intel® 386™ MicroComputer Model 300SX **Board Technical Reference Manual**

Intel 386™
MicroComputer
Model 300SX

Board Technical Reference Manual

intel®

300SX

Order Number:
459941-001

INTEL
386™
MICROCOMPUTER

intel®

Intel386™ MicroComputer Model 300SX Board Technical Reference Manual

Order Number: 459941-001

WARNING

This equipment has been certified to comply with the limits for a class B computing device, pursuant to subpart J of part 15 FCC rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

All cables used to connect to peripherals must be shielded and grounded. Operation with cables, connected to peripherals, which are not shielded and grounded may result in interference to radio and TV reception.

This equipment meets or exceeds requirements for safety in the US (UL 478 5th Edition), Canada (CSA C22.2 No. 220), and Europe (IEC 380, IEC 435, IEC 950, and VDE 0806).

This equipment has been tested for radio frequency emissions and has been verified to meet VDE 0871 Class B.

This digital apparatus does not exceed the Class B limits for radio noise emissions set out in the radio interference regulations of the Canadian Department of Communications.

Ce dispositif digital, s'il est utilise suivant les instructions et recommandations du constructeur, ne depasse pas les limites de la Classe B pour le bruit des frequences radio, etablies par les Regles sur l'interference radio du Ministere Canadian des Communications.

RADIO FREQUENCY INTERFERENCE NOTICE

This equipment generates and uses radio frequency energy and if not installed properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type-tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 FCC rules, which are designed to provide reasonable protection against interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If the equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna.
- Relocate the system with respect to the receiver.
- Move the system away from the receiver.
- Plug the system into a different outlet so that the system and receiver are on different branch circuits.
- Move the cables connected to the system to minimize the interference.
- Tighten all screws on cables and the system housing.
- Install blank panels, originally supplied with the system, in all unused card slots.

If necessary the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet, prepared by the Federal Communications Commission, helpful:

"How to Identify and Resolve Radio-TV Interference Problems"

This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402. Stock No. 004-00398-5.

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

Intel Corporation
Literature Sales
P.O. Box 58130
Santa Clara, CA 95052-8130

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel Corporation 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 its affiliates, and may be used only to identify Intel products:

Above
BITBUS
COMMputer
CREDIT
Data Pipeline
ETOX
Genius
i486
i860
⇧
i®
I^{2}CE
ICE
iCEL
ICEVIEW
iCS
iDBP
iDIS
iLBX
i$_m$®

iMDDX
iMMX
Inboard
Insite
Intel®
int$_e$l®
Intel376
Intel386
int$_e$lBOS
Intel Certified
Intelevision
int$_e$ligent Identifier
int$_e$ligent Programming
Intellec®
Intellink
iOSP
iPAT
iPDS
iPSC®
iRMK

iRMX
iSBC
iSBX
iSDM
iSXM
KEPROM
Library Manager
MAPNET
MCS®
Megachassis
MICROMAINFRAME
MULTIBUS
MULTICHANNEL
MULTIMODULE
ONCE
OpenNET
OTP
PC BUBBLE
Plug-A-Bubble
PROMPT

Promware
QueX
QUEST
Quick-Erase
Quick Pulse Programming
Ripplemode
RMX/80
RUPI
Seamless
SLD
SugarCube
UPI
VLSiCEL
376
386
386SX
387
387SX
4-SITE
486

MULTIBUS is a patented Intel bus.

Princeton is a registered trademark of Princeton Graphics Systems.

IBM is a registered trademark of International Business Machines, Inc.

PC, XT, and AT are trademarks of International Business Machines, Inc.

Microsoft, MS-DOS, OS/2, and XENIX are registered trademarks of Microsoft Corporation.

Multiplan is a trademark of Microsoft Corporation.

UNIX is a trademark of AT&T.

Lotus 1-2-3 and Symphony are registered trademarks of Lotus Development Corporation.

Framework II and dBASE III PLUS are registered trademarks of Ashton-Tate.

MultiMate is a trademark of Ashton-Tate.

REV.	REVISION HISTORY	DATE
-001	Original Issue (Release 1.0)	10/89

Contents

About This Manual

1 Board Overview

2 Central Processing Core

3 Intel 82335 High-Integration Interface Device

3 Intel 82335 High-Integration Interface Device (continued)

3 Intel 82335 High-Integration Interface Device (continued)

4 Intel 82330/82331 Chip Set

4 Intel 82330/82331 Chip Set (continued)

4 Intel 82330/82331 Chip Set (continued)

5 Communication Ports

5 Communication Ports (continued)

6 Keyboard and Mouse Controller

6 Keyboard and Mouse Controller (continued)

7 Video Display Subsystem

8 Floppy Disk Drive Controller

8 Floppy Disk Drive Controller (continued)

9 Intel ISA Bus Interface

9 Intel ISA Bus Interface (continued)

10 External Interfaces

11 Power-on Self Test and Setup

A Specifications

B System BIOS Specifications

B System BIOS Specifications (continued)

B System BIOS Specifications (continued)

B System BIOS Specifications (continued)

C Jumper Settings

D Changing CPU Speed

E Messages

F Device Mapping

G Hot Keys

H Component Installation

Glossary

Index

Tables

2 Central Processing Core

3 Intel 82335 High-Integration Interface Device

4 Intel 82330/82331 Chip Set

4 Intel 82330/82331 Chip Set (continued)

5 Communication Ports

6 Keyboard and Mouse Controller

7 Video Display Subsystem

8 Floppy Disk Drive Controller

9 Intel ISA Bus Interface

10 External Interfaces

10 External Interfaces (continued)

A Specifications

B System BIOS Specifications

C Jumper Settings

E Messages

F Device Mapping

Figures

1 Board Overview

2 Central Processing Core

3 Intel 82335 High-Integration Interface Device

4 Intel 82330/82331 Chip Set

4 Intel 82330/82331 Chip Set (continued)

5 Communication Ports

6 Keyboard and Mouse Controller

7 Video Display Subsystem

8 Floppy Disk Drive Controller

10 External Interfaces

11 Power-on Self Test and Setup

A Specifications

C Jumper Settings

D Changing CPU Speed

F Device Mapping

H Component Installation

About This Manual

PURPOSE

This manual provides reference data for the Intel386™ MicroComputer Model 300SX System Board. The Technical Reference Manual gives system designers sufficient information to allow them to integrate the board into systems, and to test and evaluate that integration. System designers may also use this manual on a continuing basis to support their customers, solve problems, or expand the system. For example, this manual may be used to:

- Design or select system components such as expansion boards, peripheral devices, and replacement parts
- Select or design systems and applications software
- Solve system integration and interfacing problems
- Troubleshoot at an advanced level
- Program complex applications and systems software such as device drivers, interrupt handlers, etc.

LIBRARY PLAN

This manual is the second in a set of three manuals written for the Intel386 MicroComputer Model 300SX board. A brief description of these manuals is as follows:

- *Intel386™ MicroComputer Model 300SX System Technical Reference Manual*. This manual and its companion volume, *Intel386™ MicroComputer Model 300SX Board Technical Reference Manual*, are written primarily for an Original Equipment Manufacturer (OEM), system engineer, or hardware or software designer. The system manual focuses on the major elements contained in the system module chassis. It provides a general description of the system, a brief overview of the system board, all internal interfaces to other components, and all external interfaces. Basic installation and removal procedures for system components and peripheral devices are also included. A detailed description of the system board is contained in a separate manual (see following).

- *Intel386™ MicroComputer Model 300SX Board Technical Reference Manual*. This manual describes the system board in detail, and is written for engineers who design system accessories and for programmers who require information on hardware and firmware specifications.

- *Intel386™ MicroComputer Model 300SX User's Guide*. This manual is written for an end-user. It describes all system features, installation and operation of the system, how to install or remove system components, and basic troubleshooting procedures should problems occur.

AUDIENCE

This manual is written for an Original Equipment Manufacturer (OEM), or a system engineer or hardware designer. As such, it assumes you are familiar with the general terminology used in the field of microprocessor and microcomputer design.

ORGANIZATION

This manual is organized into 11 chapters and eight appendixes as follows:

Chapter 1 Board Overview — provides an overview of the 300SX board. Included is a list of features, a block diagram of the board, and a description of the feature set.

Chapter 2 Central Processing Core — describes the operation of the CPU, the 387™ numeric coprocessor, and the data and controller buffers.

Chapter 3 Intel 82335 High-Integration Interface Device — presents an overview of the Intel 82335 and describes its internal functional blocks.

Chapter 4 Intel 82330/82331 Chip Set — describes the operation of the Intel 82330/82331 Chip Set which provides a high-integration interface for a PC-AT system.

Chapter 5 Communication Ports — describes the board's serial and parallel ports.

Chapter 6 Keyboard and Mouse Controller — describes the operation of the keyboard and mouse controller.

Chapter 7 Video Display Subsystem — describes the operation of the onboard graphics chip set and the digital-to-analog converter.

Chapter 8 Floppy Disk Drive Controller — describes the operation of the 87072 onboard floppy disk drive controller.

Chapter 9 Intel ISA Bus Interface — introduces the ISA bus, provides general attributes of the ISA bus, and describes the ISA bus signals.

Chapter 10 External Interfaces — describes the external interfaces included on the board.

Chapter 11 Power-on Self Test — describes the Power-on Self Test and the Setup utility stored in ROM.

Appendix A Specifications — provides specifications for the 300SX board.

Appendix B System BIOS Specifications — provides BIOS specifications for the 300SX board.

Appendix C Jumper Settings — provides jumper settings for the 300SX board.

Appendix D Changing CPU Speed — describes how to decrease the processor speed to 8 MHz.

Appendix E Messages — describes the various screen messages and error beep codes. Included are POST messages, run-time messages, 300SX board error messages, and beep codes.

Appendix F Device Mapping — provides tables that list mapping and addressing information related to 300SX board memory and onboard devices.

Appendix G Hot Keys — lists keystroke sequences used to invoke special system functions.

Appendix H Component Installation — describes how to install the Intel 387SX numeric coprocessor and single in-line memory modules (SIMMs).

Glossary List of standard acronyms and technical terms with definitions.

Index Manual index.

HOW TO OBTAIN MORE INFORMATION

You may be interested in obtaining further information about products and services relating to the Intel386 MicroComputer Model 300SX System Board. Or you may require more detailed information than is provided in this manual.

Please contact your local Intel Sales Office if you desire additional information.

NOTATIONAL CONVENTIONS

Certain notational conventions are used throughout this manual and others in the library. Refer to the glossary for specific definitions. Notational conventions include:

system — Throughout this manual, the term "300SX system" or "system" applies to the Intel386™ MicroComputer Model 300SX System.

board — Throughout this manual, the term "300SX board" or "board" applies to the Intel386™ MicroComputer Model 300SX System Board.

F1 — A letter, number, symbol, or word enclosed in a double rectangle, and printed in small type represents a a key on your keyboard. For example, the instruction "press F1" means press the key labeled "F1" on your keyboard.

Backspace — This manual refers to most keys by the symbol, letter, or name printed on the key. The exception is the Backspace key. The Backspace key is called Backspace to distinguish it from the left arrow key.

Enter — This manual uses Enter to refer to the two Enter keys. Other manuals refer to the Enter keys as RETURN, CARRIAGE RETURN, or use an arrow. All these items are interchangeable.

x + y — Two or three key names with plus signs between them indicate multiple-key entries. For example, Ctrl + Alt + Del means hold down the Ctrl and Alt keys and press the Del key.

* — In signal definitions, the asterisk (*) following a signal name indicates an active low signal; for example, IOCHECK*.

H — An H suffix to a numerical value denotes hexadecimal numbers. For example, 0F8H means 0F8 (hexadecimal).

K — A K (upper case) suffix to a numerical value is used to indicate size in kilobytes; i.e., 7168K, 640K, etc. Note, that while a kilobyte is defined as 1024 bytes, the lower case k prefix used in other measurements indicates a quantity of 1,000. The K suffix is synonymous with Kbyte. See Glossary.

Kb — A Kb suffix to a numerical value indicates size in kilobits. For example: 512Kb. (One kilobit is defined as 1024 bits.)

M — An M (uppercase) suffix to a numerical value is used to indicate size in megabytes; i.e., 1M, 256M, etc. Note, however, that while a megabyte is defined as 1,048,576 bytes the M prefix used in other measurements indicates a quantity of 1,000,000. The M suffix is synonymous with megabyte. See Glossary.

Mb — An Mb suffix to a numerical value indicates size in megabits. For example: 4Mb. (One megabit is defined as 1,048,576 bits.)

All system messages (screen display) are shown in a non-proportional font to simulate the appearance of a screen display.

An italicized word or phrase is used to represent a variable, a publication title, or occasionally, to lend emphasis in textual descriptions. Where shown, DOS, UNIX or XENIX files, path names, and directories are also italicized.

The longer POST and Boot error and informational messages in the index are followed by an ellipsis (three periods). This convention is used to denote that the message in the index is incomplete. The portion appearing in the index is of sufficient length to make a unique identification.

Four kinds of special admonishments are used throughout the text to emphasize specific information. Examples of each type of admonishment are as follows:

Note

Notes are used to provide the reader with important or explanatory information that stands out from the rest of the text.

DANGER

DANGER indicates the presence of a hazard that *will* cause death or severe personal injury if the hazard is not avoided.

WARNING

WARNING indicates the presence of a hazard that *can* cause death or severe personal injury if the hazard is not avoided.

CAUTION

CAUTION indicates the presence of a hazard that *can or will* cause minor personal injury or damage to hardware or software.

RELATED PUBLICATIONS

Refer to the following publications for additional information relating to the Intel386 MicroComputer Model 300SX Board and its operating environment.

- *Intel386™ MicroComputer Model 300SX System Technical Reference Manual* (Intel order number 459940-001)
- *Intel386™ MicroComputer Model 300SX User's Guide* (Intel order number 501329-001)
- *Introduction to the 80386* (Intel order number 231252-001)
- *80386 Programmer's Reference Manual* (Intel order number 230985-001)
- *386™ Microprocessor Hardware Reference Manual* (Intel order number 231732-003)
- *80386 System Software Writer's Guide* (Intel order number 231499-001)
- *Microprocessor and Peripheral Handbook* (Volumes I and II) (Intel order number 230843-006)

Board Overview

1

1.1 INTRODUCTION

This chapter provides an overview of the 300SX board. Included in this chapter is a list of features, a block diagram of the board, and a description of the feature set.

1.2 OVERVIEW

The 300SX board contains the following components:

- Intel 386SX™ central processing unit (CPU)
- Intel 387SX™ numeric coprocessor
- 0, 2, or 4M of random access memory
- 128K of read only memory
- 8237 direct memory access controllers
- 8259A programmable interrupt controllers
- 8254 programmable interval timer
- 6818 real time clock
- Intel 8742 keyboard and mouse controller

- Intel 82072 floppy disk controller
- AT hard disk interface
- Cirrus Logic VGA/EGA/CGA/Hercules/MDA video display controller
- Two 9-pin serial ports
- One 25-pin parallel port
- One PS/2 mouse port
- One 9-pin TTL video connector
- One 15-pin analog video connector
- Four 16-bit expansion slots
- One parallel interface controller

Figure 1-1 presents a functional block diagram of the 300SX board and Figure 1-2 shows the layout of the board

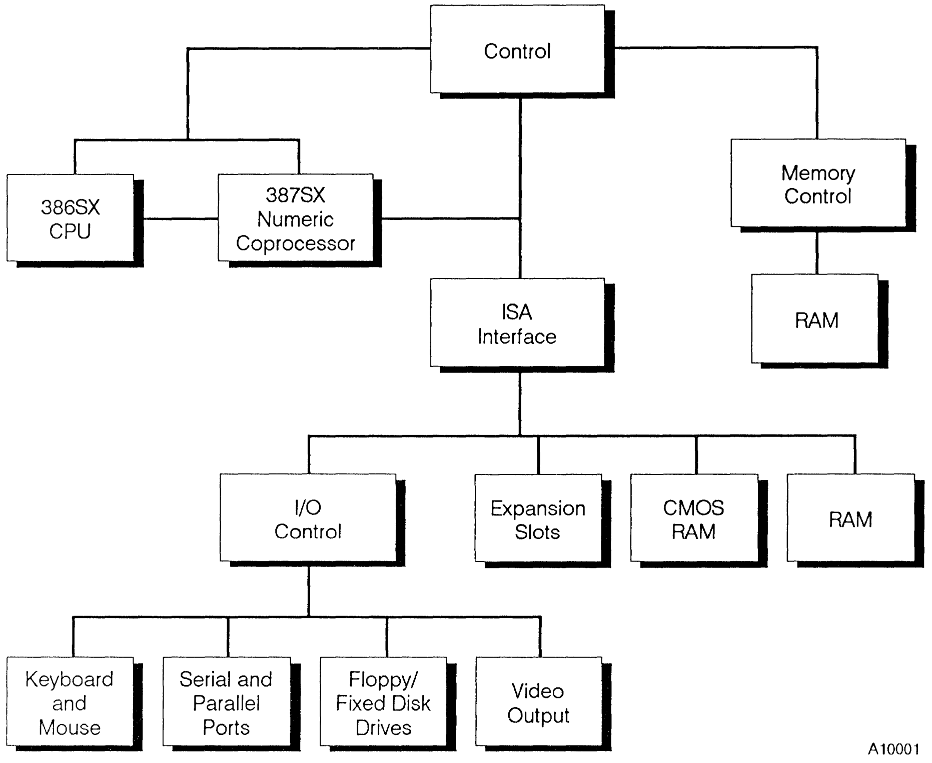


Figure 1-1. 300SX Board Functional Block Diagram

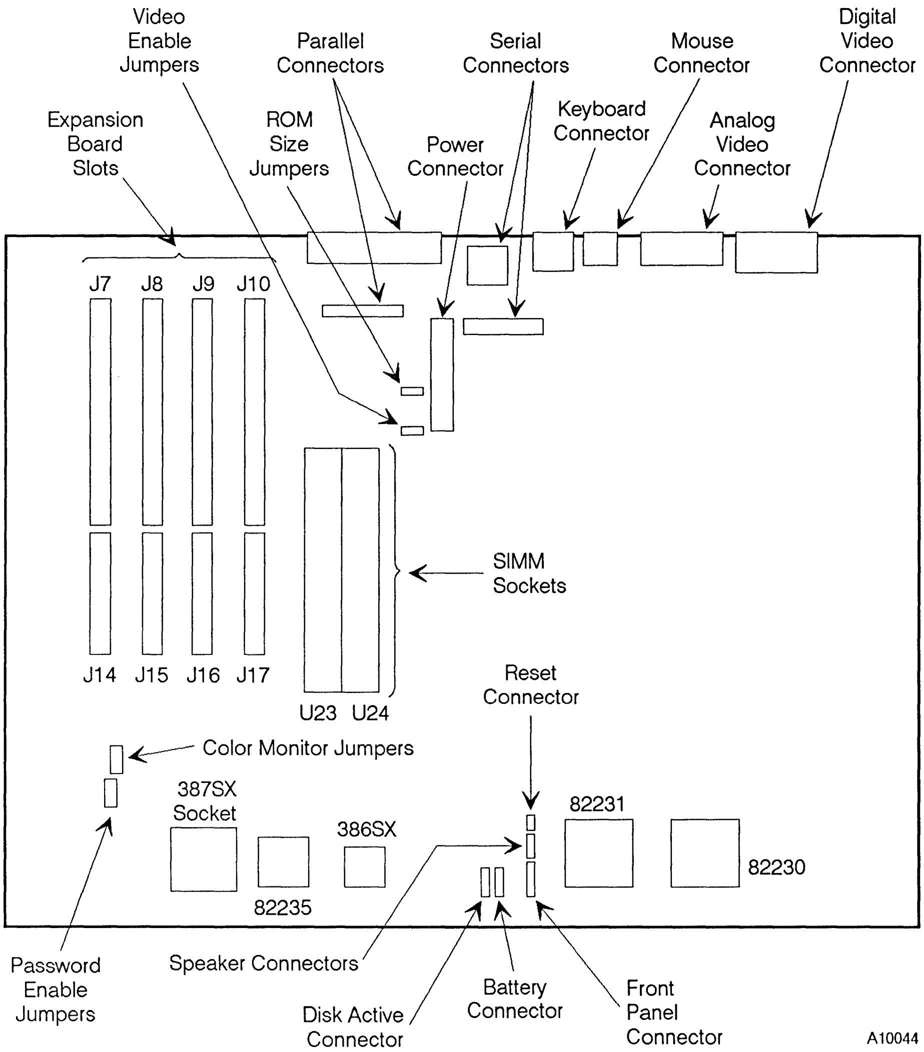


Figure 1-2. 300SX Board Layout

1.3 FEATURE SET DESCRIPTION

This section gives a detailed description of the features listed earlier.

1.3.1 Central Processing Unit (CPU)

The CPU is a 32-bit 386SX microprocessor. The 386SX incorporates multitasking support, pipelined architecture, memory management, address translation caches, and a high-speed bus interface. The CPU runs at a clock speed of 16 MHz (62.5 ns cycle time) with the system clock of 32 MHz.

The 386SX features 32-bit internal data paths, 16-bit external data paths, and directly outputs a 24-bit physical address. It also has a physical memory range of up to 16M (the maximum amount of addressable memory on the ISA bus).

1.3.2 Random Access Memory (RAM)

The 300SX board is available with 0, 2, or 4M of dynamic random access memory (DRAM).

1.3.3 Read Only Memory (ROM)

The board contains 128K of ROM, which contains a ROM-resident system and video basic input/output system (BIOS) and a power-on self test (POST) that performs automatic system diagnostics.

1.3.4 Direct Memory Access (DMA) Controller

The DMA controller, contained in the 82231 portion of the Intel 82230/82231 Chip Set, provides seven DMA channels for data transfer between main memory and I/O devices (typically disks or communications channels). The 82331 contains two 8237 DMA controllers.

1.3.5 Programmable Interrupt Controller (PIC)

Thirteen interrupt levels are provided by the 82230 portion of the 82230/82231 Chip Set. The 82230 includes two 8259A programmable interrupt controllers (PICs).

1.3.6 Programmable Interval Timer (PIT)

The 82231 portion of the 82230/82231 Chip Set provides an 8254 programmable interval timer (PIT). The 82231 generates timing for memory refresh, software timing control, and speaker frequency.

1.3.7 Real-time Clock (RTC)

The RTC, contained in the 82230 portion of the 82230/82231 Chip Set, combines a complete time-of-day clock with alarm, 100-year calendar, a programmable periodic interrupt, and 50 bytes of low power memory.

1.3.8 Keyboard and Mouse Controller

The keyboard and mouse controller supports a 101- or 102-key enhanced keyboard and a three-button mouse.

1.3.9 Floppy Disk Controller

The 300SX board uses an Intel 82072 for floppy disk control. The controller supports the following two types of floppy disk drives.

- 1.44M or 720K 3.5-inch floppy disk drive
- 1.2M or 360K 5.25-inch floppy disk drive

1.3.10 AT Fixed Disk Interface

The 300SX board provides an interface to a fixed disk controller. The interface supports the following types of AT interface fixed disk drives.

- 40M 3.5-inch fixed disk drive
- 80M 3.5-inch fixed disk drive
- 120M 3.5-inch fixed disk drive
- 170M 3.5-inch fixed disk drive

1.3.11 Video Display Subsystem

The video controller used on the 300SX board consists of a Cirrus Logic GD510A Graphics Attributes device, a Cirrus Logic GD520A Sequencer/CRT Controller device, and a digital-to-analog converter. The video controller provides MDA, CGA, EGA, VGA, extended VGA, and Hercules compatible modes.

1.3.12 Serial Port

The 300SX board is equipped with two 9-pin serial ports configured as COM1 and COM2.

1.3.13 Parallel Port

The 300SX board provides one parallel printer port configured as either LPT1 or LPT2.

1.4 OTHER FEATURES

- Fifty bytes of CMOS RAM and a battery backed-up power source for keeping system clock/calendar and system configuration parameters in permanent memory.
- Optional 387SX numeric coprocessor
- A keylock interface for prevention of unauthorized keyboard access
- Four 16-bit (36/62-pin) I/O expansion slots

Central Processing Core 2

2.1 INTRODUCTION

This chapter describes the central processing core of the 300SX board. The 386SX CPU, the 387SX numeric coprocessor, and the data and control buffers are discussed. For more detailed information on these components, refer to the Intel Microprocessor and Peripheral Handbook.

2.2 OVERVIEW

The central processing core contains the following components:

- A 386SX 32-bit CPU
- A 387SX numeric coprocessor
- Data and control buffers

Figure 2-1 diagrams the central processing core.

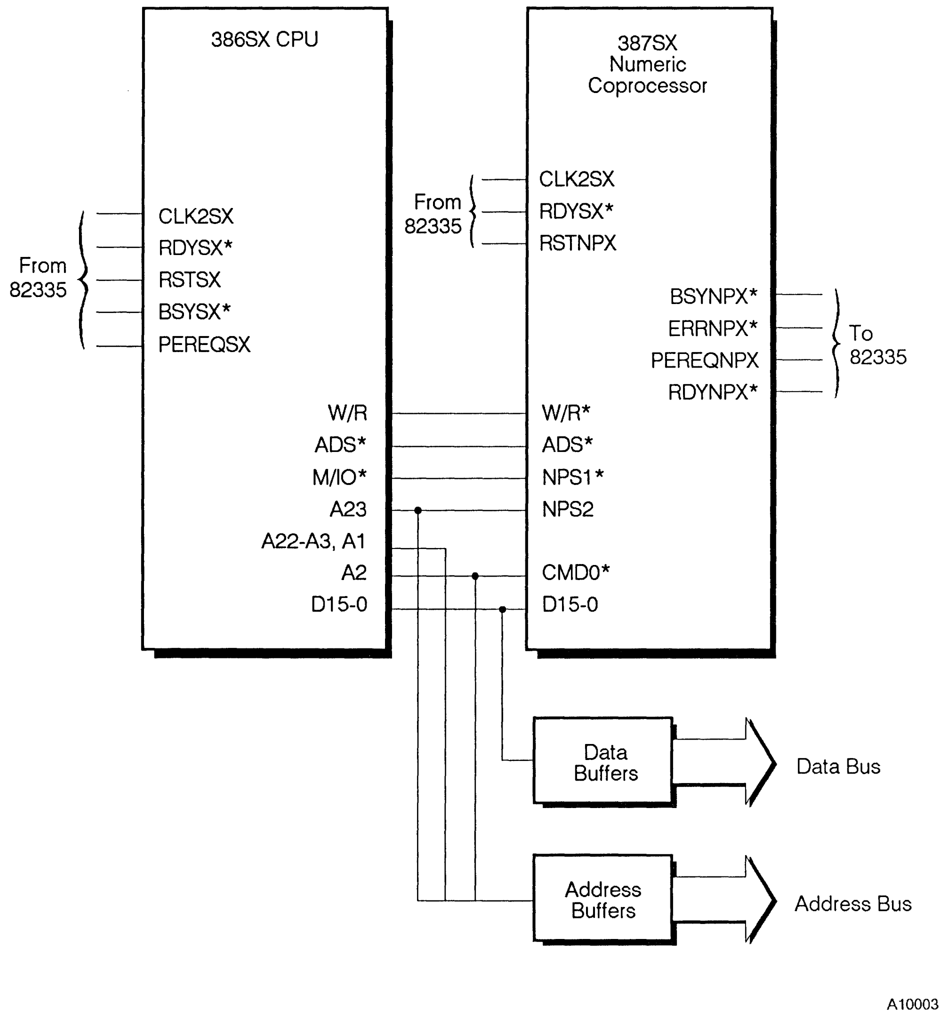


Figure 2-1. Central Processing Core

2.3 CPU

The CPU is an Intel 386SX microprocessor operating at 16 MHz with a clock cycle time of 62.5 ns. The CPU has separate 16-bit data and 24-bit address paths, plus on-chip memory management and protection. The CPU supports multi-user multi-tasking systems, memory management, virtual memory, and task isolation or memory isolation. The 386SX is functionally equivalent to the 386™ CPU, except that the 386SX CPU has a 16-bit external data bus.

2.3.1 Real Mode Architecture

Real mode is the default mode of the CPU upon reset or power up and has the same base architecture as the 8086, but provides access to a 32-bit register set. Real mode is compatible with 8086/8088 and 80286 CPUs at the object code level and has the same capabilities and limitations. In real mode, addressable physical memory is limited to 1M via segment registers, with a 64K limitation on segment size. Real mode does not provide memory protection features.

Real mode addresses are formed, as in the 8086, by combining the base address from a segment register with the offset value provided by the instruction. The CPU shifts the 16-bit base address value in the segment register left four bits, and adds the 16-bit offset value forming the 20-bit real address. For more information on real mode, refer to the Intel Microprocessor and Peripheral Handbook.

2.3.2 Protected Mode Architecture

Protected mode significantly increases the physical address space (to 16M) and allows the running of virtual memory programs of almost unlimited size (64 terabytes). In this mode, the integrated memory management and protection mechanism translates virtual addresses to physical addresses. It also isolates the operating system and enforces protection rules necessary for maintaining task integrity in a multi-tasking environment. This is useful in a multi-tasking and multi-user environment where resources are shared.

Protected mode provides memory paging, I/O protection, virtual-8086 mode, and a full 32-bit extended instruction set. Protected mode provides source code compatibility with the 8086/8088 and 80286 allowing direct execution of 16-bit applications at higher speeds. For more information on protected mode, refer to the Intel Microprocessor and Peripheral Handbook.

2.3.3 Virtual-8086 Mode

The virtual-8086 mode is an extension of the protected mode. Under this mode, the CPU provides compatibility with applications developed for the 8086/8088 while simultaneously providing a full 32-bit, large linear address programming environment in its protected mode. Virtual memory allows programs to overcome the limitation of physical memory.

The system divides virtual memory into many different segments. These segments are mapped into physical memory during virtual memory execution. The memory management system transfers code and data between physical memory and disk. For more information on virtual-8086 mode, refer to the Intel Microprocessor and Peripheral Handbook.

2.3.4 CPU Signals

This section defines the signal functions of the CPU. A signal name followed by an "I" in parentheses indicates an input signal. A signal name followed by an "O" in parentheses indicates an output signal. A signal name followed by "I/O" in parentheses indicates an input/output signal. For more detailed information on the CPU signal functions, refer to the Intel Microprocessor and Peripheral Handbook.

CLK2 (I)

CLK2 (system clock) provides the basic timing for the system. CLK2 is a 32-MHz clock divided by two inside the CPU. CLK2 generates the 16-MHz CPU clock.

D15:0 (I/O)

D15:0 (data bus lines) input data during read cycles of memory and I/O. D15:0 output data during write cycles of memory and I/O.

A23:1 (O)

A23:1 (address bus lines) output physical memory and I/O addresses.

BHE* AND BLE* (O)

BHE* and BLE* (byte enables) indicate which data bytes of the data bus take part in a bus cycle. BHE* (byte high enable) applies to D15:8 and BLE* (byte low enable) applies to D7:0. If both BHE* and BLE* are asserted, 16 bits of data are being transferred.

ADS* (O)

ADS* (address status) indicates the initiation of a valid bus cycle. It also indicates valid addresses are on the bus.

M/IO* (O)

M/IO* (memory or I/O select) distinguishes memory access from I/O access. When this signal is high, a memory or halt/shutdown cycle is in progress. When this signal is low, an I/O or interrupt acknowledge cycle is in progress.

D/C* (O)

D/C* (data/control) indicates a data access cycle when high, a control cycle when low.

W/R* (O)

W/R* (write/read) indicates a data write cycle when high, a data read cycle when low.

LOCK* (O)

LOCK* (locked bus) prevents access by an external coprocessor until the CPU completes a read or modifies a byte in memory.

READY* (I)

READY* (transfer acknowledge) indicates the current bus cycle has completed.

HOLD (I)

HOLD (bus hold request) indicates that some device other than the 386SX microprocessor requires bus mastership.

HLDA (O)

The CPU asserts HLDA (bus hold acknowledge) as a response to a HOLD request. When in the Hold Acknowledge state, the CPU tri-states its buses. When HOLD is no longer active, the CPU deactivates HLDA and drives the bus.

INTR (I)

INTR (interrupt request) requests the CPU to suspend program execution and service all external requests. The CPU samples the INTR line at the beginning of each processing cycle. INTR must be active at least two processing cycles before the current instruction ends.

NMI (I)

NMI (non-maskable interrupt) is an edge-triggered input that causes an interrupt which cannot be masked by software. NMI has the highest priority of all interrupts, and is generally not masked by software except during system initialization.

PEREQ (I)

PEREQ (coprocessor extension operand request and acknowledge) coordinates data transfer between the CPU and the numeric coprocessor. PEREQ requests the CPU to perform a data operand transfer to the numeric coprocessor.

BUSY* (I)

BUSY* (coprocessor extension busy) indicates a busy condition from the numeric coprocessor to the CPU. The CPU program execution stops as long as the signal remains active.

ERROR* (I)

ERROR* (coprocessor extension error) indicates an error condition of the numeric coprocessor to the CPU.

RESET (I)

RESET (system reset) clears the internal logic of the CPU. RESET initializes the CPU with a low-to-high transition.

2.3.5 Basic CPU Bus Operations

The CPU's bus control unit manages all bus operations. It generates the address, data, and command signals for the external memory and I/O operations, and transfers instructions to the instruction pre-fetch unit. Instructions are stored in a 16-byte queue while waiting for decoding and execution.

The instruction pre-decode unit receives and decodes the instructions from the pre-fetch queue. It places them in the decoded instruction queue for use by the execution unit.

The execution unit performs the basic processing functions. It accepts the decoded instructions from the instruction pre-decode unit and executes them. It uses the bus unit to fetch and store operands during the execution of instructions. The execution unit does not need to wait for the completion of a bus cycle before taking in a new instruction.

The address paging unit and segmentation unit provides memory management and protection services for the CPU. It also translates logical addresses into physical addresses for use by the bus unit. A register cache in the address unit contains the information used for performing the various memory translation and protection checks for each bus cycle.

The CPU uses a 32-MHz clock to control the bus timing. The CPU divides this clock by 2 to produce the internal processing clock which determines the bus cycles.

2.3.6 Pipelined Address Generation

The CPU supports address pipelining for main memory accesses. Address pipelining is the option of requesting the address and the bus cycle definition of the next internally pending bus cycle before the current bus cycle is acknowledged with READY* asserted. ADS* is asserted by the CPU when the next address is issued. The address pipelining option is controlled on a cycle-by-cycle basis with the NA* input signal. For more information on pipelined address generation, refer to the Intel Microprocessor and Peripheral Handbook.

2.4 NUMERIC COPROCESSOR

An Intel 387SX numeric coprocessor is socketed on the 300SX board. The numeric coprocessor uses the same clock generator as the CPU and fully supports single, double, and extended precision operations. The 387SX is upward object code compatible with the 8087 and 80287 numeric coprocessors and completely object code compatible with the 387 numeric coprocessor.

The 300SX board performs high speed mathematical calculations, logarithmic functions, and trigonometric operations using the numeric coprocessor. The numeric coprocessor operates in conjunction with the CPU.

The numeric coprocessor provides the 300SX board with additional data types, registers, instructions, and interrupts specifically designed to facilitate high-speed numeric processing. The registers in the numeric coprocessor hold constants and temporary results generated during calculations. Numeric coprocessor register space can be used as a stack or as a fixed register set.

2.4.1 387SX™ Data Types

The 387SX fully implements ANSI/IEEE 754-1985 standard for binary floating-point arithmetic. Table 2-1 lists the seven data types that the 387SX supports.

The 387SX directly extends the CPU instruction set. Extending the instruction set includes trigonometric, logarithmic, exponential, and arithmetic instructions for all data types.

Table 2-1. 387SX Data Types

Data Type	Precision	Range
Word integer	16 bits	$+10^{4}$
Short integer	32 bits	$+10^{9}$
Long integer	64 bits	$+10^{18}$
Packed BCD	18 digits	$+10^{18}$
Single precision	24 bits	$+10\mp^{38}$
Double precision	53 bits	$+10\mp^{308}$
Extended precision	64 bits	$+10\mp^{4932}$

2.4.2 387SX™ Programming Interface

The 387SX functions as an I/O device through the I/O ports using addresses 8000F8H and 8000FCH for sending opcodes and operands as well as receiving and storing results. The CPU outputs address 8000F8H when writing a command or reading status and outputs address 8000FCH when writing or reading data.

The CPU has two input signals for controlling data transfer to and from the 387SX: BUSY* and PEREQ (387SX request). These input signals connect to the corresponding 82335 pins.

The BUSY* signal informs the CPU that the 387SX is executing an instruction and cannot accept another. The CPU WAIT instruction informs the CPU to wait until the 387SX finishes execution.

The PEREQ signal indicates that the 387SX needs to transfer data to or from memory. Because the 387SX is never a bus owner, all input and output data transfers are performed by the CPU. PEREQ always goes inactive before BUSY* goes inactive.

The 387SX asserts an ERROR* signal after an instruction results in an error that is not masked by the 387SX control register. If an error occurs, ERROR* goes active before BUSY* goes inactive. As a result, the CPU receives an interrupt. If a higher priority task does not exist, the CPU services the interrupt.

The 387SX detects six different conditions that may occur during instruction execution. If the proper execution mask is not set by the control register, the 387SX asserts an ERROR* signal. The ERROR* signal generates a hardware interrupt holding the 387SX in a busy state until cleared by writing zeroes to address F0H.

The power-on self-test code in the system ROM enables the hardware interrupt. It also sets the hardware interrupt vector pointing to a routine in ROM. This routine clears the latch on the BUSY* signal and transfers control to the address pointed to by the non-maskable interrupt (NMI) vector. The NMI interrupt handler reads the 387SX status and determines if the 387SX caused the NMI. If it did not, control passes to the original NMI interrupt handler.

While the CPU executes numeric programs in either real or protected mode, interrupts report exception conditions (refer to Table 2-2).

All communication between the CPU and the 387SX is transparent to applications software. The 387SX operates whether the CPU executes instructions in real-address mode, protected mode, or virtual-8086 mode. The CPU handles all memory accesses. The 387SX operates on instructions and values passed to it by the CPU and is not aware of the processing mode of the CPU.

2.5 DATA AND CONTROL BUFFERS

Bidirectional buffers transfer control and data signals between the CPU bus and the AT/memory buses. Two buffers transfer CPU data (D15:0) and three buffers transfer CPU addresses (A23:1). Discrete logic provides buffer selection.

Table 2-2. Exception Conditions While Executing Numeric Programs

Interrupt Number	Cause of Interrupt
7	An ESC instruction was encountered when EM or TS of 386SX control register zero (CR0) was set. EM = 1 indicates that software emulation of the instruction is required. When TS is set, either an ESC or WAIT instruction causes interrupt 7. This indicates that the current numeric processor extension (NPX) context may not belong to the current task.
9	An operand of a coprocessor instruction wrapped around an addressing limit and spanned inaccessible addresses. The failing numeric instruction is not restartable; its address and data operand may be lost. The return address on the stack does not necessarily point to the failing instruction or to the following instruction. The interrupt can be avoided by never allowing numeric data to start within 108 bytes of the end of a segment.
13	The first word or double word of a numeric operand is not entirely within the limit of its segment. The return address pushed onto the stack of the exception handler points at the ESC instruction that caused the exception, including any prefixes. The 387SX has not executed this instruction; the instruction pointer and data pointer register refer to a previous, correctly executed instruction.
16	The previous numeric instruction caused an unmasked exception. The address of the faulty instruction and the address of its operand are stored in the instruction pointer and data pointer registers. Only ESC and WAIT instructions can cause this interrupt. The 386SX return address pushed onto the stack of the exception handler points to a WAIT or ESC instruction (including prefixes). This instruction can be restarted after clearing the exception condition in the NPX.

Intel 82335 High-Integration Interface Device

3

3.1 INTRODUCTION

This chapter describes the Intel 82335 High-Integration Interface device. An overview of the functions of the 82335 is presented, followed by descriptions of the internal functional blocks. This chapter also provides information on the 300SX board's dynamic random access memory (DRAM) resources. The chapter concludes with a description of the pin functions of the 82335.

3.2 82335 OVERVIEW

The Intel 82335 interfaces the 386SX microprocessor to the 387SX numeric coprocessor and to the Intel 82230/82231 chip set. The 82335 converts CPU bus cycles into 80286 compatible cycles, generates necessary clock signals, and provides dynamic memory control.

The 82335 is composed of the following seven functional blocks.

- Dynamic random access memory (DRAM) controller
- Address mapper/decoder
- Ready generator
- Bus cycle translator
- Numeric coprocessor interface

- Clock generator/reset synchronizer
- Parity generator/checker

See Figure 3-1 for a block diagram of the 82335 device.

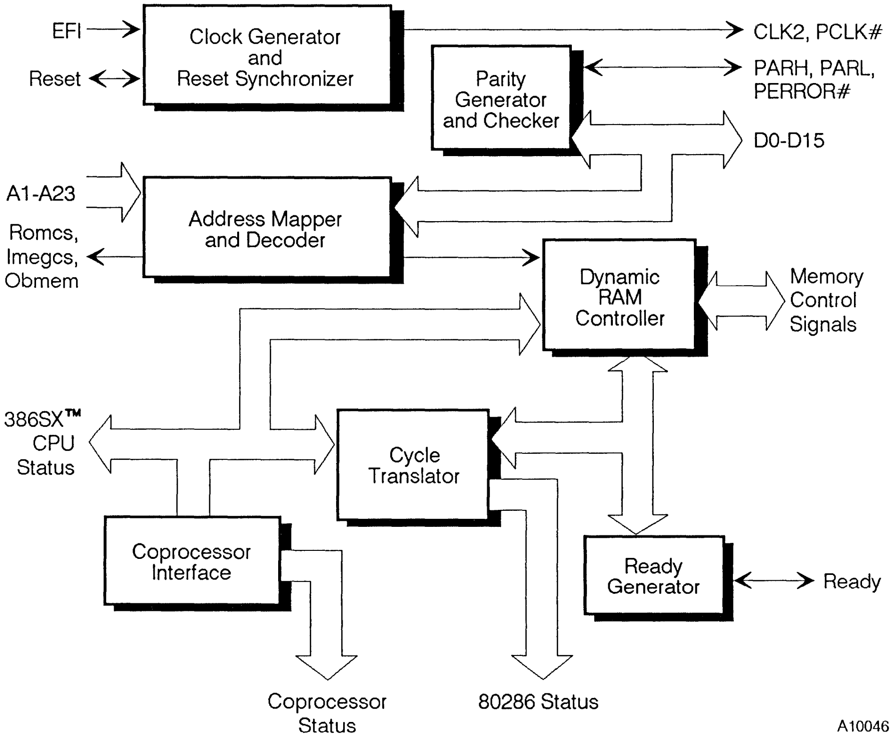


Figure 3-1. 82335 Block Diagram

3.3 DRAM CONTROLLER

The DRAM controller keeps track of CPU bus states and provides the necessary signals to address and refresh up to four 16-bit banks of 1Mb DRAMs. It has built in support for both paging and bank interleaving, and is configured for fast page 100 ns DRAM memory modes.

The DRAM controller can also be configured to operate in either turbo or non-turbo mode. In non-turbo mode, a fixed number of bus states (six per memory cycle) are used for all local memory accesses. The TURBO* input must remain static during local memory bus cycles.

3.3.1 DRAM Resources

The 300SX board has 2M of RAM installed, which can be expanded to 4M. Single in-line memory modules (SIMMs) are used for all system RAM.

ONBOARD RAM SIMMS

Onboard RAM SIMMs are small boards containing nine surface-mount dynamic random access memory (DRAM) chips. The SIMMs are organized in a 1Mb x 9 configuration. Each of the two SIMM sockets contain two SIMMs, for a maximum of four SIMMs on the system board. Refer to Appendix H for information on installing SIMMs.

3.3.2 DRAM Bank Configuration

The local DRAM system can be configured into one or two banks (2 to 4 SIMMs) of 1M x 16-bits each. Each 16-bit bank of memory is further divided into two 8-bit banks, low and high. Each 8-bit bank contains one extra bit for parity. The 300SX board contains a standard 2M of DRAM with an additional 2M optionally available (4M total configuration).

The BIOS firmware automatically determines the amount of onboard memory installed and configures the 82335 accordingly. The Setup program allows you to configure the memory between 512K and 1M. Figure 3-2 illustrates memory allocation for the system.

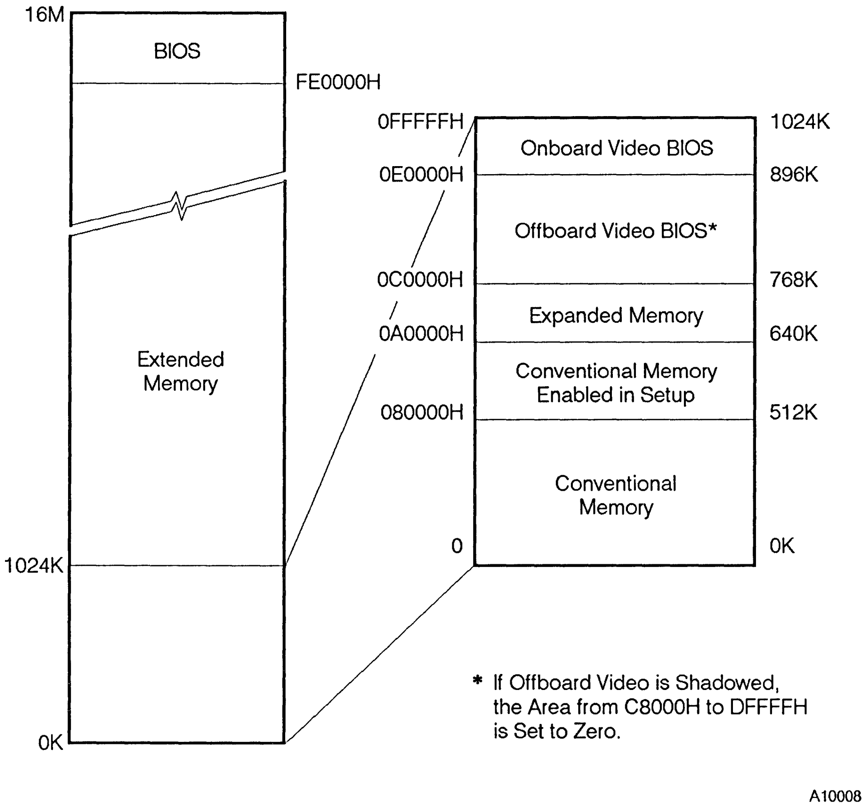


Figure 3-2. 300SX Memory Address Map

The 82335 internally decodes the address lines A23:1 from the CPU and outputs multiplexed row and column addresses (MA9:0), row address strobe (RAS), column address strobes (CASL and CASH), and write enable (WE) signals for local memory accesses. Each bank has separate RAS, CASL, and CASH signals (see Figure 3-3).

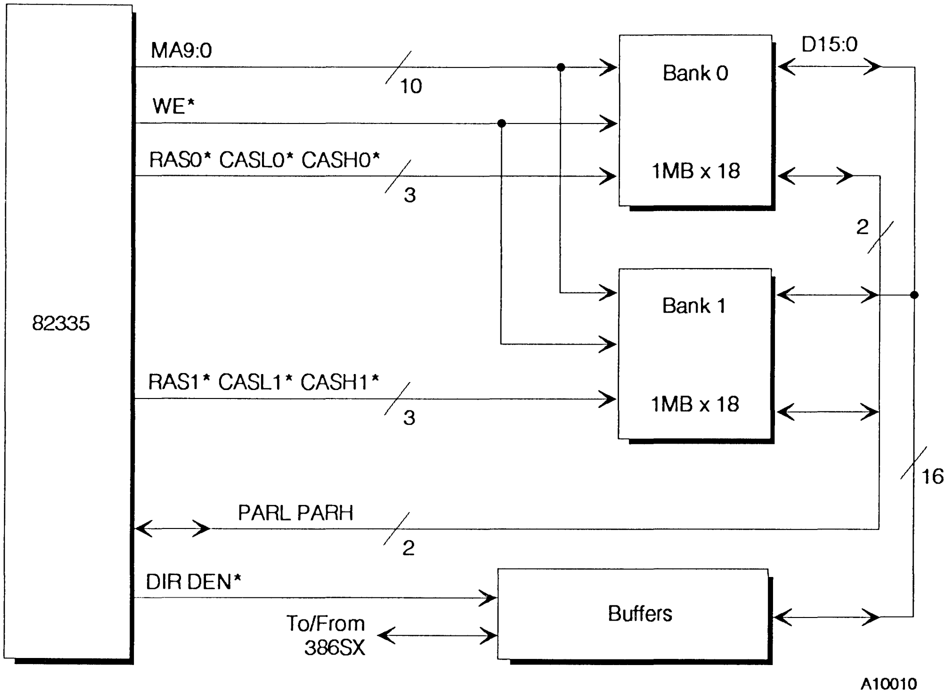


Figure 3-3. 82335 to DRAM Interface

3.3.3 Page-Mode DRAM Operation

A DRAM read or write access requires a row and column address. Successive memory accesses to the same row address and memory page (a page-hit memory access) only require a new column address. During a page-hit memory access, RAS* is kept active, and only a new CAS* is generated. Row addresses are taken from the higher order address bits, providing effective page-mode DRAM operation during successive memory accesses to the same page.

