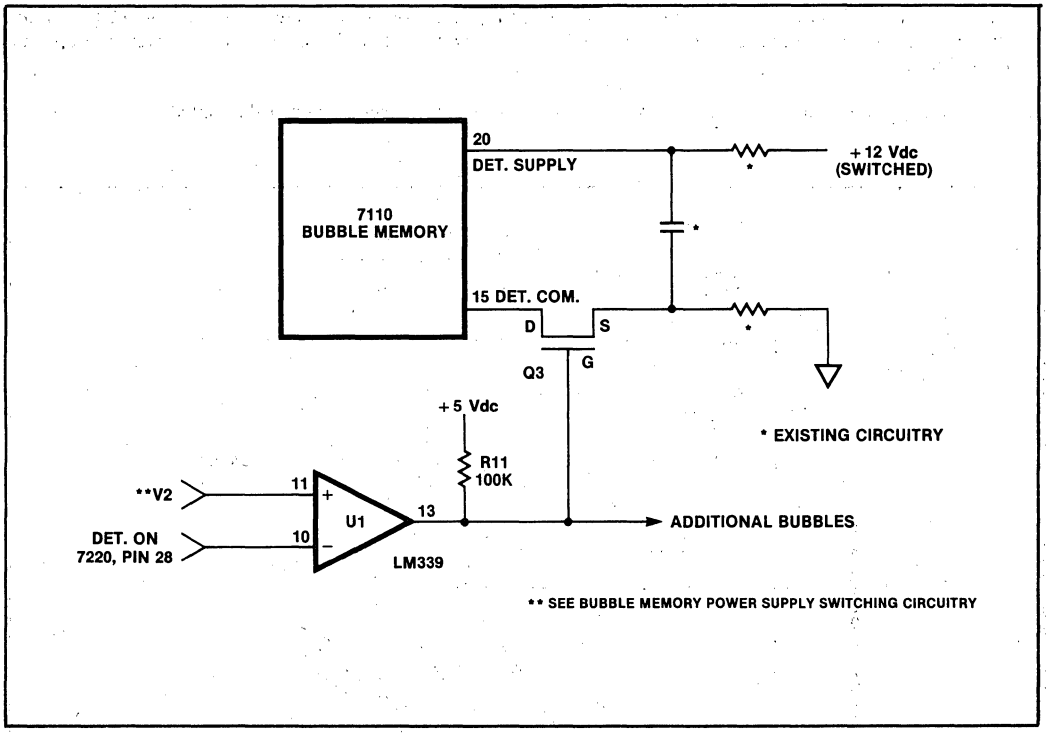


Figure 11. Bubble Memory Detector Power Supply Switching Circuitry

With a very restricted power budget, consider implementing both power and detector switching. In the sample switch, the unused fourth comparator is available for the detector switch if it is not used to invert the optional power-on interrupt.

Figure 12 is a graph comparing complete switching to the amount of average power dissipated.

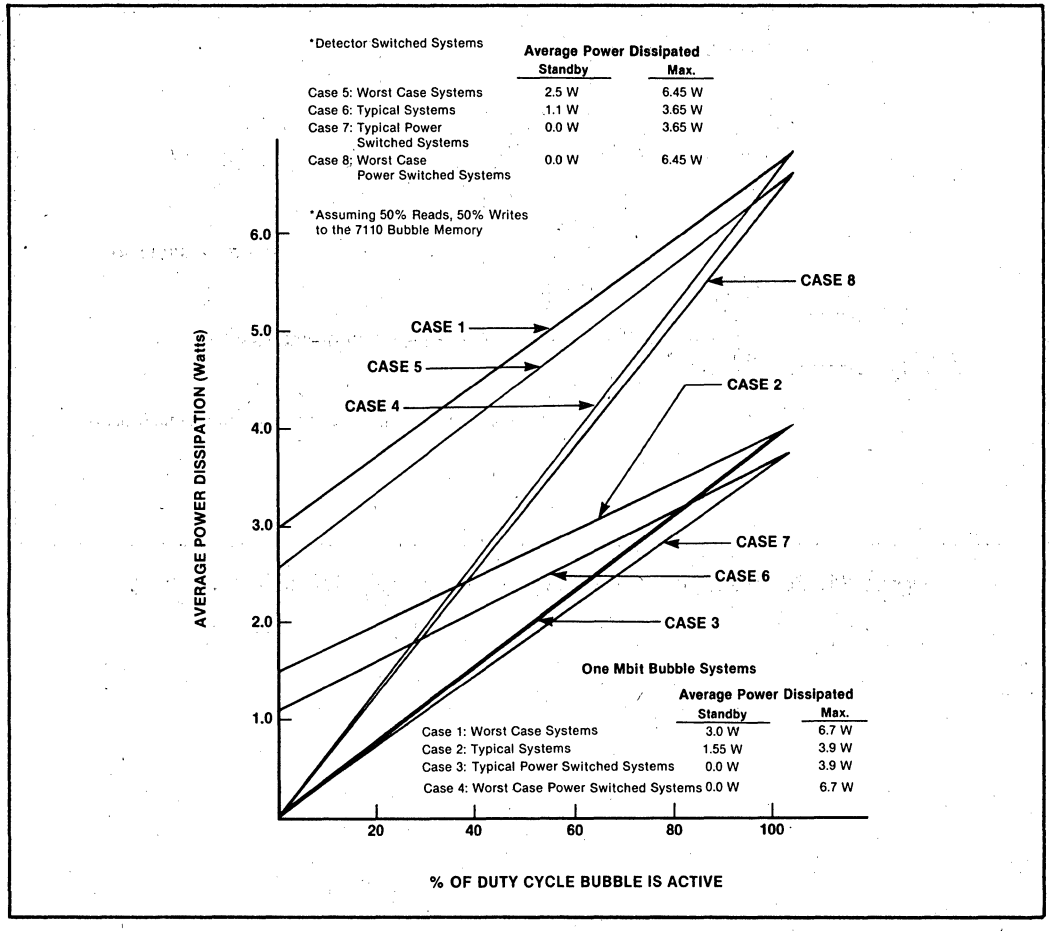


Figure 12. A Complete Comparison of Bubble Memory Activity to the Amount of Average Power Dissipated by the Bubble Memory System

SUMMARY

The main goal of this application note is to assist designers developing portable or other low power equipment. By utilizing the CMOS controlled power switch, a designer can build a simple, reliable, low power bubble memory system with a minimum of time and effort.

APPENDIX A

Typical Measured Values @ 25 °C

Configuration — One megabit Bubble Memory System incorporating a polled mode interface.

		Vs (+5VDC)	Vd (+12VDC)
Case 1:	Power-On = Off		
	Bubble Memory System Power Consumption	0.19 mW	13.66 mW
Case 2:	Power-On = On		
	Bubble Memory in Standby		
*	Total Bubble Memory System Current	260 mA	39.4 mA
*	Bubble Memory System Power Consumption	1.3 W	0.5 W
	Voltage Drop Across the FET Switch	32.1 mV	9.5 mV
Case 3:	Power-On = On		
	Bubble Memory Actively Transferring Data		
*	Total Bubble Memory System Current	262 mA	252 mA
*	Bubble Memory System Power Consumption	1.3 W	3.0 W
	Voltage Drop Across the FET Switch	32.4 mV	61.8 mV
Power-Up Rise Time (Power-On = 1)		150 μs	600 μs
Power-Down Fall Time (Power-On = 0)		15 ms	250 ms

*Includes an Intel 8284A clock generator to produce the required 4 MHz clock for the 7220 controller and BPK-70 (7242 Formatter Sense Amplifier).

Clock Specifications:	**Parameter**	**Min.**	**Max.**
	Clock Period	249.75 ns	250.25 ns
	Clock Phase Width (High)	45%	55%
	Input Signal Rise Time		30 ns

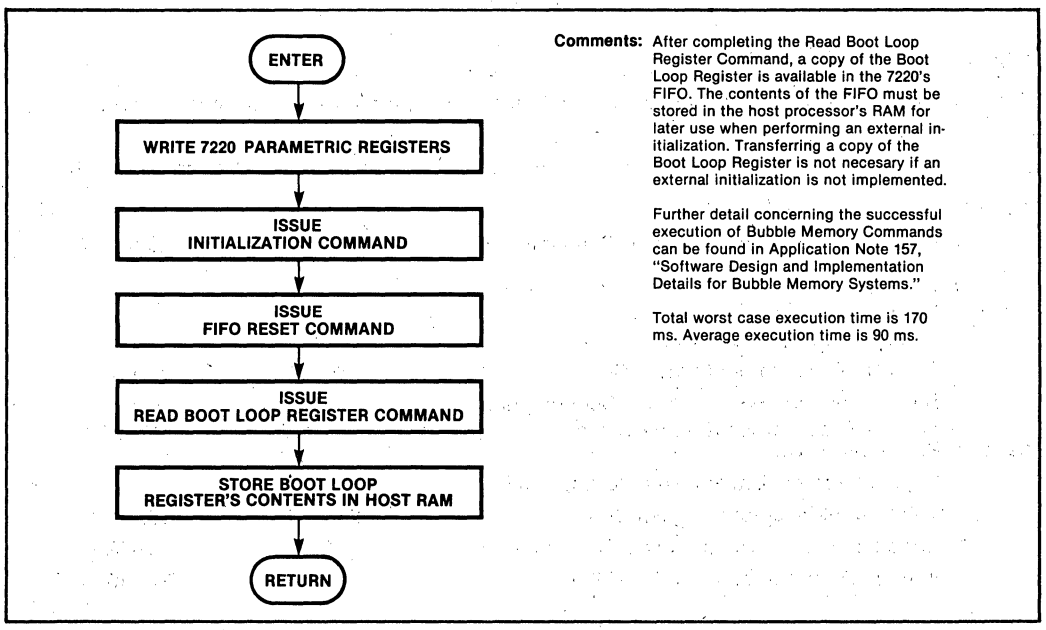


Figure 13. Internal Initialization Flowchart

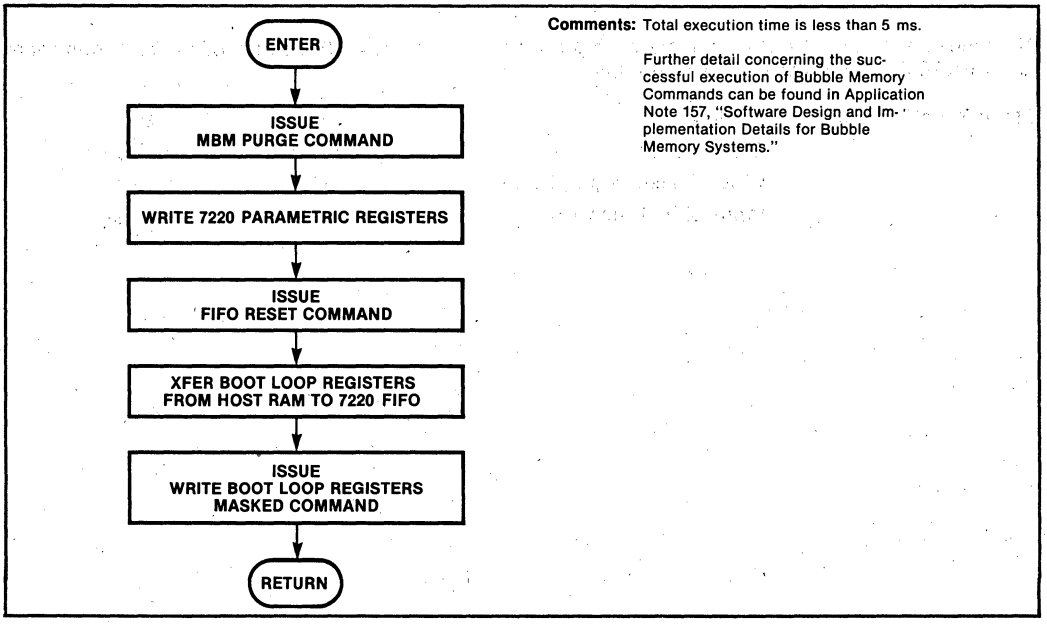


Figure 14. External Initialization Flowchart

Thin-film detectors, X-ray lithography deliver 4-Mbit bubble chip

Next-generation bubble memory chip is even smaller than the compatible, 1-Mbit device; set of support circuits takes care of memory system requirements.

Propelled by X-ray lithography and thin-film permalloy detectors, bubble memory chips have climbed to the 4-Mbit level.

Using X-ray lithography, Intel Corp. (Santa Clara, Calif.) has managed to reduce the periodicity between bubbles from 11.2 (for its 1-Mbit chip) to 5.6 μm and feature sizes from 1.25 to 0.75 μm. At the same time, thin-film permalloy detectors, replacing thick-film versions, nearly double the signal strength of the detected bubbles (Fig. 1).

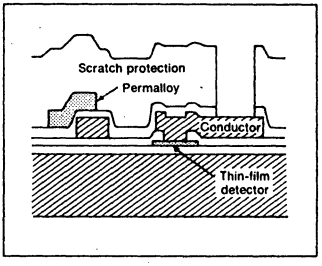
Moreover, a novel multiplexing technique handles the outputs from the eight on-chip detectors, which is double the number used on the 1-Mbit chip. This technique, which Intel is keeping under wraps, permits the higher-density chip to fit into a 22-pin package.

The outcome of all that is the 7114, plus a complement of six support circuits. The 7114 retains the basic architecture of the 1-Mbit 7110, and all the support circuits are pin-compatible with the chips that support the 7110. Aside from a few software changes to handle the larger memory space, the upgrade is totally transparent to the system user, claims Mike Eisele, bubble memory product manager. Thus in many cases the older bubble chips can be removed from a system and new ones plugged in.

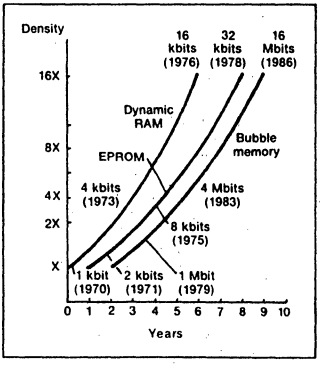
Dave Bursky

Electronic Design

However, the support chips cannot control the 1-Mbit device, and some minor hardware changes must be made to accommodate the smaller package used for the 4-Mbit chip. The package's dimensions—1.46 by 1.35 in.—represent a savings of nearly 0.9 in.^{2} over the 1-Mbit package's 1.7 by 1.68 in. In addition, the smaller package, which has DIP-like pins, eliminates the need for a socket in many cases and also has a lower profile to permit board spacings as close as 0.6 in. The same package will be used by Motorola Inc. (Phoenix, Ariz.) when it builds the second-generation 1-Mbit chip as called for in the alternative-source agreement signed earlier this year with Intel (ELECTRONIC DESIGN, July 8, p. 23).



1. A key element of Intel's 4-Mbit bubble memory is this thin-film permalloy detector structure, which delivers twice the output signal of the previously used thick-film detector.



2. Following the same growth curve as UV EPROMs and dynamic RAMs, bubble memory technology still has a good way to go to reach the 16-Mbit level projected for 1986.

However, to bring the price of the bubble memories down to what Eisele feels would be attractive for system users—about $150 for a 4-Mbit chip by 1986—Intel has turned to a Perkin-Elmer X-ray lithography system in what it believes to be the first commercial use of X-ray systems. (Other companies, though, are not very far behind—many semiconductor manufacturers have very active research and development programs to make X-ray systems practical on the production line.)

The production process for the 4-Mbit chip includes 90% of the process steps used for the 1-Mbit device, thus sharing much of the learning-curve experience, in the short run.

Functionally, the 4-Mbit device will appear to operate just like the 1-Mbit memory. However, when the 7114 operates at the 50-kHz field rate of the 1-Mbit device, the access time is double that of the smaller chip, since the loops are longer. But the data rate is double that of the 1-Mbit chip because more detector outputs are multiplexed and then fed out from the chip. Also, a version of the 4-Mbit chip will operate at twice the field rate (100 kHz), for an access time of 41 ms—almost the 40-ms access

time of the 1-Mbit chip.

There will be a full kit of parts available from Intel when samples of the memory will be available next year. The largest chip will be the 7224 controller, which duplicates the functions of the 7220 controller but has the internal changes needed to handle the larger memory space. Similarly, the other circuits are the 7234 current-pulse generator, the 7244 formatter-sense amplifier, the 7250 coil predriver, and the 7254 coil drivers.

Bubble memory capacity has been quadrupling about every four to five years. This follows very closely what happened to UV EPROMs (Fig. 2), even though EPROMs went through doubling cycles every two years.

November 1982

Bubble Chip Packs 4 Mbits Into 1-Mbit Space

Hudson Washburn
Design Engineer

Sam Nicolino
Design Engineer
Intel Corporation
Santa Clara, California

Electronic Design

ORDER NUMBER: 210842-001

BehindTheCover

Right after putting their 1-Mbit bubble memory chip into production several years ago, designers at Intel decided to try various sections of what would be needed to build a 4-Mbit device. Although several were fabricated and proved functional, priorities in ironing out the production problems for the 1-Mbit chip forced them to put the 4-Mbit design on the back burner, working on it as a secondary project. Finally, though, the years of patience are paying off, and as our cover story in this issue (p. 1) highlights, the 4-Mbit magnetic bubble memory—the i7114—is functional.

Fortunately, the designers have been able to time the developments so that both the bubble chip and its associated support chips will be ready at the same time. As Mike Eisele, product manager for the Magnetic Bubble Memory Division, notes, that wasn't the case for the 1-Mbit device—it took Intel a lot longer than it expected to make the controller fully functional.

In developing the 4-Mbit memory, Hudson Washburn, design engineer, expected that the control elements on the chip—the bubble generator, transfer gates, replicator, and detector—would be the most difficult sections to get to work, whereas he thought that the propagation paths would be relatively simple to implement. But when actually trying to create the memory chip, he and the other researchers found that the control sections performed fine after only a few iterations while the propagation paths turned out to be the tricky development problem.

Additionally, mastering the technology needed to build the 4-Mbit bubble chip was a long, hard process with many half steps back, Washburn says. However, work on the 1-Mbit device also helped the bigger memory: Every time something happened that caused yield problems on the 1-Mbit chip, work was stopped on the new circuit. When the problem or problems on the 1-Mbit process were solved, the designers applied what they learned to the 4-Mbit technology.

Also, the designers decided to use a thin-film detector structure to boost the signal-to-noise ratio of the output signal. Although building this detector adds a second critical masking level to the production process, the decrease in yield due to the additional step is expected to be more than offset by faster testing. As it turns out, testing tends to be a major part of the chip cost as the capacity reaches 4 Mbits, according to Dave Dossetter, bubble memory product marketing engineer.

Perhaps appropriately for a 4-Mbit memory, Intel worked with a manufacturer of lithography equipment and a mask maker to use X-ray lithography. Although contact printing was employed during development, Intel plans to put X-ray lithography to work for volume production, which would make it the first such commercial use.

A 4-Mbit bubble memory chip, supported by a full complement of six dedicated circuits, stands poised for applications ranging from industrial control to telecommunications to personal computers.

Bubble chip packs 4 Mbits into 1-Mbit space

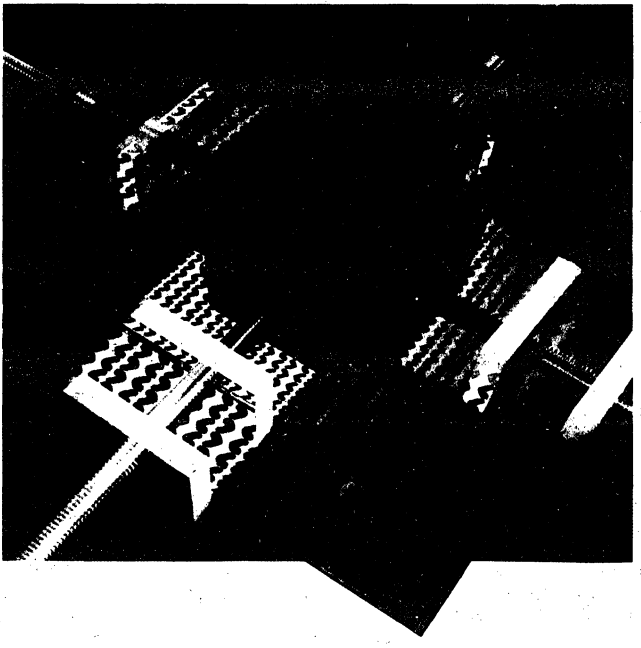
Bubble memories sport a hefty list of advantages for mass storage applications. Yet because of the complexity of interfacing them, most designers have shied away from these devices, leaving them outcasts. But the sheer appeal of 4 Mbits tucked into a 20-pin package, coupled with a set of components that takes care of the complexities of linking a bubble chip to conventional host computers, makes an extremely attractive option for those designers who have previously resigned themselves to simpler but less attractive mass storage.

As for those who have already taken the plunge into bubbles with the chip's 1-Mbit predecessor, the 7110, upgrading to the 4-Mbit 7114 requires only minimal changes.

Some of those ready to benefit from a simplified bubble memory system are portable equipment makers, who will take advantage of the compactness and nonvolatility of bubble chips. Industrial control and robotics manufacturers will appreciate bubble devices' resistance to hostile environments, since they have no moving mechanical parts to succumb to shock, corrosion, or high humidity. These last three qualities also are important to telecommunications suppliers, who need low-cost, reliable buffers for PABX and other message-carrying systems.

Still, to reap the rewards inherent in bubble memories, a full complement of support circuits must accompany the bubble chip itself. Those companions are ready, in the form of the 7224 bubble memory controller, the 7244 formatter and sense amplifier, the 7250 coil predriver, the 7254 VMOS driver transistor, and the 7234 current pulse driver.

Despite these components, a 4-Mbyte bubble memory system takes less space than the previous 1-Mbyte design, since the new bubble chip's package is both narrower, allowing more chips per board, and shorter, giving more room to stack boards next to one another (see "More Memory in Less Space"). Furthermore, the support components are interchangeable and, like the bubble chips, do not have to be matched sets, as was often true of other bubble devices. In fact, any bubble chip is guaranteed to

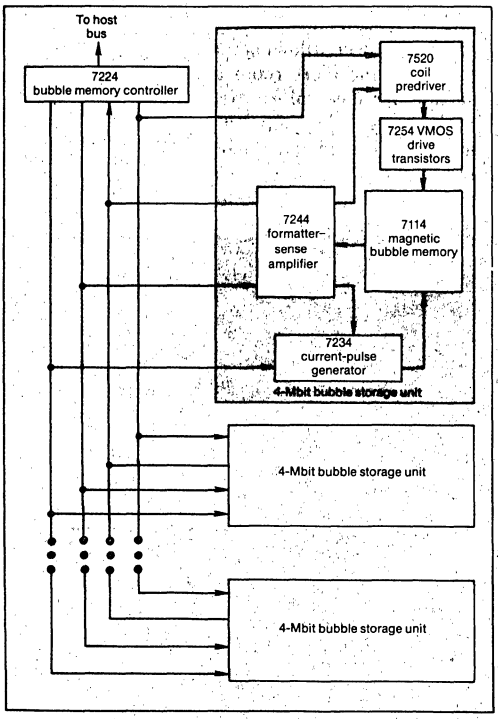


Hudson Washburn, Design Engineer
Sam Nicolino, Design Engineer
Intel Corp.
3065 Bowers Ave., Santa Clara, Calif. 95051

work with any support component, so that components can be replaced in the field without fine tuning.

Also, because the 4-Mbit bubble chip was designed to be compatible with the same hardware and software developed for the 1-Mbit version, the support circuits for both have the same pinouts. Most of the register bits are the same, too. The only differences are those in which the larger memory capacity affects how the bits are defined. Consequently, from a software perspective, any revisions to upgrade to the 4-Mbit chip are minor.

As with the 1-Mbit system, the user's interface with the 4-Mbit system remains simple. The software is written so that, first, parameters are passed to the controller by loading its registers, followed by commands. In addition, data is written or read in any of three transfer modes—DMA, polled, or interrupt—and the controller's 40-byte FIFO acts as a buffer between the host and formatter–sense amplifier chips. The formatter–sense amplifier is responsible for sending and receiving serial data between the bubble and the controller. The host system therefore need only monitor the controller's status register to determine when it is busy and to see if a transfer operation was successful.



1. The key to building a 4-Mbyte bubble memory system is the ability of the bubble chip's support ICs to simplify the interface with the host. Five such ICs plus a single 4-Mbit chip (shaded) form the basic memory block. Up to seven additional blocks in parallel, all governed by one memory controller chip, complete the system.

The bubble memory controller is the bubble chip's link to the host. It communicates with the host over an 8-bit bidirectional data bus; a single address line (A_0); and a chip-selection, a read and a write control, and an interrupt line. In addition, a ninth data bit line (D_8) can be used to detect parity errors.

The remaining input and output lines of the controller connect the formatter–sense amplifier, the coil predriver, and the current-pulse generator. These components, plus a pair of VMOS drive transistor chips, make up a 4-Mbit bubble storage unit (Fig. 1). Up to eight such units may be connected to a single controller, allowing users to trade off the number of pages against the individual page size to fit their data transfer requirements.

The controller close up

To understand the software and hardware interface with the bubble subsystem requires and understanding of the controller. An HMOS chip, it is housed in a 40-pin DIP and divided into 10 functional blocks (Fig. 2).

The host processor operates the bubble memory system by reading from, or writing to, specific registers within the bubble memory controller. The host selects each register by placing an address on lines A_0 and D_0 through D_4. Specifically, the status register and command register are directly addressed using these six bits; a third register, the register address counter, is also directly addressed and in turn indirectly addresses the remaining registers, including the block-length register, the FIFO data buffer, and the enable register. These remaining registers are called parametric registers because they contain the flags and parameters that determine exactly how the controller will respond to commands written in the command register. The parametric registers are located in a register file and are selected with addresses 1011 through 1111. In general, the parametric registers must be loaded before commands are issued to the controller.

Parametric registers are loaded when they are addressed by the register address counter. The controller automatically increments the counter by one after each data transfer between the host and a parametric register. Thus there is no need to reload the address register in the case of multiple register reads and writes.

The address register increments, starting with the address first loaded, until it reaches binary address 1111. It then wraps around to 0000 and halts until it is reloaded with another address. However, when

line A_0 is zero, all data transfers are with the FIFO. In addition, any other commands or a controlled stop sequence will reset the address counter to 0000, which is the FIFO address.

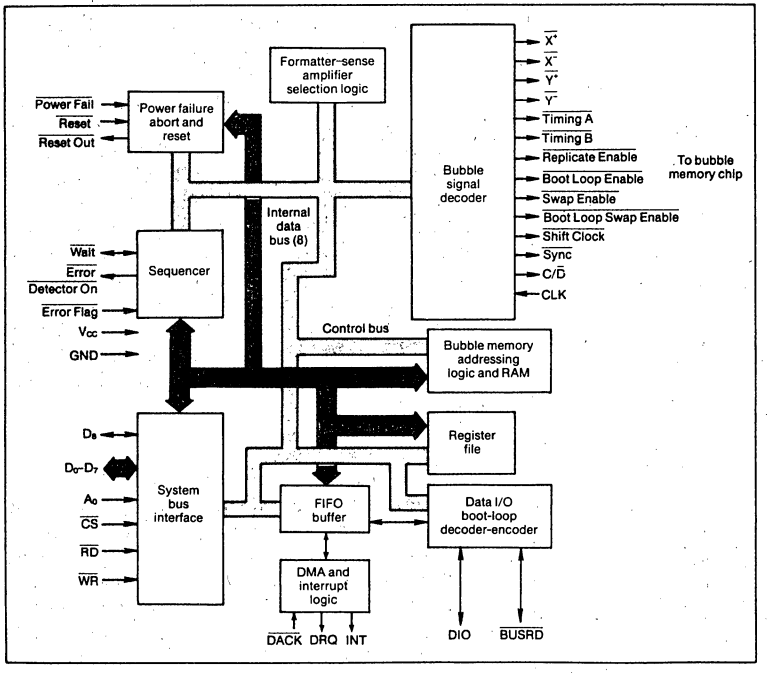
The most commonly used commands (see the table) are Initialize, Read Bubble Data, and Write Bubble Data. Others used in a typical operation are Read Seek, Write Seek, Read Formatter–Sense Amp Status, and Reset FIFO. In addition, two commands —Zero Access Read Seek and Zero Access Read Bubble Data—slash the data access time by a factor of more than 150. Zero Access Read Bubble Data returns the first byte of data in the FIFO within 50 μs after the command is sent, provided the address is known in advance of the access command.

Parameters first

Commands are written by the host into an 8-bit write-once command register. Depending on the command, certain parameters must already be written into their respective registers. For example, the Initialize command must be preceded by the number of formatter—sense amplifiers in the block-length register's first four MSB locations (Fig. 3a). Similarly, before issuing a Read Bubble Data command, the starting address information must already be set in the address register (Fig. 3b), as must be the number of system pages in the block-length register. Thus each command has its specific set of parametric requirements that must be established before it is issued.

If the parametric conditions have been set, the command is issued using a 5-bit command code. For example, Initialize is 00001, Read Bubble Data is 00010, and so on.

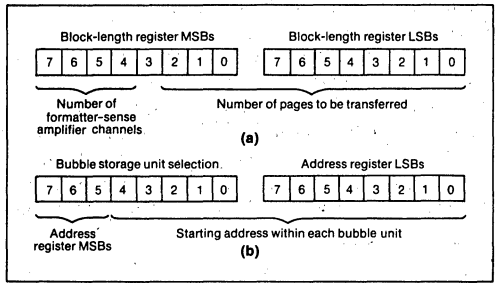
Information about any error condition, the completion or termination of a command, or the controller's readiness is stored in the status register. The host can directly address this register by setting the A_0 line and examining the eight status flags. The status register is updated every microsecond. Bits 1 through 6 (Fig. 4a) are set during command



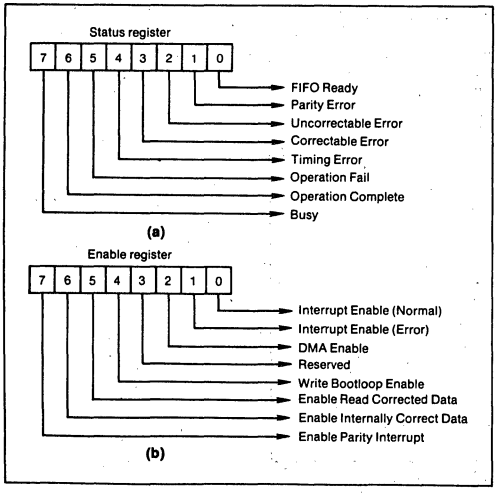
2. The 7224 bubble memory controller interfaces the bubble storage units with the host processor. It performs 10 functions, each represented by a block. The host is connected to an 8-bit data bus with an optional parity bit, a single address line, a chip-selection line, and a read and a write control line. Interrupt and DMA handshaking also are available.

execution and are reset when a new command is issued. The flags in the status register indicate whether the controller is executing a command or has completed one. In addition, they show whether an uncorrectable error or a timing error has occurred. Also, using a parity bit, the controller checks the data the host sends it and generates an odd parity for the data it sends to the host. Any parity errors are flagged.

The system page size and the number of pages to be transferred in response to a single bubble memory data read or write command are set by the block-length register, a 16-bit write-once register. The system page size is proportional to the number of bubble storage units operating in parallel during a data read or write operation. Each bubble chip requires two formatter–sense amplifier channels, with bits 4 through 7 specifying the number of such channels to be accessed. For example, in a 4-Mbyte system, if bits 7 to 4 are 0001, two channels will be accessed, each page will contain 512 bits, and there will be 65,172 pages. Setting the bits to 0100 specifies eight channels, 2048 bits per page, and 16,384 pages.



3. The parametric registers set the basic conditions for transfers between the host and the bubble memory system. The block-length register gives the number of formatter–sense amplifier channels and the number of system pages in a block (a). The address register gives the starting address for a read or write command (b).



4. The status register bits (a) tell the host about any data errors, the state of the controller's readiness, or whether a command was completed properly or not. The register is updated every microsecond and indicates whether a data error was correctable or not, in addition to pointing out parity and timing errors. The enable register bits (b) specify several conditions, including interruption on an error, DMA enabling, and parity error interruption.

The right address

Which bubble memory group is accessed and what the starting address location is within that group are determined by the contents of the address register. Each bubble chip has 8192 address locations for reading or writing data. Consequently, 13 bits are needed to specify an individual bubble storage unit's starting address. Which of the units to be read from or written to is indicated by address register bits 5 through 7. How the controller interprets these bits depends on the number of bubble storage units in a group as specified by the block-length register. For example, if the formatter–sense amplifier channels are numbered 0 through F_{16} and the number of formatter channel bits of the block-length register are set at 0000, the address register bits will specify channels 0 through 7. If, on the other hand, the block-length register bits are in the sequence 0001, the address register bits select the formatter—sense amplifier channel pairs and address register bits 0110 select channels C and D.

The address range for a 4-Mbyte subsystem is 0000–FFFF, or 65,172 pages. Selecting address register bits 0111 puts the data in the last 8192 pages of bubble storage.

Enable register controls

Certain functions in the formatter–sense amplifier and the controller are governed by setting bits in the enable register (Fig. 4b). For example, setting the Enable Parity Interrupt stops the host when the controller detects a parity error on the data bus lines (D_0–D_7). Also, the controller operates in a DMA data transfer mode when the DMA Enable bit is set. In this mode the Data Request and Data Acknowledge interface signals become operational; otherwise, the controller supports interrupt-driven or polled data transfer modes. As a result, users have a choice of three data transfer methods.

The Interrupt Enable (Normal) bit, when set to a 1, allows the controller to interrupt the host system when a command is successfully executed. The Interrupt Enable (Error) bit works in conjunction with

Bubbles by the block

The basic technology of the 7114 4-Mbit bubble chip—known as field access, conductor-first permalloy—is the same as used to build the earlier 7110, a 1-Mbit part, except for several important refinements. These refinements quadruple the bit density and the data transfer rate.

The increased density is produced by halving the period of the basic memory cell (called an asymmetric propagator) to 5.5 μm. The resultant chip size is 501 by 580 mils (compared with the 1-Mbit's 512 by 614 mils). A 0.75-μm minimum feature size, smaller than that of any silicon chip, is being printed now in development volumes using optical contact lithography. However, X-ray lithography techniques will be used for production volumes to achieve repeatible results despite the small minimum-feature size.

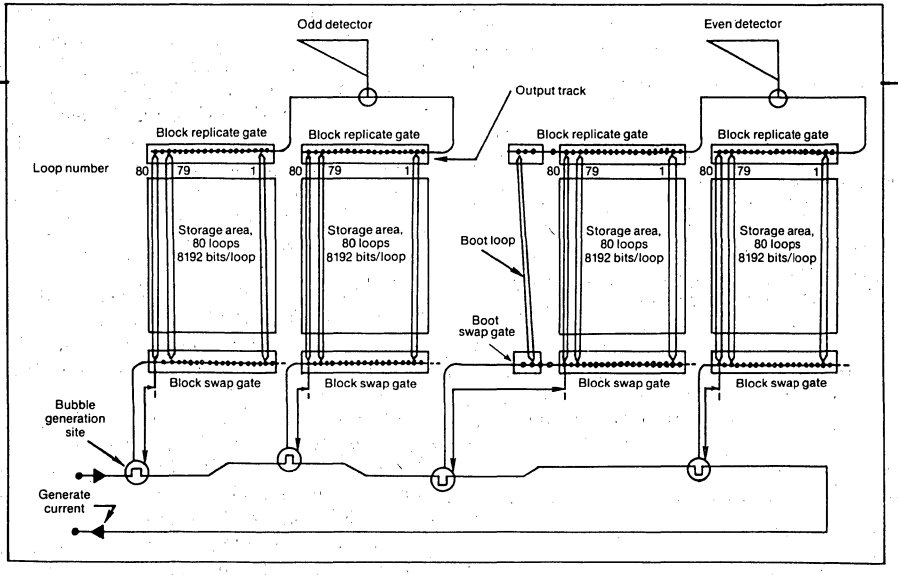
In addition, a thin-film detector was developed that doubles the detected bubble signal compared with the previous thick-film detectors. This makes doubling the data rate feasible. Further, doubling the field rotation rate from 50 to 100 kHz also doubled the data rate, producing the overall 400% increase, which also means an average random access time of 40 ms. (A 50-kHz version will be introduced first that has twice the data rate of the 1-Mbit chip and an 80-ms access time.)

Like the technology, the architecture of the 4-Mbit chip is an enhanced version of the 1-Mbit design. Both use block-swapping and replicating schemes to write and read bubbles in parallel, to ensure nonvolatile storage, and to permit the use of multiplexed replication generators to reduce the number of external pins.

The page length is fixed at 512 bits (64 bytes), but the number of pages has been quadrupled for the 4-Mbit part. Both chips are organized into identical halves. Thus, from an architectural perspective, the higher-density chip looks like a 1-Mbit part with four times the number of pages and either twice (50 kHz) or four times (100 kHz) the data rate.

Actually, the 7114 is divided into eight octants, each comprising 80 minor loops, and each loop containing 8192 bits (see the figure). The 7110, in comparison, is split into four quadrants, each with 80 minor loops, but each loop contains only 4096 bits. Also, whereas the 7110 was designed to sense one bit per side per field rotation, the 7114 senses two bits. In the 50-kHz 4-Mbit part, the longer loops are compensated for by the two-bit-per-rotation sensing.

Like the 1-Mbit device, the 4-Mbit chip has redundant loops to ensure a high yield of devices with the full 4,194,304 bits of storage capacity. Redundancy increases yields and so lowers device cost. During manufacture, each device is individually tested and a record of faulty loop locations is written and stored in the device's bootstrap loop, known as the "boot loop." The boot loop's contents are used by the 7224 bubble memory controller during initialization, reading, and writing to provide a full 4-Mbit memory space to the user while keeping redundant loops invisible. The major-track, minor-loop architecture used by both the 7114 and the 7110 to accomplish the writing, reading, and nonvolatile storage of data also maintains the reliability inherent in bubble technology.



the other enable register bits to support three levels of error correction.

At the first level, setting Enable Internally Correct Data causes the controller to send a command to a formatter–sense amplifier when an error has been detected. The formatter–sense amplifier responds by internally cycling the data through its error-correction network. On completion, it sends its status to the controller, indicating whether or not the error was corrected.

For the second level, the Enable Read Corrected Data bit prompts the controller to issue a command to the appropriate formatter-sense amplifier when an error has been detected. The formatter—sense amplifier then corrects the error if possible and transfers the corrected data to the controller. When the data transfer is complete, the controller reads the formatter–sense amplifier's status to determine whether the error was corrected. Otherwise, faulty data could be transferred to the controller and possibly to the host.

Lastly, setting the Write Bootloop Enable bit permits writing into the bootstrap loop, called here just the "boot loop." Normally, the loop should only be read, but under special circumstances a user may wish to write into it.

Bubble controller command codes

D_5	D_3	D_2	D_1	D_0	Command name
0	0	0	0	0	Write—Boot Loop Register Masked
0	0	0	0	1	Initialize
0	0	0	1	0	Read Bubble Data
0	0	0	1	1	Write Bubble Data
0	0	1	0	0	Read Seek
0	0	1	0	1	Read Boot Loop Register
0	0	1	1	0	Write Boot Loop Register
0	0	1	1	1	Write Boot Loop
0	1	0	0	0	Read Formatter–Sense Amp Status
0	1	0	0	1	Abort
0	1	0	1	0	Write Seek
0	1	0	1	1	Read Boot Loop
0	1	1	0	0	Read Corrected data
0	1	1	0	1	Reset FIFO
0	1	1	1	0	Memory Unit Purge
0	1	1	1	1	Software Reset
1	0	0	1	0	Zero Access Read Bubble Data
1	0	1	0	0	Zero Access Read Seek

More memory in less space

Instead of a leadless package requiring a second, leaded socket, the 7114 4-Mbit bubble chip is housed in a leaded package that can be placed in a socket or soldered directly to a PC board. Like the 1-Mbit package, it has 20 pins. However, the distance between pin rows is smaller, making the footprint smaller and allowing designers to incorporate more components onto the board. Also because the package's height is smaller, boards can be spaced as close as 0.6 in. to one another. Thus consequently, either more boards can be accommodated or the overall system size can be made smaller. As a result, a 4-Mbyte bubble memory system can be built in less space than a 1-Mbyte bubble system.

The FIFO as a data buffer

All data moving between the host and the bubble units passes through the 40-byte FIFO buffer. As a result, the data transfer is asynchronous, with timing constraints relaxed somewhat for both the formatter–sense amplifier and the host system. When the controller is busy executing a command, the FIFO functions as a data buffer; however, when the controller is not busy, the FIFO is available to the host as a general-purpose FIFO register bank.

Actually, a total of 43 bytes of data may be stored in the controller: 40 bytes in the FIFO, 1 byte each in its input and output latch, and 1 byte in the controller's input latch. During execution of a command involving a data transfer between the host and the formatter–sense amplifiers, the data passes through the FIFO and its status is indicated by the FIFO Ready bit in the storage register.

The FIFO is addressed automatically after the last parametric register has been written into; alternatively, the host can explicitly address the FIFO by writing the address 0000 into the register address counter. Also, after a Write Bubble Data, a Write Boot-Loop Register, or a Write Boot-Loop Register Masked command is issued, the controller delays the data transfer until there are at least two bytes of data in the FIFO. Furthermore, it is the host system's responsibility to keep up with the data transfer during execution of a command; otherwise the FIFO could underflow or overflow. If either case occurs, a Timing Error bit is set in the status register.

A look at data transfer

The boot-loop register plays a key role in data transfer both for writing and reading. This 160-bit register contains information detailing the configuration of good and bad loops in the corresponding channel of each bubble chip.

Each bit of the register corresponds to a minor loop in the bubble chip. As data passes through the latter's I/O latches, the contents of the boot-loop register are used during reading to remove the bits corresponding to bad loops and during writing the contents are used to insert 0s in those bit positions that correspond to bad loops.

Meanwhile, the error-correction block implements a 14-bit Fire code error-detection and -correction process. If it has been enabled by the user, the error-correction circuitry appends the 14-bit code to the end of each 256-bit block of data that passes through the FIFO during a data write operation. When data is being read, this circuitry checks the data block and notifies the controller with an error flag when an error has been detected.

As stated earlier, a Write Bubble Data command from the controller to the formatter–sense amplifier permits data from the controller to be written into the good loops of the memory unit. If the error correction is activated, the amplifier automatically adds the 14 error-correction bits to the end of each 256-bit data block.

Similarly, a Read Bubble Data command enables the formatter–sense amplifier to read data from the bubble chip, as was also mentioned previously. This data is sensed by the sense amplifiers and screened by the boot-loop registers so that only data from good loops is written into the FIFOs. If the error correction is selected, data to be read is first buffered. That is, a full block (270 bits) of data is collected in the FIFO before any bits are read out. As a result, the error-correction circuitry detects any errors and interrupts the controller before any data is sent. If there are no errors, the 270-bit block is read from the FIFO and sent to the controller while the next block is loaded into the FIFO.

In contrast, an Internally Correct Data sequence forces the formatter–sense amplifier to cycle the data internally through the error-correction network without sending any of it to the controller. At the end of the operation, the amplifier sets a Correctable or Uncorrectable Error bit in its status register. If the error is correctable, the controller has the option of issuing a Read Corrected Data command. This command cycles the data through the error-correction circuitry as it is being read by the controller. After all 256 bits have been transferred to the controller, the formatter–sense amplifier status register indicates whether the error was found to be correctable or not. The Read Corrected Data command is used even when the data has been previously corrected by the Internally Correct Data command. □

The authors wish to thank Dave Dossetter, Product Marketing Engineer, and Dick Pierce, Marketing Applications Engineer, for their invaluable assistance in preparing this article.

intel

ARTICLE REPRINT

AR-271

April 1983

New Bubble-Memory Packaging Cuts Board Space and Manufacturing Costs

Art Thorp
Intel Corp.
Santa Clara, Calif.

ORDER NUMBER: 230642-001

New bubble-memory packaging cuts board space and manufacturing costs

Low-profile 4-Mb bubble-memory package is interchangeable with 1-Mb types and also lets printed-circuit boards be spaced on 0.6-in. centers

by Art Thorp, *Intel Corp., Santa Clara, Calif.*

☐ Designing a second-generation product gives an engineering team the chance to put in all the improvements they realized were needed after the first design was formalized. The new 7114 4-megabit magnetic-bubble memory from Intel makes the most of this opportunity in terms of its ease of both use and manufacturing.

Despite the quadrupled bit density, the 7114's leaded package is smaller in all three dimensions than the leadless package of its 1-Mb predecessor, the 7110. It occupies less space on a printed-circuit board and has a lower profile—low enough for the boards carrying it to fit into adjacent rather than alternate slots in standard card cages. Moreover, chip and package are far easier and cheaper to assemble.

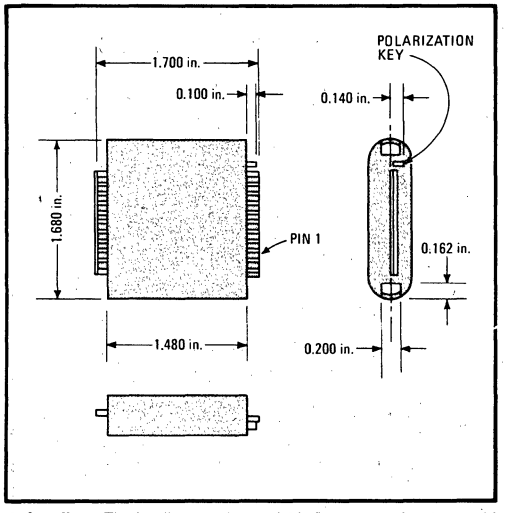
Nor is that convenience compromised by a lack of compatibility with the 7110. The pinouts are the same, and the pin spacings sufficiently similar to make it simple to upgrade from the 7110 to 7114. Also, the support circuits essential to the control of each bubble memory are either identical or so alike as to be interchangeable.

More specifically, the first-generation 1-Mb bubble device is a 520-by-620-mil chip in a leadless package that needs a socket; the assemblage has a footprint of 2.20 by 1.825 in. (Fig. 1) and an overall height of 0.430 in.

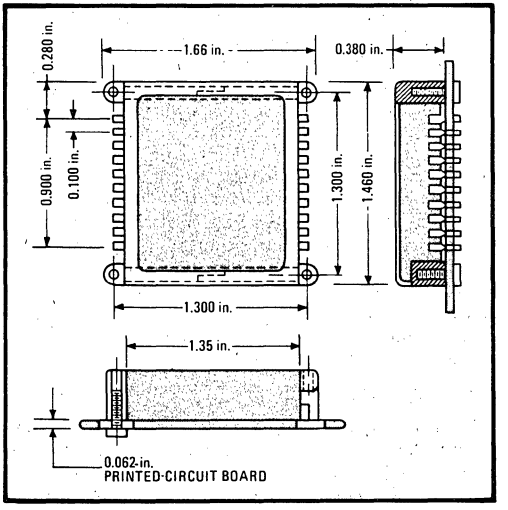
In contrast, the 4-Mb chip and a forthcoming 1-Mb device are smaller—580 by 500 mils—and their leaded "thin-C" dual in-line package has a footprint of only 1.66 by 1.46 in. (Fig. 2). Also, its height is now only 0.375 in., so that when inserted either directly into a pc board or in a zero-profile socket, it leaves ample clearance for the 0.6-in. card spacing normal in commercial card cages.

Design goals

Without scaling down device geometries, it would have been impossible to fit a garnet chip containing four times as many bubble domains into a package of the same size, let alone a smaller one. Thus the first order of business was at least to halve device geometries in both dimen-



1. Leadless. The leadless package of this first-generation 1-megabit magnetic-bubble memory has a footprint of 2.20 by I.825 inches and an overall height of 0.430 in. when socketed. Thus boards cannot be spaced the standard 0.6 in. apart.



2. Thin and leaded. Despite its 4-Mb capacity, this bubble chip fits in a leaded package with only a 1.66-by-1.46 in. footprint and 0.375-in. profile. Cards carrying these packages or using zero-profile sockets for them can be set into a card cage with the standard 0.6-in. spacing.

sions. The production application of X-ray lithography, in fact, yields 4-Mb bubble chips that are smaller than the 1-Mb one in the current 7110 leadless package.

In addition to different die dimensions, the smaller package required smaller magnets and coils with different dimensions to help control the flow of magnetic bubbles on the garnet chip. A program for designing models of coil size and shape was therefore developed and run on an IBM Personal Computer.

Either the 1- or 4-Mb scaled-down bubble device could have been placed in the same package as the original 1-Mb memory if that had been desired. Instead, it was decided not to settle for the existing package but to produce a new one that, while compatible with the 7110, would be more useful to the engineer—namely, by being smaller and allowing standard pc board spacing. Equally important, if not more so, was the decision to make the production process more efficient and cost-effective.

Nevertheless, there were to be no compromises in the stiff specifications for durability, magnetic shielding, and temperature range. In essence, the design goal was for the new package to be at least as good as the first in some aspects and better in others.

A thinsy

One major concern in moving to a new package is its effect on users who are already producing systems containing the first-generation version and who plan to continue manufacturing while introducing the later one. Unless pinout and spacings are absolutely identical, the transition to a new package cannot be totally painless.

However, in this case, maintaining the identical spacings both within and between the two rows of pins would eliminate any benefit gained by a smaller package. Thus, the decision was to keep the 7114's pinout and adjacent pin spacings the same as on the leadless 7110 package. Only the separation between the two rows of pins has been made smaller on the new memories.

As a result those engineers now manufacturing equipment using the previous 7110 model can lay out their pc boards in such a manner as to accommodate either the first-generation package or, with a minimal amount of revision, the new one. The trick is to elongate and drill the pin trace pads on the board for two holes per pin, as shown in Fig 3. For new layouts, this scheme should be used from the very start. Existing system boards can be modified this way with little effort. If board-level diagnostic and maintenance operations require the use of sockets for the packages, the pc-board holes should be dimensioned for the zero-profile–socket contacts, such as Augat Holtite types. Otherwise, the advantage of the 0.6-in. board spacing will be lost.

Attacking manufacturing costs

In many ways, making magnetic-bubble chips is similar to making integrated circuits, but there are significant differences—for instance, semiconductors do not need wire coils. Therefore, it is reasonable to assume that the major manufacturing cost factors in the one process will not be the same for the other.

In the original 7110, the garnet bubble chip is first die-bonded to a ceramic substrate and then wire-bonded to conductors metalized onto the ceramic. Next, the field and drive coils are put in place around the garnet chip, and all the components are potted using a liquid epoxy compound. Both the use of ceramic and the potting process contribute heavily to memory cost.

