Personal System/2™ Model 30 Technical Reference

80X0661

Personal System/2™ Model 30 Technical Reference

The following statement applies to this IBM product. The statement for other IBM products intended for use with this product will appear in their accompanying materials.

Federal Communications Commission (FCC) Statement

Warning: This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of 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 when this computer is operated in a residential environment. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

CAUTION
This product is equipped with a 3-wire power cord and plug for the user's safety. Use this power cord in conjunction with a properly grounded electrical outlet to avoid electrical shock.

First Edition (January 1987)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

Personal System/2 is a trademark of the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1987. All rights reserved.

Preface

This publication describes the components of the IBM Personal System/2 Model 30 and their interaction.

The information in this publication is for reference and is intended for hardware and software designers, programmers, engineers, and anyone else with a knowledge of electronics or programming who need to understand the design and operation of the Model 30.

This manual is divided into the following sections:

Section 1. "System Board" discusses the functions of the system board.

Section 2. "Coprocessor" describes the 8087 Math Co-processor and provides programming and hardware interface information.

Section 3. "Power Supply" provides electrical input/output specifications as well as a theory of operation for the power supply.

Section 4. "Keyboard" discusses the hardware, function, encoding, and layouts of the 101/102-key keyboards.

Section 5. "System BIOS" describes the basic input/output system and the interrupt interfaces. This section also contains a BIOS memory map, descriptions of vectors with special meanings, and a set of low memory maps.

Section 6. "Instruction Set" provides a quick reference for the 8086 and 8087 assembly instruction set.

Section 7. "Characters and Keystrokes" supplies the decimal and hexadecimal values for characters.

A Glossary, Bibliography, and Index are also provided.

Prerequisite Publications

- IBM Personal System/2 Model 30 *Guide to Operations*

Suggested Related Publications

- *BASIC for the IBM Personal Computer*
- *Disk Operating System (DOS)*
- *Hardware Maintenance Service* manual
- *Hardware Maintenance Reference* manual
- *Macro Assembler for the IBM Personal Computer.*

Additional Information

- Additional technical information for the IBM Personal System/2 is available from the *Technical Directory*. To receive a free copy of the *Technical Directory*, call toll free 1-800-IBM-PCTB, Monday through Friday, 8:00 a.m. to 8:00 p.m., Eastern Time.

Contents

SECTION 1. System Board **1-1**
General Description 1-3
Microprocessor 1-5
System Support Gate Array 1-6
I/O Support Gate Array 1-8
DMA Controller 1-13
Interrupts 1-13
Interrupt Sharing 1-14
Read/Write Memory 1-22
ROM 1-22
I/O Channel 1-23
Connectors 1-25
Signal Description 1-26
Signal Timings 1-28
Video Subsystem 1-36
Block Diagram 1-37
Display Support 1-38
Text Modes 1-38
Graphic Modes 1-38
Display Formats 1-40
Video Storage Organization 1-42
Video Registers 1-44
Video Initialization Tables 1-63
RAM Loadable Fonts 1-65
Programming Considerations 1-72
Connector 1-78
Diskette Drive Interface 1-79
Gate Array Registers 1-80
Controller Registers 1-82
Commands 1-84
Signal Description 1-103
Connector 1-105
Fixed Disk Connector 1-106
Serial Port 1-108
Application 1-109
Controller Registers 1-110
Programmable Baud-Rate Generator 1-119
Signal Descriptions 1-120

Connector 1-122
Parallel Port 1-123
Port Registers 1-123
Connector 1-126
Beeper 1-127
Connectors 1-128
Specifications 1-130

SECTION 2. Coprocessor 2-1
Description 2-3
Programming Considerations 2-3
Hardware Interface 2-4

SECTION 3. Power Supply 3-1
Description 3-3
Input and Output Power 3-4
Overvoltage/Overcurrent Protection 3-5
Power-Good Signal 3-5
Connectors 3-6

SECTION 4. Keyboard 4-1
Description 4-3
Power-On Routine 4-4
Clock and Data Signals 4-5
Commands 4-7
Scan Codes 4-8
Encoding 4-11
Layouts 4-23
Cables and Connectors 4-41
Specifications 4-42

SECTION 5. System BIOS 5-1
System BIOS Usage 5-3
Hardware Interrupts 5-4
Software Interrupts 5-5
Interrupt Interface Listing 5-11
Interrupt 02H - Non-Maskable Interrupt Routine 5-11
Interrupt 05H - Print Screen 5-11
Interrupt 08H - System Timer 5-12
Interrupt 09H - Keyboard 5-13
Interrupt 10H - Video 5-14
Interrupt 11H - Equipment Determination 5-29
Interrupt 12H - Memory Size Determine 5-30

Interrupt 13H - Diskette 5-31
Interrupt 13H - Fixed Disk 5-38
Interrupt 14H - Asynchronous Communications 5-46
Interrupt 15H - System Services 5-51
Interrupt 16H - Keyboard 5-58
Interrupt 17H - Printer 5-62
Interrupt 19H - Bootstrap Loader 5-63
Interrupt 1AH - System and Real-Time Clock Services 5-64
BIOS Data Area and Locations 5-67
Extended BIOS Data Area 5-72
ROM Tables 5-73
Fixed Disk Parameter Table 5-73
Asynchronous Baud Rate Initialization Table 5-75
Diskette Parameter Table 5-75
Model Byte 5-76

SECTION 6. Instruction Set 6-1
8086 Register Model 6-2
Notes 6-3
8086 Instruction Set 6-7
Data Transfer 6-7
Arithmetic 6-9
Logic 6-13
String Manipulation 6-15
Control Transfer 6-15
Processor Control 6-20
Instruction Set Matrix 6-22
8087 Coprocessor Instruction Set 6-24
Notes 6-24
Data Transfer 6-24
Comparison 6-26
Arithmetic 6-26
Transcendental 6-28
Constants 6-29
Processor Control 6-29

SECTION 7. Characters and Keystrokes 7-1
Character Codes 7-2
Table Notes 7-7
Quick Reference 7-8

Glossary X-1

Bibliography **X-19**

Index **X-21**

Figures

1-1. System Functional Diagram 1-4
1-2. Memory Map 1-5
1-3. Planar Control Register, Hex 65 1-8
1-4. System Timer Block Diagram 1-9
1-5. Output Port, Hex 61 1-11
1-6. Input Port, Hex 62 1-12
1-7. DMA Channel Assignments 1-13
1-8. DMA Page Register Addresses 1-13
1-9. Hardware Interrupt Listing 1-14
1-10. Shared Interrupt Hardware Logic 1-15
1-11. Planar RAM Control/Status Register 1-22
1-12. I/O Address Map 1-24
1-13. I/O Channel 1-25
1-14. 8-Bit I/O Timing 1-29
1-15. 8-Bit Memory Timing 1-30
1-16. 16-Bit I/O Timing 1-31
1-17. 16-Bit Memory Timing 1-32
1-18. Memory Refresh Timing 1-33
1-19. DMA Read Timing 1-34
1-20. DMA Write Timing 1-35
1-21. Video Subsystem Block Diagram 1-37
1-22. Video Mode Summary 1-39
1-23. Alphanumeric Format 1-40
1-24. Attribute Byte 1-40
1-25. Modes 4 and 5 1-41
1-26. Modes 6 and 11 1-41
1-27. Text Modes 0 to 3 1-42
1-28. Graphic Modes 4 to 6 1-43
1-29. Graphic Modes 11 and 13 1-43
1-30. Memory Controller Index Register 1-45
1-31. Sync Pulse Width Register 1-47
1-32. Vertical Total Adjust Register 1-48
1-33. Scan Lines per Character Register 1-48
1-34. Cursor Start Register 1-49
1-35. Cursor End Register 1-49
1-36. Cursor Position High Register 1-50
1-37. Mode Control, Write 1-50
1-38. Mode Control, Read 1-51

1-39. Interrupt Control Register 1-52
1-40. Character Generator Interface and Sync Polarity Register 1-52
1-41. Monitor Sense Bits 1-53
1-42. CGA Mode Register 1-55
1-43. CGA Border Control Register 1-56
1-44. Modes 4 and 5 Color Selection 1-56
1-45. Status Register 1-57
1-46. Extended Mode Control Register 1-57
1-47. Last Palette Command 1-61
1-48. Palette Data Register 1-62
1-49. Memory Controller Initialization 1-63
1-50. Video Formatter Initialization Table 1-64
1-51. 16-Color Compatibility Initialization 1-64
1-52. Font Memory Map 1-65
1-53. Sample Character 1-66
1-54. Block Specifier 1-69
1-55. Alternate Parameter Table 1-71
1-56. Write to Palette Address Register 1-73
1-57. Read Palette Address Register 1-73
1-58. Write Color followed by a Read 1-74
1-59. Write Color followed by a Write 1-75
1-60. Read Color followed by Read 1-76
1-61. Read Color followed by Write 1-77
1-62. Display Connector 1-78
1-63. RAS Port A, Hex 3F0 1-80
1-64. RAS Port B, Hex 3F1 1-80
1-65. Digital Output, Hex 3F2 1-81
1-66. Digital Input, Hex 3F7 1-81
1-67. Configuration Control, Hex 3F7 1-82
1-68. Main Status Register 1-83
1-69. Read Data Command 1-87
1-70. Read Data Result 1-87
1-71. Read Deleted Data Command 1-88
1-72. Read Deleted Data Result 1-88
1-73. Read a Track Command 1-89
1-74. Read a Track Result 1-89
1-75. Read ID Command 1-90
1-76. Read ID Result 1-90
1-77. Write Data Command 1-91
1-78. Write Data Result 1-91
1-79. Write Deleted Data Command 1-92
1-80. Write Deleted Data Result 1-92

1-81. Format a Track Command 1-93
1-82. Format a Track Result 1-93
1-83. Scan Equal Command 1-94
1-84. Scan Equal Result 1-94
1-85. Scan Low or Equal Command 1-95
1-86. Scan Low or Equal Result 1-95
1-87. Scan High or Equal Command 1-96
1-88. Scan High or Equal Result 1-96
1-89. Recalibrate Command 1-97
1-90. Sense Interrupt Status Command 1-97
1-91. Sense Interrupt Status Result 1-97
1-92. Specify Command 1-97
1-93. Sense Driver Status Command 1-98
1-94. Sense Driver Status Result 1-98
1-95. Seek Command 1-98
1-96. Invalid Command Result 1-99
1-97. Diskette Drive Connector 1-105
1-98. Fixed Disk Signal Timing 1-106
1-99. Fixed Disk Connector 1-107
1-100. Serial Port Block Diagram 1-108
1-101. Serial Port Addresses 1-110
1-102. Transmitter Holding Register 1-110
1-103. Receiver Buffer Register 1-111
1-104. Divisor Latch 1-111
1-105. Interrupt Enable Register 1-112
1-106. Interrupt Identification Register 1-112
1-107. Line Control Register 1-114
1-108. Modem Control Register 1-115
1-109. Line Status Register 1-116
1-110. Modem Status Register 1-118
1-111. Serial Port Connector 1-122
1-112. Serial Interface Specifications 1-122
1-113. Parallel Port Block Diagram 1-123
1-114. Printer Control Register 1-124
1-115. Printer Status Register 1-125
1-116. Parallel Port Signal Timing 1-126
1-117. Parallel Port Connector 1-126
1-118. Beeper Tone Generation 1-127
1-119. System Board Connector Location 1-128
1-120. Power Supply Connectors 1-129
1-121. Keyboard Connector and Pointing Device 1-129
2-1. Coprocessor Data Types 2-4
2-2. Coprocessor Interconnection 2-5

3-1. Vac Input Requirements 3-4
3-2. Vdc Output 3-4
3-3. Output Protection 3-5
3-4. Power Supply Connectors 3-6
4-1. Keyboard Data Stream 4-6
4-2. Commands from the Keyboard 4-7
4-3. Scan Codes (Part 1 of 4) 4-9
4-4. Scan Codes (Part 2 of 4) 4-10
4-5. Scan Codes (Part 3 of 4) 4-10
4-6. Scan Codes (Part 4 of 4) 4-10
4-7. 101-Key Keyboard Layout 4-12
4-8. 102-Key Keyboard Layout 4-13
4-9. Character Codes 4-14
4-10. Special Character Codes 4-17
4-11. Keyboard Extended Functions 4-18
4-12. Keyboard Cable Connectors 4-41
5-1. Software Interrupt Listing 5-5
5-2. BASIC and DOS Interrupts 5-8
5-3. Reserved Memory Locations 5-8
5-4. BASIC Workspace Variables 5-9
5-5. BIOS Memory Map 5-9
6-1. 8086 Register Model 6-2
6-2. Flag Register 6-3
6-3. Segment Override Prefix 6-4
6-4. reg Field Assignment 6-4
6-5. mod Field Assignment 6-5
6-6. r/m Field Assignments 6-5
6-7. Conditional Transfer Operations 6-19

SECTION 1. System Board

General Description 1-3
Microprocessor 1-5
System Support Gate Array 1-6
Bus Controller 1-6
Memory Controller and Parity Checker 1-7
Bus Conversion Logic and Wait-State Generator 1-7
System Clock Generator 1-7
I/O Support Gate Array 1-8
Chip Select Logic 1-8
Keyboard and Pointing Device Controller 1-9
System Timer 1-9
I/O Ports 1-11
DMA Controller 1-13
Interrupts 1-13
Interrupt Sharing 1-14
Design Overview 1-14
Program Support 1-15
Precautions 1-17
ROM Considerations 1-18
Examples 1-18
Read/Write Memory 1-22
ROM 1-22
I/O Channel 1-23
Connectors 1-25
Signal Description 1-26
Signal Timings 1-28
Video Subsystem 1-36
Block Diagram 1-37
Display Support 1-38
Text Modes 1-38
Graphic Modes 1-38
Display Formats 1-40
Video Storage Organization 1-42
Video Registers 1-44
Memory Controller Registers 1-45
Video Formatter Registers 1-54
Color Palette Registers 1-58
Video Initialization Tables 1-63

RAM Loadable Fonts 1-65
Alternate Parameter Table 1-71
Programming Considerations 1-72
Connector 1-78
Diskette Drive Interface 1-79
Gate Array Registers 1-80
Controller Registers 1-82
Commands 1-84
Command Format 1-87
Command Status Registers 1-100
Signal Description 1-103
Connector 1-105
Fixed Disk Connector 1-106
Serial Port 1-108
Application 1-109
Controller Registers 1-110
Programmable Baud-Rate Generator 1-119
Signal Descriptions 1-120
Input Signals 1-120
Output Signals 1-121
Connector 1-122
Parallel Port 1-123
Port Registers 1-123
Connector 1-126
Beeper 1-127
Connectors 1-128
Specifications 1-130
Size 1-130
Weight 1-130
Power Cable 1-130
Environment 1-130
Heat Output 1-131
Noise Level 1-131
Electrical 1-131

General Description

The IBM Personal System/2™ Model 30 is a highly integrated system using five gate arrays: two for microprocessor support, two for video support, and one for the diskette controller support. The major functional areas are:

- 8086-2 microprocessor and its support logic
- 640K read/write (R/W) memory
- 64K ROM
- I/O channel
- Integrated I/O functions
  - Color/graphic subsystem
  - Diskette drive interface
  - Fixed disk connector
  - Serial port
  - Parallel port
  - Real-time clock with battery

DC power and a 'power good' signal from the power supply enter the system board through two 6-pin connectors. Other connectors on the system board are for attaching the keyboard, pointing device, coprocessor, display, serial and parallel devices, and storage media.

Three 62-pin card-edge sockets are attached to a vertical expansion bus that is mounted to the system board. The input/output (I/O) channel is extended to these three I/O slots.

Personal System/2 is a trademark of the International Business Machines Corporation.

Functional Diagram

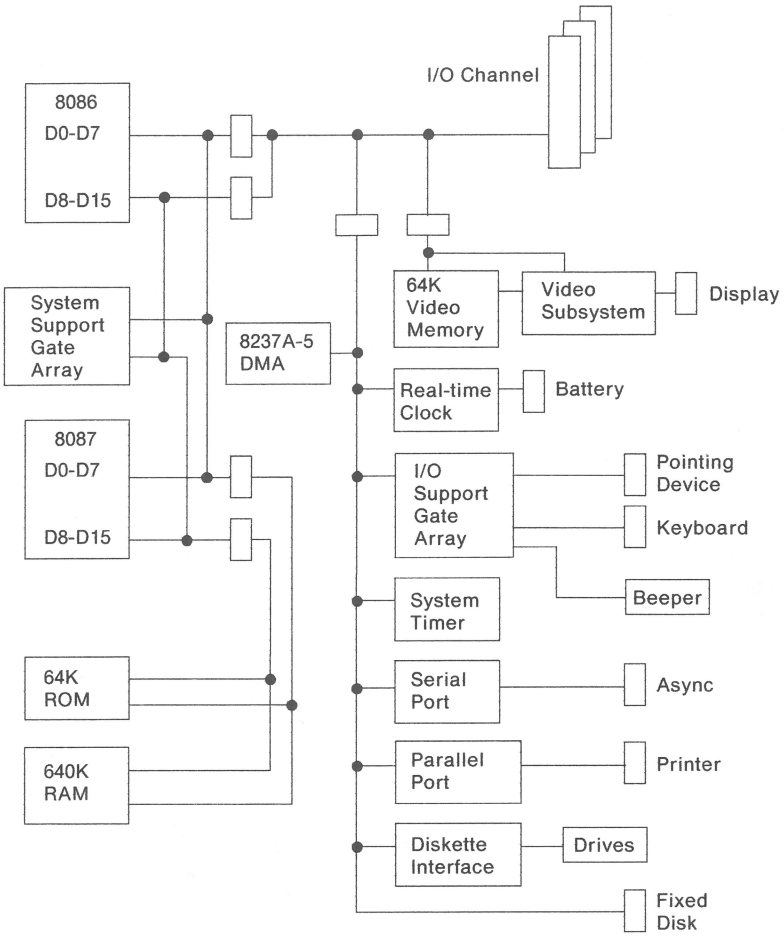


Figure 1-1. System Functional Diagram

Microprocessor

The Intel 8086-2 is a 16-bit microprocessor with a 16-bit external data bus, operating at 8 MHz. The microprocessor supports the same 20-bit addressing as IBM Personal Computers that use the 8088 microprocessor. The Model 30 uses a 16-bit data bus with the system's read-only and read/write memory, and an 8-bit bus for all I/O and direct memory access (DMA) operations.

The memory read and write are 16-bit operations, and take four clock cycles of 125 ns, with no wait states, for a cycle time of 500 ns. Normal I/O operations are 8-bit operations, and take eight clock cycles, including four wait states, for a cycle time of 1 μs. A signal on the I/O channel, IO CH RDY, allows slower devices to add more wait states to I/O and DMA operations (see "I/O Channel" later in this section).

Logic has been added to the system board to support options for the IBM Personal Computer family. This includes converting 16-bit operations to sequential 8-bit operations, inserting wait states into all I/O and DMA operations, and delaying microprocessor cycles to ensure address setup times greater than or equal to the 8088-based systems.

The 8086 supports 16-bit operations, including multiply and divide, and 20-bit addressing to access 1M (M = 1 048 576). It also operates in maximum mode, so a math coprocessor can be added as a feature. Memory is mapped as follows:

Hex Address	Function
00000 - 9FFFF	640K Read/Write Memory
A0000 - BFFFF	Video Buffer
C0000 - EFFFF	Reserved for BIOS on I/O Channel
F0000 - FFFFF	System ROM

Figure 1-2. Memory Map

The microprocessor is supported by two high-function support devices: a system support gate array and an I/O support gate array.

System Support Gate Array

The system support gate array contains the bus controller, the memory controller and parity checker, the wait-state generator and bus conversion logic, the system clock generator, and the DMA page register and support logic.

Bus Controller

The Model 30 has three bus masters on the local bus: the microprocessor, the coprocessor, and the system support gate array. The gate array seizes the bus to generate memory refresh and DMA bus cycles. It controls two request/grant lines (CPU RQ/GT and NPU RQ/GT). One is connected to the microprocessor and the other to the coprocessor.

When the coprocessor is not installed, the gate array generates a request pulse to the microprocessor to get control of the bus. The microprocessor then gives control of the bus and pulses the same line to indicate a grant. When the coprocessor is installed, the gate array generates this pulse to the coprocessor. If the coprocessor has control of the bus, it grants control to the gate array. If the coprocessor is not in control, it relays the request to the microprocessor and relays the grant back to the gate array. This arrangement gives the gate array the highest priority use of the bus.

Warning: If you are using an in-circuit 8086 emulator, care must be taken that the request/grant pulses do not get out of synchronization. Damage to the gate array will occur.

Memory Controller and Parity Checker

The memory controller functions of the gate array control memory and generate the memory refresh. Memory must be refreshed once every 4 ms. Memory refresh takes nine clock cycles of 125 ns through a dedicated refresh channel within the gate array.

The parity checker function generates the parity bits for system memory and activates the '-parity check' signal when a parity error is detected. All 640K of memory on the system board is checked.

Bus Conversion Logic and Wait-State Generator

The bus conversion logic converts word transfers to I/O devices into 2-byte transfers. Sixteen-bit transfers are only supported for the system's read-only and read/write memory.

Additional logic generates the needed wait states for the microprocessor bus cycles to I/O devices. This logic monitors the 'I/O channel ready' line (IO CH RDY) to determine the wait states required.

System Clock Generator

The system clock generator uses a 48 MHz input that is internally divided to give the output clock signal of 8 MHz with a 33% duty cycle. It also generates a 1.84 MHz signal for the serial port.

The clock generator generates the 'reset' signal after sensing the 'power good' signal from the power supply.

I/O Support Gate Array

The I/O support gate array contains the chip select logic, keyboard and pointing device controller, and the I/O ports. It also contains the interrupt controller.

Chip Select Logic

The gate array has control of the following chip select signals on the system board:

- Serial port
- Diskette controller
- Video controller
- Parallel port
- Fixed disk controller
- Real-time clock.

Except for the real-time clock, each select line can be disabled by programming the Planar Control register, address hex 65. When the bit is set to 1, that function of the system board is enabled. Bit 7, parallel port output enable, enables the parallel port's output drivers.

When the signal is enabled, the chip select signal is generated to start a read or write operation, and the read and write signals to the I/O channel are blocked. When the signal is disabled, the chip select signal is not generated and all read and write operations are directed to the I/O channel. This register is read/write.

Bit	Function
7	Parallel Port Output Enable
6	Reserved = 0
5	Reserved = 0
4	Serial CS
3	Diskette CS
2	Video CS
1	Parallel Port CS
0	Fixed Disk CS

Figure 1-3. Planar Control Register, Hex 65

Keyboard and Pointing Device Controller

The interface logic for the keyboard and the pointing device is the same, allowing the keyboard and pointing device to plug into either of the two 6-pin connectors at the rear of the system.

The interface receives the serial data and checks the parity. The data is then presented to the system at the interface's output buffer, I/O port hex 60.

System Timer

The system timer is an 8253 programmable interval timer/counter, or equivalent, that functions as an arrangement of four external I/O ports (hex 0040 through 0043). It receives its 1.19 MHz clock from the I/O support gate array. Three ports are treated as counters; the fourth, address hex 0043, is a control register for mode programming.

The content of a selected counter may be latched without disturbing the counting operation by writing to the control register. However, if the content is latched at the instant the timer is being updated, the value may be incorrect. If the content of the counter is critical to a program's operation, a second read is recommended for verification.

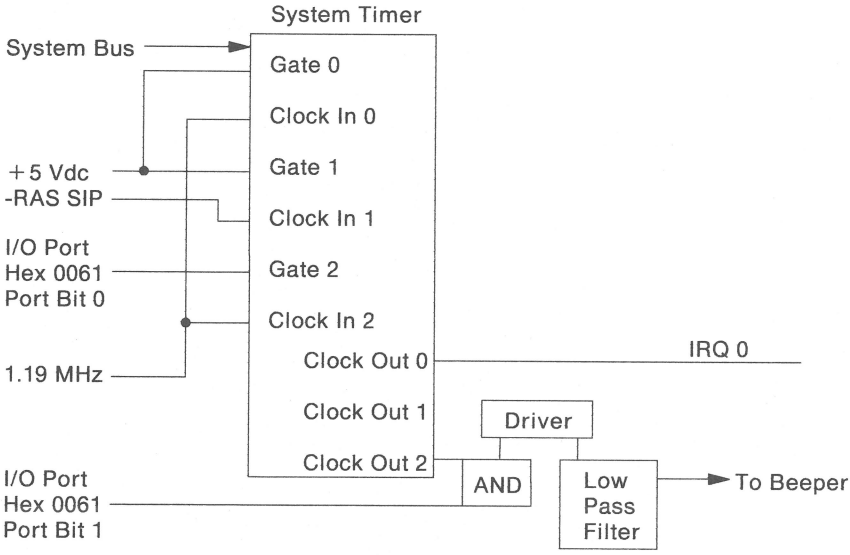


Figure 1-4. System Timer Block Diagram

The three timers are programmable and are used by the system as follows:

Channel 0 is a general-purpose timer providing a constant time base for implementing a time-of-day clock.

Channel 0	**System Timer**
GATE 0	Tied On
CLK IN 0	1.19 MHz OSC
CLK OUT 0	IRQ 0

Channel 1 is for internal diagnostic tests.

Channel 1	**Diagnostic**
GATE 1	Tied On
CLK IN 1	-RAS_SIP from system support gate array
CLK OUT 1	Not connected

Channel 2 supports the tone generation for the audio.

Channel 2	**Tone Generation**
GATE 2	Controlled by bit 0 at port hex 61
CLK IN 2	1.19 MHz OSC
CLK OUT 2	To the beeper data of the I/O support gate array

I/O Ports

The output port hex 60 is used by BIOS to store keystrokes. The output port hex 61 is used as beeper control, enables -IO CH CK and -PARITY, and is read/write. The input port hex 62 contains the status of certain signals on the system board and is read-only. The bit descriptions of ports hex 61 and hex 62 follow:

Bit	Function
7	Reserved
6	Reserved
5	-ENA IO CH CK
4	-ENA RAM Parity CK
3	Reserved
2	Reserved
1	Beeper Data
0	Timer 2 Gate (to beeper)

Figure 1-5. Output Port, Hex 61

Port 61 Bit	Connected to	Description
7 - 6	Not connected	Reserved as 0.
5	-ENA IO CH CK	When set to 1, this bit stops -IO CH CK from generating an NMI. When cleared to 0, an NMI is generated when -IO CH CK goes active.
4	-ENA RAM Parity CK	When set to 1, this bit stops a memory parity error from generating an NMI. When cleared, an NMI is generated when a memory parity error is sensed.
3 - 2	Not connected	Reserved as 0.
1	Beeper Data	This bit gates the output of timer 2. It is used to disable the timer's sound source or modify its output. When set to 1, this bit enables the output; when cleared, it forces the output to zero.
0	Timer 2 Gate	This line is routed to the timer input at GATE 2. When this bit is cleared to 0, the timer operation is halted. This bit and bit 1 (beeper data) control the operation of the timer's sound source.

Bit	Function
7	Parity
6	IO CH CK
5	Timer 2 Out
4	Reserved
3	Reserved
2	-Disk Installed
1	Coprocessor Installed
0	Reserved

Figure 1-6. Input Port, Hex 62

Port 62 Bit	Connected to	Description
7	Parity	When set to 1, this bit indicates that a memory parity error has occurred.
6	IO CH CK	When set to 1, this bit indicates that -IO CH CK is active.
5	Timer 2 Output	This bit shows the status of the timer 2 output.
4 - 3	Not connected	Reserved.
2	-Disk Installed	When cleared to 0, this bit indicates that the fixed disk and controller are installed.
1	Coprocessor Installed	When set to 1, this bit indicates that the math coprocessor is installed.
0	Not connected	Reserved.

DMA Controller

The 8237 DMA controller and its support logic in the gate array support four channels of 20-bit direct memory access (DMA). It operates at 4 MHz and handles only 8-bit transfers. The DMA channel assignments and page register addresses are:

Level	Assignment
DRQ1	Not Used
DRQ2	Diskette
DRQ3	Fixed Disk

Figure 1-7. DMA Channel Assignments

Hex Address	DMA Page Register
080	Channel 2
081	Channel 3
082	Channel 1
087	Channel 0

Figure 1-8. DMA Page Register Addresses

Three of the DMA channels (1, 2, and 3) are available on the I/O bus and support high-speed data transfers between I/O devices and memory without microprocessor intervention.

DMA data transfers take six clock cycles of 250 ns, or 1.5 μs. IO CH RDY can be pulled inactive to add wait states to allow more time for slower devices.

Interrupts

The interrupt controller has eight levels of interrupt that are handled according to priority in the I/O support gate array. Two levels are used only on the system board. Level 0, the highest priority, is attached to Channel 0 of the timer/counter and provides a periodic interrupt for the timer tick. Level 1 is shared by the keyboard, pointing device, and real-time clock; it is handled by a BIOS routine pointed to by interrupt hex 71. Level 2 is available to the video subsystem, and level 7 is available to the parallel port; however, the BIOS routines do not use interrupts 2 and 7.

This controller also has inputs from the coprocessor's '-interrupt', the memory controller's '-parity', and the '-I/O channel check' signals. These three inputs are used to generate the non-maskable interrupt (NMI) to the 8086.

The following table shows the hardware interrupts and their availability to the I/O channel.

Level	System Board	I/O Channel
NMI	Parity Check and Coprocessor	I/O Channel Check
IRQ0	Timer Channel 0	Not Available
IRQ1	Keyboard Pointing Device Real-Time Clock	Not Available
IRQ2	Video	Available
IRQ3	Not Used	Available
IRQ4	Serial Port	Available
IRQ5	Fixed Disk	Available
IRQ6	Diskette Drive	Available
IRQ7	Parallel Port	Available

Note: Interrupts are available to the I/O channel if they are not enabled by the system board function normally assigned to that interrupt.

Figure 1-9. Hardware Interrupt Listing

Interrupt Sharing

A standardized hardware design concept has been established to enable multiple adapters to share an interrupt level. The following describes this design concept and discusses the programming support required.

Design Overview

Most interrupt supporting adapters hold the IRQ line inactive and then drive the line active to cause an interrupt. In contrast, the shared interrupt hardware design allows the IRQ line to float high. Each adapter on the line may cause an interrupt by pulsing the line low. The leading edge of the pulse arms the interrupt controller; the trailing edge of the pulse causes the interrupt.

Each adapter sharing an interrupt level must monitor the IRQ line. When any adapter pulses the line, all other adapters on that interrupt must not issue an interrupt request until they are rearmed.

If an adapter's INT is active when it is rearmed, the adapter must reissue the interrupt. This prevents lost interrupts in case two adapters issue an interrupt at exactly the same time and an interrupt handler issues a global rearm after servicing one of them.

The following diagram shows the shared interrupt hardware logic.

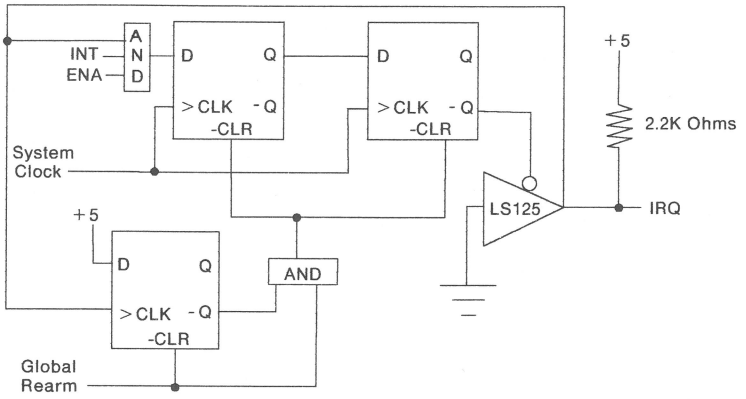


Figure 1-10. Shared Interrupt Hardware Logic

Program Support

The interrupt-sharing program support described in the following provides for an orderly means to:

- Link a task's interrupt handler to a chain of interrupt handlers.
- Share the interrupt level while the task is active.
- Unlink the interrupt handler from the chain when the task is deactivated.

Linking onto the Chain: Each newly activated task replaces the interrupt vector in low memory with a pointer to its own interrupt handler. The old interrupt vector is used as a forward pointer and is stored away at a fixed offset from the new task's interrupt handler. This method of linking means the last handler to link is the first one in the chain.

Sharing the Interrupt Level: When the new task's handler gains control as a result of an interrupt, the handler reads the contents of the adapter's interrupt status register to determine if its adapter caused the interrupt. If its adapter did cause the interrupt, the handler services the interrupt, disables the interrupts (CLI), and writes to address hex 02F*X*, where *X* corresponds to the interrupt levels 2 through 7. Each adapter in the chain decodes the address, which results in a Global Rearm. The handler then issues a nonspecific End of Interrupt (EOI) and finally issues a Return from Interrupt (IRET). If its adapter did not cause the interrupt, the handler passes control to the next interrupt handler in the chain.

Unlinking from the Chain: To unlink from the chain, a task must first locate its handler's position within the chain. By starting at the interrupt vector in low memory, and using the offset of each handler's forward pointer to find the entry point of each handler, the chain can be methodically searched until the task finds its own handler. The forward pointer of the previous handler in the chain is replaced by the task's pointer, removing the handler from the chain.

Note: If the handler cannot locate its position in the chain or the signature of any prior handler is not hex 424B, it must not unlink.

Error Recovery: If the unlinking routine discovers that the interrupt chain has been corrupted, an unlinking error recovery procedure must be in place. Each application can incorporate its own unlinking error procedure into the unlinking routine. One application may choose to display an error message requiring the operator to either correct the situation or reset the system. The application, however, must not unlink.

Precautions

The following precautions must be taken when designing hardware or programs using shared interrupts.

- Hardware designers should ensure that the adapters:
  - Do not power up with an interrupt pending or enabled.
  - Do not generate interrupts that are not serviced by a handler. Generating interrupts when a handler is not active to service them causes that interrupt level to lock up. The design concept relies on the handler to clear its adapter's interrupt and issue the Global Rearm.
  - Can be disarmed so that they do not remain active after their application has terminated.
- Programmers should:
  - Ensure that their programs contain a short routine that can be executed with the AUTOEXEC.BAT to disable their adapter's interrupts. This precaution ensures that the adapters are deactivated for a system reboot that does not clear memory.
  - Treat words as words, not bytes.

    Note: Remember that data is stored in memory using the Intel format (word hex 424B is stored as hex 4B42).

Interrupt Chaining Structure

```
ENTRY:  JMP         SHORT PAST        ; Jump around structure
        FPTR        DD      0         ; Forward Pointer
        SIGNATURE   DW      424BH     ; Used when unlinking to identify
                                      ;  compatible interrupt handlers
        FLAGS       DB      0         ; Flags
        FIRST       EQU     80H       ; Flags for being first in chain
        JMP         SHORT   RESET
        RES_BYTES   DB      DUP 7(0)  ; Future Expansion
PAST:   ...                           ; Actual start of code
```

The interrupt chaining structure is a 16-byte format containing FPTR, SIGNATURE, RES_BYTES, and a Jump instruction to a reset routine. It begins at the third byte from the interrupt handler's entry point. The first instruction of every handler is a short jump around the structure to the start of the routine.

Except for those residing in adapter ROM, handlers designed for interrupt sharing must use hex 424B as the signature to avoid corrupting the chain. Because each handler's chaining structure is known, the forward pointers can be updated when unlinking.

The flag indicates that the handler is first in the chain, and is used only with interrupt 7. The RESET routine disables the adapter's interrupt, then does a Far Return to the operating system.

ROM Considerations

Adapters with interrupt handlers residing in ROM must store the forward pointer in latches or ports on the adapter. If the adapter is sharing interrupt 7, it must, also, store FIRST. Storing this flag is necessary because its position in the chain may not always be first.

Because the forward pointer is not stored in the third byte, these handlers must contain a signature of hex 00.

Examples

In the following examples, note that interrupts are disabled before passing control to the next handler on the chain. The next handler receives control as if a hardware interrupt had caused it to receive control. Also, note that the interrupts are disabled before the nonspecific EOI is issued, and not reenabled in the interrupt handler. This ensures that the IRET is executed (at which point the flags are restored and the interrupts reenabled) before another interrupt is serviced, protecting the stack from excessive buildup.

Interrupt Handler Example

```
OUR_CARD    EQU     xxxx            ; Location of our card's interrupt
ISB         EQU     xx              ; Interrupt bit in our cards interrupt
                                    ;  control/status register
REARM       EQU     2F7H            ; Global Rearm location for interrupt 7
SPC_EOI     EQU     67H             ; Specific EOI for interrupt 7
EOI         EQU     20H             ; Nonspecific EOI
OCR         EQU     03CH            ; Location of interrupt controller
                                    ;   operational control register
IMR         EQU     21H             ; Location of interrupt mask register

MYSEG       SEGMENT PARA
            ASSUME  CS:MYSEG,DS:DSEG
ENTRY       PROC    FAR
            JMP     SHORT PAST      ; Entry point of handler
FPTR        DD      0               ; Forward Pointer
SIGNATURE   DW      424BH           ; Used when unlinking to identify
                                    ;  compatible interrupt handlers
            FLAGS   DB      0       ; Flags
            FIRST   EQU     80H
            JMP     SHORT   RESET
RES_BYTES   DB      DUP 7(0)        ; Expansion
PAST:       STI                     ; Actual start of handler code
            PUSH    ...             ; Save needed registers
            MOV     DX,OUR_CARD     ; Select our status register
            IN      AL,DX           ; Read the status register
            TEST    AL,ISB          ; Our card caused the interrupt?
            JNE     SERVICE         ; Yes, branch to service logic
            TEST    CS:FLAGS,FIRST  ; Are we the first ones in?
            JNZ     EXIT            ; If yes, branch for EOI and Rearm
            POP     ...             ; Restore registers
            CLI                     ; Disable interrupts
            JMP     DWORD PTR CS:FPTR ; Pass control to next handler on chain

SERVICE:    ...                     ; Service the interrupt
EXIT:
            CLI                     ; Disable the interrupts
            MOV     AL,EOI
            OUT     OCR,AL          ; Issue nonspecific EOI
            MOV     DX,REARM        ; Rearm our card
            OUT     DX,AL
            POP     ...             ; Restore registers
            IRET
RESET:      ...                     ; Disable our card
            RET                     ; Return Far to operating system
ENTRY:      ENDP
            MYCSEG  ENDS
            END     ENTRY
```

Linking Code Example

```
          PUSH     ES
          CLI                          ; Disable interrupts
; Set forward pointer to the value of the interrupt vector in low memory
          ASSUME   CS:CODESEG,DS:CODESEG
          PUSH     ES
          MOV      AX,350FH            ; DOS get interrupt vector
          INT      21H                 ;
          MOV      SI,OFFSET CS:FPTR   ; Set offset of our forward pointer
                                       ;  in an indexable register
          MOV      CS:[SI],BX          ; Store the old interrupt vector
          MOV      CS:[SI+2],ES        ; in our forward pointer
          CMP      ES:BYTE PTR[BX],CFH  ; Test for IRET
          JNZ      SERVECTR
          MOV      CS:FLAGS,FIRST      ;Set up first in chain flag
SERVECTR: POP      ES
          PUSH     DS
; Make interrupt vector in low memory point to our handler
          MOV      DX,OFFSET ENTRY     ; Make interrupt vector point to our
                                       ;  interrupt handler
          MOV      AX,SEG ENTRY        ; If DS not = CS, get it and
          MOV      DS,AX               ;  put it in DS
          MOV      AX,250FH            ; DOS set interrupt vector
          INT      21H                 ;
          POP      DS
; Unmask (enable) interrupts for our level
SET7:     IN       AL,IMR              ; Read interrupt mask register
          AND      AL,07FH             ; Unmask interrupt level 7
          OUT      IMR,AL              ; Write new interrupt mask
          MOV      AL,SPC_EOI          ; Issue specific EOI for level 7
          OUT      OCR,AL              ;  to allow pending level 7 interrupts
                                       ;   (if any) to be serviced
          STI                          ; Enable interrupts
          POP      ES
```

Unlinking Code Example

```
            PUSH    DS
            PUSH    ES
            CLI                     ; Disable interrupts
            MOV     AX,350FH        ; DOS get interrupt vector
            INT     21H             ; ES:BX points to the first in the chain
            MOV     CX,ES           ; Pickup segment part of interrupt vector
 ; Are we the first handler in the chain?
            MOV     AX,CS           ; Get code seg into comparable register
            CMP     BX,OFFSET ENTRY ; Interrupt vector in low memory
                                    ;  pointing to our handlers offset?
            JNE     UNCHAIN_A       ; No, branch
            CMP     AX,CX           ; Vector pointing to our handler's segment?
            JNE     UNCHAIN_A       ; No, branch
; Set interrupt vector in low memory to point to the handler
;  pointed to by our pointer
            PUSH    DS
            MOV     AX,CS:FPTR
            MOV     DX,WORD PTR CS:FPTR ; Set offset of interrupt vector
            MOV     DS,WORD PTR CS:FPTR[2] ; Set segment of interrupt vector
            MOV     AX,250FH        ; DOS set interrupt vector
            INT     21H
            POP     DS
            JMP     UNCHAIN_X
UNCHAIN_A: ; CX = FPTR segment, BX = FPTR offset
            CMP     ES:[BX+6],4B42H ; Is handler using the appropriate
                                    ;  conventions (is SIGNATURE = 424BH?
            JNE     exception       ; No, invoke error exception handler
            LDS     SI,ES:[BX+2]    ; Get FPTR's segment and offset
            CMP     SI,OFFSET ENTRY ; Is this forward pointer pointing to
                                    ;  our handler's offset?
            JNE     UNCHAIN_B       ; No, branch
            MOV     CX,DS           ; Is this forward pointer pointing to
            CMP     AX,CX           ;  our handler's segment?
            JNE     UNCHAIN_B       ; No, branch
; Located our handler in the chain
            MOV     AX,WORD PTR CS:FPTR ; Get our FPTR's offset
            MOV     ES:[BX+2],AX    ; Replace FPTR offset pointing to us
            MOV     AX,WORD PTR CS:FPTR[2] ; Get our FPTR's segment
            MOV     ES:[BX+4],AX    ; Replace FPTR segment pointing to us
            MOV     AL,CS:FLAGS
            AND     AL,FIRST
            OR      ES:[BX+6],AX    ; Replace offset of FPTR of handler
            JMP     UNCHAIN_X
UNCHAIN_B: MOV      BX,SI            ; Move new offset to BX
            PUSH    DS
            PUSH    ES
            JMP     UNCHAIN_A       ; Examine the next handler in the chain
UNCHAIN_X: STI                       ; Enable interrupts
            POP     ES
            POP     DS
```

Read/Write Memory

The system board has 640K of read/write memory. The first 128K consists of four 64K by 4-bit and two 64K by 1-bit chips. These chips are soldered to the system board.

The next 512K (from 128K to 640K) is arranged as two banks of 256K by 9-bit single-inline packages (SIPs). All read/write memory is parity checked.

The Planar RAM Control/Status register, hex 6B, is part of the system gate array and may be used to remap memory. Remapping occurs when the power-on self-test (POST) senses memory on the I/O channel that is in contention with system memory. Also, if a failure in the first 128K is sensed, POST remaps the remainder of memory to allow the system to operate, although at reduced capacity.

Bit	Function
7	Parity Check Pointer 1 = Lower 128K failed 0 = Upper 512K failed
6	-Enable RAM, 90000-9FFFF
5	-Enable RAM, 80000-8FFFF
4	-Enable RAM, 70000-7FFFF
3	-Enable RAM, 60000-6FFFF
2	-Enable RAM, 50000-5FFFF
1	-Enable RAM, 40000-4FFFF
0	Remap Low Memory

Figure 1-11. Planar RAM Control/Status Register

ROM

The system board has space for 64K by 8-bits of ROM or erasable programmable read-only memory (EPROM). Two module sockets are provided; both sockets have 32K-by-8 bits of ROM installed. This ROM contains POST, BIOS, dot patterns for 128 characters in graphics mode, and a diskette bootstrap loader. The ROM is packaged in 28-pin modules.

I/O Channel

The I/O channel is an extension of the 8086 microprocessor bus that is demultiplexed, repowered, and enhanced by the addition of interrupts and DMA functions.

The I/O channel contains:

- An 8-bit, bidirectional data bus
- 20 address lines
- Six levels of interrupt
- Control lines for memory and I/O read and write
- Clock and timing lines
- Three channels of DMA control lines
- Memory-refresh control lines
- A channel check line
- Power and ground for the adapters.

Four voltage levels are provided for I/O cards: +5 Vdc ±5%, -5 Vdc ±10%, +12 Vdc ±5%, and -12 Vdc ±10%.

The 'I/O channel ready' line (IO CH RDY) is available on the I/O channel to allow operation with slow I/O or memory devices. IO CH RDY is made inactive by an addressed device to lengthen the operation. For each clock cycle that the line is held low, one wait state is added to the I/O and DMA operations.

I/O devices are addressed using mapped I/O address space. The channel is designed so that over 64,000 device addresses are available to the adapters on the I/O channel.

The following is the I/O address map for the Model 30. Hex 0100 to FFFF are available on the I/O channel.

Hex Range	Device
0000-001F	DMA Controller, 8237A-5
0020-003F	Interrupt Controller
0040-005F	Timer
0060-0062	I/O Ports
0063-006F	System Board/Control and Status
0080-008F	DMA Page Registers
00A0-00AF*	Interrupt Controller Extension
00B0-00BF	Real-Time Clock Command/Status
00E0-00EF	Real-Time Clock Counter/RAM
0320-032F	Fixed Disk
0378-037F	Parallel Port
03C0-03DF	Video Subsystem
03F0-03F7	Diskette
03F8-03FF	Serial Port

Note: I/O Addresses, hex 000 to 0FF, are reserved for the system board I/O.

* The NMI mask can be set and reset through system software as follows:

Write hex 80 to I/O address hex A0 (enable NMI)
Write hex 00 to I/O address hex A0 (disable NMI)

Figure 1-12. I/O Address Map

The '-I/O channel check' signal (-IO CH CK) causes a non-maskable interrupt (NMI) to the microprocessor.

Connectors

The I/O channel is repowered to provide sufficient power for all three 62-pin connectors, assuming two low-power Schottky (LS) loads per slot. IBM adapters typically use only one load per adapter.

The following figures show the pin numbering and signal assignments for the I/O channel connectors.

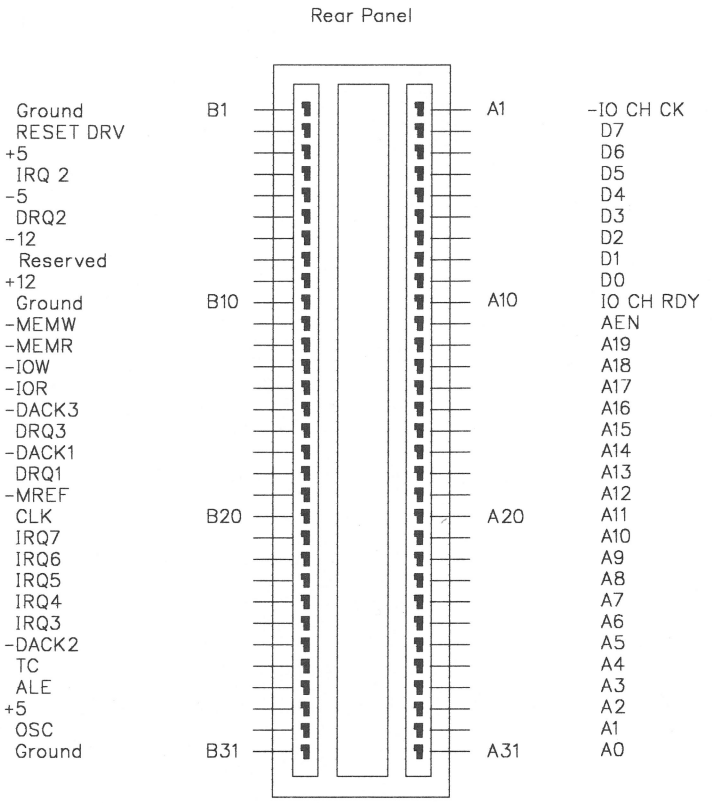


Figure 1-13. I/O Channel

Signal Description

The following is a description of the I/O channel signal lines. All lines are TTL-compatible.

A0 – A19 (O): Address bits 0 to 19: These lines are used to address memory and I/O devices within the system. The 20 address lines allow access to 1M of memory. Only the lower 16 lines are used in I/O addressing, and all 16 should be decoded by I/O devices. A0 is the least significant and A19 is the most significant. These lines are generated by either the microprocessor or DMA controller.

AEN (O): Address Enable: This line is used to de-gate the microprocessor and other devices from the I/O channel to allow DMA transfers to take place. When this line is active, the DMA controller has control of the address bus, data bus, and Read and Write command lines. When this line is inactive, the microprocessor has control. This line should be part of the adapter-select decode to prevent incorrect adapter selects during DMA operations.

ALE (O): Address Latch Enable: This line is provided by the bus controller and is used on the system board to latch valid addresses from the microprocessor. Addresses are valid at the falling edge of ALE and are latched onto the bus while ALE is inactive. This signal is forced active during DMA cycles.