3.3.4 Page-Mode Bank Interleave Operation

A larger page size increases the chance of a page-hit memory access. One page of memory is kept active per bank. In a two bank DRAM configuration, a maximum of two pages of memory are kept active at a time. A successive memory access to an active page in a different bank (a page-hit-bank-miss) does not require a new row address and additional wait states. This increases the DRAM page size by a factor of two.

The 82335 has four built-in watch-dog timers to keep track of RAS* active time. Once timed out, the timer forces RAS* of the corresponding bank to be deactivated at the end of the memory cycle such that the maximum RAS* active time of the DRAM will not be violated.

3.3.5 Refresh Cycle

The 82335 generates its own DRAM refresh address with a 10-bit refresh counter. The counter increments by one every refresh cycle. During a refresh cycle, the refresh address appears on MA9:0, followed by the activation of RAS(3:0)* in staggered cycles.

3.4 ADDRESS MAPPING/DECODING OPTIONS

The address mapping and decoding options consist of ROM and video RAM shadowing, and address rolling. Selection of these options is done via programming of the configuration, roll compare, and address compare registers.

3.4.1 Shadowing

Shadowing refers to copying data from slow memory devices (ROM) to DRAM. Independent shadowing can be selected for the BIOS ROM, adapter ROM, and video RAM by programming the ROMEN*, ENADP*, and ENV* bits, respectively, of the configuration register. Figure 3-4 is a diagram of the shadow RAM address map.

BIOS ROM and adapter ROM contents must be copied to the shadow DRAM area before setting the lock bit in the configuration register. Once the lock bit is set, both DRAM areas are read only. Also, enabling video RAM shadowing and setting the video read only (VRO) bit in the configuration register causes this shadowed DRAM area to be read only. The Setup program allows you to control the shadowing of BIOS and onboard ROM or offboard ROM.

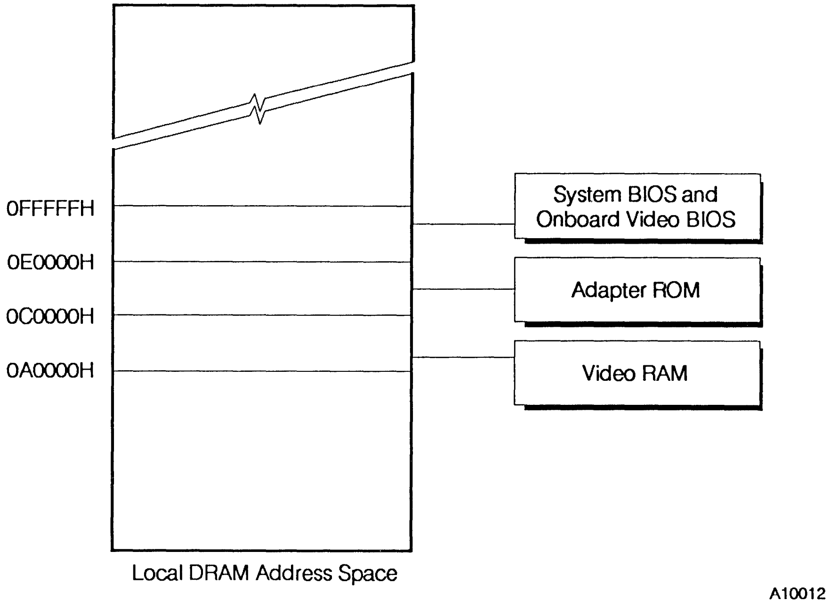


Figure 3-4. Shadow RAM Address Map

3.4.2 Roll Address Mapping

Roll address mapping is a method of utilizing DRAM memory space that may not otherwise be accessible. By not selecting BIOS ROM, adapter ROM, or video RAM shadowing, addresses generated in these areas access the ROMs or video RAM. Any local DRAM with the same physical addresses as the ROMs or video RAM cannot be directly addressed. To allow access to this DRAM space, the 82335 re-maps or rolls logical addresses above the top of the available memory in 128K segments.

There are four 128K segments of physical memory (512K total) that can be re-mapped. These segments are in the top half of the lowest megabyte of memory (see Figure 3-1). The programmed memory configuration register bits (ROMEN*, S640, ENADP*, and ENV*) control which segments are available for re-mapping. The programmed roll compare registers enable roll address mapping and specify the logical addresses to be re-mapped. Figure 3-5 diagrams a memory address map illustrating roll address mapping. The Setup program, described in Chapter 11, allows you to control address rolling.

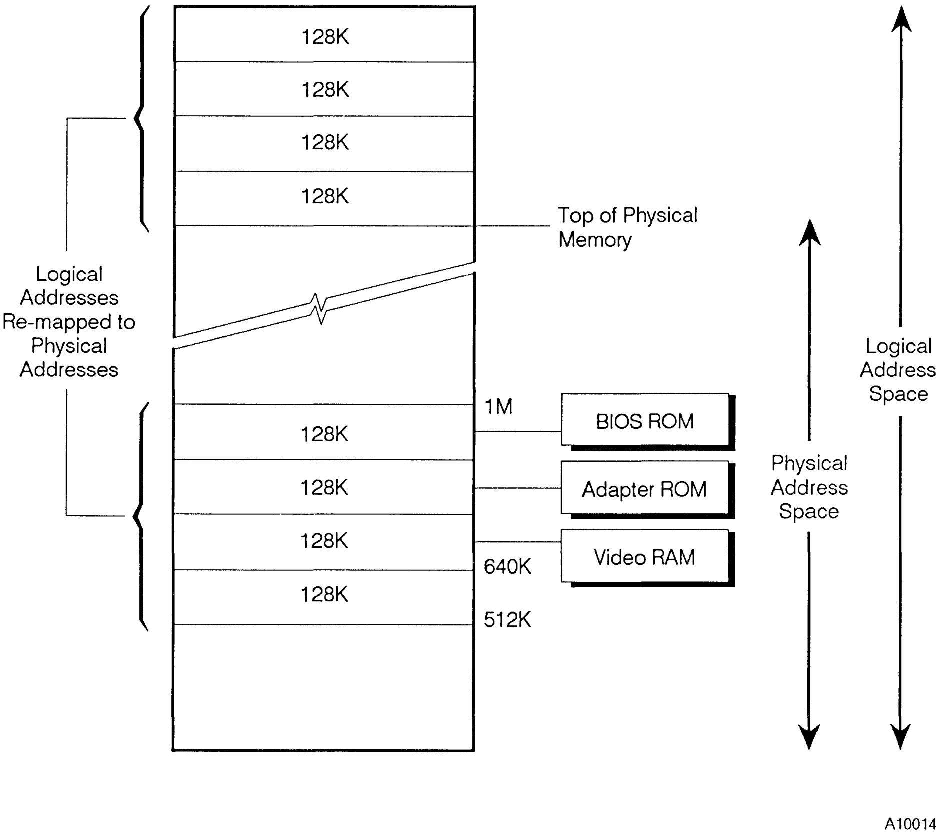


Figure 3-5. Roll Address Mapping Range

3.4.3 Address Mapping and DRAM Control Registers

There are five registers in the 82335 that control the operation of the address mapping and DRAM control options.

- Memory configuration register (MCR)
- Roll compare registers (RC1 and RC2)
- Address range compare registers (CC0 and CC1)

Each register resides in the local I/O space of the 82335 and is read/write active until the LOCK bit is set in the memory configuration register.

MEMORY CONFIGURATION REGISTER

The memory configuration register (MCR), located at I/O address 22H, selects a number of address mapping and DRAM control options. At reset, all register bits are set to zero. Table 3-1 lists the MCR bit assignments.

Table 3-1. **Memory Configuration Register Bit Assignments**

Bit	Name	Function
0	ROMEN*	0 = Enable BIOS ROM/EPROM accesses (0E0000H-0FFFFFH) 1 = Disable BIOS ROM/EPROM accesses, shadow enabled
3	S640	0 = Base memory size is 512K 1 = Base memory size is 640K
4	DSIZE	0 = 256K x 1 or 256K x 4 DRAMS installed 1 = 1M x 1 DRAMs installed
7,6	INTERL	00 = 1 memory bank installed 01 = 2 memory banks installed 10 = 3 memory banks installed 11 = 4 memory banks installed
8	ROMSIZE	0 = 256K ROM/EPROM 1 = 512K ROM/EPROM
9	ENADP*	0 = Enable adapter ROM/EPROM accesses (0C0000H-0DFFFFH) 1 = Disable adapter ROM/EPROM accesses, shadow enabled
10	ENV*	0 = Enable video RAM accesses (0A0000H-0BFFFFH) 1 = Disable video RAM accesses, shadow enabled
11	VRO	0 = Video area read-write 1 = Video area read-only
15	LOCK	0 = Enable all configuration register accesses 1 = Disable all configuration register accesses

ROMEN* Bit

The ROMEN* bit (bit 0) enables or disables shadowing of the BIOS ROM to the memory address range E0000H - FFFFFH. Setting the ROMEN* bit to 1 enables BIOS ROM shadowing. Accesses to this memory address range are from the DRAM. Setting the ROMEN* bit to 0 disables BIOS ROM shadowing. Accesses to this memory address range are from the BIOS ROM. Disabling adapter ROM shadowing allows this memory address range to be remapped above the top of the physical address space. Since onboard video ROM resides in memory address range 0E0000 - 0EFFFFH, setting the ROMEN* bit to 1 also shadows onboard video ROM.

S640 Bit

The S640 bit (bit 3) selects the base memory size. When set to 0, the S640 bit indicates a base memory size of 512K. When set to 1, the S640 bit indicates a base memory size of 640K.

DSIZE Bit

The DSIZE bit (bit 4) indicates the type of DRAM installed. For the 300SX board, the DSIZE bit is set to 1 (1Mb x 1 DRAM installed).

INTERL Bits

The INTERL bits (bits 6 and 7) indicate the number of memory banks installed. For the 300SX board, the INTERL bits are set to 00 (one memory bank installed). When an additional bank of memory is installed (2M), these bits are set to 01.

ROMSIZE Bit

The ROMSIZE bit (bit 8) indicates the size of the installed ROM. For the 300SX board, the ROMSIZE bit is set to 1 (512K-bit ROM installed).

ENADP* Bit

The ENADP* bit (bit 9) enables or disables shadowing of the adapter ROM to the memory address range 0C0000H - 0DFFFFH. Setting the ENADP* bit to 1, enables adapter ROM shadowing. Accesses to this memory address range are from the DRAM. Setting the ENADP* bit to 0, disables adapter ROM shadowing. Accesses to this memory address range are from the ROM. Disabling adapter ROM shadowing allows this memory address range to be re-mapped above the top of the physical address space.

ENV* Bit

The ENV* bit (bit 10) enables or disables shadowing of the system video RAM to the memory address range A0000H - BFFFFH. Setting the ENV* bit to 1, enables video RAM shadowing. Accesses to this memory address range are from the DRAM. Setting the ENV* bit to 0, disables video RAM shadowing. Accesses to this memory address range are from the video RAM. Disabling video RAM shadowing allows this memory address range to be re-mapped above the top of the physical address space.

VRO Bit

During video RAM shadowing, the VRO bit (bit 11) selects either read/write access or read only access from the video RAM area (A0000 - BFFFFH). When set to 1, the VRO bit allows read only access to the video RAM area. When set to 0, the VRO bit allows read/write access to the video RAM area.

LOCK Bit

The LOCK bit (bit 15) enables or disables external access to the configuration, roll compare, and address range compare registers. Setting the LOCK bit to 0, creates the following conditions:

- The configuration, roll compare, and address range compare registers are read/write active at even I/O addresses from 22H - 2BH.
- The status outputs S0* and S1* are not generated for these even I/O addresses.
- The shadowing DRAM area is available for both reading and writing.

Setting the LOCK bit to 1, creates the following conditions:

- The configuration, roll compare, and address range compare registers are not accessible.
- The status outputs S0* and S1* are generated for I/O addresses between 22H and 2BH.
- The shadowing DRAM area is read only with the exception of A0000H to BFFFFH if the VRO bit is cleared.

Once the LOCK bit is set to 1, the configuration, roll compare, and address compare registers can only be accessed by resetting the system.

When writing to the 82335 registers, all bits are written over. Care should be taken to insure that all data bits written are set for the intended operation.

ROLL COMPARE REGISTERS

There are two roll compare registers, RC1 and RC2, located at I/O addresses 24H and 26H, respectively. These registers re-map logical addresses into physical addresses above the physical address space. Table 3-2 lists the RC1 and RC2 bit assignments.

Table 3-2. Roll Compare Registers Bit Assignments

Bit	Name	Function
0	EN	Enables roll address mapping
7-1	Address Mask	Selects address bits to be included in re-mapping comparison (M23:17)
15-9	Compare Data	Selects address range to be re-mapped (C23:17)

In roll address mapping, the output of the roll comparator activates the address mapper and causes an address roll-over. This is accomplished by using the address bits A23:17, and the three outputs from each roll compare register: compare enable, address mask, and compare data. Each time the 82335 receives a new address, it compares the address bits A23:17 with the compare data C23:17 using M23:17 as a mask.

ADDRESS RANGE COMPARE REGISTERS

There are two address range compare registers, CC0 and CC1, located at I/O addresses 28H and 2AH, respectively. They are used to decode the onboard memory address range. Table 3-3 lists the CC0 and CC1 bit assignments. At reset, CC0 is enabled and CC1 is disabled.

Table 3-3. Address Range Compare Registers Bit Assignments

Bit	Name	Function
0	EN	Enables address range comparison
7-3	Address Mask	Selects address bits to be included in address range comparison (M23:19)
15-11	Compare Data	Specifies top of address range (C23:19)

The onboard memory address range is decoded in a similar manner as the roll-over address range. Each address range comparator accepts address inputs A23:19, compare data C23:19, mask data M23:19, and enable bit EN. Each comparator compares A23:19 with C23:19 using M23:19 as a mask.

3.4.4 Chip Select Signals

The address mapper/decoder uses the configuration, roll compare, and address range compare register contents along with input addresses to generate the following output signals:

ROMCS0*	ROM 0 chip select
ROMCS1*	ROM 1 chip select
LMEGCS*	Lower meg chip select
OBMEM	Onboard memory address range

The ROM chip select signals are functions of the ROMSIZE, and ROMEN* bits in the configuration register as well as the input address. Enabling ROM shadowing (ROMEN* = 1) disables the ROM chip select outputs. Disabling ROM shadowing activates the ROM chip select outputs as follows:

If ROMSIZE = 0 (256K ROM)

ROMCS0* decodes the address ranges 0E0000H:EFFFFH and FE0000H:FEFFFFH.
ROMCS1* decodes the address ranges 0F0000H:0FFFFH and FF0000H:FFFFFFH.

If ROMSIZE = 1 (512K ROM)

ROMCS0* decodes the address ranges 0E0000H:0FFFFFH and FE0000H:FFFFFH.
ROMCS1* is inactive.

The lower meg chip select output (LMEGCS) is a function of both the input address and M/IO* inputs. Decoding a memory address within the first megabyte of memory activates LMEGCS. LMEGCS is inactive during I/O cycles.

The onboard memory output (OBMEM) is a function of the address range comparators, roll comparators, and the bits in the configuration register (ROMEN*, ROMSIZE, DSIZE, and S640). OBMEM differentiates local DRAM access from system RAM, ROM, or I/O accesses. During system memory addressing, the OBMEM output is inactive and the NA* output is deactivated to extend the CPU address long enough to be latched onto the system address bus.

3.5 READY GENERATOR

The 82335 indicates completion of the current bus cycle to the CPU with the READYSX* signal. The ready generator determines the appropriate number of wait states (if any) to insert and activates the READYSX* output at the correct time. Zero wait states occur during a page-hit bank-hit and a page-hit bank-miss. Two wait states occur during a page-miss bank-hit and a page-miss bank-miss. One wait state occurs during a new RAS cycle.

The ready generator has three external inputs: READY286*, READYNPX*, and EXTRDY. The READY286* input, driven by the 82230 READY* pin, identifies the completion of system bus cycles. The READYNPX* input indicates the completion of numeric coprocessor bus cycles. The level-triggered EXTRDY input directly gates READYSX*. The READYSX* output is held inactive until the EXTRDY input is sampled active. EXTRDY extends bus cycles when using slow peripherals or off-board memory. Setup and hold times for these inputs must be met to guarantee correct operation.

3.6 BUS CYCLE TRANSLATOR (BCT)

The 82335 has a built-in interface unit that translates 386SX control signals to 80286 control signals. The bus cycle translator (BCT) identifies the bus cycle being performed, monitors the CPU T-states, and outputs 80286-like bus control signals to the 82330/82331 chip set and other components on the system board.

As the BCT monitors control inputs from both the CPU and the 82231, it determines what type of cycle is being requested. The bus tracker monitors the timing of the bus cycle and simultaneously outputs 80286-type bus control signals when one of the following cycles is being requested:

- I/O access
- System memory access
- Halt/shutdown
- Interrupt

The control signals output are S0*, S1*, and M/IO286*. The S0* and S1* outputs are not activated for local memory accesses, or for accesses to the 82335 device I/O.

In addition to controlling status output to the 82230/82231 chip set, the BCT also controls the hold request (HRQSX) input to the CPU. The BCT translates a hold request signal from the 82230 (HRQ286) into a 386SX processor-compatible output (HRQSX). The CPU responds with a hold acknowledge (HLDASX) to the 82335. The 82335 translates this signal into a HLDA output to the 82230/82231 chip set.

3.7 NUMERIC COPROCESSOR INTERFACE

The 82335 automatically senses when a numeric coprocessor is installed. If a numeric coprocessor is not installed, REFRESH* toggles BUSYSX* and PEREQSX is forced low. The numeric coprocessor interface is diagrammed in Figure 3-6.

A numeric exception latches BUSYSX* low and forces PEREQSX high. This holds processing on the CPU until numeric coprocessor transfers are completed. The ERROR* output from the numeric coprocessor becomes active, causing the 82230 to issue an interrupt request (IRQ13). The interrupt handler performs an IOWR cycle to address 0xF0H, clearing the BUSY latched hardware. The interrupt handler clears the numerics ERRORNPX signal and normal operation resumes.

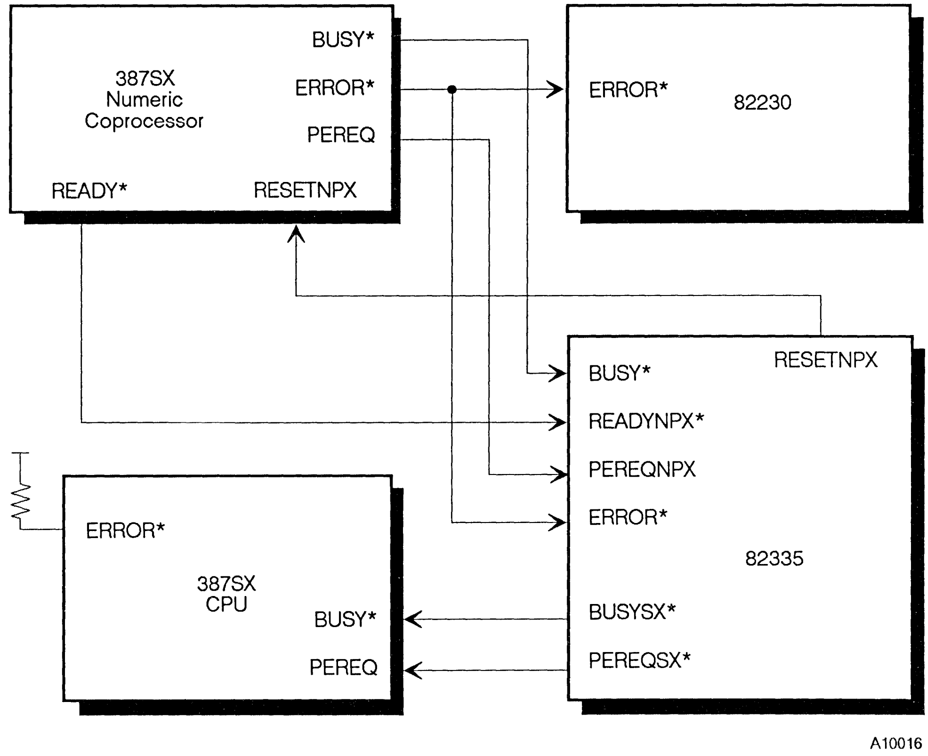


Figure 3-6. Numeric Coprocessor Interface

3.8 CLOCK GENERATOR/RESET SYNCHRONIZER

The clock generator synchronizes the CPU, numeric coprocessor, and peripherals by converting an input frequency into system clock outputs (CLK2 and PCLK*). A 32-MHz external oscillator, connected to the 82335 EFI pin, provides the input frequency. The internally buffered EFI input is output to the CPU and numeric coprocessor through the CLK2 output. The EFI input, divided by two (16 MHz), also provides the clock output (PCLK*) for the 82230/82231.

The reset synchronizer receives the RESETCPU and SYSRESET inputs from the 82230 and generates the synchronous outputs RESETSX and RESETNPX. When the SYSRESET signal is activated, both the RESETSX and RESETNPX outputs are asserted to reset the entire system. Activating the RESETCPU input generates the RESETSX output and resets the CPU. RESETCPU and SYSRESET are asynchronous inputs.

3.9 PARITY GENERATOR/CHECKER

The 82235 has a built in parity generator and checker to maintain data integrity for the DRAM. The parity generator/checker has data bus input/outputs D15:0, and two parity input/output pins, parity high byte (PARH) and parity low byte (PARL). The three-state PARH and PARL I/O pins directly drive the DRAMs without data transceivers.

During memory write cycles, the internal parity generator evaluates the D15:0 pins, and outputs PARH and PARL to the DRAM. During memory read cycles, combined DRAM data (D15:0) and parity outputs (PARH and PARL) are checked for parity errors. A detected parity error asserts parity error pin PERROR*.

A parity register bit, provided within the 82335, resets the PERROR* output. Bit 2 of the I/O address 61H is the PARCHKEN (parity check enable) bit. Programming the PARCHKEN bit to 0 enables local DRAM parity checking. The register at I/O address 61H is write-only and cannot be read.

3.10 82335 SIGNALS

This section describes the signals present at the I/O pins of the 82335. A name followed by an "I" in parenthesis, such as ADS* (I), means the signal is an input. Names followed by an "O" in parenthesis, such as CLK2 (O), are outputs. Names followed by "I/O", such as D15:0 (I/O), are input/output signals.

3.10.1 A(23:1) (I)

A23:1 (address inputs) select the dynamic RAM address for a memory read or write operation.

3.10.2 A20GATE (I)

The keyboard/mouse controller uses the A20GATE input to force A20 low. When A20GATE is low, A20 is forced low internal to the 82335 during CPU memory cycles (not DMA or master). When A20GATE is high, A20 follows the CPU address input from the A20 pin.

3.10.3 ADS* (I)

ADS* (address status) indicates a valid, CPU driven, bus cycle definition and address (W/R*, M/IO*, D/C*, BLE*, BHE*, and A23:1).

3.10.4 BHE* (I)

BHE* (byte high enable) indicates when data is being transferred on D15:8 (the physical address of the high byte of a 16-bit data word).

3.10.5 BLE* (I)

BLE* (byte low enable) indicates when data is being transferred on D7:0 (the physical address of the low byte of a 16-bit data word).

3.10.6 BUSYNPX* (I)

BUSYNPX* indicates a busy numeric coprocessor. BUSYNPX* is connected directly to BUSY* of the numeric coprocessor.

3.10.7 BUSYSX* (O)

BUSYSX* indicates a busy numeric coprocessor. BUSYSX* is connected directly to BUSY* of the CPU.

3.10.8 CASH(3:0)* (O)

CASH(3:0)* (column address strobe high byte) is used by the high byte of the DRAM array to latch the column address present on the MA9:0 pins. CASH(3:0)* directly drives the DRAM array.

3.10.9 CASL(3:0)* (O)

CASL(3:0)* (column address strobe low byte) is used by the low byte of the DRAM array to latch the column address present on the MA9:0 pins. CASL(3:0)* directly drives the DRAM array.

3.10.10 CLK2 (O)

CLK2 (32 MHz) drives the CPU and numeric coprocessor input clocks. The CPU and the numeric coprocessor internally divide CLK2 by two.

3.10.11 D/C* (I)

D/C* (data/control select) distinguishes between data and control bus cycles.

3.10.12 DEN* (O)

DEN* (data enable) enables the transfer of data between the DRAM array and the local data bus.

3.10.13 DIR (O)

DIR (direction) controls the direction of the data transceivers. An active DIR indicates data is being written into memory.

3.10.14 D15:0 (I/O)

During initialization of the 82335, D15:0 (data bus) write and read control words to and from the internal memory configuration registers. The 82335 uses D15:0 for parity generation and checking of data transferred between the bus and the DRAM array.

3.10.15 EFI (I)

EFI (external frequency in), driven by an external oscillator (32 MHz), generates the CLK2 and PCLK* output clocks. The EFI input also drives all internal 82335 logic.

3.10.16 ERROR* (I)

The ERROR* input indicates when a numeric coprocessor error has occurred. ERROR* connects directly to the ERROR* output of the numeric coprocessor and the ERROR* input of the 82230.

3.10.17 EXTRDY (I)

EXTRDY (external ready) inserts additional wait states into CPU bus cycles. Deactivation of EXTRDY during a bus cycle delays the active edge of the 82335 READYSX* output until the EXTRDY input is sampled active.

3.10.18 FM and MMS (I)

The FM and MMS inputs select the DRAM operating mode. FM and MMS are tied high, which selects the mode for fast 100 ns DRAM. This mode allows up to four pages to be active simultaneously.

3.10.19 HLDA (O)

HLDA (hold acknowledge) indicates the CPU has relinquished control of the local bus. It is asserted in response to activation of the HLDASX input from the CPU.

3.10.20 HLDASX (I)

The CPU asserts HLDASX (hold acknowledge SX) in response to the assertion of the CPU's HOLD pin; indicating the CPU has relinquished control of the bus.

3.10.21 HRQ286 (I)

The 82231 drives HRQ286 (CPU hold request) when requesting DMA or refresh cycles.

3.10.22 HRQSX (O)

HRQSX (CPU hold request) drives the CPU's HOLD input. HRQSX is the HRQ286 input synchronized to CLK2.

3.10.23 LMEGCS* (O)

LMEGCS* (lower meg chip select) decodes A23:20 and M/IO* for memory accesses in the lowest megabyte of the 386SX address range.

3.10.24 MA9:0 (O)

MA9:0 (multiplexed address) provides the row and column addresses for CPU or DMA access, and row addresses for refresh access to the DRAM array.

3.10.25 MEMR* (I)

MEMR* (memory read command) indicates an 82230 memory read cycle. The 82235 uses this signal for memory control during DMA cycles.

3.10.26 MEMW* (I)

MEMW* (memory write command) indicates an 82230 memory write cycle. The 82235 uses this signal for memory control during DMA cycles.

3.10.27 M/IO* (I)

M/IO* (memory/IO select) distinguishes between memory and I/O accesses.

3.10.28 M/IO286* (O)

M/IO286* (memory I/O select 286) emulates the M/IO* output of the 80286. The 82230/82231 and other system peripherals use M/IO286* to distinguish memory access from I/O access. It also indicates halt/shutdown and interrupt acknowledge cycles.

3.10.29 NA* (O)

NA* (next address) controls the address pipelining of the CPU. Asserting NA* validates the address of the next bus cycle in the T2P state of the current bus cycle. The 82335 asserts this signal during local DRAM bus cycles.

3.10.30 OBMEM (O)

OBMEM (onboard memory) indicates the current bus cycle is to local DRAM. The 82235 internally decodes the A23:1 and the M/IO* and D/C* inputs.

3.10.31 PARH (I/O)

PARH (parity high byte) is the upper byte parity bit of data on the bus (D15:8). For memory write cycles, the 82335 outputs the internally generated parity bit to the DRAM array via the PARH pin. During a memory read, the 82335 uses the data received at PARH to validate the upper byte of data from the DRAM array.

3.10.32 PARL (I/O)

PARL (parity low byte) is the lower byte parity bit of data on the local bus (D7:0). Its function is identical to the PARH pin.

3.10.33 PCLK* (O)

PCLK* (processor clock) drives the 82230 X3 pin. PCLK* is the EFI input (32 MHz) divide by two (16 MHz), and is equivalent to the 2X clock of an 8 MHz 80286.

3.10.34 PEREQNPX (I)

PEREQNPX (processor extension request NPX) indicates the numeric coprocessor requires a data transfer.

3.10.35 PEREQSX (O)

PEREQSX (processor extension request SX) requests the CPU to transfer data to or from the numeric coprocessor.

3.10.36 PERROR* (O)

PERROR* (parity error) indicates that the 82235 has detected a parity error in either the upper or lower byte of data from the DRAM array.

3.10.37 RAS(3:0)* (O)

RAS(3:0)* (row address strobe) is used by the DRAM array to latch the row address present on the MA9:0 pins. RAS(3:0)* supports up to a four-way interleaved DRAM configuration with page-mode access and drives the DRAM array directly and needs no external drivers.

3.10.38 READY286* (I)

READY286* indicates a completed I/O bus cycle.

3.10.39 READYNPX* (I)

READYNPX* indicates a completed numeric coprocessor bus cycle.

3.10.40 READYSX* (O)

READYSX* indicates a completed current bus cycle to the CPU. Its output is a function of the internally generated memory ready signal, and the READY286*, READYNPX*, EXTRDY, and TURBO* inputs.

3.10.41 REFRESH* (I)

REFRESH* notifies the DRAM controller that the DRAM array requires refresh.

3.10.42 RESETCPU (I)

RESETCPU is activated during system power up, or when the CPU generates a HALT status.

3.10.43 RESETNPX (O)

RESETNPX drives the RESETIN pin of the 387SX. It is active only during system power up or a front-panel reset.

3.10.44 RESETSX (O)

RESETSX drives the RESET pin of the 386SX processor. RESETSX is the logical OR of the SYSRESET and RESETCPU inputs.

3.10.45 ROMCS(1:0)* (O)

ROMCS(1:0)* (ROM chip select) selects the ROMs during system initialization. Once the ROM contents are copied into the DRAM space, the ROMCS* outputs are disabled and the ROM addresses are mapped into the DRAM physical address space by the 82335.

3.10.46 S(1:0)* (O)

S(1:0)* (bus cycle status) indicates the initiation of a system bus cycle and, along with M/IO286*, defines the type of bus cycle.

3.10.47 SYSRESET (I)

SYSRESET is driven active during system power up or a front-panel reset.

3.10.48 TEST(1:0) (I)

TEST(1:0) (test mode) is reserved for special test modes. These test inputs are pulled inactive during normal operations.

3.10.49 TURBO* (I)

When asserted, (turbo mode select) TURBO* allows the CPU local bus to run at maximum performance. Deactivating the TURBO* input causes the 82335 READY generation logic to insert additional wait states into each bus cycle. In the non-turbo mode, CPU performance approximates 8 MHz 80286 bus efficiency.

3.10.50 WE* (O)

WE* (write enable) is used by the DRAM array to enable input for a write operation. WE* drives 4M of DRAM with no additional buffering.

3.10.51 W/R* (I)

W/R* (write/read select) distinguishes between read and write cycles from the CPU.

Intel 82330/82331 Chip Set

4

4.1 INTRODUCTION

This chapter describes the Intel 82230/82331 Chip Set. These two devices work together to provide a high integration PC AT interface on the 300SX board. The chapter begins with an overview of the 82230/82331 Chip Set and concludes with a detailed description of these devices. For additional information on the 82230/82231 Chip Set, refer to the Intel Microprocessor and Peripheral Handbook.

4.2 82230/82331 OVERVIEW

The Intel 82230/82231 is a high integration chip set (see Figures 4-1 and 4-2). The 82230 performs the functions of the following devices:

- A 6818 real time clock/RAM
- Two Intel 8259A programmable interrupt controllers
- An Intel 82284 clock generator and ready interface
- An Intel 82288 bus controller

The 82231 performs the functions of the following devices:

- An Intel 8254 programmable interval timer
- Two Intel 8237 direct memory access controllers
- A 74LS612 memory mapper
- An Intel 8284A clock generator

For information on the 82284, 82288, and the 8284A, refer to the Intel Microprocessor and Peripheral Handbook. For information on the 74LS612, refer to the manufacturer's data sheet.

4.3 6818 REAL TIME CLOCK (RTC)

The 300SX board uses an RTC module of the 6818 as its real-time clock and configuration memory. The 6818 is contained in the Intel 82230 portion of the Intel 82230/82231 Chip Set. The RTC module combines a complete time-of-day clock with alarm, 100-year calendar, a programmable periodic interrupt, and 50 bytes of low power static random access memory (SRAM) for the user. System provisions allow the RTC to operate in a low power mode and protect the contents of both the RAM and clock during system power-up and power-down. The battery maintains clock and calendar information in the RAM. The system does not charge the battery. If the battery fails, it must be replaced. Figure 4-3 illustrates the memory map for the RTC.

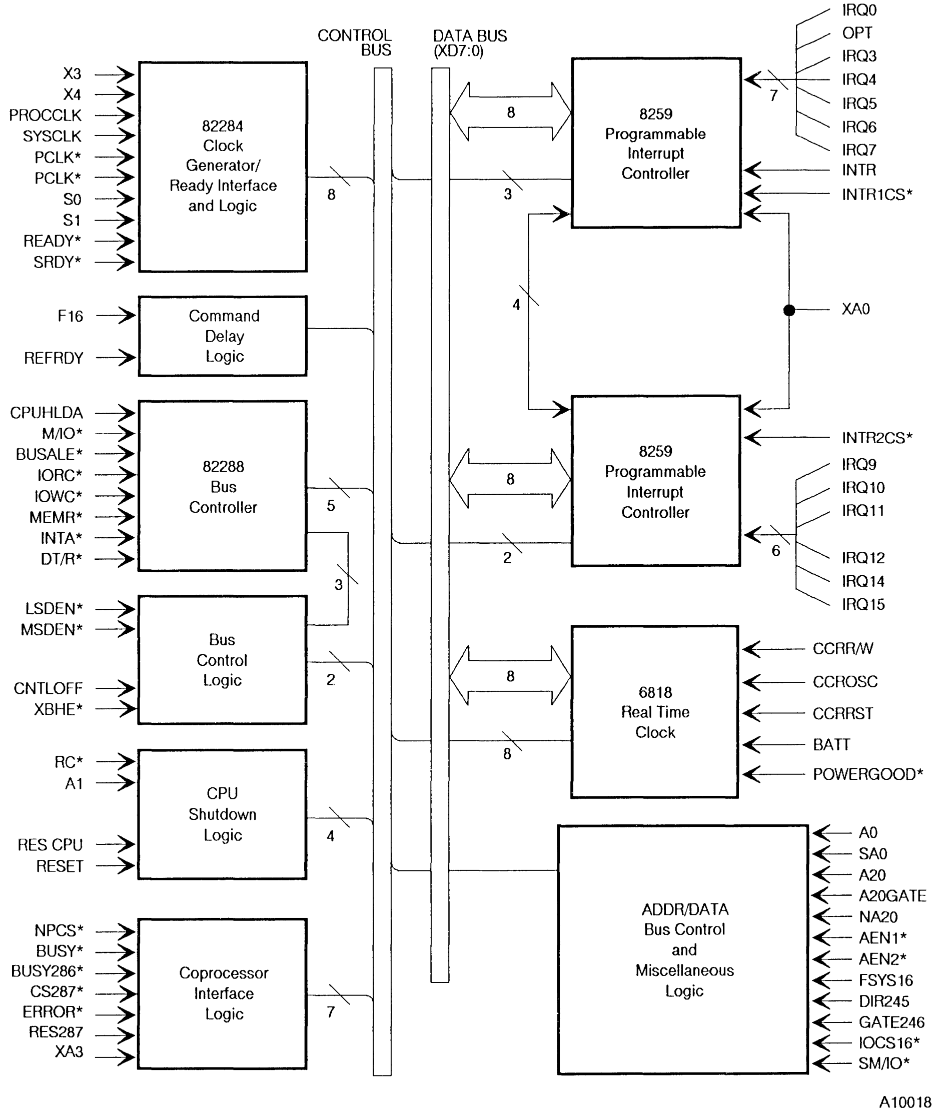


Figure 4-1. 82230 Block Diagram

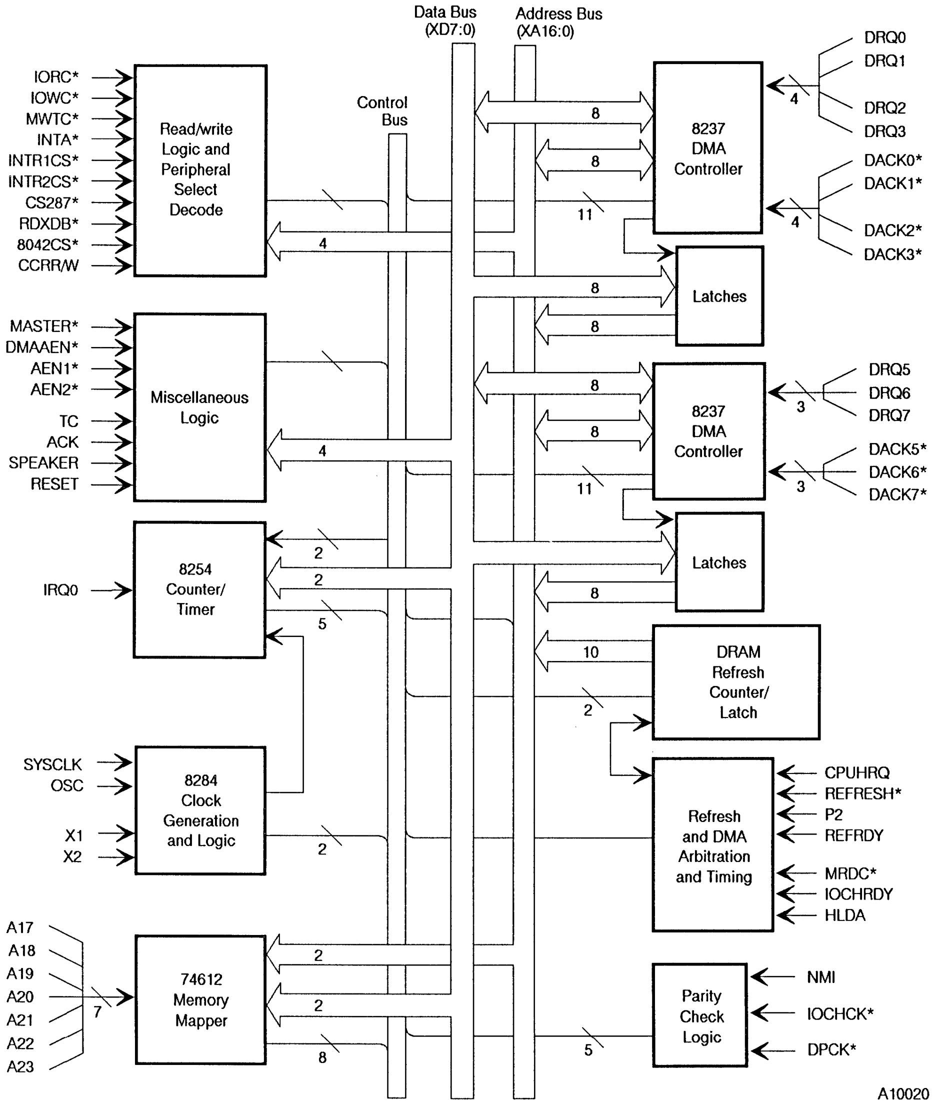


Figure 4-2. 82231 Block Diagram

4.3.1 RTC RAM I/O Operations

The RTC sets the system time and date. The RTC updates at one-second intervals and automatically adjusts at the end of months and leap years.

Writing the corresponding index address to I/O port 70H allows reading and writing of the 64 locations in the RTC. The RTC address register latches the address and points to the specified byte in the RTC.

Values can be written to or read from all 64 bytes except for the following read only bytes:

- Status registers C and D
- High-order bit (bit 7) of status register A
- High-order bit (bit 7) of the seconds byte

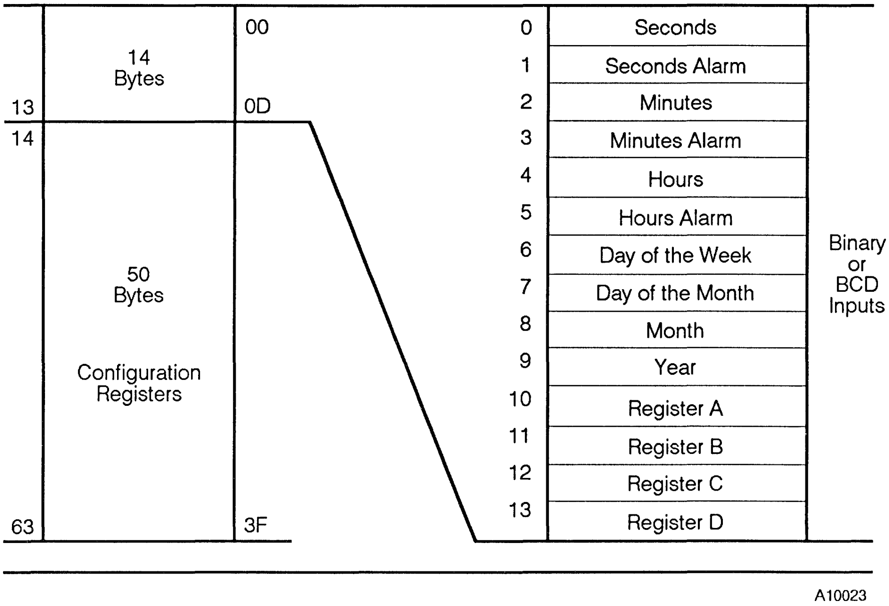


Figure 4-3. RTC Memory Map

Perform the following two steps when writing data into the RTC/RAM:

1. Write the RAM address (data = 00H through 3FH) into I/O port address 70H.

2. Write the data byte into I/O port 71H.

Perform the following two steps when reading data from the RTC/RAM:

1. Write the RAM address (data = 00H through 3FH) into I/O port address 70H.

2. Read the data byte from I/O port 71H.

NOTE
I/O port address 70H is also an output port for the NMI mask. Data bus bit 7 connects to the NMI mask bit and bits 0 through 5 connect to the RTC/RAM address lines.

During normal operation, the RTC performs an update cycle once every second. Divider bits DV2:0 and the SET bit in register B determine the performance of an update cycle. Divider bits DV2:0 must not be cleared and the register B set bit must be cleared. During an update, the lower ten registers are not available to the CPU. The update cycle increments the clock/calendar registers and compares them to the alarm registers. An interrupt is issued if a match occurs between the two sets of registers (with enabled alarm and interrupt control bits).

4.3.2 RTC Internal Addressable Locations

The 64 addressable locations in the RTC are divided into 10 bytes containing the time, calendar, alarm data, four control and status bytes, and 50 general purpose RAM bytes (refer to Table 4-1). Table 4-1 also details the internal register/RAM organization of the RTC.

Table 4-1. Real-time Clock Address Map

Function	Index
Seconds	00
Seconds Alarm	01
Minutes	02
Minutes alarm	03
Hours	04
Hours alarm	05
Day of week	06
Date of month	07
Month	08
Year	09
Status register A	0A
Status register B	0B
Status register C	0C
Status register D	0D
Diagnostic status byte	0E
Shutdown status byte	0F
Floppy disk drive type byte	10
Intel reserved	11
Fixed disk type byte	12
Intel reserved	13
Equipment byte	14
Low base memory byte	15
Requested high base memory byte	16
Requested low expansion memory byte	17
High expansion memory byte	18
Drive C extended type byte	19
Drive D extended type byte	1A
Intel Reserved	1B-1E
Features installed	1F
Drive type 48 parameters	20-27
Cache/shadow/setup	28
Intel reserved	29-2D
2-byte CMOS RAM checksum	2E-2F

Table 4-1. Real-Time Clock Address Map *(continued)*

Function	Index
Actual low expansion memory byte	30
Actual high expansion memory byte	31
Date century byte	32
Setup information	33
CPU speed	34
Drive type 49 parameters	35-3C
Reserved for user	3D-3F

TIME, CALENDAR, AND ALARM BYTES

The CPU obtains time and calendar information by reading the appropriate locations in the RTC. Writing to these locations initializes the time, calendar, and alarm information. Information stored in these locations are in binary coded decimal (BCD) format.

Before initializing the internal registers, the set bit in register B must be set to 1 to prevent RTC updates. Once set, the CPU initializes the first 10 locations in BCD format and the set bit is cleared.

Once initialized and enabled, the RTC performs clock/calendar updates at a 1-Hz rate. During updates, the 10 bytes of time, calendar, and alarm information are not available for reading or writing by the CPU for 2 ms.

Table 4-2 lists the format for the 10 clock, calendar, and alarm locations. System software sets the real-time clock to BCD data mode.

Table 4-2. Time, Calendar, and Alarm Data Format

Function	Index	BCD Data
Seconds	00	00:59
Seconds alarm	01	00:59
Minutes	02	00:59
Minute alarm	03	00:59
Hours	04	
(12 hour mode)	01:12 (AM)	
(24 hour mode)	00:23	
Hours alarm	05	
(12 hour mode)	01:12 (AM)	
(24 hour mode)	00:23	
Day of week	06	01:07
Date of month	07	01:31
Month	08	01:12
Year	09	00:99

NOTE

The RTC does not affect the 50 bytes of RAM from index address 0EH to 3FH. These bytes are accessible during the update cycle.

4.3.3 Status Registers

The four control and status bytes (status registers A:D) control the operation and monitor the status of the RTC. These registers, located at index addresses 0AH:0DH, are accessible by the CPU at all times (refer to Tables 4-3 through 4-6).

➡ NOTE
A setup program must initialize status registers A through D when setting the time and date.

STATUS REGISTER A (0AH)

Table 4-3. Status Register A (0AH)

Bit	Function
7	Update in progress (UIP) bit 1 = The time update cycle in progress 0 = The current date and time accessed
6:4	Divider selection (DV2:0) bits These bits control the divider/prescaler on the RTC. They specify the time-base frequency (in KHz) used. The system initializes to 010, specifying a time base of 32.768 KHz.
3:0	Rate selection (RS3:0) bits These bits select the divider output frequency. The system initializes to 0110, which selects a 1024-Hz divider frequency and an interrupt rate of 976.562 μs.

STATUS REGISTER B (0BH)

Table 4-4. Status Register B (0BH)

Bit	Function
7	Set update cycle (SET) bit 1 = Abort the update cycle in progress. Set to 1 for system initialization. 0 = Enable normal update cycle of one count per second.
6	Periodic interrupt enable (PIE) bit This read/write bit selects an interrupt occurring at a rate specified by the rate and divider selection bits in register A. 1 = Enable the generation of periodic interrupts 0 = Disable the interrupt (default)
5	Alarm interrupt enable (AIE) bit 1 = Enable the alarm interrupt 0 = Disable the alarm interrupt (default)
4	Update-ended interrupt enable (UIE) bit 1 = Enable the update-ended interrupt 0 = Disable the update-ended interrupt (default)

Table 4-4. Status Register B (0BH) *(continued)*

Bit	Function
3	Square wave enabled (SQWE) bit 1 = Enable the square-wave frequency set by the rate selection bits in register A 0 = Disable square-wave frequency (default)
2	Date mode (DM) bit Indicates whether the time and date calendar updates use binary or the BCD format. 1 = Select binary format. 0 = Select BCD format (default)
1	24/12-hour (24/12) bit Determines whether 24-hour mode or 12-hour mode is set in the hours byte. 1 = 24-hour mode (default) 0 = 12-hour mode
0	Daylight savings enabled (DSE) bit 1 = Daylight savings time enabled 0 = Daylight savings time disabled (default)

STATUS REGISTER C (0CH)

Table 4-5. Status Register C (0CH)

Bit	Function
7	IRQF (interrupt request flag) Set to 1 when any of the conditions cause an interrupt is true and the interrupt enable for that condition is true.
6	PF (periodic interrupt flag) Set to 1 when a transition, selected by RS3:0, occurs in the divider chain. This bit becomes active, independent of the condition of the PIE control bit. The PF bit generates an interrupt and sets IRQF if PIE = 1.
5	AF (alarm interrupt flag) Set to 1 when a match occurs between the time registers and alarm registers during an update cycle. This flag is independent of the condition of the AIE, and generates an interrupt if AIE is true.
4	UF (update ended interrupt flag) Set to 1 when an update ends. This flag is also independent of the condition of the UIE, and generates an interrupt if UIE is true.
3:0	Reserved

STATUS REGISTER D (0DH)

Table 4-6. Status Register D (0DH)

Bit	Function
7	Valid RAM and time (VRT) bit This read-only bit determines the condition of the RTC internal battery. 1 = Battery operational 0 = A low power sense. Indicates a dead battery in the RTC.
6:0	Reserved

4.3.4 Configuration Registers

The configuration bytes provide information on diagnostic status, shutdown status, equipment, memory, and other configuration parameters (refer to Tables 4-7 through 4-22).

