Phoenix

CBIOS for IBM® PS/2® Computers and Compatibles

Addison Wesley

51804

Phoenix Technical Reference Series

CBIOS for IBM® PS/2® Computers and Compatibles

The Complete Guide to ROM-Based System Software for DOS

Phoenix Technologies Ltd.

Phoenix Technical Reference Series

CBIOS for IBM® PS/2® Computers and Compatibles

The Complete Guide to ROM-based System Software for DOS

Phoenix Technologies Ltd.

Addison-Wesley Publishing Company, Inc.
Reading, Massachusetts Menlo Park, California New York
Don Mills, Ontario Wokingham, England Amsterdam Bonn
Sydney Singapore Tokyo Madrid San Juan

Limitation of Liability

Library of Congress Cataloging-in-Publication Data

CBIOS for IBM PS/2 computers and compatibles : the complete guide to
ROM-based system software for DOS / Phoenix Technologies, Ltd.
p. cm. -- (The Phoenix technical reference series)
Includes index.
ISBN 0-201-51804-X
1. IBM Personal System/2 (Computer system) 2. Systems software.
3. Read-only storage. 4. Computer input-output equipment.
I. Phoenix Technologies, Ltd. II. Title: CBIOS. III. Series.
QA76.8.I25963C35 1989 005.4'469--dc19 89-6448

Cover design by Hannus Design Associates
Text design by Phoenix Technologies Ltd.
Set in 10-point Modern by Phoenix Technologies Ltd.

ABCDEFGHIJ-AL-89
First printing, June, 1989

Trademarks

This manual acknowledges the following trademarks:

Ashton-Tate and Framework are registered trademarks of Ashton-Tate Corporation

AST is a registered trademark of AST Research, Inc.

CHIPS/250 and CHIPS/280 are trademarks of Chips and Technologies, Inc.

AT, IBM, Personal Systems/2, PS/2, and PC/AT are registered trademarks of the International Business Machines Corporation. PC-DOS, PC/XT, VGA, CGA, MCA, MDA, EGA, OS/2, and Micro Channel are trademarks of the International Business Machines Corporation.

Intel is a registered trademark of Intel Corporation.

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

Motorola is a registered trademark of Motorola Semiconductor Products, Inc.

MS, MS-DOS, and Microsoft are registered trademarks of the Microsoft Corporation.

NEC and Multisync are registered trademarks of Nippon Electric Corporation.

Quadram is a registered trademark of Quadram Corporation.

To Eric Enge, Stan Lyness, Paula Bishop, and the rest of the Phoenix Technologies Ltd. BIOS Engineering Department. Their tireless efforts have defined compatibility for the PC industry.

Table of Contents

About This Book

Acronyms and Abbreviations

Chapter 1 — The CBIOS

Chapter 2 — Hardware Environment

continued

Chapter 2 — Hardware Environment, Cont'd

Chapter 3 — System RAM Data

Chapter 4 — CMOS RAM Data

Chapter 5 — ROM BIOS Data

continued

Chapter 6 — Power-On Self Tests (POST)

Chapter 7 — INT 02h Nonmaskable Interrupt

Chapter 8 — INT 09h and INT 16h Keyboard Services

continued

Chapter 8 — INT 09h and INT 16h Keyboard Services, Cont'd

Chapter 9 — INT 10h Video Service

continued

Chapter 9 — INT 10h Video Service, Cont'd

Chapter 10 — INT 13h Diskette Service

Chapter 11 — INT 13h Fixed Disk Service

continued

Chapter 11 — INT 13h Fixed Disk Service, Cont'd

Chapter 12 — INT 14h Serial Communications Service

continued

Chapter 12 — INT 14h Serial Communications Service, Cont'd

Chapter 13 — INT 15h System Services

continued

Chapter 13 — INT 15h System Services, Cont'd

Chapter 14 — INT 17h Parallel Printer Service

Chapter 15 — INT 1Ah Time-of-Day Service

Chapter 16 — Single Function CBIOS Services

continued

Chapter 16 — Single Function CBIOS Services, Cont'd

Appendix A — CBIOS Error Messages

Appendix B — Keyboard/Mouse Controller

Appendix C — Hardware Environment for 8086-Based Systems

continued

Appendix C — Hardware Environment for 8086-Based Systems, Cont'd

Glossary

Additional Resources

Index

CBIOS for IBM® PS/2® Computers and Compatibles

The Complete Guide to ROM-based System Software for DOS

About This Book

What this book is about

CBIOS for IBM PS/2 Computers and Compatibles is a detailed technical reference that describes the CBIOS, the portion of the PS/2 ROM BIOS designed to support single-tasking operating systems such as MS-DOS. The information provided in this book is applicable to all Micro Channel Architecture-based (MCA) and non-MCA-based IBM PS/2 and compatible computers, ranging from Models 25 and 30 through Model 80.

Who should read this book

CBIOS for IBM PS/2 Computers and Compatibles provides the most comprehensive source of information about the IBM PS/2 and compatible CBIOS available today. It can be used by anyone interested in learning more about their PS/2 or compatible computer.

Applications programmers and device driver developers will find a thorough discussion of each CBIOS device service and function, including the CBIOS VGA-compatible video service. In addition, *CBIOS for IBM PS/2 Computers and Compatibles* defines, to the bit level, all IBM PS/2 and compatible I/O port addresses and CBIOS-related CMOS RAM and system RAM data.

Hardware technicians and other system troubleshooters will appreciate the complete list of all CBIOS-generated error messages, their likely causes, and suggested solutions.

What we assume you know

This book assumes a basic knowledge of 80x86 assembly language programming concepts, PC architecture, and operating system concepts. If you are new to these subjects, use this book along with some of the excellent introductory books listed at the end of this book.

How to find information

CBIOS for IBM PS/2 Computers and Compatibles is organized into two major parts. Chapters 1–6 provide a general introduction to CBIOS concepts and describe how to use CBIOS services. Chapters 7–16 describe the individual CBIOS services and serve as a technical reference. Appendix A provides a comprehensive list of potential error messages. Appendix B provides information on the 8042–based Keyboard/Mouse Controller. Appendix C describes the hardware environment needed to implement 8086–based systems. Most readers will want to read chapters 1–6 first to get a basic grasp of the CBIOS features before turning to the individual service descriptions.

Each service–specific chapter (7–16) is organized in a similar fashion. There is a description of the service and the hardware environment of this service, a discussion on how errors are handled for this service, and complete descriptions of each service function. The function descriptions include an explanation of what the function does, the function's required inputs and outputs, and any special programming considerations or hardware "gotcha's" associated with the function.

Other volumes in this series

CBIOS for IBM PS/2 Computers and Compatibles is one of several volumes about BIOS software in the Phoenix Technical Reference Series published by Addison–Wesley. Other volumes are:

ABIOS for IBM PS/2 Computers and Compatibles — a complete technical reference describing the portion of a PS/2 BIOS that supports multitasking operating systems such as OS/2.

System BIOS for IBM PC/XT/AT Computers and Compatibles — a complete technical reference for the BIOS in all standard architecture computers.

The volumes of this series are a natural companion for anyone who owns and programs an IBM PC, XT, AT, or PS/2 model, or any compatible system.

Acknowledgments

No writing project as large as this can be completed without significant contributions from many individuals. We would like to acknowledge the Phoenix employees whose time and effort helped make this book possible.

First and foremost, we gratefully acknowledge the vision and technical skill of Neil Colvin, founder, CEO, and Chief Scientist of Phoenix Technologies Ltd. We also gratefully acknowledge the expert guidance of Lance Hansche, President of Phoenix Technologies Ltd. Without Neil and Lance's leadership, Phoenix would not be in the position of technological preeminence it now enjoys.

In the marketing area, Phoenix Vice President Rich Levandov's foresight and support made this book possible. And Product Marketing Manager Henry Suwinsky's tactful and dauntless guidance shepherded this project through all phases of its existence.

On the technical side, we acknowledge the Phoenix engineers who developed the Phoenix PS/2 BIOS. Specifically, we gratefully acknowledge the efforts of Eric Enge, Director of PC Product Engineering, Paula Bishop, and Stan Lyness, who have been endlessly patient with us and tireless in their efforts to ensure quality PS/2-compatible BIOS products and documentation. We also must acknowledge the efforts expended by other Phoenix engineers, including Bruce Cairns, Paul Chicoine, Greg Honsa, Suzanne Laferriere, Malcolm Pordes, Debbie Schultz, and Trevor Western.

In the production of this volume, many people contributed significantly. Chief among these were Kathy Schiff, Manager of Technical Communications, who provided editorial guidance and direction. Writers Marianne Adams and Dr. George Elliott Tucker contributed critical and essential writing, editing, and production assistance. And last but not least, a special thanks goes to Sandie Zierak, Chief Production Coordinator, for her invaluable contributions in the fields of graphics, document design, and document production.

The Authors

Mike Boston
Paul Narushoff

Phoenix Technologies Ltd.
Norwood, MA
March, 1989

Acronyms and Abbreviations

The following abbreviations and acronyms are used in this manual:

ASCII	American Standard Code for Information Interchange
ASIC	Application-Specific Integrated Circuit
b	Binary
BCD	Binary coded decimal
BIOS	Basic input/output system
bps	Bits per second
CDA	Common data area
CGA	Color graphics adapter
CRC	Cyclic redundancy check
CMOS	Complementary metal oxide semiconductor
DMA	Direct memory access
DSR	Device service routine
ECC	Error checking and correction
EGA	Enhanced graphics adapter
EOI	End of interrupt
ESDI	Enhanced small device interface
FTT	Function transfer table
h	Hexadecimal
INT	Interrupt
I/O	Input/Output
IRQ	Interrupt request line
ISR	Interrupt service routine
K	Kilobytes
Kbs	Kilobits per second
LID	Logical ID
LSB	Least significant byte
LSI	Large scale integration
M, MB	Megabytes
MCGA	Multicolor graphics array
MDA	Monochrome Display Adapter
MFM	Modified frequency modulation
MHz	Megahertz

continued

MSB	Most significant byte
NMI	Nonmaskable interrupt
OS/2	Operating System/2
PGA	Professional graphics adapter
POST	Power-on self test
PTL	Phoenix Technologies Ltd.
RAM	Random access memory
RB	Request block
RLL	Run length limited
ROM	Read-only memory
RTC	Real time clock
VGA	Video graphics array
VLSI	Very large scale integration

1

Chapter 1
The CBIOS

Overview

What is the PS/2 BIOS?

The ROM BIOS contained in Micro Channel Architecture–based IBM PS/2 and compatible computers performs the same function that all basic input/output systems do: it isolates the operating system and application programs from direct manipulation of the hardware.

When compared with the ROM BIOS contained in IBM PC XT/AT and compatible computers, however, the PS/2 ROM BIOS has one critical difference — it is designed to support two kinds of operating systems. The PS/2 ROM BIOS is divided into two discrete parts: the Advanced BIOS (ABIOS) and the Compatibility BIOS (CBIOS).

The CBIOS

The CBIOS portion of the PS/2 ROM BIOS provides IBM PS/2 and compatible computers backward compatibility with single–tasking, Intel 80x86 real address mode operating systems such as PC– or MS–DOS. As a result, the CBIOS consists of a superset of the services and functions available in the IBM PC/XT/AT ROM BIOS, and it interfaces with the operating system in the same well understood way.

continued

The ABIOS

The ABIOS portion of the PS/2 ROM BIOS provides IBM PS/2 and compatible computers with "forward" compatibility with multitasking, bimodal (real mode, protected mode, or both) operating systems, such as IBM OS/2. The ABIOS supports the same hardware devices that are supported by the CBIOS, but its interface and data structures are specifically constructed to facilitate the multitasking, bimodal nature of its design.

For more information on the ABIOS

For a complete treatment of the ABIOS, see *ABIOS for IBM PS/2 Computers and Compatibles* in this series.

In this chapter

This chapter presents a general introduction to the major concepts and design features of the CBIOS.

The following topics are discussed:

- CBIOS Compatibility
- CBIOS Memory Usage
- Interrupts
- CBIOS Software Interrupt Handlers
- Summary: The CBIOS Services

CBIOS Compatibility

Introduction

In addition to those BIOS routines responsible for testing/initializing system hardware, all BIOSs are essentially made up of a group of ROM-based input/output device services. Each of these routines services one subsystem of system hardware. Individual BIOS routines service the display, diskette drives, fixed disk drives, the keyboard, and so on.

Operating systems and applications programs written to the BIOS device service routines instead of directly to hardware I/O ports and control words are assured a high degree of hardware independence.

CBIOS, BIOS, and application program compatibility

The CBIOS contains a superset of the service routines contained in all versions of the ROM BIOS written for PC/AT-compatible computers.

Because of this, the CBIOS ensures that operating systems (e.g. MS-DOS) and applications programs written to these older BIOS versions will be compatible with the newer IBM PS/2-compatible MCA-based hardware.

CBIOS and PS/2

Besides service routines contained in previous IBM-compatible BIOSs, the CBIOS also contains all the added functionality required of the single-tasking component of the IBM PS/2 MCA-compatible BIOS.

The CBIOS device service routines support:

- All IBM VGA video modes and video functions
- Color and monochrome direct drive analog monitors
- A "mouse" or other pointing device
- Any of 32 IBM-defined fixed disk types
- 360K and 1.2 MB 5.25-inch diskette drives
- 720K and 1.44 MB 3.5-inch diskette drives
- Micro Channel Architecture-compatible adapter boards
- All PS/2-compatible system board components
- Programmable Option Select
- Password
- Reference Diskette

CBIOS Memory Usage

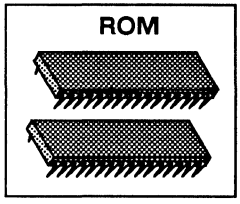
Introduction

Although the CBIOS routines reside in system Read-Only Memory (ROM), they also make use of information stored in CMOS RAM and System RAM.

The illustration on the following page outlines the location and use of CBIOS ROM, System RAM, and CMOS RAM.

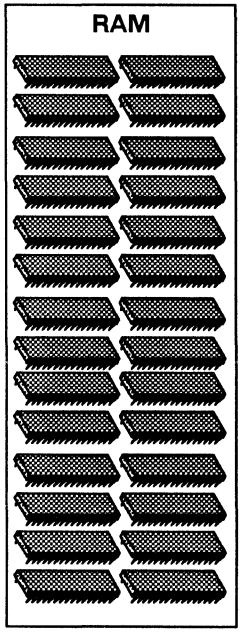
continued

CBIOS Memory Usage, Continued



ROM BIOS DATA

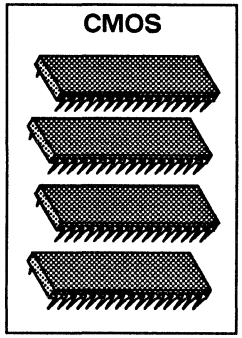
The CBIOS contains routines that test and initialize system hardware and boot the operating system. It also contains routines that service the microprocessor, the fixed disk, keyboard, display, and other peripheral devices with which the microprocessor must communicate. Each of the routines is stored in system Read-Only Memory (ROM). The CBIOS (and the ABIOS) are stored at address E0000h to FFFFFh in ROM.



SYSTEM RAM DATA

The CBIOS uses system RAM to store data that it needs to reference or update in the course of executing its device service routines. The CBIOS also uses system RAM to store the interrupt vectors.

- *Interrupt Vector Table* — An interrupt vector is the segment:offset address of a routine invoked by issuing the assembly language "INT" mnemonic. Interrupt vectors are stored in a "Table of Interrupt Vectors" in low memory beginning at 00:00h and ending at 00:3FFh. The CBIOS initializes those table entries that are reserved for BIOS use during POST. Each entry points to one ROM BIOS service routine. Other interrupt vector table entries are reserved for use by the operating system, the microprocessor, and by end users.
- *BIOS Data Area* — At POST, the CBIOS stores a set of data definitions in system RAM in absolute memory location 400h-500h. In the process of executing device service routines, the CBIOS refers to and updates this data. For example, the CBIOS updates 40:50h, the location of the cursor on the video page, each time the CBIOS routine "Set Cursor Position" is executed.
- *Extended BIOS Data Area* — The CBIOS uses the top 1K of system RAM to store additional data. The segment of this extended BIOS data area is located at 40:0Eh in the standard BIOS Data Area. The data located in this area is concerned with the functions that are CBIOS enhancements over the historical PC XT/AT BIOSs. For example, data definitions concerned with the "mouse" or other pointing devices are stored in the extended BIOS Data Area.



CMOS RAM DATA

The CBIOS uses nonvolatile CMOS RAM to store real time clock, system configuration, and system diagnostic information. Depending on the system configuration, the CBIOS can use up to two CMOS RAM storage areas.

- *CMOS Data Area* — 64 bytes (32 Words) of time-of-day and configuration data are located on the system real time clock chip. All implementations of the CBIOS make use of this information.
- *Extended CMOS Data Area* — On systems that incorporate more than four adapter slots, the CBIOS requires an additional 2K of CMOS RAM storage. This area is mainly used for configuration information associated with the additional adapter slots.

continued

For more information

To get more information on:

- **System RAM Data**, turn to Chapter 3 of this volume.
- **CMOS RAM Data**, turn to Chapter 4 of this volume.

Interrupts

Introduction

Computer systems based on the Intel 80x86 microprocessors are controlled mainly through the use of interrupts. Interrupts can be generated by the microprocessor, by system hardware, or by software.

When an interrupt occurs, control of the computer is transferred to an interrupt handling routine. Once the routine has executed, the processor's program counter and flag register are returned to their previous state.

Interrupt numbers

Every interrupt is assigned its own unique interrupt number, ranging from 00h to FFh.

By convention, certain ranges of interrupt numbers are reserved for special use. The interrupts numbers ranging from 20h–3Fh, for example, are reserved for DOS. The numbers from 60h–67h and F1h–FFh are reserved for user–created interrupt routines.

Interrupt vectors

An interrupt vector is the double word, segment:offset address of the routine assigned to a given interrupt number. Interrupt vectors are stored in a table in memory beginning at address 00:00h. The vector for INT 00h is stored at location 00:00h to 00:03h; the address for INT 02h is stored at address 00:04h to 00:07h, and so forth.

continued

How interrupt vectors are initialized

The CBIOS writes the Interrupt Vector Table to low memory as part of the power-on self test (POST) process, and it initializes the vector address of all those interrupts concerned with the BIOS. When DOS is booted, it, in turn, initializes all those interrupt vectors concerned with DOS. End users must take care to initialize the vectors concerned with their own interrupts.

How interrupts are invoked

How a given interrupt is invoked depends on what type it is.

The table below defines the range of interrupt numbers reserved for each interrupt type. It also summarizes how the various types of interrupts are invoked.

Type	Description
Processor	Microprocessor, or logical, interrupts are invoked by the processor as a result of some unusual program result, such as a program attempt to divide by zero. ▪ INTs 00h-04h are reserved for the processor.
Hardware	Hardware interrupts are invoked by peripheral devices by setting their respective Interrupt Request Line (IRQ). Each time a key is pressed, for example, the keyboard hardware generates a hardware interrupt. Hardware interrupts are vectored to Interrupt Service Routines (ISR) that generally reside in the BIOS. ▪ INTs 08h-0Fh and 70h-77h are reserved by the CBIOS.
Software	Software interrupts are invoked via the assembly language "INT" mnemonic. Most software interrupts are vectored to device service routines (DSR) located in the ROM BIOS or in DOS. There are three exceptions to this rule, however. ▪ CBIOS software INTs 1Dh, 1Eh, and 46h do not service a particular device, but act instead to return various ROM-resident hardware parameter tables. ▪ INTs 20h-3Fh are reserved for DOS. ▪ INTs 05h, 10h-1Ah, 1Dh-1Fh, 40h, 41h, 43h, and 46h are reserved for the BIOS.
User	User interrupts are a special class of software interrupt. They are invoked in the same way as software interrupts are. That is, through the INT mnemonic. ▪ INTs 60h-67h and INTs F1h-FFh are reserved for user interrupt routines.

continued

Interrupt vector table

The table below identifies each interrupt by function and type. Where applicable, it lists the interrupt vector address initialized by the CBIOS at POST.

Note: Further discussion of interrupts is limited to those interrupts directly related to the CBIOS. The column labeled "Ref. Chapter" tells where to turn for a full description of these interrupts.