CLK (O): System clock: This is the system clock signal with a frequency of 8 MHz and a 33% duty cycle.

D0—D7 (I/O): Data Bits 0 to 7: These lines provide data bus bits 0 to 7 for the microprocessor, memory, and I/O devices.

-DACK1 – -DACK3 (O): -DMA Acknowledge 1 to 3: These lines are used by the controller to acknowledge DMA requests. DACK0 is not available on the Model 30's I/O channel.

DRQ1 – DRQ3 (I): DMA Request 1 to 3: These lines are asynchronous channel requests used by peripheral devices to gain DMA services. They are prioritized with DRQ1 being the highest and DRQ3 being the lowest. A request is generated by bringing a request line to an active level. A request line is held active until the corresponding acknowledge line goes active.

-IO CH CK: -I/O Channel Check: This line generates an NMI. It is driven active to indicate an uncorrectable error and held active for at least two clock cycles.

IO CH RDY (I): I/O Channel Ready: This line, normally active (ready), is pulled inactive (not ready) by a memory or I/O device to lengthen I/O or memory cycles. It allows slower devices to attach to the I/O channel with a minimum of difficulty. Any slow device using this line should drive it inactive immediately after detecting a valid address and a Read or Write command. For every clock cycle this line is inactive, one wait state is added. This line should not be held inactive longer than 17 clock cycles.

-IOR (O): -I/O Read: This command line instructs an I/O device to drive its data onto the data bus. This signal is driven by the microprocessor or the DMA controller.

-IOW (O): -I/O Write: This command line instructs an I/O device to read the data on the data bus. This signal is driven by the microprocessor or the DMA controller.

IRQ2—IRQ7 (I): Interrupt requests 2 through 7: These lines are used to signal the microprocessor that an I/O device requires attention. They are prioritized with IRQ2 as the highest priority and IRQ7 as the lowest. When an interrupt is generated, the request line is held active until it is acknowledged by the microprocessor.

-MEMR (O): -Memory Read: This command line instructs the memory to drive its data onto the data bus. This signal is driven by the microprocessor or the DMA controller.

-MEMW (O): -Memory Write: This command line instructs the memory to store the data present on the data bus. This signal is driven by the microprocessor or the DMA controller.

-MREF (I/O): -Memory Refresh: This line indicates a refresh cycle.

OSC (O): Oscillator: High-speed clock with a 70-ns period (14.31818 MHz). It has a 50% duty cycle.

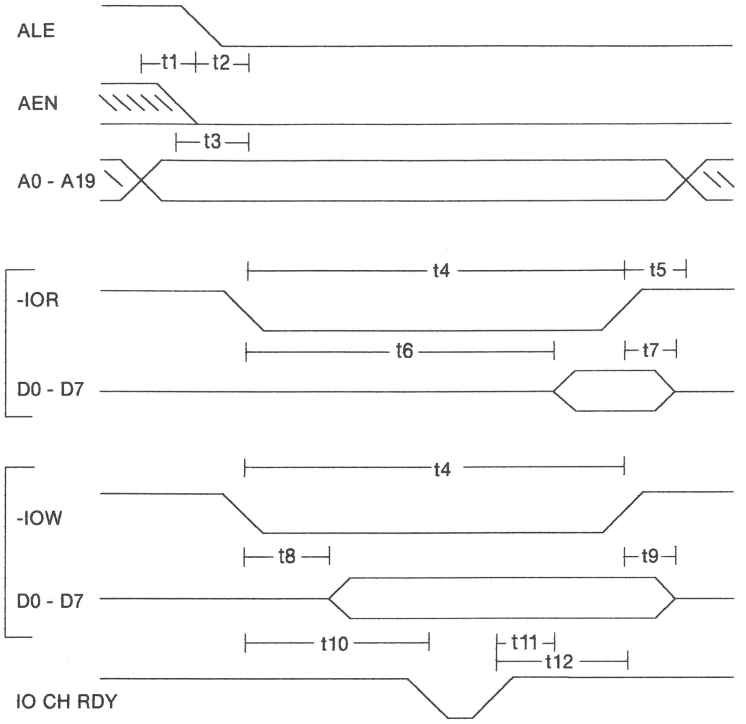
RESET DRV (O): Reset Drive: This line is used to reset or initialize system logic upon power-up or during a low line-voltage. This signal is synchronized to the falling edge of CLK.

TC (O): Terminal Count: This line provides a pulse when the terminal count for any DMA channel is reached.

Signal Timings

The following diagrams show the I/O signal timings for I/O and memory operations.

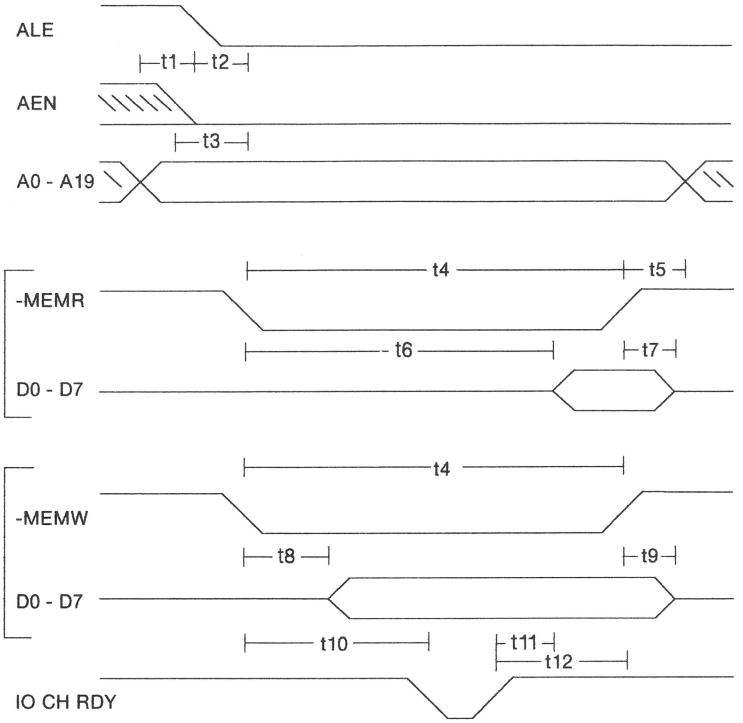
8-Bit I/O Bus Cycles



Symbol	Description	Min (ns)	Max (ns)
t1	Address valid to ALE inactive	20	
t2	ALE inactive to Command active	60	
t3	Command active from AEN inactive	95	
t4	Command pulse width	605	
t5	Address hold from Command inactive	45	
t6	Data valid from Read active		540
t7	Data hold from Read inactive	0	
t8	Data valid from Write active		120
t9	Data hold from Write inactive	25	
t10	IO CH RDY inactive from Command active		325
t11	Read Data valid from IO CH RDY active		0
t12	Command inactive from IO CH RDY active	160	

Figure 1-14. 8-Bit I/O Timing

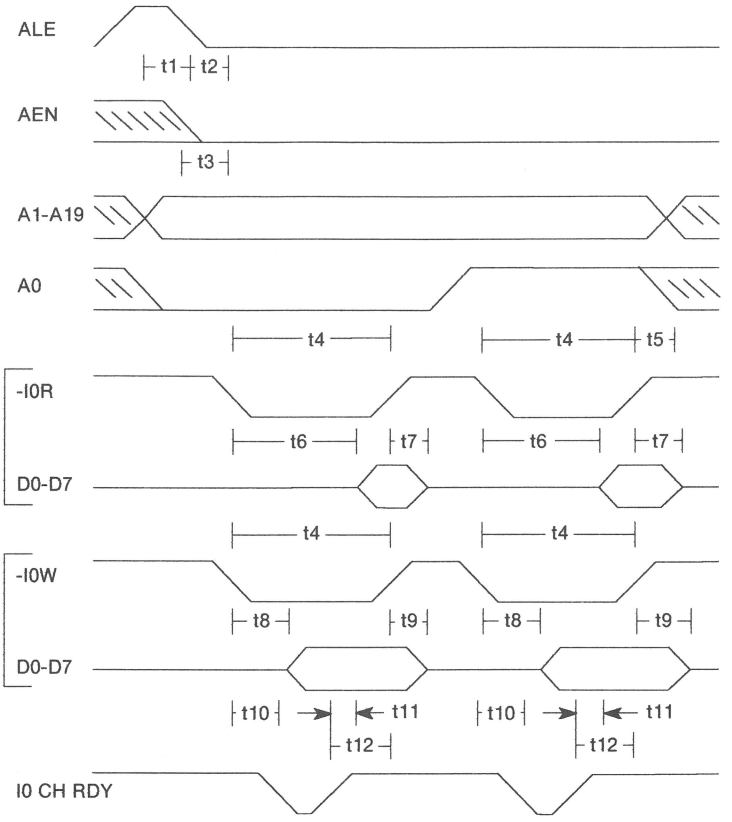
8-Bit Memory Bus Cycles



Symbol	Description	Min (ns)	Max (ns)
t1	Address valid to ALE inactive	20	
t2	ALE inactive to Command active	60	
t3	Command active from AEN inactive	95	
t4	Command pulse width	395	
t5	Address hold from Command inactive	45	
t6	Data valid from Read active		315
t7	Data hold from Read inactive	0	
t8	Data valid from Write active		120
t9	Data hold from Write inactive	25	
t10	IO CH RDY inactive from Command active		115
t11	Read Data valid from IO CH RDY active		0
t12	Command inactive from IO CH RDY active	160	

Figure 1-15. 8-Bit Memory Timing

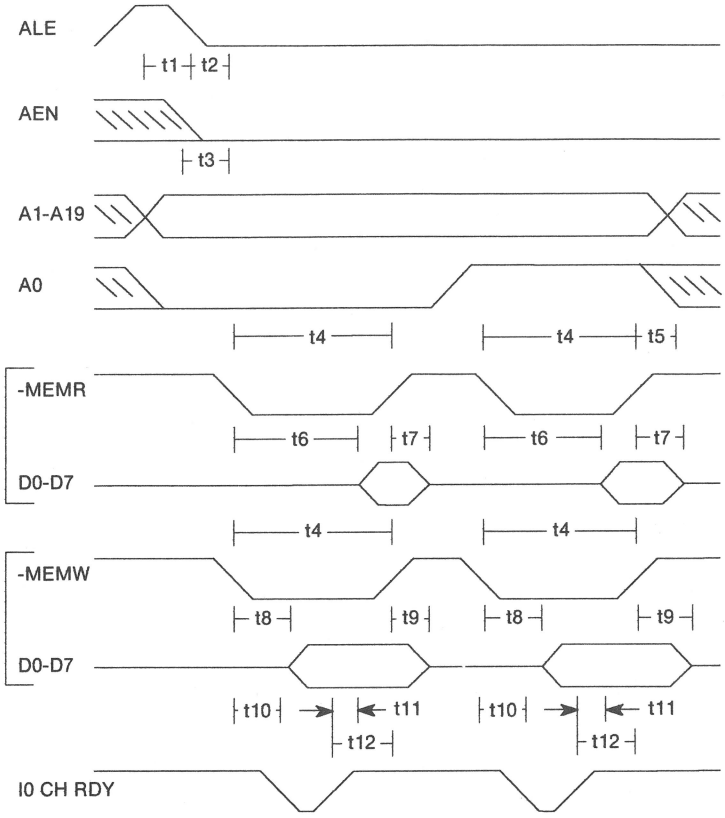
16-Bit I/O Bus Cycles



Symbol	Description	Min (ns)	Max (ns)
t1	Address valid to ALE inactive	20	
t2	ALE inactive to Command active	60	
t3	Command active from AEN inactive	95	
t4	Command pulse width	605	
t5	Address hold from Command inactive	45	
t6	Data valid from Read active		540
t7	Data hold from Read inactive	0	
t8	Data valid from Write active		120
t9	Data hold from Write inactive	25	
t10	IO CH RDY inactive from Command active		325
t11	Read Data valid from IO CH RDY active		0
t12	Command inactive from IO CH RDY active	160	

Figure 1-16. 16-Bit I/O Timing

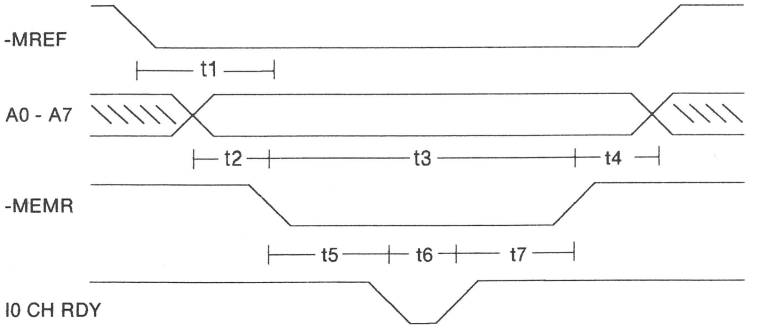
16-Bit Memory Bus Cycles



Symbol	Description	Min (ns)	Max (ns)
t1	Address valid to ALE inactive	20	
t2	ALE inactive to Command active	60	
t3	Command active from AEN inactive	95	
t4	Command pulse width	395	
t5	Address hold from Command inactive	45	
t6	Data valid from Read active		315
t7	Data hold from Read inactive	0	
t8	Data valid from Write active		120
t9	Data hold from Write inactive	25	
t10	IO CH RDY inactive from Command active		115
t11	Read Data valid from IO CH RDY active		0
t12	Command inactive from IO CH RDY active	160	

Figure 1-17. 16-Bit Memory Timing

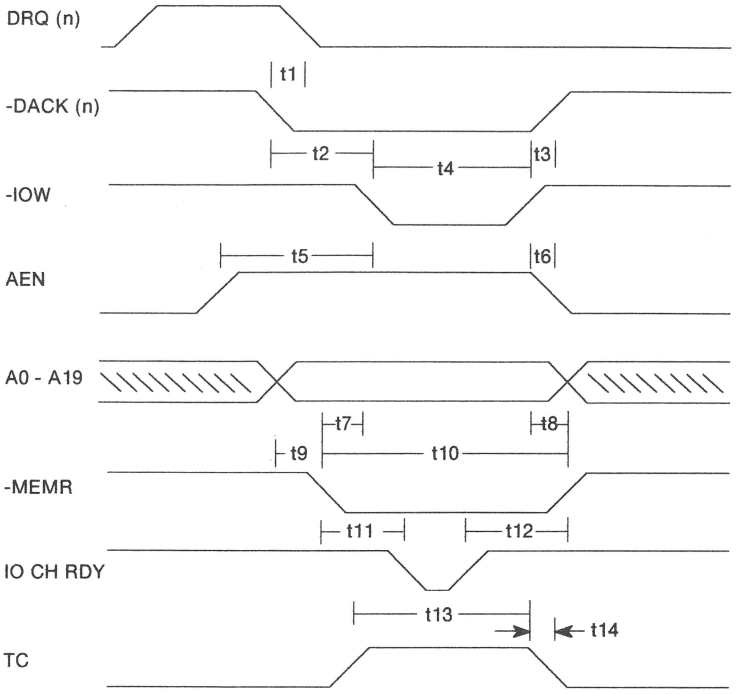
Memory Refresh



Symbol	Description	Min (ns)	Max (ns)
t1	-MREF active to -MEMR active	155	
t2	Address valid to -MEMR active	75	
t3	-MEMR pulse width	230	
t4	-MEMR inactive to -MREF inactive	10	
t5	-MEMR active to IO CH RDY inactive		60
t6	IO CH RDY pulse width		600
t7	-MEMR inactive from IO CH RDY active	0	

Figure 1-18. Memory Refresh Timing

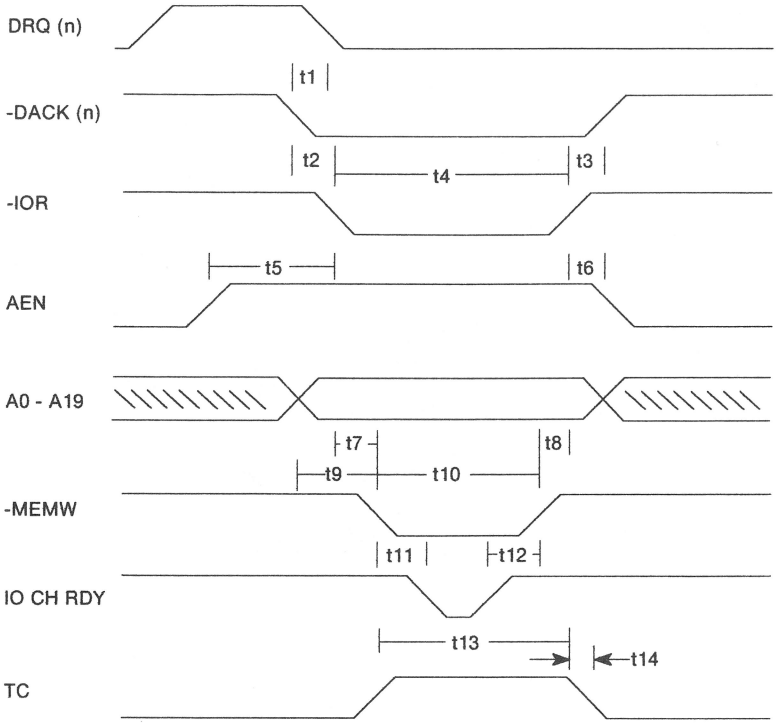
DMA Read



Symbol	Description	Min (ns)	Max (ns)
t1	-DACK active to DRQ inactive	0	
t2	-DACK active to -IOW active	200	
t3	-IOW inactive to -DACK inactive	0	
t4	-IOW pulse width	250	
t5	AEN active to -IOW active	500	
t6	-IOW inactive to AEN inactive	25	
t7	-IOW active from -MEMR active		360
t8	-IOW inactive to -MEMR inactive	0	
t9	Address valid to -MEMR active	0	
t10	-MEMR pulse width	470	
t11	-MEMR active to IO CH RDY inactive		200
t12	-MEMR inactive from IO CH RDY active	200	
t13	TC active setup to -IOW inactive	290	
t14	TC inactive from -IOW inactive	0	

Figure 1-19. DMA Read Timing

DMA Write



Symbol	Description	Min (ns)	Max (ns)
t1	-DACK active to DRQ inactive	0	
t2	-DACK active to -IOR active	0	
t3	-IOR inactive to -DACK inactive	0	
t4	-IOR pulse width	470	
t5	AEN active to -IOR active	300	
t6	-IOR inactive to AEN inactive	0	
t7	-MEMW active from -IOR active	55	
t8	-MEMW inactive to -IOR inactive	0	
t9	Address valid to -MEMW active	140	
t10	-MEMW pulse width	250	
t11	-MEMW active to IO CH RDY inactive		30
t12	-MEMW inactive from IO CH RDY active	200	
t13	TC active setup to -IOR inactive	290	
t14	TC inactive from -IOR inactive	0	

Figure 1-20. DMA Write Timing

Video Subsystem

The video subsystem is resident on the system board and consists of:

- Memory controller gate array
- Video formatter gate array
- 64K of multiport dynamic memory
- 8K static RAM character generator
- 256-by-18-bit color palette with three 6-bit digital-to-analog converters (DAC).

At the BIOS level (interrupt 10), the Model 30 maintains compatibility with the IBM Color/Graphics Adapter.

The video modes are compatible with those modes supported by the color/graphics adapter with two modes added. The additional modes are the 320-by-200 graphics with 256 colors available, and the 640-by-480 graphics with two colors available.

Block Diagram

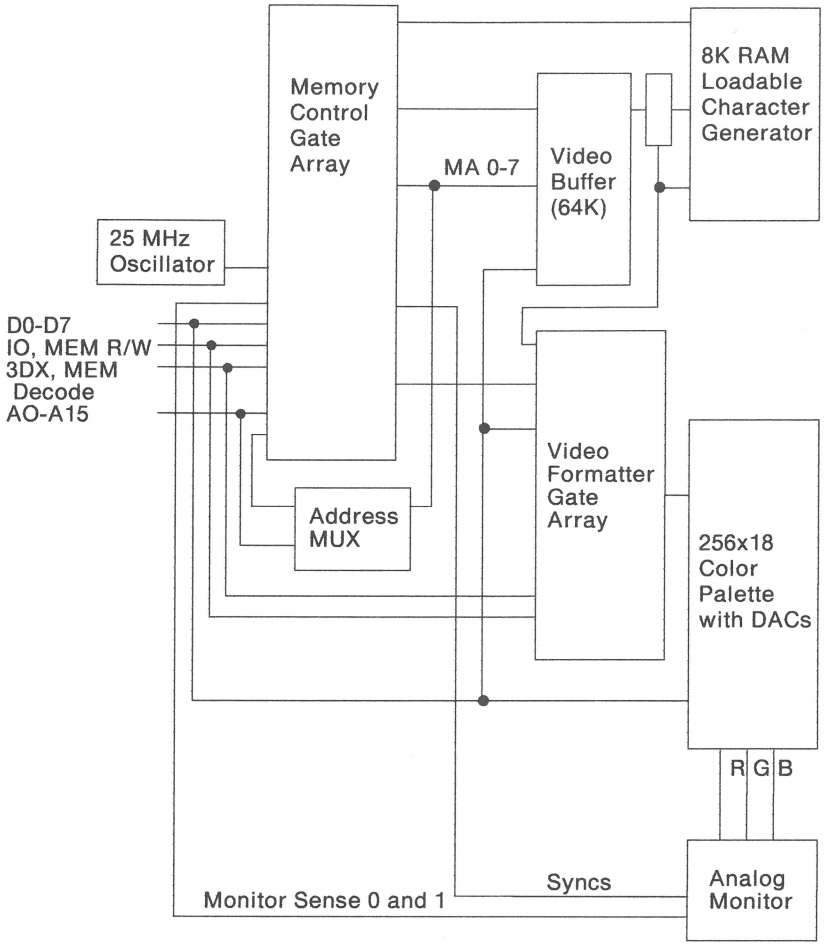


Figure 1-21. Video Subsystem Block Diagram

Display Support

The video subsystem supports a 31.5 kHz analog color display or 31.5 kHz analog monochrome display. The system senses the type of display and matches the initialization to it. The polarity of the two synchronization signals to the display determines the number of horizontal scans, either 400 or 480. The number of scan lines in relation to the polarity is:

Scan Lines	Vertical Sync	Horizontal Sync
480	Negative	Negative
400	Positive	Negative

If the system senses the presence of a monochrome display, it sums the colors and outputs the video signal to pin 2 (green). The signal returns through pin 7 (green return).

Text Modes

In the text modes, the character box size is 8 by 16. The character font table is loaded into the character generator. All 16 scan lines are programmed into the character generator.

Graphic Modes

In the graphic modes, the character font table is used to create the character PELs. For most graphics modes, the character box is an 8-by-8 character box that is double scanned to create an 8-by-16 character; however, all 16 scan lines of the 8-by-16 box are not programmable.

The 640-by-480 graphics mode is the exception. It uses an 8-by-16 character box and a separate font table. In this mode, 30 character rows are displayed.

Video Modes	Analog Display
Mode 0,1 40 Column Alphanumeric	40 column by 25 rows 8x16 character box 320 by 400 16 of 256K colors Display buffer B8000 2000 byte video buffer
Mode 2,3 80 Column Alphanumeric	80 column by 25 rows 8x16 character box 640 by 400 16 of 256K colors Display buffer B8000 4000 byte video buffer
Mode 4,5 320 by 200 Graphic	8x8 character box Double-scanned 320 by 200 4 of 256K colors Alt. palette select Display buffer B8000 16000 byte video buffer Two Row scan address partitions
Mode 6 640 by 200 Graphic	8x8 character box Double-scanned 640 by 200 2 of 256K colors Display buffer B8000 16000 byte video buffer Two Row scan address partitions
Mode 11 640 by 480 Graphic	8x16 character box 640 by 480 2 of 256K colors Display buffer A0000 38400 byte video buffer Linear addressing
Mode 13 320 by 200 Graphic	8x8 character box Double-scanned 320 by 200 256 of 256K colors Display buffer A0000 64000 byte video buffer Linear addressing

Figure 1-22. Video Mode Summary

Display Formats

In alphanumeric (text) modes 0 through 3, two bytes define each character on the display screen. The even byte accesses the character generator to create the PEL data. The odd byte defines the color of the PELs. Sixteen colors are available for foreground, and eight colors are available for background when blink is enabled (default). Blink is controlled in the CGA Mode Control register, hex 3D8.

The format of the two bytes is shown in the following:

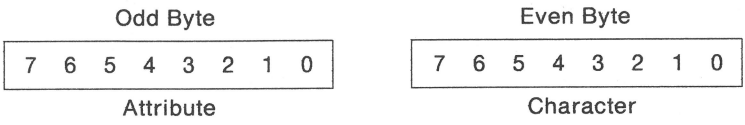


Figure 1-23. Alphanumeric Format

The following are the bit definitions of the attribute byte. Bit 7 selects a blinking character or, if blinking is disabled, selects palette addresses above hex 07 for the background color.

Bits	Function
7 to 4	Background Color Palette Address
3 to 0	Foreground Color Palette Address

Figure 1-24. Attribute Byte

In Modes 4 and 5, the bit pair C1 and C0 select one of four colors for each PEL.

Bit	PEL Definition	
7,6	C1,C0	First PEL
5,4	C1,C0	
3,2	C1,C0	
1,0	C1,C0	Last PEL

Figure 1-25. Modes 4 and 5

There are two color sets: color set 0 and color set 1. For information about the colors selected, see CGA Border Control Register, 3D9 (BCR), later in this section under "Video Formatter Registers."

In Modes 6 and 11, one bit defines each PEL, with the most significant bit defining the first PEL. The foreground color maps to the color in the CGA Border Control register if the B&W bit in the CGA Mode Control register is 0. If the B&W bit is 1, the foreground color maps to palette address hex 07. The background color always maps to address hex 00.

Bit	PEL Definition	
7	C0	First PEL
6	C0	
5	C0	
4	C0	
3	C0	
2	C0	
1	C0	
0	C0	Last PEL

Figure 1-26. Modes 6 and 11

In Mode 13, a byte defines each PEL. This allows a choice of 256 colors for each PEL.

Video Storage Organization

The following is the memory mapping for text modes 0 through 3.

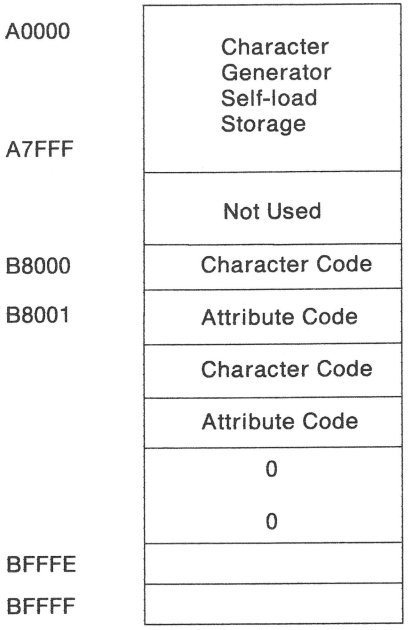


Figure 1-27. Text Modes 0 to 3

The following is the memory mapping for graphics modes 4 through 6.

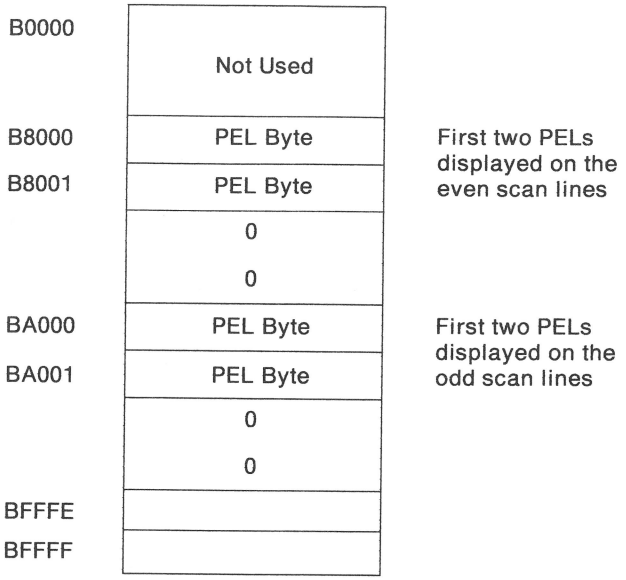


Figure 1-28. Graphic Modes 4 to 6

The following is the memory mapping for graphics modes 11 and 13. In mode 11, each byte defines eight PELs; in mode 13, each byte defines one PEL.

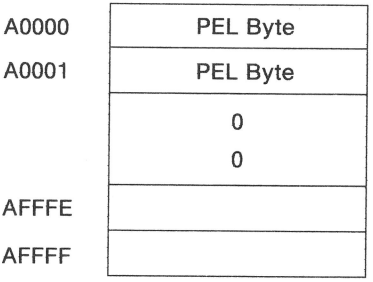


Figure 1-29. Graphic Modes 11 and 13

Video Registers

The memory controller gate array responds to I/O addresses 3D4 and 3D5. The video formatter gate array responds to I/O addresses 3D8 through 3DF.

The color palette is programmed through the video formatter at addresses 3C6 through 3C9. All registers are readable.

The following pages describe the memory controller registers, the video formatter registers, the color palette registers, and the character generator. Also, sample programs of a font load and palette load are included.

Memory Controller Registers

The memory controller contains an index register and 22 data registers. Two I/O commands are required to write to one data register: writing the desired index value to address hex 3D4, and then writing the data to address hex 3D5.

Memory Controller Index Register, Hex 3D4: This register is read and write, and points to the specific data register addressed through hex 3D5.

Bit	Function
7	Reserved
6	Reserved
5	Index5
4	Index4
3	Index3
2	Index2
1	Index1
0	Index0

Figure 1-30. Memory Controller Index Register

The following is a list of the 22 data registers and their functions.

Index (Hex)	Register Description
00	Horizontal Total
01	Horizontal Characters Displayed
02	Start Horizontal Sync
03	Sync Pulse Width
04	Vertical Total
05	Vertical Total Adjust
06	Vertical Characters Displayed
07	Start Vertical Sync
08	Reserved
09	Scan Lines per Character
0A	Cursor Start
0B	Cursor End
0C	Start of Screen High
0D	Start of Screen Low
0E	Cursor Position High
0F	Cursor Position Low
10	Mode Control
11	Interrupt Control
12	Character Generator Interface and Sync Polarity, or Display Sense
13	Character Font Pointer
14	Number of Characters to Load
20	Reserved

Horizontal Total Register, Index 00: This register contains the total number of characters in the horizontal scan interval. The number consists of both the displayed and nondisplayed characters. This register determines the frequency of the 'horizontal sync' signal.

Horizontal Characters Displayed Register, Index 01: This register determines the total number of characters to be displayed during the horizontal video scan interval. This register is loaded with a value of hex 27. The hardware calculates the correct value based on the mode selected.

Start Horizontal Sync Register, Index 02: This register specifies the character position count at which the 'horizontal sync' signal becomes active.

Sync Pulse width Register, Index 03: This register specifies the pulse widths of the horizontal and vertical synchronization signals. The horizontal pulse width is programmed in units of character clocks. The vertical pulse width is programmed in units of the horizontal synchronization period. This register is programmed to match the display specifications.

Bit	Function
7	Width VSync3
6	VSync2
5	VSync1
4	VSync0
3	Width HSync3
2	HSync2
1	HSync1
0	HSync0

Figure 1-31. Sync Pulse Width Register

Vertical Total Register, Index 04: This register contains the 8 least-significant bits for the total number of horizontal scan lines in the vertical scan interval. The ninth bit is the inversion of bit 6 of the Mode Control register. The total number consists of both the displayed and nondisplayed scan lines. This register and the Vertical Total Adjust register determine the frequency of the 'vertical sync' signal.

Vertical Total Adjust Register, Index 05: This register is used to adjust the total number of horizontal scan lines in the vertical scanning interval. It allows for an odd number of horizontal lines (525 for 60 Hz). The minimum value for this register is a hex 02.

Bit	Function
7	Reserved
6	Reserved
5	VAdjust5
4	VAdjust4
3	VAdjust3
2	VAdjust2
1	VAdjust1
0	VAdjust0

Figure 1-32. Vertical Total Adjust Register

Vertical Characters Displayed Register, Index 06: This register contains the 8 least-significant bits for the number of horizontal scan lines displayed during the vertical scan interval. The ninth bit is the inversion of bit 6 of the Mode Control register.

Start Vertical Sync Register, Index 07: This register contains the 8 least-significant bits for the vertical scan line count. It determines when the 'vertical sync' signal becomes active. The ninth bit is the inversion of bit 6 of the Mode Control register.

Scan Lines per Character Register, Index 09: This register determines the number of horizontal scan lines in a character row. In text modes, the value is hex 07. In graphics modes 4 through 6, the value is hex 01, and in modes 11 and 13, the value is hex 00. The hardware calculates the proper value based on the mode selected.

Bit	Function
7	Reserved
6	Reserved
5	Reserved
4	Reserved
3	Row Size3
2	Row Size2
1	Row Size1
0	Row Size0

Figure 1-33. Scan Lines per Character Register

Cursor Start Register, Index 0A: Bits 3 through 0 in this register determine the horizontal scan line count at which the cursor output becomes active. The cursor should always be programmed for a maximum height of eight scan lines (hex 07). The hardware will double scan the cursor to produce the proper cursor display for a 16 scan-line character box.

When bit 5 is 1, the cursor is not displayed.

Bit	Function
7	Reserved
6	Reserved
5	Blank Cursor
4	Reserved
3	Cursor Start3
2	Cursor Start2
1	Cursor Start1
0	Cursor Start0

Figure 1-34. Cursor Start Register

Cursor End Register, Index 0B: This register determines the horizontal scan line count when the cursor output becomes inactive. The cursor should always be programmed for a maximum height of eight scan lines (hex 07).

Bit	Function
7	Reserved
6	Reserved
5	Reserved
4	Reserved
3	Cursor End3
2	Cursor End2
1	Cursor End1
0	Cursor End0

Figure 1-35. Cursor End Register

Start of Screen High Register, Index 0C: This register contains the 8 most-significant bits for the starting memory address of the video display buffer. Sixteen address bits determine the starting address. This register is initialized to a value of hex 00.

Start of Screen Low Register, Index 0D: This register, together with the Start of Screen High register, gives the starting address of the display buffer. For all modes, this register is initialized to a value of hex 00.

Cursor Position High Register, Index 0E: This register contains the 4 most-significant bits for the cursor location.

Bit	Function
7	Reserved
6	Reserved
5	Reserved
4	Reserved
3	Cursor PositionB
2	Cursor PositionA
1	Cursor Position9
0	Cursor Position8

Figure 1-36. Cursor Position High Register

Cursor Position Low Register, Index 0F: This register contains the 8 least-significant bits for the location of the cursor. A value of hex 00 in both of these registers will locate the cursor in the upper left-hand corner. The cursor is not supported in any graphics mode.

Mode Control Register, Index 10: Writing to this register selects the type of display and clock times, and selects some of the graphics modes.

Bit	Function
7	Inhibit Write
6	Reserved = 0
5	Reserved
4	Clock = 1
3	Compatibility
2	Reserved
1	Mode 11
0	256 Color

Figure 1-37. Mode Control, Write

Write

Bit 7 When set to 1, the inhibit write bit prevents any writes to the horizontal and vertical registers. After a mode set, BIOS sets this bit to 1 to prevent applications designed for other color/graphics adapters from altering those registers.

Bit 6 The inverse of this bit is used as the ninth bit of the vertical compare circuits and must be set to 0.

Bit 5 Reserved.
Bit 4 This bit selects the dot clock and must be set to 1.
Bit 3 When set to 1, this bit allows the circuitry to calculate the correct horizontal register values for the 80-by-25 text modes. This bit should be set to 1 for all modes.
Bit 2 Reserved.
Bit 1 When set to 1, this bit selects mode 11.
Bit 0 When set to 1, this bit selects mode 13.

During certain operations, the circuitry calculates some of the internal signals and returns the values to the Mode Control register.

Bit	Function
7	80x25
6	Reserved
5	Clock Select
4	Clock
3	Alpha Mode
2	Double Scan
1	Reserved
0	Reserved

Figure 1-38. Mode Control, Read

Read
Bit 7 This bit indicates the state of bit 0 in the CGA Mode Control register.
Bit 6 Reserved.
Bit 5 When this bit is 1, it indicates that the clock is not divided by 2, and the resolution is 640 PELs wide. When it is 0, the resolution is 320.
Bit 4 When this bit is 1, it indicates that the dot clock is 25.175 MHz.
Bit 3 When set to 1, this bit indicates that the mode is a text mode.
Bit 2 When set to 1, this bit indicates that the scan lines are double scanned.
Bit 1 When set to 1, this bit indicates that mode 11 is selected.
Bit 0 When set to 1, this bit indicates that mode 13 is selected.

Interrupt Control Register, Index 11: This register controls IRQ2 output to the interrupt controller. It also shows the status of the interrupt. The output drivers are tri-stated (bit 7) to allow a Read of the Display Sense register.

Bit	Function
7	Tri-State Output
6	IRQ2 Status
5	-Enable IRQ2
4	-Clear IRQ2 Latch
3	Reserved
2	Reserved
1	Reserved
0	Reserved

Figure 1-39. Interrupt Control Register

Bit 7 When set to 1, this bit disables (tri-states) the output drivers and selects the Display Sense register to be read at index 12 instead of the Character Generator Interface and Sync Polarity register.

Bit 6 When set to 1, this bit indicates the memory controller is causing an interrupt. This bit is read-only.

Bit 5 When cleared to 0, this bit enables the interrupt.

Bit 4 When cleared to 0, this bit holds the interrupt latch clear.

Bits 3-0 These bits are reserved and should be 0.

Character Generator Interface and Sync Polarity Register, Index 12: This register controls the character font tables and the horizontal and vertical synchronization signals, HSYNC and VSYNC. To read this register, bit 7 of the Interrupt Control register must be 0.

Bit	Function
7	Load Character Generator
6	Load Full Character Set
5	Swap Active Font
4	Enable 512 Characters
3	Reserved = 0
2	Enable Sync Outputs
1	VSYNC Polarity
0	HSYNC Polarity

Figure 1-40. Character Generator Interface and Sync Polarity Register

Bit 7 When written as a 1, this bit loads the character generator. When reading a 0, the bit indicates that the load has finished. To start the load, this bit is first cleared and then set to 1.

Bit 6 When set to 1, this bit causes the character generator to load the display memory during normal display time. When clear, the display memory is loaded only during the vertical blanking interval.

Bit 5 This bit selects the font page that is used as font table or that the character generator loads. When set to 1, font page 1 is selected; when clear to 0, font page 0 is selected.

Bit 4 When this bit is set to 1, 512 character codes are displayable in the text modes. Bit 3 of the attribute byte then determines the font page when displaying the character. When this bit is set to 1, only eight foreground colors are supported. When this bit is cleared to 0, only 256 character codes are displayed, and bit 5 of this register determines the active font.

Bit 3 Reserved = 0.

Bit 2 When set to 1, this bit enables HSYNC and VSYNC outputs to the display.

Bit 1 When set to 1, this bit causes VSYNC to be positive polarity.

Bit 0 When set to 1, this bit causes HSYNC to be positive polarity.

Display Sense Register, Index 12: This register contains the sensed levels of the monitor sense 1 and 0 signals at pins 11 and 12 of the display connector. This information is used by BIOS to properly initialize all video registers to match the display. To read this register, bit 7 of the Interrupt Control register is set to 1.

These levels are used to determine the type of display attached as shown in the following. The bit is set when the polarity is positive.

Sense 1 Bit 1	Sense 0 Bit 0	Type of Display Attached
0	0	Reserved
0	1	Analog Monochrome Display
1	0	Analog Color Display
1	1	No Display Attached

Figure 1-41. Monitor Sense Bits

Character Font Pointer Register, Index 13: This register contains a pointer to the character font table. The only valid pointer values are hex 00, 10, 20, or 30. The pointer value doubled and the hex value A0000 make up the segment for the font table. The character value doubled is the offset into the table. See "RAM Loadable Fonts," later in this section.

Number of Characters to Load Register, Index 14: This register determines the number of characters to load into the RAM loadable character generator during one vertical retrace interval. This register is used only in the text modes.

Video Formatter Registers

The video formatter registers at I/O addresses hex 3D8 and 3D9 duplicate the functions of the 6845 registers in the color/graphics adapter. Registers are added at addresses hex 3DD through 3DF for Model 30 initialization requirements. The video formatter registers at addresses hex 3C6 through 3C9 control the color palette.

Register	Description
3D8	CGA Mode Control
3D9	CGA Border Control
3DA	CGA Status
3DB	Reserved
3DC	Reserved
3DD	Extended Mode Control
3DE	Reserved
3DF	Reserved
3C6	PEL Mask
3C7	Palette Read Address
3C8	Color Palette Address
3C9	Color Palette Data

CGA Mode Control Register, 3D8: This register contains the mode control information for color/graphics compatible functions.

Bit	Function
7	Reserved
6	Reserved
5	Enable Blink
4	640 x 200 Mono
3	Enable Video
2	B&W
1	Graphics
0	80x25 Alpha

Figure 1-42. CGA Mode Register

Bits 7,6 Reserved.

Bit 5 When set to 1, this bit selects the blink option for text modes. When cleared to 0, 16 background colors are available in the text modes.

Bit 4 When set to 1, this bit selects mode 6, 640-by-200 double-scanned graphics.

Bit 3 When set to 1, this bit enables display image.

Bit 2 When this bit is 1, palette address hex 00 and 07 are the two colors used in mode 6 and 11. When the bit is 0, address hex 00 and the address specified in the CGA Border Control register are the two colors used.

Bit 1 When set to 1, this bit selects modes 4 and 5, 320-by-200 double-scanned graphics.

Bit 0 When set to 1, this bit selects the 80-by-25 text mode.

CGA Border Control Register, 3D9: This register contains the border color information and selects the alternate color palette for modes 4 and 5. Although analog displays do not have borders, the border color information selects the alternate foreground color for modes 6 and 11, and the background color for modes 4 and 5.

Bit	Function
7	Reserved
6	Reserved
5	320 x 200 Palette Select
4	Alternate Intensity
3 to 0	Border Color

Figure 1-43. CGA Border Control Register

Bits 7,6 Reserved
Bit 5 When set to 1, this bit selects color set 1 for modes 4 and 5.
Bit 4 When set to 1 (default), this bit selects an intensified color set for modes 4 and 5.
Bits 3-0 These bits select the palette address for the border color information used by modes 4, 5, 6, and 11.

The following figure shows the effects of this register and the bit pair C1,C0 and how the two color sets map into the color palette.

BCR Bit 4	C1	C0	BCR Bit 5	Palette Address
X	0	0	X	Background color
0	0	1	0	02 Color Set 0
0	1	0	0	04 Color Set 0
0	1	1	0	06 Color Set 0
0	0	1	1	03 Color Set 1
0	1	0	1	05 Color Set 1
0	1	1	1	07 Color Set 1
				Intensified Colors
1	0	1	0	0A Color Set 0
1	1	0	0	0C Color Set 0
1	1	1	0	0E Color Set 0
1	0	1	1	0B Color Set 1
1	1	0	1	0D Color Set 1
1	1	1	1	0F Color Set 1

Figure 1-44. Modes 4 and 5 Color Selection

CGA Status Register, 3DA: This register is read-only and contains the status information for the color/graphics adapter.

Bit	Function
7	Reserved
6	Reserved
5	Reserved
4	Reserved
3	Vertical Sync
2	Reserved
1	Reserved
0	-Display Enable

Figure 1-45. Status Register

Extended Mode Control Register, 3DD: This register controls the selection of the type of display and the advanced color support. When cleared to 0, bit 7 indicates a readable DAC is installed; when set, it indicates that the DAC is not a readable type.

Bit	Function
7	-Readable DAC Installed
6	Reserved
5	Reserved
4	Reserved
3	Reserved
2	256 Colors
1	Reserved
0	Reserved = 0

Figure 1-46. Extended Mode Control Register

Color Palette Registers

Four registers are used to access the color palette: a mask register, a read address register, a write address register, and the data registers.

The color palette has 256 18-bit data registers and an 8-bit address register. Each data register is divided into three 6-bit data areas, one for each color. To load each data register takes three outputs in the sequence of red, green, blue.

When accessing the palette, the interrupts should be disabled to prevent the sequence from being interrupted. The palette supports both a single register write operation and a burst load operation.

To maintain software compatibility, programmers should use the BIOS interface when loading the color palette. BIOS supports two calls for setting and two calls for reading the color registers. The calls are through interrupt 10 with (AH) = hex 10. The value in the AL register determines the specific operation:

10 - Set individual color register
12 - Set block of color registers
15 - Read individual color register
17 - Read block of color registers

Single Register Load: The address for the specific color register (0 - 255) is loaded into the BX register. The DH, CH, and CL registers contain the red, green, and blue values, respectively. In the following example using the BIOS interface, the yellow color value is loaded into the palette address normally assigned to white. Because the Set Mode call restores the color palette to its default state, the mode must be set before changing the color palette.

```
;-----Set up the video mode

MOV     AX,0004H      ; Set mode to mode 4
INT     10H           ; Video BIOS interrupt

;-----Read color 14 to get the red, green, and blue values for yellow

MOV     AX,1015H      ; Read individual color register
MOV     BX,0EH        ; Read color register 0EH
INT     10H           ; Video BIOS interrupt
                      ; Return with DH = red value
                      ;             CH = green value
                      ;             CL = blue value

;-----Set color 15 to the red, green, and blue values of yellow

MOV     AX,1010H      ; Set individual color register
MOV     BX,0FH        ; Set color register 0FH
INT     10H           ; Video BIOS interrupt
```

Burst Load: This second call supports setting a block of color registers. Using this call, one to 256 color values can be set or read with a single BIOS call. The BX register contains the address for the first register to be set, and CX contains the number of registers. ES:DX point to a table of color values, where each table entry contains the red, green, and blue values for a color. The following example sets the first 16 colors in the color palette.

```
;-----Set colors 0 thru 15 with a set block of color registers call

CODE    SEGMENT 'CODE'
ASSUME  CS:CODE, ES:NOTHING, DS:NOTHING

SET_BLK_EX      PROC    FAR

PUSH    DS
XOR     AX,AX
PUSH    AX              ; Return address for DOS

PUSH    CS
POP     ES              ; Establish ES addressing for table
MOV     AX,1012H        ; Set block of color register call
MOV     BX,0            ; Start with color 0
MOV     CX,16                   ; Set 16 color registers
MOV     DX,OFFSET CLR_TABLE     ; ES:DX point to color table
INT     10H                     ; Make the video BIOS interrupt
RET

SET_BLK_EX  ENDP

CLR_TABLE       LABEL   BYTE

DB      00H,00H,00H         ; Black         00
DB      00H,00H,2AH         ; Blue          01
DB      00H,2AH,00H         ; Green         02
DB      00H,2AH,2AH         ; Cyan          03
DB      2AH,00H,00H         ; Red           04
DB      2AH,00H,2AH         ; Magenta       05
DB      2AH,15H,00H         ; Brown         06
DB      2AH,2AH,2AH         ; White         07
DB      15H,15H,15H         ; Gray          08
DB      15H,15H,3FH         ; Lt blue       09
DB      15H,3FH,15H         ; Lt green      0A
DB      15H,3FH,3FH         ; Lt cyan       0B
DB      3FH,15H,15H         ; Lt red        0C
DB      3FH,15H,3FH         ; Lt magenta    0D
DB      3FH,3FH,15H         ; Lt yellow     0E
DB      3FH,3FH,3FH         ; Bright White  0F

CODE    ENDS
        END
```

PEL Mask Register, 3C6: This register is initialized to a value that does not affect the color selection, hex FF. This value should not be changed because mask operations are not supported on the Model 30.

Palette Read Address Register, 3C7: This register contains the pointer to one of 256 palette data registers and is used when reading the color palette.

Reading this port returns the last command cycle to the palette. The description of bits 1 and 0 is in the following table. All other bits during a read of this port are reserved.

Bit 1	Bit 0	Last Palette Command
0	0	Write Palette Cycle
0	1	Reserved
1	0	Reserved
1	1	Read Palette Cycle

Figure 1-47. Last Palette Command

Color Palette Address Register, 3C8: This register contains the pointer to one of 256 palette data registers and is used during a palette load.

Color Palette Data Register, 3C9: This register contains a 6-bit value that yields 1 of 64 color levels. To write a color, the address is loaded into the Color Palette Address register. Three writes to this register are needed for each palette address: the first is the red color information, the second is the green, and the third is the blue.

To read a color, the address value is written to the Palette Read Address register, followed by three reads of this register. The first returns the red color information, the second returns the green, and the third returns the blue.

Bit	Function
7	Not Used
6	Not Used
5	PD 5
4	PD 4
3	PD 3
2	PD 2
1	PD 1
0	PD 0

Figure 1-48. Palette Data Register

Video Initialization Tables

The following figures show the video register values in BIOS for the various modes.