DIAGNOSTIC STATUS BYTE (0EH)

Table 4-7. Diagnostic Status Byte (0EH)

Bit	Function
7	Real-time clock chip battery power status 1 = Power off 0 = Power on
6	Configuration record (checksum status indicator) 1 = Checksum not valid 0 = Checksum valid
5	Incorrect configuration information Checks the equipment byte of the configuration record when the system powers up. 1 = Configuration information not valid 0 = Configuration information valid
4	Memory size comparison 1 = Memory size different from configuration record 0 = Memory size the same as configuration record

NOTE

In order for the configuration information to be valid, power-on check requires at least one floppy disk drive installed (bit 0 of the equipment byte set to 1) and the display switch setting matches with the display controller installed.

Table 4-7. Diagnostic Status Byte (0EH) (continued)

Bit	Function
3	Initial state of drive C or fixed disk drive controller 1 = Wrong controller or drive C. System cannot boot from drive C. 0 = Correct controller and drive. The system boots from drive C.
2	Time status indicator (post checks) 1 = Time not valid 0 = Time valid
1:0	Reserved

SHUTDOWN STATUS BYTE (0FH)

When the CPU resets, the shutdown status byte is set. The reset code identifies the type of reset and signals the system what to do after the reset. It also provides a method of resetting the system without losing previously stored data or returning the system to the real mode from protected mode.

Table 4-8. Shutdown Status Byte (0FH)

Bit	Function
7:0	Reset information 00H = Normal system reset 09H = User software reset (return from protected mode) 01H:08H = Used by hardware self-test 0AH:FFH = Used by hardware self-test

FLOPPY DISK DRIVE TYPE BYTE (10H)

Table 4-9. Floppy Disk Drive Type Byte (10H)

Bit	Function
7:4	First floppy disk drive type 0000 = No floppy disk drive 0001 = 360K drive (5.25-inch) 0010 = 1.2M high-density drive (5.25-inch) 0011 = 720K (3.5-inch drive) 0100 = 1.4M (3.5-inch drive) 0101:1111 = Reserved
3:0	Second floppy disk drive type 0000 = No floppy disk drive 0001 = 360K drive (5.25-inch) 0010 = 1.2M high-density drive (5.25-inch) 0011 = 720K (3.5-inch drive) 0100 = 1.4M (3.5-inch drive) 0101:1111 = Reserved

FIXED DISK TYPE BYTE (12H)

Table 4-10. Fixed Disk Type Byte (12H)

Bit	Function
7:4	First fixed disk drive type (drive C) 0000 = No fixed disk drive installed 0001:1110 = Types 1 - 14 1111 = Types 16 - 255 (refer to extended byte 19H)
3:0	Second fixed disk drive type (drive D) 0000 = No fixed disk drive installed 0001:1110 = Types 1 - 14 1111 = Types 16 - 255 (refer to extended byte 1AH)

EQUIPMENT BYTE (14H)

The hardware self-test uses the equipment byte. The format of the equipment byte is described in Table 4-11.

Table 4-11. Equipment Byte (14H)

Bit	Function
7:6	Number of floppy disk drives installed 00 = 1 drive 01 = 2 drives 10 = Reserved 11 = Reserved
5:4	Type of video display controller used 00 = Extended functionality controller 01 = Color graphic video display controller in 40-column mode 10 = Color graphic video display controller in 80-column mode 11 = Monochrome display controller
3:2	Not used
1	Presence of a numeric coprocessor 1 = Numeric coprocessor installed 0 = No numeric coprocessor
0	Presence of floppy disk drive 1 = Floppy disk drive installed 0 = No floppy disk drive

LOW AND HIGH BASE MEMORY BYTES (15H AND 16H)

Table 4-12. Low and High Base Memory Bytes (15H and 16H)

Bit	Function
7:0 7:0	Address 15H (low-byte base size) Address 16H (high-byte base size)
	0100H = 256K RAM 0200H = 512K RAM 0280H = 640K RAM

REQUESTED LOW AND HIGH MEMORY EXPANSION BYTES (17H AND 18H)

This word indicates the total amount of expansion memory (above 1M) set by the system configuration program before any unused memory is rolled.

Table 4-13. Low and High Memory Expansion Bytes (17H and 18H)

Bit	Function
7:0 7:0	Address 17H (low-byte expansion size) Address 18H (high-byte expansion size)
	0200H = 512K RAM expansion 0400H = 1024K RAM expansion 0600H = 153K RAM expansion 3C00H = 15360K RAM expansion

DRIVE C EXTENDED BYTE (19H)

Table 4-14. Drive C Extended Byte (19H)

Bit	Function
7:0	Addresses 00H:0FH (reserved) Addresses 10H:FFH (types 16:255)

DRIVE D EXTENDED BYTE (1AH)

Table 4-15. Drive D Extended Byte (1AH)

Bit	Function
7:0	Addresses 00H:0FH (reserved) Addresses 10H:FFH (types 16:255)

FEATURE INSTALLED BYTE (1FH)

Table 4-16. Feature Installed Byte (1FH)

Bit	Function
7:3	Reserved
2	Floppy disk drive A installed
1	Video display installed
0	Keyboard BIOS installed

CMOS RAM CHECKSUM (2EH AND 2FH)

The CMOS RAM checksum is the sum of the values from addresses 10H through 2DH.

Table 4-17. CMOS RAM Checksum (2EH and 2FH)

Bit	Function
7:0	Address 2EH (high byte of checksum)
7:0	Address 2FH (low byte of checksum)

FXD TYPE 48 PARAMETERS (20H - 27H)

Table 4-18. FXD Type 48 Parameters (20H-27H)

Parameter	Function
20H	Cylinder low byte
21H	Cylinder high byte
22H	Number of heads
23H	Write pre-compensation start cylinder low byte
24H	Write pre-compensation start cylinder high byte
25H	Landing zone cylinder low byte
26H	Landing zone cylinder high byte
27H	Sectors per track

SHADOW AND ENTER SETUP (28H)

Table 4-19. Shadow and Enter Setup (28H)

Bit	Function
7:4	Reserved
3	Enter Setup program at pre-boot only
2	Reserved
1	Video shadow disabled
0	BIOS shadow disabled

NOTE

Video BIOS, alone, cannot be shadowed. It must be shadowed with the system BIOS.

ACTUAL LOW AND HIGH EXTENDED MEMORY BYTES (30H AND 31H)

The high and low expansion memory bytes represent the total extended memory (above 1M) determined during system power up and after any unused memory has been rolled. System interrupt 15H determines extended memory size.

Table 4-20. **Low and High Extended Memory Bytes (30H and 31H)**

Bit	Function	
7:0 7:0	Address 30H (low-byte extended memory size) Address 31H (high-byte extended memory size)	
	0200H =	512K RAM extended
	0400H =	1024K RAM extended
	0600H =	1536K RAM extended
	3C00H =	15360K RAM extended

DATE CENTURY BYTE (32H)

The date century byte is the century part of the current date encoded in BCD format.

Table 4-21. **Date Century Byte (32H)**

Bit	Function
7:0	BCD value for century (BIOS sets and reads this value)

SETUP INFORMATION (33H)

Table 4-22. Setup Information (33H)

Bit	Function
7	128K ROM expansion
6	Enable user message after initial setup
5	Reserved
4	Copy of the 386 CR0 ET bit (will always be 1, regardless if an 387SX is present or not)
3:0	Reserved

CPU SPEED (34H)

Table 4-23. CPU Speed (34H)

Bit	Function
7:3	Reserved
2:0	CPU speed

FXD TYPE 49 PARAMETERS (35H - 3CH)

Table 4-24. FXD Type 49 Parameters (35H-3CH)

Parameter	Function
20H	Cylinder low byte
35H	Cylinder low byte
36H	Cylinder high byte
37H	Number of heads
38H	Write pre-compensation start cylinder low byte
39H	Write pre-compensation start cylinder high byte
3AH	Landing zone cylinder low byte
3BH	Landing zone cylinder high byte
3CH	Sectors per track

4.4 8259A PROGRAMMABLE INTERRUPT CONTROLLERS (PICS)

The system interrupt controllers (INTC1 and INTC2) are Intel 8259A programmable interrupt controllers (PICs) contained in the 82230 portion of the Intel 82230/82231 Chip Set. INTC1 is located between addresses 20H and 21H and configured for master operation. INTC2 is a slave device located between addresses A0H and A1H.

INTC1 has seven prioritized interrupt levels and INTC2 has eight. The PIC minimizes the software and real time overhead in handling multilevel priority interrupts.

The PIC functions as an overall manager in an interrupt-driven system environment. It accepts incoming interrupt requests from the various peripherals attached to the system unit. The PIC checks the interrupt requests for priority, determines whether the incoming request has a higher priority value than the current level, then sends an INT pulse to the CPU. The CPU acknowledges the INT request with an INTA signal.

Configuring a variety of priority assignment modes any time during system operation structures the system interrupt (based on the system environment).

Each peripheral device has a special program associated with its specific functional or operational requirements. The PIC, after receiving an interrupt request from the peripheral device special program, sends the interrupt request information to the CPU. This forces the program counter to the starting interrupt vector address. Interrupt vector addresses are stored in a table in low memory (locations 0 to 1024).

4.4.1 Interrupt Controller Architecture

The interrupt controller consists of an interrupt request register, cascade buffer, in-service register, interrupt mask register, and priority resolver.

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

IRR and ISR handle the interrupts at the input lines. The IRR stores interrupt levels requesting service; the ISR stores interrupt levels currently in service.

INTERRUPT MASK REGISTER

Each interrupt line can be individually masked by the interrupt mask register (IMR). Masking an interrupt line prevents the IRR from generating an interrupt. Masking a higher priority input does not affect the interrupt request lines of lower priority.

PRIORITY RESOLVER

The priority resolver determines the priorities of the bits set in the IRR. The priority resolver, following the first INTA pulse from the CPU, selects the highest priority and strobes it into the corresponding bit of the ISR.

4.4.2 Types of Interrupts

The system module utilizes the following four types of interrupts:

- Maskable interrupts
- Non-maskable interrupts
- Hardware interrupts
- Software interrupts

The CPU enables or disables the maskable interrupt (INTR). Parity check logic within the 82231 masks the non-maskable interrupt (NMI). The NMI cannot be masked off within the CPU. The NMI mask register appears at I/O address port 70H.

MASKABLE INTERRUPTS

All maskable hardware interrupts to the CPU are processed by the interrupt controllers. These devices generate interrupts on the CPU interrupt line (INTR) and can be masked in the CPU with the Clear Interrupt Enable Flag (CLI) instruction. Any or all interrupts may be masked using the IMR; however, the interrupt vector of the PIC must be initialized in advance. Several interrupts are used by the system for onboard peripherals.

Fifteen interrupt levels are available by cascading the two interrupt controllers in the 82230 (refer to Table 4-25). The slave controller signals the master to cause an interrupt. The base I/O addresses are: INTC1 (master) 20H and INTC2 (slave) A0H.

Table 4-25. Interrupt Levels

Priority	PIC Number	Interrupt Number	Interrupt Source
1	†	NMI	Parity error detected
2	1	IRQ0	Interval timer, counter 0 output
3	1	IRQ1	Keyboard output buffer full
	1	IRQ2	Interrupt from controller 2 (cascade)
4	2	IRQ8	Real-time clock
5	2	IRQ9	Onboard video, software redirected to INT 0AH (IRQ2)
6	2	IRQ10	Real-time clock INT
7	2	IRQ11	Reserved
8	2	IRQ12	Auxiliary device
9	2	IRQ13	INT from numeric coprocessor
10	2	IRQ14	Fixed disk drive controller
11	2	IRQ15	Reserved
12	1	IRQ3	Serial communications port 2
13	1	IRQ4	Serial communications port 1
14	1	IRQ5	Parallel port 2
15	1	IRQ6	Floppy disk drive controller
16	1	IRQ7	Parallel port 1

†I/O port 70H bit 7 controls the masking of the NMI signal.

The possibility of two or more interrupts demanding service at the same time exists. The PICs, after determining the priority of the interrupt requests, process the requests one at a time by transferring the control of the CPU to the higher priority service routine first.

NON-MASKABLE INTERRUPTS

NMI interrupts are caused by the following:

- Detection of parity error during memory (DRAM) read on the system board
- Parity errors on any 8-bit or 16-bit board pulling the IOCHCK* line low
- System software generating a software interrupt to the NMI routine (BIOS call)

Enable or disable the NMI interrupt by performing the following operations:

- Clearing Mask F/F: OUT 70H, 00H. Executing this instruction enables the NMI.
- Setting Mask F/F: OUT 70H, 80H. Executing this instruction disables the NMI.

At power on, the NMI is disabled until system software executes the clear mask instruction. NMI remains enabled after the power-on self test completes.

4.4.3 Programming the PICs

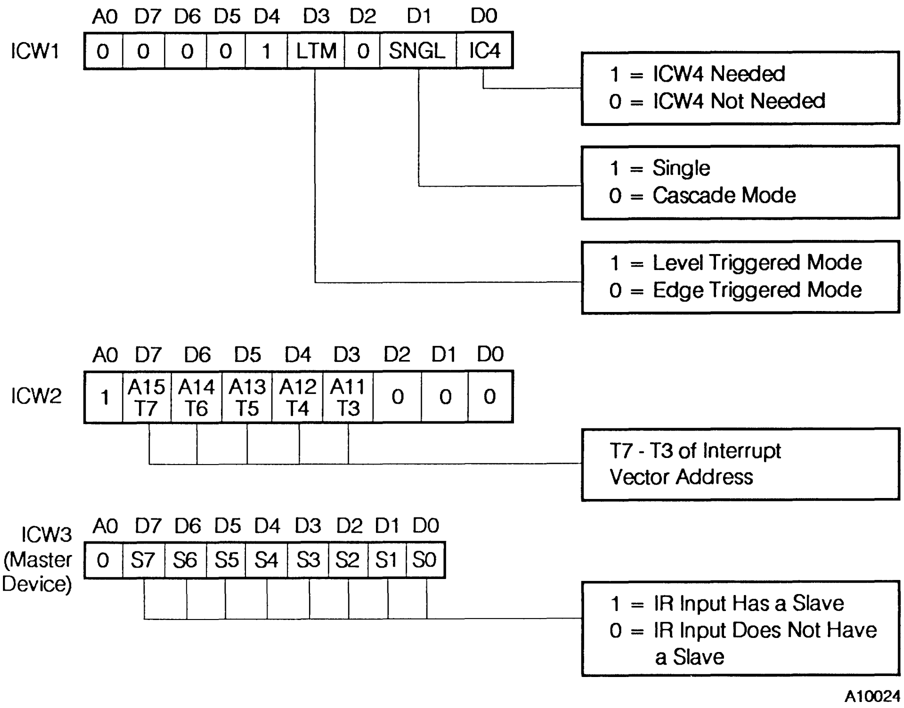
Initialization Command Words (ICWs) and Operation Command Words (OCWs) are used to program the operation of the PICs (refer to Table 4-26).

Table 4-26. Interrupt Controller I/O Address and I/O Data

Interrupt Controller	I/O Port Address	Write Data	Read Data
INTC1	0020H	ICW1	
Master		OCW2	
(IRQ0-IRQ7)		OCW3	IRR
			ISR
	0021H	ICW2	
		ICW3	
		ICW4	
		OCW1	IMR
INTC2	00A0H	ICW1	
Slave		OCW2	
(IRQ8-IRQ15)		OCW3	IRR
			ISR
	00A1H	ICW2	
		ICW3	
		ICW4	
		OCW1	IMR

INITIALIZATION COMMAND WORDS (ICWS)

The initialization command words (ICWs) initialize the PICs after power-on. Before normal operation can begin, the PICs must be brought to a starting point by writing a sequence of four ICWs to each PIC. See Figures 4-4 and 4-5 for a description of the formats of ICWs. Figure 4-6 shows the initialization sequence of the ICWs.



ICW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	LTM	0	SNGL	IC4

1 = ICW4 Needed
0 = ICW4 Not Needed

1 = Single
0 = Cascade Mode

1 = Level Triggered Mode
0 = Edge Triggered Mode

ICW2

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	A15 T7	A14 T6	A13 T5	A12 T4	A11 T3	0	0	0

T7 - T3 of Interrupt
Vector Address

ICW3
(Master
Device)

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	S7	S6	S5	S4	S3	S2	S1	S0

1 = IR Input Has a Slave
0 = IR Input Does Not Have
a Slave

Figure 4-4. Command Word Format (ICW1, ICW2, and ICW3)

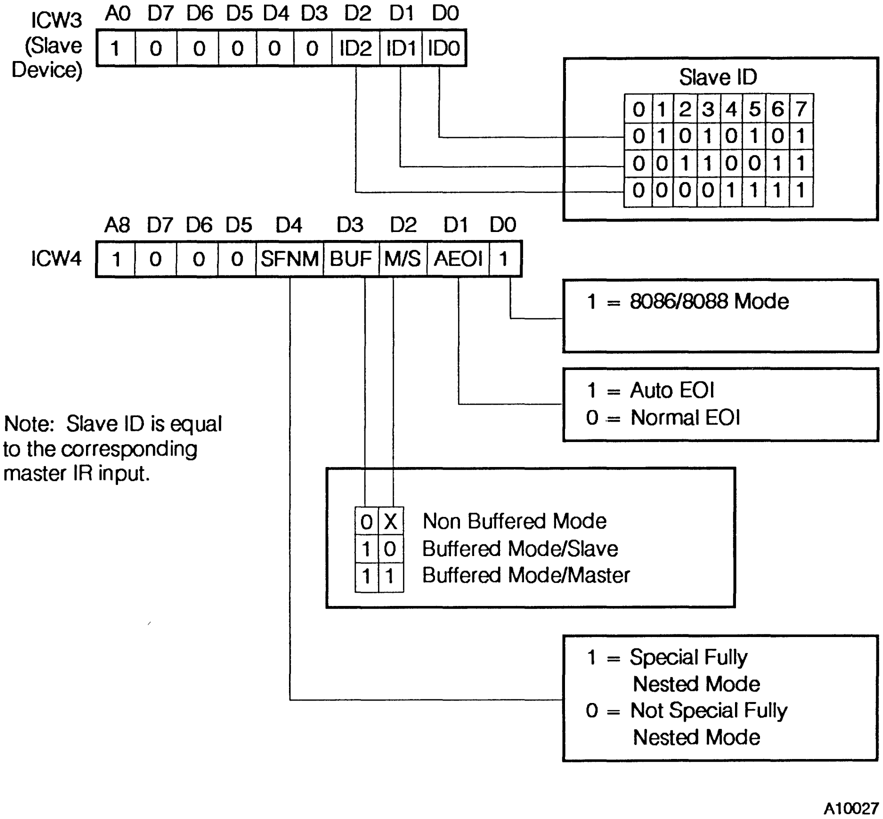


Figure 4-5. Command Word Format (ICW3 and ICW4)

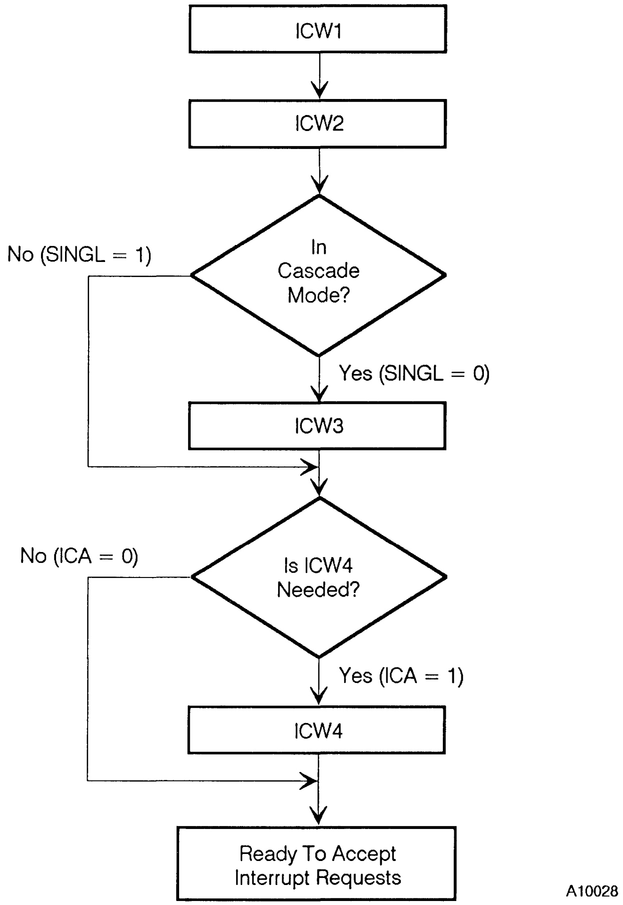


Figure 4-6. Initialization Sequence Diagram

The initialization sequence starts by writing ICW1 with a value of 11H to I/O address 020H for INTC1 (I/O address 0A0H for INTC2). The interrupt controller interprets this as the start of an initialization sequence and does the following:

1. Resets the ICW counter to zero
2. Selects the fixed priority mode
3. Assigns the highest priority to IR0
4. Clears the interrupt mask register and inservice register
5. Sets the slave mode address to 7
6. Disables the special mask mode
7. Selects the interrupt request register for status read operations
8. Resets the edge sense circuit for edge-triggered interrupt response mode

The next three I/O writes to address 021H (0A1H for INTC2) loads ICW bytes 2, 3, and 4. See Figures 4-7 and 4-8 for the initialization formats from the power-on test routine.

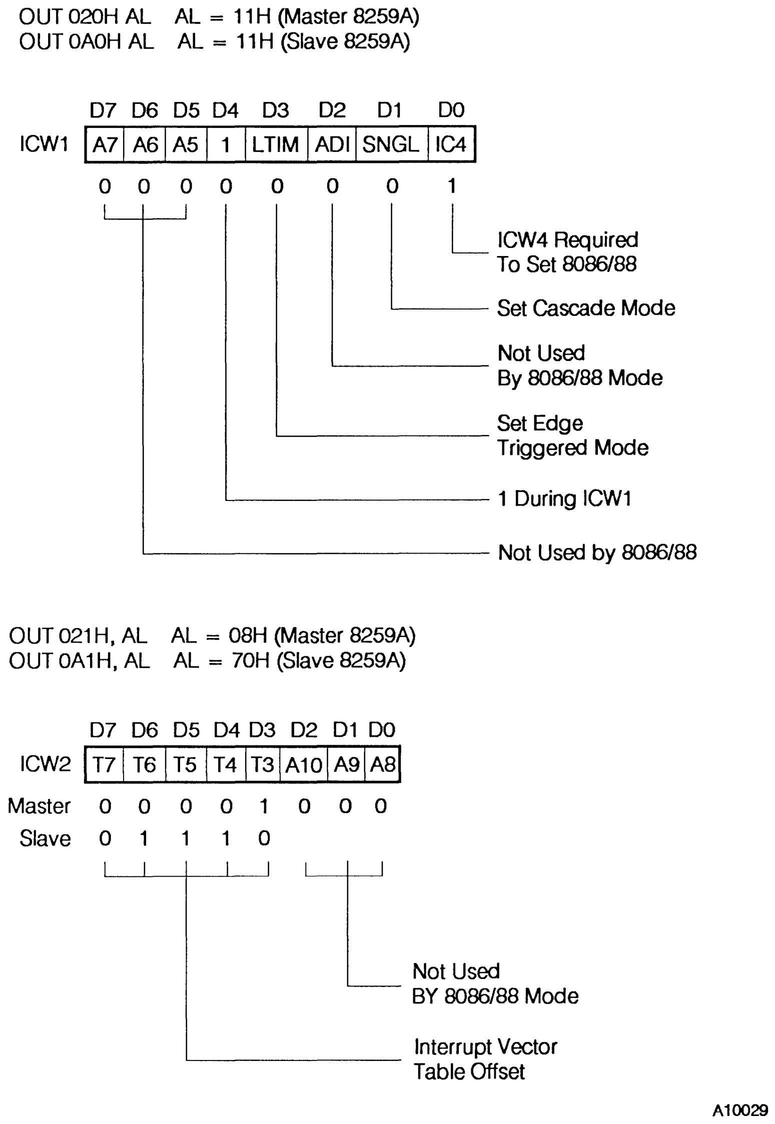
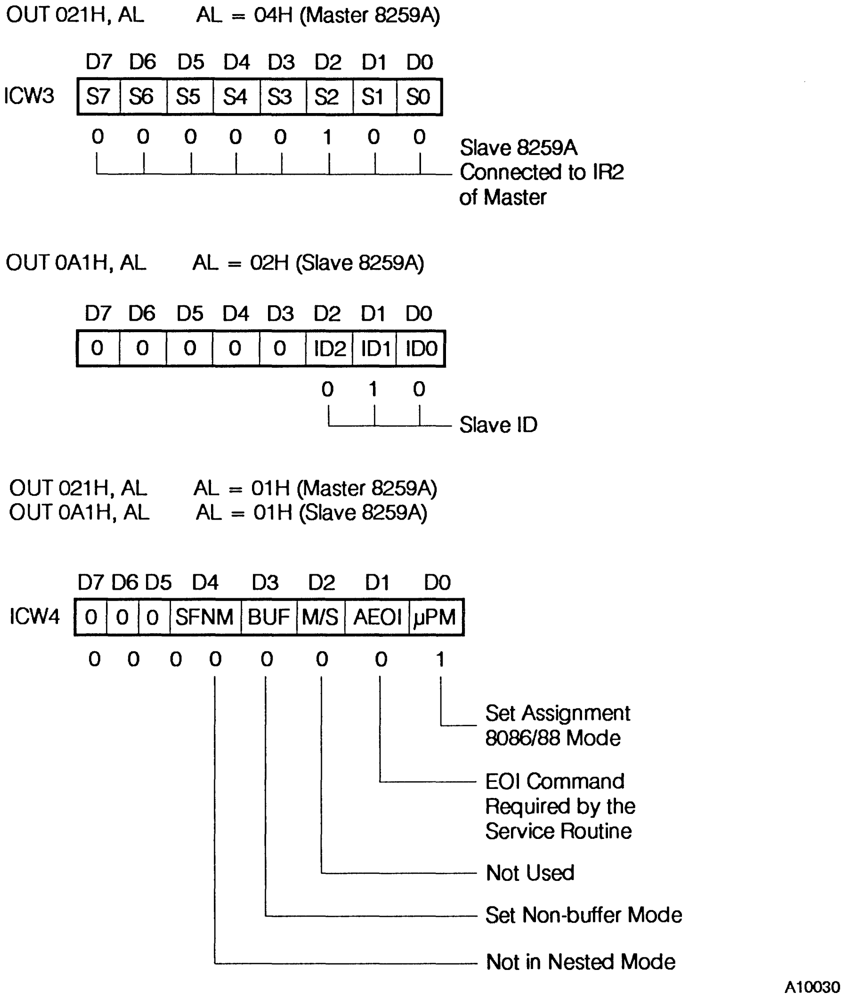


Figure 4-7. **Power-on Formats (ICW1 and ICW2)**



```
OUT 021H, AL        AL = 04H (Master 8259A)
```

	D7	D6	D5	D4	D3	D2	D1	D0
ICW3	S7	S6	S5	S4	S3	S2	S1	S0
	0	0	0	0	0	1	0	0

```
OUT 0A1H, AL        AL = 02H (Slave 8259A)
```

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	ID2	ID1	ID0
					0	1	0

```
OUT 021H, AL        AL = 01H (Master 8259A)
OUT 0A1H, AL        AL = 01H (Slave 8259A)
```

	D7	D6	D5	D4	D3	D2	D1	D0
ICW4	0	0	0	SFNM	BUF	M/S	AEOI	μPM
	0	0	0	0	0	0	0	1

Slave 8259A Connected to IR2 of Master
Slave ID
Set Assignment 8086/88 Mode
EOI Command Required by the Service Routine
Not Used
Set Non-buffer Mode
Not in Nested Mode
A10030

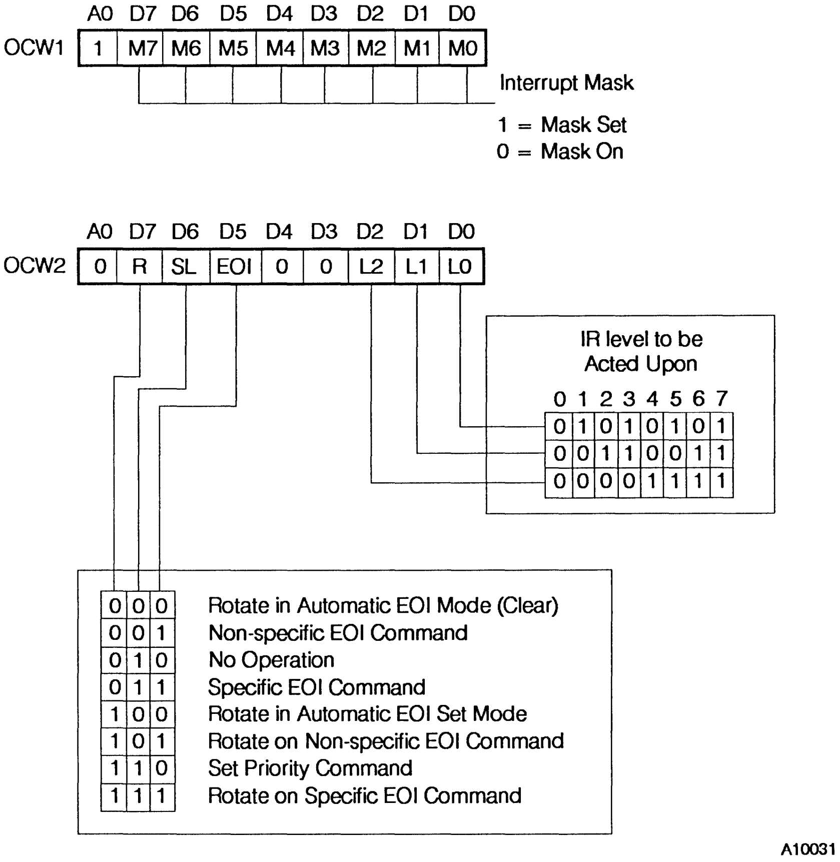
Figure 4-8. **Power-on Formats (ICW3 and ICW4)**

OPERATIONAL CONTROL WORDS (OCWS)

The OCWs command the PICs to operate in various interrupt modes. Each PIC has three OCWs which can be programmed to change the configuration and to monitor controller operation.

OCW1 can be written to address 021H (0A1H for INTC2) any time the controller is not in initialization mode. OCW2 and OCW3 are written to address 020H (0A0H for INTC2). Writing to address 020H (0A0H) with a 0 in bit 4 places the controller in operational mode and loads OCW2 (if data bit 3 = 0) or OCW3 (if data bit 3 = 1).

It is not necessary to execute the OCWs in sequence. The OCW is an output instruction defined by the software. Figures 4-9 and 4-10 diagram the OCW instruction sequence.


A0 D7 D6 D5 D4 D3 D2 D1 D0
OCW1
1 M7 M6 M5 M4 M3 M2 M1 M0
Interrupt Mask
1 = Mask Set
0 = Mask On
A0 D7 D6 D5 D4 D3 D2 D1 D0
OCW2
0 R SL EOI 0 0 L2 L1 L0
IR level to be Acted Upon
A10031

IR level to be Acted Upon

0	1	2	3	4	5	6	7
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1

0	0	0	Rotate in Automatic EOI Mode (Clear)
0	0	1	Non-specific EOI Command
0	1	0	No Operation
0	1	1	Specific EOI Command
1	0	0	Rotate in Automatic EOI Set Mode
1	0	1	Rotate on Non-specific EOI Command
1	1	0	Set Priority Command
1	1	1	Rotate on Specific EOI Command

Figure 4-9. Command Word Format (OCW1 and OCW2)

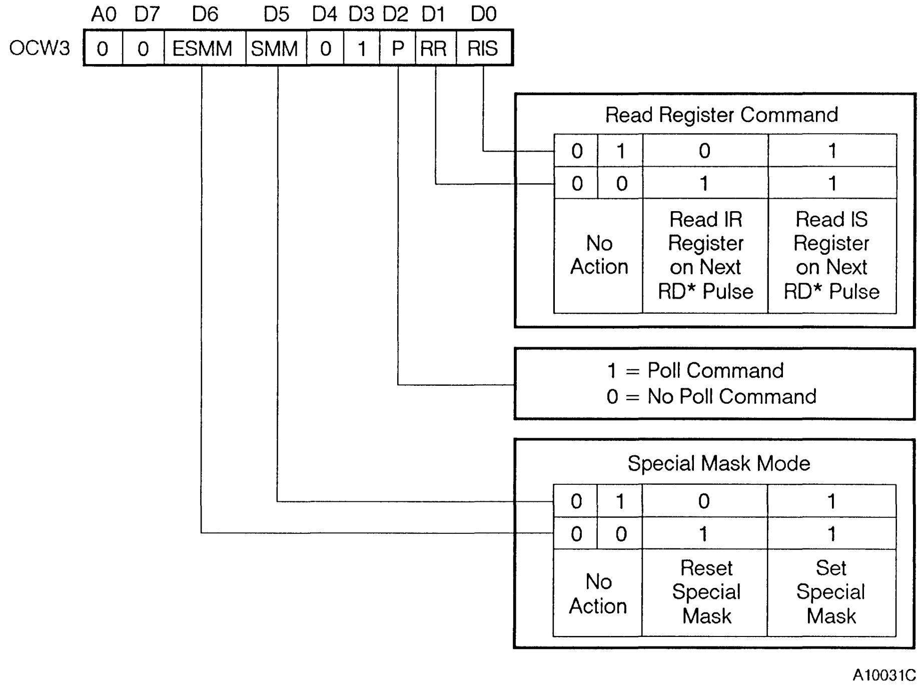


Figure 4-10. Command Word Format (OCW3)

4.5 8254 PROGRAMMABLE INTERVAL TIMER (PIT)

The system timer is an Intel 8254 programmable interval timer (PIT) located in the 82231 portion of the Intel 82230/82231 Chip Set. The PIT contains three independent 16-bit counters counting down in binary coded decimal (BCD) or binary. The counters are read, written, and configured using common control logic. Each counter operates in one of the following six modes:

- Mode 0: interrupt on terminal count
- Mode 1: hardware triggered one-shot
- Mode 2: rate generator
- Mode 3: square wave generator
- Mode 4: software triggered strobe
- Mode 5: hardware triggered strobe

4.5.1 Counter Description

A common clock drives each of the three PIT counters. The clock frequency is 1.19318 MHz, which is derived from the 14.31818 MHz crystal connected to pins X1 and X2 of the 82231. The output of counter 0 (IRQ0) connects directly to IRQ0 of the 82230 interrupt controller module and provides the system timer interrupt for time-of-day, disk timeout, and other system timing functions. System hardware uses counter 1 to generate a dynamic random access memory (DRAM) refresh operation request signal. Counter 2 generates the tone for the system speaker.

Each counter contains a control register, a status register, a 16-bit counting element (CE), a pair of 8-bit counter input latches (CIL and CIH), and a pair of 8-bit counter output latches (COL and COH). Each counter also has a clock input for loading and decrementing the CE. The CE is a mode-defined GATE input for controlling the counter and an OUT signal. The counter mode and condition of CE controls the OUT signal state and function. Table 4-27 lists the functions of the counters, the respective gates, and the clock in/clock out signals.

Table 4-27. Counter Functions, Gates, and Signals

Counter	Signals	Function
0		System timer
	GATE 0C	Always on
	CLK0C	1.193 MHz
	OUT0C	PITIRQ IR0A
1		Refresh request
	GATE 1C	Always on
	CLK1C	1.193 MHz
	OUT1C	Request refresh
2		Speaker frequency
	GATE 2C	ENBSPK
	CLK2C	1.193 MHz
	OUT2C	Speaker signal

CONTROL AND STATUS REGISTERS

The control register stores the mode and command information for each of the counters. Writing a byte to the write control word address (043H) loads the control register. The byte contains a pointer to the desired counter, the type of command, and count format information. System software uses a status register to monitor the counters and read back the contents of the control register.

COUNTING ELEMENT (CE)

The CE is a 16-bit synchronous down counter. Writing one or two bytes in the counter input latches loads the CE. The CE loads or decrements on the falling edge of the input clock. The CE contains the maximum count when loading 0 and wraps around to FFFFH in binary operation.

4.5.2 Programming the PIT

Programming the PIT requires writing a control word and initial counts to the three PIT counters at power-up. Register F receives the control word while each counter receives an initial count listed in Table 4-28.

Table 4-28. **Counter/Timer Address Map**

Address	Function
040H	Counter 0 read/write
041H	Counter 1 read/write
042H	Counter 2 read/write
043H	Control register (write-only)

CONTROL REGISTER (043H)

Programming a counter requires writing control words to control register 043H (refer to Table 4-29). Control words specify the counter, command, mode, and numeric format (BCD or binary). Control register 043H is write-only.

Table 4-29. **Control Register Bit Definition**

Bit	Function
7:4	Command to be performed 0000 = Latch counter 0 0001 = Read/write counter 0 least significant byte only 0010 = Read/write counter 0 most significant byte only 0011 = Read/write counter 0 least significant byte then most significant byte 0100 = Latch counter 1 0101 = Read/write counter 1 least significant byte only 0110 = Read/write counter 1 most significant byte only 0111 = Read/write counter 1 least significant byte then most significant byte 1000 = Latch counter 2 1001 = Read/write counter 2 least significant byte only 1010 = Read/write counter 2 most significant byte only 1011 = Read/write counter 2 least significant byte then most significant byte 11xx = Read-back command
3:1	Operating mode selection These bits are don't care during Latch Counter command. 000 = Mode 0 001 = Mode 1 x10 = Mode 2 x11 = Mode 3 100 = Mode 4 101 = Mode 5
0	Binary or BCD count down format This bit must be 0 during Read/Write command. 0 = Binary (16-bit) count down 1 = Binary-coded decimal count down (four decades)

Read/Write Counter Command

The following three conventions must be observed when loading a counter using the read/write counter command:

- Each counter control word must be written before loading the initial count.
- Writing an initial count must follow the format specified in the control word. Load the least significant byte (LSB) only, most significant byte (MSB) only, or load the LSB then the MSB.
- When writing the LSB and MSB, control must not be relinquished between modes.

A new initial count can be written into the counter at any time after programming. Writing a new initial count does not require rewriting the control word as long as the programmed format is observed.

When reading a counter, the following two conventions must be observed:

- The count must be latched.
- When reading the LSB and MSB, control must not be relinquished between modes.

Latch Counter Command

When issuing a latch counter command, the counter output latches (COL and COH) latch the current state of the CE. COL and COH remain latched until read by the CPU, or until the counter is reprogrammed. Once read or reprogrammed, COL and COH return to the value currently in the CE.

Latch counter commands can be issued to more than one counter before reading the first counter. Multiple latch counter commands issued to the same counter, without reading the counter, are ignored with the exception of the first command.

Read-back Command

The read-back command checks the count value, mode, state of the OUT signal, and NULL count flag of the selected counters (refer to Table 4-30). Latched status bytes (OUT and NULL) remain until the counter is read or reprogrammed (refer to Table 4-31).

Table 4-30. Read-Back Command Format

Bit	Function
7:6	Read-Back command specified 11 = Specifies the READ-BACK command
5:4	Latch CE/Status (LC and LS) If LS and LC = 0, status returns on the first read from the counter. The next one or two reads from the counter results in the count being returned. Bit 5 = 0 latches the state of the CE in COL and COH. Bit 4 = 0 latches status of selected counters into the status register.
3:0	Counter selection (C2:C0) Bit 3 = 1 selects counter 2 Bit 2 = 1 selects counter 1 Bit 1 = 1 selects counter 0 Bit 0 = 0 (must be zero)

Table 4-31. Format of a Status Byte Latched

Bit	Function
7	OUT signal state 0 = OUT signal 0 (low) 1 = OUT signal 1 (high)
6	NULL count flag condition 0 = Counter loaded from the counter input registers, count can be read. 1 = Write to the control register or the counter, but the new value has not been loaded into CE.
5:4	Byte transfer during Read/Write commands 00 = Reserved 01 = R/W least significant byte 10 = R/W most significant byte 11 = R/W least significant byte then most significant byte
3:1	Operating mode selection 000 = Mode 0 001 = Mode 1 010 = Mode 2 011 = Mode 3 100 = Mode 4 101 = Mode 5
0	Binary or BCD count down format 0 = Binary (16-bit) count down 1 = BCD count down (four decades)

4.6 DIRECT MEMORY ACCESS (DMA) CONTROLLERS

DMA is a process where data is transferred between I/O devices and memory without using the CPU. Intel 8237A DMA controllers within the 82231 portion of the Intel 82230/82231 Chip Set implement the process.

The 300SX board utilizes two 8237A DMA controllers. Each DMA controller device generates the memory addresses and control signals necessary to transfer information. The DMA controllers are internally cascaded and provide four DMA channels for transfers to 8-bit peripherals (DMA1) and three channels for transfers to 16-bit peripherals (DMA2).

4.6.1 DMA Operation

The DMA controllers and I/O devices use the DMA request (DRQ) and DMA acknowledge (DACK*) signals for handshaking. An I/O device activates the DRQ line when transferring a byte or a data word. The I/O device asserts the byte or data word on the data bus after receiving an active DACK* from the DMA controller. During normal operation, the DMA controller can be in an idle cycle, program condition, or active cycle.

IDLE CYCLE

The DMA controller enters the idle cycle when no I/O device requests service. During the idle cycle, the DMA controller samples the DRQ input pins every clock cycle. The DMA controller also samples the chip select (CS*) signal and determines if the CPU is attempting to access the DMA controller registers. If either situation occurs, the DMA controller exits the idle cycle.

ACTIVE CYCLE

The DMA controller enters the active cycle when a DMA request on a non-masked channel occurs. During an active cycle, the system programs the DMA controller to operate in one of four modes: single transfer, block transfer, demand transfer, or cascade.

Single Transfer Mode

During single transfer mode, the DMA controller makes one transfer. After each transfer, the current word count decrements and the current address decrements or increments. A terminal count (TC) causes an autoinitialize if the channel is programmed and the current word count goes from zero to FFFFH. Autoinitialize, enabled by a bit in the mode register, restores channels to their original condition.

An I/O device must hold DRQ active until a corresponding DACK* signal becomes active. If the I/O device holds DRQ active throughout the single transfer, the hold request (HRQ) signal goes inactive for one full cycle, then becomes active again. Another single transfer begins when the DMA controller receives a new HLDA.

Block Transfer Mode

During block transfer mode, a DRQ activates the DMA controller. DRQ remains active during an active DACK*. The controller continually makes block transfers during the service until it encounters a TC, caused by the word count going to FFFFH, or an external end of process (EOP*). If the DMA channel is programmed for autoinitialize, autoinitialization occurs at the end of the service.

Demand Transfer Mode

During demand transfer mode, the DMA controller continually makes demand transfers until it encounters a TC, an external EOP*, or when the current DRQ goes inactive. An inactive DRQ, before the last working state of the cycle, prevents another transfer. Current address and current word count registers store intermediate values of addresses and word counts between transfers. An EOP* causes an autoinitialize at the end of the DMA controller service.

Cascade Mode

During cascade mode, both DMA controllers are cascaded together for system expansion. The HRQ and HLDA signals, from the slave DMA controller, connect to a DRQ and DACK* signal of the master DMA controller. This allows the master DMA controller to request the slave DMA controller to propagate through the priority network circuitry of the preceding device (master DMA controller). The cascade channel of the master controller prioritizes the slave controller and does not output any address or control signals of its own. The DMA controller responds only to DRQ and DACK*. All other outputs, except HRQ, are disabled.

4.6.2 Transfer Types

Each of the four transfer modes performs three types of transfers: read, write, and verify (see Figure 4-11).

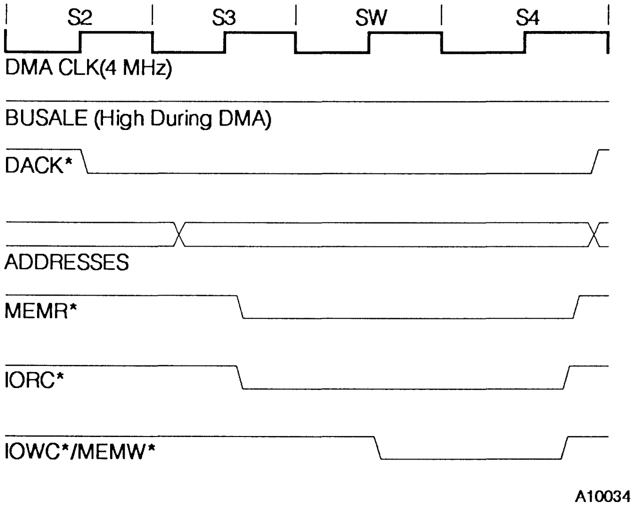


Figure 4-11. DMA Timing

READ TRANSFER

Read transfers move data from memory to an I/O device by generating the memory address and asserting MEMR* and IOWC* during the same cycle.

WRITE TRANSFER

Write transfers move data from an I/O device to memory by generating the memory address and asserting MEMW* and IORC* during the same cycle.

VERIFY TRANSFER

Verify transfer is a pseudo-transfer used for diagnostics. In this type of transfer, the DMA operates as if it were performing a read or write transfer by generating HRQ, address, and DACK*. However, memory and I/O control lines remain inactive.

4.6.3 DMA Channels

DMA1 supports channels 3:0 and DMA2 supports channels 7:5. Both DMA controllers support seven channels (refer to Table 4-32). Channel zero has the highest priority and channel seven the lowest.

Table 4-32. DMA Channel Assignment

Channel	Controller	Function
0	DMA1	Spare
1	DMA1	Spare
2	DMA1	Floppy disk drive
3	DMA1	Spare
5	DMA2	Spare
6	DMA2	Spare
7	DMA2	Spare

The DMA controller and page register provides 24 bits of address. The system assigns a page register to each DMA channel. The page register supplies the upper eight bits of each 24-bit address. The DMA controller supplies the lower 16 bits of each 24-bit address. Generated addresses do not cross page boundaries (64K for channels 3:0 and 128K for channels 7:5).

The DMA controller and page register must be initialized before initiating a DMA transfer. Initialization consists of loading the starting address, the number of bytes, and the direction of transfer. The CPU reads and writes the DMA controllers' internal registers when the controller is selected and while the CPU controls the bus. All registers, including those not used, are written to at power-up.

4.6.4 Programming the DMA Controllers

The DMA controllers can be programmed any time during an inactive HLDA. The CPU ensures no DMA activity occurs on the channel being programmed. To prevent a conflict, disable the DMA controller or mask the DMA channel before programming any registers. Disabling external interrupts protects the sections of code that set up the registers. This prevents another routine from changing the state of the circuit.

4.6.5 DMA Internal Registers

The DMA internal registers control DMA transfers. Table 4-33 lists all DMA internal registers and their size.

Table 4-33. DMA Internal Registers

Register Name	Size	Number
Base address registers	16 bits	4
Base word count registers	16 bits	4
Current address registers	16 bits	4
Current word count registers	16 bits	4
Temporary address register	16 bits	1
Temporary word count register	16 bits	1
Status register	8 bits	1
Command register	8 bits	1
Temporary register	8 bits	1
Mode registers	6 bits	4
Mask register	4 bits	1
Request register	4 bits	1

ADDRESS AND COUNT REGISTERS

The address and count registers specify the address and length of the transfer. These registers are 16 bits, requiring two successive read or write operations. The byte pointer flip-flop must be cleared before accessing these registers. This guarantees that the first operation transfers the low order byte and the second operation transfers the high order byte. After clearing the byte pointer flip-flop, control must not be relinquished by the program until both bytes are written or read. Table 4-34 lists the I/O port address assignments for the DMA address and count registers.

Table 4-34. I/O Port Addresses for DMA Address and Count Registers

Controller	I/O Address	Command Codes
DMA1	000H	Channel 0 base and current address
	001H	Channel 0 base and current word count
	002H	Channel 1 base and current address
	003H	Channel 1 base and current word count
	004H	Channel 2 base and current address
	005H	Channel 2 base and current word count
	006H	Channel 3 base and current address
	007H	Channel 3 base and current word count
DMA2	0C0H	Channel 4 base and current address
	0C2H	Channel 4 base and current word count
	0C4H	Channel 5 base and current address
	0C6H	Channel 5 base and current word count
	0C8H	Channel 6 base and current address
	0CAH	Channel 6 base and current word count
	0CCH	Channel 7 base and current address
	0CEH	Channel 7 base and current word count

Current Address Register

Each DMA channel has a 16-bit current address register for storing the value of the address used during DMA transfers. The address automatically increments or decrements after each transfer and the intermediate values of the address are stored in the register during the transfer. The CPU reads or writes the register in successive 8-bit bytes while in the program state. During autoinitialize, the original values of the current address register are automatically restored from the base count register after EOP* goes active. An I/O device or TC generates EOP*.

