



Personal System/2™  
and Personal Computer  
BIOS Interface Technical Reference



Personal System/2™  
and Personal Computer  
BIOS Interface Technical Reference

## **First Edition (April, 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.

**THE PUBLICATION OF THE INFORMATION CONTAINED HEREIN IS NOT INTENDED TO AND DOES NOT CONVEY ANY RIGHTS OR LICENSES, EXPRESS OR IMPLIED, UNDER ANY IBM PATENTS, COPYRIGHTS, TRADEMARKS, MASK WORKS, OR ANY OTHER INTELLECTUAL PROPERTY RIGHTS.**

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. No part of this work may be reproduced or distributed in any form or by any means without prior permission in writing from the IBM Corporation.

---

# Preface

This technical reference provides Basic Input/Output System (BIOS) interface information. It is intended for developers who provide hardware or software products to operate with the following IBM products:

- IBM PC Convertible
- IBM PCjr™
- IBM Personal Computer
- IBM Personal Computer AT®
- IBM Personal Computer XT™
- IBM Personal Computer XT™ Model 286
- IBM Personal System/2™ Models 30, 50, 60, and 80
- IBM *Portable* Personal Computer
- IBM Color/Graphics Monitor Adapter
- IBM Enhanced Graphics Adapter
- IBM ESDI Fixed Disk Drive Adapter/A
- IBM Monochrome Display and Printer Adapter.

You should understand the concepts of computer architecture and programming before using this publication.

This technical reference is divided into two parts: **BIOS** and **Supplements**.

**BIOS** contains the following:

Section 1, "Introduction to BIOS," provides an overview of BIOS, interrupts, parameter passing, data areas and read-only memory (ROM) tables. It also describes how to determine the system BIOS version date.

---

PCjr, Personal Computer XT, and Personal System/2 are trademarks of the International Business Machines Corporation.

Personal Computer AT is a registered trademark of the International Business Machines Corporation.

Section 2, "Interrupts," contains detailed information about how interrupts function across the IBM Personal System/2 and Personal Computer product lines. Exceptions between products are noted.

Section 3, "BIOS Data Areas and ROM Tables," contains detailed information about regular data areas, extended data areas, and ROM tables for system and adapter BIOS.

Section 4, "Additional Information," contains information about sharing interrupts in IBM Personal System/2 and Personal Computer products. It also contains information about adapter ROM calls, video compatibility, multitasking provisions, system identification bytes, keyboard keys, and scan code/character code combinations.

**Supplements** is reserved for additional BIOS interface information. A table of contents page is provided to record the supplements that you add. Supplements to this technical reference will be offered for sale as additional BIOS interface information becomes available.

System-specific hardware and software interface information for IBM systems and for IBM diskette drives, fixed disk drives, adapters, and other options is contained in separate technical reference publications.

---

# Contents - BIOS

<b>Section 1. Introduction to BIOS</b> .....	<b>1-1</b>
Interrupts .....	1-3
Parameter Passing .....	1-3
Data Areas and ROM Tables .....	1-4
BIOS Level Determination .....	1-4
System Groups .....	1-5
<b>Section 2. Interrupts</b> .....	<b>2-1</b>
Interrupt 02H - Nonmaskable Interrupt (NMI) .....	2-4
Interrupt 05H - Print Screen .....	2-6
Interrupt 08H - System Timer .....	2-7
Interrupt 09H - Keyboard .....	2-8
Interrupt 10H - Video .....	2-10
Interrupt 11H - Equipment Determination .....	2-46
Interrupt 12H - Memory Size Determination .....	2-47
Interrupt 13H - Diskette .....	2-48
Interrupt 13H - Fixed Disk .....	2-58
Interrupt 14H - Asynchronous Communications .....	2-70
Interrupt 15H - System Services .....	2-74
Interrupt 16H - Keyboard .....	2-104
Interrupt 17H - Printer .....	2-111
Interrupt 19H - Bootstrap Loader .....	2-114
Interrupt 1AH - System-Timer and Real-Time Clock Services ..	2-115
Interrupt 70H - Real-Time Clock Interrupt .....	2-122
<b>Section 3. Data Areas and ROM Tables</b> .....	<b>3-1</b>
BIOS Data Area .....	3-3
Extended BIOS Data Area .....	3-15
ROM Tables .....	3-16
Fixed Disk Drive Parameter Table .....	3-16
Diskette Drive Parameter Table .....	3-23
<b>Section 4. Additional Information</b> .....	<b>4-1</b>
Interrupt Sharing .....	4-3
Precautions .....	4-3
Interrupt Request (IRQn) Reset .....	4-4
Interrupt-Sharing Software Requirements .....	4-4
Interrupt-Sharing Chaining Structure and Signature .....	4-6
ROM Considerations .....	4-7
Implementation Information .....	4-7

Adapter ROM .....	4-12
Video Function Compatibility .....	4-14
Video Presence Test .....	4-14
Video Mode Switching .....	4-15
Multitasking Provisions .....	4-16
System Identification .....	4-18
Application Guidelines .....	4-19
Hardware Interrupts .....	4-19
Programming Considerations .....	4-21
BIOS and Operating System Function Calls .....	4-21
Scan Code/Character Code Combinations .....	4-24
<b>Glossary .....</b>	<b>X-1</b>
<b>Index .....</b>	<b>X-13</b>

---

## Figures - BIOS

2-1.	Interrupts	2-3
2-2.	INT 10H - Video Functions	2-10
2-3.	Video Modes	2-11
2-4.	Hardware Specific Video Mode Characteristics	2-11
2-5.	PC Convertible Display Types	2-38
2-6.	INT 13H - Diskette Functions	2-48
2-7.	INT 13H - Fixed Disk Functions	2-58
2-8.	INT 14H - Asynchronous Communications Functions	2-70
2-9.	INT 15H - System Services Functions	2-74
2-10.	Block Move Global Descriptor Table	2-88
2-11.	Global Descriptor Table	2-91
2-12.	INT 16H - Keyboard Functions	2-104
2-13.	INT 17H - Printer Functions	2-111
2-14.	INT 1AH - System-Timer and Real-Time Clock Services	2-115
3-1.	RS-232-C Port Base Address Data Area	3-3
3-2.	Printer Port Base Address Data Area	3-3
3-3.	System Equipment Data Area	3-4
3-4.	Miscellaneous Data Area 1	3-4
3-5.	Memory Size Data Area	3-4
3-6.	Keyboard Data Area 1	3-5
3-7.	Diskette Drive Data Area	3-6
3-8.	Video Control Data Area 1	3-7
3-9.	System Data Area 1	3-7
3-10.	System-Timer Data Area	3-7
3-11.	System Data Area 2	3-8
3-12.	Fixed Disk Drive Data Area	3-8
3-13.	Printer Time-Out Value Data Area	3-9
3-14.	RS-232-C Time-Out Value Data Area	3-9
3-15.	Keyboard Data Area 2	3-9
3-16.	Video Control Data Area 2	3-9
3-17.	Diskette Drive/Fixed Disk Drive Control Data Area	3-10
3-18.	Keyboard Data Area 3	3-11
3-19.	Real-Time Clock Data Area	3-11
3-20.	Save Pointer Data Area	3-12
3-21.	Secondary Save Pointer Data Area	3-13
3-22.	Miscellaneous Data Area 2	3-15
3-23.	Fixed Disk Drive Parameter Table Definition	3-16
3-24.	Fixed Disk Drive Parameters (AT and Personal System/2 Products)	3-17

3-25.	Fixed Disk Drive Parameter Table (Personal System/2 Products Except Model 30) .....	3-18
3-26.	Fixed Disk Drive Drive Parameter Table 00 (PC XT BIOS Dated 11/10/82) .....	3-19
3-27.	Fixed Disk Drive Parameter Table 01 (PC XT BIOS Dated 11/10/82) .....	3-19
3-28.	Fixed Disk Drive Parameter Table 02 (PC XT BIOS Dated 11/10/82) .....	3-20
3-29.	Fixed Disk Drive Parameter Table 03 (PC XT BIOS Dated 11/10/82) .....	3-20
3-30.	Fixed Disk Drive Parameter Table 00 - Type 1 (PC XT BIOS Dated 1/08/86) .....	3-21
3-31.	Fixed Disk Drive Parameter Table 01 - Type 16 (PC XT BIOS Dated 1/08/86) .....	3-21
3-32.	Fixed Disk Drive Parameter Table 02 - Type 2 (PC XT BIOS Dated 1/08/86) .....	3-22
3-33.	Fixed Disk Drive Parameter Table 03 - Type 13 (PC XT BIOS Dated 1/08/86) .....	3-22
3-34.	Diskette Drive Parameter Table .....	3-23
4-1.	System Identification .....	4-18
4-2.	Write and Format Head Settle Time .....	4-22
4-3.	Keyboard Keystrokes .....	4-24
4-4.	Shift Keyboard Keystrokes .....	4-26
4-5.	Ctrl Keyboard Keystrokes .....	4-28
4-6.	Alt Keyboard Keystrokes .....	4-31

---

# Section 1. Introduction to BIOS

Interrupts .....	1-3
Parameter Passing .....	1-3
Data Areas and ROM Tables .....	1-4
BIOS Level Determination .....	1-4
System Groups .....	1-5

## **Notes:**

The Basic Input/Output System (BIOS) for IBM Personal System/2 and Personal Computer products is a software interface or "layer" that isolates operating systems and application programs from specific hardware devices. BIOS routines allow assembly language programmers to perform block and character-level operations without concern for device addresses or hardware operating characteristics. The BIOS also provides system services such as time-of-day and memory size determination.

Operating systems and application programs should make functional requests to BIOS rather than directly manipulating I/O ports and control words of the hardware. Hardware design and timing changes then become less critical, and software compatibility across systems and features is enhanced.

---

## Interrupts

BIOS is accessed by software interrupts; each BIOS entry point is available through its own interrupt. The AH register, where appropriate, indicates the specific routine within the overall interrupt function that is being executed.

Software interrupts INT 10H through INT 1AH each access different BIOS routines. For example, INT 12H invokes the BIOS routine for determining memory size and returns the value to the caller.

See Section 2, "Interrupts," for additional information.

---

## Parameter Passing

All parameters passed to and from the BIOS routines go through the microprocessor registers. Each BIOS interrupt routine indicates the registers used on the call and the return. In general, if a BIOS routine has several possible functions, (AH) is used to select the desired function. For example, to set the time, 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, the following code is required:

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

The BIOS interrupt handlers save all registers except (AX), the flags, and those registers that return a value to the caller. In some cases other registers are modified. See Section 2, "Interrupts," for additional information.

All parameters are 1-based (that is, the count starts with 1, not 0), unless noted as 0-based.

---

## Data Areas and ROM Tables

Data areas are the memory locations allocated specifically to system BIOS and adapter BIOS to use as work areas. Read-only memory (ROM) tables are used by BIOS to define the characteristics of the hardware devices supported by a particular system BIOS or adapter BIOS.

See Section 3, "Data Areas and ROM Tables," for additional information.

---

## BIOS Level Determination

The BIOS is contained in ROM modules located on the system boards of Personal System/2 and Personal Computer products. It is also contained in ROM modules on some optional features (usually adapters) to provide device-level control of the features.

The BIOS has been amended several times since its inception. All BIOS versions are dated. In this technical reference, BIOS version dates are used when necessary to indicate interface differences in similar systems.

To determine the BIOS version date, run the following BASIC program. The date that is displayed is the version date of the BIOS for that system:

```
10 DEF SEG=&HF000
20 FOR X=&HFFF5 TO &HFFFC
30 PRINT CHR$(PEEK(X));
40 NEXT
RUN
```

See “System Identification” on page 4-18 for a list of IBM products and their BIOS version dates. To access this information, see INT 15H, “(AH) = C0H Return System Configuration Parameters” on page 2-94.

---

## System Groups

In this technical reference, IBM systems are categorized into groups with similar BIOS interfaces. These groups are referred to with any exceptions noted. The groups with similar interfaces include:

- Personal System/2 products - all models
- Personal Computer XT products - includes *Portable* Personal Computer
- Personal Computer AT products - all models.

**Important:** Information added to the **Supplements** area of this technical reference may have new information about subjects covered in other parts of this technical reference. Refer to the supplements for information that could affect your hardware or software development decisions.

## Notes:

---

## Section 2. Interrupts

Interrupt 02H - Nonmaskable Interrupt (NMI) .....	2-4
Interrupt 05H - Print Screen .....	2-6
Interrupt 08H - System Timer .....	2-7
Interrupt 09H - Keyboard .....	2-8
Interrupt 10H - Video .....	2-10
Interrupt 11H - Equipment Determination .....	2-46
Interrupt 12H - Memory Size Determination .....	2-47
Interrupt 13H - Diskette .....	2-48
Interrupt 13H - Fixed Disk .....	2-58
Interrupt 14H - Asynchronous Communications .....	2-70
Interrupt 15H - System Services .....	2-74
Interrupt 16H - Keyboard .....	2-104
Interrupt 17H - Printer .....	2-111
Interrupt 19H - Bootstrap Loader .....	2-114
Interrupt 1AH - System-Timer and Real-Time Clock Services ..	2-115
Interrupt 70H - Real-Time Clock Interrupt .....	2-122

## Notes:

The following figure lists each interrupt, its function, and, where applicable, the location of a more detailed description of the interrupt:

<b>Interrupt Number (Hex)</b>	<b>Interrupt Function</b>
00	Divide by 0
01	Single Step
02	Nonmaskable (NMI) (See page 2-4)
03	Breakpoint
04	Overflow
05	Print Screen (See page 2-6)
06 to 07	– Reserved –
08	System Timer (See page 2-7)
09	Keyboard (See page 2-8)
0A to 0D	– Reserved –
0E	Diskette (See INT 13H on page 2-48)
0F	– Reserved –
10	Video (See page 2-10)
11	Equipment Determination (See page 2-46)
12	Memory Size Determination (See page 2-47)
13	Fixed Disk/Diskette (See pages 2-48 and 2-58)
14	Asynchronous Communications (See page 2-70)
15	System Services (See page 2-74)
16	Keyboard (See page 2-104)
17	Printer (See page 2-111)
18	Resident BASIC
19	Bootstrap Loader (See page 2-114)
1A	System Timer and Real-Time Clock Services (See page 2-115)
1B	Keyboard Break (See INT 09H on page 2-8)
1C	User Timer Tick (See INT 08H on page 2-7)
1D	Video Parameters
1E	Diskette Parameters (See “Diskette Drive Parameter Table” on page 3-23)
1F	Video Graphics Characters
20 to 3F	– Reserved for Disk Operating System (DOS) –
40	Diskette BIOS Revector
41	Fixed Disk Parameters (See INT 13H on page 2-58)
42 to 45	– Reserved –
46	Fixed Disk Parameters (See INT 13H on page 2-58)
47 to 49	– Reserved –
4A	User Alarm (See INT 08H on page 2-7 and INT 70H on page 2-122)
4B to 5F	– Reserved –
60 to 67	– Reserved for User Program Interrupts –
68 to 6F	– Reserved –
70	Real-Time Clock (See page 2-122)
71 to 74	– Reserved –
75	Redirect to NMI Interrupt (See INT 02H on page 2-4)
76 to 7F	– Reserved –
80 to 85	– Reserved for BASIC –
86 to F0	Used by BASIC Interpreter When Running BASIC
F1 to FF	– Reserved for User Program Interrupts –

Figure 2-1. Interrupts

---

## Interrupt 02H - Nonmaskable Interrupt (NMI)

For PCjr the nonmaskable interrupt (NMI) is attached to the keyboard interrupt.

For PC, PC XT, AT®, and Personal System/2 Model 30, this interrupt handler displays PARITY CHECK 1 indicating a parity error occurred on the system board, or PARITY CHECK 2 indicating a parity error occurred on the I/O channel (assumes I/O channel memory). This interrupt handler attempts to find the storage location containing the bad parity, and if it is found, the segment address is displayed. If no parity error is found, ????? appears in place of the address, indicating an intermittent read problem.

For Personal System/2 products except Model 30, the above paragraph applies except PARITY CHECK 1 and PARITY CHECK 2 are replaced by error codes, **110** and **111**, respectively. In addition, the NMI detects two other errors. The error codes are as follows:

- 110 System Board Memory Failure
- 111 I/O Channel Check Activated (assumes I/O channel memory)
- 112 Watchdog Time-Out
- 113 Direct Memory Access (DMA) Bus Time-Out

When the Watchdog Time-Out is enabled and a missing timer interrupt (IRQ 0) is detected, the system generates the NMI. If this occurs the NMI interrupt handler displays **112**, indicating an expected timer interrupt was missed. Also, when a DMA-driven device uses the bus longer than the allowed 7.8 microseconds, the Central Arbitration Control Point generates the NMI and **113** is displayed, indicating a DMA bus time-out has occurred.

When an NMI occurs, the Central Arbitration Control Point is implicitly disabled. The NMI interrupt handler explicitly reenables the Central Arbitration Control Point by outputting a 00H to port 90H.

---

AT is a registered trademark of the International Business Machines Corporation

For PC Convertible, the NMI is attached to the keyboard, the diskette, the real-time clock, and the system suspend interrupts, and is activated upon the I/O channel check.

**Notes:**

1. An 8087 math coprocessor error on 8088- or 8086-based systems drives the NMI of the 8088 or 8086, respectively.
2. An 80287 or 80387 math coprocessor error on 80286- or 80386-based systems drives the IRQ 13 line. The IRQ 13 interrupt handler issues a software INT 02H to be compatible with software that expects the NMI to occur.
3. For all systems, the math coprocessor application that points the NMI vector to itself must be sensitive to NMI errors. If the NMI occurs due to an NMI error, control should be transferred to the system NMI handler.

---

## Interrupt 05H - Print Screen

This interrupt handler prints the screen to printer 1. When INT 05H is issued, the cursor position is saved and is restored upon completion of the interrupt. INT 05H runs with interrupts enabled. Additional print screen requests are ignored when a print screen is already in progress. An initial status error from the printer ends the print request. Data area address 50:00 contains the status of the print screen. The supported status values for 50:00 are as follows:

00	Print Screen not called or, on return, operation successfully completed
01	Print Screen in progress, ignore request
FF	Error encountered during printing

For PC Convertible, an initial status error ends the print request and also sounds a "beep." The Ctrl-Break sequence ends the print screen.

---

## Interrupt 08H - System Timer

This interrupt handler controls the timer interrupt from channel 0 of the system 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 at data area address 40:6C (timer counter) since power-on that may be used to establish time of day. After 24 hours of operation, 40:70 (timer overflow) is increased (made non 0).
- Decrements 40:40 (motor off counter of the diskette drive) and, when the count reaches 0, turns the diskette drive motor off, and resets the motor running flags in 40:3F (motor status).
- Calls a user routine through software interrupt 1CH every timer tick.

For PC Convertible, this interrupt handler calls a user routine through software interrupt 4AH when an alarm interrupt occurs.

---

## Interrupt 09H - Keyboard

This interrupt handler is issued upon the make or break of every keystroke.

For ASCII keys, when a make code is read from port 60H, the character code and scan code are placed in the 32-byte keyboard buffer that begins at data area address 40:1E, at the address pointed to by 40:1C (keyboard buffer tail pointer). The keyboard buffer tail pointer is then increased by 2, unless it extends past the end of the buffer. In this case it is reinitialized to the start of the buffer.

For every Ctrl, Alt, or Shift key make or break, the BIOS data areas 40:17 and 40:18 (keyboard control) and 40:96 (keyboard mode state and type flags) are updated.

The Ctrl-Alt-Del sequence causes the handler to set 40:72 (reset flag) to hex 1234 (bypass memory test), then jump to the power-on self-test (POST). The POST checks 40:72 (reset flag) and does not retest memory if it finds hex 1234. For PC Convertible, instead of a jump to POST, a processor reset is done, causing POST to execute.

The Pause key sequence causes the handler to loop until a valid ASCII keystroke is pressed. The PC Convertible issues INT 15H, (AH) = 41H (Wait on External Event) to wait for a valid ASCII keystroke.

The print screen key sequence issues an INT 05H (Print Screen).

The Ctrl-Break sequence issues an INT 1BH (Control Break).

For PC XT BIOS dated 1/10/86 and after, AT, PC XT Model 286, PC Convertible, and Personal System/2 products, System Request causes the handler to issue an INT 15H, (AH) = 85H (System Request Key Pressed) to inform the system of a System Request key make or break operation. Also, the keyboard interrupt issues an INT 15H, (AH) = 91H (Interrupt Complete) with (AL) = 02H (Type = Keyboard), indicating that a keystroke is available.

For AT BIOS dated 6/10/85 and after, PC XT Model 286, PC Convertible, and Personal System/2 products, INT 15H, (AH) = 4FH (Keyboard Intercept), is issued after reading the scan code from port 60H. This allows the system to replace or absorb the scan code. End of Interrupt (EOI) processing is done upon return.

---

## Interrupt 10H - Video

The following is a summary of the video functions of INT 10H:

(AH) = 00H	– Set Mode
(AH) = 01H	– Set Cursor Type
(AH) = 02H	– Set Cursor Position
(AH) = 03H	– Read Cursor Position
(AH) = 04H	– Read Light Pen Position
(AH) = 05H	– Select Active Display Page
(AH) = 06H	– Scroll Active Page Up
(AH) = 07H	– Scroll Active Page Down
(AH) = 08H	– Read Attribute/Character at Current Cursor Position
(AH) = 09H	– Write Attribute/Character at Current Cursor Position
(AH) = 0AH	– Write Character at Current Cursor Position
(AH) = 0BH	– Set Color Palette
(AH) = 0CH	– Write Dot
(AH) = 0DH	– Read Dot
(AH) = 0EH	– Write Teletype to Active Page
(AH) = 0FH	– Read Current Video State
(AH) = 10H	– Set Palette Registers
(AH) = 11H	– Character Generator
(AH) = 12H	– Alternate Select
(AH) = 13H	– Write String
(AH) = 14H	– Load LCD Character Font/Set LCD High-Intensity Substitute
(AH) = 15H	– Return Physical Display Parameters for Active Display
(AH) = 16H to 19H	– Reserved
(AH) = 1AH	– Read/Write Display Combination Code
(AH) = 1BH	– Return Functionality/State Information
(AH) = 1CH	– Save/Restore Video State
(AH) = 1DH to FFH	– Reserved

Figure 2-2. INT 10H - Video Functions

## (AH) = 00H - Set Mode

(AL) - Requested video mode

The following table describes the supported video modes:

Mode (Hex)	Type	Maximum Colors	Alpha Format	Buffer Start
0, 1	A/N	16	40x25	B8000
2, 3	A/N	16	80x25	B8000
4, 5	APA	4	40x25	B8000
6	APA	2	80x25	B8000
7	A/N	Mono	80x25	B0000
8	APA	16	20x25	B0000
9	APA	16	40x25	B0000
A	APA	4	80x25	B0000
B, C	—Reserved—			
D	APA	16	40x25	A0000
E	APA	16	80x25	A0000
F	APA	Mono	80x25	A0000
10	APA	16	80x25	A0000
11	APA	2	80x30	A0000
12	APA	16	80x30	A0000
13	APA	256	40x25	A0000

APA - All Points Addressable (Graphics)  
A/N - Alphanumeric (Text)

Figure 2-3. Video Modes

The following table lists hardware specific video mode characteristics:

Mode (Hex)	Display Size	Box Size	Supporting IBM Products	Maximum Pages
0, 1	320x200	8x8	PCjr, Color/Graphics Monitor Adapter (CGA), Enhanced Graphics Adapter (EGA), PC Convertible, and Personal System/2 Products Except Model 30	8
	320x350	8x14	EGA, and Personal System/2 Products Except Model 30	8
	320x400	8x16	Personal System/2 Model 30	8
	360x400	9x16	Personal System/2 Products Except Model 30	8

Figure 2-4 (Part 1 of 2). Hardware Specific Video Mode Characteristics

Mode (Hex)	Display Size	Box Size	Supporting IBM Products	Maximum Pages
2, 3	640x200	8x8	PCjr, CGA, and PC Convertible	4
	640x200	8x8	EGA, and Personal System/2 Products Except Model 30	8
	640x350	8x14	EGA, and Personal System/2 Products Except Model 30	8
	640x400	8x16	Personal System/2 Model 30	8
	720x400	9x16	Personal System/2 Products Except Model 30	8
4, 5	320x200	8x8	PCjr, CGA, EGA, and Personal System/2 Products	1
6	640x200	8x8	PCjr, CGA, EGA, and Personal System/2 Products	1
7	720x350	9x14	Monochrome Display and Printer Adapter (MDPA) and PC Convertible	1
	720x350	9x14	EGA and Personal System/2 Products Except Model 30	8
	720x400	9x16	Personal System/2 Products Except Model 30	8
	640x200	8x8	PC Convertible	4
8	160x200	8x8	PCjr	1
9	320x200	8x8	PCjr	1
A	640x200	8x8	PCjr	1
B, C	—Reserved—			
D	320x200	8x8	EGA and Personal System/2 Products Except Model 30	8
E	640x200	8x8	EGA and Personal System/2 Products Except Model 30	4
F,10	640x350	8x14	EGA and Personal System/2 Products Except Model 30	2
11	640x480	8x16	Personal System/2 Products	1
12	640x480	8x16	Personal System/2 Products Except Model 30	1
13	320x200	8x8	Personal System/2 Products	1

Figure 2-4 (Part 2 of 2). Hardware Specific Video Mode Characteristics

## Notes:

### 1. PCjr and IBM Color/Graphics Monitor Adapter (CGA):

- a. The cursor is not displayed in graphics (APA) modes.
- b. Modes 0, 2, and 5 are identical to modes 1, 3, and 4 except color burst is not enabled. Color burst on enables color information on composite displays. Color burst off disables color information on composite displays. RGB displays are not affected by the state of color burst.
- c. For PCjr during mode set, if bit 7 of (AL) is set, the video buffer is not cleared.

### 2. IBM Enhanced Graphics Adapter (EGA):

- a. The cursor is not displayed in graphics (APA) modes.
- b. Modes 0, 2, and 5 are identical to modes 1, 3, and 4 except color burst is not enabled. Color burst on enables color information on composite displays. Color burst off disables color information on composite displays. RGB displays are not affected by the state of color burst.
- c. The power-on default mode is based on switch settings on the adapter.
- d. During mode set, if bit 7 of (AL) is set, the video buffer is not cleared.

See BIOS data area address 40:A8 on page 3-12 for save pointer dynamic overrides.

### 3. PC Convertible:

- a. The cursor is not displayed in graphics (APA) modes.
- b. Modes 0, 2, and 5 are identical to modes 1, 3, and 4 except color burst is not enabled. Color burst on enables color information on composite displays. Color burst off disables color information on composite displays. RGB displays are not affected by the state of color burst.
- c. The power-on default mode for color/graphics mode is 2.
- d. The power-on default mode for monochrome mode is 7.
- e. During mode set, if bit 7 of (AL) is set, the video buffer is not cleared.
- f. Mode 7 (640x200) is used for a liquid crystal display (LCD) as monochrome.
- g. Mode 7 (720x350) is used for a monochrome display.

4. Personal System/2 Model 30:

- a. The cursor is not displayed in graphics (APA) modes.
- b. Modes 0, 2, and 5 are identical to modes 1, 3, and 4.
- c. The power-on default mode is 3.
- d. During mode set, if bit 7 of (AL) is set, the video buffer is not cleared.
- e. For all modes except mode 13H, the first 16 color registers are initialized and the values in the remaining 240 color registers are undefined.

See BIOS data area address 40:A8 on page 3-12 for save pointer dynamic overrides.

5. Personal System/2 products except Model 30:

- a. The cursor is not displayed in graphics (APA) modes.
- b. Modes 0, 2, and 5 are identical to modes 1, 3, and 4.
- c. The power-on default mode with a color display attached is 3.
- d. The power-on default mode with a monochrome display attached is 7.
- e. During mode set, if bit 7 of (AL) is set, the video buffer is not cleared.
- f. For all modes except mode 13, the first 64 color registers are initialized and the values in the remaining 192 color registers are undefined.
- g. Refer to (AH) = 12H, (BL) = 30H to select alpha mode scan lines (200, 350 or 400.)

See BIOS data area address 40:A8 on page 3-12 for save pointer dynamic overrides.

**(AH) = 01H - Set Cursor Type**

(CH) - Top line for cursor (bits 4 to 0)  
(Hardware causes blinking cursor;  
setting bit 6 or 5 causes erratic  
blinking or no cursor)

(CL) - Bottom line for cursor (bits 4 to 0)

**Notes:**

1. The BIOS maintains only one cursor type for all video pages.
2. For Personal System/2 Model 30, before writing to the hardware video ports, (CH) is multiplied by 2, and (CL) is multiplied by 2 and increased by 1.

**(AH) = 02H - Set Cursor Position**

(DH,DL) - Row, column (0,0 is upper left)

(BH) - Page number (0-based), see Figure 2-4 on page 2-11 for maximum pages

**(AH) = 03H - Read Cursor Position**

(BH) - Page number (0-based), see Figure 2-4 on page 2-11 for maximum pages

On Return:

(DH,DL) - Row, column of current cursor for requested page

(CH,CL) - Cursor type currently set

**(AH) = 04H - Read Light Pen Position**

For PC Convertible and Personal System/2 products:

On Return:

(AH) = 00H - Light pen is not supported

(BX, CX, DX) are altered on return

For all others:

On Return:

(AH) = 00H - Light pen switch not activated

(BX, CX, DX) are altered on return

(AH) = 01H - Valid light pen value in registers

(DH,DL) - Row, column of character

(CH) - Raster line (0 to 199)

(CX) - Raster line (0 to nnn) new graphics modes

(BX) - PEL column (0 to 319,639)

## **(AH) = 05H - Select Active Display Page**

For PCjr:

(AL) = 80H - Read cathode ray tube (CRT) and microprocessor page registers

(AL) = 81H - Set microprocessor page register  
(BL) - Microprocessor page register

(AL) = 82H - Set CRT page register  
(BH) - CRT page register

(AL) = 83H - Set microprocessor and CRT page registers  
(BL) - Microprocessor page register  
(BH) - CRT page register

On Return for all:

(BH) - CRT page register

(BL) - Microprocessor page register

For all others:

(AL) - New page number (0-based), see Figure 2-4 on page 2-11 for maximum pages

## **(AH) = 06H - Scroll Active Page Up**

(AL) - Number of lines blanked at bottom of window  
= 00H - Blank entire window

(CH,CL) - Row, column of upper left corner of scroll

(DH,DL) - Row, column of lower right corner of scroll

(BH) - Attribute to use on blank line

## **(AH) = 07H - Scroll Active Page Down**

(AL) - Number of input lines blanked at top of window  
= 00H - Blank entire window

(CH,CL) - Row, column of upper left corner of scroll

(DH,DL) - Row, column of lower right corner of scroll

(BH) - Attribute to use on blank line

## **(AH) = 08H - Read Attribute/Character at Current Cursor Position**

(BH) - Page number (0-based), see Figure 2-4 on page 2-11 for maximum pages

On Return:

(AL) - Character read

(AH) - Attribute of character read (alpha modes only)

## **(AH) = 09H - Write Attribute/Character at Current Cursor Position**

For the read/write character interface while in graphics modes 4, 5, and 6, the characters are formed from a character generator maintained in the system ROM that contains only the first 128 characters. To read or write the second 128 characters, initialize the pointer at INT 1FH (location 0007CH) to point to the 1Kb table containing the code points for the second 128 characters (128-255). For all other graphics modes, 256 graphics characters are supplied in the system ROM.

For the write character interface while in graphics mode, the character count contained in (CX) produces valid results for characters on the same row only. Continuation to succeeding rows produces invalid results.

(BH) - Page number (0-based), see Figure 2-4 on page 2-11 for maximum pages

(CX) - Count of characters to write

(AL) - Character to write

(BL) - Attribute of character (alpha)/color of character (graphics)

### **Notes:**

1. Functions (AH) = 09H and (AH) = 0AH are similar. Use (AH) = 09H for graphics modes.
2. For graphics modes, if bit 7 of (BL) = 01H, then color value is exclusively ORed with current video memory (except in mode 13H).
3. For mode 13H, the value passed in (BH) is used as the background color.

## **(AH) = 0AH - Write Character at Current Cursor Position**

- (BH) - Page number (0-based), see Figure 2-4 on page 2-11 for maximum pages
- (CX) - Count of characters to write
- (AL) - Character to write

**Note:** Use (AH) = 09H for graphics modes.

## **(AH) = 0BH - Set Color Palette**

- (BH) - Color ID being set (0 to 1)
  - (BL) - Color value to be used with color ID
- 
- (BH) = 00H - Set background color for 320x200 graphics modes
    - Set border color for alphanumeric modes
    - Set foreground color for 640x200 graphics
  - (BL) = (0 to 31)
- 
- (BH) = 01H - Select palette for 320x200 graphics
  - (BL) = 0 - Green (1)/red (2)/brown (3)
    - = 1 - Cyan (1)/magenta (2)/white (3)

### **Notes:**

1. This interface has meaning for 320x200 graphics only.
2. In 40x25 or 80x25 alpha modes, the value set for palette color 0 indicates the border color to use (0 to 31), where values 16 to 31 select the high-intensity background set.
3. For EGA and Personal System/2 products, when in 640x200 graphics and color ID = 0, the background color is set.