Index Pointer	Data Register Description	Modes 0,1	2,3	4,5	6	11	13
00	Horz. Total	30	30	30	30	30	30
01	Horz. Displayed	27	27	27	27	27	27
02	Start Horz. Sync	2A	2A	2A	2A	2A	2A
03	Sync Pulse width	26	26	26	26	26	26
04	Vert. Total	B0	B0	B0	B0	FF	B0
05	Vert. Adjust	0D	0D	0D	0D	0A	0D
06	Vert. Displayed	8F	8F	8F	8F	DF	8F
07	Start Vert. Sync	9B	9B	9B	9B	E9	9B
08	Reserved	XX	XX	XX	XX	XX	XX
09	Char. Scan Lines	07	07	01	01	00	00
0A	Cursor Scan Start	06	06	XX	XX	XX	XX
0B	Cursor Scan End	07	07	XX	XX	XX	XX
0C	Start of Screen (High)	00	00	00	00	00	00
0D	Start of Screen (Low)	00	00	00	00	00	00
0E	Cursor Position (High)	00	00	XX	XX	XX	XX
0F	Cursor Position (Low)	00	00	XX	XX	XX	XX
10	Mode Control	18	18	18	18	1A	19
11	Interrupt Control	30	30	30	30	30	30
12	Char. Gen/Sync Pol.	46	46	46	46	04	46
13	Char Font Pointer	00	00	XX	XX	XX	XX
14	Char to Load	FF	FF	XX	XX	XX	XX

Figure 1-49. Memory Controller Initialization

Address	Data Register Description	Modes 0,1	2,3	4,5	6	11	13
3C6	PEL Mask	FF	FF	FF	FF	FF	FF
3D8	CGA Mode Control	28	29	0A	18	18	08
3D9	CGA Border Control	30	30	30	3F	3F	30
3DA	Status	XX	XX	XX	XX	XX	XX
3DB	Reserved	XX	XX	XX	XX	XX	XX
3DC	Reserved	XX	XX	XX	XX	XX	XX
3DD	Ext Mode Control	00	00	00	00	00	04
3DE	Reserved						
3DF	Reserved						

Figure 1-50. Video Formatter Initialization Table

3C8 Index	R	3C9 G	B	Display Color
00	00	00	00	Black
01	00	00	2A	Blue
02	00	2A	00	Green
03	00	2A	2A	Cyan
04	2A	00	00	Red
05	2A	00	2A	Magenta
06	2A	15	00	Brown
07	2A	2A	2A	White
08	15	15	15	Gray
09	15	15	3F	Light Blue
0A	15	3F	15	Light Green
0B	15	3F	3F	Light Cyan
0C	3F	15	15	Light Red
0D	3F	15	3F	Light Magenta
0E	3F	3F	15	Yellow
0F	3F	3F	3F	Bright White

Figure 1-51. 16-Color Compatibility Initialization

RAM Loadable Fonts

In the text modes, the video buffer is divided into two data areas: the text area at address B8000 and the character font tables at address A0000. The text area consists of the character and attribute code for each position on the display. The font table consists of the character code and PEL data for each character in the set.

Restrictions are placed on where the character font can be loaded into the video buffer. Four fonts are supported in text modes. The memory map below shows the areas (blocks) in the video buffer where the fonts are loaded. The font tables can be swapped in synchronization with the 'vertical retrace' signal with several output commands. A maximum of four fonts can be loaded into font area, but only two can be loaded into and displayed from the character generator at any one time. Two fonts are provided in ROM, an 8-by-8 font and a 8-by-16 font. The font loaded depends on the mode that is active at the time.

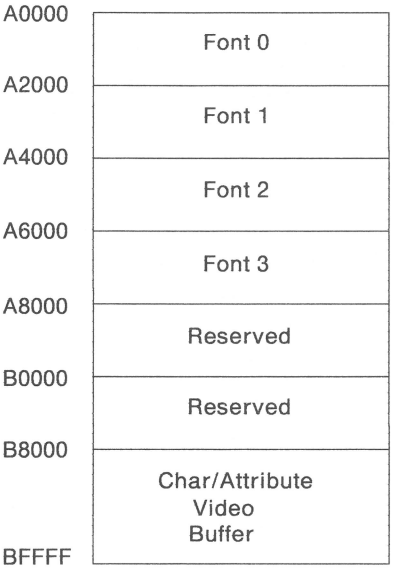


Figure 1-52. Font Memory Map

The following is an example of how the character "E" as defined in an 8-by-16 character box.

Scan Lines	Data in Hex	Data in Binary
0	00	00000000
1	00	00000000
2	7E	01111110
3	7E	01111110
4	60	01100000
5	60	01100000
6	7E	01111110
7	7E	01111110
8	60	01100000
9	60	01100000
10	7E	01111110
11	7E	01111110
12	00	00000000
13	00	00000000
14	00	00000000
15	00	00000000

Figure 1-53. Sample Character

The following programming example uses the BIOS routine to load a font table into block 0. Because of differences in the hardware, the character generator is not loaded the same for all display adapters; however, the BIOS routines are the same for all video subsystems with RAM-loadable fonts. The Model 30, for instance, supports only 8-by-8 and 8-by-16 character fonts depending on the mode selected.

```
TITLE   Load block 0 with character definitions from "SET_A"

CODE    SEGMENT PARA 'CODE'
        ASSUME  CS:CODE,ES:CODE

EX1     PROC  NEAR
        MOV   AX,0001H       ; Mode set BIOS call for mode 1
        INT   10H
        MOV   AH,11H         ; Character generator routines
        MOV   AL,00H         ; User alpha load BIOS call
        MOV   CX,100H        ; Load 256 characters into the block
        MOV   DX,0000H       ; Begin loading at offset zero
        MOV   BL,00H         ; Load the characters into block zero
        MOV   BH,10H         ; 16 bytes per character definition
        MOV   AX,SEG SET_A   ; Get the segment of the characters
        MOV   ES,AX          ; ES = segment of character definitions
        MOV   BP,OFFSET SET_A ; BP = offset of character definitions
        INT   10H
        RET
EX1     ENDP

;----8x16 definitions for "SET_A"

SET_A       LABEL   BYTE
            INCLUDE SET_A_CHARS
SET_A_END   EQU   $
CODE        ENDS
            END
```

Block 0 now contains the 256 character definitions from file SET_A. To load block 1, change the block number, the character file pointer, and the pointer for the block to be loaded, as indicated below.

```
EX2     PROC  NEAR
        MOV   AH,11H            ; Character generator routines
        MOV   AL,00H            ; User alpha load BIOS call
        MOV   CX,100H           ; Load 256 characters into the block
        MOV   DX,0000H          ; Begin loading at offset zero
        MOV   BL,01H            ; Load the characters into block one
        MOV   BH,10H            ; 16 bytes per character definition
        MOV   AX,SEG SET_B      ; Get the segment of the characters
        MOV   ES,AX             ; ES = segment of character definitions
        MOV   BP,OFFSET SET_B   ; BP = offset of character definitions
        INT   10H
        RET
EX2     ENDP

;----8x16 definitions for "SET_B"

SET_B       LABEL   BYTE
            INCLUDE SET_B_CHARS

SET_B_END   EQU   $
```

Blocks 2 and 3 can be loaded in the same manner, until all four blocks contain character font information. The characters that were loaded into the blocks are not available for display until they are transferred to the character generator.

The character generator is broken into two parts, or font pages. Each font page contains 256 character definitions. The character generator is loaded from the four blocks of 256 character definitions.

A character set of 256 characters is loaded into the character generator by selecting one of the four blocks to be transferred. Two of the four blocks are selected for a character set of 512 characters. The Set Block Specifier call is used to transfer the blocks of character definitions to the character generator.

The Set Block Specifier call uses the input parameter in BL to specify which blocks are loaded into the character generator. Only the low nibble (4 bits) of BL is used. Bits 1 and 0 specify which block to load into the first 256 positions of the character generator, or font page 0. The first 256 positions are the character definitions for characters 0 - 255. Bits 3 and 2 indicate which block to load into the second 256

positions of the character generator, or font page 1. The second 256 positions of the character generator define characters 256 - 511. If the two bit pairs are equal (bit 0 is the same as bit 2 and bit 1 is the same as bit 3) only font page 0 is loaded, which limits the character set to 256 characters. The following figure summarizes the bit patterns that indicate which blocks the character generator is loaded with.

Bit Number					
3	**2**	**1**	**0**	**Font Page 1**	**Font Page 0**
0	0	0	0	Not Used	Block 0
0	0	0	1	Block 0	Block 1
0	0	1	0	Block 0	Block 2
0	0	1	1	Block 0	Block 3
0	1	0	0	Block 1	Block 0
0	1	0	1	Not Used	Block 1
0	1	1	0	Block 1	Block 2
0	1	1	1	Block 1	Block 3
1	0	0	0	Block 2	Block 0
1	0	0	1	Block 2	Block 1
1	0	1	0	Not Used	Block 2
1	0	1	1	Block 2	Block 3
1	1	0	0	Block 3	Block 0
1	1	0	1	Block 3	Block 1
1	1	1	0	Block 3	Block 2
1	1	1	1	Not Used	Block 3

Figure 1-54. Block Specifier

To load block 0 into font page 0 and block 3 into font page 1, the following BIOS call is used.

```
MOV   AH,11H          ; Character generator routines
MOV   AL,03H          ; Set block specifier BIOS call
MOV   BL,0CH          ; Character generator block specifier
INT   10H
```

Font page 0 now contains the character definitions from block 0, and font page 1 the character definitions from block 3. Because font page 0 specifies characters 0 through 255, and font page 1 specifies the characters 256 thru 511, 512 characters are now available for display. The BIOS write character routines, however, accept the AL register as the character to be displayed. That allows a range of characters starting at 0 and stopping at 255, and appears to limit the number of characters to 256. The solution is to use a bit in the attribute byte to specify the font page (see "Programming Considerations" later in this section). Whenever a 512 character set is available, bit 3 of the

attribute byte selects between font page 0 (chars 0 - 255) and font page 1 (chars 256 - 511). If bit 3 is 1, font page 1 is used; if the bit is 0, font page 0 is used.

To display character hex 30, the following BIOS call could be used.

```
MOV   AH,09H        ; Write attribute/character at cursor pos.
MOV   AL,30H        ; AL = character to write
MOV   BH,00H        ; Display page 0
MOV   CX,1          ; Display 1 character

MOV   BL,07H        ; White character on black background
INT   10H           ; Attribute bit off selects font page 0
```

To display character hex 130 (304), the following BIOS call could be used. Attribute bit 3 is still used as the intensity bit in alpha modes.

```
MOV   AH,09H        ; Write attribute/character at cursor pos.
MOV   AL,30H        ; AL = character to write
MOV   BH,00H        ; Display page 0
MOV   CX,1          ; Display 1 character
MOV   BL,07H        ; Intense white character on black background
OR    BL,08H        ; Turn on attribute bit 3 to select font page 1
INT   10H
```

Alternate Parameter Table

A table in BIOS, SAVE _TBL, is used to maintain various tables and save areas. Each entry in this table is a doubleword. The format for this table is:

Entry	Description
1	Video Parameter Table Pointer This must point to the video parameter table in BIOS
2	Reserved = 0
3	Alpha Mode Auxiliary Font Pointer This is a pointer to a descriptor table used during a mode set to select a user font in A/N mode. The table has the following format: **Size** **Description** Byte Bytes per character Byte Block to load, should be 00 for normal operation Word Count to store, should be hex 100 for normal operation Word Character offset, should be 00 for normal operation DWord Pointer to a font table Byte Displayable rows, if the value is FF, the maximum calculated value will be used; otherwise, this value is used. Byte Consecutive bytes of mode values for which this font description is to be used. The end of this stream is indicated by a byte code of FF.
4	Graphics Mode Auxiliary Pointer This is a pointer to a descriptor table used during a mode set to select a user font in graphics mode. The table has the following format: **Size** **Description** Byte Displayable rows Word Bytes per character DWord Pointer to a font table Byte Consecutive bytes of mode values for which this font description is to be used. The end of this stream is indicated by a byte code of FF.
5 - 7	Reserved as all 0's.

Figure 1-55. Alternate Parameter Table

Normally, the auxiliary pointers, the third and fourth entries, are set to all zeroes. The Mode Set looks at these values and if they are zero, goes to the BIOS font table. If they are not zero, the Mode Set loads the user font pointed to by the auxiliary pointer.

The pointer for SAVE _TBL exists at 40:A8. When using the table, create the two tables, SAVE_TBL and the font descriptor table, and then set the pointer to point to the new SAVE_TBL.

Programming Considerations

Interrupt Usage: The Model 30 video subsystem can be programmed to create an interrupt at the end of each vertical display refresh time. An interrupt handler must be written by the application to take advantage of this feature. The vertical retrace interrupt is on IRQ2. (This interrupt does not support interrupt sharing).

The programmer can poll the Interrupt Control register, port 3D5 index 11, to determine if the video caused the interrupt. The IRQ2 status bit indicates that a vertical retrace interrupt did occur, and does not indicate that the video is still in retrace. To find the status of the 'vertical retrace' signal, check the Status register, port 3DA.

The Interrupt Control register also has 2 bits that control the interrupt circuitry and one bit that controls the outputs of the video formatter. To enable the interrupt:

1. Clear bit 4 to clear the interrupt latch.
2. Clear bit 5 to enable the interrupt.
3. Set bit 4 to enable the latch.

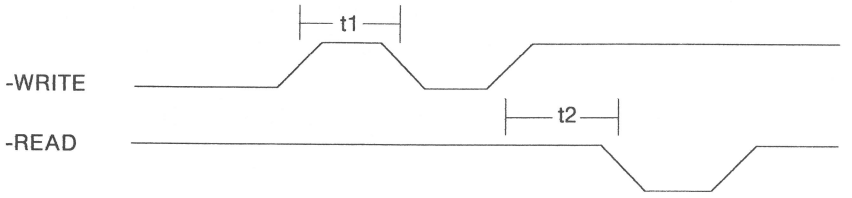
512 Character Set: When using a 512 character set on the Model 30, the following procedures are recommended to maintain consistent colors.

1. Set the block specifier, (AX) = 1103H.
2. Set the colors for 512, (AX) = 1000H (BX) = 0712H.
3. Reload the first eight colors into the palette.

Note: The character hex 20 (normally a space) is used to fill the blank area of the screen. Therefore, it is recommended that character hex 20 be a blank space.

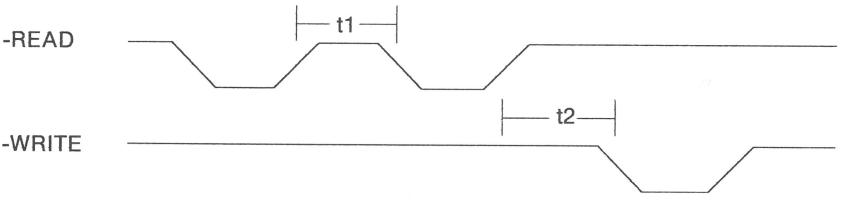
Color Palette: When the character generator is loaded during the vertical blanking interval, a maximum of 240 characters can be loaded in 80-column modes and 120 characters in 40-column modes.

To prevent screen flicker, the color palette should be accessed only during the vertical blanking interval. Also, when the palette is being accessed, certain timing requirements must be observed. The following diagrams show these timing requirements and their relationship to the type of display.



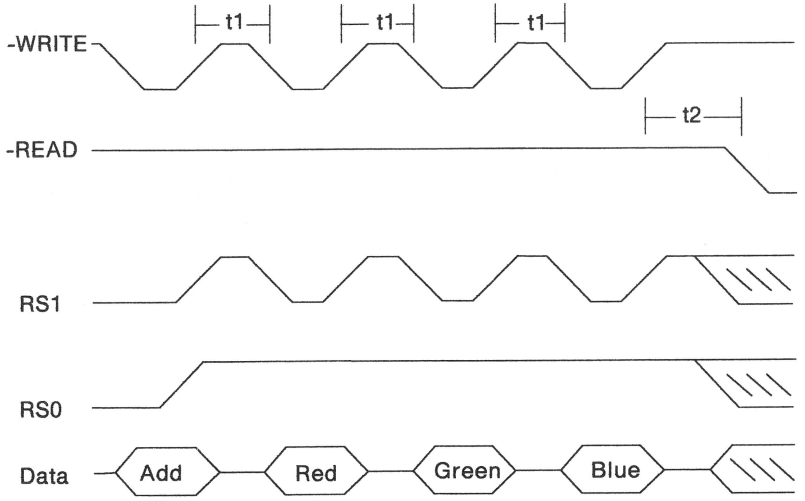
Symbol	Write to Register	40 column and 320 APA	80 column and 640 APA
t1	Followed by Write	240	120
t2	Followed by Read	240	120

Figure 1-56. Write to Palette Address Register



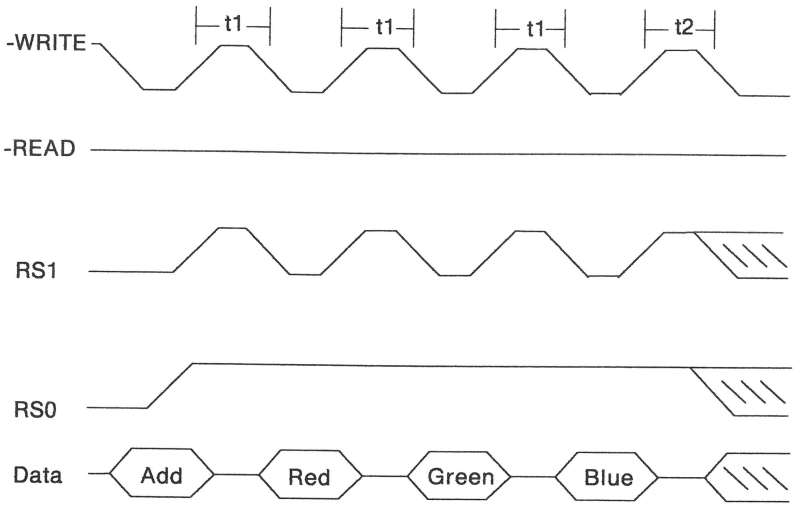
Symbol	Read from Register	40 column and 320 APA	80 column and 640 APA
t1	Followed by Read	240	120
t2	Followed by Write	240	120

Figure 1-57. Read Palette Address Register



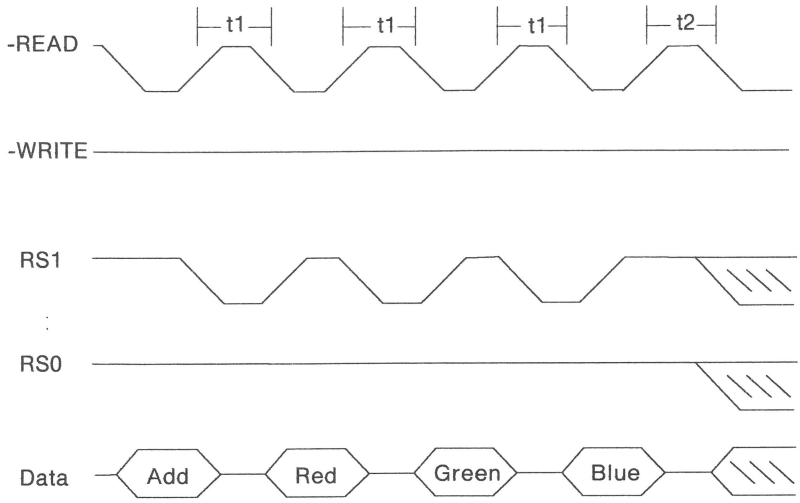
Symbol	Write Color	40 column and 320 APA	80 column and 640 APA
t1	Followed by Write Color	240	120
t2	Followed by any Read	240	120

Figure 1-58. Write Color followed by a Read



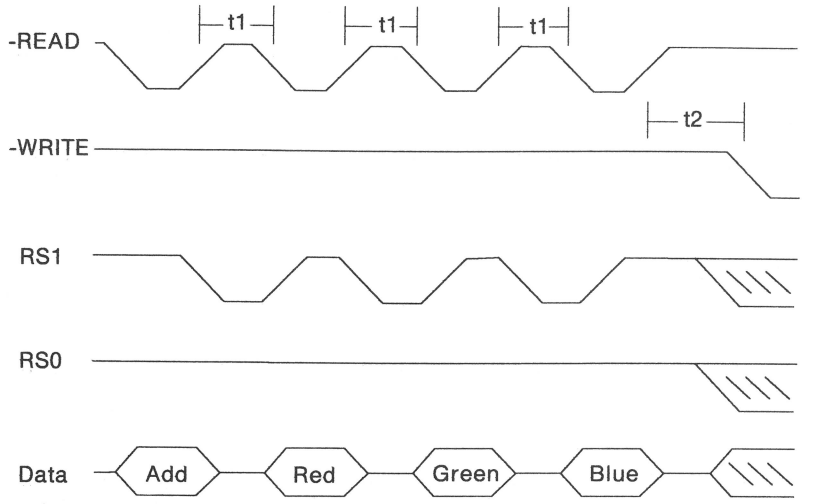
Symbol	Write Color	40 column and 320 APA	80 column and 640 APA
t1	Followed by Write Color	240	120
t2	Followed by any Write	240	120

Figure 1-59. Write Color followed by a Write



Symbol	Read Color	40 column and 320 APA	80 column and 640 APA
t1	Followed by Read Color	240	120
t2	Followed by any Read	480	240

Figure 1-60. Read Color followed by Read

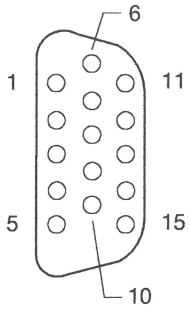


Symbol	Read Color	40 column and 320 APA	80 column and 640 APA
t1	Followed by Read Color	240	120
t2	Followed by any Write	480	240

Figure 1-61. Read Color followed by Write

Connector

The display connects to a 15-pin, subminiature D-shell connector in the rear of the system. The following are the pin numbering and signal assignments for the video connector.



Pin No.	Signal Name
1	Red
2	Green
3	Blue
4	Reserved
5	Reserved
6	Red Return
7	Green Return
8	Blue Return
9	Key
10	Ground
11	Monitor Sense 0
12	Monitor Sense 1
13	Horizontal Sync
14	Vertical Sync
15	Reserved

Figure 1-62. Display Connector

Diskette Drive Interface

The diskette gate array contains the decode logic for the internal registers, the write logic, and the read logic. The gate array:

- Controls the clock signals needed for read and write
- Controls write precompensation
- Selects the data rate of transfer
- Provides a mask for the interrupt and DMA request lines
- Provides phase error detection for input to the phase-lock loop.

The phase detector/amplifier and the VCO make up the phase-lock loop (PLL). They adjust the clock used during data read to keep it in phase with the data signal.

The drives connect to the system board through a single 40-pin connector, which supplies all signals necessary to operate two diskette drives. The diskette drives are attached to the connector through an internal, flat cable.

Gate Array Registers

The diskette drive gate array has five registers; three registers show the status of signals used in diskette operations, and two registers control certain interface signals.

RAS Port A Register: The RAS Port A register, hex 3F0, is a read-only register that shows the status of the corresponding signals.

Bit	Function
7	IRQ6
6	DRQ2
5	Step (latched)
4	Track 0
3	-Head 1 Select
2	Index
1	Write Protect
0	-Direction

Figure 1-63. RAS Port A, Hex 3F0

RAS Port B Register: The RAS Port B register, hex 3F1, is a read-only register that shows the status of signals between the diskette drive and the controller.

Bit	Function
7	Reserved
6	-Drive Select 1
5	-Drive Select 0
4	Write Data (latched)
3	Read Data (latched)
2	Write Enable (latched)
1	-Drive Select 3
0	-Drive Select 2

Figure 1-64. RAS Port B, Hex 3F1

Digital Output Register: The Digital Output register (DOR), hex 3F2, is a write-only register that controls drive motors, drive selection, and feature enables. All bits are cleared by a reset.

Bit	Function
7	Motor Enable 3
6	Motor Enable 2
5	Motor Enable 1
4	Motor Enable 0
3	DMA and Interrupt Enable
2	-Controller Reset
1,0	Drive Select 0 through 3 00 selects drive 0 01 selects drive 1 10 selects drive 2 11 selects drive 3

Figure 1-65. Digital Output, Hex 3F2

Digital Input Register: The Digital Input register, hex 3F7, is a read-only register used to sense the state of the 'diskette change' signal. It is also used for diagnostic purposes.

Bit	Function
7	-Diskette Change
6 to 4	Reserved
3	DMA Enable
2	No Write Precomp
1	250 Rate Select
0	Reserved

Figure 1-66. Digital Input, Hex 3F7

Configuration Control Register: The Configuration Control register, hex 3F7, is a write-only register used to set the transfer rate and select write precompensation.

Bit	Function
7	Reserved = 0
6	Reserved = 0
5	Reserved = 0
4	Reserved = 0
3	Reserved = 0
2	No Write Precomp
1	250 Rate Select
0	Reserved = 0

Figure 1-67. Configuration Control, Hex 3F7

Controller Registers

The diskette controller has two registers that are accessed by the microprocessor: the Main Status register and the data register. The Main Status register, hex 3F4, has the status information about the controller and may be read at any time.

Data Registers, Hex 3F5: This address, hex 3F5, consists of several registers in a stack with only one register presented to the data bus at a time. It stores data, commands and parameters, and provides diskette-drive status information. Data bytes are passed through the data register to program or obtain results after a command.

Main Status Register, Hex 3F4: This register is read-only and is used to facilitate the transfer of data between the microprocessor and the controller.

Bit	Function
7	Request for Master
6	Data Input/Output
5	Non-DMA Mode
4	Diskette Controller Busy
3, 2	Reserved
1	Drive 2 Busy
0	Drive 1 Busy

Figure 1-68. Main Status Register

The bits are defined as follows:

Bit 7 The data register is ready for transfer with the microprocessor.

Bit 6 This bit indicates the direction of data transfer between the diskette controller and the microprocessor. If this bit is set to 1, the transfer is from the controller to the microprocessor; if it is clear, the transfer is from the microprocessor.

Bit 5 When this bit is set to 1, the controller is in the non-DMA mode.

Bit 4 When this bit is set to 1, a Read or Write command is being executed.

Bits 3, 2 Reserved

Bit 1 Drive 2 Busy—When set to 1, diskette drive 2 is in the seek mode.

Bit 0 Drive 1 Busy—When set to 1, diskette drive 1 is in the seek mode.

Commands

The diskette controller performs the following commands. Each command is initiated by a multibyte transfer from the microprocessor, and the result can also be a multibyte transfer back to the microprocessor. Because of this multibyte interchange of information between the controller and the microprocessor, each command is considered to consist of three phases:

Command Phase: The microprocessor issues a series of Writes to the controller that direct it to perform a specific operation.

Execution Phase: The controller performs the specified operation.

Result Phase: After completion of the operation, status and other housekeeping information is made available to the microprocessor through a sequence of Read commands to the microprocessor.

The following is a list of controller commands.

- Read Data
- Read Deleted Data
- Read a Track
- Read ID
- Write Data
- Write Deleted Data
- Format a Track
- Scan Equal
- Scan Low or Equal
- Scan High or Equal
- Recalibrate
- Sense Interrupt Status
- Specify
- Sense Drive Status
- Seek

Symbol Descriptions: The following are descriptions of symbols used in the "Command Format" later in this section.

A0 Address Line 0—When clear, A0 selects the Main Status register; when set to 1, it selects the data register.

DTL Data Length—When N is 00, DTL is the data length to be read from or written to a sector.

EOT End of Track—The final sector number on a cylinder.

GPL Gap Length—The length of gap 3 (spacing between sectors excluding the VCO synchronous field).

H Head Address—The head number, either 0 or 1, as specified in the ID field.

HD Head—The selected head number, 0 or 1. (H = HD in all command words.)

HLT Head Load Time—The head load time in the selected drive (2 to 256 milliseconds in 2-millisecond increments).

HUT Head Unload Time—The head unload time after a read or write operation (0 to 240 milliseconds in 16-millisecond increments).

MF FM or MFM Mode—A 0 selects FM mode and a 1 selects MFM (MFM is selected only if it is implemented).

MT Multitrack—A 1 selects multitrack operation. (Both HD0 and HD1 will be read or written.)

N Number—The number of data bytes written in a sector.

NCN New Cylinder—The new cylinder number for a seek operation.

ND Nondata Mode— This indicates an operation in the nondata mode.

PCN Present Cylinder Number—The cylinder number at the completion of a Sense Interrupt Status command (present position of the head).

R Record—The sector number to be read or written.

SC Sector—The number of sectors per cylinder.

SK Skip—The skip deleted-data address mark.

SRT Stepping Rate—These four bits indicate the stepping rate for the diskette drive as follows:

1111 1 ms
1110 2 ms
1101 3 ms

ST0 - 3 Status 0 – Status 3—The four registers that store status information after a command is executed.

STP Scan Test—If STP is 01, the data in adjacent sectors is compared with the data sent by the microprocessor during a scan operation. If STP is 02, alternate sections are read and compared.

US0 - 1 Unit Select—The selected driver number encoded the same as bits 0 and 1 of the Digital Output register (DOR).

Command Format

The following are commands that may be issued to the controller. An X is used to indicate a don't-care condition.

Read Data

Command Phase

MT = Multitrack
MF = MFM Mode
SK = Skip Deleted-Data Address Mark
HD = Head Number
USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	SK	0	0	1	1	0
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
byte 6	End of Track							
Byte 7	Gap Length							
Byte 8	Data Length							

Figure 1-69. Read Data Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0							
Byte 1	Status Register 1							
Byte 2	Status Register 2							
Byte 3	Cylinder Number							
Byte 4	Head Address							
Byte 5	Sector Number							
Byte 6	Number of Data Bytes in Sector							

Figure 1-70. Read Data Result

Read Deleted Data

Command Phase

MT = Multitrack
MF = MFM Mode
SK = Skip Deleted-Data Address Mark
HD = Head Number
USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	SK	0	1	1	0	0
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End of Track							
Byte 7	Gap Length							
Byte 8	Data Length							

Figure 1-71. Read Deleted Data Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0							
Byte 1	Status Register 1							
Byte 2	Status Register 2							
Byte 3	Cylinder Number							
Byte 4	Head Address							
Byte 5	Sector Number							
Byte 6	Number of Data Bytes in Sector							

Figure 1-72. Read Deleted Data Result

Read a Track

Command Phase

MF = MFM Mode
SK = Skip Deleted-Data Address Mark
HD = Head Number
USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	0	MF	SK	0	0	0	1	0
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End of Track							
Byte 7	Gap Length							
Byte 8	Data Length							

Figure 1-73. Read a Track Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0							
Byte 1	Status Register 1							
Byte 2	Status Register 2							
Byte 3	Cylinder Number							
Byte 4	Head Address							
Byte 5	Sector Number							
Byte 6	Number of Data Bytes in Sector							

Figure 1-74. Read a Track Result

Read ID

Command Phase

MF = MFM Mode
HD = Head Number
USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	0	MF	0	0	0	1	1	0
Byte 1	X	X	X	X	X	HD	US1	US0

Figure 1-75. Read ID Command

Result Phase

	7 6 5 4 3 2 1 0
Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Cylinder Number
Byte 4	Head Address
Byte 5	Sector Number
Byte 6	Number of Data Bytes in Sector

Figure 1-76. Read ID Result

Write Data

Command Phase

MT = Multitrack
MF = MFM Mode
HD = Head Number
USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	0	0	0	1	1	0
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End of Track							
Byte 7	Gap Length							
Byte 8	Data Length							

Figure 1-77. Write Data Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0							
Byte 1	Status Register 1							
Byte 2	Status Register 2							
Byte 3	Cylinder Number							
Byte 4	Head Address							
Byte 5	Sector Number							
Byte 6	Number of Data Bytes in Sector							

Figure 1-78. Write Data Result

Write Deleted Data

Command Phase

MT = Multitrack
MF = MFM Mode
HD = Head Number
USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	0	0	0	1	1	0
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End of Track							
Byte 7	Gap Length							
Byte 8	Data Length							

Figure 1-79. Write Deleted Data Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0							
Byte 1	Status Register 1							
Byte 2	Status Register 2							
Byte 3	Cylinder Number							
Byte 4	Head Address							
Byte 5	Sector Number							
Byte 6	Number of Data Bytes in Sector							

Figure 1-80. Write Deleted Data Result

Format a Track

Command Phase

MF = MFM Mode
HD = Head Number
USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	0	MF	0	0	1	1	0	0
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Number of Data Bytes in Sector							
Byte 3	Sectors per Cylinder							
Byte 4	Gap Length							
Byte 5	Data							

Figure 1-81. Format a Track Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0							
Byte 1	Status Register 1							
Byte 2	Status Register 2							
Byte 3	Cylinder Number							
Byte 4	Head Address							
Byte 5	Sector Number							
Byte 6	Number of Data Bytes in Sector							

Figure 1-82. Format a Track Result

Scan Equal

Command Phase

MT = Multitrack
MF = MFM Mode
SK = Skip Deleted-Data Address Mark
HD = Head Number
USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	SK	1	0	0	0	1
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End of Track							
Byte 7	Gap Length							
Byte 8	Scan Test							

Figure 1-83. Scan Equal Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0							
Byte 1	Status Register 1							
Byte 2	Status Register 2							
Byte 3	Cylinder Number							
Byte 4	Head Address							
Byte 5	Sector Number							
Byte 6	Number of Data Bytes in Sector							

Figure 1-84. Scan Equal Result

Scan Low or Equal

Command Phase

MT = Multitrack
MF = MFM Mode
SK = Skip Deleted-Data Address Mark
HD = Head Number
USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	SK	1	1	0	0	1
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End of Track							
Byte 7	Gap Length							
Byte 8	Scan Test							

Figure 1-85. Scan Low or Equal Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0							
Byte 1	Status Register 1							
Byte 2	Status Register 2							
Byte 3	Cylinder Number							
Byte 4	Head Address							
Byte 5	Sector Number							
Byte 6	Number of Data Bytes in Sector							

Figure 1-86. Scan Low or Equal Result

Scan High or Equal

Command Phase

MT = Multitrack
MF = MFM Mode
SK = Skip Deleted-Data Address Mark
HD = Head Number
USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	MT	MF	SK	1	1	1	0	1
Byte 1	X	X	X	X	X	HD	US1	US0
Byte 2	Cylinder Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End of Track							
Byte 7	Gap Length							
Byte 8	Scan Test							

Figure 1-87. Scan High or Equal Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0							
Byte 1	Status Register 1							
Byte 2	Status Register 2							
Byte 3	Cylinder Number							
Byte 4	Head Address							
Byte 5	Sector Number							
Byte 6	Number of Data Bytes in Sector							

Figure 1-88. Scan High or Equal Result

Recalibrate

Command Phase: This command has no result phase.

USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	1	1	1
Byte 1	X	X	X	X	X	0	US1	US0

Figure 1-89. Recalibrate Command

Sense Interrupt Status

Command Phase

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

Figure 1-90. Sense Interrupt Status Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status Register 0							
Byte 1	Present Cylinder Number							

Figure 1-91. Sense Interrupt Status Result

Specify

Command Phase: This command does not have a result phase.

SRT = Diskette Stepping Rate
HUT = Head Unload Time
HLT = Head Load Time
ND = NonData Mode

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	0	1	1
Byte 1	SRT					HUT		
Byte 2	HLT							ND

Figure 1-92. Specify Command

Sense Drive Status

Command Phase

USx = Unit Select
HD = Head Number

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	0	1	0
Byte 1	X	X	X	X	X	HD	US1	US0

Figure 1-93. Sense Driver Status Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Status 3 Register							

Figure 1-94. Sense Driver Status Result

Seek

Command Phase

USx = Unit Select

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	1	1	1
Byte 1	X	X	X	X	X	0	US1	US0
Byte 2	New Cylinder Number for Seek							

Figure 1-95. Seek Command

Result Phase: This command has no result phase.

Invalid

Result Phase: The following status byte is returned to the microprocessor when an invalid command has been received.

	7	6	5	4	3	2	1	0
Byte 0	Status 0 Register							

Figure 1-96. Invalid Command Result

Command Status Registers

The following are definitions of the status registers ST0 through ST3.

Status 0 Register (ST0)

The following are bit definitions for the Status 0 register.

Bit 7, 6 Interrupt Code (IC)

- **00** Normal Termination of Command—The command was completed and properly executed.
- **01** Abrupt Termination of Command—The execution of the command was started but not successfully completed.
- **10** Invalid Command Issue—The issued command was never started.
- **11** Abnormal Termination—During the execution of a command, the 'ready' signal from the diskette drive changed state.

Bit 5 Seek End—Set to 1 when the controller completes the Seek command.

Bit 4 Equipment Check—Set if a 'fault' signal is received from the diskette drive, or if the 'track 0' signal fails to occur after 77 step pulses (Recalibrate command).

Bit 3 Not Ready—This flag is set when the diskette drive is in the not-ready state and a Read or Write command is issued.

Bit 2 Head Address—Indicates the state of the head at interrupt.

Bit 1, 0 Unit select 1 and 0 (US 1 and 0)—Indicate a drive's unit number at interrupt.

Status 1 Register (ST1)

The following are bit definitions for the Status 1 register.

Bit 7 End of Cylinder—Set when the controller tries to gain access to a sector beyond the final sector of a cylinder.

Bit 6 Reserved.

Bit 5 Data Error—Set when the controller detects a CRC error in either the ID field or the data field.

Bit 4 Overrun—Set if the controller is not serviced by the main system within a certain time limit during data transfers.

Bit 3 Reserved.

Bit 2 No Data—Set if the controller cannot find the sector specified in the ID register during the execution of a Read Data, Write Deleted Data, or Scan command. This flag is also set if the controller cannot read the ID field without an error during the execution of a Read ID command, or if the starting sector cannot be found during the execution of a Read Cylinder command.

Bit 1 Not Writable—Set if the controller detects a 'write-protect' signal from the diskette drive during execution of a Write Data, Write Deleted Data, or Format a Track command.

Bit 0 Missing Address Mark—Set if the controller cannot detect the ID address mark. At the same time, bit 0 of the Status 2 register is set.

Status 2 Register (ST2)

The following are bit definitions for the Status 2 register.

Bit 7 Reserved = 0.

Bit 6 Control Mark—This flag is set if the controller encounters a sector that has a deleted data-address mark during execution of a Read Data or Scan command.

Bit 5 Data Error in Data Field—Set if the controller detects an error in the data.

Bit 4 Wrong Cylinder—This flag is related to ND and is set when the content of C is different from that stored in the ID register.

Bit 3 Scan Equal Hit (SH)—Set if the adjacent sector data equals the microprocessor data during the execution of a Scan command.

Bit 2 Scan Not Satisfied (SN)—Set if the controller cannot find a sector on the cylinder that meets the condition during a Scan command.

Bit 1 Bad Cylinder—Related to ND and is set when the contents of C on the medium are different from that stored in the ID register or when the content of C is hex FF.

Bit 0 Missing Address Mark in Data Field— Set if the controller cannot find a data address mark or a deleted data address mark when data is read.

Status 3 Register (ST3)

The following are bit definitions for the Status 3 register.

Bit 7 Fault—Status of the 'fault' signal from the diskette drive.

Bit 6 Write Protect—Status of the '-write protect' signal from the diskette drive.

Bit 5 Ready—Status of the 'ready' signal from the diskette drive.

Bit 4 Track 0—Status of the '-track 0' signal from the diskette drive.

Bit 3 Two Side—Status of the 'two side' signal from the diskette drive.

Bit 2 Head Address—Status of the '-head 1 select' signal from the diskette drive.

Bit 1 Unit Select 1—Status of the '-drive select 1' signal from the diskette drive.

Bit 0 Unit Select 0—Status of the '-drive select 0' signal from the diskette drive.

Signal Description

All signals are 74HCT series compatible in both rise and fall times as well as interface levels. The following are the input signals to the diskette drive. These signals are +2.0 Vdc high and +0.8 Vdc low.

—**Drive Select 0—1:** The select lines provide the means to enable or disable the drive interface lines. When the signal is active, the drive is enabled. When the signal is inactive, all control inputs are ignored, and the drive outputs are disabled. The maximum drive-select delay time is 500 ns.

—**Motor Enable 0—1:** When this signal is made active, the spindle starts to turn. When it is made inactive, the spindle slows to a stop.

—**High Density Select:** When this signal is active, the drive is in the high density mode.

—**Step:** An active pulse on this line causes the head to move one track. The minimum pulse width is 1 μs. The direction of the head motion is determined by the state of the '-direction' signal at the trailing edge of the '-step' pulse.

—**Direction:** When this signal is active, the head moves to the next higher track (toward the spindle) for each '-step' pulse. When the signal is inactive, the head moves toward track 0. This signal must be stable for 1 μs before and after the trailing edge of the '-step' pulse.

—**Head 1 Select:** When this signal is active, the upper head (head 1) is selected. When it is inactive, the lower head (head 0) is selected.

—**Write Enable:** When this signal is active, the write-current circuits are enabled and data can be written under the control of the '-write data' signal. This signal must be active 8 μs before data can be written.

– **Write Data:** An active pulse on this line, writes a 1. These pulses have a 4-, 6-, or 8-μs spacing with a width of 250 ns for the 250,000-bps transfer rate. Write precompensation of 125 ns is done by the diskette gate array.

The following are the output signals from the diskette drive. These signals are +3.7 Vdc high and +0.4 Vdc low.

– **Index:** An active pulse of 1 ms indicates the diskette index.

– **Track 0:** When this signal is active, the head is on track 0. This signal is used to determine if the drive is present. The drive seeks track 0. If the '-track 0' signal does not go active, the drive is not present.

– **Write Protect:** This signal is active when the write-protect window is uncovered. When this happens, the write current circuits are disabled.

– **Read Data:** An active pulse on this line, writes a logical 1. The pulse width for the 250,000-bps rate is 250 ns.

– **Diskette Change:** This signal is active at power-on and whenever the diskette is removed. It remains active until a diskette is present and a '-step' pulse is received.

Connector

The following shows the signals and pin assignments for the connector.

Pin	I/O	Signal	Pin	I/O	Signal
1	N/A	Signal Ground	2	O	-High Density Select
3	N/A	Reserved	4	N/A	Reserved
5	N/A	Signal Ground	6	N/A	Reserved
7	N/A	Signal Ground	8	I	-Index
9	N/A	Signal Ground	10	O	-Motor Enable 1
11	N/A	Signal Ground	12	O	-Drive Select 0
13	N/A	Signal Ground	14	O	-Drive Select 1
15	N/A	Signal Ground	16	O	-Motor Enable 0
17	N/A	Signal Ground	18	O	-Direction
19	N/A	Signal Ground	20	O	-Step
21	N/A	Signal Ground	22	O	-Write Data
23	N/A	Signal Ground	24	O	-Write Enable
25	N/A	Signal Ground	26	I	-Track 0
27	N/A	Signal Ground	28	I	-Write Protect
29	N/A	Signal Ground	30	I	-Read Data
31	N/A	Signal Ground	32	O	-Head 1 Select
33	N/A	Signal Ground	34	I	-Diskette Change
35	N/A	Ground	36	N/A	Ground
37	N/A	Ground	38	O	+5 Vdc
39	N/A	Ground	40	O	+12 Vdc

Figure 1-97. Diskette Drive Connector

Fixed Disk Connector

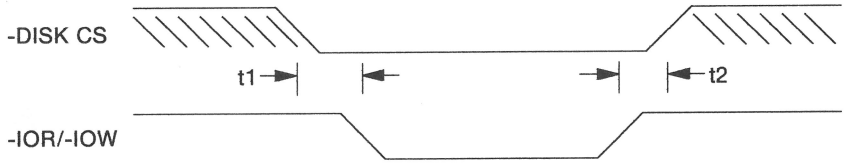
The Model 30 provides a dedicated I/O channel for the connection of the IBM Personal System/2 20MB Fixed Disk Drive and Controller, or similar attachment. The signals across this connector are the normal I/O channel signals needed for fixed disk operation: I/O read and write, reset, data lines, IO CH RDY, IRQ5, and the DMA request and acknowledge lines. These signals operate the same as the normal I/O channel signals described earlier. The additional signals are as follows:

—**Disk Card Select (** -DISK CS**):** The address decode logic for the fixed disk is on the system board. It is enabled through the Planar Control register (see "Chip Select Logic" earlier in this section). When the logic is enabled, -DISK CS goes active on a valid decode of A4 through A19 equal to hex 032x.

—**DISK Installed:** When active, this signal indicates that a fixed disk and its controller are installed.

Address 0 through 2 (A0-A2): These three address lines are used to select the specific register within the attachment.

The following shows the signal timing for -DISK CS. The other signal timings are the same as those on the I/O channel.



Symbol	Parameter Description	Min (ns)
t1	-DISK CS active to Command active	25
t2	-DISK CS inactive to Command inactive	45

Figure 1-98. Fixed Disk Signal Timing

The following shows the signal assignments for the fixed disk connector.

Pin	I/O	Signal	Pin	I/O	Signal
1	O	RESET DRV	2	I	-DISK Installed
3	I/O	D0	4	N/A	Ground
5	I/O	D1	6	N/A	Ground
7	I/O	D2	8	N/A	Ground
9	I/O	D3	10	N/A	Ground
11	I/O	D4	12	N/A	Ground
13	I/O	D5	14	N/A	Ground
15	I/O	D6	16	N/A	Ground
17	I/O	D7	18	N/A	Ground
19	O	-IOR	20	N/A	Ground
21	O	-IOW	22	N/A	Ground
23	O	-DISK CS	24	N/A	Ground
25	O	A0	26	N/A	Ground
27	O	A1	28	N/A	Ground
29	O	A2	30	O	+5
31	N/A	Reserved	32	O	+5
33	O	-DACK3	34	N/A	Ground
35	I	DRQ3	36	N/A	Ground
37	I	IRQ5	38	N/A	Ground
39	I	IO CH RDY	40	O	+12
41	N/A	Spare	42	O	+12
43	N/A	Spare	44	O	+12

Figure 1-99. Fixed Disk Connector

Serial Port

The serial port is fully programmable and supports asynchronous communications. It will add and remove start, stop, and parity bits. A programmable baud-rate generator allows operation from 50 baud to 9600 baud. The port supports 5-, 6-, 7-, and 8-bit characters with 1, 1.5, or 2 stop bits. A prioritized interrupt system controls transmit, receive, error, and line status as well as data-set interrupts.

The rear of the system unit has a 25-pin D-shell connector that contains standard EIA RS-232C interface signals.

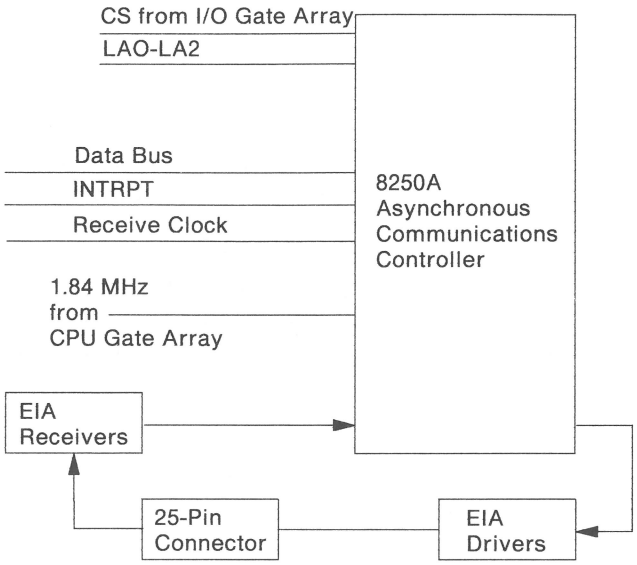


Figure 1-100. Serial Port Block Diagram

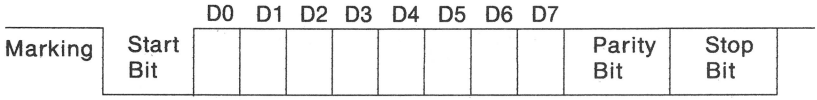
The serial port has a controller that provides the following functions:

- Adds or deletes standard, asynchronous communications bits to or from a serial data stream.
- Provides full, double buffering, which eliminates the need for precise synchronization.
- Provides a programmable baud-rate generator.
- Provides modem controls (CTS, RTS, DSR, DTR, RI, and CD).

Application

The serial port is addressed as communications port 1, addresses hex 3F8 through 3FF; the port uses interrupt 4.

The data format is:



Data bit 0 is the first bit to be sent or received. The controller automatically inserts the start bit, the correct parity bit (if programmed to do so), and the stop bit (1, 1.5, or 2, depending on the command in the Line Control register).

Controller Registers

The register addresses are hex 3F8 through 3FF. These registers control the controller's operations and are used to transmit and receive data. The divisor latch access bit (DLAB), which is the most-significant bit of the Line Control register, affects the selection of the divisor latches for the baud rate generator.

Specific registers are selected according to the following figure:

DLAB State	Port 1 Address	R/W	Register
0	03F8	W	Transmitter Holding
0	03F8	R	Receiver Buffer
1	03F8	R/W	Divisor Latch, Low Byte
1	03F9	R/W	Divisor Latch, High Byte
0	03F9	R/W	Interrupt Enable Register
X	03FA	R	Interrupt Identification Register
X	03FB	R/W	Line Control Register
X	03FC	R/W	Modem Control Register
X	03FD	R	Line Status Register
X	03FE	R	Modem Status Register
X	03FF	R/W	Scratch Register

Figure 1-101. Serial Port Addresses

Transmitter Holding Register, Hex 3F8: The Transmitter Holding register contains the character to be sent. Bit 0 is the least-significant bit and the first bit sent serially.