Current Word Count Register

Each DMA channel has a 16-bit current word count register for determining the number of transfers performed. The actual number of transfers is one more than the number programmed in the register (e.g., programming a count of 25 results in 26 transfers). The register loads or reads in successive 8-bit bytes by the CPU in the program condition. The word count decrements after each transfer until the value in the register goes from zero to FFFFH. At FFFFH the TC generates an autoinitialize. During autoinitialize, the original values of the current address register are automatically restored from the base count register after EOP* goes active. An I/O device or TC generates EOP*.

Base Address and Base Word Count Registers

Each DMA channel has a pair of base address and base word count registers. These 16-bit registers store the original value of their associated current registers. During autoinitialize, these values restore the current registers to their original values. The base registers are automatically loaded when corresponding current registers are written. The base registers cannot be read by the CPU.

PAGE REGISTERS

The system assigns each channel an 8-bit page register. These page registers extend the 16-bit DMA controller address space to 24 bits (refer to Table 4-35).

Table 4-35. **Page Register Addresses for Each Channel**

I/O Address	Function (In/Out)
081H	8-bit DMA channel 2 (DACK2*)
082H	8-bit DMA channel 3 (DACK3*)
083H	8-bit DMA channel 1 (DACK1*)
087H	8-bit DMA channel 0 (DACK0*)
089H	16-bit DMA channel 6 (DACK6*)
08AH	16-bit DMA channel 7 (DACK7*)
08BH	16-bit DMA channel 5 (DACK5*)
08FH	Refresh cycle page register

PROGRAM CONTROL REGISTERS

Program control registers program DMA transfer operations. Table 4-36 lists the I/O port addresses for the control registers.

Table 4-36. **Control Registers I/O Port Addresses**

Address DMA1	Address DMA2	Operation	Command Codes
008H	0D0H	Read	Read status register
008H	0D0H	Write	Write command register
009H	0D2H	Write	Write request register
00AH	0D4H	Write	Write single mask register bit
00BH	0D6H	Write	Write mode register
00CH	0D8H	Write	Clear byte pointer flip-flop
00DH	0DAH	Read	Read temporary register
00DH	0DAH	Write	Master clear
00EH	0DCH	Write	Clear mask register
00FH	0DEH	Write	Write all mask register bits

COMMAND REGISTER

The 8-bit command register (refer to Table 4-37) controls the overall operation of the DMA subsystem. The CPU programs the command register during a program condition. A reset or a master clear instruction clears the command register.

MODE REGISTER

Each DMA channel has an associated 6-bit mode register (refer to Table 4-38). The mode registers specify the channel operating mode. When the CPU writes to a mode register during a program state, bits 0 and 1 specify the channel selected.

Table 4-37. **Command Register Bit Assignments**

Bit	Function
7	DACK* sense bit 1 = DACK* sense active high 0 = DACK* sense active low (default)
6	DRQ sense bit 1 = DRQ sense active low 0 = DRQ sense active high (default)
5	Extended write bit 1 = Extended write selection 0 = Late write selection
4	Rotating priority bit 1 = Rotating priority scheme is used 0 = Fixed priority used (default)
3	Compressed timing bit 1 = Compressed timing enabled 0 = Normal timing used (default)
2	Controller enable bit 1 = Controller disabled 0 = Controller enabled
1	Channel address hold enable bit 1 = Channel 0 address hold enabled 0 = Channel 0 address hold disabled (default)
0	Memory-to-memory enable bit 1 = Memory-to-memory enabled for Channel 0 and Channel 1 0 = Memory-to-memory disabled (default)

Table 4-38. **Mode Register Bit Assignments**

Bit	Function
7:6	Mode selection bits 00 = Demand mode selected 01 = Single mode selected 10 = Block mode selected 11 = Cascade mode selected
5	Address increment/decrement bit 1 = Address decrement selected 0 = Address increment selected
4	Autoinitialization enable bit 1 = Autoinitialization enabled 0 = Autoinitialization disabled
3:2	Transfer bits 00 = Verify transfer 01 = Write transfer 10 = Read transfer 11 = Illegal
1:0	Channel select bits 00 = Select Channel 0 01 = Select Channel 1 10 = Select Channel 2 11 = Select Channel 3

REQUEST REGISTER

Each DMA channel has an associated 4-bit request register. The system software sets or resets each request register bit separately. A TC or EOP* clears each request register bit. A system reset clears the entire request register. The DMA channel must be in the block transfer mode and the appropriate registers must be set before initiating this request. Table 4-39 lists the format for writing the request register.

Table 4-39. Request Register Update

Bit	Function
7:3	Not used
	2 Set/reset request bit 1 = Set request bit 0 = Reset request bit
1:0	Channel selection bits 00 = Select channel 0 01 = Select channel 1 10 = Select channel 2 11 = Select channel 3

MASK REGISTER

Each DMA channel has an associated mask bit. The mask bit disables incoming DRQs. Each mask bit is set when the associated DMA channel activates EOP* (if autoinitialize is not programmed in the DMA channel). The system software sets or clears each mask register bit separately. A system reset masks all DMA channels and disables all DMA requests.

Two commands are associated with the mask register: Write Single Mask Bit and Write All Mask Bits. Tables 4-40 and 4-41 list the bit positions for both commands.

Table 4-40. Write Single Mask Bit Assignments

Bit	Function
7:3	Not used
	2 Set/reset mask bit 1 = Set mask bit 0 = Reset mask bit
1:0	Select channel mask bits 00 = Select channel 0 mask bit 01 = Select channel 1 mask bit 10 = Select channel 2 mask bit 11 = Select channel 3 mask bit

Table 4-41. **Write All Mask Bit Assignments**

Bit	Function
7:4	Not used
3	Set/reset channel 3 mask bit 1 = Set channel 3 mask bit 0 = Reset channel 3 mask bit
2	Set/reset channel 2 mask bit 1 = Set channel 2 mask bit 0 = Reset channel 2 mask bit
1	Set/reset channel 1 mask bit 1 = Set channel 1 mask bit 0 = Reset channel 1 mask bit
0	Set/reset channel 0 mask bit 1 = Set channel 0 mask bit 0 = Reset channel 0 mask bit

STATUS REGISTER

The status register contains the status of the DMA controller at the time of readout. The status information tells which channel has reached the terminal count and which channel has a DMA request pending. Bits 3:0 are set every time the corresponding channel reaches the terminal count or when a DMA channel activates EOP*. Reading or resetting the status register clears the status register bits. Status register bits 7:4 are set when the corresponding DMA channel requests service. Table 4-42 lists the status register bit positions.

Table 4-42. **Status Register Bit Assignments**

Bit	Function
7	Channel 3 DMA request bit 1 = Channel 3 DMA requested 0 = No DMA request
6	Channel 2 DMA request bit 1 = Channel 2 DMA requested 0 = No DMA request
5	Channel 1 DMA request bit 1 = Channel 1 DMA requested 0 = No DMA request
4	Channel 0 DMA request bit 1 = Channel 0 DMA requested 0 = No DMA request
3	Channel 3 TC reached bit 1 = Channel 3 DMA process completed 0 = Status register is read or reset
2	Channel 2 TC reached bit 1 = Channel 2 DMA process completed 0 = Status register is read or reset
1	Channel 1 TC reached bit 1 = Channel 1 DMA process completed 0 = Status register is read or reset
0	Channel 0 TC reached bit 1 = Channel 0 DMA process completed 0 = Status register is read or reset

TEMPORARY REGISTER

The temporary register is used to hold data during memory-to-memory transfers. Following completion of the transfers, the last word moved can be read by the CPU in the program condition. The temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a reset.

NON-REGISTER PROGRAM CONTROLS

Performing an output to the addresses specified in the program control register I/O executes special software commands. These commands do not depend on any specific bit pattern on the data bus. The three software commands are: clear byte pointer flip-flop, master clear, and clear request mask register.

Clear Byte Pointer Flip-flop

The clear byte pointer flip-flop command resets (clears) the byte pointer flip-flop so the next CPU access to the register contents addresses the low byte. This command must be executed before writing or reading new address or word count information.

Master Clear

This master clear command resets the DMA controller and has the same effect as a hardware reset. During a master clear, the command, status, request, temporary, and internal first/last flip-flop registers reset (clear) and the mask register sets. The DMA controller enters the idle condition after a master clear.

Clear Request Mask Register

The clear request mask register command resets (clears) the mask bits of all four DMA channels.

Communication Ports

5

5.1 INTRODUCTION

This chapter provides reference data for the two RS-232C serial communication ports and one parallel printer port on the 300SX board. Included are all addresses and interrupt levels.

5.2 SERIAL COMMUNICATION PORTS

The 300SX board provides two RS-232C serial communication ports (COM1 and COM2). Each port operates in an asynchronous mode for system software compatibility. An 82C605 multifunction communications controller provides the interface between the communication ports and the CPU. Refer to Table 5-1 for selection of port addresses and interrupt levels. Figure 5-1 shows an internal block diagram of the 82C605.

Table 5-1. Selection of Addresses and Interrupt Levels

Port	Designation	Address	Interrupt
1	COM1	3F8-3FFH	IRQ4
2	COM2	2F8-2FFH	IRQ3

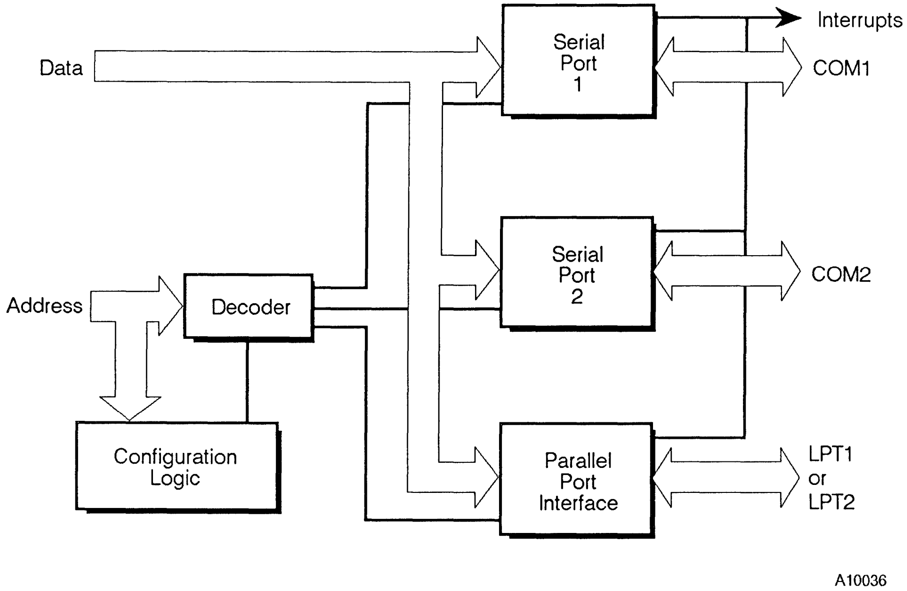


Figure 5-1. 82C605 Internal Block Diagram

5.2.1 Connectors and Pinouts

Connectors J2A, J2B, and J13 provide signals for COM1 and COM2. COM1 signals connect to a DB9 RS-232 connector at J2B. COM2 signals connect to a DB9 RS-232 connector at J2A. Both COM1 and COM2 signals are available at J13 for use with ribbon cable connectors. Refer to Chapter 10 for pinout information on the serial I/O connectors.

5.2.2 Communication Port Registers

Access to the communication port registers is determined by multiplexed address bits SAD2:0 and the divisor register address (DRA) bit. COM1 and COM2 each contain one set of the registers listed in Table 5-2.

Table 5-2. Serial Communication Port Register Selection

DRA	SAD2	SAD1	SAD0	Register
0	0	0	0	Receive buffer (read)
0	0	0	0	Transmit buffer (write)
0	0	1	0	Interrupt flag (read)
0	0	0	1	Interrupt enable
x	0	1	1	Byte format
x	1	0	0	Modem control
x	1	0	1	Line status
x	1	1	0	Modem status
x	1	1	1	Scratchpad
1	0	0	0	Divisor (LSB)
1	0	0	1	Divisor (MSB)

RECEIVE BUFFER REGISTER (RBR)

The RBR holds the incoming data byte. Bit 0 is the least significant bit, and is transmitted and received first. An additional shift register assembles the incoming byte before loading it into the RBR.

TRANSMIT BUFFER REGISTER (TBR)

The TBR holds the data byte to be sent. Bit 0 is the least significant bit, and is transmitted and received first. An additional shift register shifts the outgoing byte to the TDX pin.

INTERRUPT ENABLE REGISTER (IER)

The low order four bits of the IER control the enabling of each of the four possible types of interrupts (refer to Table 5-3). Setting the corresponding bit to 1 enables an interrupt. The interrupt flag register contents are not valid when all the interrupts are disabled.

Table 5-3. Interrupt Enable Register Bit Assignments

Bit	Function
0	1 = Generate an interrupt when the receive buffer contains valid data 0 = Disabled
1	1 = Generate an interrupt when the transmit buffer is empty 0 = Disabled
2	1 = Generate an interrupt when an error (overrun, parity, framing, or break) has been encountered. The status register must be read to determine the type of error. 0 = Disabled
3	1 = Generate an interrupt when one of the bits in the modem status register changes state.
4:7	0 These four bits are set to 0.

INTERRUPT FLAG REGISTER (IFR)

When accessed, the IFR bits report and record the highest pending interrupt (refer to Table 5-4). The following four levels of prioritized interrupts are used:

- Line status (highest priority)
- Receive buffer full
- Transmit buffer empty
- Modem status (lowest priority)

Table 5-4. Interrupt Flag Register Bit Assignments

Bit	Function
0	0 = Indicates an interrupt is pending and bits 1 and 2 can be read to determine the source of the interrupt 1 = Indicates no interrupts are pending
1:2	1-1 = Line status interrupt 0-1 = Receive buffer full interrupt 1-0 = Transmit buffer empty interrupt 0-0 = Modem status interrupt
3:7	0 Set to zero

BYTE FORMAT REGISTER (BFR)

The read/write BFR contains format information for the serial line.

Word Length (WL) Bits

The WL bits (bit 0 and 1) specify the word length for received and transmitted characters (refer to Table 5-5). Start, stop, and parity bits are not included in the word length value.

Table 5-5. Word Length Bit Assignments

Bit 0	Bit 1	Word Length
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

Stop (SP) Bit

The combination of the SP bit (bit 2) and the WL bits determines the number of stop bits used with each transmitted character (refer to Table 5-6). The receiver ignores additional stop bits beyond the first one, regardless of the number of stop bits transmitted.

Table 5-6. Stop Bits per Transmitted Character

Bit 2	Word Length	Number of Stop Bits
0	-	1
1	5 bits	1.5
1	6 bits	2
1	7 bits	2
1	8 bits	2

Parity Generator (PG) Bit

The PG bit (bit 3) enables parity generation (during transmission) and checking (during receipt). The parity bit is always after the last data bit but before the first stop bit. Setting the PG bit to 1, generates a parity bit of the proper state (0 or 1). The sum (carry ignored) of all data bits, plus the parity bit, produces either an even (even parity) or odd (odd parity) value.

Even Parity (EP) Bit

The EP bit (bit 4) controls parity sense. When set to 1, the EP bit indicates even parity. Transmitting an odd number of logic 1's generates a parity error. When set to 0, the EP bit indicates odd parity. Unless it is set to 1, the EP bit is ignored.

Force Parity (FP) Bit

The FP bit (bit 5) ensures that the parity bit and sense (even or odd) match regardless of the sum normally used to determine parity.

Break Bit

Setting the break bit (bit 6) to 1, forces the transmitted data output pin to a spacing or logic 0 condition. Setting the break bit to 0, terminates the break condition.

Divisor Register Address (DRA) Bit

Setting the DRA bit (bit 7) to 1, permits access to the divisor registers. Setting the DRA bit to 0, allows access to all other internal registers.

MODEM CONTROL REGISTER (MCR)

The byte-wide MCR manages the connection to an external modem or data set. Table 5-7 lists the bit assignments.

Table 5-7. Modem Control Register Bit Assignments

Bit	Function
0	1 = Force DTR* to its active state 0 = Force DTR* to its inactive state
1	1 = Force RTS* to its active state 0 = Force RTS* to its inactive state
2	1 = Force the RI* bit (bit 6) of the MSR active when the serial port is in a loopback mode 0 = Force the RI* bit inactive
3	1 = Force the DCD* bit (bit 7) of the MSR active when the serial port is in a loopback mode 0 = Force the DCD* bit inactive
4	Used for self-diagnostic purposes
5:7	Set to 0

LINE STATUS REGISTER (LSR)

The byte-wide LSR (read-only) supplies serial link status information to the CPU. One of the conditions flagged by bits 1 through 4 of the LSR generates a line status interrupt.

Receive Buffer Full (RBF) Bit

When set to 1, the RBF bit (bit 0) indicates an incoming character has been transferred from the receive shift register to the receive buffer. Reading the receive buffer clears the RBF bit.

Overrun Error (OE) Bit

When set to 1, the OE bit (bit 1) indicates a new character was transferred into the receive buffer before the previously received character was read by the CPU. The previously received character is lost. Reading the LSR clears the OE bit.

Parity Error (PE) Bit

When set to 1, the PE bit (bit 2) indicates a parity error was detected (received character has a parity other than that selected). Reading the LSR clears the PE bit.

Framing Error (FE) Bit

When set to 1, the FE bit (bit 3) indicates an incoming character has no stop bit after the last data bit or (if parity is enabled) after the parity bit. Reading the LSR clears the FE bit.

Break Interrupt (BI) Bit

When set to 1, the BI bit (bit 4) indicates a space condition (logic 0) is present on the corresponding RXD (RXD2 or RXD1) line for an entire character time. A character time equals a start bit time, data bit times, parity bit time, and stop bit time. Reading the LSR clears the BI bit.

Transmit Buffer Empty (TBE) Bit

When set to 1, the TBE bit (bit 5) indicates an outgoing character is loaded from the transmit buffer into the transmit shift register. If the TBE bit is set (bit one of the IER), an interrupt occurs. Writing a character to the transmit buffer clears the TBE bit.

Transmitter Empty (TEMT) Bit

When set to 1, the TEMT bit (bit 6) indicates the transmit buffer and the transmit shift register are both empty. If a character is present in either the transmit buffer or the transmit shift register, the TEMT bit is cleared.

MODEM STATUS REGISTER (MSR)

The byte-wide MSR holds the current value of the MODEM control lines. Each time one of MODEM control lines changes state, the MSR sets the corresponding bit to a 1. Reading the MSR clears the previously set bits. When set to 1, bits 0, 1, 2, and 3 generate an interrupt, provided the corresponding interrupt bit is enabled.

Clear to Send Changed (CSC) Bit

When set to 1, the CSC bit (bit 0) indicates the corresponding CTS* line (CTS2* or CTS1*) changed state since the last time the MSR was read.

Data Set Ready Changed (DSRC) Bit

When set to 1, the DSRC (bit 1) indicates the corresponding DSR* line (DSR2* or DSR1*) changed state since the last time the MSR was read.

Rising Edge of Ring Indicator (RERI) Bit

When set to 1, the RERI bit (bit 2) indicates the corresponding RI* line (RI2* or RI1*) changed state since the last time the MSR was read.

Data Carrier Detect Changed (DCD) Bit

When set to 1, the DCD bit (bit 3) indicates the corresponding DCD* line (DCD2* or DCD1*) changed state since the last time the MSR was read.

Clear to Send (CS) Bit

The CS bit (bit 4) is the complement of the corresponding CTS* line. When in the diagnostic loopback mode, the CS bit is identical to the RTS bit in the MCR.

Data Set Ready (DS) Bit

The DS bit (bit 5) is the complement of the corresponding RI* line. When in the diagnostic loopback mode, the DS bit is identical to the DTR bit in the MCR.

Ring Indicator (RI) Bit

The RI bit (bit 6) is the complement of the corresponding DSR* line. In diagnostic loopback mode, the RI bit is controlled by bit 2 of the MCR.

Data Carrier Detect (DRD) Bit

The DRD bit (bit 7) is the complement of the corresponding DCD* line. In diagnostic loopback mode, the DRD bit is controlled by bit 3 of the MCR.

DIVISOR REGISTERS (LSB AND MSB)

The two Divisor registers, Divisor register LSB (DRL) and Divisor register MSB (DRM), contain a 16-bit divisor used to generate the baud rate of the serial ports from the 300SX board's 1.8432 MHz crystal. DRL holds the least-significant byte and DRM holds the most-significant byte of the divisor.

Table 5-8 details the contents of the Divisor registers for various baud rates. All values in Table 5-8 are decimal.

Table 5-8. Divisor Register Values for Various Baud Rates

Baud Rate	Register Value	Percent Error
50	2304	none
75	1536	none
110	1047	0.026
134.5	857	0.058
150	768	none
300	384	none
600	192	none
1200	96	none
1800	64	none
2000	58	0.69
2400	48	none
3600	32	none
4800	24	none
7200	16	none
9600	12	none
19200	6	none
38400	3	none
56000	2	2.86

5.3 PARALLEL PORT

The 300SX board is equipped with one parallel printer port, configurable as either LPT1 or LPT2. The parallel printer port, at connector J1, provides a one-way interface to a printer. The parallel port is also available at connector J11 (if installed) for use with ribbon cable connectors. Refer to Chapter 10 for pinout information on the parallel printer port connectors.

5.3.1 Data Latch Register (DLR)

The read/write DLR is located at on offset of zero from the base address of the parallel port. Data written to the DLR is transmitted to the printer.

5.3.2 Printer Status Register (PSR)

The read-only PSR is located at on offset of 1H from the base address of the parallel port. Bits 2-0 of the PSR are reserved.

BUSY BIT

The busy bit (bit 7) reflects the state of the BUSY* input pin. When set to 0, the busy bit indicates that the printer is busy and cannot accept data. When set to 1, the busy bit indicates that the printer is ready to accept data.

ACKNOWLEDGE (ACK) BIT

The ACK bit (bit 6) reflects the state of the ACK input pin. When set to 0, the ACK bit indicates the printer received a character and is ready to accept another. When set to 1, the ACK bit indicates the printer is still reading the last character sent.

PAPER EMPTY (PE) BIT

The PE bit (bit 5) reflects the state of the PE* input pin. When set to 1, the PE bit indicates there is no paper in the printer. When set to 0, the PE bit indicates the presence of paper.

SLCT (SO) BIT

The SO bit (bit 4) reflects the state of the SLCT* input pin. When set to 1, the SO bit indicates the printer is online. When set to 0, the SO bit indicates the printer is not selected.

ERROR (ER) BIT

The ER bit (bit 3) reflects the inverted state of the ERROR input pin. When set to 0, the ER bit indicates an error condition has been detected. When set to 1, the ER bit indicates no errors have been detected.

5.3.3 Printer Controls Register (PCR)

The read/write PCR is located at an offset of 2H from the base address of the parallel port. Bits 7-5 of the PCR are reserved and reset to 0.

IRQ ENABLE (IE) BIT

The IE bit (bit 4) enables or disables interrupts resulting from the printer ACK signal. Setting the IEB to 1, enables interrupts. A deasserted ACK by the printer, interrupts the CPU on the IRQ line specified in the configuration RAM.

SLCTIN (SI) BIT

The SI bit (bit 3) drives the SLCTIN output pin. Setting the SI bit to 1, selects the printer.

INIT (IN*) BIT

The IN* bit (bit 2) controls the INIT output pin. Setting the IN* bit to 0, starts the printer.

AUTO FEED (AF) BIT

The AF bit controls the AUTOFD output pin. Setting the AF bit to 1, causes the printer to line feed after each printed line.

STROBE (ST) BIT

The ST bit controls the STROBE output pin. Setting the ST bit to 1, generates an active high pulse (0.5 μs pulse minimum) which clocks data into the printer.

Keyboard and Mouse Controller

6

6.1 INTRODUCTION

The 300SX board supports a 101- or 102-key enhanced keyboard and a three button mouse. An Intel 8742 microcontroller controls the keyboard and mouse system interface (see Figure 6-1). This chapter describes the keyboard and mouse interface through the 8742 microcontroller.

6.2 KEYBOARD AND MOUSE CONTROLLER SYSTEM INTERFACE

The keyboard and mouse controller communicates with the system through an 8-bit read-only status register at I/O address 64H, a read only output buffer at I/O address 60H, and an input buffer. The input buffer consists of two parts: a data byte and a command byte written at addresses 60H and 64H, respectively.

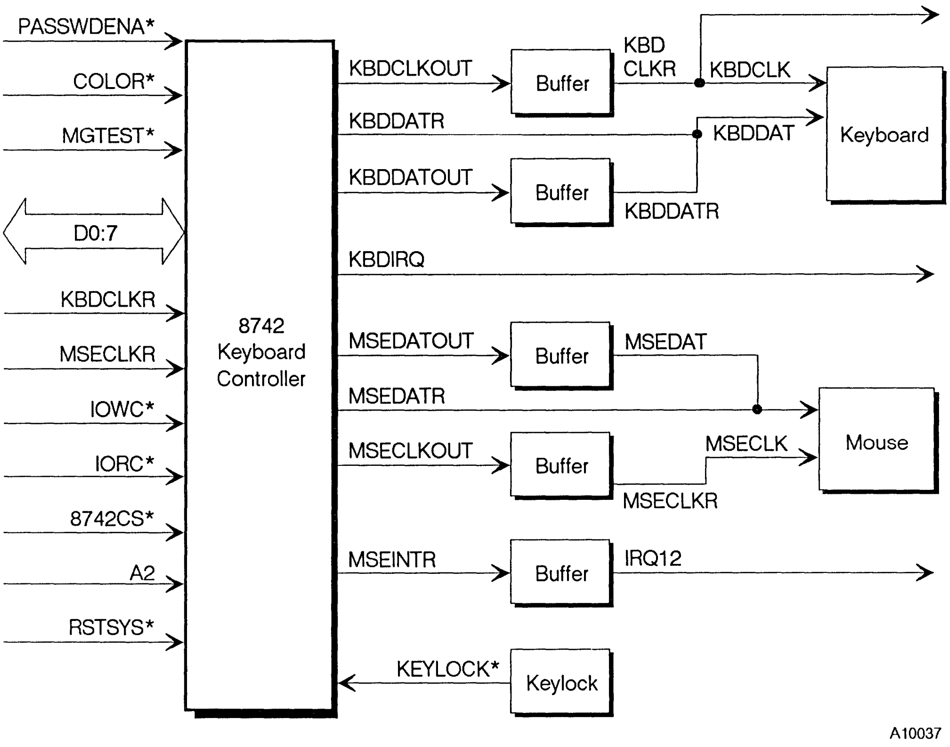


Figure 6-1. **Keyboard/Mouse Controller**

6.2.1 Status Register

The controller status register (refer to Table 6-1) contains information about the state of the controller and system interface.

Table 6-1. Status Register Bit Definition

Bit	Function
0	Output Buffer (I/O Address 60H) Full 0 = No data from the keyboard in the output buffer (DBBOUT) 1 = The keyboard controller loaded the output buffer with data. This bit is set to 0 when the CPU reads it (DBBIN).
1	Input Buffer (I/O Address 60H) Full 0 = No data from the CPU in the input buffer 1 = Data from the CPU in the input buffer. This bit is set to 0 when the keyboard controller reads the buffer.
2	System Flag 0 = Power-on reset occurred 1 = Self-test successful
3	Command/Data 0 = A data byte written (port 60H) 1 = A command byte written (port 64H)
4	Front Panel Keylock Status 0 = Keyboard inhibited 1 = Keyboard not inhibited
5	Auxiliary (Mouse) Device Output Buffer Full 0 = Output buffer is keyboard device data. 1 = Output buffer is auxiliary device data.

Table 6-1. Status Register Bit Definition *(continued)*

Bit	Function
6	General Time-Out 0 = Data reception from the keyboard/auxiliary device terminated normally within the programmed time-out delay. 1 = Data from the keyboard/auxiliary device did not terminate normally within the programmed time-out delay.
7	Parity Error 0 = Last byte of data received from the keyboard/auxiliary device had odd parity (no error). 1 = Last byte of data received from the keyboard/auxiliary device had even parity (error).

6.2.2 Output Buffer

The controller output buffer is an 8-bit read-only register at I/O address 60H. The controller sends keyboard scan codes, command requested data bytes, and mouse data to the system via the output buffer. The significant data in the output buffer can be read only when the output buffer full bit (bit 0) of the status register equals 1.

6.2.3 Input Buffer

The controller input buffer is an 8-bit write-only register at I/O addresses 60H and 64H. Data can be written to the input buffer only if the input buffer full bit (bit 1) of the status register equals 0.

Writing to address 60H clears the command/data bit (bit 3) of the status register. Once cleared, the controller processes the data in the input buffer as a data byte. Data written to address 60H is sent to the keyboard, unless a system command instructs the controller to wait for a data byte.

Writing to address 64H sets the command/data bit (bit 3) of the status register to 1. Once set, the controller processes the data in the input buffer as a command byte.

6.2.4 Controller Commands

The CPU uses controller commands (refer to Table 6-2) to control the operation of the controller and sense its status. The CPU writes controller commands into the input buffer through I/O port address 64H.

Table 6-2. Controller Commands

Code (Hex)	Description
20	Read controller command byte
21:3F	Read 8742 internal RAM locations 21-3F
60	Write controller command byte
61:7F	Write internal RAM locations 21-3F
A7	Disable auxiliary device (mouse)
A8	Enable auxiliary device
A9	Auxiliary interface test
AA	Self-test
AB	Keyboard test interface
AD	Disable keyboard
AE	Enable keyboard
C0	Read input port
C1	Poll input port low
C2	Poll input port high
D0	Read output port
D1	Write output port
D2	Write keyboard output buffer
D3	Write auxiliary output buffer
D4	Write to auxiliary device
E0	Read test input port
E1:EF	Set/Clear output pin
F0:FF	Output pulse

6.3 KEYBOARD/MOUSE INTERFACE

The keyboard and mouse connect to the controller through bidirectional synchronous serial interface cables. Refer to Chapter 10 for the pin assignments of the keyboard and mouse connectors. The controller supplies the keyboard and mouse with DC power of +5V +10% at a maximum current of 300 mA.

The controller, keyboard, and mouse communicate using data and clock lines for synchronous serial communication. Open-collector drivers, at both ends of the cable, drive the data and clock lines.

At power-up, the keyboard scans the signals on the clock and data lines and establishes a line protocol. A bidirectional serial interface in the keyboard converts the clock and data signals and transfers them to and from the keyboard through the keyboard cable. Signals include keyboard control commands from the system and keyboard scan and acknowledgment codes transferred to the controller.

The serial data from the keyboard is called a scan code. Each keyboard key has an associated 11-bit scan code. Pressing and releasing a key generates a make or break scan code. The keyboard detects all keys pressed and transfers each scan code in the correct sequence to the controller.

The controller receives serial data from the keyboard, checks the parity of the data, and translates the 8-bit scan code into system codes. It also interrupts the CPU to transfer data to the system. The controller interrupts the system when data is placed in its output buffer, or waits for the system to poll its status register to determine when data is available.

The controller transfers various commands to the keyboard at any time. When the controller transfers data to the keyboard, it sets the data line to an inactive state and allows the clock line to go active. This action serves as a request-to-send (RTS) and a start bit. Setting the clock line to an inactive state inhibits keyboard transmission.

If the controller has to transfer data to the keyboard during a keyboard transfer, the controller clamps the clock signal line to request a keyboard transfer halt. The clock line must remain low for at least 60 ms.

During the keyboard basic assurance test (BAT) or when no data transfer occurs, the clock line remains active (high). The keyboard holds the data line active (high).

An inactive signal has a value between 0V and +0.7V (logical 0). An active signal has a value between +2.4V and +5.5V (logical 1). These voltages are measured between a signal source and the DC network ground.

6.3.1 Keyboard/Mouse Data Stream

The 8-bit data stream, transferred serially over the data line, consists of one start bit, eight data bits, one odd parity bit, and one stop bit (refer to Table 6-3). A logic 1 indicates an active level and a logical 0 indicates an inactive level. The parity bit is either 1 or 0. The 8 data bits plus the parity bit always have an odd number of 1's.

Table 6-3. Data Stream Bits

Bit	Function
1	Start bit (always 0)
2	Data bit 0 (LSB)
3	Data bit 1
4	Data bit 2
5	Data bit 3
6	Data bit 4
7	Data bit 5
8	Data bit 6
9	Data bit 7 (MSB)
10	Parity bit (odd parity)
11	Stop bit (always 1)

6.3.2 Receiving Data from the Keyboard

The keyboard transfers data to the controller in a serial format using an 11-bit frame. The first bit is a start bit, followed by 8 data bits, an odd parity bit, and a stop bit. A clock, supplied by the keyboard, synchronizes the data transfer.

Before transferring data, the keyboard checks for a transmission-inhibit or controller RTS status on the clock and data lines. If transmission is inhibited (clock line inactive), keycodes are transferred to the keyboard buffer. During controller RTS, data is also stored in the keyboard buffer, while the keyboard receives controller data.

The keyboard transfers data to the controller only when both the clock and data signals are active. At the end of a transfer, the controller disables the interface until the system accepts the data byte.

If a parity check error occurs, the controller signals the keyboard to transfer the data again. If the controller does receive the data correctly (after a set number of retries), an FFH code is sent to the controller output buffer. The parity bit in the status register is also set indicating a receive parity error.

The controller times each data byte transfer from the keyboard. If a keyboard transfer does not end within 2 ms, the controller sets the receive time-out bit in the status register and writes an FFH code to its output buffer. No retries are attempted on a receive time-out error.

The following commands are sent from the keyboard to the controller.

OVERRUN OR KEY DETECTION ERROR (00H OR FFH)

If the keyboard uses scan code set 1, the code equals FFH. For sets 2 and 3, the code equals 00H. The conditions are as follows:

- The keyboard sends a key detection error character if conditions in the keyboard make it impossible to identify a switch closure.
- When the buffer in the keyboard is full, an overrun character replaces the last transmitted code in the buffer. This code is sent to the controller when it reaches the top of the buffer queue.

KEYBOARD ID (83ABH)

The keyboard ID consists of two bytes, 83ABH. The keyboard responds to the read ID with ACK, discontinues scanning, and sends the two ID bytes. The keyboard sends the low byte first, followed by the high byte. The keyboard resumes scanning following an output of the keyboard ID.

BAT COMPLETION CODE (AAH)

Each time the system is powered up, the keyboard performs a self-test operation called the basic assurance test (BAT). The BAT consists of a keyboard processor test, a checksum of the ROM, and a RAM test. Activity on the clock and data lines is ignored during a BAT. The keyboard sends an AAH command to the controller following satisfactory completion of the BAT. Any other code indicates a keyboard failure.

BAT FAILURE CODE (FCH)

If a BAT failure occurs, the keyboard sends FCH, discontinues scanning, and waits for a controller response or reset.

ECHO (EEH)

When the controller issues the echo command to the keyboard, the keyboard sends EEH as a response to the controller.

ACKNOWLEDGE (FAH)

The keyboard issues an acknowledge to any valid input other than an echo or a resend command. If the keyboard is interrupted while sending an acknowledge, it discards acknowledge and responds to the new command.

RESEND (FEH)

The keyboard sends the resend command when it receives an invalid input or any input with incorrect parity. This command signals the controller to transfer the input again.

6.3.3 Sending Data to the Keyboard

The controller transfers data to the keyboard in the same serial format used to receive data from the keyboard. Before the controller transfers data to the keyboard, it checks the keyboard and determines whether it is transferring data or not. If the keyboard is transferring data, but has not reached the tenth clock signal, the controller overrides the keyboard output by setting the keyboard clock line inactive. If the keyboard transfer is beyond the tenth clock signal, the controller waits until the keyboard completes its transfer before transferring data.

If the controller overrides the keyboard output, or if the keyboard is not transferring data, the controller sets the clock line inactive for more than 60 μs while preparing to transfer data. When the controller transfers the start bit, the clock line goes active.

Each controller command or data transmission to the keyboard requires a response before the controller transfers its next output. After the keyboard receives a controller command, it returns an acknowledge code to the controller. If the keyboard response is invalid or has a parity error, FEH is placed in the controller output buffer and the transmit time-out or parity error bits are set in the status register.

The controller sets a programmed time limit (20 ms to 25 ms) for the keyboard to respond. If the keyboard cannot complete the send-out data process within this time period, the controller places FEH in its output buffer and sets the transmit and receive time-out error bits in its status register. No retries are attempted by the controller for any transmission error. The following commands are sent from the controller to the keyboard.

SET/RESET STATUS INDICATORS (EDH)

The set/reset status indicators command activates or deactivates the three LED indicators (Num Lock, Caps Lock, and Scroll Lock) on the keyboard.

The keyboard responds to the command byte with ACK, discontinues scanning, and waits for the option byte from the controller. The contents of the option byte following the command determines the parameter for setting the LED mode. Bits 0, 1, and 2 of this byte control the Scroll Lock, Num Lock, and Caps Lock LEDs respectively. Bits 3 through 7 are reserved.

If the bit for an indicator equals 1, the indicator turns on. If the bit equals 0, the indicator turns off. The keyboard responds to the option byte with an acknowledge code, sets the indicators, and resumes scanning.

ECHO (EEH)

The echo command tests the keyboard command process. When the keyboard receives this command, it issues an EEH response, and continues scanning.

INVALID COMMAND (EFH AND F1H)

EFH and F1H are invalid commands.

SELECT ALTERNATE SCAN CODES (F0H)

The select alternate scan codes command instructs the keyboard to select one of three sets of scan codes. The keyboard acknowledges receipt of this command with an acknowledge code, then clears both the output buffer and the typematic key (if active). When the controller sends the option byte, the keyboard responds with another acknowledge code. An option byte value of 01H selects scan code set 1, 02H selects set 2, and 03H selects set 3. Byte value 00H causes the keyboard to respond with an acknowledge code and send a byte signaling the controller which scan code set is in use.

READ ID (F2H)

The read ID command requests identification information from the keyboard. The keyboard responds with an acknowledge code, discontinues scanning, and sends the two keyboard ID bytes. The second byte must follow completion of the first byte within 500 μs. After the output of the second ID byte, the keyboard resumes scanning.

SET TYPEMATIC RATE/DELAY (F3H)

This command sets the typematic rate and delay. The keyboard responds with an acknowledge code, stops scanning, and waits for the controller to issue the rate/delay value byte. Once issued, the keyboard responds with another acknowledge code, sets the rate and delay to the values indicated, and resumes scanning.

The contents of the rate/delay value byte following the command determines the parameters for these two functions. Bits 4:0 set the typematic rate, bits 5 and 6 set the delay parameter, and bit 7 is set to zero.

The following equations show the calculation for the delay and the typematic rate:

- Delay = $(1 + C) \times 250$ ms ± 20%
- Period T = $(8 + A) \times 2^{B} \times 0.00417$ seconds
- Typematic Rate = 1/T ± 20% (default is 10 characters per second)

Where:
C = Binary value of bits 5 and 6 (default equals 500 ms)
A = Binary value of bits 2, 1, and 0
B = Binary value of bits 4 and 3
T = Interval from one typematic output to the next

ENABLE (F4H)

When the keyboard receives the enable command, it responds with an acknowledge code, clears its output buffer, clears the last typematic key, and starts scanning.

DEFAULT DISABLE (F5H)

The default disable command resets all conditions to the power-on default state. The keyboard responds with an acknowledge code, clears its output buffer, sets the default key types and typematic rate/delay, and clears the last typematic key. The keyboard stops scanning and waits for further instructions from the controller.

SET DEFAULT (F6H)

The set default command is similar to default disable command F5H. However, the keyboard continues scanning instead of stopping and waiting for further instructions.

SET ALL KEYS (F7H, F8H, F9H, FAH)

The set all keys commands, F7H, F8H, F9H, and FAH, instruct the keyboard to set all keys to typematic, make/break, make, and typematic/make/break respectively. The keyboard responds with an acknowledge code, clears its output buffer, sets all keys to the type indicated by the command, and continues scanning. Although these commands are sent using any scan code set, they affect only scan code set 3.

SET KEY TYPE (FBH, FCH, FDH)

The set key type commands, FBH, FCH, and FDH, instruct the keyboard to set individual keys to typematic, make/break, and make respectively. The keyboard responds with an acknowledge, clears its output buffer, and prepares to receive key identification. The controller identifies each key by its scan code value as defined in scan code set 3. Only scan code set 3 values are valid for key identification. The type of each identified key is set to the value indicated by the command.

RESEND (FEH)

The controller transfers the resend command when it detects an error in any transfer from the keyboard. It requests the keyboard to resend a code that was detected as an error. This command transfers only after a keyboard transfer and before the controller allows the next keyboard output. When a resend command is received, the keyboard transfers the previous output again. If the previous output was resend, the keyboard transfers the last byte before the resend command.

RESET (FFH)

The controller issues the reset command to start a program reset and a keyboard internal self test. The keyboard responds with an acknowledge code and ensures the controller accepts the acknowledge code before executing the command. The controller raises the clock and data lines for a minimum of 500 μs after receiving an acknowledge code from the keyboard. The keyboard remains disabled from the time it receives the reset command until the controller responds to the acknowledge code, or until another command overrides the previous command. After the controller responds to the acknowledge code, the keyboard initializes and performs the BAT. After returning the completion code, the keyboard defaults to scan code set 2.

6.3.4 System-to-Mouse Commands

The write to auxiliary device command (D4H) instructs the 8742 to transmit the next byte it receives to the auxiliary device. All commands written to the mouse must be written must be preceded by the write to auxiliary device command to port 64H, followed by the desired mouse command. All mouse commands must be written to port 60H. If the write to auxiliary device command to port 64H is not executed first, all mouse commands will be directed to the keyboard.

RESET SCALING (E6H)

The reset scaling command resets the scaling to 1:1.

SET SCALING (E7H)

The set scaling command sets the scaling to 2:1. This command can only be used when the mouse is in stream mode. When in stream mode, the current X/Y coordinates values are converted to new values each time the sample period expires. In 2:1 scaling, the new relationship between the input and output values is as follows:

Input	Output
0	0
1	1
2	1
3	3
4	6
5	9
N (>= 6)	2.0 x N

SET RESOLUTION (E8H)

The set resolution command is a two byte command. The second byte (also written to port 60H) is interpreted as a resolution in counts per mm. There are four possible resolutions. The relationship between the second byte and the resolution is as follows:

Byte	Resolution
00H	1 count per mm
01H	2 counts per mm
02H	4 counts per mm
03H	8 counts per mm

STATUS REQUEST (E9H)

The status request command generates a three byte status report. The format of the mouse status request bytes is shown in Table 6-4.

Table 6-4. Format of Status Request Bytes

Byte	Function
3	Sampling Rate Bit 7 = Most significant bit Bit 0 = Least significant bit
2	Resolution Bit 7 = Most significant bit Bit 0 = Least significant bit
1	Mouse status Bit 7 = Reserved Bit 6 = 0 Stream mode 1 Remote mode Bit 5 = 0 Disabled 1 Enabled Bit 4 = 0 Scaling 1:1 1 Scaling 2:1 Bit 3 = Reserved Bit 2 = 1 Left mouse button pressed Bit 1 = Reserved Bit 0 = 1 Right mouse button pressed

SET STREAM MODE (EAH)

In set stream mode, the mouse transmits data to the system each time a mouse button is pressed or released, or each time the mouse detects a unit of movement. The mouse data sample rate determines the maximum number of times per second that mouse data can be transmitted to the system. If no button is pressed or if the mouse is not moved, no data is transmitted. The set stream mode command enables the stream mode.

READ DATA (EBH)

The read data command forces the transmission of one mouse data packet. The read data command is valid in both stream mode and remote mode.

RESET WRAP MODE (ECH)

The reset wrap mode command resets the mouse to normal operation.

SET WRAP MODE (EEH)

This command sets wrap mode; the mouse "echo" mode. With the exception of the reset wrap mode (ECH) and reset mouse (FFH) commands, the mouse will echo all data and commands received from the system.

SET REMOTE MODE (F0H)

This command sets remote mode; the mouse data can only be transmitted in reply to a read date command.

READ DEVICE TYPE (F2H)

The read device type command reads the mouse ID byte. The mouse returns a value of 00H to the read device command.

SET SAMPLING RATE (F3H)

This command sets the sampling rate of the mouse. The sampling rate is defined as the number of times per second that the system checks for mouse data. This is a two byte command. The set sampling rate command (F3H) must be followed by a second byte that represents the hex value of the sampling rate. The allowable values are defined below:

Hex Value	Sampling Rate
0AH	10 samples/second
14H	20 samples/second
28H	40 samples/second
3CH	60 samples/second
50H	80 samples/second
64H	100 samples/second
C8	200 samples/second

ENABLE (F4H)

The enable command enables data transmissions if the mouse has been set to stream mode. This command has no effect in remote mode.

DISABLE (F5H)

The disable command disables data transmissions if the mouse has been set to stream mode. This command has no effect in remote mode.

SET DEFAULT (F6H)

The set default command reinitializes the mouse to its power-on default state. The mouse power-on default state is shown below:

Sampling rate	100 samples/second
Scaling	Linear scaling
Mode	Stream mode
Resolution	4 counts/mm
Transmissions	Disabled

RESEND (FEH)

The resend command is issued by the system in response to transmission errors from the mouse. The mouse responds to this command by retransmitting its last data packet.

RESET (FFH)

The reset command instructs the mouse to run its internal self-test routine. This command puts the mouse into reset mode.

6.3.5 Mouse-to-System Replies

There are two mouse-to-system replies. Both replies are related to command processing and are read by the system at port 60H.

ACKNOWLEDGE (FAH)

The mouse replies with an acknowledge (FAH) whenever it receives a valid command from the system. Unlike mouse serial data packets, the acknowledge reply is not stored in a buffer in internal memory, but is discarded immediately after it is transmitted. If a new command is received while the mouse is in the acknowledge reply process, the mouse discards the acknowledge reply and begins processing the new command immediately.

NOTE

The reset wrap mode (ECH) and reset (FFH) commands are exceptions to mouse acknowledge response. The mouse does not respond with an acknowledge to either of these commands.

RESEND (FEH)

The mouse replies with a resend (FEH) whenever it receives an invalid command from the system. Two invalid commands in succession cause the mouse to send the error code FCH to the system. A single isolated invalid command does not affect mouse processing in any way. The mouse ignores single invalid commands and maintains its present operational state.

Video Display Subsystem

7

7.1 INTRODUCTION

This chapter describes the onboard video display subsystem (VDS). The chapter contains an overview of the video display controller, the graphics chip set, and the digital-to-analog converter (DAC).

7.2 VIDEO DISPLAY SUBSYSTEM OVERVIEW

The VDS consists of a graphics chip set, a digital to analog convertor (DAC), 256K of dynamic random access memory (DRAM), and support logic (see Figure 7-1). The chip set supports high resolution graphics and alphanumeric display modes for both monochrome and color, and for high resolution variable frequency monitors. Video outputs are either TTL or analog voltage levels. An external palette, provided by the DAC, supports up to 256 colors from a possible 262,144 colors. A 32-bit data bus supports video memory configurations up to 256K (four 64K planes). The graphics chip set contains a hardware implementation of video graphics array (VGA), enhanced graphics adapter (EGA), color graphics adapter (CGA) monochrome display adapter (MDA), and Hercules graphics adapter (HGC).

NOTE

The VDS does not support a video feature connector.

The VDS is controlled by the onboard video BIOS. The video BIOS is located at memory address E0000H-E7FFFH and resides in two 27512 EPROM devices. The video BIOS can be optionally mirrored to C0000H-C7FFFH during system setup.