## **(AH) = 0CH - Write Dot**

- (DX) - Row number
- (CX) - Column number
- (AL) - Color value

**Note:** If bit 7 of (AL) = 01H, then the color value is exclusively ORed with the current contents of the dot (except in mode 13H).

**For graphics modes supporting more than one page:**

(BH) - Page number (0-based), see Figure 2-4 on page 2-11 for maximum pages

**(AH) = 0DH - Read Dot**

(DX) - Row number  
(CX) - Column number

**For graphics modes supporting more than one page:**

(BH) - Page number (0-based), see Figure 2-4 on page 2-11 for maximum pages

On Return:  
(AL) returns dot read

**(AH) = 0EH - Write Teletype to Active Page**

(AL) - Character to write  
(BL) - Foreground color in graphics mode

**Notes:**

1. The screen width is controlled by the mode currently set.
2. Carriage Return, Line Feed, Backspace and Bell are treated as commands rather than printable characters.
3. For PC BIOS dated 4/24/81 and 10/19/81, (BH) must be set to the active page.

**(AH) = 0FH - Read Current Video State**

On Return:  
(AL) - Mode currently set  
[see (AH) = 00H for explanation]  
(AH) - Number of character columns on screen  
(BH) - Current active page number (0-based), see Figure 2-4 on page 2-11 for maximum pages

## **(AH) = 10H - Set Palette Registers**

For PCjr, systems with EGA capability, and Personal System/2 products except Model 30:

(AL) = 00H - Set individual palette register  
(BL) - Palette register to set  
(BH) - Value to set

(AL) = 01H - Set overscan register  
(BH) - Value to set

(AL) = 02H - Set all palette registers and overscan  
(ES:DX) - Pointer to 17-byte table  
Bytes 0 to 15 - Palette values  
Byte 16 - Overscan value

(AL) = 03H - Toggle intensify/blinking bit  
(BL) = 00H - Enable intensify  
= 01H - Enable blinking

For Personal System/2 products except Model 30:

(AL) = 04H to 06H - Reserved

(AL) = 07H - Read individual palette register  
(BL) - Palette register to read (range 0 to 15)

On Return:  
(BH) - Value read

(AL) = 08H - Read overscan register

On Return:  
(BH) - Value read

(AL) = 09H - Read all palette registers and overscan  
(ES:DX) - Pointer to 17-byte buffer for return values

On Return:  
(ES:DX) - Pointer to 17-byte table destination  
Bytes 0 to 15 - Palette values  
Byte 16 - Overscan value

- (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) = 11H - Reserved
  
- (AL) = 12H - Set block of color registers
  - (ES:DX) - Pointer to table of color values
    - Table format: red, green, blue, red,  
green, blue
  - (BX) - First color register to set
  - (CX) - Number of color register to set
  
- (AL) = 13H - Select color page (not valid for mode 13H)
  - (BL) = 00H - Select paging mode
    - (BH) - Paging mode
      - = 00H - Selects 4 register blocks of 64 registers
      - = 01H - Selects 16 register blocks of 16 registers
  
  - (BL) = 01H - Select page
    - (BH) - Page number (0-based), see Figure 2-4 on page 2-11 for maximum pages
      - For 64-register block mode:
        - = 00H - Selects first block of 64 color registers
        - = 01H - Selects second block of 64 color registers
        - = 02H - Selects third block of 64 color registers
        - = 03H - Selects fourth block of 64 color registers
      - For 16-register block mode:
        - = 00H - Selects first block of 16 color registers
        - = 01H - Selects second block of 16 color registers
        - .
        - .
        - = 0FH - Selects 16th block of 16 color registers

**Note:** Function (AH) = 00H (Set Mode) defaults to the 64-register block mode, with the first block of 64 color registers active. Only these 64 color registers are initialized during mode set. When using page selection, initialize alternate blocks of the color registers.

(AL) = 14H - Reserved

(AL) = 15H - Read individual color register

(BX) - Color register to read

On Return:

(DH) - Red value read

(CH) - Green value read

(CL) - Blue value read

(AL) = 16H - Reserved

(AL) = 17H - Read block of color registers

(ES:DX) - Pointer to destination table for values  
Table format: red, green, blue, red,  
green, blue

(BX) - First color register to read

(CX) - Number of color registers to read

On Return:

(ES:DX) - Pointer to table of values

(AL) = 18H to 19H - Reserved

(AL) = 1AH - Read color page state

On Return:

(BL) - Current paging mode

(BH) - Current page

**Note:** See (AL) = 13H on page 2-21 for paging modes and page information.

(AL) = 1BH - Sum color values to gray shades

(BX) - First color register to sum

(CX) - Number of color registers to sum

**Note:** This call reads red, green, and blue values found in color registers, performs a weighted sum (30% red + 59% green + 11% blue), then writes the result into each red, green, and blue component of the color register (original data is not retained).

## For Personal System/2 Model 30:

- (AL) = 00H
  - (BX) = 0712H - Color registers set resulting in 8 consistent colors
  
- (AL) = 01H to 02H - Reserved
  
- (AL) = 03H - Toggle intensify/blinking bit
  - (BL) = 00H - Enable intensify
  - = 01H - Enable blinking
  
- (AL) = 04H to 07H - Reserved
  
- (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) = 11H - Reserved
  
- (AL) = 12H - Set block of color registers
  - (ES:DX) - Pointer to table of color values  
          Table format: red, green, blue, red,  
                          green, blue
  
  - (BX) - First color register to set
  - (CX) - Number of color registers to set
  
- (AL) = 13H to 14H - Reserved
  
- (AL) = 15H - Read individual color register
  - (BX) - Color register to read
  
- On Return:
  - (DH) - Red value read
  - (CH) - Green value read
  - (CL) - Blue value to read
  
- (AL) = 16H - Reserved

(AL) = 17H - Read a block of color registers  
(ES:DX) - Pointer to destination table for values  
Table format: red, green, blue, red,  
green, blue

(BX) - First color register to read  
(CX) - Number of color registers to read

On Return:

(ES:DX) - Pointer to table of values

(AL) = 18H to 1AH - Reserved

(AL) = 1BH - Sum color values to gray shades  
(BX) - First color register to sum  
(CX) - Number of color registers to sum

**Note:** This call reads red, green, and blue values found in color registers, performs a weighted sum (30% red + 59% green + 11% blue), then writes result into each red, green, and blue component of the color register (original data is not retained).

For all others no action is performed.

### **(AH) = 11H - Character Generator**

For systems with EGA capability, this call initiates a mode set, completely resetting the video environment but maintaining the regenerator buffer.

(AL) = 00H - User alpha load  
(ES:BP) - Pointer to user table  
(CX) - Count to store  
(DX) - Character offset into table  
(BL) - Block to load  
(BH) - Number of bytes per character

(AL) = 01H - ROM monochrome set  
(BL) - Block to load

(AL) = 02H - ROM 8x8 double dot  
(BL) - Block to load

- (AL) = 03H - Set block specifier (valid in alpha modes)
- (BL) - Character generator block selects
  - Character attribute byte, bit 3 = 0:
    - (BL) bits 1, 0 select a block from blocks 0 to 3
  - Character attribute byte, bit 3 = 1:
    - (BL) bits 3, 2 select a block from blocks 0 to 3

For example:

- To set a 256 character set using block 3, set (BL) = 0FH; this selects a single block. Character attribute bit 3 turns foreground intensity on or off.
- To specify a 512 character set as active using blocks 0 and 3, set (BL) = 0CH; this selects block 0 when character attribute bit 3 = 0, and block 3 when character attribute bit 3 = 1.

If bits (1, 0) and bits (3, 2) are the same, only one block is selected and bit 3 of the attribute byte turns the foreground intensity on or off.

When 512 characters are active, a function call with (AX) = 1000H and (BX) = 0712H is recommended to set the color planes with eight consistent colors.

Register values, (AL) = 10H, 11H, and 12H, are similar to (AL) = 00H, 01H, and 02H, respectively, with the following exceptions:

1. Page 0 must be active.
2. Points (bytes per character) are recalculated.
3. Rows are calculated as follows:

$$\text{INT} [(200 \text{ or } 350) / \text{points}] - 1$$

4. The length of the regenerative buffer is calculated as follows:

$$(\text{Number of rows on screen}) \times (\text{Number of columns on screen}) \times 2$$

5. The CRTC registers are reprogrammed as follows:

R09H = Points - 1	Maximum scan line
R0AH = Points - 2	Cursor start
R0BH = Points - 1	Cursor end
R12H = [(Number of rows on screen) x Points] - 1	Vertical display end
R14H = Points - 1	Underline location.
(Done in mode 7H only)	

**Note:** The preceding register calculations must be close to the original table values or the results may be unpredictable.

- (AL) = 10H - User alpha load
  - (ES:BP) - Pointer to user table
  - (CX) - Count to store
  - (DX) - Character offset into table
  - (BL) - Block to load
  - (BH) - Number of bytes per character
  
- (AL) = 11H - ROM monochrome set
  - (BL) - Block to load
  
- (AL) = 12H - ROM 8x8 double dot
  - (BL) - Block to load
  
- (AL) = 20H - Set user graphics characters pointer at INT 1FH
  - (ES:BP) - Pointer to user table
  
- (AL) = 21H - Set user graphics characters pointer at INT 43H
  - (ES:BP) - Pointer to user table
  - (CX) - Points (bytes per character)
  - (BL) - Row specifier
    - = 00H - User
    - (DL) - Rows
    - = 01H - 14 (0EH)
    - = 02H - 25 (19H)
    - = 03H - 43 (2BH)
  
- (AL) = 22H - ROM 8x14 Set
  - (BL) - Row specifier
  
- (AL) = 23H - ROM 8x8 double dot
  - (BL) - Row specifier

**Note:** (AL) = 10H, 11H, 12H, 20H, 21H, 22H, or 23H should be called only immediately after a mode set is issued, or the results may not be predictable.

- (AL) = 30H - Information
  - (BH) - Font pointer
    - = 00H - Return current INT 1FH pointer
    - = 01H - Return current INT 44H pointer
    - = 02H - Return ROM 8x14 font pointer
    - = 03H - Return ROM double dot pointer
    - = 04H - Return ROM double dot pointer (top)
    - = 05H - Return ROM alpha alternate 9x14

On Return:

(CX) - Points  
(DL) - Rows  
(ES:BP) - Pointer to table

### For Personal System/2 products except Model 30:

(AL) = 00H - User alpha load  
(ES:BP) - Pointer to user table  
(CX) - Count to store  
(DX) - Character offset into table  
(BL) - Block to load  
(BH) - Number of bytes per character

(AL) = 01H - ROM 8x14 font  
(BL) - Block to load

(AL) = 02H - ROM 8x8 double dot font  
(BL) - Block to load

(AL) = 03H - Set block specifier (valid in alpha modes)  
(BL) - Character generator block selects  
Character attribute byte bit 3 = 0:  
(BL) bits 4, 1, 0 select a block from blocks 0 to 7  
Character attribute byte bit 3 = 1:  
(BL) bits 5, 3, 2 select a block from blocks 0 to 7

For example:

- To set a 256-character set using block 6, set (BL) = 03AH; this selects a single block. Character attribute bit 3 turns foreground intensity on or off.
- To specify a 512-character set as active using blocks 0 and 6, set (BL) = 028H; this selects block 0 active when character attribute bit 3 = 0, and block 6 active when character attribute bit 3 = 1.

If bits (4, 1, 0) and bits (5, 3, 2) are the same, then only one block is selected and bit 3 of the attribute byte turns foreground intensity on or off.

When 512 characters are active, a function call with (AX) = 1000H and (BX) = 0712H is recommended to set color planes with eight consistent colors.

(AL) = 04H - ROM 8x16 Font  
(BL) - Block to load

Register values (AL) = 10H, 11H, 12H, and 14H, are similar to (AL) = 00H, 01H, 02H, and 04H, respectively, with the following exceptions:

1. Page 0 is active.
2. Points (bytes per character) are recalculated.
3. Rows are calculated as follows:

$$\text{INT}[(200, 350, \text{ or } 400) / \text{points}] - 1$$

4. The length of the regenerative buffer is calculated as follows:

$$(\text{Number of rows on screen}) \times (\text{Number of columns on screen}) \times 2$$

5. The CRTC registers are reprogrammed as follows:

R09H = Points - 1	Maximum scan line
R0AH = Points - 2	Cursor start
R0BH = Points - 1	Cursor end
R12H = Vertical displacement end	
For 350 and 400 scan line modes:	
$[(\text{Number of rows on screen}) \times \text{Points}] - 1$	
For 200 scan line modes:	
$\{[(\text{Number of rows on screen}) \times \text{Points}] \times 2\} - 1$	
R14H = Points - 1	Underline location
(Done in mode 7H only)	

**Note:** The preceding register calculations must be close to the original table values or the results may be unpredictable.

(AL) = 10H - User alpha load  
(ES:BP) - Pointer to user table  
(CX) - Count to store  
(DX) - Character offset into table  
(BL) - Block to load  
(BH) - Number of bytes per character

(AL) = 11H - ROM 8x14 font  
(BL) - Block to load

(AL) = 12H - ROM 8x8 double dot font  
(BL) - Block to load

- (AL) = 14H - ROM 8x16 font
  - (BL) - Block to load
  
- (AL) = 20H - Set user graphics characters pointer at INT 1FH
  - (ES:BP) - Pointer to user table
  
- (AL) = 21H - Set user graphics characters pointer at INT 43H
  - (ES:BP) - Pointer to user table
  - (CX) - Points (bytes per character)
  - (BL) - Row specifier
    - = 00H - User
    - (DL) - Rows
    - = 01H - 14 (0EH)
    - = 02H - 25 (19H)
    - = 03H - 43 (2BH)
  
- (AL) = 22H - ROM 8x14 font
  - (BL) - Row specifier
  
- (AL) = 23H - ROM 8x8 double dot font
  - (BL) - Row specifier
  
- (AL) = 24H - ROM 8x16 font
  - (BL) - Row specifier

**Note:** (AL) = 10H, 11H, 12H, 14H, 20H, 21H, 22H, 23H or 24H should be called only immediately after a mode set is issued, or the results may not be predictable.

- (AL) = 30H - Information
  - (BH) - Font pointer
    - = 00H - Return current INT 1FH pointer
    - = 01H - Return current INT 43H pointer
    - = 02H - Return ROM 8x14 font pointer
    - = 03H - Return ROM 8x8 font pointer
    - = 04H - Return ROM 8x8 font pointer (top)
    - = 05H - Return ROM 9x14 font alternate
    - = 06H - Return ROM 8x16 pointer
    - = 07H - Return ROM 9x16 font alternate

On Return:

- (CX) - Points
- (DL) - Rows (number of character rows on screen - 1)
- (ES:BP) - Pointer to table

## For Personal System/2 Model 30:

(AL) = 00H - User alpha load  
(ES:BP) - Pointer to user table  
(CX) - Count to store  
(DX) - Character offset into table  
(BL) - Block to load  
(BH) = 16 bytes per character for 400 scan lines

**Note:** If (BH) = 14 bytes per character for 400 scan lines, characters are extended to 16 high, by extending the last line of 14-high characters.

(AL) = 01H - Reserved  
[If called, (AL) = 04H executed]

(AL) = 02H - ROM 8x8 double dot font  
(BL) - Block to load

(AL) = 03H - Set block specifier (valid in alpha modes)  
(BL) - Character generator block selects  
Character attribute byte bit 3 = 0:  
    (BL) bits 1, 0 select a block from blocks 0 to 3  
Character attribute byte bit 3 = 1:  
    (BL) bits 3, 2 select a block from blocks 0 to 3

For example:

- To specify a 256-character set active using block 2, set (BL) = 0AH; this selects a single block. Character attribute bit 3 turns foreground intensity on or off.
- To specify a 512-character set active using blocks 0 and 2, set (BL) = 08H; this selects block 0 active when character attribute bit 3 = 0, and block 2 active when character attribute bit 3 = 1.

If bits (1, 0) and bits (3, 2) are the same, then only one block is selected and bit 3 of the attribute byte turns foreground intensity on or off.

When 512 characters are active, a function call with (AX) = 1000H and (BX) = 0712H is recommended to set color registers, resulting in eight consistent colors.

A block specifier command must be issued following any character load command to make the loaded block an active character set.

(AL) = 04H - ROM 8x16 font  
(BL) - Block to load

The following register values are reserved. Calls to (AL) = 10H, 11H, 12H, and 14H are executed as if they were calls to (AL) = 00H, 01H, 02H, and 04H, respectively.

(AL) = 10H - Reserved  
[if called - (AL) = 00H executed]

(AL) = 11H - Reserved  
[if called - (AL) = 01H executed]

(AL) = 12H - Reserved  
[if called - (AL) = 02H executed]

(AL) = 14H - Reserved  
[if called - (AL) = 04H executed]

(AL) = 20H - Set user graphics characters pointer at INT 1FH  
(ES:BP) - Pointer to user table

(AL) = 21H - Set user graphics characters pointer at INT 43H  
(ES:BP) - Pointer to user table  
(CX) - Points (bytes per character)  
(BL) - Row specifier  
= 00H - User  
(DL) - Rows  
= 01H - 14 (0EH)  
= 02H - 25 (19H)  
= 03H - 43 (2BH)

(AL) = 22H - Reserved  
[if called, (AL) = 24H executed]

(AL) = 23H - ROM 8x8 double dot font  
(BL) - Row specifier

(AL) = 24H ROM 8x16 font  
(BL) - Row specifier

**Note:** (AL) = 20H, 21H, 22H, 23H or 24H should be called only immediately after a mode set is issued, or the results may not be predictable.

(AL) = 30H - Information  
(BH) - Font pointer  
= 00H - Return current INT 1FH pointer  
= 01H - Return current INT 43H pointer  
= 02H - Reserved (if called, ROM 8x16 pointer returned)  
= 03H - Return ROM 8x8 font pointer  
= 04H - Return ROM 8x8 font pointer (top)  
= 05H - Reserved  
= 06H - Return ROM 8x16 pointer  
= 07H - Reserved

On Return:

(CX) - Points  
(DL) - Rows (number of character rows on screen - 1)  
(ES:BP) - Pointer to table

For all others no action is performed.

### **(AH) = 12H - Alternate Select**

For systems with EGA capability and Personal System/2 products except Model 30:

(BL) = 10H - Return EGA information  
(BH) = 00H - Color mode in effect (3Dx address range)  
= 01H - Monochrome mode in effect (3Bx address range)  
(BL) - Memory value  
= 00H - 64Kb  
= 01H - 128Kb  
= 02H - 192Kb  
= 03H - 256Kb  
= 04H to FFH - Reserved  
(CH) = Adapter bits  
(CL) = Switch setting

(BL) = 20H - Select alternate print screen routine

## For Personal System/2 products except Model 30:

(BL) = 30H - Select scan lines for alphanumeric modes  
(Takes effect on next mode set)  
(AL) = 0 - 200 scan lines  
= 1 - 350 scan lines  
= 2 - 400 scan lines

On Return:

(AL) = 12H - Function supported

(BL) = 31H - Default palette loading during set mode  
(AH) = 00H  
(AL) = 0 - Enable default palette loading  
= 1 - Disable default palette loading

On Return:

(AL) = 12H - Function supported

**Note:** The EGA 16-palette registers, the overscan register, and the 256 color registers are not altered during any mode set when in the disabled state.

(BL) = 32H - Video  
(AL) = 0 - Enable video  
= 1 - Disable video

On Return:

(AL) = 12H - Function supported

**Note:** The video I/O port and regenerator buffer address decode is enabled/disabled for the display that is currently active.

(BL) = 33H - Summing to gray shades  
(AL) = 0 - Enable summing  
= 1 - Disable summing

On Return:

(AL) = 12H - Function supported

**Note:** When enabled, summing occurs during (AH) = 00H (Set Mode) color register loading and (AH) = 10H (Set Palette Registers).

(BL) = 34H - Cursor emulation  
(AL) = 0 - Enable cursor emulation  
      = 1 - Disable cursor emulation

On Return:  
(AL) = 12H - Function supported

**Note:** When enabled, the requested start/end value passed to (AH) = 01H (Set Cursor Type), is scaled to the current character height. The power-on default is to enable cursor emulation.

For Personal System/2 Model 30:

(BL) = 30H - Reserved

(BL) = 31H - Default palette loading during set mode (AH = 00H)  
(AL) = 0 - Enable default palette loading  
      = 1 - Disable default palette loading (the 256 color registers are not altered during any mode set when disabled)

On Return:  
(AL) = 12H - Function supported

(BL) = 32H - Video (the video I/O address and buffers are enabled/disabled)  
(AL) = 0 - Enable video  
      = 1 - Disable video

On Return:  
(AL) = 12H - Function supported

(BL) = 33H - Summing to gray shades  
(AL) = 0 - Enable summing  
      = 1 - Disable summing

On Return:

(AL) = 12H - Function supported

(BL) = 34H - Reserved

**Note:** When enabled, summing occurs during (AH) = 00H (Set Mode) color register loading, and on (AH) = 10H (Set Palette Registers).

**For Personal System/2 products:**

(BL) = 35H Display switch

(AL) = 00H - Initial adapter video off

(ES:DX) - Pointer to switch state save area of  
128 bytes

= 01H - Initial system board video on

= 02H - Switch off active video

(ES:DX) - Pointer to switch state buffer save area

= 03H - Switch on inactive video

(ES:DX) = Pointer to previously saved switch state  
buffer

On Return for all:

(AL) = 12H - Function supported

This interface allows display switching between a system board video driven display and an adapter video driven display when there is overlap in usage of the BIOS data area and in hardware capabilities.

Display switching requires that a disable function is available for the system board and the adapter video functions [(AH) = 12H, (BL) = 32H].

If there is no conflict between the adapter video and the system board video, both video functions are active in the system and display switching is not required.

If there is conflict between the adapter video and the system board video, the adapter video function is the primary video. The system board video function remains disabled until display switching is enabled.

The following steps initiate display switching:

1. Initial adapter video off, (AL) = 00H
2. Initial system board video on, (AL) = 01H.

The initiate display switching steps are valid only the first time switching is initiated. After the initiation steps, switching between the system board and adapter displays is done through the switch-off active video request, (AL) = 02H and the switch-on inactive video request, (AL) = 03H.

For a switch-off active video request, (AL) = 02H, the currently active video function and display are disabled. The switch state buffer saves the video state information. This state information is required when reactivation of this display is desired through a switch-on inactive video request, (AL) = 03H.

For a switch-on inactive video request, (AL) = 03H, the currently inactive video function and display are enabled. The switch state buffer restores the video state information. This state information was saved on a previous switch-off active video request, (AL) = 02H, for this display.

For Personal System/2 products except Model 30:

(BL) = 36H - Video screen off/on  
(AL) = 1 - Screen off  
      = 0 - Screen on

On Return:  
(AL) = 12H - Function supported

For all others no action is performed.

### **(AH) = 13H - Write String**

For PC XT BIOS dated 1/10/86 and after, AT, EGA, PC Convertible, and Personal System/2 products:

(ES:BP) - Pointer to string to write  
(CX) - Character-only count  
(DX) - Position to begin string, in cursor terms  
(BH) - Page number (0-based), see Figure 2-4 on page 2-11 for maximum pages

- (AL) = 00H
  - (BL) - Attribute
    - String - (Char, char, char, ...); Cursor not moved
- (AL) = 01H
  - (BL) - Attribute
    - String - (Char, char, char, ...); Cursor is moved
- (AL) = 02H
  - String - (Char, attr, char, attr, ...)
  - Cursor not moved, valid for alpha modes only
- (AL) = 03H
  - String - (Char, attr, char, attr, ...)
  - Cursor is moved, valid for alpha modes only

**Note:** Carriage Return, Line Feed, Backspace, and Bell are treated as commands rather than printable characters.

For all others no action is performed.

### **(AH) = 14H - Load LCD Character Font/Set LCD High-Intensity Substitute**

For PC Convertible:

- (AL) = 00H - Load user specified font
  - (ES:DI) - Point to character font within user table where loading starts
  - (CX) - Number of characters to store (1 to 256) value checked
  - (DX) - Character offset into RAM font area
  - (BL) = 00H - Load main font (block 0)
  - = 01H - Load alternate font (block 1)
  - = 02H to FFH - No operation
  - (BH) - Number of bytes per character (1 to 255) value checked
- (AL) = 01H - Load system ROM default font
  - (BL) = 00H - Load main font (block 0)
  - = 01H - Load alternate font (block 1)
  - = 02H to FFH - No operation
- (AL) = 02H - Set mapping of LCD high intensity attribute
  - (BL) = 00H - Ignore high-intensity attribute
  - = 01H - Map high-intensity to reverse image
  - = 02H - Map high-intensity to underscore
  - = 03H - Map high-intensity to select alternate font
  - = 04H to FFH - No operation

(AL) = 03H to FFH - No operation

For all others no action is performed.

**(AH) = 15H - Return Physical Display Parameters for Active Display**

For PC Convertible:

On Return:

- (AX) - Alternate display adapter type
  - = 0 - No alternate adapter
  - = 5140 - LCD
  - = 5153 - CGA type display
  - = 5151 - Monochrome type display

(ES:DI) - Points to table defined as follows:

- Word 1 - Display model number
- Word 2 - Number of vertical PELs per meter
- Word 3 - Number of horizontal PELs per meter
- Word 4 - Total number of vertical PELs
- Word 5 - Total number of horizontal PELs
- Word 6 - Horizontal PEL separation in micrometers  
(center to center)
- Word 7 - Vertical PEL separation in micrometers  
(center to center)

The PC Convertible has defined the following display types and tables:

Word	Monochrome	CGA	LCD as CGA	LCD (Monochrome)
1	5151H	5153H	5140H	5140H
2	0	0498H	08E1H	0
3	0	0A15H	0987H	0
4	0	00C8H	00C8H	0
5	0	0280H	0280H	0
6	0	0352H	01B8H	0
7	0	0184H	019AH	0

Figure 2-5. PC Convertible Display Types

For all others no action is performed.

**(AH) = 16H to 19H - Reserved**

## **(AH) = 1AH - Read/Write Display Combination Code**

For Personal System/2 products:

(AL) = 00H - Read display combination code

On Return:

(AL) = 1AH - Function supported (see display codes on page 2-39)

(BL) - Active display code

(BH) - Alternate display code

(AL) = 01H - Write display combination code (see display codes on page 2-39)

(BL) - Active display code

(BH) - Alternate display code

On Return:

(AL) = 1AH - Function supported

Display Codes:

00H - No display

01H - Monochrome with 5151 (monochrome)

02H - CGA with 5153/4 (color)

03H - Reserved

04H - EGA with 5153/4 (color)

05H - EGA with 5151 (monochrome)

06H - Professional Graphics System with 5175 (color)

07H - Personal System/2 products except Model 30 with analog monochrome

08H - Personal System/2 products except Model 30 with analog color

09H to 0AH - Reserved

0BH - Personal System/2 Model 30 video with analog monochrome

0CH - Personal System/2 Model 30 video with analog color

0DH to FEH - Reserved

-1 - Unknown

For all others no action is performed.

## **(AH) = 1BH - Return Functionality/State Information**

For Personal System/2 products:

(BX) - Implementation type

(ES:DI) - User buffer pointer for return of information

On Return:

User buffer contains functionality/state information

(AL) = 1BH - Function supported

For implementation type 00H:

(BX) = 00H

(ES:DI) = Buffer of size 40H bytes

(DI+00H) word - Offset to static functionality information

(DI+02H) word - Segment to static functionality information

Video states:

(The following information is dynamically generated and reflects the current video state.)

(DI+04H) byte - Video mode [see (AH) = 00H on page 2-11 for supported modes]

(DI+05H) word - Columns on screen (character columns on screen)

(DI+07H) word - Length of regenerator buffer (bytes)

(DI+09H) word - Starting address in regenerator buffer

(DI+0BH) word - Cursor position for eight display pages (row, column)

(DI+1BH) word - Cursor type setting (cursor start/end value)

(DI+1DH) byte - Active display page

(DI+1EH) word - CRT controller address (3BX-monochrome, 3DX-color)

(DI+20H) byte - Current setting of 3x8 register

(DI+21H) byte - Current setting of 3x9 register

(DI+22H) byte - Rows on screen (character lines on screen)

(DI+23H) word - Character height (scan lines per character)

(DI+25H) byte - Display combination code (active)

(DI+26H) byte - Display combination code (alternate)

(DI+27H) word - Colors supported for current video mode

(DI+29H) byte - Display pages supported for current video mode

(DI+2AH) byte - Scan lines in current video mode

= 0 - 200 scan lines

= 1 - 350 scan lines

= 2 - 400 scan lines

= 3 - 480 scan lines

= 4 to 255 - Reserved

(DI+2BH) byte - Primary character block (Reserved on  
Personal System/2 Model 30)  
= 0 - Block 0  
= 1 - Block 1  
= 2 - Block 2  
.  
.  
.  
.  
= 255 - Block 255

This information is based on block specifier [see (AH) = 11H,  
(AL) = 03H].

(DI+2CH) byte - Secondary character block (Reserved on  
Personal System/2 Model 30)  
= 0 - Block 0  
= 1 - Block 1  
= 2 - Block 2  
.  
.  
.  
.  
= 255 - Block 255

This information is based on block specifier [see (AH) = 11H,  
(AL) = 03H].

(DI+2DH) byte - Miscellaneous state information  
Bits 7, 6 - Reserved  
Bit 5 = 0 - Background intensity  
= 1 - Blinking  
Bit 4 = 1 - Cursor emulation active  
(Always 0 for Personal System/2 Model 30)  
Bit 3 = 1 - Mode set default palette loading disabled  
Bit 2 = 1 - Monochrome display attached  
Bit 1 = 1 - Summing active  
Bit 0 = 1 - All modes on all displays active  
(Always 0 for Personal System/2 Model 30)

(DI+2EH) byte - Reserved

(DI+2FH) byte - Reserved

(DI+30H) byte - Reserved

(DI+31H) byte - Video memory available  
= 0 - 64Kb  
= 1 - 128Kb  
= 2 - 192Kb  
= 3 - 256Kb  
= 4 to 255 - Reserved

(DI+32H) byte - Save pointer state information  
Bits 7, 6 - Reserved  
Bit 5 = 1 - DCC extension active  
Bit 4 = 1 - Palette override active  
Bit 3 = 1 - Graphics font override active  
Bit 2 = 1 - Alpha font override active  
Bit 1 = 1 - Dynamic save area active  
Bit 0 = 1 - 512-character set active

(DI+33H) to (DI+3FH) 13 bytes - Reserved

Format of static functionality table:

0 = Not supported  
1 = Supported

(00H) byte - Video modes  
Bit 7 = Mode 07H  
Bit 6 = Mode 06H  
Bit 5 = Mode 05H  
Bit 4 = Mode 04H  
Bit 3 = Mode 03H  
Bit 2 = Mode 02H  
Bit 1 = Mode 01H  
Bit 0 = Mode 00H

(01H) byte - Video modes  
Bit 7 = Mode 0FH  
Bit 6 = Mode 0EH  
Bit 5 = Mode 0DH  
Bit 4 = Mode 0CH  
Bit 3 = Mode 0BH  
Bit 2 = Mode 0AH  
Bit 1 = Mode 09H  
Bit 0 = Mode 08H

(02H) byte - Video modes  
Bits 7 to 4 - Reserved  
Bit 3 = Mode 13H  
Bit 2 = Mode 12H  
Bit 1 = Mode 11H  
Bit 0 = Mode 10H

See (AH) = 00H on page 2-11 for video mode information.

(03H) to (07H) 4 bytes - Reserved

(07H) byte - Scan lines available in text modes

Bits 7 to 3 - Reserved

Bit 2 = 400 scan lines

Bit 1 = 350 scan lines

Bit 0 = 200 scan lines

See (AH) = 12H, (BL) = 30H for text mode scan line selection.

(08H) byte - Character blocks available in text modes

(09H) byte - Maximum number of active character blocks in text modes

See (AH) = 11H for character block loading interfaces.