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

Figure 1-102. Transmitter Holding Register

Receiver Buffer Register, Hex 3F8: The Receiver Buffer register (RBR) contains the received character. Bit 0 is the least-significant bit and the first bit received serially.

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

Figure 1-103. Receiver Buffer Register

Divisor Latch, 3F9 and 3F8: These two registers access the high byte (3F9) and low byte (3F8) of the divisor latch. More information about the divisor latch may be found under "Programmable Baud-Rate Generator" later in this section.

Bit	High Byte	Low Byte
7	Bit 15	Bit 7
6	Bit 14	Bit 6
5	Bit 13	Bit 5
4	Bit 12	Bit 4
3	Bit 11	Bit 3
2	Bit 10	Bit 2
1	Bit 9	Bit 1
0	Bit 8	Bit 0

Figure 1-104. Divisor Latch

Interrupt Enable Register, Hex 3F9: This register allows the four types of controller interrupts to separately activate the 'chip-interrupt' (INTRPT) output signal. The interrupt system can be totally disabled by resetting bits 3 through 0 of the Interrupt Enable register (IER) to 0. Similarly, by setting the appropriate bits of this register to 1, selected interrupts are enabled. Disabling the interrupt system inhibits the IER and the active INTRPT output from the chip. All other system functions operate normally, including the setting of the Line Status and Modem Status registers.

Bit	Function
7	Reserved = 0
6	Reserved = 0
5	Reserved = 0
4	Reserved = 0
3	Enable Modem Status
2	Enable Rx Line Status
1	Enable Tx Buffer Empty
0	Enable Data Available

Figure 1-105. Interrupt Enable Register

Bits 7-4 Reserved = 0.

Bit 3 When set to 1, this bit enables the modem status interrupt.

Bit 2 When set to 1, this bit enables the receiver-line-status interrupt.

Bit 1 When set to 1, this bit enables the transmitter-holding-register-empty interrupt.

Bit 0 When set to 1, this bit enables the received-data-available interrupt.

Interrupt Identification Register, Hex 3FA: The controller has an internal interrupt capability that makes communications possible with reduced microprocessor intervention. To minimize programming overhead during data character transfers, the controller prioritizes interrupts into four levels: receiver line status (priority 1), received data ready (priority 2), transmitter holding register empty (priority 3), and modem status (priority 4).

Information about a pending interrupt is stored in the Interrupt Identification register (IIR). The IIR, when addressed during chip-select time, stops the pending interrupt with the highest priority, and no other interrupts are acknowledged until the microprocessor services that particular interrupt.

Bit	Function
7	Reserved = 0
6	Reserved = 0
5	Reserved = 0
4	Reserved = 0
3	Reserved = 0
2	Interrupt I/O Bit 1
1	Interrupt I/O Bit 0
0	Interrupt Not Pending

Figure 1-106. Interrupt Identification Register

Bits 7-3 Reserved = 0.

Bits 2,1 These two bits, Interrupt ID 1 and 0, identify the pending interrupts as shown.

IIR Bits				Interrupt Control	
2	1	Priority	Type	Cause	To Reset
1	1. 1	Highest	Receiver Line Status	Overrun, Parity, or Framing Error, or Break Interrupt	Read the Line Status Register
1	0	Second	Received Data Available	Data in Receiver Buffer	Read the Receiver Buffer Register
0	1	Third	Transmitter Holding Register Empty	THR is Empty	Read IIR or Write to THR
0	0	Fourth	Modem Status	Change in a Signal's Status from the Modem	Read the Modem Status Register

Bit 0 This bit can be used in either hard-wired, prioritized, or polled conditions to indicate if an interrupt is pending. When bit 0 is 0, an interrupt is pending, and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is a 1, no interrupt is pending, and polling (if used) continues.

Line Control Register, Hex 3FB: This register specifies the format of the asynchronous data communications exchange. The register can also be read at any time, eliminating the need to store line characteristics separately in memory.

Bit	Function
7	DLAB
6	Set Break
5	Stick Parity
4	Even Parity Select
3	Parity Enable
2	Number of Stop Bits
1	Word Length Select 1
0	Word Length Select 0

Figure 1-107. Line Control Register

Bit 7 This is the divisor-latch access bit. It is set to 1 to gain access to the divisor latches of the baud-rate generator during a read or write operation. It is cleared to gain access to the Receiver Buffer, the Transmitter Holding, or the Interrupt Enable registers.

Bit 6 This bit is the set-break control bit. When bit 6 is set to 1, the serial output is forced to an inactive level and remains there regardless of other transmitter activity. The set-break is disabled by clearing bit 6 to 0.

Bit 5 This bit is the stick-parity bit. When this bit is set to 1 and parity is enabled, the parity bit is sent as a 0 if parity is even, or as a 1 if parity is odd.

Bit 4 This bit is the even-parity-select bit. When set to 1 and parity is enabled, an even number of logical 1's is sent or checked. When cleared to 0, an odd number of bits is sent or checked.

Bit 3 This bit is the parity-enable bit. When this bit is set to 1, a parity bit is sent or checked. The parity bit is used to produce an even or odd number of 1's when the bits in the data word and the parity bit are summed.

Bit 2 This bit specifies the number of stop bits in each serial character that is sent or received. When set to 1 and a word length greater than 5 is specified, 2 stop bits are generated or checked. If the word length is 5 and this bit is set to 1, 1.5 stop bits are generated or checked. When this bit is cleared to 0, 1 stop bit is specified.

Bits 1,0 These 2 bits specify the number of bits in each serial character that is sent or received. The encoding of these bits is as follows:

Bit 1	Bit 0	Word Length in Bits
0	0	5
0	1	6
1	0	7
1	1	8

Modem Control Register, Hex 3FC: The Modem Control register (MCR) controls the output signals to the modem or data set (an external device acting as a modem).

Bit	Function
7 to 5	Reserved = 0
4	Loop
3	Out 2
2	Out 1
1	Request to Send
0	Data Terminal Ready

Figure 1-108. Modem Control Register

Bits 7-5 Reserved = 0.

Bit 4 This bit provides a loopback feature for diagnostic testing of the controller. When this bit is set to 1, the following events occur:

- SOUT is set to the active state.
- SIN is disconnected.
- The output of the Transmitter Shift register is "looped back" to the Receiver Shift register input.
- The four modem-control inputs (-DSR, -CTS, -RLSD, and -RI) are disconnected.
- The four modem-control outputs (-DTR, -RTS, -OUT 1, and -OUT2) are internally connected to the four modem control inputs.

In the diagnostic mode, data sent is immediately received. This feature allows the microprocessor to verify the transmit- and receive-data paths of the controller.

In the diagnostic mode, the receiver and transmitter interrupts are fully operational, as are the modem-control interrupts, however, the interrupts' sources are now the

lower 4 bits of the Modem Control register instead of the four modem-control inputs. The interrupts are still controlled by the Interrupt Enable register.

The controller's interrupt system can be tested by writing to the lower 6 bits of the Line Status register and the lower 4 bits of the Modem Status register. Setting any of these bits to 1 generates the appropriate interrupt (if enabled). Resetting these interrupts is the same as for normal controller operation. To return to normal operation, the registers must be reprogrammed for normal operation, and then bit 4 of the MCR is cleared to 0.

Bit 3 This bit controls -OUT 2; when set to 1, it forces -OUT 2 active.

Bit 2 This bit controls -OUT 1; when set to 1, it forces -OUT 1 active.

Bit 1 This bit controls -RTS; when set to 1, it forces -RTS active.

Bit 0 This bit controls -DTR; when set to 1, it forces -DTR active.

Line Status Register, Hex 3FD: This register provides the microprocessor with status information about the data transfer.

Bit	Function
7	Reserved = 0
6	Tx Empty
5	Transmitter Holding Empty
4	Break Interrupt
3	Framing Error
2	Parity Error
1	Overrun Error
0	Data Ready

Figure 1-109. Line Status Register

Bit 7 Reserved = 0.

Bit 6 This bit is the transmitter empty indicator. It is set to 1 whenever the Transmitter Holding register and the Transmitter Shift register are both empty.

Bit 5 This bit is the transmitter holding register empty (THRE) indicator. It indicates the controller is ready to accept a new character for transmission. In addition, this bit causes the controller to issue an interrupt to the microprocessor when the THRE interrupt is enabled. The THRE bit is set to 1 when a character is transferred from the Transmitter Holding register into the Transmitter Shift register. It is cleared when the microprocessor loads the Transmitter Holding register.

Bit 4 This bit is the break interrupt indicator. It is set to 1 whenever the received data input is held in the spacing state (0) for longer than a full-word transmission time (that is, the total time of start bit + data bits + parity stop bits).

Note: Bits 1 through 4 are error conditions that produce a receiver line-status interrupt whenever any of the corresponding conditions are detected.

Bit 3 This bit is the framing error indicator. It indicates the received character did not have a valid stop bit. Bit 3 is set to 1 whenever the stop bit is detected as a 0 (spacing level).

Bit 2 This bit is the parity error indicator and indicates the received data character does not have the correct even or odd parity. The parity error bit is set to 1 on a parity error, and is cleared when the Line Status register is read.

Bit 1 This bit is the overrun error indicator. It indicates that data in the Receiver Buffer register was not read by the microprocessor before the next character was transferred into the register, thereby destroying the previous character. This indicator is cleared when the microprocessor reads the contents of the Line Status register.

Bit 0 This bit is the receiver data ready indicator. It is set to 1 whenever a complete incoming character has been received and transferred into the Receiver Buffer register.

Modem Status Register, Hex 3FE: This register provides the current state of the control lines from the modem (or external device) to the microprocessor. In addition, 4 bits provide change information. They are set whenever a control input from the modem changes state; they are cleared when the microprocessor reads this register.

Bit	Function
7	Data Carrier Detect
6	Ring Indicator
5	Data Set Ready
4	Clear to Send
3	Delta Data Carrier Detect
2	Trailing Edge Ring Indicator
1	Delta Data Set Ready
0	Delta Clear to Send

Figure 1-110. Modem Status Register

Bit 7 When set to 1, this bit indicates -DCD is active. In the diagnostic mode, this bit is equivalent to OUT 2 of the Modem Control register.

Bit 6 When set to 1, this bit indicates -RI is active. In the diagnostic mode, this bit is equivalent to OUT 1 of the Mode Control register.

Bit 5 When set to 1, this bit indicates -DSR is active. In the diagnostic mode, this bit is equivalent to DTR of the Mode Control register.

Bit 4 When set to 1, this bit indicates -CTS is active. In the diagnostic mode, this bit is equivalent to RTS of the Mode Control register.

Bit 3 This bit is the delta data-carrier-detect indicator. It indicates -DCD to the chip has changed state.

Note: Whenever bit 0, 1, 2, or 3 is set, a modem status interrupt is generated.

Bit 2 This bit is the trailing-edge ring-indicate indicator. It indicates that -RI to the chip has changed from active to inactive.

Bit 1 This bit is the delta data-set-ready indicator. It indicates that -DSR to the chip has changed state.

Bit 0 This bit is the delta clear-to-send indicator. It indicates that -CTS to the chip has changed state.

Programmable Baud-Rate Generator

The controller has a programmable baud-rate generator that can divide the clock input (1.84 MHz) by any divisor from 1 to 655,535 or 2^{16}-1. The output frequency of the baud-rate generator is the baud rate multiplied by 16. Two 8-bit latches store the divisor in a 16-bit binary format. These divisor latches are loaded during setup to ensure desired operation of the baud-rate generator. When either of the divisor latches is loaded, a 16-bit baud counter is immediately loaded. This prevents long counts on the first load.

The following is a sample program that sets the baud rate at 1200 with an 8-bit data word, 1 stop bit, and odd parity.

```
BEGIN   PROC    NEAR
        MOV     AL,8BH          ; Set port parameters
        MOV     AH,00H          ; Initialize COM1 port
        INT     14H             ; Serial port BIOS interrupt

        ENDP
```

Signal Descriptions

The following describes the function of controller input/output signals.

Input Signals

-Clear to Send: (-CTS)—This signal is an input from the modem. The status of this signal is reflected in bit 4 of the Modem Status register. Bit 0 of the same register indicates whether -CTS has changed state since the last reading.

-Data Set Ready: (-DSR)—When active, this signal indicates the modem or data set is ready to establish the communications link and transfer data with the controller. This signal is an input from the modem. Its status is reflected in bit 5 of the Modem Status register. Bit 1 of the same register indicates whether this signal has changed state since the last reading.

Note: Whenever the bit 5 of the Modem Status register changes state, an interrupt is generated if the modem status interrupt is enabled.

-Data Carrier Detect: (-DCD)—When active, this signal indicates the modem or data set detected a data carrier. This signal is an input from the modem. Its status is reflected in bit 7 of the Modem Status register. Bit 3 of the same register indicates whether the signal has changed state since the last reading.

-Ring Indicate: (-RI)—When active, this signal indicates the modem or data set detected a telephone ringing signal. This signal is an input from the modem. Its status is reflected in bit 6 of the Modem Status register. Bit 2 of the same register indicates whether the signal has changed from active to inactive.

Note: Whenever the bit 6 of the Modem Status register changes from 0 to 1, an interrupt is generated if the modem status interrupt is enabled.

Output Signals

-Data Terminal Ready: (-DTR)—When active, this signal informs the modem or data set that the controller is ready to communicate. This signal can be made active by setting bit 0 of the Modem Control register. It is inactive after a master reset operation.

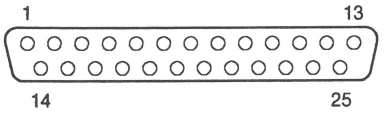
-Request to Send: (-RTS)—When active, this signal informs the modem or data set that the controller is ready to send data. It can be made active by setting bit 1 of the Modem Control register. This signal is inactive after a master reset operation.

-Output 1: (-OUT 1)—User-designated output that can be made active by setting bit 2 of the Modem Control register. -OUT 1 is inactive after a master reset operation.

-Output 2: (-OUT 2)—User-designated output that can be made active by setting bit 3 of the Modem Control register. -OUT 2 is inactive after a master reset operation. This signal controls interrupts to the system.

Connector

The following figure shows the pin assignments for the serial port in a communications environment.



Pin	I/O	Signal Name	Pin	I/O	Signal Name
1	N/A	Not Connected	14	N/A	Not Connected
2	O	Transmit Data	15	N/A	Not Connected
3	I	Receive Data	16	N/A	Not Connected
4	O	Request to Send	17	N/A	Not Connected
5	I	Clear to Send	18	N/A	Not Connected
6	I	Data Set Ready	19	N/A	Not Connected
7	N/A	Signal Ground	20	O	Data Terminal Ready
8	I	RLSD	21	N/A	Not Connected
9	N/A	Not Connected	22	I	Ring Indicate
10	N/A	Not Connected	23	N/A	Not Connected
11	N/A	Tied to line 20	24	N/A	Not Connected
12	N/A	Not Connected	25	N/A	Not Connected
13	N/A	Not Connected			

Figure 1-111. Serial Port Connector

The following are the specifications for the serial interface.

Function	Condition
On	Spacing condition (binary 0, positive voltage).
Off	Marking condition (binary 1, negative voltage).

Voltage	Function
Above +15 Vdc	Invalid
+3 to +15 Vdc	On
-3 to +3 Vdc	Invalid
-3 to -15 Vdc	Off
Below -15 Vdc	Invalid

Figure 1-112. Serial Interface Specifications

Parallel Port

The parallel port makes possible the attachment of various devices that accept 8 bits of parallel data at standard TTL levels. The rear of the system unit has a 25-pin, D-shell connector. This port is addressed as parallel port 1.

To allow the parallel port to receive data from external devices, disable the output buffer by writing a 0 to bit 7 of the Planar Control register.

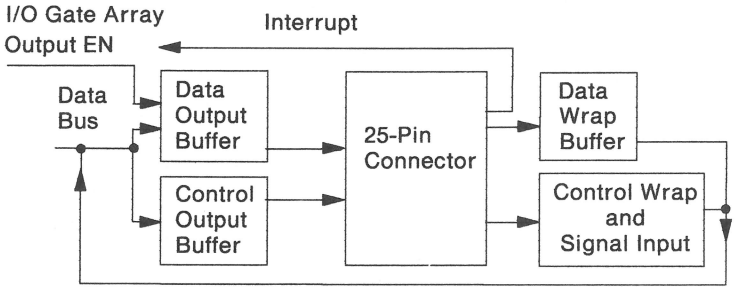


Figure 1-113. Parallel Port Block Diagram

Port Registers

The following describe the registers used for this port in a parallel printer application.

Data Latch, Hex 378: Writing to this address causes data to be stored in the device data buffer. Reading this address returns the contents of the buffer.

The output drivers for this data port will source 2.6 mA at a V_{OH} of 2.4 Vdc and sink 24 mA at a V_{OL} of .5 Vdc. Thirty-nine Ohm resistors are in series with the output drivers.

Printer Controls, Hex 37A: Parallel port control signals are controlled through this address and can be read by the microprocessor. These signals are driven by open collector devices pulled up to +5 Vdc through 4.7K Ohm resistors. The output drivers can sink 16 mA at a V_{OL} of .4 Vdc.

Bit	Function
7	Reserved
6	Reserved
5	Reserved
4	IRQ Enable
3	Select Input
2	-Initialize
1	Auto Feed
0	-Strobe

Figure 1-114. Printer Control Register

The following are bit definitions.

Bit 7-5 Reserved.

Bit 4 IRQ Enable—When set to 1, this bit allows an interrupt to occur when -ACK changes from active to inactive.

Bit 3 Slct In—When set to 1, this bit selects the device.

Bit 2 -Init—When cleared to 0, this bit starts the device (50-microsecond pulse, minimum).

Bit 1 Auto FD XT—When set to 1, this bit causes the device to line feed after a line is printed.

Bit 0 Strobe—An active pulse, minimum of 0.5 μs, clocks data into the device. Valid data must be present for a minimum of 0.5μs before and after the strobe pulse.

Printer Status - Address 379: Parallel port status is stored at this address to be read by the microprocessor. The following are bit definitions for this byte.

Bit	Function
7	-Busy
6	-Acknowledge
5	Page End
4	Selected
3	-Error
2	Reserved
1	Reserved
0	Reserved

Figure 1-115. Printer Status Register

Bit 7 -Busy—This bit indicates the status of the device's 'busy' signal. When the signal is active, this bit is a 0, and the device cannot accept data. It is active during data entry, while the device is offline, or while in an error state.

Bit 6 -ACK—This bit represents the current state of the '-acknowledge' signal. A 0 means the device has received the character and is ready to accept another. Normally, this signal is be active for approximately 5 μs before -BUSY goes active.

Bit 5 PE—When set to 1, this bit indicates a printer has detected the end of the paper.

Bit 4 Slct—When set to 1, this bit indicates the device is selected.

Bit 3 -Error—When cleared to 0, this bit indicates the device has encountered an error condition.

Bits 2-0 Not used.

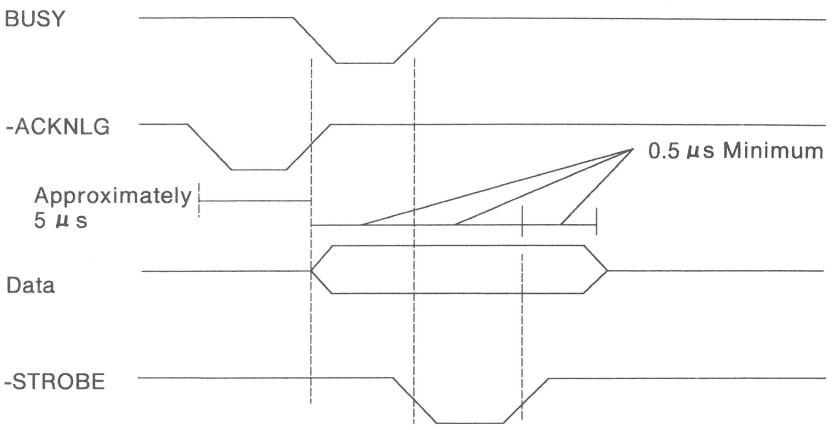
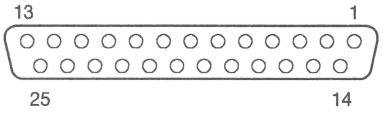


Figure 1-116. Parallel Port Signal Timing

Connector

The port has a 25-pin, D-shell connector at the rear of the system unit. The following figure shows the signals and their pin assignments. Typical printer input signals also are shown.



Pin	I/O	Signal Name	Pin	I/O	Signal Name
1	O	-Strobe	14	O	-Auto Feed XT
2	I/O	D0	15	I	-Error
3	I/O	D1	16	O	-Init
4	I/O	D2	17	O	-Slct In
5	I/O	D3	18	N/A	Ground
6	I/O	D4	19	N/A	Ground
7	I/O	D5	20	N/A	Ground
8	I/O	D6	21	N/A	Ground
9	I/O	D7	22	N/A	Ground
10	I	-ACK	23	N/A	Ground
11	I	Busy	24	N/A	Ground
12	I	PE	25	N/A	Ground
13	I	Slct			

Figure 1-117. Parallel Port Connector

Beeper

The beeper and its control circuits and driver are on the system board. The beeper drive circuit is capable of approximately 1/2 watt of power. The control circuits allow the beeper to be driven three different ways:

1. A direct program control register bit may be toggled to generate a pulse train.
2. The clock input to the timer can be modulated with a program-controlled I/O port bit.
3. The output from channel 2 of the timer may be programmed to generate a waveform to the beeper.

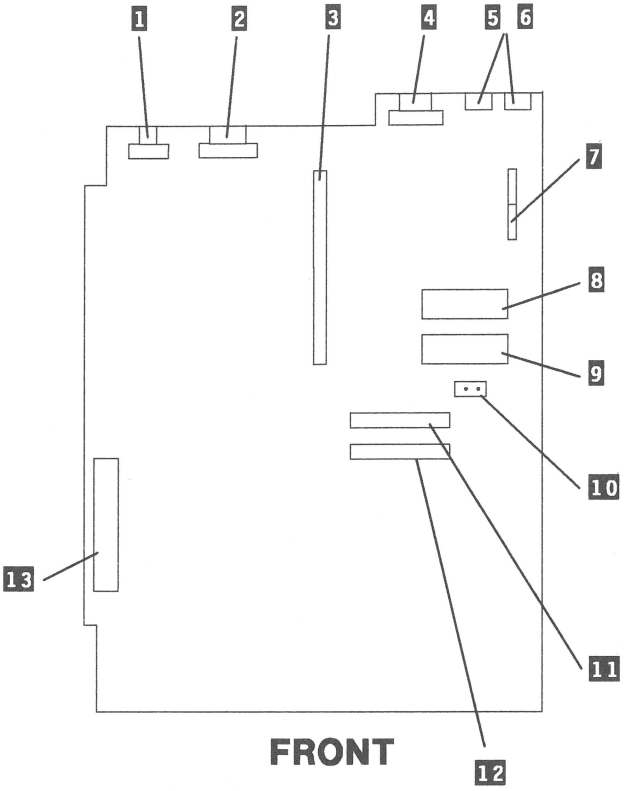
Channel 2	**(Tone generation for beeper)**
Gate 2	-- Controlled by I/O Port Bit 1
Clock In 2	-- 1.9318 MHz OSC
Clock Out 2	-- Used to drive speaker

Figure 1-118. Beeper Tone Generation

All three methods may be performed simultaneously. For more information, see "System Timer" and "I/O Ports" earlier in this section.

Connectors

The following diagram shows the connector locations on the system board.



Ref. #	Description	Ref. #	Description
1	Display Connector	7	Power Supply Connector
2	Serial Connector	8	Microprocessor
3	80-pin I/O Connector	9	Math Coprocessor
4	Parallel Connector	10	Keylock Connector
5 and 6	Keyboard and Pointing Device Connector	11	Fixed Disk Connector
		12	Diskette Drive Connector
		13	Memory SIP's

Figure 1-119. System Board Connector Location

The pin assignments for the power-supply connectors, P3 and P4, are as follows. The pins are numbered 1 through 6 from the rear of the system.

Connector	Pin	Assignments
P3	1	Power Good
	2	Ground
	3	+12 Vdc
	4	-12 Vdc
	5	Ground
	6	Ground
P4	1	Ground
	2	Ground
	3	-5 Vdc
	4	+5 Vdc
	5	+5 Vdc
	6	+5 Vdc

Figure 1-120. Power Supply Connectors

The keyboard and pointing device connectors, J1 and J2, are each six-pin, 90-degree printed circuit board (PCB) mounting, miniature DIN connector. For pin numbering, see the "Keyboard" section. The pin assignments are as follows:

Pin	Assignments
1	Keyboard Data
2	Reserved
3	Ground
4	+5 Vdc
5	Keyboard Clock
6	Reserved

Figure 1-121. Keyboard Connector and Pointing Device

Specifications

Size

- Length: 406 millimeters (16 inches)
- Depth: 397 millimeters (15.6 inches)
- Height: 102 millimeters (4 inches)

Weight

- 7.1 kilograms (15.7 pounds)

Power Cable

- Length: 1.8 meters (6 feet)

Environment

- Air Temperature
  - System On: 15°C to 32°C (60°F to 90°F)
  - System Off: 10°C to 43°C (50°F to 110°F)
- Wet Bulb Temperature
  - System On: 23°C (73°F)
  - System Off: 27°C (80°F)

- Humidity
  - System On: 8% to 80%
  - System Off: 20% to 80%
- Altitude
  - Maximum altitude: 2133.6 meters (7000 feet)

Heat Output

- 341 British Thermal Units per hour

Noise Level

- 38 decibels average-noise rating (without printer)

Electrical

- Power: 240 VA
- Input
  - Nominal: 120 Vac 220 Vac
  - Minimum: 90 Vac 180 Vac
  - Maximum: 137 Vac 265 Vac

Notes:

SECTION 2. Coprocessor

Description 2-3
Programming Considerations 2-3
Hardware Interface 2-4

Notes:

Description

The Math Coprocessor (8087) enables the Model 30 to perform high-speed arithmetic functions, logarithmic functions, and trigonometric operations with extreme accuracy.

The 8087 coprocessor works in parallel with the microprocessor. The parallel operation decreases operating time by allowing the coprocessor to do mathematical calculations while the microprocessor continues to do other functions.

The first five bits of every instruction's operation code for the coprocessor are identical (binary 11011). When the microprocessor and the coprocessor see this operation code, the microprocessor calculates the address of any variables in memory, while the coprocessor checks the instruction. The coprocessor takes the memory address from the microprocessor if necessary. To gain access to locations in memory, the coprocessor takes the local bus from the microprocessor when the microprocessor finishes its current instruction. When the coprocessor is finished with the memory transfer, it returns the local bus to the microprocessor.

The coprocessor works with seven numeric data types divided into the following three classes:

- Binary integers (3 types)
- Decimal integers (1 type)
- Real numbers (3 types).

Programming Considerations

The coprocessor extends the data types, registers, and instructions to the microprocessor.

The coprocessor has eight 80-bit registers, which provide the equivalent capacity of the forty 16-bit registers found in the microprocessor. This register space allows constants and temporary results to be held in registers during calculations, thus reducing memory access and improving speed as well as bus availability. The register space can be used as a stack or as a fixed register set.

When used as a stack, only the top two stack elements are operated on. The figure below shows representations of large and small numbers in each data type.

Data Type	Bits	Significant Digits (Decimal)	Approximate Range (Decimal)
Word Integer	16	4	-32,768 through +32,767
Short Integer	32	9	-2×10^{9} through $+2 \times 10^{9}$
Long Integer	64	18	-9×10^{18} through $+9 \times 10^{18}$
Packed Decimal	80	18	-9..99 through +9..99 (18 digits)
Short Real *	32	6-7	8.43×10^{-37} through 3.37×10^{38}
Long Real *	64	15-16	4.19×10^{-307} through 1.67×10^{308}
Temporary Real	80	19	3.4×10^{-4932} through 1.2×10^{4932}

* The Short Real and Long Real data types correspond to the single and double-precision data types.

Figure 2-1. Coprocessor Data Types

Hardware Interface

The coprocessor uses the same clock generator and system bus interface components as the microprocessor. The microprocessor's queue status lines (QS0 and QS1) enable the coprocessor to obtain and decode instructions simultaneously with the microprocessor. The coprocessor's 'busy' signal (BUSY) informs the microprocessor that it is executing and the coprocessor's Wait instruction forces the microprocessor to wait until the coprocessor is finished executing (Wait for not busy).

When an incorrect instruction is sent to the coprocessor (for example, divide by 0 or load a full register), the coprocessor can signal the microprocessor with an interrupt on the NMI. There are two

conditions that will disable the coprocessor interrupt to the microprocessor:

1. Exception and interrupt-enable bits of the control word are set to 1's.
2. NMI is masked off.

Any program using the coprocessor's interrupt capability must ensure that the second condition is never met during the operation of the software or an "endless wait" will occur. An "endless wait" has the microprocessor waiting for BUSY to go inactive from the coprocessor while the coprocessor is waiting for the microprocessor to interrupt.

Because a memory parity error may also cause an NMI, the program should check the coprocessor status for an exception condition. If a coprocessor exception condition is not found, control is passed to the normal NMI handler. If an 8087 exception condition is found, the program clears the exception by executing the FNSAVE or the FNCLEX instruction, and the exception can be identified and acted upon.

The NMI and the coprocessor's interrupt are tied to the NMI line through the NMI interrupt logic.

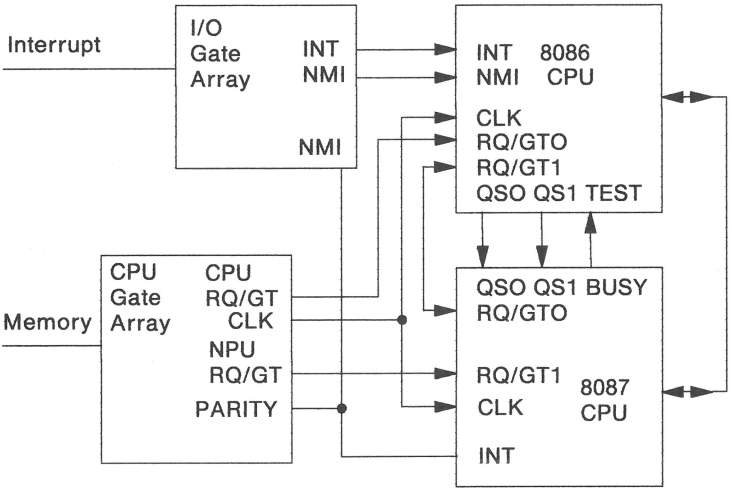


Figure 2-2. Coprocessor Interconnection

Notes:

SECTION 3. Power Supply

Description 3-3
Input and Output Power 3-4
Overvoltage/Overcurrent Protection 3-5
Power-Good Signal 3-5
Connectors 3-6

Notes:

Description

The system dc power supply is a single-phase, 70-watt, four voltage level supply. It is internal to the system unit and supplies power for the system unit, its options, and the keyboard. The supply provides 9.0 A of +5 Vdc, 1.8 A of +12 Vdc, 110 mA of -5 Vdc, and 300 mA of -12 Vdc. All power levels are monitored with undervoltage and overcurrent protection.

The power supply handles either 120 Vac or 220/240 Vac inputs. The input is protected by an internal fuse. If dc overcurrent or undervoltage conditions exist, the supply automatically shuts down until the condition is corrected.

The system board and storage devices take approximately 5 A of +5 Vdc, thus allowing approximately 4 A of +5 Vdc for the adapters in the system expansion slots. The -5 Vdc level is used for adapters. The +12 Vdc power level is designed to power the internal diskette drives and the fixed disk drive. The +12 Vdc and -12 Vdc are used for powering the Electronic Industries Association (EIA) drivers for the serial port. All four power levels are bussed across the three system expansion slots.

Input and Output Power

The nominal power requirements and voltages are listed in the following tables.

Nominal (Vac)	Minimum (Vac)	Maximum (Vac)	Maximum Current
110	90	137	2.0 A
220	180	265	1.0 A

Figure 3-1. Vac Input Requirements

Nominal (Vdc)	Load Current (A) Min	Load Current (A) Max	Regulation Tolerance	Ripple (mV p-p)
+5 Vdc	1.5	9.0	+5% to -4%	50
-5 Vdc	0.0	0.11	+10% to -8%	100
+12 Vdc	0.0	1.8	+5% to -4%	80
-12 Vdc	0.0	0.30	+10% to -9%	250

Figure 3-2. Vdc Output

Overvoltage/Overcurrent Protection

The sense levels of the dc outputs are:

Output (Vdc)	Minimum (Vdc)
+5 Vdc	+4.5
-5 Vdc	-4.3
+12 Vdc	+10.8
-12 Vdc	-10.2

Figure 3-3. Output Protection

Power-Good Signal

The power supply provides a 'power good' signal to reset the system logic, to indicate proper operation of the power supply, and to give advance warning when the power is turned off.

The signal has a TTL-compatible active level of 2.4 to 5.25 Vdc during normal operation, or an inactive level of 0.0 to 0.4 Vdc. The signal is inactive if an undervoltage condition occurs, or during the power-on and power-off sequence. The 'power good' signal has a turn-on delay that is at least 100 ms but no greater than 500 ms. This line can sink 2 mA or source 100 μA.

Connectors

The power supply attaches to the system board through two 6-pin connectors. Connector P3 is the one closest to the rear of the system. The pin numbering and signal assignments are shown below.

Connector	Pin	Assignments
P3	1	Power Good
	2	Ground
	3	+12 Vdc
	4	-12 Vdc
	5	Ground
	6	Ground
P4	1	Ground
	2	Ground
	3	-5 Vdc
	4	+5 Vdc
	5	+5 Vdc
	6	+5 Vdc

Figure 3-4. Power Supply Connectors

SECTION 4. Keyboard

Description 4-3
Sequence Key-Code Scanning 4-3
Keyboard Buffer 4-3
Keys 4-3
Power-On Routine 4-4
Power-On Reset 4-4
Basic Assurance Test 4-4
Clock and Data Signals 4-5
Data Stream 4-6
Data Output 4-6
Commands 4-7
Scan Codes 4-8
Encoding 4-11
Character Codes 4-11
Extended Functions 4-17
Shift States 4-19
Special Handling 4-21
Other Characteristics 4-22
Layouts 4-23
Arabic 4-24
Belgian 4-25
Canadian 4-26
Danish 4-27
Dutch 4-28
French 4-29
German 4-30
Israeli 4-31
Italian 4-32
Latin American 4-33
Norwegian 4-34
Portuguese 4-35
Spanish 4-36
Swedish 4-37
Swiss 4-38
U.K. English 4-39
U.S. English 4-40
Cables and Connectors 4-41
Specifications 4-42

Power Requirements 4-42
Size 4-42
Weight 4-42

Description

The keyboard has 101 keys (102 in countries outside of the U. S.). At system power-on, the keyboard monitors the signals on the 'clock' and 'data' lines and establishes its line protocol.

Sequence Key-Code Scanning

The keyboard detects each key pressed, and sends each scan code in the sequence pressed. When not serviced by the system, the keyboard stores the scan codes in its buffer.

Keyboard Buffer

A 16-byte first-in-first-out (FIFO) buffer in the keyboard stores the scan codes until the system is ready to receive them.

A buffer-overrun condition occurs when more than 16 bytes are placed in the keyboard buffer. An overrun code replaces the 17th byte. If more keys are pressed while the buffer is full, the additional keystrokes are lost.

When the keyboard is allowed to send data, the bytes in the buffer are sent as in normal operation, and any additional keystrokes are sent. Response codes do not occupy a buffer position.

When keystrokes generate a multiple-byte sequence, the entire sequence fits into the available buffer space; otherwise a buffer-overrun condition occurs.

Keys

With the exception of the Pause key, all keys are *make/break*. The break code is the make code prefixed by hex F0 (in the case of a multiple byte make code, only the last byte is prefixed). The make scan code of a key is sent to the keyboard controller when the key is pressed. When the key is released, its break scan code is sent.

Additionally, except for the Pause key, all keys are *typematic*. When a key is pressed and held down, the keyboard sends the make code for that key, delays 500 ms $\pm$20%, and begins sending a make code for that key at a rate of 10.9 characters per second $\pm$20%.

If two or more keys are held down, only the last key pressed repeats at the typematic rate. Typematic operation stops when the last key pressed is released, even if other keys are still held down. If a key is pressed and held down while keyboard transmission is inhibited, only the first make code is stored in the buffer. This prevents buffer overflow as a result of typematic action.

Power-On Routine

The following activities take place when power is first applied to the keyboard.

Power-On Reset

The keyboard logic generates a 'power-on reset' signal (POR) when power is first applied to the keyboard. POR occurs during 150 ms to 2.0 seconds from the time power is first applied to the keyboard.

Basic Assurance Test

The basic assurance test (BAT) consists of a keyboard processor test, a checksum of the read-only memory (ROM), and a random-access memory (RAM) test. During the BAT, activity on the 'clock' and 'data' lines is ignored. The BAT takes from 300 to 500 ms. This is in addition to the time required by the POR.

Upon satisfactory completion of the BAT, a completion code (hex AA) is sent to the system, and keyboard scanning begins. If a BAT failure occurs, the keyboard sends an error code to the system. The keyboard is then disabled pending command input. Completion codes are sent 600 ms to 2.5 seconds after POR, and between 300 and 500 ms after a Reset command is acknowledged.

Following a successful POR, the keyboard sets the line protocol to Scan Set 1.

Clock and Data Signals

The keyboard and system communicate over the 'clock' and 'data' lines. The source of each of these lines is an open-collector device on the keyboard that allows either the keyboard or the system to force a signal inactive. When no communication is occurring, the 'clock' line is active. The state of the 'data' line is held inactive by the keyboard.

An inactive signal is between 0.0 and +0.7 volts. A signal at the inactive level is a logical 0. An active signal is between +2.4 and +5.5 volts. A signal at the active level is a logical 1. Voltages are measured between a signal source and the dc ground.

The keyboard 'clock' line provides the clocking signals used to clock serial data from the keyboard. If the system forces the 'clock' line inactive, keyboard transmission is inhibited.

When the keyboard sends data to the system, it generates the 'clock' signal to time the data. The system can prevent the keyboard from sending data by forcing the 'clock' line inactive, or by holding the 'data' line inactive.

During the BAT, the keyboard allows the 'clock' and 'data' lines to go active.

Data Stream

Data transmissions from the keyboard consist of a 11-bit data stream sent serially over the 'data' line. The following table shows the data stream.

Bit	Function
1	Start Bit (Always 0)
2	Data Bit 0 (Least-Significant)
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 (Most-Significant)
10	Parity Bit (Odd Parity)
11	Stop Bit (Always 1)

Figure 4-1. Keyboard Data Stream

Data Output

When the keyboard is ready to send data, it first checks the status of the 'clock' and 'data' lines. When the 'clock' line is inactive (keyboard inhibit) the keyboard stores the data in its buffer. When the 'data' line is inactive and the 'clock' line is active (system request to send), the keyboard stores the data in its buffer and accepts the system input.

If both lines are active, the keyboard sends the data stream. During transmission, the keyboard monitors the 'clock' line. If the line goes inactive before the parity bit is sent, the keyboard stores the data in its buffer and returns both the 'clock' and 'data' line to active and waits on the system. If the parity bit has been sent, the keyboard completes the transmission.

Commands

Reset (Hex FF): The system issues a Reset command to initiate a keyboard reset and internal self-test. The keyboard acknowledges (ACK) receiving the command, but before executing the reset, the keyboard waits for the system to accept the ACK. If the system accepts the ACK, it pulses the clock and data lines with a 500-ms active pulse. The keyboard remains in a reset mode until the clock and data lines are pulsed or until another command is sent.

The following describes the commands that the keyboard sends to the system, and shows their hexadecimal values.

Command	Hex Value
BAT Completion Code	AA
BAT Failure Code	FC
Key Detection Error/Overrun	FF

Figure 4-2. Commands from the Keyboard

BAT Completion Code (Hex AA): Following satisfactory completion of the BAT, the keyboard sends hex AA. Any other code indicates a failure of the keyboard.

BAT Failure Code (Hex FC): If a BAT failure occurs, the keyboard sends this code, discontinues scanning, and waits for a system response or reset.

Key Detection Error (Hex FF): The keyboard sends a key detection error character (hex FF) if conditions in the keyboard make it impossible to identify a switch closure.

Overrun (Hex FF): An overrun character (hex FF) is placed in the keyboard buffer and replaces the last code when the buffer capacity has been exceeded. The code is sent to the system when it reaches the top of the buffer queue.

Scan Codes

Each key is assigned a make and break scan code and, in some cases, an extra set of codes to generate artificial shift states in the system. The typematic scan codes are identical to the make scan code for each key.

The following keys send the codes shown, regardless of the shifted states of the keyboard. Refer to the keyboard layout to determine the character associated with each key.

Key No.	Make	Break	Key No.	Make	Break
1	29	A9	47	2D	AD
2	02	82	48	2E	AE
3	03	83	49	2F	AF
4	04	84	50	30	B0
5	05	85	51	31	B1
6	06	86	52	32	B2
7	07	87	53	33	B3
8	08	88	54	34	B4
9	09	89	55	35	B5
10	0A	8A	57	36	B6
11	0B	8B	58	1D	9D
12	0C	8C	60	38	B8
13	0D	8D	61	39	B9
15	0E	8E	62	E0 38	E0 B8
16	0F	8F	64	E0 1D	E0 9D
17	10	90	90	45	C5
18	11	91	91	47	C7
19	12	92	92	4B	CB
20	13	93	93	4F	CF
21	14	94	96	48	C8
22	15	95	97	4C	CC
23	16	96	98	50	D0
24	17	97	99	52	D2
25	18	98	100	37	B7
26	19	99	101	49	C9
27	1A	9A	102	4D	CD
28	1B	9B	103	51	D1
29 +	2B	AB	104	53	D3
30	3A	BA	105	4A	CA
31	1E	9E	106	4E	CE
32	1F	9F	108	E0 1C	E0 9C
33	20	A0	110	01	81
34	21	A1	112	3B	BB
35	22	A2	113	3C	BC
36	23	A3	114	3D	BD
37	24	A4	115	3E	BE
38	25	A5	116	3F	BF
39	26	A6	117	40	C0
40	27	A7	118	41	C1
41	28	A8	119	42	C2
42 +	2B	AB	120	43	C3
43	1C	9C	121	44	C4
44	2A	AA	122	D9	D7
45 +	D5	D6	123	DA	D8
46	2C	AC	125	46	C6

+ Key 29 on U.S keyboard only, keys 42 and 45 on all but U.S. keyboard.

Figure 4-3. Scan Codes (Part 1 of 4)

The remaining keys send a series of codes depending on the state of various shift keys (Ctrl, Alt, and Shift), and the state of Num Lock (On or Off). Because the base scan code is identical to that of another key, an extra code (hex E0) is added to the base code to make it unique.

The following show the make/break code using the left shift key. If the right shift key is used, substitute its make/break for that of the left shift key.

Key	Base Case, or Shift + Num Lock	Shift Case	Num Lock
75	E0 52/E0 D2	E0 AA E0 52/E0 D2 E0 2A	E0 2A E0 52/E0 D2 E0 AA
76	E0 53/E0 D3	E0 AA E0 53/E0 D3 E0 2A	E0 2A E0 53/E0 D3 E0 AA
79	E0 4B/E0 CB	E0 AA E0 4B/E0 CB E0 2A	E0 2A E0 4B/E0 CB E0 AA
80	E0 47/E0 C7	E0 AA E0 47/E0 C7 E0 2A	E0 2A E0 47/E0 C7 E0 AA
81	E0 4F/E0 CF	E0 AA E0 4F/E0 CF E0 2A	E0 2A E0 4F/E0 CF E0 AA
83	E0 48/E0 C8	E0 AA E0 48/E0 C8 E0 2A	E0 2A E0 48/E0 C8 E0 AA
84	E0 50/E0 D0	E0 AA E0 50/E0 D0 E0 2A	E0 2A E0 50/E0 D0 E0 AA
85	E0 49/E0 C9	E0 AA E0 49/E0 C9 E0 2A	E0 2A E0 49/E0 C9 E0 AA
86	E0 51/E0 D1	E0 AA E0 51/E0 D1 E0 2A	E0 2A E0 51/E0 D1 E0 AA
89	E0 4D/E0 CD	E0 AA E0 4D/E0 CD E0 2A	E0 2A E0 4D/E0 CD E0 AA

Figure 4-4. Scan Codes (Part 2 of 4)

Key No.	Base Case	Shift Case
95	E0 35/E0 B5	AA E0 35/E0 B5 2A

Figure 4-5. Scan Codes (Part 3 of 4)

Key No.	Base Case	Shift or Ctrl Case	Alt Case
124	E0 2A E0 37/E0 B7 E0 AA	E0 37/E0 B7	54/D4
126 *	E1 1D 45 E1 9D C5	E0 46 E0 C6	
	* All associated scan codes are sent on the make of the key.		

Figure 4-6. Scan Codes (Part 4 of 4)

Encoding

The keyboard routine, provided in ROM BIOS, converts the keyboard scan codes into *Extended ASCII*. The extended ASCII codes returned by the BIOS routine are mapped to the U.S. English keyboard layout. Operating systems can make provisions for alternate keyboard layouts by providing another keyboard routine. This section discusses only the ROM routine.

Extended ASCII encompasses 1-byte character codes, with possible values of 0 to 255, an extended code for certain extended keyboard functions, and functions handled within the keyboard routine or through interrupts.

Character Codes

The character codes are passed through the BIOS keyboard routine to the system or application program. A "-1" in the following figures indicates the combination is suppressed in the keyboard routine. The codes are returned in the AL register. See "Characters and Keystrokes" later in this manual for the exact codes.

Figure 4-7. 101-Key Keyboard Layout

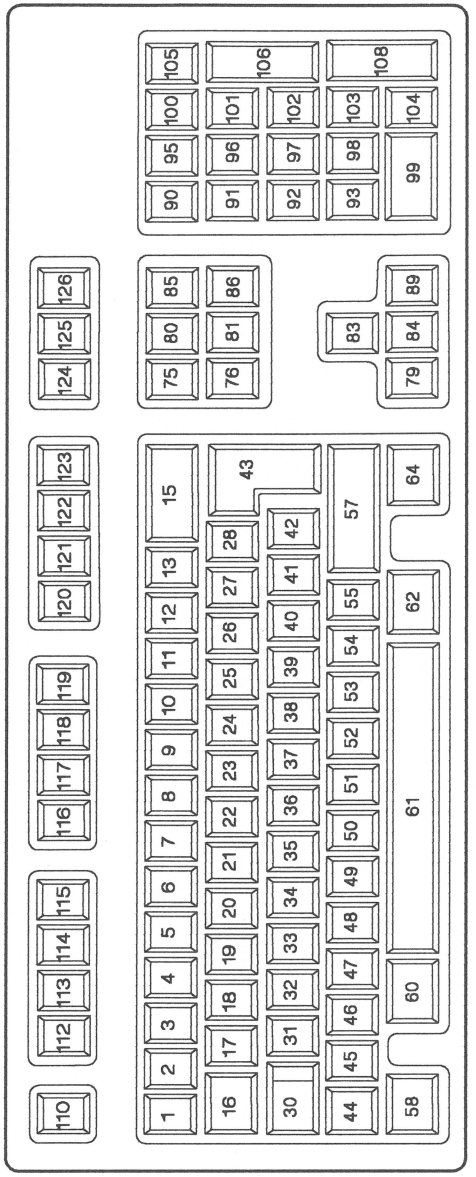


Figure 4-8. 102-Key Keyboard Layout

Key	Base Case	Uppercase	Ctrl.	Alt	
1	'	~	-1	(*)	
2	1	!	-1	(*)	
3	2	@	Nul(000) (*)	(*)	
4	3	#	-1	(*)	
5	4	$	-1	(*)	
6	5	%	-1	(*)	
7	6	^	RS(030)	(*)	
8	7	&	-1	(*)	
9	8	*	-1	(*)	
10	9	(	-1	(*)	
11	0	)	-1	(*)	
12	-	_	US(031)	(*)	
13	=	+	-1	(*)	
15	Backspace (008)	Backspace (008)	Del(127)	(*)	
16	→| (009)	|← (*)	(*)	(*)	
17	q	Q	DC1(017)	(*)	
18	w	W	ETB(023)	(*)	
19	e	E	ENQ(005)	(*)	
20	r	R	DC2(018)	(*)	
21	t	T	DC4(020)	(*)	
22	y	Y	EM(025)	(*)	
23	u	U	NAK(021)	(*)	
24	i	I	HT(009)	(*)	
25	o	O	SI(015)	(*)	
26	p	P	DLE(016)	(*)	
27	{	{	Esc(027)	(*)	
28	}	}	GS(029)	(*)	
29	\	¦	FS(028)	(*)	
30	Caps Lock		-1	-1	-1 -1
31	a	A	SOH(001)	(*)	
32	s	S	DC3(019)	(*)	
33	d	D	EOT(004)	(*)	
34	f	F	ACK(006)	(*)	
35	g	G	BEL(007)	(*)	
36	h	H	BS(008)	(*)	
37	j	J	LF(010)	(*)	
38	k	K	VT(011)	(*)	
39	l	L	FF(012)	(*)	
40	;	:	-1	(*)	
41	'	"	-1	(*)	
43	CR	CR	LF(010)	(*)	

Note:
(*) Refer to "Extended Functions" in this section.