INT	Function	Type	Vector	Ref. Chap.
00h	Divide by Zero	Logical		
01h	Single Step	Logical		
02h	Nonmaskable Interrupt (NMI)	Logical	FE2C3h	7
03h	Breakpoint	Logical		
04h	Overflow	Logical		
05h	Print Screen	Software	FFF54h	16
06h	Invalid Op Code	Hardware		
07h	Math Coprocessor Not Present	Hardware		
08h	System Timer	Hardware	FFEA5h	15
09h	Keyboard	Hardware	FE987h	8
0Ah	Invalid Task Segment State	Hardware		
0Bh	Serial Communications (COM2)	Hardware		
0Ch	Serial Communications (COM1)	Hardware		
0Dh	Parallel Printer (LPT2)	Hardware		
0Eh	Diskette	Hardware	FEF57h	8
0Fh	Parallel Printer (LPT1)	Hardware		
10h	Video	Software	FF065h	9
11h	Equipment List	Software	FF84Dh	16
12h	Memory Size	Software	FF841h	16
13h	Fixed Disk/Diskette	Software	FE3FEh	11
14h	Serial Communication	Software	FE739h	12
15h	System Services	Software	FF859h	13
16h	Keyboard	Software	FE84Eh	8
17h	Parallel Printer	Software	FEFD2h	14
18h	Load ROM BASIC	Software	F1C90h	16

continued

Interrupt vector table, cont'd

INT	Function	Type	Vector	Ref. Chap.
19h	Bootstrap Loader	Software	FE6F2h	16
1Ah	System Timer & Real-Time Clock	Software	FFE6Eh	15
1Bh	Keyboard Break	Software	FFF53h	8
1Ch	User Timer Tick	User	FFF53h	15
1Dh	Video Parameters Table	BIOS Table	FF0A4h	9
1Eh	Diskette Parameters Table	BIOS Table	FEFC7h	10
1Fh	Video Graphics Characters	User	F7F67h	9
20h-3Fh	Reserved for DOS			
40h	Diskette BIOS Revector	Software	FEC59h	10
41h	Fixed Disk Parameter Table	BIOS Table	FE401h	11
42h	EGA Default Video Driver	BIOS Table		
43h	Video Graphics Characters	User		9
44h-45h	Reserved			
46h	Fixed Disk Parameter Table	BIOS Table	FE401h	11
47h-49h	Reserved			
4Ah	User Alarm	User		15
4Bh-5Fh	Reserved			
60h-66h	Reserved for User Program Interrupts	User		1
67h	LIM EMS Driver			
68h-6Fh	Reserved			
70h	Real Time Clock	Hardware	F5124h	15
71h	Redirect Cascade	Hardware	F5266h	
72h-73h	Reserved			
74h	Pointing Device (Mouse)	Hardware	F4910h	13
75h	80287 Exception	User	F5257h	7
76h	Fixed Disk	Hardware	FE2C3h	11
77h-7Fh	Reserved			
80h-F0h	Reserved for BASIC	BASIC		16
F1h-FFh	Reserved for User Program Interrupts	User		

Software DSRs, functions, and parameters

Most of the CBIOS software interrupt handlers are capable of executing more than one device related routine, or function.

Individual functions within a given software interrupt handler are identified by a hex number, and they are selected by specifying that number in the AH register when the interrupt is invoked.

Note: Some software DSRs also contain subfunctions. Subfunctions are usually selected via the AL or the BL register.

Parameter passing

In addition to function numbers, all other parameters are passed to and from the CBIOS routines via the microprocessor registers. All CBIOS software interrupt handlers save the program counter registers, the flag register, and all other registers except the AX register and those registers that return a value to the caller.

continued

How functions are described

Throughout this manual, the functions performed by the CBIOS software interrupt handlers are described in terms of their input and output parameters. The sample description below is excerpted from Chapter 9, which describes the CBIOS Video Service functions.

Example of a function description

Function: AH = 02h Set Cursor Position

Description

The Set Cursor Position function sets the cursor position (in terms of row-by-column coordinates) for the display page indicated in BL. It saves the position as a two-byte row-by-column table entry in the cursor coordinates byte at 40:50h. Row and column coordinates are indicated in registers DH and DL respectively.

The Set Cursor Position function applies to both text and graphics video modes. In text modes, if the display page selected in BH is the active display page, the on-screen cursor will move to the coordinates indicated in registers DH and DL. In graphics modes, the cursor is invisible, but it is used to define a position on the screen.

Input/output

Input:	AH	=	02h
	BH	=	Display page number (see function AH = 05h)
	DH	=	Row (0 is top row of screen)
	DL	=	Column (0 is leftmost column)
Output:	AX	=	00h

continued

Example program

This assembly language program uses INT 10h Video Service function AH = 02h Set Cursor Position to move the cursor on video page 0 to row 3, column 14.

```
MOV AH, 2          ;Select "Set Cursor Position" function
MOV DH, 3          ;Input row parameter into DH register
MOV DL, 14         ;Input column parameter into DL register
INT 10H            ;Invoke INT 10h, CBIOS Video Service
```

Summary: The CBIOS Services

Introduction

Each CBIOS service executes at least one function. When a given CBIOS service is capable of executing more than one function, functions are selected by placing the proper function number in the AH register. Subfunctions are selected via either the AL register or the BL register.

The tables below briefly define each CBIOS service and list each CBIOS function and subfunction.

Print Screen Service		The CBIOS Print Screen Service prints the contents of the current video screen to printer number 0.
INT	**PARAMETER**	**FUNCTION**
05h	None	Print Screen

Video Service		The CBIOS Video Service provides I/O support for both color and monochrome analog monitors. They allow the display of all 17 IBM PS/2 video modes.
INT	**PARAMETER**	**FUNCTION**
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

continued

Introduction, cont'd

Video Service, cont'd

The CBIOS Video Service provides I/O support for both color and monochrome analog monitors. They allow the display of all 17 IBM PS/2 video modes.

INT	PARAMETER	FUNCTION
10h	AH = 0Ah	Write Character At Current 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:

Parameter	Subfunction
AL = 00h	Set individual palette register
AL = 01h	Set overscan register
AL = 02h	Set all palette registers and overscan
AL = 03h	Toggle intensify/blinking bit
AL = 04h–06h	Reserved
AL = 07h	Read individual palette register
AL = 08h	Read overscan register
AL = 09h	Read all palette registers and overscan
AL = 10h	Set individual color register
AL = 11h	Reserved
AL = 12h	Set block of color registers
AL = 13h	Select color page (not valid for mode 13h) BL = 00h Select paging mode BL = 01h Select page
AL = 14h	Reserved
AL = 15h	Read individual color register
AL = 16h	Reserved
AL = 17h	Read block of color registers
AL = 18h–19h	Reserved
AL = 1Ah	Read color paging status
AL = 1Bh	Sum color values to gray shades

continued

Introduction, cont'd

Video Service cont'd

The CBIOS Video Service provides I/O support for both color and monochrome analog monitors. They allow the display of all 17 IBM PS/2 video modes.

INT	PARAMETER	FUNCTION
10h	AH = 11h	CHARACTER GENERATOR:

Parameter	Subfunction
AL = 00h	User alpha load
AL = 01h	ROM 8x16 font
AL = 02h	ROM 8x8 double dot font
AL = 03h	Set block specifier (valid in alpha modes)
AL = 04h	ROM 8x16 font
AL = 10h	User alpha load (call after mode set)
AL = 11h	ROM 8x14 font (call after mode set)
AL = 12h	ROM 8x8 double dot font (call after mode set)
AL = 14h	ROM 8x16 font (call after mode set)
AL = 20h	Set user graphics characters pointer at INT 1Fh
AL = 21h	Set user graphics characters pointer at INT 43h
AL = 22h	ROM 8x14 font
AL = 23h	ROM 8x8 double dot font
AL = 24h	ROM 8x16 font
AL = 30h	Get font information

INT	PARAMETER	FUNCTION
10h	AH = 12h	ALTERNATE SELECT:

Parameter	Subfunction
BL = 10h	Return EGA information
BL = 20h	Select alternate print screen routine
BL = 30h	Select scan lines for alphanumeric modes
BL = 31h	Default palette loading during set mode
BL = 32h	Video
BL = 33h	Summing to gray shades
BL = 34h	Cursor emulation
BL = 35h	Display switch
BL = 36h	Video screen off/on

continued

Introduction, cont'd

Video Service cont'd

The CBIOS Video Service provides I/O support for both color and monochrome analog monitors. They allow the display of all 17 IBM PS/2 video modes.

INT	PARAMETER	FUNCTION
10h	AH = 13h	ALTERNATE SELECT: Parameter: AL = 00h — Subfunction: Cursor not moved AL = 01h — Cursor is moved AL = 02h — Cursor not moved (text modes only) AL = 03h — Cursor is moved (text modes only)
	AH = 14h–19h	Reserved
	AH = 13h	READ/WRITE DISPLAY COMBINATION CODE: Parameter: AL = 00h — Subfunction: Read display combination code AL = 01h — Write display combination code
	AH = 1Bh	Read/Write Display Combination Code
	AH = 1Ch	Save/Restore Video State
	AH = 1Dh–FFh	Reserved

Equipment List Service

The CBIOS Equipment List Service returns the system equipment list, as determined by the CBIOS POST routine.

INT	PARAMETER	FUNCTION
11h	None	Read Equipment List

continued

Introduction, cont'd

Memory Size Service		The CBIOS Memory Size Service returns the amount of available base memory (in Kilobytes), as determined by the POST routine.
INT	PARAMETER	FUNCTION
12h	None	Read Memory Size

Diskette Service		The CBIOS Diskette Service performs BIOS-level read, write, format, initialization, and diagnostic support for up to two internal diskette drives. **Note:** If a fixed disk is installed, the CBIOS automatically revectors all 13h Diskette Service requests to INT 40h. This revectoring is transparent to end users. End users should continue to use INT 13h for both fixed disk and diskette service requests.
INT	PARAMETER	FUNCTION
13h	AH = 00h	Reset Diskette System
	AH = 01h	Read Diskette Status
	AH = 02h	Read Diskette Sectors
	AH = 03h	Write Diskette Sectors
	AH = 04h	Verify Diskette Sectors
	AH = 05h	Format Diskette Track
	AH = 06h-07h	Reserved
	AH = 08h	Read Drive Parameters
	AH = 09h-14h	Reserved
	AH = 15h	Read Drive Type
	AH = 16h	Detect Media Change
	AH = 17h	Set Diskette Type
	AH = 18h	Set Media Type for Format
	AH = 19h-FFh	Reserved

continued

Introduction, cont'd

Fixed Disk Service		The CBIOS Fixed Disk Service performs BIOS level read, write, format, initialization, and diagnostic functions for up to two fixed disk drives.
INT	PARAMETER	FUNCTION
13h	AH = 00h	Reset Disk System
	AH = 01h	Read Disk Status
	AH = 02h	Read Disk Sectors
	AH = 03h	Write Disk Sectors
	AH = 04h	Verify Disk Sectors
	AH = 05h	Format Disk Cylinder
	AH = 06h-07h	Reserved
	AH = 08h	Read Drive Parameters
	AH = 09h	Initialize Drive Parameters
	AH = 0Ah	Read Long Disk Sectors
	AH = 0Bh	Write Long Disk Sectors
	AH = 0Ch	Seek to Cylinder
	AH = 0Dh	Alternate Disk Select
	AH = 0Eh	Diagnostics
	AH = 0Fh	Diagnostics
	AH = 10h	Test Drive Ready
	AH = 11h	Recalibrate
	AH = 12h-14h	Reserved
	AH = 15h	Read Disk Type
	AH = 16h-18h	Reserved
	AH = 19h	Park Heads
	AH = 1Ah-FFh	Reserved

continued

Introduction, cont'd

Serial Communication Service		The CBIOS Serial Communication Service performs RS-232-C character I/O for IBM-compatible serial port adapters.
INT	**PARAMETER**	**FUNCTION**
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-FFh	Reserved

System Services		The CBIOS System Services is composed of several miscellaneous system-level subservices, all of which are invoked via the INT 15h mnemonic.
INT	**PARAMETER**	**FUNCTION**
15h	AH = 00h-03h	Reserved
	AH = 04h	Build ABIOS System Parameters Table
	AH = 05h	Build ABIOS Initialization Table
	AH = 06h - 20h	Reserved
	AH = 21h	POWER-ON SELF TEST ERROR LOG:
	AH = 22h-4Eh	Reserved
	AH = 4Fh	Keyboard Intercept
	AH = 50h-7Fh	Reserved
	AH = 80h	Device Open
	AH = 81h	Device Close

Parameter	Subfunction
AL = 00h	Read POST error log
AL = 01h	Write error code to POST error log

continued

Introduction, cont'd

System Services, cont'd		The CBIOS System Services is composed of several miscellaneous system-level subservices, all of which are invoked via the INT 15h mnemonic.
INT	**PARAMETER**	**FUNCTION**
15h	AH = 82h	Program Termination
	AH = 83h	Set Event Wait Interval
	AH = 84h	Joystick Support
	AH = 85h	System Request Key
	AH = 86h	Wait
	AH = 87h	Move Block
	AH = 88h	Read Extended Memory Size
	AH = 89h	Set Protected Mode
	AH = 8Ah-8Fh	Reserved
	AH = 90h	Device Busy Loop
	AH = 91h	Interrupt Complete
	AH = 92h-BFh	Reserved
	AH = C0h	Return System Configuration Parameters
	AH = C1h	Return Extended BIOS Data Area Segment Address
	AH = C2h	POINTING DEVICE INTERFACE

Parameter	Subfunction
AL = 00h	Enable/Disable pointing device
AL = 01h	Reset pointing device
AL = 02h	Set sample rate
AL = 03h	Set resolution
AL = 04h	Read device type
AL = 05h	Pointing device initialization
AL = 06h	Extended commands BH = 00h Return status BH = 01h Set scaling to 1:1 BH = 02h Set scaling to 2:1
AL = 07h	Device driver far call

continued

Introduction, cont'd

System Services, cont'd

The CBIOS System Services is composed of several miscellaneous system-level subservices, all of which are invoked via the INT 15h mnemonic.

INT	PARAMETER	FUNCTION
15h	AH = C3h	ENABLE/DISABLE WATCHDOG TIME-OUT Parameter: AL = 00h — Subfunction: Disable watchdog time-out Parameter: AL = 01h — Subfunction: Enable watchdog time-out
	AH = C4h	PROGRAMMABLE OPTION SELECT Parameter: AL = 00h — Subfunction: Enable cursor for setup Parameter: AL = 01h — Subfunction: Adapter enable
	AH = C5h-FFh	Reserved

Keyboard Service

The CBIOS Keyboard Service interfaces the operating system and application programs with the keyboard.

INT	PARAMETER	FUNCTION
16h	AH = 00h	Read Next Keyboard Character
	AH = 01h	Return Keyboard Status
	AH = 02h	Return Shift Flag Status
	AH = 03h	Set Typematic Rate And Delay
	AH = 05h	Store Key Data
	AH = 06h-0Fh	Reserved
	AH = 010h	Return Extended Key Values
	AH = 011h	Return Extended Key Status
	AH = 012h	Return Extended Shift Flags
	AH = 013h-FFh	Reserved

continued

Introduction, cont'd

Parallel Printer Service		The CBIOS Printer Service provides I/O support for parallel printer ports.
INT	**PARAMETER**	**FUNCTION**
17h	AH = 00h	Print Character
	AH = 01h	Initialize Printer
	AH = 02h	Read Status
	AH = 03h–FFh	Reserved

Time-of-Day Service		The CBIOS Time-of-Day Service contains functions that support and maintain the time-of-day portion of the Motorola MC 146818A (or equivalent) CMOS clock chip.
INT	**PARAMETER**	**FUNCTION**
1Ah	AH = 00h	Read System Timer Counter
	AH = 01h	Set System Timer 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

2

Chapter 2
Hardware Environment

Overview

Introduction

This chapter describes the hardware environment supported by the CBIOS.

Note: The hardware environment described here also applies to the ABIOS portion of PS/2-compatible ROM BIOSs.

Hardware overview only

The hardware contained in IBM PS/2 and compatible systems provides the user with a rich and powerful computing environment. A precise description of each IBM PS/2 hardware component, however, is beyond the scope of this book. The material presented here is intended to give CBIOS users a general survey of the PS/2 hardware and is not intended to substitute for the hardware data sheet or documentation available from individual hardware manufacturers.

continued

CBIOS benefits

By its very nature, the CBIOS is designed to isolate the user from direct manipulation of hardware I/O ports, registers, and control words. As such, the CBIOS provides a high level interface to hardware that is guaranteed to remain the same when hardware components change.

This has a number of benefits:

- If programs use CBIOS calls instead of reads and writes to hardware, they will be more portable across differing hardware environments.
- Using the CBIOS services spares the programmer from having to master many details about hardware that may change.

Additional information

Two appendices are included in this reference with additional hardware information.

- Appendix B — provides a detailed look at the 8042 controller chip.
- Appendix C — describes the Model 25/30 and compatibles hardware environment.

In this chapter

The following hardware information is presented:

- 80286/80386/80386SX Microprocessors
- Math Coprocessors
- Micro Channel
- I/O Devices
- System Time-Related Devices
- CMOS RAM Service
- DMA Controller
- Programmable Option Select (POS)
- Intel 8259A Programmable Interrupt Controllers
- System Control Port Definitions
- Power-On Password
- NMI Mask
- Hardware I/O Port List

80286/80386/80386SX Microprocessors

Introduction

The CBIOS must be implemented in systems that employ the Intel 80286, 80386, or 80386SX microprocessor, an equivalent, or a superset.

The table below lists the CBIOS microprocessor support.

Item	Support
Processor Speeds	6 to 20 MHz (80286); 6 to 33 MHz (80386 and 80386SX) **Note:** Currently verified range of supported speeds.
Wait States	0 or 1 wait states
Address Modes	Real and protected address modes **Note:** See "Address mode support" below.

Address mode support

The ROM BIOS for IBM PS/2 and compatible systems can be addressed in both the real and the protected address modes of the microprocessor.

This statement holds true for both the CBIOS and the ABIOS portions of the ROM BIOS. The difference between the CBIOS and the ABIOS portions of the ROM BIOS lies not in the modes under which they can be addressed but in the kind of operating systems each is designed to support.

- The ABIOS is designed specifically to support multitasking operating systems (e.g. IBM OS/2) that execute exclusively in the microprocessor's real mode, exclusively in protected mode, or bimodally — switching between real and protected modes.
- CBIOS is primarily designed to support single-tasking operating systems (e.g. MS-DOS) that execute exclusively in the microprocessor's real mode.

Math Coprocessors

Introduction

The CBIOS can be implemented in systems that include an Intel 80287 or 80387 math coprocessor (80387SX for 80386SX processors) or an equivalent math coprocessor.

Description

The math coprocessor performs high speed arithmetic, logarithmic, and trigonometric floating point arithmetic calculations, permitting much speedier processing for mathematically intensive processes.

The math coprocessor works in parallel with the microprocessor, allowing both to process instructions separately. See the user manual for the coprocessor for details about the extended data types, registers, and instructions available with this chip.

Math coprocessor hardware interface

The coprocessor operates in an asynchronous mode and can use the same clock generator as the microprocessor. It functions as an I/O device and can be accessed through I/O ports 00F8h, 00FAh, and 00FCh.

The coprocessor BUSY signal tells the microprocessor that the coprocessor is operating. The WAIT signal means that it is executing, and forces the microprocessor to wait until the coprocessor is done.

The coprocessor can operate in either the real or protected address mode. It is in real address mode after a power-on, a reset, or when returning from protected address mode.

The math coprocessor generates an error signal which sets IRQ 13, and the BUSY signal from the coprocessor is then held in the BUSY state. The BUSY signal can be cleared by an I/O Write to I/O port 00F0h or by a write of zero to I/O port address 00F0h with bits 0-7 set to zeros, which also clears IRQ 13.

continued

CBIOS: INT 75h Numeric exception error handling

IRQ 13 (which is generated by a math coprocessor numeric exception error) is vectored to INT 75h hardware interrupt service routine as part of the CBIOS initialization. When IRQ 13 goes high, INT 75h clears the BUSY signal and issues a software interrupt, INT 02h.

Note: User programs which enable numeric exceptions must provide a method for intercepting and processing either INT 75h or INT 02h.

Micro Channel

Introduction

The CBIOS supports such IBM Micro Channel-compatible features as Programmable Option Select (POS) and configurable Direct Memory Access (DMA) arbitration levels.

The following table lists the Micro Channel hardware supported by the CBIOS:

Feature	Description
I/O address width is 8-bit or 16-bits.	Allows either 8-bit or 16-bit I/O transfers.
Central arbitration control point.	Arbitrates among as many as 15 devices.
8 DMA channels.	Serial DMA protocol for 8 channels; either 8-bit or 16-bit DMA transfer.
Level-sensitive interrupts.	Interrupt controller operated in level-triggered mode to allow devices to share interrupt levels.
Programmable option select	Eliminates need for jumpers and configuration switches.
Channel extension connectors	Support future growth and additional channel features.
No support for PC-type adapters.	Supports only adapters designed for the Micro Channel.

continued

Micro Channel connectors

The CBIOS supports three types of channel connectors:

- 16-bit, made up of an 8-bit section, and a 16-bit extension,
- 16-bit with auxiliary video extensions, and
- 32-bit connectors in Intel 80386-based systems.

Programmable option select (POS)

POS data is accumulated in adapter description files (ADFs) which are created by adapter manufacturers for each adapter. The reference diskette supplied with MCA-compatible systems reads .ADF files and stores configuration information in CMOS RAM. The CBIOS power-on self test (POST) reads CMOS RAM and writes the configuration information to the POS registers of the adapters.

The POS I/O addresses are 0094h - 0097h and 0100h - 0107h.

I/O Devices: Introduction

The CBIOS supports the following I/O devices:

- a diskette controller,
- a fixed disk controller,
- a VGA video controller,
- an Intel 8042/8242 or equivalent keyboard controller,
- a serial port, and
- a parallel port.

Each of these capabilities is discussed below.

I/O Devices: Diskette and Disk Hardware

Diskette controller

The CBIOS supports an NEC 765 or equivalent diskette controller. Four types of diskette drives are supported:

- 720K 3.5-inch,
- 1.44 MB 3.5-inch,
- 360K 5.25-inch, and
- 1.2 MB 5.25-inch drives.

Diskette drive configuration

The ROM BIOS supports a maximum of two diskette drives.

Fixed disk controller

The ROM BIOS supports up to two fixed disks. The CBIOS supports an ST506 fixed disk adapter or equivalent. This adapter should be PS/2-compatible. The ST506 adapter is single-tasking and must complete one operation before starting another, even though the next operation may be for the other fixed disk. The hardware interrupt request for fixed disk is 14.