(0AH) byte - Miscellaneous functions

Bit 7 = Color paging [see (AH) = 10H]

(Always 0 for Personal System/2 Model 30)

Bit 6 = Color palette [see (AH) = 10H]

Bit 5 = EGA palette [see (AH) = 10H]

Bit 4 = Cursor emulation [see (AH) = 01H]

Bit 3 = Mode set default palette loading [see

(AH) = 12H]

Bit 2 = Character font loading [see (AH) = 11H]

Bit 1 = Summing [see (AH) = 10H and (AH) = 12H]

Bit 0 = All modes on all displays

(Always 0 for Personal System/2 Model 30)

(0BH) byte - Miscellaneous functions

Bits 7 to 4 - Reserved

Bit 3 = DCC [see (AH) = 1AH]

Bit 2 = Background intensity/blinking control [see

(AH) = 10H]

Bit 1 = Save/restore [see (AH) = 1CH]

(Always 0 for Personal System/2 Model 30)

Bit 0 = Light pen [see (AH) = 04H]

(0CH) to (0DH) 2 bytes - Reserved

(0EH) byte - Save pointer functions  
Bits 7, 6 = Reserved  
Bit 5 = DCC extension  
    (Always 0 for Personal System/2 Model 30)  
Bit 4 = Palette override  
Bit 3 = Graphics font override  
Bit 2 = Alpha font override  
Bit 1 = Dynamic save area  
Bit 0 = 512-character set

(0FH) byte - Reserved

For all others no action is performed.

### **(AH) = 1CH - Save/Restore Video State**

For Personal System/2 products except Model 30:

(AL) = 00H - Return save/restore state buffer size  
(CX) - Requested states (see supported save/restore states  
on page 2-45)

On Return:

(AL) = 1CH - Function supported  
(BX) - Save/restore buffer size block count [number of  
64-byte blocks for saving requested states in (CX)]

(AL) = 01H - Save state  
(CX) = Requested states (see supported save/restore states  
on page 2-45)  
(ES:BX) = Buffer pointer to save state

On Return:

(AL) = 1CH - Function supported

Requested states saved

(AL) = 02H - Restore state  
(CX) - Requested states (see supported save/restore states  
on page 2-45)  
(ES:BX) - Buffer pointer to restore state

On Return:

(AL) = 1CH - Function supported

Requested states restored

Supported save/restore states

Bits 15 to 3 - Reserved and set to 0

Bit 2 = 1 - Save/restore video DAC state and color registers

Bit 1 = 1 - Save/restore video BIOS data area

Bit 0 = 1 - Save/restore video hardware state

**Note:** The current video state is altered during save a state operation. To maintain the current video state, perform a restore state operation.

For all others no action is performed.

**(AH) = 1DH to FFH - Reserved**

---

## Interrupt 11H - Equipment Determination

This routine returns the optional devices that are attached to the system. BIOS data area 40:10 (installed hardware) is set during the POST as follows:

On Return:

- (AX) - Equipment flags
  - Bits 15,14 - Number of printers attached
  - Bit 13 - Internal modem installed
  - Bit 12 - Not used
  - Bits 11,10,9 - Number of RS-232-C cards attached
  - Bit 8 - Not used
  - Bits 7,6 - Number of diskette drives, if bit 0 = 1
    - (values are binary)
    - = 00 - 1 drive
    - = 01 - 2 drives
  - Bits 5,4 - Video mode type (values are binary)
    - = 00 - Reserved
    - = 01 - 40x25 (color)
    - = 10 - 80x25 (color)
    - = 11 - 80x25 (monochrome)
  - Bit 3 - Not used
  - Bit 2 - Pointing device installed
  - Bit 1 = Math coprocessor installed
  - Bit 0 = IPL diskette installed

---

## Interrupt 12H - Memory Size Determination

This routine returns the amount of RAM up to 640Kb in the system as determined by the POST, minus the memory allocated to the Extended BIOS Data Area. See INT 15H, (AH) = C1H (Return Extended BIOS Data Area Segment Address) on page 2-96, and INT 15H, (AH) = 88H (Extended Memory Size Determine) on page 2-89 for additional information.

The following assumptions are made during memory size determination:

- All installed memory is functional
- All memory from 0 to 640Kb is contiguous.

On Return, (AX) contains the number of contiguous 1Kb blocks of memory.

---

## Interrupt 13H - Diskette

This interface provides access to diskette drives. The following is a summary of the diskette functions of Interrupt 13H:

(AH) = 00H	– Reset Diskette System
(AH) = 01H	– Read Status of Last Operation
(AH) = 02H	– Read Desired Sectors into Memory
(AH) = 03H	– Write Desired Sectors from Memory
(AH) = 04H	– Verify Desired Sectors
(AH) = 05H	– Format Desired Track
(AH) = 06H to 07H	– Reserved for Fixed Disk
(AH) = 08H	– Read Drive Parameters
(AH) = 09H to 14H	– Reserved for Fixed Disk
(AH) = 15H	– Read DASD Type
(AH) = 16H	– Diskette Change Line Status
(AH) = 17H	– Set DASD Type for Format
(AH) = 18H	– Set Media Type for Format
(AH) = 19H to FFH	– Reserved

Figure 2-6. INT 13H - Diskette Functions

**Note:** For the diskette drive parameters see “Diskette Drive Parameter Table” on page 3-23.

For AT, PC XT BIOS dated 1/10/86 and after, PC XT Model 286, PC Convertible, and Personal System/2 products, operations that require the diskette drive motor to be turned on call INT 15H, (AX) = 90FDH (Diskette Drive Motor Start). This allows the operating system to perform a different task while waiting for the diskette drive motor to accelerate.

Prior to waiting for the diskette interrupt, BIOS calls INT 15H, (AH) = 90H (Device Busy) with (AL) = 01H (Type = Diskette). This informs the operating system of the wait. The complementary INT 15H, (AH) = 91H (Interrupt Complete) with (AL) = 01H (Type = Diskette) is called indicating the operation is complete. See “Multitasking Provisions” on page 4-16 for additional information.

### (AH) = 00H - Reset Diskette System

(DL) - Drive number (0-based)  
Bit 7 = 0 - Diskette

On Return:

CF = 1 - Status is non 0  
= 0 - Status is 0

(AH) - Status of operation  
= 80H - Diskette drive not ready  
= 40H - Seek operation failed  
= 20H - General controller failure  
= 10H - Cyclic redundancy check (CRC) error on diskette read  
= 0CH - Media type not found  
= 09H - Attempt to DMA across a 64Kb boundary  
= 08H - DMA overrun on operation  
= 06H - Diskette change line active  
= 04H - Requested sector not found  
= 03H - Write protect error  
= 02H - Address mark not found  
= 01H - Invalid diskette parameter  
= 00H - No error

Diskette status at 40:41 - Status of operation

### Notes:

1. If an error is reported by the diskette BIOS, reset the diskette system and retry the operation.
2. If (DL) is greater than or equal to hex 80, the diskette system is reset then the fixed disk system is reset. The status returned in (AH) is the status of fixed disk reset. Read the status of the diskette system after completion of the operation.

### **(AH) = 01H - Read Status of Last Operation**

(DL) - Drive number (0-based)  
Bit 7 = 0 - Diskette (value checked)

On Return:

CF = 1 - Status is non 0  
= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-49)

## **(AH) = 02H - Read Desired Sectors into Memory**

(DL) - Drive number (0-based)  
Bit 7 = 0 - Diskette (value checked)  
(DH) - Head number (not value checked, 0-based)  
(CH) - Track number (not value checked, 0-based)  
(CL) - Sector number (not value checked)  
(AL) - Number of sectors (not value checked)  
(ES:BX) - Address of buffer

### On Return:

CF = 1 - Status is non 0  
= 0 - Status is 0  
(AL) - Number of sectors actually transferred  
(AH) - Status of operation (see values for the status of operation on page 2-49)

Diskette status at 40:41 - Status of operation

**Note:** If an error is reported by the diskette BIOS, reset the diskette system, then retry the operation.

## **(AH) = 03H - Write Desired Sectors from Memory**

(DL) - Drive number (0-based)  
Bit 7 = 0 - Diskette (value checked)  
(DH) - Head number (not value checked, 0-based)  
(CH) - Track number (not value checked, 0-based)  
(CL) - Sector number (not value checked)  
(AL) - Number of sectors (not value checked)  
(ES:BX) - Address of buffer

### On Return:

CF = 1 - Status is non 0  
= 0 - Status is 0  
(AL) - Number of sectors actually transferred  
(AH) - Status of operation (see values for the status of operation on page 2-49)

Diskette status at 40:41 - Status of operation

**Notes:**

1. If an error is reported by the diskette BIOS, reset the diskette system, then retry the operation.
2. For PC XT Model 286, (AL) is not required.

**(AH) = 04H - Verify Desired Sectors**

(DL) - Drive number (0-based)  
Bit 7 = 0 - Diskette (value checked)  
(DH) - Head number (not value checked, 0-based)  
(CH) - Track number (not value checked, 0-based)  
(CL) - Sector number (not value checked)  
(AL) - Number of sectors (not value checked)  
(ES:BX) - Address of buffer

## On Return:

CF = 1 - Status is non 0  
= 0 - Status is 0  
(AL) - Number of sectors verified  
(AH) - Status of operation (see values for the status of operation on page 2-49)

Diskette status at 40:41 - Status of operation

**Notes:**

1. If an error is reported by the diskette BIOS, reset the diskette system, then retry the operation.
2. ES:BX is not required for AT BIOS dated 11/15/85 and after, PC XT Model 286, PC Convertible, or Personal System/2 products.

**(AH) = 05H - Format Desired Track**

The buffer pointer (ES:BX) must point to the collection of desired address fields for the track. Each field has the following four bytes:

Byte 0 - Track number  
Byte 1 - Head number  
Byte 2 - Sector number

Byte 3 - Number of bytes per sector  
= 00H - 128-bytes per sector  
= 01H - 256-bytes per sector  
= 02H - 512-bytes per sector  
= 03H - 1024-bytes per sector

There must be one entry for every sector on the track. This information is used to find the requested sector during read/write access. Prior to formatting a diskette, if there is more than one supported format for the drive in question, it is necessary to call (AH) = 17H (Set DASD Type for Format), or (AH) = 18H (Set Media Type for Format) to set the diskette type to be formatted.

(AL) - Number of sectors to format (not value checked)  
(DL) - Drive number (0-based)  
    Bit 7 = 0 - Diskette (value checked)  
(DH) - Head number (not value checked, 0-based)  
(CH) - Track number (not value checked, 0-based)  
(ES:BX) - Address of buffer

On Return:

CF = 1 - Status is non 0  
    = 0 - Status is 0  
(AH) - Status of operation (see values for the status of operation on page 2-49)

Diskette status at 40:41 - Status of operation

**Notes:**

1. If an error is reported by the diskette BIOS, reset the diskette system, then retry the operation.
2. The diskette parameter table is used to format the diskette. See "Diskette Drive Parameter Table" on page 3-23.
3. For PC XT Model 286, (AL) is not required.

**(AH) = 06H to 07H - Reserved for Fixed Disk**

## **(AH) = 08H - Read Drive Parameters**

There is a parameter table for each supported media type.

For PCjr, PC, PC XT, or for AT BIOS dated 1/10/84:

On Return:

CF = 1 - Error

(AH) - Status of operation

= 01H - Invalid command

Diskette status at 40:41 - Status of operation

For all others:

(DL) - Drive number (0-based)

Bit 7 = 0 - Diskette (value checked)

On Return:

(ES:DI) - Pointer to 11-byte parameter table

associated with the maximum supported media type

within the drive in question (see

"Diskette Drive Parameter Table" on page 3-23.)

(CH) - Maximum number of tracks (low 8 bits of 10-bit track number, 0-based)

(CL) - Bits 7, 6 - Maximum number of tracks (high 2 bits of 10-bit track number, 0-based)

- Bits 5 to 0 - Maximum sectors per track

(DH) - Maximum head number

(DL) - Number of diskette drives installed

(BH) = 0

(BL) - Bits 7 to 4 = 0

Bits 3 to 0 - Valid drive type value in CMOS

= 01H - 360Kb, 5.25 inch, 40 track

= 02H - 1.2Mb, 5.25 inch, 80 track

= 03H - 720Kb, 3.5 inch, 80 track

= 04H - 1.44Mb, 3.5 inch, 80 track

(AX) = 0

When the drive type is known but the CMOS type is invalid, CMOS is not present, CMOS battery is discharged or CMOS checksum is invalid, all registers are returned as above except (BL) = 0.

If the requested drive is not installed, then (AX), (BX), (CX), (DX), (DI), and (ES) = 0.

Diskette status 40:41 = 0 and CF = 0

For drive number 80H or above (indicating fixed disks):

CF = 1 - Error  
(AH) - Status of operation  
= 01H - Invalid command

(ES), (AX), (BX), (CX), (DH), and (DI) all equal 0 and (DL) contains the number of drives when any of the following conditions exist:

- Drive number is invalid
- Drive type is unknown and the CMOS is not present
- CMOS battery is discharged or CMOS checksum is invalid
- Drive type is unknown and the CMOS drive type is invalid.

Diskette status 40:41 = 0 and CF = 0

**(AH) = 09H to 14H - Reserved for Fixed Disk**

**(AH) = 15H - Read DASD Type**

For AT, PC XT BIOS dated 1/10/86 and after, PC XT Model 286, PC Convertible, and Personal System/2 products:

(DL) - Drive number (0-based)  
Bit 7 = 0 - Diskette (value checked)

On Return:

CF = 0 - Operation successfully completed  
(AH) = 00H - Drive not present  
= 01H - Diskette, no change line available  
= 02H - Diskette, change line available  
= 03H - Reserved for fixed disk interface

Diskette status at 40:41 - Status of operation

## For all others:

On Return:

CF = 1 - Error

(AH) - Status of operation

= 01H - Invalid command

Diskette status at 40:41 - Status of operation

## **(AH) = 16H - Diskette Change Line Status**

For AT, PC XT BIOS dated 1/10/86 and after, PC XT Model 286, PC Convertible, and Personal System/2 products:

(DL) - Drive number (0-based)

Bit 7 = 0 - Diskette (value checked)

On Return:

(AH) = 00H - 'Diskette change' signal not active

= 01H - Invalid diskette parameter

= 06H - 'Diskette change' signal active

= 80H - Diskette drive not ready

CF = 0 if (AH) is 0

= 1 if (AH) is non 0

Diskette status at 40:41 - (AH) on return

## For all others:

On Return:

(AH) - Status of operation

= 01H - Invalid command

CF = 1 - Error

Diskette status at 40:41 - Status of operation

## **(AH) = 17H - Set DASD Type for Format**

The 'diskette change' signal is checked for all drives that support it. If found active, the logic attempts to reset 'diskette change' to the inactive state. If successful, the BIOS sets the data rate for format and returns the disk change error code. If the attempt fails, the logic returns the time-out error code and sets the DASD type to a predetermined state, indicating that the media type is unknown.

When the 'diskette change' signal is found active, as it is after a diskette is changed, this function is called again.

For PC XT BIOS dated 1/10/86 and after, AT, PC Convertible, and Personal System/2 products:

- (DL) - Drive number (0-based)
  - Bit 7 = 0 - Diskette (value checked)
- (AL) = 00H - Invalid request
  - = 01H - Diskette 320/360Kb in 360Kb drive
  - = 02H - Diskette 360Kb in 1.2Mb drive
  - = 03H - Diskette 1.2Mb in 1.2Mb drive
  - = 04H - AT BIOS before 6/10/85: Invalid request
    - All others: Diskette 720Kb in 720Kb drive
  - = 05H through 0FFH - Invalid request

On Return:

- CF = 1 - Status is non 0
- = 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-49)

Diskette status at 40:41 - Status of operation

For all others:

On Return:

- (AH) - Status of operation
  - = 01H - Invalid command
- CF = 1 - Error

Diskette status at 40:41 - Status of operation

### **(AH) = 18H - Set Media Type for Format**

For AT BIOS dated 11/15/85 and after, PC XT BIOS dated 1/10/86 and after, PC XT Model 286, and Personal System/2 products, this function is called before issuing INT 13H, (AH) = 05H (Format the Desired Track). If the diskette is changed, the function is called again. A diskette must be present in the drive.

There is one parameter table for each supported media type.

- (DL) - Drive number (0-based)
  - Bit 7 = 0 - Diskette (value checked)
- (CH) - Number of tracks (low 8 bits, 0-based)
- (CL) - Bits 7, 6 - Number of tracks (high two bits, 0-based)
  - Bits 5 to 0 - Sectors per track

On Return:

- (ES:DI) - Pointer to 11-byte parameter table for this media type, unchanged if (AH) is non 0 (see "Diskette Drive Parameter Table" on page 3-23.)

- CF = 1 - Status is non 0
- = 0 - Status is 0

- (AH) - Status of operation (see values for the status of operation on page 2-49)

**Note:** For PC XT Model 286 and Personal System/2 products, this function monitors the 'diskette change' signal. If the signal is active, the logic attempts to reset the change line to the inactive state. If the attempt succeeds (for example, as when media is present), the BIOS sets the correct data rate for format. If the attempt fails (for example, as when no media is present), then the BIOS returns (AH) = 80H (Diskette Drive Not Ready) and the carry flag is set.

When the 'diskette change' signal is inactive, the BIOS performs the function as requested.

For all others:

On Return:

- (AH) - Status of operation
  - = 01H - Invalid command
- CF = 1 - Error

Diskette status at 40:41 - Status of operation

**(AH) = 19H to FFH - Reserved**

---

## Interrupt 13H - Fixed Disk

This interface provides access to fixed disk drives. The following is a summary of the fixed disk functions of INT 13H:

(AH) = 00H - Reset Disk System
(AH) = 01H - Read Status of Last Operation
(AH) = 02H - Read Desired Sectors into Memory
(AH) = 03H - Write Desired Sectors from Memory
(AH) = 04H - Verify Desired Sectors
(AH) = 05H - Format Desired Cylinder
(AH) = 06H - Format Desired Cylinder and Set Bad Sector Flags
(AH) = 07H - Format Drive Starting at Desired Cylinder
(AH) = 08H - Read Drive Parameters
(AH) = 09H - Initialize Drive Pair Characteristics
(AH) = 0AH to 0BH - Reserved for Diagnostics
(AH) = 0CH - Seek
(AH) = 0DH - Alternate Disk Reset
(AH) = 0EH to 0FH - Reserved for Diagnostics
(AH) = 10H - Test Drive Ready
(AH) = 11H - Recalibrate
(AH) = 12H to 14H - Reserved for Diagnostics
(AH) = 15H - Read DASD Type
(AH) = 16H to 18H - Reserved for Diskette
(AH) = 19H - Park Heads
(AH) = 1AH - Format Unit
(AH) = 1BH to FFH - Reserved

Figure 2-7. INT 13H - Fixed Disk Functions

### Notes:

1. If a fixed disk drive adapter is not installed, the code is not hooked into INT 13H. The returns are described in the diskette interface.
2. For the fixed disk interface, the drive number (DL) is value checked for all functions that use (DL).
3. For AT, PC XT Model 286, and Personal System/2 products, prior to waiting for interrupt, the BIOS calls INT 15H, (AH) = 90H (Device Busy) with (AL) = 00H (Type = Disk), informing the operating system of the wait. The complementary INT 15H, (AH) = 91H (Interrupt Complete) with (AL) = 00H (Type = Disk), is called indicating the operation is complete.
4. For Personal System/2 products, prior to waiting for the fixed disk reset the BIOS calls INT 15H, (AH) = 90H (Device Busy) with (AH) = FCH (Type = Fixed Disk Reset). This is a time-out only

function. There is no complementary Post operation. See "Multitasking Provisions" on page 4-16.

5. Bit 7 of the drive number must be set upon entry to the fixed disk BIOS.
6. For the drive parameters see "Fixed Disk Drive Parameter Table" on page 3-16.

### **(AH) = 00H - Reset Disk System**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

On Return:

CF = 1 - Status is non 0  
= 0 - Status is 0

(AH) - Status of operation

- = 00H - No error
- = 01H - Invalid function request
- = 02H - Address mark not found
- = 03H - Write protect error
- = 04H - Sector not found
- = 05H - Reset failed
- = 07H - Drive parameter activity failed
- = 08H - DMA overrun on operation
- = 09H - Data boundary error
- = 0AH - Bad sector flag detected
- = 0BH - Bad cylinder detected
- = 0DH - Invalid number of sectors on format
- = 0EH - Control data address mark detected
- = 0FH - DMA arbitration level out of range
- = 10H - Uncorrectable error checking and correction (ECC) or cyclic redundancy check (CRC) error
- = 11H - ECC corrected data error
- = 20H - General controller failure
- = 40H - Seek operation failed
- = 80H - Time-out
- = BBH - Undefined error occurred
- = CCH - Write fault on selected drive
- = E0H - Status error/error register = 0
- = FFH - Sense operation failed

**Notes:**

1. Reset Disk System is issued only if the 7-bit drive number is less than or equal to the maximum number of fixed disk drives. The diskette system is also reset for all values of (DL).
2. For Personal System/2 products, prior to waiting for the fixed disk reset, the BIOS calls INT 15, (AH) = 90H (Device Busy) with (AL) = 00H (Type = Disk) informing the operating system of the wait.

**(AH) = 01H - Read Status of Last Operation**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

On Return:

CF = 1 - Status is non 0  
= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-59)

Disk status is reset to 0

**(AH) = 02H - Read Desired Sectors into Memory**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

(DH) - Head number (0-based, not value checked)

(CH) - Cylinder number (low 8 bits of 10-bit cylinder number, 0-based, not value checked)

(CL) - Bits 7, 6 - Cylinder number (high 2 bits of 10-bit cylinder number, 0-based, not value checked)  
- Bits 5 to 0 - Sector number (not value checked)

(AL) - Number of sectors

(ES:BX) - Address of buffer

On Return:

CF = 1 - Status is non 0  
= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-59)

## Notes:

1. An 11H error indicates the data read had a recoverable error that was corrected by the ECC algorithm. The data may be good; however, the BIOS routine indicates an error to allow the controlling program to make this determination. The error may not recur if the data is rewritten.
2. If an error is reported by the fixed disk BIOS, reset the disk system, then retry the operation.

### **(AH) = 03H - Write Desired Sectors from Memory**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)  
(DH) - Head number (0-based, not value checked)  
(CH) - Cylinder number (low 8 bits of 10-bit cylinder number, 0-based, not value checked)  
  
(CL) - Bits 7, 6 - Cylinder number (high 2 bits of 10-bit cylinder number, 0-based, not value checked)  
      - Bits 5 to 0 - Sector number (not value checked)  
(AL) - Number of sectors  
(ES:BX) - Address of buffer

#### On Return:

CF = 1 - Status is non 0  
      = 0 - Status is 0  
(AH) - Status of operation (see values for the status of operation on page 2-59)

**Note:** If an error is reported by the fixed disk BIOS, reset the disk system, then retry the operation.

### **(AH) = 04H - Verify Desired Sectors**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)  
(DH) - Head number (0-based, not value checked)  
(CH) - Cylinder number (low 8 bits of 10-bit cylinder number, 0-based, not value checked)  
  
(CL) - Bits 7, 6 - Cylinder number (high 2 bits of 10-bit cylinder number, 0-based, not value checked)  
      - Bits 5 to 0 - Sector number (not value checked)  
(AL) - Number of sectors

On Return:

CF = 1 - Status is non 0

= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-59)

**Note:** If an error is reported by the fixed disk BIOS, reset the disk system, then retry the operation.

**(AH) = 05H - Format Desired Cylinder**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

(DH) - Head number (0-based, not value checked)

(CH) - Cylinder number (low 8 bits of 10-bit cylinder number, 0-based, not value checked)

(CL) - Bits 7, 6 - Cylinder number (high 2 bits of 10-bit cylinder number, 0-based, not value checked)

For PC XT:

(AL) - Contains interleave value

On Return:

CF = 1 - Status is non 0

= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-59)

For AT, PC XT Model 286, and Personal System/2 products:

(ES:BX) - Address of buffer

(ES:BX) points to a 512-byte buffer. The first 2 x (Sectors per cylinder) bytes contain F, N for each sector.

F = 00H - Good sector

= 80H - Bad sector

N - Sector number

On Return:

CF = 1 - Status is non 0

= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-59)

**For the IBM ESDI Fixed Disk Drive Adapter/A:**

On Return:

(AH) - Status of operation = 01H - Invalid function request  
CF = 1 - Error

**Note:** If an error is reported by the fixed disk BIOS, reset the disk system, then retry the operation.

**(AH) = 06H - Format Desired Cylinder and Set Bad Sector Flags**

**Warning:** Formatting destroys all information on the fixed disk drive.

For PC XT:

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)  
(DH) - Head number (0-based, not value checked)  
(CH) - Cylinder number (low 8 bits of 10-bit cylinder number, 0-based, not value checked)  
(CL) - Bits 7, 6 - Cylinder number (high 2 bits of 10-bit cylinder number, 0-based, not value checked)  
(AL) - Interleave value

On Return:

CF = 1 - Status is non 0  
= 0 - Status is 0  
(AH) - Status of operation (see values for the status of operation on page 2-59)

**For AT, PC XT Model 286, Personal System/2 products, and the IBM ESDI Fixed Disk Drive Adapter/A:**

On Return:

(AH) - Status of operation = 01H - Invalid function request  
CF = 1 - Error

**Note:** If an error is reported by the fixed disk BIOS, reset the disk system, then retry the operation.

## **(AH) = 07H - Format Drive Starting at Desired Cylinder**

For PC XT:

- (DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)
- (CH) - Cylinder number (low 8 bits of 10-bit cylinder number, 0-based, not value checked)
- (CL) - Bits 7, 6 - Cylinder number (high 2 bits of 10-bit cylinder number, 0-based, not value checked)
- (AL) - Interleave value

On Return:

- CF = 1 - Status is non 0
- = 0 - Status is 0
- (AH) - Status of operation (see values for the status of operation on page 2-59)

For AT, PC XT Model 286, Personal System/2 products, and the IBM ESDI Fixed Disk Drive Adapter/A:

On Return:

- (AH) - Status of operation = 01H - Invalid function request
- CF = 1 - Error

**Note:** If an error is reported by the fixed disk BIOS, reset the disk system, then retry the operation.

## **(AH) = 08H - Read Drive Parameters**

If the drive number is invalid then (AH) and 40:74 = 07H (last fixed disk drive operation status), (CX) and (DX) = 0, and CF is set. If no fixed disk drive is attached or no fixed disk drive adapter is installed, (AH) and 40:41 = 01H (last diskette drive operation status), and CF is set.

- (DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

On Return:

- (DL) - Number of consecutive drives attached (1, 2; controller card 0 tally only)
- (DH) - Maximum value for head number (range 0-3FH)
- (CH) - Maximum value for cylinder number (range 0-3FFFH)
- (CL) - Maximum value for sector and high order 2 bits of cylinder numbers

## **(AH) = 09H - Initialize Drive Pair Characteristics**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

On Return:

CF = 1 - Status is non 0

= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-59)

For PC XT:

Interrupt 41H points to the parameter tables. Four entries in the PC XT table correspond to the switch settings on the fixed disk drive adapter. The switches act as an index into the parameter table. For example, if both switches are set to the On position, the drive is initialized with the first entry of the parameter table. If the drive number is an allowable value [ $80H \leq (DL) \leq 87H$ ] then both drives 0 and 1 are initialized. For all other values, an invalid command status is returned. If drive 0 initialization fails, drive 1 initialization is not attempted. If either attempt fails, 40:74 = 07H (last fixed disk drive operation status) and (AH) are updated with the appropriate error.

For AT, PC XT Model 286, and Personal System/2 products:

Interrupt 41H points to the single parameter table for drive 0, and interrupt 46H points to the single parameter table for drive 1. If (DL) = 80H, then drive 0 is initialized using interrupt 41H. If (DL) = 81H, then drive 1 is initialized using interrupt 46H. For all other values, an invalid command status is returned.

For the IBM ESDI Fixed Disk Drive Adapter/A:

This function performs no action. Drive configuration information is obtained from the drive, not from a table in the host ROM. Drive type initialization is performed automatically by the controller.

**Note:** If an error is reported by the fixed disk BIOS, reset the disk system, then retry the operation.

**(AH) = 0AH to 0BH - Reserved for Diagnostics**

**(AH) = 0CH - Seek**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

(DH) - Head number (0-based, not value checked)

(CH) - Cylinder number (low 8 bits of 10-bit cylinder number, 0-based, not value checked)

(CL) - Bits 7, 6 - Cylinder number (high 2 bits of 10-bit cylinder number, 0-based, not value checked)

On Return:

CF = 1 - Status is non 0

= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-59)

**Note:** If an error is reported by the fixed disk BIOS, reset the disk system, then retry the operation.

**(AH) = 0DH - Alternate Disk Reset**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

On Return:

CF = 1 - Status is non 0

= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-59)

**Notes:**

1. Alternate Disk Reset is issued only if the 7-bit drive number is less than or equal to the maximum number of fixed disk drives.
2. For the IBM ESDI Fixed Disk Drive Adapter/A, (AH) = 0DH is reserved for diagnostics.

**(AH) = 0EH to 0FH - Reserved for Diagnostics**

**(AH) = 10H - Test Drive Ready**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

On Return:

CF = 1 - Status is non 0

= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-59)

### **(AH) = 11H - Recalibrate**

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

On Return:

CF = 1 - Status is non 0

= 0 - Status is 0

(AH) - Status of operation (see values for the status of operation on page 2-59)

**Note:** If an error is reported by the fixed disk BIOS, reset the disk system, then retry the operation.

### **(AH) = 12H to 14H - Reserved for Diagnostics**

### **(AH) = 15H - Read DASD Type**

For PC XT:

On Return:

(AH) - Status of operation = 01H - Invalid function request

CF = 1 - Error

For AT, PC XT Model 286, and Personal System/2 products:

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

On Return:

(AH) = 00H - Drive not present or (DL) invalid

= 01H - Reserved for diskette interface

= 02H - Reserved for diskette interface

= 03H - Fixed disk

(CX,DX) - Number of 512-byte blocks

If (AH) = 0 then (CX) and (DX) = 0

CF = 0 - Operation successfully completed

**(AH) = 16H to 18H - Reserved for Diskette**

**(AH) = 19H - Park Heads**

For PC XT, AT, and PC XT Model 286:

On Return:

(AH) - Status of operation = 01H - invalid function request

CF = 1 - Error

For Personal System/2 products:

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

On Return:

(AH) - Status of operation (see values for the status of operation on page 2-59)

CF = 1 - Error

**(AH) = 1AH - Format Unit**

For the IBM ESDI Fixed Disk Drive Adapter/A:

**Warning:** Formatting destroys all information on the fixed disk drive.

This command may take more than an hour to complete.

(AH) = 1AH - Format fixed disk drive

(AL) - Relative block address (RBA) defect table block count  
= 0 - No RBA table associated with this format request  
> 0 - RBA table used

(ES:BX) - Address of RBA table

(CL) - Modifier bits

Bits 7, 6, 5 - Must be 0

Bit 4 - Periodic interrupt. The controller interrupts the host for every cylinder completed during each phase of the formatting operation. This feature allows the host to display formatting progress. The phase is defined as follows:

0 - Reserved

1 - Surface analysis

2 - Formatting

An INT 15H (AH) = 0FH, (AL) = Phase Code is executed by BIOS. The return must clear CF to allow formatting to continue. Set CF to end formatting. The host must keep a running count of interrupts for each phase. This running count is the cylinder number. The host may display formatting progress in granularities other than 1, although interrupts occur for every cylinder.

- Bit 3 - Perform extended surface analysis. (A format with this bit set to 0 must have occurred before attempting to format with this bit set.)
- Bit 2 - Update secondary defect map. This map is updated to reflect defects found during surface analysis and those passed with the format command. If this bit is set with bit 1, the secondary defect map is replaced.
- Bit 1 - Ignore secondary defect map. The secondary defect map is not processed.
- Bit 0 - Ignore primary defect map. The primary defect map is not processed.

(DL) - Drive number, bit 7 = 1 for fixed disk drive (0-based)

For all others this function is reserved.

**(AH) = 1BH to FFH - Reserved**

---

## Interrupt 14H - Asynchronous Communications

These routines provide RS-232-C support. The following is a summary of the RS-232-C support functions of Interrupt 14H:

(AH) = 00H - Initialize the Communications Port  
(AH) = 01H - Send Character  
(AH) = 02H - Receive Character  
(AH) = 03H - Read Status  
(AH) = 04H - Extended Initialize  
(AH) = 05H - Extended Communications Port Control  
(AH) = 06H to FFH - Reserved

Figure 2-8. INT 14H - Asynchronous Communications Functions

### (AH) = 00H - Initialize the Communications Port

(AL) - Parameters for initialization

Bits 7, 6, 5 - Baud rate (values are binary)

= 000 - 110

= 001 - 150

= 010 - 300

= 011 - 600

= 100 - 1200

= 101 - 2400

= 110 - 4800

= 111 - 9600

On Personal System/2 products, for baud rates above 9600, see INT 14H, (AH) = 04H and (AH) = 05H.