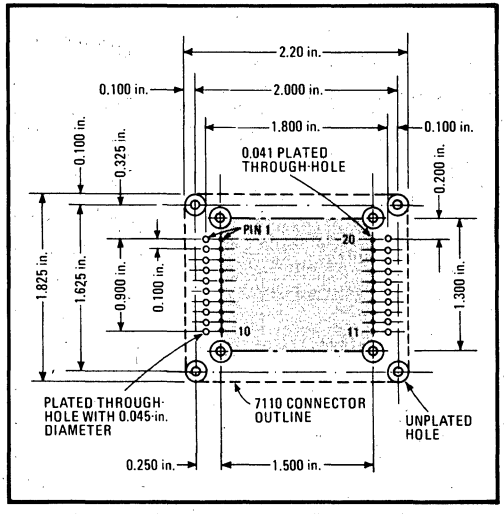
The 7114 package design is more economical on both counts. For the ceramic substrate, the design team substituted a special pc-board material that is thinner, lighter in weight, and lower-cost.

Next, the designers tackled the problem of potting the chip, substrate, and coil combination. Packages containing ICs often employ transfer molding of a thermoset epoxy compound. But when applied to the bubble assembly, the high pressures involved in this process—about 500 to 1,000 pounds per square inch—routinely deformed the bubble memory's wire coils and degraded their electrical performance unacceptably.

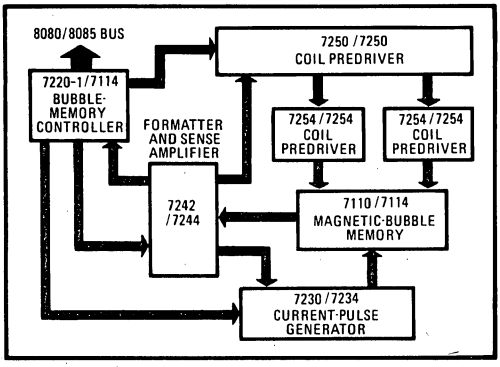
Indeed, for the 7110, manual potting had seemed unavoidable, even though production personnel spent an average of 1 hour on each bubble assemblage, first placing it in a mold, then pouring in liquid epoxy, placing it in a high-temperature oven to cure, removing the mold from the oven, and finally extracting the assembly.

Nonetheless, for the 7114 a fast alternative was found in the liquid injection-molding process used by some manufacturers for high-voltage insulators. Unlike transfer molding, it employs only low pressure, in the region of 15 psi, and it speeds up the potting process to more than 50 devices per hour.

Thus by replacing the ceramic substrate with pc-board material and by substituting liquid injection molding for the manual pouring of a potting compound, the package engineering team had a very favorable impact on the cost and throughput of manufacturing. An added advantage is that, with the ceramic removed, the possibility of chip-



3. Four for one. If this pc board layout is followed, it is possible to plug either a socketed 7110, a leaded 4-Mb, or a leaded 1-Mb bubble-memory package into this hole pattern. The socketed 7110 goes into the outer set of holes, while the newer packages fit the inner set.



4. Upgrade. Along with the bubble chip, all the other ICs of the 1- and 4-Mb memory systems are mechanically and electrically interchangeable so far as the pc board layout is concerned. The diagram shows the chip set for a 4-Mb system in color and the 1-Mb system in black.

ping the exposed edges has been reduced. The table on this page summarizes the packaging and manufacturing aspects of the new and old bubble package methods.

There is always the danger that improving one aspect of a product may inadvertently degrade another. In this case, however, that has not been the result. For example, the original 7110 package was designed to offer protection from external magnetic fields to a level of 20 oersteds. Furthermore, the 7110 is specified to operate over a standard temperature range of 0° to 75°C.

In each case, the new package offers the same or better specifications than the leadless package. In terms of mechanical reliability, both the leadless and leaded package meet and exceed all vibration and shock test limits specified in MIL-STD-883. The leaded package precludes many mechanical problems since it does not depend on a leadless package socket for interfacing with the board.

One for one

The original 1-Mb bubble memory was developed along with a set of support ICs that handled all of its complex timing and drive functions and made its interface with a microprocessor bus indistinguishable from that of any *bona fide* peripheral semiconductor. Considerable engineering effort was applied to make these support circuits interchangeable.

Consequently, any support chip works with any bubble memory. This is in marked contrast to otherwise similar devices that need matched sets of support components.

Intel's 7114 4-Mb device uses the same architecture as the 7110, but now has eight identical sections (called octants) instead of four, and each section is enlarged to store double the number of bubbles it does in the current 7110. The result is a fourfold increase in capacity. However, all of the same pins are brought out in the same order on both bubble memories. Thus the pinout is identical and allows for the use of the same new package for both the new 1-Mb and 4-Mb devices.

Those support ICs that are not affected by the increased capacity, such as the coil predriver (7250) and drivers (7254), are used with either memory.

FIRST- VERSUS SECOND-GENERATION MAGNETIC-BUBBLE MEMORIES

Type	7110 Leadless	7110 Leaded Thin C	7114 Leaded Thin C
Memory (Mb)	1	1	4
Die size (in.)	0.620 by 0.520	0.580 by 0.500	0.580 by 0.500
Substrate material	ceramic	printed-circuit board	printed-circuit board
Potting technique	liquid epoxy by hand	liquid injection molding	liquid injection molding

Those support ICs that have been designed in conjunction with the new memory are the same size and have the same pinouts as their counterparts in the 7110 1-Mb subsystems. What has changed is some of the programmable parameters and the descriptions of their associated registers. These are all involved with the new bubble-memory controller chip—the IC that interfaces the microprocessor bus with the 4-Mb memory.

Examining the effect of these changes in upgrading from a 7110 1-Mb to a 7114 4-Mb package reveals that the modifications are really minimal—the new support ICs can be designed into the same board layouts as their 1-Mb cousins. Users will have only to make some modifications in their software to handle minor differences in addressing and configuration initialization.

Intel's bubble-memory system therefore can have the same configuration whether working with 1-Mb or 4-Mb devices. A single bubble-memory controller acts as the interface between the microprocssor bus and one or more bubble storage subsystems.

The 7224 controller for the 4-Mb bubble-memory device is housed in a standard 40-pin, dual in-line package and takes up about 2 by 0.5 in. on a board. A single controller operates up to eight storage subsystems for a maximum capacity of 4 megabytes.

Each bubble-memory subsystem contains a monolithic formatter and sense amplifier in a standard 20-pin DIP; a current-pulse–generator chip in a 22-pin DIP; a coil-predriver chip in a 16-pin DIP; a pair of quad V-groove MOS driver chips, each in a 14-pin DIP; and of course the bubble device itself. One subsystem takes up less than 3 by 4 in., and a 4-megabyte board of eight of them plus a a controller could be constrained to 6.75 by 12 in.

Obviously, boards laid out for the 7110 1-Mb memory and its support family would be approximately the same size for one fourth the amount of memory. However, in a card cage with a standard 0.6-in. spacing, boards built using the original leadless packages could not be stacked in adjacent slots.

Converting from the earlier 7110 1-Mb to a 4-Mb system essentially requires modifications to the pin pads of the 7110's leadless package. The 7220-1 bubble-memory controller is the same size and has the same pinout as the 7224. The same is true for the 7242 formatter and sense amplifier and its 7244 replacement, as well as for the 7230 current-pulse generator and its 7234 substitute. Figure 4 shows how an identical board can support either a 7114 4-Mb or a leaded or leadless 7110 1-Mb system.

Space limitations, interface details, and other such criteria will typically dictate the actual board layout. □

intel

ARTICLE REPRINT

AR-272

April 1983

Bubble-Memory Support Chips Allow Tailored-System Design

Richard Pierce
Applications Engineer
Intel Corporation

APRIL 1983
ORDER NUMBER: 230645-001

Bubble-memory support chips allow tailored-system design

Using special support chips gives you flexibility in designing a bubble-memory system. And understanding design tradeoffs helps configure a system that best suits your needs.

Richard Pierce, Intel Corp

Designing a bubble-memory system—with its advantages of small size, high reliability and nonvolatility—is easy when you use system support devices. Such a family of LSI chips allows you to tailor a system to meet requirements on specs such as access time and power consumption and to modularly expand the system for increased storage capacity. How you configure and use a bubble-memory system involves tradeoffs, though, and balancing those tradeoffs requires knowledge of the intended system application. This article discusses bubble-system tradeoffs and other design considerations and presents a specific design using Intel's 1M-bit bubble-memory device and family of support chips.

Interface appears as a peripheral controller

The devices available for building a 1M-bit bubble-memory system are the 7110 magnetic-bubble module, the 7220-1 bubble-memory controller, the 7242 formatter/sense amplifier, the 7230 current-pulse generator and the 7250 coil predriver (see **box,** "Bubble-memory devices"). Communication with the 7220-1 controller (**Fig 1**) is the key function, because this device provides the bubble memory's only interface with the outside world. It allows you to interface with a bubble-memory system through a standard μP bus just as with any peripheral. Software directs the controller to choose one of three memory-access methods: direct memory access (DMA), interrupt or poll.

The 7220-1 bubble-memory controller (BMC) has several functional sections. The system-bus interface provides an asynchronous interface to a host processor, transferring 12.5k bytes/sec with a 4-MHz system clock. The controller also has a 40-byte first-in first-out (FIFO) memory buffer through which data passes on its way to the 7242 formatter/sense amplifier (FSA). This FIFO's primary purpose is to reconcile timing differences between the user and the FSA, because the system bus typically can transfer data much more rapidly than the bubble memory can.

The BMC's DMA and interrupt logic handle data transfers. In addition, an internal register file contains six user-accessible 8-bit registers: a command register, a status register and four registers that store information pertaining to the system's operational mode and configuration. The command register accepts user-issued codes that initiate data transfers, and the status register indicates the BMC's current state.

The remaining functional sections of the BMC

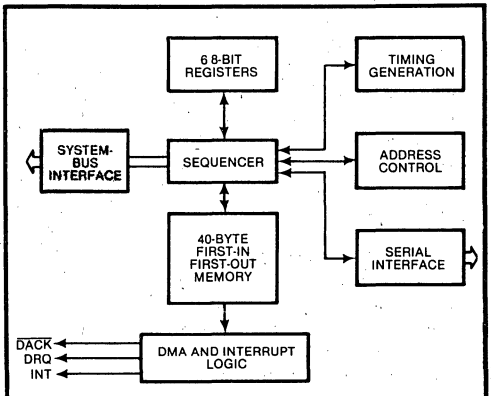


Fig 1—A bubble-memory controller, *which interfaces to a host μP through a system bus, generates all support-circuit timing and control signals, maintains memory-address information and performs data conversions for transfers to a bubble-memory module.*

Communicate with a bubble memory as with a peripheral controller

generate all the support-circuit timing and control signals, maintain memory-address information and perform parallel-to-serial and serial-to-parallel conversions for transfers to the bubble-memory module.

Interface circuitry depends on transfer mode

The type of circuitry you design to interface with the BMC depends on the mode of data transfer chosen. For a polled implementation, the requirements reduce to interfacing to a standard μP bus. One such interface design, for the 8088, appears in **Fig 2;** it consists of address-decode logic, data-bus-decode and buffering logic, a clock circuit and miscellaneous control logic. The clock circuit must provide a 4-MHz (±0.1%) system clock with a 50% (±5%) duty cycle.

Fig 2's system operates from 12 and 5V only, and one important part of this design is its automatic power-fail circuitry **(Fig 3),** which monitors these voltages. An important aspect of bubble-memory devices is that the coil drive current (which moves bubbles around within the device) must always have the proper phase and amplitude, without transients, to ensure data integrity. **Fig 3**'s power-fail circuit prevents transients and—when voltages drop 6% below normal level—stops the coil currents while maintaining the proper phase. You can also expand the power-fail circuit to include recommended features such as ac power-fail and ac or dc overvoltage protection.

Bubble-memory devices

Several different devices **(figure)** make up a bubble-memory system. At the heart of the system is the 7110 magnetic-bubble memory (MBM), with a user data capacity of 1M bits (128k bytes).

This chip embodies a major-track/minor-loop architecture, in which bubbles serially propagate into and out of the memory tracks and get stored in minor (storage) loops (EDN, September 1, 1982, pg 198). This organization permits creation of redundant storage loops, increasing device yield by tolerating defects in as many as 15% of the loops. Testing by the manufacturer detects the unusable loops and writes a map—showing the usable loops—into an additional storage loop called a boot loop. The map also appears on the 7110's label.

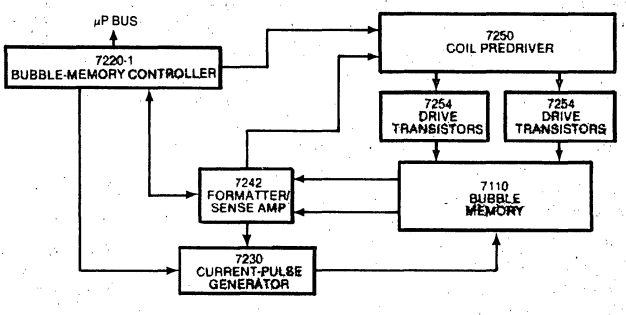
Internally, the 7110 consists of two identical 512k-bit sections. Although these sections are essentially independent, they operate simultaneously to give a factor-of-two data-rate improvement.

User interface to the system occurs through a 7220-1 bubble-memory controller (BMC). This device provides the system-bus interface, performs serial-to-parallel and parallel-to-serial data conversions, generates all timing and control signals necessary for proper operation of support circuitry and interprets and executes user requests for data transfers. The BMC's interface makes the bubble-memory system look like a peripheral to a μP-system bus.

The 7242 formatter/sense amplifier (FSA) interfaces independently to each half of the bubble memory. Its integrated sense amplifier accepts low-level voltage signals from the MBM's bubble detectors during read operations, and the device also performs data-formatting tasks that include the transparent handling of the MBM's redundant loops. In addition, the FSA sends TTL-level control signals to the 7230 current-pulse generator (CPG) during write operations.

The 7230 supplies the current pulses that generate bubbles in the MBM and transfer them into and out of the storage loops. The CPG's integrated power-fail-detection circuitry initiates an orderly shutdown of the current sources when power fails.

The 7520 coil predriver (CPD) interfaces the 7220-1 BMC to the two 7254 drive transistors, which supply the relatively high peak currents required by the 7110 MBM's X and Y coils. These currents induce the in-plane rotating magnetic field that moves the magnetic bubbles.



Support chips *allow the design of a complete bubble-memory system around Intel's 7110 magnetic-bubble memory device.*

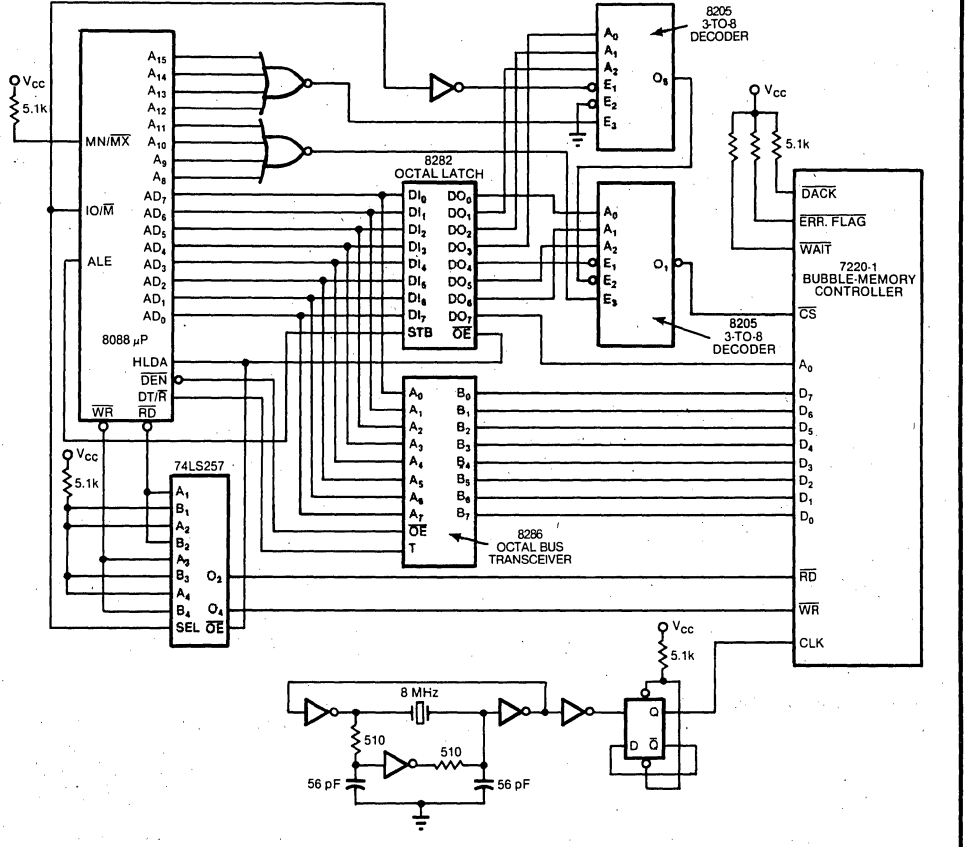


Fig 2—Interfacing a bubble memory *can be almost as simple as interfacing a standard μP bus. This interface connects a 7220-1 bubble-memory controller with an 8088 μP.*

Guaranteeing power-supply requirements is an important part of the design process. The supply voltages must be within 5% of their specified values, and the power-off/power-fail decay rates (**Table 1**) must allow 150 μsec max for an orderly shutdown and reset of all support circuits. No restrictions apply to voltage rise times or sequencing.

A simple calculation determines your system's required storage capacitance to achieve **Table 1**'s voltage decay rates. The worst-case power-supply capacitance requirement is

$$\frac{Q_{MAX}}{V_{MIN}} = \frac{\Delta I_{MAX} \Delta T_{MAX}}{\Delta V_{MIN}}$$

Typical capacitance values for a system with one bubble-memory module, excluding any additional current drain from unrelated circuitry, are 805 and 350 μF for the 5 and 12V supplies, respectively.

Another important design factor in custom bubble-memory boards is careful circuit layout to minimize interference from the 7110's large drive signals with

Choose a memory-access method: direct, interrupt or poll

nearby small sense signals. The pin assignments of the 7110 and its support devices optimize board layout and maximize circuit density, and the package layout used in Intel's Bubble Prototype Kit (BPK-72) helps ensure error-free operation. As shown in **Fig 4**, this layout places all support circuits near the 7110; note particularly that the 7110 and the 7242 FSA must not be physically separated.

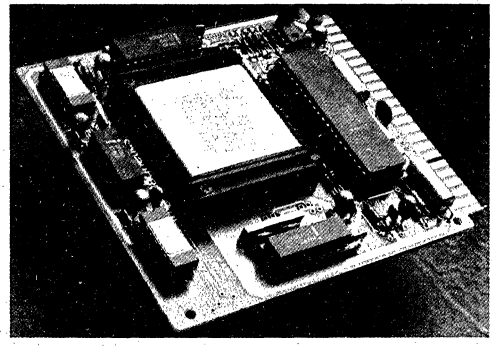


Fig 4—Proper component layout *helps ensure error-free bubble-memory operation. In this Intel Bubble Prototype Kit (BPK-72), all support circuits are close to the 7110 memory module.*

Software controls the system

Software is a vital part of bubble-memory design, too; it's the major contributor to a system's efficient and reliable operation. Software interface modules (often called bubble drivers) accept processor-issued commands and control and monitor command execution; they also return status information to the processor.

Software drivers depend on the mode of data transfer: interrupt-driven I/O, DMA or polled. In a DMA implementation, the software need only set up the DMA controller's memory-address and transfer counts and initiate the data transfer; Intel's 8257 DMA-controller hardware automatically handshakes with the μP and BMC to perform each transfer.

In the interrupt mode, on the other hand, the software is responsible for performing memory-read and -write operations to transfer 22-byte data blocks to and from the BMC's FIFO on receipt of an interrupt. A BMC output signal—typically wired as a second-level interrupt—indicates when the BMC's FIFO becomes half full (during read operations) or half empty (during write operations).

The third I/O method—polling—is similar to the interrupt-driven mode except that in it, data moves one byte at a time. The software determines when to transfer data by continually polling a bit in the BMC's status register. This status bit indicates the presence or

TABLE 1 — BUBBLE-MEMORY POWER-SUPPLY REQUIREMENTS

VOLTAGE	MARGIN	POWER-OFF/POWER-FAIL DECAY RATE
12V	±5%	< 1.10 V/mSEC
5V	±5%	< 0.45 V/mSEC

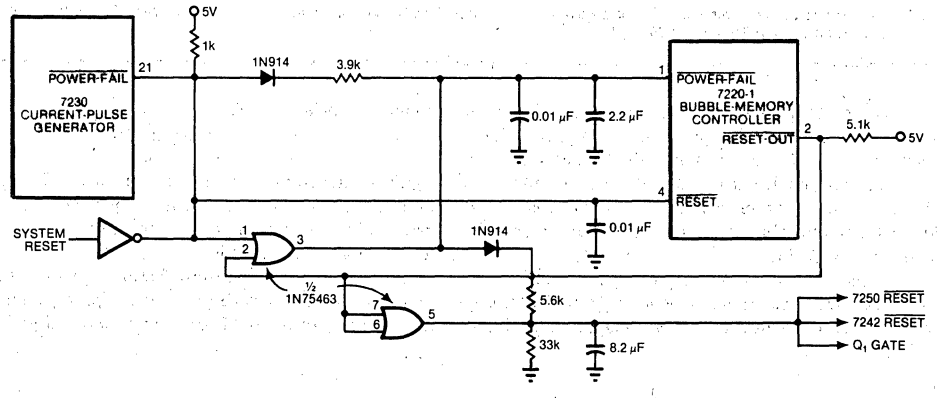


Fig 3—Power-fail circuitry *monitors the 12 and 5V supplies for* **Fig 2***'s circuit and maintains data integrity in case of power failure by stopping coil currents in the proper phase.*

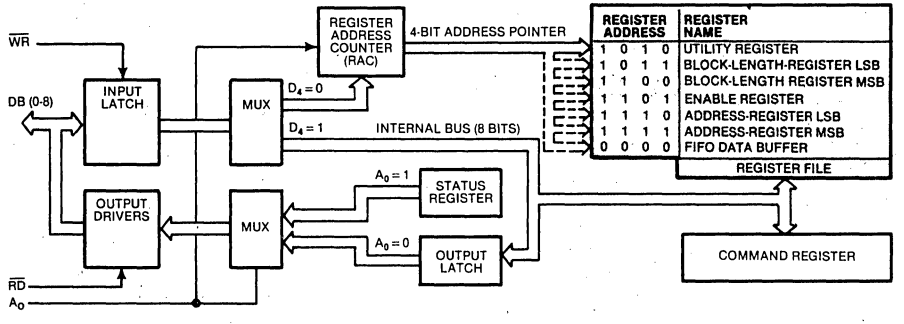


Fig 5—Software accesses the bubble-memory controller *through a system-bus interface consisting of two I/O ports. A ONE on address line A_0 selects the command/status port, and a ZERO selects the bidirectional data port.*

absence of data—to be read or written by the host processor—in the BMC's FIFO.

Although the polling mode is simple to implement, its software requirements are the most demanding. Because data transfers one byte at a time, the software must continually monitor the status register to ensure that the FIFO doesn't underflow or overflow with data.

Each of these data-transfer modes has unique advantages that must be weighed with each particular application. The DMA mode, for example, permits the processor to continue executing instructions while a transfer is in progress. The interrupt and polled modes, on the other hand, offer lower cost but are often too slow in systems incorporating multiple bubble devices connected in parallel. Eight 7110 bubble modules in parallel can transfer data at rates as high as 100k bytes/sec, and these high-performance systems normally use the DMA mode.

Understand protocols and definitions

Developing the software drivers for any bubble-memory system requires a clear understanding of command protocols and register definitions. The software communicates with the bubble-memory controller through the system-bus interface, consisting of two I/O ports selected by the state of the least significant address line (**Fig 5**). When A_0 is ONE (active), the command/status port gets selected; when it's ZERO (inactive), the bidirectional data port gets selected.

The command/status port serves a dual function: Reading the port accesses the status register, and writing to the port accesses a register determined in part by data bit D_4. This register is the command register if D_4 is ONE; it's one of the six parametric registers or the BMC's FIFO if D_4 is ZERO. In the latter case, the contents of a register-address counter (RAC) specify one of the seven possibilities.

The command register accepts one of 16 interface commands. The initiation of nondata transfers occurs merely by writing the proper command code to the command register, while a data transfer (initialize or read/write) requires prior loading of the parametric registers. Those registers specify the operating mode and system configuration, plus the data transfer's starting address and length (in 64-byte pages). After loading of the command register, the BMC automatically executes the command. An indication of success or failure returns in the status register.

An autoincrementing feature in the RAC facilitates loading the parametric registers before data-transfer commands. After the processor loads a value—specifying a particular parametric register—into the RAC via the command port, the next write operation to the data port accesses that register. Each write operation also increments the RAC, so each subsequent operation accesses the next register in sequence. After incrementing to address 0 (the BMC's FIFO), however, the RAC stops incrementing and continues to point to the FIFO until modified by software.

Software drivers perform data transfers

An example set of BMC software drivers (**Fig 6**) shows how data-transfer operations occur in the polled mode. The three drivers perform a power-up procedure, write registers, and read and write data.

The power-up procedure (**Fig 6a**) enables bubble-memory support devices in an orderly fashion and permits subsequent reading and writing of memory

Meet power-supply requirements to ensure reliable operation

pages. The first command it issues is Abort; the host then loads the parametric registers with appropriate values for a 1M-bit system configuration and issues an Initialize command. During command execution, the host processor constantly polls the BMC's status register to determine when the command finishes. An additional status-register check determines whether the command and/or data transfer is successful.

Successful completion of the Initialize command

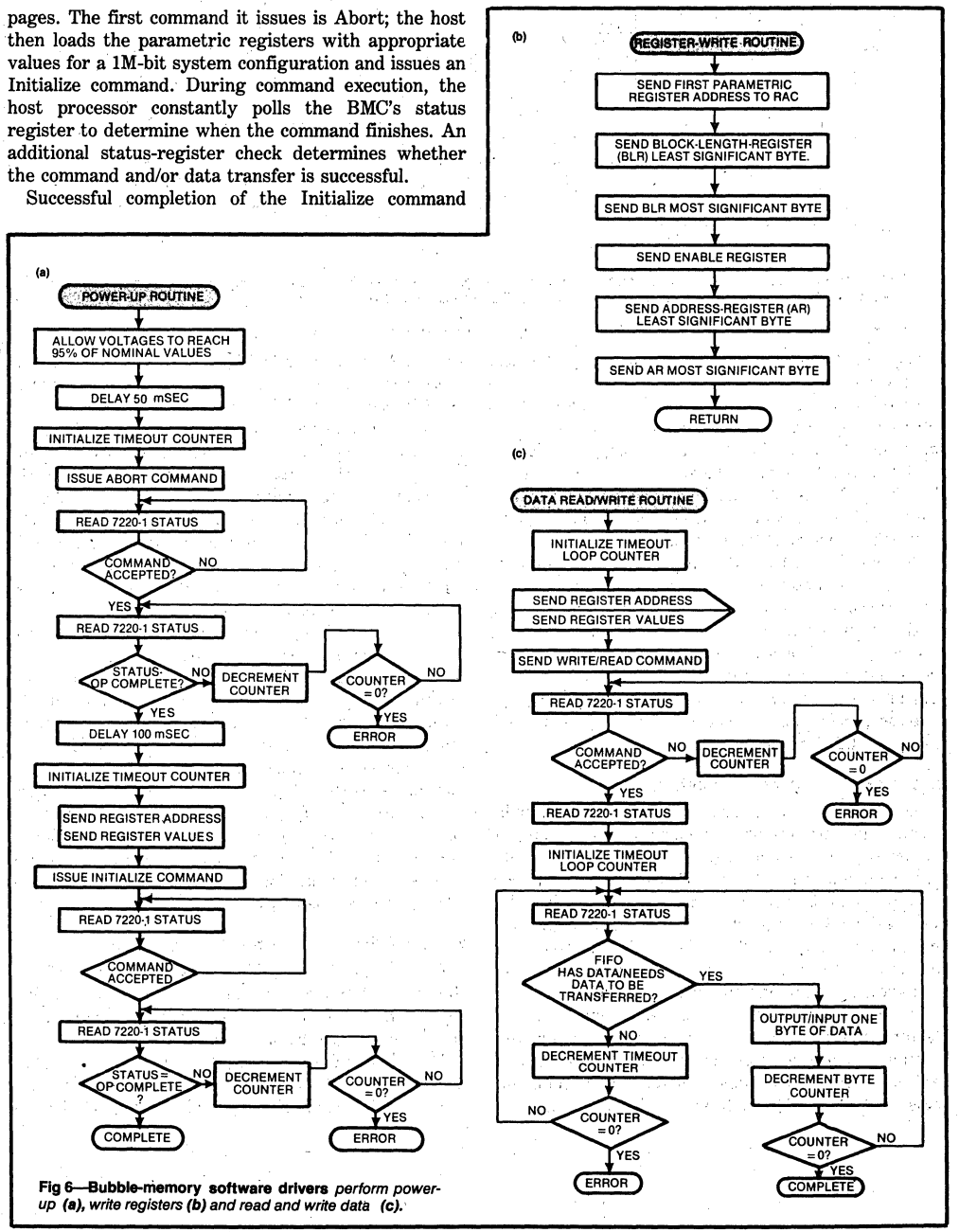


Fig 6—Bubble-memory software drivers *perform power-up* **(a)**, *write registers* **(b)** *and read and write data* **(c)**.

Proper physical layout helps reduce errors

indicates that the system is ready to transfer data. To initiate a data transfer, the host loads the parametric registers (**Fig 6b**) with the memory-page address and number of pages to transfer and then issues a read or write command (**Fig 6c**). This command transfers data between the system bus and the BMC's FIFO at a rate of 80 μsec/byte, excluding access time. The software polls the status register to determine when to transfer each byte and also maintains a count of the bytes transferred. In systems operating in the DMA or interrupt mode, however, maintaining this count is unnecessary.

Of course, all software driver routines should contain error-handling capabilities, but the examples shown here ignore these capabilities for the sake of simplicity. Numerous error-correction options exist in a bubble-memory system, though, and all are selectable under software control. As an example, the 7242 formatter/sense amplifier uses a 14-bit error-correction code with each 256-bit block of data. This code can correct all single-error bursts of five bits or less, improving the data error rate by several orders of magnitude while remaining user transparent.

Expand memory in modules

Having considered a bubble-memory-system design, turn to memory-expansion considerations. One BMC can control as many as eight bubble-memory modules, and with multiple BMCs, you can configure even larger systems (**Fig 7**). A memory module—providing expansion in increments of 128k bytes—consists of a 7110 bubble device, one 7230 current-pulse generator, one 7250 coil predriver, a 7242 formatter/sense amplifier and two 7254 drive-transistor packages.

Expansion can occur in three ways. The first approach uses the BMC's built-in ability to handle multiple bubble devices. This scheme relies on the BMC to time-slice the serial bus between the BMC and the appropriate number of 7242 FSAs in the system and to output appropriate control signals to the 7242, the 7230 and the 7250. Data flow in this expanded system is similar to that in a single-module system.

A second expansion approach takes advantage of provisions in the BMC for paralleling controllers. This approach provides a greater word width at the system bus and still allows each BMC to accommodate as many as eight bubble devices.

A third option switches banks of bubble devices into or out of a circuit under external control. (Switching is possible because each support device has a chip-select pin.) The maximum number of devices in each bank remains eight, but the number of banks is unlimited. You can even multiplex entire subsystems of this type, because the BMC chip itself has a chip-select input.

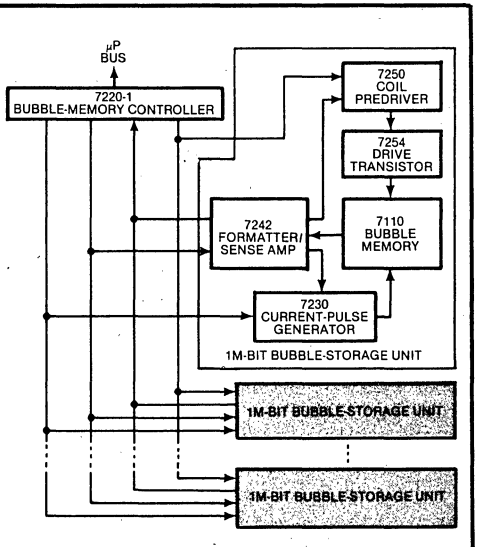


Fig 7—Bubble-memory expansion *occurs in increments. Each 7220-1 controller can handle as many as eight 128k-byte modules.*

Nonvolatility permits power savings

Your design of a bubble-memory system can also take into account the system's nonvolatility in order to minimize power consumption. Consider, for example, a bubble system for a portable terminal that stores inventory and sales figures and periodically transmits them to a central computer. The bubble memory in such a terminal requires power only during memory access, so you can remove power from the bubble system during much of the time that the terminal is in use.

In some applications—for example, systems that only occasionally transmit a small amount of data—you can also reduce power consumption by using a faster initialization scheme. The usual initialization procedure, which reads boot-loop information from a bubble device to identify redundant storage loops, requires as much as 160 msec per device—a significant percentage of power-on time for such systems. If you store the boot-loop information in EPROM, though, you can download it from there much more rapidly and thus with less power.

Design tradeoffs occur at system level

Still other design considerations exist at the system level; design tradeoffs concern such factors as memory capacity, access time, data rate and power consump-

Software interface modules contribute to system efficiency

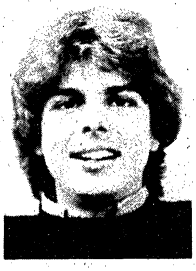
TABLE 2 — BUBBLE-MEMORY PERFORMANCE PARAMETERS

	ONE MBM	FOUR MBMs	EIGHT MBMs OPERATED IN PARALLEL	EIGHT MBMs MULTIPLEXED ONE AT A TIME
CAPACITY	128k BYTES	512k BYTES	1M BYTES	1M BYTES
NOMINAL DATA RATE	68 kHz	272 kHz	544 kHz	68 kHz
AVERAGE ACCESS TIME	48 mSEC	48 mSEC	48 mSEC	48 mSEC
POWER DISSIPATION (100% DUTY FACTOR)	6W	20W	40W	11W
STANDBY POWER	1.55W	3.7W	7.0W	7.0W
BOARD AREA	16 IN.^{2}	45 IN.^{2}	90 IN.^{2}	90 IN.^{2}

tion. As **Table 2** shows, the 7110 bubble module's access time averages 48 msec (7.4 msec best case, 80 msec worst case), independent of the number of modules in a system. You can increase the data rate, however, by operating the devices in parallel. The approach requires more power, but using eight 7110s instead of one increases the nominal bit rate from 68 to 544 kHz.

Finally, you can improve the overall data rate by reducing the average time required to access a memory page. The access time for any single page is random, but no access delay occurs for succeeding pages. Thus, in time-critical applications where successive page accesses aren't random, least recently used (LRU) and lookahead algorithms can help reduce page-access time. **EDN**

Author's biography



Richard Pierce works in Intel's Nonvolatile Memory Div (Santa Clara, CA), supplying customer support and developing new applications. He holds a BSEE degree from Purdue University and previously worked for Cincinnati Electronics. Dick is a member of the IEEE, and his hobbies include biking, photography and skiing.

BPK 5V74
4MBIT BUBBLE MEMORY SUBSYSTEM

BPK 5V74-4	10°C To 55°C

- **4 Mbit (512K Bytes) Non-Volatile, Solid-state, Read/Write Bubble Memory Subsystem**
- **Interfaces to Host Microprocessor Via Additional Bubble Memory Controller**
- **Contains Bubble Memory and ICs for Production with 4Mbit Bubble Memory**
- **Modularity Provides Expansion Up to Eight Subsystems Per Controller**
- **Maximum Data Rate of 200K bit/sec with One Subsystem**
- **Maximum Data Rate of 1.6M bit/sec with Eight Subsystems in Parallel and Time Multiplexed**
- **Average Random Access Time of 88 ms**
- **Bubble Memory in Leaded Package**

The BPK 5V74 Bubble Memory Subsystem is a modular building block used to design bubble memory systems. In a complete bubble memory system, an 7224 Bubble Memory Controller (BMC) interfaces the BPK 5V74 subsystem to the host processor.

The modular Intel subsystem provides a path for density expansion. One BMC can interface up to eight 4 Mbit subsystems. Thus, a 4 MBit (512 KByte) system can be expanded up to a 32 MBit system by adding subsystems. BMC's can be combined in parallel to further expand the system memory capacity.

Together, the BPK 5V74 Bubble Storage Subsystem and a 7224 controller provide a reliable mass storage system for any application. This bubble memory system can be customized to the particular layout and form factor of many different systems.

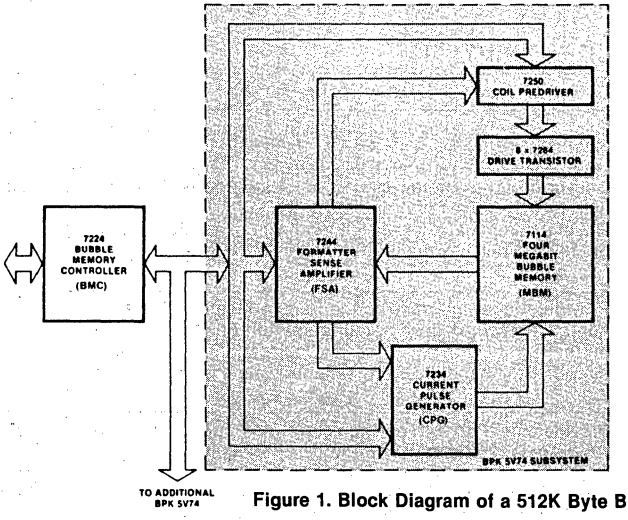


Figure 1. Block Diagram of a 512K Byte Bubble Storage System

OCTOBER 1983
ORDER NO. 210804-002

FUNCTIONAL DESCRIPTION

An BPK 5V74 subsystem and a 7224 controller comprise a complete bubble memory system. The 4 MBit BMC, the 7224, provides the interface between the host microprocessor and the bubble memory subsystem and provides all the timing and control signals to the subsystem. The user interface of the BMC is compatible with microprocessor bus systems for 8080, 8085, 8086, 8088, 80186, 80286 and other standard microprocessors. The BMC is a software driven device utilizing 18 convenient commands. The design engineer's primary responsibility is interfacing to the BMC. This is comparable to interfacing a disk drive controller.

The BPK 5V74 consists of one 4 MBit Magnetic Bubble Memory (MBM) and additional support IC's (see Figure 1). These are the basic components to build a non-volatile, solid-state, read/write military memory system utilizing 4 MBit bubble memory. The bubble memory is in a leaded package. The complete family of LSI support circuits has been designed to handle the complex analog interface associated with bubble devices. The immediate support circuitry for the MBM consists of — an 7250 Coil Predriver (CPD), eight 7264 MOS FETs Transistor Packs, an 7234 Current Pulse Generator (CPG), and an 7244 Formatter/Sense Amplifier (FSA).

Data integrity is insured by the automatic error correction designed into the BPK 5V74.

The average random access time of a 4 MBit subsystem is 88 ms with a 200Kbit/sec maximum data transfer rate. Operating several subsystems in parallel, the BMC uses time division multiplexing. Therefore, the maximum data rate increases correspondingly for the whole system.

Operating subsystems serially, one MBM being accessed at a time, the maximum data transfer rate is still 200Kbit/sec. If low power consumption is a critical design goal, the bubble memory subsystem can be powered down when it is not being accessed, thus reducing the average power consumption.

The data in the 4 Mbit subsystem is organized in 8192 pages, each with 64 bytes. Conceptually, the data organization with pages is analogous to a disk system's sectors. In system's with multiple bubble memories, the page size can vary from 64 bytes to 512 bytes depending on the number of subsystems and if the subsystems are operating in parallel or serially, being accessed one at a time.

The BPK 5V74 subsystem has matched components. Each of the components in the subsystem is described in more detail in the rest of this data sheet.

BPK 5V74 FUNCTIONAL DESCRIPTION

Item	Description	Part Number
4 MBit Bubble Memory	20-pin leaded package which provides 4 megabit of non-volatile storage.	7114
Current Pulse Generator	Converts digital timing signals to analog current pulses suited to the drive requirements of the MBM. The CPG provides the replicate, swap, generate, boot replicate, and bootswap pulses required by the MBM. 22 Pin DIP Package.	7234
Dual Formatter/Sense Amp	Provides direct interface to the Bubble Memory. The FSA contains on-chip sense amplifiers, a full FIFO data block buffer, burst error detection and correction circuits, and circuitry for handling of the bubble memory redundant loops. 20 Pin DIP package.	7244
Coil Predriver.	Provides the high voltage, high current outputs to drive the Quad VMOS transistors. 16 Pin DIP package.	7250
VMOS Coil Drive Transistors (8)	Switches the required current to drive the X and Y coils of the Bubble Memory. 3 Pin Discrete.	7264

For additional packaging information see the packaging information section.

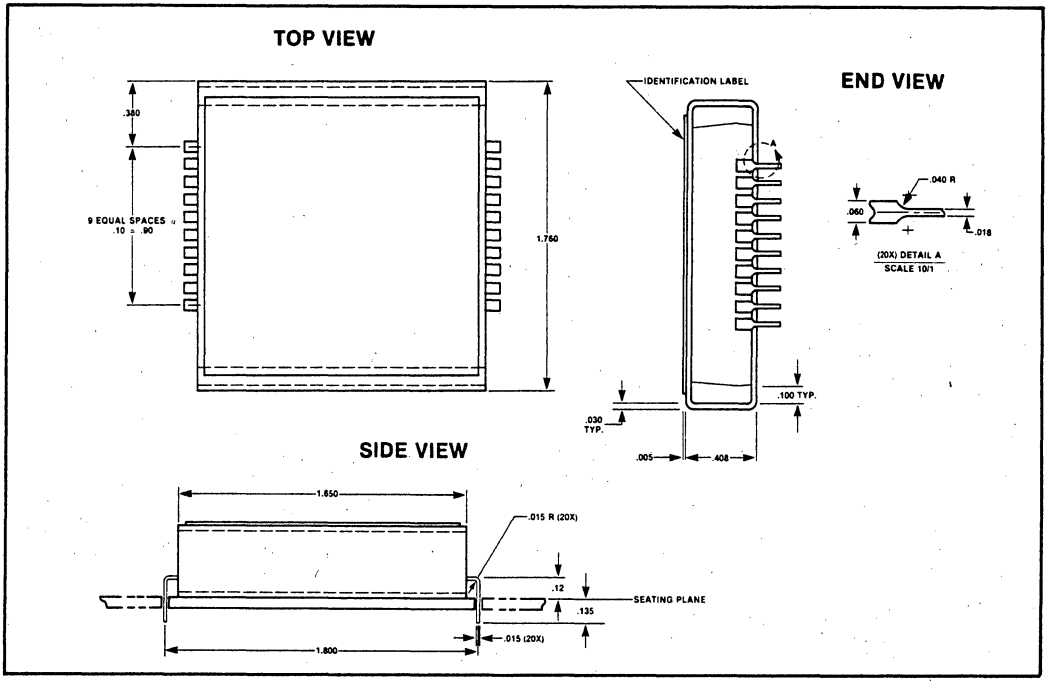


Figure 2. 4MBit Leaded Bubble Memory Package

BPK 5V74 TEMPERATURE RANGE

Bubble Memory Temperature Ranges		Support Circuits Min. Operating Temperature	Description
Operating	Non-Volatile Storage		
10° to 55°C Case	−20 to +75°C	10° to +55°C Ambient	4 Mbit Bubble Storage Sub-System

SPECIFICATIONS

Capacity

512K Byte per BPK 5V74
Maximum of 8 BPK 5V74 per 7224 Controller

Performance

Avg. Access Time ..... 88 msec

Data Organization

64 bytes per page
8192 pages per BPK 5V74

Addressing Scheme

Logical page number

Environmental

Temperature: See Ordering Information
Operating Humidity: 0—95% Non-Condensing

DATA TRANSFER RATES (Examples of System Configurations)

Parameter	One BPK 5V74 Unit	Four BPK 5V74 Operated in Parallel[1]	Eight BPK 5V74 Operated in Parallel[1]	Eight BPK 5V74 Multiplexed One at a Time[1]
Capacity	512 kilobytes	2048 kilobytes	8 megabyte	8 megabyte
Average Data Rate (kilobits/sec)	136	544	1088	136
Maximum Date Rate (kilobits/sec) (Burst)	200	800	1600	200

NOTE:
1. Multiple Bubble subsystems can be operated in parallel for maximum performance or multiplexed to conserve power.

BPK 5V74 POWER SUPPLY REQUIREMENTS

Voltage	Margin	Power Off/Power Fail Decay Rate
+12 Volt	±1%	less than 1.10 volts/msec
+5 Volt	±5%	less than 0.45 volts/msec

- Voltage sequencing — no restrictions
- Power on voltage rate of rise — no restrictions
- The power supply requirements based on recommended power fail circuitry as shown in Figure 3.
- The 12V ±1% may be supplied by:
  1. Using such power supply.
  2. Use voltage regulator with 5V ± 5% input and 12V ±1% output as used in BPK 5V75 prototype kit. Circuitry — See Figure 4.

BPK 5V74 POWER CONSUMPTION

(Includes 7114, 7234, 7244, 7250, 7264)

Standby	Typical:	0.7 W
	Maximum:	1.8 W
Active	Typical:	3.7 W
	Maximum:	6.2 W

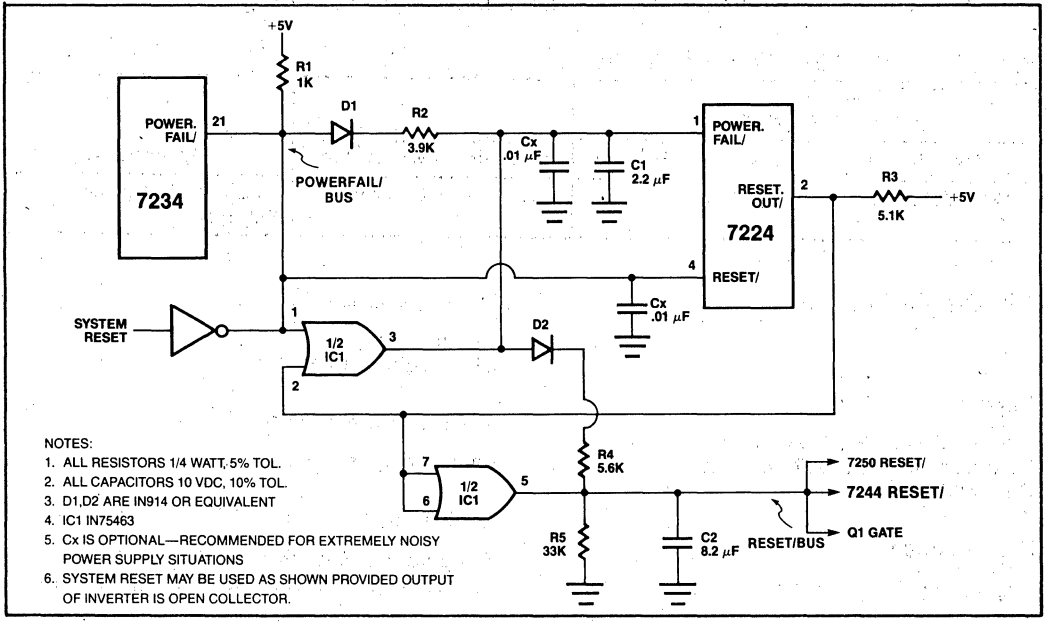


Figure 3. Power Fail Circuit

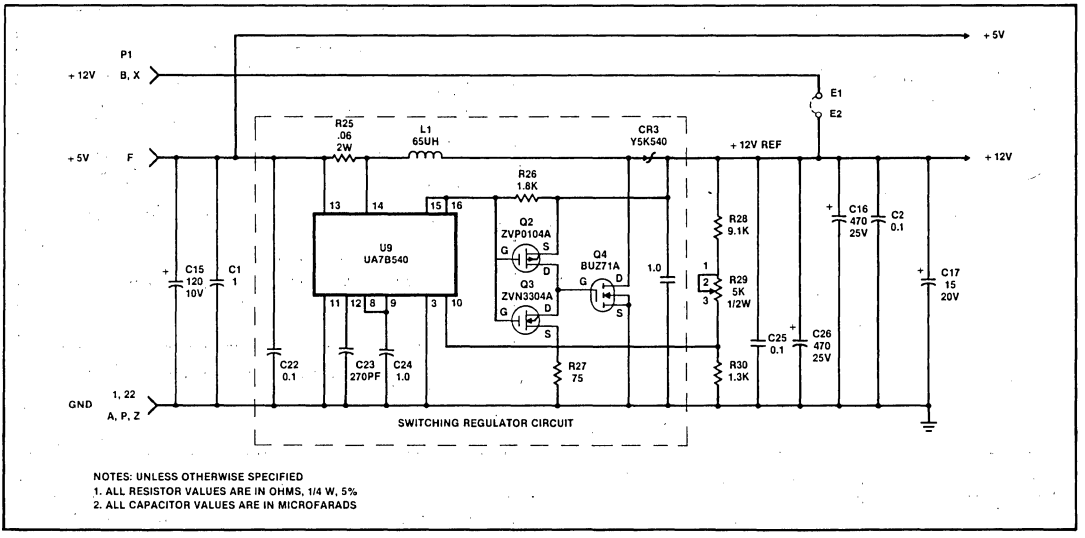


Figure 4. BPK 5V75 Voltage Regulator Circuit

PIN DESCRIPTIONS

7114

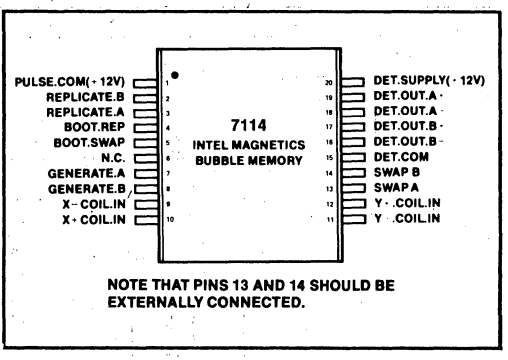


Figure 5. 7114 Pin Configuration

Table 1. 7114 Pin Description

Symbol	Pin No.	I/O	Source/Destination	Description
BOOT.REP	4	I	7234 CPG	Two-level current pulse input for reading the boot loop.
BOOT.SWAP	5	I	7234 CPG	Single-level current pulse for writing data into the boot loop. This pin is normally used only in the manufacture of the MBM.
DET.COM	15	I		Ground return for the detector bridge.
DET.OUT	16 — 19	O	7244 FSA	Differential pair (A+, A− and B+, B−) outputs which have signals of several millivolts peak amplitude.
DET.SUPPLY	20	I		+12 volt supply pin.
GEN.A and GEN.B	7, 8	I	7234 CPG	Two-level current pulses for writing data onto the input track.
PULSE.COM	1	I		+12 volt supply pin.
REP.A and REP.B	3, 2	I	7234 CPG	Two-level current pulses for replicating data from storage loops to output track.
SWAP.A and SWAP.B	13, 14	I	7234 CPG	Single-level current pulse for swapping data from input track to storage loops.
X−.COIL.IN. X+COIL.IN.	9, 10	I	7264	Terminals for the X or inner coil.
Y−.COIL.IN. Y+.COIL.IN.	11, 12	I	7264	Terminals for the Y or outer coil.