RLL and ESDI fixed disk controllers are also supported by the CBIOS.

I/O Devices: Video Hardware

Introduction

The CBIOS video service supports IBM VGA-compatible hardware, including:

- a VGA-compatible chip or chip set that includes a:
  - CRT Controller
  - Sequencer
  - Graphics Controller, and
  - Attribute Controller
- DAC chip — INMOS G171 or compatible Digital-to-Analog Converter
- Video RAM — 256K of dynamic read/write RAM configured as four 64K maps
- Monochrome or color direct drive analog monitor
- Monochrome or color multiple sync frequency monitors

VGA-compatible chip (or chip set)

The VGA chip (or chip set) provides all CRT control signals. It consists of four components: CRT Controller, Sequencer, Graphics Controller, and Attribute Controller.

The function of each VGA component is summarized in the table below.

Component	Function CRT Controller
CRT Controller	Generates horizontal and vertical CRT sync timings, cursor and underline timings, video buffer addressing, and refresh addressing.
Sequencer	Arbitrates system access to display RAM and fonts. The sequencer allows up to eight fonts with two fonts displayable at any one time.
Graphics Controller	Handles read/write operation on 4 parallel bit planes. Outputs data to Attribute Controller.
Attribute Controller	Converts incoming text mode attribute data or graphics mode pixel data into 8-bit indices into the Digital-to-Analog Converter (DAC) color registers (see below).

Note: For a complete description of VGA-compatible components, I/O ports, and registers, refer to the hardware documentation accompanying your particular VGA-compatible chip or chip set.

continued

DAC

The video DAC contains 256 individual color registers which can be accessed by the BIOS as either four 64-color registers or sixteen 16-color registers.

Each DAC color register contains one 18-bit RGB analog value. Six bits of each register are allocated to each primary color. Thus, the color represented in each DAC color register may be any of 256K possible colors (i.e. 2^{3*6} = 256K).

Video RAM

The ROM BIOS Video Services require 256K of read/write video RAM formatted into four banks (or maps) of 64K.

To maintain compatibility, display memory for each of the historical MDA, CGA, and EGA-compatible modes is mapped exactly as it was in the original display adapter. The display memory organization for the new VGA modes is outlined in the CBIOS Video Services chapter.

Analog monitor support

To display all modes, the Video Service requires either a monochrome or a color direct drive analog monitor with a 31.5 KHz horizontal scan frequency.

The display's vertical gain is adjusted automatically by the VGA-compatible circuitry. Thus, video modes with 350, 400, and 480 horizontal scan lines can be displayed without requiring manual adjustment.

Multiscan monitor support

In addition to 31.5 KHz direct drive analog monitors, the Video Service also supports multiscan rate monitors capable of operating in analog modes (e.g. NEC Multisync monitor). Monitors of this type require an adapter cable that matches the signal assignments and monitor ID circuitry of the DAC external video controller.

I/O Devices: Keyboard Hardware

Introduction

The CBIOS supports an intelligent keyboard subsystem based on the Intel 8042 or equivalent keyboard controller.

The hardware interrupt level associated with the CBIOS Keyboard Service is IRQ 1.

The 8042 controller chip

The Intel 8042 peripheral controller (or compatible) is a single chip microcomputer that can be programmed to allow bidirectional communication between the master microprocessor and up to two auxiliary serial input devices. The 8042 chip, generally, is mounted on the system motherboard. 8042 programs reside as firmware in the 8042 chip itself.

Device support

The kind of devices a given 8042 chip supports are dependent on how the 8042 is programmed.

On IBM PS/2-compatible systems, the 8042 is programmed to allow bi-directional communication between the system and the keyboard, as well as between the system and one other auxiliary serial device, such as a mouse, joystick, or trackball.

Pointing device interface

The pointing device in PS/2-compatible systems is controlled by the 8042. INT 74h handles interrupts from the pointing device. The INT 15h Pointing Device Service controls the device operations.

Reference

For more information on the 8042 Keyboard Controller Chip, refer to Appendix B.

I/O Devices: Parallel Port Hardware

Introduction

The CBIOS Parallel Port Service is associated with hardware interrupt request 7. The CBIOS supports a parallel port that can transfer eight bits of data at standard TTL levels. The parallel port can be called port 1 through 8, must be IBM PS/2-compatible, and must have a bidirectional mode, supporting both input and output. The parallel port also supports level-sensitive interrupts and a readable interrupt pending status.

Parallel port addresses

The following table lists the most common parallel port addresses. Up to 8 parallel ports are supported.

Parallel Port Number	Data Address	Status Address	Control Address
1	03BCh	03BDh	03BEh
2	0378h	0379h	037Ah
3	0278h	0279h	027Ah

Parallel port extended mode

The extended mode of the parallel port can be selected through the port system-based POS registers. The extended mode adds a bidirectional interface.

I/O Devices: Serial Port Hardware

Introduction

The CBIOS supports a National Semiconductor 16550 serial port controller or equivalent logic. The serial ports can be addressed as Serial 1–8. See the equipment list at 40:10h to find out how many serial ports are available. Serial 1 and 3 interrupts are on IRQ 4; Serial 2 and 4 interrupts are on IRQ 3.

Serial port addresses/interrupt levels

The serial port addresses and interrupt levels are listed in the table below.

Serial Port Number	Base Address	Interrupt Level
1	03F8h	4
2	02F8h	3
3	3220h	3
4	3228h	3
5	4220h	3
6	4228h	3
7	5220h	3
8	5228h	3

NS 16550 characteristics

The NS 16550, which is functionally compatible with the NS 16450 and the NS 8250, supports:

- Characters of 5, 6, 7, or 8 bits,
- 1, 1.5, or 2 stop bits, and
- even, odd, or no parity modes.

continued

NS 16550 Serial Communications Controller

The NS 16550 does serial-to-parallel conversions on data received from a peripheral device or a modem, and parallel-to-serial conversion on data received from the system processor. The system processor can read the status of the NS 16550 at any time during its operation. The information furnished includes the type and condition of transfer operations in progress, as well as any error conditions (parity, overrun, framing, or break interrupt) present. The NS 16550 provides complete modem control, and has a user programmable processor-interrupt system.

NS 16550 Serial Controller Registers

The NS 16550 has 12 accessible registers:

- Receiver Buffer Register (Read Only)
- Transmitter Holding Register (Write Only)
- Interrupt Enable Register (Read/Write)
- Interrupt Identification Register (Read Only)
- FIFO (First in/First out) Control Register (Write Only)
- Line Control Register (Read/Write)
- Modem Control Register (Read/Write)
- Line Status Register (Read Only)
- Modem Status Register (Read Only)
- Scratch Register (Read/Write)
- Divisor Latch (LSB) (Read/Write)
- Divisor Latch (MSB) (Read/Write)

Information on the operation of these registers is contained in the National Semiconductor NS 16550 Data Sheet. However, to avoid any incompatibility problems introduced by direct hardware programming, use the access to the serial controller provided through the BIOS services.

Programmable baud rate generator

The serial port controller can operate at speeds of from 110 to 19,200 bps.

System Time-Related Devices: Introduction

System time-related components

The CBIOS supports that the following time-handling chips, or their equivalents:

- Intel 82284 Clock Generator
- Intel 8254A Programmable Interval Timer (PIT)
- Motorola MC146818A Real Time Clock

System Time-Related Devices: 82284 Clock Generator

Description

The Intel 82284 (or compatible) Clock Generator chip interfaces directly to the system microprocessor (CPU). The 82284 chip:

- Provides the CPU with two clock inputs
- Generates the READY input to the CPU
- Synchronizes the system RESET input to the CPU.

System Time-Related Devices: 8254A PIT Chip

Introduction

The 8254A Programmable Interval Timer (PIT) is a counter and timer that provides three channel timers. All channels are driven by a 1.19 MHz oscillator signal. Each "tick" of the PIT generates hardware interrupt request 0.

The BIOS supports the Intel 8254A programmable counter chip or its equivalent. The timer chip need not include a timer/counter 1 but should provide a limited-function timer/counter 3.

Timer channel differences

There are some differences between the three timer channels.

Counters 0 and 2:

- are independent 16-bit counters,
- can be preset, and
- can count in BCD (binary coded decimal) or in binary.

Counter 3:

- is only 8 bits,
- can be preset,
- counts in binary only, and
- can only count downward.

System Timer Modes

The system timer has six modes:

Mode	Name
0	Interrupt on Terminal Count
1	Hardware Retriggerable One-Shot
2	Rate Generator
3	Square Wave
4	Software Triggered Strobe
5	Hardware Retriggerable Strobe

continued

Common timer mode operations

All modes have the following operations in common:

- All control logic resets when control bytes are written to a counter.
- Counters do not stop when they reach zero.
- In Modes 0, 1, 4, and 5 the counter wraps to the highest possible count, and continues to count.
- In Modes 2 and 3, the counter reloads the initial count and continues to count.

Timer Channels

The following table describes the functions of the timer channels. The system timer is treated as a series of I/O ports. Three are count registers, and two are control registers.

Channel	I/O Port	Read/Write Status
0 System Timer	0040h	R/W
2 Tone Generation for Speaker	0042h	R/W
3 Watchdog Timer	0044h	R/W
Control Register 3	0047h	W
Control Register 0, 2	0043h	W

Watchdog Timer

The Watchdog Timer is used to find out if IRQ 0 is not being serviced, which can be used to detect a user program in a tight loop.

The Watchdog Timer is enabled or disabled by the CBIOS INT 15h, System Services, function C3h.

System Time-Related Devices: MC146818A RTC Chip

Introduction

The BIOS supports a Motorola MC146818A Real Time Clock, or equivalent. This chip is assumed to have at least 64 bytes of nonvolatile CMOS RAM available to store configuration data.

RTC CMOS RAM Addresses

The following table lists the CMOS RAM addresses:

I/O Address	Length	Description
0070h	1 Byte	CMOS RAM address, where: Bit 7 = 1 NMI disabled Bits 6-0 = 0 CMOS RAM address
0071h	1 Byte	CMOS RAM data port

- **To write to CMOS RAM:**
  - Inhibit interrupts.
  - Write the CMOS RAM address to which the data is to be written to I/O port 0070h.
  - Write the data to be written to I/O port 0071h.
- **To read from CMOS RAM:**
  - Inhibit interrupts.
  - Write the CMOS RAM address from which the data is to be read to I/O Port 0070h.
  - Read from I/O port 0071h.

CMOS RAM Service

Introduction

Information may be stored in up to two areas of CMOS RAM. The table below describes the CMOS RAM areas available to the ROM BIOS.

Table of CMOS RAM areas

Data Area	I/O Location	Size	Description
CMOS RAM Data Area	0070h and 0071h	64 bytes	These bytes are located on the Motorola MC146818A Real Time Clock CMOS RAM chip (or its equivalent). All implementations of the BIOS make use of this area to store real time clock, POST, and system configuration data.
Extended CMOS RAM Data Area	0074h, 0075h, and 0076h	2K	When implemented on systems that employ more than four adapter slots, the BIOS requires an additional 2K of CMOS RAM. This extended CMOS RAM is primarily used to store POS data.

DMA Controller

DMA functionality

The hardware environment for DMA transfers is described here in order to explain the background against which the DMA CBIOS functions operate. The CBIOS, however, serves as a shield between underlying hardware and requests of the operating system, obviating the need for the caller to directly access the DMA controller.

Direct Memory Access (DMA) allows large amounts of data to be transferred from a physical device to system memory or vice versa without microprocessor involvement. A program may initiate a DMA transfer and have no need to copy each byte or word individually, freeing the processor for more complex tasks. DMA transfers are typically from/to a fixed I/O port address to/from a continually incremented memory address.

DMA functionality in Micro Channel systems is a superset of the functionality of two Intel 8237 DMA Controllers, one addressed at every port, starting with Port 0000h, and one addressed at every other port, starting at port 00C0h. Access to 8237-compatible DMA functions and to additional functions *for all channels* is provided at I/O ports 0018h and 001Ah. Data output to port 0018h selects the channel and function, and data output to or input from 001Ah goes to or from the selected internal register.

Bus sharing

The system microprocessor and any currently-transferring DMA users can share the bus by taking turns directing bus cycles (driving the Micro Channel's address lines and certain control signals). An arbitration process determines which of these possible bus masters are ready to direct a cycle. Competing bus masters (DMA devices) are assigned varying priorities, which are weighed during arbitration. Each bus master gets control of the bus for a number of cycles as determined by the arbitration process.

DMA device

A DMA device (or bus master) is one that enters into arbitration for the channel. If it wins, it receives addresses and control signals from the DMA controller so it can read or write data.

---

continued

DMA Controller functions

A DMA controller is a device which monitors the arbitration process and gives addresses and control signals to the device that has won the bus through arbitration. The controller does not enter into the arbitration itself. PS/2-compatible Micro Channel-based systems provide a DMA controller that supports DMA transfers to/from up to eight devices at once.

DMA hardware registers

The DMA controller maintains several hardware registers for each DMA channel. The key registers are:

- a memory address where the next byte or word will be transferred to or from
- a count of the remaining bytes to transfer ((transfer count)
- a flag (mode) controlling the transfer direction (to memory or to the device), and
- transfer status flags for each channel (status)

continued

DMA hardware registers

The DMA Controller has ten sets of registers, summarized below:

Register	Size (bits)	Number of Registers	How Allocated
Memory Address	24	8	1 per Channel
I/O Address	16	8	1 Per Channel
Transfer Count	16	8	1 Per Channel
Temporary Holding	16	1	All Channels
Mask	4	2	1 for Channels 7-4 1 for Channels 3-0
Arbus	4	2	1 for Channel 4 1 for Channel 0
Mode	8	8	1 per Channel
Status	8	2	1 for Channel 7-4 1 for Channel 3-0
Function	8	1	All Channels
Refresh	9	1	Independent of DMA

Mode Control Field

The Mode Control field provides an opportunity for the caller to use the Autoinitialization and Programmed I/O (PIO) features of the DMA Controller.

- **Autoinitialization**

  Specifies if the DMA Controller will initialize automatically when the transfer reaches the terminal count.

- **Programmed I/O**

  Specifies that the I/O address is to be programmed to the DMA Controller, driving the I/O address on the bus during the DMA cycles.

continued

Transfer Control Bytes

These fields provide an opportunity for the caller to specify the physical address of the memory and I/O fields for CBIOS DMA Service functions 10h, 11h, and 12h.

- **Count Control**

  Specifies if the physical address is decremented or incremented during a transfer.

- **Device Size**

  Specifies whether this is an 8-bit or 16-bit transfer.

DMA controlled by microprocessor

The microprocessor can address the DMA controller and access the DMA registers. The microprocessor can control the DMA modes, transfer addresses, transfer counts, channel masks, and page registers.

Direct DMA Controller access

Reading directly from or writing directly to any DMA Controller port may cause unpredictable results.

DMA data transfer

After a DMA device has won the arbitration bus and the DMA controller is programmed to service the request, a transfer can take place.

DMA transfers can be:

- single transfer,
- multiple transfer (burst mode), or
- read verification.

continued

Burst mode

Burst mode is a method of DMA transfer that allows a device to remain inactive for long periods and then send large amounts of data in a short time. Some peripheral devices, e.g. a fixed disk, transfer their data in bursts that are frequently separated by long periods of inactivity. Burst mode is a way of making these devices more efficient. The device asks to be serviced only when it has data to transfer and then does so in large quantities.

Arbitration levels

Arbitration is a process through which devices compete for control of the Micro Channel on a prioritized basis. Arbitration levels are predefined or programmable levels of priority assigned to devices that compete for possession of the channel.

Central Arbitration Control is a hardware function that allows intelligent peripherals to share and control access to the system. Arbitration is organized in levels of priority, and on each level there can be a number of competing devices.

Arbitration Levels are numbered from 00h to 0Fh. In addition, there are Arbitration Levels -1 and -2, which exist only on the system board. Of the former set, Arbitration Level 00h has the highest priority; Level 0Eh has the lowest. All arbitration level priorities are assigned sequentially; 00h through 0Eh are the highest through lowest priorities. Level 0Fh is reserved.

continued

Arbitration levels, cont'd

The following table summarizes the arbitration levels:

Arbitration Levels	Primary Assignment
FEh	Memory Refresh
FFh	NMI
00h	DMA Channel 0*
01h	DMA Channel 1
02h	DMA Channel 2
03h	DMA Channel 3
04h	DMA Channel 4*
05h	DMA Channel 5
06h	DMA Channel 6
07h	DMA Channel 7
08h	Reserved
09h	Reserved
0Ah	Reserved
0Bh	Reserved
0Ch	Reserved
0Dh	Reserved
0Eh	Reserved
0Fh	System Microprocessor
* These DMA Channels can be programmed to any arbitration level	

continued

DMA channel flags

Additional channel flags control whether

- a transfer is to be repeated forever,
- the memory address is to be decremented or incremented after each cycle,
- the DMA controller maintains I/O address (this is hardwired in most devices that use DMA), or
- a byte or a word is transferred at each cycle.

Physical and Virtual DMA channels

DMA channels can be either physical or virtual. A physical channel can only have one arbitration level, but a virtual channel can be programmed to own any arbitration level not currently assigned to a different channel. Thus, a virtual DMA channel can have many arbitration levels.

Functionally, there is no difference between physical and virtual channels. Priority is determined by the arbitration level only, where level 00h is the highest priority and level 0Eh the lowest.

Virtual DMA Channels and the Arbus register

The arbitration level assignment for channels 0 and 4 can be programmed using the two 4-bit Arbus registers. The Arbus registers permit dynamic reassignment of the arbitration ID value by which the DMA controller responds to DMA requests for bus arbitration. Channels 0 and 4 can then service devices at any arbitration level.

continued

DMA Channels

The DMA channel I/O addresses are defined in the I/O Port Address section later in this chapter. They range from 0000h - 000Fh, 0018h, 001Ah, 0081h - 008Fh, and 00C0h - 00DEh.

DMA extended mode

An extended mode register is available for each programmable DMA channel and is used when a DMA channel requests a DMA data transfer. DMA channels must match the transfer size of the DMA slave, which is programmed by Bit 6 of the extended mode register. DMA read transfers of 16 bits from 8-bit memory or 8-bit memory-mapped I/O devices are not supported.

The following table describes the DMA extended mode register:

Bit Number	Description
7	= 0 Reserved
6	= 0 8-bit transfer = 1 16-bit transfer
5	= 0 Reserved
4	= 0 Reserved
3	= 0 Read memory transfer = 1 Write memory transfer
2	= 0 Verify = 1 Transfer data
1	= 0 Reserved
0	= 0 I/O address 00h = 1 User programs the I/O address

Programmable Option Select (POS)

Introduction

Because it is an integral part of the Micro Channel Architecture (MCA), Programmable Option Select support is assumed.

Adapter slots

The CBIOS supports up to eight adapter cards (or more on systems with a customized BIOS). The CBIOS will work with any system regardless of the number of adapter slots available.

Adapter card identification

Each adapter card must have a unique 2-byte identifier.

Adapter description files

POS data is accumulated in adapter description files (ADFs) that are created by adapter manufacturers for each adapter. The reference diskette supplied with MCA-compatible systems reads the .ADF files and stores the configuration information in CMOS RAM. This information is read by the BIOS power-on self test (POST), which writes it to the POS registers of the adapters and the system board.

Intel 8259A Programmable Interrupt Controllers

Introduction

The CBIOS supports two cascaded Intel 8259A Programmable Interrupt Controller (PIC) chips, or their functional equivalent.

Description

The PIC handles all maskable hardware interrupts. The CBIOS insures compatibility with hardware interrupts expected by existing software. Formerly reserved, IRQ2 is now associated with the 8042 Auxiliary Device (mouse) controller.

Levels of interrupt

There are 16 levels of interrupts. Interrupts are level-sensitive, allowing several devices to share the same hardware interrupt. This reduces the interrupt controller's sensitivity to a transient signal on the Micro Channel bus.

Programmable Interrupt Controller addresses

The following table lists all PIC addresses:

I/O Address	Functional Equivalents	IRQs Routed	Read/Write Status	Description
20h	master 8259A	IRQ 0 - IRQ 7	R/W	Base
21h	master 8259A	IRQ 0 - IRQ 7	W	Mask
A0h	slave 8259A	IRQ 8 - IRQ 15	R/W	Base
A1h	slave 8259A	IRQ 8 - IRQ 15	W	Mask

continued

Table of IRQ assignments

The following table shows the assignment of interrupt requests to commonly interrupting devices, listed from highest to lowest priority. Other devices not listed may use interrupt requests and may share requests with these devices.

Request	Device
IRQ 0	Timer tick
IRQ 1	Keyboard
IRQ 2	IRQ 8-15 are cascaded through IRQ 2
IRQ 8	Real time clock
IRQ 9	Redirect cascade
IRQ 10	Reserved
IRQ 11	Reserved
IRQ 12	Mouse
IRQ 13	80x87 math coprocessor exception
IRQ 14	Fixed disk controllers
IRQ 15	Reserved
IRQ 3	Serial port 2
IRQ 4	Serial port 1
IRQ 5	Parallel port 2
IRQ 6	Diskette
IRQ 7	Parallel port 1

System Control Port Definitions

Introduction

The CBIOS supports certain system control functions at I/O Port addresses 0061h and 0092h.