Bits 4, 3 - Parity (values are binary)

= 00 - None

= 01 - Odd

= 10 - None

= 11 - Even

Bit 2 - Stop bit

= 0 - 1

= 1 - 2

Bits 1, 0 - Word length (values are binary)

= 10 - 7 Bits

= 11 - 8 Bits

(DX) - RS-232-C Communications line to use (0,1,2,3) corresponding to actual port base address at 40:00

On Return:

- (AL) - Modem status
  - Bit 7 - Received line signal detect
  - Bit 6 - Ring indicator
  - Bit 5 - Data set ready
  - Bit 4 - Clear to send
  - Bit 3 - Delta receive line signal detect
  - Bit 2 - Trailing edge ring detector
  - Bit 1 - Delta data set ready
  - Bit 0 - Delta clear to send

- (AH) - Line status
  - Bit 7 - Time-out
  - Bit 6 - Transmitter shift register empty
  - Bit 5 - Transmitter holding register empty
  - Bit 4 - Break detect
  - Bit 3 - Framing error
  - Bit 2 - Parity error
  - Bit 1 - Overrun error
  - Bit 0 - Data ready

**Note:** If bit 7 of the line status byte is set to one, other bits are unpredictable.

**(AH) = 01H - Send Character**

- (AL) - Character to send
- (DX) - RS-232-C communications line to use (0,1,2,3) corresponding to actual port base addresses at 40:00

On Return:

- (AL) is preserved
- (AH) - Line status (see values for the line status on page 2-71)

**(AH) = 02H - Receive Character**

- (DX) - RS-232-C communications line to use (0,1,2,3) corresponding to actual port base addresses at 40:00

On Return:

- (AL) - Character received
- (AH) - Line status (see values for the line status on page 2-71)

**Note:** The routine waits for the character.

## **(AH) = 03H - Read Status**

(DX) - RS-232-C communications line to use (0,1,2,3) corresponding to actual port base addresses at 40:00

On Return:

(AL) - Modem status (see values for the modem status on page 2-71)

(AH) - Line status (see values for the line status on page 2-71)

## **(AH) = 04H - Extended Initialize**

For Personal System/2 products:

(DX) - RS-232-C communications line to use (0,1,2,3) corresponding to actual port base addresses at 40:00

(AL) - Break  
= 00H - No break  
= 01H - Break

(BH) - Parity  
= 00H - None  
= 01H - Odd  
= 02H - Even  
= 03H - Stick parity odd  
= 04H - Stick parity even

(BL) - Stop bit  
= 00H - One  
= 01H - Two if 6-, 7-, or 8-bit word length  
- One-and-one-half if 5-bit word length

(CH) - Word length  
= 00H - 5 bits  
= 01H - 6 bits  
= 02H - 7 bits  
= 03H - 8 bits

(CL) - Baud rate  
= 00H - 110 baud  
= 01H - 150 baud  
= 02H - 300 baud  
= 03H - 600 baud  
= 04H - 1200 baud  
= 05H - 2400 baud  
= 06H - 4800 baud  
= 07H - 9600 baud  
= 08H - 19200 baud

On Return:

(AL) - Modem status (see values for the modem status on page 2-71)

(AH) - Line status (see values for the line status on page 2-71)

For all others no action is performed.

### **(AH) = 05H - Extended Communications Port Control**

For Personal System/2 products:

(AL) = 00H - Read modem control register

(DX) - RS-232-C communications line to use (0,1,2,3)  
corresponding to actual port base addresses at 40:00

On Return:

(BL) - Modem control register

Bit 7 to 5 - Reserved

Bit 4 = 1 - Loop

Bit 3 = 1 - Out2

Bit 2 = 1 - Out1

Bit 1 = 1 - Request to send

Bit 0 = 1 - Data terminal ready

(AL) = 01H - Write modem control register

(DX) - RS-232-C communications line to use (0,1,2,3)  
corresponding to actual port base addresses at 40:00

On Return:

(BL) - Modem control register

Bit 7 to 5 - Reserved

Bit 4 = 1 - Loop

Bit 3 = 1 - Out2

Bit 2 = 1 - Out1

Bit 1 = 1 - Request to send

Bit 0 = 1 - Data terminal ready

On Return:

(AL) - Modem status (see values for the modem status on page 2-71)

(AH) - Line status (see values for the line status on page 2-71)

For all others no action is performed.

### **(AH) = 06H to FFH - Reserved**

## Interrupt 15H - System Services

The following is a summary of the system services of Interrupt 15H:

(AH) = 00H	– Turn Cassette Motor On
(AH) = 01H	– Turn Cassette Motor Off
(AH) = 02H	– Read Blocks from Cassette
(AH) = 03H	– Write Blocks to Cassette
(AH) = 04H to 0EH	– Reserved
(AH) = 0FH	– Format Unit Periodic Interrupt
(AH) = 10H to 20H	– Reserved
(AH) = 21H	– Power-On Self-Test Error Log
(AH) = 22H to 3FH	– Reserved
(AH) = 40H	– Read/Modify Profiles
(AH) = 41H	– Wait for External Event
(AH) = 42H	– Request System Power-Off
(AH) = 43H	– Read System Status
(AH) = 44H	– Activate/Deactivate Internal Modem Power
(AH) = 45H to 4EH	– Reserved
(AH) = 4FH	– Keyboard Intercept
(AH) = 50H to 7FH	– Reserved
(AH) = 80H	– Device Open
(AH) = 81H	– Device Close
(AH) = 82H	– Program Termination
(AH) = 83H	– Event Wait
(AH) = 84H	– Joystick Support
(AH) = 85H	– System Request Key Pressed
(AH) = 86H	– Wait
(AH) = 87H	– Move Block
(AH) = 88H	– Extended Memory Size Determine
(AH) = 89H	– Switch Processor to Protected Mode
(AH) = 8AH to 8FH	– Reserved
(AH) = 90H	– Device Busy
(AH) = 91H	– Interrupt Complete
(AH) = 92H to BFH	– Reserved
(AH) = C0H	– Return System Configuration Parameters
(AH) = C1H	– Return Extended BIOS Data Area Segment Address
(AH) = C2H	– Pointing Device BIOS Interface
(AH) = C3H	– Enable/Disable Watchdog Time-Out
(AH) = C4H	– Programmable Option Select
(AH) = C5H to FFH	– Reserved

Figure 2-9. INT 15H - System Services Functions

### **(AH) = 00H - Turn Cassette Motor On**

For PCjr and PC:

On Return:

(AH) = 00H

CF = 0

**For all others:**

On Return:  
(AH) = 86H  
CF = 1

**(AH) = 01H - Turn Cassette Motor Off**

**For PCjr and PC:**

On Return:  
(AH) = 00H  
CF = 0

**For all others:**

On Return:  
(AH) = 86H  
CF = 1

**(AH) = 02H - Read Blocks from Cassette**

**For PCjr and PC:**

(ES:BX) - Pointer to data buffer  
(CX) - Count of bytes to read

On Return:  
(ES:BX) - Pointer to last byte read + 1  
(DX) - Count of bytes read  
CF = 0 - No error  
= 1 - Error

For PCjr when CF = 1, (AH) contains:

01H = CRC error  
02H = Lost data transitions  
04H = No data found

**For all others:**

On Return:  
(AH) = 86H  
CF = 1

## **(AH) = 03H - Write Blocks to Cassette**

For PCjr and PC:

(ES:BX) - Pointer to data buffer

(CX) - Count of bytes to write

On Return:

(ES:BX) - Pointer to last byte written + 1

(CX) = 00H

CF = 0 - No error

= 1 - Error

For PCjr when CF = 1, (AH) contains:

01H = CRC error

02H = Lost data transitions

04H = No data found

For all others:

On Return:

(AH) = 86H

CF = 1

## **(AH) = 04H to 0EH - Reserved**

## **(AH) = 0FH - Format Unit Periodic Interrupt**

For the IBM ESDI Fixed Disk Drive Adapter/A:

(AL) - Phase code

= 00H - Reserved

= 01H - Surface analysis

= 02H - Formatting

On Return:

CF = 0 - Continue formatting or scanning

= 1 - End formatting or scanning

**Note:** Function (AH) = 0FH provides a hook to the caller upon completion of formatting or scanning each cylinder. If no handler is hooked, CF is set to 1 on return.

**For PCjr and PC:**

On Return:  
(AH) = 80H  
CF = 1

**For all others:**

On Return:  
(AH) = 86H  
CF = 1

**(AH) = 10H to 20H - Reserved**

**(AH) = 21H - Power-On Self-Test Error Log**

**For PCjr and PC:**

On Return:  
(AH) = 80H  
CF = 1

**For Personal System/2 products except Model 30:**

(AL) = 00H - Read POST error log

On Return:  
(ES:DI) - Pointer to POST error log  
(BX) - Number of POST error codes stored  
CF = 0  
(AH) = 00H

(AL) = 01H - Write error code to POST error log  
(BX) - POST error code (word)  
(BH) - Device code  
(BL) - Device error

On Return:  
CF = 0 - Successfully stored  
= 1 - Error code location full  
(AH) = 00H - Successfully stored  
= 01H - Error code location full

For all others:

On Return:  
(AH) = 86H  
CF = 1

**(AH) = 22H to 3FH - Reserved**

**(AH) = 40H - Read/Modify Profiles**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC Convertible:

(AL) = 00H - Read system profile

On Return:  
(CX,BX) - Profile information

(AL) = 01H - Modify system profile  
(CX,BX) - Profile information

(AL) = 02H - Read internal modem profile

On Return:  
(BX) - Profile information

(AL) = 03H - Modify internal modem profile  
(BX) - Profile information

On Return for all:  
(AL) = 00H - Operation successfully completed  
= 80H - Profile execution failed

CF = 0 - Operation successfully completed  
= 1 - Profile execution failed

For all others:

On Return:  
(AH) = 86H  
CF = 1

**(AH) = 41H - Wait for External Event**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC Convertible:

(ES:DI) - Pointer to byte in user area for event  
determination (event type codes 01H to 04H)  
-or-  
- (DX) contains the address of the I/O port address  
to read for event determination

(AL) - Event type code  
= 00H - Return after any event has occurred  
= 01H - Compare value, return if equal  
= 02H - Compare value, return if not equal  
= 03H - Test bit, return if not 0  
= 04H - Test bit, return if 0

(BH) - Condition compare or mask value  
(BL) - Time-out value (in 55 millisecond units), 0 = No time-out

On Return:  
CF = 1 - Time-out

**Note:** Event type codes (AH) = 11H, 12H, 13H, and 14H are the same as codes (AH) = 01H, 02H, 03H, and 04H, respectively, except that (DX) is used to contain the event determination address.

For all others:

On Return:  
(AH) = 86H  
CF = 1

**(AH) = 42H - Request System Power-Off**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC Convertible:

(AL) = 00H - Use system profile for  
suspend/IPL determination

(AL) = 01H - Force system suspend mode  
regardless of profile

On Return:  
(AX) is modified

For all others:

On Return:  
(AH) = 86H  
CF = 1

**(AH) = 43H - Read System Status**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

## For PC Convertible:

On Return:

(AL) - Status

- Bit 7 - Low battery indication
- Bit 6 - Operating on external power source
- Bit 5 - Standby power lost (real-time clock time bad)
- Bit 4 - Power activated by real-time clock alarm
- Bit 3 - Internal modem power-on
- Bit 2 - RS-232-C/parallel power-on
- Bit 1 - Reserved
- Bit 0 - LCD detached

(AH) is modified

## For all others:

On Return:

(AH) = 86H

CF = 1

## **(AH) = 44H - Activate/Deactivate Internal Modem Power**

### For PCjr and PC:

On Return:

(AH) = 80H

CF = 1

## For PC Convertible:

(AL) = 00H - Power-off internal modem

(AL) = 01H - Power-on internal modem and configure according to system profile

On Return:

(AL) = 00H - Operation successfully completed

= 80H - Request failed

CF = 0 - Operation successfully completed

= 1 - Request failed

## For all others:

On Return:

(AH) = 86H

CF = 1

**(AH) = 45H to 4EH - Reserved**

**(AH) = 4FH - Keyboard Intercept**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC XT BIOS dated 11/08/82, and AT BIOS dated 1/10/84:

On Return:  
(AH) = 86H  
CF = 1

For all others, the keyboard intercept (keyboard escape), is called by the INT 09H (keyboard) routine to allow the keystroke to be changed or absorbed. Normally, the system returns the scan code unchanged, but the operating system can point INT 15H to itself and do one of the following:

1. Replace (AL) with a different scan code and return with the carry flag set, effectively changing the keystroke.
2. Process the keystroke and return with the carry flag reset causing the INT 09H routine to ignore the keystroke.

(AL) - Scan code  
CF = 1

On Return:  
(AL) - New scan code  
CF = 1  
-or-  
(AL) - Unchanged scan code  
CF = 0

**Note:** To dynamically determine the products that support this feature, see INT 15H, (AH) = C0H (Return System Configuration Parameters) on page 2-94.

**(AH) = 50H to 7FH - Reserved**

**(AH) = 80H - Device Open**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC XT BIOS dated 11/08/82:

On Return:  
(AH) = 86H  
CF = 1

For all others:

(BX) - Device ID  
(CX) - Process ID

**(AH) = 81H - Device Close**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC XT BIOS dated 11/08/82:

On Return:  
(AH) = 86H  
CF = 1

For all others:

(BX) - Device ID  
(CX) - Process ID

## **(AH) = 82H - Program Termination**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC XT BIOS dated 11/08/82:

On Return:  
(AH) = 86H  
CF = 1

For all others:

(BX) - Device ID

## **(AH) = 83H - Event Wait**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC XT:

On Return:  
(AH) = 86H  
CF = 1

For AT BIOS dated 1/10/84:

(ES:BX) - Pointer to byte in caller's memory that has the high order bit set as soon as possible after interval expires.

(CX,DX) - Microseconds until posting  
(Granularity is 976 microseconds)

On Return:  
CF = 0 - Operation successfully completed  
= 1 - Operation unsuccessful, function busy

## For all others:

(AL) = 00H - Set interval  
(ES:BX) - Pointer to byte in caller's memory that has the high order bit set as soon as possible after interval expires.  
(CX,DX) - Microseconds until posting  
(Granularity is 976 microseconds)

### On Return:

CF = 0 - Operation successfully completed  
= 1 - Operation unsuccessful, function busy

(AL) = 01H - Cancel set interval

### On Return:

CF = 0 - Operation successfully completed  
= 1 - Operation unsuccessful, function busy

(Personal System/2 Model 30 always returns with CF = 1)

## **(AH) = 84H - Joystick Support**

### For PCjr, PC, and PC Convertible:

#### On Return:

(AH) = 80H  
CF = 1

### For PC XT BIOS dated 11/08/82:

#### On Return:

(AH) = 86H  
CF = 1

## For all others:

(DX) = 00H - Read current switch settings

### On Return:

(AL) - Switch settings (bits 7 to 4)  
CF = 1 - Invalid call

(DX) = 01H - Read resistive inputs

On Return:  
(AX) - A(x) value  
(BX) - A(y) value  
(CX) - B(x) value  
(DX) - B(y) value  
CF = 1 - Invalid call

### **(AH) = 85H - System Request Key Pressed**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC XT BIOS dated 11/08/82:

On Return:  
(AH) = 86H  
CF = 1

For all others:

(AL) = 00H - Key make  
(AL) = 01H - Key break

### **(AH) = 86H - Wait**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC XT:

On Return:  
(AH) = 86H  
CF = 1

**For all others:**

(CX,DX) - Time before return to caller, in microseconds  
(Granularity is 976 microseconds)  
CF = 0 - Successful wait  
= 1 - Wait function already in progress

**(AH) = 87H - Move Block**

**For PCjr and PC:**

On Return:  
(AH) = 80H  
CF = 1

**For PC XT, PC Convertible, and Personal System/2 Model 30:**

On Return:  
(AH) = 86H  
CF = 1

**For AT, PC XT Model 286, and Personal System/2 products except Model 30, this function allows a real mode program or system to transfer a block of data to and from storage above the 1Mb protected mode address range by switching to the protected mode.**

(AH) = 87H - Block move.  
(CX) - Word count of storage block to be moved.  
[maximum count = 8000H for 32Kb words (65Kb)]  
(ES:SI) - Location of a global descriptor table (GDT)  
built by routine using this function.

(ES:SI) points to a global descriptor table (GDT) built before interrupting to this function. The descriptors are used to perform the block move in the protected mode. The source and target descriptors built by the user must have a segment length = 2 x CX-1 or greater. The data access rights byte must be set to CPL0-R/W (93H). The 24-bit address (byte high, word low) must be set to the target/source.

**Note:** No interrupts are allowed during transfer. Large block moves may cause lost interrupts.

On Return:

(AH) = 00H - Operation successfully completed

(AH) = 01H - RAM parity (parity error registers cleared)

(AH) = 02H - Other exception interrupt error occurred

(AH) = 03H - Gate address line 20H failed

All registers are restored except (AH)

If (AH) = 00H:

CF = 0

ZF = 1

If (AH) = 01H to 03H:

CF = 1

ZF = 0

The following shows the organization of a block move GDT:

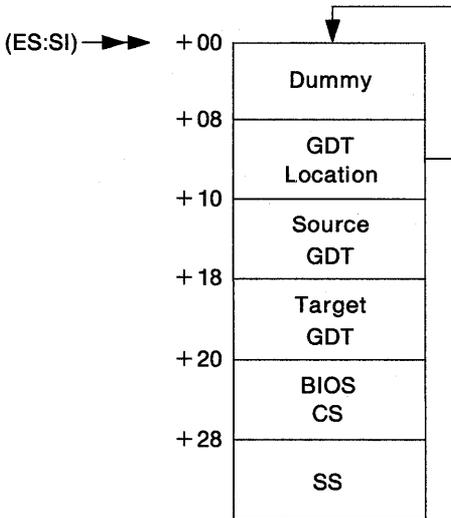


Figure 2-10. Block Move Global Descriptor Table

The descriptors are defined as follows:

- The first is the required dummy and is user initialized to 0.
- The second points to the GDT as a data segment. It is user initialized to 0 and can be modified by the BIOS.
- The third points to the source to be moved and is user initialized.

- The fourth points to the destination segment and is user initialized.
- The fifth is used by the BIOS to create the protected mode code segment. It is user initialized to 0 and can be modified by the BIOS.
- The sixth is used by the BIOS to create a protected mode stack segment. It is user initialized to 0, can be modified by the BIOS, and points to the user stack.

The following is a sample of a source or target descriptor:

```

SOURCE_TARGET_DEF      STRUC

    SEG_LIMIT           DW    ?    ; Segment limit (1 to 65536 bytes)
    LO_WORD             DW    ?    ; 24-bit segment physical
    HI_BYTE             DB    ?    ; address [0 to (16Mb-1)]
    DATA_ACC_RIGHTS   DB    93H  ; Access rights byte (CPL0-R/W)
    Reserved            DW    0    ; Reserved word (must be 0)

SOURCE_TARGET_DEF      ENDS

```

The global descriptor table [actual location pointed to by (ES:SI)]

```

BLOCKMOVE_GDT_DEF      STRUC
    DW    0,0,0,0      ; First descriptor not accessible
CGDT_LOC DW    ?,?,?,0 ; Location of calling routine GDT
SOURCE   DW    ?,?,?,0 ; Source descriptor
TARGET   DW    ?,?,?,0 ; Target descriptor
BIOS_CS  DW    ?,?,?,0 ; BIOS code descriptor
TEMP_SS  DW    ?,?,?,0 ; Stack descriptor
BLOCKMOVE_GDT_DEF      ENDS

```

### **(AH) = 88H - Extended Memory Size Determine**

For PCjr and PC:

```

On Return:
    (AH) = 80H
    CF = 1

```

For PC XT, PC Convertible, and Personal System/2 Model 30:

```

On Return:
    (AH) = 86H
    CF = 1

```

For AT, PC XT Model 286, and Personal System/2 products except Model 30, this routine returns the amount of system memory beginning at address 100000H, as determined by the POST. The system may not be able to use I/O memory unless the system board is fully populated.

On Return:

(AX) - Contiguous 1Kb blocks of memory available beginning at address 100000H

### **(AH) = 89H - Switch Processor to Protected Mode**

For PCjr and PC:

On Return:

(AH) = 80H  
CF = 1

For PC XT, PC Convertible, and Personal System/2 Model 30:

On Return:

(AH) = 86H  
CF = 1

For AT, PC XT Model 286, and Personal System/2 products except Model 30, this function allows the user to switch into protected (virtual address) mode. Upon completion, the processor is in the protected mode and control is transferred to the code segment specified by the user.

Entry requirements:

(ES:SI) points to a GDT built before calling this function. These descriptors initialize the interrupt descriptor table (IDT) register, the GDT register, and the stack segment (SS) selector. The data segment (DS) selector and the extra segment (ES) selector are initialized from descriptors built by the routine using this function.

(BH) contains an index into the interrupt descriptor table that states where the first eight hardware interrupts begin (interrupt level 1). (BL) contains an index into the interrupt descriptor table that states where the second eight hardware interrupts begin (interrupt level 2).

The following shows the organization of a GDT; actual location pointed to by (ES:SI):

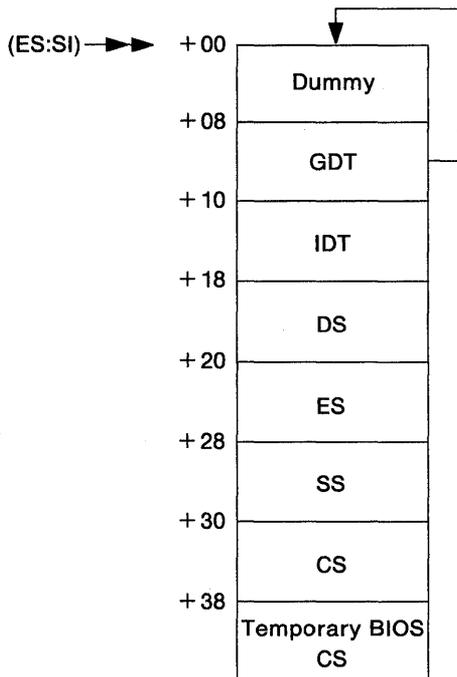


Figure 2-11. Global Descriptor Table

Each descriptor must contain the limit, the base address, and the access rights byte. The descriptors are defined as follows:

- The first is the required dummy and is user initialized to 0.
- The second points to the GDT as a data segment and is user initialized.
- The third points to the user-defined interrupt descriptor table and is user initialized.
- The fourth points to the user data segment (DS) and is user initialized.
- The fifth points to the user extra segment (ES) and is user initialized.
- The sixth points to the user stack segment (SS) and is user initialized.

- The seventh points to the code segment that this function returns to the user, initialized to the user code segment.
- The eighth is used to establish a code segment for itself. This is necessary for this function to complete its operation while in the protected mode. When control is passed to the user code, this descriptor can be reused.

(AH) = 89H  
 (ES:SI) - Location of GDT built by routine using this function.

On Return:  
 (AH) = 00H - Operation successfully completed

All segment registers are changed, (AX) and (BP) are modified.

### Considerations:

1. BIOS functions are not available to the user. The user must handle all I/O commands.
2. Interrupt vector locations must be moved, due to the 80286 reserved areas.
3. The hardware interrupt controllers must be reinitialized to define locations that do not reside in the 80286 reserved areas.
4. An exception interrupt table and handler must be initialized by the user.
5. The interrupt descriptor table cannot overlap the real mode BIOS interrupt descriptor table.

The following is an example of a way to call the protected (virtual address) mode:

```
- User code -
MOV  AX,GDT SEGMENT
MOV  ES,AX
MOV  SI,GDT OFFSET
MOV  BH,HARDWARE INT LEVEL 1 OFFSET
MOV  BL,HARDWARE INT LEVEL 2 OFFSET
MOV  AH,89H
INT  15H
- User code -
(Protected mode established)
```

**(AH) = 8AH to 8FH - Reserved**

**(AH) = 90H - Device Busy**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC XT BIOS dated 11/08/82:

On Return:  
(AH) = 86H  
CF = 1

For all others, this function is called to tell the operating system that the system is about to wait for a device.

The type code assignments for (AH) = 90H and 91H use the following general guidelines:

**00H to 7FH:** Serially reusable devices (operating system must serialize access).

**80H to BFH:** Reentrant devices; (ES:BX) is used to distinguish different calls (multiple I/O calls are allowed simultaneously).

**C0H to FFH:** Wait only calls; there is no complementary Post for these waits. These are time-out only. Times are function number dependent.

(AL) - Type code  
= 00H - Disk (time-out)  
= 01H - Diskette (time-out)  
= 02H - Keyboard (no time-out)  
= 03H - Pointing device (time-out)  
= 80H - Network (no time-out)  
(ES:BX) = Network control block (NCB)  
= FCH - Fixed disk reset for Personal System/2 products only (time-out)  
= FDH - Diskette drive motor start (time-out)  
= FEH - Printer (time-out)

On Return:

CF = 0 - Wait not satisfied  
= 1 - Minimum wait time satisfied for  
this type code

**(AH) = 91H - Interrupt Complete**

For PCjr and PC:

On Return:

(AH) = 80H  
CF = 1

For PC XT BIOS dated 11/08/82:

On Return:

(AH) = 86H  
CF = 1

For all others the interrupt complete flag is set to tell the operating system that the interrupt has occurred.

(AL) - Type code [see (AH) = 90H (Device Busy)]

**(AH) = 92H to BFH - Reserved**

**(AH) = C0H - Return System Configuration Parameters**

For PCjr and PC:

On Return:

(AH) = 80H  
CF = 1

For PC XT BIOS dated 11/08/82 and AT BIOS dated 1/10/84:

On Return:

(AH) = 86H  
CF = 1

For AT BIOS dated 6/10/85 and after, PC XT BIOS dated 1/10/86 and after, PC XT Model 286, PC Convertible, and Personal System/2 products:

(AH) = COH

On Return:

(ES:BX) - Pointer to system descriptor vector in ROM

(AH) = 0

CF = 0

System Descriptor:

DW	XXXX	Byte count of data that follows; minimum length = 8
DB	XX	Model byte See "System Identification" on page 4-18
DB	XX	Submodel byte See "System Identification" on page 4-18
DB	XX	BIOS revision level See "System Identification" on page 4-18 00 = First release Revision level is increased by one for each subsequent release of code
DB	XX	Feature information byte 1  Bit 7 = 1 - Fixed disk BIOS uses DMA channel 3 = 0 - DMA channel 3 not used by fixed disk BIOS or channel 3 usage cannot be determined  Bit 6 = 1 - 2nd interrupt chip present = 0 - 2nd interrupt chip not present  Bit 5 = 1 - Real-time clock present = 0 - Real-time clock not present

Bit 4 = 1 - Keyboard intercept sequence  
(INT 15H) called in keyboard  
interrupt (INT 09H)  
= 0 - Keyboard intercept sequence not  
called

Bit 3 = 1 - Wait for external event  
supported  
= 0 - Wait for external event not  
supported

Bit 2 = 1 - Extended BIOS data area is  
allocated  
= 0 - Extended BIOS data area is  
not allocated

Bit 1 = 1 - Micro Channel implemented  
= 0 - PC type I/O channel implemented

Bit 0 - Reserved

DB	XX	Feature information byte 2 - Reserved
DB	XX	Feature information byte 3 - Reserved
DB	XX	Feature information byte 4 - Reserved
DB	XX	Feature information byte 5 - Reserved

**Note:** For Personal System/2 products except Model 30, if the system model cannot be determined, (AH) = 86H, CF = 1, and (ES:BX) is not changed.

**(AH) = C1H - Return Extended BIOS Data Area Segment Address**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

**For PC XT, AT, PC XT Model 286, and PC Convertible:**

On Return:  
(AH) = 86H  
CF = 1

**For all others:**

On Return:  
(ES) - Extended BIOS data area segment address  
CF = 0 - No error  
= 1 - Error

**(AH) = C2H - Pointing Device BIOS Interface**

**For PCjr and PC:**

On Return:  
(AH) = 80H  
CF = 1

**For PC XT, AT, PC XT Model 286, and PC Convertible:**

On Return:  
(AH) = 86H  
CF = 1

**For all others:**

(AL) = 00H - Enable/disable pointing device  
(BH) = 00H - Disable  
= 01H - Enable

On Return:  
CF = 0 - Operation successfully completed  
CF = 1 - Operation unsuccessful  
(AH) - Status  
= 00H - No error  
= 01H - Invalid function call  
= 02H - Invalid input  
= 03H - Interface error  
= 04H - Resend  
= 05H - No far call installed

(AL) = 01H - Reset pointing device

On Return:

See Return for (AL) = 00H on page 2-97

If the operation successfully completed:

(BH) - Device ID  
= 00H

The pointing device state is as follows:

- Disabled
- Sample rate at 100 reports per second
- Resolution at 4 counts per millimeter
- Scaling at 1 to 1
- Data package size remains the same as before this function was called

(BL) is modified on return

(AL) = 02H - Set sample rate

(BH) - Sample rate value

- = 00H - 10 reports per second
- = 01H - 20 reports per second
- = 02H - 40 reports per second
- = 03H - 60 reports per second
- = 04H - 80 reports per second
- = 05H - 100 reports per second
- = 06H - 200 reports per second

On Return:

See Return for (AL) = 00H on page 2-97

(AL) = 03H - Set resolution

(BH) - Resolution value

- = 00H - 1 count per millimeter
- = 01H - 2 counts per millimeter
- = 02H - 4 counts per millimeter
- = 03H - 8 counts per millimeter

On Return:

See Return for (AL) = 00H on page 2-97

(AL) = 04H - Read device type

On Return:

See Return for (AL) = 00H on page 2-97

If the operation successfully completed:

(BH) - Device ID  
= 00H

(AL) = 05H - Pointing device interface initialization

(BH) - Data package size  
= 00H - Reserved  
= 01H - 1 byte  
= 02H - 2 bytes  
= 03H - 3 bytes  
= 04H - 4 bytes  
= 05H - 5 bytes  
= 06H - 6 bytes  
= 07H - 7 bytes  
= 08H - 8 bytes

On Return:

See Return for (AL) = 00H on page 2-97

The pointing device state is as follows:

-Disabled  
-Sample rate at 100 reports per second  
-Resolution at 4 counts per millimeter  
-Scaling at 1 to 1

(AL) = 06H - Extended commands

(BH) = 00H - Return status

On Return:

See Return for (AL) = 00H on page 2-97

If the operation successfully completed:

(BL) - Status byte 1  
Bit 7 = 0 - Reserved  
Bit 6 = 0 - Stream mode  
= 1 - Remote mode  
Bit 5 = 0 - Disable  
= 1 - Enable  
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  
= 00H - 1 count per millimeter  
= 01H - 2 counts per millimeter  
= 02H - 4 counts per millimeter  
= 03H - 8 counts per millimeter

(DL) - Status byte 3  
= 0AH - 10 reports per second  
= 14H - 20 reports per second  
= 28H - 40 reports per second  
= 3CH - 60 reports per second  
= 50H - 80 reports per second  
= 64H - 100 reports per second  
= C8H - 200 reports per second

(BH) = 01H - Set scaling to 1:1

On Return:

See Return for (AL) = 00H on page 2-97

(BH) = 02H - Set scaling to 2:1

On Return:

See Return for (AL) = 00H on page 2-97

(AL) = 07H - Device driver far call initialization  
(ES) - Segment  
(BX) - Offset

On Return:

See Return for (AL) = 00H on page 2-97

**The user codes a routine to receive control upon pointing device data availability. The device driver far call initialization communicates the address of this routine to the BIOS. Each time the pointing device data is available, the pointing device interrupt handler calls the user routine, with the following parameters on the stack:**

Status - First word pushed on the stack  
X data - Second word pushed on the stack  
Y data - Third word pushed on the stack  
Z data - Fourth word pushed on the stack

Word 1 on stack:  
Low byte - Status  
Bit 7 - Y data overflow  
= 1 - Overflow  
  
Bit 6 - X data overflow  
= 1 - Overflow  
  
Bit 5 - Y data sign  
= 1 - Negative  
  
Bit 4 - X data sign  
= 1 - Negative  
  
Bit 3 - Reserved (must be 1)  
Bit 2 - Reserved (must be 0)  
  
Bit 1 - Right button status  
= 1 - Pressed  
  
Bit 0 - Left button status  
= 1 - Pressed  
High byte = 0

Word 2 on stack:  
Low byte - X data  
Bit 7 = Most significant bit  
Bit 0 - Least significant bit  
High byte = 0

Word 3 on stack:  
Low byte - Y data  
Bit 7 = Most significant bit  
Bit 0 - Least significant bit  
High byte = 0

Word 4 on stack:  
High byte = 0  
Low byte = 0

The pointing device interrupt handler uses a far call to transfer control to the user routine. This routine should be coded as a far procedure and should not pop the parameters off the stack before returning.