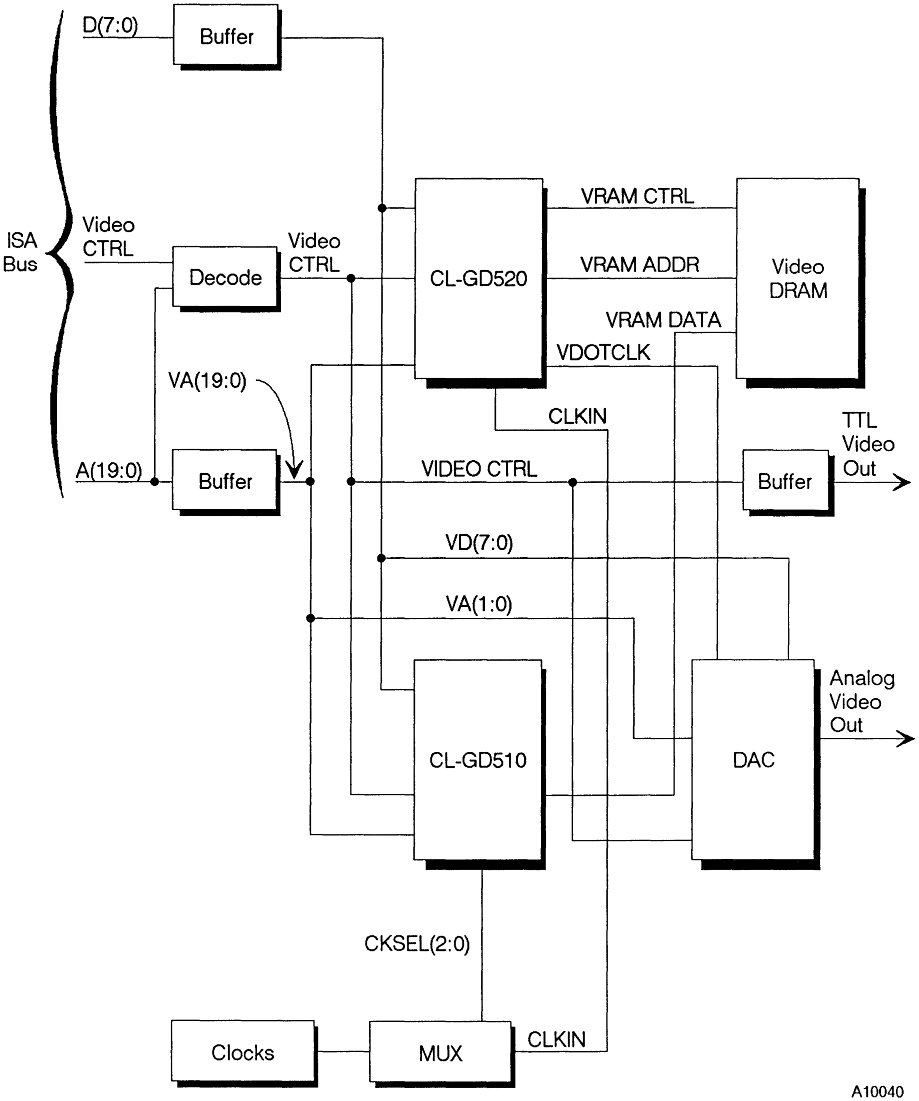


Figure 7-1. Video Display Subsystem Block Diagram

7.3 GRAPHICS CHIP SET

The Cirrus Logic GD510A Graphics Attributes and GD520A Sequencer/CRT Controller devices comprise the graphics chip set on the 300SX board. The four major operations supported by the graphics chip set are as follows:

- CPU access to the registers within the graphics chip set
- CPU access to video memory
- Video memory refresh
- Display access to video memory

7.3.1 CPU Access to Registers

The CPU accesses the graphics chip set registers by setting up 20-bit addresses and generating IOR* and IOW* signals to read or write 8-bit data.

7.3.2 CPU Access to Video Memory

CPU access to video memory is channeled through the graphics chip set. The CPU must set up the proper address, data, and timing parameters in the graphics chip set registers in order to connect one of the four video memory byte plane buses to the CPU data bus.

The graphics chip set also contains an intelligent address sequencer that allocates video memory cycles not only to the CPU, but also to the DRAM refresh controller and the display CRT controller.

7.3.3 Video Memory Refresh

Memory bandwidth is allocated to each process according to the actual real time needs of the process. This ensures efficient use of the available bandwidth. During horizontal and vertical retrace intervals the display is blanked, which frees up memory bandwidth for host access and/or memory refresh.

7.3.4 Display Access to Video Memory

The GD520A sequencer/CRT controller device works closely with the GD510A graphics/attributes device in all video modes. The GD510A contains the video memory data interface as well as the video outputs to the monitor. Display data is latched in the GD510A after the GD520A determines where the data is located. The GD510A contains video shift registers to interface to a monitor.

7.4 DIGITAL-TO-ANALOG CONVERTOR (DAC)

The digital-to-analog convertor device (DAC0631) contains a 256 x 18-bit color palette. The color palette provides a display of 256 colors selected from a total of 262,144. Three internal 6-bit video DACs convert the 18-bit digital signal to a red, green, and blue (RGB) analog output (0V to 0.7V).

7.4.1 DAC to CPU Interface

The DAC to CPU interface consists of three internal registers: pixel address, color value, and pixel mask. The contents of the color palette are accessed through the color value and pixel address registers. Register select signals, RS0 and RS1, determine which register is accessed by the CPU (refer to Table 7-1).

Table 7-1. Register Select Signals Bit Assignments

RS0	RS1	Register
0	0	Pixel Address (write only)
1	1	Pixel Address (read only)
1	0	Color Value
0	1	Pixel Mask

PIXEL ADDRESS REGISTER (PAR)

The PAR is a byte-wide latch that receives and latches address information applied to D7:0. The register select signals (RS0 and RS1) determine if the PAR is used in the read or write mode. The PAR latches color palette addresses prior to the CPU writing new color definitions to the color palette. The PAR also latches color palette addresses prior to the CPU reading the color definitions in the color palette.

COLOR VALUE REGISTER (CVR)

The 18-bit wide CVR buffers the interface between the CPU and the color palette. A color definition can be read from or written to the CVR by a sequence of three byte-wide transfers. Only the least significant six bits (D5:0) of a byte contain color information when reading from or writing to the CVR; the most significant bits are set to 0. The first byte placed on the data I/O lines during a CVR read contains the red value. The next byte contains the green value and the last byte contains the blue value. After a completed read or write sequence to the CVR, the PAR is automatically incremented.

PIXEL MASK REGISTER (PMR)

The byte-wide PMR masks selected bits of the pixel address values applied to the pixel address inputs (P7:0). A 1 in any location in the PMR leaves the corresponding bit in the pixel address unchanged. A 0 resets the corresponding bit to 0. The operation of the PMR does not affect the address of the color definition during a CPU color palette access.

7.5 VIDEO MODES

The VDS includes all registers and data paths required for VGA/EGA, CGA, MDA, and HGC controllers. In addition, several clocks are provided on the board to support the various video modes and monitors. The graphics chip set selects the appropriate clock for the current mode and monitor type.

Extended graphics resolutions beyond the 720 x 400 VGA standard are possible using a multiple frequency monitor such as the NEC Multisync. Extended modes include:

- 720 x 540 graphics resolution with 4:3 aspect ratio
- 800 x 600 graphics resolution with 4:3 aspect ratio
- High resolution text modes

The VDS also supports an extended mode 13 where four pages of 64K blocks of memory can be switched and displayed instead of the standard single page. This allows for animation using 256 displayable colors without requiring a large amount of data to be manipulated (64K maximum size per image). Table 7-2 lists the standard video modes and Table 7-3 lists the extended video modes available on the VDS.

Table 7-2. VDS Standard Video Modes

Mode (Hex)	Type	Colors	Columns	Rows	Character Size	Resolution
0	Text	4	40	25	8 x 8	320 x 200
0[‡]	Text	16	40	25	8 x 14	320 x 350
0[†]	Text	16	40	25	9 x 16	360 x 400
1	Text	4	40	25	8 x 8	320 x 200
1[‡]	Text	16	40	25	8 x 14	320 x 350
1[†]	Text	16	40	25	9 x 16	360 x 400
2	Text	4	80	25	8 x 8	640 x 200
2[‡]	Text	16	80	25	8 x 14	640 x 350
2[†]	Text	16	80	25	9 x 16	720 x 400
3	Text	4	80	25	8 x 8	640 x 200
3[‡]	Text	16	80	25	8 x 14	640 x 350
3[†]	Text	16	80	25	9 x 16	720 x 400
4	Graphics	4	40	25	8 x 8	320 x 200
5	Graphics	4	40	25	8 x 8	320 x 200
6	Graphics	2	80	25	8 x 8	640 x 200
7	Text	4	80	25	9 x 14	720 x 350
7[†]	Text	4	80	25	9 x 16	720 x 400
D	Graphics	16	40	25	8 x 8	320 x 200
E	Graphics	16	80	25	8 x 8	640 x 200
F‡	Graphics	4	80	25	8 x 14	640 x 350
10[‡]	Graphics	16	80	25	8 x 14	640 x 350
11	Graphics	2	80	30	8 x 16	640 x 480
12	Graphics	16	80	30	8 x 16	640 x 480
13	Graphics	256	40	25	8 x 8	320 x 200
HGC	Graphics	2				720 x 348

[‡] EGA style text modes with 8 x 14 and 9 x 14 character sizes and 350 lines vertical resolution.

[†] VGA style text modes with 9 x 16 character size and 400 lines vertical resolution.

Table 7-3. VDS Extended Video Modes

Mode (Hex)	Type	Colors	Columns	Rows	Character Size	Resolution
40	Text	16	100	30	9 x 13	900 x 390
41	Text	16	100	50	8 x 8	800 x 400
42	Text	16	100	60	8 x 8	800 x 480
43	Text	16	100	75	8 x 8	800 x 600
50	Text	16	132	30	8 x 13	1056 x 390
51	Text	16	132	50	8 x 8	1056 x 400
52	Text	16	132	60	8 x 8	1056 x 480
53	Text	16	80	60	8 x 8	640 x 480
63	Graphics	16				720 x 540
64	Graphics	4				800 x 600

7.6 VIDEO INTERFACE

The 300SX board has two video interface connectors, J5 and J6. J5 is a 15-pin female connector intended for use with an analog monitor. J6 is a 9-pin female connector intended for use with a digital (TTL) monitor. The pinouts of the two video interface connectors are listed in Chapter 10

Floppy Disk Drive Controller

8

8.1 INTRODUCTION

This chapter describes the operation of the onboard Intel 82072 floppy disk drive controller. The architecture of the 82072 is discussed with regard to its software interface, followed by a description of its control and data transfer commands, and a definition of its input and output signals. The chapter concludes with a description of two external registers used in conjunction with the controller.

8.2 82072 ARCHITECTURE

An Intel 82072 floppy disk drive controller provides the data interface between the floppy disk drive and the central processing unit (CPU). The system interface consists of a parallel interface unit (PIU) and several registers (see Figure 8-1). The PIU has an 8-bit bi-directional data bus which handles all of the data transfers to and from the system bus.

The 82072 contains the following registers:

- Main status (MSR)
- Data rate select (DSR)
- First in first out (FIFO)

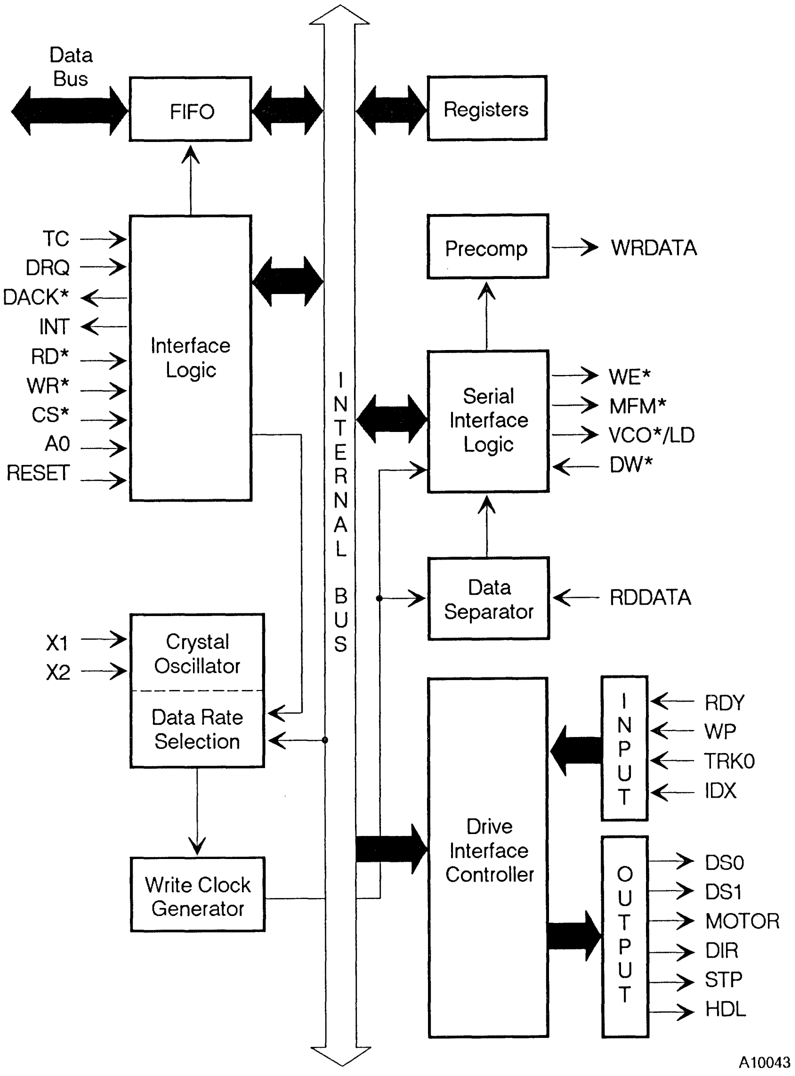


Figure 8-1. 82072 Block Diagram

The MSR, DSR, and FIFO are selected with different combinations of the RD*, WR*, and A0 pins (refer to Table 8-1).

Table 8-1. Controller Register Set Selection

A0	RD*	WR*	Function
0	0	0	Illegal
0	0	1	Read the MSR
0	1	0	Write to the DSR
0	1	1	Data bus is tri-stated
1	0	0	Illegal
1	0	1	Read the FIFO
1	1	0	Write to the FIFO
1	1	1	Data bus is tri-stated

8.2.1 Main Status Register

The MSR is an 8-bit read-only register and may be accessed at any time (before, during, and after a command). The CPU reads the MSR to determine if the controller is ready. If the controller is ready, the CPU sends a command, followed by the correct parameters, to the controller through the FIFO (data port). The FIFO should be accessed only when the request for master (RQM) bit in the MSR is set, or when the DRQ input is active during a DMA transfer.

REQUEST FOR MASTER (RQM) BIT

When set to 1, the RQM bit (bit 7) indicates the CPU can access the FIFO. If the RQM bit is set to 0, accesses to the FIFO should not be attempted.

DATA IN/OUT (DIO) BIT

The DIO bit indicates the direction of data transfer from the FIFO (the RQM bit must be set). When the DIO bit is set to 1, the CPU removes data bytes from the FIFO. When the DIO bit is set to 0, the CPU writes data bytes into the FIFO.

NON-DMA MODE (NDM) BIT

Selecting the non-DMA mode in the specify command, sets the NDM bit (bit 5) to 1, during the execution phase of a command. The NDM bit supports polled data transfers along with the RQM and DIO bits. The NDM bit remains a zero if DMA mode is selected. The NDM bit differentiates between the execution and result phases of a non-DMA Data command.

CONTROLLER BUSY (CB) BIT

When set to 1, the CB bit (bit 4) indicates the controller is processing a command. The CB bit is set during the command phase after the command byte has been accepted. When set to 0 at the end of the result phase, the CB bit indicates the start of the next command phase. If the command has no result phase (i.e., seek and recalibrate), the CB bit is set to 0 after receiving the last byte in the command phase.

DRIVE BUSY (DB) BITS

The DB bits (bit 3:0) are set only when a drive is in the seek portion of a command, including implied seeks, overlapped seeks, and recalibrates.

8.2.2 Data Rate Select Register (DSR)

The DSR (write-only) specifies the data transfer rate between the controller and the floppy disk drive. The data transfer rate can be selected for the internal and external data separator, the data transfer rate, and the write pre-compensation delays. Setting the appropriate bit in the DSR invokes the power-down mode and a software reset.

The CPU loads data into the DSR after an internal synchronization delay. The CPU should not try to perform successive writes to the DSR until the synchronization time has elapsed (24 clock periods at 24 MHz or 1 μs). The DSR should be programmed before issuing a command that accesses the disk and uses values that the DSR controls. There is no minimum delay between writing the DSR and accessing the FIFO and MSR.

The contents of the DSR, and the logic it controls, are altered if a write to the DSR occurs during data transfers. Data rate and pre-compensation values change and may give undesirable results.

Upon a hardware reset, the DDRE pin setting determines the configuration of the DSR . Tieing DDRE high configures the DSR upon power up and reset. When DDRE is tied low, the DSR retains the current data rate and pre-compensation values.

In the case of the software reset, if EPLL equals zero and DDRE equals one, the data rate select and pre-compensation field resets to 00010. Under these conditions, a second write to the DSR without a software reset is needed if a different data rate or pre-compensation value is required.

SOFTWARE RESET (SWR) BIT

Setting the SWR bit (bit 7) to 1, enables software reset of the controller. During a software reset, the controller internally holds the reset active for 12 to 15 clock cycles, depending on the state of the internal state machine. The CPU must wait for the request for master bit (bit 7 of the MSR) to be set before issuing any commands. During the initialization period following a software reset, the controller does not accept commands from the CPU.

A command may be aborted by setting the software reset bit. However, after a software reset, the specify command has to be re-issued because a software reset clears the parameters set by this command. The specify command initializes the step rate time, the head load time, and the head unload time.

POWER DOWN (PD) BIT

Setting the PD bit (bit 6) to 1, initiates the controller into its power down mode. The controller performs a software sequence before entering the power down state. During power down, the controller shuts off the clock oscillator. Only write operations to the data rate select register are allowed during power down.

ENABLE PLL (EPLL*) BIT

Setting the EPLL* bit (bit 5) to 0, enables the internal PLL data separator, and the DW/DDRE pin operates in DDRE mode. When the EPLL* bit is set to 1, the external data window signal is ignored. The EPPL* bit internally generates the data window signal used to extract data bits from the serial clock and data bit stream.

PRE-COMPENSATION (PRECOMP) BITS

The controller has write pre-compensation circuitry that internally adjusts the write data pulses before sending it to the disk drive. A programmed compensation interval (refer to Table 8-2) is added to or subtracted from the normal write pulse timing as a function of the data pattern. The configure command specifies the track number that pre-compensation starts upon. If a configure command is not issued, the controller defaults to beginning pre-compensation on track 0.

The controller defaults to PC-AT compatible pre-compensation values when the DSR register is programmed with PRECOMP bits (bits 2 - 4) set to zeroes. The pre-compensation value chosen (refer to Table 8-3) depends upon the programmed data transfer rate (DRATESEL bits 0 and 1).

Table 8-2. Programmed Pre-Compensation Delays

PRECOMP Bits 4 3 2	Pre-Compensation Delay
1 1 1	0.00 ns (Disabled)
0 0 1	41.67 ns
0 1 0	83.34 ns
0 1 1	125.00 ns
1 0 0	166.67 ns
1 0 1	208.33 ns
1 1 0	250.00 ns
0 0 0	Default

Table 8-3. Default Pre-Compensation Delays

Data Rate	Pre-Compensation Delay
1Mb/s	41.67 ns
500Kb/s	125 ns
300Kb/s	125 ns
250Kb/s	125 ns

DATA RATE SELECT (DRATESEL) BITS

The DRATESEL bits (bits 0 and 1) program both the read and write data rates (refer to Table 8-4).

Table 8-4. Controller Supported Data Transfer Rate

DRATESEL		Data Rate	
Bit 1	Bit 0	MFM	FM
1	1	1Mb/s	--
0	0	500Kb/s	250Kb/s
0	1	300Kb/s	150Kb/s
1	0	250Kb/s	125Kb/s

8.2.3 FIFO

The controller has a 16-byte FIFO with a programmable threshold set by the configure command. Threshold is the number of bytes available to the controller when service is requested from the CPU, and ranges from 1 to 16.

The parameters determining the optimal threshold value are the parallel bus frequency (Fp), serial data transfer rate (Fs), and the bus latency time (in the range between NAmax and NAmin). The ratio of Fp/Fs determines the relationship between how fast the FIFO fills and how fast it empties. The bus latency time determines how long the controller has to wait before being serviced. If Fs is the serial data rate, then Fs/8 is the serial data rate in bytes per second. The data rate of the system depends on the system clock frequency and the number of clock cycles required to transfer one byte of data (n).

In order for the parallel side to keep up with the serial side, the following must hold true: (Fp/n)/(Fs/8) > 1. This equation assumes the parallel interface section reads and writes data at a faster rate than the serial section.

It takes a number of clock cycles to acquire the system bus. NAmin is the minimum time required to acquire the system bus and NAmax is the maximum. To prevent underrun or overrun, the maximum bus acquisition time must be less than the time to fill the FIFO from the threshold limit. For a write operation, the maximum bus acquisition should be less than the time to empty the FIFO from the threshold limit.

8.2.4 Control Commands

The control commands position the head on the desired track and obtain information regarding the status of the floppy disk drives. The control commands are as follows:

- Read id
- Seek
- Recalibrate
- Sense interrupt status
- Sense drive status
- Specify
- Configure
- Relative seek
- Motor on/off
- Dumpreg

The read id command is used to find the present position of the recording heads.

The seek and recalibrate commands, executed prior to a data transfer command, position the read/write head over the desired cylinder. CPU involvement does not occur during the execution phase of the seek and recalibrate commands. At the end of the execution phase, the controller generates an interrupt. A sense interrupt status command must be issued in response to the interrupt.

The sense drive status command performs between other commands to immediately obtain the status of any one of the floppy disk drives. The sense drive status command has no execution phase and generates no interrupts.

The specify command sets the initial values for each of the three internal timers: Head Unload Time (HUT), Step Rate Time (SRT), and Head Load Time (HLT).

The configure command is issued to select special features of the 82072. A configure command need not be issued if the default values of the 82072 meet the system requirements.

The relative seek command differs from the seek command in that it steps the head by the absolute number of tracks specified in the command instead of making a comparison against an internal register.

The motor on/off (mon/moff) command provides software control of the motor pin. This command has no execution phase and generates no interrupts.

The dumpreg command supports system run-time diagnostics and application software development and debug. The dumpreg command can be issued following the specify or the configure command, to determine if proper communication between the CPU and the controller exists. The dumpreg command has no execution phase and generates no interrupts.

8.2.5 Data Transfer Commands

The controller supports six data transfer commands. They are as follows:

- Read data
- Read deleted data
- Read a track
- Write data
- Write deleted data
- Format track

The data transfer commands all require the same parameter bytes and return the same status bytes. The only difference between the individual data transfer commands are the coding of bits 4 - 0 (D3:0) in the first command byte sent to the controller (refer to Table 8-5).

Table 8-5. Data Transfer Coding

Command	D3	D2	D1	D0
Read data	0	1	1	0
Read deleted data	1	1	0	0
Write data	0	1	0	1
Write deleted data	1	0	0	1
Read track	0	0	1	0

8.2.6 Signal Definitions

TC (I)

TC (terminal count) terminates requests for data transfers. Disk read and write commands complete the transfer to the current sector with valid CRC checking/generation.

IDX (I)

IDX (index) indicates the beginning of a track. It counts retries and delay periods for internal (i.e., motor on/off) timers and is rising-edge triggered.

INT (O)

INT (interrupt) indicates command completion or a required data transfer (depending upon the data transfer mode). Command completion interrupts are cleared by reading the STO status register. Data transfer interrupts are cleared when the amount of data in the FIFO reaches the full or empty level (depending on FIFO direction) or a TC is issued.

RESET (I)

RESET places the controller in a known idle state. All disk outputs are set to a low level. All registers, except those set by the specify command, are cleared.

From the trailing edge of reset, there is a maximum delay of 8 μs until the main status register is valid. Following reset, the controller defaults to polling enabled. The default values are as follows:

- Internal data separator is enabled.
- Write pre-compensation value is 125 ns.
- Motor on delay is 0.0 seconds.
- Motor off delay is 5.2 s.
- Data rate is dependent on DDRE setting.
- FIFO is disabled.

DW/DDRE (I)

DW/DDRE (data window/default data rate enabled) samples the read data input. When the internal PLL is used, this input pin defines the data rate and write pre-compensation values after reset. DDRE tied high causes the data rate and pre-compensation bits of the DSR to reinitialize to the default values of 250Kb/s and 125 ns delay when a hardware/software reset is issued. DDRE tied low causes the current data rate and pre-compensation values in the DSR to be retained when a hardware reset is issued. During a software reset, the DSR contains those values written into the register. DDRE tied low should be used in applications where data rate and pre-compensation information needs to be retained regardless of device reset.

RDDATA (I)

RDDATA (read data) is serial FM or MFM encoded data from the disk drive.

VCO (O)

VCO (read data gate) enables an external PLL to synchronize to read data input from the disk drive.

LD (O)

LD (low density) modifies read/write head and data channel characteristics. This signal activates when internal PLL is enabled and a data transfer rate of 250 or 300Kb/s is chosen.

WE (O)

WE (write enable) enables the head to write onto the disk.

MFM (O)

MFM (MFM mode) selects between single (FM) and double density (MFM) modes. A 1 = MFM and an 0 = FM mode.

MOTOR (O)

MOTOR (motor enable) activates the drive motor on the selected drive. Delays are programmable. With one output, this pin must be qualified with the drive select logic to provide motor enables for each drive.

HDSEL (O)

HDSEL (head select) selects one of two sides on the disk. A 0 = side 0 and a 1 = side 1.

DS1:0 (O)

DS1:0 (drive select) selects one of four disk drives. DS0, DS1 = 0,0 selects drive 0.

WRDATA (O)

WRDATA (write data) writes FM or MFM encoded serial data to the disk drive. No external pre-compensation is required.

TRK0 (I)

TRK0 (track 0) indicates the head is on physical track 0 (outermost track).

WP (I)

WP (write protect) indicates if the disk is physically write protected.

RDY (I)

RDY (ready input) indicates whether the drive is ready for an operation.

HDL (O)

HDL (head load) loads the head onto the disk drive if required.

STP (O)

STP (step) supplies step pulses to the disk drive.

DIR (O)

DIR (direction), in conjunction with the STP input, causes the drive to move the head out if 0, and in if 1.

8.3 EXTERNAL REGISTERS

To support different capacity drives, the controller circuitry of the 300SX board incorporates two registers that are external to the 82072. The two registers are as follows:

- Digital Output Register (DOR)
- Digital Input Register (DIR)

These two registers are described in the following sections.

8.3.1 Digital Output Register (DOR)

The Digital Output register is a write-only register located at I/O address 3F2H. Bits 1, 5, 6, and 7 of the register are reserved and can be set to either a 1 or a 0. The remaining bits are described below.

DRIVE SELECT

When set to 0, the Drive Select bit (bit 0) indicates that drive A is selected. If the Drive Select bit is set to 1, it indicates that drive B is selected.

DISKETTE FUNCTION RESET

When the Diskette Function Reset bit (bit 2) is set to 0, the diskette reset function is disabled.

ENABLE DISKETTE DMA AND INTERRUPTS

When the Enable Diskette DMA and Interrupts bit (bit 3) is set to 1, the DMA and interrupt lines are enabled.

DRIVE A MOTOR ENABLE

When the Drive A Motor Enable bit (bit 4) is set to 1, the Motor A enable signal is activated. A timer is dedicated to perform the motor disable function. The timer is initialized by the BIOS, based on the command selected.

DRIVE B MOTOR ENABLE

When the Drive B Motor Enable bit (bit 5) is set to 1, the Motor A enable signal is activated. A timer is dedicated to perform the motor disable function. The timer is initialized by the BIOS, based on the command selected.

8.3.2 Digital Input Register (DIR)

The Digital Input register is a read-only register located at I/O address 3F7H. Bit 7 is the only bit used by the floppy disk drive controller. Bit 7 can only be driven when a high capacity disk drive is active. If bit 7 is set to a 1, it indicates to the system that the floppy disk in the disk drive has been changed. This bit is automatically reset by the disk drive when a seek operation is performed.

Intel ISA Bus Interface

9

9.1 INTRODUCTION

This chapter provides an overview of the ISA bus interface as implemented on the 300SX board.

9.2 BUS AGENTS

The ISA Bus allows several different bus agents. A bus agent is a physical unit which has an interface directly to the ISA Bus. A memory expansion board, a LAN controller, and a modem are all examples of bus agents. The two basic types of bus agents are Requesting Agents and Replying Agents.

9.2.1 Requesting Agents

Requesting Agents initiate an ISA Bus cycle. Requesting Agents can be either a Primary Requesting Agent (PRA) or a Secondary Requesting Agent (SRA).

The PRA has immediate access to the ISA Bus when control has not been granted to a Secondary Requesting Agent. The only PRA allowed is the 300SX board.

The SRA is an optional Requesting Agent that normally does not have immediate control of the ISA Bus. Control of the bus is requested from the Primary Requesting Agent. Multiple Secondary Requesting Agents are allowed. An SRA must have a 16-bit bus interface. No 8-bit SRA's are allowed on the ISA Bus.

9.2.2 Replying Agents

A Replying Agent (RPA) responds to ISA Bus cycles initiated by a Requesting Agent. An RPA cannot initiate ISA Bus cycles.

9.2.3 Configuration of Bus Agents

The ISA Bus services 5 agents via 8-bit and 16-bit portions of the bus. The number of each agent type that is supported is as follows:

Requesting Agent	1 - 5 (PRA + 4)
Primary Requesting Agent	1 (PRA)
Secondary Requesting Agent	0 - 4
Replying Agent	0 - 4

9.2.4 Agent Functional Model

Table 9-1 describes the types of ISA Bus cycles in which ISA bus agents may participate.

Table 9-1. ISA Bus Cycles

Type of Agent	Agent's Action	
Primary Requesting Agent	Initiates:	Memory access I/O access DMA access Global refresh
	Responds to:	Interrupt request DMA request Bus arbitration request
Secondary Requesting Agent	Initiates:	Memory access I/O access
	Responds to:	Global refresh
	Gains Bus Ownership by:	DMA request Assuming bus ownership on DMA grant Assuming responsibility for refresh initiation
Replying Agent	Responds to:	Memory access I/O access DMA access Global refresh
	Seeks PRA service through:	Interrupt request DMA request

9.3 GENERAL ISA BUS ATTRIBUTES

The ISA Bus has the following general attributes:

- The memory address space is 24 bits long and the data path is 16 bits wide, providing a 16M memory address space with 8 and 16-bit data transfers.

- The I/O address space is 16 bits long and the data path is 16 bits wide, providing a 64K I/O address space with 8-bit and 16-bit data transfers.

- Interrupt lines support signalling between agents on the bus and the Primary Requesting Agent.

- The DMA capability allows 8-bit or 16-bit data transfer between memory and I/O agents without direct intervention of the CPU on the Primary Requesting Agent.

- The Primary Requesting Agent refreshes all agents with refresh cycles. The Primary Requesting Agent will initiate refresh cycles at the request of a Secondary Requesting Agent that is in control of the bus in order to maintain integrity of the data in system DRAM.

- Multiple agent support is provided. The ISA Bus supports up to five agents, including one Primary Requesting Agent and up to a total of four Secondary Requesting Agents and Replying Agents. A Primary Requesting Agent (the 300SX board itself) is required in all implementations. Secondary Requesting Agents and Replying Agents are strictly optional.

9.4 SIGNAL GROUPS

The ISA Bus contains seven groups of signals: address, data, cycle control, central control, interrupt, direct memory access (DMA), and power. The bus signals can support a Primary Requesting Agent, which has an onboard DMA controller, optional Secondary Requesting Agents, and Replying Agents providing DMA or I/O memory expansion. The input and output direction designations for each signal are referenced to the Primary Requesting Agent.

9.4.1 Address Signal Group

The address signal group consists of signals driven by the Requesting Agent in order to specify both the address and data transfer width.

A(19:0) (I/O)

The A(19:0) (Address) bus signals are latched outputs driven by a Requesting Agent. They represent the least significant 20 bits of a positive logic binary number, defining a 1M address space. A(19:0) become valid when BUSALE is asserted, and they may be latched by Responding Agents on the falling edge of BUSALE.

A(19:16) are driven low during I/O cycles. During refresh cycles, the Primary Requesting Agent drives A(7:0) with the DRAM row address to be refreshed and drives A(19:8) to 0.

LA(23:17) (I/O)

The LA(23:17) (Unlatched Address) bus signals are driven by a Requesting Agent. These signals are not latched by the Primary Requesting Agent. However, they are valid when BUSALE is asserted, and they may be latched on the falling edge of this signal. LA(23:17) represent bits 17 through 23 of the memory address presented on the bus. LA(23:17) should be used by 16-bit Replying Agents in generation of SRDY*, MCS16*, and IOCS16*.

The Requesting Agent drives LA(23:17) during any transfer cycle. LA(23:17) are driven to 0 during I/O cycles. During Secondary Requesting Agent cycles, LA(23:17) must be valid throughout the entire transfer cycle, and BUSALE is asserted by the Primary Requesting Agent.

The Primary Requesting Agent drives LA(23:17) to 0 during refresh cycles.

SBHE* (I/O)

SBHE* (System Bus High Enable) is asserted by a Requesting Agent to indicate a transfer of data on lines D(15:8).

BUSALE (O)

BUSALE (Bus Address Latch Enable) is an address strobe driven by the Primary Requesting Agent. LA(23:17) are valid when BUSALE is asserted, and they may be latched on the falling edge of BUSALE. A(19:0) are latched by the Primary Requesting Agent on the leading edge of BUSALE during bus cycles initiated by the PRA.

All agents must be level sensitive with respect to BUSALE. This means that although the address signal group signals or decodes generated from them may be latched by agents on the falling edge of BUSALE, all agents must monitor the address signal group signals whenever BUSALE is asserted. This is especially important during DMA cycles, Secondary Requesting Agent cycles, and refresh cycles.

For all DMA controller cycles (including Secondary Requesting Agent cycles), the Primary Requesting Agent asserts BUSALE to allow the LA(23:17) address to "pass through" transparent address latches to the bus.

AEN (O)

AEN (DMA Address Enable) is asserted by the Primary Requesting Agent when its CPU is in the hold mode and its DMA controller has control of the bus. AEN is negated by the Primary Requesting Agent when its CPU is in control of the bus or when the DMA controller has granted the bus to a Secondary Requesting Agent. When AEN is asserted, all agents other than the Primary Requesting Agent must tri-state their Address Signal Group and Cycle Control Signal Group outputs to the ISA Bus.

During DMA cycles, the validity of LA(23:17),SA(19:0) is indicated by the assertion of both AEN and BUSALE.

Secondary Requesting Agents cannot conduct DMA cycles because only the Primary Requesting Agent can drive the DACKn* and AEN signals.

9.4.2 Data Signal Group

The data signal group consists of one set of 16 data bits. Data transfers may occur over either of the two bytes independently of one another.

D(15:0) (I/O)

On the D(15:0) (Data Bus), D15 is the most significant bit and D0 is the least significant bit. All 8-bit Replying Agents must connect only to the least significant eight data lines, D(7:0). To support communication of 8-bit Replying Agents to 16-bit Requesting Agents, both data swapping and transfer reformatting are supported by the Primary Requesting Agent. During odd-byte transfers between a 16-bit Requesting Agent and an 8-bit Replying Agent, the PRA drives the data appearing on D(7:0) onto D(15:8). Transfer reformatting is accomplished by the Primary Requesting Agent when it formats 16-bit accesses to 8-bit Replying Agents as two consecutive 8-bit ISA Bus cycles.

The Primary Requesting Agent tristates D(15:00) during refresh operations.

9.4.3 Cycle Control Signal Group

The cycle control signals control the duration and type of cycles. The group consists of six command signals, two ready signals, and three signals which specify the cycle type.

The command signals define the address space (memory or I/O) and the data direction (read or write). The ready signals modify the command pulse widths to lengthen or shorten the default cycle timings.

MRDC*, MEMR* (I/O)

MRDC* (Memory Read command) is asserted when the Requesting Agent is ready for a Replying Agent to drive the data bus with the contents of the memory location specified by LA(23:17), A(19:0). MEMR* is identical in function to MRDC*, except that it is asserted only when the memory read access falls below 1M. Eight-bit agents will receive only MEMR*.

The refresh circuitry on the Primary Requesting Agent asserts MEMR* and MRDC* during refresh cycles initiated by a Secondary Requesting Agent in control of the bus.

MEMW*, MWTC* (I/O)

MEMW* (Memory Write command) is asserted during a write cycle when the Requesting Agent is driving the data bus. MWTC* is identical in function to MEMW*, except that it is asserted only when the memory write access falls below 1M. Eight-bit agents receive only MWTC*.

IORC* (I/O)

IORC* (I/O Read command) is asserted when the Requesting Agent is ready for a Replying Agent to drive the data bus with the data available from the I/O port specified by A(15:0).

IOWC* (I/O)

IOWC* (I/O Write command) is asserted during an I/O write cycle when the Requesting Agent is driving the data bus and it is negated when a Replying Agent must clock the data into the I/O port specified by A(15:0).

MCS16* (I)

MCS16* (16-bit Memory Cycle Select) is asserted by a 16-bit memory agent to indicate to the Requesting Agent that a 16-bit cycle may be executed. Replying Agents generate MCS16* based on a decode of LA(23:17). Timing requirements placed on MCS16* prevent use of the memory command signals MEMR*/MRDC* and MEMW*/MWTC* in generation of MCS16*. The Requesting Agent ignores MCS16* on I/O cycles.

IOCS16* (I)

IOCS16* (16-bit I/O Cycle Select) is asserted by a 16-bit I/O agent to indicate to the Requesting Agent that a 16-bit cycle may be executed. Replying agents generate IOCS16* based on a decode of A(15:0). Timing requirements placed on IOCS16* prevent use of IOWC* and IORC* in generation of IOCS16*. The Requesting Agent ignores IOCS16* on memory cycles.

IOCHRDY* (I)

IOCHRDY* (I/O Channel Ready) is an asynchronous ready signal from a Replying Agent. It is negated to force the Requesting Agent to lengthen the bus cycle by inserting an integral number of wait states (one-half of an ISA Bus SYSCLK period, or 62.5 ns). IOCHRDY* must not be negated for longer than 15 μs. IOCHRDY* is ignored by the Primary Requesting Agent during zero-wait state cycles.

SRDY* (I)

SRDY* (Synchronous Ready) is asserted by the Replying Agent to terminate the current bus cycle without any further wait states. The absolute minimum command pulse width is nominally 1 SYSCLK period (125 ns) in length, and is known as a zero-wait state cycle. Secondary Requesting Agents are not required to support SRDY*.

MEMREF* (I/O)

MEMREF* (Refresh) is asserted during a DRAM refresh cycle. Only memory read cycles may occur while MEMREF* is asserted. The address present on A(7:0) is used by memory agents as the address of the row to be refreshed.

A Secondary Requesting Agent may, if it is the current bus owner, tri-state its address, command, and data drivers and assert MEMREF* to force the PRA to conduct a refresh cycle. Secondary Requesting Agents must do this every 15 μs if they retain ownership of the bus, or the contents of the system DRAM will be lost. When a refresh cycle is initiated in this manner, the PRA will assert A(7:0) and MEMR*/MRDC*.

Refresh cycles occur at a period of 15 μs. Each of the 256 possible refresh addresses must, therefore, be refreshed at least once every 4 ms.

9.4.4 Central Control Signal Group

The central control group consists of special timing, control, and error signals. The function of these signals is as follows.

SECMAST* (I)

SECMAST* (Secondary Master) is asserted by a Secondary Requesting Agent to gain control of the bus after receiving the appropriate DACKn* from the Primary Requesting Agent. When SECMAST* is asserted, all other Requesting Agents must tri-state their address, data, and control signals. After SECMAST* is asserted, the Secondary Requesting Agent must wait at least one SYSCLK period before driving the address and data group signals, and it must wait at least two SYSCLK periods before driving the cycle control group signals. If SECMAST* is asserted for longer than 15 μs, the SRA must initiate refresh cycles to maintain DRAM data integrity. Note that only DMA channels programmed in the cascade mode may be used by SRA's wishing to gain control of the bus.

IOCHCK* (I)

IOCHCK* (I/O Channel Check) may be asserted by any agent to signal an error condition that cannot be corrected, such as a memory parity error. IOCHCK* must be asserted for at least 15 ns for the Primary Requesting Agent to recognize that an error condition has occurred.

RSTDEV (O)

RSTDEV (Reset Drive) is asserted by the Primary Requesting Agent to initialize all agents on the ISA Bus after powerup or during a low-voltage condition.

SYSCLK (O)

SYSCLK (System Clock) has a frequency of 8 MHz with a 50% duty cycle, and it is driven by the Primary Requesting Agent. Bus cycle times are directly proportional to the clock period. All synchronous signals on the ISA Bus are synchronous to SYSCLK.

Bus cycles are lengthened by IOCHRDY* or shortened by SRDY* in integer multiples of one-half the SYSCLK period. For example, SRDY* could be asserted during a 16-bit cycle to reduce the command pulse width to 1.5 SYSCLK periods. Likewise, IOCHRDY* could lengthen a 16-bit cycle to N + 2.5 SYSCLK periods, where N is the number of wait states that the accessed device requests. Since the DMA controller operates off of a 4 MHz clock, DMA cycles are extended in multiples of 2 SYSCLK periods.

84OSC (O)

84OSC (Oscillator Output) is a 50-percent duty cycle clock signal with a frequency of 14.31818 MHz. 84OSC is not synchronous to either SYSCLK or any other signals on the ISA Bus; therefore, it must not be used in applications which require synchronization to the bus.

9.4.5 Interrupt Signal Group

The interrupt signal group consists of a set of signals that can be used by a Replying Agent to obtain interrupt service from a Requesting Agent.

IRQ (I)

Asserting the IRQ(15,14,12:09,07:03) (Interrupt Request) line requests an interrupt. The line must remain asserted until the interrupt is acknowledged by the appropriate software interrupt service routine.

9.4.6 Direct Memory Access Signal Group

These signals control direct memory access service and transfer of bus ownership from the Primary Requesting Agent to a Secondary Requesting Agent.

DRQ (I)

DRQ(7:5,3:0) (Direct Memory Access Request) signals are asynchronous channel requests used to gain either DMA service or control of the bus from the Primary Requesting Agent. DMA service or bus control can be attained by asserting a DRQ line and keeping it asserted until the corresponding DACK* line is asserted by the Primary Requesting Agent. When Secondary Requesting Agents wish to gain control of the bus, they must only use DMA channels that have been programmed to operate in the cascade mode.

DACK* (O)

DACK(7:5,3:0)* (DMA Request Acknowledge) lines are driven by the Primary Requesting Agent to acknowledge DMA requests DRQ(7:5,3:0). I/O repliers use DMA acknowledge signals for address selection during DMA cycles when AEN is asserted.

TC (O)

TC (Terminal Count) is asserted by the Primary Requesting Agent when any one of its DMA channels has reached its terminal count, signalling the end of the pre-programmed DMA transfer.

9.4.7 Power Signal Group

The ISA Bus provides DC power at +5V, −5V, +12V, −12V, and 0V (Ground).

+5 VOLTS

Three pins supply current for 16-bit agents. Two (2) pins supply current for 8-bit agents.

−5 VOLTS

One pin supplies current.

+12 VOLTS

One pin supplies current.

−12 VOLTS

One pin supplies current.

0 VOLTS (GROUND)

Four pins provide a return path for the currents supplied by the other power pins for 16-bit agents. Three pins provide this return path for 8-bit agents.

9.5 ISA BUS PIN ASSIGNMENTS

Tables 9-2 and 9-3 list the pin assignments for the ISA bus 8-bit and 16-bit connectors, respectively

Table 9-2. ISA Bus 8-bit Connector Pin Assignments

Pin No.	Signal	Pin No.	Signal
B1	Ground	A1	IOCHCK
B2	RSTDEV	A2	D07
B3	+ 5V	A3	D06
B4	IRQ09	A4	D05
B5	- 5V	A5	D04
B6	DRQ2	A6	D03
B7	- 12V	A7	D02
B8	SRDY	A8	D01
B9	+ 12 V	A9	D00
B10	Ground	A10	IOCHRDY*
B11	MEMW*	A11	AEN
B12	MEMR*	A12	A19
B13	IOWC*	A13	A18
B14	IORC*	A14	A17
B15	DACK3*	A15	A16
B16	DRQ3	A16	A15
B17	DACK1*	A17	A14
B18	DRQ1	A18	A13
B19	MEMREF*	A19	A12
B20	SYSCLK	A20	A11
B21	IRQ07	A21	A10
B22	IRQ06	A22	A09
B23	IRQ05	A23	A08
B24	IRQ04	A24	A07
B25	IRQ03	A25	A06
B26	DACK2*	A26	A05
B27	TC	A27	A04
B28	BUSALE	A28	A03
B29	+ 5V	A29	A02
B30	84OSC	A30	A01
B31	Ground	A31	A00

Table 9-3. ISA Bus 16-bit Connector Pin Assignments

Pin No.	Signal	Pin No.	Signal
D1	MCS16*	C1	SBHE*
D2	IOCS16*	C2	LA23
D3	IRQ10	C3	LA22
D4	IRQ11	C4	LA21
D5	IRQ12	C5	LA20
D6	IRQ15	C6	LA19
D7	IRQ14	C7	LA18
D8	DACK0*	C8	LA17
D9	DRQ0	C9	MEMR*
D10	DACK5*	C10	MEMW*
D11	DRQ5	C11	D08
D12	DACK6*	C12	D09
D13	DRQ6	C13	D10
D14	DACK7*	C14	D11
D15	DRQ7	C15	D12
D16	+5V	C16	D13
D17	SECMAST*	C17	D14
D18	Ground	C18	D15

External Interfaces 10

10.1 INTRODUCTION

This chapter describes the external interfaces found on the 300SX board.

10.2 PARALLEL PORT CONNECTORS

The parallel printer port, at connector J1, provides a one-way interface to a printer. J1 is a 25-pin D-subminiature female connector. J11, a 2 x 13 header, is provided for connecting to ribbon cables. Table 10-1 lists the pinouts for J1 and Table 10-2 lists the pinouts for J11.

Table 10-1. J1 Parallel Port Connector Pinouts

Pin	Signal	Function
1	STROBE*	
2	PRTD0	Data bit 0
3	PRTD1	Data bit 1
4	PRTD2	Data bit 2
5	PRTD3	Data bit 3
6	PRTD4	Data bit 4
7	PRTD5	Data bit 5
8	PRTD6	Data bit 6
9	PRTD7	Data bit 7
10	GACK*	Acknowledge
11	PPBUSY	
12	PE	Paper end
13	SLCT	Select
14	AUTOFDT*	Auto feed
15	PPERROR*	
16	INIT*	
17	SLCTIN*	Select input
18-25	Ground	

Table 10-2. J11 Parallel Port Connector Pinouts

Pin	Signal	Function
1	STROBE*	
2	AUTOFDT*	Auto feed
3	PRTD0	Data bit 0
4	PPERROR*	
5	PRTD1	Data bit 1
6	INIT*	
7	PRTD2	Data bit 2
8	SLCTIN*	Select
9	PRTD3	Data bit 3
10	Ground	
11	PRTD4	Data bit 4
12	Ground	
13	PRTD5	Data bit 5
14	Ground	
15	PRTD6	Data bit 6
16	Ground	
17	PRTD7	Data bit 7
18	Ground	
19	GACK*	Acknowledge
20	Ground	
21	PPBUSY	
22	Ground	
23	PE	Paper end
24	Ground	
25	SLCT	Select

10.3 SERIAL PORT CONNECTORS

Connectors J2A and J2B provide signals for serial I/O ports COM1 and COM2. COM1 signals connect to the DB9 RS-232 male connector at J2B. COM2 signals connect to the DB9 RS-232 male connector at J2A. COM1 and COM2 signals are also available at J13, a 2 x 13 header. Table 10-3 lists the pinouts for the J2A and J2B connectors and Table 10-4 lists the pinouts for J13.

Table 10-3. J2A and J2B Serial Port Connector Pinouts

Pin	Signal	Function
1	DCD	Carrier detect
2	RXD	Receive data
3	TXD	Transmit data
4	DTR	Data terminal ready
5	Ground	
6	DSR	Data set ready
7	RTS	Request to send
8	CTS	Clear to send
9	RI	Ring indicator

Table 10-4. J13 Serial Port Pinouts

Pin	Signal	Function
1	DCD1	Carrier detect (COM1)
2	DSR1	Data set ready (COM1)
3	RXD1	Receive data (COM1)
4	RTS1	Request to send (COM1)
5	TXD1	Transmit data (COM1)
6	CTS1	Clear to send (COM1)
7	DTR1	Data terminal ready (COM1)
8	RI1	Ring indicator (COM1)
9	Ground	
10	DCD2	Carrier detect (COM2)
11	DSR2	Data set ready (COM2)
12	RXD2	Receive data (COM2)
13	RTS2	Request to send (COM2)
14	TXD2	Transmit data (COM2)
15	CTS2	Clear to send (COM2)
16	DTR2	Data terminal ready (COM2)
17	RI2	Ring indicator (COM2)
18	Ground	

10.4 KEYBOARD INTERFACE CONNECTOR

The keyboard interface connects to a 5-pin DIN connector at J3B. Table 10-5 lists the pinout of the J3B connector.

Table 10-5. J3B Keyboard Connector Pinouts

Pin	Signal
1	+KBD CLK
2	+KBD DATA
3	Reserved
4	Ground
5	+5VDC (fused)
Shield	Frame ground

10.5 MOUSE INTERFACE CONNECTOR

The mouse interface connects to a 6-pin mini-DIN connector at J4. Table 10-6 lists the pinouts of the J4 connector.

Table 10-6. J4 Connector Pinouts

Pin	Signal
1	Data
2	Reserved
3	Ground
4	+5VDC (fused)
5	Clock
6	Reserved

10.6 VIDEO INTERFACE CONNECTORS

The 300SX board has two video interface connectors, J5 and J6. J5 is a 15-pin female connector intended for use with an analog (VGA) monitor. J6 is a 9-pin female connector intended for use with a digital (TTL) monitor. Table 10-7 lists the pinouts of the J5 connector. Table 10-8 lists the pinouts of the J6 connector.