Figure 4-9 (Part 1 of 3). Character Codes

Key	Base Case	Uppercase	Ctrl.	Alt
44 Shift (Left)	-1	-1	-1	-1
46	z	Z	SUB(026)	(*)
47	x	X	CAN(024)	(*)
48	c	C	ETX(003)	(*)
49	v	V	SYN(022)	(*)
50	b	B	STX(002)	(*)
51	n	N	SO(014)	(*)
52	m	M	CR(013)	(*)
53	,	<	-1	(*)
54	.	>	-1	(*)
55	/	?	-1	(*)
57 Shift (Right)	-1	-1	-1	-1
58 Ctrl (Left)	-1	-1	-1	-1
60 Alt (Left)	-1	-1	-1	-1
61	Space	Space	Space	Space
62 Alt (Right)	-1	-1	-1	-1
64 Ctrl (Right)	-1	-1	-1	-1
90 Num Lock	-1	-1	-1	-1
95	/	/	(*)	(*)
100	*	*	(*)	(*)
105	-	-	(*)	(*)
106	+	+	(*)	(*)
108	Enter	Enter	LF(010)	(*)

Notes:
(*) Refer to "Extended Functions" in this section.

Figure 4-9 (Part 2 of 3). Character Codes

Key	Base Case	Uppercase	Ctrl.	Alt
110	Esc	Esc	Esc	(*)
112	Null (*)	Null (*)	Null (*)	Null(*)
113	Null (*)	Null (*)	Null (*)	Null(*)
114	Null (*)	Null (*)	Null (*)	Null(*)
115	Null (*)	Null (*)	Null (*)	Null(*)
116	Null (*)	Null (*)	Null (*)	Null(*)
117	Null (*)	Null (*)	Null (*)	Null(*)
118	Null (*)	Null (*)	Null (*)	Null(*)
119	Null (*)	Null (*)	Null (*)	Null(*)
120	Null (*)	Null (*)	Null (*)	Null(*)
121	Null (*)	Null (*)	Null (*)	Null(*)
122	Null (*)	Null (*)	Null (*)	Null(*)
123	Null (*)	Null (*)	Null (*)	Null(*)
125 Scroll Lock		-1 Num Lock	84/85-key	-1 -1
126	Pause(**)	Pause(**)	Break(**)	Pause(**)

Notes:
(*) Refer to "Extended Functions" in this section.
(**) Refer to "Special Handling" in this section.

Figure 4-9 (Part 3 of 3). Character Codes

The following figure lists keys that have meaning only in Num Lock, Shift, or Ctrl states. The Shift key temporarily reverses the current Num Lock state.

Key	Num Lock	Base Case	Alt.	Ctrl
91	7	Home (*)	-1	Clear Screen
92	4	←(*)	-1	Reverse Word(*)
93	1	End (*)	-1	Erase to EOL(*)
96	8	↑(*)	-1	(*)
97	5	(*)	-1	(*)
98	2	↓(*)	-1	(*)
99	0	Ins	-1	(*)
101	9	Page Up (*)	-1	Top of Text and Home
102	6	→(*)	-1	Advance Word (*)
103	3	Page Down (*)	-1 (*)	Erase to EOS
104	.	Delete (*,**)	(**)	(**)
105	-	Sys Request	-1	-1
106	+	+ (*)	-1	-1

Notes:
(*) Refer to "Extended Functions" in this section.
(**) Refer to "Special Handling" in this section.

Figure 4-10. Special Character Codes

Extended Functions

For certain functions that cannot be represented by a standard ASCII code, an extended code is used. A character code of 00 (null) is returned in AL. This indicates that the system or application program should examine a second code, which will indicate the actual function. Usually, but not always, this second code is the scan code of the primary key that was pressed. This code is returned in AH.

The following figure is a list of the extended codes and their functions.

Second Code	Function
1	Alt Esc
3	Nul Character
14	Alt Backspace
15	⇤ (Back-tab)
16-25	Alt Q, W, E, R, T, Y, U, I, O, P
26-28	Alt [ ] ↵
30-38	Alt A, S, D, F, G, H, J, K, L
39-41	Alt ; ' `
43	Alt \
44-50	Alt Z, X, C, V, B, N, M
51-53	Alt , . /
55	Alt Keypad *
59-68	F1 to F10 Function Keys (Base Case)
71	Home
72	↑ (Cursor Up)
73	Page Up
74	Alt Keypad -
75	← (Cursor Left)
76	Center Cursor
77	→ (Cursor Right)
78	Alt Keypad +
79	End
80	↓ (Cursor Down)
81	Page Down
82	Ins (Insert)
83	Del (Delete)
84-93	Shift F1 to F10
94-103	Ctrl F1 to F10
104-113	Alt F1 to F10
114	Ctrl PrtSc (Start/Stop Echo to Printer)
115	Ctrl ← (Reverse Word)
116	Ctrl → (Advance Word)
117	Ctrl End (Erase to End of Line-EOL)
118	Ctrl PgDn (Erase to End of Screen-EOS)
119	Ctrl Home (Clear Screen and Home)
120-131	Alt 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, = keys 2-13
132	Ctrl PgUp (Top 25 Lines of Text and Cursor Home)
133-134	F11, F12
135-136	Shift F11, F12
137-138	Ctrl F11, F12
139-140	Alt F11, F12
141	Ctrl Up/8
142	Ctrl Keypad -
143	Ctrl Keypad 5
144	Ctrl Keypad
145	Ctrl Down/2
146	Ctrl Ins/0
147	Ctrl Del/.
148	Ctrl Tab

Figure 4-11 (Part 1 of 2). Keyboard Extended Functions

Second Code	Function
149	Ctrl Keypad /
150	Ctrl Keypad *
151	Alt Home
152	Alt Up
153	Alt Page Up
155	Alt Left
157	Alt Right
159	Alt End
160	Alt Down
161	Alt Page Down
162	Alt Insert
163	Alt Delete
164	Alt Keypad /
165	Alt Tab
166	Alt Enter

Figure 4-11 (Part 2 of 2). Keyboard Extended Functions

Shift States

Most shift states are handled within the keyboard routine, and are not apparent to the system or application program. In any case, the current status of active shift states is available by calling an entry point in the BIOS keyboard routine. The following keys result in altered shift states:

Shift: This key temporarily shifts keys 1 through 13, 16 through 29, 31 through 41, and 46 through 55, to uppercase (base case if in Caps Lock state). Also, the Shift key temporarily reverses the Num Lock or non-Num Lock state of keys 91 through 93, 96, 98, 99, and 101 through 104.

Ctrl: This key temporarily shifts keys 3, 7, 12, 15 through 29, 31 through 39, 43, 46 through 52, 75 through 89, 91 through 93, 95 through 108, 112 through 124 and 126 to the Ctrl state. The Ctrl key is also used with the Alt and Del keys to cause the system-reset function, with the Scroll Lock key to cause the break function, and with the Num Lock key to cause the pause function. The system-reset, break, and pause functions are described under "Special Handling" later in this section.

Alt: This key temporarily shifts keys 1 through 29, 31 through 43, 46 through 55, 75 through 89, 95, 100, and 105 through 124 to the Alt state. The Alt key is also used with the Ctrl and Del keys to cause a system reset.

The Alt key also allows the user to enter any character code from 0 to 255. The user holds down the Alt key and types the decimal value of the characters desired on the numeric keypad (keys 91 through 93, 96 through 99, and 101 through 103). The Alt key is then released. If the number is greater than 255, a modulo-256 value is used. This value is interpreted as a character code and is sent through the keyboard routine to the system or application program. Alt is handled in the keyboard routine.

Caps Lock: This key shifts keys 17 through 26, 31 through 39, and 46 through 52 to uppercase. When Caps Lock is pressed again, it reverses the action. Caps Lock is handled in the keyboard routine.

Scroll Lock: When interpreted by appropriate application programs, this key indicates that the cursor-control keys cause windowing over the text rather than moving the cursor. When the Scroll Lock key is pressed again, it reverses the action. The keyboard routine simply records the current shift state of the Scroll Lock key. It is the responsibility of the application program to perform the function.

Num Lock: This key shifts keys 91 through 93, 96 through 99, and 101 through 104 to uppercase. When Num Lock is pressed again, it reverses the action. Num Lock is handled in the keyboard routine.

Shift Key Priorities and Combinations: If combinations of the Alt, Ctrl, and Shift keys are pressed and only one is valid, the priority is as follows: Alt key first, Ctrl key second, and Shift key third. The only valid combination is Alt and Ctrl, which is used in the system-reset function.

Special Handling

System Reset: The combination of Alt, Ctrl, and Del keys results in the keyboard routine that starts a system reset or restart. System reset is handled by BIOS.

Break: The combination of the Ctrl and Pause/Break keys results in the keyboard routine signaling interrupt hex 1B. The extended characters (AL) = hex 00, and (AH) = hex 00 are also returned.

Pause: The Pause key causes the keyboard interrupt routine to loop, waiting for any character or function key to be pressed. This provides a method of temporarily suspending an operation, such as listing or printing, and then resuming the operation. The method is not apparent to either the system or the application program. The keystroke used to resume operation is discarded. Pause is handled in the keyboard routine.

Print Screen: The Print Screen key results in an interrupt invoking the print-screen routine. This routine works in the alphanumeric or graphics mode, with unrecognizable characters printing as blanks.

System Request: When the System Request (Alt and Print Screen) key is pressed, a hex 8500 is placed in AX, and an interrupt hex 15 is executed. When the Sys Req key is released, a hex 8501 is placed in AX, and another interrupt hex 15 is executed. If an application is to use System Request, the following rules must be observed:

Save the previous address.

Overlay interrupt vector hex 15.

Check AH for a value of hex 85:

If yes, process may begin.
If no, go to previous address.

The application program preserves the value in all registers, except AX, upon return. System request is handled in the keyboard routine.

Other Characteristics

The keyboard routine does its own buffering (16 bytes). If a key is pressed when the buffer is full, that key is ignored and the beeper sounds.

The keyboard routine also suppresses the typematic action of the following keys: Ctrl, Shift, Alt, Num Lock, Scroll Lock, Caps Lock, and Ins.

During each interrupt hex 09 from the keyboard, an interrupt hex 15, function (AH) = hex 4F is generated by the BIOS after the scan code is read from the keyboard adapter. The scan code is passed in the AL register with the carry flag set. This allows an operating system to intercept each scan code before it is handled by the interrupt hex 09 routine, and to change or act on the scan code. If the carry flag is changed to 0 on return from interrupt hex 15, the scan code is ignored by interrupt hex 09.

Layouts

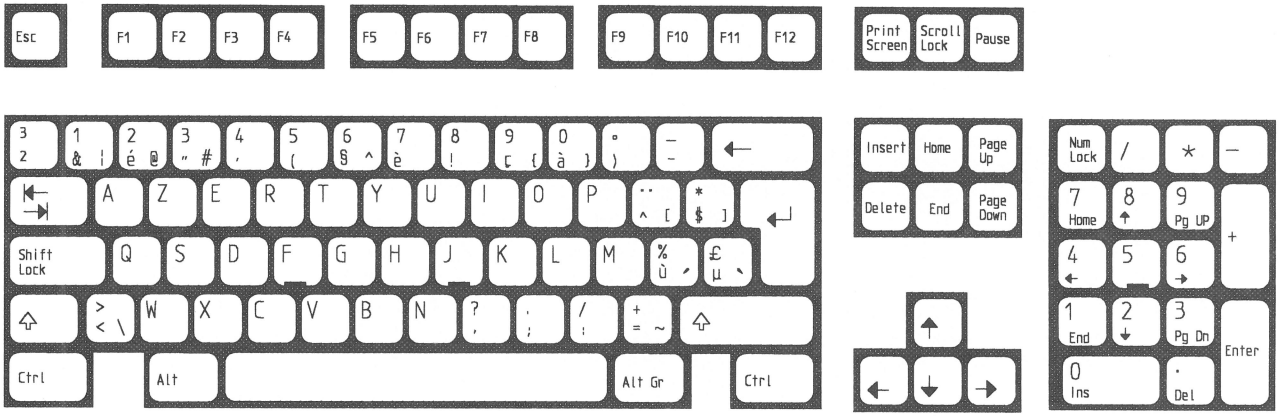
The keyboard is available in the 17 layouts shown below:

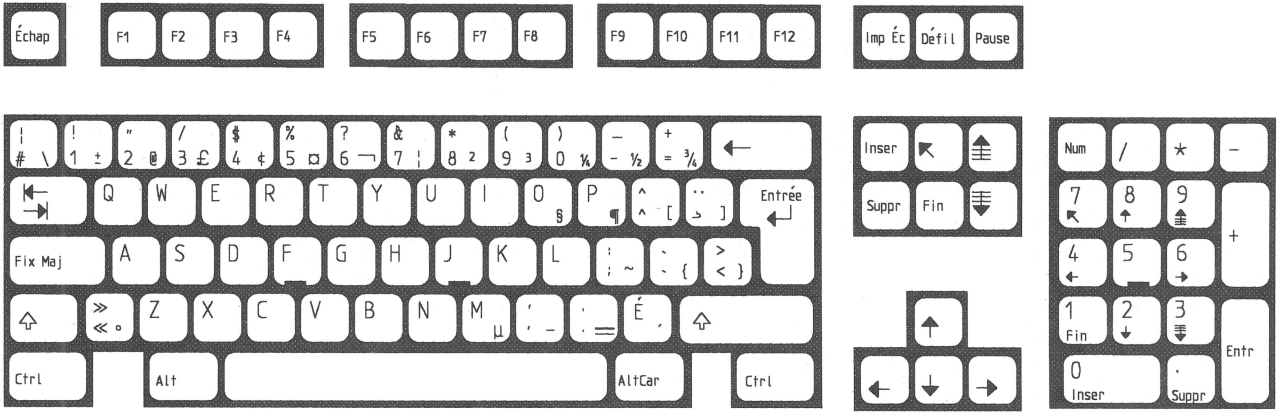
- Arabic
- Belgian
- Canadian French
- Danish
- Dutch
- French
- German
- Israeli
- Italian
- Latin American
- Norwegian
- Portuguese
- Spanish
- Swedish
- Swiss
- U.K. English
- U.S. English

The layouts are shown in alphabetic order on the following pages. The characters normally found on the front face of the keybuttons are shown on the lower right corner of the keys in the layouts.

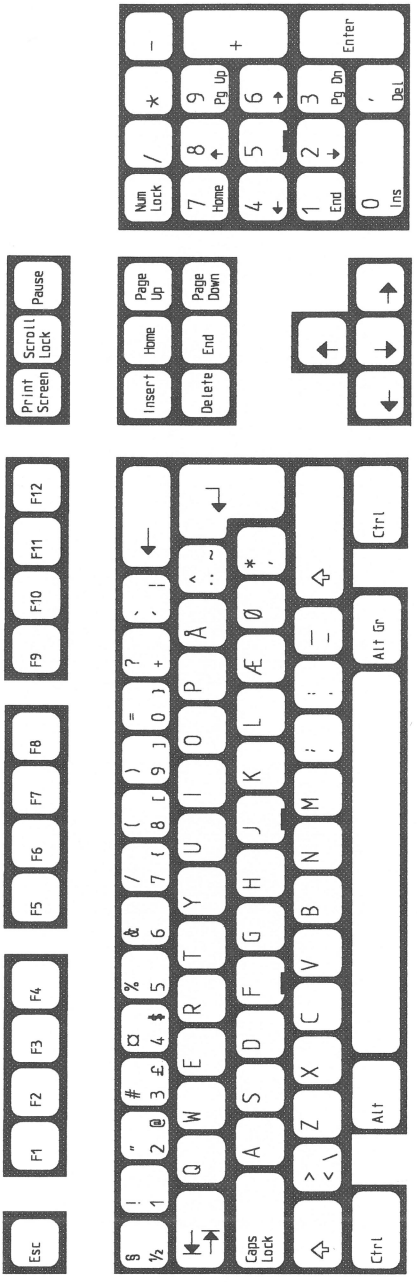
Arabic

Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Scroll Lock
Pause
Backspace
Tab
Caps Lock
New Line
Ctrl
Alt
Ctrl
Home
End
Num Lock
Home
Pg Up
End
Pg Dn
Insert
Del
Enter

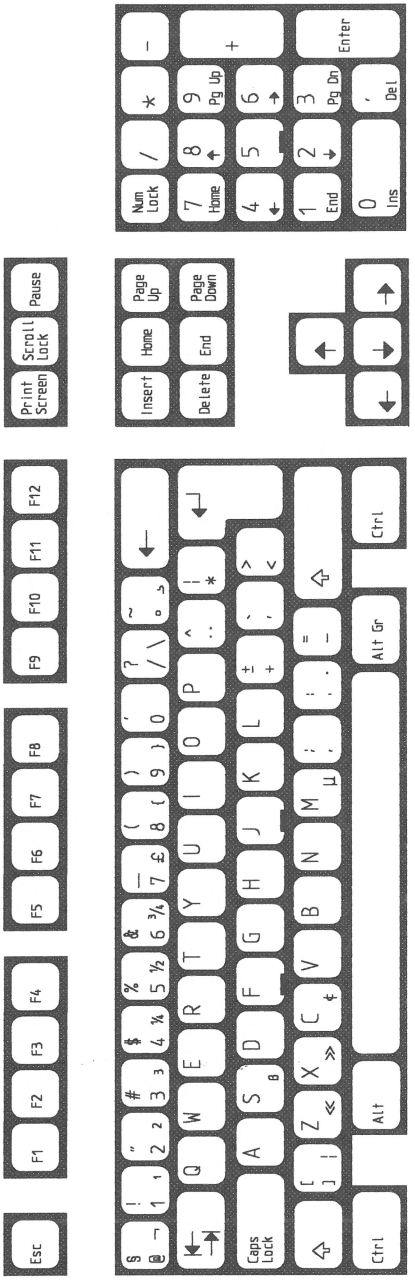

Esc
F1 F2 F3 F4
F5 F6 F7 F8
F9 F10 F11 F12
Print Screen
Scroll Lock
Pause
Insert Home Page Up
Delete End Page Down
Num Lock
Shift Lock
Ctrl
Alt
Alt Gr
Ctrl
Enter


Échap
F1 F2 F3 F4
F5 F6 F7 F8
F9 F10 F11 F12
Imp Éc Défil Pause
Entrée
Fix Maj
Ctrl
Alt
AltCar
Ctrl
Inser
Suppr Fin
Num
Entr
Inser
Suppr

Danish

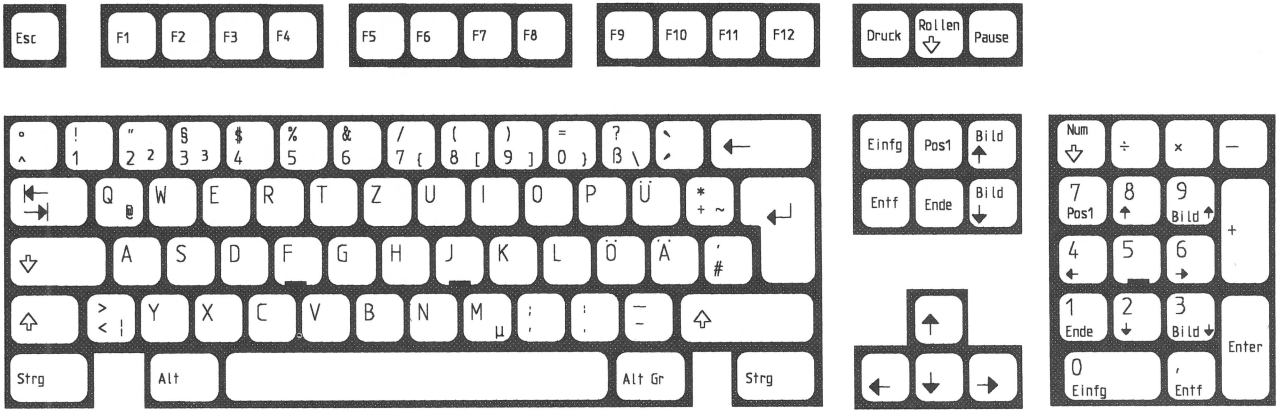

Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Print Screen
Scroll Lock
Pause
Caps Lock
Ctrl
Alt
Alt Gr
Ctrl
Insert
Home
Page Up
Delete
End
Page Down
Num Lock
Home
Pg Up
End
Pg Dn
Ins
Del
Enter

Dutch


Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Print Screen
Scroll Lock
Pause
Caps Lock
Ctrl
Alt
Alt Gr
Ctrl
Insert
Home
Page Up
Delete
End
Page Down
Num Lock
Home
Pg Up
End
Pg Dn
Ins
Del
Enter

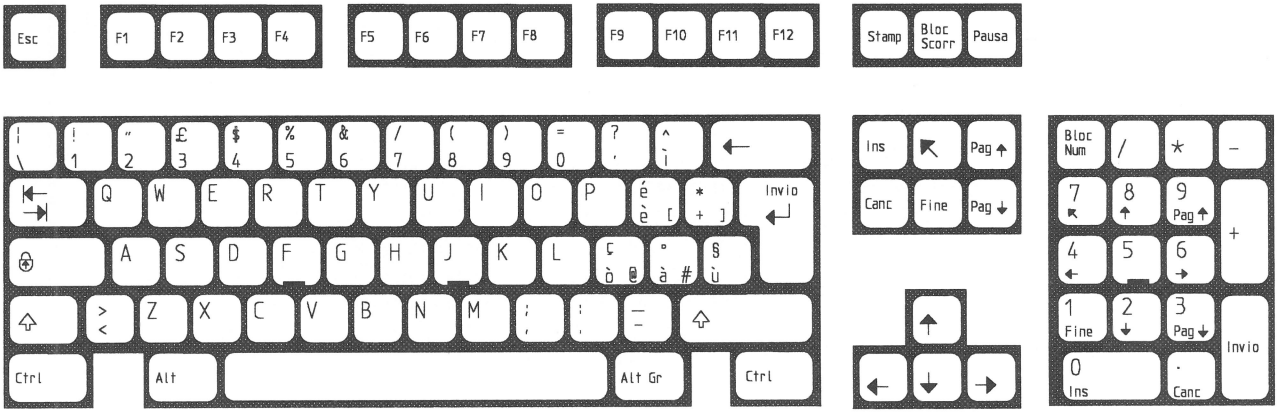
French

Échap
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Impr écran
Arrêt défil
Pause
Entrée
Ctrl
Alt
Alt Gr
Ctrl
Insert
Suppr
Fin
Verr num
Entr
Inser
Suppr

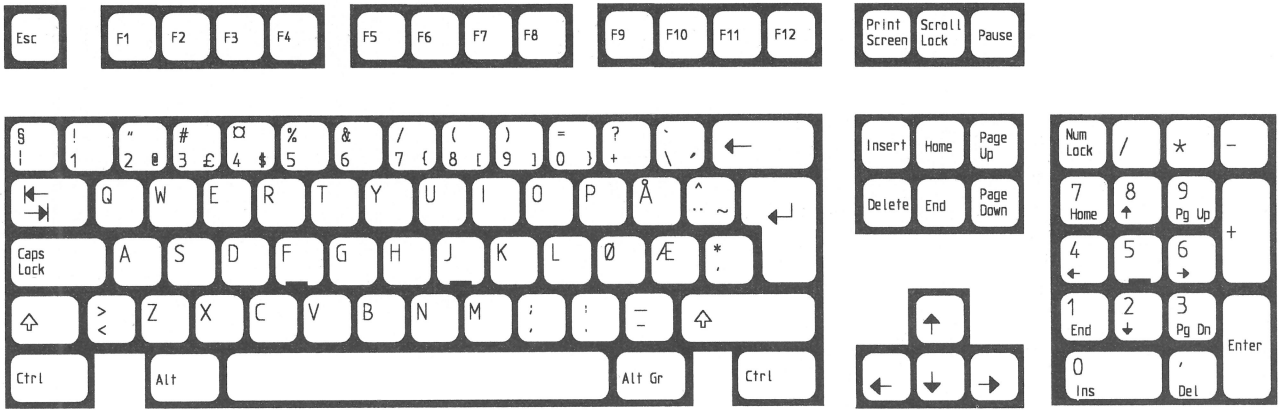

Esc
F1 F2 F3 F4
F5 F6 F7 F8
F9 F10 F11 F12
Druck Rollen Pause
Einfg Pos1 Bild
Entf Ende Bild
Num
Pos1
Bild
Ende
Bild
Enter
Einfg
Entf
Strg
Alt
Alt Gr
Strg

Israeli

Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Print Screen
Scroll Lock
Pause
~ ;
! 1
@ 2
# 3
$ 4
% 5
^ 6
& 7
* 8
(9
) 0
_ -
+ =
Backspace
Insert
Home
Page Up
Num Lock
/
*
-
B Tab Tab
Q
W
E
R
T
Y
U
I
O
P
{ [
}]
| \
Delete
End
Page Down
7 Home
8
9 Pg Up
+
Caps Lock
A
S
D
F
G
H
J
K
L
: ;
" '
Enter
4
5
6
Shift
Z
X
C
V
B
N
M
< ,
> .
? /
Shift
1 End
2
3 Pg Dn
Enter
Ctrl
Alt
Alt
Ctrl
0 Ins
. Del

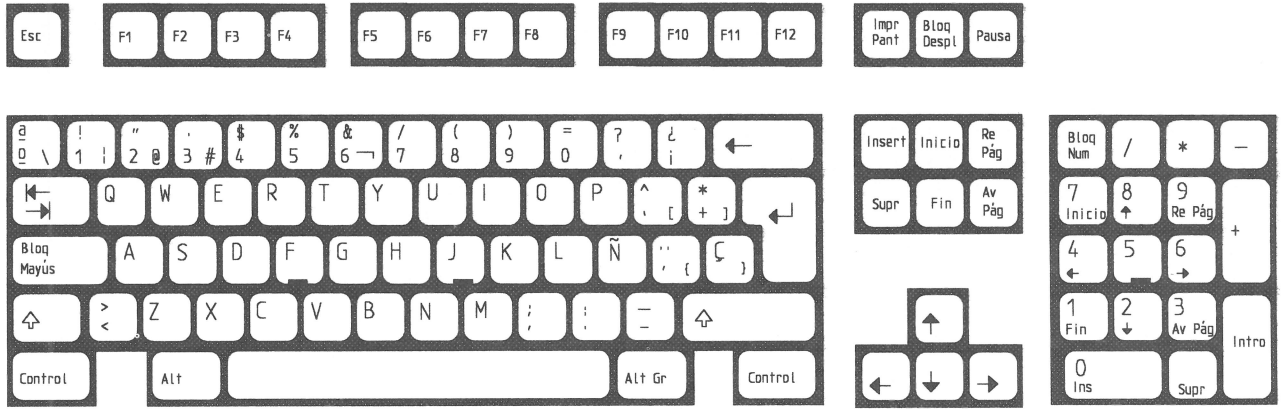

Esc
F1 F2 F3 F4
F5 F6 F7 F8
F9 F10 F11 F12
Stamp Bloc Scorr Pausa
Invio
Ctrl Alt Alt Gr Ctrl
Ins Pag
Canc Fine Pag
Bloc Num
Pag
Fine
Invio
Ins
Canc

Latin American


Esc
F1 F2 F3 F4
F5 F6 F7 F8
F9 F10 F11 F12
Print Screen
Scroll Lock
Pause
§ ½
! 1
" 2 @
3 £
¤ 4 $
% 5
& 6
/ 7 {
(8 [
) 9]
= 0 }
? + \ ´
` ¨
Q W E R T Y U I O P Å ^ ~
Caps Lock
A S D F G H J K L Ø Æ *
> <
Z X C V B N M ; , : . _ -
Ctrl
Alt
Alt Gr
Ctrl
Insert
Home
Page Up
Delete
End
Page Down
Num Lock
/
*
-
7 Home
8
9 Pg Up
+
4
5
6
1 End
2
3 Pg Dn
Enter
0 Ins
, Del

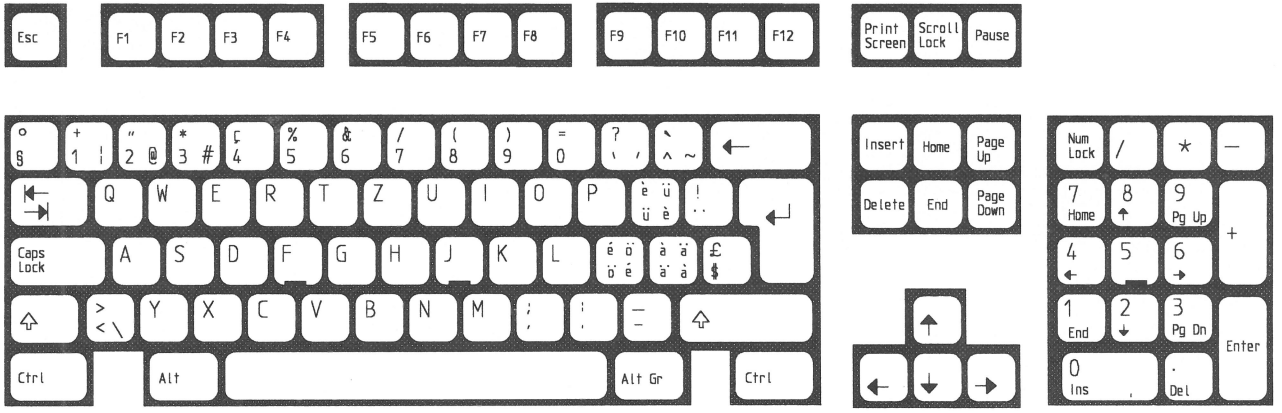
Portuguese

Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Print Screen
Scroll Lock
Pause
Insert
Home
Page Up
Delete
End
Page Down
Num Lock
Caps Lock
Alt
Alt Gr
Ctrl
Enter
Pg Up
Pg Dn
Del
Ins


Esc
F1 F2 F3 F4
F5 F6 F7 F8
F9 F10 F11 F12
Impr Pant
Bloq Despl
Pausa
Insert Inicio Re Pág
Supr Fin Av Pág
Bloq Num
Bloq Mayús
Control
Alt
Alt Gr
Control
Intro

Swedish

Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Print Screen
Scroll Lock
Pause
½ § 1 ! 2 " @ 3 # £ 4 ¤ $ 5 % 6 & 7 / { 8 ([9)] 0 = } + ? \ ´ `
Tab
Q W E R T Y U I O P Å ^ ~ ¨
Enter
Caps Lock
A S D F G H J K L Ö Ä * '
> < | Z X C V B N M ; , : . - _
Ctrl
Alt
Alt Gr
Ctrl
Insert
Home
Page Up
Delete
End
Page Down
Num Lock
÷ / × * -
7 Home 8 ↑ 9 Pg Up +
4 ← 5 6 →
1 End 2 ↓ 3 Pg Dn Enter
0 Ins , Del


Esc
F1 F2 F3 F4
F5 F6 F7 F8
F9 F10 F11 F12
Print Screen
Scroll Lock
Pause
Caps Lock
Ctrl
Alt
Alt Gr
Ctrl
Insert
Home
Page Up
Delete
End
Page Down
Num Lock
Enter

U.K. English

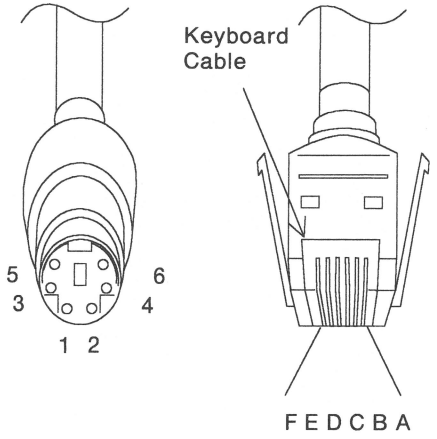
Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Print Screen
Scroll Lock
Pause
~ \
! 1
@ 2
# 3
$ 4
% 5
^ 6
& 7
* 8
(9
) 0
_ -
+ =
Backspace
Tab
Q
W
E
R
T
Y
U
I
O
P
{ [
}]
| \
Caps Lock
A
S
D
F
G
H
J
K
L
: ;
" '
Enter
Shift
Z
X
C
V
B
N
M
< ,
> .
? /
Shift
Ctrl
Alt
Alt
Ctrl
Insert
Home
Page Up
Delete
End
Page Down
Num Lock
/
*
-
7 Home
8
9 Pg Up
+
4
5
6
1 End
2
3 Pg Dn
Enter
0 Ins
. Del

U.S. English

Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Print Screen
Scroll Lock
Pause
~ `
! 1
@ 2
# 3
$ 4
% 5
^ 6
& 7
* 8
(9
) 0
_ -
+ =
Backspace
Tab
Q
W
E
R
T
Y
U
I
O
P
{ [
}]
| \
Caps Lock
A
S
D
F
G
H
J
K
L
: ;
" '
Enter
Shift
Z
X
C
V
B
N
M
< ,
> .
? /
Shift
Ctrl
Alt
Alt
Ctrl
Insert
Home
Page Up
Delete
End
Page Down
Num Lock
/
*
-
7 Home
8
9 Pg Up
+
4
5
6
1 End
2
3 Pg Dn
Enter
0 Ins
. Del

Cables and Connectors

The keyboard cable connects to the system with a 6-pin miniature DIN connector, and to the keyboard with a 6-position connector. The following table shows the pin configuration and signal assignments.



DIN Connector Pins	Signal Name	Connector Pins
1	+KBD DATA	B
2	Reserved	F
3	Ground	C
4	+5.0 Vdc	E
5	+KBD CLK	D
6	Reserved	A
Shield	Frame Ground	Shield

Figure 4-12. Keyboard Cable Connectors

Specifications

The specifications for the keyboards are:

Power Requirements

- +5 Vdc ± 10%
- Current cannot exceed 275 mA.

Size

- Length: 492 millimeters (19.4 inches)
- Depth: 210 millimeters (8.3 inches)
- Height: 58 millimeters (2.3 inches), legs extended

Weight

- 2.25 kilograms (5.0 pounds)

SECTION 5. System BIOS

System BIOS Usage 5-3
Parameter Passing 5-4
Hardware Interrupts 5-4
Software Interrupts 5-5
Vectors with Special Meanings 5-6
Interrupt Interface Listing 5-11
Interrupt 02H - Non-Maskable Interrupt Routine 5-11
Interrupt 05H - Print Screen 5-11
Interrupt 08H - System Timer 5-12
Interrupt 09H - Keyboard 5-13
Interrupt 10H - Video 5-14
Interrupt 11H - Equipment Determination 5-29
Interrupt 12H - Memory Size Determine 5-30
Interrupt 13H - Diskette 5-31
Interrupt 13H - Fixed Disk 5-38
Interrupt 14H - Asynchronous Communications 5-46
Interrupt 15H - System Services 5-51
Interrupt 16H - Keyboard 5-58
Interrupt 17H - Printer 5-61
Interrupt 19H - Bootstrap Loader 5-62
Interrupt 1AH - System and Real-Time Clock Services 5-63
BIOS Data Area and Locations 5-66
Extended BIOS Data Area 5-71
ROM Tables 5-72
Fixed Disk Parameter Table 5-72
Asynchronous Baud Rate Initialization Table 5-74
Diskette Parameter Table 5-74
Model Byte 5-75

Notes:

System BIOS Usage

The basic input/output system (BIOS) resides in ROM on the system board and provides device-level control for the major I/O devices in the system. Additional ROM modules may be located on adapters to provide device-level control for that adapter. BIOS routines enable the assembler language programmer to perform block (disk and diskette) or character-level I/O operations without concern for device address and operating characteristics. System services, such as time-of-day and memory size determination, are provided by the BIOS.

The goal of BIOS is to provide an operational interface to the system and relieve the programmer of the concern about the characteristics of hardware devices. The BIOS interface insulates the user from the hardware, allowing new devices to be added to the system, yet retaining the BIOS level interface to the device. In this manner, hardware modifications and enhancements are transparent to user programs.

The IBM Personal Computer *Macro Assembler* manual and the IBM Personal Computer *Disk Operating System (DOS)* manual provide useful programming information related to this section. A description of the BIOS interface is given in this section.

Access to the BIOS is through the 8086 software interrupts. The software interrupts hex 10 through 1A each access a different BIOS routine. For example, to determine the amount of memory available in the system,

INT 12H

invokes the BIOS routine for determining memory size and returns the value to the caller.

Parameter Passing

All parameters passed to and from the BIOS routines go through the microprocessor registers. The description of each BIOS function shows the registers used on the Call and the Return. For the memory size example, no parameters are passed. The memory size, in 1K increments, is returned in the AX register.

If a BIOS function has several possible operations, the AH register is used as input to indicate the desired operation. For example, to set the time of day, the following code is required:

```
MOV  AH,1           ;Function is to set time of day
MOV  CX,HIGH_COUNT ;Establish the current time
MOV  DX,LOW_COUNT
INT  1AH            ;Set the time
```

To read the time of day:

```
MOV  AH,0           ;Function is to read time of day
INT  1AH            ;Read the timer
```

Generally, the BIOS routines save all registers except for AX and the flags. Other registers are modified on return only if they are returning a value to the caller. The exact register use is in the description of each BIOS function.

Hardware Interrupts

For the hardware interrupt assignments, see the Software Interrupt Listing table later in this section. Interrupt level 0 corresponds to interrupt vector 8, level 1 to interrupt vector 9, and so forth, including interrupt level 7 which corresponds to interrupt vector 0F.

For information about sharing interrupts, see "Interrupt Sharing" in Section 1.

Software Interrupts

With the advent of software interrupt sharing, software interrupt routines can "daisy chain" BIOS interrupts hex 10 through 1F similar to hardware interrupt routines. The routine must check the function value in AH and, if the value is not in the routine's range of function calls, transfer control to the next routine in the chain.

Int	Address in Hex	Name
0	0-3	Divide by Zero
1	4-7	Single Step
2	8-B	Non-Maskable
3	C-F	Breakpoint
4	10-13	Overflow
5	14-17	Print Screen
6	18-1B	Reserved
7	1C-1F	Reserved
8	20-23	Timer
9	24-27	Keyboard
A	28-2B	Reserved
B	2C-2F	Communications
C	30-33	Communications
D	34-37	Fixed Disk
E	38-3B	Diskette
F	3C-3F	Printer
10	40-43	Video BIOS
11	44-47	Equipment Check
12	48-4B	Memory
13	4C-4F	Diskette/Fixed Disk
14	50-53	Communications
15	54-57	System Services
16	58-5B	Keyboard
17	5C-5F	Printer
18	60-63	Resident BASIC
19	64-67	Bootstrap
1A	68-6B	Time of Day
1B	6C-6F	Keyboard Break
1C	70-73	Timer Tick
1D	74-77	Video
1E	78-7B	Diskette Parameters
1F	7C-7F	Video Graphics Chars
40	100-103	Diskette Pointer Save Area for Fixed Disk
41	104-107	Fixed Disk Parameters
42	108-10B	Video
43	10C-10F	Character Graphics Table
46	118-11B	Extended Disk Parameters
4A	128-12B	Real-time Clock Alarm
60-67	180-19F	Reserved for User Programs

Figure 5-1. Software Interrupt Listing

Vectors with Special Meanings

Interrupt Hex 1B - Keyboard Break Address: This vector points to the code to be used when the Ctrl and Break keys are pressed on the keyboard. The vector is invoked while responding to the keyboard interrupt, and control is returned through an IRET instruction. The power-on routines initialize this vector to an IRET instruction; nothing occurs when the Ctrl and Break keys are pressed unless the application program sets a different value.

Control may be retained by this routine, with the following considerations. The Break may have occurred during interrupt processing, so that one or more End of Interrupt commands must be sent to the interrupt controller. Also, all I/O devices should be reset in case an operation was underway at that time.

Interrupt Hex 1C - Timer Tick: This vector points to the code to be executed on every timer-tick interrupt. This vector is invoked while responding to the timer interrupt, and control is returned through an IRET instruction. The power-on routines initialize this vector to point to an IRET instruction; nothing occurs unless the application modifies the pointer. It is the responsibility of the application to save and restore all registers that are modified.

Interrupt Hex 1D - Video Parameters: This vector is maintained for compatibility with earlier IBM display adapters. For the current video parameters, see "Alternate Parameter Table" in Section 1.

Interrupt Hex 1E - Diskette Parameters: This vector points to a data region containing the parameters required for the diskette drive. The power-on routines initialize the vector to point to the parameters contained in the BIOS diskette routine. These default parameters represent the specified values for any IBM drives attached to the system. Changing this parameter block may be necessary to reflect the specifications of other drives attached.

Interrupt Hex 1F - Graphics Character Extensions: When operating in the graphics modes, the read/write character interface forms the character from the ASCII code point, using a set of dot patterns. The dot patterns for the first 128 code points are contained in BIOS. To gain access to the second 128 code points, this vector must be established to point at a table of up to 1024 bytes, where each code point is represented by 8 bytes of graphic information. At power-on, this vector is initialized to 000:0, and it is the responsibility of the application to change this vector if additional code points are needed.

Interrupt Hex 40 - Reserved: When a fixed disk is installed, the BIOS routines use interrupt hex 40 to revector the diskette pointer.

Interrupt Hex 41 - Fixed Disk Parameters: This vector points to a data region containing the parameters required for the fixed disk drive. The power-on routines initialize the vector to point to the parameters contained in the ROM disk routine. These default parameters represent the specified values for the fixed disk drive. Changing this parameter block may be necessary to reflect the specifications of other fixed disk drives attached.

Other Read/Write Memory Usage: The BIOS routines use 256 bytes of memory from absolute hex 400 to 4FF. This memory is called the BIOS data area. The routines also use an expandable memory segment called the extended BIOS data area. Location hex 40E and 40F in the BIOS data Area points to the extended data area. Both memory segments are defined later in this section.

Memory locations hex 300 to 3FF are used as a stack area during the power-on initialization and during bootstrap when it receives control from power-on. The application can set its own stack area.

Interrupt	Address	Function
20	80-83	DOS Program Terminate
21	84-87	DOS Function Call
22	88-8B	DOS Terminate Address
23	8C-8F	DOS Ctrl Break Exit Address
24	90-93	DOS Irrecoverable Error Vector
25	94-97	DOS Absolute Disk Read
26	98-9B	DOS Absolute Disk Write
27	9C-9F	DOS Terminate, Fix in Storage
28-3F	A0-FF	Reserved for DOS
40-5F	100-17F	Reserved for BIOS
60-67	180-19F	Reserved for User Program Interrupts
68-6F	1A0-1BF	Reserved
70	1C0-1C3	Real-Time Clock Interrupt
71-7F	1E0-1FF	Reserved
80-85	200-217	Reserved for BASIC
86-F0	218-3C3	Used by BASIC Interpreter while BASIC is Running
F1-FF	3C4-3FF	Reserved

Figure 5-2. BASIC and DOS Interrupts

Address	Mode	Function
400-4A0	BIOS	See BIOS Data Area
4A1-4EF		Reserved
4F0-4FF		Reserved as Intraapplication Communication Area for any Application
500-5FF		Reserved for DOS and BASIC
500	DOS	Print Screen Status Flag Store 00 = Print Screen Not Active, or Successful Print Screen Operation 01 = Print Screen in Progress FF = Error Encountered during Print Screen Operation
504	DOS	Single Drive Mode Status Byte
510-511	BASIC	BASIC Segment Address Store
512-515	BASIC	Clock Interrupt Vector Segment:Offset Store
516-519	BASIC	Break Key Interrupt Vector Segment:Offset Store
51A-51D	BASIC	Disk Error Interrupt Vector Segment:Offset Store

Figure 5-3. Reserved Memory Locations

If you do Default Work Space Segment:

Offset	Length	Function
2E	2	Line Number of Current Line being Executed
30	2	Offset into Start of Program Text
4E	1	Character Color in Graphics Mode*
6A	1	Keyboard Buffer Contents 0=No Characters in Buffer 1=Characters in Buffer
347	2	Line Number of Last Error
358	2	Offset into Start of Variables (End of Program Text 1-1)

* Set to 1, 2, or 3 to get text in colors 1-3. The default is 3.

Figure 5-4. BASIC Workspace Variables

Starting Address	Function
00000	BIOS Interrupt Vectors
00080	Available Interrupt Vectors
00400	BIOS Data Area
00500	User Read/Write Memory
F0000	Read-Only Memory

Figure 5-5. BIOS Memory Map

BIOS Programming Considerations

Warning: When using an in-circuit emulator in place of the system microprocessor, take care to prevent the request/grant signals between the emulator and the system support gate array from getting out of synchronization. Damage to the gate array will result.

The BIOS code is invoked through software interrupts. The programmer should not "hard code" BIOS addresses into application programs. The internal workings and absolute addresses within BIOS are subject to change without notice.

If an error is reported by the disk or diskette code, reset the drive adapter and retry the operation. A specified number of retries is needed on diskette reads to ensure the problem is not due to motor startup or head settling.

When altering I/O-port bit values, the programmer should change only the bits necessary to the current task. When finished, the programmer should restore the original environment. Not following these guidelines may cause incompatibility with present and future applications.

Adapters with System-Accessible ROM Modules: The ROM BIOS provides a means to integrate ROM code on adapters into the system's code. During the POST, interrupt vectors are established for the BIOS calls. After the default vectors are in place, a scan for additional ROM modules takes place. At this point, a ROM routine on the adapter gains control and establishes or intercepts interrupt vectors to hook itself into the system's code.

During POST, the absolute addresses hex C0000 through EFFFF are scanned in 2K increments searching for valid adapter ROM. Addresses hex C0000 through C7FFF are scanned before the video is initialized and hex C8000 through EFFFF are scanned at the end of POST. A valid ROM is defined as follows:

Byte 0 Hex 55
Byte 1 Hex AA
Byte 2 A length indicator representing the number of 512-byte blocks in the ROM (length/512). A checksum is also done to test the integrity of the ROM module. Each byte in the defined ROM is summed modulo hex 100. This sum must be 0 for the module to be deemed valid.

When the POST identifies a valid ROM, it does a Far Call to byte 3 of the ROM (which should be executable code). The adapter may now perform its power-on initialization tasks. The feature ROM should return control to the BIOS routines by executing a Far Return.

Interrupt Interface Listing

The following contains the BIOS interrupts and their registers used on the Call and Return.

Interrupt 02H - Non-Maskable Interrupt Routine

This routine attempts to find the storage location containing the bad parity. If found, the segment address is displayed; if not found, four question marks are displayed. An NMI is generated by a system memory or I/O channel memory failure.

Interrupt 05H - Print Screen

This logic is invoked to print the screen. The cursor position at the time this routine is invoked is saved and restored upon completion. The routine is intended to run with interrupts enabled. If a subsequent print screen key is pressed while this routine is printing, it is ignored. The base printer status is checked for not busy and not out of paper. An initial status error aborts the print request. Address 50:00 contains the status of the print screen:

```
50:0    = 00    Print screen has not been called, or upon return
                from a call, indicates a successful operation.
        = 01    Print screen is in progress - ignore this request.
        = FF    Error encountered during printing.
```

Interrupt 08H - System Timer

This routine handles the timer interrupt from channel 0 of the timer. The input frequency is 1.19318 MHz and the divisor is 65536, resulting in approximately 18.2 interrupts every second.

The interrupt handler:

- Maintains a count of interrupts (doubleword at 40:6C) since power-on time, which may be used to establish time of day. If the system has been powered on for 24 hours, the overflow flag at 40:70 is increased. The day counter word at 40:CE must be updated when the time counter crosses a day boundary.
- Decrements the motor control count (40:40) of the diskette. Upon reaching 0, it turns off the diskette drive motor, and resets the motor running flags.
- Invokes a user routine through interrupt hex 1C at every timer tick. The user must code a routine and place the correct address in the vector table.

Interrupt 09H - Keyboard

This routine is invoked upon the make or break of every keystroke.

For ASCII keys, when a make code is read from port 60, the character and scan code are placed in the keyboard buffer (40:1E for a length of 32 bytes) at the address pointed to by the buffer tail pointer word at 40:1C. The buffer tail pointer is then increased by 2 unless it wraps past the end of the buffer, in which case it is reinitialized to the start of the buffer.

For shift keys, the keyboard flags are updated accordingly on makes or breaks.

For the sequence Ctrl-Alt-Del, the handler sets the memory-test-complete word at 40:72 to hex 1234; then jumps to POST. POST checks the memory-test-complete word and does not retest memory if hex 1234 exists.

For the Pause key, the handler loops until a valid ASCII keystroke is pressed.

For the Print Screen key, interrupt 05 is invoked to print the screen.

For a Ctrl-Break sequence, the control break interrupt handler, interrupt hex 1B, is invoked.

For the System Request key, interrupt hex 15 is invoked with (AH) = 85H; for Interrupt Complete, interrupt hex 15 is invoked with (AH) = 91H.