**(AH) = C3H - Enable/Disable Watchdog Time-Out**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC XT, AT, PC XT Model 286, PC Convertible, and Personal System/2 Model 30:

On Return:  
(AH) = 86H  
CF = 1

For Personal System/2 products except Model 30:

(AL) = 00H - Disable watchdog time-out  
      = 01H - Enable watchdog time-out  
(BX) - Watchdog timer count  
      (1 to 255 is valid for Personal System/2 products)

On Return:  
CF = 0 - Operation successfully completed  
      = 1 - Operation unsuccessful

**(AH) = C4H - Programmable Option Select (POS)**

For PCjr and PC:

On Return:  
(AH) = 80H  
CF = 1

For PC XT, AT, PC XT Model 286, PC Convertible, and Personal System/2 Model 30:

On Return:  
(AH) = 86H  
CF = 1

## For Personal System/2 products except Model 30:

(AL) = 00H - Return base POS adapter register address

On Return:

(AL) = 00H

(DX) - Base POS adapter register address

(AL) = 01H - Enable slot for setup

(BL) - Slot number

On Return:

(AL) = 01H

(BL) - Slot number

(AL) = 02H - Adapter enable

On Return:

(AL) = 02H

On Return for all:

CF = 0 - Operation successfully completed

= 1 - Request failed

**(AH) = C5H to FFH - Reserved**

---

## Interrupt 16H - Keyboard

These routines provide keyboard support. The following is a summary of the keyboard functions of Interrupt 16H:

(AH) = 00H - Keyboard Read
(AH) = 01H - Keystroke Status
(AH) = 02H - Shift Status
(AH) = 03H - Set Typematic Rate
(AH) = 04H - Keyboard Click Adjustment
(AH) = 05H - Keyboard Write
(AH) = 06H to 0FH - Reserved
(AH) = 10H - Extended Keyboard Read
(AH) = 11H - Extended Keystroke Status
(AH) = 12H - Extended Shift Status
(AH) = 13H to FFH - Reserved

Figure 2-12. INT 16H - Keyboard Functions

The extended functions, (AH) = 10H, 11H, and 12H, have been added to the BIOS interface to support the 101/102-Key Keyboard. The extended-function keyboard scan codes fall into one of three categories:

1. When only one key produces an ASCII character, the scan code read from the keyboard port is the same as with the standard keyboards.
2. When more than one key produces the same character, one of the keys generates the standard keyboard scan code. The other key generates a unique sequence of scan codes, enabling the system to differentiate between the keys.
3. New scan codes are assigned to keys that did not exist on the standard keyboards.

The extended functions allow new programs to take advantage of all categories and avoid compatibility problems with existing programs.

If the extended functions are not supported by the system BIOS, the scan code/character code combination placed in the keyboard buffer by the keyboard interrupt handler are returned without change upon calling (AH) = 00H (Keyboard Read) and (AH) = 01 (Keystroke Status).

if the extended functions are supported by the system BIOS:

- The character code placed in the keyboard buffer by the keyboard interrupt handler differentiates between keys with identical nomenclature.
- The keyboard interrupt handler places the scan code/character code combination for new keys in the keyboard buffer.
- (AH) = 10H (Extended Keyboard Read) and (AH) = 11H (Extended Keystroke Status) extract the scan code/character code combination from the buffer as is, and return it to the caller. The scan code/character code combination is returned for new keys. The scan code/character code combination is returned for like keys, with the character code used to differentiate between them. If the character code is equal to hex F0H and the scan code is not equal to hex 00H, the character code is set to hex 00H.
- (AH) = 00H (Keyboard Read) and (AH) = 01H (Keystroke Status) extract the scan code/character code combination and translate it, if necessary, to the the scan code/character code combination compatible with previous keyboards. The translation:
  1. Converts like codes to compatible codes
  2. Extracts the scan code/character code combination until a compatible combination is found.
- (AH) = 12H (Extended Shift Status) returns the existing keyboard shift state and the shift state of the separate Ctrl and Alt keys.

To determine if the extended functions, (AH) = 10H, 11H, and 12H, are supported by the system BIOS, the program must use INT 16H, (AH) = 05H (Keyboard Write) to write an scan code/character code combination of hex FFFF to the buffer. If on return (AL) = 00H, the function successfully inserted hex FFFF into the buffer. Next, INT 16H, (AH) = 10H (Extended Keyboard Read) is issued to read the scan code/character code combination from the keyboard buffer. If on return (AX) is hex FFFF, the extended keyboard functions are supported. If on return (AX) is not hex FFFF, INT 16H, (AH) = 10H (Extended Keyboard Read) is issued until (AX) is hex FFFF on return. If after 16 tries (the buffer size) or each of the calls to the Extended Keyboard Read function yields an (AX) not equal to hex FFFF, the extended keyboard functions are not supported.

See "Scan Code/Character Code Combinations" on page 4-24 for scan code/character code combinations.

### **(AH) = 00H - Keyboard Read**

The scan code/character code is extracted from the buffer. The keyboard buffer head pointer (40:1A) is increased by 2 or, if the pointer is already at the end, is reinitialized to the start of the buffer.

On Return:

(AL) - ASCII character code

(AH) - Scan code

For AT, PC XT Model 286, PC Convertible, and Personal System/2 products, if no keystroke is available, INT 15H, (AH) = 90 (Device Busy) is called with (AH) = 02H (Type = Keyboard), to inform the operating system that a keyboard loop is about to take place, allowing the operating system to perform another task. When the keyboard operation is completed, INT 09H calls INT 15H, (AH) = 91H (Interrupt Complete) with (AH) = 02H (Type = Keyboard). See "Multitasking Provisions" on page 4-16 for additional information.

**Note:** Control is returned only when a keystroke is available. The keystroke is removed from buffer.

### **(AH) = 01H - Keystroke Status**

On Return:

ZF = 1 - No code available

= 0 - Code is available

If code is available:

(AL) - ASCII character code

(AH) - Scan code

**Note:** The keystroke is not removed from the buffer.

## **(AH) = 02H - Shift Status**

On Return:

- (AL) - Current shift status
  - Bit 7 = 1 - Insert locked
  - Bit 6 = 1 - Caps Lock locked
  - Bit 5 = 1 - Num Lock locked
  - Bit 4 = 1 - Scroll Lock locked
  - Bit 3 = 1 - Alt key pressed
  - Bit 2 = 1 - Ctrl key pressed
  - Bit 1 = 1 - Left Shift key pressed
  - Bit 0 = 1 - Right Shift key pressed
- (AH) - Reserved

## **(AH) = 03H - Set Typematic Rate**

For PCjr only:

- (AL) = 00H - Returns to default, restores original state (typematic on, normal initial delay and normal typematic rate)
- (AL) = 01H - Increases initial delay (this is the delay between first character and the burst of typematic characters)
- (AL) = 02H - Slows typematic characters by one-half
- (AL) = 03H - Increases initial delay and slows typematic characters by one-half
- (AL) = 04H - Turns off typematic characters

For AT BIOS dated 11/15/85 and after, PC XT Model 286, and Personal System/2 products:

(AL) = 05H - Set typematic rate and delay

(BL) - Typematic rate (in characters per second)

00H = 30.0	0BH = 10.9	16H = 4.3
01H = 26.7	0CH = 10.0	17H = 4.0
02H = 24.0	0DH = 9.2	18H = 3.7
03H = 21.8	0EH = 8.6	19H = 3.3
04H = 20.0	0FH = 8.0	1AH = 3.0
05H = 18.5	10H = 7.5	1BH = 2.7
06H = 17.1	11H = 6.7	1CH = 2.5
07H = 16.0	12H = 6.0	1DH = 2.3
08H = 15.0	13H = 5.5	1EH = 2.1
09H = 13.3	14H = 5.0	1FH = 2.0
0AH = 12.0	15H = 4.6	20H to FFH - Reserved

(BH) - Delay value (in milliseconds)

00H = 250
01H = 500
02H = 750
03H = 1000
04H to FFH - Reserved

For all others no action is performed.

### **(AH) = 04H - Keyboard Click Adjustment**

For PCjr and PC Convertible:

(AL) = 00H - Set keyboard click off

(AL) = 01H - Set keyboard click on

For all others no action is performed.

### **(AH) = 05H - Keyboard Write**

For AT BIOS dated 11/15/85 and after, PC XT dated 1/10/86 and after, PC XT Model 286, and Personal System/2 products, this function places scan code/character code combination in the keyboard buffer as if they came from the keyboard.

(CL) - ASCII character code  
(CH) - Scan code

On Return:

(AL) = 00H - Operation successfully completed  
= 01H - Buffer full

For all others no action is performed.

**(AH) = 06H to 0FH - Reserved**

**(AH) = 10H - Extended Keyboard Read**

For AT BIOS dated 11/15/85 and after, PC XT dated 1/10/86 and after, PC XT Model 286, and Personal System/2 products, the scan code/character code combination is extracted from the buffer. The keyboard buffer head pointer (40:1A) is increased by 2. If the pointer is already at the end, it is reinitialized to the start of the buffer .

On Return:

(AL) - ASCII character code  
(AH) - Scan code

**Note:** Control is returned only when a keystroke is available.  
The keystroke is removed from buffer.

For all others no action is performed.

**(AH) = 11H - Extended Keystroke Status**

For AT BIOS dated 11/15/85 and after, PC XT dated 1/10/86 and after, PC XT Model 286, and Personal System/2 products:

On Return:

ZF = 1 - No code available  
= 0 - Code is available

If code is available:

(AL) - ASCII character code  
(AH) - Scan code

**Note:** The keystroke is not removed from the buffer.

For all others no action is performed.

**(AH) = 12H - Extended Shift Status**

For AT BIOS dated 11/15/85 and after, PC XT dated 1/10/86 and after, PC XT Model 286, and Personal System/2 products:

On Return:

(AL) - Shift status

- Bit 7 = 1 - Insert locked
- Bit 6 = 1 - Caps Lock locked
- Bit 5 = 1 - Num Lock locked
- Bit 4 = 1 - Scroll Lock locked
- Bit 3 = 1 - Alt key pressed
- Bit 2 = 1 - Ctrl key pressed
- Bit 1 = 1 - Left Shift key pressed
- Bit 0 = 1 - Right Shift key pressed

(AH) - Extended shift status

- Bit 7 = 1 - SysRq key pressed
- Bit 6 = 1 - Caps Lock key pressed
- Bit 5 = 1 - Num Lock key pressed
- Bit 4 = 1 - Scroll Lock key pressed
- Bit 3 = 1 - Right Alt key pressed
- Bit 2 = 1 - Right Ctrl key pressed
- Bit 1 = 1 - Left Alt key pressed
- Bit 0 = 1 - Left Ctrl key pressed

For all others no action is performed.

**(AH) = 13H to FFH - Reserved**

---

## Interrupt 17H - Printer

These routines provide printer support. The following is a summary of the printer support functions of Interrupt 17H:

(AH) = 00H - Print Character  
(AH) = 01H - Initialize the Printer Port  
(AH) = 02H - Read Status  
(AH) = 03H to FFH - Reserved

Figure 2-13. INT 17H - Printer Functions

### **(AH) = 00H - Print Character**

(AL) - Character to print  
(DX) - Printer to use (0,1,2); index into the port base address table at 40:08

On Return:

(AH) - Status  
Bit 7 = 1 - Not busy  
Bit 6 = 1 - Acknowledge  
Bit 5 = 1 - Out of paper  
Bit 4 = 1 - Selected  
Bit 3 = 1 - I/O error  
Bits 2, 1 - Reserved  
Bit 0 = 1 - Time-out

### **(AH) = 01H - Initialize the Printer Port**

(DX) - Printer to use (0,1,2); index into the port base address table at 40:08

On Return:

(AH) - Status  
Bit 7 = 1 - Not busy  
Bit 6 = 1 - Acknowledge  
Bit 5 = 1 - Out of paper  
Bit 4 = 1 - Selected  
Bit 3 = 1 - I/O error  
Bits 2, 1 - Reserved  
Bit 0 - Time-out

## **(AH) = 02H - Read Status**

(DX) - Printer to use (0,1,2); index into the port base address table at 40:08

On Return:

(AH) - Status

- Bit 7 = 1 - Not busy
- Bit 6 = 1 - Acknowledge
- Bit 5 = 1 - Out of paper
- Bit 4 = 1 - Selected
- Bit 3 = 1 - I/O error
- Bits 2, 1 - Reserved
- Bit 0 - Time-out

## **(AH) = 03H to FFH - Reserved**

### **Notes:**

1. For AT, PC XT Model 286, PC Convertible, and Personal System/2 products, when the printer is busy, the BIOS calls INT 15H, (AH) = 90 (Device Busy) with (AL) = FEH (Type = Printer), informing the operating system that a time-out loop is about to begin. See "Multitasking Provisions" on page 4-16 for additional information.
2. For AT BIOS dated before 11/15/85, PCjr, PC, and PC XT BIOS dated 11/08/82, the printer port number associated with (DX) is tested for 0. If found to be 0, no action occurs. If it is non 0, the print operation is performed. The (DX) register is not tested for a valid printer port number at the offset into the printer base address data area at 40:08.
3. For PC XT BIOS dated 1/10/86 and after, if (DX) is greater than 3 or the printer port associated with (DX) is 0, no action is performed and, on return (AH) = 29H.
4. For PC Convertible, if the printer port associated with (DX) is 0, the return is (AH) = 01H. If (DX) is non 0, the print operation is performed. No test is made on (DX) to see if a valid printer port number exists at the offset into the printer base address data area at 40:08.
5. For AT BIOS dated 11/15/85 and PC XT Model 286, if (DX) is greater than 3 or the printer port associated with (DX) is 0, no action is performed and (AH) is returned unchanged.

6. For Personal System/2 products, if (DX) is greater than 2 or the printer port associated with (DX) is 0, no action is performed and (AH) is returned unchanged.

---

## Interrupt 19H - Bootstrap Loader

Track 0, sector 1 is read into segment 0, offset 7C00. Control is then transferred as follows:

(CS) = 0000H

(IP) = 7C00H

(DL) - Drive where bootstrap sector was read

**Note:** If there is a hardware error, control is transferred to the ROM BASIC entry point.

---

## Interrupt 1AH - System-Timer and Real-Time Clock Services

The following is a summary of the system-timer and real-time clock services of Interrupt 1AH:

(AH) = 00H	– Read System-Timer Time Counter
(AH) = 01H	– Set System-Timer Time Counter
(AH) = 02H	– Read Real-Time Clock Time
(AH) = 03H	– Set Real-Time Clock Time
(AH) = 04H	– Read Real-Time Clock Date
(AH) = 05H	– Set Real-Time Clock Date
(AH) = 06H	– Set Real-Time Clock Alarm
(AH) = 07H	– Reset Real-Time Clock Alarm
(AH) = 08H	– Set Real-Time Clock Activated Power-On Mode
(AH) = 09H	– Read Real-Time Clock Alarm Time and Status
(AH) = 0AH	– Read System-Timer Day Counter
(AH) = 0BH	– Set System-Timer Day Counter
(AH) = 0CH to 7FH	– Reserved
(AH) = 80H	– Set Up Sound Multiplexer
(AH) = 81H to FFH	– Reserved

Figure 2-14. INT 1AH - System-Timer and Real-Time Clock Services

### (AH) = 00H - Read System-Timer Time Counter

On Return:

(CX) - High portion of count

(DX) - Low portion of count

(AL) = 0 - Timer has not passed 24 hours worth of counts since power-on, last system reset, last system-timer time counter read, or last system-timer time counter set.

> 0 - Timer has passed 24 hours worth of counts since power-on, last system reset, last system-timer time counter read, or last system-timer time counter set.

**Note:** Execution causes the timer overflow (40:70) flag to be reset to 0. Counts occur at the rate of  $1193180 \div 65536$  counts per second (about 18.2 per second).

## **(AH) = 01H - Set System-Timer Time Counter**

(CX) - High portion of count

(DX) - Low portion of count

**Note:** Execution causes the timer overflow (40:70) flag to be reset to 0. Counts occur at the rate of  $1193180 \div 65536$  counts per second (about 18.2 per second).

## **(AH) = 02H - Read Real-Time Clock Time**

For AT BIOS dated before 6/10/85:

On return:

(CH) - Hours in BCD

(CL) - Minutes in BCD

(DH) - Seconds in BCD

CF = 0 - Clock operating

= 1 - Clock not operating

For AT BIOS dated 6/10/85 and after, PC XT Model 286, PC Convertible, and Personal System/2 products:

On Return:

(CH) - Hours in BCD

(CL) - Minutes in BCD

(DH) - Seconds in BCD

(DL) = 01H - Daylight saving time option

= 00H - No daylight saving time option

CF = 0 - Clock operating

= 1 - Clock not operating

For all others no action is performed.

**(AH) = 03H - Set Real-Time Clock Time**

For AT, PC XT Model 286, PC Convertible, and Personal System/2 products:

- (CH) - Hours in BCD
- (CL) - Minutes in BCD
- (DH) - Seconds in BCD
- (DL) = 01H - Daylight saving time option  
= 00H - No daylight saving time option

**Note:** For Personal System/2 Model 30, (DL) is not used.

For all others no action is performed.

**(AH) = 04H - Read Real-Time Clock Date**

For AT, PC XT Model 286, PC Convertible, and Personal System/2 products:

- On Return:
- (CH) - Century in BCD (19 OR 20)
  - (CL) - Year in BCD
  - (DH) - Month in BCD
  - (DL) - Day in BCD

- CF = 0 - Clock operating
- = 1 - Clock not operating

For all others no action is performed.

**(AH) = 05H - Set Real-Time Clock Date**

For AT, PC XT Model 286, PC Convertible, and Personal System/2 products:

- (CH) - Century in BCD (19 OR 20)
- (CL) - Year in BCD
- (DH) - Month in BCD
- (DL) - Day in BCD

For all others no action is performed.

### **(AH) = 06H - Set Real-Time Clock Alarm**

For AT, PC XT Model 286, PC Convertible, and Personal System/2 products:

(CH) - Hours in BCD  
(CL) - Minutes in BCD  
(DH) - Seconds in BCD

On Return:

CF = 0 - Operation successfully completed  
= 1 - Alarm already set or clock not operating

**Note:** The alarm interrupt occurs at the specified hour, minute, and second passed in (CH), (CL), and (DH) respectively. When the alarm interrupt occurs, a software interrupt 4AH is issued. The user must point software interrupt 4AH to an alarm routine prior to setting the real-time clock alarm INT 1AH, (AH) = 06H. Only one alarm function may be active at any time. The alarm interrupt occurs every 24 hours at the specified time until it is reset.

For all others no action is performed.

### **(AH) = 07H - Reset Real-Time Clock Alarm**

For AT, PC XT Model 286, PC Convertible, and Personal System/2 products, this function stops the real-time clock alarm interrupt from occurring.

For all others no action is performed.

### **(AH) = 08H - Set Real-Time Clock Activated Power-On Mode**

For PC Convertible:

(CH) - Hours in BCD  
(CL) - Minutes in BCD  
(DH) - Seconds in BCD

On Return:

CF = 0 - Operation successfully completed  
= 1 - Alarm already set or clock not operating

For AT BIOS dated 6/10/85 and after, PC XT Model 286, and Personal System/2 products:

On Return:

CF = 1 - Invalid function request

For all others no action is performed.

**(AH) = 09H - Read Real-Time Clock Alarm Time and Status**

For PC Convertible and Personal System/2 Model 30:

On Return:

(CH) - Hours in BCD

(CL) - Minutes in BCD

(DH) - Seconds in BCD

(DL) - Alarm status

00H = Alarm not enabled

01H = Alarm enabled but will not power-on system

02H = Alarm enabled and will power-on system

**Note:** Personal System/2 Model 30 does not support the power-on system feature.

For AT BIOS dated 6/10/85 and after, PC XT Model 286, and Personal System/2 products except Model 30:

On Return:

CF = 1 - Invalid function request

For all others no action is performed.

**(AH) = 0AH - Read System-Timer Day Counter**

For AT and PC XT Model 286:

On Return:

CF = 1 - Invalid function request

For PC XT BIOS dated 1/10/86 and after, and Personal System/2 products:

On Return:  
(CX) - Count of days since 1-1-1980

**Note:** For PC XT dated 1/10/86 and after, and Personal System/2 Model 30, the count of days since 1/1/80 is initialized to 0 during the POST.

For all others no action is performed.

**(AH) = 0BH - Set System-Timer Day Counter**

For AT BIOS dated 6/10/85 and after and PC XT Model 286:

On Return:  
CF = 1 - Invalid function request

For PC XT BIOS dated 1/10/86 and after, and Personal System/2 products:

(CX) - Count of days since 1-1-1980

**Note:** For PC XT dated 1/10/86 and after, and Personal System/2 Model 30, the count of days since 1/1/80 is initialized to 0 during the POST.

For all others no action is performed.

**(AH) = 0CH to 7FH - Reserved**

**(AH) = 80H - Set Up Sound Multiplexer**

For PCjr:

(AL) - Source of sound ("Audio Out" or RF modulator)  
00H = 8253 channel 2  
01H = Cassette input  
02H = "Audio In" line on I/O channel  
03H = Complex sound generator chip

For AT BIOS dated 6/10/85 and after, PC XT Model 286, PC Convertible, and Personal System/2 products:

On Return:

CF = 1 - Invalid function request

For all others no action is performed.

**(AH) = 81H to FFH - Reserved**

---

## Interrupt 70H - Real-Time Clock Interrupt

For AT, PC XT Model 286, and Personal System/2 products:

This interrupt handler controls the periodic and alarm interrupt functions from the real-time clock.

**Periodic function** — When activated, the interrupt occurs approximately 1024 times per second. The doubleword microsecond counter is decremented by a value of 976 microseconds (1/1024 of a second). When the counter becomes less than or equal to 0, bit 7 of the designated location is set. For INT 15H, (AH) = 83H (Event Wait), the designated location is provided by the user. For INT 15H, (AH) = 86H (Wait), the designated location is bit 7 of BIOS data area 40:A0 (wait active flag).

**Alarm function** — When activated, the interrupt occurs at the specified time and a software interrupt 4AH is issued. The user must point interrupt 4AH to an alarm routine prior to setting INT 1AH, (AH) = 06H (Real-Time Clock Alarm).

For all others, the Real-Time Clock Interrupt is not supported.

### Notes:

1. The PC Convertible provides the above functions, but the Real-Time Clock Interrupt generates a nonmaskable interrupt rather than INT 70H. Additionally, PC Convertible uses the real-time clock update ended interrupt function (interrupts once per second) when certain system profiles are enabled.
2. For Personal System/2 Model 30, the periodic function is not supported.

---

## Section 3. Data Areas and ROM Tables

BIOS Data Area .....	3-3
Extended BIOS Data Area .....	3-15
ROM Tables .....	3-16
Fixed Disk Drive Parameter Table .....	3-16
Diskette Drive Parameter Table .....	3-23

## Notes:

---

## BIOS Data Area

The BIOS Data Area is allocated specifically as a work area for system BIOS and adapter BIOS. The BIOS routines use 256 bytes of memory from absolute address hex 400 to hex 4FF. A description of the BIOS data area follows:

Address	Function	Size
40:00	RS-232-C Communications Line 1 Port Base Address	Word
40:02	RS-232-C Communications Line 2 Port Base Address	Word
40:04	RS-232-C Communications Line 3 Port Base Address	Word
40:06	RS-232-C Communications Line 4 Port Base Address	Word

**Note:** The RS-232-C communications line port base address fields may be initialized to 0 by the POST if the system configuration contains less than four serial ports. The POST never places 0 in the RS-232-C communications line port base address table between two valid RS-232-C communications line port base addresses.

Figure 3-1. RS-232-C Port Base Address Data Area

Address	Function	Size
40:08	Printer 1 Port Base Address	Word
40:0A	Printer 2 Port Base Address	Word
40:0C	Printer 3 Port Base Address	Word
40:0E	Reserved	Word

Exceptions

40:0E	Printer 4 Port Base Address (PC, PC XT, AT, and PC Convertible)	Word
-------	---	------

**Note:** The printer port base address fields may be initialized to 0 by the POST if the system configuration contains less than four parallel ports. The POST never places 0 in the printer port base address table between two valid printer port base addresses.

Figure 3-2. Printer Port Base Address Data Area

Address	Function	Size
40:10	Installed Hardware	Word
Bits 15,14	Number of Printer Adapters	
Bit 13	Reserved	
Bit 12	Reserved	
Bits 11,10,9	Number of RS-232-C Adapters	
Bit 8	Reserved	
Bits 7,6	Number of Diskette Drives (0-based)	
Bits 5,4	Video Mode Type (Values are Binary)	
	00 = Reserved	
	01 = 40x25 Color	
	10 = 80x25 Color	
	11 = 80x25 Monochrome	
Bit 3	Reserved	
Bit 2	Pointing Device	
Bit 1	Math Coprocessor	
Bit 0	IPL Diskette	
Exceptions		
Bit 13	Internal Modem (PC Convertible Only)	
Bit 2	Reserved (PC, PC XT, AT, and PC Convertible)	
<b>Note:</b> Refer to INT 11H for equipment return information .		

Figure 3-3. System Equipment Data Area

Address	Function	Size
40:12	Reserved	Byte
Exceptions		
40:12	Power-On Self-Test Status (PC Convertible Only)	Byte

Figure 3-4. Miscellaneous Data Area 1

Address	Function	Size
40:13	Memory Size in Kb (Range 0 to 640)	Word
40:15 to 40:16	Reserved	Byte

Figure 3-5. Memory Size Data Area

<b>Address</b>	<b>Function</b>	<b>Size</b>
40:17	Keyboard Control	Byte
Bit 7	Insert Locked	
Bit 6	Caps Lock Locked	
Bit 5	Num Lock Locked	
Bit 4	Scroll Lock Locked	
Bit 3	Alt Key Pressed	
Bit 2	Ctrl Key Pressed	
Bit 1	Left Shift Key Pressed	
Bit 0	Right Shift Key Pressed	
40:18	Keyboard Control	Byte
Bit 7	Insert Key Pressed	
Bit 6	Caps Lock Key Pressed	
Bit 5	Num Lock Key Pressed	
Bit 4	Scroll Lock Key Pressed	
Bit 3	Pause Locked	
Bit 2	System Request Key Pressed	
Bit 1	Left Alt Key Pressed	
Bit 0	Left Ctrl Key Pressed	
40:19	Alternate Keypad Entry	Byte
40:1A	Keyboard Buffer Head Pointer	Word
40:1C	Keyboard Buffer Tail Pointer	Word
40:1E	Keyboard Buffer	32 Bytes

Figure 3-6. Keyboard Data Area 1

Address	Function	Size
40:3E	Recalibrate status	Byte
Bit 7	Interrupt Flag	
Bit 6	Reserved	
Bit 5	Reserved	
Bit 4	Reserved	
Bit 3	Recalibrate Drive 3	
Bit 2	Recalibrate Drive 2	
Bit 1	Recalibrate Drive 1	
Bit 0	Recalibrate Drive 0	
40:3F	Motor Status	Byte
Bit 7	Write/Read Operation	
Bit 6	Reserved	
Bits 5,4	Diskette Drive Select Status (Values in Binary)	
	00 = Diskette Drive 0 Selected	
	01 = Diskette Drive 1 Selected	
	10 = Diskette Drive 2 Selected	
	11 = Diskette Drive 3 Selected	
Bit 3	Diskette Drive 3 Motor On Status	
Bit 2	Diskette Drive 2 Motor On Status	
Bit 1	Diskette Drive 1 Motor On Status	
Bit 0	Diskette Drive 0 Motor On Status	
40:40	Motor off counter	Byte
40:41	Last Diskette Drive Operation Status	Byte
	00H = No Error	
	01H = Invalid Diskette Drive Parameter	
	02H = Address Mark not Found	
	03H = Write-protect Error	
	04H = Requested Sector not Found	
	06H = Diskette Change Line Active	
	08H = DMA Overrun on Operation	
	09H = Attempt to DMA Across a 64Kb Boundary	
	0CH = Media Type not Found	
	10H = CRC Error on Diskette Read	
	20H = General Controller Failure	
	40H = Seek Operation Failed	
	80H = Diskette Drive not Ready	
40:42	Diskette Drive Controller Status Bytes	7 Bytes

Figure 3-7. Diskette Drive Data Area

<b>Address</b>	<b>Function</b>	<b>Size</b>
40:49	Display Mode set	Byte
40:4A	Number of Columns	Word
40:4C	Length of Regen Buffer in Bytes	Word
40:4E	Starting Address in Regen Buffer	Word
40:50	Cursor Position Page 1	Word
40:52	Cursor Position Page 2	Word
40:54	Cursor Position Page 3	Word
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 Type	Word
40:62	Display Page	Byte
40:63	CRT Controller Base Address	Word
40:65	Current Setting of 3x8 Register	Byte
40:66	Current Setting of 3x9 Register	Byte

Figure 3-8. Video Control Data Area 1

<b>Address</b>	<b>Function</b>	<b>Size</b>
40:67	Reserved	DWord
40:6B	Reserved	Byte
Exceptions		
40:67	Pointer to reset code upon system reset with memory preserved (Personal System/2 products except Model 30). Reset Flag at 40:72 = 4321H	DWord

Figure 3-9. System Data Area 1

<b>Address</b>	<b>Function</b>	<b>Size</b>
40:6C	Timer Counter	DWord
40:70	Timer Overflow (If non 0, timer has counted past 24 hours.)	Byte

Figure 3-10. System-Timer Data Area

Address	Function	Size
40:71	Break Key State	Byte
40:72	Reset Flag	Word
	1234H = Bypass Memory Test	
	4321H = Preserve Memory (Personal System/2 products except Model 30)	
	5678H = System Suspended (PC Convertible)	
	9ABCH = Manufacturing Test Mode (PC Convertible)	
	ABCDH = System POST Loop Mode (PC Convertible)	

Figure 3-11. System Data Area 2

Address	Function	Size
40:74	Last Fixed Disk Drive Operation Status	Byte
	00H = No Error	
	01H = Invalid Function Request	
	02H = Address Mark not Found	
	03H = Write Protect Error	
	04H = Sector not Found	
	05H = Reset Failed	
	07H = Drive Parameter Activity Failed	
	08H = DMA Overrun on Operation	
	09H = Data Boundary Error	
	0AH = Bad Sector Flag Detected	
	0BH = Bad Track Detected	
	0DH = Invalid Number of Sectors on Format	
	0EH = Control Data Address Mark Detected	
	0FH = DMA Arbitration Level Out of Range	
	10H = Uncorrectable ECC or CRC Error	
	11H = ECC Corrected Data Error	
	20H = General Controller Failure	
	40H = Seek Operation Failed	
	80H = Time Out	
	AAH = Drive not Ready	
	BBH = Undefined Error Occurred	
	CCH = Write Fault on Selected Drive	
	E0H = Status Error/Error Register is 0	
	FFH = Sense Operation Failed	
40:75	Number of Fixed Disk Drives Attached	Byte
40:76	Reserved	Byte
40:77	Reserved	Byte
Exceptions		
40:74	Reserved (IBM ESDI Fixed Disk Drive Adapter/A)	Byte
40:76	Fixed Disk Drive Control (PC XT)	Byte
40:77	Fixed Disk Drive Controller Port (PC XT)	Byte

Figure 3-12. Fixed Disk Drive Data Area

Address	Function	Size
40:78	Printer 1 Time-out Value	Byte
40:79	Printer 2 Time-out Value	Byte
40:7A	Printer 3 Time-out Value	Byte
40:7B	Reserved	
Exceptions		
40:7B	Printer 4 Time-out Value (PC, PC XT, and AT)	Byte

Figure 3-13. Printer Time-Out Value Data Area

Address	Function	Size
40:7C	RS-232-C Communications Line 1 Time-out Value	Byte
40:7D	RS-232-C Communications Line 2 Time-out Value	Byte
40:7E	RS-232-C Communications Line 3 Time-out Value	Byte
40:7F	RS-232-C Communications Line 4 Time-out Value	Byte

Figure 3-14. RS-232-C Time-Out Value Data Area

Address	Function	Size
40:80	Keyboard Buffer Start Offset Pointer	Word
40:82	Keyboard Buffer End Offset Pointer	Word

Figure 3-15. Keyboard Data Area 2

Address	Function	Size
40:84	Number of Rows on the Screen (Minus 1)	Byte
40:85	Character Height (Bytes/Character)	Word
40:87	Video Control States	Byte
40:88	Video Control States	Byte
40:89	Reserved	Byte
40:8A	Reserved	Byte

Figure 3-16. Video Control Data Area 2

Address	Function	Size
40:8B	Media Control	Byte
Bits 7,6	Last Diskette Drive Data Rate Selected (Values in Binary) 00 = 500Kb Per Second 01 = 300Kb Per Second 10 = 250Kb Per Second 11 = Reserved	
Bits 5,4	Last Diskette Drive Step Rate Selected	
Bit 3	Reserved	
Bit 2	Reserved	
Bit 1	Reserved	
Bit 0	Reserved	
40:8C	Fixed Disk Drive Controller Status	Byte
40:8D	Fixed Disk Drive Controller Error Status	Byte
40:8E	Fixed Disk Drive Interrupt Control	Byte
40:8F	Reserved	Byte
40:90	Drive 0 Media State	Byte
40:91	Drive 1 Media State	Byte
Bits 7,6	Diskette Drive Data Rate (Values in Binary) 00 = 500Kb Per Second 01 = 300Kb Per Second 10 = 250Kb Per Second 11 = Reserved	
Bit 5	Double Stepping Required	
Bit 4	Media Established	
Bit 3	Reserved	
Bits 2,1,0	Drive/Media State (Values in Binary) 000 = 360Kb Diskette/360Kb Drive not Established 001 = 360Kb Diskette/1.2Mb Drive not Established 010 = 1.2Mb Diskette/1.2Mb Drive not Established 011 = 360Kb Diskette/360Kb Drive Established 100 = 360Kb Diskette/1.2Mb Drive Established 101 = 1.2Mb Diskette/1.2Mb Drive Established 110 = Reserved 111 = None of the Above	
40:92	Reserved	Byte
40:93	Reserved	Byte
40:94	Drive 0 Current Cylinder	Byte
40:95	Drive 1 Current Cylinder	Byte
Exceptions		
40:8B to 40:95	Reserved (PC, PCjr, PC XT BIOS Dated 11/8/82, and PC Convertible)	Byte

Figure 3-17. Diskette Drive/Fixed Disk Drive Control Data Area

<b>Address</b>	<b>Function</b>	<b>Size</b>
40:96	Keyboard Mode State and Type Flags	Byte
Bit 7	Read ID in Progress	
Bit 6	Last Character was First ID Character	
Bit 5	Force Num Lock if Read ID and KBX	
Bit 4	101/102-Key Keyboard Installed	
Bit 3	Right Alt Key Pressed	
Bit 2	Right Ctrl Key Pressed	
Bit 1	Last Code was E0 Hidden Code	
Bit 0	Last Code was E1 Hidden Code	
40:97	Keyboard LED Flags	Byte
Bit 7	Keyboard Transmit Error Flag	
Bit 6	Mode Indicator Update	
Bit 5	Resend Receive Flag	
Bit 4	Acknowledgment Received	
Bit 3	Reserved (Must be 0)	
Bits 2,1,0	Keyboard LED State Bits	

Figure 3-18. Keyboard Data Area 3

<b>Address</b>	<b>Function</b>	<b>Size</b>
40:98	Offset Address to User Wait Complete Flag	Word
40:9A	Segment Address to User Wait Complete Flag	Word
40:9C	User Wait Count - Low Word (Microseconds)	Word
40:9E	User Wait Count - High Word (Microseconds)	Word
40:A0	Wait Active Flag	Byte
Bit 7	Wait Time Elapsed and Post	
Bits 6 to 1	Reserved	
Bit 0	INT 15H, AH = 86H (Wait) has Occurred	
40:A1 to 40:A7	Reserved	Byte

Figure 3-19. Real-Time Clock Data Area

For systems with EGA capability and Personal System/2 products, the save pointer table contains pointers that define specific dynamic overrides for the video mode set function, INT 10H, (AH) = 00H.