NOTE

Pin one of the J5 connector is located in the upper left-hand corner of the connector. Pin 15 is at the lower right-hand corner.

Table 10-7. J5 Video Connector Pinouts

Pin	Function
1	Red video
2	Green video
3	Blue video
4	Monitor identification bit 2
5	Ground
6	Red return (ground)
7	Green return (ground)
8	Blue return (ground)
9	Key position (no pin)
10	Sync return (ground)
11	Monitor identification bit 0
12	Monitor identification bit 1
13	Horizontal sync
14	Vertical sync
15	Not used

NOTE
Monochrome video displays use green video for all video input and ignore red and blue video.

Table 10-8. J6 Video Connector Pinouts

Pin	Function
1	Ground
2	Red video
3	Intensified red video
4	Intensified green video
5	Intensified blue video
6	Green video
7	Blue video
8	Horizontal sync
9	Vertical sync

10.7 POWER SUPPLY CONNECTOR

The 12-pin connector (J12) on the board (see Figure 10-1) distributes power to the board components and the four expansion slots.

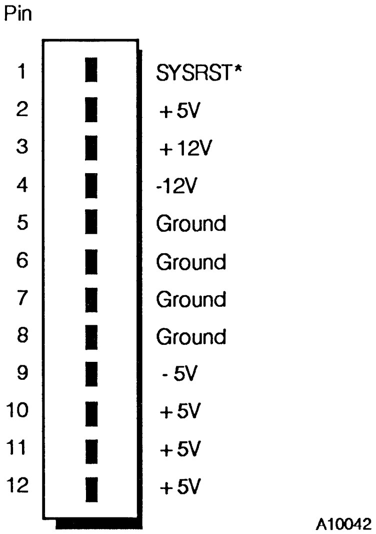


Figure 10-1. J12 Power Supply Connector

10.8 FLOPPY DISK INTERFACE CONNECTOR

Table 10-9 lists the pinouts for the J18 floppy disk interface connector.

Table 10-9. J18 Floppy Disk Interface Connector Pinouts

Pin	Signal	Function
1	GROUND	Ground
2	LD*	Head Load
3	GROUND	Ground
4	NC	No connect
5	GROUND	Ground
6	NC	No connect
7	GROUND	Ground
8	INDEX*	Beginning of track
9	GROUND	Ground
10	MOTENA*	Motor Enable A
11	GROUND	Ground
12	DRVSELB*	Drive Select B
13	GROUND	Ground
14	DRVSELA*	Drive Select A
15	GROUND	Ground
16	MOTENB*	Motor Enable B
17	GROUND	Ground
18	DIR*	Direction select
19	GROUND	Ground
20	STEP*	Motor step
21	GROUND	Ground
22	WRDATA*	Write data strobe
23	GROUND	Ground
24	FLPYWE*	Floppy disk write enable
25	GROUND	Ground
26	TRACK0*	Track 0 indicator
27	GROUND	Ground
28	WP*	Write protect
29	GROUND	Ground
30	RDDATA*	Read data strobe
31	GROUND	Ground
32	HDSEL*	Head Select
33	GROUND	Ground
34	DSKCHNG*	Drive door open

10.9 FIXED DISK INTERFACE CONNECTOR

Table 10-10 lists the pinouts for the J19 fixed disk interface connector.

Table 10-10. J19 Fixed Disk Interface Connector Pinouts

Pin	Signal	Function
1	HOST RESET*	Reset signal from CPU
2	GROUND	Ground
3	HOST DATA 7	Data bit 7
4	HOST DATA 8	Data bit 8
5	HOST DATA 6	Data bit 6
6	HOST DATA 9	Data bit 9
7	HOST DATA 5	Data bit 5
8	HOST DATA 10	Data bit 10
9	HOST DATA 4	Data bit 4
10	HOST DATA 11	Data bit 11
11	HOST DATA 3	Data bit 3
12	HOST DATA 12	Data bit 12
13	HOST DATA 2	Data bit 2
14	HOST DATA 13	Data bit 13
15	HOST DATA 1	Data bit 1
16	HOST DATA 14	Data bit 14
17	HOST DATA 0	Data bit 0
18	HOST DATA 15	Data bit 15
19	GROUND	Ground
20	KEY	No connection
21	HOST IO CH RDY*	I/O channel ready
22	GROUND	Ground
23	HOST IOW*	I/O write
24	GROUND	Ground
25	HOST IOR*	I/O read
26	GROUND	Ground
27	HOST IOCHRDY*	I/O channel ready
28	HOST ALE	Address latch enable
29	RESERVED	No Connection
30	GROUND	Ground
31	HOST IRQ14	IRQ14
32	HOST IO CS16*	Drive register enabled
33	HOST ADDR 1	Address bit 1
34	RESERVED	No connection

Table 10-10. J19 Fixed Disk Interface Connector Pinouts *(continued)*

Pin	Signal	Function
35	HOST ADDR 0	Address bit 0
36	HOST ADDR 2	Address bit 2
37	HOST CS0*	Reg. access chip select 0
38	HOST CS1*	Reg. access chip select 1
39	DISK ACTIVE*	Disk activity indicator
40	GROUND	Ground

10.10 FRONT-PANEL CONNECTORS

The system board has three connectors providing signals for front panel indicators. The following list defines the function of each connector:

- J20 connects to the front panel reset button.
- J22 provides the signal for the fixed disk drive access indicator.
- J24 supplies the signal for the power-on indicator.

10.11 SPEAKER INTERFACE

A 74F125 high current analog driver provides audible tone generation for the speaker. The 74F125 uses an OSC signal derived from a counter on the 8254 programmable interval timer (PIT) module to drive the SPKOUT signal. The OSC signal is gated on and off by bit 0 (ENABLE SPEAKER) of port 61H. Bit 1 (SPEAKER DATA) supplies the data for sounding the speaker. A 1 x 4 speaker header (J21) provides the connection between the speaker and the system (refer to Table 10-11).

Table 10-11. J21 Speaker Header Pinouts

Pin	Signal/Function
1	SPKOUT speaker drive out
2	Key (pin missing)
3	Ground
4	+5VDC

10.12 BATTERY CONNECTOR

The system board includes a two connectors (J23 and J25) for installing batteries. The batteries backup CMOS RAM to retain system configuration and password information when power is removed. These two connectors provide the identical function; use J23 if the battery is mounted near the front of the board and J25 if the battery is mounted near the back of the board.

10.13 KEYLOCK INTERFACE

The keylock interface lockouts all data from the keyboard. The keyboard is locked out anytime the controller receives the KEYLOCK* signal. A 1 x 5 keylock header (J24) provides the connection between the system and the controller (see Table 10-12). J24 also connects to the power-on indicator LED mentioned above.

Table 10-12. J24 Keylock Header Pinouts

Pin	Signal/Function
1	Power for LED power-on light
2	Key (pin missing)
3	Ground
4	KEYLOCK*
5	Ground

Power-on Self Test and Setup

11

11.1 INTRODUCTION

This chapter describes the Power-on Self Test (POST), which executes automatically each time a 300SX system is booted, and the Setup program. Both the POST and the Setup program are stored in ROM on the 300SX board. This chapter assumes that the 300SX board has been correctly integrated into a computer system environment with all necessary I/O devices connected, peripheral devices installed, and configuration performed.

11.2 POWER-ON SELF TEST

Each time the system is turned on or reset, the POST runs automatically and checks the CPU, keyboard, video display, memory, and most onboard peripheral devices.

During the POST memory test, the amount of memory being tested is displayed on the screen. Depending on the amount of extended memory installed, the POST memory test takes 3 to 15 seconds to complete.

During a soft boot, the system executes all POST tests except memory.

When POST is complete the system beeps once, if no configuration errors are detected, and displays a message similar to the following :

```
Phoenix 80386 ROM BIOS PLUS Version x.xx yy F0B
Copyright (c) 1985-1989 Phoenix Technologies Ltd.
All Rights Reserved

640K Base Memory, 03072K Extended
```

If configuration errors are detected, the system beeps twice and displays the following message:

```
Invalid configuration information -
     please run the SETUP program
     Strike the F1 key to continue
```

NOTE
It is normal for this message to appear the first time you start the system.

If you press F1 to continue, the system will operate correctly, but not at full potential.

11.3 SETUP PROGRAM

The Setup program stores system configuration settings in the battery-backed memory of the CMOS real-time clock chip. You can change system settings at any time by running the Setup program. The stored information includes the following:

- Date and time
- Number and capacity of floppy disk drives
- Number and type of fixed disk drives
- Amount of base memory
- Amount of extended memory
- Availability and type of primary video display controller
- Availability of keyboard
- CPU speed
- Availability of numeric coprocessor
- Shadow or do not shadow the system BIOS and video BIOS
- Shadow or do not shadow the offboard video BIOS
- Enable/Disable onboard winchester and floppy disks
- Enable/Disable COM1, COM2, LPT1, and LPT2
- Set the power-on password
- Enable/disable console redirection
- Enable/disable rolled memory

When the system boots, the POST checks the stored Setup information against the hardware configuration. If the data does not agree, the POST displays an invalid configuration message. If such a message appears, you can run Setup to enter the correct configuration parameters.

The Setup program performs the following operations:

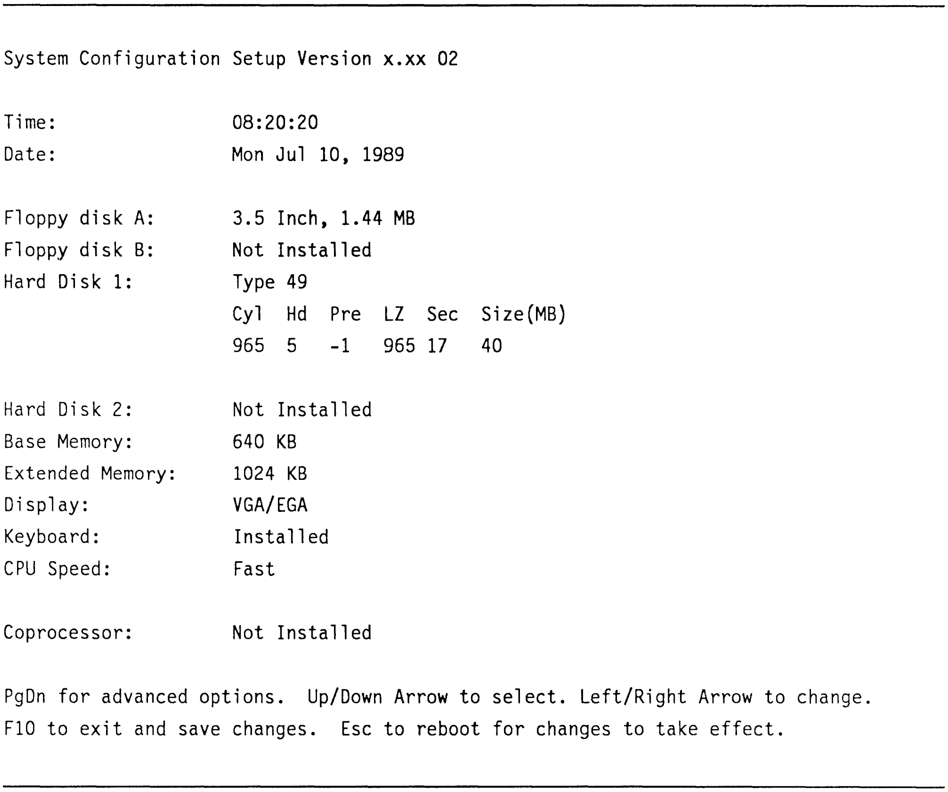
- Displays date, time, and current values for system options.
- Allows you to accept current (default) values or enter different values for time, date, and system options.
- Provides instructions on booting the system so new settings take effect.

11.4 RUNNING THE SETUP PROGRAM

The Setup program is permanently stored in ROM, and can be running with or without an operating system present.

To run Setup, wait for POST to complete (you'll hear a short beep), then immediately press Ctrl + Alt + Ins.

The Setup program starts by displaying a screen similar to that shown in Figure 11-1. The screen shows the current settings for the system. Because it reflects the exact configuration of your system, the actual display may not match this example.



```
System Configuration Setup Version x.xx 02

Time:                 08:20:20
Date:                 Mon Jul 10, 1989

Floppy disk A:        3.5 Inch, 1.44 MB
Floppy disk B:        Not Installed
Hard Disk 1:          Type 49
                      Cyl  Hd  Pre  LZ  Sec  Size(MB)
                      965  5   -1   965 17   40

Hard Disk 2:          Not Installed
Base Memory:          640 KB
Extended Memory:      1024 KB
Display:              VGA/EGA
Keyboard:             Installed
CPU Speed:            Fast

Coprocessor:          Not Installed

PgDn for advanced options.  Up/Down Arrow to select. Left/Right Arrow to change.
F10 to exit and save changes.  Esc to reboot for changes to take effect.
```

Figure 11-1. Setup Screen 1

To change options, use the ↑ or ↓ key to move the cursor to a selected option. (The cursor moves only to the options that can be changed.) Change the selected option by pressing the ← or → key. Each time you press one of these keys, the Setup program displays one of the possible values for the selected option. You cannot set the base memory, extended memory, or disk parameters using the ← or → key. Instead, you must enter the numeric values using the number keys at the top of the keyboard. If you enter an incorrect number, you can correct the entry by using the Backspace key. The fixed disk parameter table can be displayed by selecting "Hard Disk 1" or "Hard Disk 2" and pressing the F1 key.

The screen shown in Example 1 is the first of three screens displaying the current settings for the system. Press the Page Down key to view the second Setup screen.

When you finish setting options, exit Setup by pressing the F10 key or the Esc key. You can exit Setup from either screen.

Press F10 to exit the Setup program without booting the system. Changes are saved, but only the date and time information take effect. The other changes take effect when the system boots. Press the Esc key to save the Setup changes and boot the system.

11.4.1 Moving Through Setup Screen 1

Make sure the first Setup screen is displayed (refer to Figure 11-1). If it is not, press the Page Up key.

SETTING SYSTEM TIME

If the time is incorrect, use the ↑ or ↓ key to move the cursor to the time option fields on the Setup screen.

```
  Time:          08:20:20
```

The time option contains three fields: hours, minutes, and seconds. Set the hours and minutes fields using the ← or → key. Pressing the gray + key moves the clock forward. Pressing the gray − key moves the clock backward. Holding down the ← or → key causes the value to increase or decrease continuously. Set the hour field first and then press the ↓ key to move to the minutes field. Set the minutes field using the ← or → key. Reset the seconds field to 00 by selecting this field and pressing either the + or − key.

SETTING SYSTEM DATE

If the date is incorrect, use the ↑ or ↓ key to move the cursor to the date option fields on the Setup screen.

```
Date:          Tue May 23, 1989
```

The date option contains four fields: day of the week, month, day of the month, and year. You can set all fields except the day of the week. When any of the other three fields are changed, the Setup program automatically resets the day of the week accordingly.

Increase or decrease the date fields using the ← or → key. When one field is set, press the ↑ or ↓ key to move to a different field.

SETTING FLOPPY DISK DRIVE TYPES

The Setup program maintains information about two floppy disk drives (drive A and drive B). If the information about either of these floppy disk drives is incorrect, use the ↑ or ↓ keys to move the cursor to the appropriate field.

```
Floppy disk A: 3.5 Inch, 1.44 MB
Floppy disk B: Not Installed
```

Change the floppy disk options by pressing the ← or → keys. The allowable floppy disk options are as follows:

```
3.5 Inch, 1.44 MB
3.5 Inch, 720 KB
5.25 Inch, 1.2 MB
5.25 Inch, 360 KB
Not Installed
```

If only one floppy disk drive is installed, it is always drive A. Set drive B to Not Installed. You cannot change the individual fields within the options.

SETTING FIXED DISK DRIVE TYPES

The Setup program maintains drive type information for two fixed disk drives (drive 1 and drive 2).

```
Hard Disk 1:   Type 49
Hard Disk 2:   Not Installed
```

If the drive type for either of these disks is incorrect, use the ↑ or ↓ key to move the cursor to the appropriate field.

If you know the drive type associated with the fixed disk, press the ← or → key until the drive type appears on the screen.

The fixed disk parameter table can be displayed by selecting "Hard Disk 1" or "Hard Disk 2" and pressing the F1 key. Compare this list with the specifications of your disk drive to determine the correct drive type.

CAUTION
It is essential to specify the correct fixed disk drive type because the Setup program cannot independently verify this information. Specifying an incorrect drive type may damage the disk.

If the list does not include the drive type in the system, choose the user-configurable disk type. Set the number of cylinders, number of heads, and disk capacity parameters to match the specifications of the fixed disk.

If only one fixed disk drive is installed, it is always drive 1.

SETTING EXTENDED MEMORY

Extended memory is RAM above 1024K (1M). The first 1M of memory is used for base and dedicated RAM. To compute extended memory, this amount must be subtracted from the total amount of onboard RAM in the system.

If the POST indicates "Invalid configuration information," when Setup is executed, Setup will attempt to correct the error if it is due to an invalid memory configuration.

SETTING VIDEO DISPLAY CONTROLLER TYPE

The type of video display controller installed in the system must be correctly specified for POST to function correctly. If the information listed on the Setup screen is incorrect, use the ↑ or ↓ key to move the cursor to the display field.

```
Display:        VGA/EGA
```

Change the display options by pressing the ← or → key.

NOTE

Set the display to VGA/EGA even if another controller board is present and designated as the primary controller. This enables the POST to properly configure the onboard VGA controller.

The following video display controller selections are available:

`VGA/EGA` video graphics array (VGA) or enhanced graphics adapter (EGA)

`CGA40` color/graphics adapter in 40-column mode

`CGA80` color/graphics adapter in 80-column mode

`MONO` monochrome video display controller

`Not Installed` No display attached. Use this setting to suppress errors relating to the video display. This permits systems such as network servers to operate without video displays. If a display is attached, it will operate in the CGA80 mode.

SETTING KEYBOARD AVAILABILITY

The presence or absence of a keyboard must be correctly specified for the POST to function correctly. If the information listed on the Setup screen is not correct, use the ↑ or ↓ key to move the cursor to the display field.

```
  Keyboard:       Installed
```

The keyboard option can be changed by pressing the ← or → key. The following choices are available:

`Installed` Keyboard attached.

`Not Installed` Keyboard not present. Use this setting to prevent the POST from pausing when it detects the absence of the keyboard and reports an error. This permits systems such as network servers to operate without keyboards. If a keyboard is attached, it will operate correctly.

SETTING CPU SPEED

The CPU speed setting determines the speed used by the system each time you turn on the power. If the information listed in the Setup screen is not correct, use the ↑ or ↓ key to move the cursor to the display field.

```
  CPU Speed:      Fast
```

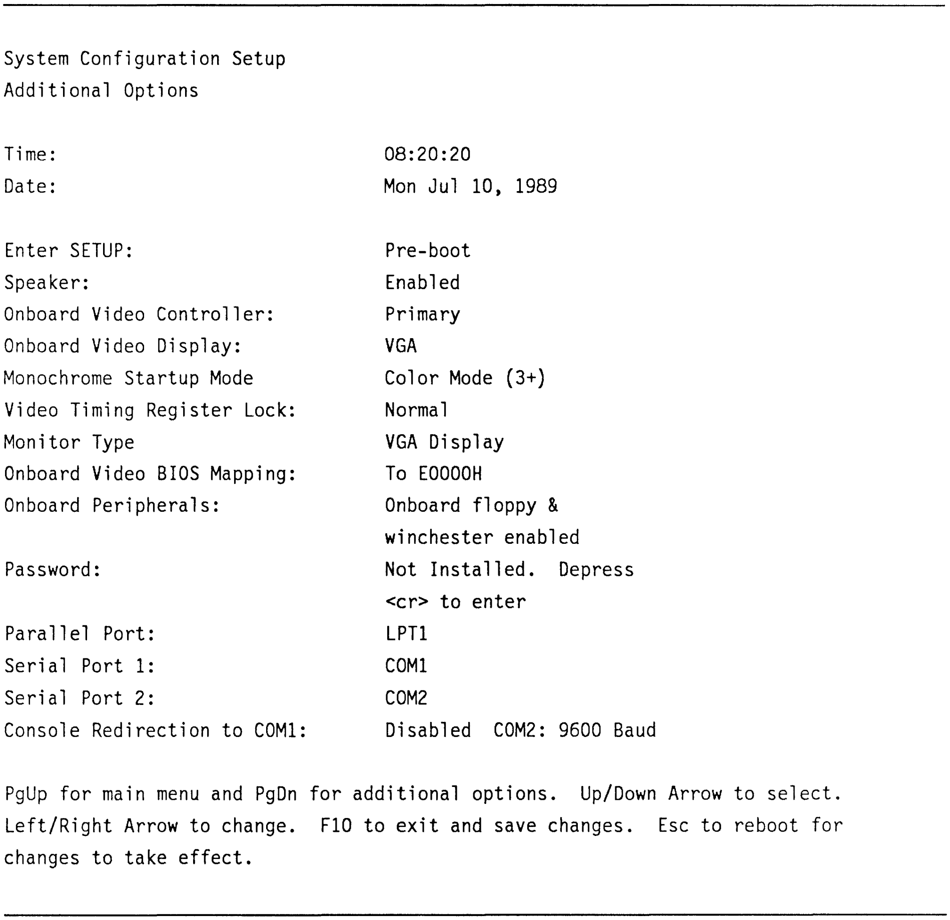
Change the CPU speed option by pressing the ← or → key. The following choices are available:

`Fast` Fast (16 MHz) is the normal setting for the CPU speed.

`Slow` Slow (8 MHz) is used to reduce CPU speed to be compatible with some applications programs. If a program does not run correctly at 16 MHz, change the CPU speed to slow.

11.4.2 Moving Through Setup Screen 2

Press the Page Down key. A screen display similar to the one shown in Figure 11-2 appears.



```
System Configuration Setup
Additional Options

Time:                              08:20:20
Date:                              Mon Jul 10, 1989

Enter SETUP:                       Pre-boot
Speaker:                           Enabled
Onboard Video Controller:          Primary
Onboard Video Display:             VGA
Monochrome Startup Mode            Color Mode (3+)
Video Timing Register Lock:        Normal
Monitor Type                       VGA Display
Onboard Video BIOS Mapping:        To E0000H
Onboard Peripherals:               Onboard floppy &
                                   winchester enabled
Password:                          Not Installed.  Depress
                                   <cr> to enter
Parallel Port:                     LPT1
Serial Port 1:                     COM1
Serial Port 2:                     COM2
Console Redirection to COM1:       Disabled  COM2: 9600 Baud

PgUp for main menu and PgDn for additional options.  Up/Down Arrow to select.
Left/Right Arrow to change.  F10 to exit and save changes.  Esc to reboot for
changes to take effect.
```

Figure 11-2. Setup Screen 2

These options are preset to the recommended choice. In most cases they need not be changed.

ENTER SETUP

Two options are available in the enter Setup selection. You can always enter Setup (from the DOS prompt) or from a pre-boot. Set either option using the ↑ or ↓ key to move the cursor to the display field.

```
Enter SETUP:   Pre-boot
```

Change the option by pressing the ← or → key.

SPEAKER

To enable or disable the speaker, use the ↑ or ↓ key to move the cursor to the display field.

```
Speaker:       Enabled
```

Change the option by pressing the ← or → key.

ONBOARD VIDEO CONTROLLER

Two options are available in the onboard video controller selection: primary and secondary. To change the onboard video controller option, use the ↑ or ↓ key to move the cursor to the display field.

```
Onboard Video Controller:   Primary
```

Change the option by pressing the ← or → key.

ONBOARD VIDEO DISPLAY

The onboard video selection is used to set up the onboard video display controller. The following onboard video display controller options are available:

`Auto`	Automatic mode. In this mode the Setup program attempts to configure automatically.
`CGA`	Color/graphics adapter mode
`MGA`	Monochrome/graphics adapter mode
`EGA`	Enhanced graphics array mode
`VGA`	Video graphics array mode

Change the option by pressing the ← or → key.

MONOCHROME MONITOR STARTUP MODE

This selection allows you to set the startup mode for a VGA monochrome monitor to monochrome or color (displayed as 64 shades of gray). To change the monochrome monitor startup mode option, use the ↑ or ↓ key to move the cursor to the display field.

```
  Monochrome Startup Mode: Color Mode (3+)
```

Change the option by pressing the ← or → key.

VIDEO TIMING REGISTER LOCK

This selection has two options: Normal and Protected. Normal allows programs to write data to the timing registers of the CRT controller while Protected inhibits programs from writing to the CRT controller registers. To change the video timing register lock option, use the ↑ or ↓ key to move the cursor to the display field.

```
Video Timing Register Lock:     Normal
```

Change the option by pressing the ← or → key.

MONITOR TYPE

This selection allows you to choose between three monitor types. You can select monochrome, color, or EGA. To change the monitor type option, use the ↑ or ↓ key to move the cursor to the display field.

```
Monitor Type:  EGA Display
```

Change the option by pressing the ← or → key.

ONBOARD VIDEO BIOS MAPPING

The Video BIOS is normally mapped to memory location E0000H, however, some application programs require the Video BIOS to be located at C0000H also. To change the onboard Video BIOS mapping option, use the ↑ or ↓ key to move the cursor to the display field.

```
Onboard Video BIOS Mapping:     To E0000H
```

Change the option by pressing the ← or → key.

ONBOARD PERIPHERALS

To enable or disable the onboard peripherals option, use the ↑ or ↓ key to move the cursor to the display field.

```
Onboard Peripherals:    On-board floppy and winchester enabled
```

Change the option by pressing the ← or → key.

PASSWORD

Set the system password by performing the following:

1. Use the ↑ or ↓ key to highlight the display field.

2. Follow on-screen procedures.

After validation, Setup stores the password in the RTC.

PARALLEL PORT

Three options are available in the parallel port selection: LPT1, LPT2, and Disabled. To change the parallel port option, use the ↑ or ↓ key to move the cursor to the display field.

```
Parallel Port: LPT1
```

Change the option by pressing the ← or → key.

SERIAL PORT 1

Three options are available in the serial port 1 selection: COM1, COM2, and Disabled. To change the serial port 1 option, use the ↑ or ↓ key to move the cursor to the display field.

```
Serial Port 1: COM1
```

Change the option by pressing the ← or → key.

SERIAL PORT 2

Three options are available in the serial port 2 selection: COM1, COM2, and Disabled. To change the serial port 2 option, use the ↑ or ↓ key to move the cursor to the display field.

```
Serial Port 2: COM2
```

Change the option by pressing the ← or → key.

NOTE

Setup allows serial port 2 to be set to the same value as serial port 1. However, the POST will automatically change the serial port 2 setting to the opposite setting when the system is rebooted.

CONSOLE REDIRECTION TO COM1:

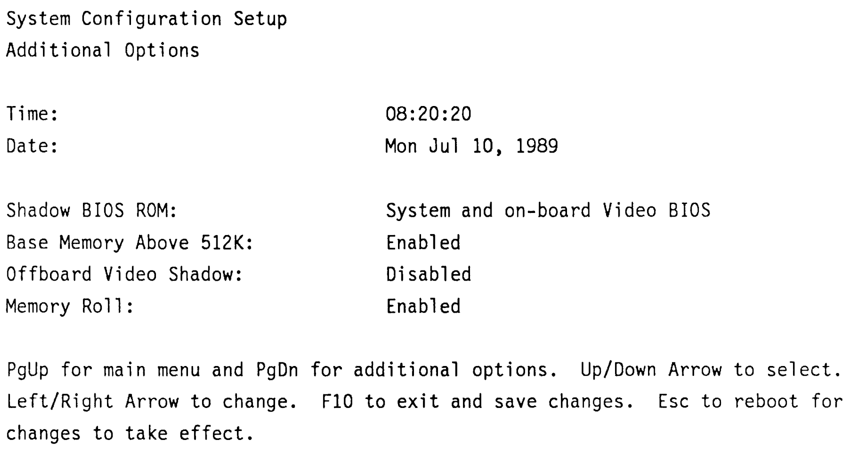
Three options are available in the console redirection to COM:1 selection. You can set COM1: at 9600 baud, 1200 baud, or disable COM1. You can also set COM2: using the same options. You can set any of the three options using the ↑ or ↓ key to move the cursor to the display field.

```
Console Redirect. to COM1:     Disabled COM2:9600 Baud
```

Change the option by pressing the ← or → key.

11.4.3 Moving Through Setup Screen 3

Press the Page Down key. A screen display similar to the one shown in Figure 11-3 appears.



```
System Configuration Setup
Additional Options

Time:                              08:20:20
Date:                              Mon Jul 10, 1989

Shadow BIOS ROM:                   System and on-board Video BIOS
Base Memory Above 512K:            Enabled
Offboard Video Shadow:             Disabled
Memory Roll:                       Enabled

PgUp for main menu and PgDn for additional options.  Up/Down Arrow to select.
Left/Right Arrow to change.  F10 to exit and save changes.  Esc to reboot for
changes to take effect.
```

Figure 11-3. Setup Screen 3

These options are preset to the recommended choice. In most cases they need not be changed.

SHADOWING SYSTEM BIOS AND VIDEO BIOS

The system board reserves an area of RAM for a copy of the system BIOS and video BIOS. This memory, called shadow memory, is write-protected and has the same addresses as the ROM locations. If you select the shadow option, the system copies the system BIOS and the video BIOS to this area and disables the ROM. System I/O performance increases significantly, because the information is maintained in fast RAM instead of ROM.

Setup Screen 3 presents the Shadow BIOS ROM option. It is recommended that you select the system and onboard video BIOS option.

```
Shadow BIOS ROM:  System and on-board video BIOS
```

If the system and onboard video BIOS option is selected, the system copies both the system BIOS and video BIOS into their corresponding RAM locations on power up. Programs then access the information from the RAM copy.

NOTE
Some high resolution plug-in video display controllers do not work properly when Video BIOS shadowing is enabled. If you have a high resolution video display controller installed and you experience display problems, set the shadow option to disabled.

BASE MEMORY ABOVE 512K

To enable or disable the memory between 512K and 640K, use the ↑ or ↓ key to move the cursor to the display field.

```
Base Memory Above 512K:  Enabled
```

Change the option by pressing the ← or → key.

OFFBOARD VIDEO SHADOW

To enable or disable the offboard video BIOS shadow option, use the ↑ or ↓ key to move the cursor to the display field.

```
Offboard Video Shadow:   Disabled
```

NOTE
Enabling this option shadows the offboard video BIOS from C0000H to DFFFFH not just from C0000H to C8000H. Therefore, use this option with caution.

Change the option by pressing the ← or → key.

MEMORY ROLL

Enabling this option allows the system to add any memory not used for the shadowing function to end of extended memory. To enable or disable the memory roll option, use the ↑ or ↓ key to move the cursor to the display field.

```
Memory Roll:   Disabled
```

Change the option by pressing the ← or → key.

11.5 EXITING THE SETUP PROGRAM

When you finish setting system options, follow the instructions on the screen to exit the Setup program.

Specifications

A

A.1 SYSTEM BOARD SPECIFICATIONS

- CPU: 386SX microprocessor
- Clock rate: 16 MHz
- ISA Bus speed: 8 MHz
- Data path: 16 bits
- Physical addressing: 16M
- Virtual addressing: 64 terabytes

A.1.1 ROM

- Size: 128K

A.1.2 Onboard RAM

- Base memory size: 2M
- Base memory expandable to: 4M

A.1.3 Expansion Slots

- Four 8- and 16-bit slots, Intel ISA standard

A.2 300SX BOARD PHYSICAL CHARACTERISTICS

A.2.1 Dimensions

Note
Figure A-1 shows the 300SX board dimensions.

- Width: 12 inches (30.48 cm)
- Length: 10.0 inches (25.4 cm)
- Height: 0.75 inch (1.8 cm)
- Weight: 3.3 pounds (1.5 kg)

A.2.2 Environment

- Operating temp: 0°C to 55° C (32°F to 131° F)
- Storage temp: -40°C to 65° C (-40° F to 149° F)
- Operating humidity: 20% to 80% RH
- Storage humidity: 5% to 95% noncondensing
- Altitude: 10,000 feet (3048 m) maximum

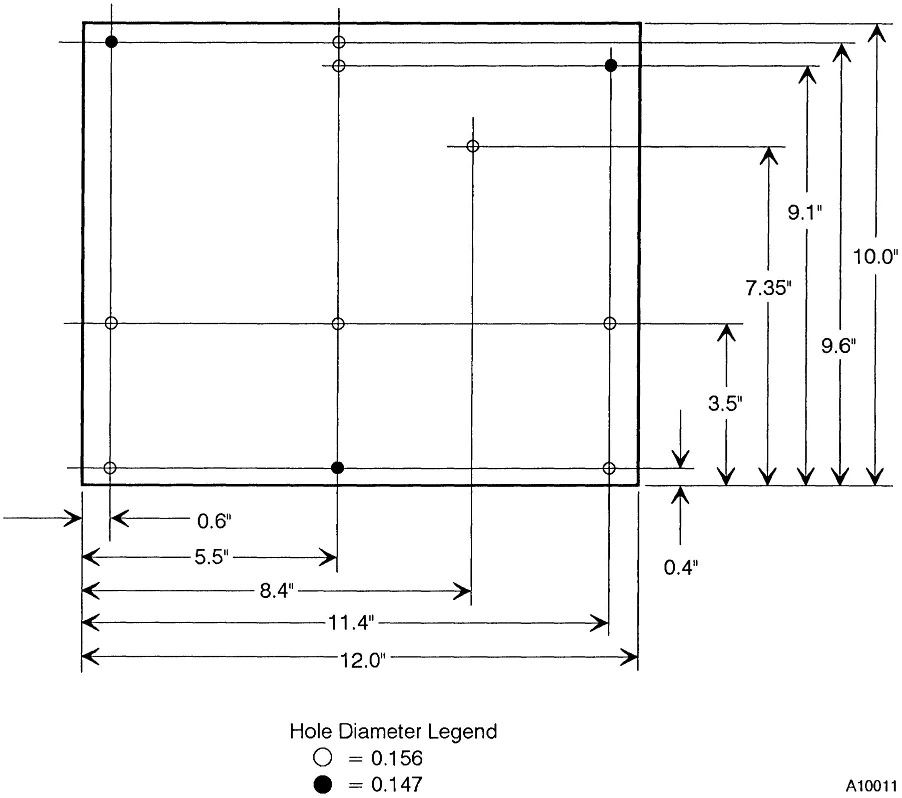


Figure A-1. **Model 300SX System Board Dimensions**

A.3 BOARD POWER REQUIREMENTS

The board requires the power consumption and current ratings listed in Table A-1 for normal operation.

Table A-1. Power Consumption

Voltage	Nominal Current
+5V ±5%	4A
+12V ±10%	0.1A
-12V ±10%	0.1A

The 300SX board contains four 8- or 16-bit expansion slots.

The maximum available +5V current allowable to any expansion slot depends upon the following parameters:

- The number of +5V pins in the slot providing current
- The power supply capacity
- The current-carrying capability of the power connectors
- The power demands of both the board and all other slots in use
- The board circuit traces

Each expansion slot +5V pin can carry a maximum of 3A. Table A-2 lists the current ratings for each slot type.

Table A-2. Maximum Current Ratings

Slot Type	Maximum Allowable Current Per Slot
8-bit	6A (3A per pin with 2 pins)
16-bit	9A (3A per pin with 3 pins)

CAUTION

The ISA bus connectors are limited to 3A per +5V pin maximum. The power supply connectors are limited to 5A per pin maximum. These limitations plus the maximum capacity of the power supply itself may be more restrictive than the current limitations of the pins/connectors and expansion slots. As an overall limitation, do not exceed 12A total when computing total +5V current drain for the system board. This avoids damage to the power supply and board.

Table A-2 specifies the nominal current drain for various system board configurations.

Table A-3. Maximum +5V Current Drain on 300SX Board

Element	Maximum +5V Current Required
System board (2M onboard RAM, no numeric coprocessor, no expansion boards)	4.0A
387SX numeric coprocessor	0.4A

System BIOS Specifications

B

B.1 INTRODUCTION

The BIOS is a software interface that isolates operating systems and application programs from specific hardware devices. BIOS routines allow assembly language programmers to do block and character-level operations without concern for device addresses or hardware operating characteristics. The BIOS also provides system services such as time-of-day and memory size information.

This appendix provides a description of the BIOS routines.

B.2 SYSTEM ID STRING

The system BIOS reserves seven bytes starting at location F000:ED00 for an ASCII System Identification string.

Table B-1. System ID String

Byte	ASCII Value	Function
F000:ED00	I	Constant
F000:ED01	D	Constant
F000:ED02	N	Constant
F000:ED03	O	Constant
F000:ED04	D	System type
F000:ED05	1	Board version

B.3 NON-MASKABLE INTERRUPT (INT 02H)

The system board and I/O channel parity error signals connect to the NMI pin of the CPU. If a memory parity error occurs, the hardware invokes this routine. The routine examines a register and ensures the reason for the interrupt is a memory parity error. If no memory parity error occurred, the routine returns to the interrupted operation. Some video controllers may make use of the NMI. The parity checking aspect is never impaired. Note that the I/O channel error and parity error sources of an NMI can be masked.

Input: None

Output: Error message; system halts

B.4 PRINT SCREEN (INT 05H)

Invoking this routine prints the screen. The cursor position at the time this routine invokes is saved and restored upon completion. During execution of this routine, the PRINT SCREEN key is ignored.

Input: None

Output: Location 50:0H

= 0	Normal termination or print screen not busy
= 1	Print screen in progress
= FFH	Error during print screen

B.5 SYSTEM TIMER H/W INTERRUPT (INT 08H)

This interrupt occurs when channel zero of the system timer counts down to zero. The BIOS handler keeps a count of interrupts starting at power-on for use as a system clock. It also decrements the disk drive motor control counter and shuts off the drive when the counter expires. In addition, the BIOS handler invokes INT 1CH allowing a user routine to execute.

Input: None

Output: None

B.6 KEYBOARD H/W INTERRUPT (INT 09H)

This interrupt occurs each time a key is pressed. The BIOS handler checks for certain special keys or combinations, such as CTRL, ALT, DEL or PAUSE and takes action if one occurs. Otherwise the key scan code and its ASCII value, if applicable, are moved into the keyboard buffer (provided enough space is available).

Input: None

Output: None

B.7 DISK H/W INTERRUPT (INT 0EH)

This interrupt occurs when the disk drive requires attention. During multi-sector transfers, the controller interrupts for each transfer. A completed operation also generates an interrupt. The BIOS handler sets bit 7 of the byte at 40:30H indicating an interrupt occurred.

Input: None

Output: None

B.8 VIDEO I/O (INT 10H)

This program interface allows control of the video display. Detailed descriptions are provided with each function.

Functions:	AH = 00H	Set display mode
	AH = 01H	Set cursor size
	AH = 02H	Set cursor position
	AH = 03H	Read cursor position
	AH = 04H	Read light pen position
	AH = 05H	Select active display page (valid only for text modes)
	AH = 06H	Scroll window up
	AH = 07H	Scroll window down
	AH = 08H	Read attribute/character at current cursor position
	AH = 09H	Write attribute/character at current cursor position
	AH = 0AH	Write character only at current cursor position
	AH = 0BH	Set color palette
	AH = 0CH	Write dot
	AH = 0DH	Read dot
	AH = 0EH	Write character as teletype to active page
	AH = 0FH	Get current video mode
	AH = 13H	Write string

B.8.1 Set Display Mode

This function sets the display mode for the active controller. A monochrome display controller only uses modes 0 and 7.

Input: AH = 00H
AL Mode value

Text Mode:

AL = 00H	40 x 25 B&W
AL = 01H	40 x 25 color
AL = 02H	80 x 25 B&W
AL = 03H	80 x 25 color
AL = 07H	80 x 25 B&W Card

Graphics Mode:

AL = 04H	320 x 200 color
AL = 05H	320 x 200 B&W
AL = 06H	640 x 200 B&W
AL = 11H	640 x 480 color

Enhanced Graphics Mode:

AL = 0DH	320 x 200 16 colors
AL = 0EH	640 x 200 16 colors
AL = 0FH	640 x 350 B&W
AL = 10H	640 x 350 16/64 colors

Output: None

B.8.2 Set Cursor Size

This function programs the CRTC cursor start/end register to set the desired cursor size.

Input:	AH = 01H	Set cursor type
	CH (bits 4:0)	Start line for cursor
	(bit 5)	Control cursor display
		0 = Normal
		1 = No cursor
	(bit 6)	Set to 0
	CL (bits 4:0)	End line for cursor

Output: None

B.8.3 Set Cursor Position

This function programs the CRTC cursor position register to display the cursor at the desired location. When the specified page number differs from the active display page, no visible reaction occurs.

Input:	AH = 02H	
	DH,DL	Row, column (0,0 upper left corner)
	BH	Page number (must be 0 for graphics modes)

Output: None

B.8.4 Read Cursor Position

This function returns the current cursor position of the specified page.

Input: AH = 03H
BH Page number (must be 0 for graphics modes)

Output: DH,DL Row, column of current cursor position
CH,CL Cursor size parameters

B.8.5 Read Light Pen Position

This function returns the current light pen position.

Input: AH = 04H

Output: AH = 00H Light pen switch not down/not triggered
AH = 01H Valid light pen value in registers
DH,DL Row,column of light pen character position
CH Raster line (0:199)
BX Pixel column (0:319, 0:639)

B.8.6 Select Active Display Page (Valid Only for Text Modes)

This function selects the active display page.

Input: AH = 05H
AL New page value (0:7 for modes 0 & 1, 0:3 for modes 2 & 3)

Output: None

B.8.7 Scroll Window Up

This function scrolls up the given area in the active page.

Input: AH = 06H
AL Number of lines blanked at bottom of window
0 = Blank entire window
CH,CL Row, column of upper left corner of window
DH,DL Row, column of lower right corner of window
BH Attribute used on blank line

Output: None

B.8.8 Scroll Window Down

This function scrolls down the given area in the active page.

Input: AH = 07H
AL Number of input lines blanked at top of window
0 = Blank entire window
CH,CL Row, column of upper left corner of window
DH,DL Row, column of lower right corner of window
BH Attribute used on blank line

Output: None

B.8.9 Read Attribute/Character Current Cursor Position

This function gets the attribute and character displayed at the cursor position.

Input: AH = 08H
BH Display page (used for text modes only)

Output: AL Character read
AH Attribute of character read (text modes only)

B.8.10 Write Attribute/Character at Current Cursor Position

This function writes the specified attribute and character to the display at the current cursor position. Cursor position does not change.

Input: AH = 09H
AL Character to write
BH Display page (used for text modes only)
BL Attribute of character (text) or color of character (graphics)
CX Number of times to write the character and attribute
In graphics mode the write does not continue to the next line

Output: None

B.8.11 Write Character Only at Current Cursor Position

This function writes the character on the display at the current cursor position. Cursor position is unchanged.

Input:	AH = 0AH	
	AL	Character written
	BH	Display page (used for text modes only)
	CX	Number of times to write the character (in graphics mode the write does not continue to the next line)

Output: None

B.8.12 Set Color Palette

This function programs the CRTC border control register to set the desired color.

Input:	AH = 0BH	
	BH	Color ID set (0-1)
	BL	Color value used with color ID
	BH = 00H	Set background color for 320x200 graphics modes Set border color for alphanumeric modes set foreground color for 640x200 graphics
	BL = 0-31	
	BH = 01H	Select palette for 320x200 graphics
	BL = 0	Green (1)/red (2)/brown (3)
	BL = 1	Cyan (1)/magenta (2)/white (3)

Output: None

B.8.13 Write Dot

This function writes a dot at the specified location.

Input:	AH = 0CH	
	AL	Color value The color value is exclusive ORed with the current color at that location if bit 7 = 1
	CX	Column number
	DX	Row number
Output:	None	

B.8.14 Read Dot

This function reads the dot at the specified location.

Input:	AH = 0DH	
	CX	Column number
	DX	Row number
Output:	AL	Color data

B.8.15 Write Character as Teletype to Active Page

This function writes a character at the cursor position of the active page and moves the cursor to the next position. The attribute remains the same in text mode. The cursor moves to the next line and scrolls the screen, if applicable.

Input: AH = 0EH
AL Character to write
BL Foreground color in graphics mode

Output: None

B.8.16 Get Current Video Mode

This function returns the current video mode.

Input: AH = 0FH

Output: AH Number of columns on screen
AL Mode currently set
BH Current active display page

B.8.17 Write String

This function displays a string of characters on the display.

Input:	AH = 13H	
	ES:BP	Pointer to string to write
	CX	Length of character string to write
	DH,DL	Row, column string write
	BH	Page number
	AL = 00H	BL = Attribute, string {char, ..., char} Cursor does not move
	AL = 01H	BL = Attribute, string {char, ..., char} Cursor moves
	AL = 02H	String {char, attr, ..., char, attr} Cursor does not move
	AL = 03H	String {char, attr, ..., char, attr} Cursor moves

Output: None

B.9 EQUIPMENT DETERMINATION (INT 11H)

This program interface describes the hardware installed on the system.

Input: None

Output: AX

Bits 15:14	Number of printers attached
Bit 13	Internal modem
Bit 12	Not used
Bits 11:9	Number of RS-232 ports attached
Bit 8	Not used
Bits 7:6	Number of floppy disk drives (00 = 1; 01 = 2 only if bit 0 = 1)
Bits 5:4	Initial video mode 00 = Not used 01 = 40 x 25 color 10 = 80 x 25 color 11 = 80 x 25 monochrome
Bit 3	Not used
Bit 2	Pointing device 1=pointing device installed 2=pointing device not installed
Bit 1	Numeric coprocessor 1 = Numeric coprocessor installed 0 = Numeric coprocessor not installed
Bit 0	Floppy disk drive 1 = Floppy disk drive installed 0 = Floppy disk drive not installed

B.10 MEMORY SIZE DETERMINATION (INT 12H)

This routine returns the memory size below address 10000H.

Input:	None	
Output:	AX	Number of contiguous 1K blocks of memory

B.11 FLOPPY DISK DRIVE I/O (INT 13H, PART 1)

This interface provides access to the floppy disk drives supported by the system.

Functions:	AH = 00H	Reset disk drive
	AH = 01H	Read status
	AH = 02H	Disk read
	AH = 03H	Disk write
	AH = 04H	Disk verify
	AH = 05H	Format disk track
	AH = 08H	Disk drive parameters
	AH = 15H	Read direct access storage device (DASD) type
	AH = 16H	Disk change line status
	AH = 17H	Set DASD type for format
	AH = 18H	Set media type for format
Input:	AH	Function number
	AL	Number of sectors
	CH	Track number
	CL	Sector number
	DH	Head number
	DL	Drive number
	ES:BX	Transfer address

Output:	Carry flag = 1	If operation error
	Carry flag = 0	If operation correct

AH   Status of operation

AH = 00H	No error
AH = 01H	Invalid function request
AH = 02H	Address mark not found
AH = 03H	Write protect error
AH = 04H	Requested sector not found
AH = 06H	Media changed
AH = 08H	DMA overrun
AH = 09H	Attempt to DMA across 64K boundary
AH = 0CH	Media type not found
AH = 10H	CRC error on disk read
AH = 20H	General controller failure
AH = 40H	Seek operation failed
AH = 80H	Timeout

B.11.1 Reset Disk Drive

Input: AH = 00H

Output: DL   Drive number (0-based)
Bit 7=0 floppy disk

B.11.2 Read Status

Input: AH = 01H

Output: DL   Drive number (0-based)
Bit 7=0 floppy disk (value checked)

B.11.3 Disk Read

Input:	AH = 02H	
	AL	Number of sectors
	CH	Track number
	CL	Sector number
	DH	Head number
	DL	Drive number
	ES:BX	Address of buffer
Output:	AH	Status of operation
	AL	Number of sectors transferred
	CY = 1	Error
	CY = 0	No error

B.11.4 Disk Write

Input:	AH = 03H	
	AL	Number of sectors
	CH	Track number
	CL	Sector number
	DH	Head number
	DL	Drive number
	ES:BX	Address of buffer
Output:	AH	Status of operation
	AL	Number of sectors transferred
	CY = 1	Error
	CY = 0	No error

B.11.5 Disk Verify

Input:	AH = 04H	
	AL	Number of sectors
	CH	Track number
	CL	Sector number
	DH	Head number
	DL	Drive number
Output:	AH	Status of operation
	AL	Number of sectors verified
	CY = 1	Error
	CY = 0	No error

B.11.6 Format Disk Track

Input:	AH = 05H	
	AL	Number of sectors
	CH	Track number
	DH	Head number
	DL	Drive number
	ES:BX	Address of buffer containing a series of 4 byte fields for each sector:
		Byte 1 = Track
		Byte 2 = Head
		Byte 3 = Sector
		Byte 4 = Bytes/sector
		0 = 128 bytes/sector
		1 = 256 bytes/sector
		2 = 512 bytes/sector
		3 = 1024 bytes/sector
Output:	AH =	Status of operation
	CY = 1	Error
	CY = 0	No error

B.11.7 Disk Drive Parameters

Input:	AH = 08H	
	DL	Drive number
Output:	ES:DI	Pointer to drive parameter table
	CH	Maximum number of tracks (low order 8 bits)
	CL (bits 7:6)	Maximum number of tracks (high order 2 bits)
	(bits 5:0)	Maximum sectors per track
	DH	Maximum head number
	DL	Number of floppy disk drives installed
	BH	0
	BL (bits 7:4)	0
	(bits 3:0)	Valid drive type value stored in CMOS RAM 01 - 360K drive 02 - 1.2M drive 03 - 720K drive 04 - 1.4M drive
	AX	0

B.11.8 Read DASD Type

Input:	AH = 15H	
	DL	Drive number 0-3=floppy
Output:	AH = 00H	Drive not present
	AH = 01H	Disk, no change line available
	AH = 02H	Disk, change line available
	AH = 03H	Fixed disk drive
	CY = 1	Error
	CY = 0	No error

B.11.9 Disk Change Line Status

Input:	AH = 16H	
	DL	Drive number
Output:	AH = 01H	Invalid parameter
	AH = 80H	Drive not ready
	CY = 1H	AH=non 0
AH=0		

B.11.10 Set DASD Type For Format

Input:	AH = 17H	
	AL = 00H	Not used
	AL = 01H	360K floppy disk in 360K drive
	AL = 02H	360K floppy disk in 1.2M drive
	AL = 03H	1.2M floppy disk in 1.2M drive
	AL = 04H	720K floppy disk in 720K drive
	AL = 05H	720K floppy disk in 1.4M drive
	DL	Drive number 0-3
Output:	AH	Status of operation
	CY = 1	Error
	CY = 0	No error

B.11.11 Set Media Type For Format

Input:	AH = 18H	
	CH	Maximum number of tracks (low order 8 bits)
	CL (bits 7:6)	Maximum number of tracks (high order 2 bits)
	(bits 5:0)	Maximum sectors per track
	DL	Drive number
Output:	ES:DI	Pointer to drive parameter table for this media type unchanged if AH is non-zero
	CY = 0	
	AH = 00H	Successful Track and sector combination supported
	CY = 1	
	AH = 01H	Function not available
	AH = 0CH	Track and sector combination not supported

B.12 FIXED DISK I/O (INT 13H, PART 2)

This interface provides access to fixed disk drives through the fixed disk controller.