The keyboard intercept is handled through interrupt 15, with (AH) = 4FH.

Interrupt 10H - Video

(AH) = 00H Set Mode

The AL register contains the mode value; if bit 7 in AL is set, the video buffer is not cleared.

The cursor is not supported in graphics modes. With an analog display, each color has 256K possibilities.

For the graphics modes 4, 5, 6, and 13, the font is an 8x8 character box that is double scanned to generate the 8x16 character. The box size refers to the font supported by BIOS.

Mode in Hex	Type	Colors	Alpha Format	Buffer Start	Box Size	No. Pages	PEL Dimensions
0, 1	A/N	16	40x25	B8000	8x16	8	320x400
2, 3	A/N	16	80x25	B8000	8x16	8	640x400
4, 5	APA	2	40x25	B8000	8x8	1	320x200
6	APA	2	80x25	B8000	8x8	1	640x200
11	APA	2	80x30	A0000	8x16	1	640x480
13	APA	256	40x25	A0000	8x8	1	320x200

(AH) = 01H Set Cursor Type

BIOS maintains only one cursor type for all video pages. If an application requires that different cursor types be preserved for different pages, it must maintain the types itself.

When operating with 400 scan lines, the hardware modifies the cursor type as follows:

Start line = (CH)*2
End line = [(CL)*2-1]

```
(CH) -  Bits 4-0 = Start line for cursor
         Hardware controls the cursor blink
(CL) -  Bits 4-0 = End line for cursor
```

(AH) = 02H Set Cursor Position

```
(DH,DL) - Row,column  (00,00) is upper left
(BH) - Page number (00 for graphics)
```

(AH) = 03H Read Cursor Position

```
(BH) - Page number (00 for graphics)

ON RETURN:
(DH,DL) - Row,column of cursor for
           requested page
(CH,CL) - Cursor mode currently set
```

(AH) = 04H Reserved

(AH) = 05H Select Active Display Page

This function is valid for alphanumeric modes (only these modes support more than one page).

```
(AL) - New page value
       0-7 for modes 0,1
       0-3 for modes 2,3
```

(AH) = 06H Scroll Active Page Up

```
(AL) - Number of lines; input lines blanked
        at bottom of window.
       AL = 0 means blank entire window
(BH) - Attribute to be used on blank line
(CH,CL) - Row,column of upper left corner of scroll
(DH,DL) - Row,column of lower right corner of scroll
```

(AH) = 07H Scroll Active Page Down

```
(AL) - Number of lines, input lines blanked at top
        of window
       AL = 0 means blank entire window
(BH) - Attribute to be used on blank line
(CH,CL) - Row,column of upper left corner of scroll
(DH,DL) - Row,column of lower right corner of scroll
```

(AH) = 08H Read Attribute/Character at Current Cursor Position

```
(BH) - Display page (alpha)

ON RETURN:
(AH) - Attribute of character read (alpha)
(AL) - Character read
```

(AH) = 09H Write Attribute/Character at Current Cursor Position

These two functions, (AH) = 09H and 0AH, are similar. The function (AH)=09 is used for the graphics modes. For the read/write character interface in graphics mode, the characters are formed from a character image maintained in the system ROM, which contains only the first 128 characters. To read or write the second 128 characters, the user must initialize the pointer at interrupt 1F (location 0007C) to point to the table containing the code points for the second 128 characters (128-255). For the graphics modes 11 and 13, 256 graphics characters are supplied in the system ROM.

For the write character interface in graphics mode, the character count in CX produces valid results only for characters on the same row. Continuation to following lines will not produce correctly.

For graphics modes other than mode 13, if bit 7 in BL is set to 1, the color value is exclusively OR'd with the current contents of the video memory.

```
(AL) - Character to write
(BH) - Display page (alpha)
(BL) - Attribute of character (alpha)/color
       of character (graphics)
(CX) - Count of characters to write
```

(AH) = 0AH Write Character Only at Current Cursor Position

```
(AL) - Character to write
(BH) - Display page (alpha)
(CX) - Count of characters to write
```

(AH) = 0BH Set Color Palette

The following are the results for modes 4 and 5:

Color ID = 0 selects the background color (0-15)
Color ID = 1 selects the color set to be used.

In the alpha modes, the value set for palette color 0 indicates the border color to be used (decimal values 0-31, where 16-31 select the high intensity background set).

```
(BH) - Palette color ID being set (0-127)
(BL) - Color value to be used with that color ID
```

(AH) = 0CH Write Dot

If bit 7 of AL is set to 1, the color value is exclusively OR'd with the current contents of the dot (except mode 13).

```
(AL) - Color value
(BH) - Display page (alpha)
(CX) - Column number
(DX) - Row number
```

(AH) = 0DH Read Dot

```
        (BH) - Display page (alpha)
        (CX) - Column number
        (DX) - Row number

ON RETURN:
        (AL) - Dot read
```

(AH) = 0EH Write Teletype to Active Display

The screen width is controlled by the previous Mode Set.

```
        (AL) - Character to write
        (BL) - Foreground color in graphics mode
```

(AH) = 0FH Current Video State

```
ON RETURN:
        (AH) - Number of character columns on the screen
        (AL) - Mode currently set (see AH=00)
        (BH) - Current active display page
```

(AH) = 10H Color Palette Interface

(AL) = 00H Set Individual Register

On the Model 30, this routine is used only to inhibit bit 3 of the attribute byte when 512 characters are active to provide eight consistent colors. The only value supported is (BX) = 0712H.

```
        (BH) - Value to set
        (BL) - Register to set
```

(AL) = 03H Toggle Intensify/Blinking Bit

```
        (BL) = 00H  Enable intensify
             = 01H  Enable blinking
```

(AL) = 10H Set Individual Color Register

```
        (BX)= Color register to set
        (DH)= Red value to set
        (CH)= Green value to set
        (CL)= Blue value to set
```

(AL) = 12H Set Block of Color Registers

The table format is red value, green value, blue value.

```
(ES:DX)= Pointer to a table of color values
(BX)= First color register to set
(CX)= Number of color registers to set
```

(AL) = 15H Read Individual Color Register

```
(BX)= Color register to read
ON RETURN:
(DH)= Red value returned
(CH)= Green value returned
(CL)= Blue value returned
```

(AL) = 17H Read Block of Color Registers

The table format is red value, green value, blue value.

```
(ES:DX)= Pointer to a destination table
          for values
(BX)= First color register to be read
(CX)= Number of color registers to be read
```

(AL) = 1BH Sum Color Values to Grey Shades

This routine reads the red, green, and blue values found in the color registers, performs a weighted sum (30% red, 59% green, and 11% blue), then writes the result into the red, green, and blue components of the color register. The original data in each color register is not retained; if those values are needed later, they must be preserved by the calling routine.

```
(BX)= First color register to sum
(CX)= Number of color registers to sum
```

(AH) = 11H Character Generator Load

This function initiates a mode set, completely resetting the video environment but maintaining the regen buffer.

(AL) = 00H User Alpha Load

BH contains the value hex 10 for normal operation. If (BH) = 0EH, the characters are extended to 16 high by extending the last line of the 14 high character.

```
(ES:BP)= Pointer to user table
(BH)= Number of bytes per character
(BL)= Block to load
(CX)= Count to store
(DX)= Character offset into table
```

(AL) = 01H Reserved If called, (AL) = 04H is executed.

(AL) = 02H ROM 8x8 Double Dot Font

```
(BL)= Block to load
```

(AL) = 03H Set Block Specifier

This routine is executed after loading a font to make that character font active. This routine is valid in alpha modes only. For more information on block specifier, see "RAM Loadable Font" in Section 1.

When 512 characters are active, a function call with (AX) = 1000H and (BX) = 0712H should be executed to set eight consistent colors.

```
(BL)= Character genenerator block selects
```

(AL) = 04H ROM 8x16 Font

```
(BL)= Block to load
```

(AL) = 10H Reserved If called, (AL) = 00H is executed.
(AL) = 11H Reserved If called, (AL) = 04H is executed.
(AL) = 12H Reserved If called, (AL) = 02H is executed.
(AL) = 14H Reserved If called, (AL) = 04H is executed.

(AL) = 20H User Graphics Chars (INT 1FH - 8x8)

The following routines are designed to be called only immediately after a Mode Set. Performing them at any other time will have undetermined results.

```
(ES:BP) - Pointer to user table
```

(AL) = 21H User Graphics Chars

```
(ES:BP) - Pointer to user table
(CX)  - Points (bytes per character)
(BL)  - Row specifier
      = 00 - User specified in DL
      = 01  (0EH) - 14
      = 02  (19H) - 25
      = 03  (2BH) - 43
```

(AL) = 22H Reserved If called, (AL) = 24H is executed.

(AL) = 23H ROM 8x8 Double Dot Font

```
(BL)= Row specifier
```

(AL) = 24H ROM 8x16 Font

```
(BL)= Row specifier
```

(AL) = 30H Information

```
(CX)= Points
(DL)= Rows

ON RETURN:
    (ES:BP)= Pointer to table
```

(BH) = 00H Return Current INT 1F Pointer

(BH) = 01H Return Current INT 44 Pointer

(BH) = 02H Reserved If called, (BL) = 06H is executed.

(BH) = 03H Return ROM 8x8 Font Pointer

(BH) = 04H Return ROM 8x8 Font Pointer (Top)

(BH) = 06H Return ROM Alpha Alternate 8x16

(AH) = 12H Alternate Select

(BL) = 20H Select Alternate Print Screen Routine

(BL) = 31H Default Palette Loading During Mode Set

The color registers are not altered during Mode Set if disable default palette loading is selected.

```
        (AL) = 00  Enable default palette loading
             = 01  Disable default palette loading

    ON RETURN:
        (AL) = 12H  Function supported
```

(BL) = 32H Video

The routine enables and disables the address decode for the video I/O port and regen buffer.

```
        (AL) = 00  Enable video
             = 01  Disable video

    ON RETURN:
        (AL) = 12H  Function supported
```

(BL) = 33H Summing to Gray Shades

When enabled, summing occurs during the loading of the color palette during mode set and color palette interface routines.

```
        (AL) = 00  Enable summing
             = 01  Disable summing

    ON RETURN:
        (AL) = 12H  Function supported
```

(BL) = 35H Display Switch

When the system video and adapter video have the same BIOS data areas and hardware capabilities, they are in conflict. If POST detects the conflict, the system video is disabled and the adapter video becomes the primary.

This routine allows switching the active display between these two videos. The following shows the procedure when initially switching to the system video:

1. Initial Feature Video Off, (AL) = 00.
2. Initial System Video On, (AL) = 01.

Afterwards, switching displays is done through the sequence:

1. Switch Off Active Video, (AL) = 02
2. Switch On Inactive Video, (AL) = 03

Switching off the active video disables the video function that is active at that time. The Switch State buffer saves the video state information used when that video is reactivated.

Switching on the inactive video enables the video function that was inactive, using its buffer to retrieve the video information.

All subroutines under display switching return a value of hex 12 to indicate that the function is supported.

(AL) = 00H Initial Feature Video Off

(ES:DX)- Pointer to Switch State buffer of 128 bytes

(AL) = 01H Initial System Video On

(AL) = 02H Switch Active Video Off

(ES:DX)- Pointer to Switch State buffer

(AL) = 03H Switch Inactive Video On

(ES:DX)- Pointer to Switch State buffer saved earlier

(AH) = 13H Write String

CAR RET, LINE FEED, BACKSPACE, and BELL are treated as commands rather than printable characters.

```
(ES:BP)= Pointer to string to be written
(CX)= Character only count
(DX)= Position to begin string, in cursor terms
(BH)= Page number
```

(AL) = 00H Write Character String

```
(BL)= Attribute
      String is (CHAR, CHAR, CHAR, ...)
      Cursor not moved
```

(AL) = 01H Write Character String and Move Cursor

```
(BL)= Attribute
      String is (CHAR, CHAR, CHAR, ...)
      Cursor not moved
```

(AL) = 02H Write Character and Attribute Strings

This function is for alpha modes only.

```
String - (CHAR, ATTR, CHAR, ATTR, ...)
Cursor not moved
```

(AL) = 03H Write Character And Attribute Strings

This function is for alpha modes only.

```
STRING - (CHAR, ATTR, CHAR, ATTR, ...)
CURSOR IS MOVED
```

(AH) = 1AH Read/Write Display Combination Code

(AL) = 00H Read Display Combination Code

Display Code	Description
00H	No Display
0BH	Analog Monochrome
0CH	Analog Color

```
ON RETURN:
     (AL) = 1AH - Function supported
     (BL) - Active display code
     (BH) - Alternate display code
```

(AL) = 01H Write Display Combination Code

```
     (BL) - Active display code
     (BH) - Alternate display code

ON RETURN:
     (AL) = 1AH - Function supported
```

(AH) = 1BH Return Functionality and Video State Information

The user buffer contains functionality and video state information as described by the requested implementation type. When the implementation type in BX is set to 0, the buffer size is 64 bytes.

```
     (BX)= Implementation type
     (ES:DI)= User buffer pointer for return of
               fuctionality/state information

ON RETURN:
     (AL)= 1BH -Function supported
```

(AH) = 1CH - FFH Reserved

The following is the format of the functionality and state information table for the video. The size is 64 bytes and the offset is shown as the hex value from (DI).

Offset	Size	Description
00	Word	Offset to Static Functionality Table
02	Word	Segment to Static Functionality Table
04	Byte	Video Mode (Refer to AH=00 for Supported Modes)
05	Word	Columns on Screen (No. of Char. Columns)
07	Word	Length of Regen Buffer in Bytes
09	Word	Start Address in Regen Buffer
0B	Word	Cursor Position for 8 Display Pages (Row,Col)
1B	Word	Cursor Mode Setting (Cusor Start/End Value)
1D	Byte	Active Display Page
1E	Word	Controller Address
20	Byte	CRT Mode Set
21	Byte	CRT Palette
22	Byte	Rows on Screen (No. of Char. Lines)
23	Word	Character Height (Scan Lines/Char.)
25	Byte	Display Combination Code (Active)
26	Byte	Display Combination Code (Alternate)
27	Word	No. of Colors Supported for Current Mode
29	Byte	No. of Display Pages Supported for Current Mode
2A	Byte	Scan Lines in Current Mode = 0 - 200 = 1 - 350 = 2 - 400 = 3 - 480
2B - 2C	Byte	Reserved = 0
2D	Byte	Miscellaneous State Information Bit 7,6 - Reserved Bit 5 - Blink Enabled Bit 4 - Reserved = 0 Bit 3 - Default Palette Loading Bit 2 - Monochrome Display Attached Bit 1 - Summing Active Bit 0 - Reserved = 0
2E - 30	Byte	Reserved
31	Byte	Video Memory Available = 0 - 64K = 1 - 128K = 2 - 192K = 3 - 256K
32	Byte	Save Pointer State Information Bits 7-5 Reserved = 0 Bit 4 Palette Override Active Bit 3 Graphics Font Override Active Bit 2 Alpha Font Override Active Bit 1 Dynamic Save Area Active Bit 0 512 Character Set Active
33 - 3F	Byte	Reserved

The following is the format of the static functionality table pointed to by the start of the functionality and state information table. The table is 16 bytes long. A bit is set to 1 if the function is supported.

	Bit	Function
Byte 0		Video Modes (3 Bytes)
	7	Mode 7
	6	Mode 6
	5	Mode 5
	4	Mode 4
	3	Mode 3
	2	Mode 2
	1	Mode 1
	0	Mode 0
Byte 1	7	Mode F
	6	Mode E
	5	Mode D
	4	Mode C
	3	Mode B
	2	Mode A
	1	Mode 9
	0	Mode 8
Byte 2	7-4	Reserved
	3	Mode 13
	2	Mode 12
	1	Mode 11
	0	Mode 10
Bytes 3-6		Reserved
Byte 7		Scan Lines Available in Text Modes
	7-3	Reserved
	2	400
	1	350
	0	200
Byte 8		Character Blocks Available in Text Modes
Byte 9		Max. Number of Active Character Blocks in Text Modes
Byte A		Miscellaneous Functions
	7	Reserved = 0
	6	Color Register, See (AH) = 10
	5	Palette, See (AH) = 10
	4	Reserved = 0
	3	Default Palette Loading, See (AH) = 12
	2	Character Font Loading, See (AH) = 11
	1	Summing
	0	Reserved = 0
Byte B		Miscellaneous Functions
	7-4	Reserved = 0
	3	DCC
	2	Blink Enabled
	1	Reserved = 0
	0	Reserved = 0

	Bit	Function
Bytes C,D		Reserved
Byte E		Save Pointer Functions
	7-5	Reserved = 0
	4	Palette Override
	3	Graphics Font Override
	2	Alpha Font Override
	1	Dynamic Save Area
	0	512 Character Set
Byte F		Reserved

The following is the format for the SAVE _TBL. All entries are doubleword. For more information, see "Alternate Parameter Table" on page 1-71.

Entry	Description
1	Video Parameter Table Pointer This must point to the video parameter table in BIOS
2	Reserved as all 0's
3	Alpha Mode Auxiliary Font Pointer This is a pointer to a descriptor table used during a mode set to select a user font in A/N mode. The table has the following format: **Size** **Description** Byte Bytes per character Byte Block to load, should be 00 for normal operation Word Count to store, should be hex 100 for normal operation Word Character offset, should be 00 for normal operation DWord Pointer to a font table Byte Displayable rows, if the value is FF, the maximum calculated value will be used; otherwise, this value is used. Byte Consecutive bytes of mode values for which this font description is to be used. The end of this stream is indicated by a byte code of FF.
4	Graphics Mode Auxiliary Pointer This is a pointer to a descriptor table used during a mode set to select a user font in graphics mode. The table has the following format: **Size** **Description** Byte Displayable rows Word Bytes per character DWord Pointer to a font table Byte Consecutive bytes of mode values for which this font description is to be used. The end of this stream is indicated by a byte code of FF.
5 - 7	Reserved as all 0's.

Interrupt 11H - Equipment Determination

This routine determines what optional devices are attached. The EQUIP_FLAG variable is set during the power-on diagnostics, using the following hardware assumptions:

```
        Port 3FA - Interrupt ID register (primary)
             2FA - Interrupt ID register (secondary)
                   Bits 7-3 are always 0
        Port 378 - Output port of printer 1
             278 - Output port of printer 2
             3BC - Output port of printer 3

ON RETURN:
       (AX) - Equipment flag
            Bit 15,14   = Number of printers attached
            Bit 13,12   = Reserved
            Bit 11,10,9 = Number of RS232 ports attached
            Bit 8       = Reserved
            Bit 7,6     = Number of diskette drives
                          00=1, 01=2 only if bit 0 = 1
            Bit 5,4     = Initial video mode
                          00 - reserved
                          01 - 40x25 using color
                          10 - 80x25 using color
                          11 - 80x25 using BW
            Bit 3       = Reserved
            Bit 2       = Pointing device attached
            Bit 1       = Math coprocessor installed
            Bit 0       = IPL diskette installed
```

Interrupt 12H - Memory Size Determine

This routine returns the amount of RAM in the system as determined by the POST routines.

The following are the memory determination assumptions:

- All installed memory is functional. If the memory test during POST indicates less, that value becomes the default.
- All memory from 0 to 640K must be contiguous.

```
ON RETURN:
      (AX) - Number of contiguous 1K blocks of memory
```

Interrupt 13H - Diskette

For operations requiring the diskette drive motor, the multitasking hook function (INT 15, AX=90FDH) is called, telling the operating system that the BIOS is waiting for the motor to accelerate, allowing the operating system to perform a different task.

Before waiting for an interrupt, BIOS calls Device Busy (INT 15, AX = 9001H) telling the operating system of the Wait. The complementary Interrupt Complete (INT 15, AX = 9101H) is called indicating the operation is complete.

(AH) = 00H Reset Diskette System

This function issues a hard reset to the controller, then generates a Prepare command. The drive is recalibrated when the next drive operation is initiated.

If an error is reported by the diskette code, the appropriate action is to reset the diskette, then retry the operation.

```
        (DL) - Drive number
             Bit 7 = 0 for diskette (value checked)

ON RETURN:
        (CY) - Set indicates status is nonzero
        (AH) - Status of operation (see Read Status)

        Diskette status at 40:41 = status of operation
```

(AH) = 01H Read Status of Last Operation

```
        (DL) - Drive number
             Bit 7 = 0 for diskette (value checked)

    ON RETURN:
        (CY) - Set indicates status is nonzero
        (AH) - Status of operation
```

(AH)	Error	(AH)	Error
80	Time Out	08	Reserved
40	Seek Failure	06	Media Has Been Changed
20	General Controller Failure	04	Sector Not Found
10	Bad CRC Error	03	Write Protect Error
0C	Unsupported Track, Sectors/Track Combination	02	Bad Address Mark
09	DMA Boundary Error	01	Invalid Function Request
		00	No Error

(AH) = 02H Read Desired Sectors into Memory

The 2 high-order bits in CL are the 2 high-order bits of the 10-bit track number.

If an error is reported by the diskette code, the appropriate action is to reset the diskette, then retry the operation.

```
        (DL) - Drive number,
                  Bit 7 = 0 for diskette (value checked)

        (DH) - Head number, (origin of 0, not value checked)
        (CH) - Track number, (origin of 0, not value checked)
        (CL) - Sector number, (origin of 1, not value checked)
        (AL) - Number of sectors (not value checked)
        (ES:BX) - Address of buffer

    ON RETURN:
        (CY) - Set indicates status is nonzero
        (AL) - Number of sectors actually transferred
        (AH) - Status of operation (see Read Status)

        Diskette status at 40:41 = status of operation
```

(AH) = 03H Write Desired Sectors from Memory

The 2 high-order bits in CL are the 2 high-order bits of the 10-bit track number. If an error is reported by the diskette code, the appropriate action is to reset the diskette, then retry the operation.

```
        (DL) - Drive number,
               Bit 7 = 0 for diskette (value checked)
        (DH) - Head number (origin of 0, not value checked)
        (CH) - Track number (origin of 0, not value checked)
        (CL) - Sector number (origin of 1, not value checked)
        (AL) - Number of sectors (not value checked)
        (ES:BX) - Address of buffer

ON RETURN:
        (CY) - Set indicates status is nonzero
        (AL) - Number of sectors actually transferred
        (AH) - Status of operation (see Read Status)

        Diskette status at 40:41 = status of operation
```

(AH) = 04H Verify Desired Sectors

The 2 high-order bits in CL are the 2 high-order bits of the 10-bit track number. If an error is reported by the diskette code, the appropriate action is to reset the diskette, then retry the operation.

```
        (DL) - Drive number,
               Bit 7 = 0 for diskette (value checked)
        (DH) - Head number (origin of 0, not value checked)
        (CH) - Track number (origin of 0, not value checked)
        (CL) - Sector number (origin of 1, not value checked)
        (AL) - Number of sectors (not value checked)

ON RETURN:
        (CY) - Set indicates status is nonzero
        (AL) - Number of sectors verified
        (AH) - Status of operation (see Read Status)

        Diskette status at 40:41 = status of operation
```

(AH) = 05H Format Desired Track

When using this function, (ES:BX) points to the buffer containing the collection of desired address fields for the track. Each field has 4 bytes with a format as follows:

Byte 0 Track number
Byte 1 Head number
Byte 2 Sector number
Byte 3 Number of bytes per sector

- 00 = 128
- 01 = 256
- 02 = 512
- 03 = 1024

There must be one entry for every sector on the track. This is used to find the requested sector during read/write access. Before formatting a diskette when there is more than one format, Set Media Type (AH = 18H) must be called. If its not called, the default is the maximum capacity of the drive.

The 2 high-order bits in CL are the 2 high-order bits of the 10-bit track number. If an error is reported by the diskette code, the appropriate action is to reset the diskette, then retry the operation.

```
        (DL) - Drive number,
               Bit 7 = 0 for diskette (value checked)
        (DH) - Head number (origin of 0, not value checked)
        (CH) - Track number (origin of 0, not value checked)
        (AL) - Number of sectors (origin of 1, not value checked)
        (ES:BX) - Address of buffer

ON RETURN:
        (CY) - Set indicates status is nonzero
        (AH) - Status of operation (see Read Status)

        Diskette status at 40:41 = status of operation
```

(AH) = 06H - 07H Reserved for Fixed Disk Interface

```
ON RETURN:
        (CY) - Set indicates error
        (AH) - Status of operation = 01 for invalid command

        Diskette status at 40:41 = status of operation
```

(AH) = 08H Read Drive Parameters

There is a parameter table for each supported media type.

```
        (DL) - Drive number,
               Bit 7 = 0 for diskette (value checked)

ON RETURN:
      (ES:DI) - Pointer to 11 byte parameter table
               associated with the maximum supported media types
               on the drive in question.
      (CH) - Low-order 8 of 10 bits maximum number of tracks
               (origin of 0)
      (CL) - Bits 7 & 6 - 2 high-order bits of maximum tracks
           - Bits 5 thru 0 - maximum sectors per track
               (origin of 1)

      (DH) - Maximum head number
               (origin of 0)
      (DL) - Number of diskette drives installed

      (BH) = 0
      (BL) - Bits 7 through 4 = 0
             Bits 3 through 0 - valid drive type
               03 = 720 K,  3.5 in, 80 track

      (AX) = 0

  If the drive number is invalid,
  ES,AX,BX,CX,DH,DI = 0 ; DL = number of drives.
         If no drives are present, DL = 0

      Diskette status 40:41 = 0 and (CY) = 0
```

(AH) = 09H - 14H Reserved for Fixed Disk Interface

```
ON RETURN:
      (CY) - Set indicates error
      (AH) - Status of operation = 01 for invalid command

      Diskette status at 40:41 = status of operation
```

(AH) = 15H Read DASD Type

```
        (DL) - 7-bit drive number, bit 7 = 0 for diskette
               (value checked)

  ON RETURN:
        (AH) = 00 - Drive not present
             = 01 - Diskette, no change line available
             = 02 - Diskette, change line available
             = 03 - Reserved for fixed disk interface

        Diskette status at 40:41 = status of operation
```

(AH) = 16H Disk Change Line Status

```
        (DL) - 7-bit drive number, bit 7 = 0 for diskette
               (value checked)

  ON RETURN:
        (CY) - Set if (AH) is not zero
        (AH) = 00 - Disk change line not active
               01 - Invalid drive number
               06 - Disk change line active

        Diskette status at 40:41 = (AH) on return
```

(AH) = 17H Set DASD Type for Format

The disk change line status is checked for all drives supporting the 'disk change' signal. This function is supported for compatibility purposes, however, Set Media Type for Format, (AH) = 18H, is the suggested function to use.

```
        (DL) - 7-bit drive number, bit 7 = 0 for diskette
               (value checked)
        (AL) = 04 - 720K diskette in a 720K diskette drive

  ON RETURN:
        (CY) - Set indicates error
        (AH) - Status of operation
             = 01 for invalid request

        Diskette status at 40:41 = status of operation
```

(AH) = 18H Set Media Type For Format

This function is called before issuing the first Format Desired Track command. If the diskette is changed, this function is called again. The diskette must be present.

This function monitors the 'disk change' signal. If the signal is active:

1. The logic attempts to reset the signal to the inactive state.
2. If the attempt succeeds, BIOS sets the correct data rate for formatting.
3. If the attempt fails, BIOS returns the time-out error (hex 80) in AH.

There is one parameter table for each supported medium type.

```
        (DL) - 7-bit drive number, bit 7 = 0 for diskette
               (value checked)
        (CH) - Low order 8 of 10 bits, number of tracks
               (origin of 0)
        (CL) - Bits 7 & 6 - 2 high-order bits of number of tracks
             - Bits 5 through 0 - sectors per track
               (origin of 1)

ON RETURN:
        (ES:DI) - Pointer to 11-byte parameter table
                  for this medium type, unchanged if AH is nonzero

        (CY) - Set if track and sectors/track is not supported
        (AH) - Status of operation = 01 for invalid request
```

(AH) = 19H - FFH Reserved

```
ON RETURN:
        (CY) - Set indicates error
        (AH) - Status of operation
             = 01 for invalid command

        Diskette status at 40:41 = status of operation
```

Interrupt 13H - Fixed Disk

This interface provides access to fixed disks through the controller. It is assumed that upon entry to the fixed disk portion of Interrupt 13, bit 7 of the drive number is set, indicating a fixed disk operation. The drive number, (DL), is the only parameter that is value checked.

Before waiting for an interrupt, BIOS calls Device Busy with type = disk (INT 15, AX = 9000H), telling the operating system of the wait. The complementary Interrupt Complete (INT 15, AX = 9100H) is called indicating the operation is complete.

The function number (AH) is also checked for read/write. The sector number (AL) is also checked for a valid range of hex 01 to 80.

Registers will be preserved except when they are used to return values.

(AH) = 00H Reset Disk System

Before waiting on a disk reset, the BIOS calls Device Busy (INT 15, AX = 9000H). The reset is a time-out of approximately 3 seconds. This time-out value depends on the function number.

Diskette reset is invoked for all values of (DL). Disk Reset is invoked only if the drive number is less than or equal to the maximum number of fixed disks.

If an error is reported, reset the disk, then retry the operation.

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk

ON RETURN:
        (CY) - Set indicates status is not zero
        (AH) - Status of operation (see Read Status)

        Disk status at 40:74 = status of operation
```

(AH) = 01H Read Status of Last Operation

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk

ON RETURN:
        (CY) - Set indicates status in AH is not zero
        (AL) - Status of last operation
        (AH) - Status of operation

        Disk status at 40:74 is reset to 0
```

(AH)	Error	(AH)	Error
FF	Sense Operation Failed	0D	Invalid Number Of Sectors On Format
E0	Status Error/Error REG = 0	0B	Bad Track Detected
CC	Write Fault On Selected Drive	0A	Bad Sector Flag Detected
BB	Undefined Error Occurred	09	DMA Boundary Error
AA	Drive Not Ready	08	DMA Failure
80	Time Out	07	Drive Parameter Activity Failed
40	Seek Failure	05	Reset Failed
20	General Controller Failure	04	Sector Not Found
11	ECC Corrected Data Error	03	Write Protect Error (Diskette Only)
10	Bad ECC On Disk Read	02	Bad Address Mark
0E	Controlled Data Address Mark Detected	01	Invalid Function Request
		00	No Error

(AH) = 02H Read Desired Sectors into Memory

The error code 11 indicates that the data read had a recoverable error that was corrected by the ECC algorithm. The data is probably good; however, the BIOS routine indicates an error to allow the controlling program a chance to decide for itself. The error may not recur if the data is rewritten.

The 2 high-order bits of the track number (10 bits total) are the 2 high-order bits of CL.

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk
        (DH) - Head number (origin of 0, not value checked)
        (CH) - Track number (origin of 0, not value checked)
        (CL) - Sector number (origin of 1, not value checked)
        (AL) - Number of sectors
        (ES:BX) - Address of buffer

ON RETURN:
        (CY) - Set indicates status is not zero
        (AH) - Status of operation (see Read Status)

        Disk status at 40:74 = STATUS OF OPERATION
```

(AH) = 03H Write Desired Sectors from Memory

The 2 high-order bits of the track number (10 bits total) are the 2 high-order bits of CL.

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk
        (DH) - Head number (origin of 0, not value checked)
        (CH) - Track number (origin of 0, not value checked)
        (CL) - Sector number (origin of 1, not value checked)
        (AL) - Number of sectors
        (ES:BX) - Address of buffer

ON RETURN:
        (CY) - Set indicates status is not zero
        (AH) - Status of operation (see Read Status)

        Disk status at 40:74 = status of operation
```

(AH) = 04H Verify Desired Sectors

The 2 high-order bits of the track number (10 bits total) are the 2 high-order bits of CL.

```
            (DL) - 7-bit drive number, bit 7 = 1
                   for fixed disk
            (DH) - Head number (origin of 0, not value checked)
            (CH) - Track number (origin of 0, not value checked)
            (CL) - Sector number (origin of 1, not value checked)
            (AL) - Number of sectors

      ON RETURN:
            (CY) - Set indicates status is not zero
            (AH) - Status of operation (see Read Status)

            Disk status at 40:74 = status of operation
```

(AH) = 05H Format Desired Track

The 2 high-order bits of the track number (10 bits total) are the 2 high-order bits of CL.

```
            (DL) - 7-bit drive number, bit 7 = 1
                   for fixed disk
            (DH) - Head number
            (CH) - Track number

            (ES:BX) - Address of buffer
                  points to a 512 byte buffer.  The first
                  2 bytes (sectors/track) contain F,N
                  for each sector.

                  F = 00 - for a good sector
                      80 - for a bad sector
                  N - sector number
```

For an interleave of 2 and 17 sectors per track, the table is:

```
      DB      00H,01H,00H,0AH,00H,02H,00H,0BH,00H,03H,00H,0CH
      DB      00H,04H,00H,0DH,00H,05H,00H,0EH,00H,06H,00H,0FH
      DB      00H,07H,00H,10H,00H,08H,00H,11H,00H,09H

      ON RETURN:
            (CY) - Set indicates status is not zero
            (AH) - Status of operation (see Read Status)

            Disk status at 40:74 = status of operation
```

(AH) = 08H Read Drive Parameters

If drive number is invalid, then AH and DISK _STATUS contain the value hex 07, and the carry flag is set.

If no fixed disks are attached, then AH and DISK _STATUS contain the value hex 01, and the carry flag is set. The number of drives attached, (DL), will never be returned as 0; therefore the value (DL) is either 01 or 02.

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk

ON RETURN:
        (DL) - Number of consecutive drives attached
               (1-2) (controller card zero tally only)
        (DH) - Maximum usable value for head number
               (origin of 0)
        (CH) - Maximum usable value for cylinder number
               (origin of 0)
        (CL) - Maximum usable value for sector number
               and cylinder number high bits (origin of 1)
```

(AH) = 09H Initialize Drive Pair Characteristics

Interrupt hex 41 points to the single parameter table for drive 0. If (DL) is hex 80, drive 0 is initialized using interrupt hex 41. For all other values, an invalid command status is returned.

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk

ON RETURN:
        (CY) - Set indicates status is not zero
        (AH) - Status of operation (see Read Status)
```

(AH) = 0AH and 0BH These functions are reserved for diagnostics.

(AH) = 0CH Seek

If an error is reported by the disk code, the appropriate action is to reset the disk, then retry the operation.

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk
        (DH) - Head number
        (CH) - Track number

ON RETURN:
        (CY) - Set indicates status is not zero
        (AH) - Status of operation (see Read Status)

        Disk status at 40:74 = status of operation
```

(AH) = 0DH Alternate Disk Reset

Disk Reset is invoked only if the drive number is less than or equal to the maximum number of fixed disks.

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk

ON RETURN:
        (CY) - Set indicates status is not zero
        (AH) - Status of operation (see Read Status)

        Disk status at 40:74 = status of operation
```

(AH) = 0EH and 0FH These functions are reserved for diagnostics.

(AH) = 10H Test Drive Ready

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk

ON RETURN:
        (CY) - Set indicates status is not zero
        (AH) - Status of operation (see Read Status)

        Disk status at 40:74 = status of operation
```

(AH) = 11H Recalibrate

If an error is reported by the disk code, reset the disk, then retry the operation.

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk

ON RETURN:
        (CY) - Set indicates status is not zero
        (AH) - Status of operation (see Read Status)

        Disk status at 40:74 = status of operation
```

(AH) = 14H Reserved for diagnostics.

(AH) = 15H Read DASD Type

If the drive number is out of range, AH contains 00 (Drive Not Present) and (CX, DX) = 00,00.

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk

ON RETURN:
        (AH) = 00 - Drive not present
             = 01 - Reserved for diskette interface
             = 02 - Reserved for diskette interface
             = 03 - Fixed disk
        (CX,DX) - Number of 512 byte blocks

        Disk status at 40:74 = status of operation
```

(AH) = 16H - 18H Reserved for diskette drive

(AH) = 19H Park Heads on Specified Drive

```
        (DL) - 7-bit drive number, bit 7 = 1
               for fixed disk

ON RETURN:
        (CY) - Set indicates error
        (AH) - Status of operation
             = 01 for invalid command

        Disk status at 40:74 = status of operation
```

(AH) = 1AH - FFH Reserved

```
ON RETURN:
      (CY) - Set indicates error
      (AH) = 01 for invalid command

      Disk status at 40:74 = status of operation
```

Interrupt 14H - Asynchronous Communications

These routines provide RS232 support.

(AH) = 00H Initialize the Communications Port

(AL) - Parms for initialization
(DX) - RS232 card number (0 based)

7 6 5 Baud Rate	4 3 Parity	2 Stopbit	1 0 Word Length
000 - 110	X0 - None	0 - 1	10 - 7 Bits
001 - 150	01 - Odd	1 - 2	11 - 8 Bits
010 - 300	11 - Even		
011 - 600			
100 - 1200			
101 - 2400			
110 - 4800			
111 - 9600			

ON RETURN:

(AL) - Modem status
- Bit 7 = Receved 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

(AH) - Line control status
- Bit 7 = Time out
- Bit 6 = Tx shift register empty
- Bit 5 = Tx holding register empty
- Bit 4 = Break detect
- Bit 3 = Framing error
- Bit 2 = Parity error
- Bit 1 = Overrun error
- Bit 0 = Data ready

(AH) = 01H Send Character

```
        (AL) - Character to send
        (DX) - RS232 card number (0 based)

ON RETURN:
        (AL) is preserved
        (AH) - Status
               Bit 7 = 1 unable to transmit
                     If bit 7 = 0 (able to transmit),
                     then bits 6 thru 0 are:
               Bit 6 = Tx shift register empty
               Bit 5 = Tx holding register empty
               Bit 4 = Break detect
               Bit 3 = Framing error
               Bit 2 = Parity error
               Bit 1 = Overrun error
               Bit 0 = Data ready
```

(AH) = 02H Receive Character

The routine waits for the character. If bit 7 of status is set, the other bits are unpredictable.

```
        (DX) - RS232 card number (0 based)

ON RETURN:
        (AL) - Character received
        (AH) - Line status
               Bit 7 = Timeout
               Bit 4 = Break detect
               Bit 3 = Framing error
               Bit 2 = Parity error
               Bit 1 = Overrun error
```

(AH) = 03H Read Status

```
        (DX) - RS232 card number (0 based)

ON RETURN:
      (AL) - Modem status register
             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

      (AH) - Line status register
             Bit 7 = Time out
             Bit 6 = Tx shift register empty
             Bit 5 = Tx holding register empty
             Bit 4 = Break detect
             Bit 3 = Framing error
             Bit 2 = Parity error
             Bit 1 = Overrun error
             Bit 0 = Data ready

      (DX) - RS232 card number (0 based)
```

(AH) = 04H Extended Initialize

```
        (DX) - RS232 card number (0 based)
        (AL) - Break
               00 - No break
               01 - Break

        (BH) - Parity
               00 - None
               01 - Odd
               02 - Even
               03 - Stick parity odd
               04 - Stick parity even

        (BL) - Stop bit
               00 - One
               01 - Two  if 6-, 7-, or 8-bit word length
                    One and a half if 5-bit word length

        (CH) - Word length
               00 - 5 bits
               01 - 6 bits
               02 - 7 bits
               03 - 8 bits

        (CL) - Baud rate
               00 -   110  Baud
               01 -   150  Baud
               02 -   300  Baud
               03 -   600  Baud
               04 -  1200  Baud
               05 -  2400  Baud
               06 -  4800  Baud
               07 -  9600  Baud
               08 - 19200  Baud

ON RETURN:
        (AL) - Modem status register, see (AH)=03
        (AH) - Line status register, see (AH)=03
```

(AH) = 05H Extended Communications Port Control

(AL) = 00H Read Modem Control Register

```
ON RETURN:
      (AL) - Modem status register, see (AH)=03
      (AH) - Line status register, see (AH)=03
      (BL) - Modem control register
             Bits 7-5 Reserved = 0

             Bit 4 = Loop
             Bit 3 = Out 2
             Bit 2 = Out 1
             Bit 1 = Request to Send
             Bit 0 = Data Terminal Ready
```

(AL) = 01H Write Modem Control Register

```
      (BL) - Modem control register
             Bits 7-5 Reserved = 0
             Bit 4 = Loop
             Bit 3 = Out 2
             Bit 2 = Out 1
             Bit 1 = Request to Send
             Bit 0 = Data Terminal Ready

ON RETURN:
      (AL) - Modem status register, see (AH)=03
      (AH) - Line status register, see (AH)=03
      (BL) - Modem control register
```

(AH) = 06F - FFH Reserved

Interrupt 15H - System Services

(AH) = 00 - 4EH Reserved

```
ON RETURN:
      (CY) - Carry flag set
      (AH) = 86 invalid function
```

(AH) = 4FH Keyboard Intercept

Keyboard intercept (keyboard escape) is called asynchronously by the keyboard interrupt 09 routine. This allows for a keystroke to be changed or absorbed. Normally the system returns with the scan code unchanged, but the operating system can redirect an interrupt 15 to its own routine and do the following:

- Replace (AL) with a different scan code and return with the carry flag set, effectively changing the keystroke
- Process the keystroke and return with the carry flag clear causing the interrupt 09 routine to ignore the keystroke.

```
      (CY) - Set to change keystroke
      (AL) = Scan code

ON RETURN:
      (CY) - Carry flag set
      (AL) = Scan code
```

(AH) = 51H - 7FH Reserved

```
ON RETURN:
      (CY) - Carry flag set
      (AH) = 86H
```

(AH) = 80H Device Open

```
      (BX) = Device ID
      (CX) = Process ID
```

(AH) = 81H Device Close

```
      (BX) = Device ID
      (CX) = Process ID
```

(AH) = 82H Program Termination

(BX) = Device ID

(AH) = 83H Event Wait

(AL) = 00 Set interval
= 01 Cancel
(ES:BX) - Pointer to a byte in callers memory that will have the high-order bit set as soon as possible after the interval expires.
(CX,DX) - Number of microseconds to elapse before posting.

ON RETURN:
(CY) - Clear if (AL) not zero
- Set if function already busy

(AH) = 84H Joystick Support
(DX) = 00H Read Current Switch Settings

ON RETURN:
(CY) - Set if invalid call
(AL) = Switch settings (bits 7-4)

(DX) = 01H Read Resistive Inputs

ON RETURN:
(CY) - Set if invalid call
(AX) = A(x) value
(BX) = A(y) value
(CX) = B(x) value
(DX) = B(y) value

(AH) = 85H System Request Key Pressed

(AL) = 00 - Make of key
= 01 - Break of key

(AH) = 86H Wait

(CX,DX) - Number of microseconds to elapse before return to caller

(AH) = 87H - 8FH Reserved

ON RETURN:
(CY) - Carry flag set
(AH) = 86H

(AH) = 90H Device Busy

This function tells the operatintg system that the system is about to wait for device.

```
ON RETURN:
     (AL)   Type code (see AH = 91)
```

(AH) = 91H Interrupt Complete

This function is called to tell the operating system that an interrupt has occurred. The type codes for functions 90 and 91 are in the following catagories:

00 to 7F Indicates serially reusable devices. The operating system must serialize the access.

80 to BF Indicates reentrant devices; ES:BX is used to distinguish different calls (multiple I/O calls are allowed simultaneously).

C0 to FF Indicates wait-only calls; there are no complementary Posts for these Waits - they are timeout only. Times depend on the type of device.

```
(AL) - Type Code

 Type  Description          Timeout
 00  =  Disk                  Yes
 01  =  Diskette              Yes
 02  =  Keyboard               No
 80  =  Network                No
     ES:BX --> NCB
 FD  =  Diskette motor start  Yes
 FE  =  Printer               Yes
 FC  =  Fixed disk reset      Yes
```

(AH) = 92H - BFH Reserved

```
(CY) - Carry flag set
(AH) = 86H
```

(AH) = C0H Return System Configuration Parameters

```
ON RETURN:
        (ES:BX)= Pointer to system descriptor vector in ROM
           (CY) - Carry flag clear
           (AH) = 0
```

The following is the format of the system descriptor table.

Size	Description
Word	Length of Descriptor in Bytes, Minimum is 8 Bytes
Byte	Model Byte
Byte	Submodel Byte
Byte	BIOS Revision Level
Byte	Feature Information Byte 1 Bit 7 = 1 BIOS uses DMA channel 3 Bit 6 = 0 One interrupt controller Bit 5 = 1 Real-time clock present Bit 4 = 1 Keyboard escape sequence (INT 15) called in keyboard interrupt (INT 09) Bit 2 = 1 Extended BIOS data area is allocated

(AH) = C1H Return Extended BIOS Data Area Segment Address

```
ON RETURN:
        (CY) = Set on error
        (ES) = Segment to extended BIOS data area
```

(AH) = C2H Pointing Device

After POST, the following default parameters are set:

- Package size is set to 3 bytes.
- Pointing device is disabled.
- Sample rate is set to 100 reports per second.
- Resolution is set to 4 counts per mm.
- Scaling is set to 1:1.

When the device driver is called, the following information is on the stack (each entry is word length):

Entry	Description
1	Status (High Byte = 0) Low Byte Bit 7 1 = Y data overflow Bit 6 1 = X data overflow Bit 5 Y data, 1 = negative Bit 4 X data, 0 = positive Bits 3,2 Reserved Bit 1 1 = Right button pressed Bit 0 1 = Left button pressed
2	X Data (High Byte = 0) Low Byte - Bit 7 MSB, Bit 0 LSB
3	Y Data (High Byte = 0) Low Byte - Bit 7 MSB, Bit 0 LSB
4	Z Data (High Byte = 0) Low Byte = 0

The following are the return values for all functions of pointing device

```
ON RETURN:
        (CY) = Set if unsuccessful operation
        (AH) = Status
             00 - No error
             01 - Invalid function call
             02 - Invalid input
             03 - Error
             04 - Reserved
             05 - No Far Call installed
             06 - Reserved
```

(AL) = 00H Enable Pointing Device

(BH) = 0 Disable
= 1 Enable

(AL) = 01H Reset Pointing Device

(AL) = 02H Set Sample Rate

(BH) - Rate value
0 - 10 reports/sec
1 - 20 reports/sec
2 - 40 reports/sec
3 - 60 reports/sec
4 - 80 reports/sec
5 - 100 reports/sec
6 - 200 reports/sec

(AL) = 03H Set Resolution

(BH) - Resolution value
0 - 1 count /mm
1 - 2 counts/mm
2 - 4 counts/mm
3 - 8 counts/mm

(AL) = 04H Read Device Type

(BH) = Device ID

(AL) = 05H Initialization

(BH) - Data package size
1 - 1 Byte
2 - 2 Bytes
3 - 3 Bytes
4 - 4 Bytes
5 - 5 Bytes
6 - 6 Bytes
7 - 7 Bytes
8 - 8 Bytes

(AL) = 06H Extended Commands

(BH) = 00H Return Status

```
ON RETURN:
      (BL) - Status Byte 1
        Bit 7 = 0 - Reserved
        Bit 6 = 0 - Stream mode
              = 1 - Remote mode
        Bit 5 = 1 - Pointer enabled
        Bit 4 = 0 - 1:1 scaling
              = 1 - 2:1 scaling
        Bit 3 = 0 - Reserved
        Bit 2 = 1 - Left button pressed
        Bit 1 = 0 - Reserved
        Bit 0 = 1 - Right button pressed

      (CL) - Status Byte 2
        00 - 1  count/mm
        01 - 2 counts/mm
        02 - 4 counts/mm
        03 - 8 counts/mm

      (DL) - Status Byte 3
        0A -  10 reports/sec
        14 -  20 reports/sec
        2B -  40 reports/sec
        3C -  60 reports/sec
        50 -  80 reports/sec
        64 - 100 reports/sec
        C8 - 200 reports/sec
```

(BH) = 01H Set Scaling to 1:1
(BH) = 02H Set Scaling to 2:1

(AL) = 07H Device Driver Far Call

Setting the segment and offset to all 0's unsets the device driver.

```
(ES) = Segment pointer
(BX) = Offset pointer
```

Interrupt 16H - Keyboard

(AH) = 00H Keyboard Read

The ASCII characters and the scan code are extracted from the buffer (40:1E for a length of 32 bytes). The keyboard buffer pointer (word at 40:1A) is increased by 2 or reinitialized to the start of the buffer if the pointer is already at the end.

This function returns control only upon a keystroke being available; the keystroke is removed from buffer. If no keystroke is available, Device Busy (INT 15, AX = 9002H) is called to tell the operating system that a keyboard loop is about to take place, allowing the operating system to perform another task. Eventually, the keyboard interrupt (INT 09) calls Interrupt Complete (INT 15, AX = 9102H) to Post the operation complete.

```
ON RETURN:
      (AH) - Scan code
      (AL) - ASCII character
```

(AH) = 01H Keystroke Status

The keystoke is not removed from the buffer.

```
ON RETURN:
      (ZF) = Set if no code is available
           = Clear if code is available and
      (AL) - ASCII character
      (AH) - Scan code
```

(AH) = 02H Shift Status

The bits in AL are set for the following conditions.

```
ON RETURN:
        (AL) - Shift status
           Bit 7 - Insert locked
           Bit 6 - Caps locked
           Bit 5 - Nums locked
           Bit 4 - Scroll locked
           Bit 3 - Alt key pressed
           Bit 2 - Ctrl key pressed
           Bit 1 - Left shift key pressed
           Bit 0 - Right shift key pressed
```

(AH) = 03H Set Typematic Rate

(AL) = 05H Set Typamatic Rate and Delay

If the typematic rate or delay is not within the supported range, the function returns with no action taken.