Address	Function	Size
40:A8	Pointer to Video Parameters and Overrides	DWord
DWord 1	Video Parameter Table Pointer  Initialized to the BIOS video parameter table. This value must contain a valid pointer.	
DWord 2	Dynamic Save Area Pointer (except Personal System/2 Model 30)  Initialized to 00:00, this value is optional. When non 0, this value points to an area in RAM where certain dynamic values are saved. This area holds the 16 EGA palette register values plus the overscan value in bytes (0-16), respectively. A minimum of 256 bytes must be allocated for this area.	
DWord 3	Alpha Mode Auxiliary Character Generator Pointer  Initialized to 00:00, this value is optional. When non 0, this value points to a table that is described as follows:	
	Bytes/Character	Byte
	Block to Load, 0 = Normal Operation	Byte
	Count to Store, 256 = Normal Operation	Word
	Character Offset, 0 = Normal Operation	Word
	Pointer to a Font Table	DWord
	Displayable Rows If 0FFH, the maximum calculated value is used, otherwise this value is used.	Byte
	Consecutive bytes of mode values for this font description. The end of this stream is indicated by a byte code of 0FFH.	Byte
<b>Note:</b>	Use of the DWord 3 pointer may cause unexpected cursor type operation. For an explanation of cursor type, see INT 10H, (AH) = 01H.	
DWord 4	Graphics Mode Auxiliary Character Generator Pointer  Initialized to 00:00, this value is optional. When non 0, this value points to a table that is described as follows:	

Figure 3-20 (Part 1 of 2). Save Pointer Data Area

Address	Function	Size
	Displayable Rows	Byte
	Bytes Per Character	Word
	Pointer to a Font Table	DWord
	Consecutive bytes of mode values for this font description. The end of this stream is indicated by a byte code of 0FFH.	Byte
DWord 5	Secondary Save Pointer (except EGA and Personal System/2 Model 30)	
	Initialized to the BIOS secondary save pointer. This value must contain a valid pointer.	
DWord 6	Reserved and set to 00:00.	
DWord 7	Reserved and set to 00:00.	

Figure 3-20 (Part 2 of 2). Save Pointer Data Area

Address	Function	Size
Word 1	Table Length	
	Initialized to the BIOS secondary save pointer table length.	
DWord 2	Display Combination Code (DCC) Table Pointer	
	Initialized to ROM DCC table. This value must exist. It points to a table described as follows:	
	Number of Entries in Table	Byte
	DCC Table Version Number	Byte
	Maximum Display Type Code	Byte
	Reserved	Byte
	00,00 Entry 0 No Displays	
	00,01 Entry 1 MDPA	
	00,02 Entry 2 CGA	
	02,01 Entry 3 MDPA + CGA	
	00,04 Entry 4 EGA	
	04,01 Entry 5 EGA + MDPA	
	00,05 Entry 6 MEGA	
	02,05 Entry 7 MEGA + CGA	
	00,06 Entry 8 PGC	
	01,06 Entry 9 PGC + MDPA	
	05,06 Entry 10 PGC + MEGA	
	00,08 Entry 11 CVGA	

Figure 3-21 (Part 1 of 3). Secondary Save Pointer Data Area

Address	Function	Size
	01,08 Entry 12 CVGA + MDPA	
	00,07 Entry 13 MVGA	
	02,07 Entry 14 MVGA + CGA	
	02,06 Entry 15 MVGA + PGC	
	Abbreviation Meanings:	
	MDPA = Monochrome Display and Printer Adapter	
	CGA = Color/Graphics Monitor Adapter	
	EGA = Enhanced Graphics Adapter	
	MEGA = EGA with monochrome display	
	PGC = Professional Graphics Controller	
	VGA = Video Graphics Array	
	MVGA = VGA based with monochrome display	
	CVGA = VGA based with color display	
DWord 3	Second Alpha Mode Auxiliary Character Generator Pointer	
	Initialized to 00:00, this value is optional. When non 0, this value points to a table that is described as follows:	
	Bytes/Character	Byte
	Block to load, should be non 0 for normal operation.	Byte
	Reserved	Byte
	Pointer to a Font Table	DWord
	Consecutive bytes of mode values for this font description. The end of this stream is indicated by a byte code of 0FFH.	Byte
<b>Note:</b> Attribute bit 3 is used to switch between primary and secondary fonts. It may be desirable to use the user palette profile to define a palette of consistent colors independent of attribute bit 3.		
DWord 4	User Palette Profile Table Pointer	
	Initialized to 00:00, this value is optional. When non 0, this value points to a table that is described as follows:	
	Underlining flag (1 = On, 0 = Ignore, -1 = Off; 0 = Normal Operation)	Byte
	Reserved	Byte
	Reserved	Word
	Internal Palette Count (0-17; 17 = Normal Operation)	Word

Figure 3-21 (Part 2 of 3). Secondary Save Pointer Data Area

Address	Function	Size
	Internal Palette Index (0-16; 0 = Normal Operation)	Word
	Pointer to Internal Palette	DWord
	External Palette Count (0-256; 256 = Normal Operation)	Word
	External Palette Index (0-255; 0 = Normal Operation)	Word
	Pointer to External Palette	DWord
	Consecutive bytes of mode values for this font description. The end of this stream is indicated by a byte code of 0FFH.	Byte
DWord 5 to DWord 7	Reserved and set to 00:00.	

Figure 3-21 (Part 3 of 3). Secondary Save Pointer Data Area

Address	Function	Size
40:AC to 40:FF	Reserved	Byte
50:00	Print Screen Status Byte (INT 05H Status)	Word

Figure 3-22. Miscellaneous Data Area 2

## Extended BIOS Data Area

The Extended BIOS Data Area is supported on Personal System/2 products only. The POST allocates the highest possible (n) Kb of memory below 640Kb to be used as this data area. The word value at 40:13 (memory size), indicating the number of Kb below the 640Kb limit, is decremented by (n). The first byte in the Extended BIOS Data Area is initialized to the length in Kb of the allocated area.

To access the Extended BIOS Data Area segment, issue an INT 15, (AH) = C1H (Return Extended BIOS Data Area Segment Address). To determine if an Extended BIOS Data Area is allocated, use INT 15, (AH) = C0H (Return System Configuration Parameters).

---

## ROM Tables

The following ROM tables are used by the BIOS to define the characteristics of the hardware devices supported by the system or adapter BIOS.

### Fixed Disk Drive Parameter Table

The fixed disk drive parameter table is defined as follows:

Offset	Length	Description
0	1 Word	Maximum Number of Cylinders
2	1 Byte	Maximum Number of Heads
3	1 Word	For PC XT: Starting Reduced Write Current Cylinder All Others: Not Used
5	1 Word	Starting Write Precompensation Cylinder
7	1 Byte	For PC XT: Maximum ECC Data Burst Length All Others: Not Used
8	1 Byte	Control Byte For PC XT: Bit 7 - Disable Disk-access Retries Bit 6 - Disable ECC Retries Bits 5 to 3 = 0 Bits 2,1,0 - Drive Option All Others: Bit 7 - Disable Retries -or- Bit 6 - Disable Retries Bit 5 - Manufacturer's Defect Map Present at Maximum Cylinders + 1 Bit 3 - More than 8 Heads
9	1 Byte	For PC XT: Standard Time-out Value All Others: Not Used
10	1 Byte	For PC XT: Time-out Value For Format Drive All Others: Not Used
11	1 Byte	For PC XT: Time-out Value For Check Drive All Others: Not Used
12	1 Word	For PC XT: Reserved All Others: Landing Zone
14	1 Byte	For PC XT: Reserved
15	1 Byte	All others: Number of Sectors Per Track Reserved

Figure 3-23. Fixed Disk Drive Parameter Table Definition

For AT and Personal System/2 products, the following lists the fixed disk drive parameters for the various fixed disk drive types. Values are decimal unless noted otherwise:

Type	Number of Cylinders	Number of Heads	Number Write Precompensation	Landing Zone	Defect Map
0		—No fixed disk drive installed—			
1	306	4	128	305	No
2	615	4	300	615	No
3	615	6	300	615	No
4	940	8	512	940	No
5	940	6	512	940	No
6	615	4	0FFFFH (None)	615	No
7	462	8	256	511	No
8	733	5	0FFFFH (None)	733	No
9	900	15	0FFFFH (None)	901	No
10	820	3	0FFFFH (None)	820	No
11	855	5	0FFFFH (None)	855	No
12	855	7	0FFFFH (None)	855	No
13	306	8	128	319	No
14	733	7	0FFFFH (None)	733	No
15		—Reserved—			
16	612	4	0 (All Cylinders)	663	No
17	977	5	300	977	No
18	977	7	0FFFFH (None)	977	No
19	1024	7	512	1023	No
20	733	5	300	732	No
21	733	7	300	732	No
22	733	5	300	733	No
23	306	4	0 (All Cylinders)	336	No
24	612	4	305	663	No
25	306	4	0FFFFH (None)	340	No
26	612	4	0FFFFH (None)	670	No
27	698	7	300	732	Yes
28	976	5	488	977	Yes
29	306	4	0 (All Cylinders)	340	No
30	611	4	306	663	Yes
31	732	7	300	732	Yes
32	1023	5	0FFFFH(None)	1023	Yes
33 to 255		—Reserved—			

Figure 3-24. Fixed Disk Drive Parameters (AT and Personal System/2 Products)

**Notes:**

1. Software interrupt 41H points to the entry in the table for drive 0. Software interrupt 46H points to the entry in the table for drive 1.
2. AT BIOS dated 1/10/84 contains entries 0 through 14.
3. AT BIOS dated 6/10/85 or 11/15/85 contains entries 0 through 23.

4. PC XT Model 286 contains entries 0 through 24.
5. Personal System/2 products except Model 30 contain entries 0 through 32.
6. Personal System/2 Model 30 contains entries 0 through 26.

For Personal System/2 products except Model 30, the following fixed disk drive parameter table applies:

Offset	Length	Value	Description
0	2	41	Length of Fixed Disk Drive Table
2	22	(ID)	ASCII string 'IBM HARDFILE TYPE xxx', where xxx is the type number in ASCII.
24	1	yyy	Type Number (Values are Binary)
25	2	*	Maximum Number of Cylinders
27	1	*	Maximum Number of Heads
28	2	0	Reserved
30	2	*	Start Write Precompensation Cylinder
32	1	0	Reserved
33	1	*	Control Byte
			Bit 7 or 6 - Disable Retries
			Bit 5 - Defect Map Installed
			Bit 3 - More Than 8 Heads (AT Only)
34	3	0	Reserved
37	2	*	Landing Zone
39	1	*	Number of Sectors Per Track
40	1	0	Reserved

Figure 3-25. Fixed Disk Drive Parameter Table (Personal System/2 Products Except Model 30)

**Note:** This information is located at head 0, track 0, sector 2.

For PC XT BIOS dated 11/10/82 the following fixed disk drive parameter tables apply:

Size	Value	Description
DW	0306	Maximum Cylinders
DB	02	Maximum Heads
DW	0306	Start Reduced Write Current Cylinder
DW	0000	Start Write Precompensation Cylinder
DB	0BH	Maximum ECC Burst Data Length
DB	00H	Control Byte
DB	0CH	Standard Time-out
DB	0B4H	Time-out for Format Drive
DB	028H	Time-out for Check Drive
DB	0,0,0,0	Reserved

Figure 3-26. Fixed Disk Drive Drive Parameter Table 00 (PC XT BIOS Dated 11/10/82)

Size	Value	Description
DW	0375	Maximum Cylinders
DB	08	Maximum Heads
DW	0375	Start Reduced Write Current Cylinder
DW	0000	Start Write Precompensation Cylinder
DB	0bh	Maximum ECC Burst Data Length
DB	05H	Control Byte
DB	0CH	Standard Time-out
DB	0B4H	Time-out for Format Drive
DB	028H	Time-out for Check Drive
DB	0,0,0,0	Reserved

Figure 3-27. Fixed Disk Drive Parameter Table 01 (PC XT BIOS Dated 11/10/82)

Size	Value	Description
DW	0306	Maximum Cylinders
DB	06	Maximum Heads
DW	0128	Start Reduced Write Current Cylinder
DW	0256	Start Write Precompensation Cylinder
DB	0BH	Maximum ECC Burst Data Length
DB	05H	Control Byte
DB	0CH	Standard Time-out
DB	0B4H	Time-out for Format Drive
DB	028H	Time-out for Check Drive
DB	0,0,0,0	Reserved

Figure 3-28. Fixed Disk Drive Parameter Table 02 (PC XT BIOS Dated 11/10/82)

Size	Value	Description
DW	0306	Maximum Cylinders
DB	04	Maximum Heads
DW	0306	Start Reduced Write Current Cylinder
DW	0000	Start Write Precompensation Cylinder
DB	0BH	Maximum ECC Burst Data Length
DB	05H	Control Byte
DB	0CH	Standard Time-out
DB	0B4H	Time-out for Format Drive
DB	028H	Time-out for Check Drive
DB	0,0,0,0	Reserved

Figure 3-29. Fixed Disk Drive Parameter Table 03 (PC XT BIOS Dated 11/10/82)

**Note:** INT 41H points to the beginning of the table and the switch settings on the adapter are used as an index into the table.

For PC XT BIOS dated 1/08/86 and after the following fixed disk drive parameter tables apply:

Size	Value	Description
DW	306	Maximum Cylinders
DB	4	Maximum Heads
DW	306	Start Reduced Write Current Cylinder
DW	0	Start Write Precompensation Cylinder
DB	0BH	Maximum ECC Burst Data Length
DB	05H	Control Byte
DB	0CH	Standard Time-out
DB	0B4H	Time-out for Format Drive
DB	028H	Time-out for Check Drive
DB	0,0,0,0	Reserved

Figure 3-30. Fixed Disk Drive Parameter Table 00 - Type 1 (PC XT BIOS Dated 1/08/86)

Size	Value	Description
DW	612	Maximum Cylinders
DB	4	Maximum Heads
DW	612	Start Reduced Write Current Cylinder
DW	0	Start Write Precompensation Cylinder
DB	0BH	Maximum ECC Burst Data Length
DB	05H	Control Byte
DB	20H	Standard Time-out
DB	0B4H	Time-out for Format Drive
DB	028H	Time-out for Check Drive
DB	0,0,0,0	Reserved

Figure 3-31. Fixed Disk Drive Parameter Table 01 - Type 16 (PC XT BIOS Dated 1/08/86)

Size	Value	Description
DW	615	Maximum Cylinders
DB	4	Maximum Heads
DW	615	Start Reduced Write Current Cylinder
DW	300	Start Write Precompensation Cylinder
DB	0BH	Maximum ECC Burst Data Length
DB	05H	Control Byte
DB	18H	Standard Time-out
DB	0B4H	Time-out for Format Drive
DB	028H	Time-out for Check Drive
DB	0,0,0,0	Reserved

Figure 3-32. Fixed Disk Drive Parameter Table 02 - Type 2 (PC XT BIOS Dated 1/08/86)

Size	Value	Description
DW	306	Maximum Cylinders
DB	8	Maximum Heads
DW	306	Start Reduced Write Current Cylinder
DW	128	Start Write Precompensation Cylinder
DB	0BH	Maximum ECC Burst Data Length
DB	05H	Control Byte
DB	0CH	Standard Time-out
DB	0B4H	Time-out for Format Drive
DB	028H	Time-out for Check Drive
DB	0,0,0,0	Reserved

Figure 3-33. Fixed Disk Drive Parameter Table 03 - Type 13 (PC XT BIOS Dated 1/08/86)

**Note:** INT 41H points to the beginning of the table. The switch settings on the adapter are used as an index into the table.

## Diskette Drive Parameter Table

The diskette drive parameter table is defined as follows:

Offset	Length	Description
0	1 byte	First Specify Byte
1	1 byte	Second Specify Byte
2	1 byte	Number of Timer Ticks to Wait Prior to Turning Diskette Drive Motor Off
3	1 byte	Number of Bytes Per Sector 00H = 128 Bytes Per Sector 01H = 256 Bytes Per Sector 02H = 512 Bytes Per Sector 03H = 1024 Bytes Per Sector
4	1 byte	Sectors Per Track
5	1 byte	Gap Length
6	1 byte	Dtl (Data Length)
7	1 byte	Gap Length for Format
8	1 byte	Fill Byte for Format
9	1 byte	Head Settle Time (Milliseconds)
10	1 byte	Motor Startup Time (1/8 Seconds) For Example: 8 = 1 Second Wait

Figure 3-34. Diskette Drive Parameter Table

**Note:** The diskette drive parameter table is pointed to by INT 1EH.

## Notes:

---

# Section 4. Additional Information

- Interrupt Sharing ..... 4-3
  - Precautions ..... 4-3
  - Interrupt Request (IRQn) Reset ..... 4-4
  - Interrupt-Sharing Software Requirements ..... 4-4
  - Interrupt-Sharing Chaining Structure and Signature ..... 4-6
  - ROM Considerations ..... 4-7
  - Implementation Information ..... 4-7
- Adapter ROM ..... 4-12
- Video Function Compatibility ..... 4-14
  - Video Presence Test ..... 4-14
  - Video Mode Switching ..... 4-15
- Multitasking Provisions ..... 4-16
- System Identification ..... 4-18
- Application Guidelines ..... 4-19
  - Hardware Interrupts ..... 4-19
  - Programming Considerations ..... 4-21
  - BIOS and Operating System Function Calls ..... 4-21
- Scan Code/Character Code Combinations ..... 4-24

**Notes:**

---

## Interrupt Sharing

This section defines an interrupt sharing protocol that allows multiple hardware adapters on the PC type I/O channel and Micro Channel™ to share a single interrupt request line.

### Precautions

Take the following precautions before implementing interrupt sharing:

- This interrupt sharing protocol is intended to run only in the real address mode. It is not intended to run in the protected (virtual address) mode.
- This interrupt sharing protocol does not apply to the sharing of an interrupt level between an interrupt handler running in the real mode and an interrupt handler running in the protected mode.
- This interrupt sharing protocol is not necessarily compatible with all operating systems.
- Interrupts must be disabled before control is passed to the next handler on the chain. The disabling of the interrupts allows the next handler to receive control as if a hardware interrupt had caused it to receive control.
- Interrupts must be disabled before the non-specific End of Interrupt (EOI) is issued and not reenabled in the interrupt handler to ensure that the Return from Interrupt (IRET) is executed. The flags are restored and the interrupts reenabled before another interrupt is serviced, protecting the stack from excessive build-up.
- All interrupt handlers must have a routine that can be executed after power-on to disable their adapters' interrupts. Executing this routine and resetting the interrupt sharing hardware ensures that adapters are deactivated if the user resets the system.

---

Micro Channel is a trademark of the International Business Machines Corporation.

- Interrupt handler implementations must store data in memory using Intel™ format; that is, word hex 424B is stored as 4BH,42H in memory.

## **Interrupt Request (IRQn) Reset**

The Micro Channel interrupt mechanism is level sensitive as opposed to the edge sensitive mechanism of the PC type I/O channel. The level sensitive Micro Channel mechanism simplifies the interrupt hardware needed for the adapters.

An interrupt request in the PC type I/O channel is implicitly reset due to the edge sensitive characteristic of the signal. In the Micro Channel, due to the level sensitive characteristic of the signal, an interrupt request must be explicitly reset by the bus slave interrupt handler software. This is not the case if the bus slave hardware implicitly resets the interrupt request. An example of a bus slave device that implicitly resets an interrupt request is the system timer.

## **Interrupt-Sharing Software Requirements**

All interrupt sharing software developed for Micro Channel bus slaves must reset the interrupt request. The interrupt-sharing chaining structure must be provided by all interrupt handlers. The 16-byte interrupt-sharing chaining structure must begin at the third byte from the entry point of the interrupt handler. Pointers and flags stored in the interrupt-sharing chaining structure must be stored in Intel data format (see “Interrupt-Sharing Chaining Structure and Signature” on page 4-6). These requirements are specified to support the portability of the interrupt handlers across hardware operating environments.

The interrupt handling software for all adapters sharing an interrupt request line must implement this interrupt sharing software standard. Interrupt sharing software operating in a multitasking environment must support the linking of a task’s interrupt handler to a chain of interrupt handlers, the sharing of the interrupt level while that task is active, and the unlinking of the interrupt handler from the chain once the task is complete.

---

Intel is a trademark of the Intel Corporation.

To link an interrupt handler, the newly activated task's interrupt handler replaces the interrupt vector in low memory with a pointer to its own interrupt handler. (See "ROM Considerations" on page 4-7 for interrupt handlers stored in ROM.) The interrupt handler must preserve the interrupt vector it is replacing and use it as a forward pointer to the next interrupt handler in the chain. This old interrupt vector must be stored at a fixed offset from the entry point of the new task's interrupt handler.

When the system acknowledges an interrupt request, each interrupt handler must determine whether it is the appropriate interrupt handler for the adapter presenting the interrupt request. This is accomplished by the handler reading the contents of the interrupt status register of the adapter.

If the handler's device caused the interrupt, the handler must service the interrupt, reset the interrupt status bit, clear the interrupts, issue a non-specific EOI to the interrupt controller, then execute an IRET.

If the handler's device did not cause the interrupt, the handler passes control to the next interrupt handler in the chain using the previously stored forward pointer.

An interrupt handler is unlinked from a chain by the task first locating its handler's position within the chain. The chain can be searched by starting at the interrupt vector in low memory and using the offset of each handler's forward pointer to locate the entry point of each handler. This is done until the task finds its own handler. Each interrupt handler's signature (424BH) must be checked to ensure that a valid forward pointer exists. The task's forward pointer replaces the forward pointer of the previous handler in the chain, thus removing the handler from the chain.

**Note:** If the interrupt handler cannot locate its position in the chain, the interrupt handler cannot unlink.

An application-dependent unlinking error-recovery procedure must be incorporated into the unlinking routine for those situations where the unlinking routine discovers that the interrupt chain has been corrupted (an interrupt handler is linked but does not have a valid signature). All interrupt sharing handlers, except those in ROM (see "ROM Considerations" on page 4-7), must use 424BH as the signature to avoid corrupting the chain.

During a system-reset condition, a routine for each interrupt handler must be executed after power-on to disable interrupts from their responsible devices.

Operating system environments that support dynamic relocation of software must manage the entire interrupt sharing process. Interrupt handler software written exclusively for dynamic-relocation operating-system environments does not have to provide the interrupt-sharing chaining structure. These interrupt handlers do not have to provide linking and unlinking support. They must provide support for disabling the interrupting capability of the bus slave they support.

## Interrupt-Sharing Chaining Structure and Signature

The interrupt-sharing software chaining structure is in a 16-byte format containing a 4-byte forward pointer (FPTR), a 2-byte signature, and 8 reserved bytes (RES\_BYTES), as depicted in the following example:

```
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  ; Flag for being first in chain
        JMP         SHORT RESET
        RES_BYTES   DB          DUP 7(0) ; -Reserved-
PAST:   ...                               ; Actual start of code
```

The interrupt-sharing software chaining structure begins at the third byte from the interrupt handler's entry point. The first instruction of each handler is a short jump around the structure, placing the structure at a known offset from the beginning of the handler routine. Since the position of each interrupt handler's chaining structure is known (except for the handlers on adapter ROM), the FPTRs can be updated when linking and unlinking.

The FIRST flag is used to determine the handler's position in the chain when linking and unlinking for shared interrupt levels. The contents of the FLAGS byte is changed to the value of the FIRST flag (80H) to indicate that the handler is the first handler linked in the chain. All interrupt handlers not stored in ROM must store the FIRST

flag (80H) in the FLAGS byte when they are the first handler in the chain.

The Reset routine, an entry point for the operating system, must disable the adapter interrupt and return to the operating system.

## ROM Considerations

Adapters with interrupt handlers in ROM must implement chaining by storing the FPTR in latches or ports on the adapter. If the adapter is sharing interrupt levels 7 or 15, it must also store the FIRST flag that indicates whether it is positioned first in the chain of interrupt handlers. Storage of this information is required because it cannot be guaranteed that handlers in ROM will always link first and never unlink. The ROM handler must contain the signature 0000H beginning at the seventh byte from the handler entry point since the forward pointer in ROM handlers is not stored at the third byte from the handler entry point.

## Implementation Information

The Interrupt Mask register is located at I/O port 21H. Specific End of Interrupt (EOI) values for the various interrupt levels are listed (67H for level 7). The specific EOI is accomplished by issuing an OUT to the 8259A operational control register (port 20H) using Operational Control Word 2 (OCW2). A non-specific EOI is accomplished by issuing an OUT value of hex 20 to the 8259A operational control register (port 20H).

The following are examples of code used to implement interrupt sharing:

### Linking

```
PUSH     ES
CLI                               ;Clear interrupts
;Set forward pointer to value of 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       ;Get offset of your forward pointer
                                           ; in an indexable register
MOV      CS:[SI],BX              ;Store the old interrupt vector
```

```

        MOV     CS:[SI+2],ES      ; in your forward pointer for
                                ; chaining
        CMP     ES:BYTE PTR[BX],0CFH ;Test for IRET
if  iret_test_only_is_needed  ; See NOTE below
        JNE     SETVECTR
else
        JE     FRSTVCTR
        CMP     ES:WORD PTR[BX+6],424BH ; Is signature present?
        JE     SETVECTR
        MOV     AX,ES
        CMP     AX,0F000H          ;See if pointing to dummy handler
        JNE     SETVCTR
        CMP     BX,WORD PTR ES:[0FF01H] ; Dummy Vector Pointer?
        JNE     SETVECTR          ;If dummy, then first
FRSTVCTR:
endif
        MOV     CS:FLAGS,FIRST    ;Set up first in chain flag
SETVECTR: POP     ES
        PUSH    DS
;Make interrupt vector in low memory point to your handler
        MOV     DX,OFFSET ENTRY   ;Make interrupt vector point to
                                ; your handler
        MOV     AX,SEG ENTRY      ;If DS ≠ 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 your level
        IN      AL,IMR            ;Read interrupt mask register
        JMP     $+2                ;I/O delay
        AND     AL,07FH          ;Unmask interrupt level 7
        OUT     IMR,AL            ;Write new interrupt mask
        MOV     AL,SPC_EOI        ;Issue specific EOI for level 7
        JMP     $+2                ; to allow pending level 7 interrupts
        OUT     OCR,AL            ; (if any) to be serviced
        STI     ;Enable interrupts
        POP     ES

```

## Notes:

1. The operating system must ensure that the SEG:OFF points to a valid interrupt handler or to an IRET (CFH) for levels 7 and 15.
2. ROM interrupt handlers during ROMSCAN (before the operating system is loaded) and handlers on other than IRQ 7, must test the SEG:OFF as shown in the "else" clause in this listing to determine if they are the first handler in the chain. Checking the SEG:OFF to see if it points to an IRET as the sole determination of

FIRST is allowed only on IRQ 7, and then only after the operating system is loaded.

## Interrupt Handler

```

YOUR_CARD EQU      xxxx           ;Location of your card interrupt
                                           ; control/status register
ISB        EQU      xx            ;Interrupt bit in your card
                                           ; interrupt control/status register
REARM      EQU      2F7H          ;Global Rearm location for
                                           ; interrupt level 7
SPC_EOI    EQU      67H           ;Specific EOI for 8259 interrupt
                                           ; level 7
EOI        EQU      20H           ;Non-specific EOI
OCR        EQU      20H           ;Location of 8259 operational
                                           ; control register
IMR        EQU      21H           ;Location of 8259 interrupt mask
MYCSEG     SEGMENT PARA
ASSUME     CS:MYCSEG,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        7 DUP (0)     ;Future expansion
PAST:     STI
          PUSH    ...             ;Save needed registers
          MOV     DX,YOUR_CARD     ;Select your status register
          IN      AL,DX            ;Read the status register
          TEST    AL,ISB           ;Your card caused interrupt?
          JNZ    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     ...             ;Clear interrupts
          JMP     DWORD PTR CS:FPTR ;Pass control to next handler on chain
SERVICE: ...                     ;Service the interrupt
EXIT:
          CLI     ...             ;Clear interrupts
          MOV     AL,EOI
          OUT    OCR,AL           ;Issue non-specific EOI to 8259
          MOV     DX,REARM         ;Rearm the cards
          OUT    DX,AL
          POP     ...             ;Restore registers
          IRET
RESET:    ...                     ;Disable your card
          RET
ENTRY     ENDP

```

```

MYCSEG   ENDS
         END     ENTRY

```

## Unlinking

```

PUSH     DS
PUSH     ES
CLI                                     ;Clear interrupts
MOV      AX,350FH                       ;DOS get interrupt vector
INT      21H                             ;ES:BX points to first of chain
MOV      CX,ES                           ;Pick up 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 your handler offset?
JNE      UNCHAIN_A                       ;No, branch
CMP      AX,CX                           ;Vector pointing to your handler
                                           ; segment?
JNE      UNCHAIN_A                       ;No, branch
;Set interrupt vector in low memory to point to the handler pointed to
; by your pointer
PUSH     DS
MOV      DX,WORD PTR CS:FPTR
MOV      DS,WORD PTR CS:FPTR[2]
MOV      AX,250FH                         ;DOS set interrupt vector
INT      21H
POP      DS
JMP      UNCHAIN_X
UNCHAIN_A: ; BX = FPTR offset, ES = FPTR segment, CX = CS
CMP      ES:[BX+6],4B42H                 ;Is handler using the appropriate
                                           ; conventions (is SIGNATURE present in
                                           ; the interrupt chaining structure)?
JNE      exception                       ;No, invoke error exception handler
LDS      SI,ES:[BX+2]                     ;Get FPTR segment and offset
CMP      SI,OFFSET ENTRY                 ;Is this forward pointer pointing to
                                           ; your handler offset?
JNE      UNCHAIN_B                       ;No, branch
MOV      CX,DS                            ;Move to compare
CMP      AX,CX                           ;Is this forward pointer pointing to
                                           ; your handler segment?
JNE      UNCHAIN_B                       ;No, branch
;Locate your handler in the chain
MOV      AX,WORD PTR CS:FPTR             ; Get your FPTR offset
MOV      ES:[BX+2],AX                    ;Replace offset of FPTR of handler
                                           ; that points to you
MOV      AX,WORD PTR CS:FPTR[2]          ; Get your FPTR segment
MOV      ES:[BX+4],AX                    ;Replace segment of FPTR of handler
                                           ; that points to you
MOV      AL,CS:FLAGS                      ;Get your flags
AND      AL,FIRST                         ;Isolate FIRST flag
OR       ES:[BX+6],AL                    ;Set your first flag into prior routine

```

```
                JMP     UNCHAIN_X
UNCHAIN_B: MOV   BX,SI           ;Move new offset to BX
              PUSH  DS           ;Set pointer to next in chain
              POP   ES
              JMP   UNCHAIN_A    ;Examine next handler in chain
UNCHAIN_X: STI                    ;Enable interrupts
              POP   ES
              POP   DS
```

---

## Adapter ROM

The BIOS provides a method for integrating adapters with on-board ROM code into the system. During the POST, interrupt vectors are established for BIOS calls. After the default vectors are in place, a scan for adapter ROM modules occurs. At this point, an adapter ROM routine can gain control. The routine can establish or intercept interrupt vectors to hook into the system.