7234

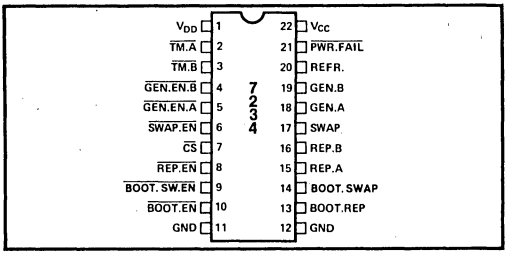


Figure 6. 7234 Pin Configuration

Table 2. 7234 Pin Description

Symbol	Pin No.	I/O	Source/Destination	Description
$\overline{\text{BOOT.EN}}$	10	I	7224 BMC	An active low input enabling the BOOT.REP output current pulse.
BOOT.REP	13	O	7114 MBM	An output providing the current pulse for bootstrap loop replication in the bubble memory.
BOOT.SWAP	14	O	7114 MBM	An output providing a current pulse which may be used for writing data into the bootstrap loop.
$\overline{\text{BOOT.SW.EN}}$	9	I	7224 BMC	An active low input enabling the BOOT.SWAP output current pulse.
$\overline{\text{CS}}$	7	I	7244 FSA	An active low input for selecting the chip. The chip powers down during deselect.
GEN.A	18	O	7114 MBM	An output providing the current pulse for writing data into the "A" quads of the bubble memory.
GEN.B	19	O	7114 MBM	An output providing the current pulse for writing data into the "B" quads of the bubble memory.
$\overline{\text{GEN.EN.A}}$	5	I	7244 FSA	An active low input enabling the GEN.A output current pulse.
$\overline{\text{GEN.EN.B}}$	4	I	7244 FSA	An active low input enabling the GEN.B output current pulse.
$\overline{\text{PWR.FAIL}}$	21	O	7224 BMC	An active low, open collector output indicating that either V_{CC} or V_{DD} is below its threshold value.
REFR.	20	I	External Resistor	The pin for the reference current generator to which an external resistance must be connected.
REP.A	15	O	7114 MBM	An output providing the current pulse for replication of data in the "A" quads of the bubble memory.
REP.B	16	O	7114 MBM	An output providing the current pulse for replication of data in the "B" quads of the bubble memory.
$\overline{\text{REP.EN}}$	8	I	7224 BMC	An active low input enabling the REP.A and REP.B outputs.
SWAP	17	O	7114 MBM	An output providing the current pulse for exchanging the data between the input track and the storage loops in the bubble memory.
$\overline{\text{SWAP.EN}}$	6	I	7224 BMC	An active low input enabling the SWAP output.
$\overline{\text{TM.A}}$	2	I	7224 BMC	An active low timing signal determining the cut pulse widths of the BOOT.REP, GEN.A, GEN.B, REP.A and REP.B outputs.
$\overline{\text{TM.B}}$	3	I	7224 BMC	An active low timing signal determining the transfer pulse widths of the BOOT.REP, GEN.A, GEN.B, REP.A and REP.B outputs.

7244

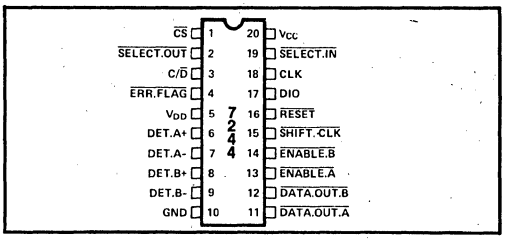


Figure 7. 7244 Pin Configuration

Table 3. 7244 Pin Description

Symbol	Pin No.	I/O	Source/Destination	Description
C/$\overline{D}$	3	I	7224 BMC	Command/$\overline{\text{Data}}$ signal. This signal shall cause the FSA to enter a receive command mode when high and to interpret the serial data line as data when low. Any previously active command will be immediately terminated by C/$\overline{D}$.
CLK	18	I	Clock	Same TTL-level clock used to generate internal timing as used for 7220-1.
$\overline{CS}$	1	I	External	An active low signal used for multiplexing of FSAs. The FSA is disabled whenever $\overline{CS}$ is high (i.e., it presents a high impedance to the bus and ignores all bus activity).
$\overline{DATA.OUT.A}$, $\overline{DATA.OUT.B}$	11, 12	O	7234 CPG	Output data from the FIFO to the MBM generate circuitry. Used to write data into the bubble device (active low).
DET.A+, DET.A−, DET.B+, DET.B−	6, 7, 8, 9	I	7114 MBM	Differential signal lines from the MBM detector.
DIO	17	I/O	7224 BMC	The Serial Bus data line (a bidirectional active high signal).
$\overline{ENABLE.A}$, $\overline{ENABLE.B}$	13, 14	O	7234 CPG/7250	TTL-level outputs utilized as chip selects for other interface circuits. They shall be set and reset by the Command Decoder under instruction of the Controller (active low).
$\overline{ERR.FLG}$	4	O	7224 BMC	An error flag used to interrupt the Controller to indicate that an error condition exists. It shall be an open drain, active low signal.
$\overline{RESET}$	16	I	Power Fail Circuit	An active low signal that shall reset all flags and pointers in the FSA as well as disabling the chip as the CS signal does. The $\overline{RESET}$ pulse width must be 5 clock periods to assure the FSA is properly reset.
$\overline{SELECT.IN}$	19	I	7224 BMC	An input utilized for time-division multiplexing. An active low signal whose presence indicates that the FSA is to send or receive data from the Serial Bus during the next two clock periods.
$\overline{SELECT.OUT}$	2	O	7244 FSA	The $\overline{SELECT.IN}$ pulse delayed by two clocks. It shall be connected to the $\overline{SELECT.IN}$ pin of the next FSA. It is delayed by two clocks because the FSA is a dual-channel device. Channel A shall internally pass $\overline{SELECT.IN}$ to Channel B (delayed by one clock).
$\overline{SHIFT.CLK}$	15	I	7224 BMC	A Controller-generated clock signal that shall be used to clock data out of the bubble I/O Output Latch to the bubble module during a write operation and to cause bubble signals to be converted by the Sense Amp and clocked into the Bubble I/O Input Latch on a read.

7250

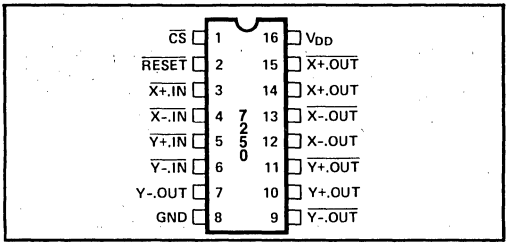


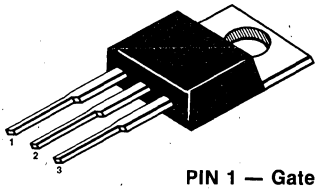
Figure 8. 7250 Pin Configuration

Table 4. 7250 Pin Description

Symbol	Pin No.	I/O	Source/Destination	Description
$\overline{CS}$	1	I	7244 FSA	Chip select. It is active low. When high chip is deselected and I_{DD} is significantly reduced.
$\overline{RESET}$	2	I	Power Fail Circuit	Active low input from RESET.OUT of MD7220-5 Controller forces 7250 outputs inactive so that bubble memory is protected in the event of power supply failure.
$\overline{X+.IN}$, $\overline{X-.IN}$	3, 4	I	7224 BMC	Active low inputs from Controller which turn on the high-current X outputs.
X−.OUT $\overline{X-.OUT}$ X+.OUT $\overline{X+.OUT}$	12, 13, 14, 15	O	7264	High-current outputs and their complements for driving the gates of the 7264 transistors which in turn drive the X coils of the bubble memory.
$\overline{Y+.IN}$, $\overline{Y-.IN}$	5, 6	I	7224 BMC	Active low inputs from Controller which turn on the high-current Y outputs.
Y−OUT $\overline{Y-.OUT}$ $\overline{Y+.OUT}$ Y+.OUT	7,9,10,11	O	7264	High-current outputs and their complements for driving the gates of the 7264 quad transistors which in turn drive the Y coils of the bubble memory.

7264

Four matched pair of N- and P-channel transistors. In industry standard TO-220 Discrete packaging.



PIN 1 — Gate
PIN 2 & TAB — Drain
PIN 3 — Source

Symbol	Pin No.	I/O	Source/Destination	Description
N-Channel				
G	1	I	7250	Gate Drive Signal
D	2	O	7114	Coil Drive Current
S	3	I	Ground	—
P-Channel				
G	1	I	7250	Gate Drive Signal
D	2	O	7114	Coil Drive Current
S	3	I	Ground	—

ABSOLUTE MAXIMUM RATINGS*

7114

Operating Temperature ..... 10°C to 55°C Case
Relative Humidity 95%
Shelf Storage Temperature (Data Integrity Not Guaranteed) ..... −55°C to +125°C
Voltage Applied to DET.SUPPLY ..... 14 Volts
Voltage Applied to PULSE. COM ..... 14 Volts
Continuous Current between DET.COM and Detector Outputs ..... 20 mA
Coil Current ..... 0.5A D.C.
External Magnetic Field for Non-Volatile Storage ..... 20 Oersteds
Non-Operating Handling Shock (without socket) 200G
Operating Vibration (2 Hz to 2 kHz with socket) 20G

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

SUPPORT I.C.'S

	7234	7244	7250	7264
Temperature Under Bias	−40 to 100°C	−10 to +85°C	−40 to 100°C	−40 to 100°C
Storage Temperature	−65 to +150°C	−65 to +150°C	−65 to +150°C	−55 to +150°C
Voltage Input	−0.5 to +7V		−0.5 to V_{DD} +0.5	
V_{CC}	−0.5 to +7V	−0.5 to +7V		
V_{DD}	−0.5 to +12.6V	−0.5 to +14V		
Gate Voltage				15V
Output Current			250mA	
Power Dissipation 80°C	1W			1.05W
Power Dissipation 25°C		1W		2W
Continuous Drain Current				2A
Peak Drain Current				3A

D.C. CHARACTERISTICS

The BPK 5V74 is designed as a true subsystem. All D.C. characteristics that describes the interfacing to the subsystem is included in this section.

7234 (T_A 0°C to +70°C; V_{CC} = 5.0V ±5%, ±5% V_{DD} = 12V ±5%; unless otherwise specified.)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
I_{IL}	Input Low Current			−0.4	mA	V_{IL} = 0.4V, V_{CC} = 5.25V
I_{IH}	Input High Current			20	μA	V_{IH} = V_{CC} = 5.25V
V_{IL}	Input Low Voltage			0.8	V	
V_{IH}	Input High Voltage	2.0			V	
V_C	Input Clamp Voltage			−1.5	V	I = −18 mA, V_{CC} = 4.75V
I_{CEX1}	Output Leakage Current (All Outputs except PWR.FAIL)			1.0	mA	V_{CC} = 5.25V, V_{DD} = 12.6V
I_{CEX2}	$\overline{\text{PWR.FAIL}}$ Output Leakage Current			40	μA	V_{OH} = V_{CC} = 5.25V
V_{OL}	$\overline{\text{PWR.FAIL}}$ Output Low Voltage			0.4	V	I_{OL} = 4 mA, V_{CC} = 4.75V
I_{CC1}	Current from V_{CC}—Selected		30	45	mA	CS = V_{IL}, V_{CC} = 5.25V
I_{DD1}	Current from V_{DD}—Selected		20	35	mA	CS = V_{IL}, V_{CC} = 5.25V
I_{DD2}	Current from V_{DD}—Power Down		12	19	mA	CS = V_{IH}, V_{DD} = 12.6V

7244 (T_A = 0°C to 70°C; V_{CC} = 5.0V +5%, −10%: V_{DD} = 12V ±5%)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
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 (All Outputs Except $\overline{\text{SELECT.OUT}}$)			0.45	V	I_{OL} = 3.2mA
V_{OLSO}	Output Low Voltage ($\overline{\text{SELECT.OUT}}$)			0.45	V	I_{OL} = 1.6mA
V_{OH}	Output High Voltage (All Outputs Except $\overline{\text{SELECT.OUT}}$)	2.4			V	I_{OH} = 400 μA
V_{OHSO}	Output High Voltage ($\overline{\text{SELECT.OUT}}$)	2.4			V	I_{OH} = 200 μA
V_{THR}	Detector Threshold		6.8		mV	V_{DD} = 12.0V
$\lvert I_{IN} \rvert$	Input Leakage Current			10	μA	$0 \leq V_{IN} \leq V_{CC}$
$\lvert I_{OFL} \rvert$	Output Float Leakage			10	μA	$0.45 \leq V_{OUT} \leq V_{CC}$
I_{CC}	Power Supply Current from V_{CC}			120	mA	
I_{DD}	Power Supply Current from V_{DD}			30	mA	

7250 (T_A = 0° to 70°C; V_{DD} = 12V +5%, −10%; unless otherwise specified)

Symbol	Parameter	Limits				Test Conditions
		Min.	Typ.	Max.	Unit	
$\|I_{IN}\|$	Input Current			5	μA	V_I = 0.8V
V_{IL}	Low-Level Input Voltage			0.8	V	
V_{IH}	High-Level Input Voltage	2.2			V	
I_{DD0}	Supply Current			4.5	mA	Chip Deselected: $\overline{CS}$ = V_{IH}, V_{DD} = 12.6V
I_{DD1}	Supply Current			75	mA	f = 100 kHz, V_{DD} = 12.6V, Outputs Unloaded

ADVANCE INFORMATION

BPK 5V75
4MBIT BUBBLE MEMORY PROTOTYPE KIT

BPK 5V75-4	10°C To 55°C

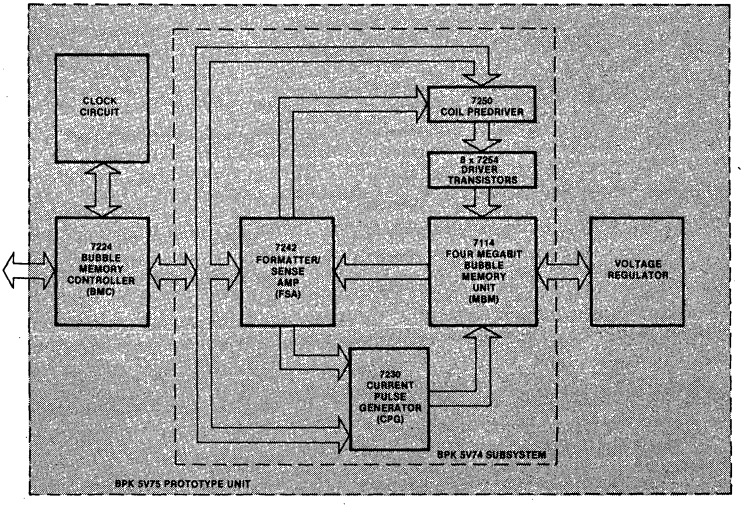
- **Assembled and Tested 4MBit Bubble Memory Prototype Kit on 6" x 4" PC Board**
- **Complete with Powerfail Data Protection and Clock Circuitry**
- **Built-in Error Detection/Correction**
- **Interfaces with Intel 8080/85/86/88 186/286 and Other Standard Microprocessors**
- **Software Driver for Bubble Memory Kit on Diskette for Intel Microcomputer Development System**
- **4Mbit, Non-Volatile, Read-Write, Solid-State Memory in Leaded Dense Package**
- **Average Random Access Time of 88ms**
- **Maximum Data Rate of 200K bit/sec**
- **Operates From + 5V Only**
- **Complete Documentation and Interfacing Information Included**

The BPK 5V75 prototype kit is a completely assembled and tested 4Mbit bubble memory evaluation tool. It is ideal for the design engineer that wants the opportunity to quickly evaluate how a bubble memory solution improves and adds value to an end-product by providing a compact solid-state memory that also reliably keeps the data at any power down.

Application information on microprocessor interfacing is included in the kit. A Bubble Memory Kit software driver is also included on a diskette for the Intel Microcomputer Development System.

The bubble memory (7114) and the support circuits (7234, 7244, 7250, 7264) in the kit are described in more detail on the BPK 5V74 Bubble Memory Subsystem data sheet. The Bubble Memory Controller (7224) in the kit is described in more detail on the 7224 data sheet.

For production purposes, the bubble memory and the support circuit components can be ordered as kits in the BPK 5V74 Subsystem. The 7224, controlling up to eight BPK 5V74's, is ordered separately.



OCTOBER 1983
ORDER NUMBER: 230868-001

COMPONENT TEMPERATURE SPECIFICATIONS*

Part Number	7110A Magnetic Bubble Memory Temperature		Support Circuits Min. Operating Temperature	Description
	Operating	Non-Volatile Storage		
BPK 5V75-4	10° to 55°C Case	−20° to 75°C	10° to 55°C Ambient	4 Mbit Bubble Memory Prototype Kit

* The bubble memory prototype kit is assembled and functionally tested to facilitate the prototyping process. The board is tested at 10°C and 40°C ambient temperature.

BPK 5V75 BUBBLE MEMORY PROTOTYPE BOARD

Item	Description	Part Number
4 Mbit Bubble Memory	20-pin leaded package which provides 4 megabit of non-volatile storage.	7114-4
Bubble Memory Controller	User interface, performs serial-to-parallel and parallel-to-serial data conversions. Generates timing signals.	7224
Current Pulse Generator	Converts digital timing signals to analog current pulses suited to the drive requirements of the 7114 MBM. The CPG provides the replicate, swap, generate, boot replicate, and bootswap pulses required by the MBM.	7234
Dual Formatter/Sense Amp	Provides direct interface to the 7114 Bubble Memory. The FSA contains on-chip sense amplifiers, a full FIFO data block buffer, burst error detection and correction circuits, and circuitry for handling of the bubble memory redundant loops.	7244
Coil Predriver	Provides the high voltage, high current outputs to drive the 7264 transistors.	7250
8 Coil Drive Transistors	Switches the required current to drive the X and Y coils of the 7114 Bubble Memory.	7264
Prefabricated Printed Circuit Board		IMB 75
Clock Circuit		
Voltage Regulator		

ADDITIONAL ITEMS

Diskette	Software driver for bubble memory kit, configured for Intel Microcomputer Development System.	
BPK 75 Bubble Memory Prototype Kit User's Manual	Literature	

SPECIFICATIONS

Capacity

512K Byte per BPK 5V75

Performance

Avg. Access Time ..... 88 msec
Maximum Data Transfer Rate .... 200 Kbits/sec
Average Data Transfer Rate ..... 136 Kbits/sec

Data Organization

64 bytes per page
8192 pages per BPK 5V75

Addressing Scheme

Logical page number

Environmental

Temperature: Temperature specifications.
Operating Humidity: 0—95% Non-Condensing

BPK 5V75 POWER SUPPLY REQUIREMENTS

Voltage	Margin	Power Off/Power Fail Decay Rate
+5 Volt	±5%	less than 0.45 volts/msec

- Power on voltage rate of rise — no restrictions

BPK 5V75 POWER CONSUMPTION

Board including bubble memory, support ICs, controller and voltage regulator.

Total Standby
- Typical 1.4 W
- Maximum 3.2 W

Total Active
- Typical 5.5 W
- Maximum 9.6 W

PRELIMINARY

BPK 70
1MBIT BUBBLE MEMORY SUBSYSTEM

BPK 70-1	0°C to 75°C
BPK 70-4	10°C to 55°C
BPK 70-5	−20°C to 85°C

- **1 Mbit (128K Bytes) Non-volatile, Solid-state, Read/Write Bubble Memory Subsystem**
- **Interfaces to Host Microprocessor Via 7220 Bubble Memory Controller**
- **Maximum Data Rate of 100K bit/sec with One Subsystem**
- **Maximum Data Rate of 800K bit/sec with Eight Subsystems in Parallel and Time Multiplexed**
- **Contains Bubble Memory and all ICs for Production with 1Mbit Bubble Memory**
- **Modularity Provides Expansion Up to Eight Subsystems per Controller**
- **Average Random Access Time of 48 ms**
- **Bubble Memory in 20-pin Leadless Package Using Socket**

The BPK 70 Bubble Memory Subsystem is a modular building block used to design bubble memory systems. In a complete bubble memory system, a 7220 (7220-4 or 7220-5) Bubble Memory Controller (BMC) interfaces the BPK 70 subsystem to the host processor.

The modular Intel subsystem provides a path for density expansion. One 7220/7220-4 BMC can interface up to eight 1 Mbit BPK 70-1/BPK 70-4 subsystems. Thus, a 1 MBit (128 KByte) system can be expanded up to a 8 MBit system by adding subsystems. A 7220-5 can interface up to four BPK 70-5 subsystems. BMC's can be combined in parallel to further expand the system memory capacity.

Together, the BPK 70 Bubble Storage Subsystem and a 7220 controller provide a reliable mass storage system for any application. This bubble memory system can be customized to the particular layout and form factor of many different systems.

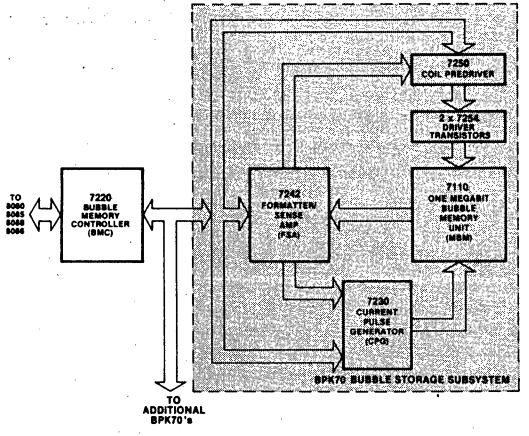


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

OCTOBER 1983
ORDER NO. 230870-001

FUNCTIONAL DESCRIPTION

A BPK 70 subsystem and a controller comprise a complete bubble memory system. The 1 MBit BMC, the 7220, provides the interface between the host microprocessor and the bubble memory subsystem and provides all the timing and control signals to the subsystem. The user interface of the BMC is compatible with microprocessor bus systems for 8080, 8085, 8086, 8088, 80186, 80286 and other standard microprocessors. The BMC is a software driven device utilizing 16 convenient commands. The design engineer's primary responsibility is interfacing to the BMC. This is comparable to interfacing a disk drive controller.

The BPK 70 consists of one 1 MBit Magnetic Bubble Memory (MBM) and five support IC's (see Figure 1). These are the basic components to build a non-volatile, solid-state, read/write military memory system utilizing 1 MBit bubble memory. The bubble memory is in a leadless package and can directly be soldered to tbe board. The complete family of LSI support circuits has been designed to handle the complex analog interface associated with bubble devices. The immediate support circuitry for the MBM consists of five integrated circuit components — an 7250 Coil Predriver (CPD), two 7254 Quad VMOS Drive Transistor Packs, an 7230 Current Pulse Generator (CPG), and an 7242 Formatter/Sense Amplifier (FSA).

Data integrity is insured by the automatic error correction designed into the BPK 70-5.

The average random access time of a 1 MBit subsystem is 48 ms with a 100Kbit/sec maximum data transfer rate. Operating several subsystems in parallel, the BMC uses time division multiplexing. Therefore, the maximum data rate increases correspondingly for the whole system.

Operating subsystems serially, one MBM being accessed at a time, the maximum data transfer rate is still 100Kbit/sec. If low power consumption is a critical design goal, the bubble memory subsystem can be powered down when it is not being accessed, thus reducing the average power consumption.

The data in the 1 Mbit subsystem is organized in 2048 pages, each with 64 bytes. Conceptually, the data organization with pages is analogous to a disk system's sectors. In systems with multiple bubble memories, the page size can vary from 64 bytes to 512 bytes (256 for BPK 70-5) depending on the number of subsystems and if the subsystems are operating in parallel or serially, being accessed one at a time. In the 10° to 55°C and 0 to 75°C temperature range, respectively, the components in the subsystem are fully interchangeable.

Each of the components in the subsystem is described in more detail in the rest of this data sheet. For additional information, consult the Intel **BPK-72 Bubble Memory Prototype Kit User's Manual,** Order Number: 121685.

BPK 70 FUNCTIONAL DESCRIPTION

Item	Description	Part Number
1 MBit Bubble Memory	20-pin leadless package which provides 1 megabit of non-volatile storage.	7110
Socket	Provides reliable mounting of 1Mbit bubble memory.	7905
Current Pulse Generator	Converts digital timing signals to analog current pulses suited to the drive requirements of the MBM. The CPG provides the replicate, swap, generate, boot replicate, and bootswap pulses required by the MBM. 22 Pin DIP Package.	7230
Dual Formatter/Sense Amp	Provides direct interface to the Bubble Memory. The FSA contains on-chip sense amplifiers, a full FIFO data block buffer, burst error detection and correction circuits, and circuitry for handling of the bubble memory redundant loops. 20 Pin DIP package.	7242
Coil Predriver	Provides the high voltage, high current outputs to drive the Quad VMOS transistors. 16 Pin DIP package.	7250
Quad VMOS Coil Drive Transistors	Switches the required current to drive the X and Y coils of the Bubble Memory. 14 Pin DIP Package.	7254

For additional packaging information, see the last section of Packaging Information in the Memory Components Handbook.

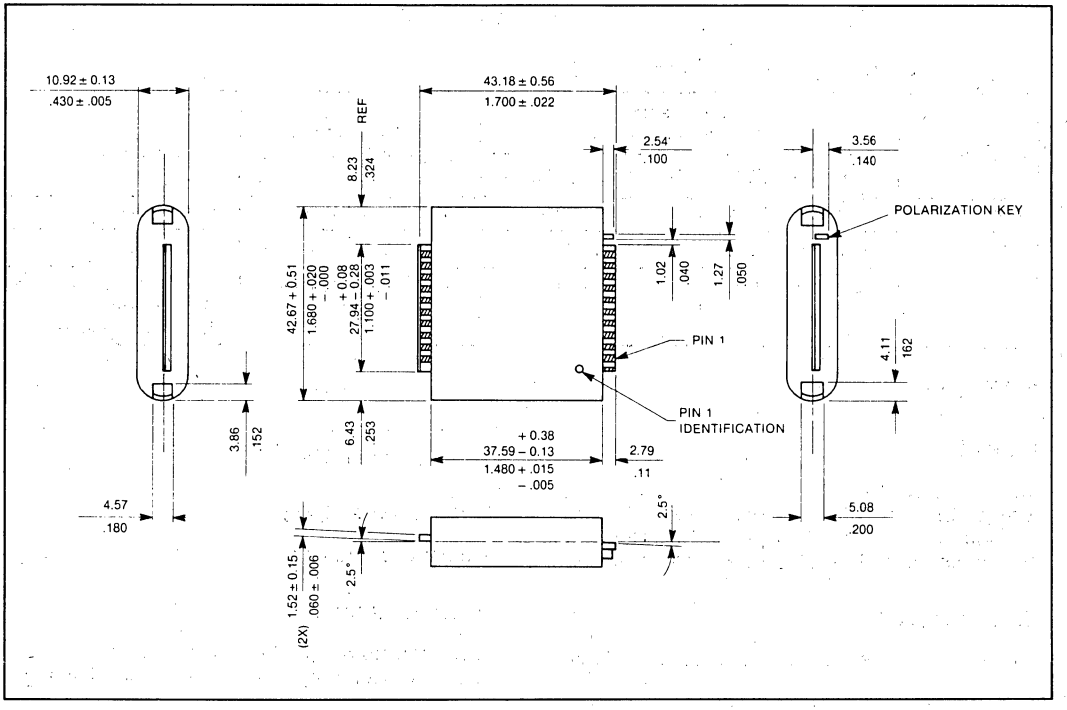


Figure 2. Bubble Leadless Package

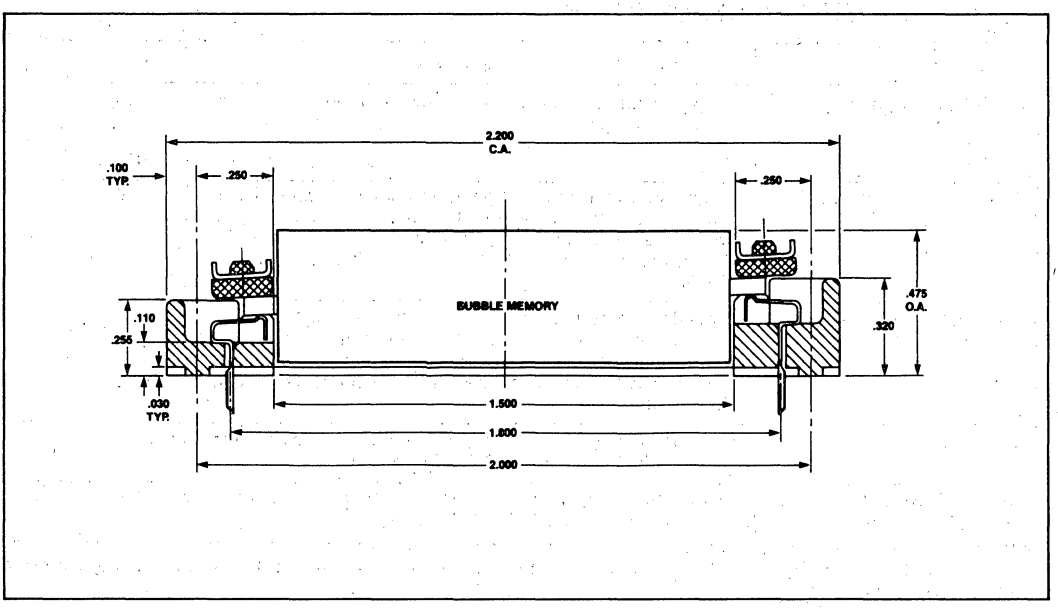


Figure 3. Socket 7905

BPK 70 TEMPERATURE RANGE

Part Number	Bubble Memory Temperature Ranges		Support Circuits Operating Temperature (T_A)	Description
	Operating (T_C)	Non-Volatile Storage		
BPK 70-1	0 to 75°C Case	−40 to 90°C	0 to 70°C Ambient	See Below
BPK 70-4	10 to 55°C Case	−20 to 75°C	10 to 55°C Ambient	See Below
BPK 70-5	−20 to 85°C Case	−55 to +100°C	−20 to 85°C Ambient	1 Mbit Bubble Storage Sub-system, Leadless Package

SPECIFICATIONS

Capacity

128K Byte per BPK 70
Maximum of 8 BPK 70-1 per 7220 Controller
Maximum of 8 BPK 70-4 per 7220-4 Controller
Maximum of 4 BPK 70-5 per 7220-5 Controller

Performance

Avg. Access Time..... 48 msec

Data Organization

64 bytes per page
2048 pages per BPK 70

Addressing Scheme

Logical page number

Environmental

Temperature: See Ordering Information
Operating Humidity: 0—95% Non-Condensing

BPK 70 POWER CONSUMPTION

BPK 70 Components	Power (Watts)					
	+5V (Maximum)	+12V (Maximum)	Total Active (Maximum)	Total Active (Typical)	Total Standby (Maximum)	Total Standby (Typical)
7110	0	1.740	1.740	1.480	0.440	0.290
7230	0.235	0.440	0.675	0.390	0.475	0.225
7242	0.630	0.375	1.005	0.500	1.005	0.500
7250	0	0.945	0.945	0.480	0.060	0.030
7254	0	1.300	1.300	0.550	0	0
7220	1.050	0	1.050	0.500	1.050	0.500
System	1.92	4.80	6.72	3.90	3.03	1.55

For larger systems in which a 7220 is controlling several BPK 70 subsystems, the power consumption can be estimated by using the above data. By using power switching techniques the bubble memory system's standby power can be reduced to almost zero. For details, see AP-164, **Using CMOS to Minimize Bubble Memory Power Consumption.**

BPK 70 POWER SUPPLY REQUIREMENTS

Voltage	Margin	Power Off/Power Fail Decay Rate
+12 Volt	±5%	less than 1.10 volts/msec
+5 Volt	±5%	less than 0.45 volts/msec

- Voltage sequencing — no restrictions
- Power on voltage rate of rise — no restrictions
- The power supply requirements shown are based on the recommended power fail circuitry as shown in Figure 4.

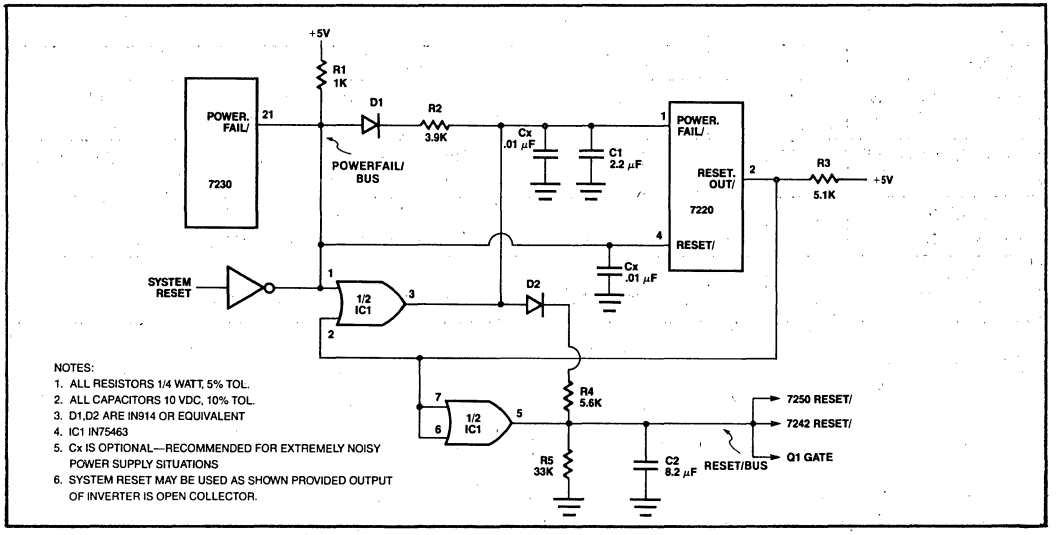


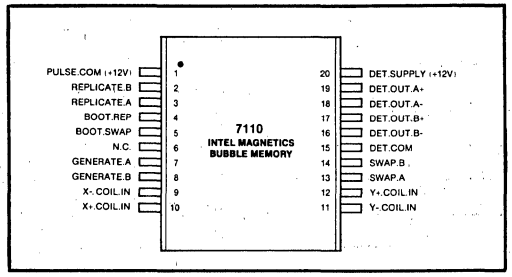
Figure 4. Recommended Power Fail Circuitry

Below the 7110 MBM is described in more detail. For other support ICs are described in the BPK70A data sheet.

Consult the **BPK72 User's Manual** for Additional Information.

PIN DESCRIPTION

7110



NOTE THAT PINS 13 AND 14 SHOULD BE EXTERNALLY CONNECTED.

Figure 5. 7110 Pin Configuration

Table 1. 7110 Pin Description

Symbol	Pin No.	I/O	Source/Destination	Description
BOOT.REP	4	I	7230 CPG	Two-level current pulse input for reading the boot loop.
BOOT.SWAP	5	I	7230 CPG	Single-level current pulse for writing data into the boot loop. This pin is normally used only in the manufacture of the MBM.
DET.COM	15	I		Ground return for the detector bridge.
DET.OUT	16 — 19	O	7242 FSA	Differential pair (A+, A− and B+, B−) outputs which have signals of several millivolts peak amplitude.
DET.SUPPLY	20	I		+ 12 volt supply pin.
GEN.A and GEN.B	7, 8	I	7230 CPG	Two-level current pulses for writing data onto the input track.
PULSE.COM	1	I		+ 12 volt supply pin.
REP.A and REP.B	3, 2	I	7230 CPG	Two-level current pulses for replicating data from storage loops to output track.
SWAP.A and SWAP.B	13, 14	I	7230 CPG	Single-level current pulse for swapping data from input track to storage loops.
X− .COIL.IN. X+ COIL.IN.	9, 10	I	7254/VMOS	Terminals for the X or inner coil.
Y− .COIL.IN. Y+ .COIL.IN.	11, 12	I	7254/VMOS	Terminals for the Y or outer coil.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature ..... −20°C to +85°C Case
Relative Humidity 95%
Shelf Storage Temperature (Data Integrity Not Guaranteed) ..... −65°C to +150°C
Voltage Applied to DET.SUPPLY ..... 14 Volts
Voltage Applied to PULSE.COM ..... 12.6 Volts
Continuous Current between DET.COM and Detector Outputs ..... 10 mA
Coil Current ..... 0.5A D.C.
External Magnetic Field for Non-Volatile Storage ..... 20 Oersteds
Non-Operating Handling Shock (without socket) 200G
Operating Vibration (2 Hz to 2 kHz with socket) 20G

COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS (T_C = Specified in earlier temperature range table, V_{DD} = 12V ± 5%)

7110

Parameter	7110-1, -4 Limits			7110-5 Limits		Unit
	Min.	Nom.[1]	Max.	Min.	Max.	
RESISTANCE: PULSE.COM to GEN.A or GEN.B	9	30	59	8	61.5	ohms
RESISTANCE: PULSE.COM to REP.A or REP.B	9	20	26	8	27	ohms
RESISTANCE: PULSE.COM to SWAP.A or SWAP.B	44	100	149	40	155.5	ohms
RESISTANCE: PULSE.COM to BOOT.REP	3.5	8	24	3	25	ohms
RESISTANCE: PULSE.COM to BOOT.SWAP	5	15	36	4.5	37.5	ohms
RESISTANCE: DET.OUT A+ to DET.OUT.A−	670	1030	1903	620	1984	ohms
RESISTANCE: DET.OUT B+ to DET.OUT B−	670	1030	1903	620	1984	ohms
RESISTANCE: DET.COM to DET.SUPPLY	355	600	1050	338	1095	ohms
X.COIL RESISTANCE		4.6				ohms
Y.COIL RESISTANCE		2.0				ohms
X.COIL INDUCTANCE		97				μH
Y.COIL INDUCTANCE		80				μH
OPERATING POWER		1.20	1.75			watts
STANDBY POWER		0.25	.45			watts

DRIVE REQUIREMENTS CHARACTERISTICS[2] (T_C = Specified in temperature range table)

7110

Symbol	Parameter	Min.	Nom.[1]	Max.	Units
f_R	Field Rotation Frequency	49.95	50.000	50.05	kHz
I_{px}	X.Coil Peak Current		600		ma
I_{py}	Y.Coil Peak Current		750		ma
Θ_{1x}	X.Coil Positive Turn-On Phase	268	270	272	degrees
Θ_{2x}	X.Coil Positive Turn-Off Phase	16	18	20	degrees
Θ_{3x}	X.Coil Negative Turn-On Phase	88	90	92	degrees
Θ_{4x}	X.Coil Negative Turn-Off Phase	196	198	200	
Θ_{1y}	Y.Coil Positive Turn-On Phase	0	0	0	degrees
Θ_{2y}	Y.Coil Positive Turn-Off Phase	106	108	110	degrees
Θ_{3y}	Y.Coil Negative Turn-On Phase	178	180	182	degrees
Θ_{4y}	Y.Coil Negative Turn-Off Phase	286	288	290	degrees

CONTROL PULSE REQUIREMENTS (T_C = Specified in temperature range table)[2]

7110

Pulse	Amplitude			Pulse of Leading Edge (Degrees)			Width (Degrees)		
	Min.	Nom[1]	Max.	Min.	Nom.[1]	Max.	Min.	Nom.[1]	Max.
GEN.A, GEN.B CUT	62	75	81	266 86	270 (Odd) 90 (Even)	274 94	3	6.75	8
GEN.A, GEN.B TRANSFER	34	40	49	266 86	270 (Odd) 90 (Even)	274 94	86	90	94
REP.A, REP.B CUT	170	200	240	268	270	277	3	6.75	8
REP.A, REP.B TRANSFER	126	145	160	268	270	277	86	90	94
SWAP	111	125	134	176	180	184	513	517	521
BOOT.REP CUT	85	100	110	268	270	277	3	6.75	8
BOOT.REP TRANSFER	63	75	80	268	270	277	86	90	94
BOOT.SWAP	63	75	80	176	180	184		360	

NOTES:

1. Nominal values are measured at T_C = 25°C.
2. 7110-5 is sold only as a matched part with the 7230-5. Matched parts are tested over temperature range for V_{DD} = 12V ±5%.

PRELIMINARY

BPK 70A
1MBIT BUBBLE MEMORY SUBSYSTEM

BPK 70A-1	0°C to 75°C
BPK 70A-4	10°C to 55°C
BPK 70A-5	– 20°C to 85°C

- **1 Mbit (128K Bytes) Non-volatile, Solid-state, Read/Write Bubble Memory Subsystem**
- **Interfaces to Host Microprocessor Via 7220 Bubble Memory Controller**
- **Maximum Data Rate of 100K bit/sec with One Subsystem**
- **Maximum Data Rate of 800K bit/sec with Eight Subsystems in Parallel and Time Multiplexed**
- **Contains Bubble Memory and all ICs for Production with 1Mbit Bubble Memory**
- **Modularity Provides Expansion Up to Eight Subsystems per Controller**
- **Average Random Access Time of 48 ms**
- **Bubble Memory in 20-pin Leaded Package**

The BPK 70A Bubble Memory Subsystem is a modular building block used to design bubble memory systems. In a complete bubble memory system, a 7220 (7220-4 or 7220-5) Bubble Memory Controller (BMC) interfaces the BPK 70A subsystem to the host processor.

The modular Intel subsystem provides a path for density expansion. One 7220/7220-4 BMC can interface up to eight 1 Mbit BPK 70A-1/BPK 70A-4 subsystems. Thus, a 1 MBit (128 KByte) system can be expanded up to a 8 MBit system by adding subsystems. A 7220-5 can interface up to four BPK 70A-5 subsystems. BMC's can be combined in parallel to further expand the system memory capacity.

Together, the BPK 70A Bubble Storage Subsystem and a 7220 controller provide a reliable mass storage system for any application. This bubble memory system can be customized to the particular layout and form factor of many different systems.

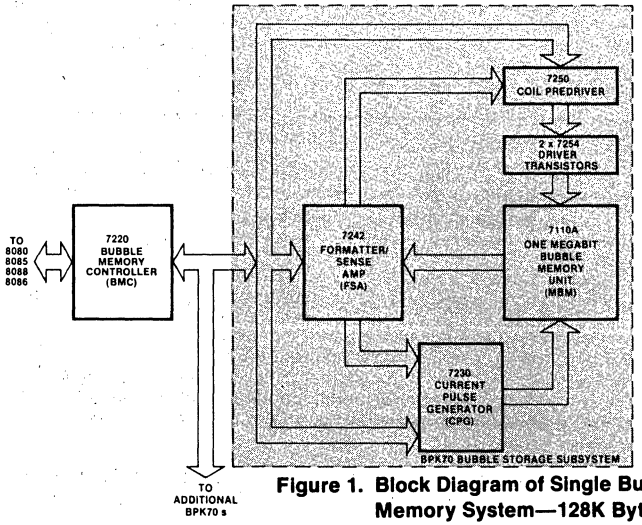


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

OCTOBER 1983
ORDER NO. 210804-002

FUNCTIONAL DESCRIPTION

An BPK 70A-5 subsystem and a controller comprise a complete bubble memory system. The 1 MBit BMC, the 7220, provides the interface between the host microprocessor and the bubble memory subsystem and provides all the timing and control signals to the subsystem. The user interface of the BMC is compatible with microprocessor bus systems for 8080, 8085, 8086, 8088, 80186, 80286 and other standard microprocessors. The BMC is a software driven device utilizing 16 convenient commands. The design engineer's primary responsibility is interfacing to the BMC. This is comparable to interfacing a disk drive controller.

The BPK 70A consists of one 1 MBit Magnetic Bubble Memory (MBM) and five support IC's (see Figure 1). These are the basic components to build a non-volatile, solid-state, read/write military memory system utilizing 1 MBit bubble memory. The bubble memory is in a leaded package. The complete family of LSI support circuits has been designed to handle the complex analog interface associated with bubble devices. The immediate support circuitry for the MBM consists of five integrated circuit components — an 7250 Coil Predriver (CPD), two 7254 Quad VMOS Drive Transistor Packs, an 7230 Current Pulse Generator (CPG), and an 7242 Formatter/Sense Amplifier (FSA).

Data integrity is insured by the automatic error correction designed into the BPK 70A.

The average random access time of a 1 MBit subsystem is 48 ms with a 100Kbit/sec maximum data transfer rate. Operating several subsystems in parallel, the BMC uses time division multiplexing. Therefore, the maximum data rate increases correspondingly for the whole system.

Operating subsystems serially, one MBM being accessed at a time, the maximum data transfer rate is still 100Kbit/sec. If low power consumption is a critical design goal, the bubble memory subsystem can be powered down when it is not being accessed, thus reducing the average power consumption.

The data in the 1 Mbit subsystem is organized in 2048 pages, each with 64 bytes. Conceptually, the data organization with pages is analogous to a disk system's sectors. In systems with multiple bubble memories, the page size can vary from 64 bytes to 512 bytes (256 for BPK 70-5) depending on the number of subsystems and if the subsystems are operating in parallel or serially, being accessed one at a time. In the 10° to 55°C and 0 to 75°C temperature range, respectively, the components in the subsystem are fully interchangeable.

Each of the components in the subsystem is described in more detail in the rest of this data sheet. For additional information, consult the Intel **BPK-72 Bubble Memory Prototype Kit User's Manual,** Order Number: 121685.

BPK 70A FUNCTIONAL DESCRIPTION

Item	Description	Part Number
1 MBit Bubble Memory	20-pin leaded package which provides 1 megabit of non-volatile storage.	7110A
Current Pulse Generator	Converts digital timing signals to analog current pulses suited to the drive requirements of the MBM. The CPG provides the replicate, swap, generate, boot replicate, and bootswap pulses required by the MBM. 22 Pin DIP Package.	7230
Dual Formatter/Sense Amp	Provides direct interface to the Bubble Memory. The FSA contains on-chip sense amplifiers, a full FIFO data block buffer, burst error detection and correction circuits, and circuitry for handling of the bubble memory redundant loops. 20 Pin DIP package.	7242
Coil Predriver	Provides the high voltage, high current outputs to drive the Quad VMOS transistors. 16 Pin DIP package.	7250
Quad VMOS Coil Drive Transistors	Switches the required current to drive the X and Y coils of the Bubble Memory. 14 Pin DIP package.	7254

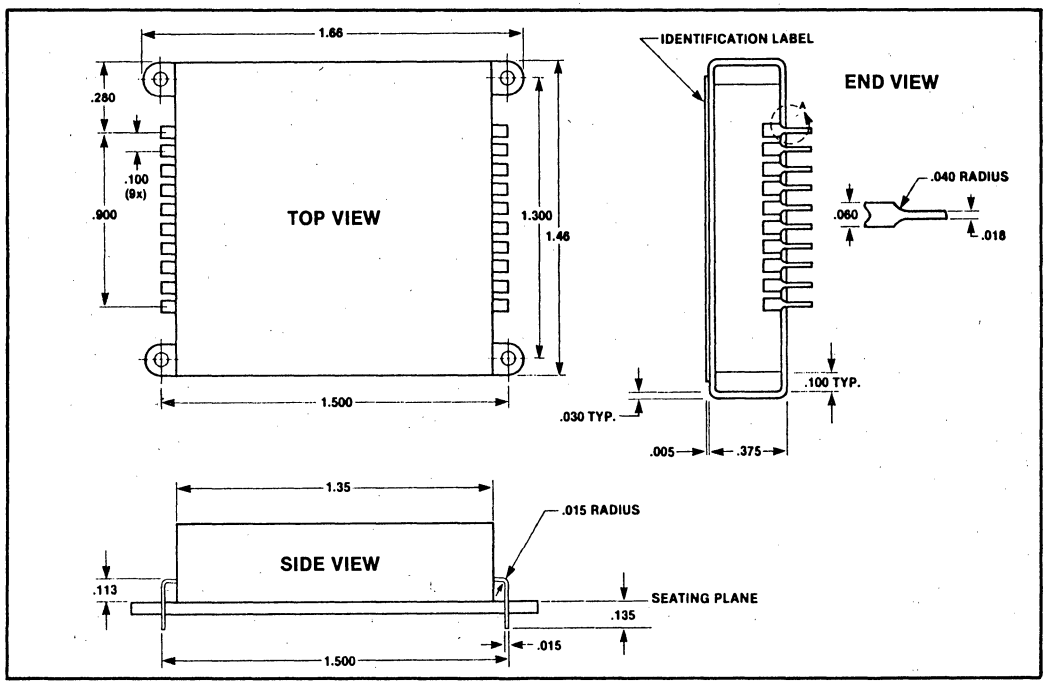


Figure 2. 1MBit Leaded Bubble Memory Package

BPK 70A TEMPERATURE RANGE

Part Number	Bubble Memory Temperature Ranges		Support Circuits Operating Temperature (T_A)	Description
	Operating (T_C)	Non-Volatile Storage		
BPK 70A-1	0 to 75°C Case	−40 to 90°C	0 to 70°C Ambient	See Below
BPK 70A-4	10 to 55°C Case	−20 to 75°C	10 to 55°C Ambient	See Below
BPK 70A-5	−20 to 85°C Case	−55 to +100°C	−20 to 85°C Ambient	1 Mbit Bubble Storage Sub-system, Leaded Package

SPECIFICATIONS

Capacity

128K Byte per BPK 70A
Maximum of 8 BPK 70A-1 per 7220 Controller
Maximum of 8 BPK 70A-4 per 7220-4 Controller
Maximum of 4 BPK 70A-5 per 7220-5 Controller

Performance

Avg. Access Time . . . . . 48 msec

Data Organization

64 bytes per page
2048 pages per BPK 70A

Addressing Scheme

Logical page number

Environmental

Temperature: See Ordering Information
Operating Humidity: 0—95% Non-Condensing

BPK 70A POWER CONSUMPTION

BPK70A Components	Power (Watts)					
	+5V (Maximum)	+12V (Maximum)	Total Active (Maximum)	Total Active (Typical)	Total Standby (Maximum)	Total Standby (Typical)
7110A	0	1.740	1.740	1.480	0.440	0.290
7230	0.235	0.440	0.675	0.390	0.475	0.225
7242	0.630	0.375	1.005	0.500	1.005	0.500
7250	0	0.945	0.945	0.480	0.060	0.030
7254	0	1.300	1.300	0.550	0	0
7220-5	1.050	0	1.050	0.500	1.050	0.500
System	1.92	4.80	6.72	3.90	3.03	1.55

For larger systems in which a 7220 is controlling several BPK 70A subsystems, the power consumption can be estimated by using the above data. By using power switching techniques the bubble memory system's standby power can be reduced to almost zero. For details, see AP-164, **Using CMOS to Minimize Bubble Memory Power Consumption.**

BPK 70A POWER SUPPLY REQUIREMENTS

Voltage	Margin	Power Off/Power Fail Decay Rate
+12 Volt	±5%	less than 1.10 volts/msec
+5 Volt	±5%	less than 0.45 volts/msec

- Voltage sequencing — no restrictions
- Power on voltage rate of rise — no restrictions
- The power supply requirements shown are based on the recommended power fail circuitry as shown in Figure 3.