Port 61h: System Control Port B

The read/write definitions for I/O port address 0061h, System Control Port B, are:

Port 61h: Write Operations	
Bit	**Description**
7	= 1 Reset timer 0 output latch (IRQ 0)
6–4	= 0 Reserved
3	= 0 Enable channel check
2	= 0 Enable parity check
1	= 1 Speaker data enable
0	= 1 Enable Timer 2 gate to speaker

Port 61h: Read Operations	
Bit	**Description**
7	= 1 Parity check
6	= 1 Channel check
5	= 1 Timer 2 output
4	= 1 Toggles with each refresh request
3	= 0 Channel check enabled
2	= 0 Parity check enabled
1	= 1 Speaker data enabled
0	= 1 Timer 2 gate to speaker enabled

continued

Port 92h: System Control Port A

I/O port 92h (R/W), System Control Port A, contains system control flag data. System control Port A is defined below:

Bit	Description
7	1 = Fixed disk activity light A on
6	1 = Fixed disk activity light B on
5	0 = Reserved (must be zero)
4	1 = Watchdog time-out occurred
3	1 = Security lock latch is locked
2	0 = Reserved (must be zero)
1	1 = Alternate gate A20 line active
0	1 = Alternate hot CPU reset

Power-On Password

There are eight bytes in CMOS RAM reserved for a 1-7 byte password and a check character for that password. The microprocessor accesses these bytes during POST. These bytes cannot be accessed by a user program.

These eight bytes are initialized to zeros and are changed to the password during password installation. Passwords can only be changed during POST via the program on the reference diskette which is supplied with MCA-compatible systems.

Models 25 and 30: Power-on password is not supported in IBM PS/2 Models 25 and 30, and compatibles. For a complete description of the Model 25/30 hardware environment turn to Appendix C.

NMI Mask

The NMI (nonmaskable interrupt) input to the microprocessor is masked off at power-on reset. Bit 7 of I/O address 0070h is set to zero to enable the NMI. A power-on reset sets this bit to one.

Table: Hardware I/O Port Definitions

I/O Address	Read/Write Status	Description
0000h	R/W	DMA channel 0, memory address register
0001h	R/W	DMA channel 0, transfer count register
0002h	R/W	DMA channel 1, memory address register
0003h	R/W	DMA channel 1, transfer count register
0004h	R/W	DMA channel 2, memory address register
0005h	R/W	DMA channel 2, transfer count register
0006h	R/W	DMA channel 3, memory address register
0007h	R/W	DMA channel 3, transfer count register
0008h	R	DMA channel 0-3, status register, where: Bit 7 = 1 Channel 3 request Bit 6 = 1 Channel 2 request Bit 5 = 1 Channel 1 request Bit 4 = 1 Channel 0 request Bit 3 = 1 Terminal count on channel 3 Bit 2 = 1 Terminal count on channel 2 Bit 1 = 1 Terminal count on channel 1 Bit 0 = 1 Terminal count on channel 0
000Ah	R/W	DMA channel 0-3, mask register, where: Bits 7-3 = 0 Reserved Bit 2 = 0 Clear mask Bit = 1 Set mask Bit Bits 1-0 = 00b Select channel 0 = 01b Select channel 1 = 10b Select channel 2 = 11b Select channel 3
000Bh	W	DMA channel 0-3, mode register, where: Bits 7-6 = 00b Demand mode = 01b Signal mode = 10b Block mode = 11b Cascade mode Bits 5-4 = 0 Reserved Bits 3-2 = 00b Verify operation = 01b Write operation = 10b Read operation = 11b Reserved Bits 1-0 = 00b Select channel 0 = 01b Select channel 1 = 10b Select channel 2 = 11b Select channel 3
000Ch	W	DMA Clear Byte Pointer
000Dh	W	DMA Master Clear Byte

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
000Eh	W	DMA Channel 0-3 Clear Mask Register
000Fh	W	DMA channel 0-3, write mask register, where: Bits 7-4 = 0 reserved Bit 3 = 0 unmask channel 3 mask bit = 1 set channel 3 mask bit Bit 2 = 0 unmask channel 2 mask bit = 1 set channel 2 mask bit Bit 1 = 0 unmask channel 1 mask bit = 1 set channel 1 mask bit Bit 0 = 0 unmask channel 0 mask bit = 1 set channel 0 mask bit
0018h	W	DMA extended function register, where: Bits 7-4 = Progress command, where: 00h I/O address register 01h Reserved 02h Memory address register write 03h Memory address register read 04h Transfer count register write 05h Transfer count register read 06h Status register read 07h Mode register 08h Arbus register 09h Mask register set single bit 0Ah Mask register reset single bit 0Bh-0Ch Reserved 0Dh Master clear 0Eh-0Fh Reserved Bits 3-0 = 0 Reserved
001Ah	R/W	DMA extended function execute register
0020h	R	PIC, Interrupt request/In-service registers programmed by Operation Command Word 3 (OCW3): Interrupt request register, where: Bits 7-0= 0 No active request for the corresponding interrupt line = 1 Active request for the corresponding interrupt line Interrupt in-service register, where: Bits 7-0= 0 The corresponding interrupt line is not currently being serviced = 1 The corresponding interrupt line is currently being serviced

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0020h	W	PIC, Initialization Command Word 1 (ICW1) (Bit 4 is one), where: Bits 7-5 = 000 Only used in 80/85 mode Bit 4 = 1 Reserved Bit 3 = 0 Edge triggered mode = 1 Level triggered mode Bit 2 = 0 Successive interrupt vectors are separated by eight bytes = 1 Successive interrupt vectors are separated by four bytes Bit 1 = 0 Cascade mode = 1 Single mode — no ICW3 needed Bit 0 = 0 No ICW4 needed = 1 ICW4 needed
0021h	W	PIC, ICW2, ICW3, or ICW4 in sequential order after ICW1 written to Port 0020h ICW2, where: Bits 7-3 = Address lines A0-A3 of base vector address for interrupt controller Bits 2-0 = 000 Reserved ICW3, where: Bits 7-0 = 0 Slave controller not attached to corresponding interrupt pin = 0 Slave controller attached to corresponding interrupt pin ICW4, where: Bits 7-5 = 000 Reserved Bit 4 = 0 No special fully-nested mode = 1 Special fully-nested mode Bits 3-2 = 00 Non-buffered mode = 01 Non-buffered mode = 10 Buffered mode/slave = 11 Buffered mode/master Bit 1 = 0 Normal EOI = 1 Auto EOI Bit 0 = 0 80/85 mode = 1 8086/8088 mode
0021h	R/W	PIC, Interrupt mask register (OCW1), where: Bit 7 = 0 Enable parallel printer interrupt Bit 6 = 0 Enable diskette interrupt Bit 5 = 0 Enable fixed disk interrupt Bit 4 = 0 Enable serial port 1 interrupt Bit 3 = 0 Enable serial port 2 interrupt Bit 2 = 0 Enable video interrupt Bit 1 = 0 Enable keyboard interrupt Bit 0 = 0 Enable timer interrupt

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0021h	W	PIC, OCW2 (Bit 4 is zero, Bit 3 is zero), where: Bits 7-5 = 000 Rotate in automatic EOI mode (clear) = 001 Non-specific EOI = 010 No operation = 011 Specific EOI = 100 Rotate in automatic EOI mode (set) = 101 Rotate on non-specific EOI command = 110 Set priority command = 111 Rotate on specific EOI command Bit 4 = 0 Reserved Bit 3 = 0 Reserved Bits 2-0 = Interrupt request to which the command applies
0020h	W	PIC, OCW3 (Bit 4 is zero, Bit 3 is one), where: Bit 7 = 0 Reserved Bits 6-5 = 00 No operation = 01 No operation = 10 Reset special mask = 11 Set special mask Bit 4 = 0 Reserved Bit 3 = 1 Reserved Bit 2 = 0 No poll command = 1 Poll command Bits 1-0 = 00 No operation = 01 No operation = 10 Read interrupt request register on next read at Port 0020h = 11 Read interrupt in-service register on next read at Port 0020h
0040h	R/W	Programmable Interrupt Timer - Read/write counter 0
0042h	R/W	Programmable Interrupt Timer - Read/write counter 2

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0043h	W	PIT, control word register for counters 0 and 2, where: Bits 7-6 = 00b Select counter 0 = 01b Reserved = 10b Select counter 2 Bits 5-4 = 00b Counter latch command = 01b Read/write counter bits 0-7 only = 10b Read/write counter bits 8-15 only = 11b Read/write counter bits 0-7 first, then bits 8-15 Bits 3-0 = 000b Mode 0 select = 001b Mode 1 select = X10b Mode 2 select = X11b Mode 3 select = 100b Mode 4 select = 101b Mode 5 select Bit 0 = 0 Binary counter 16 bits = 1 Binary coded decimal counter
0044h	W	PIT, read/write counter 3
0047h	W	PIT, control word register for counter 3, where: Bits 7-6 = 00b Select counter 3 = 01b Reserved = 10b Reserved = 11b Reserved Bits 5-4 = 00b Counter latch command select counter 0 = 01b Read/write counter bits 0-7 only = 10b Reserved = 11b Reserved
0060h	R/W	Keyboard/auxiliary data port
0061h	R	System control port B, where: Bit 7 = 1 Parity check Bit 6 = 1 Channel check Bit 5 = 1 Timer 2 output Bit 4 = 1 Toggle with each refresh request Bit 3 = 0 Channel check enabled Bit 2 = 0 Parity check enabled Bit 1 = 1 Speaker data enabled Bit 0 = 1 Timer 2 gate to speaker enabled

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0061h	W	System control port B, where: Bit 7 = 1 Reset timer 0 output latch (IRQ 0) Bits 6-4 = Reserved Bit 3 = 0 Enable channel check Bit 2 = 0 Enable parity check Bit 1 = 1 Speaker data enable Bit 0 = 1 Enable timer 2 gate to speaker
0064h	W	8042 Commands
0064h	R	8042 Status, where: Bit 7 = 1 Parity error Bit 6 = 1 General time out Bit 5 = 1 Auxiliary output buffer full Bit 4 = 1 Inhibit switch Bit 3 = 1 Command/data Bit 2 = System flag Bit 1 = 1 Input buffer full Bit 0 = 1 Output buffer full
0070h	W	CMOS RAM address register port, where: Bit 7 = 1 NMI disable = 0 NMI enabled Bits 6-0 = 0 CMOS RAM address
0071h	R/W	CMOS RAM data register port
0074h	W	Extended CMOS RAM address register port, least significant byte
0075h	W	Extended CMOS RAM address register port, most significant byte
0076h	R/W	Extended CMOS RAM data register port
0080h	R	DMA access
0081h	R/W	DMA channel 2, page table address register
0082h	R/W	DMA channel 3, page table address register
0083h	R/W	DMA channel 1, page table address register
0087h	R/W	DMA channel 0, page table address register
0089h	R/W	DMA channel 6, page table address register
008Ah	R/W	DMA channel 7, page table address register
008Bh	R/W	DMA channel 5, page table address register
008Fh	R/W	DMA channel 4, page table address register

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0090h	R	DMA arbitration register, where: Bit 7 = 1 System microprocessor cycles enabled Bit 6 = 1 Arbitration mask by NMI Bit 5 = 1 Bus timeout Bit 4 = 0 Reserved Bits 3-0 = Arbitration level
0090h	W	DMA arbitration register, where: Bit 7 = 1 Enable system microprocessor cycle Bit 6 = 1 Arbitration mask Bit 5 = 1 Enable extended arbitration Bits 4-0 = 0 Reserved
0091h	W	DMA card selected feedback register, where: Bits 7-1 = Reserved Bit 0 = 1 Card selected feedback signal active on previous cycle or system board I/O functions accessed by an I/O cycle
0092h	R/W	System control port A, where: Bit 7 = 1 Fixed disk activity light bit A on Bit 6 = 1 Fixed disk activity light bit B on Bit 5 = X Reserved Bit 4 = 1 Watchdog timeout occurred Bit 3 = 1 Security lock latch locked Bit 2 = X Reserved Bit 1 = 1 Alternate gate A20 active Bit 0 = 1 Alternate hot reset
0094h	R/W	System board setup enable register, where: Bit 7 = 0 Enable system board setup = 1 Disable system board setup Bit 6 = 1 Reserved Bit 5 = 0 Enable VGA setup = 1 Disable VGA setup Bits 4-0 = 1 Reserved
0096h	R/W	POS channel position select register, where: Bit 7 = 1 Channel 1 reset Bits 6-4 = Reserved (written as 0, read as 1) Bit 3 = 1 Channel select Bits 2-0 = 0 Channel number
00A0h	R/W	Programmable Interrupt Controller 2

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
00A1h	R/W	Programmable Interrupt Controller 2 mask, where: Bit 7 = 0 Reserved Bit 6 = 0 Enable fixed disk interrupt Bit 5 = 0 Enable 80387 exception interrupt Bit 4 = 0 Enable mouse interrupt Bit 3 = 0 Reserved Bit 2 = 0 Reserved Bit 1 = 0 Enable redirect cascade Bit 0 = 0 Enable real time clock interrupt
00C0h	R/W	DMA channel 4, memory address register
00C2h	R/W	DMA channel 4, transfer count register
00C4h	R/W	DMA channel 5, memory address register
00C6h	R/W	DMA channel 5, transfer count register
00C8h	R/W	DMA channel 6, memory address register
00CAh	R/W	DMA channel 6, transfer count register
00CCh	R/W	DMA channel 7, memory address register
00CEh	R/W	DMA channel 7, transfer address register
00D0h	R	DMA channel 4-7, status register, where: Bit 7 = 1 Channel 7 request Bit 6 = 1 Channel 6 request Bit 5 = 1 Channel 5 request Bit 4 = 1 Channel 4 request Bit 3 = 1 Terminal count on channel 7 Bit 2 = 1 Terminal count on channel 6 Bit 1 = 1 Terminal count on channel 5 Bit 0 = 1 Terminal count on channel 4
00D4h	R/W	DMA channel 4-7, mask register, where: Bits 7-3 = 0 Reserved Bit 2 = 0 Clear mask bit = 1 Set mask bit Bits 1-0 = 00b Select channel 4 = 01b Select channel 5 = 10b Select channel 6 = 11b Select channel 7

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
00D6h	R/W	DMA channel 4-7, Mode Register, where: Bits 7-6 = 00b Demand mode = 01b Single mode = 10b Block mode = 11b Cascade mode Bit 5 = 0 Reserved Bit 4 = 0 Reserved Bits 3-2 = 00b Verify operation = 01b Write operation = 10b Read operation = 11b Reserved Bits 1-0 = 00b Select channel 4 = 01b Select channel 5 = 10b Select channel 6 = 11b Select channel 7
00D8h	W	DMA Clear Byte Pointer
00DAh	W	DMA Master Clear
00DCh	W	DMA Channel 4-7, Clear Mask Register
00DEh	W	DMA Channel 4-7, Write Mask Register, where: Bits 7-4 = 0 Reserved Bit 3 = 0 Unmask channel 7 mask bit = 1 Set channel 7 mask bit Bit 2 = 0 Unmask channel 6 mask bit = 1 Set channel 6 mask bit Bit 1 = 0 Unmask channel 5 mask bit = 1 Set channel 5 mask bit Bit 0 = 0 Unmask channel 4 mask bit = 1 Set channel 4 mask bit
F0-FFh	R/W	Math Coprocessor
0100h	R	POS Adapter Identification (least significant byte)
0101h	R	POS Adapter Identification (most significant byte)

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0102h	R/W	POS register 2 for the system board setup, where: Bit 7 = 0 Enable parallel port extended mode = 1 Disable parallel port extended mode Bits 6, 5 = Parallel port select, where: = 00 Parallel 1 3BC-3BE = 01 Parallel 2 278-37A = 10 Parallel 3 278-27A = 11 Reserved Bit 4 = 1 Enable parallel port, if bit 0 = 1 = 0 Disable parallel port Bit 3 = 1 System board serial is serial 1 = 0 System board serial is serial 2 Bit 2 = 1 Enable serial port, if bit 1 = 1 = 0 Disable serial port Bit 1 = 1 Enable diskette, if bit 0 = 1 = 0 Disable diskette Bit 0 = 1 Allows bits 4, 2, and 1 to enable/disable devices = 0 Disables system board devices For an adapter card, the following bit is defined: Bit 0 = 1 Enable adapter card
0103h	R/W	POS register 3, where: Bits 7-2 = Reserved Bit 1 = 0 Password disable Bit 0 = Reserved
0104h	R/W	POS register 4
0105h	R/W	POS register 5 for an adapter card, where: Bit 7 = 0 Channel check condition occurred = 1 Channel reset Bit 6 0 Channel check exception status available
0106h	R/W	POS subaddress extension (least significant byte)
0107h	R/W	POS subaddress extension (most significant byte)
1F0-1F8h	R/W	Fixed disk
200-20Fh	R/W	Game port
0278h	R/W	Parallel 3, Data Port

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0279h	R/W	Parallel 3, Status Port, where: Bit 7 = 0 Busy Bit 6 = 0 Acknowledge Bit 5 = 1 Out of paper Bit 4 = 1 Printer is selected Bit 3 = 0 Error Bit 2 = 0 IRQ has occurred Bits 1-0 = Reserved
027Ah	R/W	Parallel 3, Control Port, where: Bits 7-6 = Reserved Bit 5 = 0 Direction is write to port = 1 Direction is read from port Bit 4 = 1 Enable IRQ Bit 3 = 1 Select printer Bit 2 = 0 Initialize printer Bit 1 = 1 Automatic line feed Bit 0 = 1 Strobe
02F8h	W	Serial 2, transmitter holding register, where: Bits 7-0 Data bits 7-0, respectively, when Divisor Latch Access Bit = 0
02F8h	R	Serial 2, receiver buffer register, where: Bits 7-0 = Data bits 7-0, respectively, when DLAB = 0
02F8h	R/W	Serial 2, divisor latch, low byte, where: Bits 7-0 = Bits 7-0 of divisor, when DLAB = 1
02F9h	R/W	Serial 2, divisor latch, high byte Bits 7-0 = Bits 15-8 of divisor, when DLAB = 1
02F9h	R/W	Serial 2, interrupt enable register, where: Bits 7-4 = 0 Reserved Bit 3 = 1 Modem status interrupt enable Bit 2 = 1 Receiver line status interrupt enable Bit 1 = 1 Transmitter holding register empty interrupt enable Bit 0 = 1 Received data available interrupt enable when DLAB = 0
02FAh	R	Serial 2, interrupt identification register, where: Bits 7-3 = 0 Reserved Bits 2-1 = 00b Modem status interrupt = 01b Transmitter holding register empty interrupt = 10b Received data available register interrupt = 11b Receiver line status interrupt Bit 0 = 0 Interrupt pending

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
02FAh	W	Serial 2, FIFO control register, where: Bits 7-6 = Receiver FIFO register trigger 00b = 1 byte 01b = 4 bytes 10b = 8 bytes 11b = 14 bytes Bits 5-3 = 0 Reserved Bit 2 = Transmitter FIFO register reset 1 = transmit FIFO register cleared, counter cleared, bit is self-clearing Bit 1 = Receiver FIFO register reset 1 = receiver FIFO register cleared, counter cleared, bit is self-clearing Bit 0 = FIFO enable 1 = receiver and transmitter FIFOs enabled, must be 1 to program FIFO registers 0 = clears receive and transmit FIFO registers, enters character mode
02FBh	R/W	Serial 2, line control register, where: Bit 7 = 1 Divisor latch access = 0 Receiver buffer, transmitter holding, or interrupt enable registers access Bit 6 = 1 Set break enable Bit 5 = Stick parity Bit 4 = Even parity select Bit 3 = Parity enable Bit 2 = Number of stop bits Bits 1-0 = 00b Word length is 5 bits = 01b Word length is 6 bits = 10b Word length is 7 bits = 11b Word length is 8 bits
02FCh	R/W	Serial 2, modem control register, where: Bits 7-5 = 0 Reserved Bit 4 = 1 Loopback mode Bit 3 = 1 Enable OUT2 interrupt Bit 2 = 1 Force OUT1 active Bit 1 = 1 Force request-to-send active Bit 0 = 1 Force data-terminal-ready active

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
02FDh	R	Serial 2, line status register, where: Bit 7 = 0 Reserved Bit 6 = 1 Transmitter shift and holding registers empty Bit 5 = 1 Transmitter holding register empty Bit 4 = 1 Break interrupt Bit 3 = 1 Framing error Bit 2 = 1 Parity error Bit 1 = 1 Overrun error Bit 0 = 1 Data ready
02FEh	R	Serial 2, modem status register, where: Bit 7 = 1 Data carrier detect Bit 6 = 1 Ring indicator Bit 5 = 1 Data set ready Bit 4 = 1 Clear to send Bit 3 = 1 Delta to carrier detect Bit 2 = 1 Trailing edge ring indicator Bit 1 = 1 Delta data set ready Bit 0 = 1 Delta clear to send
02FFh	R/W	Serial 2, scratch register
0320h	R/W	Fixed Disk Adapter Register (8 or 16 bit)
0322h	W	Fixed Disk Adapter Control Register, where: Bit 7 = 1 Reset Bit 6 = 1 Reserved (except during reset) Bit 5 = 1 16-bit mode (must match bit 2) = 0 8-bit mode Bit 4-3 = 0 Reserved Bit 2 = 1 16-bit mode (must match bit 5) = 0 8-bit mode Bit 1 = 1 Enable interrupt through Programmable Interrupt Controller (hardware interrupt) = 0 enable interrupt through Interrupt Status Register (port 324h) Bit 0 = 1 DMA mode = 0 PIO mode
0322h	R	Fixed Disk Adapter Status Register, where: Bit 7-6 = 0 Reserved Bit 5 = 1 16-bit mode = 0 8-bit mode Bit 4 = 1 Data transfer requested by adapter Bit 3 = 1 Direction is adapter to system = 0 Direction is system to adapter Bit 2 = 1 Busy Bit 1 = 1 Interrupt request (notification) Bit 0 = 1 Transfer in progress