Early in the POST the absolute addresses hex C0000 through hex C7FFF are scanned in 2Kb blocks in search of adapter ROM modules that need to be initialized (for example, valid video adapter ROM).

The absolute addresses hex C8000 through hex DFFFF are scanned in 2Kb blocks in search of devices with valid adapter ROM modules. Valid adapter 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 (limit hex 7F) in the ROM (length/512). A checksum tests 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 valid.

When the POST identifies valid adapter ROM, it executes a far call to byte 3 of the ROM (which should contain executable code). The device can now perform power-on initialization. The adapter ROM should return control to the POST by executing a far return.

For PC Convertible, if the adapter ROM diagnoses a self-test error, the following should be done before returning:

- Set bit 4 of 40:12 (POST status) to 1
- Set the device number for the supported adapter into (AH)
- Set a two-digit error code into (AL).

If no self-test error is found, the adapter ROM should reset bit 4 of 40:12 (POST status) to 0 before returning.

For Personal System/2 products, video adapters in the channel have a ROM signature code that identifies the video adapter. During the POST, when CMOS is not valid (abnormal condition), the signature code is used to find the first video adapter and set up its ROM programmable option select (POS) parameters.

The code starts at 0CH in the ROM address space and consists of:

77H, CCH, 'VIDEO ', POS Byte 1, POS Byte 2, POS Byte 3, POS Byte 4

Video ROM scan remains C0000H to C7FFFH.

For PC Convertible, during early ROM scan the following protocol is established to determine the video support:

Upon return from a call to a video adapter ROM module, (BH) indicates the following:

- (BH) = 00H - Not a video adapter
- = 02H - Video adapter supporting video in the color/graphics adapter range
- = 04H - Video adapter supporting video in the monochrome adapter range

---

## Video Function Compatibility

The following procedures are recommended to provide video function compatibility to application software.

### Video Presence Test

Use this video presence test to determine which IBM video functions are present.

1. The first step is to issue an INT 10H, with (AH) = 1AH and (AL) = 00H (Read Display Combination Code).

If on return (AL) is not equal to 1AH, the Read/Write Display Combination Code function is not supported, and step 2 should be followed to determine video presence.

If on return (AL) = 1AH, the information returned in (BX) defines the video environment. The active display code is returned in (BL). The alternate display code, if any, is returned in (BH). Refer to INT 10H, (AH) = 1AH on page 2-39 for display code definitions.

2. To determine the presence of an IBM Enhanced Graphics Adapter (EGA) when the Display Combination Code function is not supported, issue an INT 10H with (AH) = 12H and (BL) = 10H (Return EGA Information).

If on return, (BL) = 10H, an EGA is not present and step 3 should be followed.

If on return (BL) is not equal to 10H then an EGA is present. Note that an IBM Color/Graphics Monitor Adapter or an IBM Monochrome Display and Printer Adapter may also be present, depending on the EGA switch settings.

3. Complete steps 1 and 2 before performing this step. The video functions that may be present at this point are the IBM Color/Graphics Monitor Adapter, the IBM Monochrome Display and Printer Adapter, or both. Perform a presence test on video buffer addresses 0B8000H, 0B0000H to determine which video functions are present.

## Video Mode Switching

Use the following video mode switching procedure when applications will switch between monochrome and color video modes. A correct video function presence test, as previously described, is required. The following three system video environments are possible:

1. A single video function that supports either monochrome or color video modes. If a monochrome function is present, only monochrome video modes are available. If a color function is present, only color video modes are available.
2. Two video functions; one supporting color and the other supporting monochrome video modes. In this case both monochrome and color video modes are available. To switch from monochrome to color or from color to monochrome, the application should change the system equipment video mode type bits (see data area 40:10, bits 5, 4 on page 3-4) to monochrome or color and issue a INT 10H, (AH) = 00H (Set Mode).
3. A single video function that supports both monochrome and color video modes. To determine if a single video function supports both monochrome and color video modes the application should issue an INT 10H, (AH) = 1BH (Return Functionality/State Information).

If on return (AL) is not equal to 1BH, the Return Functionality/State function is not supported. Support for both monochrome and color video modes on a single video function is not available.

If on return (AL) = 1BH, use the returned information to determine if the All Modes on All Displays function is active. If active, color and monochrome modes are available and the application should change the system equipment video mode type bits to monochrome or color, and issue a INT 10H, (AH) = 00H (Set Mode). If inactive, only color modes or monochrome modes are available, depending on the results of the video presence test.

---

## Multitasking Provisions

The BIOS provides hooks to assist in multitasking implementation. Whenever a busy (Wait) loop occurs in the BIOS, a hook is provided for the program to break out of the loop. Also, when the BIOS services an interrupt, a corresponding Wait loop is exited, and another hook is provided. A program may be written that employs the bulk of the device driver code. The following is valid only in the microprocessor real address mode and must be taken by the code to allow this support.

The program is responsible for matching corresponding Wait and Post calls and for the serialization of access to the device driver. The BIOS code is not reentrant.

The following four interfaces are used by the multitasking dispatcher:

**Startup:** The startup code hooks INT 15H. The dispatcher is responsible to check for function codes of (AH) = 90H or (AH) = 91H (see the following descriptions of Wait and Post). The dispatcher must pass all other functions to the previous user of INT 15H (use a JMP or a CALL). If (AH) = 90H or (AH) = 91H, the dispatcher must do the appropriate processing, and return by the IRET instruction.

**Serialization:** The multitasking system must ensure that the device driver code is used serially. Multiple entries into the code can result in errors.

**Wait (Busy):** Whenever the BIOS is about to enter a Wait loop, it first issues an INT 15H, (AH) = 90H. This signals a wait condition. At this point, the dispatcher should save the task status and dispatch another task. This allows overlapped execution of tasks when the hardware is busy. The following is an outline of the code that has been added to the BIOS to perform this function.

```

MOV AX, 90xxH      ;Wait code in AH and
                   ; type code in AL
INT 15H           ;Issue call
JC TIMEOUT        ;Optional: for time-out or
                   ; if carry is set, time-out
                   ; occurred
NORMAL TIMEOUT LOGIC ;Normal time-out

```

**Post (Interrupt):** Whenever the BIOS has set an interrupt flag for a corresponding busy loop, an INT 15H, (AH) = 91H occurs. This signals a Post condition. At this point, the dispatcher must set the task status to “ready to run” and return to the interrupt routine. The following is an outline of the code added to the BIOS that performs this function.

```

MOV AX, 91xxH      ;Post code AH and
                   ; type code AL
INT 15H           ;Issue call

```

Three Wait loop function code classes are supported:

- The first (hex 0 to 7F) is serially reusable. This means that for the devices that use these codes, access to the BIOS must be restricted to one task at a time and the operating system must serialize access.
- The second (hex 80 to BF) is for reentrant devices. There is no restriction on the number of tasks that may access the device. ES:BX is used to distinguish different calls.
- The third (hex C0 to FF) is non interrupt (Wait-only calls). There is no corresponding interrupt for the Wait loop. The dispatcher must take the appropriate action to satisfy this condition, and exit from the loop. There is no complementary Post for these Waits. They are time-out only and the times are function-number dependent.

To support time-outs properly, the multitasking dispatcher must be aware of time. If a device enters a busy loop, it generally should remain there for a specific amount of time before indicating an error. The dispatcher must return to the BIOS Wait loop with the carry bit set if a time-out occurs.

## System Identification

Each BIOS ROM module has a model byte located at F000:FFFE in ROM. In some cases a submodel byte and a BIOS revision level byte are used to further distinguish the various BIOS ROM modules. To access this information, see INT 15H, (AH) = C0H (Return System Configuration Parameters) on page 2-94.

Product	BIOS Date	Model Byte	Submodel Byte	Revision
PC	04/24/81	FF	—	—
PC	10/19/81	FF	—	—
PC	10/27/82	FF	—	—
PC XT	11/08/82	FE	—	—
PC XT	01/10/86	FB	00	01
PC XT	05/09/86	FB	00	02
PCjr	06/01/83	FD	—	—
AT	01/10/84	FC	—	—
AT	06/10/85	FC	00	01
AT	11/15/85	FC	01	00
PC XT Model 286	04/21/86	FC	02	00
PC Convertible	09/13/85	F9	00	00
Personal System/2 Model 30	09/02/86	FA	00	00
Personal System/2 Model 50	*	FC	04	00
Personal System/2 Model 60	*	FC	05	00
Personal System/2 Model 80	*	F8	00	00
Personal System/2 Model 80	*	F8	01	00

\* BIOS date not available.

Figure 4-1. System Identification

---

## Application Guidelines

Use the following information to develop application programs for the IBM Personal System/2 and Personal Computer products. Whenever possible, BIOS should be used as an interface to hardware in order to provide maximum compatibility and portability of applications across systems.

### Hardware Interrupts

Hardware interrupts are level-sensitive for systems using the Micro Channel architecture while systems using the PC type I/O channel have edge-sensitive hardware interrupts. On edge-sensitive interrupt systems, the interrupt controller clears its internal interrupt-in-progress latch when the interrupt routine sends an end of interrupt (EOI) command to the controller. The EOI is sent whether the incoming interrupt request to the controller is active or inactive.

In level-sensitive systems, the interrupt-in-progress latch is readable at an I/O address bit position. This latch is read during the interrupt service routine and may be reset by the read operation or may require an explicit reset.

**Note:** Designers may want to limit the number of devices sharing an interrupt level for performance and latency considerations.

The interrupt controller on level-sensitive systems requires the interrupt request to be inactive at the time the EOI is sent; otherwise, a “new” interrupt request will be detected and another microprocessor interrupt caused.

To avoid this problem, a level-sensitive interrupt handler must clear the interrupt condition (usually by a Read or Write to an I/O port on the device causing the interrupt). After clearing the interrupt condition, a `JMP $+2` should be executed prior to sending the EOI to the interrupt controller. This ensures that the interrupt request is removed prior to reenabling the interrupt controller. Another `JMP $+2` should be executed after sending the EOI, but prior to enabling the interrupt through the Set Interrupt Flag (STI) command.

I/O commands followed immediately by an STI instruction do not permit enough recovery time for some system board and channel operations. To ensure enough time, a `JMP SHORT $+2` must be inserted between the I/O command and the STI instruction.

**Notes:**

1. `MOV AL,AH` type instructions do not allow enough recovery time. An example of the correct procedure follows:

```
OUT  IO_ADD,AL
JMP  SHORT $+2
MOV  AL,AH
STI
```

2. Prior to programming the interrupt controllers, interrupts should be disabled by issuing a Clear Interrupt Flag (CLI) command. This includes the Mask register, EOIs, initialization control words, and operational control words.

In the level-sensitive systems, hardware prevents the interrupt controllers from being set to the edge-sensitive mode.

Hardware interrupt IRQ9 is defined as the replacement interrupt level for the cascade level IRQ2. Program interrupt sharing should be implemented on IRQ2, INT 0AH. The following processing occurs to maintain compatibility with the IRQ2 used by IBM Personal Computer products:

1. A device drives the interrupt request active on IRQ2 of the channel.
2. This interrupt request is mapped in hardware to IRQ9 input on the second interrupt controller.
3. When the interrupt occurs, the system microprocessor passes control to the IRQ9 (INT 71H) interrupt handler.
4. This interrupt handler performs an EOI to the second interrupt controller and passes control to IRQ2 (INT 0AH) interrupt handler.
5. When handling the interrupt, the IRQ2 interrupt handler causes the device to reset the interrupt request prior to performing an EOI to the master interrupt controller that finishes servicing the IRQ2 request.

## **Programming Considerations**

The IBM-supported languages of IBM C, BASIC, FORTRAN, COBOL, and Pascal are the best choices for writing compatible programs. If a program uses specific features of the hardware, that program may not be compatible with all IBM Personal System/2 and Personal Computer products.

Any program that requires precise timing information should obtain it through an operating system or language interface; for example, TIME\$ in BASIC. The use of programming loops may prevent a program from being compatible with other Personal System/2 and IBM Personal Computer products, and software.

## **BIOS and Operating System Function Calls**

For maximum portability, programs should perform all I/O operations through operating system function calls. In environments where the operating system does not provide the necessary programming interfaces, programs should access the hardware through BIOS function calls, if permissible. When writing programs, consider the following:

- In some environments, program interrupts are used for access to these functions. This practice removes the absolute addressing from the program. Only the interrupt number is required.
- The system can mask hardware sensitivity. New devices can change the BIOS to accept the same programming interface on the new device.
- In cases where BIOS provides parameter tables, such as for video or diskette, a program can substitute new parameter values by building a new copy of the table and changing the vector to point to that table. The program should copy the current table, using the current vector, and then modify those locations in the table that need to be changed. In this way, the program does not inadvertently change any values that should be left the same.
- The Diskette Parameters Table pointed to by INT 1EH consists of 11 parameters required for diskette operation. It is recommended that the values supplied in ROM be used. If it becomes necessary to modify any of the parameters, build

another parameter block and modify the address at INT 1EH (0:78) to point to the new block.

The parameters were established to allow:

- Some models of the IBM Personal Computer to operate both the 5.25-inch high capacity diskette drive (96 tracks per inch) and the 5.25-inch double-sided diskette drive (48 tracks per inch).
- Some models of the Personal System/2 to operate both the 3.5-inch 1.44Mb diskette drive and the 3.5-inch 720Kb diskette drive.

The Gap Length Parameter is not always retrieved from the parameter block. The gap length used during diskette read, write, and verify operations is derived from within diskette BIOS. The gap length for format operations is still obtained from the parameter block.

If a parameter block contains a head settle time parameter value of 0 milliseconds, and a write or format operation is being performed, the following minimum head settle times are enforced.

<b>Drive Type</b>	<b>Head Settle Time</b>
5.25-Inch Diskette Drives:	
Double Sided (48 TPI)	20 milliseconds
High Capacity (96 TPI)	15 milliseconds
3.5-Inch Diskette Drives:	
720Kb	20 milliseconds
1.44Mb	15 milliseconds

Figure 4-2. Write and Format Head Settle Time

Read and verify operations use the head settle time provided by the parameter block.

If a parameter block contains a motor start wait parameter of less than 500 milliseconds (1 second for a Personal Computer product) for a write or verify operation, diskette BIOS enforces a minimum time of 500 milliseconds (1 second for a Personal Computer product). Read and write operations use the motor start time provided by the parameter block.

- Programs may be designed to reside on both 5.25-inch and 3.5-inch diskettes. Since not all programs are operating-system dependent, the following procedure can be used to determine the type of media inserted into a diskette drive:

1. Verify track 0, head 0, sector 1 (1 sector): This allows diskette BIOS to determine if the format of the media is a recognizable type.

If the verify operation fails, issue the reset function (AH=0) to diskette BIOS and try the operation again. If another failure occurs, the media needs to be formatted or is defective.

2. Verify track 0, head 0, sector 16 (1 sector).

If the verify operation fails, either a 5.25-inch (48 TPI) or 3.5-inch 720Kb diskette is installed. The type can be determined by verifying track 78, head 1, sector 1 (1 sector). A successful verification of track 78 indicates a 3.5-inch 720Kb diskette is installed; a verification failure indicates a 5.25-inch (48 TPI) diskette is installed.

**Note:** Refer to the *DOS Technical Reference* for the File Allocation Table parameters for single-sided and double-sided diskettes.

3. Read the diskette controller status in BIOS starting with address 40:42. The fifth byte defines the head that the operation ended with. If the operation ended with head 1, the diskette is a 5.25-inch high-capacity (96 TPI) diskette; if the operation ended with head 0, the diskette is a 3.5-inch 1.44Mb diskette.

# Scan Code/Character Code Combinations

The following lists the keyboard keystrokes and the scan code/character code combinations that are returned through INT 16H:

Keystroke	83- and 84-Key Standard Function	101/102-Key Standard Function	101/102-Key Extended Function
Esc	01/1B	01/1B	01/1B
1	02/31	02/31	02/31
2	03/32	03/32	03/32
3	04/33	04/33	04/33
4	05/34	05/34	05/34
5	06/35	06/35	06/35
6	07/36	07/36	07/36
7	08/37	08/37	08/37
8	09/38	09/38	09/38
9	0A/39	0A/39	0A/39
0	0B/30	0B/30	0B/30
-	0C/2D	0C/2D	0C/2D
=	0D/3D	0D/3D	0D/3D
Backspace	0E/08	0E/08	0E/08
Tab	0F/09	0F/09	0F/09
q	10/71	10/71	10/71
w	11/77	11/77	11/77
e	12/65	12/65	12/65
r	13/72	13/72	13/72
t	14/74	14/74	14/74
y	15/79	15/79	15/79
u	16/75	16/75	16/75
i	17/69	17/69	17/69
o	18/6F	18/6F	18/6F
p	19/70	19/70	19/70
[	1A/5B	1A/5B	1A/5B
]	1B/5D	1B/5D	1B/5D
Return	1C/0D	1C/0D	1C/0D
Ctrl	**	**	**
a	1E/61	1E/61	1E/61
s	1F/73	1F/73	1F/73
d	20/64	20/64	20/64
f	21/66	21/66	21/66
g	22/67	22/67	22/67
h	23/68	23/68	23/68
j	24/6A	24/6A	24/6A
k	25/6B	25/6B	25/6B
l	26/6C	26/6C	26/6C
:	27/3B	27/3B	27/3B
'	28/27	28/27	28/27
.	29/60	29/60	29/60
Shift	**	**	**

Figure 4-3 (Part 1 of 3). Keyboard Keystrokes

<b>Keystroke</b>	<b>83- and 84-Key Standard Function</b>	<b>101/102-Key Standard Function</b>	<b>101/102-Key Extended Function</b>
\	2B/5C	2B/5C	2B/5C
z	2C/7A	2C/7A	2C/7A
x	2D/78	2D/78	2D/78
c	2E/63	2E/63	2E/63
v	2F/76	2F/76	2F/76
b	30/62	30/62	30/62
n	31/6E	31/6E	31/6E
m	32/6D	32/6D	32/6D
,	33/2C	33/2C	33/2C
.	34/2E	34/2E	34/2E
/	35/2F	35/2F	35/2F
*	37/2A	37/2A	37/2A
Alt	**	**	**
Space	39/20	39/20	39/20
Caps Lock	**	**	**
F1	3B/00	3B/00	3B/00
F2	3C/00	3C/00	3C/00
F3	3D/00	3D/00	3D/00
F4	3E/00	3E/00	3E/00
F5	3F/00	3F/00	3F/00
F6	40/00	40/00	40/00
F7	41/00	41/00	41/00
F8	42/00	42/00	42/00
F9	43/00	43/00	43/00
F10	44/00	44/00	44/00
F11	(no key)	--	85/00
F12	(no key)	--	86/00
Num Lock	**	**	**
Scroll Lock	**	**	**
Home	47/00	47/00	47/00
Up Arrow	48/00	48/00	48/00
PgUp	49/00	49/00	49/00
-	4A/2D	4A/2D	4A/2D
Left Arrow	4B/00	4B/00	4B/00
Center Key	--	--	4C/00
Right Arrow	4D/00	4D/00	4D/00
+	4E/2B	4E/2B	4E/2B
End	4F/00	4F/00	4F/00
Down Arrow	50/00	50/00	50/00
PgDn	51/00	51/00	51/00
Ins	52/00	52/00	52/00
Del	53/00	53/00	53/00
SysReq	**	(no key)	(no key)
Key 45	(no key)	56/5C	56/5C
Enter	(no key)	1C/0D	E0/0D
/	(no key)	35/2F	E0/2F
PrtSc	(no key)	**	**
Pause	(no key)	**	**
Home	(no key)	47/00	47/E0
Up Arrow	(no key)	48/00	48/E0
PageUp	(no key)	49/00	49/E0
Left Arrow	(no key)	4B/00	4B/E0

Figure 4-3 (Part 2 of 3). Keyboard Keystrokes

<b>Keystroke</b>	<b>83- and 84-Key Standard Function</b>	<b>101/102-Key Standard Function</b>	<b>101/102-Key Extended Function</b>
Right Arrow	(no key)	4D/00	4D/E0
End	(no key)	4F/00	4F/E0
Down Arrow	(no key)	50/00	50/E0
PageDown	(no key)	51/00	51/E0
Insert	(no key)	52/00	52/E0
Delete	(no key)	53/00	53/E0

\*\* These combinations do not provide a keystroke for the application but perform some other action. They are not put in the INT 16H queue.  
-- These combinations have no function and are ignored.

Figure 4-3 (Part 3 of 3). Keyboard Keystrokes

The following lists the Shift keyboard keystrokes and the scan code/character code combinations that are returned through INT 16H:

<b>Keystroke</b>	<b>83- and 84-Key Standard Function</b>	<b>101/102-Key Standard Function</b>	<b>101/102-Key Extended Function</b>
Shift Esc	01/1B	01/1B	01/1B
Shift !	02/21	02/21	02/21
Shift @	03/40	03/40	03/40
Shift #	04/23	04/23	04/23
Shift \$	05/24	05/24	05/24
Shift %	06/25	06/25	06/25
Shift ^	07/5E	07/5E	07/5E
Shift &	08/26	08/26	08/26
Shift *	09/2A	09/2A	09/2A
Shift (	0A/28	0A/28	0A/28
Shift )	0B/29	0B/29	0B/29
Shift _	0C/5F	0C/5F	0C/5F
Shift +	0D/2B	0D/2B	0D/2B
Shift Backspace	0E/08	0E/08	0E/08
Shift Tab (Backtab)	0F/00	0F/00	0F/00
Shift Q	10/51	10/51	10/51
Shift W	11/57	11/57	11/57
Shift E	12/45	12/45	12/45
Shift R	13/52	13/52	13/52
Shift T	14/54	14/54	14/54
Shift Y	15/59	15/59	15/59
Shift U	16/55	16/55	16/55
Shift I	17/49	17/49	17/49
Shift O	18/4F	18/4F	18/4F
Shift P	19/50	19/50	19/50
Shift {	1A/7B	1A/7B	1A/7B
Shift }	1B/7D	1B/7D	1B/7D
Shift Return	1C/0D	1C/0D	1C/0D
Shift Ctrl	**	**	**

Figure 4-4 (Part 1 of 3). Shift Keyboard Keystrokes

<b>Keystroke</b>	<b>83- and 84-Key Standard Function</b>	<b>101/102-Key Standard Function</b>	<b>101/102-Key Extended Function</b>
Shift A	1E/41	1E/41	1E/41
Shift S	1F/53	1F/53	1F/53
Shift D	20/44	20/44	20/44
Shift F	21/46	21/46	21/46
Shift G	22/47	22/47	22/47
Shift H	23/48	23/48	23/48
Shift J	24/4A	24/4A	24/4A
Shift K	25/4B	25/4B	25/4B
Shift L	26/4C	26/4C	26/4C
Shift :	27/3A	27/3A	27/3A
Shift "	28/22	28/22	28/22
Shift ~	29/7E	29/7E	29/7E
Shift	2B/7C	2B/7C	2B/7C
Shift Z	2C/5A	2C/5A	2C/5A
Shift X	2D/58	2D/58	2D/58
Shift C	2E/43	2E/43	2E/43
Shift V	2F/56	2F/56	2F/56
Shift B	30/42	30/42	30/42
Shift N	31/4E	31/4E	31/4E
Shift M	32/4D	32/4D	32/4D
Shift <	33/3C	33/3C	33/3C
Shift >	34/3E	34/3E	34/3E
Shift ?	35/3F	35/3F	35/3F
Shift *	37/2A	37/2A	37/2A
Shift Alt	**	**	**
Shift Space	39/20	39/20	39/20
Shift Caps Lock	**	**	**
Shift F1	54/00	54/00	54/00
Shift F2	55/00	55/00	55/00
Shift F3	56/00	56/00	56/00
Shift F4	57/00	57/00	57/00
Shift F5	58/00	58/00	58/00
Shift F6	59/00	59/00	59/00
Shift F7	5A/00	5A/00	5A/00
Shift F8	5B/00	5B/00	5B/00
Shift F9	5C/00	5C/00	5C/00
Shift F10	5D/00	5D/00	5D/00
Shift F11	(no key)	--	87/00
Shift F12	(no key)	--	88/00
Shift Num Lock	**	**	**
Shift Scroll Lock	**	**	**
Shift 7	47/37	47/37	47/37
Shift 8	48/38	48/38	48/38
Shift 9	49/39	49/39	49/39
Shift -	4A/2D	4A/2D	4A/2D
Shift 4	4B/34	4B/34	4B/34
Shift 5	4C/35	4C/35	4C/35
Shift 6	4D/36	4D/36	4D/36
Shift +	4E/2B	4E/2B	4E/2B
Shift 1	4F/31	4F/31	4F/31
Shift 2	50/32	50/32	50/32

Figure 4-4 (Part 2 of 3). Shift Keyboard Keystrokes

<b>Keystroke</b>	<b>83- and 84-Key Standard Function</b>	<b>101/102-Key Standard Function</b>	<b>101/102-Key Extended Function</b>
Shift 3	51/33	51/33	51/33
Shift 0	52/30	52/30	52/30
Shift .	53/2E	53/2E	53/2E
Shift SysReq	**	(no key)	(no key)
Shift Key 45	(no key)	56/7C	56/7C
Shift Enter	(no key)	1C/0D	E0/0D
Shift /	(no key)	35/2F	E0/2F
Shift PrtSc	(no key)	**	**
Shift Pause	(no key)	**	**
Shift Home	(no key)	47/00	47/E0
Shift Up Arrow	(no key)	48/00	48/E0
Shift PgUp	(no key)	49/00	49/E0
Shift Left Arrow	(no key)	4B/00	4B/E0
Shift Right	(no key)	4D/00	4D/E0
Shift End	(no key)	4F/00	4F/E0
Shift Down Arrow	(no key)	50/00	50/E0
Shift PgDn	(no key)	51/00	51/E0
Shift Insert	(no key)	52/00	52/E0
Shift Delete	(no key)	53/00	53/E0

\*\* These combinations do not provide a keystroke for the application presently running but perform some other action. They are not put in the INT 16H queue.  
-- These combinations have no function and are ignored.

Figure 4-4 (Part 3 of 3). Shift Keyboard Keystrokes

The following lists the Ctrl keyboard keystrokes and the scan code/character code combinations that are returned through INT 16H:

<b>Keystroke</b>	<b>83- and 84-Key Standard Function</b>	<b>101/102-Key Standard Function</b>	<b>101/102-Key Extended Function</b>
Ctrl Esc	01/1B	01/1B	01/1B
Ctrl 1	--	--	--
Ctrl 2 (NUL)	03/00	03/00	03/00
Ctrl 3	--	--	--
Ctrl 4	--	--	--
Ctrl 5	--	--	--
Ctrl 6 (RS)	07/1E	07/1E	07/1E
Ctrl 7	--	--	--
Ctrl 8	--	--	--
Ctrl 9	--	--	--
Ctrl 0	--	--	--
Ctrl _	0C/1F	0C/1F	0C/1F
Ctrl =	--	--	--
Ctrl Backspace (DEL)	0E/7F	0E/7F	0E/7F
Ctrl Tab	--	--	94/00
Ctrl q (DC1)	10/11	10/11	10/11

Figure 4-5 (Part 1 of 3). Ctrl Keyboard Keystrokes

<b>Keystroke</b>	<b>83- and 84-Key Standard Function</b>	<b>101/102-Key Standard Function</b>	<b>101/102-Key Extended Function</b>
Ctrl w (ETB)	11/17	11/17	11/17
Ctrl e (ENQ)	12/05	12/05	12/05
Ctrl r (DC2)	13/12	13/12	13/12
Ctrl t (DC4)	14/14	14/14	14/14
Ctrl y (EM)	15/19	15/19	15/19
Ctrl u (NAK)	16/15	16/15	16/15
Ctrl i (HT)	17/09	17/09	17/09
Ctrl o (SI)	18/0F	18/0F	18/0F
Ctrl p (DLE)	19/10	19/10	19/10
Ctrl [ (ESC)	1A/1B	1A/1B	1A/1B
Ctrl ] (GS)	1B/1D	1B/1D	1B/1D
Ctrl Return (LF)	1C/0A	1C/0A	1C/0A
Ctrl a (SOH)	1E/01	1E/01	1E/01
Ctrl s (DC3)	1F/13	1F/13	1F/13
Ctrl d (EOT)	20/04	20/04	20/04
Ctrl f (ACK)	21/06	21/06	21/06
Ctrl g (BEL)	22/07	22/07	22/07
Ctrl h (Backspace)	23/08	23/08	23/08
Ctrl j (LF)	24/0A	24/0A	24/0A
Ctrl k (VT)	25/0B	25/0B	25/0B
Ctrl l (FF)	26/0C	26/0C	26/0C
Ctrl ;	--	--	--
Ctrl ' (Apostrophe)	--	--	--
Ctrl ` (Backquote)	--	--	--
Ctrl Shift	**	**	**
Ctrl \ (FS)	2B/1C	2B/1C	2B/1C
Ctrl z (SUB)	2C/1A	2C/1A	2C/1A
Ctrl x (CAN)	2D/18	2D/18	2D/18
Ctrl c (ETX)	2E/03	2E/03	2E/03
Ctrl v (SYN)	2F/16	2F/16	2F/16
Ctrl b (STX)	30/02	30/02	30/02
Ctrl n (SO)	31/0E	31/0E	31/0E
Ctrl m (CR)	32/0D	32/0D	32/0D
Ctrl ,	--	--	--
Ctrl .	--	--	--
Ctrl /	--	--	--
Ctrl * (Asterisk)	--	--	96/00
Ctrl Alt	**	**	**
Ctrl Space	39/20	39/20	39/20
Ctrl Caps Lock	--	--	--
Ctrl F1	5E/00	5E/00	5E/00
Ctrl F2	5F/00	5F/00	5F/00
Ctrl F3	60/00	60/00	60/00
Ctrl F4	61/00	61/00	61/00
Ctrl F5	62/00	62/00	62/00
Ctrl F6	63/00	63/00	63/00
Ctrl F7	64/00	64/00	64/00
Ctrl F8	65/00	65/00	65/00
Ctrl F9	66/00	66/00	66/00
Ctrl F10	67/00	67/00	67/00
Ctrl F11	(no key)	--	89/00
Ctrl F12	(no key)	--	8A/00

Figure 4-5 (Part 2 of 3). Ctrl Keyboard Keystrokes

<b>Keystroke</b>	<b>83- and 84-Key Standard Function</b>	<b>101/102-Key Standard Function</b>	<b>101/102-Key Extended Function</b>
Ctrl Num Lock	--	--	--
Ctrl Scroll Lock	--	--	--
Ctrl Home	77/00	77/00	77/00
Ctrl Up Arrow	--	--	8D/00
Ctrl PgUp	84/00	84/00	84/00
Ctrl Keypad -	--	--	8E/00
Ctrl Left Arrow	73/00	73/00	73/00
Ctrl Center	--	--	8F/00
Ctrl Right Arrow	74/00	74/00	74/00
Ctrl Keypad +	--	--	90/00
Ctrl End	75/00	75/00	75/00
Ctrl Down Arrow	--	--	91/00
Ctrl PgDn	76/00	76/00	76/00
Ctrl Ins	--	--	92/00
Ctrl Del	--	--	93/00
Ctrl SysReq	**	(no key)	(no key)
Ctrl Key 45	(no key)	--	--
Ctrl Enter	(no key)	1C/0A	E0/0A
Ctrl /	(no key)	--	95/00
Ctrl PrtSc	(no key)	72/00	72/00
Ctrl Break	(no key)	00/00	00/00
Ctrl Home	(no key)	77/00	77/E0
Ctrl Up	(no key)	--	8D/E0
Ctrl PageUp	(no key)	84/00	84/E0
Ctrl Left	(no key)	73/00	73/E0
Ctrl Right	(no key)	74/00	74/E0
Ctrl End	(no key)	75/00	75/E0
Ctrl Down	(no key)	--	91/E0
Ctrl PageDown	(no key)	76/00	76/E0
Ctrl Insert	(no key)	--	92/E0
Ctrl Delete	(no key)	--	93/E0

\*\* These combinations do not provide a keystroke for the application presently running but perform some other action. They are not put on the INT 16H queue.  
-- These combinations have no function and are ignored.