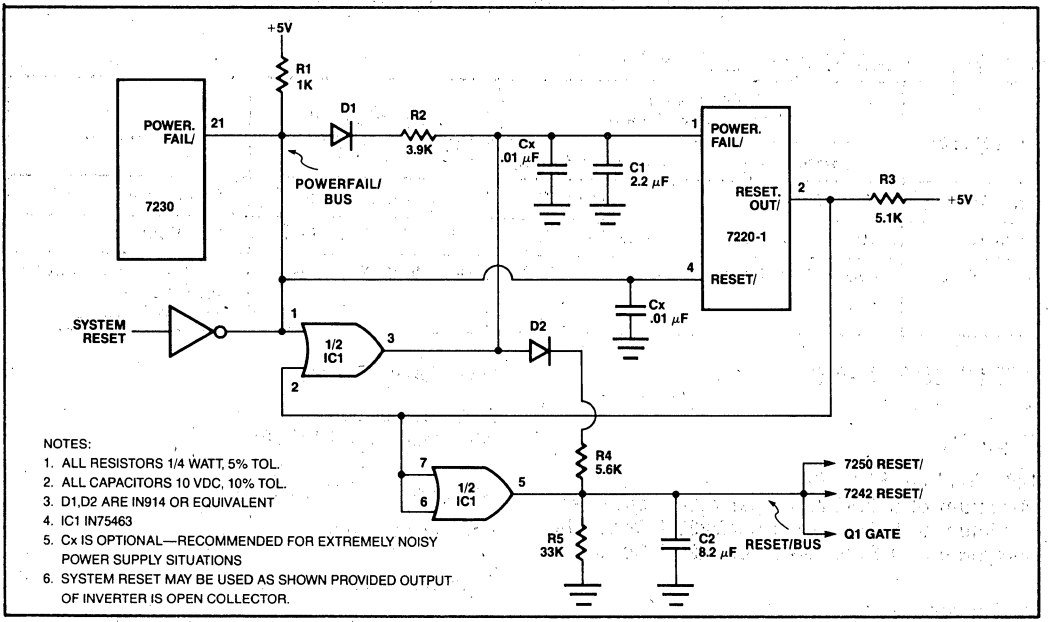


Figure 3. External Powerfail Circuit Solution

PIN DESCRIPTIONS

7110A

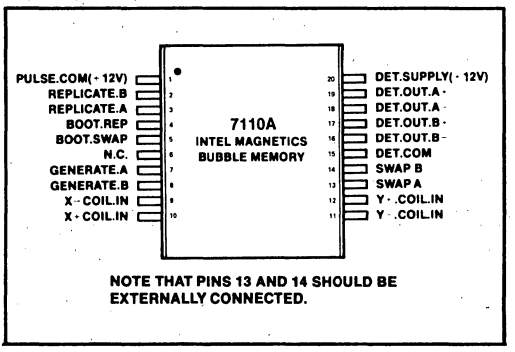


Figure 4. 7110A Pin Configuration

Table 1. 7110A Pin Description

Symbol	Pin No.	I/O	Source/Destination	Description
BOOT.REP	4	I	7230 CPG	Two-level current pulse input for reading the boot loop.
BOOT.SWAP	5	I	7230 CPG	Single-level current pulse for writing data into the boot loop. This pin is normally used only in the manufacture of the MBM.
DET.COM	15	I		Ground return for the detector bridge.
DET.OUT	16 — 19	O	7242 FSA	Differential pair (A+, A− and B+, B−) outputs which have signals of several millivolts peak amplitude.
DET.SUPPLY	20	I		+ 12 volt supply pin.
GEN.A and GEN.B	7, 8	I	7230 CPG	Two-level current pulses for writing data onto the input track.
PULSE.COM	1	I		+ 12 volt supply pin.
REP.A and REP.B	3, 2	I	7230 CPG	Two-level current pulses for replicating data from storage loops to output track.
SWAP.A and SWAP.B	13, 14	I	7230 CPG	Single-level current pulse for swapping data from input track to storage loops.
X− .COIL.IN. X+ COIL.IN.	9, 10	I	7254/VMOS	Terminals for the X or inner coil.
Y− .COIL.IN. Y+ .COIL.IN.	11, 12	I	7254/VMOS	Terminals for the Y or outer coil.

7230

EXTERNAL RESISTOR REQUIREMENTS

Connect a 1% 3.48K ohm resistor based between pin 20 and ground or rreferenced current switch as outlined in BPK72 User's Manual.

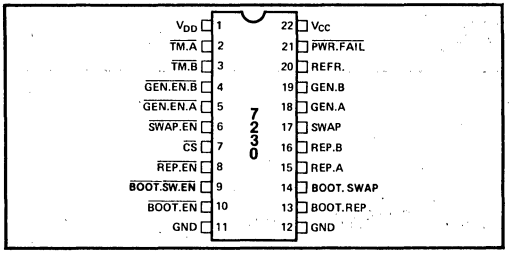


Figure 5. 7230 Pin Configuration

Table 2. 7230 Pin Description

Symbol	Pin No.	I/O	Source/Destination	Description
$\overline{\text{BOOT.EN}}$	10	I	7220 BMC	An active low input enabling the BOOT.REP output current pulse.
BOOT.REP	13	O	7110A MBM	An output providing the current pulse for bootstrap loop replication in the bubble memory.
BOOT.SWAP	14	O	7110A MBM	An output providing a current pulse which may be used for writing data into the bootstrap loop.
$\overline{\text{BOOT.SW.EN}}$	9	I	7220 BMC	An active low input enabling the BOOT.SWAP output current pulse.
CS	7	I	7242 FSA	An active low input for selecting the chip. The chip powers down during deselect.
GEN.A	18	O	7110A MBM	An output providing the current pulse for writing data into the "A" quads of the bubble memory.
GEN.B	19	O	7110A MBM	An output providing the current pulse for writing data into the "B" quads of the bubble memory.
$\overline{\text{GEN.EN.A}}$	5	I	7242 FSA	An active low input enabling the GEN.A output current pulse.
$\overline{\text{GEN.EN.B}}$	4	I	7242 FSA	An active low input enabling the GEN.B output current pulse.
$\overline{\text{PWR.FAIL}}$	21	O	7220 BMC	An active low, open collector output indicating that either V_{CC} or V_{DD} is below its threshold value.
REFR.	20	I	External Resistor	The pin for the reference current generator to which an external resistance must be connected.
REP.A	15	O	7110A MBM	An output providing the current pulse for replication of data in the "A" quads of the bubble memory.
REP.B	16	O	7110A MBM	An output providing the current pulse for replication of data in the "B" quads of the bubble memory.
$\overline{\text{REP.EN}}$	8	I	7220 BMC	An active low input enabling the REP.A and REP.B outputs.
SWAP	17	O	7110A MBM	An output providing the current pulse for exchanging the data between the input track and the storage loops in the bubble memory.
$\overline{\text{SWAP.EN}}$	6	I	7220 BMC	An active low input enabling the SWAP output.
$\overline{\text{TM.A}}$	2	I	7220 BMC	An active low timing signal determining the cut pulse widths of the BOOT.REP, GEN.A, GEN.B, REP.A and REP.B outputs.
$\overline{\text{TM.B}}$	3	I	7220 BMC	An active low timing signal determining the transfer pulse widths of the BOOT.REP, GEN.A, GEN.B, REP.A and REP.B outputs.

7242

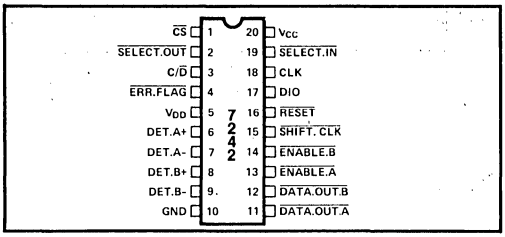


Figure 6. 7242 Pin Configuration

Table 3. 7242 Pin Description

Symbol	Pin No.	I/O	Source/Destination	Description
C/$\overline{D}$	3	I	7220 BMC	Command/$\overline{\text{Data}}$ signal. This signal shall cause the FSA to enter a receive command mode when high and to interpret the serial data line as data when low. Any previously active command will be immediately terminated by C/$\overline{D}$.
CLK	18	I	Clock	Same TTL-level clock used to generate internal timing as used for 7220.
$\overline{CS}$	1	I	External	An active low signal used for multiplexing of FSAs. The FSA is disabled whenever $\overline{CS}$ is high (i.e., it presents a high impedance to the bus and ignores all bus activity).
$\overline{DATA.OUT.A}$, $\overline{DATA.OUT.B}$	11, 12	O	7230 CPG	Output data from the FIFO to the MBM generate circuitry. Used to write data into the bubble device (active low).
DET.A+, DET.A–, DET.B+, DET.B–	6, 7, 8, 9	I	7110A MBM	Differential signal lines from the MBM detector.
DIO	17	I/O	7220 BMC	The Serial Bus data line (a bidirectional active high signal).
$\overline{ENABLE.A}$, $\overline{ENABLE.B}$	13, 14	O	7230 CPG/7250	TTL-level outputs utilized as chip selects for other interface circuits. They shall be set and reset by the Command Decoder under instruction of the Controller (active low).
$\overline{ERR.FLG}$	4	O	7220 BMC	An error flag used to interrupt the Controller to indicate that an error condition exists. It shall be an open drain, active low signal.
$\overline{RESET}$	16	I	Power Fail Circuit	An active low signal that shall reset all flags and pointers in the FSA as well as disabling the chip as the CS signal does. The $\overline{RESET}$ pulse width must be 5 clock periods to assure the FSA is properly reset.
$\overline{SELECT.IN}$	19	I	7220 BMC	An input utilized for time-division multiplexing. An active low signal whose presence indicates that the FSA is to send or receive data from the Serial Bus during the next two clock periods.
$\overline{SELECT.OUT}$	2	O	7242 FSA	The $\overline{SELECT.IN}$ pulse delayed by two clocks. It shall be connected to the $\overline{SELECT.IN}$ pin of the next FSA. It is delayed by two clocks because the FSA is a dual-channel device. Channel A shall internally pass $\overline{SELECT.IN}$ to Channel B (delayed by one clock).
$\overline{SHIFT.CLK}$	15	I	7220 BMC	A Controller-generated clock signal that shall be used to clock data out of the bubble I/O Output Latch to the bubble module during a write operation and to cause bubble signals to be converted by the Sense Amp and clocked into the Bubble I/O Input Latch on a read.

7250

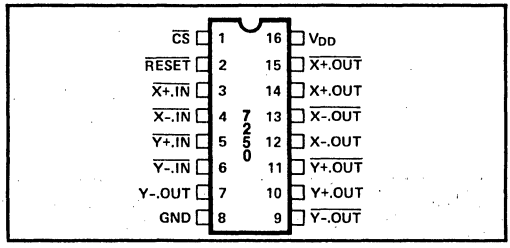


Figure 7. 7250 Pin Configuration

Table 4. 7250 Pin Description

Symbol	Pin No.	I/O	Source/Destination	Description
$\overline{CS}$	1	I	7242 FSA	Chip select. It is active low. When high chip is deselected and I_{DD} is significantly reduced.
$\overline{RESET}$	2	I	Power Fail Circuit	Active low input from RESET.OUT of 7220 Controller forces 7250 outputs inactive so that bubble memory is protected in the event of power supply failure.
$\overline{X+IN}$, $\overline{X-.IN}$	3, 4	I	7220 BMC	Active low inputs from Controller which turn on the high-current X outputs.
X−.OUT $\overline{X-.OUT}$ $\overline{X+.OUT}$ X+.OUT	12, 13, 14, 15	O	7254/VMOS	High-current outputs and their complements for driving the gates of the 7254 VMOS quad transistors which in turn drive the X coils of the bubble memory.
$\overline{Y+.IN}$, $\overline{Y-.IN}$	5, 6	I	7220 BMC	Active low inputs from Controller which turn on the high-current Y outputs.
Y−OUT $\overline{Y-.OUT}$ $\overline{Y+.OUT}$ Y+.OUT	7,9,10,11	O	7254/VMOS	High-current outputs and their complements for driving the gates of the 7254 VMOS quad transistors which in turn drive the Y coils of the bubble memory.

7254

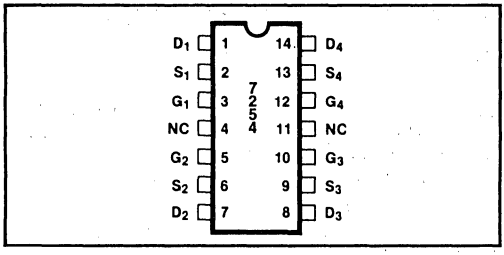


Figure 8. 7254 Pin Configuration

Table 5. 7254 Pin Configuration

Symbol	Pin No.	I/O	Source/Destination	Description
D_1	1	O	7110A MBM	Coil Drive Current
S_1	2	I	Ground	
G_1	3	I	7250 CPD	Gate Drive Signal
G_2	5	I	7250 CPD	Gate Drive Signal
S_2	6	I	+ 12V	
D_2	7	O	Coil	Coil Drive Current
D_3	8	O	Coil	Coil Drive Current
S_3	9	I	Ground	
G_3	10	I	7250 CPD	Gate Drive Signal
G_4	12	I	7250 CPD	Gate Drive Signal
S_4	13	I	+ 12V	
D_4	14	O	Coil	Coil Drive Current

ABSOLUTE MAXIMUM RATINGS*

7110A

Operating Temperature ..... −20°C to +85°C Case
Relative Humidity 95%
Shelf Storage Temperature (Data Integrity Not Guaranteed) ..... −65°C to +150°C
Voltage Applied to DET.SUPPLY ..... 14 Volts
Voltage Applied to PULSE.COM ..... 12.6 Volts
Continuous Current between DET.COM and Detector Outputs ..... 10 mA
Coil Current ..... 0.5A D.C.
External Magnetic Field for Non-Volatile Storage ..... 20 Oersteds
Non-Operating Handling Shock 200G
Operating Vibration (2 Hz to 2 kHz) 20G

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

SUPPORT I.C.'S*

	7230	7242	7250	7254
Temperature Under Bias	−40 to 100°C	−40 to 100°C	−40 to 100°C	−40 to 100°C
Storage Temperature	−65 to +150°C	−65 to +150°C	−65 to +150°C	−65 to +150°C
Voltage Input	−0.5 to +7V		−0.5 to V_{DD} +0.5	
V_{CC}	−0.5 to +7V	−0.5 to +7V		
V_{DD}	−0.5 to +12.6V	−0.5 to +14V		
Gate Voltage				15V
Output Current			250mA	
Power Dissipation 80°C	1W			1.05W
Power Dissipation 25°C				1.75W
Continuous Drain Current				2A
Peak Drain Current				3A

D.C. CHARACTERISTICS

7110A (T_C = Range Specified in temperature range table, V_{DD} = 12V ± 5%)

Parameter	7110A-1, 7110A-4 Limits			7110A-5 Limits[5]		Unit
	Min.	Nom.[1]	Max.	Min.	Max.	
RESISTANCE: PULSE.COM to GEN.A or GEN.B	7.5	30	59	8	61.5	ohms
RESISTANCE: PULSE.COM to REP.A or REP.B	7.5	20	26	8	27	ohms
RESISTANCE: PULSE.COM to SWAP.A or SWAP.B	42	100	149	40	155.5	ohms
RESISTANCE: PULSE.COM to BOOT.REP	2	8	24	3	25	ohms
RESISTANCE: PULSE.COM to BOOT.SWAP	3.5	15	36	4.5	37.5	ohms
RESISTANCE: DET.OUT A+ to DET.OUT.A−	570	1030	1903	530	1984	ohms
RESISTANCE: DET.OUT B+ to DET.OUT B−	570	1030	1903	530	1984	ohms
RESISTANCE: DET.COM to DET.SUPPLY	320	600	1050	290	1095	ohms
X.COIL RESISTANCE		4.2		329		ohms
Y.COIL RESISTANCE		2.7				ohms
X.COIL INDUCTANCE		135				μH
Y.COIL INDUCTANCE		93				μH
OPERATING POWER		1.20	1.75			watts
STANDBY POWER		0.25	.45			watts

7230 (T_A = Specified in temperature range table; V_{CC} = 5.0V ±5%, ±5% V_{DD} = 12V ±5%; unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
I_{IL}	Input Low Current			−0.4	mA	V_{IL} = 0.4V, V_{CC} = 5.25V
I_{IH}	Input High Current			20	μA	V_{IH} = V_{CC} = 5.25V
V_{IL}	Input Low Voltage			0.8	V	
V_{IH}	Input High Voltage	2.0			V	
V_C	Input Clamp Voltage			−1.5	V	I = −18 mA, V_{CC} = 4.75V
I_{CEX1}	Output Leakage Current (All Outputs except PWR.FAIL)			1.0	mA	V_{CC} = 5.25V, V_{DD} = 12.6V
I_{CEX2}	$\overline{\text{PWR.FAIL}}$ Output Leakage Current			40	μA	V_{OH} = V_{CC} = 5.25V
V_{OL}	$\overline{\text{PWR.FAIL}}$ Output Low Voltage			0.4	V	I_{OL} = 4 mA, V_{CC} = 4.75V
I_{CC1}	Current from V_{CC}—Selected		30	45	mA	CS = V_{IL}, V_{CC} = 5.25V
I_{DD1}	Current from V_{DD}—Selected		20	35	mA	CS = V_{IL}, V_{CC} = 5.25V
I_{DD2}	Current from V_{DD}—Power Down		12	19	mA	CS = V_{IH}, V_{DD} = 12.6V

7242 (T_A = Specified in temperature range table; V_{CC} = 5.0V +5%, −10% V_{DD} = 12V ±5%)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage	−0.5		0.8	V	
V_{IH}	Input High Voltage	2.2		V_{CC} +0.5	V	
V_{OL}	Output Low Voltage (All Outputs Except $\overline{\text{SELECT.OUT}}$)		.2	0.45	V	I_{OL} = 3.2mA
V_{OLSO}	Output Low Voltage ($\overline{\text{SELECT.OUT}}$)		.2	0.45	V	I_{OL} = 1.6mA
V_{OH}	Output High Voltage (All Outputs Except $\overline{\text{SELECT.OUT}}$)	2.4	3.0		V	I_{OH} = 400 μA
V_{OHSO}	Output High Voltage ($\overline{\text{SELECT.OUT}}$)	2.4			V	I_{OH} = 200 μA
V_{THR}	Detector Threshold	2.3	2.5	2.7	mV	V_{DD} = 12.0V
$\lvert I_{IL} \rvert$	Input Leakage Current		0	5	μA	$0 \leq V_{IN} \leq V_{CC}$
$\lvert I_{OFL} \rvert$	Output Float Leakage		0	10	μA	$0.45 \leq V_{OUT} \leq V_{CC}$
I_{CC}	Power Supply Current from V_{CC}		35	120	mA	
I_{DD}	Power Supply Current from V_{DD}		5	30	mA	

*Minimum V_{IH} is 2.2V for the 7242-5 device.

7250 (T_A = Specified in temperature range table; V_{DD} = 12V ±5%; −10%; unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
$\lvert I_{IN} \rvert$	Input Current			5	μA	V_I = 0.8V
V_{IL}	Low-Level Input Voltage			0.8	V	
V_{IH}	High-Level Input Voltage	2.2			V	
V_{OL1}	Output Low Voltage			2.0	V	I_{OL} = 100 mA
V_{OL2}	Output Low Voltage			0.2	V	I_{OL} = 10 mA
V_{OH1}	Output High Voltage	V_{DD}−2			V	I_{OH} = −100 mA
V_{OH2}	Output High Voltage	V_{DD}−0.2			V	I_{OH} = −10 mA
I_{OL}	Output Sink Current	100			mA	V_{OL} = 2.0V
$\lvert I_{OH} \rvert$	Output Source Current	100			mA	V_{OH} = V_{DD}−2.0V
I_{DD0}	Supply Current			4.5	mA	Chip Deselected: $\overline{CS}$ = V_{IH}, V_{DD} = 12.6V
I_{DD1}	Supply Current			75	mA	f = 100 kHz, V_{DD} = 12.6V, Outputs Unloaded

7254 All limits apply for N- and P-Channel transistors, T_A = specified in temperature range table; unless otherwise specified.

Symbol	Parameter	Limits				Test Conditions
		Min.	Typ.	Max.	Unit	
BV_{DSS}	Drain-Source Breakdown Voltage	20			V	V_{GS} = 0, I_D = 10 μA
$V_{GS}(th)$	Gate-Source Threshold Voltage	0.8			V	V_{GS} = V_{DS}, I_D = 1 mA, T_A = 25°C
		0.65			V	V_{GS} = V_{DS}, I_D = 1 mA, T_A = 85°C
I_{GSS}	Gate Leakage Current			100	nA	V_{GS} = 12V, V_{DS} = 0, T_A = 25°C
I_{DSS}	Drain Leakage Current			500	nA	V_{GS} = 0, V_{DS} = 20V, T_A = 25°C
R_{DS}	On-Resistance for sum of Q1+Q2, Q3+Q4 (Note 1)	2.0	2.5	3.0	Ω	V_{GS} = 11.4V, 1_D = 1A, T_A = 25°C
V_{F1}	Parasitic Diode Forward Voltage (Note 1)			.75	V	V_{GS} = 0V, I_D = 50 mA, T_A = 25°C
V_{F2}	Parasitic Diode Forward Voltage (Note 1)			1.20	V	V_{GS} = 0V, I_D = 1000 mA, T_A = 25°C

NOTE:
1. Pulse test—80 μs pulse, 1% duty cycle, r_{DS} increase 0.8%/°C.

DRIVE REQUIREMENTS CHARACTERISTICS[2] (T_C = Specified in temperature range table)

7110A

Symbol	Parameter	Min.	Nom.[1]	Max.	Units
f_R	Field Rotation Frequency	49.95	50.000	50.05	kHz
I_{px}	X.Coil Peak Current		600		ma
I_{py}	Y.Coil Peak Current		750		ma
Θ_{1x}	X.Coil Positive Turn-On Phase	268	270	272	degrees
Θ_{2x}	X.Coil Positive Turn-Off Phase	16	18	20	degrees
Θ_{3x}	X.Coil Negative Turn-On Phase	88	90	92	degrees
Θ_{4x}	X.Coil Negative Turn-Off Phase	196	198	200	
Θ_{1y}	Y.Coil Positive Turn-On Phase	0	0	0	degrees
Θ_{2y}	Y.Coil Positive Turn-Off Phase	106	108	110	degrees
Θ_{3y}	Y.Coil Negative Turn-On Phase	178	180	182	degrees
Θ_{4y}	Y.Coil Negative Turn-Off Phase	286	288	290	degrees

CONTROL PULSE REQUIREMENTS (T_C = Specified in temperature range table)[2]

7110A

Pulse	Amplitude			Pulse of Leading Edge (Degrees)			Width (Degrees)		
	Min.	Nom[1]	Max.	Min.	Nom.[1]	Max.	Min.	Nom.[1]	Max.
GEN.A, GEN.B CUT	62	70	81	266 86	270 (Odd) 90 (Even)	274 94	3	6.75	8
GEN.A, GEN.B TRANSFER	34	40	49	266 86	270 (Odd) 90 (Even)	274 94	86	90	94
REP.A, REP.B CUT	170	190	240	268	270	277	3	6.75	8
REP.A, REP.B TRANSFER	126	140	160	268	270	277	86	90	94
SWAP	111	120	134	176	180	184	513	517	521
BOOT.REP CUT	85	95	110	268	270	277	3	6.75	8
BOOT.REP TRANSFER	63	70	80	268	270	277	86	90	94
BOOT.SWAP [2]	63	70	80	176	180	184		360	

NOTES:

1. Nominal values are measured at T_C = 25°C.
2. Boot.Swap is not normally accessed during operating. It is utilized at the factory to write the index address and redundant loop information onto the bootstrap loops before shipment.

A.C. CHARACTERISTICS*

7230 V_{CC} = 5V ±5%, V_{DD} = 12V ±5%

Symbol	Parameter	Min.	Max.	Unit
t_{ENON}	Delay On		260	ns
t_{DISOFF}	Delay Off		70	ns
t_{CSON}	CS Enable		500	ns
t_{CSOFF}	CS Disable		70	ns

*These parameters are sample tested, not 100% tested.

POWER FAIL CHARACTERISTICS**

7230 T_A = See temperature range table.

	Min.	Typ.	Max.
V_{CCTH}	4.43V	4.60V	4.70V
V_{DDTH}	10.75V	11.10V	11.28V

**Power fail characteristics apply to 7110A Bubble Memory Data Integrity only and not to full memory operation.

A.C. CHARACTERISTICS (T_A = Specified in earlier temperature range table; V_{CC} = 5.0V +5%, −10%; V_{DD} = 12V ±5%; C_L = 120pF; unless otherwise noted)

7242

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_p	Clock Period	240	500	ns	
t_f	Clock Phase Width	.45 t_p	.55 t_p		
t_n t_f	Clock Rise and Fail Time		30	ns	
t_{SIC}	$\overline{\text{SELECT.IN}}$ Setup Time to CLK	50		ns	
t_{CDC}	C/$\overline{\text{D}}$ Setup Time to CLK	50		ns	
t_{CYC}	$\overline{\text{SELECT.IN}}$ or $\overline{\text{SHIFT.CLK}}$ Cycle Time	20 t_p			
t_{DC}	DIO Setup Time to Clock (Read Mode)	50		ns	
t_{CSC}	$\overline{\text{CS}}$ Setup Time to CLK	100		ns	
t_{RIC}	$\overline{\text{RESET.IN}}$ Setup Time to CLK	100		ns	
t_{IH}	Control Input Hold Time for C/D, $\overline{\text{SELECT.IN}}$ and DIO	10		ns	
t_{CSOL}	CLK to $\overline{\text{SELECT.OUT}}$ Leading Edge Delay		100	ns	C_L = 50 pF
t_{CSOT}	CLK to $\overline{\text{SELECT.OUT}}$ Trailing Edge Delay		80	ns	C_L = 50 pF
t_{CDV}	CLK to DIO Valid Delay*		100	ns	
t_{CDH}	CLK to DIO Hold Time*	0		ns	
t_{CDE}	CLK to DIO Enabled from Float*		100	ns	
t_{SIDE}	$\overline{\text{SELECT.IN}}$ Trailing Edge to DIO Enabled from Float*		70	ns	
t_{CDF}	CLK to DIO Entering Float*		100	ns	
t_{SCDO}	$\overline{\text{SHIFT.CLK}}$ to DATAOUT Delay*		200	ns	
t_{SCWR}	$\overline{\text{SHIFT.CLK}}$ Width (Read)	4 t_p	t_{CYC} − 11 t_p		
t_{SCWW}	$\overline{\text{SHIFT.CLK}}$ Width (Write)	t_p	t_{CYC} − 2 t_p		

7250 (T_A = Specified in earlier temperature range table; V_{DD} = 12V ±5%, unless otherwise specified)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
t_{p1}	Propagation Delay from $\overline{X+}$.IN, $\overline{X-}$.IN, $\overline{Y+}$.IN, $\overline{Y-}$.IN			100	ns	500 pF Load
t_{p2}	Propagation Delay from $\overline{CS}$ or $\overline{RESET}$			150	ns	500 pF Load
t_r	Rise Time (10% to 90%)			45	ns	500 pF Load
t_F	Fail Time (90% to 10%)			45	ns	500 pF Load
t_S	Skew Between an Output and Its Complements			20	ns	

7254 (T_A = 25°C)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
$T_{ON}(N)$	N-Channel Turn-On Time			20	ns	
$t_{ON}(P)$	P-Channel Turn-On Time			30	ns	
$t_{OFF}(N)$	N-Channel Turn-Off Time			20	ns	
$t_{OFF}(P)$	P-Channel Turn-Off Time			30	ns	

CAPACITANCE* (T_A = 25°C)

7230

Symbol	Parameter	Typ.	Max.	Unit	Test Conditions*
C_{IN}	Input Capacitance		10	pF	

*This parameter is periodically sampled and not 100% tested. Condition of measurement is f = 1 MHz.

7242 (T_A = 25°C, V_{CC} = 0V, f = 1 MHz)*

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance		10	pF	
C_{OUT}	Output Capacitance		10	pF	
C_{DIO}	DIO Capacitance		10	pF	

*DIO Write Mode.

7250 (T_A = 25°C, V_{DD} = 0V, V_{BIAS} = 2V, f = 1 MHz)*

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	

*This parameter is periodically sampled and is not 100% tested.

7254 T_A = 25°C

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
$C_{iss}(N)$	N-Channel Input Capacitance			175	pF	V_{GS} = 0, V_{DS} = 12V, f = 1 MHz
$C_{iss}(P)$	P-Channel Input Capacitance			190	pF	V_{GS} = 0, V_{DS} = 12V, f = 1 MHz

OUTPUT CURRENTS

7230 (T_A = Specified in earlier temperature range table; V_{CC} = 5.0V ±5%, V_{DD} = 12V ±5%)

Parameter	Current (mA)			Test Conditions Voltage Out	
	Min.	Nom.	Max.	Min.	Max.
GEN.A, GEN.B CUT	62	75	81	5.5	11.6
GEN.A, GEN.B TRANSFER	34	40	49	5.5	12.2
REP.A, REP.B CUT	170	200	240	3.4	9.3
REP.A, REP.B TRANSFER	126	145	160	3.4	11.4
SWAP	111	125	134	2.7	9.9
BOOT.REP CUT	85	100	110	7.7	12.1
BOOT.REP TRANSFER	63	75	80	7.7	12.4
BOOT.SWAP	63	75	80	9.0	12.3

PRELIMINARY

BPK 72A
1MBIT BUBBLE MEMORY PROTOTYPE KIT

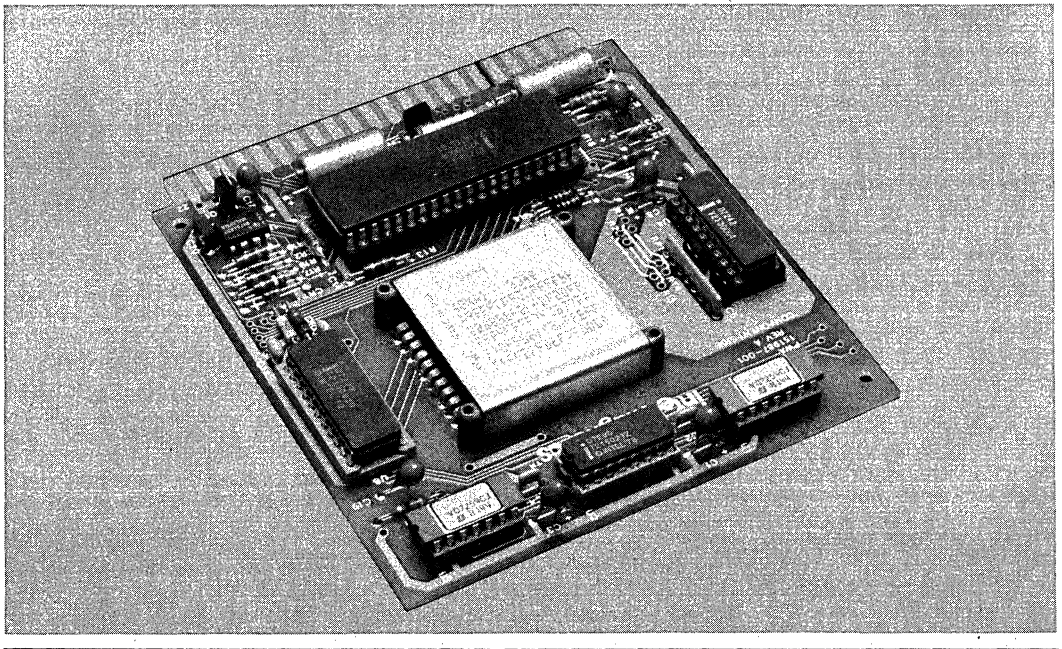
BPK 72A-1	0 To 75°C
BPK 72A-4	10°C To 55°C
BPK 72A-5*	– 20°C To 85°C

- **Assembled and Tested 1MBit Bubble Memory Prototype Kit on 4" x 4" PC Board**
- **Complete with Powerfail Data Protection and Clock Circuitry**
- **Built-in Error Detection/Correction**
- **Interfaces with Intel 8080/85/86/88 186/286 and Other Microprocessors**
- **Software Driver for Bubble Memory Kit Diskette for Intel MDS* System**
- **1Mbit, Non-Volatile, Read-Write, Solid-State Memory in Leaded Dense Package**
- **Average Random Access Time of 48ms**
- **Maximum Data Rate of 100K bit/sec**
- **Operates from + 5V and + 12V Power Supplies**
- **Complete Documentation and Interfacing Information Included**

The BPK 72A prototype kit is a completely assembled and tested 1Mbit bubble memory evaluation tool. It is ideal for the design engineer that wants the opportunity to fast evaluate how a bubble memory solution improves and adds value to an end-product by providing a compact solid-state memory that also reliably keeps the data at any powerdown.

Application information on microprocessor interfacing is included in the kit. A Bubble Memory Kit software driver is also included on a diskette for the Intel MDS* System.

*MDS is a registered trademark of Mohawk Data Sciences Corp.



OCTOBER 1983
ORDER NO. 230871-001

COMPONENT TEMPERATURE SPECIFICATIONS

Part Number	7110A Magnetic Bubble Memory Temperature		Support Circuits Min. Operating Temperature	Description
	Operating	Non-Volatile Storage		
BPK 72A-1*	0° to 75°C Case	−40° to 90°C	0° to 70°C Ambient	1 Mbit Bubble Memory Prototype Kit Assembled
BPK 72A-4*	10° to 55°C Case	−20° to 75°C	10° to 55°C Ambient	1 Mbit Bubble Memory Prototype Kit Assembled
BPK 72A-5	−20° to 85°C Case	−40° to 100°C	−20° to 85°C Ambient	1 Mbit Bubble Memory Prototype Kit

(* The bubble memory prototype kit is assembled and functionally tested to facilitate the prototyping process. The board is tested at the following ambient temperatures:

BPK 72A-1 0°C and 55°C BPK 72A-4 10°C and 40°C)

BPK 72A BUBBLE MEMORY PROTOTYPE KIT

The BPK 72A has a compact leaded bubble memory package and does not require a socket.

The bubble memory (7110A) and the support circuits (7230, 7242, 7250, 7254) in the BPK 72A kit are described in more detail on the BPK 70A Bubble Memory Subsystem data sheet. The Bubble Memory Controller (7220) in the BPK 72A kit is described in more detail on the 7220 data sheet.

For production purposes, the bubble memory and the support circuit components can be ordered as the BPK 70A Subsystem. The 7220, controlling up to eight BPK 70A's, is ordered separately.

Item	Description	Part Number
1 Mbit Bubble Memory	20-pin leaded package which provides 1 megabit of non-volatile storage.	7110A-1/7110A-4/ 7110A-5
Bubble Memory Controller	User interface, performs serial-to-parallel and parallel-to-serial data conversions. Generates timing signals.	7220-1/7220-4/7220-5
Current Pulse Generator	Converts digital timing signals to analog current pulses suited to the drive requirements of the 7110 MBM. The CPG provides the replicate, swap, generate, boot replicate, and bootswap pulses required by the MBM.	7230/7230-4/7230-5
Dual Formatter/Sense Amp	Provides direct interface to the 7110 Bubble Memory. The FSA contains on-chip sense amplifiers, a full FIFO data block buffer, burst error detection and correction circuits, and circuitry for handling of the bubble memory redundant loops.	7242/7242-5
Coil Predriver	Provides the high voltage, high current outputs to drive the 7254 Quad VMOS transistors.	7250/7250-5
2 Quad VMOS Coil Drive Transistors	Switches the required current to drive the X and Y coils of the 7110 Bubble Memory.	7254
Prefabricated Printed Circuit Board		IMB 72
Additional Items		
BPK 72 Bubble Memory Prototype Kit User's Manual	Literature, User's Manual, Software Guide	121685-002
Diskette	Software driver for bubble memory kit, configured for Intel MDS.	
Memory Components Handbook	Application Notes and Data Sheets.	210830
Seed Module	Recreates a lost seed bubble.	7901

SPECIFICATIONS

Capacity

128K Byte per BPK 72A

Performance

Avg. Access Time ..... 48 msec
Maximum Data Transfer Rate .... 100 Kbits/sec
Average Data Transfer Rate ..... 68 Kbits/sec

Data Organization

64 bytes per page
2048 pages per BPK 72A

Addressing Scheme

Logical page number

Environmental

Temperature: Temperature specifications
Operating Humidity: 0—95% Non-Condensing

BPK 72A POWER SUPPLY REQUIREMENTS

Voltage	Margin	Power Off/Power Fail Decay Rate
+12 Volt	±5%	less than 1.10 volts/msec
+5 Volt	±5%	less than 0.45 volts/msec

- Voltage sequencing — no restrictions
- Power on voltage rate of rise — no restrictions

BPK 72A POWER CONSUMPTION

Power (Watts)					
+5V (Maximum)	+12V (Maximum)	Total Active (Maximum)	Total Active (Typical)	Total Standby (Maximum)	Total Standby (Typical)
1.92	4.80	6.72	3.90	3.03	1.55

By using power switching techniques the bubble memory system's standby power consumption can be reduced to virtually zero. Application Note 164: **Using CMOS to Minimize Bubble Memory Power Consumption.** (Order Number: 230826-001)

7220
CONTROLLER FOR 1MBIT BPK 70A BUBBLE MEMORY SUBSYSTEM

7220-1	0° To 75°C
7220-4	10°C To 55°C
7220-5	−20°C To 85°C

- **Provides Interface between Host Microprocessor and 1 Mbit Bubble Subsystems**
- **Interfaces to 8080/85/86/88/186/286 and other Standard Microprocessors**
- **Controls Up to Eight BPK 70A-1, -4, Subsystems (or BPK70-1, -4)**
- **Controls up to Four BPK 70A-5 Bubble Memory Subsystems or BPK70-5**
- **16 Easy-to-Use Commands**
- **Three Modes of Data Transfer**
  - **— DMA**
  - **— Polled**
  - **— Interrupt**
- **Transfer of Single (64 bytes) or Multiple Pages of Data**

The 7220 is a complete 1 Mbit Bubble Memory Controller (BMC) that provides the interface between the microprocessor host and the 1 Mbit Bubble Memory Subsystem. All communication between the host processor and the bubble memory is performed through the controller.

The BMC interfaces easily to any Intel microprocessor or other standard microprocessor. The user has 16 easy-to-use commands available. Information such as the starting page location, the number of pages to be transferred and a read or write command is passed to the BMC before the read or write operation is initiated.

The design engineer writes a bubble memory software driver to integrate the bubble memory into his system. This interfacing with the BMC is similar to interfacing a disk drive controller. Application notes and manuals describe the details of interfacing to the BMC.

The BMC can transfer data in DMA, interrupt or polled mode. Data is transferred in and out of the bubble memory subsystem via the controller in single or multiple pages. A page size may vary from 64 bytes in a single bubble system and up to 512 bytes in an eight bubble system. (Maximum page size of 256 bytes for 7220-5.)

The BMC has an eight bit data bus plus parity bit. Word length expansion to 16 bit is possible by operating two controllers in parallel.

The BMC generates all the timing and control signals to the subsystem.

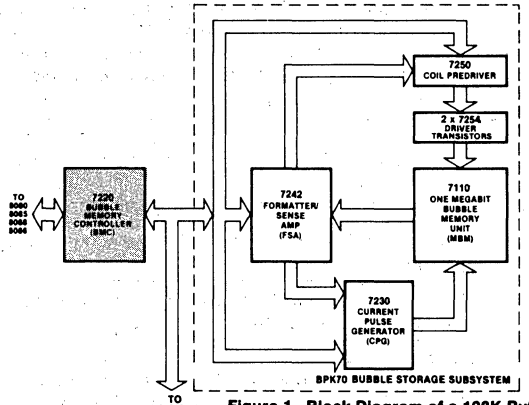


Figure 1. Block Diagram of a 128K Byte Bubble Storage System

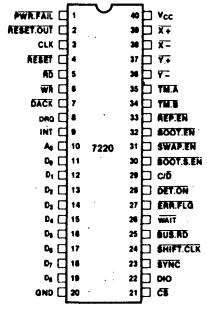


Figure 2. 7220 Pin Configuration

OCTOBER 1983
ORDER NO. 210286-002

One 7220-1 or 7220-4 BMC can control up to eight BPK 70A-1, -4 subsystems. This provides an easy expansion path to expand any 1 Mbit (128K byte) system by adding on additional subsystems. One 7220-5 can control up to four BPK 70A-5 subsystems.

The BMC is manufactured using Intel's high performance HMOS process and is packaged in a standard 40-pin dual-in-line package. All inputs are directly TTL compatible and the device uses a single +5 Volt supply.

HARDWARE DESCRIPTION

The 7220 Bubble Memory Controller is packaged in a 40-pin Dual In-Line Package (DIP). The following lists the individual pins and describes their function.

Table 1. Pin Description

Signal Name	Pin No.	I/O	Source/Destination	Description
V_{CC}	40	I		+5 VDC Supply
GND	20	I		Ground
$\overline{\text{PWR.FAIL}}$	1	I	7230 CPG	A low forces a controlled stop sequence and holds BMC in an IDLE state (similar to $\overline{\text{RESET}}$).
$\overline{\text{RESET.OUT}}$	2	O	7250 CPD/7242 FSA 7230 Reference Current Switch	An active low signal to disable external logic initiated by $\overline{\text{PWR.FAIL}}$ or $\overline{\text{RESET}}$ signals, but not active until a stopping point in a field rotation is reached (if the BMC is causing the bubble memory drive field to be rotated).
CLK	3	I	Host Bus	4 MHz, TTL-level clock.
$\overline{\text{RESET}}$	4	I	Host Bus	A low on this pin forces the interruption of any BMC sequencer activity, performs a controlled shut-down, and initiates a reset sequence. After the reset sequence is concluded, a low on this pin causes a low on the $\overline{\text{RESET.OUT}}$ pin, furthermore, the next BMC sequencer command must be either the Initialize or Abort command; all other commands are ignored.
$\overline{\text{RD}}$	5	I	Host Bus	A low on this pin enables the BMC output data to be transferred to the host data bus (D_0-D_8).
$\overline{\text{WR}}$	6	I	Host Bus	A low on this pin enables the contents of the host data bus (D_0-D_8) to be transferred to the BMC.
$\overline{\text{DACK}}$	7	I	Host Bus	A low signal is a DMA acknowledge. This notifies the BMC that the next memory cycle is available to transfer data. This line should be active only when DMA transfer is desired and the DMA ENABLE bit has been set. $\overline{\text{CS}}$ should not be active during DMA transfers except to read status. If DMA is not used, $\overline{\text{DACK}}$ requires an external pullup to V_{CC} (5.1K ohm).
DRQ	8	O	Host Bus	A high on this pin indicates that a data transfer between the BMC and the host memory is being requested.
INT	9	O	Host Bus	A high on this pin indicates that the BMC has a new status and requires servicing when enabled by the host CPU.
A_0	10	I	Host Bus	A high on this pin selects the command/status registers. A low on this pin selects the data register.
D_0-D_7	11-18	I/O	Host Bus	Host CPU data bus. An eight-bit bidirectional port which can be read or written by using the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ strobes. D_0 shall be the LSB.
D_8	19	I/O	Host Bus	Parity bit.

Table 1. Pin Description (Continued)

Signal Name	Pin No.	I/O	Source/Destination	Description
$\overline{CS}$	21	I	Host Bus	Chip Select Input. A high on this pin shall disable the device to all but DMA transfers (i.e., it ignores bus activity and goes into a high impedance state).
DIO	22	I/O	7242 FSA	A bidirectional active high data line that shall be used for serial communications with 7242 FSA devices.
$\overline{SYNC}$	23	O	7242 FSA	An active low output utilized to create time division multiplexing slots in a 7242 FSA chain. It shall also indicate the beginning of a data or command transfer between BMC and 7242 FSA.
$\overline{SHIFT.CLK}$	24	O	7242 FSA	A controller generated clock that initiates data transfer between selected FSAs and their corresponding bubble memory devices. The timing of $\overline{SHIFT.CLK}$ shall vary depending upon whether data is being read or written to the bubble memory.
$\overline{BUS.RD}$	25	O	*	An active low signal that indicates that the DIO line is in the output mode, i.e., BMC is sending data to FSA. It shall be used to allow off-board expansion of 7242 FSA devices.
$\overline{WAIT}$	26	I/O	*	A bidirectional pin that shall be tied to the $\overline{WAIT}$ pin on other BMCs when operated in parallel. It shall indicate that an interrupt has been generated and that the other BMCs should halt in synchronization with the interrupting BMC. $\overline{WAIT}$ is an open collector active low signal. Requires an external pullup resistor to V_{CC} (5.1K ohm).
$\overline{ERR.FLG}$	27	I	7242 FSA	An active low input generated externally by 7242 FSA indicating that an error condition exists. It is an open collector input which requires an external pullup resistor (5.1K ohm).
$\overline{DET.ON}$	28	O	*	An active low signal that indicates the system is in the read mode and may be detecting. It is useful for power saving in the MBM.
C/$\overline{D}$	29	O	7242 FSA	A high on this line indicates that the BMC is beginning an FSA command sequence. A low on this line indicates that the BMC is beginning a data transmit or receive sequence.
$\overline{BOOT.SW.EN}$	30	O	7230 CPG	An active low signal which may be used for enabling the BOOT.SWAP of the 7230 CPG.
$\overline{SWAP.EN}$	31	O	7230 CPG	An active low signal used to create the swap function in external circuits.
$\overline{BOOT.EN}$	32	O	7230 CPG	An active low signal enabling the bootstrap loop replicate function in external circuitry.
$\overline{REP.EN}$	33	O	7230 CPG	An active low signal used to enable the replicate function in external circuitry.
$\overline{TM.B}$	34	O	7230 CPG	An active low timing signal generated by the decoder logic for determining TRANSFER pulse width.
$\overline{TM.A}$	35	O	7230 CPG	An active low timing signal generated by the decoder logic for determining CUT pulse width.
$\overline{Y-}$, $\overline{Y+}$, $\overline{X-}$, $\overline{X+}$	36-39	O	7250 CPD	Four active low timing signals generated by the decoding logic and used to create coil drive currents in the bubble memory device.

*Not used in minimum (128K byte) system.

FUNCTIONAL DESCRIPTION

The 7220 Bubble Memory Controller provides the user interface to the bubble memory system. The BMC generates all memory system timing and control, maintains memory address information, interprets and executes user request for data transfers, and provides a Microprocessor-Bus compatible interface for the magnetic bubble memory system.

Figure 3 is a block diagram of the 7220 Bubble Memory Controller (BMC). The following paragraphs describe the functions of the individual functional sections of the BMC.

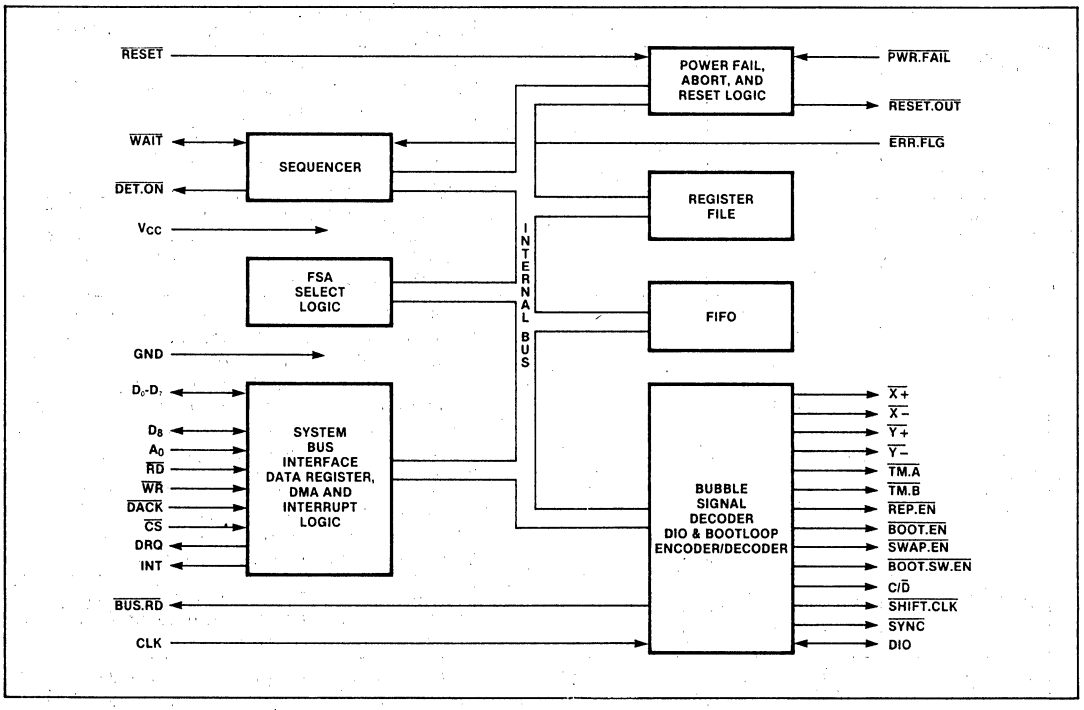


Figure 3. 7220 Bubble Memory Controller (BMC), Block Diagram

System Bus Interface—The System Bus Interface (SBI) logic contains the timing and control logic required to interface the BMC to a non-multiplexed bus. The logic also contains the circuitry to check and generate odd parity on transfers across the bus. The interface has input data, output data, and status data latches. The BMC can interface asynchronously to the host CPU. With a 4-MHz clock, it is capable of sustaining a 1.14 Mbyte per second transfer rate, while data is available in the BMC FIFO.

FIFO—The FIFO consists of a 40 × 8 bit FIFO RAM for data storage. The FIFO block also contains input and output data latches, providing double data buffering, to improve the R/W cycle times seen at the system bus interface. The FIFO may be used as a general purpose FIFO when a command is not being executed by the BMC Sequencer. In this mode, the FIFO READY status bit becomes a FIFO not-empty indicator indicating that the RAM and input/output latches have at least one byte of data.