```
(BH) - Delay value
(BL) - Typamatic rate
```

Value in BL	Char/Sec	Value in BL	Char/Sec	Value in BL	Char/Sec
00	30.0	0B	10.9	16	4.3
01	26.7	0C	10.0	17	4.0
02	24.0	0D	9.2	18	3.7
03	21.8	0E	8.6	19	3.3
04	20.0	0F	8.0	1A	3.0
05	18.5	10	7.5	1B	2.7
06	17.1	11	6.7	1C	2.5
07	16.0	12	6.0	1D	2.3
08	15.0	13	5.5	1E	2.1
09	13.3	14	5.0	1F	2.0
0A	12.0	15	4.6		

Value in BH	Delay Value
0	250 ms
1	500 ms
2	750 ms
3	1000 ms

(AH) = 05H Keyboard Write

This function places ASCII character scan code combination in keyboard buffer the same as if that key had been pressed.

```
(CL) - ASCII character
(CH) - Scan code

ON RETURN:
     (AL) = 00  Successful operation
          = 01  Buffer full
```

(AH) = 10H Extended Keyboard Read

The ASCII character and the scan code are extracted from the buffer (40:1E for a length of 32 bytes). The keyboard buffer pointer (word at 40:1A) is increased by 2 or reinitialized to the start of the buffer if the pointer is already at the end.

This function returns control only upon a keystroke being available; the keystroke is removed from buffer.

```
ON RETURN:
      (AL) - ASCII Character
      (AH) - Scan code
```

(AH) = 11H Extended Keystroke Status

This function does not remove the keystroke from the buffer.

```
ON RETURN:
      (ZF) - Set if no code is available
             Clear if code is available

If code is available:
      (AL) - ASCII character
      (AH) - Scan code
```

(AH) = 12H Extended Shift Status

The bits in AL and AH are set for the following conditions. Only AX and the flags are changed. All other registers are preserved.

```
ON RETURN:
      (AL) - Shift status
          Bit 7 - Insert locked
          Bit 6 - Caps locked
          Bit 5 - Nums locked
          Bit 4 - Scroll locked
          Bit 3 - Alt key pressed
          Bit 2 - Ctrl key pressed
          Bit 1 - Left shift key pressed
          Bit 0 - Right shift key pressed

      (AH) - Extended shift status
          Bit 7 - System request key pressed
          Bit 6 - Caps lock key pressed
          Bit 5 - Num lock key pressed
          Bit 4 - Scroll lock key pressed
          Bit 3 - Right Alt key pressed
          Bit 2 - Right Ctrl key pressed
          Bit 1 - Left Alt key pressed
          Bit 0 - Left Ctrl key pressed
```

Interrupt 17H - Printer

These routines provide printer support. When the printer is busy, BIOS calls Device Busy (INT 15, AX = 90FEH) telling the operating system a time out loop is about to begin.

(AH) = 00H Print Character

```
        (AL) - Character to print
        (DX) - Printer to be used (0,1,2) corresponding
               to actual values in PRINTER_BASE area

ON RETURN:
        (AH) - Status
             Bit 7  - Not busy
             Bit 6  - Acknowledge
             Bit 5  - Out of paper
             Bit 4  - Selected
             Bit 3  - I/O error
             Bit 2,1 - Unused
             Bit 0  - Time out
```

(AH) = 01H Initialize the Printer Port

```
        (DX) - Printer to be used (0, 1, 2) corresponding to
               actual values in PRINTER_BASE area

ON RETURN:
        (AH) - Status - same as function 00
```

(AH) = 02H Read Status

```
        (DX) = Printer to be used (0,1,2) corresponding to
               actual values in PRINTER_BASE area

ON RETURN:
        (AH) - Status - same as function 00
```

(AH) = 03H - FFH Reserved

Interrupt 19H - Bootstrap Loader

Track 0, sector 1 is read into the boot location (segment 0 offset 7C00) and control is transferred there with the following values. If there is a hardware error, control is transferred to the ROM BASIC entry point.

```
(CS) = 00H
(IP) = 7C00H
(DL) = Drive that boot sector was read from
```

Interrupt 1AH - System and Real-Time Clock Services

(AH) = 00H Read System Time Counter

This function causes the timer overflow flag to be reset to 0. Timer counts occur at the rate of 1193180/65536 counts per second, or about 18.2 per second.

```
ON RETURN:
      (CX) = High portion of count
      (DX) = Low portion of count
      (AL) = 0 if timer has not passed 24 hours worth of counts
             since power-on, last system time counter read or write
           > 0 if timer has passed 24 hours worth of counts
             since power-on, last system time counter read or write
```

(AH) = 01H Set System Time Counter

This function causes timer overflow flag to be reset to 0. Timer counts occur at the rate of 1193180/65536 counts per second, or about 18.2 per second.

```
ON RETURN:
      (CX) - High portion of count
      (DX) - Low portion of count
```

(AH) = 02H Read Real-Time Clock Time

```
ON RETURN:
      (CY) - Clear if clock operating
             Set if not operating
      (CH) - Hours in BCD
      (CL) - Minutes in BCD
      (DH) - Seconds in BCD
```

(AH) = 03H Set Real-time Clock Time

```
      (CH) - Hours in BCD
      (CL) - Minutes in BCD
      (DH) - Seconds in BCD
```

(AH) = 04H Read Real-Time Clock Date

```
ON RETURN:
      (CY) = Clear if clock operating
             Set if not operating
      (CH) - Century in BCD (19 or 20)
      (CL) - Year in BCD
      (DH) - Month in BCD
      (DL) - Day in BCD
```

(AH) = 05H Set Real-Time Clock Date

```
(CH) - Century in BCD (19 or 20)
(CL) - Year in BCD
(DH) - Month in BCD
(DL) - Day in BCD
```

(AH) = 06H Set Real-Time Clock Alarm

The alarm interrupts at the specified hours, minutes, and seconds passed in CH, CL, and DH respectively. One alarm function can be active at any time and interrupts every 24 hours at the specified time until the alarm is reset. When the alarm interrupts, software interrupt 4A is invoked.

The user must code a routine and place the correct address in the vector table.

```
(CH) - Hours in BCD
(CL) - Minutes in BCD
(DH) - Seconds in BCD

ON RETURN:
     (CY) - Set if alarm is already set or
            clock not operating
```

(AH) = 07H Reset Real-Time Clock Alarm

This function stops the alarm interrupt from occurring.

(AH) = 09H Read Real-Time Clock Alarm Time and Status

```
ON RETURN:
     (CH) - Hours in   BCD
     (CL) - Minutes in BCD
     (DH) - Seconds in BCD
     (DL) - Alarm status
            00 - alarm not enabled (AIE=0)
            01 - alarm enabled but will not power
                 on system (AIE=1, EN_PON_ALRM=0)
            02 - alarm enabled and will power on
                 system (AIE=1, EN_PON_ALRM=1)
```

(AH) = 0AH Read System Day Counter