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0324h	W	Fixed Disk Adapter Attention Register, where: Bit 7 = 1 Command control block Bit 6 = 1 Command specify block Bit 5 = 1 Sense summary block Bit 4 = 1 Data transfer requested by system Bit 3 = 0 Reserved Bit 2 = 0 Drive 0 select = 1 drive 1 select Bit 1 = 0 Reserved Bit 0 = 1 Abort current command
0324h	R	Fixed Disk Adapter Interrupt Status Register where: Bit 7 = 1 Termination error, bits 0-6 indicate what the error is Bit 6 = 1 Invalid command Bit 5 = 1 Command reject Bit 4-3 = 0 Reserved Bit 2 = 0 Drive 0 selected = 1 Drive 1 selected Bit 1 = 1 Error recovery procedure invoked Bit 0 = 1 Equipment check
0378h	R/W	Parallel 2, data port
0379h	R/W	Parallel 2, status port, where: Bit 7 = 0 Busy Bit 6 = 0 Acknowledge Bit 5 = 1 Out of paper Bit 4 = 1 Printer is selected Bit 3 = 0 Error Bit 2 = 0 IRQ has occurred Bits 1, 0 = Reserved
037Ah	R/W	Parallel 2, control port, where: Bit 7-6 = Reserved Bit 5 = 0 Direction is write to port = 1 Direction is read from port Bit 4 = 1 Enable IRQ Bit 3 = 1 Select printer Bit 2 = 0 Initialize printer Bit 1 = 1 Automatic line feed Bit 0 = 1 Strobe
03B4h	R/W	VGA CRT controller index register (mono)
03B5h	R/W	Other VGA CRT controller registers (mono)
03BAh	R	VGA input status register 1 (mono)
03BAh	W	VGA feature control register (mono)

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
03BCh	R/W	Parallel 1, data port
03BDh	R/W	Parallel 1, status port, where: Bit 7 = 0 Busy Bit 6 = 0 Acknowledge Bit 5 = 1 Out of paper Bit 4 = 1 Printer is selected Bit 3 = 0 Error Bit 2 = 0 IRQ has occurred Bits 1, 0 = Reserved
03BEh	R/W	Parallel 1, control port, where: Bit 6 = Reserved Bit 5 = 0 Direction is write to port = 1 Direction is read from port Bit 4 = 1 Enable IRQ Bit 3 = 1 Select printer Bit 2 = 0 Initialize printer Bit 1 = 1 Automatic line feed Bit 0 = 1 Strobe
03C0h	R/W	VGA attribute address register
03C0h	W	Other VGA attribute registers
03C1h	R	Other VGA attribute registers
03C2h	W	VGA miscellaneous output register
03C2h	R	VGA input status register 0
03C3h	R/W	VGA video subsystem enable
03C4h	R/W	VGA sequencer address register
03C5h	R/W	Other VGA sequencer registers
03C6h	R/W	Video DAC PEL mask
03C7h	W	Video DAC PEL address, read mode
03C7h	R	Video DAC state register
03C8h	R/W	Video DAC PEL address, write mode
03C9h	R/W	Video DAC PEL data register
03CAh	R	VGA feature control register
03CCh	R	VGA miscellaneous output register
03CEh	R/W	VGA graphics registers, address register
03CFh	R/W	VGA and other graphics registers

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
03D4h	R/W	VGA CRT controller index register (color)
03D5h	R/W	Other VGA CRT controller registers (color)
03DAh	R	VGA input status register 1 (color)
03DAh	W	VGA feature control register (color)
03F0h	R	Diskette controller status register A, where: Bit 7 = 1 Interrupt pending Bit 6 = 0 Second drive installed Bit 5 = 1 Step Bit 4 = 1 Track 0 Bit 3 = 1 Head 1 select Bit 2 = 0 Index Bit 1 = 0 Write protect Bit 0 = 0 Data received by controller
03F1h	R	Diskette controller status register B, where: Bit 7-6 = 0 Reserved Bit 5 = Drive select Bit 4 = Write data Bit 3 = Read data Bit 2 = Write enable Bit 1 = 1 Motor enable 1 Bit 0 = 1 Motor enable 0
03F2h	W	Diskette controller digital output register, where: Bit 7-6 = 0 Reserved Bit 5 = 1 Motor enable 1 Bit 4 = 1 Motor enable 0 Bit 3 = 0 Allow interrupts Bit 2 = 0 Controller reset Bit 1 = 0 Reserved Bit 0 = 0 Drive select 0 = 1 Drive select 1
03F4h	R	Diskette controller status register, where: Bit 7 = 1 Data register is ready Bit 6 = 1 Transfer is from controller to system = 0 Transfer is from system to controller Bit 5 = 1 Non-DMA mode Bit 4 = 1 Diskette controller busy Bit 3-2 = Reserved Bit 1 = 1 Drive 1 busy Bit 0 = 0 Drive 0 busy
03F5h	R/W	Diskette controller data registers

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
03F7h	R	Diskette controller digital input register where: Bit 7 = Diskette change Bits 6-1 = Reserved Bit 0 = 0 High density select
03F7h	W	Diskette controller configuration control register, where: Bits 7-2 = Reserved Bits 1-0 = 00b 500 kbs mode = 01b Reserved = 10b 250 kbs mode = 11b Reserved
03F8h	W	Serial 1, transmitter holding register, where: Bits 7-0 = Data bits 7-0, respectively, when Divisor Latch Access Bit (DLAB) = 0
03F8h	R	Serial 1, receiver buffer register, where: Bits 7-0 = Data bits 7-0, respectively, when DLAB = 0
03F8h	R/W	Serial 1, divisor latch, low byte, where: Bits 7-0 = Bits 7-0 of divisor, when DLAB = 1
03F9h	R/W	Serial 1, divisor latch, high byte, where: Bits 7-0 = Bits 15-8 of divisor, when DLAB = 1
03F9h	R/W	Serial 1, interrupt enable register, where: Bits 7-4 = 0 Reserved Bit 3 = 1 Modem status interrupt enable Bit 2 = 1 Receiver line status interrupt enable Bit 1 = 1 Transmitter holding register empty interrupt enable Bit 0 = 1 Received data available interrupt enable when DLAB = 0
03FAh	R	Serial 1 Interrupt ID Register, where: Bits 7-3 = 0 Reserved Bits 2-1 = 00b Modem status interrupt 01b Transmitter holding register empty interrupt 10b Received data available register interrupt 11b Receiver line status interrupt Bit 0 = 0 Interrupt pending

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
03FAh	W	Serial 1, FIFO Control Register, where: Bits 7-6 = Receiver FIFO register trigger 00b = 1 byte 01b = 4 bytes 10b = 8 bytes 11b = 14 bytes Bits 5-3 = 0 Reserved Bit 2 = Transmitter FIFO register reset 1 = transmit FIFO register cleared, counter cleared, bit is self-clearing Bit 1 = Receiver FIFO register reset 1 = receiver FIFO register cleared, counter cleared, bit is self-clearing Bit 0 = FIFO enable 1 = receiver and transmitter FIFOs enabled, must be 1 to program FIFO registers 0 = clears receive and transmit FIFO registers, enters character mode
03FBh	R/W	Serial 1, Line Control Register, where: Bit 7 = 0 Receiver Buffer, Transmitter Holding, or Interrupt Enable Registers Access = 1 Divisor Latch Access Bit 6 = 1 Set Break Enabled Bit 5 = Stick Parity Bit 4 = Even Parity Select Bit 3 = Parity Enable Bit 2 = 0 1 Stop Bit = 1 0 Stop Bits Bits 1-0 = 00b 5 Bit Word Length 01b 6 Bit Word Length 10b 7 Bit Word Length 11b 8 Bit Word Length
03FCh	R/W	Serial 1, Modem Control Register, where: Bits 7-5 = 0 Reserved Bit 4 = 1 Loopback mode Bit 3 = 1 Enable OUT2 interrupt Bit 2 = 1 Force OUT1 active Bit 1 = 1 Force request to send active Bit 0 = 1 Force data terminal ready active

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
03FDh	R/W	Serial 1, Line Status Register, where: Bit 7 = 0 Reserved Bit 6 = 1 Transmitter shift and holding registers empty Bit 5 = 1 Transmitter holding register empty Bit 4 = 1 Break interrupt Bit 3 = 1 Framing error Bit 2 = 1 Trailing edge ring indicator Bit 1 = 1 Overrun error Bit 0 = 1 Data ready
03FEh	R	Serial 1 Modem Status Register, where: Bit 7 = 1 Data carrier detect Bit 6 = 1 Ring indicator Bit 5 = 1 Data set ready Bit 4 = 1 Clear to send Bit 3 = 1 Delta data carrier detect Bit 2 = 1 Trailing edge ring indicator Bit 1 = 1 Delta data set ready Bit 0 = 1 Delta clear to send
03FFh	R	Serial 1, Scratch register
0680h	W	Manufacturing checkpoint port
3220-3227h	See 03F8h-03FFh	Serial Port 3 (see description for addresses 03F8h-03FFh for details).
3228-322Fh	See 03F8h-03FFh	Serial Port 4 (see description for addresses 03F8h-03FFh for details).
4220-4227h	See 03F8h-03FFh	Serial Port 5 (see description for addresses 03F8h-03FFh for details).
4228-422Fh	See 03F8h-03FFh	Serial Port 6 (see description for addresses 03F8h-03FFh for details).
5220-5228h	See 03F8h-03FFh	Serial Port 7 (see description for addresses 03F8h-03FFh for details).
5228-522Fh	See 03F8h-03FFh	Serial Port 8 (see description for addresses 03F8h-03FFh for details).

3

Chapter 3
System RAM Data

Overview

Introduction

The CBIOS makes use of information stored in system RAM. The table below characterizes each of these areas.

Data Area	Location	Description
Interrupt Vector Table	00h to 3FFh	Interrupt vectors stored as offset/segment format.
CBIOS Data Area	400h to 4FFh	CBIOS work area. Contains data definitions related to CBIOS fixed disk, diskette, keyboard, video, and other BIOS features and functions.
Extended CBIOS Data Area	Varies: See INT 15h, AH=C1h	Extended CBIOS work area. Contains pointing device (mouse) data definitions and fixed disk parameters.

In this chapter

This chapter presents the following information:

- Interrupt Vectors
- CBIOS Data Area
- CBIOS Extended Data Area

Interrupt vector table

The table below identifies each interrupt by function and type. Where applicable, it lists the interrupt vector address initialized by the CBIOS at POST.

Note: Further discussion of interrupts is limited to those interrupts directly related to the CBIOS. The column labeled "Ref. Chapter" tells where to turn for more information on a particular interrupt.

INT	Function	Type	Vector	Ref. Chap.
00h	Divide by Zero	Logical		
01h	Single Step	Logical		
02h	Nonmaskable Interrupt (NMI)	Logical	FE2C3h	7
03h	Breakpoint	Logical		
04h	Overflow	Logical		
05h	Print Screen	Software	FFF54h	16
06h	Invalid Op Code	Hardware		
07h	Math Coprocessor Not Present	Hardware		
08h	System Timer	Hardware	FFEA5h	15
09h	Keyboard	Hardware	FE987h	8
0Ah	Invalid Task Segment State	Hardware		
0Bh	Serial Communications (COM2)	Hardware		
0Ch	Serial Communications (COM1)	Hardware		
0Dh	Parallel Printer (LPT2)	Hardware		
0Eh	Diskette	Hardware	FEF57h	8
0Fh	Parallel Printer (LPT1)	Hardware		
10h	Video	Software	FF065h	9
11h	Equipment List	Software	FF84Dh	16
12h	Memory Size	Software	FF841h	16
13h	Fixed Disk/Diskette	Software	FE3FEh	11
14h	Serial Communication	Software	FE739h	12
15h	System Services	Software	FF859h	13
16h	Keyboard	Software	FE84Eh	8
17h	Parallel Printer	Software	FEFD2h	14
18h	Load ROM BASIC	Software	F1C90h	16

continued

Interrupt vector table, cont'd

INT	Function	Type	Vector	Ref. Chap.
19h	Bootstrap Loader	Software	FE6F2h	16
1Ah	System Timer & Real-Time Clock	Software	FFE6Eh	15
1Bh	Keyboard Break	Software	FFF53h	8
1Ch	User Timer Tick	User	FFF53h	15
1Dh	Video Parameters Table	BIOS Table	FF0A4h	9
1Eh	Diskette Parameters Table	BIOS Table	FEFC7h	10
1Fh	Video Graphics Characters	User	F7F67h	9
20h-3Fh	Reserved for DOS			
40h	Diskette BIOS Revector	Software	FEC59h	10
41h	Fixed Disk Parameter Table	BIOS Table	FE401h	11
42h	EGA Default Video Driver	BIOS Table		
43h	Video Graphics Characters	User		9
44h-45h	Reserved			
46h	Fixed Disk Parameter Table	BIOS Table	FE401h	11
47h-49h	Reserved			
4Ah	User Alarm	User		15
4Bh-5Fh	Reserved			
60h-66h	Reserved for User Program Interrupts	User		1
67h	LIM EMS Driver			
68h-6Fh	Reserved			
70h	Real Time Clock	Hardware	F5124h	15
71h	Redirect Cascade	Hardware	F5266h	
72h-73h	Reserved			
74h	Pointing Device (Mouse)	Hardware	F4910h	13
75h	80287 Exception	User	F5257h	7
76h	Fixed Disk	Hardware	FE2C3h	11
77h-7Fh	Reserved			
80h-F0h	Reserved for BASIC	BASIC		16
F1h-FFh	Reserved for User Program Interrupts	User		

CBIOS Data Area

Introduction

The CBIOS uses system RAM for storage of data definitions specific to the CBIOS.

System RAM summary

The table below lists the system RAM data definitions in offset order. Offsets are from segment address 40:0h.

Location	BIOS Service	Description
00h	INT 14h	I/O address of up to 4 asynchronous communications adapters
08h	INT 17h	I/O address of up to 3 printer adapters
0Eh		Extended CBIOS data area segment
10h	INT 11h	Number of devices installed, where: Bits 15-14 = Number of printer adapters Bits 13-12 = Reserved Bits 11-9 = Number of asynchronous adapters (RS232) Bits 8 = Reserved Bits 7-6 = Number of disk drives, where: 00b = 1 Disk drive 01b = 2 Disk drives Bits 5-4 = Initial video mode, where: 00b = EGA/VGA or PGA 01b = 40x25 color 10b = 80x25 color 11b = 80x25 black and white Bit 3 = Reserved Bit 2 = Pointing device Bit 1 = 1 If math coprocessor Bit 0 = Diskette available for boot
12h	POST	Reserved
13h	INT 12h	Installed memory in Kilobytes minus 1K used by the extended BIOS data area
15h	POST	Reserved

continued

System RAM summary, cont'd

Location	BIOS Service	Description
17h	INT 16h	Keyboard shift flags, where: Bit 7 = 1 Insert active Bit 6 = 1 Caps Lock active Bit 5 = 1 Num Lock active Bit 4 = 1 Scroll Lock active Bit 3 = 1 Alt pressed Bit 2 = 1 Ctrl pressed Bit 1 = 1 Left Shift pressed Bit 0 = 1 Right Shift pressed
18h	INT 16h	More keyboard shift flags, where: Bit 7 = 1 Insert pressed Bit 6 = 1 Caps Lock pressed Bit 5 = 1 Num Lock pressed Bit 4 = 1 Scroll Lock pressed Bit 3 = 1 Ctrl-Num Lock state active Bit 2 = 1 Sys Req pressed Bit 1 = 1 Left Alt pressed Bit 0 = 1 Left Ctrl pressed
19h	INT 16h	Work area for Alt key and numeric keypad input
1Ah	INT 16h	Pointer to next character in keyboard buffer
1Ch	INT 16h	Pointer to first available spot in keyboard buffer
1Eh	INT 16h	Keyboard buffer of 16 word entries
3Eh	INT 13h	Diskette drive recalibrate status, where: Bit 7 = 1 Diskette hardware interrupt has occurred Bits 6-4 = Not used Bits 3,2 = Reserved Bit 1 = Recalibrate drive 1 Bit 0 = Recalibrate drive 0
3Fh	INT 13h	Diskette motor status, where: Bit 7 = 1 Current operation is a write or format = 0 Current operation is a read or verify Bit 6 = Reserved Bits 5,4 = Drive select states where: 00 = Drive 0 selected 01 = Drive 1 selected 10 = Reserved 11 = Reserved Bit 1 = 1 Drive 1 motor is on Bit 0 = 1 Drive 0 motor is on
40h	INT 13h	Diskette motor time-out count

continued

System RAM summary, cont'd

Location	BIOS Service	Description
41h	INT 13h	Diskette status return code, where: Bit 7 = 1 Drive not ready Bit 6 = 1 Seek error occurred Bit 5 = 1 Diskette controller failed Bits 4-0 = Error codes, where: = 00h No error = 01h Illegal function was requested = 02h Address mark not found = 03h Write protect error = 04h Sector not found = 06h Drive door was opened = 08h DMA overrun error = 09h DMA boundary error = 0Ch Media type unknown = 10h CRC failed on disk read
42h-48h	INT 13h	Diskette controller status bytes and command bytes for fixed disk controller
49h	INT 10h	Video mode setting
4Ah	INT 10h	Number of columns on screen
4Ch	INT 10h	Current page size
4Eh	INT 10h	Current page address
50h	INT 10h	Cursor position on each page. Two bytes/page. First byte of each pair is column; second byte is row. 0,0 is upper left corner of screen.
60h	INT 10h	Cursor type defined as 6845 video chip-compatible starting and ending scan lines. High byte is starting scan line; low is ending scan line.
62h	INT 10h	Current page number
63h	INT 10h	6845-compatible I/O port number for current mode (Port 03D4h or 03B4h)
65h	INT 10h	Current mode select register
66h	INT 10h	Current palette value
67h-6Ah	POST	Pointer to reset code upon system reset for real mode re-entry
6Bh	POST	Last unexpected interrupt that occurred
6Ch	INT 1Ah	Least significant timer count
6Eh	INT 1Ah	Most significant timer count
70h	INT 1Ah	24 hour roll over flag

continued

System RAM summary, cont'd

Location	BIOS Service	Description
71h	INT 16h	Ctrl-Break flag
72h	POST	System Reset Flag, where: 1234h = Bypass memory test 4321h = Reserved for ABIOS
74h	INT 13h	Status from last fixed disk operation, where: 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 EOh = Status error/error register is 0 FFh = Sense operation failed
75h	INT 13h	Number of fixed disks
76h	INT 13h	Fixed disk control byte
77h	INT 17h	Fixed disk port offset
78h	INT 14h	Printer time-out table (ports 1-3)
7Ch	INT 16h	Serial time-out table (ports 0-3)
80h	INT 16h	Offset to start of keyboard buffer (from segment 40h)
82h	INT 10h	Offset to end of keyboard buffer (from segment 40h)
84h	INT 10h	Number of rows on screen (24/25)
85h	INT 10h	Character height (bytes/character)

continued

System RAM summary, cont'd

Location	BIOS Service	Description
87h	INT 10h	Video control bits, where: Bit 7 = Clear RAM Bit 6,5 = Memory on adapter as follows: 00b = 64K 01b = 128K 10b = 192K 11b = 256K Bit 4 = Not used Bit 3 = 0 EGA/VGA-compatible adapter active Bit 2 = Wait for display enable Bit 1 = 0 Color or ECD monitor is attached to EGA/VGA-compatible adapter = 1 Monochrome monitor is attached to EGA/VGA-compatible adapter Bit 0 = 0 Translate cursor video modes 0-3 when using ECD monitor in 350 line mode = 1 Inhibit cursor translation
88h	INT 10h	EGA/VGA switch data where: Bits 7-4 = Feature connector bits 3-0, respectively Bits 3-0 = Option switches 3-0, respectively
89h	INT 10h	EGA/VGA control bits, where: Bit 7 = 200 lines Bits 6-5 = Reserved Bit 4 = 400 lines Bit 3 = No palette load Bit 2 = Mono Monitor Bit 1 = Gray Scalling Bit 0 = Reserved
8Ah	INT 13h	Reserved
8Bh	INT 13h	Diskette data rate information Bits 7,6 = Last data rate set by controller, where: 00b = 500 Kilobytes/second (Kbs) 01b = 300 Kbs 10b = 250 Kbs 11b = Reserved Bits 5,4 = Last diskette drive step rate selected Bits 3,2 = Data transfer rate at operation start, where: 00b = 500 Kbs 01b = 300 Kbs 10b = 250 Kbs 11b = Reserved Bits 1,0 = Reserved
8Ch	INT 13h	Fixed disk status register
8Dh	INT 13h	Fixed disk error register
8Eh	INT 13h	Fixed disk interrupt flag

continued

System RAM summary, cont'd

Location	BIOS Service	Description
8Fh	INT 13h	Diskette controller information, where: Bit 7 = Reserved Bit 6 = 1 Drive determined for drive 1 Bit 5 = 1 Drive 1 is multirate Bit 4 = 1 Drive 1 supports change line Bit 3 = Reserved Bit 2 = 1 Drive determined for drive 0 Bit 1 = 1 Drive 0 is multirate Bit 0 = 1 Drive 0 supports change line
90h-91h	INT 13h	Media type of both drives: (One byte per drive. drive 0 at 40:90h; drive 1 at 40:91h) Bits 7,6 = Data Transfer Rate, where: 00b = 500 Kbs 01b = 300 Kbs 10b = 250 Kbs Bit 5 = 1 Double stepping required (360K media/1.2 MB drive) Bit 4 = 1 Known media in drive Bit 3 = Reserved Bits 2-0 = Definitions on return to user: 111b = 720K media in 720K or 1.44 MB drive; or 1.44 MB media in 1.44 MB drive 101b = Known 1.2 MB media in 1.2 MB drive 100b = Known 360K media in 1.2 MB drive 011b = Known 360K media in 360K drive 010b = Trying 1.2 MB media in 1.2 MB drive 001b = Trying 360K media in 1.2 MB drive 000b = Trying 360K media in 360K drive
92h	INT 13h	Diskette device service work area. Each entry is first diskette device service value tried. One byte per drive. Drive 0 at 92h, Drive 1 at 93h.
94h	INT 13h	Current track number for both drives. One byte per drive. Drive 0 at 94h, drive 1 at 95h.
96h	INT 16h	Status byte: Bit 7 = 1 Read ID in progress Bit 6 = 1 Last code was first ID Bit 5 = 1 Forced Num Lock Bit 4 = 1 101/102 keyboard used Bit 3 = 1 Right Alt active Bit 2 = 1 Right Ctrl active Bit 1 = 1 Last code was E0h Bit 0 = 1 Last code was E1h

continued

System RAM summary, cont'd

Location	BIOS Service	Description
97h	INT 16h	Status byte: Bit 7 = Error flag for keyboard command Bit 6 = LED update in progress Bit 5 = RESEND received from keyboard Bit 4 = ACK received from keyboard Bit 3 = Reserved Bit 2 = Current status of Caps Lock LED Bit 1 = Current status of Num Lock LED Bit 0 = Current status of Scroll Lock LED
98h	INT 15h	User wait flag offset address
9Ah	INT 15h	User wait flag segment address
9Ch	INT 15h	Least significant wait count
9Eh	INT 15h	Most significant wait count
A0h	INT 15h	Wait active flag, where: Bit 7 = 1 Wait time elapsed Bits 6-1 = Reserved Bit 0 = 1 INT 15h, AH = 86h occurred
A8h	INT 10h	Pointer to video parameters and overrides (in segment:offset format)
B0h-B5h		Reserved
B6h-B8h	POST	Reserved for POST
0C0h-0CDh		Reserved
CEh	INT 15h	Count of days since 1-1-80
CFh-FFh		Reserved
100h	INT 05h	Print screen status byte

CBIOS Extended Data Area

The following table lists the definitions for the extended data area located at the top 1K of system RAM. The segment of the extended data area is defined at location 40:0Eh of the BIOS data area.