DMA and Interrupt Logic—The DRQ pin has two functions:

(1) If the DMA enable bit in the enable register is set, the DRQ pin, in conjunction with the $\overline{DACK}$ pin, provides a standard DMA transfer capability; i.e., it has the ability to handshake with an 8257 or 9517/8237 DMA controller chip.

(2) If the DMA enable bit is reset, the DRQ pin acts as a "ready for data transfer interrupt" pin. It becomes active when 22 bytes may be read from or written into the BMC; it is reset when this condition no longer exists.

Register File—The register file contains 7 eight-bit registers that are accessible by the host CPU. Refer to the Register Section for details.

MBM Address Logic and RAM—The MBM address logic consists of the block length counter, starting address counter, adder, and MBM Address RAM. The MBM Address RAM is used to store the next available page address for each of up to 8 dual FSAs. The address maintained is the read address; the write address is generated, when needed, by adding a constant to the stored read address.

The block length counter enables multiple page transfers of up to 2048 pages in length.

The starting address counter is used as a register to hold the desired start address. Once the start address is reached, the counter is incremented on each subsequent page transfer so that its value is equal to the present read address.

DIO Bootloop Decoder/Encoder—Performs parallel-to-serial and serial-to-parallel conversions between the FIFO data and the serial bit stream on the DIO line. This block also generates the $\overline{\text{BUS.RD}}$ signal, which indicates the direction of data transfer on the DIO line (this is useful in situations which require external buffering on the DIO line). This block also contains the circuitry which decodes the bootloop data during a Read Bootloop or Initialize operation, and encodes the bootloop data during a Write Bootloop operation.

Sequencer—Controls the execution of commands by decoding the contents of its own internal ROM in which the BMC firmware is located. This block also sets and resets flags and status bits, and controls actions in other parts of the BMC.

Power Fail and Reset—Provides a means of resetting the bubble systems in an orderly manner, when activated by the $\overline{\text{PWR.FAIL}}$ signal, the $\overline{\text{RESET}}$ signal, or the ABORT command. The additive noise on the $\overline{\text{PWR.FAIL}}$ pin should be less than 150 mV for proper powerfail operation.

FSA Select Logic block contains the logic which controls the timing of the interaction between the BMC and the FSAs. The FSA selection is determined by the four high-order bits in the BLR and the four high-order bits in the AR, both set by the user.

Bubble Signal Decoder block contains the logic for creating all the MBM timing signals. The BMC to bubble memory interface consists of active low timing signals. The starting and stopping point of each signal is determined by the decoder logic. Each signal may occur every field rotation or only once in a number of field rotations. The field rotation in which a timing pulse occurs is controlled by the sequencer logic.

Figure 4 and Table 2 illustrate the typical timing signals for the BMC. These signals are described in the following paragraphs.

$\overline{X+}$, $\overline{X-}$, $\overline{Y+}$, and $\overline{Y-}$ go to the 7250 CPDs, and are used to enable the coil drive currents in the MBMs.

$\overline{\text{TM.A}}$ and $\overline{\text{TM.B}}$ go to the 7230 CPGs, and are used to determine, respectively, the pulse widths for the CUT and TRANSFER functions used in replicating and generating the bubbles.

Table 2. 7220 BMC Timing (Degrees)**

Signal	Start	Width	End
$\overline{X+}$	270°	108°	378°
$\overline{Y+}$	0°	108°	108°
$\overline{X-}$	90°	108°	198°
$\overline{Y-}$	180°	108°	288°
$\overline{\text{TM.A}}$ (ODD)	270°	4°	274.5°
$\overline{\text{TM.A}}$ (EVEN)	90°	4°	94.5°
$\overline{\text{TM.B}}$ (ODD)	270°	90°	360°
$\overline{\text{TM.B}}$ (EVEN)	90°	90°	180°
$\overline{\text{BOOT.EN}}$	252°	108°	360°
$\overline{\text{REP.EN}}$	252°	108°	360°
$\overline{\text{SWAP.EN}}$	180°	5.7°	697°
$\overline{\text{BOOT.SW.EN}}$*	180°	DC*	180°
$\overline{\text{SHIFTCLK (RD)}}$	186.75°	99°	285.75°
$\overline{\text{SHIFTCLK (WR)}}$	72°	288°	360°

*Stays low for 4118 field rotation periods when writing the MBM Bootloop.

**All phases relative to $\overline{Y-}$ start phase. All entries ± 1.26° except $\overline{\text{TM.A}}$ width which is ± 0.5°.

$\overline{\text{SWAP.EN}}$, $\overline{\text{REP.EN}}$, $\overline{\text{BOOT.SW.EN}}$, and $\overline{\text{BOOT.EN}}$ all go to the 7230 CPG. They are used to enable, respectively, the data swap, data replicate, boot swap, and boot replicate functions within the MBMs.

$\overline{\text{SHIFT.CLK}}$ goes to the FSAs. It is used to control the timing of events at the interface between each FSA and its corresponding MBM. (Refer to 7242 FSA Specification for a description of the BMC/FSA interface.)

$\overline{\text{SYNC}}$ and C/$\overline{\text{D}}$ control the serial communications between the BMC and the FSAs (on the DIO line).

USER-ACCESSIBLE REGISTERS

The user operates the bubble memory system by reading from or writing to specific registers within the bubble memory controller (BMC). The following paragraphs identify these registers and gives brief functional descriptions, including bit configurations and address assignments.

Register Addressing

Selection of the user-accessible registers depends on register address information sent from the user to the BMC. This address information is sent via a single address line (designated A_0) and data bus lines D_0 through D_4.

Both Command Register (CMDR) and Register Address Counter (RAC) are 4-bit registers which are loaded from D_0-D_3. The status register is selected and read by a single read request. The command register is selected and loaded by a single write request. The remaining registers are accessed indirectly, and the desired register is first selected by placing its address in the RAC, and then read or written with a subsequent read or write request.

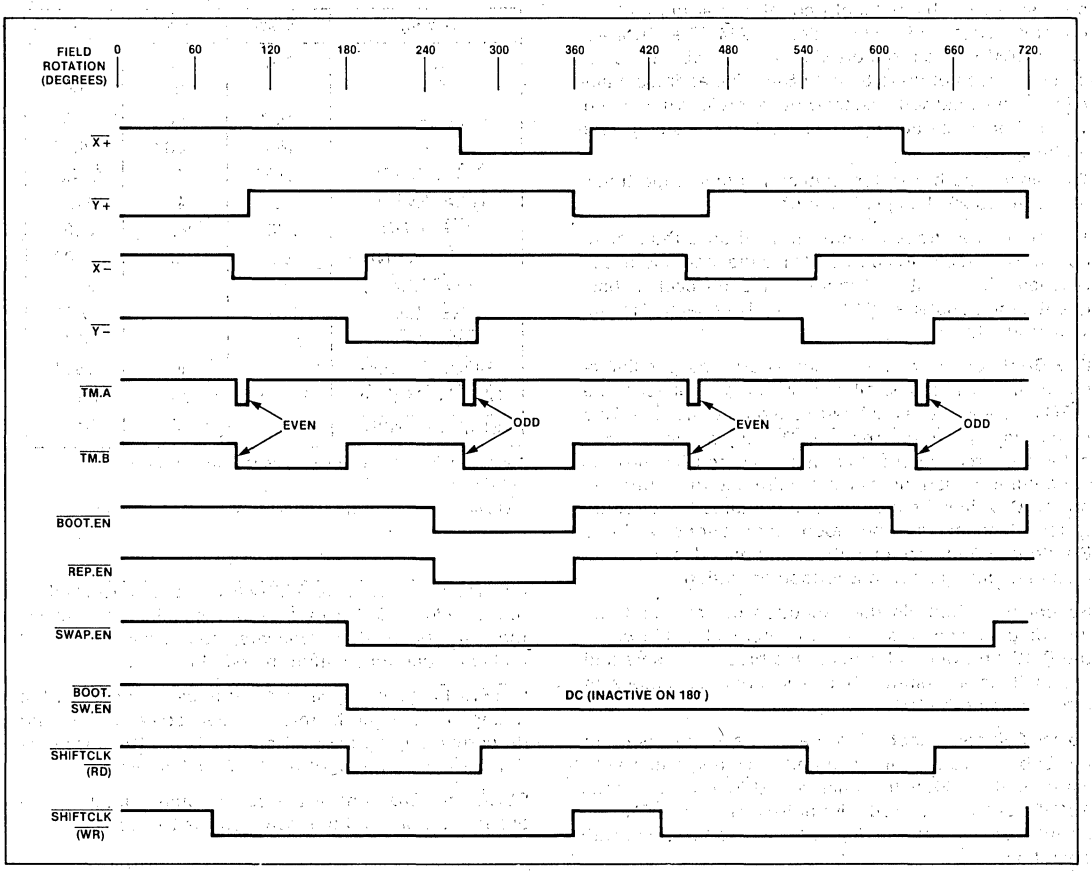


Figure 4. 7220 BMC Timing Diagram

Table 3 gives a complete listing of the address assignments for the user-accessible registers. The registers are listed in two groups. The first group (STR, CMDR, RAC) consists of those registers that are selected and accessed in one operation. The second group (UR, BLR, ER, AR, FIFO) consists of those registers that are addressed indirectly by the contents of RAC.

Table 3. Address Assignments for the User-Accessible Registers

A0 D7 D6 D5 D4 D3 D2 D1 D0	Symbol	Name of Register	Read/Write
1 0 0 0 1 C C C C	CMDR	Command Register	Write Only
1 0 0 M 0 B B B B	RAC	Register Address Counter	Write Only
1 S S S S S S S S	STR	Status Register	Read Only

Table 3. Address Assignments for the User-Accessible Registers (Continued)

RAC A0 B3 B2 B1 B0	Symbol	Name of Register	Read/Write
0 1 0 1 0	UR	Utility Register	Read or Write
0 1 0 1 1	BLR LSB	Block Length Register LSB	Write Only
0 1 1 0 0	BLR MSB	Block Length Register MSB	Write Only
0 1 1 0 1	ER	Enable Register	Write Only
0 1 1 1 0	AR LSB	Address Register LSB	Read or Write
0 1 1 1 1	AR MSB	Address Register MSB	Read or Write
0 0 0 0 0	FIFO	FIFO Data Buffer	Read or Write

SSSSSSSS = 8-bit status information returned to the user from the STR
CCCC = 4-bit command code sent to the CMDR by the user.
BBBB = 4-bit register address sent to the RAC by the user.
B3B2B1B0 = 4-bit contents of RAC at the time the user makes a read or write request with A0 = 0.
LSB = Least Significant Byte
MSB = Most Significant Byte
M = Modifier

The register file contains the registers with address 1010 through 1111. These registers are also called parametric registers because they contain flags and parameters that determine exactly how the BMC will respond to commands written to the CMDR.

To facilitate such operations, the BMC automatically increments the RAC by one count after each transfer of data to or from a parametric register.

The RAC increments from the initially loaded value through address 1111 and then on to 0000 (the FIFO address). When it has reached 0000, it no longer increments. All subsequent data transfers (with A0 = 0) will be to or from the FIFO until such time as the RAC is loaded with a different register address.

REGISTER DESCRIPTIONS

Command Register (CMDR) 4 Bits, Write Only

The user issues a command to the BMC by writing a 4-bit command code to the CMDR.

Table 4 lists the 4-bit command codes used to issue the sixteen commands recognized by the BMC:

Table 7 is a listing of the commands and their functions.

Table 4. Command Code Definitions

D3	D2	D1	Do	Command Name
0	0	0	0	Write Bootloop Register Masked
0	0	0	1	Initialize
0	0	1	0	Read Bubble Data
0	0	1	1	Write Bubble Data
0	1	0	0	Read Seek
0	1	0	1	Read Bootloop Register
0	1	1	0	Write Bootloop Register
0	1	1	1	Write Bootloop
1	0	0	0	Read FSA Status
1	0	0	1	Abort
1	0	1	0	Write Seek
1	0	1	1	Read Bootloop
1	1	0	0	Read Corrected Data
1	1	0	1	Reset FIFO
1	1	1	0	MBM Purge
1	1	1	1	Software Reset

The most commonly used commands in normal operation are:

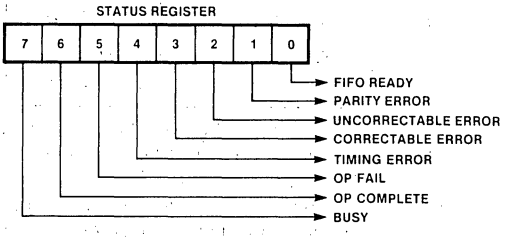
Initialize
Read Bubble Data
Write Bubble Data
Reset FIFO
Read Seek
Write Seek
Abort
Read Corrected Data
Software Reset
Read FSA Status
MBM Purge

Commands relating to the bootloop, and used only for diagnostic purposes, are:

Read Bootloop Register
Write Bootloop Register
Write Bootloop Register Masked
Read Bootloop
Write Bootloop

Status Register (STR) 8 Bits, Read Only

The user reads the BMC status register in response to an interrupt signal, or as part of the polling process in a polled data transfer mode. The status register provides information about error conditions, completion or termination of commands, and about the BMC's readiness to transfer data or accept new commands. The individual bit descriptions are as follows:



BUSY (when = 1) indicates that the BMC is in the process of executing a command. When equal to 0, BUSY indicates that the BMC is ready to receive a new command. In the case of Read Bubble Data, Read Bootloop, read Bootloop Register, or Read Corrected Data commands, BUSY may also indicate that the data has not been completely removed from the FIFO, and that DRQ is still active. BUSY will then drop as soon as DRQ does (after the user has finished reading the data remaining in the FIFO).

OP COMPLETE (when = 1) indicates the successful completion of a command.

OP FAIL (when = 1) indicates that the BUSY bit has gone inactive with either the TIMING ERROR or UNCORRECTABLE ERROR bits active.

TIMING ERROR (when = 1) indicates that a FSA has reported a timing error to the BMC, or that the host system has failed to keep up with the BMC, thereby causing the BMC FIFO to overflow or to underflow. TIMING ERROR is also set if no bootloop sync word is found during initialization, or if a Write Bootloop command is issued when the WRITE BOOTLOOP ENABLE bit is equal to zero in the enable register.

CORRECTABLE ERROR (when = 1) indicates that a FSA has reported to the BMC that a correctable error has been detected in the last data block transferred.

UNCORRECTABLE ERROR (when = 1) indicates that at least one FSA has reported to the BMC that an uncorrectable error has been detected in the last data block transferred.

PARITY ERROR (when = 1) indicates that the BMC's parity check circuitry has detected a parity error on a data byte sent to the BMC by the user on the data lines D_0-D_8.

FIFO READY has two functions. The FIFO READY functions are as follows:

NOTE: IF RAC ≠ FIFO, FIFO READY = 1

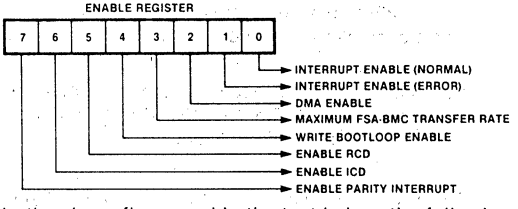
STATUS BITS		READ	WRITE
FIFO READY	BUSY		
1	1	data in FIFO	space in FIFO
0	1	no data	no space
1	0	— data in FIFO —	
0	0	— FIFO empty —	

Although the status word can be read at any time, the status information, bit 1 through 6, is not valid until the BUSY bit is low.

STR Bits 1 through 6 are reset when a new command is issued. They may also be reset by making a write request (WR = 0) to the BMC with $A_0 = 1$, $D_4 = 0$, and $D_5 = 1$ (Modifier Bit) (that is, writing the RAC with $D_5 = 1$). This operation also resets the "INT" pin to "0". NOTE: A byte of FIFO data can be lost when using this procedure if the RAC is written to other than the FIFO address when data is still present in FIFO.

Enable Register (ER) 8 Bits, Write Only

The user sets various bits of the enable register to enable or disable various functions within the BMC or the FSAs. The individual bit descriptions are as follows:



In the above figure and in the text below, the following abbreviations are used:

ICD = INTERNALLY CORRECT DATA
RCD = READ CORRECTED DATA
UCE = UNCORRECTABLE ERROR
CE = CORRECTABLE ERROR
TE = TIMING ERROR

ENABLE PARITY INTERRUPT enables the BMC to interrupt the host system (via the INT line) when the BMC detects a parity error on the data bus lines D_0-D_7.

ENABLE ICD enables the BMC to give the Internally Correct Data command to the FSAs when an error has been detected by the FSA's error detection and correction circuitry. Each FSA responds to such a command by internally cycling the data through its error correction network. When finished, the FSA returns status to the BMC as to whether or not the error is correctable. The value of ENABLE ICD affects the action of INTERRUPT ENABLE (ERROR).

ENABLE RCD enables the BMC to give the Read Corrected Data command to the FSAs when an error has been detected. This causes each FSA to correct the error (if possible) and also to transfer the corrected data to the BMC. The Read Corrected Data command is also used to read into the BMC data previously corrected by the FSA in response to an Internally Correct Data command. In either case, when the data transfer has been completed, the BMC reads each FSA's status to determine whether or not the error was correctable. In the case of an uncorrectable error, bad data may have been sent to the user. The value of ENABLE RCD affects the action of INTERRUPT ENABLE (ERROR).

WRITE BOOTLOOP ENABLE (when = 1) enables the bootloop to be written. If this bit is equal to zero, and a Write Bootloop command is received by the BMC, the command is aborted and the TIMING ERROR bit is set in the STR.

MFBTR controls the maximum burst transfer rate from FSA(s) to BMC FIFO. This rate is variable on the "last page" of a multiple page transfer. (In one page transfers the last page is the only page.) See Table 5 for effects of this bit on the various 7220 commands.

Table 5. MFBTR Bit Definitions

Number of MBMs Operated in Parallel	Maximum Required Host Interface Data Rate	MFBTR Bit	
		Read Command	Write Command
1	50K byte/sec	0	N/A
2	100K byte/sec	0	N/A
4	200K byte/sec	0	N/A
8	400K byte/sec	0	N/A
1	12.5K byte/sec	1	0
2	25K byte/sec	1	0
4	50K byte/sec	1	0
8	100K byte/sec	1	0

NOTE: The MFBTR bit should always be set to "0" for all commands except "Read Bubble Data."

DMA ENABLE (when = 1) enables the BMC to operate in DMA data transfer mode, using the DRQ and $\overline{DACK}$ signals in interaction with a DMA controller. When equal to zero, DMA ENABLE sets up the controller to support interrupt driven or polled data transfer.

INTERRUPT ENABLE (ERROR) selects error conditions under which the BMC stops command execution and interrupts the host processor (via the INT line). INTERRUPT ENABLE (ERROR) operates in conjunction with ENABLE ICD and ENABLE RCD.

Enable ICD	Enable RCD	Interrupt Enable (ERROR)	Interrupt Action
0	0	0	No interrupts due to errors
0	0	1	Interrupt on TE only
0	1	0	Interrupt on UCE or TE
0	1	1	Interrupt on UCE, CE, or TE
1	0	0	Interrupt on UCE or TE
1	0	1	Interrupt on UCE, CE, or TE
1	1	0	Not used
1	1	1	Not used

TE = Timing Error, CE = Correctable Error,
UCE = Uncorrectable Error.

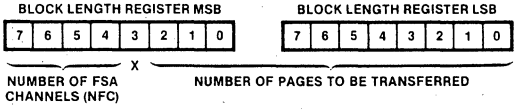
INTERRUPT ENABLE (NORMAL) (when = 1) enables the BMC to interrupt the host system (via the INT line), when a command execution has been successfully completed (OP COMPLETE = 1 in the STR).

Utility Register (UR) 8 Bits, Read or Write

The utility register is a general purpose register available to the user in connection with bubble memory system operations. It has no direct effect on the BMC operation, but is provided as a convenience to the user.

Block Length Register (BLR) 16 Bits, Write Only

The contents of the block length register determine the system page size and also the number of pages to be transferred in response to a single bubble data read or write command. The bit configuration is as follows:

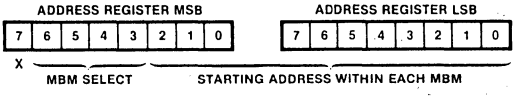


The system page size is proportional to the number of magnetic bubble memory modules (MBMs) operating in parallel during the data read or write operation. Each MBM requires two FSA channels. Bits 4 through 7 of BLR MSB actually specify the number of FSA channels to be accessed.

The BLR LSB, together with the 3 least significant bits of the BLR MSB, specify the number of pages to be transferred. Up to 2048 pages can be transferred in response to a single bubble data read or write command, hence the requirement for 11 bits. All 11 bits equal to zero specifies a 2048 page transfer.

Address Register (AR) 16 Bits, Read or Write

The contents of the address register determine which MBM group is to be accessed, and, within that group, what starting address location shall be used in a data read or write operation. The bit configuration is as follows:



Within each MBM there are 2048 possible starting address locations for a data read or write operation, hence the requirement for 11 bits in the starting address.

The selection of the MBMs to be read or written is specified by AR MSB Bits 3-6. The BMCs interpretation of these bits depends on the number of MBMs in a group, which is specified by BLR MSB Bits 4-7.

Table 6 shows which MBM groups are selected in response to given values for BLR MSB Bits 4-7 and AR MSB Bits 3-6. A 1-megabyte system (8 MBMs) is represented, with the FSA channels numbered 0 through F:

Table 6. Selection of FSA Channels

AR MSB Bits (6,5,4,3)	BLR MSB Bits (7,6,5,4)				
	0000	0001	0010	0100	1000
0000	0	0,1	0,1,2,3	0 to 7	0 to F
0001	1	2,3	4,5,6,7	8 to F	
0010	2	4,5	8,9,A,B		
0011	3	6,7	C,D,E,F		
0100	4	8,9			
0101	5	A,B			
0110	6	C,D			
0111	7	E,F			
1000	8				
1001	9				
1010	A				
1011	B				
1100	C				
1101	D				
1110	E				
1111	F				

The accessing of single FSA channels is done only as part of diagnostic processes. AR MSB Bit 7 is not used.

FIFO Data Buffer (FIFO) 40 x 8 Bits, Read or Write

The BMC FIFO is a 40-byte buffer through which data passes on its way from the FSAs to the user, or from the user to the FSAs. The FIFO allows the data transfer to proceed in an asynchronous and flexible manner, and relaxes timing constraints, both to the FSAs and also to the user's equipment. The user's system must, however, meet the data rate requirements. When the BMC is busy (executing a command) the FIFO functions as a data buffer. When the BMC is not busy, the FIFO is available to the user as a general purpose FIFO.

FUNCTIONAL OPERATION

The IC components used in the bubble memory systems have been designed with transparency in mind—that is, a maximum number of operations are handled by the hardware and firmware of these components.

Each one-Megabit Bubble Memory (MBM) operates in its own domain, and is unaffected by the number of bubble memories in the system. The roles played by the MBM's immediate support circuitry can be described as if the system contained only one MBM module.

Data Flow Within the Magnetic Bubble Memory (MBM) System (Single MBM Systems)

During a read operation, data flows as follows: The data from the MBM is input to the Formatter/Sense Amplifier (FSA). Data from each channel (A channel or B channel) of the MBM goes to the corresponding channel of the FSA. In the FSA, the data is paired up with the corresponding bit in the FSA's bootloop register to determine whether it represents data from a 'good' loop. If it does, the data bit is stored in the FSA FIFO. Error detection and correction (if enabled by the user) is applied to each block of 256 data bits.

From the FSA FIFO, data is sent to the bubble memory controller (BMC) in the form of a serial bit stream, via a one-line bidirectional data bus (DIO). The data is multiplexed onto the DIO line, with data bits coming alternately from the A and B channels of the FSA. The BMC outputs a $\overline{\text{SYNC}}$ pulse to the $\overline{\text{SELECT.IN}}$ input of the FSA. The FSA responds by placing a data bit from the A channel FIFO on the DIO line. One clock cycle later, a data bit from the B channel FIFO is placed on the DIO line. The BMC continues to output $\overline{\text{SYNC}}$ pulses, once every 20 or 80 clock cycles, each time receiving two data bits in return.

In the BMC, the data undergoes serial-to-parallel conversion, and is assembled into bytes, which are then placed in the BMC FIFO, which can hold 40 bytes of data. From this FIFO, the data bytes are written onto the user interface.

During a write operation, the data flow consists of the corresponding operations in the reverse order.

INTERFACING REQUIREMENTS

All communications between the host microprocessor, and the bubble memory is performed through the 7220 BMC. The **BPK 72 Bubble Memory Prototype Kit User's Manual**, Order Number: 121685, contains detailed information on how to use and interface the BMC. Below the general principles are described, for detailed guidelines please refer to the BPK 72 Manual. For software considerations, please also see Application Note AP-157. (Order Number: 230707)

First the hardware interfacing requirements and second the software interfacing requirements are described.

HARDWARE INTERFACE REQUIREMENTS

User Interface Signals

The source, destination and function of the user interface signals are described in Table 1 in the data sheet.

Table 7. Detailed Command Descriptions

Initialize	The BMC executes the Initialize command by first interrogating the bubble system to determine how many FSAs are present, then reading and decoding the bootloop from each MBM and storing the results in the corresponding FSA's bootloop register. All the parametric registers must be properly set up before issuing the Initialize command.
Read Bubble Data	The Read Bubble Data command causes data to be read from the MBMs into the BMC FIFO. The selection of the MBMs to be accessed and the starting address for the read operation is specified in the address register (AR). The block length register (BLR) specifies the number of system pages to be read. All the parametric registers must be properly set up before issuing the Read Bubble Data command.
Write Bubble Data	The Write Bubble Data command causes data to be read from the BMC FIFO and written into the MBMs. The selection of the MBMs to be accessed and the starting address for the write operation is specified in the address register (AR). The block length register (BLR) specifies the number of system pages to be written. All the parametric registers must be properly set up before issuing the Write Bubble Data command.
Read Seek	The Read Seek command rotates the selected MBMs to a designated page address location. No data transfer occurs. The positioning is such that the next data location available to be read is the specified (in AR) page address plus one. The Read Seek command may be used to reduce latency (access time) in cases where information is available for the user to predict the location of an impending read reference to the MBMs.

Table 7. Detailed Command Descriptions (Continued)

Write Seek	The Write Seek command rotates the selected MBMs to a designated page address location. No data transfer occurs. The positioning is such that the next data location available to be written is the specified (in AR) page address plus one. The Write Seek command may be used to reduce latency (access time) in cases where information is available for the user to predict the location of an impending write reference to the MBMs.
Abort	The Abort command causes a controlled termination of the command currently being executed by the BMC. The Abort command will be accepted by the BMC (and is typically issued) when the BMC is busy.
MBM Purge	The MBM Purge command clears all BMC registers, counters, and the MBM address RAM Furthermore, it determines how many FSA channels are present in the system and stores this value in the 7220. The "INITIALIZE" command uses this command as a subroutine.
Read Corrected Data	The Read Corrected Data command causes the BMC to read into the BMC FIFO a 256-bit block of data from the FIFO of each selected FSA channel, after an error has been detected. The data cycles through the error correction network of the FSA. After the data has been read, the FSA reports to the BMC whether or not the error was correctable. The Read Corrected Data command is used only when the system is in error correction mode (ENABLE ICD or ENABLE RCD set in the ER).
Software Reset	The Software Reset command clears the BMC FIFO and all registers, except those containing initialization parameters. It also causes the BMC to send the Software Reset command to selected FSAs in the system. No reinitialization is needed after this command.
Read FSA Status	The Read FSA Status command causes the BMC to read the 8-bit status register of all FSAs, and to store this information in the BMC FIFO. The Read FSA Status command is independent all parametric registers.
Read Bootloop Register	The Read Bootloop Register command causes the BMC to read the bootloop register of the selected FSA channels and to store this information in the BMC FIFO. Twenty bytes are transferred for each FSA channel selected.
Write Bootloop Register Masked	Proper operation of the FSAs during data transfer to or from the MBMs requires that the bootloop register contain (if error correction is used) exactly 270 logic 1s for each FSA bootloop register. The user may select any subset of 270 "good" loops from the total number of available loops (if error correction is not used, 270 replaced by 272). As an alternative, the Write Bootloop Register Masked command may be used. This command counts the number of logic 1s and masks out the remaining 1s after the proper count has been reached. The Initialize command uses this command as a subroutine.
Read Bootloop	The Read Bootloop command causes the BMC to read the bootloop from the selected MBM, and to store the decoded bootloop information in the BMC FIFO. The Initialize command uses this command as a subroutine.
Write Bootloop	The Write Bootloop command causes the existing contents of the selected MBM's bootloop to be replaced by new bootloop data based on 40 bytes of information stored in the FIFO (the user must actually write 41 bytes, where the 41st byte is all 0s). Encoding of the bootloop data is done by the BMC hardware.

System Timing

As shown on the timing diagrams in the WAVEFORM section the typical read/write cycle timing provides sufficient tolerance to allow most currently available microprocessors to be easily adapted to the BMC timing requirements.

User Data Transfer Rate Requirements

The maximum data rate for the user interface is a function of the number of MBMs operated in parallel as outlined in table 8. The rates listed must be considered in relation to the data transfer mode (polled, interrupt-driven, or DMA) to be implemented in order to be sure that the host system software and hardware are capable of keeping up with the data transfer. In other words, the BMC requires the host CPU to be able to sustain the maximum data rate transfer rate for the minimum data transfer (e.g., for a one bubble system keep up the transfer rate for at least 64 bytes = one page).

Table 8. User Data User Transfer Rate Requirements

Number of MBMs Operating in Parallel	Maximum Data Transfer Rate Between BMC FIFO and the FSAs during Write Bubble Data Commands	Maximum Data Transfer Rate Between BMC FIFO and the FSAs during Read Bubble Data Commands	
		MFBTR = 0	MFBTR = 1
1	12.5 kbytes/second	50 kbytes/second	12.5 kbytes/second
2	25 kbytes/second	100 kbytes/second	25 kbytes/second
4	50 kbytes/second	200 kbytes/second	50 kbytes/second
8	100 kbytes/second	400 kbytes/second	100 kbytes/second

Hardware Interfacing for Data Transfer

The BMC supports three data transfer modes, i.e. DMA, interrupt-driven and polled.

To support DMA, a hardware mechanism is required for servicing the BMC's data transfer requests. While several hardware implementations are possible, one common configuration is the Intel 8257 DMA controller.

To support an interrupt-driven system an Intel 8259 Programmable Interrupt Controller is often used.

The polled data transfer mode relies almost exclusively on the software interaction between the host processor and the BMC to control the transfer of data.

Multiple MBM-System

A BMC is capable of processing data and of supplying the required timing and control signals for operating up to four Bubble Storage Units (BSUs), each of which is capable of storing 128 kbytes of user data. A BSU consists of a 128 kbyte MBM and its five immediate IC support chips (i.e. a MBPK 70A-5 Kit).

SOFTWARE INTERFACE REQUIREMENTS

To use the BMC, the user has to write a "bubble memory software driver".

The bubble driver is responsible for all the system interaction with the bubble memory controller and is intrinsic to the efficient and reliable operation of the bubble system. The driver accepts bubble memory commands and command execution parameters from the application program, controls and monitors command execution, and returns operational status information to the application program at command completion. To perform all of these operations, the bubble driver must support the bit/byte level of the bubble memory controller's command and status registers and the parametric registers that define the operating mode, system configuration, and extent of the transfer.

The level of the software driver complexity is a function of the specific application needs. Regardless, a set of basic drivers must be developed that in turn are integrated into a system at the appropriate level. If an application program is small and simple, a basic bubble driver may simply be called from the main program.

At the highest level of driver sophistication, the application program treats the bubble system as a collection of named data areas of files similar to the way in which data is stored and retrieved in disk operating systems. At the file system level, an application program can ignore the mechanics of bubble storage and access and merely present a file name to the driver to open, read, or write, then close the desired bubble file.

Data Organization

From a software viewpoint, data logically is organized into blocks of bytes called pages. During data transfer operations, one or more of these pages are transferred between the bubble(s) and the host microprocessor. A page is the smallest increment of data that can be transferred; single bytes cannot be transferred. Conceptually, the data organization within a bubble memory is analogous to a disk system. Just as disk sector sizes are fixed when a disk is formatted, bubble page sizes are established, under software control when the bubble system is initialized.

For a single bubble system, the page size is fixed at either 64 bytes when error correction is implemented or 68 bytes without error correction, and the total number of pages available is 2048. In systems with multiple bubbles, page size can vary from 64 bytes (68 bytes without error correction) to 256 bytes (272 bytes without error correction) depending on the number of bubble devices in the system. Page size is directly proportional to the system data rate and also determines the total number of available pages (address field size). The selection of the appropriate page size depends primarily on the data rate supported by the system. The higher the data rate, the faster the microprocessor must respond to the demands of the bubble memory controller.

Buffering

The bubble memory controller includes a FIFO data buffer that, although only 40 bytes long, reconciles timing

differences between the parallel data transfer to or from the host microprocessor and the serial data transfer to or from the Bubble Memory Subsystem. Accordingly, when an application program requests data from a bubble, the software driver is responsible for keeping up with the FIFO for the duration of the data transfer in order to prevent the FIFO from overflowing or underflowing.

Command Execution

Command execution can be performed either in an interrupt driven mode or in a polled mode irrespective of the data transfer mode (polled, interrupt-driven, or DMA).

Data Transfer Mode

As described earlier in the hardware section, three data transfer modes are available (polled, interrupt-driven or DMA).

System performance, additional hardware and software overhead are all important considerations when choosing the appropriate mode for your application.

Error Correction

The bubble memory system has a built-in error correction. To insure highest data integrity, the error correction should always be used. Three levels of error correction are available.

Communication with the BMC

All communications between the host and the bubble memory actually are performed through the BMC. The BMC has two input/output (I/O) ports, an eight-bit bidirectional data port, and an eight-bit command/status port. Conceptually, a bubble memory system can be thought of as a disk system in that data in the bubble memory is organized into blocks called pages in bubble technology that are similar to disk sectors. Information such as starting page location, direction of transfer, and the number of pages to be transferred is passed to the BMC before the desired read or write operation is initiated.

The general procedure for communicating with the BMC is:

Set-up the BMC for data transfer communication by loading specific parameters in user-accessible registers.

Send the desired command.

Read the status register to determine if command is accepted.

If applicable, transfer (i.e., read or write) data.

Read the status register until BMC is not busy (or under some conditions "INT" pin).

Examine the status register to determine whether the operation was successful.

For all details and exceptions to this general description, see AP-157 or the BPK 72 User's Manual.

ABSOLUTE MAXIMUM RATINGS

Temperature under bias ..... −40 to +100°C
Storage Temperature ..... −65°C to +150°C
All Input or Output Voltages and
V_{CC} Supply Voltage ..... −0.5V to 7V

**NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS (T_A = see front page; V_{CC} = 5.0V +5%, −10%; * = 7220-5)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
V_{IL}	Input Low Voltage		0.8	V	
$V_{IH(1)}$	Input High Voltage (all but $\overline{PWR.FAIL}$)	2.0(2.2*)	V_{CC}+0.5V	V	
$V_{IH(2)}$	Input High Voltage ($\overline{PWR.FAIL}$)	2.5	V_{CC}+0.5V	V	
$V_{OL(1)}$	Output Low Voltage (All outputs except $\overline{DET.ON}$, $\overline{BUS.RD}$, $\overline{SHIFT.CLK}$, and $\overline{SYNC}$		.45	V	I_{OL} = 3.2 mA
$V_{OL(2)}$	Output Low Voltage $\overline{DET.ON}$, $\overline{BUS.RD}$, $\overline{SHIFT.CLK}$, $\overline{SYNC}$		.45	V	I_{OL} = 1.6 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 400μA
$\|I_{IL}\|$	Input Leakage Current		10	μA	$0 \leq V_{IN} \leq V_{CC}$
$\|I_{OFL}\|$	Output Float Leakage		10	μA	$0.45 \leq V_{OUT} \leq V_{CC}$
I_{CC}	Power Supply Current from V_{CC}		200	mA	

A.C. CHARACTERISTICS

(T_A = see front page; V_{CC} = 5.0V + 5%, −10%, C_L = 150pF; unless otherwise noted.)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_P	Clock Period	249.75	250.25	ns	
t_ϕ	Clock Phase Width (High Time)	.45 t_P	.55 t_P		
t_R,t_F	Input Signal Rise and Fall Time		30	ns	

FSA INTERFACE TIMINGS (under pin loading)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{CDV}	CLK to DIO Valid Delay		150	ns	Under Pin Loads*
t_{CDF}	CLK to DIO Entering Float	10	250	ns	Under Pin Loads*
t_{CDE}	CLK to DIO Enabled from Float		150	ns	Under Pin Loads*
t_{CDH}	CLK to DIO Hold Time	0		ns	Under Pin Loads*
t_{CSOL}	CLK to $\overline{SYNC}$ Leading Edge Delay		120	ns	Under Pin Loads*
t_{CSOT}	CLK to $\overline{SYNC}$ Trailing Edge Delay	10	100	ns	Under Pin Loads*
t_{DC}	DIO Setup Time to Clock	80		ns	Under Pin Loads*
t_{DHC}	DIO Hold Time from Clock	0		ns	Under Pin Loads*
t_{COL}	CLK to Output Leading Edge		150	ns	Under Pin Loads*
t_{COT}	CLK to Output Trailing Edge	0	190	ns	Under Pin Loads*
t_{EW}	$\overline{ERR.FLG}$ Pulse Width	200		ns	Under Pin Loads*
t_{SCFT}	$\overline{SHIFTCLK}$ to $\overline{Y-}$ Trailing Edge	80	200	ns	Under Pin Loads*

A.C. CHARACTERISTICS (Continued)

(T_A = see front page; V_{CC} = 5.0 ± 5%, −10%; C_L = 150pF; unless otherwise noted; ** = 7220-5.)

READ CYCLE (HOST INTERFACE)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{AC}	Select Setup to $\overline{RD}\downarrow$	0		ns	
t_{CA}	Select Hold from $\overline{RD}\uparrow$	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	200		ns	
t_{AD}	Data Delay from Address		150(200**)	ns	
t_{RD}	Data Delay from $\overline{RD}\downarrow$		150(200**)	ns	
t_{DF}	Output Float Delay	10	100	ns	
t_{DC}	DACK Setup to $\overline{RD}\downarrow$	0		ns	
t_{CD}	DACK Hold from $\overline{RD}\uparrow$	0		ns	
t_{KD}	Data Delay from $\overline{DACK}\downarrow$		150(200**)	ns	
t_{CYCR}	"Read" Cycle Time	(DMA Mode) $4t_p$-t_ϕ		ns	In non DMA mode. t_{CYCR} Min. = $6t_p$-t_ϕ

WRITE CYCLE (HOST INTERFACE)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{AC}	Select Setup to $\overline{WR}\downarrow$	0		ns	
t_{CA}	Select Hold from $\overline{WR}\uparrow$	0		ns	
t_{WW}	$\overline{WR}$ Pulse Width	200		ns	
t_{DW}	Data Setup to $\overline{WR}\uparrow$	200		ns	
t_{WD}	Data Hold from $\overline{WR}\uparrow$	0		ns	
t_{DC}	$\overline{DACK}$ Setup to $\overline{WR}\downarrow$	0		ns	
t_{CD}	$\overline{DACK}$ Hold from $\overline{WR}\uparrow$	0		ns	
t_{CYCW}	"Write" Cycle Time	$4t_P + t_{ww}$			
t_{CQ}	Request Hold from $\overline{RD}$ or $\overline{WR}$ (Non-Burst Mode)		200	ns	
t_{DEADW}	Inactive Time between $\overline{WR}\uparrow$ and $\overline{WR}\downarrow$	$4t_P$		ns	
t_{DEADR}	Inactive Time between $\overline{RD}\uparrow$ and $\overline{RD}\downarrow$	150			

MD7250-MD7230 INTERFACE TIMINGS

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{CBL}	CLK to Bubble Signal Leading Edge		250(275**)	ns	Under Pin Loads*
t_{CBT}	CLK to Bubble Signal Trailing Edge		250(275**)	ns	Under Pin Loads*

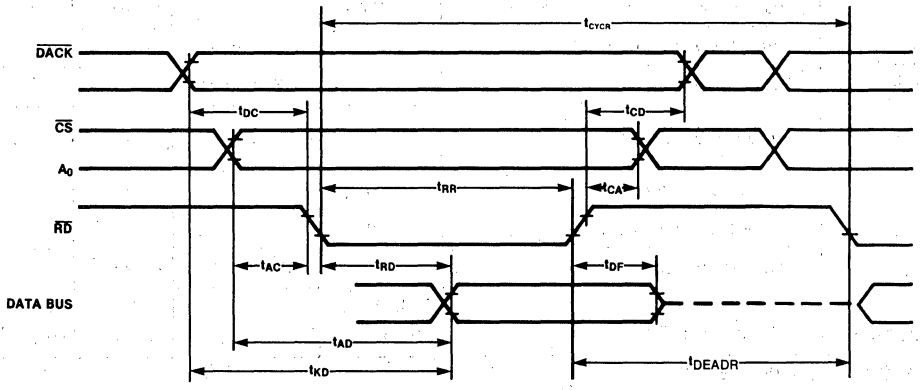
*Bubble Pin Loads Shown Below

PIN LOADINGS

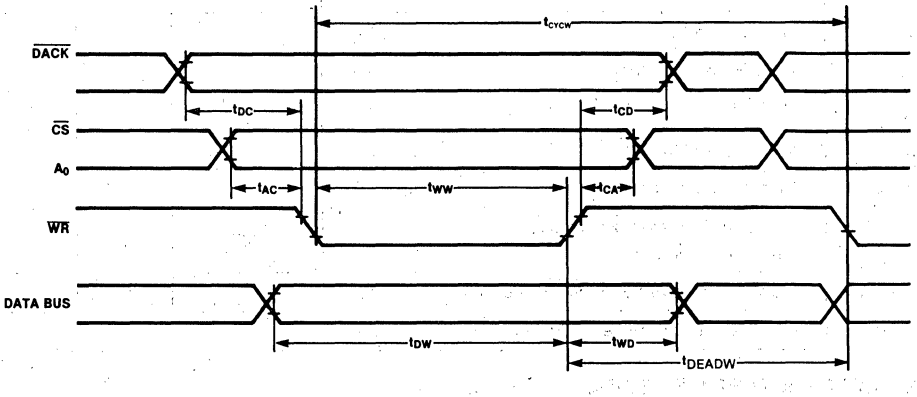
Pin Names	Value	Unit
$\overline{X+}$, $\overline{X-}$, $\overline{Y+}$, $\overline{Y-}$	150	pF
$\overline{TM.A}$, $\overline{TM.B}$, $\overline{REP.EN}$, $\overline{BOOT.EN}$, $\overline{SWAP.EN}$, $\overline{BOOT.SW.EN}$, C/D, $\overline{ERR.FLG}$, $\overline{WAIT}$, $\overline{SYNC}$, DIO	100	pF
$\overline{DET.ON}$ & $\overline{SHIFT.CLK}$	100	pF
$\overline{BUS.READ}$	10	pF

WAVEFORMS

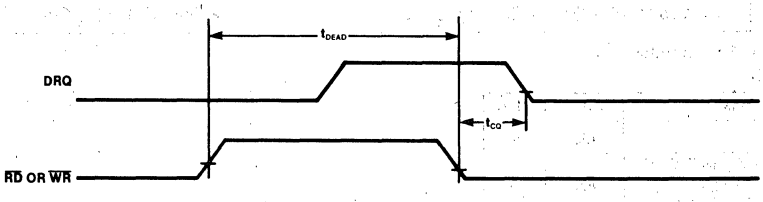
READ (HOST INTERFACE)



WRITE (HOST INTERFACE)



DMA (HOST INTERFACE)



7224
CONTROLLER FOR 4MBIT BPK 5V74 BUBBLE MEMORY SUBSYSTEM

- **Provides Interface between Host Microprocessor and 4 Mbit Bubble Memory Subsystems**
- **Interfaces to 8080/85/86/88/186/286 and Other Standard Microprocessors**
- **Controls up to Eight Bubble Memory Subsystems**
- **18 Easy-to-Use Commands**
- **Three Modes of Data Transfer**
  - **— DMA**
  - **— Polled**
  - **— Interrupt**
- **Transfer of Single (64 bytes) or Multiple Pages of Data**

The 7224 is a complete 4 Mbit Bubble Memory Controller (BMC) that provides the interface between the microprocessor host and the 4 Mbit Bubble Memory Subsystem. All communication between the host processor and the bubble memory is performed through the controller.

The BMC interfaces easily to any Intel microprocessor or other standard microprocessor. The user has 18 easy-to-use commands available. Information such as the starting page location, the number of pages to be transferred and a read or write command is passed to the BMC before the read or write operation is initiated.

The 18 commands of the 4Mbit BMC is a superset of the 1Mbit BMC's 16 commands providing an easy up-grade of software from the 1Mbit to the 4Mbit system.

The design engineer writes a bubble memory software driver to integrate the bubble memory into his system. This interfacing with the BMC is similar to interfacing a disk drive controller. Application notes and manuals describe the details of interfacing to the BMC.

The BMC can transfer data in DMA, interrupt or polled mode. Data is transferred in and out of the bubble memory subsystem via the controller in single or multiple pages. A page size may vary from 64 bytes in a single bubble system and up to 512 bytes in an eight bubble system.

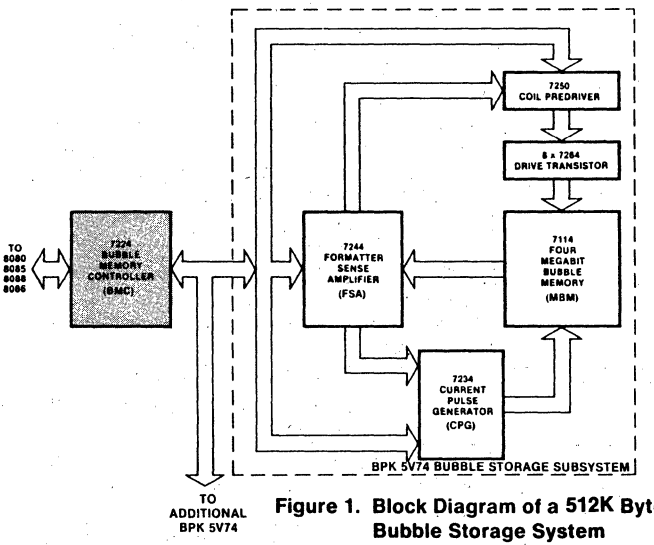


Figure 1. Block Diagram of a 512K Byte Bubble Storage System

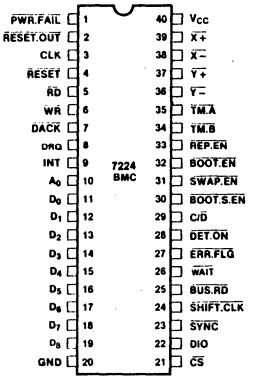


Figure 2. Pin Configuration

OCTOBER 1983
ORDER NO. 230865-001

The BMC has an eight bit data bus plus parity bit. Word length expansion to 16 bit is possible by operating two controllers in parallel.

The BMC generates all the timing and control signals to the BPK 5V74 subsystem.

One BMC can control up to eight BPK 5V74 subsystems. This provides an easy expansion path to expand any 4 Mbit (512K byte) system up to 32Mbit (4Mbyte) by adding on additional BPK 5V74 subsystems.

The BMC is manufactured using Intel's high performance HMOS process and is packaged in a standard 40-pin dual-in-line package. All inputs are directly TTL compatible and the device uses a single +5 Volt supply.

HARDWARE DESCRIPTION

The 7224 Bubble Memory Controller is packaged in a 40-pin Dual In-Line Package (DIP). The following lists the individual pins and describes their function.

Table 1. Pin Description

Signal Name	Pin No.	I/O	Source/Destination	Description
V_{CC}	40	I		+5 VDC Supply
GND	20	I		Ground
$\overline{PWR.FAIL}$	1	I	7234 CPG	A low forces a controlled stop sequence and holds BMC in an IDLE state (similar to $\overline{RESET}$).
$\overline{RESET.OUT}$	2	O	7250 CPD/7244 FSA 7234 Reference Current Switch	An active low signal to disable external logic initiated by $\overline{PWR.FAIL}$ or $\overline{RESET}$ signals, but not active until a stopping point in a field rotation is reached (if the BMC is causing the bubble memory drive field to be rotated).
CLK	3	I	Host Bus	2 MHz, TTL-level clock.
$\overline{RESET}$	4	I	Host Bus	A low on this pin forces the interruption of any BMC sequencer activity, performs a controlled shut-down, and initiates a reset sequence. After the reset sequence is concluded, a low on this pin causes a low on the $\overline{RESET.OUT}$ pin, furthermore, the next BMC sequencer command must be either the Initialize or Abort command; all other commands are ignored.
$\overline{RD}$	5	I	Host Bus	A low on this pin enables the BMC output data to be transferred to the host data bus (D_0-D_8).
$\overline{WR}$	6	I	Host Bus	A low on this pin enables the contents of the host data bus (D_0-D_8) to be transferred to the BMC.
$\overline{DACK}$	7	I	Host Bus	A low signal is a DMA acknowledge. This notifies the BMC that the next memory cycle is available to transfer data. This line should be active only when DMA transfer is desired and the DMA ENABLE bit has been set. $\overline{CS}$ should not be active during DMA transfers except to read status. If DMA is not used, $\overline{DACK}$ requires an external pullup to V_{CC} (5.1K ohm).
DRQ	8	O	Host Bus	A high on this pin indicates that a data transfer between the BMC and the host memory is being requested.
INT	9	O	Host Bus	A high on this pin indicates that the BMC has a new status and requires servicing when enabled by the host CPU.
A_0	10	I	Host Bus	A high on this pin selects the command/status registers. A low on this pin selects the data register.
D_0-D_7	11-18	I/O	Host Bus	Host CPU data bus. An eight-bit bidirectional port which can be read or written by using the $\overline{RD}$ and $\overline{WR}$ strobes. D_0 shall be the LSB.
D_8	19	I/O	Host Bus	Parity bit.