```
ON RETURN:
     (CX) = Count of days since 1-1-1980
```

(AH) = 0BH Set System Day Counter

(CX) = Count of days since 1-1-1980

(AH) = 0CH - FFH Reserved

ON RETURN:
(CY) - Set for invalid function request

BIOS Data Area and Locations

The IBM BIOS routines use 256 bytes of memory from absolute address hex 400 to 4FF.

Address	Function
40:0	COM1 Port Address (Word)
40:2	COM2 Port Address (Word)
40:4	COM3 Port Address (Word)
40:6	COM4 Port Address (Word)
40:8	LPT1 Port Address (Word)
40:A	LPT2 Port Address (Word)
40:C	LPT3 Port Address (Word)
40:E	Extended BIOS Data Area Segment (Word)
40:10	Equipment Word (Word) 15,14 Number of Printers Attached 13,12 Reserved 11-9 Number of RS232 Cards Attached 8 Reserved 7,6 Number of Diskette Drives 5,4 Initial Video Mode 00 = Unused 01 = 40x25 Color 10 = 80x25 Color 11 = 80x25 Monochrome 3 Reserved 2 Mouse Present 1 Coprocessor Installed 0 IPL Diskette Installed
40:12	Reserved
40:13	Memory Size in K Bytes (Word)
40:15	Reserved
40:16	BIOS Control Flags
40:17	Keyboard Flags (Byte) Alt and Ctrl bits are set if either Alt and Ctrl keys are pressed. 7 Insert Locked 6 Caps Locked 5 Nums Locked 4 Scroll Locked 3 Alt Key Depressed 2 Ctrl Key Depressed 1 Left Shift Key Depressed 0 Right Shift Key Depressed

Address	Function
40:18	Keyboard Flags 1 (Byte) 7 Insert Key Pressed 6 Caps Lock Key Pressed 5 Num Lock Key Pressed 4 Scroll Lock Key Pressed 3 Pause Locked 2 System Key Pressed 1 Left Alt Key Pressed 0 Left Ctrl Key Pressed
40:19	Storage For Alternate Keypad Entry (Byte)
40:1A	Pointer To Buffer Head Within Data Segment 40 (Word)
40:1C	Pointer To Buffer Tail Within Data Segment 40 (Word)
40:1E	Keyboard Buffer (32 Bytes)
40:3E	Drive Recalibration Status (Byte) 7 Working Interrupt Flag Always 0 on Return from Diskette BIOS 3 Recalibrate Drive 3 2 Recalibrate Drive 2 1 Recalibrate Drive 1 0 Recalibrate Drive 0
40:3F	Motor Status (Byte) 7 Write Operation Otherwise Read 3 Drive 3 Motor On 2 Drive 2 Motor On 1 Drive 1 Motor On 0 Drive 0 Motor On
40:40	Motor Off Counter (Byte), Decremented by Timer. When 0, All Drive Motors Turned Off
40:41	Status of Last Diskette Operation (Byte) 80 - Time Out 40 - Seek Failure 20 - General Controller Failure 10 - Bad CRC Error 0C - Unsupported Track, Sectors/Track Combination 09 - DMA Boundary Error 08 - DMA Failure 06 - Media Has Been Changed 04 - Sector Not Found 03 - Write Protect Error 02 - Bad Address Mark 01 - Invalid Function Request 00 - No Error
40:42	Status Returned from Controller (7 Bytes)
40:49	Current CRT Mode (Byte) See Interrupt 10H
40:4A	Number of Columns on Screen (Word)
40:4C	Regen Buffer Length in Bytes (Word)
40:4E	Starting Address Offset of Regen Buffer (Word)
40:50	Cursor Position Page 1 (Word)
40:52	Cursor Position Page 2 (Word)
40:54	Cursor Position Page 3 (Word)

Address	Function
40:56	Cursor Position Page 4 (Word)
40:58	Cursor Position Page 5 (Word)
40:5A	Cursor Position Page 6 (Word)
40:5C	Cursor Position Page 7 (Word)
40:5E	Cursor Position Page 8 (Word)
40:60	Cursor Mode (Word)
40:60	End Line for Cursor
40:61	Start Line for Cursor
40:62	Current Page being Displayed (Byte)
40:63	Base Port Address for Active Display (Word)
40:65	Current Setting of the 3x8 Register (Byte) Mirror Image Written to Base Port Address + 4 for Set Mode
40:66	Current Pallete Setting Color Card (Byte) Mirror Image Written to Base Port Address + 5
40:67 - 6B	Reserved
40:6C	Timer Counter Low Word,High Word (DWord) Increased Approximately 18 Times per Second
40:70	Timer Overflow (Byte) Not 0 = Timer Counted Past 24 Hours 0 = NOT
40:71	BIOS Break Flag (Byte) Bit 7 - Set if Break Key Pressed
40:72	Reset Flag (Word), If Hex 1234, Then No Need to Test Memory on POST
40:74	Status of Last Disk Operation (Byte) FF - Sense Operation Failed E0 - Status Error/Error Reg = 0 CC - Write Fault on Selected Drive BB - Undefined Error Occurred AA - Drive Not Ready 80 - Time Out 40 - Seek Failure 20 - General Controller Failure 11 - ECC Corrected Data Error 10 - Bad ECC on Disk Read 0E - Controlled Data Address Mark Detected 0D - Invalid Number of Sectors on Format 0A - Bad Sector Flag Detected 0B - Bad Track Detected 09 - DMA Boundary Error 08 - DMA Failure 07 - Drive Parameter Activity Failed 05 - Reset Failed 04 - Sector Not Found 03 - Write Protect Error 02 - Bad Address Mark 01 - Invalid Function Request 00 - No Error

Address	Function
40:75	Number Of Fixed Disks Attached To System (Byte)
40:76	Reserved
40:77	Reserved
40:78	LPT1 Timeout Value (Byte)
40:79	LPT2 Timeout Value (Byte)
40:7A	LPT3 Timeout Value (Byte)
40:7C	COM1 Timeout Value (Byte)
40:7D	COM2 Timeout Value (Byte)
40:7E	COM3 Timeout Value (Byte)
40:7F	COM4 Timeout Value (Byte)
40:80	Start of Keyboard Buffer within Data Segment 40 (Word)
40:82	End of Keyboard Buffer within Data Segment 40 (Word)
40:84	Rows on the Screen (Byte)
40:85	Bytes per Character (Word)
40:87	Mode Options (Byte) = 00
40:88	Reserved
40:89	7-5 Reserved 4 1 - 8x16 Text Font 0 - 8x8 Text Font 3 0 - Default Palette Loading Enabled 2 0 - Color Monitor Attached 1 - Monochrome Attached 1 Video Summing Enabled 0 Reserved
40:8B	Last Diskette (Byte) Bits 7,6 Data Rate Selected 00 = 500K bps 01 = 300K bps 10 = 250K bps 11 = Reserved Bits 5,4 Step Rate Time Selected 00 = for SRT = 0C 01 = for SRT = 0D 10 = for SRT = 0A 11 = Reserved
40:8C	Fixed Disk Status Returned by Controller (Byte)
40:8D	Fixed Disk Error Returned by Controller (Byte)
40:8E	Reserved = 00
40:8F	Reserved

Address	Function
40:90	Media State Drive 0 (Byte) See Below
40:91	Media State Drive 1 (Byte) See Below Bit Description For 40:90 & 40:91 7,6 Data rate 00 - 500K bps 01 - Reserved 10 - 250K bps 11 - Reserved 5 Reserved 4 0 - Media/Drive Unestablished 3 Reserved 2-0 Reserved = 111B
40:93	Reserved
40:94	Track Currently Seeked to, Drive 0 (Byte)
40:95	Track Currently Seeked to, Drive 1 (Byte)
40:96	Keyboard Type (Byte) 7 Read ID in Process 6 Last Char was First ID Char 5 Force Num Lock if Rd ID & KBX 4 101/102-key Keyboard Installed 3 Right Alt Key Depressed 2 Right Ctrl Key Depressed 1 Last Code was E0 Hidden Code 0 Last Code was E1 Hidden Code
40:97	LED Flags (Byte)
40:98	Pointer to Users Wait Flag (DWord)
40:9C	User Timeout Value Low Word, High Word (DWord) in Microseconds
40:A0	RTC Wait Function in Use Flag (Byte) Bit 7 - RTC Periodic Time Elapsed Bit 0 - Function in Use Otherwise Not
40:A1-A3	Reserved
40:A4-A7	Saved Fixed Disk Interrupt Vector
40:A8-AB	Pointer to Alternate Parameter Table (Video)
40:AC-CD	Reserved
40:CE	Day Counter (Word)
40:CF-EF	Reserved
40:F0-FF	Reserved for User
50:00	Print Screen Status Byte

Extended BIOS Data Area

Power-on-self-test (POST) carves out the highest possible 1K of memory below 640K to be used as the extended data area. The word pointer at 40:0E in the BIOS data area, points to the segment. The first byte in the extended BIOS data area is initialized to the length, in K bytes, allocated. The allocation of the data area within the carved segment is:

Offset (Hex)	Function
00	Number of bytes allocated in multiples of K (Byte)
01-21	Reserved
22-2F	Pointing device interface BIOS data area (14 Bytes)
	22 Device Driver Far Call Offset (Word)
	24 Device Driver Far Call Segment (Word)
	26 Pointing Device Flag (1st Byte)
	7 Command in Progress
	6 Resend
	5 Acknowledge
	4 Error
	3 Reserved = 0
	2-0 Index Count
	27 Pointing Device Flag (2nd Byte)
	7 Device Driver Far Call flag
	6-3 Reserved
	2-0 Package Size
	28 - 2F Reserved

ROM Tables

The following tables are located in ROM.

Fixed Disk Parameter Table

The following shows the table format and the table entries for the fixed disk.

Offset (Hex)	Size	Function
0	Word	Maximum number of cylinders
2	Byte	Maximum number of heads
3	Word	Reserved
5	Word	Starting write precompensation cyl
7	Byte	Not used
8	Byte	Control byte
9	Byte	Reserved
A	Byte	Reserved
B	Byte	Reserved
C	Word	Landing zone
E	Byte	Number of sectors/track,
F	Byte	Reserved

Type	Cyls	Heads	Precomp at Cyl	Landing Zone
0	Indicates No Fixed Disk Installed			
1	306	4	128	305
2	615	4	300	615
3	615	6	300	615
4	940	8	512	940
5	940	6	512	940
6	615	4	None	615
7	462	8	256	511
8	733	5	None	733
9	900	15	None	901
10	820	3	None	820
11	855	5	None	855
12	855	7	None	855
13	306	8	128	319
14	733	7	None	733
15	Indicates Parameters In Expanded Table			
16	612	4	0	663
17	977	5	300	977
18	977	7	None	977
19	1024	7	512	1023
20	733	5	300	732
21	733	7	300	732
22	733	5	300	733
23	306	4	None	336
24	612	4	305	663
25	306	4	None	340
26*	612	4	None	670
27 - 255	Reserved			

* Type for IBM Personal System/2 20MB Fixed Disk Drive and Controller.

Asynchronous Baud Rate Initialization Table

Offset (Hex)	Size	Function
0	Word	Init value for 110 Baud
2	Word	Init value for 150 Baud
4	Word	Init value for 300 Baud
6	Word	Init value for 600 Baud
8	Word	Init value for 1200 Baud
A	Word	Init value for 2400 Baud
C	Word	Init value for 4800 Baud
E	Word	Init value for 9600 Baud

Diskette Parameter Table

Offset (Hex)	Size	Function
0	Byte	First specify byte
1	Byte	Second specify byte
2	Byte	Number of timer ticks to wait prior to turning diskette motor off
3	Byte	Number of bytes/sector = 0 128 bytes/sector = 1 256 bytes/sector = 2 512 bytes/sector = 3 1024 bytes/sector
4	Byte	Sectors/track
5	Byte	Gap length
6	Byte	Data length
7	Byte	Gap length for format
8	Byte	Fill byte for format
9	Byte	Head settle time in ms
A	Byte	Motor startup time in 1/8 seconds

Model Byte

The model byte is located at F000:FFFE in ROM. Use the read system configuration parameters (INT 15, AH = C0H) to find the model and sub-model byte. For the Model 30, the model byte is hex FA and the sub-model is 00.

SECTION 6. Instruction Set

8086 Register Model 6-2
Notes 6-3
8086 Instruction Set 6-7
Data Transfer 6-7
Arithmetic 6-9
Logic 6-13
String Manipulation 6-15
Control Transfer 6-15
Processor Control 6-20
Instruction Set Matrix 6-22
8087 Coprocessor Instruction Set 6-24
Notes 6-24
Data Transfer 6-24
Comparison 6-26
Arithmetic 6-26
Transcendental 6-28
Constants 6-29
Processor Control 6-29

8086 Register Model

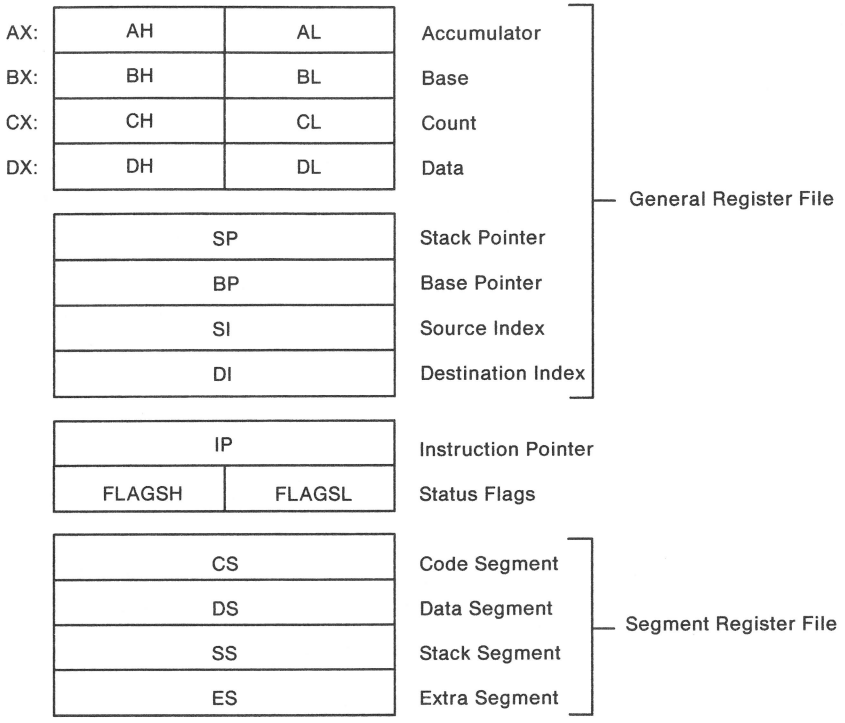


Figure 6-1. 8086 Register Model

Flag Register

Bit	Function
15 to 12	Don't Care
11	Overflow Flag
10	Direction Flag
9	Interrupt Enable Flag
8	Trap-Single Step Flag
7	Sign Flag
6	Zero Flag
5	Don't Care
4	Auxiliary Carry - BCD
3	Don't Care
2	Parity Flag
1	Don't Care
0	Carry Flag

Figure 6-2. Flag Register

Notes

If d = 1 then "to"; if d = 0 then "from"
If w = 1 then word size; if w = 0 then byte size
If sw = 01 then 16 bits of immediate data from the operand
If sw = 11 then an immediate data byte is signed extended to form the 16-bit operand
If v = 0 the "count" = 1; if v = 1 the "count" is in (CL) or (CX)
x = don't care
z is used for string primitives for comparison with zero flag

Segment Override Prefix

001reg110

Operand Register	Default	With Prefix
IP (Code Address)	CS	Never
SP (Stack Address)	CS	Never
BP (Stack Address or Stack Marker)	SS	BP + DS or ES, or CS
SI or DI (not including strings)	DS	ES, SS, or CS
SI (Implicit Source Address for strings)	DS	ES, SS, OR CS
DI (Implicit Destination Address for strings)	ES	Never

Figure 6-3. Segment Override Prefix

reg Field Assignments

16-Bit	8-Bit	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Figure 6-4. reg Field Assignment

Second Instruction Byte

mod	xxx	r/m

mod	Displacement
00	DISP = 0*, disp-low and disp-high are absent
01	DISP = disp-low sign-extended to 16-bits, disp-high is absent
10	DISP = disp-high: disp-low
11	DISP = r/m is treated as a "reg" field

DISP follows 2nd byte of instruction (if required)
* If mod = 00 and r/m = 110, then Effective Address = disp-high:disp-low.

Figure 6-5. mod Field Assignment

r/m	Operand Address
000	(BX) + (SI) + DISP
001	(BX) + (DI) + DISP
010	(BP) + (SI) + DISP
011	(BP) + (DI) + DISP
100	(SI) + DISP
101	(DI) + DISP
110	(BP) + DISP
100	(BX) + DISP

Figure 6-6. r/m Field Assignments

Notes:

8086 Instruction Set

Data Transfer

MOV = Move

Register/Memory to/from Register

100010dw	mod reg r/m

Immediate to Register/Memory

1100011w	mod 000 r/m	data	data if w = 1

Immediate to Register

1011wreg	data	data if w = 1

Memory to Accumulator

1010000w	addr-low	addr-high

Accumulator to Memory

1010001w	addr-low	addr-high

Register/Memory to Segment Register

10001110	mod 0 reg r/m

Segment Register to Register/Memory

10001100	mod 0 reg r/m

PUSH = Push

Register/Memory

11111111	mod 110 r/m

Register

01010reg

Segment Register

000reg110

POP = Pop

Register/Memory

10001111	mod 000 r/m

Register

01011reg

Segment Register

000reg111

XCHG = Exchange

Register/Memory with Register

1000011w	mod reg r/m

Register with Accumulator

10010reg

IN = Input to AL/AX from

Fixed Port

1110010w	port

Variable Port

1110110w

OUT = Output from AL/AX to

Fixed Port

1110011w	port

Variable Port (DX)

1110110w

XLAT = Translate Byte to AL

11010111

LEA = Load EA to Register

10001101	mod reg r/m

LDS = Load Pointer to DS

11000101	mod reg r/m

LES = Load Pointer to ES

11000100	mod reg r/m

LAHF = Load AH with Flags

10011111

SAHF = Store AH with Flags

10011110

PUSHF = Push Flags

10011100

POPF = Pop Flags

10011101

Arithmetic

ADD = Add

Register/Memory with Register to Either

000000dw	mod reg r/m

Immediate to Register/Memory

100000sw	mod 000 r/m	data	data if sw = 01

Immediate to Accumulator

0000010w	data	data if w = 1

ADC = Add with Carry

Register/Memory with Register to Either

000100dw	mod reg r/m

Immediate to Register/Memory

100000sw	mod 010 r/m	data	data if sw = 01

Immediate to Accumulator

0001010w	data	data if w = 1

INC = Increment

Register/Memory

1111111w	mod 000 r/m

Register

01000reg

AAA = ASCII Adjust for Add

00110111

DAA = Decimal Adjust for Add

00100111

SUB = Subtract

Register/Memory and Register to Either

001010dw	mod reg r/m

Immediate from Register/Memory

100000sw	mod 010 r/m	data	data if sw = 01

Immediate from Accumulator

0010110w	data	data if w = 1

SBB = Subtract with Borrow

Register/Memory and Register to Either

000110dw	mod reg r/m

Immediate from Register/Memory

100000sw	mod 011 r/m	data	data if sw = 01

Immediate to Accumulator

0001110w	data	data if w = 1

DEC = Decrement

Register/Memory

1111111w	mod 001 r/m

Register

01001reg

NEG = Change Sign

1111011w	mod 011 r/m

CMP = Compare

Register/Memory and Register

001110dw	mod reg r/m

Immediate with Register/Memory

100000sw	mod 111 r/m	data	data if sw = 01

Immediate with Accumulator

0011110w	data	data if w = 1

AAS = ASCII Adjust for Subtract

00111111

DAS = Decimal Adjust for Subtract

00101111

MUL = Multiply (Unsigned)

1111011w	mod 100 r/m

IMUL = Integer Multiply (Signed)

1111011w	mod 101 r/m

AAM = ASCII Adjust for Multiply

11010100	00001010

DIV = Divide (Unsigned)

1111011w	mod 110 r/m

IDIV = Integer Divide (Signed)

1111011w	mod 111 r/m

AAD = ASCII Adjust for Divide

11010101	00001010

CBW = Convert Byte to Word

10011000

CWD = Convert Word to Double Word

10011001

Logic

NOT = Invert Register/Memory

1111011w	mod 010 r/m

SHL/SAL = Shift Logical/Arithmetic Left

110100vw	mod 100 r/m

SHR = Shift Logical Right

110100vw	mod 101 r/m

SAR = Shift Arithmetic Right

110100vw	mod 111 r/m

ROL = Rotate Left

110100vw	mod 000 r/m

ROR = Rotate Right

110100vw	mod 001 r/m

RCL = Rotate through Carry Left

110100vw	mod 010 r/m

RCR = Rotate through Carry Right

110100vw	mod 011 r/m

AND = And

Register/Memory and Register to Either

001000dw	mod reg r/m

Immediate to Register/Memory

1000000w	mod 100 r/m	data	data if w = 1

Immediate to Accumulator

0010010w	data	data if w = 1

TEST = AND Function to Flags; No Result

Register/Memory and Register

1000010w	mod reg r/m

Immediate Data and Register/Memory

1111011w	mod 000 r/m	data	data if w = 1

Immediate Data and Accumulator

1010100w	data	data if w = 1

OR = Or

Register/Memory and Register to Either

000010dw	mod reg r/m

Immediate to Register/Memory

1000000w	mod 001 r/m	data	data if w = 1

Immediate to Accumulator

0000110w	data	data if w = 1

XOR = Exclusive Or

Register/Memory and Register to Either

001100dw	mod reg r/m

Immediate to Register/Memory

1000000w	mod 110 r/m	data	data if w = 1

Immediate to Accumulator

0011010w	data	data if w = 1

String Manipulation

REP = Repeat

1111001z

MOVS = Move String

1010010w

CMPS = Compare String

1010011w

SCAS = Scan String

1010111w

LODS = Load String

1010110w

STOS = Store String

1010101w

Control Transfer

Call = Call

Direct within Segment

11101000	disp-low	disp-high

Indirect within Segment

11111111	mod 010 r/m

Direct Intersegment

10011010	offset-low	offset-high
	seg-low	seg-high

Indirect Intersegment

11111111	mod 011 r/m

JMP = Unconditional Jump

Direct within Segment-Short

11101011	disp

Indirect within Segment

11111111	mod 100 r/m

Direct Intersegment

11101010	offset-low	offset-high
	seg-low	seg-high

Indirect Intersegment

11111111	mod 101 r/m

RET = Return from Call

Within Segment

11000011

Within Segment Adding Immediate to SP

11000010	data-low	data-high

Intersegment

11000011

Intersegment Adding Immediate to SP

11000010	data-low	data-high

JE/JZ = Jump on Equal/Zero

01110100	disp

JI/JNGE = Jump on Less/Not Greater, or Equal

01111100	disp

JLE/JNG = Jump on Less, or Equal/Not Greater

01111110	disp

JB/JNAE = Jump on Below/Not Above, or Equal

01110010	disp

JBE/JNA = Jump on Below, or Equal/Not Above

01110110	disp

JP/JPE = Jump on Parity/Parity Even

01111010	disp

JO = Jump on Overflow

01110000	disp

JS = Jump on Sign

01111000	disp

JNE/JNZ = Jump on Not Equal/Not Zero

01110101	disp

JNL/JGE = Jump on Not Less/Greater, or Equal

01111101	disp

JNLE/JG = Jump on Not Less, or Equal/Greater

01111111	disp

JNB/JAE = Jump on Not Below/Above, or Equal

01110011	disp

JNBE/JA = Jump on Not Below, or Equal/Above

01110111	disp

JNP/JPO = Jump on Not Parity/Parity Odd

01111011	disp

JNO = Jump on Not Overflow

01110001	disp

JNS = Jump on Not Sign

01111001	disp

LOOP = Loop CX Times

11100010	disp

LOOPZ/LOOPE = Loop while Zero/Equal

11100001	disp

LOOPNZ/LOOPNE = Loop while Not Zero/Not Equal

11100000	disp

JCXZ = Jump on CX Zero

11100011	disp

Instruction	Condition	Interpretation
JE or JZ	ZF = 1	"equal" or "zero"
JL or JNGE	(SF xor OF) = 1	"less" or "not greater or equal"
JLE or JNG	((SF xor OF) or ZF) = 1	"less or equal" or "not greater"
JB or JNAE or JC	CF = 1	"below" or "not above or equal"
JBE or JNA	(CF or ZF) = 1	"below or equal" or "not above"
JP or JPE	PF = 1	"parity" or "parity even"
JO	OF = 1	"overflow"
JS	SF = 1	"sign"
JNE or JNZ	ZF = 0	"not equal" or "not zero"
JNL or JGE	(SF xor OF) = 0	"not less" or "greater or equal"
JNLE or JG	((SF xor OF) or ZF) = 0	"not less or equal" or "greater"
JNB or JAE or JNC	CF = 0	"not below" or "above or equal"
JNBE or JA	(CF or ZF) = 0	"not below or equal" or "above"
JNP or JPO	PF = 0	"not parity" or "parity odd"
JNO	OF = 0	"not overflow"
JNS	SF = 0	"not sign"

"Above" and "below" refer to the relation between two unsigned values, while "greater" and "less" refer to the relation between two signed values.

Figure 6-7. Conditional Transfer Operations

INT = Interrupt

Type Specified

11001101	Type

Type 3

11001100

INTO = Interrupt on Overflow

11001110

IRET = Interrupt Return

11001111

Processor Control

CLC = Clear Carry

11111000

STC = Set Carry

11111001

CMC = Complement Carry

11110101

NOP = No Operation

10010000

CLD = Clear Direction

11111100

STD = Set Direction

11111101

CLI = Clear Interrupt

11111010

STI = Set Interrupt

11111011

HLT = Halt

11110100

WAIT = Wait

10011011

LOCK = Bus lock prefix

11110000

ESC = Escape (to 8087)

11011xxx	mod xxx r/m

Instruction Set Matrix

HI \ LO	0	1	2	3	4	5	6	7
0	ADD b,f,r/m	ADD w,f,r/m	ADD b,t,r/m	ADD w,t,r/m	ADD b,ia	ADD w,ia	PUSH ES	POP ES
1	ADC b,f,r/m	ADC w,f,r/m	ADC b,t,r/m	ADC w,t,r/m	ADC b,i	ADC w,i	PUSH SS	POP SS
2	AND b,f,r/m	AND w,f,r/m	AND b,t,r/m	AND w,t,r/m	AND b,i	AND w,i	DEG =ES	DAA
3	XOR b,f,r/m	XOR w,f,r/m	XOR b,t,r/m	XOR w,t,r/m	XOR b,i	XOR w,i	SEG =S+	AAA
4	INC AX	INC CX	INC DX	INC BX	INC SP	INC BP	INC SI	INC DI
5	PUSH AX	PUSH CX	PUSH DX	PUSH BX	PUSH SP	PUSH BP	PUSH SI	PUSH DI
6								
7	JO	JNO	JB/ JNAE	JNB/ JAE	JE/ JZ	JNE/ JNZ	JBE/ JNA	JNBE/ JA
8	Immed b,r/m	Immed w,r/m	Immed b,r/m	Immed is,r/m	TEST b,r/m	TEST w,r/m	XCHG b,r/m	XCHG w,r/m
9	NOP	XCHG CX	XCHG DX	XCHG BX	XCHG SP	XCHG BP	XCHG SI	XCHG DI
A	MOV m AL	MOV m AL	MOV AL m	MOV AL m	MOVS b	MOVS w	CMPS b	CMPS w
B	MOV i AL	MOV i CL	MOV i DL	MOV i BL	MOV i AH	MOV i CH	MOV i DH	MOV i BH
C			RET (I+SP)	RET	LES	LDS	MOV b,i,r/m	MOV w,i,r/m
D	Shift b	Shift w	Shift b,v	Shift w,v	AAM	AAD		XLAT
E	LOOPNZ/ LOOPNE	LOOPZ/ POOPE	LOOP	JCXZ	IN b	IN w	OUT b	OUT w
F	LOCK		REP	REP z	HLT	CMC	Grp 1 b,r/m	Grp 1 w,r/m

Code	Definition	Code	Definition
b	Byte	m	Memory
d	Direct	r/m	EA is Second Byte
i	Immediate	si	Short, Intrasegment
ia	Immed. to Accum.	t	To CPU Register
id	Indirect	v	Variable
is	Immed. Byte, Sign Ext.	w	Word
l	Long, Intersegment	z	Zero

LO / HI	8	9	A	B	C	D	E	F
0	OR b,f,r/m	w,f,r/m	OR b,t,r/m	OR w,t,r/m	OR b,i	OR w,i	PUSH CS	
1	SBB b,f,r/m	SBB w,f,r/m	SBB b,t,r/m	SBB w,t,r/m	SBB b,i	SBB w,i	PUSH DS	POP DS
2	SUB b,f,r/m	SUB w,f,r/m	SUB b,t,r/m	SUB w,t,r/m	SUB b,i	SUB w,i	SEG= CS	DAS
3	CMP b,f,r/m	CMP w,f,r/m	CMP b,t,r/m	CMP w,t,r/m	CMP b,i	CMP w,i	SEG= CS	AAS
4	DEC AX	DEC CX	DEC DX	DEC BX	DEC SP	DEC BP	DEC SI	DEC DI
5	POP AX	POP CX	POP DX	POP BX	POP SP	POP BP	POP SI	POP DI
6								
7	JS	JNS	JP/ JPE	JNP/ JPO	JL/ JNGE	JNL/ JGE	JLE/ JNG	JNLE/ JG
8	MOV b,f,r/m	MOV w,f,r/m	MOV b,t,r/m	MOV w,t,r/m	MOV sr,t,r/m	LEA	MOV sr,f,r/m	POP r/m
9	CBW	CWD CX	CALL l,d	WAIT BX	PUSHF SP	POPF BP	SAHF SI	LAHF DI
A	TEST b,i	TEST w,i	STOS b	STOS w	LODS b	LODS w	SCAS b	SCAS w
B	MOV i AX	MOV i CX	MOV i DX	MOV i BX	MOV i SP	MOV i BP	MOV i SI	MOV i DI
C			RET l,(I+SP)	RET l	INT Type 3	INT (Any)	INTO	IRET
D	ESC 0	ESC 1	ESC 2	ESC 3	ESC 4	ESC 5	ESC 6	ESC 7
E	CALL d	JMP d	JMP l,d	JMP si,d	IN v,b	IN v,w	OUT v,b	OUT v,w
F	CLC	STC	CLI	STI	CLD	STD	GRP 2 b,r/m	GRP 3 w,r/m

Where: mod xxx r/m

xxx	000	001	010	011	100	101	110	111
Immed	ADD	OR	ADC	SBB	AND	SUB	XOR	CMP
Shift	ROL	ROR	RCL	RCR	SHL/SAL	SHR	--	SAR
Grp 1	TEST	--	NOT	NEG	MUL	IMUL	DIV	DIV
Grp 2	INC	DEC	CALL id	CALL l,id	JMP id	JMP l,id	PUSH	--

Notes

MF = Memory format

00 - 32-bit Real
01 - 32-bit Integer
10 - 64-bit Real
11 - 64-bit Integer

ST(0) = Current Stack top
ST(i) = ith register below Stack top
d = Destination
0—Destination is ST(0)
1—Destination is ST(i)

P = POP
0—No Pop
1—Pop ST(0)

R = Reverse

0—Destination (op) Source
1—Source (op) Destination

For **FSQRT:** $-0 \leq ST(0) \leq +\infty$
For **FSCALE:** $-2^{15} \leq ST(1) < +2^{15}$ and ST(1) interger
For **F2XM1:** $0 \leq ST(0) \leq 2-1$
For **FYL2X:** $0 < St(0) < \infty - \infty < ST(1) < +\infty$
For **FYL2XP1:** $0 < |ST(0)| < (2-\sqrt{2})/2 - \infty < ST(1) < \infty$
For **FPTAN:** $0 \leq ST(0) < \pi/4$
For **FPATAN:** $0 \leq ST(0) < ST(1) < +\infty$

The following is an instruction set summary for the 8087 coprocessor. In the following, the bit pattern for escape is 11011.

Data Transfer

FLD = Load

Interger/Real Memory to ST(0)

escape MF 1	mod 000 r/m	disp-low	disp-high

Long Integer Memory to ST(0)

escape 111	mod 101 r/m	disp-low	disp-high

Temporary Real Memory to ST(0)

escape 011	mod 101 r/m	disp-low	disp-high

BCD Memory to ST(0)

escape 111	mod 100 r/m	disp-low	disp-high

ST (i) to ST(0)

escape 001	11 000ST(i)

FST = Store

ST(0) to Integer/Real Memory

escape MF 1	mod 010 r/m	disp-low	disp-high

ST(0) to ST(i)

escape 101	11 010 ST(i)

FSTP = Store and Pop

ST(0) to Integer/Real Memory

escape MF 1	mod 011 r/m	disp-low	disp-high

ST(0) to Long Integer Memory

escape 111	mod 111 r/m	disp-low	disp-high

ST(0) to Temporary Real Memory

escape 011	mod 111 r/m	disp-low	disp-high

ST(0) to BCD Memory

escape 111	mod 110 r/m	disp-low	disp-high

ST(0) to (ST(i)

escape 101	11 011 ST(i)

FXCH = Exchange ST(i) and ST(0)

escape 001	11 001 ST(i)

Comparison

FCOM = Compare

Integer/Real Memory to ST(0)

escape MF 0	mod 010 r/m	disp-low	disp-high

ST(i) to ST(0)

escape 000	11 010 ST(i)

FCOMP = Compare and Pop

Integer/Real Memory to ST(0)

escape MF 0	mod 011 r/m	disp-low	disp-high

ST(1) to ST(0)

escape 000	11 011 ST(i)

FCOMPP = Compare ST(i) to ST(0) and Pop Twice

escape 110	11 011 001

FTST = Test ST(0)

escape 001	11 100 100

FXAM = Examine ST(0)

escape 001	11 100 101

Arithmetic

FADD = Addition

Integer/Real Memory with ST(0)

escape MF 0	mod 000 r/m	disp-low	disp-high

ST(i) to ST(0)

escape dP 0	11 000 ST(i)

FSUB = Subtraction

Integer/Real Memory with ST(0)

escape MF 0	mod 10R r/m	disp-low	disp-high

ST(i) to ST(0)

escape dP 0	11 10R r/m

FMUL = Multiplication

Integer/Real Memory with ST(0)

escape MF 0	mod 001 r/m	disp-low	disp-high

ST(i) to ST(0)

escape dP 0	11 001 r/m

FDIV = Division

Integer/Real Memory with ST(0)

escape MF 0	mod 11R r/m	disp-low	disp-high

ST(i) to ST(0)

escape dP 0	1111R r/m

ST(i) to ST(0)

escape dP 0	1111R r/m

FSQRT = Square Root of ST(0)

escape 001	11111010

FSCALE = Scale ST(0) of ST(1)

escape 001	11111101

FPREM = Partial Remainder of ST(0) ÷ ST(1)

escape 001	11111000

FRNDINT = Round ST(0) to Integer

escape 001	11111100

FXTRACT = Extract Components of ST(0)

escape 001	11110100

FABS = Absolute Value of ST(0)

escape 001	11100001

FCHS = Change Sign of ST(0)

escape 001	11100000

Transcendental

FPTAN = Partial Tangent of ST(0)

escape 001	11110010

FPATAN = Partial Archtangent of ST(0) ÷ ST(1)

escape 001	11110011

F2XM1 = $2^{ST(0)} - 1$

escape 001	11110000

FYL2X = ST(1) x Log_2 ffIST(0)"

escape 001	11111001

FYL2XP1 = ST(1) x Log_2 ffIST(0) + 1"

escape 001	11111001

Constants

FLDZ = Load + 0.0 into ST(0)

escape 001	11101110

FLD1 = Load + 1.0 into ST(0)

escape 001	11101000

FLDP1 = Load π into ST(0)

escape 001	11101011

FLDL2T = Load Log_2 10 into ST(0)

escape 001	11101001

FLDLG2 = Load Log_{10} 2 into ST(0)

escape 001	11101100

FLDLN2 = Load Log_e 2 into ST(0)

escape 001	11101101

Processor Control

FINIT = Initialize NDP

escape 011	11100011

FENI = Enable Interrupts

escape 011	11100000

FDISI = Disable Interrupts

escape 011	11100001

FLDCW = Load Control Word

escape 001	mod 101 r/m	disp-low	disp-high

FSTCW = Store Control Word

escape 001	mod 111 r/m	disp-low	disp-high

FSTSW = Store Status Word

escape 101	mod 111 r/m	disp-low	disp-high

FCLEX = Clear Exceptions

escape 011	11100010

FSTENV = Store Environment

escape 001	mod 110 r/m	disp-low	disp-high

FLDENV = Load Environment

escape 100	mod 100 r/m	disp-low	disp-high

FSAVE = Save State

escape 101	mod 110 r/m	disp-low	disp-high

FRSTOR = Restore State

escape 101	mod 100 r/m	disp-low	disp-high

FINCSTP = Increment Stack Pointer

escape 001	11110111

FDECSTP = Decrement Stack Pointer

escape 001	11110110

FFREE = Free ST(i)

escape 001	11000ST(i)

FNOP = No Operation

escape 001	11010000

FWAIT = CPU Wait for NDP

10011011

Notes:

SECTION 7. Characters and Keystrokes

Character Codes 7-2
Table Notes 7-7
Quick Reference 7-8

Character Codes

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
00	0	Blank (Null)	Ctrl 2	
01	1	☺	Ctrl A	
02	2	☻	Ctrl B	
03	3	♥	Ctrl C	
04	4	♦	Ctrl D	
05	5	♣	Ctrl E	
06	6	♠	Ctrl F	
07	7	•	Ctrl G	
08	8	◘	Ctrl H, Backspace, Shift Backspace	
09	9	○	Ctrl I	
0A	10	◙	Ctrl J, Ctrl ↵	
0B	11	♂	Ctrl K	
0C	12	♀	Ctrl L	
0D	13	♪	Ctrl M, ↵, Shift ↵	
0E	14	♫	Ctrl N	
0F	15	☼	Ctrl O	
10	16	►	Ctrl P	
11	17	◄	Ctrl Q	
12	18	↕	Ctrl R	
13	19	‼	Ctrl S	
14	20	¶	Ctrl T	
15	21	§	Ctrl U	
16	22	▬	Ctrl V	
17	23	↨	Ctrl W	

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
18	24	↑	Ctrl X	
19	25	↓	Ctrl Y	
1A	26	→	Ctrl Z	
1B	27	←	Ctrl [Esc, Shift Esc, Crtl Esc	
1C	28	∟	Ctrl \	
1D	29	↔	Ctrl]	
1E	30	▲	Ctrl 6	
1F	31	▼	Ctrl -	
20	32	Blank Space	Space Bar, Shift, Space, Ctrl Space, Alt Space	
21	33	!	!	Shift
22	34	"	"	Shift
23	35	#	#	Shift
24	36	$	$	Shift
25	37	%	%	Shift
26	38	&	&	Shift
27	39	'	'	Shift
28	40	(	(	Shift
29	41	)	)	
2A	42	*	*	Note 1
2B	43	+	+	Shift
2C	44	,	,	
2D	45	-	-	
2E	46	.	.	Note 2

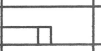
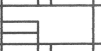
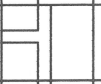
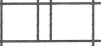
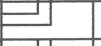
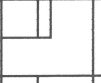
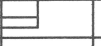
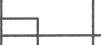
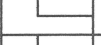
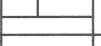
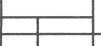
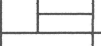
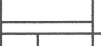
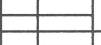
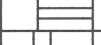
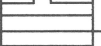
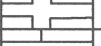
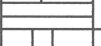
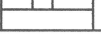
Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
2F	47	/	/	
30	48	0	0	Note 3
31	49	1	1	Note 3
32	50	2	2	Note 3
33	51	3	3	Note 3
34	52	4	4	Note 3
35	53	5	5	Note 3
36	54	6	6	Note 3
37	55	7	7	Note 3
38	56	8	8	Note 3
39	57	9	9	Note 3
3A	58	:	:	Shift
3B	59	;	;	
3C	60	<	<	Shift
3D	61	=	=	
3E	62	>	>	Shift
3F	63	?	?	Shift
40	64	@	@	Shift
41	65	A	A	Note 4
42	66	B	B	Note 4
43	67	C	C	Note 4
44	68	D	D	Note 4
45	69	E	E	Note 4
46	70	F	F	Note 4
47	71	G	G	Note 4
48	72	H	H	Note 4
49	73	I	I	Note 4
4A	74	J	J	Note 4

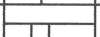
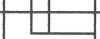
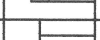
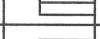
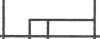
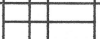
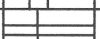
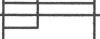
Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
4B	75	K	K	Note 4
4C	76	L	L	Note 4
4D	77	M	M	Note 4
4E	78	N	N	
4F	79	O	O	Note 4
50	80	P	P	Note 4
51	81	Q	Q	Note 4
52	82	R	R	Note 4
53	83	S	S	Note 4
54	84	T	T	Note 4
55	85	U	U	Note 4
56	86	V	V	Note 4
57	87	W	W	Note 4
58	88	X	X	Note 4
59	89	Y	Y	Note 4
5A	90	Z	Z	Note 4
5B	91	[	[	
5C	92	\	\	Note 4
5D	93	]	]	
5E	94	^	^	Shift
5F	95	_	_	Shift
60	96	`	`	
61	97	a	a	Note 5
62	98	b	b	Note 5
63	99	c	c	Note 5
64	100	d	d	Note 5
65	101	e	e	Note 5
66	102	f	f	Note 5

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
67	103	g	g	Note 5
68	104	h	h	Note 5
69	105	i	i	Note 5
6A	106	j	j	Note 5
6B	107	k	k	Note 5
6C	108	l	l	Note 5
6D	109	m	m	Note 5
6E	110	n	n	Note 5
6F	111	o	o	Note 5
70	112	p	p	Note 5
71	113	q	q	Note 5
72	114	r	r	Note 5
73	115	s	s	Note 5
74	116	t	t	Note 5
75	117	u	u	Note 5
76	118	v	v	Note 5
77	119	w	w	Note 5
78	120	x	x	Note 5
79	121	y	y	Note 5
7A	122	z	z	Note 5
7B	123	{	{	Shift
7C	124	¦	¦	Shift
7D	125	}	}	Shift
7E	126	~	~	Shift
7F	127	△	Ctrl-	

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
80	128	Ç	Alt 128	Note 6
81	129	ü	Alt 129	Note 6
82	130	é	Alt 130	Note 6
83	131	â	Alt 131	Note 6
84	132	ä	Alt 132	Note 6
85	133	à	Alt 133	Note 6
86	134	å	Alt 134	Note 6
87	135	ç	Alt 135	Note 6
88	136	ê	Alt 136	Note 6
89	137	ë	Alt 137	Note 6
8A	138	è	Alt 138	Note 6
8B	139	ï	Alt 139	Note 6
8C	140	î	Alt 140	Note 6
8D	141	ì	Alt 141	Note 6
8E	142	Ä	Alt 142	Note 6
8F	143	Å	Alt 143	Note 6
90	144	É	Alt 144	Note 6
91	145	æ	Alt 145	Note 6
92	146	Æ	Alt 146	Note 6
93	147	ô	Alt 147	Note 6
94	148	ö	Alt 148	Note 6
95	149	ò	Alt 149	Note 6
96	150	û	Alt 150	Note 6
97	151	ù	Alt 151	Note 6
98	152	ÿ	Alt 152	Note 6
99	153	Ö	Alt 153	Note 6
9A	154	Ü	Alt 154	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
9B	155	¢	Alt 155	Note 6
9C	156	£	Alt 156	Note 6
9D	157	¥	Alt 157	Note 6
9E	158	Pt	Alt 158	Note 6
9F	159	ƒ	Alt 159	Note 6
A0	160	á	Alt 160	Note 6
A1	161	í	Alt 161	Note 6
A2	162	ó	Alt 162	Note 6
A3	163	ú	Alt 163	Note 6
A4	164	ñ	Alt 164	Note 6
A5	165	Ñ	Alt 165	Note 6
A6	166	a	Alt 166	Note 6
A7	167	o	Alt 167	Note 6
A8	168	¿	Alt 168	Note 6
A9	169	⌐	Alt 169	Note 6
AA	170	¬	Alt 170	Note 6
AB	171	½	Alt 171	Note 6
AC	172	¼	Alt 172	Note 6
AD	173	¡	Alt 173	Note 6
AE	174	<<	Alt 174	Note 6
AF	175	>>	Alt 175	Note 6
B0	176	⁞	Alt 176	Note 6
B1	177	▒	Alt 177	Note 6
B2	178	▓	Alt 178	Note 6
B3	179		Alt 179	Note 6
B4	180		Alt 180	Note 6
B5	181		Alt 181	Note 6
B6	182		Alt 182	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
B7	183		Alt 183	Note 6
B8	184		Alt 184	Note 6
B9	185		Alt 185	Note 6
BA	186		Alt 186	Note 6
BB	187		Alt 187	Note 6
BC	188		Alt 188	Note 6
BD	189		Alt 189	Note 6
BE	190		Alt 190	Note 6
BF	191		Alt 191	Note 6
C0	192		Alt 192	Note 6
C1	193		Alt 193	Note 6
C2	194		Alt 194	Note 6
C3	195		Alt 195	Note 6
C4	196		Alt 196	Note 6
C5	197		Alt 197	Note 6
C6	198		Alt 198	Note 6
C7	199		Alt 199	Note 6
C8	200		Alt 200	Note 6
C9	201		Alt 201	Note 6
CA	202		Alt 202	Note 6
CB	203		Alt 203	Note 6
CC	204		Alt 204	Note 6
CD	205		Alt 205	Note 6
CE	206		Alt 206	Note 6
CF	207		Alt 207	Note 6
D0	208		Alt 208	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
D1	209		Alt 209	Note 6
D2	210		Alt 210	Note 6
D3	211		Alt 211	Note 6
D4	212		Alt 212	Note 6
D5	213		Alt 213	Note 6
D6	214		Alt 214	Note 6
D7	215		Alt 215	Note 6
D8	216		Alt 216	Note 6
D9	217		Alt 217	Note 6
DA	218		Alt 218	Note 6
DB	219		Alt 219	Note 6
DC	220		Alt 220	Note 6
DD	221		Alt 221	Note 6
DE	222		Alt 222	Note 6
DF	223		Alt 223	Note 6
EO	224	α	Alt 224	Note 6
E1	225	β	Alt 225	Note 6
E2	226	Γ	Alt 226	Note 6
E3	227	π	Alt 227	Note 6
E4	228	Σ	Alt 228	Note 6
E5	229	σ	Alt 229	Note 6
E6	230	μ	Alt 230	Note 6
E7	231	τ	Alt 231	Note 6
E8	232	Φ	Alt 232	Note 6
E9	233	θ	Alt 233	Note 6
EA	234	Ω	Alt 234	Note 6
EB	235	δ	Alt 235	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
EC	236	∞	Alt 236	Note 6
ED	237	φ	Alt 237	Note 6
EE	238	ε	Alt 238	Note 6
EF	239	∩	Alt 239	Note 6
F0	240	≡	Alt 240	Note 6
F1	241	±	Alt 241	Note 6
F2	242	≥	Alt 242	Note 6
F3	243	≤	Alt 243	Note 6
F4	244		Alt 244	Note 6
F5	245		Alt 245	Note 6
F6	246	÷	Alt 246	Note 6
F7	247	≈	Alt 247	Note 6
F8	248	°	Alt 248	Note 6
F9	249	∙	Alt 249	Note 6
FA	250	·	Alt 250	Note 6
FB	251	√	Alt 251	Note 6
FC	252	^{n}	Alt 252	Note 6
FD	253	^{2}	Alt 253	Note 6
FE	254	■	Alt 254	Note 6
FF	255	**BLANK**	Alt 255	Note 6

Table Notes

1. Asterisk (*) can be typed by pressing the * key or, in the shift mode, pressing the 8 key.
2. Period (.) can be typed by pressing the . key or, in the shift or Num Lock mode, pressing the Del key.
3. Numeric characters 0-9 can be typed by pressing the numeric keys on the top row of the keyboard or, in the shift or Num Lock mode, pressing the numeric keys in the keypad portion of the keyboard.
4. Uppercase alphabetic characters (A-Z) can be typed by pressing the character key in the shift mode or the Caps Lock mode.
5. Lowercase alphabetic characters (a-z) can be typed by pressing the character key in the normal mode or in Caps Lock and shift mode combined.
6. The three digits after the Alt key is typed from the numeric keypad. Character codes 001-255 may be entered in this fashion (with Caps Lock activated, character codes 97-122 display uppercase).

Quick Reference

DECIMAL VALUE	➡	0	16	32	48	64	80	96	112
⬇	HEXA-DECIMAL VALUE	0	1	2	3	4	5	6	7
0	0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	‘	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	☻	↕	"	2	B	R	b	r
3	3	♥	!!	#	3	C	S	c	s
4	4	♦	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↨	’	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◙	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	¦
13	D	♪	↔	—	=	M	]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	☼	▼	/	?	O	_	o	△

DECIMAL VALUE	➡	128	144	160	176	192	208	224	240
⬇	HEXA-DECIMAL VALUE	8	9	A	B	C	D	E	F
0	0	Ç	É	á	▒	└	╨	∝	≡
1	1	ü	æ	í	▒	┴	╤	β	±
2	2	é	Æ	ó	▒	┬	╥	Γ	≥
3	3	â	Ô	ú	│	├	╙	π	≤
4	4	ä	ö	ñ	┤	─	╘	Σ	⌠
5	5	à	ò	Ñ	╡	┼	╒	σ	⌡
6	6	å	û	ª	╢	╞	╓	μ	÷
7	7	ç	ù	º	╖	╟	╫	τ	≈
8	8	ê	ÿ	¿	╕	╚	╪	Φ	°
9	9	ë	Ö	⌐	╣	╔	┘	Θ	•
10	A	è	Ü	¬	║	╩	┌	Ω	•
11	B	ï	¢	½	╗	╦	■	δ	√
12	C	î	£	¼	╝	╠	▄	∞	n
13	D	ì	¥	¡	╜	═	▌	φ	2
14	E	Ä	₧	«	╛	╬	▐	∈	▮
15	F	Å	ƒ	»	┐	╧	▀	∩	BLANK 'FF'

Notes:

Glossary

This glossary includes terms and definitions from the *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems,* GC20-1699.

μ. Prefix micro; 0.000 001.

μs. Microsecond; 0.000 001 second.

A. Ampere.

ac. Alternating current.

accumulator. A register in which the result of an operation is formed.

active high. Designates a signal that has to go high to produce an effect. Synonymous with positive true.

active low. Designates a signal that has to go low to produce an effect. Synonymous with negative true.

adapter. An auxiliary device or unit used to extend the operation of another system.

address bus. One or more conductors used to carry the binary-coded address from the microprocessor throughout the rest of the system.

all points addressable (APA). A mode in which all points of a displayable image can be controlled by the user.

alphanumeric (A/N). Pertaining to a character set that contains letters, digits, and usually other characters, such as punctuation marks.

alternating current (ac). A current that periodically reverses its direction of flow.

American National Standard Code for Information Interchange (ASCII). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information exchange between data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

ampere (A). The basic unit of electric current.

A/N. Alphanumeric

analog. (1) Pertaining to data in the form of continuously variable physical quantities. (2) Contrast with digital.

AND. A logic operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the AND of P, Q, R,...is true if all statements are true, false if any statement is false.

AND gate. A logic gate in which the output is 1 only if all inputs are 1.

APA. All points addressable.

ASCII. American National Standard Code for Information Interchange.

assemble. To translate a program expressed in an assembler language into a computer language.

assembler. A computer program used to assemble.

assembler language. A computer-oriented language whose instructions are usually in one-to-one correspondence with computer instructions.

asynchronous transmission. (1) Transmission in which the time of occurrence of the start of each character, or block of characters, is arbitrary; once started, the time of occurrence of each signal representing a bit within a character, or block, has the same relationship to significant instants of a fixed time frame. (2) Transmission in which each information character is individually transmitted (usually timed by the use of start elements and stop elements).

audio frequencies. Frequencies that can be heard by the human ear (approximately 15 hertz to 20,000 hertz).

auxiliary storage. (1) A storage device that is not main storage. (2) Data storage other than main storage; for example, storage on magnetic disk. (3) Contrast with main storage.

BASIC. Beginner's all-purpose symbolic instruction code.

basic input/output system (BIOS). The feature of the IBM Personal System/2 that provides the level control of the major I/O devices and relieves the programmer from concern about hardware device characteristics.

baud. (1) A unit of signaling speed equal to the number of discrete conditions or signal events per second. For example, one baud equals one bit per second in a train of binary signals, one-half dot cycle per second in Morse code, and one 3-bit value per second in a train of signals each of which can assume one of eight states. (2) In asynchronous transmission, the unit of modulation rate corresponding to one unit of interval per second; that is, if the duration of the unit interval is 20 milliseconds, the modulation rate is 50 baud.

BCC. Block-check character.

BCD. Binary-coded decimal

beginner's all-purpose symbolic instruction code (BASIC). A programming language with a small repertoire of commands and a simple syntax, primarily designed for numeric applications.

binary. (1) Pertaining to a selection, choice, or condition that has two possible values or states. (2) Pertaining to a fixed radix numeration system having a radix of 2.

binary digit. (1) In binary notation, either of the characters 0 or 1. (2) Synonymous with bit.

binary notation. Any notation that uses two different characters, usually the binary digits 0 and 1.

binary synchronous communications (BSC). A uniform procedure, using a standardized set of control characters and control character sequences for synchronous transmission of binary – coded data between stations.

BIOS. Basic input/output system.

bit. Synonym for binary digit

bits per second (bps). A unit of measure representing the number of discrete binary digits transmitted by a device in one second.

block. (1) A string of records, a string of words, or a character string formed for technical or logic reasons, to be treated as an entity. (2) A set of things, such as words, characters, or digits, treated as a unit.

block-check character (BCC). In cyclic redundancy checking, a character that is transmitted by the sender after each message block and is compared with a block-check character computed by the receiver to determine if the transmission was successful.

boolean operation. (1) Any operation in which each of the operands and the result take one of two values. (2) An operation that follows the rules of boolean algebra.

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.

bps. Bits per second.

BSC. Binary synchronous communications.

buffer. (1) An area of storage that is temporarily reserved for use in performing an input/output operation, into which data is read or from which data is written. Synonymous with I/O area. (2) A portion of 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 representation of a character.

C. Celsius.

Cartesian coordinates. A system of coordinates for locating a point on a plane by its distance from each of two intersecting lines, or in space by its distance from each of three mutually perpendicular planes.

CAS. Column address strobe.

CCITT. International Telegraph and Telephone Consultative Committee.

Celsius (C). A temperature scale. Contrast with Fahrenheit (F).

CGA. Color/graphics adapter.

channel. A path along which signals can be sent; for example, data channel, output channel.

character generator. (1) In computer graphics, a functional unit that converts the coded representation of a graphic 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 data.

character set. (1) A finite set of characters upon which agreement has been reached and that is considered complete for some purpose. (2) A set of unique representations called characters. (3) A defined collection of characters.

characters per second (cps). A standard unit of measurement for the speed at which a printer prints.

chip select (CS). A signal, line, or bit that activates a specified device or circuit logic.

collector. An element in a transistor toward which current flows.

column address strobe (CAS). A signal that latches the column addresses in a memory chip.

complement. A number that can be derived from a specified number by subtracting it from a second specified number.

conjunction. Synonym for AND operation.

contiguous. Touching or joining at the edge or boundary; adjacent.

cps. Characters per second.

CRC. Cyclic redundancy check.

CS. Chip select.

CTS. Clear to send. Associated with modem control.

cyclic redundancy check (CRC). (1) A redundancy check in which the check key is generated by a cyclic algorithm. (2) A system of error checking performed at both the sending and receiving station after a block-check character has been accumulated.

cylinder. (1) The set of all tracks with the same nominal distance from the axis about which the disk rotates. (2) The tracks of a disk storage device that can be accessed without repositioning the access mechanism.

daisy-chained. Two or more devices or programs attached or linked in series.

DAC. Digital-to-analog converter

dB. 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.

DIN connector. One of the connectors specified by the DIN committee.

DIP. 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.

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.

diskette. A thin, flexible magnetic disk and a protective jacket, in which the disk is permanently enclosed. Synonymous with flexible or floppy disk.

diskette drive. A device for storing data on and retrieving data from a diskette.

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.

DMA. Direct memory access.

DSR. Data set ready. Associated with modem control.

DTL. Data length - a field value for diskette and fixed disk operation.

DTR. In the IBM Personal Computer, data terminal ready. Associated with modem control.

dual in-line package (DIP). A widely used container for an integrated circuit. DIPs have pins in two parallel rows. The pins are spaced 1/10 inch apart. See also DIP switch.

duplex. (1) In data communication, pertaining to a simultaneous two-way independent transmission in both directions. (2) Contrast with half-duplex.

EBCDIC. Extended binary-coded decimal interchange code.

ECC. Error checking and correction.

EIA. Electronic Industries Association.

enable. To initiate the operation of a circuit or device.

end of block (EOB). A code that marks the end of a block of data.

end of file (EOF). An internal label, immediately following the last record of a file, signaling the end of that file. It may include control totals for comparison with counts accumulated during processing.

end-of-text (ETX). A transmission control character used to terminate text.

end-of-transmission (EOT). A transmission control character used

to indicate the conclusion of a transmission, which may have included one or more texts and any associated message headings.

end-of-transmission-block (ETB). A transmission control character used to indicate the end of a transmission block of data when data is divided into such blocks for transmission purposes.

EOB. End of block.

EOF. End of file.

EOI. End of interrupt.

EOT. End-of-transmission.

erasable programmable read-only memory (EPROM). A PROM in which the user can erase old information and enter new information.

error checking and correction (ECC). The detection and correction of all single-bit errors, plus the detection of double-bit and some multiple-bit errors.

ESC. The escape character.

escape character (ESC). A code extension character used, in some cases, with one or more succeeding characters to indicate by some convention or agreement that the coded representations following the character or the group of characters are to be interpreted according to a different code or according to a different coded character set.

extended binary-coded decimal interchange code (EBCDIC). A set of 256 characters, each represented by 8 bits.

F. Fahrenheit.

Fahrenheit (F). A temperature scale. Contrast with Celsius (C).

falling edge. Synonym for negative-going edge.

FCC. Federal Communications Commission.

fetch. To locate and load a quantity of data from storage.

FF. The form feed character.

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.

FIFO (first-in-first out). A queuing technique in which the next item to be retrieved is the item that has been in the queue for the longest time.

fixed disk drive. A unit consisting of nonremovable magnetic disks, and a device for storing data on and retrieving data from the disks.

flag. (1) Any of various types of indicators used for identification. (2) A character that signals the occurrence of some condition, such as the end of a word. (3) Deprecated term for mark.

flexible disk. Synonym for diskette.

flip-flop. A circuit or device containing active elements, capable

of assuming either one of two stable states at a given time.

font. A family or assortment of characters of a given size and style; for example, 10 point Press Roman medium.

format. The arrangement or layout of data on a data medium.

frame. (1) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures. Each frame begins and ends with a flag. (2) In data transmission, the sequence of contiguous bits bracketed by and including beginning and ending flag sequences.

g. Gram.

G. (1) Prefix giga; 1 000 000 000. (2) When referring to computer storage capacity, 1 073 741 824 bytes (2 to the 30th power).

gate. (1) A combinational logic circuit having one output channel and one or more input channels, such that the output channel state is completely determined by the input channel states. (2) A signal that enables the passage of other signals through a circuit.

gram (g). A unit of weight (equivalent to 0.035 ounces).

graphic. A symbol produced by a process such as handwriting, drawing, or printing.

hertz (Hz). A unit of frequency equal to one cycle per second.

hex. Common abbreviation for hexadecimal. Also, hexadecimal can be noted with an H following the value.

hexadecimal. 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.

high-order position. The leftmost position in a string of characters. See also most-significant digit.

Hz. Hertz

immediate instruction. An instruction that contains within itself an operand for the operation specified, rather than an address of the operand.

index register. A register whose contents may be used to modify an operand address during the execution of computer instructions.

inhibited. Pertaining to a state of a device that does not allow interruptions, or instructions.

initialize. To set counters, switches, addresses, or contents of storage to 0 or other starting values at the beginning of, or at prescribed points in, the operation of a computer routine.

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. (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. In a programming language, a meaningful expression that specifies one operation and identifies its operands, if any.

instruction set. The set of instructions of a computer, of a programming language, or of the programming languages in a programming system.

intensity. In computer graphics, the amount of light emitted at a display point

interface. A device that alters or converts electrical signals between distinct devices, programs, or systems.

interleave. To arrange parts of a sequence 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.

irrecoverable error. An error that makes recovery impossible without the use of recovery techniques external to the computer program or run.

k. Prefix kilo; 1000.

K. 1024 (1024 = 2 to the 10th power.). When referring to storage capacity, 1024 bytes.

keylock. A device that deactivates the keyboard and locks the cover on for security.

kg. Kilogram; 1000 grams.

kHz. Kilohertz; 1000 hertz.

latch. (1) A simple logic-circuit storage element. (2) A feedback loop in sequential digital circuits used to maintain a state.

least-significant digit. The rightmost digit. See also low-order position.

load. In programming, to enter data into storage or working registers.

low-order position. The rightmost position in a string of characters. See also least-significant digit.

m. (1) Prefix milli; 0.001. (2) Meter.

M. (1) Prefix mega; 1 000 000. (2) When referring to computer storage capacity, 1 048 576 bytes (1 048 576 = 2 to the 20th power.)

mA. Milliampere; 0.001 ampere.

machine code. The machine language used for entering text and program instructions onto the recording medium or into storage and which is subsequently used for processing and printout.

machine language. (1) A language that is used directly by a machine. (2) Deprecated term for computer instruction code.

magnetic disk. (1) A flat circular plate with a magnetizable surface layer on which data can be stored by magnetic recording. (2) See also diskette.

mark. A symbol or symbols that indicate the beginning or the end of a field, of a word, of an item of data, or of a set of data such as a file, a record, or a block.

mask. (1) A pattern of characters that is used to control the retention or elimination of portions of another pattern of characters. (2) To use a pattern of characters to control the retention or elimination of portions of another pattern of characters.

masked. Synonym for disabled.

mega (M). Prefix 1 000 000.

megahertz (MHz). 1 000 000 hertz.

MFM. Modified frequency modulation.

micro (μ). Prefix 0.000,001.

microcode. A code, representing the instructions of an instruction set, implemented in a part of storage that is not program-addressable.

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.

milli (m). Prefix 0.001.

milliampere (mA). 0.001 ampere.

millisecond (ms). 0.001 second.

mnemonic. A symbol chosen to assist the human memory; for example, an abbreviation such as "mpy" for "multiply."

mode. (1) A method of operation; for example, the binary mode, the interpretive mode, the alphanumeric mode. (2) The most frequent value in the statistical sense.

modem (modulator-demodulator). A device that converts serial (bit by bit) digital signals from a business machine (or data communication equipment) to analog signals that are suitable for transmission in a telephone network. The inverse function is also performed by the modem on reception of analog signals.

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.

modulation. The process by which some characteristic of one wave (usually high frequency) is varied in accordance with another wave or signal (usually low frequency). This technique is used in modems to

make business-machine signals compatible with communication facilities.

modulo check. A calculation performed on values entered into a system. This calculation is designed to detect errors.

modulo- check. A check in which an operand is divided by a number N (the modulus) to generate a remainder (check digit) that is retained with the operand. For example, in a modulo-7 check, the remainder will be 0, 1, 2, 3, 4, 5, or 6. The operand is later checked by again dividing it by the modulus; if the remainder is not equal to the check digit, an error is indicated.

most-significant digit. The leftmost (nonzero) digit. See also high-order position.

ms. Millisecond; 0.001 second.

multiplexer. A device capable of interleaving the events of two or more activities, or capable of distributing the events of an interleaved sequence to the respective activities.

n. Prefix nano; 0.000,000,001.

NAND. A logic operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the NAND of P, Q ,R,... is true if at least one statement is false, false if all statements are true.

NAND gate. A gate in which the output is 0 only if all inputs are 1.

nano (n). Prefix 0.000,000,001.

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.

NMI. Non-maskable interrupt

nonreturn-to-zero change-on-ones recording (NRZI). A transmission encoding method in which the data terminal changes the signal to the opposite state to send a binary 1 and leaves it in the same state to send a binary 0.

nonreturn-to-zero (inverted) recording (NRZI). Deprecated term for non-return-to-zero change-on-ones recording.

NOR. A logic operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the NOR of P, Q, R,... is true if all statements are false, false if at least one statement is true.

NOR gate. A gate in which the output is 0 only if at least one input is 1.

NOT. A logical operator having the property that if P is a statement, then the NOT of P is true if P is false, false if P is true.

NRZI. Nonreturn-to-zero (inverted) recording.

ns. Nanosecond; 0.000,000,001 second.

NUL. The null character.

null character (NUL). A control character that is used to accomplish media-fill or time-fill, and that may be inserted into or removed from, a sequence of characters without affecting the meaning of the sequence; however, the control of the equipment or the format may be affected by this character.

open collector. A switching transistor without an internal connection between its collector and the voltage supply. A connection from the collector to the voltage supply is made through an external (pull-up) resistor.

operand. (1) An entity to which an operation is applied. (2) That which is operated upon. An operand is usually identified by an address part of an instruction.

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.

OR. A logic operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the OR of P, Q, R,...is true if at least one statement is true, false if all statements are false.

OR gate. A gate in which the output is 1 only if at least one input is 1.

output. Pertaining to a device, process, or channel involved in an output process, or to the data or states involved in an output process.

output process. (1) The process that consists of the delivery of data from a data processing system, or from any part of it. (2) The return of information from a data processing system to an end user, including the translation of data from a machine language to a language that the end user can understand.

overcurrent. A current of higher than specified strength.

overflow indicator. (1) An indicator that signifies when the last line on a page has been printed or passed. (2) An indicator that is set on if the result of an arithmetic operation exceeds the capacity of the accumulator.

overrun. Loss of data because a receiving device is unable to accept data at the rate it is transmitted.

overvoltage. A voltage of higher than specified value.

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) Contrast with 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.

parity bit. A binary digit appended to a group of binary digits to make the sum of all the digits either always odd (odd parity) or always even (even parity).

parity check. A redundancy check that uses a parity bit.

picture element (PEL). In computer graphics, a basic graphic element that can be used to construct a display image; for example, a dot, a line segment, a character.

polling. (1) Interrogation of devices for purposes such as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (2) The process whereby stations are invited, one at a time, to transmit.

POR. Power-on-reset

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.

priority. A rank assigned to a task that determines its precedence in receiving system resources.

propagation delay. (1) The time necessary for a signal to travel from one point on a circuit to another. (2) The time delay between a signal change at an input and the corresponding change at an output.

protocol. (1) A specification for the format and relative timing of information exchanged between communicating parties. (2) The set of rules governing the operation of functional units of a communication system that must be followed if communication is to be achieved.

pulse. A variation in the value of a quantity, short in relation to the time schedule of interest, the final value being the same as the initial value.

radix. Another term for base.

radix numeration system. A positional representation system in which the ratio of the weight of any one digit place to the weight of the digit place with the next lower weight is a positive integer (the radix). The permissible values of the character in any digit place range from 0 to one less than the radix.

RAM. Random access memory. Read/write memory.

RAS. Row address strobe.

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.

read/write memory. A storage device whose contents can be modified. Also called RAM.

recoverable error. An error condition that allows continued execution of a program.

red-green-blue-intensity (RGBI). The description of a direct-drive color monitor that accepts input signals of red, green, blue, and intensity.

redundancy check. A check that depends on extra characters attached to data for the detection of errors. See cyclic redundancy check.

register. (1) A storage device, having a specified storage capacity such as a bit, a byte, or word, and usually intended for a special purpose. (2) A storage device in which specific data is stored.

retry. To resend the current block of data (from the last EOB or ETB) a prescribed number of times, or until it is entered correctly or accepted.

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.

RF modulator. The device used to convert the composite video signal to the antenna level input of a home TV.

RGBI. Red-green-blue-intensity.

RI. Ring indicate; a signal associated with modem control.

rising edge. Synonym for positive-going edge.

ROM. Read-only memory.

ROM/BIOS. The ROM resident basic input/output system, which provides the level control of the major I/O devices in the computer system.

row address strobe (RAS). A signal that latches the row address in a memory chip.

RS-232C. A standard by the EIA for communication between computers and external equipment.

RTS. Request to send. Associated with modem control.

SDLC. Synchronous Data Link Control.

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. In English, 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) Contrast with parallel.

serializer/deserializer (SERDES). A device that serializes output from, and deserializes input to, a business machine.

short circuit. A low-resistance path through which current flows, rather than through a component or circuit.

sink. A device or circuit into which current drains.

SIP. Single-inline package.

source. The origin of a signal or electrical energy.

square wave generator. A signal generator delivering an output signal having a square waveform.

start bit. A signal to a receiving mechanism to get ready to receive data or perform a function.

stop bit. A signal to a receiving mechanism to wait for the next signal.

strobe. An instrument that emits adjustable-rate flashes of light. Used to measure the speed of rotating or vibrating objects.

synchronization. The process of adjusting the corresponding significant instants of two signals to obtain the desired phase relationship between these instants.

Synchronous Data Link Control (SDLC). A protocol for management of data transfer over a data link.

synchronous transmission. (1) Data transmission in which the time of 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, by means of correction, in a desired phase relationship.

time-out. (1) A parameter related to an enforced event designed to occur at the conclusion of a predetermined elapsed time. A time-out condition can be cancelled by the receipt of an appropriate time-out cancellation signal. (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

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.

transmission. (1) The sending of data from one place for reception elsewhere. (2) In ASCII and data communication, a series of characters including headings and text. (3) One or more blocks or messages. For BSC and start-stop devices, a transmission is terminated by an EOT character. (4) Synonymous with data transmission.

TTL. Transistor-transistor logic.

typematic key. A keyboard key that repeats its function when held pressed.

vector. In computer graphics, a directed line segment.

video. Computer data or graphics displayed on a cathode ray tube, monitor, or display.

volt. The basic practical unit of electric pressure. The potential that causes electrons to flow through a circuit.

W. Watt.

watt. The practical unit of electric power.

word. (1) A sequence of 16 adjacent binary digits that are operated upon as a unit. (2) A character string or a bit string considered as an entity.

write. To make a permanent or transient recording of data in a storage device or on a 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.

Bibliography

Intel Corporation. *The 8086 Family User's Manual*. This manual introduces the 8086 family of microcomputing components and serves as a reference in system design and implementation.

Intel Corporation. *8086/8087/8088 Macro Assembly Reference Manual for 8088/8085 Based Development System*. This manual describes the 8086/8087/8088 Macro Assembly Language and is intended for persons who are familiar with assembly language.

Intel Corporation. *Component Data Catalog*. This book describes Intel components and their technical specifications.

Motorola, Inc. *The Complete Microcomputer Data Libary*. This book describes Motorola components and their technical specifications.

National Semiconductor Corporation. *8250 Asynchronous Communications Element*. This book documents physical and operating characteristics of the INS 8250.

Notes:

Index

Special Characters

-MEMW (memory write) 1-27

A

AAA 6-10
AAD 6-12
AAM 6-12
AAS 6-12
adapters with ROM 5-10
ADC 6-10
ADD 6-9
address
  map, I/O 1-24
  serial port 1-109
  video subsystem 1-44
AEN (address enable) 1-26
ALE (address latch enable), I/O channel 1-26
alphanumeric (A/N) modes 1-40
alternate disk reset 5-43
alternate key 4-20
alternate parameter table, video 1-71
alternate select 5-22
AND 6-13
Arabic keyboard 4-24
arithmetic instructions 6-9, 6-26
ASCII characters 7-2
ASCII, extended 4-11
async baud rate table 5-75
asynchronous communications, interrupt 14 5-46

B

basic assurance test, keyboard 4-4
BASIC reserved interrupts 5-7
BAT commands, keyboard 4-7
baud rate table 5-75
baud-rate generator 1-119
beeper 1-127
beeper control 1-11
Belgian keyboard 4-25
binary integers (coprocessor) 2-4
BIOS
  alternate disk reset 5-43
  alternate select 5-22
  baud rate initialization table 5-75
  character generator routine 5-20
  communications 5-46
  current video state 5-18
  data area 5-67
  device busy 5-53
  device close 5-51
  device open 5-51
  disk change status 5-36
  diskette 5-31
  diskette parameter table 5-75
  display code, r/w 5-25
  equipment determination 5-29
  event wait 5-52
  extended data area 5-72
  extended initialize 5-49
  extended keyboard read 5-60
  extended keystroke status 5-60
  extended shift status 5-61
  fixed disk 5-38
  fixed disk parameter table 5-73
  format track 5-34
  format track (disk) 5-41
  hardware interrupts 5-4
  initialize drive 5-42
  initialize the communications port 5-46

initialize the printer port 5-62
int 1A, clock services 5-64
int 17, printer 5-62
int 19, bootstrap 5-63
interrupt complete 5-53
interrupt 02, NMI routine 5-11
interrupt 05, print screen 5-11
interrupt 08, system timer 5-12
interrupt 09, keyboard 5-13
interrupt 10, video 5-14
interrupt 16 5-58
joystick 5-52
keyboard intercept 5-51
keyboard write 5-59
keystroke status 5-58
memory size determine 5-30
model byte 5-76
palette registers 5-18
parameter passing 5-4
park heads 5-44
pointing device 5-55
print character 5-62
program termination 5-52
programming considerations 5-9
read alarm time and status 5-65
read clock date 5-64
read clock time 5-64
read cursor position 5-15
read DASD type 5-36
read DASD type (disk) 5-44
read day counter 5-65
read dot 5-18
read drive parameters 5-35, 5-42
read sectors into memory 5-32, 5-40
read status 5-32, 5-39, 5-48, 5-62
read system time counter 5-64
read value at cursor position 5-16
recalibrate 5-44
receive character 5-47
reset disk system 5-38
reset diskette 5-31
reset real time clock alarm 5-65
return config parameters 5-54
return ext segment address 5-54
ROM table 5-73
scroll active page down 5-16
scroll active page up 5-16
seek 5-43
select active display page 5-15
send character 5-47
set clock alarm 5-65
set clock date 5-65
set clock time 5-64
set color palette 5-17
set cursor position 5-15
set cursor type 5-15
set DASD type 5-36
set day counter 5-66
set media type 5-37
set system time counter 5-64
set typamatic rate 5-59
shift status 5-58
software interrupts 5-5
system request 5-52
system services 5-51
test drive ready 5-43
verify sectors 5-33, 5-41
video state information 5-25
video tables 5-26
wait 5-52
write character at cursor 5-17
write dot 5-17
write sectors from memory 5-33, 5-40
write string 5-24
write teletype to display 5-18
write value at cursor 5-16
block diagram
coprocessor 2-5
parallel port 1-123
serial port 1-108
shared interrupt 1-15
system board 1-4
system timer 1-9
video 1-37
bootstrap 5-63
border control register 1-56

break code 4-3
break key 4-21
buffer, keyboard 4-3
bus card 1-23

C

cable 4-41
CALL 6-15
Canadian keyboard 4-26
caps lock key 4-20
CBW 6-12
CGA border control register 1-56
CGA mode control register 1-55
channel check (-IO CH CK) 1-27
channel ready (IO CH RDY) 1-27
channel, I/O
character box 1-38
character codes 4-11
character generator routine 5-20
character matrix 1-66
character set, 512 1-68
characters 7-2
chip select 1-8
CLC 6-20
CLD 6-20
CLI 6-20
CLK 1-26
clock (CLK) 1-26
clock and data signals 4-5
CMC 6-20
CMP 6-11
CMPS 6-15
codes
  character 4-11
  extended 4-17
  make/break 4-8
color palette load 1-59
color/graphics
  See video
colors, default 1-64
colors, mode 4,5 1-56
commands
  diskette drive 1-84
  keyboard 4-7
comparison instructions 6-26
configuration register 1-82
connectors
  fixed disk 1-107
  I/O channel 1-25
  keyboard and pointing device 4-41
  parallel port 1-126
  power supply 1-129, 3-6
  serial port 1-122
  system board 1-128
  video 1-78
constants instructions 6-29
control key 4-19
control transfer instructions 6-15
controller, diskette 1-82
Ctrl state 4-17
current video state 5-18
cursor position 5-15
CWD 6-12

D

Danish keyboard 4-27
data
data area, BIOS 5-67
data bits (D0-D7) 1-26
data output, keyboard 4-6
data stream, keyboard 4-6
data stream, serial port 1-109
data transfer instructions 6-7, 6-24
DEC 6-11
decimal integers (coprocessor) 2-4
default colors 1-64
delay, typematic 4-3
description
  fixed disk connector 1-106
  I/O channel 1-26
  keyboard 4-3
  parallel port 1-123
  video 1-36
device busy 5-53
device close 5-51
device open 5-51
digital I/O registers 1-81
disk change line status 5-36
diskette controller select 1-8
diskette drive interface 1-79
  change signal 1-81
  commands 1-84
  data transfer 1-82

parameter table 5-75
phase-lock loop 1-79
registers 1-80
status registers 1-100
diskette interrupt, BIOS 5-31
diskette parameter table 5-75
display combination code, r/w 5-25
display support, video 1-38
DIV 6-12
DMA request 1 to 3 (DRQ1 – DRQ3) 1-26
DOS reserved interruts 5-7
drive types 5-73
Dutch keyboard 4-28

E

encoding, keyboard 4-11
equipment determination 5-29
ESC 6-21
event wait 5-52
extended ASCII 4-11
extended codes 4-17
extended codes, keyboard 4-19
extended data area, BIOS 5-72
extended initialize 5-49
extended keyboard read 5-60
extended keystroke status 5-60
extended shift status 5-61

F

FABS 6-28
FADD 6-26
FCHS 6-28
FCLEX 6-30
FCOM 6-26
FCOMP 6-26
FCOMPP 6-26
FDECSTP 6-30
FDISI 6-29
FDIV 6-27
FENI 6-29
FFREE 6-31
FIFO 4-3
FINCSTP 6-30
FINIT 6-29
fixed disk 1-106
parameter table, BIOS 5-73
fixed disk controller select 1-8
fixed disk routines 5-38
FLD 6-24
FLDCW 6-30
FLDENV 6-30
FLDLG2 6-29
FLDLN2 6-29
FLDL2T 6-29
FLDP1 6-29
FLDZ 6-29
FLD1 6-29
FMUL 6-27
FNOP 6-31
font save table 1-71
fonts, RAM loadable 1-65
format track 5-34
format track (disk) 5-41
FPATAN 6-28
FPREM 6-27
FPTAN 6-28
French keyboard 4-29
FRNDINT 6-28
FRSTOR 6-30
FSAVE 6-30
FSCALE 6-27
FSQRT 6-27
FST 6-25
FSTCW 6-30
FSTENV 6-30
FSTP 6-25
FSTSW 6-30
FSUB 6-27
FTST 6-26
function enable 1-8
functionality and state information 5-26
FWAIT 6-31
FXAM 6-26
FXCH 6-25
FXTRACT 6-28
FYL2X 6-28
FYL2XP1 6-28
F2XM1 6-28

G

gate array
diskette drive 1-79
I/O support 1-8
system support 1-6
German keyboard 4-30
graphics modes 1-41

H

hardware interrupts 1-13, 5-4
HLT 6-20

I

I/O channel 1-23
-memory refresh 1-27
-MEMR, -MEMW 1-27
address map 1-24
ALE (address latch enable) 1-26
channel check (-IO CH CK) 1-27
channel ready (IO CH RDY) 1-27
CLK 1-26
connector 1-25
description 1-26
oscillator (OSC) 1-27
read (-IOR) 1-27
Reset Drive (RESET DRV) 1-27
terminal count (TC) 1-28
timing 1-28
write (-IOW) 1-27
I/O port 1-11, 1-127
I/O support gate array 1-8
IDIV 6-12
IMUL 6-12
IN 6-8
INC 6-10
information, return video 5-25
initialization tables 1-63
initialize
initialize drive 5-42
initialize the communications port 5-46
initialize the printer port 5-62
instructions
arithmetic 6-9, 6-26
comparison 6-26
constants 6-29
control transfer 6-15
data transfer 6-7, 6-24
diskette drive 1-84
logic 6-13
rotate 6-13
shift 6-13
string manipulation 6-15
INT 6-19
interrupt 1-13
BIOS interface listing 5-5
bootstrap 5-63
hardware 1-13, 5-4
keyboard 5-13
NMI routine 5-11
print screen 5-11
printer 5-62
real time clock services 5-64
shared logic 1-15
sharing 1-14
software 5-5
system timer 5-12
10, video 5-14
11, equipment determination 5-29
12, memory size 5-30
13, diskette 5-31
13, fixed disk 5-38
14, communications 5-46
15, system services 5-51
16, keyboard 5-58
interrupt complete 5-53
interrupt identification register 1-112
interrupt requests 1-27
INTO 6-19
IO CH CK 1-27
IRET 6-19
Israeli keyboard 4-31
Italian keyboard 4-32

J

JB/JNAE 6-17
JBE/JNA 6-17
JCXZ 6-18
JE/JZ 6-16
JL/JNGE 6-17
JLE/JNG 6-17
JMP 6-16
JNB/JAE 6-18
JNBE/JA 6-18
JNE/JNZ 6-17
JNL/JGE 6-17
JNLE/JG 6-17
JNO 6-18
JNP/JPO 6-18
JNS 6-18
JO 6-17
joystick 5-52
JP/JPE 6-17
JS 6-17

K

key-code scanning 4-3
keyboard 4-3
- basic assurance test 4-4
- buffer 4-3
- cable 4-41
- commands 4-7
- connector 1-129
- data output 4-6
- data stream 4-6
- encoding 4-11
- interrupt 09 5-13
- interrupt 16 5-58
- key-code scanning 4-3
- layout 4-12
- line protocol 4-4
- make/break 4-3
- POR (power-on reset) 4-4
- routine 4-22
- scan codes 4-8
- shift states 4-19

keyboard controller 1-9
keyboard intercept 5-51
keyboard layouts
- Arabic 4-24
- Belgian 4-25
- Canadian 4-26
- Danish 4-27
- Dutch 4-28
- French 4-29
- German 4-30
- Israeli 4-31
- Italian 4-32
- Latin American 4-33
- Norwegian 4-34
- Portuguese 4-35
- Spanish 4-36
- Swedish 4-37
- Swiss 4-38
- UK English 4-39
- US English 4-40

keyboard write 5-59
keys, typematic 4-3
keystroke status 5-58

L

LAHF 6-9
Latin American keyboard 4-33
layout, keyboard 4-12
layouts, keyboard
- Arabic 4-24
- Belgian 4-25
- Canadian 4-26
- Danish 4-27
- Dutch 4-28
- French 4-29
- German 4-30
- Israeli 4-31
- Italian 4-32
- Latin American 4-33
- Norwegian 4-34
- Portuguese 4-35
- Spanish 4-36
- Swedish 4-37
- Swiss 4-38
- UK English 4-39
- US English 4-40

LDS 6-9
LEA 6-9
LES 6-9
line protocol 4-4

listing, software interrupt 5-5
loadable fonts 1-65
loading color palette 1-59
locations, system board 1-128
LOCK 6-21
LODS 6-15
logic instructions 6-13
LOOP 6-18
LOOPNZ/LOOPNE 6-18
LOOPZ/LOOPE 6-18

M

main status register 1-82
make code 4-3
math coprocessor
  block diagram 2-5
  control word 2-5
  data types 2-4
  hardware interface 2-4
  NMI 2-5
  QS1 2-4
memory
  control/status register 1-22
  read-only 1-22
  read/write 1-22
  reserved locations 5-8
  ROM table 5-73
memory map
  BIOS 5-8
  video font 1-65
  video storage 1-42
memory map, system 1-5
memory read (-MEMR) 1-27
memory refresh (-MREF) 1-27
memory size determine 5-30
microprocessor 1-5
mode control register 1-55
mode 4,5 colors 1-56
model byte 5-76
modem control/status registers 1-115
modes, video 1-39
modules, RAM 1-22
modules, ROM/EPROM 1-22
MOV 6-7
MOVS 6-15
MUL 6-12

N

NEG 6-11
NMI (coprocessor) 2-5
non-maskable interrupt routine 5-11
NOP 6-20
Norwegian keyboard 4-34
NOT 6-13
Num Lock state 4-17
number lock key 4-20

O

OR 6-14
oscillator (OSC), I/O channel 1-27
OUT 6-8
output, keyboard 4-6
Overrun command 4-7

P

page down 5-16
palette registers 5-18
parallel port 1-123
parallel port select 1-8
parameter passing (BIOS) 5-4
parameter table, fixed disk 5-73
park heads 5-44
pause key 4-21
planar control register 1-8
planar RAM control register 1-22
pointing device 5-55
pointing device controller 1-9
POP 6-8
POPF 6-9
Portuguese keyboard 4-35
power good 3-5
power requirements 4-42
power supply 3-3
  circuit protection 3-5
  connectors 1-129, 3-6
  inputs 3-4
  outputs 3-4
  power good signal 3-5
power-on routine, keyboard 4-4

print character 5-62
print screen key 4-21
print screen, interrupt 05 5-11
printer interrupt 5-62
priorities, shift key 4-20
processor control, 8087 6-29
program termination 5-52
programming considerations
  BIOS 5-9
  chip selects 1-8
  interrupt sharing 1-15
  video 1-72
protocol 4-4
PUSH 6-7
PUSHF 6-9

Q

queue status line (QS) 2-4
quick reference charts 7-8
quick reference, character set 7-8

R

RAM modules 1-22
RAM subsystem 1-22
RAS port registers 1-80
rate, typematic 4-3
RCL 6-13
RCR 6-13
read alarm time and status 5-65
read clock time 5-64
read cursor position 5-15
read dasd type 5-36
read DASD type (disk) 5-44
read day counter 5-65
read dot 5-18
read drive parameters 5-35, 5-42
read sectors into memory 5-32, 5-40
read status 5-39, 5-48, 5-62
read status diskette 5-32
read system time counter 5-64
read value at cursor position 5-16
read/write display code 5-25
read, I/O channel 1-27
read, memory (-MEMR) 1-27
readclock date 5-64
ready (IO CH RDY) 1-27
real numbers (coprocessor) 2-4
real-time clock services 5-64
recalibrate 5-44
receive character 5-47
register, border control 1-56
registers
  color palette 1-58
  diskette drive 1-80
  memory controller 1-45
  parallel port 1-123
  planar control 1-8
  planar RAM control 1-22
  serial port 1-110
  video 1-44
  video formatter 1-54
REP 6-15
request/grant 1-6
requests
  DMA 1-26
  interrupts 1-27
reserved interrupts, BASIC and DOS 5-7
reset disk system 5-38
reset diskette system 5-31
reset drive signal (RESET DRV) 1-27
reset real time clock alarm 5-65
reset, power-on 4-4
reset, system 4-21
RET 6-16
return config parameters 5-54
return ext segment address 5-54
return video information 5-25
ROL 6-13
ROM subsystem 1-22
ROM table 5-73
ROM, adapters with 5-10
ROR 6-13
rotate instructions 6-13
routine, keyboard 4-22

S

SAHF 6-9
SAR 6-13
save table 1-71
SBB 6-11
scan codes 4-8
scanning, key-code 4-3
SCAS 6-15
scroll active page down 5-16
scroll active page up 5-16
scroll lock key 4-20
seek 5-43
select active display page 5-15
send character 5-47
serial port 1-108
  connector 1-122
  signals 1-120
serial port interrupt call 5-46
serial port select 1-8
set clock alarm 5-65
set clock date 5-65
set clock time 5-64
set color palette 5-17
set cursor position 5-15
set cursor type 5-15
set DASD type 5-36
set day counter 5-66
set media type 5-37
set system time counter 5-64
set typamatic rate 5-59
shared interrupt logic 1-15
sharing, interrupt 1-14
shift instructions 6-13
shift key 4-19
shift key priorities 4-20
shift states 4-19
shift status 5-58
SHL/SAL 6-13
SHR 6-13
signals (I/O)
  diskette 1-103
  fixed disk connector 1-106
  I/O channel 1-26
  keyboard 4-5
  parallel port 1-123
  RQ/GT 1-6
  serial port 1-120
software interrupts 5-5
Spanish keyboard 4-36
speaker (beeper) 1-127
speaker tone generation 1-10
specifications
  keyboard 4-42
  parallel port 1-123
  serial port 1-122
  system board 1-130
states, shift
static functionality table 5-27
status registers, diskette 1-100
STC 6-20
STD 6-20
STI 6-20
STOS 6-15
string manipulation
 instructions 6-15
SUB 6-10
subsystem
  RAM 1-22
  ROM 1-22
  video 1-36
support, joystick 5-52
supported drives 5-31
Swedish keyboard 4-37
Swiss keyboard 4-38
system board
  functional diagram 1-4
  locations 1-128
system clock (CLK) 1-26
system memory map 1-5
system request 5-52
system request key 4-21
system reset 4-21
system services interrupt 5-51
system services, interrupt 1A 5-64
system support gate array 1-6
system timer 1-9
system timer, interrupt 08 5-12
system, return parameters 5-54

T

tables, video 5-26
terminal count (TC) 1-28
TEST 6-14
test drive ready 5-43
text modes 1-40
timer/counter 1-9
timer, system 1-9
timing
  color palette 1-73
  DMA operation 1-34
  fixed disk 1-106
  I/O channel 1-28
  parallel port 1-125
tone generation, beeper 1-10
typematic keys 4-3

U

U.K. English keyboard 4-39
US English keyboard 4-40

V

vectors with special meanings 5-6
verify sectors 5-33, 5-41
video 1-36
  alternate parameters table 1-71
  BIOS tables 5-26
  border control 1-56
  character size 1-38
  color palette registers 1-58
  connector 1-78
  considerations 1-72
  default tables 1-63
  display format 1-40
  display support 1-38
  formatter registers 1-54
  interrupt 10 5-14
  loadable fonts 1-65
  memory controller registers 1-45
  memory maps 1-42
  mode 4,5 colors 1-56
  modes 1-39
  timing 1-73
  512 characters 1-68
video controller select 1-8
video state information 5-25

W

wait 5-52, 6-21
write character at cursor 5-17
write dot 5-17
write memory command (-MEMW) 1-27
write sectors from memory 5-33, 5-40
write string 5-24
write teletype to display 5-18
write value at cursor 5-16
write, I/O channel (-IOW) 1-27

X

XCHG 6-8
XLAT 6-9
XOR 6-14

Numerics

512 character set 1-68
8086 microprocessor 1-5
8253 timer/counter 1-9

© Copyright
International Business
Machines Corporation, 1987
All Rights Reserved

Printed in the
United States of America

References in this
publication to IBM
products or services do not
imply that IBM intends
to make them available
outside the United States.

80X0661