Location	Length	Description
00h	1 Byte	Number of bytes allocated in multiples of Kilobytes
01h-16h	16 Words	Reserved
17h	1 Byte	Number of POST error codes
18h	5 Words	POST error log
19h-21h		Reserved
22h	1 Word	Device driver far call offset
24h	1 Word	Device driver far call segment
26h	1 Byte	Pointing device flag (1st byte) Bit 7 = 1 Command in Progress Bit 6 = 1 Resend Bit 5 = 1 Acknowledge Bit 4 = 1 Error Bit 3 = 0 Reserved Bits 2-0 = Index count
27h	1 Byte	Pointing device flag (2nd byte) Bit 7 = 1 Device driver far call flag Bits 6-3 = 0 Reserved Bits 2-0 = Package Size
28h	1 Byte	Auxiliary device data
29h	1 Byte	Auxiliary device data
2Ah	1 Byte	Auxiliary device data
2Bh	1 Byte	Auxiliary device data
2Ch	1 Byte	Auxiliary device data
2Dh	1 Byte	Auxiliary device data
2Eh	1 Word	Auxiliary device data
2Fh		Reserved
30h	2 Words	The interrupt vector for INT 07h is stored here during 80387 interrupt code
34h	2 Words	The interrupt vector for INT 01h is stored here during INT 07h emulation code

continued

CBIOS Extended Data Area, Continued

Location	Length	Description
38h	1 Byte	Where the low byte of the MSW is stored by the 80287/80387 interrupt code
39h	1 Word	Watchdog Timer (Counter 3) initial count
3Dh	16 Bytes	Fixed disk parameter table for drive 0, where: **Offset** **Size** **Description** 00h Word Max number of cylinders 02h Byte Max. number of heads 03h Word Reserved 05h Word Starting write precompensation cylinder 07h Byte Not used 08h Byte Control byte 09h Byte Reserved 0Ah Byte Reserved 0Bh Byte Reserved 0Ch Word Landing zone 0Eh Byte Number of sectors/track 0Fh Byte Reserved
4Dh	16 Bytes	Fixed disk parameter table for drive 1, where: **Offset** **Size** **Description** 00h Word Max number of cylinders 02h Byte Max. number of heads 03h Word Reserved 05h Word Starting write precompensation cylinder 07h Byte Reserved 08h Byte Control byte 09h Byte Reserved 0Ah Byte Reserved 0Bh Byte Reserved 0Ch Word Landing zone 0Eh Byte Number of sectors/track
6Ch	1 Byte	Fixed Disk arbitration level and channel number, where: Bits 7-4 = Channel number (00h-0Fh) Bits 3-0 = DMA arbitration level (00h-0Eh)
3F0h	1 Byte	Fixed disk buffer

4

Chapter 4
CMOS RAM Data

Overview

Introduction

The ROM uses nonvolatile CMOS RAM to store real time clock, system configuration, system diagnostic, and other information.

Note: The information contained in CMOS is referenced by both the CBIOS and the ABIOS portions of the BIOS. CMOS RAM information is included here for the convenience of CBIOS users. ABIOS users will find this information repeated in *ABIOS for IBM PS/2 Computers and Compatibles.*

continued

Where CMOS RAM data is located

Information may be stored in up to two areas of CMOS RAM. The table below describes the CMOS RAM areas available to the CBIOS.

Data Area	I/O Location	Size	Description
CMOS RAM Data Area	070h & 071h	64 bytes	These bytes are located on the Motorola MC146818A Real Time Clock CMOS chip (or its equivalent). All implementations of the BIOS make use of this area to store real time clock, POST, and system configuration data.
Extended CMOS RAM Data Area	074h, 075h, and 076h	2K bytes	When implemented on systems that employ more than four adapter slots, the BIOS requires an additional 2K of CMOS RAM. This extended CMOS RAM is primarily used to store POS data.

In this chapter

This chapter defines the contents of the CMOS RAM data areas. The following topics are presented:

- CMOS RAM I/O Ports
- CMOS RAM Data
- Extended CMOS RAM Data

CMOS RAM I/O Ports

The CBIOS accesses the CMOS RAM data areas through the following I/O ports.

I/O Address	Read/Write Status	Description
0070h	W	CMOS RAM address register port, where: Bit 7 = 1 NMI disable = 0 NMI enabled Bits 6-0 = 0 CMOS RAM address
0071h	R/W	CMOS RAM data register port
0074h	W	Extended CMOS RAM address register port, least significant byte
0075h	W	Extended CMOS RAM address register port, most significant byte
0076h	R/W	Extended CMOS RAM data register port

CMOS RAM Data

Introduction

The CBIOS uses 64 bytes of CMOS RAM located in the Motorola MC146818A Real Time Clock CMOS chip to store real time clock and configuration data. The tables below present the following CMOS RAM data definitions:

- Real time clock
- Real time clock status registers
- Configuration

Real time clock data definitions

Real time clock information uses CMOS RAM addresses 00h–09h. These data definitions are presented below in offset order.

Location	Size	Description in binary coded decimal
00h	1 Byte	Current second
01h	1 Byte	Second alarm
02h	1 Byte	Current minute
03h	1 Byte	Minute alarm
04h	1 Byte	Current hour
05h	1 Byte	Hour alarm
06h	1 Byte	Current day of week
07h	1 Byte	Current date
08h	1 Byte	Current month
09h	1 Byte	Current year

continued

Status registers data definitions

The real time status registers use CMOS RAM addresses 0Ah–0Dh. These data definitions are presented below in offset order.

Location	Size	Description
0Ah	1 Byte	Status register A, where: Bit 7 = 1 Update in progress Bits 6–4 = Divider identifying the time-based frequency to use Bits 3–0 = Rate selection bits that define output frequency and periodic interrupt rate
0Bh	1 Byte	Status register B, where: Bit 7 = 0 Run = 1 Halt Bit 6 = 1 Enable periodic interrupt Bit 5 = 1 Enable alarm interrupt Bit 4 = 1 Enable update-ended interrupt Bit 3 = 1 Enable square wave interrupt Bit 2 = 1 Calendar is in binary format = 0 Calendar is in BCD format Bit 1 = 1 24-hour mode = 0 12-hour mode Bit 0 = 1 Enable Daylight Savings Time
0Ch	1 Byte	Status register C, where: Bits 7–4 = IRQF, PF, AF, and UF flags, respectively Bits 3–0 = Reserved
0Dh	1 Byte	Status register D, where: Bit 7 = 1 Real time clock has power Bits 6–0 = Reserved

continued

Configuration data definitions

The system configuration information data definitions use CMOS RAM addresses 0Eh–3Fh. These data definitions are presented below in offset order.

Location	Size	Description
0Eh	1 Byte	Diagnostic status, where: Bit 7 = 1 Real time clock lost power Bit 6 = 1 CMOS RAM checksum is bad Bit 5 = 1 Invalid configuration information found at POST Bit 4 = 1 Memory size compare error at POST Bit 3 = 1 Fixed disk or adapter fails initialization Bit 2 = 1 CMOS RAM time found invalid Bit 1 = 1 Adapters do not match configuration Bit 0 = 1 Time-out reading an adapter ID
0Fh	1 Byte	Reason for shutdown, where: 00h = Power on or soft reset 01h = Memory size pass 02h = Memory test pass 03h = Memory test fail 04h = POST end; boot system 05h = JMP DWord pointer with end-of-interrupt 06h = Protected tests pass 07h = Protected tests fail 08h = Memory size fail 09h = INT 15h Block Move 0Ah = JMP DWord pointer without end-of-interrupt 0Bh = Used by ABIOS
10h	1 Byte	Type of diskette drives: Bits 7–4 = Drive type of drive 0, where: 0000b = No drive 0001b = 360K drive 0010b = 1.2 MB drive 0011b = 720K 0100b = 1.44 MB Bits 3–0 = Drive type of drive 1, where: 0000b = No drive 0001b = 360K drive 0010b = 1.2 MB drive 0011b = 720K 0100b = 1.44 MB
11h	1 Byte	Type of fixed disk drive 0
12h	1 Byte	Type of fixed disk drive 1

continued

Configuration data definitions, cont'd

Location	Size	Description
13h	1 Byte	Password Configuration Bits 7-5 = 0 Reserved Bit 4 = 0 BIOS initializes keyboard to normal speed 1 BIOS initializes keyboard to fast speed* Bits 3-2 = Reserved Bit 1 = 1 Network password installed = 0 Network password not installed Bit 0 = 1 Power-on password installed = 0 Power-on password not installed
14h	1 Byte	Equipment installed, where: Bits 7-6 = Number of diskette drives, where: 00b = 1 Diskette drive 01b = 2 Diskette drives Bits 5-4 = Primary display, where: 00b = Reserved 01b = VGA in 40-column mode 10b = VGA in 80-column mode 11b = VGA in monochrome mode Bits 3-2 = Reserved Bit 1 = 1 80387 installed Bit 0 = 0 Diskette drive installed
15h	1 Byte	Base memory in 1K, low byte
16h	1 Byte	Base memory in 1K, high byte
17h	1 Byte	Expansion memory in 1K, low byte
18h	1 Byte	Expansion memory in 1K, high byte
19h	1 Byte	Adapter ID for channel 0, low byte
1Ah	1 Byte	Adapter ID for channel 0, high byte
1Bh	1 Byte	Adapter ID for channel 1, low byte
1Ch	1 Byte	Adapter ID for channel 1, high byte
1Dh	1 Byte	Adapter ID for channel 2, low byte
1Eh	1 Byte	Adapter ID for channel 2, high byte
1Fh	1 Byte	Adapter ID for channel 3, low byte
20h	1 Byte	Adapter ID for channel 3, high byte
21h	1 Byte	POS 2 configuration byte for channel 0
22h	1 Byte	POS 3 configuration byte for channel 0
* If the machine has extended CMOS RAM, the value for the fast speed is taken from offset 702h.		

continued

Configuration data definitions, cont'd

Location	Size	Description
23h	1 Byte	POS 4 configuration byte for channel 0
24h	1 Byte	POS 5 configuration byte for channel 0
25h	1 Byte	POS 2 configuration byte for channel 1
26h	1 Byte	POS 3 configuration byte for channel 1
27h	1 Byte	POS 4 configuration byte for channel 1
28h	1 Byte	POS 5 configuration byte for channel 1
29h	1 Byte	POS 2 configuration byte for channel 2
2Ah	1 Byte	POS 3 configuration byte for channel 2
2Bh	1 Byte	POS 4 configuration byte for channel 2
2Ch	1 Byte	POS 5 configuration byte for channel 2
2Dh	1 Byte	POS 2 configuration byte for channel 3
2Eh	1 Byte	POS 3 configuration byte for channel 3
2Fh	1 Byte	POS 4 configuration byte for channel 3
30h	1 Byte	POS 5 configuration byte for channel 3
31h	1 Byte	Copy of system board POS 2
32h	1 Byte	CRC for offsets 10-31, high byte
33h	1 Byte	CRC for offsets 10-31, low byte
34h	1 Byte	Miscellaneous information Bits 7-4 = Actual number of RS-232 ports installed Bits 3-0 = Block move status before reset to real mode
35h	1 Byte	Low byte of actual expansion memory size
36h	1 Byte	High byte of actual expansion memory size
37h	1 Byte	Century in BCD
38h-3Eh	1 Byte	Power-on password
39h	1 Byte	Power-on password checksum

Extended CMOS RAM Data

Introduction

The BIOS uses an additional 2K of CMOS RAM to store Programmable Option Select (POS) data for MCA-compatible computer systems that include more than four expansion slots. Fixed disk parameter data for fixed disks other than those listed in the ROM BIOS Fixed Disk Parameter Table are also stored in this extra CMOS RAM.

Extended CMOS RAM data definitions

The table below outlines the contents of the extended CMOS RAM data area.

CMOS RAM Offset	Size	Description
0000h	1 Byte	LSB of adapter ID for channel 0
0001h	1 Byte	MSB of adapter ID for channel 0
0002h	1 Byte	Number of POS values used
0003h	1 Byte	POS 2 for channel 0
0004h	1 Byte	POS 3 for channel 0
0005h	1 Byte	POS 4 for channel 0
0006h	1 Byte	POS 5 for channel 0
0007-0022h		Reserved
0023h	1 Byte	LSB of adapter ID for channel 1
0024h	1 Byte	MSB of adapter ID for channel 1
0025h	1 Byte	Number of POS values used
0026h	1 Byte	POS 2 for channel 1
0027h	1 Byte	POS 3 for channel 1
0028h	1 Byte	POS 4 for channel 1
0029h	1 Byte	POS 5 for channel 1
002A-0045h		Reserved
0046h	1 Byte	LSB of adapter ID for channel 2
0047h	1 Byte	MSB of adapter ID for channel 2
0048h	1 Byte	Number of POS values used
0049h	1 Byte	POS 2 for channel 2
004Ah	1 Byte	POS 3 for channel 2

continued

Extended CMOS RAM data definitions, cont'd

CMOS RAM Offset	Size	Description
004Bh	1 Byte	POS 4 for channel 2
004Ch	1 Byte	POS 5 for channel 2
0040-0068h		Reserved
0069h	1 Byte	LSB of adapter ID for channel 3
006Ah	1 Byte	MSB of adapter ID for channel 3
006Bh	1 Byte	Number of POS values used
006Ch	1 Byte	POS 2 for channel 3
006Dh	1 Byte	POS 3 for channel 3
006Eh	1 Byte	POS 4 for channel 3
006Fh	1 Byte	POS 5 for channel 3
0070-008Bh		Reserved
008Ch	1 Byte	LSB of adapter ID for channel 4
008Dh	1 Byte	MSB of adapter ID for channel 4
008Eh	1 Byte	Number of POS values used
008Fh	1 Byte	POS 2 for channel 4
0090h	1 Byte	POS 3 for channel 4
0091h	1 Byte	POS 4 for channel 4
0092h	1 Byte	POS 5 for channel 4
0093-00AEh		Reserved
00AFh	1 Byte	LSB of adapter ID for channel 5
00B0h	1 Byte	MSB of adapter ID for channel 5
00B1h	1 Byte	Number of POS values used
00B2h	1 Byte	POS 2 for channel 5
00B3h	1 Byte	POS 3 for channel 5
00B4h	1 Byte	POS 4 for channel 5
00B5h	1 Byte	POS 5 for channel 5
00B6-00D1h		Reserved
00D2h	1 Byte	LSB of adapter ID for channel 6
00D3h	1 Byte	MSB of adapter ID for channel 6

continued

Extended CMOS RAM data definitions, cont'd

CMOS RAM Offset	Size	Description
00D4h	1 Byte	Number of POS values used
00D5h	1 Byte	POS 2 for channel 6
00D6h	1 Byte	POS 3 for channel 6
00D7h	1 Byte	POS 4 for channel 6
00D8h	1 Byte	POS 5 for channel 6
00D9-00F4h		Reserved
00F5h	1 Byte	LSB of adapter ID for channel 7
00F6h	1 Byte	MSB of adapter ID for channel 7
00F7h	1 Byte	Number of POS values used
00F8h	1 Byte	POS 2 for channel 7
00F9h	1 Byte	POS 3 for channel 7
00FAh	1 Byte	POS 4 for channel 7
00FBh	1 Byte	POS 5 for channel 7
00FC-0160h		Reserved
0161-0162h	1 Word	Set to a value to make POST's CRC for extended CMOS RAM locations 0-162 equal zero. Maintained by the Reference Diskette.
0163-0165h	3 Bytes	Actual extended memory size when over 65 MB
0166-0175h	16 Bytes	Fixed Disk Parameter Table for drive 0
0176-0185h	16 Bytes	Fixed Disk Parameter Table for drive 1
0186h	1 Byte	POST uses this offset to test whether extended CMOS RAM can be accessed correctly.
0187h-0188h		Reserved
0189h-018Dh		Reserved for Reference Diskette POST use
018Eh	1 Byte	Number of Micro Channel slots.
018Fh-0388h		Reserved
0389h	1 Byte	Error Log Number (0-5)
038Ah	20 Bytes	Error Log Block 0
039Eh	20 Bytes	Error Log Block 1
03B2h	20 Bytes	Error Log Block 2

continued

Extended CMOS RAM data definitions, cont'd

CMOS RAM Offset	Size	Description
03C6h	20 Bytes	Error Log Block 3
03DAh	20 Bytes	Error Log Block 4
03EEh	20 Bytes	Error Log Block 5
0402h-06FFh	20 Bytes	Reserved
0700h-07FFh	20 Bytes	Reserved

5

Chapter 5
ROM BIOS Data

Overview

Introduction

In addition to the code for the various CBIOS services, the ROM BIOS also contains tabular data that is used to initialize devices and to insure IBM family compatibility.

In this chapter

This chapter defines the contents of the ROM BIOS data areas. The following topics are presented:

- ROM Address Compatibility Table
- System Configuration Data Table
- Feature Control
- Diskette Parameters Table
- Fixed Disk Parameters Table
- Baud Rate Initialization

ROM Address Compatibility Table

Description

The Phoenix CBIOS maintains compatibility by providing a list of jump instructions that insure that both table information and calls to specific CBIOS services are located at the IBM-compatible entry points.

Compatible ROM Addresses

Address	Description
FE05Bh	Compatible POST Entry Point
FE2C3h	NMI Handler Entry Point
FE3FEh	Compatible INT 13h Fixed Disk Service Entry Point
FE401h	Fixed Disk Parameters Table
FE6F2h	Compatible INT 19h Boot Load Service Entry Point
FE6F5h	Configuration Data Table
FE729h	Baud Rate Generator Table
FE739h	Compatible INT 14h Serial Communications Service Entry Point
FE82Eh	Compatible INT 16h Keyboard Service Entry Point
FE987h	Compatible INT 09h Keyboard Service Entry Point
FEC59h	Compatible INT 13h Diskette Service Entry Point
FEF57h	Compatible INT 0Eh Diskette Hardware ISR Entry Point
FEFC7h	Diskette Controller Parameter Table
FEFD2h	Compatible INT 17h Printer Service Entry Point
FF045h	Compatible INT 10 Functions 0-Fh Entry Point
FF065h	Compatible INT 10h Video Support Service Entry Point
FF0A4h	MDA/CGA Video Parameters Table (INT 1Dh)
FF841h	Compatible INT 12h Memory Size Service Entry Point
FF84Dh	Compatible INT 11h Equipment List Service Entry Point
FF859h	Compatible INT 15h System Services Entry Point
FFA6Eh	Character Font for 320X200 and 640X200 Graphics (lower 128 characters)
FFE6Eh	Compatible INT 1Ah Time-of-day Service Entry Point
FFEA5h	Compatible INT 08h System Timer ISR Entry Point

continued

Compatible ROM Addresses, cont'd

Address	Description
FFEF3h	Initial Interrupt Vector Offsets Loaded by POST
FFF53h	IRET Instruction for Dummy Interrupt Handler
FFF54h	Compatible INT 05h Print Screen Service Entry Point
FFFF0h	Power-up Entry Point
FFFF5h	ASCII Date ROM was built — 8 characters xx/xx/xx
FFFFEh	System Model ID (See System Services)

System Configuration Data Table

Description

The System Configuration Table is located in the CBIOS ROM at F000:E6F5h. This table can be called into RAM via INT 15h function AH = C0h Return System Configuration Parameters.