Table 1. Pin Description (Continued)

Signal Name	Pin No.	I/O	Source/Destination	Description
$\overline{CS}$	21	I	Host Bus	Chip Select Input. A high on this pin shall disable the device to all but DMA transfers (i.e., it ignores bus activity and goes into a high impedance state).
DIO	22	I/O	7244 FSA	A bidirectional active high data line that shall be used for serial communications with 7244 FSA devices.
$\overline{SYNC}$	23	O	7244 FSA	An active low output utilized to create time division multiplexing slots in a 7244 FSA chain. It shall also indicate the beginning of a data or command transfer between BMC and 7244 FSA.
$\overline{SHIFT.CLK}$	24	O	7244 FSA	A controller generated clock that initiates data transfer between selected FSAs and their corresponding bubble memory devices. The timing of $\overline{SHIFT.CLK}$ shall vary depending upon whether data is being read or written to the bubble memory.
$\overline{BUS.RD}$	25	O	To User External Circuit	An active low signal that indicates that the DIO line is in the output mode, i.e., BMC is sending data to FSA. It shall be used to allow off-board expansion of 7244 FSA devices.
$\overline{WAIT}$	26	I/O	To Alternate Controller(s) When User System Uses More Than One Controller.	A bidirectional pin that shall be tied to the $\overline{WAIT}$ pin on other BMCs when operated in parallel. It shall indicate that an interrupt has been generated and that the other BMCs should halt in synchronization with the interrupting BMC. $\overline{WAIT}$ is an open collector active low signal. Requires an external pullup resistor to V_{CC} (5.1K ohm).
$\overline{ERR.FLG}$	27	I	7244 FSA	An active low input generated externally by 7244 FSA indicating that an error condition exists. It is an open collector input which requires an external pullup resistor (5.1K ohm).
$\overline{DET.ON}$	28	O	To User External Circuit	An active low signal that indicates the system is in the read mode and may be detecting. It is useful for power saving in the MBM.
C/$\overline{D}$	29	O	7244 FSA	A high on this line indicates that the BMC is beginning an FSA command sequence. A low on this line indicates that the BMC is beginning a data transmit or receive sequence.
$\overline{BOOT.SW.EN}$	30	O	7234 CPG	An active low signal which may be used for enabling the BOOT.SWAP of the 7234 CPG.
$\overline{SWAP.EN}$	31	O	7234 CPG	An active low signal used to create the swap function in external circuits.
$\overline{BOOT.EN}$	32	O	7234 CPG	An active low signal enabling the bootstrap loop replicate function in external circuitry.
$\overline{REP.EN}$	33	O	7234 CPG	An active low signal used to enable the replicate function in external circuitry.
$\overline{TM.B}$	34	O	7234 CPG	An active low timing signal generated by the decoder logic for determining TRANSFER pulse width.
$\overline{TM.A}$	35	O	7234 CPG	An active low timing signal generated by the decoder logic for determining CUT pulse width.
$\overline{Y-}$, $\overline{Y+}$, $\overline{X-}$, $\overline{X+}$	36-39	O	7250 CPD	Four active low timing signals generated by the decoding logic and used to create coil drive currents in the bubble memory device.

FUNCTIONAL DESCRIPTION

The 7224 Bubble Memory Controller provides the user interface to the bubble memory system. The BMC generates all memory system timing and control, maintains memory address information, interprets and executes user request for data transfers, and provides a Microprocessor-Bus compatible interface for the magnetic bubble memory system.

Figure 3 is a block diagram of the 7224 Bubble Memory Controller (BMC). The following paragraphs describe the functions of the individual functional sections of the BMC.

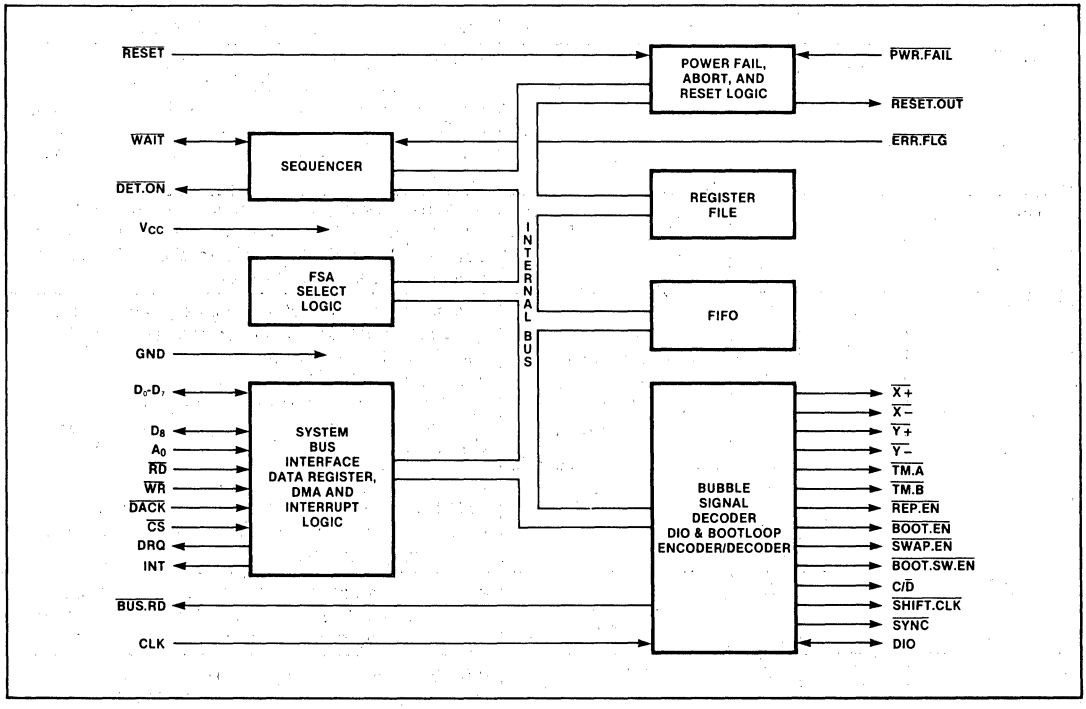


Figure 3. 7224 Bubble Memory Controller (BMC), Block Diagram

System Bus Interface—The System Bus Interface (SBI) logic contains the timing and control logic required to interface the BMC to a non-multiplexed bus. The logic also contains the circuitry to check and generate odd parity on transfers across the bus. The interface has input data, output data, and status data latches. The BMC can interface asynchronously to the host CPU. With a 2-MHz clock, it is capable of sustaining a 1 Mbyte per second transfer rate while data is available in the BMC FIFO.

FIFO—The FIFO consists of a 40 x 8 bit FIFO RAM for data storage. The FIFO block also contains input and output data latches, providing double data buffering, to improve the R/W cycle times seen at the system bus interface. The FIFO may be used as a general purpose FIFO when a command is not being executed by the BMC Sequencer. In this mode, the FIFO READY status bit becomes a FIFO not-empty indicator indicating that the RAM and input/output latches have at least one byte of data.

DMA and Interrupt Logic—The DRQ pin has two functions:

(1) If the DMA enable bit in the enable register is set, the DRQ pin, in conjunction with the $\overline{DACK}$ pin, provides a standard DMA transfer capability; i.e., it has the ability to handshake with an 8257 or 9517/8237 DMA controller chip.

(2) If the DMA enable bit is reset, the DRQ pin acts as a "ready for data transfer interrupt" pin. It becomes active when 22 bytes may be read from or written into the BMC; it is reset when this condition no longer exists.

Register File— The register file contains 6 eight-bit registers that are accessible by the host CPU. Refer to the Register Section for details.

MBM Address Logic and RAM—The MBM address logic consists of the block length counter, starting address counter, adder, and MBM Address RAM. The MBM Address RAM is used to store the next available page address for each of up to 8 dual FSAs. The address maintained is the read address; the write address is generated, when needed, by adding a constant to the stored read address.

The block length counter enables multiple page transfers of up to 2048 pages in length.

The starting address counter is used as a register to hold the desired start address. Once the start address is reached, the counter is incremented on each subsequent page transfer so that its value equal to the present read address. There are 8192 possible starting addresses.

DIO Bootloop Decoder/Encoder—Performs parallel-to-serial and serial-to-parallel conversions between the FIFO data and the serial bit stream on the DIO line. This block also generates the $\overline{\text{BUS.RD}}$ signal, which indicates the direction of data transfer on the DIO line (this is useful in situations which require external buffering on the DIO line). This block also contains the circuitry which decodes the bootloop data during a Read Bootloop or Initialize operation, and encodes the bootloop data during a Write Bootloop operation.

Sequencer—Controls the execution of commands by decoding the contents of its own internal ROM in which the BMC firmware is located. This block also sets and resets flags and status bits, and controls actions in other parts of the BMC.

Power Fail and Reset—Provides a means of resetting the bubble systems in an orderly manner, when activated by the $\overline{\text{PWR.FAIL}}$ signal, the $\overline{\text{RESET}}$ signal, or the ABORT command. The additive noise on the $\overline{\text{PWR.FAIL}}$ pin should be less than 150 mV for proper powerfail operation.

FSA Select Logic block contains the logic which controls the timing of the interaction between the BMC and the FSAs. The FSA selection is determined by the four high-order bits in the BLR and the three high-order bits in the AR, both set by the user.

Bubble Signal Decoder block contains the logic for creating all the MBM timing signals. The BMC to bubble memory interface consists of active low timing signals. The starting and stopping point of each signal is determined by the decoder logic. Each signal may occur every field rotation or only once in a number of field rotations. The field rotation in which a timing pulse occurs is controlled by the sequencer logic.

Figure 4 and Table 2 illustrate the typical timing signals for the BMC. These signals are described in the following paragraphs.

$\overline{X+}$, $\overline{X-}$, $\overline{Y+}$, and $\overline{Y-}$ go to the 7250 CPDs, and are used to enable the coil drive currents in the MBMs.

$\overline{\text{TM.A}}$ and $\overline{\text{TM.B}}$ go to the 7234 CPGs, and are used to determine, respectively, the pulse widths for the CUT and TRANSFER functions used in replicating and generating the bubbles.

Table 2. 7224 BMC Timing (Degrees)**

Signal		Start	Width
$\overline{X\pm}$		270°	108°
$\overline{Y+}$		0°	108°
$\overline{X-}$		90°	108°
$\overline{Y-}$		180°	108°
$\overline{\text{TM.A}}$ (LATE)		279°	94.5°
$\overline{\text{TM.A}}$ (EARLY)		99°	94.5°
$\overline{\text{TM.B}}$ (LATE)		279°	90°
$\overline{\text{TM.B}}$ (EARLY)		99°	90°
$\overline{\text{BOOT.EN}}$		261°	126°
$\overline{\text{REP.EN}}$		261°	126°
$\overline{\text{SWAP.EN}}$		180°	153°
$\overline{\text{BOOT.SW.EN}}$		180°	DC*
SHIFTCLK (RD)	LATE	230.5°	45°
	EARLY	40.5°	63°
SHIFTCLK (WR)	LATE	261°	126°
	EARLY	81°	126°

*Stays low for 8211 field rotation periods when writing the MBM Bootloop.

**All phases relative to Y+ start phase. All entries ±1.26 except $\overline{\text{TM.A}}$ width which is ±0.5.

$\overline{\text{SWAP.EN}}$, $\overline{\text{REP.EN}}$, $\overline{\text{BOOT.SW.EN}}$, and $\overline{\text{BOOT.EN}}$ all go to the 7234 CPG. They are used to enable, respectively, the data swap, data replicate, boot swap, and boot replicate functions within the MBMs.

$\overline{\text{SHIFT.CLK}}$ goes to the FSAs. It is used to control the timing of events at the interface between each FSA and its corresponding MBM. (Refer to 7244 FSA Specification for a description of the BMC/FSA interface.)

$\overline{\text{SYNC}}$ and C/$\overline{\text{D}}$ control the serial communications between the BMC and the FSAs (on the DIO line).

USER-ACCESSIBLE REGISTERS

The user operates the bubble memory system by reading from or writing to specific registers within the bubble memory controller (BMC). The following paragraphs identify these registers and gives brief functional descriptions, including bit configurations and address assignments.

Register Addressing

Selection of the user-accessible registers depends on register address information sent from the user to the BMC. This address information is sent via a single address line (designated A_0) and data bus lines D_0 through D_5.

The Command Register (CMDR) is an 8-bit register which is loaded from D_0 — D_7. Register Address Counter (RAC) is a 4-bit register which is loaded from D_0 — D_3. The status register is selected and read by a single read request. The command register is selected and loaded by a single write request. The remaining registers are accessed indirectly, and the desired register is first selected by placing its address in the RAC, and then read or written with a subsequent read or write request.

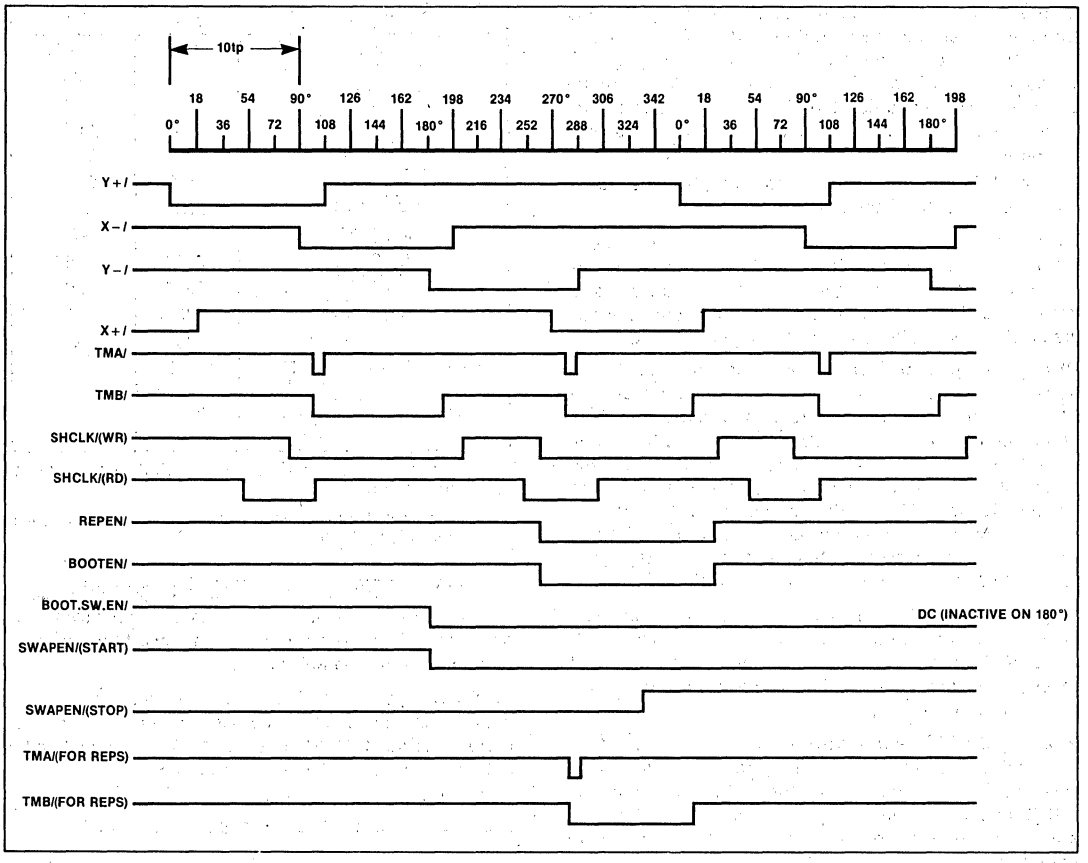


Figure 4. 7224 BMC Timing Diagram

Table 3 gives a complete listing of the address assignments for the user-accessible registers. The registers are listed in two groups. The first group (STR, CMDR, RAC) consists of those registers that are selected and accessed in one operation. The second group (UR, BLR, ER, AR, FIFO) consists of those registers that are addressed indirectly by the contents of RAC.

Table 3. Address Assignments for the User-Accessible Registers

A0 D7 D6 D5 D4 D3 D2 D1 D0	Symbol	Name of Register	Read/Write
1 0 0 C 1 C C C C	CMDR	Command Register	Write Only
1 0 0 M 0 B B B B	RAC	Register Address Counter	Write Only
1 S S S S S S S S	STR	Status Register	Read Only

Table 3. Address Assignments for the User-Accessible Registers (Continued)

RAC A0	B3	B2	B1	B0	Symbol	Name of Register	Read/Write
0	1	0	1	1	BLR LSB	Block Length Register LSB	Write Only
0	1	1	0	0	BLR MSB	Block Length Register MSB	Write Only
0	1	1	0	1	ER	Enable Register	Read or Write
0	1	1	1	0	AR LSB	Address Register LSB	Read or Write
0	1	1	1	1	AR MSB	Address Register MSB	Read or Write
0	0	0	0	0	FIFO	FIFO Data Buffer	Read or Write

SSSSSSSS = 8-bit status information returned to the user from the STR
CCCCC = 5-bit command code sent to the CMDR by the user.
BBBB = 4-bit register address sent to the RAC by the user.
B3B2B1B0 = 4-bit contents of RAC at the time the user makes a read or write request with A0 = 0.
LSB = Least Significant Byte
MSB = Most Significant Byte
M = Modifier (When written high will clear any pending interrupt from 7224 without destroying any data present in the FIFO and its associated latches.

The register file contains the registers with address 1011 through 1111. These registers are also called parametric registers because they contain flags and parameters that determine exactly how the BMC will respond to commands written to the CMDR.

To facilitate such operations, the BMC automatically increments the RAC by one count after each transfer of data to or from a parametric register.

The RAC increments from the initially loaded value through address 1111 and then on to 0000 (the FIFO address). When it has reached 0000, it no longer increments. All subsequent data transfers (with A0 = 0) will be to or from the FIFO until such time as the RAC is loaded with a different register address.

REGISTER DESCRIPTIONS

Command Register (CMDR) 4 Bits, Write Only

The user issues a command to the BMC by writing a 5-bit command code to the CMDR.

Table 4 lists the 5-bit command codes used to issue the eighteen commands recognized by the BMC.

Table 7 is a listing of the commands and their functions.

Table 4. Command Code Definitions

D5	D3	D2	D1	Do	Command Name
1	0	0	0	0	Write Bootloop Register Masked
0	0	0	0	1	Initialize
0	0	0	1	0	Read Bubble Data
0	0	0	1	1	Write Bubble Data
0	0	1	0	0	Read Seek
0	0	1	0	1	Read Bootloop Register
0	0	1	1	0	Write Bootloop Register
0	0	1	1	1	Write Bootloop
0	1	0	0	0	Read FSA Status
0	1	0	0	1	Abort
0	1	0	1	0	Write Seek
0	1	0	1	1	Read Bootloop
0	1	1	0	0	Read Corrected Data
0	1	1	0	1	Reset FIFO
0	1	1	1	0	MBM Purge
0	1	1	1	1	Software Reset
1	0	0	0	1	Zero Access Read Seek
1	0	0	1	0	Zero Access Read Bubble Data

The most commonly used commands in normal operation are:

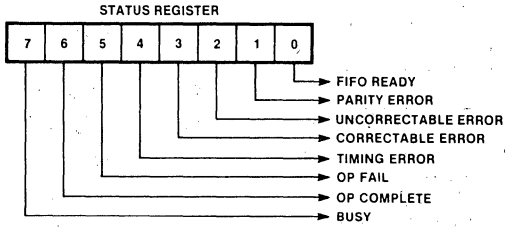
Initialize
Read Bubble Data
Write Bubble Data
Reset FIFO
Read Seek
Write Seek
Abort
Read Corrected Data
Software Reset
Read FSA Status
MBM Purge
Zero Access Read Seek
Zero Access Read Bubble Data

Commands relating to the bootloop, and used only for diagnostic purposes, are:

Read Bootloop Register
Write Bootloop Register
Write Bootloop Register Masked
Read Bootloop
Write Bootloop

Status Register (STR) 8 Bits, Read Only

The user reads the BMC status register in response to an interrupt signal, or as part of the polling process in a polled data transfer mode. The status register provides information about error conditions, completion or termination of commands, and about the BMC's readiness to transfer data or accept new commands. The individual bit descriptions are as follows:



BUSY (when = 1) indicates that the BMC is in the process of executing a command. When equal to 0, BUSY indicates that the BMC is ready to receive a new command.

OP COMPLETE (when = 1) indicates the successful completion of a command.

OP FAIL (when = 1) indicates that the BUSY bit has gone inactive with either the TIMING ERROR or UNCORRECTABLE ERROR bits active.

TIMING ERROR (when = 1) indicates that a FSA has reported a timing error to the BMC, or that the host system has failed to keep up with the BMC, thereby causing the BMC FIFO to overflow or to underflow. TIMING ERROR is also set if no bootloop sync word is found during initialization, or if a Write Bootloop command is issued when the WRITE BOOTLOOP ENABLE bit is equal to zero in the enable register, or the Write Bootloop Register Masked command is sent without an adequate number of 1's present in data pattern.

CORRECTABLE ERROR (when = 1) indicates that a FSA has reported to the BMC that a correctable error has been detected in the last data block transferred.

UNCORRECTABLE ERROR (when = 1) indicates that at least one FSA has reported to the BMC that an uncorrectable error has been detected in the last data block transferred.

PARITY ERROR (when = 1) indicates that the BMC's parity check circuitry has detected a parity error on a data byte sent to the BMC by the user on the data lines D_0-D_8.

FIFO READY has two functions. The FIFO READY functions are as follows:

NOTE: IF RAC ≠ FIFO, FIFO READY = 1

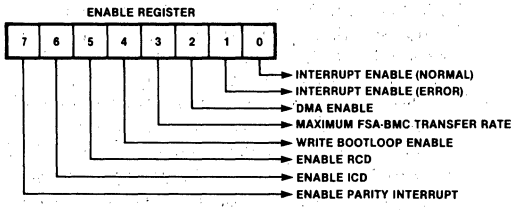
STATUS BITS		READ	WRITE
FIFO READY	BUSY		
1	1	data in FIFO	space in FIFO
0	1	no data	no space
1	0	— data in FIFO —	
0	0	— FIFO empty —	

Although the status word can be read at any time, the status information, bit 1 through 6, is not valid until the BUSY bit is low.

STR Bits 1 through 6 are reset when a new command is issued. They may also be reset by making a write request (WR=0) to the BMC with A_0=1, D_4=0, and D_5=1 (Modified bit) (that is, writing the RAC with D_5 = 1). This operation also resets the "INT" pin to "0". NOTE: A byte of FIFO data can be lost when using this procedure if the RAC is written to other than the FIFO address when data is still present in FIFO.

Enable Register (ER) 8 Bits, Write Only

The user sets various bits of the enable register to enable or disable various functions within the BMC or the FSAs. The individual bit descriptions are as follows:



In the above figure and in the text below, the following abbreviations are used:

ICD = INTERNALLY CORRECT DATA
RCD = READ CORRECTED DATA
UCE = UNCORRECTABLE ERROR
CE = CORRECTABLE ERROR
TE = TIMING ERROR

ENABLE PARITY INTERRUPT enables the BMC to interrupt the host system (via the INT line) when the BMC detects a parity error on the data bus lines D_0-D_7.

ENABLE ICD enables the BMC to give the Internally Correct Data command to the FSAs when an error has been detected by the FSA's error detection and correction circuitry. Each FSA responds to such a command by internally cycling the data through its error correction network. When finished, the FSA returns status to the BMC as to whether or not the error is correctable. The value of ENABLE ICD affects the action of INTERRUPT ENABLE (ERROR).

ENABLE RCD enables the BMC to give the Read Corrected Data command to the FSAs when an error has been detected. This causes each FSA to correct the error (if possible) and also to transfer the corrected data to the BMC. The Read Corrected Data command is also used to read into the BMC data previously corrected by the FSA in response to an Internally Correct Data command. In either case, when the data transfer has been completed, the BMC reads each FSA's status to determine whether or not the error was correctable. In the case of an uncorrectable error, bad data may have been sent to the user. The value of ENABLE RCD affects the action of INTERRUPT ENABLE (ERROR).

WRITE BOOTLOOP ENABLE (when = 1) enables the bootloop to be written. If this bit is equal to zero, and a Write Bootloop command is received by the BMC, the command is aborted and the TIMING ERROR bit is set in the STR.

DMA ENABLE (when = 1) enables the BMC to operate in DMA data transfer mode, using the DRQ and $\overline{\text{DACK}}$ signals in interaction with a DMA controller. When equal to zero, DMA ENABLE sets up the controller to support interrupt driven or polled data transfer.

INTERRUPT ENABLE (ERROR) selects error conditions under which the BMC stops command execution and interrupts the host processor (via the INT line). INTERRUPT ENABLE (ERROR) operates in conjunction with ENABLE ICD and ENABLE RCD.

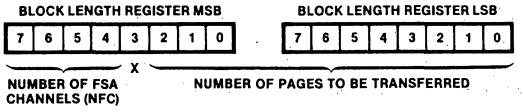
Enable ICD	Enable RCD	Interrupt Enable (ERROR)	Interrupt Action
0	0	1	Interrupt on TE only
0	1	0	Interrupt on UCE or TE
0	1	1	Interrupt on UCE, CE, or TE
1	0	0	Interrupt on UCE or TE
1	0	1	Interrupt on UCE, CE, or TE
1	1	0	Not used
1	1	1	Not used

TE = Timing Error, CE = Correctable Error,
UCE = Uncorrectable Error.

INTERRUPT ENABLE (NORMAL) (when = 1) enables the BMC to interrupt the host system (via the INT line), when a command execution has been successfully completed (OP COMPLETE = 1 in the STR).

Block Length Register (BLR) 16 Bits, Write Only

The contents of the block length register determine the system page size and also the number of pages to be transferred in response to a single bubble data read or write command. The bit configuration is as follows:

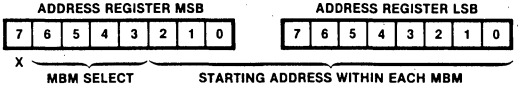


The system page size is proportional to the number of magnetic bubble memory modules (MBMs) operating in parallel during the data read or write operation. Each MBM requires two FSA channels. Bits 4 through 7 of BLR MSB actually specify the number of FSA channels to be accessed.

The BLR LSB, together with the 3 least significant bits of the BLR MSB, specify the number of pages to be transferred. Up to 2048 pages can be transferred in response to a single bubble data read or write command, hence the requirement for 11 bits. All 11 bits equal to zero specifies a 2048 page transfer.

Address Register (AR) 16 Bits, Read or Write

The contents of the address register determine which MBM group is to be accessed, and, within that group, what starting address location shall be used in a data read or write operation. The bit configuration is as follows:



Within each MBM there are 8192 possible starting address locations for a data read or write operation, hence the requirement for 13 bits in the starting address.

The selection of the MBMs to be read or written is specified by AR MSB Bits 57. The BMCs interpretation of these bits depends on the number of MBMs in a group, which is specified by BLR MSB Bits 4-7.

Table 6 shows which MBM groups are selected in response to given values for BLR MSB Bits 4-7 and AR MSB Bits 3-6. A 4-megabyte system (8 MBMs) is represented, with the FSA channels numbered 0 through F:

Table 6. Selection of FSA Channels

AR MSB Bits (7,6,5)	BLR MSB Bits (7,6,5,4)				
	0000	0001	0010	0100	1000
000	0	0,1	0,1,2,3	0 to 7	0 to 7
001	1	2,3	4,5,6,7		
010	2	4,5			
011	3	6,7			
100	4				
101	5				
110	6				
111	7				

Table 5. 4Mbyte System Page Size, Page Address Range, and Data Transfer Performance Configuration

BLR MSB 7 6 5 4	NFC	System Page Size	# of Pages	Address Range	MBM Data Transfer Rate
0 0 0 1	2	64 byte	64K	0000 — FFFF	25K bytes/sec
0 0 1 0	4	128 byte	32K	0000 — 7FFF	50K bytes/sec
0 1 0 0	8	256 byte	16K	0000 — 3FFF	100K bytes/sec
1 0 0 0	16	512 byte	8K	0000 — 1FFF	200K bytes/sec

FIFO Data Buffer (FIFO) 40 x 8 Bits, Read or Write

The BMC FIFO is a 40-byte buffer through which data passes on its way from the FSAs to the user, or from the user to the FSAs. The FIFO allows the data transfer to proceed in an asynchronous and flexible manner, and relaxes timing constraints, both to the FSAs and also to the user's equipment. The user's system must, however, meet the data rate requirements. When the BMC is busy (executing a command) the FIFO functions as a data buffer. When the BMC is not busy, the FIFO is available to the user as a general purpose FIFO.

FUNCTIONAL OPERATION

The IC components used in the bubble memory systems have been designed with transparency in mind—that is, a maximum number of operations are handled by the hardware and firmware of these components.

Each four Megabit Bubble Memory (MBM) operates in its own domain, and is unaffected by the number of bubble memories in the system. The roles played by the MBM's immediate support circuitry can be described as if the system contained only one MBM module.

Data Flow Within the Magnetic Bubble Memory (MBM) System (Single MBM Systems)

During a read operation, data flows as follows: The data from the MBM is input to the Formatter/Sense Amplifier (FSA). Data from each channel (A channel or B channel) of the MBM goes to the corresponding channel of the FSA. In the FSA, the data is paired up with the corresponding bit in the FSA's bootloop register to determine whether it represents data from a 'good' loop. If it does, the data bit is stored in the FSA FIFO. Error detection and correction is applied to each block of 256 data bits.

From the FSA FIFO, data is sent to the bubble memory controller (BMC) in the form of a serial bit stream, via a one-line bidirectional data bus (DIO). The data is multiplexed onto the DIO line, with data bits coming alternately from the A and B channels of the FSA. The BMC outputs a $\overline{\text{SYNC}}$ pulse to the $\overline{\text{SELECT.IN}}$ input of the FSA. The FSA responds by placing a data bit from the A channel FIFO on the DIO line. One clock cycle later, a data bit from the B channel FIFO is placed on the DIO line. The BMC continues to output $\overline{\text{SYNC}}$ pulses, once every 20 or 80 clock cycles, each time receiving two data bits in return.

In the BMC, the data undergoes serial-to-parallel conversion, and is assembled into bytes, which are then placed in the BMC FIFO, which can hold 40 bytes of data. From this FIFO, the data bytes are written onto the user interface.

During a write operation, the data flow consists of the corresponding operations in the reverse order.

INTERFACING REQUIREMENTS

All communications between the host microprocessor, and the bubble memory is performed through the 7224 BMC. Below the general principles are described, for detailed guidelines please refer to the BPK 75 Manual.

First the hardware interfacing requirements and second the software interfacing requirements are described.

Table 7. Detailed Command Descriptions

Initialize	The BMC executes the Initialize command by first interrogating the bubble system to determine how many FSAs are present, then reading and decoding the bootloop from each MBM and storing the results in the corresponding FSA's bootloop register. All the parametric registers must be properly set up before issuing the Initialize command.
Read Bubble Data	The Read Bubble Data command causes data to be read from the MBMs into the BMC FIFO. The selection of the MBMs to be accessed and the starting address for the read operation is specified in the address register (AR). The block length register (BLR) specifies the number of system pages to be read. All the parametric registers must be properly set up before issuing the Read Bubble Data command.
Write Bubble Data	The Write Bubble Data command causes data to be read from the BMC FIFO and written into the MBMs. The selection of the MBMs to be accessed and the starting address for the write operation is specified in the address register (AR). The block length register (BLR) specifies the number of system pages to be written. All the parametric registers must be properly set up before issuing the Write Bubble Data command.
Read Seek	The Read Seek command rotates the selected MBMs to a designated page address location. No data transfer occurs. The positioning is such that the next data location available to be read is the specified (in AR) page address plus one. The Read Seek command may be used to reduce latency (access time) in cases where information is available for the user to predict the location of an impending read reference to the MBMs.

Table 7. Detailed Command Descriptions (Continued)

Write Seek	The Write Seek command rotates the selected MBMs to a designated page address location. No data transfer occurs. The positioning is such that the next data location available to be written is the specified (in AR) page address plus one. The Write Seek command may be used to reduce latency (access time) in cases where information is available for the user to predict the location of an impending write reference to the MBMs.
Abort	The Abort command causes a controlled termination of the command currently being executed by the BMC. The Abort command will be accepted by the BMC (and is typically issued) when the BMC is busy.
MBM Purge	The MBM Purge command clears all BMC registers, counters, and the MBM address RAM. Furthermore, it determines how many FSA channels are present in the system and stores this value in the 7224. The "INITIALIZE" command uses this command as a subroutine.
Read Corrected Data	The Read Corrected Data command causes the BMC to read into the BMC FIFO a 256-bit block of data from the FIFO of each selected FSA channel, after an error has been detected. The data cycles through the error correction network of the FSA. After the data has been read, the FSA reports to the BMC whether or not the error was correctable. The Read Corrected Data command is used only when the system is in error correction mode (ENABLE ICD or ENABLE RCD set in the ER).
Software Reset	The Software Reset command clears the BMC FIFO and all registers, except those containing initialization parameters. It also causes the BMC to send the Software Reset command to selected FSAs in the system. No reinitialization is needed after this command.
Read FSA Status	The Read FSA Status command causes the BMC to read the 8-bit status register of all FSAs, and to store this information in the BMC FIFO. The Read FSA Status command is independent all parametric registers.
Read Bootloop Register	The Read Bootloop Register command causes the BMC to read the bootloop register of the selected FSA channels and to store this information in the BMC FIFO. Twenty bytes are transferred for each FSA channel selected.
Write Bootloop Register Masked	Proper operation of the FSAs during data transfer to or from the MBMs requires that the bootloop register contain exactly 270 logic 1s for each FSA bootloop register. The user may select any subset of 270 "good" loops from the total number of available loops. As an alternative, the Write Bootloop Register Masked command may be used. This command counts the number of logic 1s and masks out the remaining 1s after the proper count has been reached. The Initialize command uses this command as a subroutine.
Read Bootloop	The Read Bootloop command causes the BMC to read the bootloop from the selected MBM, and to store the decoded bootloop information in the BMC FIFO. The Initialize command uses this command as a subroutine.
Write Bootloop	The Write Bootloop command causes the existing contents of the selected MBM's bootloop to be replaced by new bootloop data based on 40 bytes of information stored in the FIFO (the user must actually write 41 bytes, where the 41st byte is all 0s). Encoding of the bootloop data is done by the BMC hardware.
Zero Access Read Bubble Data	The Zero Access Read Bubble Data command functions exactly the same as the Read Bubble Data command except it must be preceeded by the Zero Access Read Seek command. The parametric registers are written prior to the Zero Access Read Seek command and should not be rewritten for the Zero Access Read Bubble Data command.
Zero Access Read Seek	The Zero Access Read Seek command operates similiarly to the Read Seek command, but it reads the first page of data from the MBM into the FSA(s) FIFO. This eliminates the first page overhead involved in the Read Bubble Data command. The latency for the first page data is only the time required to read the data from the 7244(s). The values written into the parametric registers prior to the issuance of the Zero Access Read Seek command are identical to those written for a Read Bubble Data command. The seeking address for the Zero Access Read Seek command is equal to the desired read address. The Zero Access Read Seek command increments the Ad and decrements the BLR. Since the Zero Access Read Bubble Data command expects this increment and decrement, it is used for this data transfer instead of the Read Bubble Data command.

HARDWARE INTERFACE REQUIREMENTS

User Interface Signals

The source, destination and function of the user interface signals are described in Table 1 in the data sheet.

System Timing

As shown on the timing diagrams in the WAVEFORM section the typical read/write cycle timing provides sufficient tolerance to allow most currently available microprocessors to be easily adapted to the BMC timing requirements.

User Data Transfer Rate Requirements

The maximum data rate for the user interface is a function of the number of MBMs operated in parallel as outlined in table 8. The rates listed must be considered in relation to the data transfer mode (polled, interrupt-driven, or DMA) to be implemented in order to be sure that the host system software and hardware are capable of keeping up with the data transfer. In other words, the BMC requires the host CPU to be able to sustain the maximum data rate transfer rate for the minimum data transfer (e.g., for a one bubble system keep up the transfer rate for at least 64 bytes = one page).

Table 8. User Data User Transfer Rate Requirements

Number of MBMs Operating in Parallel	Maximum Data Transfer Rate Between BMC FIFO and the FSAs during Write Bubble Data Commands
1	12.5 kbytes/second
2	25 kbytes/second
4	50 kbytes/second
8	100 kbytes/second

Hardware Interfacing for Data Transfer

The BMC supports three data transfer modes, i.e. DMA, interrupt-driven and polled.

To support DMA, a hardware mechanism is required for servicing the BMC's data transfer requests. While several hardware implementations are possible, one common configuration is the Intel 8257 DMA controller.

To support an interrupt-driven system an Intel 8259 Programmable Interrupt Controller is often used.

The polled data transfer mode relies almost exclusively on the software interaction between the host processor and the BMC to control the transfer of data.

Multiple MBM-System

A BMC is capable of processing data and of supplying the required timing and control signals for operating up to eight Bubble Storage Units (BSUs), each of which is capable of storing 512 kbytes of user data. A BSU consists of a 512 kbyte MBM and its five immediate IC support chips (i.e. a BPK 74 Kit).

SOFTWARE INTERFACE REQUIREMENTS

To use the BMC, the user has to write a "bubble memory software driver".

The bubble driver is responsible for all the system interaction with the bubble memory controller and is intrinsic to the efficient and reliable operation of the bubble system. The driver accepts bubble memory commands and command execution parameters from the application program, controls and monitors command execution, and returns operational status information to the application program at command completion. To perform all of these operations, the bubble driver must support the bit/byte level of the bubble memory controller's command and status registers and the parametric registers that define the operating mode, system configuration, and extent of the transfer.

The level of the software driver complexity is a function of the specific application needs. Regardless, a set of basic drivers must be developed that in turn are integrated into a system at the appropriate level. If an application program is small and simple, a basic bubble driver may simply be called from the main program.

At the highest level of driver sophistication, the application program treats the bubble system as a collection of named data areas of files similar to the way in which data is stored and retrieved in disk operating systems. At the file system level, an application program can ignore the mechanics of bubble storage and access and merely present a file name to the driver to open, read, or write, then close the desired bubble file.

Data Organization

From a software viewpoint, data logically is organized into blocks of bytes called pages. During data transfer operations, one or more of these pages are transferred between the bubble(s) and the host microprocessor. A page is the smallest increment of data that can be transferred; single bytes cannot be transferred. Conceptually, the data organization within a bubble memory is analogous to a disk system. Just as disk sector sizes are fixed when a disk is formatted, bubble page sizes are established, under software control when the bubble system is initialized.

For single bubble system, the page size is fixed at 64 bytes with error correction. The total number of pages available is 8092. In systems with multiple bubbles, page size can vary from 64 bytes to 512 bytes depending on the number of bubble devices in the system. Page size is directly proportional to the system data rate and also determines the total number of available pages (address field size). The selection of the appropriate page size depends primarily on the data rate supported by the system. The higher the data rate, the faster the microprocessor must respond to the demands of the bubble memory controller.

Buffering

The bubble memory controller includes a FIFO data buffer that, although only 40 bytes long, reconciles timing differences between the parallel data transfer to or from the host microprocessor and the serial data transfer to or from the Bubble Memory Subsystem. Accordingly, when an application program requests data from a bubble, the software driver is responsible for keeping up with the FIFO for the duration of the data transfer in order to prevent the FIFO from overflowing or underflowing.

Command Execution

Command execution can be performed either in an interrupt driven mode or in a polled mode irrespective of the data transfer mode (polled, interrupt-driven, or DMA).

Data Transfer Mode

As described earlier in the hardware section, three data transfer modes are available (polled, interrupt-driven or DMA).

System performance, additional hardware and software overhead are all important considerations when choosing the appropriate mode for your application.

Error Correction

The bubble memory has a built-in error detection. Three levels of error correction are available.

Communication with the BMC

All communications between the host and the bubble memory actually are performed through the BMC. The BMC has two input/output (I/O) ports, an eight-bit bidirectional data port, and an eight-bit command/status port. Conceptually, a bubble memory system can be thought of as a disk system in that data in the bubble memory is organized into blocks called pages in bubble technology that are similar to disk sectors. Information such as starting page location, direction of transfer, and the number of pages to be transferred is passed to the BMC before the desired read or write operation is initiated.

The general procedure for communicating with the BMC is:

Set-up the BMC for data transfer communication by loading specific parameters in user-accessible registers.

Send the desired command.

Read the status register to determine if command is accepted.

If applicable, transfer (i.e., read or write) data.

Read the status register until BMC is not busy (or under some conditions "INT" pin).

Examine the status register to determine whether the operation was successful.

For all details and exceptions to this general description consult the BPK 75 User's Manual. (Available 1st Quarter 1984).

ABSOLUTE MAXIMUM RATINGS

Temperature under bias ..... −40 to +100°C
Storage Temperature ..... −65°C to +150°C
All Input or Output Voltages and
V_{CC} Supply Voltage ..... −0.5V to 7V

**NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

(T_A = 0 to 70°C, V_{CC} = 5.0V +5%, −10%)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
V_{IL}	Input Low Voltage		0.8	V	
$V_{IH(1)}$	Input High Voltage (all but $\overline{PWR.FAIL}$)	2.0	V_{CC}+0.5V	V	
$V_{IH(2)}$	Input High Voltage ($\overline{PWR.FAIL}$)	2.5	V_{CC}+0.5V	V	
$V_{OL(1)}$	Output Low Voltage (All outputs except $\overline{DET.ON}$, $\overline{BUS.RD}$, $\overline{SHIFT.CLK}$, and $\overline{SYNC}$		.45	V	I_{OL} = 3.2 mA
$V_{OL(2)}$	Output Low Voltage $\overline{DET.ON}$, $\overline{BUS.RD}$, $\overline{SHIFT.CLK}$, $\overline{SYNC}$		.45	V	I_{OL} = 1.6 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = 400μA
I_{IN}	Input Leakage Current		10	μA	$0 \leq V_{IN} \leq V_{CC}$
$\lvert I_{OFL} \rvert$	Output Float Leakage		10	μA	$0.45 \leq V_{OUT} \leq V_{CC}$
I_{CC}	Power Supply Current from V_{CC}		200	mA	

A.C. CHARACTERISTICS

(T_A = 0 to 70°C, V_{CC} = 5.0V + 5%, −10%; C_L = 150pF; unless otherwise noted.)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_P	Clock Period	499.5	500.5	ns	
t_ϕ	Clock Phase Width (High Time)	.45 t_P	.55 t_P		
t_R-t_F	Input Signal Rise and Fall Time		30	ns	

FSA INTERFACE TIMINGS (under pin loading)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{CDV}	CLK to DIO Valid Delay		200	ns	Under Pin Loads*
t_{CDF}	CLK to DIO Entering Float		250	ns	Under Pin Loads*
t_{CDE}	CLK to DIO Enabled from Float		200	ns	Under Pin Loads*
t_{CDH}	CLK to DIO Hold Time	0		ns	Under Pin Loads*
t_{CSOL}	CLK to $\overline{SYNC}$ Leading Edge Delay		200	ns	Under Pin Loads*
t_{CSOT}	CLK to $\overline{SYNC}$ Trailing Edge Delay		150	ns	Under Pin Loads*
t_{DC}	DIO Setup Time to Clock	150		ns	Under Pin Loads*
t_{DHC}	DIO Hold Time from Clock	0		ns	Under Pin Loads*
t_{COL}	CLK to Output Leading Edge		200	ns	Under Pin Loads*
t_{COT}	CLK to Output Trailing Edge		200	ns	Under Pin Loads*
t_{EW}	$\overline{ERR.FLG}$ Pulse Width	200		ns	Under Pin Loads*
t_{SCFT}	$\overline{SHIFTCLK}$ to $\overline{Y-}$ Trailing Edge	80	200	ns	Under Pin Loads*

A.C. CHARACTERISTICS (Continued) (T_A = 0 to 70°C, V_{CC} = 5.0 + 5%, −10%; C_L = 150 pF; unless otherwise noted.)

READ CYCLE (HOST INTERFACE)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{AC}	Select Setup to $\overline{RD}\downarrow$	0		ns	
t_{CA}	Select Hold from $\overline{RD}\uparrow$	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	200		ns	
t_{AD}	Data Delay from Address		200	ns	
t_{RD}	Data Delay from $\overline{RD}\downarrow$		200	ns	
t_{DF}	Output Float Delay	10	100	ns	
t_{DC}	DACK Setup to $\overline{RD}\downarrow$	0		ns	
t_{CD}	DACK Hold from $\overline{RD}\uparrow$	0		ns	
t_{KD}	Data Delay from $\overline{DACK}\downarrow$		200	ns	
t_{CYCR}	"Read" Cycle Time	(DMA Mode) $4t_p$		ns	In non DMA mode. t_{CYCR} Min. = $6t_p$

WRITE CYCLE (HOST INTERFACE)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{AC}	Select Setup to $\overline{WR}\downarrow$	0		ns	
t_{CA}	Select Hold from $\overline{WR}\uparrow$	0		ns	
t_{WW}	$\overline{WR}$ Pulse Width	200		ns	
t_{DW}	Data Setup to $\overline{WR}\uparrow$	200		ns	
t_{WD}	Data Hold from $\overline{WR}\uparrow$	0		ns	
t_{DC}	$\overline{DACK}$ Setup to $\overline{WR}\downarrow$	0		ns	
t_{CD}	$\overline{DACK}$ Hold from $\overline{WR}\uparrow$	0		ns	
t_{CYCW}	"Write" Cycle Time	$4t_p$			
t_{CQ}	Request Hold from $\overline{RD}$ or $\overline{WR}$ (Non-Burst Mode)		200	ns	
t_{DEADW}	Inactive Time between $\overline{WR}\uparrow$ and $\overline{WR}\downarrow$	$4t_p$		ns	
t_{DEADR}	Inactive Time between $\overline{RD}\uparrow$ and $\overline{RD}\downarrow$	150			

7250-7234 INTERFACE TIMINGS

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{CBL}	CLK to Bubble Signal Leading Edge		275	ns	Under Pin Loads*
t_{CBT}	CLK to Bubble Signal Trailing Edge		275	ns	Under Pin Loads*

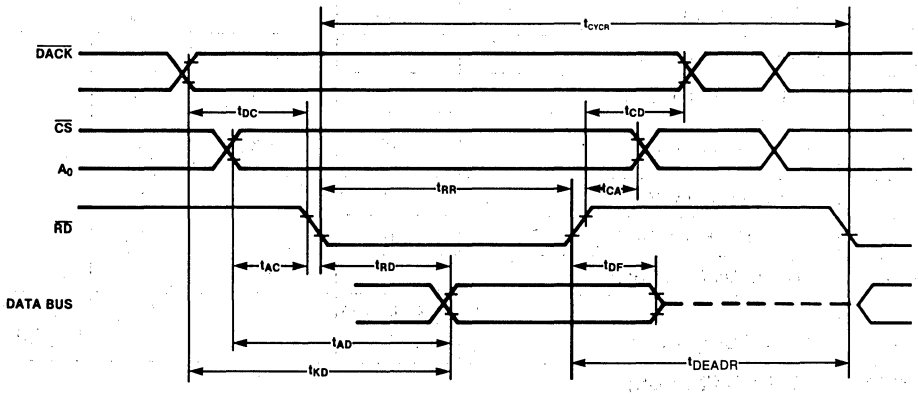
*Bubble Pin Loads Shown Below

PIN LOADINGS

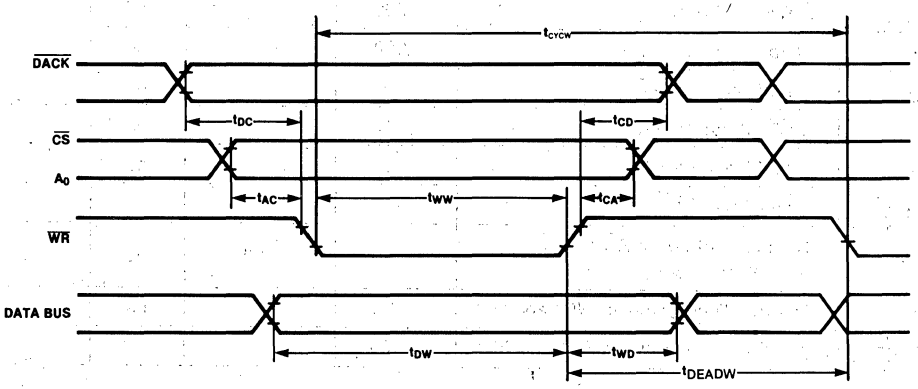
Pin Names	Value	Unit
$\overline{X+}$, $\overline{X-}$, $\overline{Y+}$, $\overline{Y-}$	150	pF
$\overline{TM.A}$, $\overline{TM.B}$, $\overline{REP.EN}$, $\overline{BOOT.EN}$, $\overline{SWAP.EN}$, $\overline{BOOT.SW.EN}$, $\overline{C/D}$, $\overline{ERR.FLG}$, $\overline{WAIT}$, $\overline{SYNC}$, DIO	100	pF
$\overline{DET.ON}$ & $\overline{SHIFT.CLK}$		
$\overline{BUS.READ}$	10	pF

WAVEFORMS

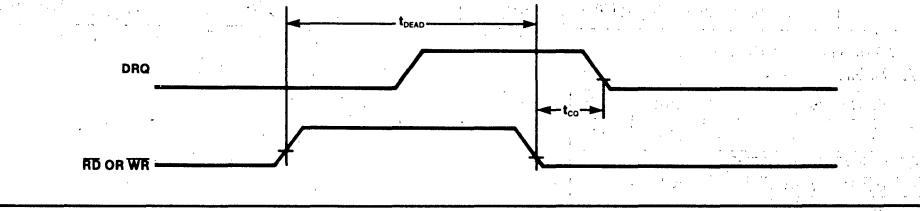
READ (HOST INTERFACE)



WRITE (HOST INTERFACE)

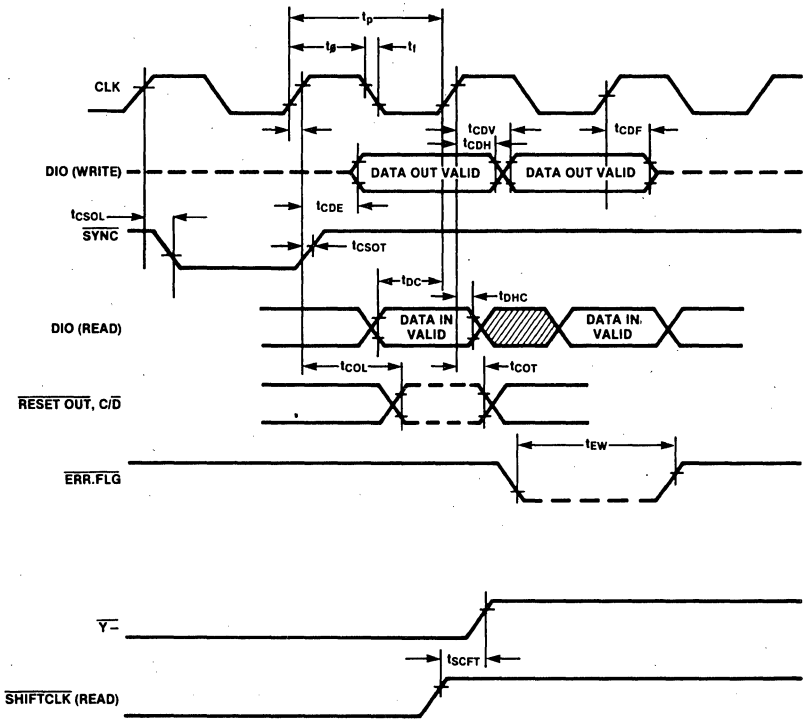


DMA (HOST INTERFACE)

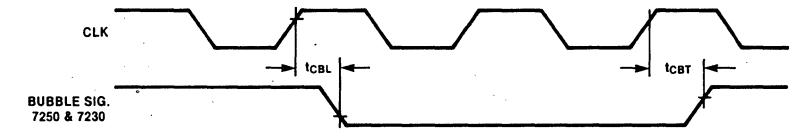


WAVEFORMS (Continued)

7244 INTERFACE TIMINGS



7250 & 7234 INTERFACE TIMINGS



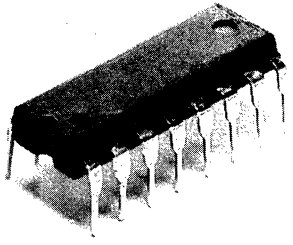
General Information 7

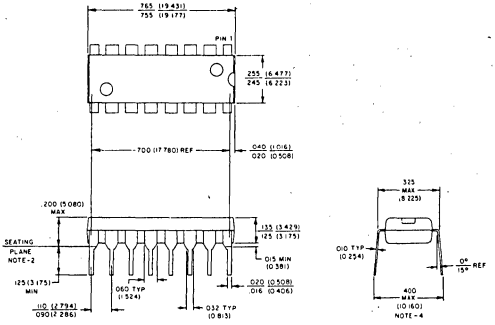
NOTES:

1. All packages drawings not to scale.
2. All packages seating plane defined by .0415 to .0430 PCB holes.
3. Type P packages only. Package length does not include end flash burr. Burr is .005 nominal, can be .010 max. at one end.
4. All package drawings end view dimensions are to outside of leads.