Functions:	AH = 00H	Disk reset
	AH = 01H	Read status
	AH = 02H	Read disk
	AH = 03H	Write disk
	AH = 04H	Verify disk sectors
	AH = 05H	Format disk track
	AH = 08H	Disk drive parameters
	AH = 09H	Initialize disk parameters
	AH = 0AH	Disk read long
	AH = 0BH	Disk write long
	AH = 0CH	Disk seek
	AH = 0DH	Disk alternate reset
	AH = 10H	Disk ready test
	AH = 11H	Disk recalibrate
	AH = 14H	Disk diagnostics
	AH = 15H	Read DASD type

Input:	AH	Function number
	AL	Number of sectors
	CH	Cylinder number (0:1023)
	CL	Sector number (1:17) High 2 bits of cylinder number are placed in the high 2 bits of the CL register
	DH	Head number (0:15 are allowed)
	DL	Drive number (80H:81H)
	ES:BX	Transfer address

Output: See individual functions

B.12.1 Disk Reset

Input:	AH = 00H	
	DL	Drive number
Output:	AH	Status
	CY = 1	Error
	CY = 0	No error

B.12.2 Read Status

Input:	AH = 01H	
	DL	Drive number bit7=1
Output:	AH	Status of the system
	CY = 1	Error
	CY = 0	No error

B.12.3 Read Disk

Input:	AH = 02H	
	AL	Number of sectors
	CH	Cylinder number (low order 8 bits)
	CL (bits 7:6)	Cylinder number (high order 2 bits)
	(bits 5:0)	Sector number
	DL	Drive number bit7=1
	DH	Head number
	ES:BX	Address of buffer
Output:	AH	Status of operation
	CY = 1	Error
	CY = 0	No error

B.12.4 Write Disk

Input:	AH = 03H	
	AL	Number of sectors
	CH	Cylinder number (low order 8 bits)
	CL (bits 7:6)	Cylinder number (high order 2 bits)
	(bits 5:0)	Sector number
	DH	Head number
	DL	Drive number
	ES:BX	Address of buffer
Output:	AH	Status of operation
	CY = 1	Error
	CY = 0	No error

B.12.5 Verify Disk Sectors

Input:	AH = 04H	
	AL	Number of sectors
	CH	Cylinder number (low order 8 bits)
	CL (bits 7:6)	Cylinder number (high order 2 bits)
	(bits 5:0)	Sector number
	DL	Drive number
	DH	Head number
Output:	CY = 1	Error
	CY = 0	No error

B.12.6 Format Disk Track

Input:	AH = 05H	
	AL	Number of sectors
	CH	Cylinder number (low order 8 bits)
	CL (bits 7:6)	Cylinder number (high order 2 bits)
	(bits 5:0)	Sector number
	DL	Drive number
	DH	Head number
	ES:BX	Address of buffer for reads and writes
		This points to a 512 byte buffer
		The first 2 x (number of sectors/track) bytes
		contain F, N for each sector
		where: F = 00H for a good sector,
		80H for a bad sector, and
		N = logical sector number
Output:	AH	Status of operation
	CY = 1	Error
	CY = 0	No error

B.12.7 Disk Drive Parameters

Input:	AH = 08H	
	DL	Drive number
Output:	AX = 0	
	BH = 0	
	DL	Number of disk drives installed
	DH	Maximum usable head number
	CH	Cylinder number (low order 8 bits)
	CL (bits 7:6)	Cylinder number (high order 2 bits)
	(bits 5:0)	Sector number
	CY = 1	Error
	CY = 0	No error

B.12.8 Initialize Disk Parameters

Input:	AH = 09H	
	DL	Drive number
Output:	AH	Status
	CY = 1	Error
	CY = 0	No error

B.12.9 Disk Read Long

Input:	AH = 0AH	
	AL	Number of sectors
	CH	Cylinder number (low order 8 bits)
	CL (bits 7:6)	Cylinder number (high order 2 bits)
	(bits 5:0)	Sector number
	DH	Head number
	DL	Drive number
	ES:BX	Buffer address (must accommodate four bytes of ECC per sector)
Output:	AH	Status of operation
	CY = 1	Error
	CY = 0	No error

B.12.10 Disk Write Long

Input:	AH = 0BH	
	AL	Number of sectors
	CH	Cylinder number (low order 8 bits)
	CL (bits 7:6)	Cylinder number (high order 2 bits)
	(bits 5:0)	Sector number
	DH	Head number
	DL	Drive number
	ES:BX	Buffer address Each sector of data is followed by the 4-byte ECC
Output:	AH	Status of operation
	CY = 1	Error
	CY = 0	No error

B.12.11 Disk Seek

Input:	AH = 0CH	
	CH	Cylinder number (low order 8 bits)
	CL (bits 7:6)	Cylinder number (high order 2 bits)
	(bits 5:0)	Sector number
	DH	Head number
	DL	Drive number
Output:	AH	Status of operation
	CY = 1	Error
	CY = 0	No error

B.12.12 Disk Alternate Reset (Same as Function 00H)

Input: AH = 0DH
DL Drive number

Output: AH Status
CY = 1 Error
CY = 0 No error

B.12.13 Disk Ready Test

Input: AH = 10H
DL Drive number

Output: AH Status (if fixed disk drive present)
CY = 1 Error
CY = 0 No error

B.12.14 Disk Recalibrate

Input: AH = 11H
DL Drive number

Output: AH Status (if fixed disk drive present)
CY = 1 Error
CY = 0 No error

B.12.15 Disk Diagnostics

Input:	AH = 14H	
	DL	Drive number
Output:	AH	Status
	CY = 1	Error
	CY = 0	No error

B.12.16 Read DASD Type

Input:	AH = 15H	
	DL	Drive number
Output:	AH = 00H	Drive not present
	AH = 01H	Disk, no change line available
	AH = 02H	Disk, change line available
	AH = 03H	Fixed disk drive
	CX:DX	Contains total number of sectors
	CY = 1	Invalid drive number
	CY = 0	No error

B.13 RS232 I/O (INT 14H)

This program interface provides access to the asynchronous communication I/O port. The communication port number is not checked.

Functions:	AH = 00H	Initialize the communication port
	AH = 01H	Send a character
	AH = 02H	Receive a character
	AH = 03H	Return the communication port status

Input:	AH	Function number
	DX	Communication port number (0:3)

Output: See individual functions

B.13.1 Initialize the Communication Port

Input:	AH = 00H	
	DX	Communication port number (0:3)
	AL	Parameters for initialization
	Bits 7:5	Baud rate
		000 = 110 bps
		001 = 150 bps
		010 = 300 bps
		011 = 600 bps
		100 = 1200 bps
		101 = 2400 bps
		110 = 4800 bps
		111 = 9600 bps
	Bits 4:3	Parity
		00 = No parity
		01 = Odd parity
		10 = No parity
		11 = Even parity
	Bit 2	Stop bit
		0 = 1 stop bit
		1 = 2 stop bits
	Bits 1:0	Word length
		10 = 7 bits
		11 = 8 bits

Output:	AH	Line control status:
	Bit 7	Time out
	Bit 6	Transmitter shift register empty
	Bit 5	Transmitter holding register empty
	Bit 4	Break detected
	Bit 3	Framing error
	Bit 2	Parity error
	Bit 1	Overrun error
	Bit 0	Data ready
	AL	Modem status:
	Bit 7	Received line signal detect
	Bit 6	Ring indicator
	Bit 5	Data set ready
	Bit 4	Clear to send
	Bit 3	Delta receive line signal detect
	Bit 2	Trailing edge ring detector
	Bit 1	Delta data set ready
	Bit 0	Delta clear to send

B.13.2 Send a Character

Input:	AH = 01H	
	AL	Character
	DX	Communication port number (0:3)

Output:	AL	Character
	AH	Status of operation
	Bit 7 = 0	See below
	Bit 7 = 1	Time out error

If bit 7 of AH is 0, then:

Bit 6	Transmitter shift register empty
Bit 5	Transmitter holding register empty
Bit 4	Break detected
Bit 3	Framing error
Bit 2	Parity error
Bit 1	Overrun error
Bit 0	Data ready

B.13.3 Receive a Character

Input:	AH = 02H	
	DX	Communication port number (0:3)

Output:	AH = 00H	No error
	AL	Character received
	AH	Status of operation
	Bit 7 = 0	Character was received
	Bit 7 = 1	Error encountered

If bit 7 of AH is 0, then:

Bit 6	Transmitter shift register empty
Bit 5	Transmitter holding register empty
Bit 4	Break detected
Bit 3	Framing error
Bit 2	Parity error
Bit 1	Overrun error
Bit 0	Data ready

B.13.4 Return the Communication Port Status

Input:	AH = 03H	
	DX	Communication port number (0:3)
Output:	AH	Communication line status:
	Bit 7	Time out
	Bit 6	Transmitter shift register empty
	Bit 5	Transmitter holding register empty
	Bit 4	Break detected
	Bit 3	Framing error
	Bit 2	Parity error
	Bit 1	Overrun error
	Bit 0	Data ready
	AL	Modem status:
	Bit 7	Received line signal detect
	Bit 6	Ring indicator
	Bit 5	Data set ready
	Bit 4	Clear to send
	Bit 3	Delta receive line signal detect
	Bit 2	Trailing edge ring detector
	Bit 1	Delta data set ready
	Bit 0	Delta clear to send

B.14 SYSTEM SERVICE ROUTINES (INT 15H)

This program interface allows access to the 386 extended functions.

Functions:	AH = 44H	Onboard video setup
	AH = 4FH	Keyboard intercept (null)
	AH = 80H	Device open (null)
	AH = 81H	Device close (null)
	AH = 82H	Program termination (null)
	AH = 83H	Event wait (null)
	AH = 84H	Joystick support
	AH = 85H	System request key pressed (null)
	AH = 86H	Wait
	AH = 87H	Move block
	AH = 88H	Extended memory size determination
	AH = 89H	Switch coprocessor to virtual mode
	AH = 90H	Device busy (null)
	AH = 91H	Interrupt complete (null)
	AH = C0H	Return system configuration parameters
	AH = C1H	Get mouse data area
Input:	AH	Function number

Output: See individual functions

B.14.1 Get Video Configuration Byte

Input: AH = 44H
AL = 8EH

Output:	DH (bit 0)	State of Setup "Analog Monochrome Monitor"
	DL (bits 3:0)	State of Setup "Monitor Type"

B.14.2 Get Video Options Byte

Input:	AH = 44H	
	AL = 8FH	
Output:	AH = 0	
	DH (bit 7)	State of Setup "Video Registers"
	(bit 6)	State of CMOS byte 2CH bit 4
	DL (bits 7:5)	State of CMOS byte 2DH bits 7:5

B.14.3 Keyboard Intercept

The H/W keyboard interrupt INT 09H routine calls the keyboard intercept asynchronously. This allows keystroke changes. Normally the system returns with the scan code unchanged, but the operating system can redirect interrupt 15H to its own routine and

- Replaces (AL) with a different scan code and return with the carry flag set, effectively changing the keystroke.
- Processes the keystroke and return with the carry flag cleared, causing the interrupt 09H routine to ignore the keystroke.

The RET 2 instruction must be used when returning from scan code handling. This clears the stack and prevents the flag just set from being overwritten with its previous state.

Input:	AH = 4FH	
Output:	AL	Scan code
	CY = 1	Handle code
	CY = 0	Ignore keystroke

B.14.4 Device Open

This function reserved for the operating system.

Input: AH = 80H
BX Device ID
CX Process ID

Output: AH = 0
CY = 0

B.14.5 Device Close

This function reserved for the operating system.

Input: AH = 81H
BX Device ID
CX Process ID

Output: AH = 0
CY = 0

B.14.6 Program Termination

This function reserved for the operating system.

Input: AH = 82H
BX Device ID

Output: AH = 0
CY = 0

B.14.7 Event Wait

This function sets a timer which counts off CX:DX microseconds. When the timer expires, the high bit of the byte pointed to by ES:BX sets. Make sure the bit clears initially, and monitor the bit after invoking this function. The real-time clock periodic interrupt counts the interval (1/1024th of a second). The event wait handler takes this into account, however, the shortest interval available is 1 ms.

Input:	AH = 83H	
	AL = 0	Set interval
	AL = 1	Cancel
	ES:BX	Pointer to a byte
	CX:DX	Number of microseconds to wait
Output:	CY = 0	Not busy (AL not equal to 0)
	CY = 1	Busy (AL = 0)
	AH = 0	

B.14.8 Joystick Support

This routine supports an interface between a joystick interface and the system.

Input: AH = 84H
DX = 0 Read current switch settings
DX = 1 Read resistive inputs

Output: CY = 1 Invalid call
If DX = 0
AL = Switch setting (bits 7:4)

If DX = 1

AX = a(x) value
BX = a(y) value
CX = b(x) value
DX = b(y) value

B.14.9 System Request Key Pressed

Pressing the system request key (SysRq) key loads the BIOS keyboard hardware interrupt routine, loads AX with 8500H, and invokes INT 15H. This vector can be revectored. Releasing the key invokes INT 15H, this time with 8501H in AX.

Input: AH = 85H
AL = 00 key pressed
AL = 01 key released

Output: AH = 0
CY = 0

B.14.10 Wait

This function waits for the specified number of microseconds before returning. The real-time clock periodic interrupt counts the interval (1/1024th of a second). The event wait handler takes this into account, however, the shortest interval available is 1 ms.

Input:	AH = 86H	
	CX,DX	Number of microseconds to elapse before return to caller
Output:	CY = 0	Function successful
	CY = 1	Wait function already in progress

B.14.11 Move Block

This function transfers up to 32,768 words to or from memory.

Input:	AH = 87H	
	CX	Number of words to move, maximum count = 8000H (32K) words
	ES:SI	Global descriptor table (GDT) pointer. User must set up the GDT.
Output:	AH = 00H	Successful
	AH = 01H	RAM parity (parity error cleared on return)
	AH = 02H	Exception interrupt occurred
	AH = 03H	Gate address line 20 failed
	CY = 0	Successful
	CY = 1	Error

B.14.12 Extended Memory Size Determination

This routine returns the number of consecutive 1K blocks above 1M. CMOS RAM locations 30H and 31H, set when the system was started, obtains this value.

Input: AH = 88H

Output: AX Number of consecutive 1K blocks starting at 1M

B.14.13 Switch Processor to Protected Mode

This routine sets the coprocessor to protected mode.

Input: AH = 89H
ES:SI Pointer to GDT. User must set up the GDT
BH Offset into interrupt descriptor table (IDT) where first eight 8259 interrupts are to occur
BL Offset into IDT where second eight 8259 interrupts occur

Output: AH = 0 If successful
Segment registers set for protected mode operation, AX and BP destroyed

B.14.14 Device Busy

This function reserved for the operating system.

Input: AH = 90H
AL Type code

Output: CY = 0 No error
CY = 1 Timeout

Type	Description	Timeout
00	Fixed disk	yes
01	Disk	yes
02	Keyboard	no
80	Network	no
FC	Fixed disk reset	yes
FD	Drive motor start	yes
FE	Printer	yes

B.14.15 Interrupt Complete

This function reserved for the operating system.

Input: AH = 91H
AL Type code

Output: None, AX destroyed

B.14.16 Return System Configuration Parameters

This routine gives information about the model of the board, the BIOS revision level, and hardware features.

Input: AH = C0H

Output: CY = 0
AH = 0
ES:BX Pointer to system descriptor table in ROM

System Descriptor Table:

Size	Description	Value
Word	Descriptor length (bytes)	008H
Byte	Model	FCH
Byte	Submodel	81H
Byte	BIOS revision level	00H
Byte	Feature information	70H
Bit 7 = 1	BIOS uses DMA channel 3	
Bit 6 = 0	One interrupt controller	
Bit 5 = 1	Real-time clock present	
Bit 4 = 1	System hook in keyboard interrupt routine	
Bit 3 = 0	Reserved	
Bit 2 = 1	Extended BIOS data area allocated	
Bit 1 = 0	Reserved	
Bit 0 = 0	Reserved	

B.14.17 Get Mouse Data Area

This function reserved for the operating system.

Input: AH = C1H

Output: CF = 0
ES = Segment of mouse data
Mouse data starts at ES:22

NOTE
This is not a complete extended BIOS data area. It is only for mouse data.

B.15 KEYBOARD I/O (INT 16H)

This program interface provides an interface to access keys placed into the keyboard buffer by the keyboard hardware interrupt.

Functions:	AH = 00H	Read next character
	AH = 01H	Read buffer status
	AH = 02H	Return shift status
	AH = 03H	Set typematic rate and delay
	AH = 05H	Place ASCII character/scan code in keyboard buffer
	AH = 10H	Extended read interface for the enhanced keyboard
	AH = 11H	Extended buffer status for the enhanced keyboard
	AH = 12H	Return the extended shift status for the enhanced keyboard

B.15.1 Read Next Character

This function returns the scan code and ASCII code of the next character in the keyboard buffer, and updates the buffer pointer. If the keyboard buffer is empty, the function waits for a key.

Input: AH = 00H

Output: AH Scan code
AL ASCII character (0 for special keys, function keys, etc.)

B.15.2 Read Buffer Status

This function returns the keyboard buffer status (indicates whether a keystroke is available or not). The buffer pointer is not updated, even though the key returns. Use function 00H to update the buffer pointer.

Input: AH = 01H

Output: ZF = 1 No keystroke queued
ZF = 0 Keystroke available in queue, key in AX

B.15.3 Return Shift Status

Pressing a shift key does not transfer code/ASCII code into the keyboard buffer. Instead, a shift status bit sets. This function indicates which key is pressed.

Input: AH = 02H

Output:

AL	Shift status
Bit 0 = 1	Right-Shift pressed
Bit 1 = 1	Left-Shift pressed
Bit 2 = 1	Ctrl-Shift pressed
Bit 3 = 1	Alt-Shift pressed
Bit 4 = 1	Scroll Lock state
Bit 5 = 1	Num Lock state
Bit 6 = 1	Caps Lock state
Bit 7 = 1	Insert state

All other registers are restored

B.15.4 Set Typematic Rate and Delay

Pressing a key (except a shift key) causes the keyboard handler to repeat the key until the key is released. This function allows the keyboard typematic rate and keyboard delay time to be set. The typematic rate is the rate at which the handler repeats the key. Delay time is the time between the first keystroke and the repeated keystrokes.

Input: AH = 03H

BL Typematic rate (bits 4:0)

BL = 00H	30.0	BL = 10H	7.5
BL = 01H	26.7	BL = 11H	6.7
BL = 02H	24.0	BL = 12H	6.0
BL = 03H	21.8	BL = 13H	5.5
BL = 04H	20.0	BL = 14H	5.0
BL = 05H	18.5	BL = 15H	4.6
BL = 06H	17.1	BL = 16H	4.3
BL = 07H	16.0	BL = 17H	4.0
BL = 08H	15.0	BL = 18H	3.7
BL = 09H	13.3	BL = 19H	3.3
BL = 0AH	12.0	BL = 1AH	3.0
BL = 0BH	10.9	BL = 1BH	2.7
BL = 0CH	10.0	BL = 1CH	2.5
BL = 0DH	9.2	BL = 1DH	2.3
BL = 0EH	8.5	BL = 1EH	2.1
BL = 0FH	8.0	BL = 1FH	2.0

BH Delay value (bits 1:0)

BH = 00H	250 milliseconds
BH = 01H	500 milliseconds
BH = 02H	750 milliseconds
BH = 03H	1000 milliseconds

Output: None

B.15.5 Place ASCII Character/Scan Code in Keyboard Buffer

This function places an ASCII character/scan code combination into the keyboard buffer.

Input:	AH = 05H	
	CL	ASCII code
	CH	Scan code
Output:	AL = 00H	Successful operation
	AL = 01H	Keyboard buffer full

B.15.6 Extended Read Interface for the Enhanced Keyboard

This function corresponds to AH = 0, but supports the enhanced keyboard.

Input:	AH = 10H	
Output:	AH	Scan code
	AL	ASCII code

B.15.7 Extended Buffer Status for the Enhanced Keyboard

This function corresponds to AH = 1, but supports the enhanced keyboard.

Input:	AH = 11H	
Output:	ZF = 1	No keystroke queued
	ZF = 0	Keystroke is available, key in AX

B.15.8 Return the Extended Shift Status for the Enhanced Keyboard

This function corresponds to AH = 2, but supports the enhanced keyboard.

Input: AH = 12H

Output: AL

Bit 0 = 1	Right-Shift key pressed
Bit 1 = 1	Left-Shift key pressed
Bit 2 = 1	Control key pressed
Bit 3 = 1	Alt key pressed
Bit 4 = 1	Scroll Lock state
Bit 5 = 1	Num Lock state
Bit 6 = 1	Caps key state
Bit 7 = 1	Insert state

Output: AH

Bit 0 = 1	Left-Ctrl-Shift
Bit 1 = 1	Left-Alt-Shift
Bit 2 = 1	Right-Ctrl-Shift
Bit 3 = 1	Right-Alt-Shift
Bit 4 = 1	Scroll-Lock-Shift
Bit 5 = 1	Num-Lock-Shift
Bit 6 = 1	Caps-Lock-Shift
Bit 7 = 1	SysRq-Shift

B.16 PRINTER I/O (INT 17H)

This program interface allows access to the printer

Functions:	AH = 00H	Print character
	AH = 01H	Initialize printer port
	AH = 02H	Read printer status
Input:	AH	Function number
	DX	Device number (0:2)

Output: See individual functions

B.16.1 Print Character

This function writes the character in AL to the specified printer.

Input:	AH = 00H	
	AL	Character
	DX	Device number (0:2)
Output:	AH = 0FH	Invalid port
	AH	Printer status
	Bit 0 = 1	Timeout
	Bits 1:2	Reserved
	Bit 3 = 1	I/O error
	Bit 4 = 1	Printer selected
	Bit 5 = 1	Out of paper
	Bit 6 = 1	Acknowledge
	Bit 7 = 1	Not busy

B.16.2 Initialize Printer Port

Input:	AH = 01H	
	DX	Device number (0:2)
Output:	AH	Printer status
	Bit 0 = 1	Timeout
	Bits 1:2	Reserved
	Bit 3 = 1	I/O error
	Bit 4 = 1	Printer selected
	Bit 5 = 1	Out of paper
	Bit 6 = 1	Acknowledge
	Bit 7 = 1	Not busy

B.16.3 Read Printer Status

Input:	AH = 02H	
	DX	Device number (0:2)
Output:	AH	Printer status
	Bit 0 = 1	Timeout
	Bits 1:2	Reserved
	Bit 3 = 1	I/O error
	Bit 4 = 1	Printer selected
	Bit 5 = 1	Out of paper
	Bit 6 = 1	Acknowledge
	Bit 7 = 1	Not busy

B.17 SYSTEM BOOT (INT 19H)

This routine reads the boot-sector from the disk drive into main memory and executes it. Memory does not clear unless invoked by the BIOS self-test and initialization. If a disk time-out is encountered, the routine attempts to boot from the fixed disk drive. If this fails, INT 18H invokes.

Input: None

Output: None

B.18 CLOCK SERVICES (INT 1AH)

This routine allows the clock to be set or read.

Functions:	AH = 00H	Read the system timer count
	AH = 01H	Set the system timer count
	AH = 02H	Read the real-time clock time
	AH = 03H	Set the real-time clock time
	AH = 04H	Read the real-time clock date
	AH = 05H	Set the real-time clock date
	AH = 06H	Set the real-time clock alarm
	AH = 07H	Reset the real-time clock alarm

Input: Function number

Output: See individual functions

B.18.1 Read the System Timer Count

This function reads the system timer from the BIOS data area. Each system timer interrupt (INT 08H), approximately 18.2 times per second, increments this value.

Input: AH = 00H

Output: CX High count word
DX Low count word
AL = 0 Timer has not passed midnight since last read
AL > 0 Timer has passed midnight since last read
AH = 0

B.18.2 Set the System Timer Count

This function sets the system timer in the BIOS data area.

Input: AH = 01H
CX High count word
DX Low count word

Output: AH = 0

B.18.3 Read the Real-Time Clock Time

This function reads hours, minutes, and seconds from the real-time clock.

Input:	AH = 02H	
Output:	CH	Hours in BCD
	CL	Minutes in BCD
	DH	Seconds in BCD
	DL = 1	Daylight savings time
	DL = 0	Standard time
	CY = 0	Successful
	CY = 1	Clock is busy

B.18.4 Set the Real-Time Clock Time

This function sets hours, minutes, and seconds in the real-time clock.

Input:	AH = 03H	
	CH	Hours in BCD
	CL	Minutes in BCD
	DH	Seconds in BCD
	DL = 1	Daylight savings time
	DL = 0	Standard time
Output:	AH = 00H	
	CY = 0	Successful
	CY = 1	Clock is busy

B.18.5 Read the Real-Time Clock Date

This function reads century, year, month, and day from the real-time clock.

Input:	AH = 04H	
Output:	CH	Century in BCD (19 or 20)
	CL	Year in BCD
	DH	Month in BCD
	DL	Day in BCD
	AH = 00H	
	CY = 0	Successful
	CY = 1	Clock busy

B.18.6 Set the Real-Time Clock Date

This function sets century, year, month, and day into the real-time clock.

Input:	AH = 05H	
	CH	Century in BCD (19 or 20)
	CL	Year in BCD
	DH	Month in BCD
	DL	Day in BCD
Output:	AH = 00H	
	CY = 0	Successful
	CY = 1	Clock busy

B.18.7 Set the Real-Time Clock Alarm

This function sets the alarm (hours, minutes, and seconds) to the real-time clock. When the time is up, INT 4AH invokes.

```
Input:  AH = 06H
        CH      Hours in BCD
        CL      Minutes in BCD
        DH      Seconds in BCD

Output: AH = 00H
        CY = 0  Successful
        CY = 1  Clock busy
```

B.18.8 Reset the Real-Time Clock Alarm

This function resets the alarm and turns off the alarm enable.

```
Input:  AH = 07H

Output: AH = 00H
        CY = 0  Successful
        CY = 1  Clock busy
```

B.19 REAL-TIME CLOCK (INT 70H)

The following three conditions in the real-time clock may cause this interrupt:

- The current time matches the time set in the alarm.
- The periodic interval counter expires.
- The real-time clock completes an internal update cycle.

Each separately enabled condition must be enabled for an interrupt to occur. In a standard system, only the periodic interrupt is enabled. BIOS uses this periodic interrupt to time certain events in the system. If an alarm interrupt occurs, BIOS invokes INT 4AH and gives the program a chance to handle the alarm condition. The update cycle complete interrupt is not used.

B.20 AUXILIARY DEVICE (MOUSE) INTERRUPT (INT 74H)

Interrupt 70H is the interrupt service routine for auxiliary devices. When the mouse has data to transmit to the system, it sends the data to the keyboard and mouse controller. The data is transmitted to a data package. After the keyboard and mouse controller receives the data, it interrupts the system on IRQ12. INT 74H collects each byte of the data package and stores it in the BIOS extended data area. INT 74H also sets appropriate flags if an acknowledge, resend, or self-test completion code is received. When all information in the data package has been collected and stored, INT 74H pushes the data onto the stack and calls the auxiliary device driver.

B.21 COPROCESSOR INTERRUPT (INT 75H)

This interrupt is caused by the 387SX™ numeric coprocessor error, which occurs with interrupts enabled at the coprocessor. To retain compatibility with 8088-based systems, the IRQ13 handler issues a software INT 02H command. INT 02H causes a software generated NMI.

If an application preempts the NMI vector to handle coprocessor errors (as it must to retain compatibility with 8088-based systems), it must be sensitive to NMIs generated by hardware, and pass them on to the system NMI.

B.22 FIXED DISK H/W INTERRUPT (INT 76H)

This interrupt occurs when the fixed disk drive requires attention. During multi-sector transfers, the controller interrupts for each transfer. A completed operation also causes an interrupt. The BIOS handler moves 0FFH to 40:8EH indicating the interrupt occurred.

Jumper Settings

C

C.1 INTRODUCTION

Jumper pins allow specific system operating parameters to be set into the system. Various options require changing jumper blocks to reflect the correct system configuration.

Figure C-1 diagrams the system board jumper pin locations. Jumper pin blocks are arranged into the following functional groups:

- Monitor type monochrome/color
- Password enable/clear
- ROM size (512K/256K)
- Video enable/disable

NOTE
The POST must be executed after any changes are made to the jumper settings.

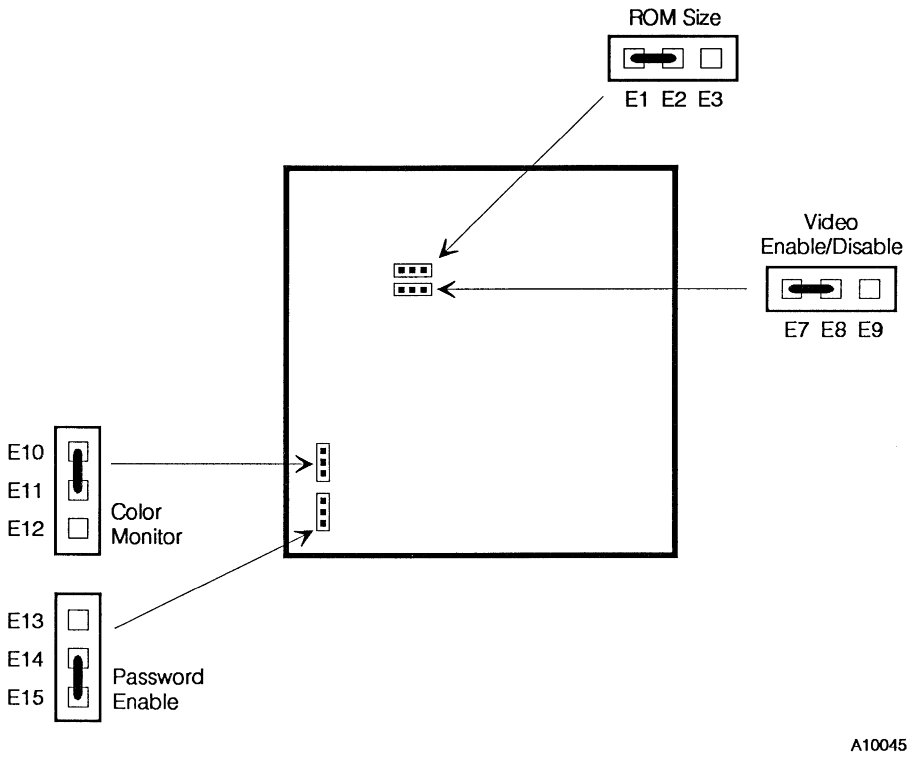


Figure C-1. System Board Jumper Locations

C.2 MONITOR TYPE

The monitor type jumper allows you to indicate the type of monitor connected to the 300SX board's TTL video output connector. If the monitor is one of the following, the jumper must be set to MD (monochrome display) as indicated in Table C-1:

Monochrome Displays:

- Princeton Graphics MAX-12
- Amdek 310A
- IBM 5151

If you are connecting any other monitor, the jumper should be set to color. If the TTL connector is not used, the jumper should be set to color.

Table C-1. Color/Monochrome Monitor

Monitor Type	Jumper Pin Setting
Color	E10 to E11†
Monochrome	E11 to E12

† Default setting.

C.3 PASSWORD ENABLE/CLEAR

This jumper enables or clears the password function. If the password is enabled, a password may be entered into the board ROM using the SETUP utility. The jumper settings are shown in Table C-2.

Table C-2. Password Enable/Clear

Option	Jumper Pin Setting
Enable	E13 to E14
Clear	E14 to E15†

† Default setting.

C.4 ROM SIZE

If the onboard video subsystem is being used, the ROM size jumper must be set to 512 to indicate 512K devices. If the onboard video subsystem is disabled, the ROM sized jumper should be set to 256K devices, to disable the onboard video BIOS. The jumper settings are shown in Table C-3.

Table C-3. Password Enable/Clear

Option	Jumper Pin Setting
512K Devices	E1 to E2†
256K Devices	E2 to E3

† Default setting.

C.5 VIDEO ENABLE/DISABLE

If an offboard video controller is installed in one of the ISA bus expansion slots, the onboard video controller must be disabled to prevent a conflict between the two controllers. The jumper settings are listed in Table C-4.

Table C-4. Video Enable/Disable

Option	Jumper Pin Setting
Onboard Video enabled	E7 to E8†
Onboard video disabled	E8 to E9

† Default setting.

⇛ **NOTE**

To completely disable the onboard video circuitry, the ROM size jumper should be set to 256K devices when you disable onboard video

Changing CPU Speed

D

D.1 INTRODUCTION

This chapter describes how to change the speed of the CPU on the 300SX board.

D.2 CPU CLOCK SPEED

The CPU runs at a clock speed of 16 MHz, resulting in system clock cycles of 625 ns. However, there are some applications (such as those installing some copy-protected software), that cannot operate at a clock speed of 16 MHz, and instead require slower operation. To meet this requirement, a special mode can be initiated which enables the board to effectively operate at a slower speed to simulate the performance of a 8 MHz IBM AT. This special mode of operation is called *deturbo* mode. The opposite of deturbo mode (normal system operation) is sometimes referred to as *turbo* mode. Deturbo mode can be enabled in the setup program, from the keyboard, or with a software program. Conversely, the normal or turbo mode can be enabled in the same manner.

Deturbo mode does not actually affect the clock rate of the 386SX nor the 387SX. Deturbo mode inserts CPU wait states between each instruction, resulting in an effective CPU clock speed of 8 MHz.

If the system is put in deturbo mode, it will remain in deturbo mode even after warm reboots (Ctrl + Alt + Delete) or other software resets. However, a power reset will put the CPU back into normal (turbo) 16 MHz operation.

D.3 KEYBOARD METHOD

The following keystroke sequences can be used to set the CPU speed.

Ctrl + Alt + 1 Hold down the Ctrl and Alt keys and press 1 on the numeric keypad to place the system in deturbo mode.

Ctrl + Alt + 2 Hold down the Ctrl and Alt keys and press 2 on the numeric keypad to place the system in turbo mode.

An audible tone occurs when you change the CPU clock speed via the keyboard. A low-pitched tone will be emitted when the system is placed into deturbo mode. A high-pitched tone will be emitted when the system is returned to normal operation.

D.4 SETUP METHOD

The CPU clock speed can be changed using the Setup program. The CPU clock speed chosen determines the speed used by the 300SX system each time it is powered up.

To invoke Setup, press Ctrl + Alt + Ins. Press the ↑ or ↓ to move the cursor to the CPU speed field. Press the ← or → to display the turbo or deturbo options. Exit Setup by pressing Esc.

D.5 SOFTWARE METHOD

Figure D-1 lists the assembly language code for the *SLOW.COM* routine. This routine is used to place the system in deturbo mode. Figure D-2 lists the assembly language code for the *FAST.COM* routine. This routine is used to return the system to full speed operation. Both of these programs run under DOS and can be assembled using the Microsoft Macro Assembler. Assuming that the *FAST.COM* routine shown in Figure D-2 is contained in a file called *FAST.ASM*, the following commands can be used to assemble the program and make it ready to run (user input is in italic type). The procedure for preparing the *SLOW.COM* program shown in Figure D-1 is analogous.

```
C:>masm fast, fast, fast;

Microsoft (R) Macro Assembler Version 4.00
Copyright (C) Microsoft Corp 1981, 1983, 1984, 1985.  All rights reserved.

  50460 Bytes symbol space free
        0 Warning Errors
        0 Severe Errors

C:>link fast;

The IBM Personal Computer Linker
Version 2.40  (C) Copyright International Business Machines Corp 1981, 1985 (C)
Copyright Microsoft Corp. 1981, 1985

Warning: no stack segment

C:>exe2bin fast.exe fast.com
C:>del fast.obj
C:>del fast.exe
```

```
      name    slow
        title   'SLOW.COM -- puts board into deturbo mode
                (simulated 8 MHz)'

cr      equ     0dh        ; ASCII carriage return
lf      equ     0ah        ; ASCII line feed
cmnd    equ     64h        ; command register

code    segment public

        org     100h       ; COM file

                assume  cs:code,ds:code

start:
        mov     ax, cs     ; set ds equal to cs
        mov     ds,ax

        mov     al,0eah    ; value for deturbo mode
        out     cmnd,al    ; write to command port

        mov     dx,offset deturbo ; tell user that switch occurred

        mov     ah,9      ; use DOS function 9 to print the string
        int     21h

        mov     ax,4c00h  ; exit back to DOS with
        int     21h       ; a return code of zero

deturbo db      cr,lf,'Now running in deturbo mode.'cr,lf,'$'

code    ends
end     start
```

Figure D-1. Code Listing for SLOW.COM Routine (Setting Deturbo Mode)

```
        name    fast
        title   'FAST.COM -- puts board into turbo mode

cr      equ     0dh        ; ASCII carriage return
lf      equ     0ah        ; ASCII line feed
cmnd    equ     64h        ; command register

code    segment public

        org     100h       ; COM file

                assume  cs:code,ds:code

start:
        mov     ax, cs     ; set ds equal to cs
        mov     ds,ax

        mov     al,0e5h    ; value for turbo mode
        out     cmnd,al    ; write to command port

        mov     dx,offset turbo ; tell user that switch occurred

        mov     ah,9      ; use DOS function 9 to print the string
        int     21h

        mov     ax,4c00h  ; exit back to DOS with
        int     21h       ; a return code of zero

turbo   db      cr,lf,'Now running in turbo mode.'cr,lf,'$'

code    ends
end     start
```

Figure D-2. Code Listing for FAST.COM Routine (Setting Turbo Mode)

Messages

E

E.1 INTRODUCTION

This appendix describes the various system screen messages and error beep codes. Information is grouped as follows:

- POST and boot error messages
- POST and boot informational messages
- Run-time messages
- System board errors
- Beep codes for fatal errors
- Beep codes for non-fatal errors

E.2 POST AND BOOT MESSAGES

The POST displays messages to indicate errors in hardware, software, or firmware, or to provide other information.

If the POST can display a message on the video display screen, it will beep the speaker twice as the message appears. However, when an error occurs before the video display is initialized, the POST cannot display messages on the screen. POST sounds a series of beeps instead.

The next three sections provide a general grouping of messages, with each group arranged in alphabetical order. Each message is accompanied by a short paragraph describing the message and gives a recommended solution to the problem.

Italics indicate variable parts of a message such as memory addresses. These variable parts of the message may differ at each occurrence.

E.2.1 POST and Boot Error Messages

Message: `Coprocessor failed`

Possible Cause: Coprocessor not correctly sealed in its socket or the coprocessor has failed.

Solution: Reseat the coprocessor chip. Rerun the Setup program. If the coprocessor still fails, call your service representative.

Message: `Diskette drive 0 seek failure`

Possible Cause: Drive A has either failed or is missing.

Solution: Make sure drive A is present and the floppy disk is inserted properly. If they are, then drive A may have failed.

Message: `Diskette drive 1 seek failure`

Possible Cause: Drive B has either failed or is missing.

Solution: Make sure drive B is present and the floppy disk is inserted properly. If they are, then drive B may have failed.

Message:

```
Diskette read failure -
strike F1 to retry boot
```

Possible Cause: The two most likely causes for this condition are: (1) the floppy disk is not bootable, and (2) the floppy disk is defective.

Solution: Replace the floppy disk with a bootable floppy disk and try again. Clean the drive heads if necessary.

Message: `Diskette subsystem reset failed`

Possible Cause: The floppy disk adapter cable has failed.

Solution: Check the floppy disk adapter cable.

Message: `Display adapter failed; using alternate`

Possible Cause: The video display type jumpers are set incorrectly or the primary video adapter failed.

Solution: Make sure the video display type jumpers are set correctly. Check the primary video adapter.

NOTE

The following basic message precedes any of the possible errors numbered 1 through 6. These errors can only occur when the setup program is run.

Message:

```
Errors have been found during the power on self
test in your system.  The errors were:

1  Clock chip lost power

2  CMOS checksum invalid

3  Incorrect configuration data in CMOS

4  Memory size in CMOS invalid

5  Disk C: failed initialization

6  Time or Date in CMOS is invalid

          Hit any key to continue
```

Possible Cause: The configuration information stored in the real-time clock CMOS memory chip does not agree with the hardware configuration of the system.

Solution: Make sure the jumper settings on the system board agree with the hardware configuration. Run Setup and reenter data accounting for the indicated error.

Message: Gate A20 failure

Possible Cause: The system cannot switch into protected mode.

Solution: Call your service representative.

Message: Hard disk configuration error

Possible Cause: The specified configuration is incorrect.

Solution: Run Setup and enter the correct fixed disk drive type.

Message: Hard disk controller failure

Possible Cause: The fixed disk controller has failed.

Solution: Check the controller cable connections. If the message recurs, replace the fixed disk controller.

Message: Hard disk failure

Possible Cause: The fixed disk is defective.

Solution: Check the system configuration and drive type, and run Setup. Check the controller cable. If the message recurs, replace the fixed disk drive.

Message: Hard disk read failure -
strike F1 to retry boot

Possible Cause: The fixed disk is defective.

Solution: Check the system configuration and drive type, and run Setup. Check the controller cable. If the message recurs, replace the fixed disk drive.

Message: *hex-value* Optional ROM bad checksum = *hex-value*

Possible Cause: A peripheral card contains a defective ROM or its address conflicts with another card.

Solution: Replace the ROM or the peripheral card, or correct the address conflict.

Message: Invalid configuration information - please run Setup program

Possible Cause: The memory size is incorrect, the display is configured incorrectly, or the number of floppy disk drives is incorrect.

Solution: Check the system configuration and run Setup.

Message: Keyboard clock line failure

Possible Cause: Either the keyboard or the keyboard cable connection is defective.

Solution: Check the keyboard connection. If connection is good, the keyboard may have failed.

Message: Keyboard controller failure

Possible Cause: The keyboard controller has failed.

Solution: Call your service representative.

Message: Keyboard data line failure

Possible Cause: Either the keyboard or the keyboard cable connection is defective.

Solution: Check the keyboard connection. If connection is good, the keyboard may have failed.

Message: Keyboard is locked - please unlock -
Strike the F1 key to continue

Possible Cause: The keyboard lock (located at the front of the system) is activated.

Solution: Unlock the keyboard and try again.

Message: Keyboard stuck key failure

Possible Cause: One or more of the keys is pressed.

Solution: Release the key or keys and try again.

Message: Memory address line failure at *hex-value*, read *hex-value* expecting *hex-value*

Possible Cause: One of the SIMMs or associated circuitry has failed.

Solution: Check for defective SIMM and replace if necessary. If the message repeats, contact your service representative.

Message: Memory data line failure at *hex-value*, read *hex-value*, expecting *hex-value*

Possible Cause: One of the SIMMs or associated circuitry has failed.

Solution: Check for defective SIMM and replace if necessary. If the message repeats, contact your service representative.

Message: Memory double word logic failure at *hex-value*, read *hex-value* expecting *hex-value*

Possible Cause: One of the SIMMs or associated circuitry has failed.

Solution: Check for defective SIMM and replace if necessary. If the message repeats, contact your service representative.

Message: Memory high address line failure at *hex-value*, read *hex-value* expecting *hex-value*

Possible Cause: One of the SIMMs or associated circuitry has failed.

Solution: Check for defective SIMM and replace if necessary. If the message repeats, contact your service representative.

Message: Memory odd/even logic failure at *hex-value*, read *hex-value* expecting *hex-value*

Possible Cause: One of the SIMMs or associated circuitry has failed.

Solution: Check for defective SIMM and replace if necessary. If the message repeats, contact your service representative.

Message: Memory parity failure at *hex-value*, read *hex-value* expecting *hex-value*

Possible Cause: One of the SIMMs or associated circuitry has failed.

Solution: Check for defective SIMM and replace if necessary. If the message repeats, contact your service representative.

Message: Memory write/read failure at *hex-value*, read *hex-value* expecting *hex-value*

Possible Cause: One of the SIMMs or associated circuitry has failed.

Solution: Check for defective SIMM and replace if necessary. If the message repeats, contact your service representative.

Message: No boot device available -
strike F1 to retry boot

Possible Cause: If booting from a floppy disk, it is a non-bootable type or the floppy disk drive is defective. If booting from a fixed disk, it may not be formatted or the disk drive is defective. The problem could also be in the disk controller.

Solution: Make sure the floppy disk in drive A contains an operating system floppy disk. If applicable, make sure the fixed disk drive contains an operating system. Check the disk controller.

Message: No boot sector on hard disk -
strike F1 to retry boot

Possible Cause: The fixed disk is not formatted as a system disk.

Solution: Format the disk with the /S option.

Message: No timer tick interrupt

Possible Cause: The timer chip on the system board may have failed.

Solution: Contact your service representative.

Message: `Not a boot diskette -`
`strike F1 to retry boot`

Possible Cause: The floppy disk in drive A is not formatted as a system disk.

Solution: Replace the floppy disk with a bootable system floppy disk and try again.

Message: `Shadow of System BIOS failed - Executing from ROM -`
`Strike the F1 key to continue`

Possible Cause: System RAM is defective.

Solution: Check SIMMs and replace defective module.

Message: `Shadow of Video BIOS failed - Executing from ROM -`
`Strike the F1 key to continue`

Possible Cause: System RAM is defective or the video BIOS cannot be shadowed.

Solution: Check for defective SIMMs. If none are found, run the Setup program and turn Shadow of Video off.

Message: `Shutdown failure`

Possible Cause: The keyboard controller or its associated logic has failed.

Solution: Call your service representative.

Message: Time-of-day clock stopped

Possible Cause: The external battery for the clock is probably dead.

Solution: Replace the battery.

Message: Time-of-day not set -
Please run SETUP program

Possible Cause: The date and time information is not set in the real-time clock.

Solution: Run the Setup program and set the date and time.

Message: Timer chip counter 2 failed

Possible Cause: The timer chip on the system board may have failed.

Solution: Contact your service representative.

Message: Timer or interrupt controller bad

Possible Cause: The timer chip or the interrupt controller on the system board may have failed.

Solution: Contact your service representative.

Message: Unexpected interrupt in protected mode

Possible Cause: The system received an interrupt when in protected mode - probably while testing memory.

Solution: Contact your service representative.

E.2.2 POST and Boot Informational Messages

These messages do not indicate error conditions.

Message: *Hex-value*k Base Memory

Meaning: Indicates amount of base memory tested successfully.

Message: *Hex-value*k extended

Meaning: Indicates amount of extended memory tested successfully.

Message: *Hex-value*k of unshadowed memory added to extended memory and removed from dedicated memory.

Meaning: Indicates the amount of unused memory from 80000H to F0000H that has been added to the end of extended memory.

Message: Decreasing available memory

Meaning: Follows any memory error message. Informs you available memory size is adjusted to avoid use of the failed memory.