Parameter table structure

Offset	Initial value	Length	Description
00h	0	1 Word	Number of bytes in this table (minimum = 8)
02h	1	1 Byte	Model byte, where: Model 80 = F8h Model 70 = F8h Model 60 = FCh Model 50 = FCh Model 30 = FAh Model 25 = FAh Unknown system board = FFh
03h	1	1 Byte	Submodel byte, where: Model 80 = 00h or 01h Model 70 = 09h Model 60 = 05h Model 50 = 04h Model 30 = 00h Model 25 = 01h Unknown system board = FFh
04h	1	1 Byte	BIOS revision level (first release = 00b)
05h	1	1 Byte	Feature information byte, where: Bit 7 = 1 Fixed disk BIOS uses DMA channel 3 Bit 6 = 1 Second interrupt chip present Bit 5 = 1 Real time clock present Bit 4 = 1 Keyboard intercept (INT 15h, function AH = 4Fh) called by keyboard interrupt service (INT 09h) Bit 3 = 1 Wait for external event supported Bit 2 = 1 Extended BIOS data area is allocated Bit 1 = 1 Micro Channel supported = 0 PC-type I/O channel implemented Bit 0 = 0 Reserved
06h	0	1 Byte	Feature information byte 2 (reserved, zeros)
07h	1	1 Byte	Feature information byte 3 (reserved, zeros)
08h	1	1 Byte	Feature information byte 4 (reserved, zeros)
09h	0	1 Byte	Feature information byte 5 (reserved, zeros)

Feature Control

Description

The PS/2 BIOS is designed to use the configuration control information in the System Configuration Table, which is accessed using CBIOS INT 15h function C0h, Return System Configuration Parameters.

However, the System Configuration Table does not provide enough information for a PS/2 BIOS to operate properly. Additional data about the system configuration is stored at location F000:E97Fh, and is described below.

Using PS/2 model features as configuration parameters

There are several system features which can be used as system configuration parameters to define PS/2-compatible models. These features are:

- number of adapter slots,
- existence of extended CMOS RAM,
- position of the password disable bit, and
- the memory encoding and memory definition register implementation.

The features listed above are used as parameters by the BIOS to distinguish various PS/2-compatible system configurations.

Feature table

The following table describes how these features vary among several PS/2 models.

Feature	Model 50	Model 60	16 MHz Model 80	20 MHz Model 80
Adapter slots	4	8	8	8
Extended CMOS	No	Yes	Yes	Yes
Password disable bit	Bit 1	Bit 1	Bit 4	Bit 4
Memory registers	None	None	Definition 1	Definition 2

continued

Setting the adapter board data

The number of adapter boards is described at address F000:E97Fh. This number can be 1 to n.

If...	Then...
four or fewer adapter slots on the board and extended CMOS is available	the PS/2 BIOS expects that all slot definitions are stored in extended CMOS
five or more adapter slots on the board	extended CMOS RAM must be available.

Use a reference diskette utility to copy the adapter card definitions to CMOS RAM to ensure proper system operation.

Password disable bit

The password disable bit is in the POS 3 register of the system board at I/O address 0103h. This bit specifies whether the power-on password should be checked for or not. The password disable bit is described at address F000:E980h. A one is required to proceed normally to query the user for a password. A zero indicates that the password will be zeroed out.

If the memory registers are defined, the password disable bit cannot be in bits 0-3 of the system board POS slot 3, and must be in bits 4-7.

Memory register definition

There are three parts in each memory definition, each one byte long:

- memory card register,
- memory encoding register 1, and
- memory encoding register 2.

The memory registers for IBM Model 80-compatible systems are implemented differently in 16 MHz and 20 MHz versions. The memory registers start at address F000:E981h.

PS/2-compatible systems may implement memory register configuration on Model 50 or Model 60-compatible systems, since the Intel 80286 and 80386SX microprocessors support memory encoding.

Diskette Parameters Table

Introduction

The diskette device service routine in older systems supplied a table of parameters used to manipulate diskette drives. For compatibility, the diskette device service routine must still provide the same table although some of its values are modified so that the table is more suited to the default 1.44 MB diskette drives. The default 11-byte table is located at F000:EFC7h and is pointed to by the interrupt 1Eh vector.

Parameter table structure

Byte	Description
00h	The format for the first data byte of the diskette specify command is: Bits 7-4 = Stepping rate Bits 3-0 = Head unload time The default stepping rate is 0Ah for the 1.44 MB drive with a 1.44 MB diskette inserted. The default head unload time is the maximum value of 240 ms (0Fh).
01h	The format for the second data byte of the diskette specify command is: Bits 7-1 = Head load time Bit 0 = Non-DMA mode flag The default head load delay time is set to the minimum value of 4 ms (01h). The heads are loaded at the same time as the motor is started, but the motor delay is much longer so the head load time delay is not really needed. The non-DMA mode flag is always set to zero to indicate that DMA is being used.
02h	Motor turn-off delay. This is the amount of time in timer ticks that the diskette device service routine waits before turning off an inactive diskette drive motor. Timer ticks occur 18.2 times per second and the routine waits about two seconds; therefore the default value for this field is 25h.
03h	Bytes per sector. This field is encoded in the following way to match the encoding used by the diskette controller: 00 = 128 bytes per sector 01 = 256 bytes per sector 02 = 512 bytes per sector 03 = 1024 bytes per sector The standard sector size is 512 bytes per sector so the default value for this field is 02h.

continued

Parameter table structure, cont'd

Byte	Description
04h	End of track. Since the sector numbering is one relative, this field is actually the number of sectors per track. For the 1.44 MB diskette in the 1.44 MB drive, this field is 18 sectors per track.
05h	Gap length. This field is the length of the gap between sectors. The default value for this field is 1Bh for the 1.44 MB diskette.
06h	Data length. Since the bytes per sector field is non-zero, this field is meaningless and is set to FFh.
07h	Gap length for format. This field is the length of the gap between sectors to maintain when formatting. The default format gap length for the 1.44 MB diskette is 6Ch.
08h	Filler byte. This byte is the format filler byte and is F6h.
09h	Head seek delay time. This is the amount of time in milliseconds the diskette device service routine must wait for the heads to settle after doing a seek operation. For the 1.44 MB diskette drive, this field is 0Fh.
0Ah	Motor start time. This is the amount of time in eighths of a second that the diskette device service routine must wait for the motor to come up to speed before doing an I/O operation. The 1.44 MB drive has a motor start time of one second so the default value for this field is set to 08h.

Fixed Disk Parameters Table

Description

The Fixed Disk Parameters Table defines the types of fixed disk drives that can be used. As part of system configuration, the INT 41h vector is set up to point to the fixed disk parameters entry associated with drive 0. If fixed disk drive 1 is installed, INT 46h is initialized to point to its corresponding entry in the Fixed Disk Parameters table. To maintain compatibility, the Fixed Disk Parameters Table is based at F000:E401h.

Parameter table structure

Each entry in the Fixed Disk Parameters Table occupies 8 words as defined below.

Offset	Size	Description
00h	2 Bytes	Maximum number of cylinders
02h	1 Byte	Maximum number of heads
03h	2 Byte	Reserved
05h	2 Bytes	Starting write precompensation cylinder
07h	1 Byte	Reserved
08h	1 Byte	Control byte, where: Bit 7,6 = 1 Disable retries (either bit disables) Bit 5 = 1 Defect map present at max, cylinder + 1 Bit 4 = 0 Reserved Bit 3 = 1 More than 8 heads Bits 2-0 = 0 Reserved
09h	3 Bytes	Reserved
0Ch	2 Bytes	Landing zone
0Eh	1 Byte	Number of sectors per track
0Fh	1 Byte	Reserved

continued

Fixed disk parameter entries

The CBIOS portion of the ROM BIOS defines only the first 32 entries in the Fixed Disk Table. Fixed disk type 15 is not used. Definitions for fixed disk types 1–32 are located in a 16–byte table starting at F000:E401h.

Table of fixed disk definitions

The defined entries into the Fixed Disk Parameters table are listed below. Wherever possible, the manufacturer name and model number associated with a given drive type are listed in the column, "Manufacturer."

Type	Manufacturer	Cyl.	Heads	Write Precomp	Landing Zone	Sectors/ Track	Defect Map
1	IBM 10 MB	306	4	128	305	17	No
2	IBM 20 MB Seagate ST-225 CDC Wren II 9415-5-25 Miniscribe 8438F	615	4	300	615	17	No
3	IBM 30 MB	615	6	300	615	17	No
4	IBM 62 MB	940	8	512	940	17	No
5	IBM 46 MB	940	6	512	940	17	No
6	IBM 20 MB Miniscribe MS 8425 Seagate ST4026 Tandon TM 262 Tandon TM 702AT	615	4	0FFFFh	615	17	No
7	IBM 30 MB	462	8	256	511	17	No
8	IBM 30 MB Seagate ST-4038 CDC Wren II 9415-5-38 Tandon TM 703AT	733	5	0FFFFh	733	17	No
9	IBM 112 MB	900	15	0FFFFh	901	17	No
10	IBM 20 MB	820	3	0FFFFh	820	17	No
11	IBM 35 MB	855	5	0FFFFh	855	17	No
12	IBM 49 MB	855	7	0FFFFh	855	17	No
13	IBM 20 MB	306	8	128	319	17	No

Note: If a table entry contains 0FFFFh for Write Precompensation, then there is no write precompensation for this disk. If the write precompensation is zero, then there is write precompensation for all cylinders.

continued

Fixed Disk Parameters Table, Continued

Table of fixed disk definitions, cont'd

Type	Manufacturer	Cyl.	Heads	Write Precomp	Landing Zone	Sectors/ Track	Defect Map
14	IBM 42 MB	733	7	0FFFFh	733	17	No
15	Not used						
16	IBM 20 MB	612	4	0	663	17	No
17	IBM 40 MB	977	5	300	977	17	No
18	IBM 56 MB	977	7	0FFFFh	977	17	No
19	IBM 59 MB	1024	7	512	1023	17	No
20	IBM 30 MB	733	5	300	732	17	No
21	IBM 42 MB	733	7	300	732	17	No
22	IBM 30 MB	733	5	300	733	17	No
23	IBM 10 MB	306	4	0	336	17	No
24	IBM 20 MB	612	4	305	663	17	No
25	IBM 10 MB	306	4	0FFFFh	340	17	No
26	IBM 20 MB	612	4	0FFFFh	670	17	No
27	IBM 40.5 MB	698	7	300	732	17	Yes
28	IBM 40.5 MB	976	5	488	977	17	Yes
29	IBM 10 MB	306	4	0FFFFh	340	17	No
30	IBM 20 MB	611	4	306	663	17	Yes
31	IBM 42.5 MB	732	7	300	732	17	Yes
32	IBM 42.5 MB	1023	5	0FFFFh	1023	17	Yes

Note: If a table entry contains 0FFFFh for Write Precompensation, then there is no write precompensation for this disk. If the write precompensation is zero, then there is write precompensation for all cylinders.

Baud Rate Initialization

Baud rate initialization table

The baud rate initialization table is located at F000:E729h in the ROM BIOS.

Baud Rate	Divisor
110	0417h
150	0300h
300	0180h
600	00C0h
1200	0060h
2400	0030h
4800	0018h
9600	000Ch
19200	0006h

How baud rate divisor is calculated

The input frequency to the device is 1.8432 MHz. The values in the table are calculated as follows:

1,843,200 / 16 = 115200 / Baud Rate = Divisor

For example, a baud rate of 2400 has a divisor of 115200/2400, which equals 48 decimal, 30 hex.

6

Chapter 6
Power-On Self Tests (POST)

Overview

Introduction

Before a computer system can be used, all system components must be tested and initialized, and the operating system must be bootstrapped into memory. The process of system test and initialization is generally under the control of the computer system's ROM BIOS. It is commonly referred to by the acronym POST, for power-on self test and initialization procedure.

ABIOS and POST

The POST procedure is contained in the CBIOS portion of the BIOS code. The ABIOS is not in any way associated with POST.

Model 25/30 POST

Error messages and POST procedures that are specific to the PS/2 Model 25 and Model 30 are documented in Appendix C.

continued

How POST is started

The CBIOS POST process can be started in any of the ways described in the table below:

Start Method	CBIOS Behavior
Apply power to system *(i.e. turn system on.)*	Jump to entry point indicated by the processor reset vector. All POST tests and initializations are executed. POST invokes CBIOS INT 19h, Operating System Bootstrap Loader.
Reset system via optional hardware reset button.	Jump to entry point indicated by the processor reset vector. All POST tests and initializations are executed. POST invokes CBIOS INT 19h, Operating System Bootstrap Loader.
Press <Ctrl> <Alt> <Del> *(warm boot)*	INT 09h keyboard hardware interrupt service routine transfers control to CBIOS POST. The POST test and initialization of memory above 64K is **not** executed. All other POST tests and initializations are executed. POST invokes CBIOS INT 19h, Operating System Bootstrap Loader.

Processor reset vector location

To maintain compatibility, the processor reset vector points to address F000:0000h. This vector is also dual mapped by external address selection logic to point to FFFFF:00000h — the 80286/80386 reset vector address.

In this chapter

This chapter presents the following information:

- POST Procedures
- Rules for Positioning I/O Expansion ROM
- Re-entering Real Mode
- INT 19h, INT 18h, and System Boot
- POST Error Handling

POST Procedures

Introduction

The initialization and self test functions of the POST process are tightly interwoven. The processes can be generally thought of, however, as falling into two categories:

- Processes related to the central system hardware
- Processes related to configuration and nonsystem board hardware

How POST handles errors

If POST finds errors on its first pass through the boot procedure, it looks for the Reference Diskette in Diskette Drive A:. If the Reference Diskette is not in Drive A:, POST generates a message, **Press F1 to Retry.**

If the Reference Diskette is in Diskette Drive A:, POST causes it to load so the error message can be handled by the end user.

Note: See the POST Error Handling information in this chapter.

Central hardware POST tests

Because the proper functioning of the central hardware is required before further POST tests can occur, the most central hardware is first tested, then initialized.

POST tests and initializes the following central system components in the order listed below:

- The Central Processing Unit (CPU)
- ROM BIOS (checksum)
- CMOS RAM
- Intel 8237 Direct Memory Access (DMA) Controller
- Intel 8042/8742 Keyboard Controller
- Base 64K System RAM
- Intel 8259A Programmable Interrupt Controller
- Intel 8254A Programmable Timer
- Intel 82385 Cache Controller, if present

continued

Nonsystem hardware POST tests

Once the central hardware has been tested and initialized, POST verifies that the system configuration specified in CMOS matches the actual hardware present. The remaining POST procedures test and initialize memory above 64K, the keyboard, the diskette and fixed disk drives, the CRT controller, and any nonsystem board hardware.

POST tests and initializes the following configuration and nonsystem board components in the order listed below:

- CMOS RAM Configuration data
- CRT controllers
- RAM memory above 64K
- Keyboard
- Pointing device (mouse, etc.)
- Diskette drive A availability
- Serial interface circuitry
- Diskette controllers
- Fixed disk controllers

Rules for Positioning I/O Expansion ROM

Introduction

An I/O expansion ROM is an optional extension of the BIOS which is inserted into the machine environment as part of an added I/O subsystem. I/O ROMs often replace part of the BIOS for a certain function. (e.g. ESDI fixed disk ROM BIOS.)

I/O expansion ROMs are detected by POST and are given an opportunity to initialize themselves and their hardware environment.

Predefined address ranges

POST searches for expansion ROM over two predefined address ranges. The predefined address ranges are listed in the table below.

Address Range	Description
C0000 - C7FFFh	VIDEO EXPANSION ROM POST searches the address range beginning at C0000h and up to but not including C8000h for the existence of Video Expansion ROM. This search proceeds in 2K increments.
C8000 - DFFFFh	GENERAL EXPANSION ROM POST searches the address range beginning at C8000h and up to but not including E0000h for the existence of General Expansion ROM. This search proceeds in 2K increments.

Rules for detecting ROM

For POST to detect an expansion ROM, the first two words of each expansion ROM area must be set up as described below.

I/O ROM Byte	Value
0	55h
1	AAh
2	ROM length in 512-byte blocks
3	Entry point for ROM initialization (via FAR CALL)

Note: POST calculates the checksum of the area indicated by I/O ROM Byte 2. If the checksum is zero, POST will call the expansion ROM initialization code, which must begin at Byte 3.

Re-entering Real Mode

Re-entry into real mode

Several POST processes require that the system be put into protected mode.

When returning to real mode from protected mode, the processor is reset, therefore POST is re-entered. In order to prevent re-initialization of the system, POST reads the shutdown code stored in location 0Fh in CMOS RAM. It then jumps around the initialization procedures to the appropriate entry point.

Table of shutdown code values

As POST proceeds, it will use the shutdown codes with values 01h, 02h, 03h, 04h, 06h, 07h, and 08h to return the processor to real mode.

Shutdown codes 05h and 0Ah are more flexible; they return control to the address stored in a double word pointer at 40:67h. These vectors are provided for use by operating system extensions such as I/O ROMs. Shutdown code 05h flushes the keyboard buffer and performs an end-of-interrupt (EOI) prior to vectoring. Shutdown code 0Ah simply vectors.

CMOS RAM Location	Size	Description
0Fh	1 Byte	SHUTDOWN CODE 00h = Power-on or soft reset 01h = Memory size pass 02h = Memory test pass 03h = Memory test fail 04h = POST end; boot system 05h = JMP dword pointer with end-of-interrupt 06h = Protected tests pass 07h = Protected tests fail 08h = Memory size fail 09h = INT 15h Block Move 0Ah = JMP dword pointer without end-of-interrupt

INT 19h, INT 18h, and System Boot

Introduction

Once the POST test and initialization procedures have been completed, POST boots the system by issuing an INT 19h.

INT 19h behavior

INT 19h behaves as described in the table below:

If...	And...	Then...
a diskette drive is configured	a bootable diskette is in place	INT 19h reads the diskette boot sector and places its contents at address 0000:7C00h
a fixed disk is configured	there is no bootable diskette in the diskette drive	INT 19h reads the fixed disk boot sector and places its contents at address 0000:7C00h
no boot sector is found on the diskette	no boot sector is found on the fixed disk	INT 19h issues INT 18h. INT 18h displays the message: "NO BOOT DEVICE AVAILABLE"

Note: INT 19h performs password checking routines after the boot sector is loaded. For more information on INT 19h password checking, refer to Chapter 16.

INT 18h behavior

INT 18h can be vectored to a "no boot routine" which takes over the boot process. An example of such a routine would be one that allows the system to be booted over a network.

By default, the CBIOS initializes INT 18h to point to a routine that displays the message: "NO BOOT DEVICE AVAILABLE."

POST Error Handling

Introduction

POST reports test or initialization failures in one of the following ways:

- Beep codes
- System failure messages
- Boot failure messages
- Informational messages

See Appendix A, CBIOS Error Messages, for a complete listing of POST messages.

7

Chapter 7
INT 02h Nonmaskable Interrupt

Overview

Introduction

When the system hardware activates the Nonmaskable Interrupt (NMI) pin on the microprocessor:

- the Central Arbitration Control Point is implicitly disabled and
- the CBIOS INT 02h NMI interrupt service routine (ISR) is invoked.

CBIOS NMI Interrupt Service Routine (ISR)

The CBIOS NMI ISR determines the reason for the NMI and takes appropriate action. In general, this involves:

- explicitly re-enabling the central arbitration control point,
- displaying a run time error message, and
- halting the processor.

Note: NMIs can only be cleared by a system reboot. CBIOS supports NMI clearing by warm (<Ctrl> <Alt> <Del>) reboot or cold reboot (system reset button or power switch).

continued

INT 02h vector

The CBIOS initializes the INT 02h vector to F000:E2C3h. The INT 02h vector resides at address 00:08h in the CBIOS Interrupt Vector Table.

In this chapter

This chapter describes the various NMI sources and tells how the CBIOS Nonmaskable Interrupt ISR handles them.