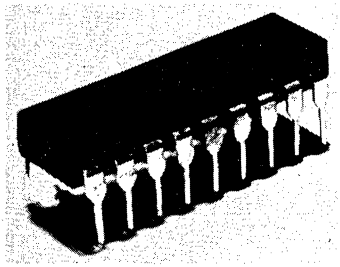
PLASTIC DUAL IN-LINE PACKAGE TYPE P

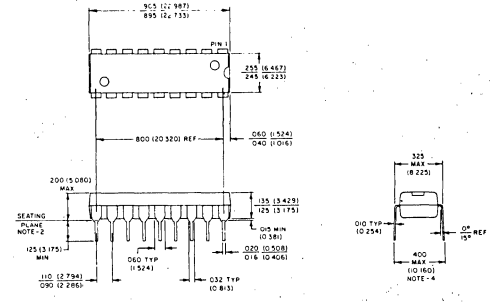
16-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



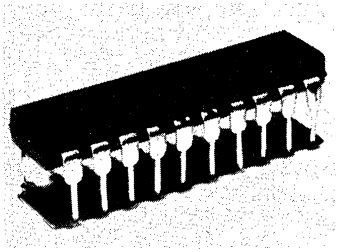


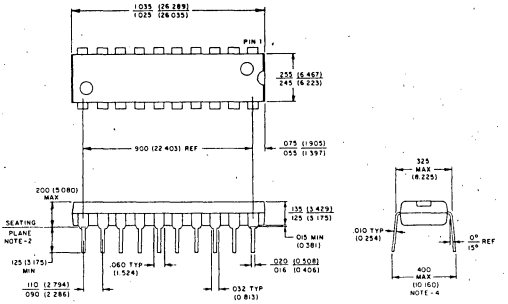
18-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



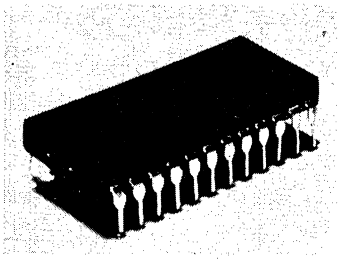


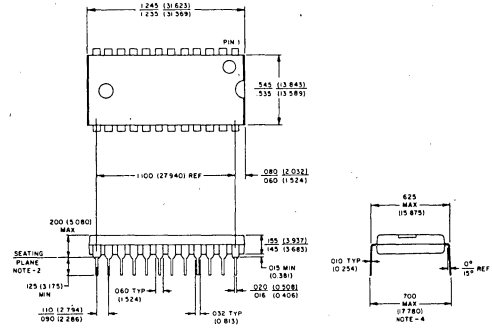
20-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P





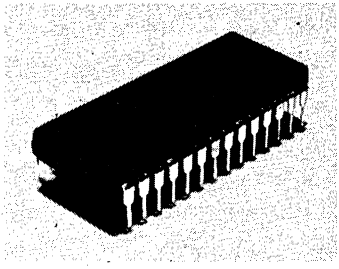
24-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P

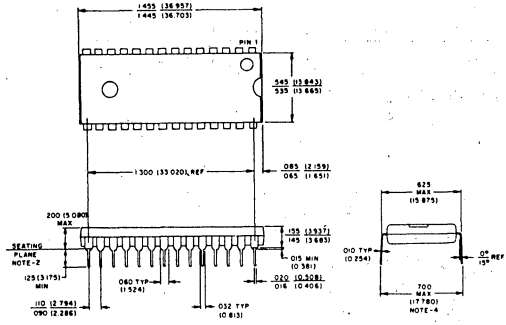




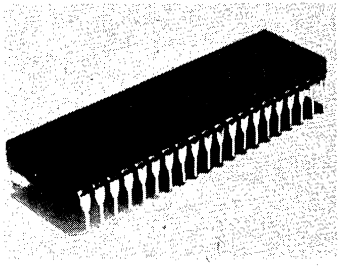
PLASTIC DUAL IN-LINE PACKAGE TYPE P

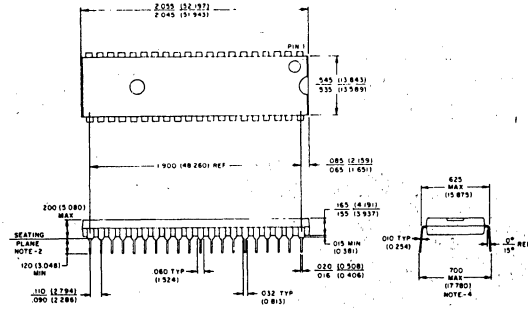
28-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P





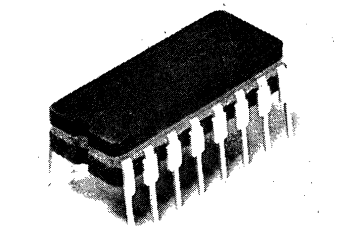
40-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P

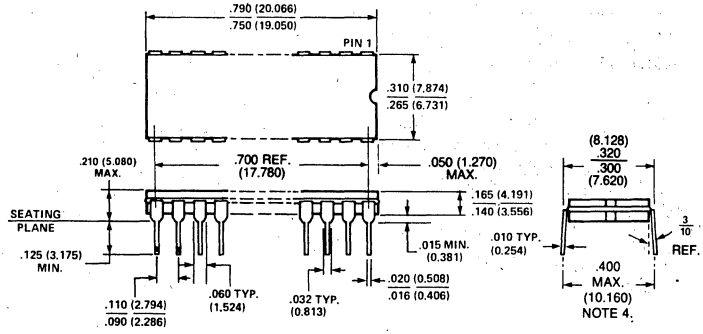




CERAMIC DUAL IN-LINE PACKAGE TYPE D

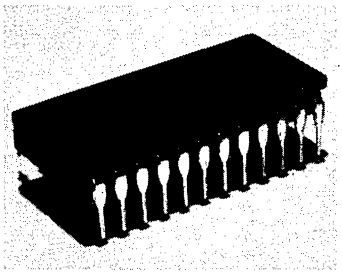
16-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D

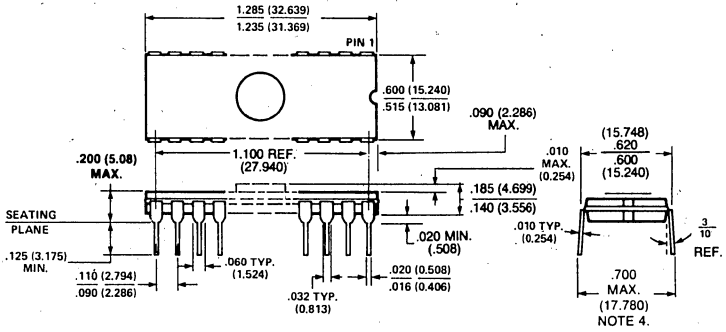




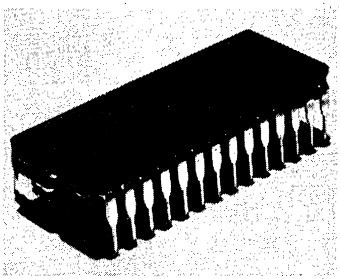
CERAMIC DUAL IN-LINE PACKAGE TYPE D

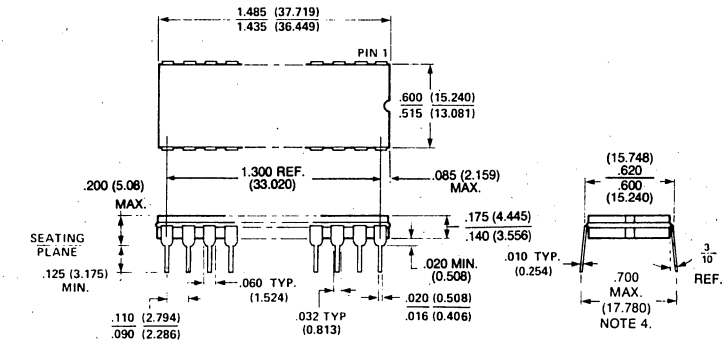
24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



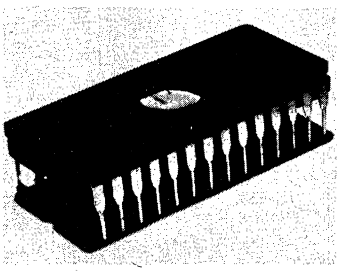


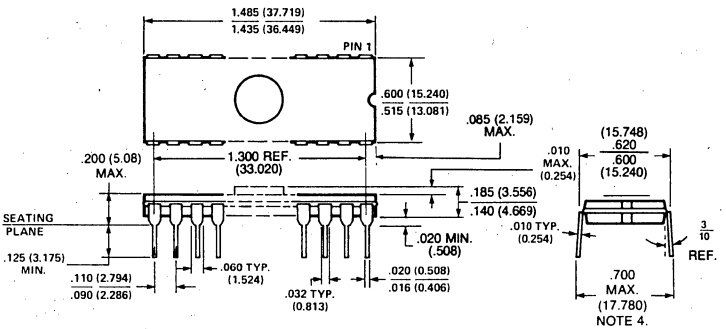
28-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



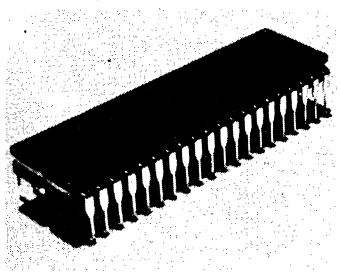


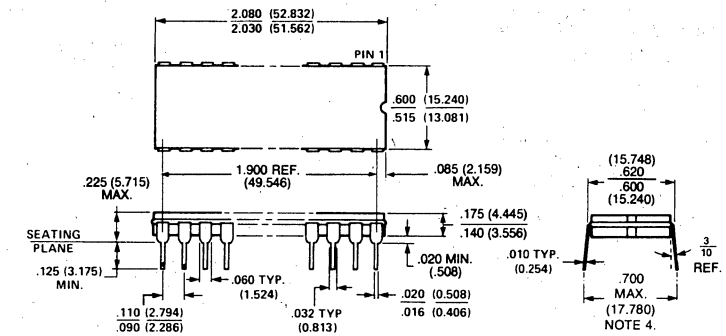
28-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D





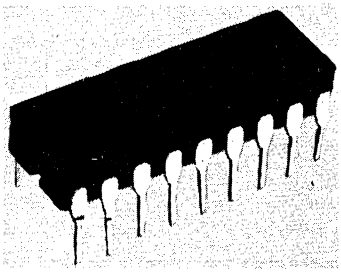
40-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D

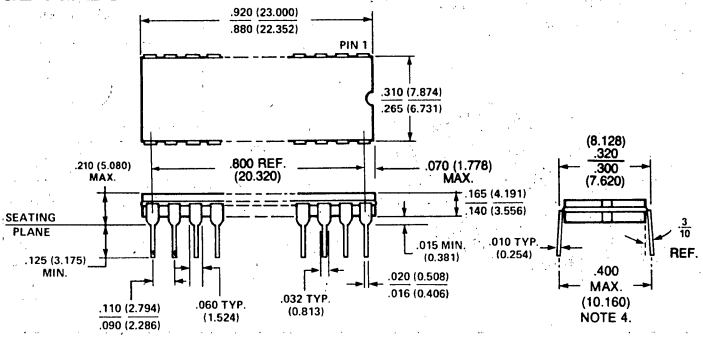




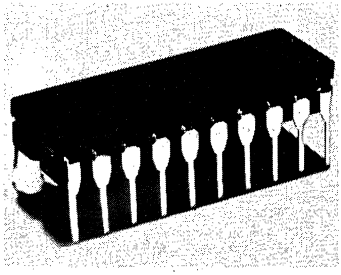
CERAMIC DUAL IN-LINE PACKAGE TYPE D

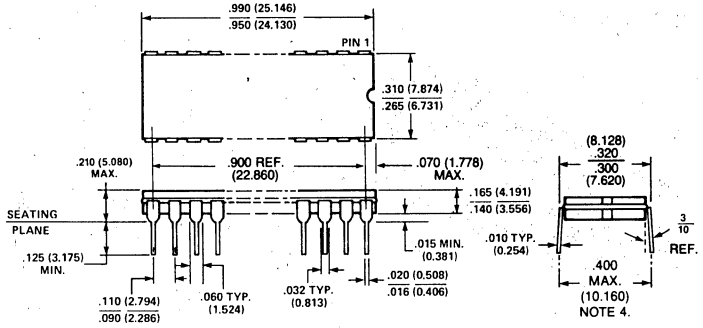
18-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



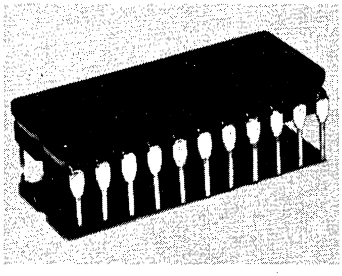


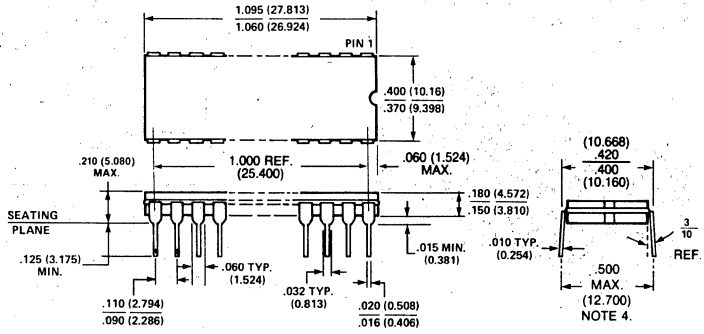
20-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



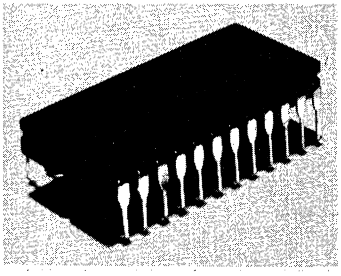


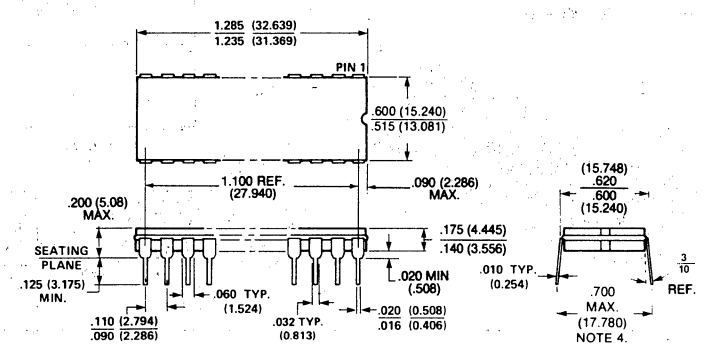
22-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D





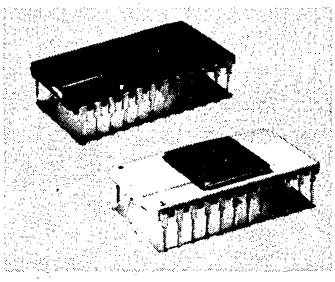
24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D

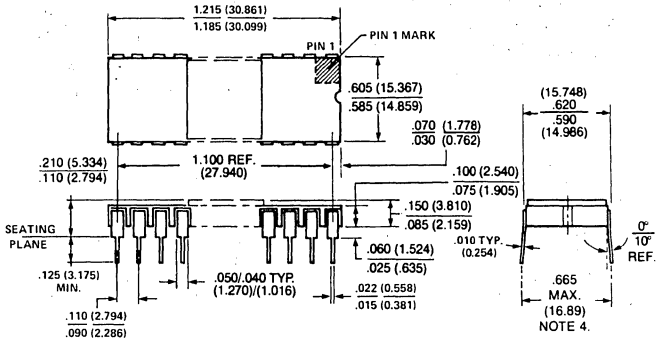




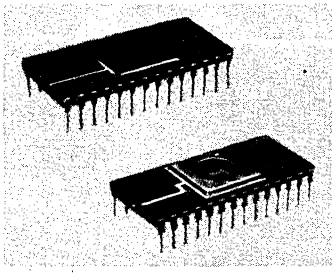
CERAMIC DUAL IN-LINE PACKAGE TYPE C

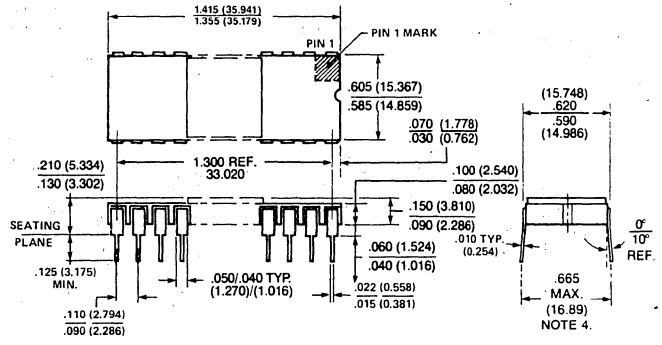
24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C





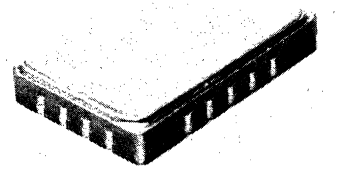
28-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C

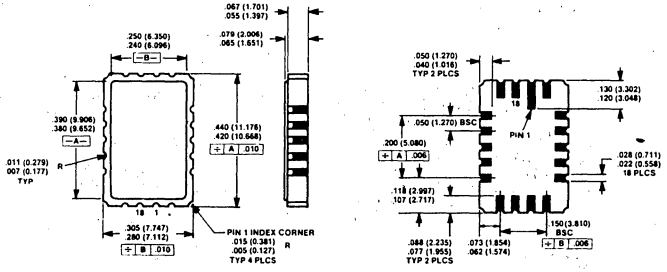




CERAMIC LEADLESS CHIP CARRIERS

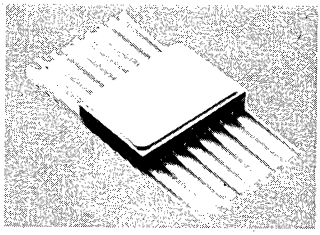
18-LEAD CERAMIC LEADLESS PACKAGE TYPE F

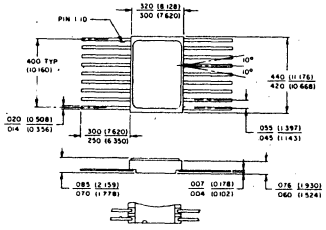




CERAMIC FLAT PACK PACKAGE TYPE CF

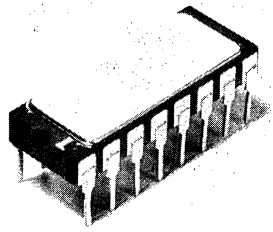
18-LEAD HERMETIC FLAT PACK PACKAGE TYPE CF

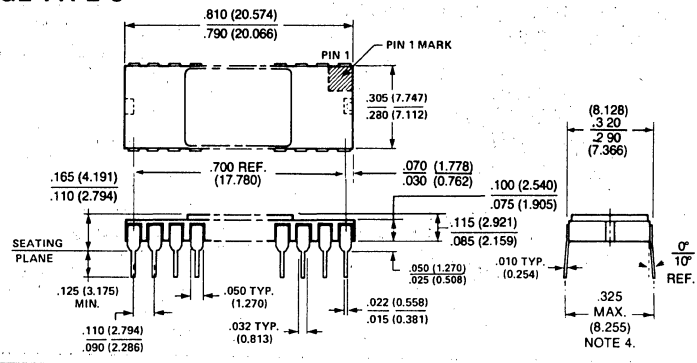




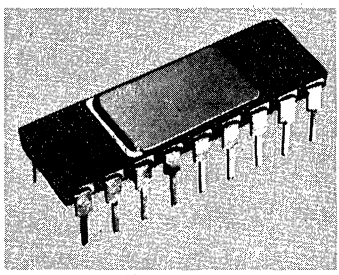
CERAMIC DUAL IN-LINE PACKAGE TYPE C

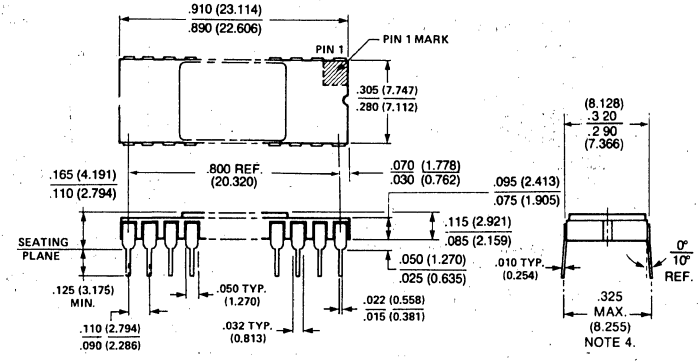
16-LEAD CERAMIC DUAL IN-LINE PACKAGE TYPE C





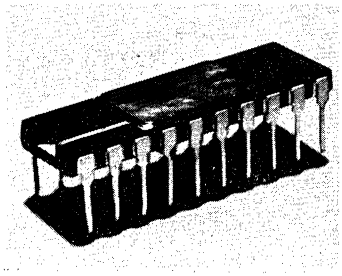
18-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C

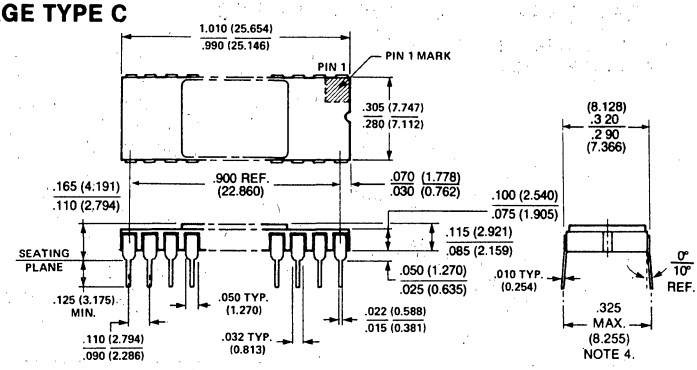




CERAMIC DUAL IN-LINE PACKAGE TYPE C

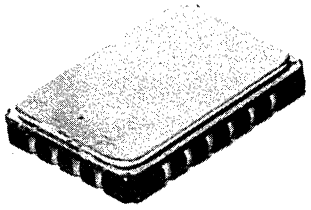
20-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C

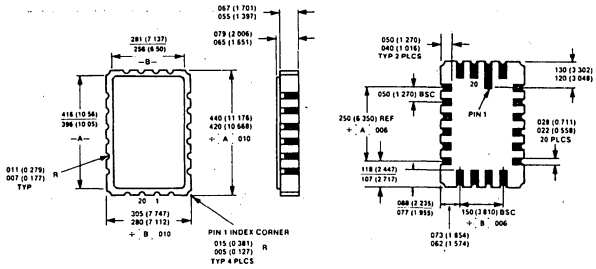




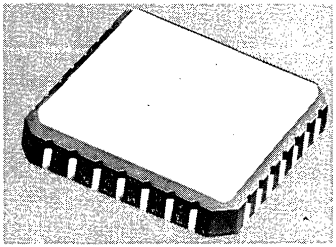
CERAMIC LEADLESS CHIP CARRIER

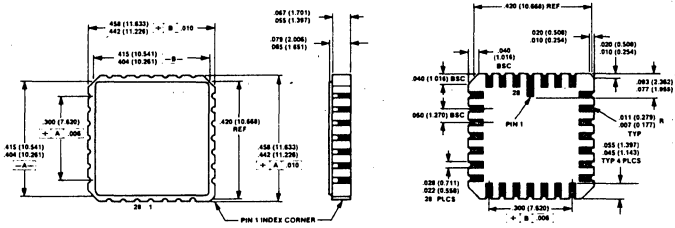
20-LEADLESS HERMETIC CHIP CARRIER
JEDEC PACKAGE TYPE F



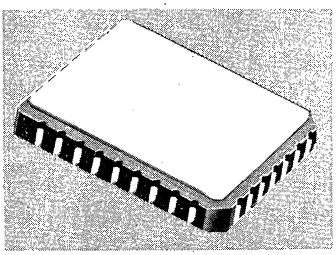


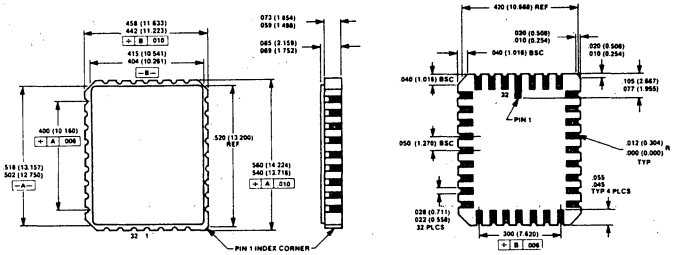
28-LEAD CERAMIC LEADLESS
PACKAGE TYPE C



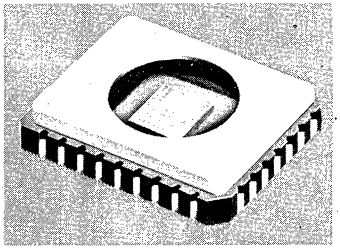


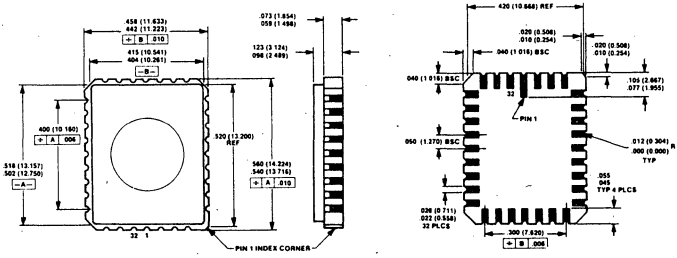
32-LEAD CERAMIC LEADLESS
PACKAGE TYPE E



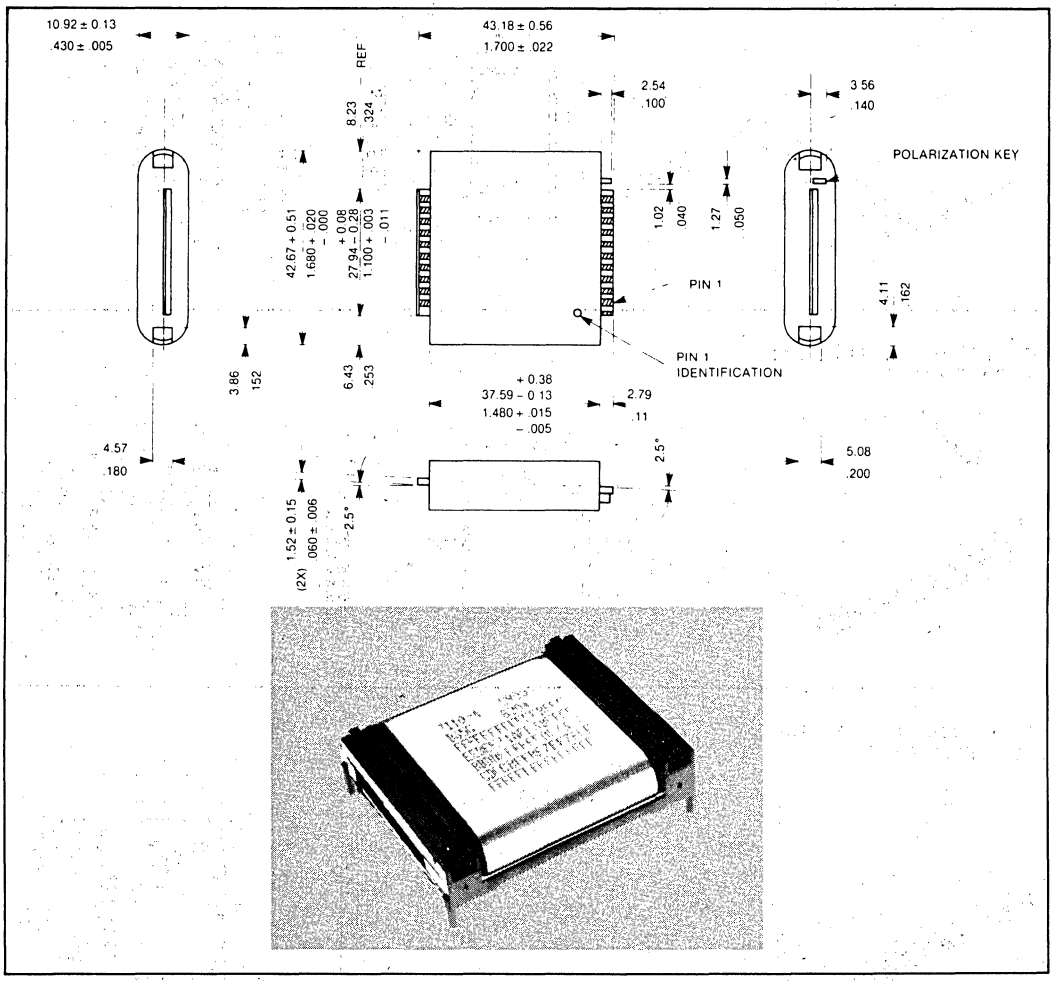


32-LEAD CERAMIC LEADLESS
PACKAGE TYPE E



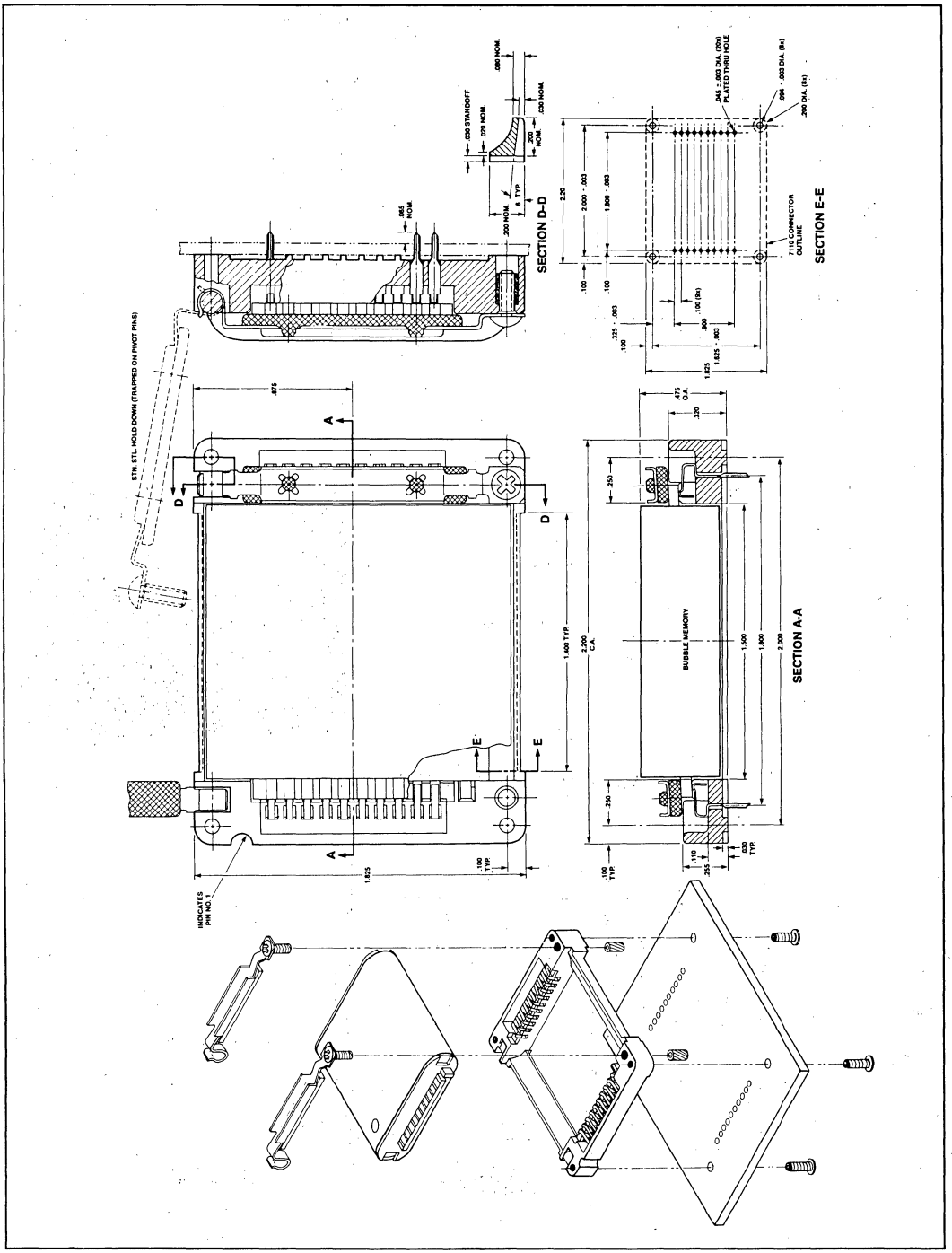


BUBBLE LEAD-LESS PACKAGE AND SOCKET



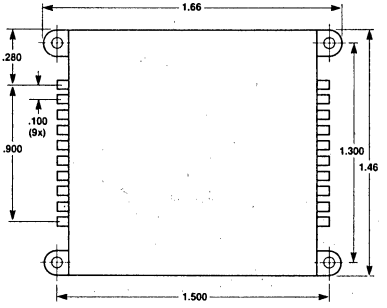
BUBBLE LEAD-LESS PACKAGE AND SOCKET

BUBBLE SOCKET #7905

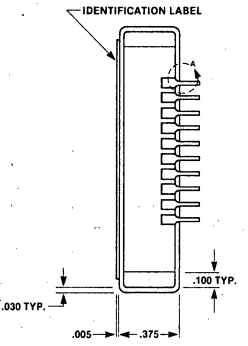


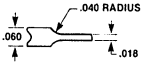
1MBIT LEADED BUBBLE MEMORY PACKAGE

TOP VIEW

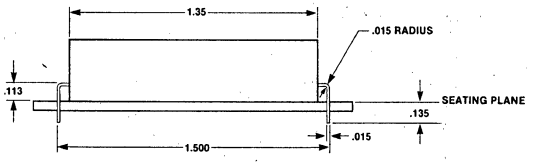


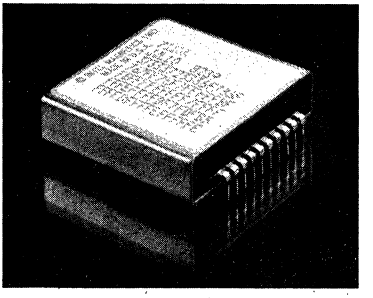
END VIEW



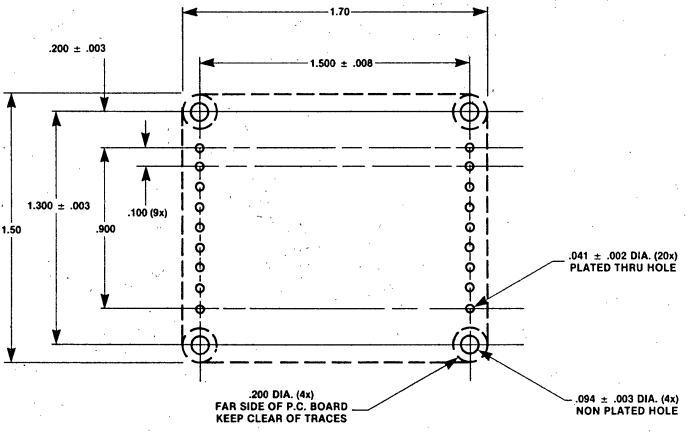


SIDE VIEW



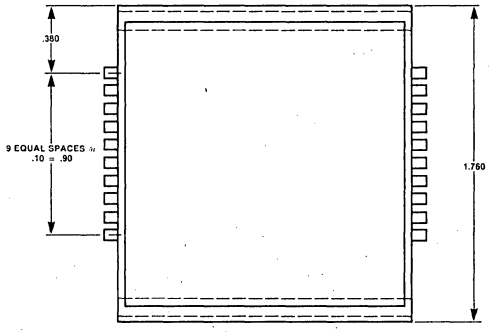


P.C. BOARD SPACE REQUIREMENTS

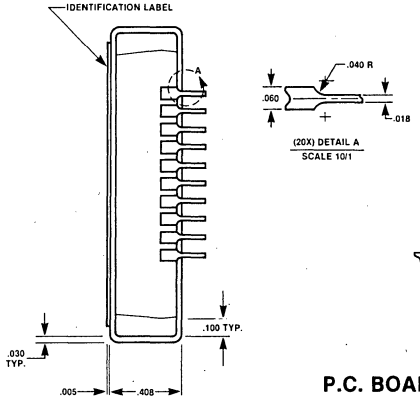


4MBIT BUBBLE MEMORY PACKAGE

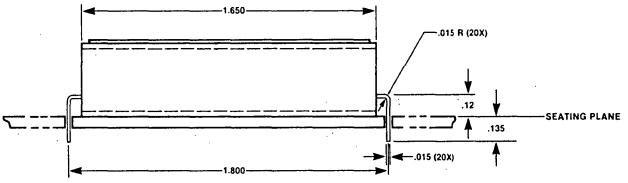
TOP VIEW



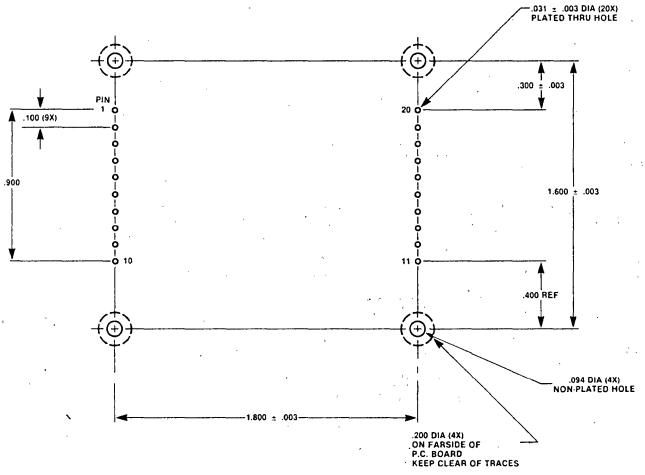
END VIEW



SIDE VIEW

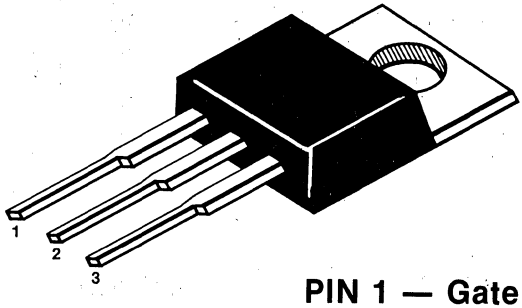


P.C. BOARD SPACE REQUIREMENTS

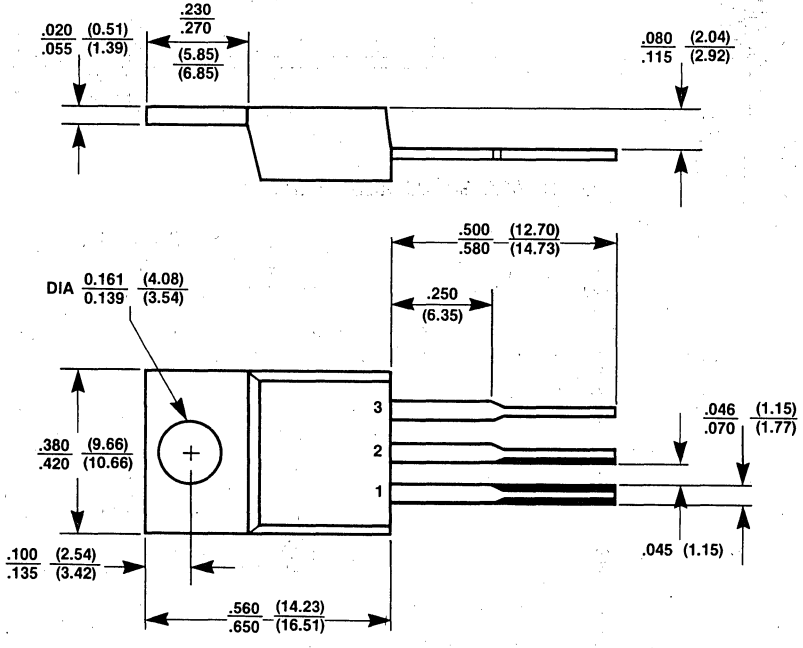


PACKAGING INFORMATION All dimensions in inches and (millimeters)

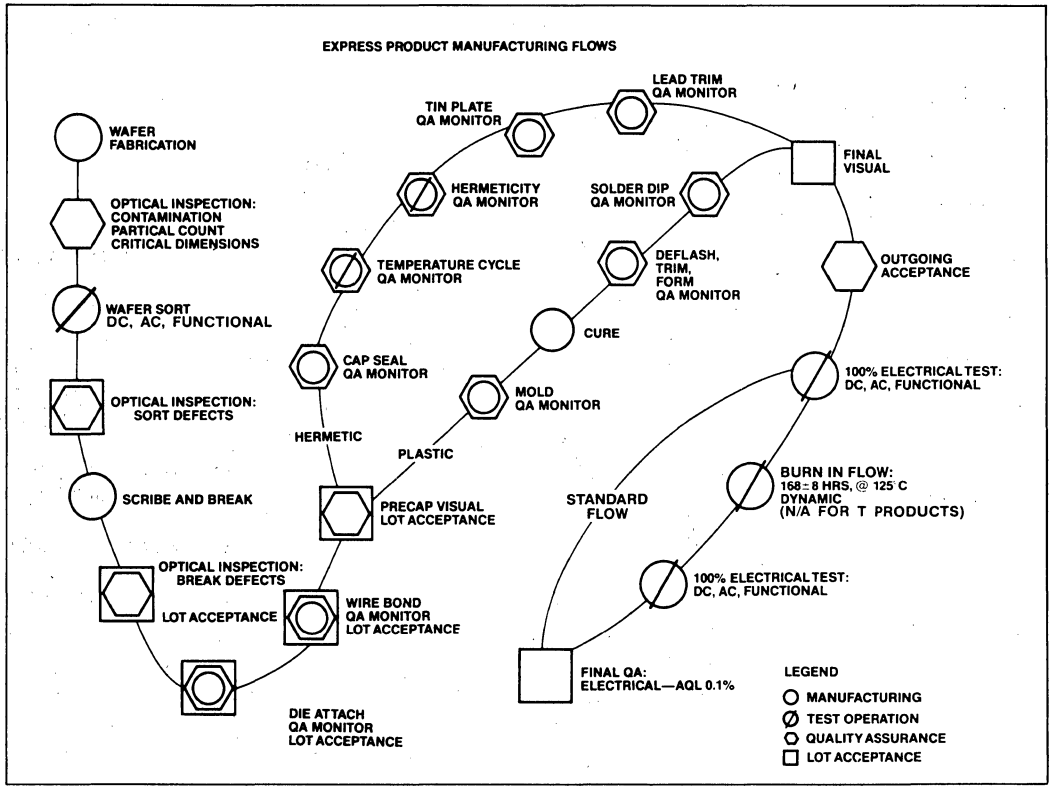
7264 (PART OF 4MBIT BUBBLE MEMORY SYSTEM)



PIN 1 — Gate
PIN 2 & TAB — Drain
PIN 3 — Source

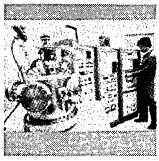


TO220


EXPRESS PRODUCT MANUFACTURING FLOWS
WAFER FABRICATION
OPTICAL INSPECTION: CONTAMINATION PARTICAL COUNT CRITICAL DIMENSIONS
WAFER SORT DC, AC, FUNCTIONAL
OPTICAL INSPECTION: SORT DEFECTS
SCRIBE AND BREAK
OPTICAL INSPECTION: BREAK DEFECTS
LOT ACCEPTANCE
DIE ATTACH QA MONITOR LOT ACCEPTANCE
WIRE BOND QA MONITOR LOT ACCEPTANCE
HERMETIC
PLASTIC
CAP SEAL QA MONITOR
PRECAP VISUAL LOT ACCEPTANCE
TEMPERATURE CYCLE QA MONITOR
HERMETICITY QA MONITOR
TIN PLATE QA MONITOR
LEAD TRIM QA MONITOR
MOLD QA MONITOR
CURE
DEFLASH, TRIM, FORM QA MONITOR
SOLDER DIP QA MONITOR
FINAL VISUAL
OUTGOING ACCEPTANCE
100% ELECTRICAL TEST: DC, AC, FUNCTIONAL
BURN IN FLOW: 168 ± 8 HRS, @ 125 C DYNAMIC (N/A FOR T PRODUCTS)
100% ELECTRICAL TEST: DC, AC, FUNCTIONAL
STANDARD FLOW
FINAL QA: ELECTRICAL—AQL 0.1%
LEGEND
MANUFACTURING
TEST OPERATION
QUALITY ASSURANCE
LOT ACCEPTANCE

QUALITY ASSURANCE OPERATIONS

PROCESS TECHNOLOGY



Reliability: Generate Design Rules
Generate Test Pattern
Process Qualification
Stress Testing
Failure Mode Analysis

PACKAGE TECHNOLOGY

Reliability: Process Qualification
Materials Test Methods
Materials Characterization

WAFER FABRICAT

Quality: Incoming Inspectio
High Magnification

Reliability: Process Qualificatio
Process Control, Mo
Analytical Test Labo

PRODUCT DESIGN

Quality: Parameter Limits, Testability
Test Program Control
Target Specification

PRODUCT VALIDATION/QUALIFICATION

Design Engineering: Design Verification
Design Engineering: Performance Verification
Applications Engineering: System Verification
Quality/Reliability: Qualification

ASSEMBLY

Quality: Incoming Inspection Gates
High/Low Magnification Visual Gates

Assembly Q.A. Acceptance

External Visual	Open/Short Test
Fine/Gross Leak®	Bond Pull
Centrifuge	Die Shear
Acoustic (PIND)	X-Ray
Mark Permanency	Internal Visual

Reliability: Process Qualification
Assembly Monitor Program

ASSEMBLY MONITOR PROGRAM

External Visual	Internal Visual
Fine/Gross Leak®	Temperature Cycle
Lid Torque	Thermal Shock
Temp/Humidity℗	Steam
Moisture Resistance®	

TEST AND FINISH

Quality: Final Q.A. Acceptance

Electrical Test Sample	Physical Dimensions
Solderability	External Visual
Mark Permanency	Conformance to Sales Order

Reliability: Reliability Monitor Program

RELIABILITY MONITOR PROGRAM

48 Hr, 125°C Dynamic Burn-In
1000 Hr, 125°C Dynamic Lifetest

NOTES:
® Hermetic packages, only.
℗ Plastic packages, only.

PLANT CLEARANCE

Quality: External Visual
Sales Order Requirements

Enclosed is my check for $ _____ payable to INTEL CORPORATION.

Please enroll me in the following workshop(s):

Workshop(s) _____ Date(s) _____

_____ _____

_____ _____

Confirmation number(s): _____ _____

_____ _____

Name: _____ Tel. No. _____

Please put correct mailing address on reverse side.

☐ Please keep me on your mailing list.

☐ Please remove my name from your mailing list.

BOSTON AREA
Intel Customer Training
27 Industrial Ave.
Chelmsford, MA 01824
(617) 256-1374

WASHINGTON D.C. AREA
Intel Customer Training
7833 Walker Dr., 5th Fl.
Greenbelt, MD 20770
(301) 474-2878

CANADA
Intel Customer Training
190 Atwell Drive
Toronto, Ontario M9W 6H8
(416) 494-6831

CHICAGO AREA
Intel Customer Training
Gould Center, East Tower
2550 Golf Rd., Suite 815
Rolling Meadows, IL 60008
(312) 981-7250

DALLAS AREA
Intel Customer Training
12300 Ford Road, Suite 380
Dallas, TX 75234
(214) 241-8087

SAN FRANCISCO AREA
Intel Customer Training
1350 Shorebird Way
Mt. View, CA 94043
(415) 940-7800

LOS ANGELES
Intel Customer Training
Kilroy Airport Center
2250 Imperial Highway
El Segundo, CA 90245
(415) 940-7800

Registration
To enroll, call the appropriate Intel Training Center between the hours of 8:30–12 and 1–4:30 and ask for Customer Training. Since enrollment is limited, registration 4–8 weeks in advance is recommended.

Confirmation No: _____
Your confirmation number, issued upon registration, guarantees your enrollment.

Payment:
Check or money order payable to Intel Corporation is due before workshop begins. To ensure proper credit, please include your enrollment confirmation number with your payment and send to the appropriate Training Center.

Workshop Hours: Workshops begin at 8:30 a.m. and end at approximately 5:00 p.m.

The best training you can get your hands on.