Message: Memory test terminated by keystroke

Meaning: The spacebar was pressed during the memory test. Reboot the system if you want to run to the POST.

Message: Strike the F1 key to continue.

Meaning: The POST detected an error prior to boot. Pressing the F1 key lets the system try to boot.

E.3 RUN-TIME MESSAGES

Run-time messages are displayed if an error occurs after the boot process is complete.

Message: I/O card parity interrupt at address *hex-value*. Type (S)hut off NMI, (R)eboot, other keys to continue

Possible Cause: A peripheral card has failed.

Solution: Type S to shut off the nonmaskable interrupt (NMI). This will temporarily allow you to continue. Replace the peripheral card.

Message: Memory parity interrupt at address *hex-value*. Type (S)hut off NMI, (R)eboot, other keys to continue

Possible Cause: One or more memory chips has failed.

Solution: Type S to shut off the nonmaskable interrupt (NMI). This will temporarily allow you to continue. Check the seating of the SIMMs and replace any defective SIMM.

Message: Unexpected HW interrupt at address *hex-value*. Type (R)eboot, other keys to continue

Possible Cause: This could be any hardware-related problem.

Solution: Check the hardware or call your service representative.

Message: Unexpected SW interrupt at address *hex-value*. Type (R)eboot, other keys to continue

Possible Cause: There is an error in the software program.

Solution: Try turning the system off and then on again. If that does not work, check the program.

Message:	`Unexpected type 02 interrupt at address` *`hex-value`*`. Type (S)hut off NMI, (R)eboot, other keys to continue`
Possible Cause:	There is an error in the software program.
Solution:	Try turning the system off and then on again. If that does not work, check the program.

E.4 SYSTEM BOARD ERRORS

If the POST finds an error and cannot display a message on the video display, the POST issues a series of beeps indicating the error and places a value in I/O port 80H.

For example, a failure of bit 3 in the first 64K of RAM is indicated by a 2-1-4 beep code (a burst of two beeps, a single beep, and a burst of four beeps). In addition, the POST writes a value to I/O port 80H to enable debugging tools to identify the area of failure.

Tables E-1 and E-2 list the beep codes and I/O the values that the POST writes to I/O port 80H when it encounters error conditions. Table E-1 lists fatal errors (errors that halt the system). Table E-2 lists the non-fatal errors (errors that are not serious enough to halt the system). Both tables list other conditions that have no beep codes.

One beep code is not listed in Tables E-1 or E-2: a long beep followed by one or more short beeps indicates a video adapter failure. No beep code is sounded if a test is aborted while in progress.

Table E-1. Beep Codes for Fatal Errors

Beep Code	Description of Error	Contents of I/O Port 80H
None	386SX register test in progress	01H
1-1-3	Real-time clock write/read failure	02H
1-1-4	ROM BIOS checksum failure	03H
1-2-1	Programmable interval timer failure	04H
1-2-2	DMA initialization failure	05H
1-2-3	DMA page register write/read failure	06H
1-3-1	RAM refresh verification failure	08H
None	1st 64K RAM test in progress	09H
1-3-3	1st 64K RAM chip or data line failure multi-bit	0AH
1-3-4	1st 64K RAM odd/even logic failure	0BH
1-4-1	1st 64K RAM address line failure	0CH
1-4-2	1st 64K RAM parity test in progress or failure	0DH
2-1-1	Bit 0 1st 64K RAM failure	10H
2-1-2	Bit 1 1st 64K RAM failure	11H
2-1-3	Bit 2 1st 64K RAM failure	12H
2-1-4	Bit 3 1st 64K RAM failure	13H
2-2-1	Bit 4 1st 64K RAM failure	14H
2-2-2	Bit 5 1st 64K RAM failure	15H
2-2-3	Bit 6 1st 64K RAM failure	16H
2-2-4	Bit 7 1st 64K RAM failure	17H
2-3-1	Bit 8 1st 64K RAM failure	18H
2-3-2	Bit 9 1st 64K RAM failure	19H
2-3-3	Bit A 1st 64K RAM failure	1AH
2-3-4	Bit B 1st 64K RAM failure	1BH
2-4-1	Bit C 1st 64K RAM failure	1CH
2-4-2	Bit D 1st 64K RAM failure	1DH
2-4-3	Bit E 1st 64K RAM failure	1EH
2-4-4	Bit F 1st 64K RAM failure	1FH
3-1-1	Slave DMA register failure	20H
3-1-2	Master DMA register failure	21H
3-1-3	Master interrupt mask register failure	22H
3-1-4	Slave interrupt mask register failure	23H
None	Interrupt vector loading in progress	25H
3-2-4	Keyboard controller test failure	27H
None	Real-time clock power failure or checksum failure	28H

Table E-2. Beep Codes for Non-fatal Errors

Beep Code	Description of Error	Contents of I/O Port 80H
none	Real-time clock configuration	29H
3-3-4	Screen memory test failure	2BH
3-4-1	Screen initialization failure	2CH
3-4-2	Screen retrace test failure	2DH
None	Search for video ROM in progress	2EH
None	Screen running with video ROM	30H
None	Monochrome display operable	31H
None	Color display (40 column) operable	32H
None	Color display (80 column) operable	33H

Device Mapping

F

F.1 INTRODUCTION

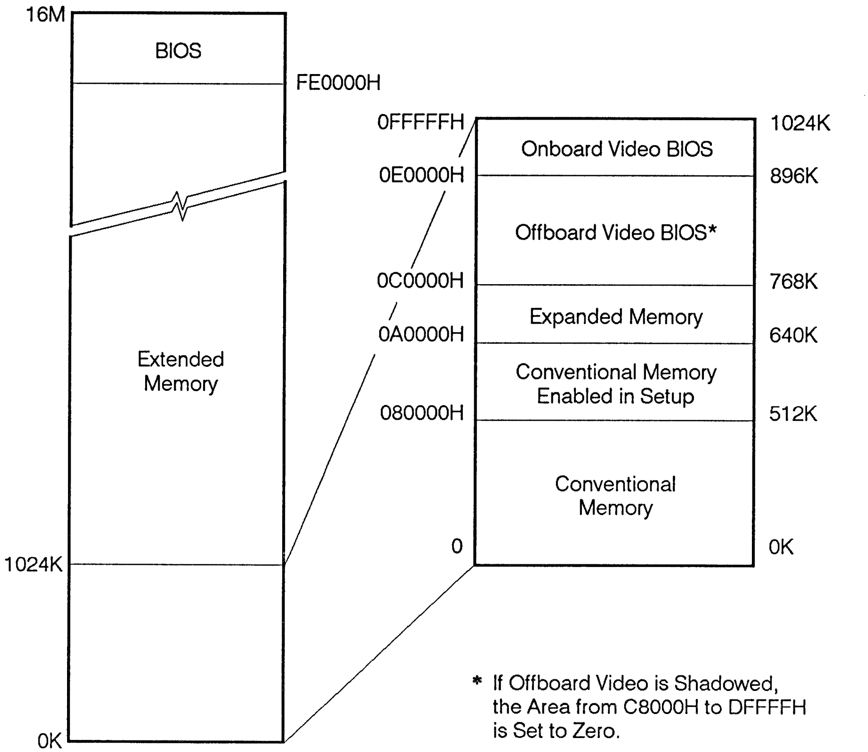
This appendix provides a series of tables listing mapping and address information related to system memory and onboard devices. Topics presented include the following:

- System memory map
- I/O address map
    - Control port bit assignments (61H)
    - Auxiliary control port bit assignments (78H and 79H)
- Interrupt priority levels
- DMA controller channel assignments
- Real-time clock map

You can find detailed information concerning the topics discussed in this appendix in other sections of this manual.

F.2 SYSTEM MEMORY MAP

Figure F-1 illustrates the memory map for the 300SX board.



* If Offboard Video is Shadowed, the Area from C8000H to DFFFFH is Set to Zero.

A10008

Figure F-1. Memory Address Map

F.3 I/O ADDRESSES

Tables F-1 through F-3 list the 300SX board I/O address maps.

Table F-1. I/O Address Map

Address (Hex)	Device
000-01F	DMA controller-1
020-3F	Interrupt controller-1
040-5F	System timer, PIT
060, 064	Keyboard controller
61	Control port
70	NMI mask (bit 7)
70-71	Real-time clock
78	Aux control port
080-09F	DMA page registers
0A0-0BF	Interrupt controller-2
0C0-0DF	DMA controller-2
0F0	Clear numeric coprocessor BUSY
0F1	Reset numeric coprocessor
0F8-0FF	Numeric coprocessor
102	Video display controller
1F0-1F8	Fixed disk controller
200-207	Game I/O port
278-27F	Parallel printer port 2
2F8-2FF	Serial port 2
378-37F	Parallel printer port 1 (primary)
3B0-3BF	Monochrome display/printer port
3C0-3CF	Enhanced graphics controller
3D0-3DF	Color/graphics controller
3F0-3F7	Floppy disk drive controller
3F8-3FF	Serial port 1 (primary)
400-44F	Multi-terminal adapter board
480-49F	DMA page registers
46E8	Video display controller
56E8	Video display controller
66E8	Video display controller
76E8	Video display controller
4BC4-4BC5	Video display controller

F.3.1 I/O Address 61H

I/O address 61H is an eight-bit control port. Table F-2 lists the 61H bit assignments.

Table F-2. I/O Address 61H Bit Assignment

Bit	Function
7	Onboard parity error
6	ISA parity error
5	Speaker signal
4	Refresh signal
3	Enable ISA parity error
2	Enable onboard parity error
1	Speaker data
0	Enable speaker

F.3.2 I/O Address 78H

I/O address 78H is an eight-bit auxiliary control port. Table F-3 lists the 78H bit assignments.

Table F-3. I/O Address 78H Bit Assignments

Bit	Function	True State
7	0=Enable speaker	L
6	1=Video BIOS shadowed	H
5	0=Floppy disk enabled	L
4	0=Fixed disk enabled	L
3	0=Onboard video enabled	L
2	1=Monochrome monitor	H
1	0=A20 enabled	L
0	1=Slow (CPU speed)	H

F.4 INTERRUPT PRIORITY LEVELS

Table F-4 lists the interrupt priority assignments for the 300SX board.

Table F-4. Interrupt Levels

Priority Number	PIC Number	Interrupt	Interrupt Source
1	†	NMI	Parity error detected
2	1	IRQ0	Interval timer, counter 0 output
3	1	IRQ1	Keyboard output buffer full
4	2	IRQ8	Real-time clock
5	2	IRQ9	On-board video, software redirected to INT 0AH (IRQ2)
6	2	IRQ10	Real-time clock INT
7	2	IRQ11	Reserved
8	2	IRQ12	Auxiliary device
9	2	IRQ13	INT from math coprocessor
10	2	IRQ14	Fixed disk controller
11	2	IRQ15	Reserved
12	1	IRQ3	Serial communications port 2
13	1	IRQ4	Serial communications port 1
14	1	IRQ5	Parallel port 2
15	1	IRQ6	Floppy disk controller
16	1	IRQ7	Parallel port 1

† I/O port 70H bit 7 controls the NMI signal.

The possibility of two or more interrupts demanding service at the same time exists. The PICs, after determining the priority of the interrupt requests, process the requests one at a time by transferring control of the CPU to the higher priority service routine first.

F.5 DMA CHANNEL ASSIGNMENTS

Each DMA controller has four ports: DMA4:1. DMA1 supports channels 3:0 and DMA2 supports channels 7:4. Channel four is used to cascade from DMA2 to DMA1. Both DMA controllers support eight channels (refer to Table F-5). All HOLD requests for DMA1 are processed via DMA2 channel four. This forces all channels in DMA1 to operate at a higher priority than those in DMA2. Channel zero has the highest priority and channel seven the lowest. DMA channels 0-3 are 8-bit channels and DMA channels 4-8 are 16-bit channels.

Table F-5. DMA Channel Assignment

Channel	Controller	Function
0	DMA1	Spare
1	DMA1	Spare
2	DMA1	Disk
3	DMA1	Spare
5	DMA2	Spare
6	DMA2	Spare
7	DMA2	Spare

F.6 REAL-TIME CLOCK MAPPING

The 64 addressable locations in the real-time clock are divided into two areas. The first area includes ten bytes containing the time, calendar, alarm data, and four control and status bytes. The remaining area, comprised of 50 general purpose RAM locations, stores system configuration information (refer to Table F-6).

Table F-6. Real-time Clock Address Map

Function	Index
Seconds	00
Seconds alarm	01
Minutes	02
Minutes alarm	03
Hours	04
Hours alarm	05
Day of week	06
Date of month	07
Month	08
Year	09
Status register A	0A
Status register B	0B
Status register C	0C
Status register D	0D
Diagnostic status byte	0E
Shutdown status byte	0F
Floppy disk drive type byte	10
Reserved	11
Fixed disk type byte	12
Reserved	13
Equipment byte	14
Low base memory byte	15
High base memory byte	16
Low expansion memory byte	17
High expansion memory byte	18
Drive C extended type byte	19
Drive D extended type byte	1A
Reserved	1B-1E
Features installed	1F
Drive type 48 parameters	20-27
Cache/shadow/setup	28
Reserved	29-2D
2-byte CMOS RAM checksum	2E-2F

(continued)

Table F-6. Real-Time Clock Address Map *(continued)*

Function	Index
Low expansion memory byte	30
High expansion memory byte	31
Date century byte	32
Setup information	33
CPU speed	34
Drive type 49 parameters	35-3C
Reserved	3D-3F

Hot Keys

G

This appendix lists the "hot keys." Hot keys are keystroke sequences used to invoke special system functions. Note that in these descriptions, all numbers refer to numeric pad keys.

Keystroke Sequence	Function
Ctrl + Alt + Insert	Enter ROM-based Setup program.
Ctrl + Alt + 0	Enter ROM-based Setup program.
Ctrl + Alt + 1	Set deturbo mode. CPU runs at 8 MHz.
Ctrl + Alt + 2	Set turbo mode. CPU runs at 16 MHz.
Ctrl + Alt + 4	Forces default video monitor mode (CGA).

Component Installation

H

H.1 INTRODUCTION

This appendix describes how to install and remove SIMMs and a 387SX numeric coprocessor on the 300SX board.

H.2 INSTALLING SIMMS

Installing SIMMs requires inserting two modules each in sockets U23 and U24. For each socket, insert the first SIMM module into the left-hand slot and the second SIMM module into the right-hand slot. To install SIMMs, perform the following:

C **CAUTION**
Do not touch any electronic component unless you are properly grounded. Proper grounding can be established by wearing a grounded wrist strap or touching an exposed metal part of the system module chassis. A static discharge from your fingers can result in permanent damage to the electronic component.

C **CAUTION**
Use extreme care when installing or removing SIMMs. The plastic retaining clips on the sockets are easily broken by using too much force.

1. Holding the SIMMs only by the edges, remove them from their antistatic package.

2. Position the SIMM correctly (see Figure H-1) and insert the bottom edge into the socket slot, beginning with the empty slot farthest to the left. Press down firmly while maintaining the angle of insertion.

3. Make sure the SIMM seats correctly. If not, gently spread the retaining clips just enough to permit the top edge of the SIMM to be pulled away from the clips and reseat the SIMM.

4. When the SIMM seats correctly, hold it at each end, and gently push the top edge toward the slot retaining clips until it snaps into place.

5. Repeat steps one through four and install the remaining SIMMs into the socket slots, working from left to right.

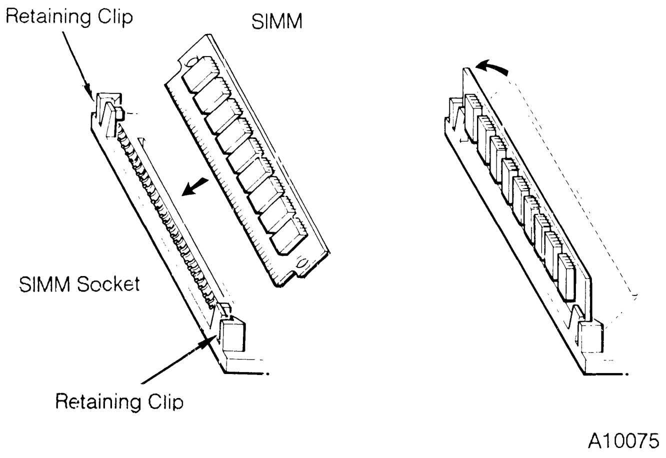


Figure H-1. Installing SIMMs

H.3 REMOVING SIMMS

Use the following procedures to remove SIMM modules from the system board.

When removing SIMMs, remove them one at a time from right to left. That is, remove the right-hand SIMM first, and the left-hand SIMM last.

CAUTION
Apply only enough pressure on the retaining clips to release the module. Too much pressure can break the retaining clips or damage the socket slot.

CAUTION
Do not touch any electronic component unless you are properly grounded. Proper grounding can be established by wearing a grounded wrist strap or touching an exposed metal part of the system module chassis. A static discharge from your fingers can result in permanent damage to the electronic component.

To remove a SIMM:

1. Locate the SIMM in the right-hand slot of the right-most socket of the group to be removed (see Figure H-2).
2. Gently spread the retaining clips just enough to pull the top edge of the SIMM away from the retaining clips.
3. Carefully lift the SIMM away from the socket and store it in a suitable static-free protective wrapper.
4. Repeat steps 2 and 3, as necessary, to remove and store SIMMs from the remaining sockets.

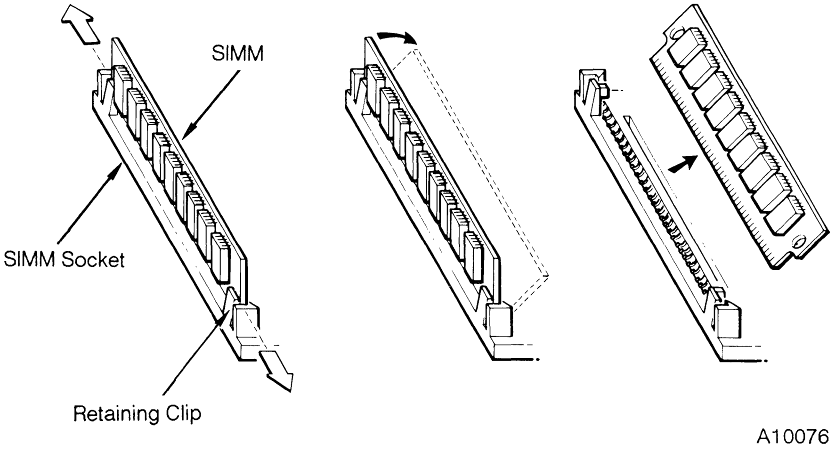


Figure H-2. Removing SIMMs

H.4 INSTALLING A NUMERIC COPROCESSOR

CAUTION
Do not touch any electronic component unless you are properly grounded. Proper grounding can be established by wearing a grounded wrist strap or touching an exposed metal part of the system module chassis. A static discharge from your fingers can result in permanent damage to the electronic component.

CAUTION
To avoid damaging the numeric coprocessor, make sure the circled indentation on the chip is positioned at the top left of the socket looking from the front of the board to the back.

The numeric coprocessor plugs directly into the socket on the board (see Figure 1-2). Follow these procedures to install the numeric coprocessor:

1. Remove the numeric coprocessor from its antistatic package, being careful not to touch the pins on the chip.

2. Align the numeric coprocessor's pins with the socket contacts.

3. Position the numeric coprocessor's pins in the socket receptacles. Press the chip down firmly until it seats (see Figure H-3). Be careful not to bend the pins.

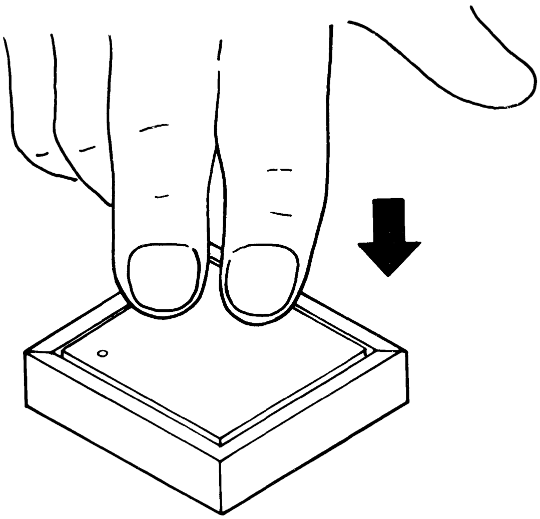


Figure H-3. Installing a Numeric Coprocessor

H.5 REMOVING A NUMERIC COPROCESSOR

Use the following procedures to remove the numeric coprocessor from the system board:

C **CAUTION**

Do not touch any electronic component unless you are properly grounded. Proper grounding can be established by wearing a grounded wrist strap or touching an exposed metal part of the system module chassis. A static discharge from your fingers can result in permanent damage to the electronic component.

1. Remove the numeric coprocessor from its socket on the board (see Figure 1-2) using a 68-pin grid array device extraction tool. Be careful not to touch the device pins.

2. Place the numeric coprocessor in an antistatic container to protect the device from static electricity.

If you are transferring the numeric coprocessor to a replacement board or replacing a defective device, follow the procedures described in section H.4.

Glossary

A

A
Ampere.

AC
Alternating current. A current that periodically reverses its direction of flow.

accuracy
In scientific measurements, accuracy is the degree of conformity to an absolute standard. For example, a specification of +5V ±10%, signifies how accurate +5V is with respect to the absolute volt as defined by the U.S. National Bureau of Standards or other governing standards organizations. Do not confuse accuracy with precision. Contrast with precision.

active high
Designates a signal that has to go high to produce an effect.

active low
Designates a signal that has to go low to produce an effect.

adapter

1. An auxiliary device or unit used to extend the operation of another system.
2. An electronic part used to connect two dissimilar parts or machines.

address

1. A name, label, or number identifying a location in storage, a device in a network, or any other data source.
2. A number that identifies the location of data in memory.

address bus
One or more conductors used to carry the binary-coded address from the processor throughout the rest of the system.

algorithm
A finite set of well-defined rules for the solution of a problem in a finite number of steps.

ampere(A)
The basic unit of electric current.

analog
Pertaining to data in the form of continuously variable physical quantities. Contrast with digital.

application
A program or set of programs used to do work on the computer. Some categories of application programs are word processors, database managers, spreadsheet managers, and project managers. Specific examples of application programs are MultiMate, dBase III PLUS, Lotus 1-2-3, Framework II, and Symphony.

array
An arrangement of elements in one or more dimensions.

ASCII
American Standard Code for Information Interchange. The code developed by ANSI for information interchange among data processing systems, data communications systems, and associated equipment. The ASCII character set consists of 7-bit control characters and symbolic characters.

asynchronous
In data communications, a method of transmission in which the bits included in a character or block of characters occur during a specific time interval. However, the start of each character or block of characters can occur at any time during this interval. Contrast with synchronous.

AUTOEXEC.BAT
A special-purpose batch file. When you turn on your computer, or restart it by pressing the Ctrl + Alt + Del combination, DOS searches the system disk for the AUTOEXEC.BAT file. If DOS finds one, it executes the commands in the file.

B

base address
The beginning address for resolving symbolic references to locations in storage.

base memory
Up to 640K of memory accessible to DOS. This is also referred to as conventional memory. Contrast with expanded memory and extended memory.

base register
A general purpose register that the programmer chooses to contain a base address.

BASIC
A programming language that uses common English words.

basic input/output system

The feature of a computer that provides a basic level of control of the major I/O devices, and relieves programmers of having to learn about system hardware device characteristics.

See BIOS.

batch file

A file that saves time and effort and which is identified by the .BAT extension following the file name. If you use a sequence of DOS commands frequently, you can create a batch file containing the commands, and then execute the entire sequence by typing the name of the file. This reduces the number of keystrokes needed to execute a sequence of commands.

binary

1. Involving a a choice of two conditions, such as on-off or yes-no.
2. Pertaining to a fixed radix numeration system having a radix of 2, wherein the binary digits are 0 and 1.

BIOS

The feature of a computer that provides a basic level of control of the major I/O devices, and relieves programmers of having to learn about system hardware device characteristics.

Acronym for basic input/output system.

bit

Synonym for binary digit. Either of the binary digits 0 or 1 used in computers to store information. (see also byte)

bits per second (bps)

A unit of measurement representing the number of discrete binary digits transmitted by a device in one second.

board

A rectangular piece of fiberglass that has pins on one side and electronic parts on the other; also called a card, PC board or PCB (printed circuit board). The system is always supplied with a system board. Other boards can include a video adapter board, a disk controller board, a network communication board, memory boards, and multifunction boards.

boot

(see bootstrap)

bootstrap

A technique or device designed to bring itself into a desired state by means of its own action; for example, a machine routine whose first few instructions are sufficient to bring the rest of itself into the computer from an input device. For example, a computer that runs DOS boots itself by causing the computer to read the first few of its instructions from disk. Those instructions are sufficient to read in the rest of DOS from disk.

bps
Bits per second.

buffer
1. An area of storage that is temporarily reserved for use in performing an input/output operation into which data is written or from which data is read. Synonymous with I/O area.
2. A portion of memory storage for temporarily holding input or output data.

bus
One or more conductors used for transmitting signals or power.

byte
1. A sequence of eight adjacent binary digits that are operated upon as a unit.
2. A binary character operated upon as a unit.
3. The amount of storage used to represent one character.

C

C
1. Celsius (centigrade).
2. A programming language.

cache memory
A small, high-speed memory block that improves CPU performance by minimizing the number of memory accesses on the bus. This action is accomplished by storing the CPU's most recently used data/instructions in the cache memory rather than main memory.

card
(see Board)

CAS
Column address strobe; a signal that latches the column addresses in a memory chip.

Cathode ray tube (CRT)
A vacuum tube in which a stream of electrons is projected onto a fluorescent screen producing a luminous spot. The location of the spot can be controlled. A CRT is the main element in a video display or monitor.

Celsius (C)
A temperature scale; also called Centigrade. Contrast with Fahrenheit .

Central processing unit (CPU)
Term for processing unit; i.e. 80386.

CFR Part 15 Subpart J
Federal Communications Commission Specification for EMI suppression.

channel
A path along which signals can be sent; for example, data channel, output channel.

character
A letter, digit, or other symbol.

character generator

1. In computer graphics, a functional unit that converts the coded representation of a graphics character into the shape of the character for display.
2. In word processing, the means within equipment for generating visual characters or symbols from coded sets.

character key
A keyboard key that allows the user to enter the character shown on the key. Compare with function key.

character set
A group of characters used for a specific reason; for example, the set of characters a printer can print or a keyboard can support.

Class A device
Broadly defined, a Class A device complies with the various regulatory agencies that certify equipment for operation in a commercial (office or factory) environment.

Class B device
Broadly defined, a Class B device complies with the various regulatory agencies that certify equipment for operation in a home or residential environment.

CMOS
Complementary metal oxide semiconductor. A logic circuit family that uses very little power. It works with a wide range of power supply voltages.

COM1, COM2, COM3, COM4
These are the names DOS assigns to the serial communications port(s). Some systems only provide one serial port; others provide two.

CONFIG.SYS file
A special-purpose file that provides DOS with information about the special kinds of hardware or software used with the computer. Whenever the computer is turned on or rebooted by pressing Ctrl + Alt + Del, DOS searches the system disk for the CONFIG.SYS file. If DOS finds one, it reads the commands from the file and uses them to prepare the computer for operation.

configuration

1. The arrangement of a computer system or network as defined by the nature, number, and chief characteristics of its functional units. More specifically, the term configuration may refer to a hardware configuration or a software configuration.
2. The devices and programs that make up a system, subsystem or network.

connector
A device allowing the connection of various electrical elements. See edge connector.

conventional memory
(see base memory)

cps
Characters per second.

CPU
Central processing unit.

CRT
Cathode ray tube.

CSA 22.2 # 220
Canadian Standards Association Standard for Safety of Information Processing and Business Equipment.

cursor

1. In computer graphics, a movable marker that is used to indicate position on a display.
2. A displayed symbol that acts as a marker to help the user locate a point in text, in a system command, or in storage.
3. A movable spot of light on the screen of a display device, usually indicating where the next character is to be entered, replaced, or deleted.

cycle time
Defines the minimum amount of time in which subsequent accesses to a DRAM device can occur.

cylinder
All fixed disk or diskette tracks that can be read or written without moving the disk drive or diskette drive read/write mechanism

D

data base
A collection of data that can be immediately accessed and operated upon by a data processing system for a specific purpose.

dB
(see Decibel)

DC
Direct current.

decibel

1. A unit that expresses the ratio of two power levels on a logarithmic scale.
2. A unit for measuring relative power.

Deutsche Industrie Norm (DIN)

1. German Industrial Norm.
2. The committee that sets German dimension standards.

diagnostic

Pertaining to the detection and isolation of a malfunction or mistake.

digital

Pertaining to data in the form of digits. Contrast with analog.

DIN

Deutsche Industrie Norm.

DIN connector

One of the connectors specified by the DIN committee.

DIP

Dual in-line package. DIPs have pins in two parallel rows. The pins are spaced 1/10 inch apart. (See also DIP switch.)

DIP switch

One of a set of small switches mounted in a dual in-line package.

direct current (dc)

A current that always flows in one direction.

direct memory access (DMA)

A method of transferring data between main storage and I/O devices that does not require processor intervention.

disable

To stop the operation of a circuit or device; specifically applied to communications ports.

disabled

Pertaining to a state of a processing unit that prevents the occurrence of certain types of interruptions. Synonymous with masked.

disk

Loosely, a magnetic disk.

disk drive

A device for storing data on and retrieving data from a fixed disk or diskette.

diskette
A thin, flexible magnetic disk, permanently sealed in a protective jacket, that is used to store information. Synonymous with floppy and flexible disk. Most commonly available in 5.25-inch and 3.5-inch sizes.

display

1. A visual presentation of data.
2. A device for visual presentation of information on any temporary character imaging device.
3. To present data visually.
4. See cathode ray tube.

DMA
Direct memory access.

DOS
Disk Operating System. (see operating system)

double precision
Pertaining to the use of two computer words to represent a number in accordance with the required precision. Contrast with single precision.

DRAM
Dynamic RAM. A type of RAM comprised of capacitive cells that require periodic refresh to maintain data. While the DRAM is a slower than the SRAM, its cell is much smaller. This enables the DRAM to be a higher density device and generally less expensive.

drop card
Drop cards are expansion boards that gain extra surface area by dropping down immediately after the 8-bit edge connector. Drop cards are not compatible in 16-bit and 32-bit expansion slots because the dropped portion of the card interferes with the unused connectors in these longer slots.

dual in-line package (DIP)
A widely used container for an integrated circuit.

dynamic memory
Random access memory (RAM). Read/write memory. See DRAM.

E

edge connector
A terminal block with a number of contacts attached to the edge of a printed-circuit board to facilitate plugging into a foundation circuit.

EIA
Electronic Industries Association.

EMI
Electromagnetic Interference.

enable
To initiate the operation of a circuit or device; specifically applied to communications ports.

EPROM
Erasable programmable read-only memory. A PROM that allows the user to change its code.

E^{2}PROM
Electrically erasable programmable read-only memory. An EPROM which allows the user to change its code by means of appropriate electrical signals.

ESDI
Enhanced Small Device Interface, which achieves faster throughput than standard fixed-disk controllers by shifting functionality to the fixed-disk drive.

expanded memory
Certain expansion boards can provide additional memory to a personal computer. Expanded memory is distinguished from conventional memory in that it cannot be addressed directly by DOS but must be accessed through the expanded memory manager. It is used directly by application programs such as Symphony and Framework. Contrast with extended memory.

expansion slot
A series of connectors mounted on the system board into which expansion boards can be inserted. Depending upon the system model, the type and number of expansion slots will vary.

extended memory
Memory whose addresses start at 1M and which can be accessed only when the processor is running in protected mode. Extended memory can be used by operating systems, such as UNIX, that run in protected mode, and by certain DOS programs, such as the RAMDRIVE.SYS virtual disk, that switch in and out of protected mode to perform special operations. Ordinary DOS applications cannot directly access extended memory.

F

Fahrenheit (F)
A temperature scale. Contrast with Celsius (C).

falling edge
Synonym for negative-going edge.

fast paged mode

A dynamic memory mode of operation that allows successive addresses to the same DRAM page (defined by the row addresses strobed (RAS) into the devices by simply changing the column addresses. In this mode, RAS is held active and the various accesses are initiated by strobing the new column addresses with the falling edge of CAS. The mode is available in CMOS parts and allows fewer cycle times.

FCC

Federal Communications Commission.

field

1. In a record, a specified area used for a particular category of data.
2. In a data base, the smallest unit of data that can be referred to by name.

firmware

1. Instructions or programs stored permanently in read-only memory (ROM) and unchangeable.
2. Internal connections that permanently determine the function of a device or system.

fixed disk

A nonflexible, flat, circular plate with a magnetizable surface layer on which data can be stored by magnetic recording. Synonymous with hard disk. Fixed disks are permanently mounted within a fixed-disk drive.

fixed disk drive

A computer unit that consists of nonremovable magnetic disks, and a device for storing and retrieving data from the disks. Synonymous with hard disk drive.

flexible disk

See diskette.

floppy disk

Synonym for flexible disk. See diskette.

floppy disk drive

A computer unit that can store and retrieve data from floppy disks. See diskette.

function keys

Keys that request actions but do not display or print characters. Included are the keys that normally produce a printed character, but when used with the code key produce a function instead. Compare with character key.

G

G

A symbol used to represent the prefix giga. When describing computer storage capacity, common usage has made G synonymous with GB, G-byte or gigabyte.

GB
Abbreviation for gigabyte.

giga
A prefix normally used to indicate a quantity of 1,000,000,000. However, when referring to computer storage capacity, the prefix giga represents a quantity of 1,073,741,824 or 2 raised to the 30th power.

gigabyte
A term used when referring to computer storage capacity. A gigabyte is defined as 1,073,741,824 bytes.

gram (g)
A unit of weight equivalent to 0.035 ounces.

graphics
A type of data created from fundamental drawing units such as lines, splines, curves, polygons, and so forth.

H

hard disk
(See fixed disk.)

hardware

1. Physical equipment used in data processing, as opposed to programs, procedures, rules, and associated documentation.
2. Contrast with software.

head
A device that reads, writes, or erases data on a storage medium; for example, a small electromagnet used to read, write, or erase data on a magnetic disk.

header
A connector located on the system board. Usually consisting of one or more rows of evenly-spaced pins.

Hertz (Hz)
A unit of frequency equal to one cycle per second.

hex
Common abbreviation for hexadecimal.

hexadecimal

1. Pertaining to a selection, choice, or condition that has 16 possible different values or states. These values or states are usually symbolized by the ten digits 0 through 9, and the six letters A through F.
2. Pertaining to a fixed radix numeration system having a radix of 16.

Hz

See Hertz.

I

icon

Icon is a term used to describe graphic display symbols commonly used on video displays. A small symbol that can be easily identified with a device or function; e.g., a graphics symbol of a printer or keyboard. Selecting the icon will allow you to access the device or function it represents.

IEC 435

TUV certified in accordance with International Electrotechnical Commissions Standard for Safety of Information Technology Equipment.

IEC 950

TUV certified in accordance with International Electrotechnical Commissions Standard for Safety of Information Technology Equipment including Electrical Business Equipment.

input/output (I/O)

1. Pertaining to a device or to a channel that may be involved in an input process and, at a different time, in an output process. Input/output may be used in place of input/output data, input/output signal, and input/output terminals, when such usage is clear in a given context.
2. Pertaining to a device whose parts can be performing an input process and an output process at the same time.
3. Pertaining to either input or output, or both.

instruction

A statement that specifies an operation to be performed by the computer, along with the values or locations of operands, if any exist. This statement represents the programmer's request to the processor to perform a specific operation.

instruction set

The set of instructions of a computer, of a programming language, or of the programming languages in a programming system.

interface

A device that alters or converts actual electrical signals between distinct devices, programs, or systems.

interleave

To arrange parts of one sequence of things or events so that they alternate with parts of one or more other sequences of the same nature and so that each sequence retains its identity.

interrupt

1. A suspension of a process, such as the execution of a computer program, caused by an event external to that process and performed in such a way that the process can be resumed.
2. In a data transmission, to take an action at a receiving station that causes the transmitting station to terminate a transmission.
3. Synonymous with interruption.

I/O

Input/Output.

K

K

A symbol used to represent kilobyte, a computer storage quantity representing 1024 bytes, wherein 1024 is equal to 2 raised to the 10th power. Common usage has made it synonymous with KB, Kbyte or kilobyte. See kilobyte. Contrast with k.

k

A symbol used to represent the prefix kilo; 1,000.

KB

Abbreviation for kilobyte.

Kb

A symbol used to represent kilobit, a computer storage quantity representing 1024 bits, wherein 1024 is equal to 2 raised to the 10th power.

keylock

A device that can deactivate a keyboard (if implemented) and locks the cover on for security.

kilo

A prefix used to indicate a quantity of 1000. Abbreviation symbol k. Contrast with K.

kilobyte

A term used when referring to computer storage capacity. A kilobyte is defined as 1024 bytes. *Note that in all other usages, the prefix kilo (k) indicates a quantity of 1,000.*

kilogram (kg)

1000 grams.

kilohertz (kHz)

1000 Hertz.

L

leading edge
The first occurring edge of a pulse.

LIM
Lotus/Intel/Microsoft Expanded Memory Manager specification.

LED
Light-emitting diode.

LPT1, LPT2, LPT3
These are the names DOS assigns to the parallel printer ports in a system. The three names reflect the fact that DOS permits as many as three parallel printer ports in a system.

M

M
A symbol used to represent the prefix mega. When describing computer storage capacity, common usage has made M synonymous with MB, Mbyte or megabyte. See mega.

m

1. Prefix milli; 0.001.
2. Meter.

mA
Milliampere; 0.001 ampere.

machine language
A language that can be used directly by a computer without intermediate processing.

magnetic disk
A flat circular plate with a magnetizable surface layer on which data can be stored by magnetic recording.

main storage

1. Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing.
2. Contrast with mass storage.

mapping
Pertaining to the geographic location for a resource within the address space.

masked
Synonym for disabled.

mass storage
Auxiliary storage in a computer system as differentiated from RAM. Mass storage most commonly refers to floppy and fixed disks and magnetic tape.

MB
Abbreviation for megabyte.

Mb
A symbol used to represent megabit, a computer storage quantity representing 1,048,576 bits, wherein 1,048,576 is equal to 2 raised to the 20th power.

mega
A prefix normally used to indicate a quantity of 1,000,000. However, when referring to computer storage capacity, the prefix mega represents a quantity of 1,048,576 or 2 raised to the 20th power.

megabyte
A term used when referring to computer storage capacity. A megabyte is defined as 1,048,576 bytes.

memory
Storage on electric memory such as random access memory (RAM), read-only memory (ROM), or CPU registers.

MFM
Modified frequency modulation.

MHz
Megahertz; 1,000,000 Hertz.

micro (μ)
Prefix 0.000 001.

microprocessor
An integrated circuit that accepts coded instructions for execution; the instructions may be entered, integrated, or stored internally.

microsecond (μs)
0.000 001 second.

modified frequency modulation (MFM)
The process of varying the amplitude and frequency of the write signal. MFM pertains to the number of bytes of storage that can be stored on the recording media. The number of bytes is twice the number contained in the same unit area of recording media at single density. MFM recording is commonly used on floppy- and fixed-disk drives.

monitor

1. A device for visual presentation of information as temporary images. A video display.
2. Synonym for cathode ray tube display (CRT display).

N

nanosecond (ns)
0.000 000 001 second.

negative-going edge
The edge of a pulse or signal changing in a negative direction. Synonymous with falling edge.

negative true
Synonym for active low.

network
A group of computers connected and configured such that they can share resources.

nonrecoverable error
An error that makes recovery impossible without the use of recovery techniques external to the computer program run.

ns
nanosecond; 0.000 000 001 second.

O

OEM
Original Equipment Manufacturer.

offline
Pertaining to the operation of a functional unit without the continual control of a computer.

online
Pertaining to the operation of a functional unit under the continual control of a computer.

operating system
Software that controls the execution of programs; an operating system may provide services such as resource allocation, scheduling, input/output control, and data management.

P

paged mode
The same basic functionality as fast paged-mode, except that the access time is the same as a normal RAS/CAS access. This mode is a feature on NMOS-type DRAM parts.

PAL
Program array logic

parallel

1. Pertaining to the concurrent or simultaneous operation of two or more devices, or to the concurrent performance of two or more activities.
2. Pertaining to the concurrent or simultaneous occurrence of two or more related activities in multiple devices or channels.
3. Pertaining to the simultaneity of two or more processes.
4. Pertaining to the simultaneous processing of the individual parts of a whole, such as the bits of a character and the characters of a word, using separate facilities for the various parts.
5. An alternative to serial.

parameter

1. A variable that is given a constant value for a specified application and that may denote the application.
2. A name in a procedure that is used to refer to an argument passed to that procedure.

pel
Picture element.

picture element (pel)
In computer graphics, the smallest element of a display space that can be independently assigned color and intensity. Synonymous with pixel.

pixel
The smallest displayable unit on a monitor or picture tube element. Synonymous with pel.

platform system
A basic OEM product-line system which combines computers or computer subsystems with special, unique, and proprietary hardware and/or software for added value.

port
An access point for data entry or exit.

positive-going edge
The edge of a pulse or signal changing in a positive direction. Synonymous with rising edge.

positive true
Synonym for active high.

POST
Acronym for power-on self test.

power-on self test
A series of diagnostic tests that are run each time the computer's power is turned on. See POST.

power supply
A device that produces the power needed to operate electronic equipment.

precision
In science, a measure of the ability to differentiate quantities; the degree of agreement of repeated measurements of a quantity. Not to be confused with accuracy.

printed circuit
A pattern of conductors (corresponding to the wiring of an electronic circuit) formed on a board of insulating material.

printed-circuit board
Usually a copper-clad fiberglass board used to make a printed circuit. Also, refers to a board on which a printed circuit has been made.

processing unit
A functional unit that consists of one or more processors and all or part of internal memory.

processor

1. In a computer, a functional unit that interprets and executes instructions.
2. A functional unit, a part of another unit such as a terminal or processing unit, that interprets and executes instructions. (see microprocessor)

program
A file containing a set of instructions conforming to a particular programming language syntax.

PROM
Programmable read-only memory. A type of ROM that contains a programmed set of code. A PROM code cannot be changed once programmed. See also EPROM, E^{2}PROM and ROM.

protected mode
A mode of the 80386 microprocessor enabling it to provide advanced features, such as accessing large amounts of memory and enforcing hardware protection of memory segments. Current versions of DOS do not support protected mode operation, except for special utilities such as the RAMDRIVE.SYS virtual disk.

R

RAM
Random access memory. Read/write memory.

RAS

Row address strobe; a technique used in dynamic RAM addressing.

RAS/CAS

A mode of DRAM operation where every access is begun by strobing the row addresses with RAS and column addresses with CAS.

raster

In computer graphics, a predetermined pattern of lines that provides uniform coverage of a display space.

read

To acquire or interpret data from a storage device, from a data medium, or from another source.

read-only memory (ROM)

A storage device whose contents cannot be modified. The memory is retained when power is removed.

recoverable error

An error condition that allows continued execution of a program.

register

1. A storage device, having a specified storage capacity such as a bit, a byte, or a computer word, and usually intended for a special purpose.
2. A storage device in which specific data is stored.

reverse video

A form of highlighting a character, field, or cursor by reversing the color of the character, field, or cursor with its background; for example, changing a red character on a black background to a black character on a red background.

RFI

Radio frequency interference.

ROM

Read-only memory. See also PROM, EPROM, and E^{2}PROM.

ROM BIOS

The ROM resident basic input/output (BIOS) system which controls the major I/O devices in a computer system.

RS-232C

A standard by the Electronics Industries Association (EIA) for serial communication between computers and external equipment.

S

scratch disk

A scratch disk is usually a formatted floppy disk that can be used for test purposes. A floppy disk that contains no information of value. If data on a scratch disk is lost or destroyed during tests, it is of no consequence.

sector

That part of a track or band on a magnetic drum, a magnetic disk, or a disk pack that can be accessed by the magnetic heads in the course of a predetermined rotational displacement of the particular device.

serial

1. Pertaining to the sequential performance of two or more activities in a single device. The modifiers serial and parallel usually refer to devices, as opposed to sequential and consecutive, which refer to processes.
2. Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel.
3. Pertaining to the sequential processing of the individual parts of a whole, using the same facilities for successive parts.
4. An alternative to parallel.

setup

1. In a computer that consists of an assembly of individual computing units, the arrangement of interconnections between the units, and the adjustments needed for the computer to operate.
2. The preparation of the system for normal operation.

shadow memory

A portion of RAM to which selected BIOS information is copied from ROM. This ROM-to-RAM copying technique is referred to as shadowing. Shadow memory is write-protected and often has the same addresses as the original ROM locations. Shadowing greatly enhances system performance because ROM information is available from fast 32-bit RAM chips instead of the slower ROM chips.

signal

A variation of physical quantity, used to convey data.

SIMM

Single in-line memory module. A small plug-in board containing nine DRAM chips. A SIMM DRAM chip is organized in a specific configuration; i.e. 256Kb × 1 or 1Mb × 1 organization. For example, eight 256Kb × 1 DRAM devices combine to form 256K of memory. The ninth device provides parity checking.

single precision

Pertaining to the use of one computer word to represent a number in accordance with the required precision. Contrast with double precision and precision.

software

1. Computer programs, procedures, and rules concerned with the operation of a data processing system.
2. Contrast with hardware.

SRAM

Static RAM. RAM comprised of static RAM chips. Unlike DRAMs, SRAMs require no refresh and are faster devices. The SRAM cell is larger than the DRAM cell and for this reason, SRAMs are lower density devices.

static column

A mode of operation of DRAM operation that allows successive accesses to the same DRAM page (defined by the row addresses initially strobed into the devices) by simply changing the column addresses. This mode differs from fast paged-mode because both RAS and CAS are held active, whereas CAS is strobed in fast paged-mode. The access time is limited to the address access time of the part as new column addresses are presented.

static memory

RAM using flip-flops as the memory elements. Data is retained as long as power is applied to the flip-flops. Contrast with dynamic memory.

storage

1. A storage device.
2. A device, or part of a device that can retain data.
3. The retention of data in a storage device.
4. The placement of data into a storage device.

synchronization

The process of adjusting the corresponding significant instants of two signals to obtain the desired phase relationship between these instants.

synchronous

1. Data transmission in which the time of transmission occurrence of each signal representing a bit is related to a fixed time frame.
2. Data transmission in which the sending and receiving devices are operating continuously at substantially the same frequency and are maintained in a desired phase relationship by means of correction.

Contrast with asynchronous.

T

T

A symbol used to represent the prefix tera. When describing computer storage capacity, common usage has made T synonymous with TB, Tbyte or terabyte.

TB

Abbreviation for terabyte.

tera

A prefix normally used to indicate a quantity of 1,000,000,000,000. However, when referring to computer storage capacity, the prefix tera represents a quantity of 1,099,571,627,300 or 2 raised to the 40th power.

terabyte

A term used when referring to computer storage capacity. A terabyte is defined as 1,099,571,627,300 bytes.

tpi

Tracks per inch. A specification used in formatting floppy disks and fixed disks.

TTL

Transistor-transistor logic. A popular logic circuit family that uses multiple-emitter transistors.

track

1. The path or one of the set of paths, parallel to the reference edge on a data medium, associated with a single reading or writing component as the data medium moves past the component.
2. The portion of a moving data medium such as a drum, or disk, that is accessible to a given reading head position.

trailing edge

The second edge of a pulse.

TUV

Technischer Ueberwachungs-Verein. TUV is an testing organization that evaluates and certifies electronic data processing equipment to specific International safety standards.

typematic key

A key that repeats its function multiple times when held down.

U

UL 478

Underwriter Laboratories Standard for Safety of Information Processing and Business Equipment.

V

V

Volt.

VAC

Volts (alternating current).

VDE 0806/IEC 380

TUV certified in accordance with International Electrotechnical Commissions Standard for Safety of Electrical Energized Office Machines.

VDE 0871

Verband Deutscher Electrotechnikes Specification for EMI Suppression.

VDC

Volts (direct current).

video

Computer data or graphics displayed on a CRT, monitor, or display.

video adapter

A special board that provides a suitable interface between a computer and a video display device such as a CRT or monitor. A video controller.

video controller

A special board that provides a suitable interface between a computer and a video display device such as a CRT or monitor. A video adapter.

video display

A device for visual presentation of information as temporary images. A monitor. See also CRT.

virtual address

A 32-bit address on the internal bus intended to be translated by memory management.

volt

The basic unit of electric pressure. The potential that causes electrons to flow through a circuit.

W

W
Watt.

Watt (W)
The basic unit of electric power.

word
A character string or bit string considered as an entity in computer architecture.

write
To make a permanent or transient recording of data in a storage device or on data medium.

write precompensation
The varying of the timing of the head current from the outer tracks to the inner tracks of the diskette to keep a constant write signal.

Index

INTEL CORPORATION
3065 Bowers Avenue
Santa Clara, CA 95052-8130