Figure 4-5 (Part 3 of 3). Ctrl Keyboard Keystrokes

The following lists the Alt keyboard keystrokes and the scan code/character code combinations that are returned through INT 16H:

<b>Keystroke</b>	<b>83- and 84-Key Standard Function</b>	<b>101/102-Key Standard Function</b>	<b>101/102-Key Extended Function</b>
Alt Esc	--	--	01/00
Alt 1	78/00	78/00	78/00
Alt 2	79/00	79/00	79/00
Alt 3	7A/00	7A/00	7A/00
Alt 4	7B/00	7B/00	7B/00
Alt 5	7C/00	7C/00	7C/00
Alt 6	7D/00	7D/00	7D/00
Alt 7	7E/00	7E/00	7E/00
Alt 8	7F/00	7F/00	7F/00
Alt 9	80/00	80/00	80/00
Alt 0	81/00	81/00	81/00
Alt -	82/00	82/00	82/00
Alt =	83/00	83/00	83/00
Alt Backspace	--	--	0E/00
Alt Tab	--	--	A5/00
Alt q	10/00	10/00	10/00
Alt w	11/00	11/00	11/00
Alt e	12/00	12/00	12/00
Alt r	13/00	13/00	13/00
Alt t	14/00	14/00	14/00
Alt y	15/00	15/00	15/00
Alt u	16/00	16/00	16/00
Alt i	17/00	17/00	17/00
Alt o	18/00	18/00	18/00
Alt p	19/00	19/00	19/00
Alt [	--	--	1A/00
Alt ]	--	--	1B/00
Alt Return	--	--	1C/00
Alt Ctrl	**	**	**
Alt a	1E/00	1E/00	1E/00
Alt s	1F/00	1F/00	1F/00
Alt d	20/00	20/00	20/00
Alt f	21/00	21/00	21/00
Alt g	22/00	22/00	22/00
Alt h	23/00	23/00	23/00
Alt j	24/00	24/00	24/00
Alt k	25/00	25/00	25/00
Alt l	26/00	26/00	26/00
Alt ;	--	--	27/00
Alt '	--	--	28/00
Alt `	--	--	29/00
Alt Shift	**	**	**
Alt \	--	--	2B/00
Alt z	2C/00	2C/00	2C/00
Alt x	2D/00	2D/00	2D/00
Alt c	2E/00	2E/00	2E/00
Alt v	2F/00	2F/00	2F/00
Alt b	30/00	30/00	30/00

Figure 4-6 (Part 1 of 2). Alt Keyboard Keystrokes

Keystroke	83- and 84-Key Standard Function	101/102-Key Standard Function	101/102-Key Extended Function
Alt n	31/00	31/00	31/00
Alt m	32/00	32/00	32/00
Alt ,	--	--	33/00
Alt .	--	--	34/00
Alt /	--	--	35/00
Alt *	--	--	37/00
Alt Space	39/20	39/20	39/20
Alt Caps Lock	**	**	**
Alt F1	68/00	68/00	68/00
Alt F2	69/00	69/00	69/00
Alt F3	6A/00	6A/00	6A/00
Alt F4	6B/00	6B/00	6B/00
Alt F5	6C/00	6C/00	6C/00
Alt F6	6D/00	6D/00	6D/00
Alt F7	6E/00	6E/00	6E/00
Alt F8	6F/00	6F/00	6F/00
Alt F9	70/00	70/00	70/00
Alt F10	71/00	71/00	71/00
Alt F11	(no key)	--	8B/00
Alt F12	(no key)	--	8C/00
Alt Num Lock	**	**	**
Alt Scroll Lock	**	**	**
Alt Keypad -	--	--	4A/00
Alt Keypad +	--	--	4E/00
Alt Keypad Nos.	#	#	#
Alt Del	--	--	--
Alt SysRq	**	(no key)	(no key)
Alt Key 45	(no key)	--	--
Alt Enter	(no key)	--	A6/00
Alt /	--	--	A4/00
Alt Print Screen	(no key)	**	**
Alt Pause	(no key)	**	**
Alt Home	(no key)	--	97/00
Alt Up	(no key)	--	98/00
Alt PageUp	(no key)	--	99/00
Alt Left	(no key)	--	9B/00
Alt Right	(no key)	--	9D/00
Alt End	(no key)	--	9F/00
Alt Down	(no key)	--	A0/00
Alt PageDown	(no key)	--	A1/00
Alt Insert	(no key)	--	A2/00
Alt Delete	(no key)	--	A3/00

# See the following page for use of Alt key with number keys.  
\*\* These combinations do not provide a keystroke for the application presently running but perform some other action. They are not put on the INT 16H queue.  
-- These combinations have no function and are ignored.

Figure 4-6 (Part 2 of 2). Alt Keyboard Keystrokes

For all keyboards, the numeric keypad can be used in combination with the Alt key to input any ASCII character. The scan code (always 00) and character code are returned after the Alt key is released. For example, pressing Alt and Keypad 1, then releasing Alt returns scan code/character code combination hex 00/01; pressing Alt and Keypad 255, then releasing Alt returns scan code/character code combination hex 00/FF.

**Notes:**

---

# Glossary

This glossary includes terms and definitions from the *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699.

**adapter.** An auxiliary device or unit used to extend the operation of another 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. Synonymous with alphameric.

**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.

**analog.** (1) Pertaining to data in the form of continuously variable physical quantities. (2) Contrast with digital.

**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.

**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).

**BASIC.** Beginner's all-purpose symbolic instruction code.

**basic input/output system (BIOS).**

The feature 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 different 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.

**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.

**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.

**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.

**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.

**cathode ray tube (CRT).** A vacuum tube in which a stream of electrons is projected onto a fluorescent screen producing a luminous spot. The location of the spot can be controlled.

**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 different 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.

**CMOS.** Complementary metal oxide semiconductor.

**code.** (1) A set of unambiguous rules specifying the manner in which data may be represented in a discrete form. Synonymous with coding scheme. (2) A set of items, such as abbreviations, representing the members of another set. (3) To represent data or a computer program in a symbolic form that can be accepted by a data processor. (4) Loosely, one or more computer programs, or part of a computer program.

**complement.** A number that can be derived from a specified number by subtracting it from a second specified number.

**complementary metal oxide semiconductor (CMOS).** A logic

circuit family that uses very little power. It works with a wide range of power supply voltages.

**computer.** A functional unit that can perform substantial computation, including numerous arithmetic operations or logic operations, without human intervention during a run.

**computer program.** A sequence of instructions suitable for processing by a computer.

**computer word.** A word stored in one computer location and capable of being treated as a unit.

**configuration.** (1) The arrangement of a computer system or network as defined by the nature, number, and the chief characteristics of its functional units. More specifically, the term configuration may refer to a hardware configuration or a software configuration. (2) The devices and programs that make up a system, subsystem, or network.

**cps.** Characters per second.

**CRC.** Cyclic redundancy check.

**CRT.** Cathode ray tube.

**cursor.** (1) In computer graphics, a movable marker that is used to indicate position on a display. (2) A displayed symbol that acts as a marker to help the user locate a point in text, in a system command, or in storage. (3) A movable spot of light on the screen of a display device, usually indicating where the next character is to be entered, replaced, or deleted.

**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.

**data.** (1) A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by human or automatic means. (2) Any representations, such as characters or analog quantities, to which meaning is, or might be assigned.

**dc.** Direct current.

**device driver.** A device handler routine in the operating system.

**digit.** (1) A graphic character that represents an integer; for example, one of the characters 0 to 9. (2) A symbol that represents one of the non-negative integers smaller than the radix. For example, in decimal notation, a digit is one of the characters 0 to 9.

**digital.** (1) Pertaining to data in the form of digits. (2) Contrast with analog.

**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 semirigid protective jacket, in which the disk is permanently enclosed. Synonymous with flexible 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. (4) See cathode ray tube display.

**DMA.** Direct memory access.

**duplex.** (1) In data communication, pertaining to a simultaneous two-way independent transmission in both directions. d. Contrast with half-duplex.

**ECC.** Error checking and correction.

**enable.** To initiate the operation of a circuit or device.

**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.

**EOT.** End-of-transmission.

**ETB.** End-of-transmission-block.

**ETX.** End-of-text.

**FCC.** Federal Communications Commission.

**field.** (1) In a record, a specified area used for a particular category of data. (2) In a data base, the smallest unit of data that can be referred to.

**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.

**font.** A family or assortment of characters of a given size and style; for example, 10 point Press Roman medium.

**foreground.** (1) In multiprogramming, the environment in which high-priority programs are executed. (2) On a color display screen, the characters as opposed to the background.

**format.** The arrangement or layout of data on a data medium.

**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.

**graphic.** A symbol produced by a process such as handwriting, drawing, or printing.

**half-duplex.** (1) In data communication, pertaining to an alternate, one way at a time, independent transmission. (2) Contrast with duplex.

**hardware.** Physical equipment used in data processing, as opposed to programs, procedures, rules, and associated documentation.

**head.** A device that reads, writes, or erases data on a storage medium; for example, a small electromagnet used to read, write, or erase data on a magnetic disk.

**hertz (Hz).** A unit of frequency equal to one cycle per second.

**hex.** Common abbreviation for hexadecimal.

**hexadecimal.** (1) Pertaining to a selection, choice, or condition that has 16 possible different values or states. These values or states are usually symbolized by the ten digits 0 through 9 and the six letters A through F. (2) Pertaining to a fixed radix numeration system having a radix of 16.

**Hz.** Hertz

**image.** A fully processed unit of operational data that is ready to be transmitted to a remote unit; when loaded into control storage in the remote unit, the image determines the operations of the unit.

**indicator.** (1) A device that may be set into a prescribed state, usually according to the result of a previous process or on the occurrence of a specified condition in the equipment, and that usually gives a visual or other indication of the existence of the prescribed state, and that may in some cases be used to determine the selection among alternative processes; for example, an overflow indicator. (2) An item of data that may be interrogated to determine whether a particular condition has been satisfied in the execution of a computer program; for example, a switch indicator, an overflow indicator.

**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. In the English language, "input/output" may be used in place of such terms as "input/output data," "input/output signal," and "input/output terminals," when such usage is clear in a given context. (2) Pertaining to a device whose parts can be performing an input process and an output process at the same time. (3) Pertaining to either input or output, or both.

**instruction.** In a programming language, a meaningful expression that specifies one operation and identifies its operands, if any.

**intensity.** In computer graphics, the amount of light emitted at a display point

**interface.** A device that alters or converts actual electrical signals between distinct devices, programs, or systems.

**interleave.** To arrange parts of one sequence of things or events so that they alternate with parts of one or more other sequences of the same nature and so that each sequence retains its identity.

**interrupt.** (1) A suspension of a process, such as the execution of a computer program, caused by an event external to that process, and performed in such a way that the process can be resumed. (2) In a data transmission, to take an action at a receiving station that causes

the transmitting station to terminate a transmission. (3) Synonymous with interruption.

**I/O.** Input/output.

**I/O area.** Synonym for buffer.

**joystick.** In computer graphics, a lever that can pivot in all directions and that is used as a locator device.

**Kb.** 1024 bytes.

**latch.** (1) A simple logic-circuit storage element. (2) A feedback loop in sequential digital circuits used to maintain a state.

**LED.** Light-emitting diode.

**light-emitting diode (LED).** A semiconductor device that gives off visible or infrared light when activated.

**load.** In programming, to enter data into storage or working registers.

**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.

**Mb.** 1 048 576 bytes.

**mega (M).** Prefix 1,000,000.

**megahertz (MHz).** 1,000,000 hertz.

**memory.** Term for main storage.

**meter (m).** A unit of length (equivalent to 39.37 inches).

**MHz.** Megahertz; 1,000,000 hertz.

**micro.** Prefix 0.000,001.

**microprocessor.** An integrated circuit that accepts coded instructions for execution; the instructions may be entered, integrated, or stored internally.

**microsecond.** 0.000,001 second.

**milli (m).** Prefix 0.001.

**milliampere (mA).** 0.001 ampere.

**millisecond (ms).** 0.001 second.

**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.

**module.** (1) A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading. (2) A packaged functional hardware unit

designed for use with other components.

**monitor.** Synonym for cathode ray tube display (CRT display).

**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.

**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.

**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.

**overflow.** Loss of data because a receiving device is unable to accept data at the rate it is transmitted.

**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 check.** (1) A redundancy check that uses a parity bit. (2) Synonymous with odd-even check.

**PEL.** Picture element.

**picture element (PEL).** The smallest displayable unit on a display.

**pointers.** A double word entity that specifies an address.

**port.** An access point for data entry or exit.

**processor.** (1) In a computer, a functional unit that interprets and executes instructions. (2) A functional unit, a part of another unit such as a terminal or a processing unit, that interprets and executes instructions. (3) Deprecated term for processing program. (4) See microprocessor.

**program.** (1) A series of actions designed to achieve a certain result. (2) A series of instructions telling the computer how to handle a problem or task. (3) To design, write, and test computer programs.

**programmable read-only memory (PROM).** A read-only memory that can be programmed by the user.

**programming language.** (1) An artificial language established for expressing computer programs. (2) A set of characters and rules with meanings assigned prior to their use, for writing computer programs.

**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.

**radio frequency (RF).** An ac frequency that is higher than the highest audio frequency. So called because of the application to radio communication.

**RAM.** Random access memory. Read/write memory.

**random access memory (RAM).** Read/write memory.

**raster.** In computer graphics, a predetermined pattern of lines that provides uniform coverage of a display space.

**read.** To acquire or interpret data from a storage device, from a data medium, or from another source.

**read-only memory (ROM).** A storage device whose contents cannot be modified. The memory is retained when power is removed.

**read/write memory.** A storage device whose contents can be modified. Also called RAM.

**register.** (1) A storage device, having a specified storage capacity such as a bit, a byte, or a computer word, and usually intended for a special purpose. (2) A storage device in which specific data is stored.

**reset/initialize.** Set Hardware to known state

**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.

**RF.** Radio frequency.

**RF modulator.** The device used to convert the composite video signal to the antenna level input of a home TV.

**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.

**RS-232-C.** A standard by the EIA for communication between computers and external equipment.

**run.** A single continuous performance of a computer program or routine.

**scaling.** In computer graphics, enlarging or reducing all or part of a display image by multiplying the coordinates of the image by a constant value.

**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) Pertaining to the sequential processing of the individual parts of a whole, such as the bits of a character or the characters of a word, using the same facilities for successive parts. (4) Contrast with parallel.

**setup.** (1) In a computer that consists of an assembly of individual computing units, the arrangement of interconnections between the units, and the adjustments needed for the computer to operate. (2) The preparation of the system for normal operation.

**software.** Computer programs, procedures, and rules concerned with the operation of a data processing system.

**source.** The origin of a signal or electrical energy.

**start-of-text (STX).** A transmission control character that precedes a text and may be used to terminate the message heading.

**stop bit.** (1) A signal to a receiving mechanism to wait for the next signal. (2) In a start-stop system, a signal following a character or block that prepares the receiving device for the reception of a subsequent character or block.

**storage.** (1) A storage device. (2) A device, or part of a device, that can retain data. (3) The retention of data in a storage device.

**STX.** Start-of-text.

**text.** In ASCII and data communication, a sequence of characters treated as an entity if preceded and terminated by one STX and one ETX transmission control character, respectively.

**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.

**window.** (1) A predefined part of the virtual space. (2) The visible area of a viewplane.

**word.** (1) A character string or a bit string considered as an entity. (2) See computer word.

**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.

**Notes:**

---

# Index

## A

- activate/deactivate internal modem power 2-81
- adapter ROM 4-12
- alternate disk reset 2-66
- alternate select 2-32
- application guidelines 4-19
  - BIOS function calls 4-21
  - hardware interrupts 4-19
  - operating system function calls 4-21
  - programming considerations 4-21
- asynchronous communications interrupt (14H) 2-70
  - extended communications port control 2-73
  - extended initialize 2-72
  - initialize the communications port 2-70
  - read status 2-72
  - receive character 2-71
  - send character 2-71

## B

- BASIC 4-21
- BIOS data area 3-3
- BIOS data area, extended 3-15
- BIOS function calls 4-21
- BIOS level determination 1-4
- bootstrap loader interrupt (19H) 2-114

## C

- character code combinations 4-24
- character generator 2-24
- COBOL 4-21
- compatibility, video 4-14

## D

- data areas 1-4, 3-3, 3-15
- device busy 2-93
- device close 2-83
- device open 2-83
- diskette change line status 2-55
- diskette drive data area 3-6
- diskette drive parameter table 3-23
- diskette drive/fixed disk drive control data area 3-10
- diskette interrupt (13H) 2-48
  - diskette change line status 2-55
  - format desired track 2-51
  - read DASD type 2-54
  - read desired sectors into memory 2-50
  - read drive parameters 2-53
  - read status of last operation 2-49
  - reset diskette system 2-48
  - set DASD type for format 2-55
  - set media type for format 2-56
  - verify desired sectors 2-51
  - write desired sectors from memory 2-50

## E

- edge-sensitive interrupts 4-19
- enable/disable watchdog time-out 2-102
- equipment determination interrupt (11H) 2-46
- event wait 2-84
- extended BIOS data area 3-15
- extended communications port control 2-73
- extended initialize 2-72
- extended keyboard read 2-109
- extended keystroke status 2-109

extended memory size  
determine 2-89  
extended shift status 2-110

## F

fixed disk drive data area 3-8  
fixed disk drive parameter  
table 3-16  
fixed disk interrupt (13H) 2-58  
alternate disk reset 2-66  
format desired cylinder 2-62  
format desired cylinder and set  
bad sector flags 2-63  
format drive starting at desired  
cylinder 2-64  
format unit 2-68  
initialize drive pair  
characteristics 2-65  
park heads 2-68  
read DASD type 2-67  
read desired sectors into  
memory 2-60  
read drive parameters 2-64  
read status of last  
operation 2-60  
recalibrate 2-67  
reset disk system 2-59  
seek 2-66  
test drive ready 2-66  
verify desired sectors 2-61  
write desired sectors from  
memory 2-61  
format desired cylinder 2-62  
format desired cylinder and set bad  
sector flags 2-63  
format desired track 2-51  
format drive starting at desired  
cylinder 2-64  
format unit 2-68  
format unit periodic interrupt 2-76  
FORTRAN 4-21  
function calls 4-21

## G

gap length 4-22  
guidelines, application 4-19  
BIOS function calls 4-21  
hardware interrupts 4-19  
operating system function  
calls 4-21  
programming  
considerations 4-21

## H

hardware interrupts  
edge-sensitive interrupts 4-19  
I/O commands 4-20  
level-sensitive interrupts 4-19  
STI instructions 4-20  
hardware specific video mode  
characteristics 2-11

## I

I/O commands 4-20  
IBM C 4-21  
identification, system 4-18  
initialize drive pair  
characteristics 2-65  
initialize the communications  
port 2-70  
initialize the printer port 2-111  
INT 1AH (system-timer and real-time  
clock services interrupt) 2-115  
read real-time clock alarm time  
and status 2-119  
read real-time clock date 2-117  
read real-time clock time 2-116  
read system-timer day  
counter 2-119  
read system-timer time  
counter 2-115  
reset real-time clock  
alarm 2-118  
set real-time clock activated  
power-on mode 2-118  
set real-time clock alarm 2-118  
set real-time clock date 2-117

- set real-time clock time 2-117
- set system-timer day counter 2-120
- set system-timer time counter 2-116
- set up sound multiplexer 2-120
- INT 10H (video) 2-10
  - alternate select 2-32
  - character generator 2-24
  - load LCD character font/set LCD high-intensity substitute 2-37
  - read attribute/character at current cursor position 2-17
  - read current video state 2-19
  - read cursor position 2-15
  - read dot 2-19
  - read light pen position 2-15, 2-16
  - read/write display combination code 2-39
  - return functionality/state information 2-40
  - return physical display parameters for active display 2-38
  - save/restore video state 2-44
  - scroll active page down 2-16
  - scroll active page up 2-16
  - set color palette 2-18
  - set cursor position 2-15
  - set cursor type 2-14
  - set mode 2-11
  - set palette registers 2-20
  - write attribute/character at current cursor position 2-17
  - write character at current cursor position 2-18
  - write dot 2-18
  - write string 2-36
  - write teletype to active page 2-19
- INT 13H (diskette) 2-48
  - diskette change line status 2-55
  - format desired track 2-51
  - read DASD type 2-54
  - read desired sectors into memory 2-50
  - read drive parameters 2-53
  - read status of last operation 2-49
  - reset diskette system 2-48
  - set DASD type for format 2-55
  - set media type for format 2-56
  - verify desired sectors 2-51
  - write desired sectors from memory 2-50
- INT 13H (fixed disk) 2-58
  - alternate disk reset 2-66
  - format desired cylinder 2-62
  - format desired cylinder and set bad sector flags 2-63
  - format drive starting at desired cylinder 2-64
  - format unit 2-68
  - initialize drive pair characteristics 2-65
  - park heads 2-68
  - read DASD type 2-67
  - read desired sectors into memory 2-60
  - read drive parameters 2-64
  - read status of last operation 2-60
  - recalibrate 2-67
  - reset disk system 2-59
  - seek 2-66
  - test drive ready 2-66
  - verify desired sectors 2-61
  - write desired sectors from memory 2-61
- INT 14H (asynchronous communications) 2-70
  - extended communications port control 2-73
  - extended initialize 2-72
  - initialize the communications port 2-70
  - read status 2-72
  - receive character 2-71
  - send character 2-71
- INT 15H (system services) 2-74
  - activate/deactivate internal modem power 2-81
  - device busy 2-93
  - device close 2-83

- device open 2-83
- enable/disable watchdog
  - time-out 2-102
- event wait 2-84
- extended memory size
  - determine 2-89
- format unit periodic
  - interrupt 2-76
- interrupt complete 2-94
- joystick support 2-85
- keyboard intercept 2-82
- move block 2-87
- pointing device BIOS
  - interface 2-97
- power-on self-test error
  - log 2-77
- program termination 2-84
- programmable option select (POS) 2-102
- read blocks from cassette 2-75
- read system status 2-80
- read/modify profiles 2-78
- request system power-off 2-80
- return extended BIOS data area
  - segment address 2-96
- return system configuration
  - parameters 2-94
- switch processor to protected mode 2-90
- system request key
  - pressed 2-86
- turn cassette motor off 2-75
- turn cassette motor on 2-74
- wait 2-86
  - wait for external event 2-79
- write blocks to cassette 2-76
- INT 16H (keyboard) 2-104
  - extended keyboard read 2-109
  - extended keystroke
    - status 2-109
  - extended shift status 2-110
  - keyboard click
    - adjustment 2-108
  - keyboard read 2-106
  - keyboard write 2-108
  - keystroke status 2-106
  - set typematic rate 2-107
  - shift status 2-107
- INT 17H (printer) 2-111
  - initialize the printer port 2-111
  - print character 2-111
  - read status 2-112
- interrupt complete 2-94
- interrupt sharing 4-3
  - implementation information 4-7
  - interrupt request 4-4
  - interrupt-sharing chaining
    - structure and signature 4-6
  - interrupt-sharing software
    - requirements 4-4
    - precautions 4-3
    - ROM considerations 4-7
- interrupts 2-3
  - asynchronous communications (14H) 2-70
  - bootstrap loader (19H) 2-114
  - diskette (13H) 2-48
  - edge-sensitive interrupts 4-19
  - equipment determination (11H) 2-46
  - fixed disk (13H) 2-58
  - I/O commands 4-20
  - INT 02H 2-4
  - INT 05H (print screen) 2-6
  - INT 08H (system timer) 2-7
  - INT 09H (keyboard) 2-8
  - INT 1AH (system-timer and real-time clock services) 2-115
  - INT 10H (video) 2-10
  - INT 11H (equipment determination) 2-46
  - INT 12H (memory size determination) 2-47
  - INT 13H (diskette) 2-48
  - INT 13H (fixed disk) 2-58
  - INT 14H (asynchronous communications) 2-70
  - INT 15H (system services) 2-74
  - INT 16H (keyboard) 2-104
  - INT 17H (printer) 2-111
  - INT 19H (bootstrap loader) 2-114
  - INT 70H (real-time clock interrupt) 2-122
  - keyboard (16H) 2-104

- keyboard interrupt (09H) 2-8
- level-sensitive interrupts 4-19
- memory size determination (12H) 2-47
- NMI (02H) 2-4
- nonmaskable interrupt (02H) 2-4
- print screen (05H) 2-6
- printer (17H) 2-111
- real-time clock (70H) 2-122
- real-time clock services (1AH) 2-115
- STI instructions 4-20
- system services (15H) 2-74
- system timer (08H) 2-7
- system-timer services (1AH) 2-115
- video (10H) 2-10
- introduction 1-3

## J

- joystick support 2-85

## K

- keyboard click adjustment 2-108
- keyboard data area 3-4, 3-9, 3-11
- keyboard intercept 2-82
- keyboard interrupt (09H) 2-8
- keyboard interrupt (16H) 2-104
  - extended keyboard read 2-109
  - extended keystroke status 2-109
  - extended shift status 2-110
- keyboard click adjustment 2-108
- keyboard read 2-106
- keyboard write 2-108
- keystroke status 2-106
- set typematic rate 2-107
- shift status 2-107
- keyboard read 2-106
- keyboard write 2-108
- keystroke status 2-106

## L

- level-sensitive interrupts 4-19
- load LCD character font/set LCD high-intensity substitute 2-37

## M

- memory size data area 3-4
- memory size determination interrupt (12H) 2-47
- mode switching, video 4-15
- model byte 2-95, 4-18
- move block 2-87
- multitasking 4-16
  - device driver code 4-16
  - post 4-17
  - serialization 4-16
  - startup 4-16
  - time-outs, multitasking 4-17
  - wait 4-16, 4-17

## N

- NMI (nonmaskable interrupt) 2-4
- nonmaskable interrupt (02H) 2-4

## O

- operating system function calls 4-21

## P

- parameter passing 1-3
- park heads 2-68
- Pascal 4-21
- PC Convertible display types 2-38
- pointing device BIOS interface 2-97
- power-on self-test error log 2-77
- presence test, video 4-14
- print character 2-111
- print screen interrupt (05H) 2-6
- printer interrupt (17H) 2-111
  - initialize the printer port 2-111
  - print character 2-111

- read status 2-112
- printer port base address data area 3-3
- printer time-out value data area 3-8
- program termination 2-84
- programmable option select (POS) 2-102
- programming considerations 4-21
  - BASIC 4-21
  - COBOL 4-21
  - FORTRAN 4-21
  - gap length 4-22
  - IBM C 4-21
  - Pascal 4-21

## R

- read attribute/character at current cursor position 2-17
- read blocks from cassette 2-75
- read current video state 2-19
- read cursor position 2-15
- read DASD type 2-54, 2-67
- read desired sectors into memory 2-50, 2-60
- read dot 2-19
- read drive parameters 2-53, 2-64
- read light pen position 2-15
- read real-time clock alarm time and status 2-119
- read real-time clock date 2-117
- read real-time clock time 2-116
- read status 2-72, 2-112
- read status of last operation 2-49, 2-60
- read system status 2-80
- read system-timer day counter 2-119
- read system-timer time counter 2-115
- read/modify profiles 2-78
- read/write display combination code 2-39
- real-time clock data area 3-11
- real-time clock interrupt (70H) 2-122
- recalibrate 2-67

- receive character 2-71
- request system power-off 2-80
- reset disk system 2-59
- reset diskette system 2-48
- reset real-time clock alarm 2-118
- return extended BIOS data area segment address 2-96
- return functionality/state information 2-40
- return physical display parameters for active display 2-38
- return system configuration parameters 2-94
- revision level byte 2-95, 4-18
- ROM tables 1-4, 3-3, 3-16
- RS-232-C port base address data area 3-3
- RS-232-C time-out value data area 3-9

## S

- save pointer data area 3-12
- save/restore video state 2-44
- scan code/character code combinations 4-24
- scroll active page down 2-16
- scroll active page up 2-16
- secondary save pointer data area 3-13
- seek 2-66
- select active display page 2-16
- send character 2-71
- set color palette 2-18
- set cursor position 2-15
- set cursor type 2-14
- set DASD type for format 2-55
- set media type for format 2-56
- set mode 2-11
- set palette registers 2-20
- set real-time clock activated power-on mode 2-118
- set real-time clock alarm 2-118
- set real-time clock date 2-117
- set real-time clock time 2-117
- set system-timer day counter 2-120

- set system-timer time counter 2-116
- set typematic rate 2-107
- set up sound multiplexer 2-120
- sharing, interrupt 4-3
  - implementation information 4-7
  - interrupt request 4-4
  - interrupt-sharing chaining structure and signature 4-6
  - interrupt-sharing software requirements 4-4
  - precautions 4-3
  - ROM considerations 4-7
- shift status 2-107
- STI instructions 4-20
- submodel byte 2-95, 4-18
- summary of interrupt functions
  - INT 1AH (system-timer and real-time clock services) 2-115
  - INT 10H (video) 2-10
  - INT 13H (diskette) 2-48
  - INT 13H (fixed disk) 2-58
  - INT 14H (asynchronous communications) 2-70
  - INT 15H (system services) 2-74
  - INT 16H (keyboard) 2-104
  - INT 17H (printer) 2-111
- switch processor to protected mode 2-90
- system data area 3-7, 3-8
- system equipment data area 3-4
- system groups 1-5
- system identification 4-18
- system request key pressed 2-86
- system services interrupt (15H) 2-74
  - activate/deactivate internal modem power 2-81
  - device busy 2-93
  - device close 2-83
  - device open 2-83
  - enable/disable watchdog time-out 2-102
  - event wait 2-84
  - extended memory size determine 2-89
  - format unit periodic interrupt 2-76
  - interrupt complete 2-94
  - joystick support 2-85
  - keyboard intercept 2-82
  - move block 2-87
  - pointing device BIOS interface 2-97
  - power-on self-test error log 2-77
  - program termination 2-84
  - programmable option select (POS) 2-102
  - read blocks from cassette 2-75
  - read system status 2-80
  - read/modify profiles 2-78
  - request system power-off 2-80
  - return extended BIOS data area segment address 2-96
  - return system configuration parameters 2-94
  - switch processor to protected mode 2-90
  - system request key pressed 2-86
  - turn cassette motor off 2-75
  - turn cassette motor on 2-74
  - wait 2-86
  - wait for external event 2-79
  - write blocks to cassette 2-76
- system timer interrupt (08H) 2-7
- system-timer and real-time clock services interrupt (1AH) 2-115
  - read real-time clock alarm time and status 2-119
  - read real-time clock date 2-117
  - read real-time clock time 2-116
  - read system-timer day counter 2-119
  - read system-timer time counter 2-115
  - reset real-time clock alarm 2-118
  - set real-time clock activated power-on mode 2-118
  - set real-time clock alarm 2-118
  - set real-time clock date 2-117
  - set real-time clock time 2-117

- set system-timer day counter 2-120
- set system-timer time counter 2-116
- set up sound multiplexer 2-120
- system-timer data area 3-7

## T

- test drive ready 2-66
- turn cassette motor off 2-75
- turn cassette motor on 2-74

## V

- verify desired sectors 2-51, 2-61
- video compatibility 4-14
  - mode switching 4-15
  - presence test 4-14
- video control data area 3-6, 3-9
- video interrupt (10H) 2-10
  - alternate select 2-32
  - character generator 2-24
  - load LCD character font/set LCD high-intensity substitute 2-37
  - read attribute/character at current cursor position 2-17
  - read current video state 2-19
  - read cursor position 2-15
  - read dot 2-19
  - read light pen position 2-15, 2-16
  - read/write display combination code 2-39
  - return functionality/state information 2-40

- return physical display parameters for active display 2-38
- save/restore video state 2-44
- scroll active page down 2-16
- scroll active page up 2-16
- set color palette 2-18
- set cursor position 2-15
- set cursor type 2-14
- set mode 2-11
- set palette registers 2-20
- write attribute/character at current cursor position 2-17
- write character at current cursor position 2-18
- write dot 2-18
- write string 2-36
- write teletype to active page 2-19
- video mode switching 4-15
- video modes 2-11
- video presence test 4-14

## W

- wait 2-86, 4-16
- wait for external event 2-79
- write attribute/character at current cursor position 2-17
- write blocks to cassette 2-76
- write character at current cursor position 2-18
- write desired sectors from memory 2-50, 2-61
- write dot 2-18
- write string 2-36
- write teletype to active page 2-19



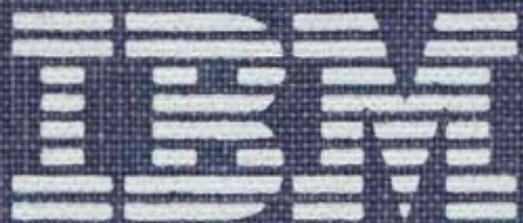


© Copyright  
International Business  
Machines Corporation, 1987  
All Rights Reserved

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.

84X1514



Personal System/2™  
and Personal Computer  
BIOS Interface Technical Reference

68X2260  
S68X-2260-00