Intel Workshop Schedule 1984

Workshops (All Workshops 5 days unless otherwise specified)	JANUARY				FEBRUARY				MARCH				APRIL					MAY				JUNE			
	9	16	23	30	6	13	21	27	5	12	19	26	2	9	16	23	30	7	14	21	29	4	11	18	25
Introduction to Microprocessors P. 4 (4 days) M–TH	T		W	S			D L			B	C					S		B			S		W		
MCS-51 Microcontrollers P. 5		S			D			L			S		B			W	S		C			B		S	
MCS-96 16-Bit Microcontrollers P. 6 (4 Days) M-TH					C				S		B			D				C					B		
MCS-80/85 Microprocessors P. 7	D		S			B			W	C		S		N		L	T			B				S	W
iAPX 86, 88, 186 Microprocessors, Part I P. 8	S	B	D		S W	C		B	S	W	L N	T	S	B	S W	C	S	B	D L	M		S W	C	B	S T
iAPX 86, 88, 186 Microprocessors, Part II P. 9			B			S					W		T	S			C		B				S		
iAPX 286 Microprocessors P. 10		S		B				C		S			D	S		W		S		L	T				B
MCS-48/49 Microcontrollers P. 11				B													B								
Data Communications (Including Ethernet) P. 12 (4 Days)					B						S					B						D			
Speech Communication with Computers P. 13 (4 Days) M-TH		W										B							S						
Beginning Programming Using Pascal P. 15 (4 Days) M-TH	C				B			S				W				S				D					L
PLM Programming P. 16	B	C		S W		T		D		S			C		N	D		S	B				S	W	C
Ada* Programming, P. 17	Scheduled on Demand in San Francisco and Washington, D.C. Area																								
XENIX*/C Programming, P. 18				S		D		B		L		D		W			S			C			B		D
System 86/300 Applications Programming P. 19 (3 Days)			S		C				B		D			S						W	L	B		D	
iRMX 86 Operating System P. 20-21 (10 Days)	S	S B	C B	C	S W	S W		T L	T D L	D B	S B	S		C	S C	S N	D N	D W	S W	S		C L	C L	S B	S B
iRMX 88, 80 Operating System P. 22	Scheduled on Demand in Chicago Area																								
iDIS Database Information System P. 23 (4 Days) T-F			S				W				T		C		N			L			W		D		
iTPS Transaction Processing System P. 24 (4 Days)	Scheduled on Demand in San Francisco and Washington, D.C. Area																W								
System 2000 for Non-Programmers P. 27 (3 Days) M-W	0					W			Cg						W		T								N
System 2000 Technical Fundamentals P. 28		W			S T			N		Cg	De	W	O		S	C	W	T				W	De		
System 2000 Applications Programming P. 29			W			S T			N		Cg	De	W	O		S	C	W	T				W	De	
System 2000 Report Writing P. 30 (3 days) M-W							S W			N		Cg		C	O		S		W	T					De
System 2000 Database Design & Implementation P. 31		O		W						C				N					De		T			W	

	Other Areas	Call
De	Denver	312-981-7250
M	Minneapolis	312-981-7250
N	New York	301-474-2878
	Canada	**Call**
Cg	Calgary	416-494-6831
O	Ottawa	416-494-6831
T	Toronto	416-494-6831

	Area	USA Training Center	Telephone
B	**Boston**	27 Industrial Ave., Chelmsford, MA 01824	617-256-1374
C	**Chicago**	2550 Golf Rd., Suite 815, Rolling Meadows, IL 60008	312-981-7250
D	**Dallas**	12300 Ford Rd., Suite 380, Dallas, TX 75234	214-241-8087
L	**Los Angeles**	2250 Imperial Hwy., El Segundo, CA 90245	415-940-7800
S	**San Francisco**	1350 Shorebird Way, Mt. View, CA 94043	415-940-7800
W	**Washington, DC**	7833 Walker Drive, 5th Fl., Greenbelt, MD 20770	301-474-2878

Workshop Tuition

	LENGTH	USA		Canada		Prepaid Tuition
		Standard	Group	Standard	Group	
Introduction to Microprocessors Audio Cassette	Self Study	$750	NA	$950	NA	1
Introduction to Microprocessors	4 Days	$995	NA	$1295	NA	1
Beginning Programming Using Pascal	4 Days	$995	$895	$1295	$1165	1
MCS®-48/49 Microcontrollers	5 Days	$995	$895	$1295	$1165	1
MCS®-51 Microcontrollers	5 Days	$995	$895	$1295	$1165	1
MCS®-96 Microcontrollers	5 Days	$995	$895	$1295	$1165	1
MCS®-80/85 Microprocessors	5 Days	$995	$895	$1295	$1165	1
iAPX 86, 88 Microprocessors, Part I	5 Days	$995	$895	$1295	$1165	1
iAPX 86, 88, 186 Microprocessors, Part II	5 Days	$995	$895	$1295	$1165	1
iAPX 286 Microprocessors	5 Days	$995	$895	$1295	$1165	1
Data Communications including Ethernet*	4 Days	$995	$895	$1295	$1165	1
Speech Communication with Computers	4 Days	$995	$895	$1295	$1165	1
Ada* Programming	5 Days	$995	$885	$1295	$1165	1
PL/M Programming	5 Days	$995	$895	$1295	$1165	1
XENIX*/C Programming	5 Days	$995	$895	$1295	$1165	1
System 86/300 Applications Programming	3 Days	$995	$895	$1295	$1165	1
iRMX™-86 Operating System for Programmers	10 Days	$1995	$1795	$2595	$2335	2
iRMX™-88 Operating System	5 Days	$995	$895	$1295	$1165	1
iDIS™ Database Information System	4 Days	$995	$895	$1295	$1165	1
iTPS/TAPS Transaction Processing System	4 Days	$995	$895	$1295	$1165	1
System 2000® For Non-Programmers	3 Days	$495	$450	$625	$560	1/2 Attendees
System 2000® Technical Fundamentals	5 Days	$795	$720	$995	$895	1
System 2000® Report Writing	3 Days	$595	$550	$750	$675	3/2 Attendees
System 2000® Applications Programming	5 Days	$895	$795	$1100	$995	1
System 2000® Database Design and Implementation	5 Days	$895	$795	$1100	$995	1
Customer Site Training	Contact Regional Training Center					
Prepaid Tuition	$9600 for 12 ($12500 Canadian)					

*Ethernet is a trademark of Xerox Corp.; Ada is a Registered Trademark of the Ada Joint Program Office-US Government; XENIX is a trademark of Microsoft Corp.

DOMESTIC SALES OFFICES

ALABAMA

Intel Corp.
303 Williams Avenue, S.W.
Suite 1422
Huntsville 35801
Tel: (205) 533-9353

ARIZONA

Intel Corp.
11225 N. 28th Drive
Suite 214D
Phoenix 85029
Tel: (602) 869-4980

CALIFORNIA

Intel Corp.
1010 Hurley Way
Suite 300
Sacramento 95825
Tel: (916) 929-4078

Intel Corp.
7670 Opportunity Road
Suite 135
San Diego 92111
(714) 268-3563

Intel Corp.*
2000 East 4th Street
Suite 100
Santa Ana 92705
Tel: (714) 835-9642
TWX: 910-595-1114

Intel Corp.*
1350 Shorebird Way
Mt. View 94043
Tel: (415) 968-8086
TWX: 910-339-9279
910-338-0255

Intel Corp.*
5530 Corbin Avenue
Suite 120
Tarzana 91356
Tel: (213) 708-0333
TWX: 910-495-2045

COLORADO

Intel Corp.
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel: (303) 594-6622

Intel Corp.*
650 S. Cherry Street
Suite 720
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

CONNECTICUT

Intel Corp.
36 Padanaram Road
Danbury 06810
Tel: (203) 792-8366
TWX: 710-456-1199

EMC Corp.
393 Center Street
Wallingford 06492
Tel: (203) 265-6991

FLORIDA

Intel Corp.
1500 N.W. 62nd Street
Suite 104
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407

Intel Corp.
500 N. Maitland
Suite 205
Maitland 32751
Tel: (305) 628-2393
TWX: 810-853-9219

GEORGIA

Intel Corp.
3300 Holcombe Bridge Road
Suite 225
Norcross 30092
Tel: (404) 449-0541

ILLINOIS

Intel Corp.*
2550 Golf Road
Suite 815
Rolling Meadows 60008
Tel: (312) 981-7200
TWX: 910-651-5881

INDIANA

Intel Corp.
9100 Purdue Road
Suite 400
Indianapolis 46268
Tel: (317) 875-0623

IOWA

Intel Corp.
St. Andrews Building
1930 St. Andrews Drive N.E.
Cedar Rapids 52402
Tel: (319) 393-5510

KANSAS

Intel Corp.
8400 W. 110th Street
Suite 170
Overland Park 66210
Tel: (913) 642-8080

LOUISIANA

Industrial Digital Systems Corp.
2332 Severn Avenue
Suite 202
Metairie 70001
Tel: (504) 831-8492

MARYLAND

Intel Corp.*
7257 Parkway Drive
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

Intel Corp.
1620 Elton Road
Silver Spring 20903
Tel: (301) 431-1200

MASSACHUSETTS

Intel Corp.*
27 Industrial Avenue
Chelmsford 01824
Tel: (617) 256-1800
TWX: 710-343-6333

EMC Corp.
385 Elliot Street
Newton 02164
Tel: (617) 244-4740
TWX: 922531

MICHIGAN

Intel Corp.*
26500 Northwestern Hwy.
Suite 401
Southfield 48075
Tel: (313) 353-0920
TWX: 810-244-4915

MINNESOTA

Intel Corp.
3500 W. 80th Street
Suite 360
Bloomington 55431
Tel: (612) 835-6722
TWX: 910-576-2867

MISSOURI

Intel Corp.
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel: (314) 291-1990

NEW JERSEY

Intel Corp.*
Raritan Plaza III
Raritan Center
Edison 08837
Tel: (201) 225-3000
TWX: 710-480-6238

NEW MEXICO

Intel Corp.
1120 Juan Tabo N.E.
Albuquerque 87112
Tel: (505) 292-8086

NEW YORK

Intel Corp.*
300 Vanderbilt Motor Parkway
Hauppauge 11788
Tel: (516) 231-3300
TWX: 510-227-6236

Intel Corp.
80 Washington Street
Poughkeepsie 12601
Tel: (914) 473-2303
TWX: 510-248-0060

Intel Corp.*
211 White Spruce Boulevard
Rochester 14623
Tel: (716) 424-1050
TWX: 510-253-7391

T-Squared
6443 Ridings Road
Syracuse 13206
Tel: (315) 463-8592
TWX: 710-541-0554

T-Squared
7353 Pittsford
Victor Road
Victor 14564
Tel: (716) 924-9101
TWX: 510-254-8542

NORTH CAROLINA

Intel Corp.
2306 W. Meadowview Road
Suite 206
Greensboro 27407
Tel: (919) 294-1541

OHIO

Intel Corp.*
6500 Poe Avenue
Dayton 45414
Tel: (513) 890-5350
TWX: 810-450-2528

Intel Corp.*
Chagrin-Brainard Bldg., No. 300
28001 Chagrin Boulevard
Cleveland 44122
Tel: (216) 464-6915
TWX: 810-427-9298

OKLAHOMA

Intel Corp.
4157 S. Harvard Avenue
Suite 123
Tulsa 74135
Tel: (918) 749-8688

OREGON

Intel Corp.
10700 S.W. Beaverton
Hillsdale Highway
Suite 22
Beaverton 97005
Tel: (503) 641-8086
TWX: 910-467-8741

PENNSYLVANIA

Intel Corp.*
510 Pennsylvania Avenue
Fort Washington 19034
Tel: (215) 641-1000
TWX: 510-661-2077

Intel Corp.*
201 Penn Center Boulevard
Suite 301W
Pittsburgh 15235
Tel: (412) 823-4970

Q.E.D. Electronics
300 N. York Road
Hatboro 19040
Tel: (215) 674-9600

TEXAS

Intel Corp.*
12300 Ford Road
Suite 380
Dallas 75234
Tel: (214) 241-8087
TWX: 910-860-5617

Intel Corp.*
7322 S.W. Freeway
Suite 1490
Houston 77074
Tel: (713) 988-8086
TWX: 910-881-2490

Industrial Digital Systems Corp.
5925 Sovereign
Suite 101
Houston 77036
Tel: (713)988-9421

Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628

UTAH

Intel Corp.
268 West 400 South
Salt Lake City 84101
Tel: (801) 533-8086

VIRGINIA

Intel Corp.
1603 Santa Rosa Road
Suite 109
Richmond 23288
Tel: (804) 282-5668

WASHINGTON

Intel Corp.
110 110th Avenue N.E.
Suite 510
Bellevue 98004
Tel: (206) 453-8086
TWX: 910-443-3002

WISCONSIN

Intel Corp.
450 N. Sunnyslope Road
Suite 130
Brookfield 53005
Tel: (414) 784-9060

CANADA

ONTARIO

Intel Semiconductor of Canada, Ltd.
39 Hwy. 7, Bell Mews
Nepean K2H 8R2
Tel: (613) 829-9714
TELEX: 053-4115

Intel Semiconductor of Canada, Ltd.
50 Galaxy Boulevard
Suite 12
Rexdale M9W 4Y5
Tel: (416) 675-2105
TELEX: 06983574

Intel Semiconductor of Canada, Ltd.
201 Consumers Road
Suite 200
Willowdale M2J 4G8
Tel: (416) 494-6831
TELEX: 4946831

QUEBEC

Intel Semiconductor of Canada, Ltd.
3860 Cote Vertu Road
Suite 210
St. Laurent H4R 1V4
Tel: (514) 334-0560
TELEX: 05-824172

*Field Application Location

DOMESTIC DISTRIBUTORS

ALABAMA

†Arrow Electronics, Inc.
3611 Memorial Parkway So.
Huntsville 35405
Tel: (205) 882-2730

†Hamilton/Avnet Electronics
4812 Commercial Drive N.W.
Huntsville 35805
Tel: (205) 837-7210
TWX: 810-726-2162

†Pioneer/Huntsville
1207 Putnam Drive N.W.
Huntsville 35805
Tel: (205) 837-9300
TWX: 810-726-2197

ARIZONA

†Hamilton/Avnet Electronics
505 S. Madison Drive
Tempe 85281
Tel: (602) 231-5140
TWX: 910-950-0077

†Wyle Distribution Group
8155 N. 24th Street
Phoenix 85021
Tel: (602) 249-2232
TWX: 910-951-4282

CALIFORNIA

†Arrow Electronics, Inc.
521 Weddell Drive
Sunnyvale 94086
Tel: (408) 745-6600
TWX: 910-339-9371

†Arrow Electronics, Inc.
19748 Dearborn Street
Chatsworth 91311
Tel: (213) 701-7500
TWX: 910-493-2086

†Hamilton/Avnet Electronics
350 McCormick Avenue
Costa Mesa 92626
Tel: (714) 754-6051
TWX: 910-595-1928

†Hamilton/Avnet Electronics
19515 So. Vermont Avenue
Torrance 90502
Tel: (213) 615-3909
TWX: 910-349-6263

†Hamilton/Avnet Electronics
1175 Bordeaux Drive
Sunnyvale 94086
Tel: (408) 743-3300
TWX: 910-339-9332

†Hamilton/Avnet Electronics
4545 Viewridge Avenue
San Diego 92123
Tel: (714) 641-1850
TWX: 910-595-2638

†Hamilton/Avnet Electronics
10912 W. Washington Boulevard
Culver City 20230
Tel: (213) 558-2458
TWX: 910-340-6364

†Hamilton Avnet/Electronics
21050 Erwin Street
Woodland Hills 91367
Tel: (213) 883-0000
TWX: 910-494-2207

†Hamilton Electro Sales
3170 Pullman Street
Costa Mesa 92626
Tel: (714) 641-4109
TWX: 910-595-2638

†Hamilton/Avnet Electronics
4103 Northgate Boulevard
Sacramento 95834
Tel: (916) 920-3150

Kieruff Electronics, Inc.
3969 E. Bayshore Road
Palo Alto 94303
Tel: (415) 968-6292
TWX: 910-379-6430

Kieruff Electronics, Inc.
14101 Franklin Avenue
Tustin 92680
Tel: (714) 731-5711
TWX: 910-595-2599

Kieruff Electronics, Inc.
2585 Commerce Way
Los Angeles 90040
Tel: (213) 725-0325
TWX: 910-580-3666

†Wyle Distribution Group
124 Maryland Street
El Segundo 90245
Tel: (213) 322-8100
TWX: 910-348-7140 or 7111

CALIFORNIA (Cont'd)

†Wyle Distribution Group
9525 Chesapeake Drive
San Diego 92123
Tel: (714) 565-9171
TWX: 910-335-1590

†Wyle Distribution Group
3000 Bowers Avenue
Santa Clara 95051
Tel: (408) 727-2500
TWX: 910-338-0296

†Wyle Distribution Group
17872 Cowan Avenue
Irvine 92714
Tel: (714) 641-1600
TWX: 910-595-1572

COLORADO

†Wyle Distribution Group
451 E. 124th Avenue
Thornton 80241
Tel: (303) 457-9953
TWX: 910-936-0770

†Hamilton/Avnet Electronics
8765 E. Orchard Road
Suite 708
Englewood 80111
Tel: (303) 740-1017
TWX: 910-935-0787

CONNECTICUT

†Arrow Electronics, Inc.
12 Beaumont Road
Wallingford 06492
Tel: (203) 265-7741
TWX: 710-476-0162

†Hamilton/Avnet Electronics
Commerce Industrial Park
Commerce Drive
Danbury 06810
Tel: (203) 797-2800
TWX: 710-456-9974

†Harvey Electronics
112 Main Street
Norwalk 06851
Tel: (203) 853-1515
TWX: 710-468-3373

FLORIDA

†Arrow Electronics, Inc.
1001 N.W. 62nd Street
Suite 108
Ft. Lauderdale 33309
Tel: (305) 776-7790
TWX: 510-955-9456

†Arrow Electronics, Inc.
50 Woodlake Drive W.
Bldg. B
Palm Bay 32905
Tel: (305) 725-1480
TWX: 510-959-6337

†Hamilton/Avnet Electronics
6801 N.W. 15th Way
Ft. Lauderdale 33309
Tel: (305) 971-2900
TWX: 510-956-3097

†Hamilton/Avnet Electronics
3197 Tech. Drive North
St. Petersburg 33702
Tel: (813) 576-3930
TWX: 810-863-0374

†Pioneer/Alta Monte Springs
221 N. Lake Blvd.
Suite 412
Alta Monte Springs 32701
Tel: (305) 859-3600
TWX: 810-853-0284

†Pioneer/Ft. Lauderdale
1500 62nd Street N.W.
Suite 506
Ft. Lauderdale 33309
Tel: (305) 771-7520
TWX: 510-955-9653

GEORGIA

†Arrow Electronics, Inc
2979 Pacific Drive
Norcross 30071
Tel: (404) 449-8252
TWX: 810-766-0439

†Hamilton/Avnet Electronics
5825 D. Peachtree Corners
Norcross 30092
Tel: (404) 447-7500
TWX: 810-766-0432

†Pioneer/Georgia
5835B Peachtree Corners E
Norcross 30092
Norcross 30092
Tel: (404) 448-1711
TWX: 810-766-4515

ILLINOIS

†Arrow Electronics, Inc.
2000 E. Alonquin Street
Schaumberg 60195
Tel: (312) 397-3440
TWX: 910-291-3544

†Hamilton/Avnet Electronics
1130 Thorndale Avenue
Bensenville 60106
Tel: (312) 860-7780
TWX: 910-227-0060

†Pioneer/Chicago
1551 Carmen Drive
Elk Grove Village 60007
Tel: (312) 437-9680
TWX: 910-222-1834

INDIANA

†Arrow Electronics, Inc.
2718 Rand Road
Indianapolis 46241
(317) 243-9353
TWX: 810-341-3119

†Hamilton/Avnet Electronics
485 Gradle Drive
Carmel 46032
Tel: (317) 844-9333
TWX: 810-260-3966

†Pioneer/Indiana
6408 Castleplace Drive
Indianapolis 46250
Tel: (317) 849-7300
TWX: 810-260-1794

KANSAS

†Hamilton/Avnet Electronics
9219 Quivera Road
Overland Park 66215
Tel: (913) 888-8900
TWX: 910-743-0005

MARYLAND

†Hamilton/Avnet Electronics
6822 Oak Hall Lane
Columbia 21045
Tel: (301) 995-3500
TWX: 710-862-1861

†Mesa Technology Corporation
16021 Industrial Drive
Gaithersburg 20877
Tel: (301) 948-4350TWX: 710-828-9702

†Pioneer
9100 Gaither Road
Gaithersburg 20877
Tel: (301) 948-0710
TWX: 710-828-0545

MASSACHUSETTS

†Arrow Electronics, Inc.
1 Arrow Drive
Woburn 01801
Tel: (617) 933-8130
TWX: 710-393-6770

†Hamilton/Avnet Electronics
50 Tower Office Park
Woburn 01801
Tel: (617) 935-9700
TWX: 710-393-0382

†Harvey/Boston
44 Hartwell Avenue
Lexington 02173
Tel: (617) 863-1200
TWX: 710-326-6617

MICHIGAN

†Arrow Electronics, Inc.
3810 Varsity Drive
Ann Arbor 48104
Tel: (313) 971-8220
TWX: 810-223-6020

†Pioneer/Michigan
13485 Stamford
Livonia 48150
Tel: (313) 525-1800
TWX: 810-242-3271

†Hamilton/Avnet Electronics
32487 Schoolcraft Road
Livonia 48150
Tel: (313) 522-4700
TWX: 810-242-8775

†Hamilton/Avnet Electronics
2215 29th Street S.E.
Space A5
Grand Rapids 49508
Tel: (616) 243-8805
TWX: 810-273-6921

MINNESOTA

†Arrow Electronics, Inc.
5230 W. 73rd Street
Edina 55435
Tel: (612) 830-1800
TWX: 910-576-3125

†Hamilton/Avnet Electronics
10300 Bren Road East
Minnetonka 55343
Tel: (612) 932-0600
TWX: (910) 576-2720

†Pioneer/Twin Cities
10203 Bren Road East
Minnetonka 55343
Tel: (612) 935-5444
TWX: 910-576-2738

MISSOURI

†Arrow Electronics, Inc.
2380 Schuetz
St. Louis 63141
Tel: (314) 567-6888
TWX: 910-764-0882

†Hamilton/Avnet Electronics
13743 Shoreline Court
Earth City 63045
Tel: (314) 344-1200
TWX: 910-762-0684

NEW HAMPSHIRE

†Arrow Electronics, Inc.
1 Perimeter Road
Manchester 03103
Tel: (603) 668-6968
TWX: 710-220-1684

NEW JERSEY

†Arrow Electronics, Inc.
Pleasant Valley Avenue
Moorestown 08057
Tel: (215) 928-1800
TWX: 710-897-0829

†Arrow Electronics, Inc.
2 Industrial Road
Fairfield 07006
Tel: (201) 575-5300
TWX: 710-998-2206

†Hamilton/Avnet Electronics
1 Keystone Avenue
Bldg. 36
Cherry Hill 08003
Tel: (609) 424-0110
TWX: 710-940-0262

†Hamilton/Avnet Electronics
10 Industrial
Fairfield 07006
Tel: (201) 575-3390
TWX: 710-734-4388

†Harvey Electronics
45 Route 46
Pinebrook 07058
Tel: (201) 575-3510
TWX: 710-734-4382

†MTI Systems Sales
383 Route 46 W
Fairfield 07006
Tel: (201) 227-5552

NEW MEXICO

†Alliance Electronics Inc.
11030 Cochiti S.E.
Albuquerque 87123
Tel: (505) 292-3360
TWX: 910-989-1151

†Hamilton/Avnet Electronics
2524 Baylor Drive S.E.
Albuquerque 87106
Tel: (505) 765-1500
TWX: 910-989-0614

NEW YORK

†Arrow Electronics, Inc.
900 Broad Hollow Road
Farmingdale 11735
Tel: (516) 694-6800
TWX: 510-224-6126

†Arrow Electronics, Inc.
3000 South Winton Road
Rochester 14623
Tel: (716) 275-0300
TWX: 510-253-4766

†Arrow Electronics, Inc.
7705 Maltage Drive
Liverpool 13088
Tel: (315) 652-1000
TWX: 710-545-0230

†Arrow Electronics, Inc.
20 Oser Avenue
Hauppauge 11788
Tel: (516) 231-1000
TWX: 510-227-6623

†Microcomputer System Technical Demonstrator Centers.

DOMESTIC DISTRIBUTORS

NEW YORK (Cont'd)

†Hamilton/Avnet Electronics
333 Metro Park
Rochester 14623
Tel: (716) 475-9130
TWX: 510-253-5470

†Hamilton/Avnet Electronics
16 Corporate Circle
E. Syracuse 13057
Tel: (315) 437-2641
TWX: 710-541-1560

†Hamilton/Avnet Electronics
5 Hub Drive
Melville, Long Island 11747
Tel: (516) 454-6000
TWX: 510-224-6166

†Harvey Electronics
P.O. Box 1208
Binghamton 13902
Tel: (607) 748-8211
TWX: 510-252-0893

†Harvey Electronics
60 Crossway Park West
Woodbury, Long Island 11797
Tel: (516) 921-8700
TWX: 510-221-2184

†Harvey/Rochester
840 Fairport Park
Fairport 14450
Tel: (716) 381-7070
TWX: 510-253-7001

†MTI Systems Sales
38 Harbor Park Drive
Port Washington 11050
Tel: (516) 621-6200
TWX: 510-223-0846

NORTH CAROLINA

†Arrow Electronics, Inc.
938 Burke Street
Winston-Salem 27101
Tel: (919) 725-8711
TWX: 510-931-3169

†Hamilton/Avnet Electronics
3510 Spring Forest Drive
Raleigh 27604
Tel: (919) 878-0819
TWX: 510-928-1836

†Pioneer/Carolina
103 Industrial Avenue
Greensboro 27406
Tel: (919) 273-4441
TWX: 510-925-1114

OHIO

†Arrow Electronics, Inc.
7620 McEwen Road
Centerville 45459
Tel: (513) 435-5563
TWX: 810-459-1611

†Arrow Electronics, Inc.
6238 Cochran Road
Solon 44139
Tel: (216) 248-3990
TWX: 810-427-9409

†Hamilton/Avnet Electronics
954 Senate Drive
Dayton 45459
Tel: (513) 433-0610
TWX: 810-450-2531

†Hamilton/Avnet Electronics
4588 Emery Industrial Parkway
Warrensville Heights 44128
Tel: (216) 831-3500
TWX: 810-427-9452

OHIO (Cont'd)

†Pioneer/Dayton
4433 Interpoint Boulevard
Dayton 45424
Tel: (513) 236-9900
TWX: 810-459-1622

†Pioneer/Cleveland
4800 E. 131st Street
Cleveland 44105
Tel (216) 587-3600
TWX: 810-422-2211

OKLAHOMA

†Arrow Electronics, Inc.
4719 S. Memorial Drive
Tulsa 74145
Tel: (918) 665-7700

OREGON

†Almac Electronics Corporation
6022 S.W. Nimbus, Bldg. 7
Beaverton 97005
Tel: (503) 641-9070
TWX: 910-467-8743

†Hamilton/Avnet Electronics
6024 S.W. Jean Road
Bldg. C, Suite 10
Lake Oswego 97034
Tel: (503) 635-7848
TWX: 910-455-8179

PENNSYLVANIA

†Arrow Electronics, Inc.
650 Seco Road
Monroeville 15146
Tel: (412) 856-7000

†Pioneer/Pittsburgh
259 Kappa Drive
Pittsburgh 15238
Tel: (412) 782-2300
TWX: 710-795-3122

†Pioneer/Delaware Valley
261 Gibralter Road
Horsham 19044
Tel: (215) 674-4000
TWX: 510-665-6778

TEXAS

†Arrow Electronics, Inc.
13715 Gamma Road
Dallas 75234
Tel: (214) 386-7500
TWX: 910-860-5377

†Arrow Electronics, Inc.
10899 Kinghurst
Suite 100
Houston 77099
Tel: (713) 530-4700
TWX: 910-880-4439

†Arrow Electronics, Inc. 10125
Metropolitan
Austin 78758
Tel: (512) 835-4100
TWX: 910-874-1348

†Hamilton/Avnet Electronics
2401 Rutland
Austin 78757
Tel: (512) 837-8911
TWX: 910-874-1319

†Hamilton/Avnet Electronics
2111 W. Walnut Hill Lane
Irving 75062
Tel: (214) 659-4100
TWX: 910-860-5929

TEXAS (Cont'd)

†Hamilton/Avnet Electronics
8750 West Park
Houston 77063
Tel: (713) 780-1771
TWX: 910-881-5523

†Pioneer/Austin
9901 Burnet Road
Austin 78758
Tel: (512) 835-4000
TWX: 910-874-1323

†Pioneer/Dallas
13710 Omega Road
Dallas 75234
Tel: (214) 386-7300
TWX: 910-850-5563

†Pioneer/Houston
5853 Point West Drive
Houston 77036
Tel: (713) 988-5555
TWX: 910-881-1606

UTAH

†Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City 84119
Tel: (801) 972-2800
TWX: 910-925-4018

Wyle Distribution Group
1959 South 4130 West, Unit B
Salt Lake City 84104
Tel: (801) 974-9953

WASHINGTON

†Almac Electronics Corporation
14360 S.E. Eastgate Way
Bellevue 98007
Tel: (206) 643-9992
TWX: 910-444-2067

†Arrow Electronics, Inc.
14320 N.E. 21st Street
Bellevue 98007
Tel: (206) 643-4800
TWX: 910-444-2017

†Hamilton/Avnet Electronics
14212 N.E. 21st Street
Bellevue 98005
Tel: (206) 453-5874
TWX: 910-443-2469

WISCONSIN

†Arrow Electronics, Inc.
430 W. Rausson Avenue
Oakcreek 53154
Tel: (414) 764-6600
TWX: 910-262-1193

†Hamilton/Avnet Electronics
2975 Moorland Road
New Berlin 53151
Tel: (414) 784-4510
TWX: 910-262-1182

CANADA

ALBERTA

†Hamilton/Avnet Electronics
2816 21st Street N.E.
Calgary T2E 6Z3
Tel: (403) 230-3586
TWX: 03-827-642

†L.A. Varah, Ltd.
4742 14th Street N.E.
Calgary T2D 6L7
Tel: (403)230-1235
TWX: 038-258-97

ALBERTA (Cont'd)

Zentronics
Bay #1
3300 14th Avenue N.E.
Calgary T2A 6J4
Tel: (403) 272-1021

BRITISH COLUMBIA

L.A. Varah, Ltd.
2077 Alberta Street
Vancouver V5Y 1C4
Tel: (604) 873-3211
TWX: 610-929-1068

Zentronics
108-11400 Bridgeport Road
Richmond V6X 1T2
Tel: (604) 273-5575
TWX: 04-5077-89

MANITOBA

L.A. Varah, Ltd.
12-1832 King Edward Street
Winnipeg R2R ON1
Tel: (204) 633-6190
TWX: 07-55-365

Zentronics
590 Berry Street
Winnipeg R3H OS1
Tel: (204) 775-8661

ONTARIO

Hamilton/Avnet Electronics
6845 Rexwood Road
Units G & H
Mississauga L4V 1R2
Tel: (416) 677-7432
TWX: 610-492-8867

Hamilton/Avnet Electronics
210 Colonnade Road South
Nepean K2E 7L5
Tel: (613) 226-1700
TWX: 05-349-71

L.A. Varah, Ltd.
505 Kenora Avenue
Hamilton L8E 3P2
Tel: (416) 561-9311
TWX: 061-8349

Zentronics
8 Tilbury Court
Brampton L6T 3T4
Tel: (416) 451-9600
TWX: 06-976-78

Zentronics
564/10 Weber Street North
Waterloo N2L 5C6
Tel: (519) 884-5700

Zentronics
590 Berry Street
Winnipeg R3H OS1
Tel: (204) 775-8661

QUEBEC

Hamilton/Avnet Electronics
2670 Sabourin Street
St. Laurent H4S 1M2
Tel: (514) 331-6443
TWX: 610-421-3731

Zentronics
505 Locke Street
St. Laurent H4T 1X7
Tel: (514) 735-5361
TWX: 05-827-535

†Microcomputer System Technical Demonstrator Centers

EUROPEAN SALES OFFICES

BELGIUM

Intel Corporation S.A.
Parc Seny
Rue du Moulin a Papier 51
Boite 1
B-1160 Brussels
Tel: (02)661 07 11
TELEX: 28414

DENMARK

Intel Denmark A/S*
Lyngbyvej 32F 2nd Floor
DK-2100 Copenhagen East
Tel: (01) 18 20 00
TELEX: 19567

FINLAND

Intel Finland OY
Hameentie 103
SF - 00550 Helsinki 55
Tel: 0/716 955
TELEX: 123 332

FRANCE

Intel Corporation, S.A.R.L.*
5 Place de la Balance
Silic 223
94528 Rungis Cedex
Tel: (01) 687 22 21
TELEX: 270475

FRANCE (Cont'd)

Intel Corporation, S.A.R.L.
Immeuble BBC
4 Quai des Etroits
69005 Lyon
Tel: (7) 842 40 89
TELEX: 305153

WEST GERMANY

Intel Semiconductor GmbH*
Seidlstrasse 27
D-8000 Muenchen 2
Tel: (89) 53891
TELEX: 05-23177 INTL D

Intel Semiconductor GmbH*
Mainzer Strasse 75
D-6200 Wiesbaden 1
Tel: (6121) 70 08 74
TELEX: 04186183 INTW D

Intel Semiconductor GmbH
Brueckstrasse 61
7012 Fellbach
West Germany
Tel: (711) 58 00 82
TELEX: 7254826 INTS D

Intel Semiconductor GmbH*
Hohenzollern Strasse 5*
3000 Hannover 1
Tel: (511) 34 40 81
TELEX: 923625 INTH D

Intel Semiconductor GmbH
Ober-Ratherstrasse 2
D-4000 Dusseldorf 30
Tel: (211) 65 10 54
TELEX: 08-58977 INTL D

ISRAEL

Intel Semiconductor Ltd.*
P.O. Box 1659
Haifa
Tel: 4/524
TELEX: 46511

ITALY

Intel Corporation Italia Spa*
Milanofiori, Palazzo E
20094 Assago (Milano)
Tel: (02) 824 00 06
TELEX: 315183 INTMIL

NETHERLANDS

Intel Semiconductor Nederland B.V.*
Alexanderpoort Building
Marten Meesweg 93
3068 Rotterdam
Tel: (10) 21 23 77
TELEX: 22283

NORWAY

Intel Norway A/S
P.O. Box 92
Hvamveien 4
N-2013
Skjetten
Tel: (2) 742 420
TELEX: 18018

SWEDEN

Intel Sweden A.B.*
Box 20092
Archimedesvagen 5
S-16120 Bromma
Tel: (08) 98 53 85
TELEX: 12261

SWITZERLAND

Intel Semiconductor A.G.*
Forchstrasse 95
CH 8032 Zurich
Tel: (01) 55 45 02
TELEX: 57989 ICH CH

UNITED KINGDOM

Intel Corporation (U.K.) Ltd.*
5 Hospital Street
Nantwich, Cheshire CW5 5RE
Tel: (0270) 626 560
TELEX: 36620

Intel Corporation (U.K.) Ltd.*
Pipers Way
Swindon, Wiltshire SN3 1RJ
Tel: (0793) 488 388
TELEX: 444447 INT SWN

*Field Application Location

EUROPEAN DISTRIBUTORS/REPRESENTATIVES

AUSTRIA

Bacher Elektronische Geraete GmbH
Rotenuehlgasse 26
A 1120 Vienna
Tel: (222) 83 63 96
TELEX: 11532 BASAT A

BELGIUM

Inelco Belgium S.A.
Ave. des Croix de Guerre 94
B1120 Brussels
Tel: (02) 216 01 60
TELEX: 25441

DENMARK

MultiKomponent A/S
Fabriksparken 31
DK-2600 Glostrup
Tel: (02) 45 66 45
TX: 33355

Scandinavian Semiconductor
Supply A/S
Nannasgade 18
DK-2200 Copenhagen
Tel: (01) 83 50 90
TELEX: 19037

FINLAND

Oy Fintronic AB
Melkonkatu 24 A
SF-00210
Helsinki 21
Tel: (0) 692 60 22
TELEX: 124 224 Ftron SF

FRANCE

Generim
Z.I. de Courtaboeuf
Avenue de la Baltique
91943 Les Ulis Cedex-B.P.88
Tel: (6) 907 78 79
TELEX: F691700

Jermyn S.A.
rue Jules Ferry 35
93170 Bagnolet
Tel: (1) 859 04 04
TELEX: 21810 F

Metrologie
La Tour d' Asnieres
1, Avenue Laurent Cely
92606-Asnieres
Tel: (1) 791 44 44
TELEX: 611-448

FRANCE (Cont'd)

Tekelec Airtronic
Cite des Bruyeres
Rue Carle Vernet
F-92310 Sevres
Tel: (01) 534 75 35
TELEX: 204552

WEST GERMANY

Electronic 2000 Vertriebs A.G.
Neumarkter Strasse 75
D-8000 Munich 80
Tel: (89) 43 40 61
TELEX 522561 EIEC D

Jermyn GmbH
Postfach 1180
Schulstrasse 48
D-6277 Bad Camberg
Tel: (06434) 231
TELEX: 484426 JERM D

Celdis Enatechnik Systems GmbH
Gutenbergstrasse 4
2359 Henstedt-Ulzburg 1
Tel: (04193) 4026
TELEX: 2180260

Proelectron Vertriebs GmbH
Max Planck Strasse 1-3
6072 Dreieich bei Frankfurt
Tel: (6103) 33564
TELEX: 417983

IRELAND

Micro Marketing
Glenageary Office Park
Glenageary
Co. Dublin
Tel: (1) 85 62 88
TELEX: 31584

ISRAEL

Eastronics Ltd.
11 Rozanis Street
P.O. Box 39300
Tel Aviv 61390
Tel: (3) 47 51 51
TELEX: 33638

ITALY

Eledra 3S S.P.A.
Viale Elvezia, 18
I 20154 Milano
Tel: (2) 34 97 51
TELEX: 332332

ITALY (Cont'd)

Intesi
Milanfiori Pal. E/5
20090 Assago
Milano
Tel: (02) 82470
TELEX: 311351

NETHERLANDS

Koning & Hartman
Koperwerf 30
P.O. Box 43220
2544 EN's Gravenhage
Tel: 31 (70) 210.101
TELEX: 31528

NORWAY

Nordisk Elektronic (Norge) A/S
Postoffice Box 122
Smedsvingen 4
1364 Hvalstad
Tel: (2) 786 210
TELEX: 77546

PORTUGAL

Ditram
Componentes E Electronica LDA
Av. Miguel Bombarda, 133
P1000 Lisboa
Tel: (19) 545 313
TELEX: 14182 Brieks-P

SPAIN

Interface S.A.
Ronda San Pedro 22,3
Barcelona 10
Tel: (3) 301 78 51
TWX: 51508

ITT SESA
Miguel Angel 23-3
Madrid 10
Tel: (1) 419 54 00
TELEX: 27707

SWEDEN

AB Gosta Backstrom
Box 12009
Alstroemergatan 22
S-10221 Stockholm 12
Tel: (8) 541 080
TELEX: 10135

SWEDEN (Cont'd)

Nordisk Electronik AB
Box 27301
Sandhamnsgatan 71
S-10254 Stockholm
Tel: (8) 635 040
TELEX: 10547

SWITZERLAND

Industrade AG
Gemsenstrasse 2
Postcheck 80 - 21190
CH-8021 Zurich
Tel: (01) 363 23 20
TELEX: 56788 INDEL CH

UNITED KINGDOM

Bytech Ltd.
Unit 57
London Road
Earley, Reading
Berkshire
Tel: (0734) 61031
TELEX: 848215

Comway Microsystems Ltd.
Market Street
UK-Bracknell, Berkshire
Tel: 44 (344) 55333
TELEX: 847201

Jermyn Industries
Vestry Estate
Sevenoaks, Kent
Tel: (0732) 450144
TELEX: 95142

M.E.D.L.
East Lane Road
North Wembley
Middlesex HA9 7PP
Tel: (01) 904 93 07
TELEX: 28817

Rapid Recall, Ltd.
Rapid House/Denmark St
High Wycombe
Berks, England HP11 2ER
Tel: (0494) 26 271
TELEX: 837931

YUGOSLAVIA

H. R. Microelectronics Enterprises
P.O. Box 5604
San Jose, California 95150
Tel: 408/978-8000
TELEX: 278-559

INTERNATIONAL SALES OFFICES

AUSTRALIA

Intel Semiconductor Pty. Ltd.
Spectrum Building
200 Pacific Highway
Level 6
Crows Nest, NSW, 2089
Australia
Tel: 011-61-2-436-2744
TELEX: 790-20097
FAX: 011-61-2-923-2632

HONG KONG

Intel Semiconductor Ltd.
13/F Hong Kong Trade Centre
161-167 Des Voeux Road Central
Tel: 011-852-5-450-885
TELEX: 63869 ISLHKHX

JAPAN

Intel Japan K.K.
5-6 Tokodai, Toyosato-machi
Tsukuba-gun, Ibaraki-ken 300-26
Tel: 029747-8511
TELEX: 03656-160

JAPAN (Cont'd)

Intel Japan K.K.*
2-1-15 Naka-machi
Atsugi, Kanagawa 243
Tel: 0462-23-3511

Intel Japan K.K.*
2-51-2 Kojima-cho
Chofu, Tokyo 182
Tel: 0424-88-3151

Intel Japan K.K.*
2-69 Hon-cho
Kumagaya, Saitama 360
Tel: 0485-24-6871

JAPAN (Cont'd)

Intel Japan K.K.*
2-4-1 Terauchi
Toyonaka, Osaka 560
Tel: 06-863-1091

Intel Japan K.K.
1-5-1 Marunouchi
Chiyoda-ku, Tokyo 100
Tel: 03-201-3621/3681

Intel Japan K.K.*
1-23-9 Shinmachi
Setagaya-ku, Tokyo 154
Tel: 03-426-2231

*Field Application Location

INTERNATIONAL DISTRIBUTORS/REPRESENTATIVES

ARGENTINA

VLC S.R.L.
Sarmiento 1630, 1 Piso
1042 Buenos Aires
Tel: 35-1201/9242
TELEX: Public Booth 9900 or 9901

Mailing Address
Soimex International Corporation
15 Park Row, Room #1730
New York, New York 10038
(212) 406-3052
Attn: Gaston Briones

AUSTRALIA

Total Electronics
9 Harker Street
Burwood
Victoria 3125
Tel: 61 3 288-4044
TELEX: AA 31261

Mailing Address
Private Bag 250
Burwood, Victoria 3125
Australia

Total Electronics
#1 Johnstone Lane
Lane Cove, N.S.W. 2066
TELEX: 26297

BRAZIL

Icotron S.A.
05110 Av. Mutinga 3650-6 Andar
Pirituba Sao Paulo
Tel: 261-0211
TELEX: 1122274/ICOTBR

CHILE

DIN
AV. VIC MCKENNA 204
Casilla 6055
Santiago
Tel: 227 564
TELEX: 352 003

COLUMBIA

International Computer Machines
Carrera 7 No. 72-34
Apdo. Aereo 19403
Bogota 1
Tel: 211-7282
TELEX: 45716 ICM CO

HONG KONG

Schmidt & Co. Ltd.
Wing on Centre, 28th Floor
111 Connaught Road Central
Tel: 5 8521 222
TELEX: 74766 SCHMC HX

INDIA

Micronic Devices
104/109C, Nirmal Industrial Estate
Sion (E)
Bombay 400022
Tel: 486-170
TELEX: 011-71447 MDEV IN

JAPAN

Asahi Electronics Co. Ltd.
KMM Bldg. Room 407
2-14-1 Asano, Kokura
Kita-Ku, Kitakyushu City 802
Tel: (093) 511-6471
TELEX: AECKY 7126-16

JAPAN (Cont'd)

Hamilton-Avnet Electronics Japan Ltd.
YU and YOU Bldg. 1-4 Horidome-Cho
Nihonbashi Chuo-Ku, Tokyo 103
Tel: (03) 662-9911
TELEX: 2523774

Ryoyo Electric Corp.
Konwa Bldg.
1-12-22, Tsukiji
Chuo-Ku, Tokyo 104
Tel: (03) 543-7711/541-7311

Tokyo Electron Ltd.
Shin Juku, Nomura Bldg.
26-2 Nishi-Shin Juku-Ichome
Shin Juku-Ku, Tokyo 160
Tel. (03) 343-4411
TELEX: 232-2220 LABTEL J

KOREA

Koram Digital
2nd Floor, Government Pension Bldg.
1-589, Yoido-Dong
Youngdungpo-Ku
Seoul 150
Tel: 782-8039 or 8049
TELEX: KODIGIT K25 299

NEW ZEALAND

McLean Information Technology Ltd.
459 Kyber Pass Road, Newmarket,
P.O. Box 9464, Newmarket
Auckland 1, New Zealand
Tel: 501-801, 501-219, 587-037
TELEX: NZ21570 THERMAL

SINGAPORE

General Engineers Corporation Pty. Ltd.
18 Pasir Panjang Road
11-05/08 PSA Multi Storey Complex
Singapore 0511
Tel: 011-65-271-3163
TELEX: RS23987 GENERCO

SOUTH AFRICA

Electronic Building Elements, Pty. Ltd.
P.O. Box 4609
Hazelwood, Pretoria 0001
Tel: 011-27-12-46-9221 or 9227
TELEX: 3-0181 SA

TAIWAN

Taiwan Automation Corp.*
3rd Floor #75, Section 4
Nanking East Road
Taipei
Tel: 771-0940 or 0941
TELEX: 11942 TAIAUTO

YUGOSLAVIA

H. R. Microelectronics Enterprises
P.O. Box 5604
San Jose, California 95150
Tel: (408) 978-8000
TELEX: 278-559

*Field Application Location

U.S. SERVICE OFFICES

CALIFORNIA

Intel Corp.
1350 Shorebird Way
Mt. View 94043
Tel: (415) 968-8211
TWX: 910-339-9279
910-338-0255

Intel Corp.
2000 E. 4th Street
Suite 110
Santa Ana 92705
Tel: (714) 835-5577
TWX: 910-595-2475

Intel Corp.
7670 Opportunity Road
San Diego 92111
Tel: (714) 268-3563

Intel Corp.
5530 N. Corbin Avenue
Suite 120
Tarzana 91356
Tel: (213) 708-0333

COLORADO

Intel Corp.
650 South Cherry
Suite 720
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

CONNECTICUT

Intel Corp.
36 Padanaram Road
Danbury 06810
Tel: (203) 792-8366

FLORIDA

Intel Corp.
1500 N.W. 62nd Street
Suite 104
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407

Intel Corp.
500 N. Maitland Avenue
Suite 205
Maitland 32751
Tel: (305) 628-2393
TWX: 810-853-9219

Intel Corp.
5151 Adanson Street
Orlando 32804
Tel: (305) 628-2393

GEORGIA

Intel Corp.
3300 Holcombe Bridge Road
Suite 225
Norcross 30092
Tel: (404) 441-1171

ILLINOIS

Intel Corp.
2550 Golf Road
Suite 815
Rolling Meadows 60008
Tel: (312) 981-7270
TWX: 910-253-1825

KANSAS

Intel Corp.
8400 W. 110th Street
Suite 170
Overland Park 66210
Tel: (913) 642-8080

MARYLAND

Intel Corp. 7257 Parkway Drive
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

MASSACHUSETTS

Intel Corp.
27 Industrial Avenue
Chelmsford 01824
Tel: (617) 256-1800
TWX: 710-343-6333

MICHIGAN

Intel Corp.
26500 Northwestern Highway
Suite 401
Southfield 48075
Tel: (313) 354-1540
TWX: 810-244-4915

MINNESOTA

Intel Corp.
7401 Metro Boulevard
Suite 355
Edina 55435
Tel: (612) 835-6722
TWX: 910-567-2867

MISSOURI

Intel Corp.
4203 Earth City Expressway
Suite 143
Earth City 63045
Tel: (314) 291-2015

NEW JERSEY

Intel Corp.
385 Sylvan Avenue
Englewood Cliffs 07632
Tel: (201) 567-0820
TWX: 710-991-8593

NEW YORK

Intel Corp.
2255 Lyell Avenue
Rochester 14606
Tel: (716) 254-6120

NORTH CAROLINA

Intel Corp.
5600 Executive Drive
Suite 113
Charlotte 28212
Tel: (704) 568-8966

Intel Corp.
2306 W. Meadowview Road
Suite 206
Greensboro 27407
Tel: (919) 294-1541

OHIO

Intel Corp.
Chagrin-Brainard Bldg.
Suite 305
28001 Chagrin Boulevard
Cleveland 44122
Tel: (216) 464-6915
TWX: 810-427-9298

Intel Corp.
6500 Poe Avenue
Dayton 45414
Tel: (800) 325-4415
TWX: 810-450-2528

OKLAHOMA

Intel Corp.
4157 S. Harvard
Suite 123
Tulsa 74101
Tel: (918) 744-8068

OREGON

Intel Corp.
10700 S.W. Beaverton-Hillsdale
Highway
Suite 22
Beaverton 97005
Tel: (503) 641-8086
TWX: 910-467-8741

PENNSYLVANIA

Intel Corp.
500 Pennsylvania Avenue
Fort Washington 19034
Tel: (215) 641-1000
TWX: 510-661-2077

Intel Corp.
201 Penn Center Boulevard
Suite 301 W
Pittsburgh 15235
Tel: (313) 354-1540

TEXAS

Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512)454-3628
TWX: 910-874-1347

Intel Corp.
12300 Ford Road
Suite 380
Dallas 75234
Tel: (214) 241-8087
TWX: 910-860-5617

Intel Corp.
7322 S.W. Freeway
Suite 1490
Houston 77074
Tel: (713) 988-8088
TWX: 910-881-2490

VIRGINIA

Intel Corp.
7700 Leesburg Pike
Suite 412
Falls Church 22043
Tel: (703) 734-9707
TWX: 710-931-0625

WASHINGTON

Intel Corp.
110 110th Avenue N.E.
Suite 510
Bellevue 98004
Tel: 1-800-538-0662
TWX: 910-443-3002

WISCONSIN

Intel Corp.
150 S. Sunnyslope Road
Suite 148
Brookfield 53005
Tel: (414) 784-9060

Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

Intel International (U.K.) Ltd.
Piper's Way
Swindon, SN3 1RJ
Wiltshire, England

Intel Japan K.K.
5-6 Tokodai Toyosato-machi
Tsukuba-gun, Ibaraki-ken 300-26
Japan

Printed in U.S.A./C-626/115K/RRD/JR
Memory