The following topics are presented:

- Nonmaskable Interrupt I/O Ports
- Parity Error NMI Handling
- Time-Out Error Handling
- 80X87 Coprocessor Exceptions

Nonmaskable Interrupt I/O Ports

INT 02h, Nonmaskable Interrupt, references the following I/O ports:

I/O Address	Read/Write Status	Description
0018h	W	DMA extended function register, where: Bits 7-4 = Progress command, where: 00h I/O address register 01h Reserved 02h Memory address register write 03h Memory address register read 04h Transfer count register write 05h Transfer count register read 06h Status register read 07h Mode register 08h Arbus register 09h Mask register set single bit 0Ah Mask register reset single bit 0Bh-0Ch Reserved 0Dh Master clear 0Eh-0Fh Reserved Bits 3-0 = 0 Reserved
0021h	R/W	Interrupt controller 1 mask, where: Bit 7 = 0 Enable parallel port 1 interrupt Bit 6 = 0 Enable diskette interrupt Bit 5 = 0 Enable fixed disk interrupt Bit 4 = 0 Enable serial port 1 interrupt Bit 3 = 0 Enable serial port 2 interrupt Bit 2 = 0 Enable cascade interrupt Bit 1 = 0 Enable keyboard interrupt Bit 0 = 0 Enable timer channel 0
0047h	W	Control word register for counter 3 Bits 7-6 = 00b Select counter 3 = 01b Reserved = 10b Reserved = 11b Reserved Bits 5-4 = 00b Counter latch command select counter 0 = 01b Read/write counter bits 0-7 only = 10b Reserved = 11b Reserved Bits 3-0 = 00b
0061h	R	System control port B, where: Bit 7 = 1 Parity check Bit 6 = 1 Channel check Bit 5 = 1 Timer 2 output Bit 4 = 1 Toggles with each refresh request Bit 3 = 0 Channel check enabled Bit 2 = 0 Parity check enabled Bit 1 = 1 Speaker data enabled Bit 0 = 1 Timer 2 gate to speaker enabled

continued

Nonmaskable Interrupt I/O Ports, Continued

I/O Address	Read/Write Status	Description
0061h	W	System control port B, where: Bit 7 = 1 Reset timer 0 output latch (IRQ0) Bits 6-4 = Reserved Bit 3 = 0 Channel check enabled Bit 2 = 0 Parity check enabled Bit 1 = 1 Speaker data enabled Bit 0 = 1 Timer 2 gate to speaker enabled
0070h	W	CMOS RAM address register port, where: Bit 7 = 1 NMI disable = 0 NMI enabled Bits 6-0 = CMOS address
0071h	R/W	CMOS RAM data register port
0090h	W	Arbitration register, where: Bit 7 = 1 Enable system microprocessor cycle Bit 6 = 1 Arbitration mask Bit 5 = 1 Enable extended arbitration Bits 4-0 = 0 Reserved
0090h	R	Arbitration register, where: Bit 7 = 1 System microprocessor cycle enabled Bit 6 = 1 Arbitration mask by NMI Bit 5 = 1 Bus time-out Bit 4 = 0 Reserved Bits 3-0 = Arbitration level
0092h	R/W	System control port A, where: Bit 7 = 1 Fixed disk activity light bit A on Bit 6 = 1 Fixed disk activity light bit B on Bit 5 = Reserved Bit 4 = 1 Watchdog time-out occurred Bit 3 = 1 Security lock latch locked Bit 2 = Reserved Bit 1 = 1 Alternate gate A20 active Bit 0 = 1 Alternate hot reset
0096h	R/W	Channel position select register, where: Bit 7 = 1 Channel 1 reset Bits 6-4 = Reserved (written as 0, read as 1) Bit 3 = 1 Channel select Bits 2-0 = 0 Channel number
0105h	R/W	POS register 5 where: Bit 7 = 0 Channel check condition occurred = 1 Channel reset Bit 6 = 0 Channel check exception status available = 1 No status available

Parity Error NMI Handling

Introduction

Parity errors in system memory cause system hardware to activate the processor's NMI pin.

There are two types of memory parity errors:

- system board memory failure
- I/O board memory failure

Parity error handling in the Phoenix CBIOS

The table below describes how parity errors are handled by the Phoenix CBIOS NMI service, and it suggests possible causes and solutions for each error type.

Error Code	Message	Possible Cause	Suggested Solution
110	Memory Parity Error at *xxxxx*	Memory is failing. If the NMI handler can determine the address of the failing memory, it will be displayed in place of *xxxxx*. If the failing memory is not found, then the message will read **"Memory Parity Error ????"**.	Check the system board and all installed devices for proper operation. Replace if necessary.
111	I/O Card Parity Error at *xxxx*	The peripheral card has failed. If the NMI handler can determine the address of the failing memory, it will be displayed in place of *xxxxx*. If the failing memory is not found, then the message will read **"I/O Card Parity Error ????"**.	Check the system board and all installed devices for proper operation. Replace if necessary.

continued

Parity error handling in IBM PS/2 systems

IBM PS/2 systems have a different set of error messages. The messages are followed by the segment address of the bad memory location or ????? if it cannot be found. These error codes and messages are listed in the following table.

Error Code	Message
110	System Board Memory Failure
111	I/O Channel Check Activated

Models 25 and 30: IBM PS/2 Models 25 and 30 (and compatibles) support memory parity error in the same way as other PS/2 systems but substitute the message PARITY CHECK1 for message 110 and PARITY CHECK2 for message 111.

Time-Out Error Handling

Introduction

Time-out errors cause the system hardware to set the processor's NMI pin. There are two kinds of NMI-related time-out errors:

- Watchdog time-out errors
- Direct Memory Access (DMA) bus time-out errors

Watchdog time-out errors

The watchdog time-out function is enabled via INT 15h, AH = C3h, Watchdog Time-Out.

When enabled, the watchdog time-out function ensures that a program does not turn interrupts off for too long a period. It does this by "watching" for a missing timer interrupt IRQ 0. (Timer interrupts occur 1024 times per second.) Should a timer interrupt be missed, the watchdog time-out function generates an NMI.

DMA bus time-out errors

When a DMA driven device uses the bus longer that 7.8 microseconds, the Central Arbitration Control Point generates an NMI.

continued

Time-out and error handling in the Phoenix CBIOS

The table below describes how the CBIOS NMI ISR handles watchdog and DMA bus time-out errors and suggests possible causes and solutions for each error type.

Error Code	Message	Possible Cause	Suggested Solution
112	Watchdog Time-Out	The watchdog timer has detected that the timer 0 interrupt has not been serviced. A program or device has probably failed while interrupts were off.	Reboot the system. If the problem persists, check all installed devices for proper operation. Replace if necessary. If the problem still occurs, examine all programs for errors which cause the interrupts to remain off.
113	DMA Bus Time-Out	A device has driven the -BURST signal line for more than 7.8 microseconds, causing the central arbiter device (on the motherboard) to generate a bus time-out.	Check system board and all installed devices for proper operation. Replace if necessary.

Time-out and error handling in IBM PS/2 systems

IBM PS/2 systems have a different set of time-out error messages. They are listed in the table below.

Error Code	Message
112	Watchdog Time-Out
113	Direct Memory Access (DMA) Bus Time-Out

Models 25 and 30: Time-out error handling is not supported on IBM PS/2 Model 25 and 30 systems.

80x87 Coprocessor Exceptions

How CBIOS maintains compatibility

Unless it is revectored by a user program, the CBIOS INT 75h ISR is initialized to the same entry point as INT 02h. This insures that 8087 software running under CBIOS is given the opportunity to handle coprocessor exceptions by intercepting INT 02h.

8087 coprocessor exceptions

Systems based on the Intel 8088 that incorporate the 8087 math coprocessor indicate math coprocessor exceptions by generating an NMI (i.e. INT 02h).

A coprocessor exception routine should be written to handle 8087 coprocessor exceptions by intercepting INT 02h. The coprocessor exception routine should determine if the NMI was generated by the 8087. If the NMI was not coprocessor-related, the routine should transfer control to the BIOS NMI ISR.

80287/80387 coprocessor exceptions

Systems that incorporate the 80287 or 80387 math coprocessor indicate math coprocessor exceptions by setting IRQ 13. IRQ 13 invokes INT 75h, the CBIOS 80287/80387 Math Coprocessor Exception.

A software routine should be written to handle the coprocessor exceptions by intercepting INT 75h. The coprocessor exception routine determines if the NMI was generated by the coprocessor. If the NMI was not coprocessor-related, this routine transfers control to the CBIOS NMI ISR.

INT 75h vector

The CBIOS initializes the INT 75h vector to F000:E2C3h (i.e. the INT 02h entry point). The INT 75h vector resides at address 00:4Bh in the CBIOS Interrupt Vector Table.

8

Chapter 8
INT 09h and INT 16h Keyboard Services

Overview

Introduction

The CBIOS contains two keyboard-related services:

- **CBIOS keyboard hardware ISR**
  The CBIOS keyboard interrupt service routine (ISR) is invoked via hardware INT 09h each time a key is pressed. The CBIOS keyboard ISR translates System Scan Codes (provided by the 8042 keyboard controller at port 60h) into:
  - Updates of the keyboard shift key and toggle key flags
  - Requests for keyboard ISR internal functions (e.g. Break, Pause, Reset)
  - Entries into the CBIOS keyboard buffer.
- **CBIOS keyboard software DSR**
  The CBIOS keyboard device service routine (DSR) is invoked via software INT 16h. INT 16h is invoked by the operating system or by application programs running under the operating system. The CBIOS keyboard DSR contains functions that read the CBIOS keyboard buffer, write to the buffer, and return the status of the keyboard service flags.

continued

Vector values

CBIOS initializes two vectors:

- INT 09h to point to ROM address F000:E987h, the IBM-compatible entry point. INT 09h resides at address 00:24h in the Interrupt Vector Table, and
- INT 16h to point to ROM address F000:E82Eh, the IBM-compatible entry point. INT 16h resides at address 00:58h in the Interrupt Vector Table.

In this chapter

Because they are so closely interrelated, this chapter presents information on both INT 09h and INT 16h Keyboard Services. The following topics are discussed:

- Hardware Environment
- Keyboard-to-System Communication
- Keyboard I/O Ports
- System RAM Data
- INT 09h: Internal Function Requests
- INT 16h: Keyboard Services
- How to Read Character Code Tables
- Character Codes: AH = 00h/01h
- Character Codes: AH = 10h/11h

Hardware Environment

Introduction

The ROM BIOS supports an intelligent keyboard subsystem based on the Intel 8042.

The hardware interrupt level associated with the CBIOS Keyboard Service is IRQ 1.

The 8042 controller chip

The Intel 8042 peripheral controller (or compatible) is a single-chip microcomputer that can be programmed to allow bidirectional communication between the master microprocessor and up to two auxiliary serial input devices. The 8042 chip, generally, is mounted on the system motherboard. 8042 programs reside as firmware in the 8042 chip itself.

Device support

The kind of devices a given 8042 chip supports are dependent on how the 8042 is programmed.

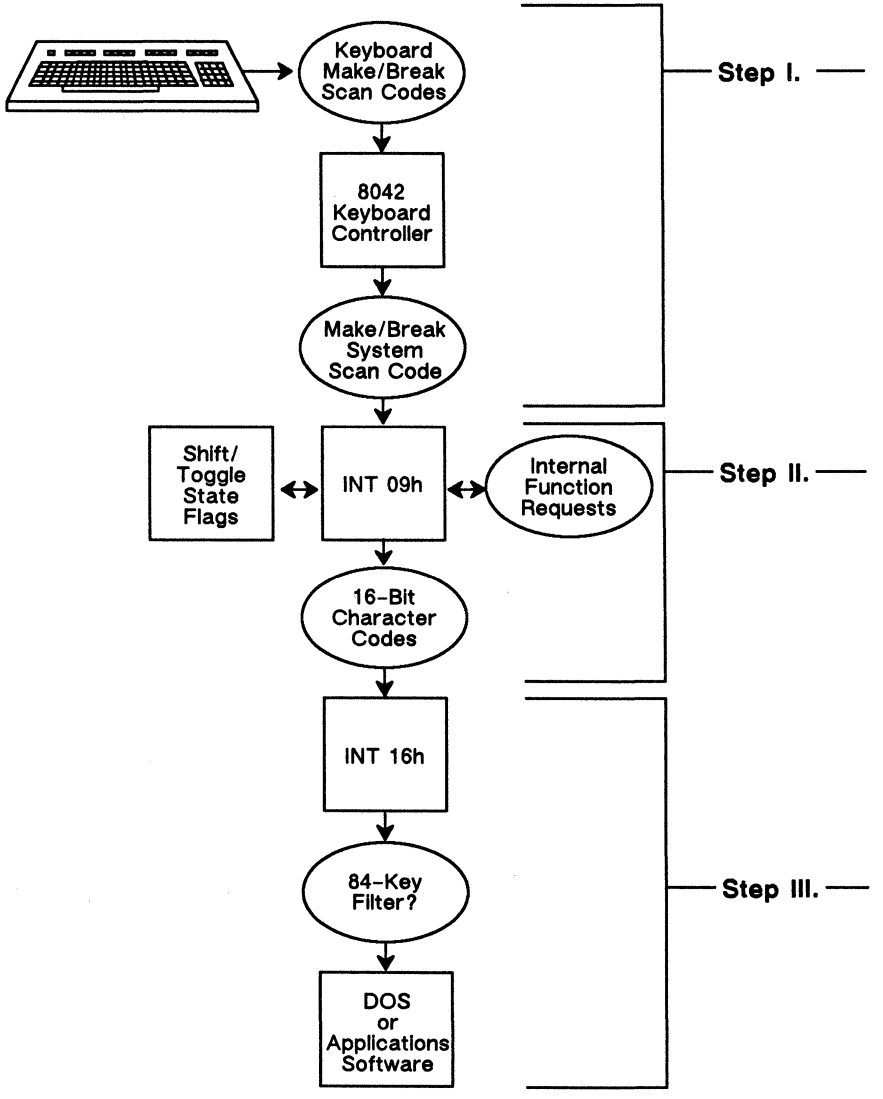
On IBM PS/2-compatible systems, the 8042 is programmed to allow bidirectional communication between the system and the keyboard, as well as between the system and one other auxiliary serial device, such as a mouse, joystick, or trackball.

Reference

For more information on the 8042 keyboard controller, refer to Appendix B.

Keyboard-to-System Communication

The diagram below shows how the keyboard communicates with the system.



Step I.

8042 Keyboard Controller translates keyboard make or break scan codes into make or break System Scan Codes. 8042 System Scan Codes are the only codes the INT 09h keyboard service can interpret.

Step II.

INT 09h accepts the System Scan Codes placed at Port 60h by the 8042 Keyboard Controller and processes each code as follows:

1. System Scan Codes which correspond to the keyboard shift or toggle keys (Shift, Alt, Ctrl, Caps Lock, Num Lock or, Scroll Lock) cause INT 09h to update the Shift/Toggle data variables located in segment 40h of system RAM.

2. System Scan Codes corresponding to nonshift/toggle keys are first compared to the Shift/Toggle state data to see if the shift or toggle of the Character Code corresponding to that key should be generated. For example, the Character Code for a shifted "A" key is different from the one for an unshifted "a" key. If no shift/toggle state data applies, then INT 09h generates the unshifted Character Code for that key. All Character Codes occupy one word and are placed by INT 09h into a 16-word keyboard buffer.

3. Some combinations of Shift and Toggle keys (such as Ctrl/Break) and of Shift keys and nonshift keys (such as Ctrl/Alt/Del) are interpreted by INT 09h as a request for internal function services.

Step III.

When invoked by the operating system or by a software application, INT 16h acts on the contents of the 16-word keyboard Character Code buffer. If desired, character codes can be filtered for compatibility with application software that accepts only those character codes generated by 84-key keyboards.

Keyboard I/O Ports

INT 09h and 16h Keyboard Services reference the following I/O ports:

I/O Address	Read/Write Status	Description
0060h	R/W	8042 Data port
0064h	R	8042 Status port

System RAM Data

Introduction

Throughout the course of operation, the CBIOS keyboard services monitor and maintain several status and buffer-control data definitions.

Keyboard data definitions

The keyboard data definitions, which are stored in segment 40h of the system RAM, are presented below in offset order.

Location	Size	Description
17h	1 Byte	Keyboard shift flags, where: Bit 7 = 1 Insert active Bit 6 = 1 Caps Lock active Bit 5 = 1 Num Lock active Bit 4 = 1 Scroll Lock active Bit 3 = 1 Alt pressed Bit 2 = 1 Ctrl pressed Bit 1 = 1 Left Shift pressed Bit 0 = 1 Right Shift pressed

continued

Keyboard data definitions, cont'd

Location	Size	Description
18h	1 Byte	More keyboard shift flags, where: Bit 7 = 1 Insert pressed Bit 6 = 1 Caps Lock pressed Bit 5 = 1 Num Lock pressed Bit 4 = 1 Scroll Lock pressed Bit 3 = 1 Ctrl-Num Lock state active Bit 2 = 1 Sys Req pressed Bit 1 = 1 Left Alt pressed Bit 0 = 1 Left Ctrl pressed
19h	1 Byte	Work area for Alt key and numeric keypad input
1Ah	1 Byte	Offset to next character in keyboard buffer
1Ch	1 Word	Offset to first available spot in keyboard buffer
1Eh	16 Words	Keyboard buffer of 16 word entries (a maximum of 15 are used at a time)
71h	1 Byte	Bit 7 = 1 If <Ctrl> <Break> pressed
72h	1 Word	Set to 1234h if <Ctrl> <Alt> <Del> is pressed
80h	1 Word	Address to start of keyboard buffer (from segment 40h)
82h	1 Word	Address to end of keyboard buffer (from segment 40h)
96h	1 Byte	Status byte: Bit 7 = 1 Read ID in progress Bit 6 = 1 Last code was first ID Bit 5 = 1 Forced Num Lock Bit 4 = 1 101/102 keyboard used Bit 3 = 1 Right Alt active Bit 2 = 1 Right Ctrl active Bit 1 = 1 Last code was E0h Bit 0 = 1 Last code was E1h
97h	1 Byte	Status byte: Bit 7 = Error flag for keyboard command Bit 6 = LED update in progress Bit 5 = RESEND received from keyboard Bit 4 = ACK received from keyboard Bit 3 = Reserved Bit 2 = Current status of Caps Lock LED Bit 1 = Current status of Num Lock LED Bit 0 = Current status of Scroll Lock LED

INT 09h: Internal Function Requests

Introduction

The CBIOS INT 09h ISR is programmed to interpret certain predefined key combinations as requests for internal functions.

The key or key combinations which the INT 09h ISR interprets as internal function calls are listed below:

- <Ctrl> <Alt> <Del>: SYSTEM RESET
- <Ctrl> <Break> (<Ctrl> <Scroll Lock>): BREAK
- <Ctrl> <Num Lock>: PAUSE
- <Shift> <Print Screen>: PRINT SCREEN
- <SysReq>: SYSTEM REQUEST

INT 09h: System Reset (<Ctrl> <Alt> <Del>)

Description

The CBIOS INT 09h ISR recognizes the <Ctrl> <Alt> <Del> key combination as an internal request for the system RESET routine.

The INT 09h ISR invokes the RESET operation with a direct call to the ROM BIOS code location where the system RESET routine resides.

Input/Output

Input: Keyboard Shift Flags byte (40:17h)
Bit 3 = 1 (Alt pressed)
Bit 2 = 1 (Ctrl pressed)
System Scan Code = 53h (Del pressed)

Output: Reset Flag word (40:72h)
(1 Word) = 1234h
INT 09h initiates system RESET routine

INT 09h: Break (<Ctrl> <Break> or <Ctrl> <Scroll Lock>)

Description

The CBIOS INT 09h ISR recognizes the <Ctrl> <Break> key combination as an internal function request for INT 1Bh. <Break> and <Scroll Lock> are equivalent keys on many keyboards.

INT 1Bh DSR is simply a vector which points to a body of programmer-created or application-program-resident code intended for execution when the <Ctrl> <Break> key combination is received from the keyboard.

Input/Output

Input: Keyboard Shift Flags byte (40:17h):
Bit 2 = 1 (Ctrl pressed)
System Scan Code = 46h (Break pressed)

Output: INT 09h invokes INT 1Bh, a vector address which points to a break routine, if any.

INT 09h: Pause (<Ctrl> <Num Lock>)

Description

The CBIOS INT 09h ISR recognizes the <Ctrl> <Num Lock> key combination as an internal function request for the INT 09h controlled PAUSE state. On a 101-key keyboard, the <PAUSE> key causes the INT 09h ISR to initiate the Pause State.

When the INT 09h ISR puts the computer system into the Pause State, the INT 09h ISR Pause-State routine loops until a valid key or key combination is received from the keyboard.

Input/Output

Input: Keyboard Shift Flags byte (40:17h)
Bit 2 = 1 (Ctrl pressed)
Keyboard Shift Flags byte 2 (40:18h)
Bit 5 = 1 (Num Lock pressed)
Bit 3 = 1 (Ctrl/Num Lock in active state: suspend toggles)
System Scan Code = 45h (Num Lock pressed)

Output: INT 09h loops PAUSE routine until a valid key or key combination is received from 8042 controller.

Ending PAUSE state

To end Pause State, Bit 3 of Keyboard Shift Flags byte 2 (40:18h) must be reset to 0.

The INT 09h ISR resets Bit 3 of Keyboard Shift Flags byte 2 to zero as soon as it receives:

1. The make System Scan Code for any non-shift key except the SYS REQ key, or
2. Any valid shift/toggle key combination.

The valid shift/toggle key combinations needed to reset Bit 3 of Keyboard Shift Flags Byte 2 to zero are shown on the following page.

continued

PAUSE state table

The table below outlines the behavior of the various shift/toggle key combinations during the INT 09h ISR pause state. The key combinations necessary to reset Bit 3 of Keyboard Shift Flags Byte 2 to zero (i.e. end Pause State) are in category #1.

<table>
<tr><th rowspan="2">Shift Keys</th><th colspan="4">** Toggle Keys **</th></tr>
<tr><th>Ins</th><th>Caps Lock</th><th>Scroll Lock</th><th>Num Lock</th></tr>
<tr><td>None
Shift (L or R)</td><td rowspan="2">2.</td><td colspan="3" rowspan="2">3.</td></tr>
<tr><td>Alt
Alt/Shift</td></tr>
<tr><td>Ctrl
Ctrl/Shift</td><td colspan="3" rowspan="2">1.</td><td>4.</td></tr>
<tr><td>Alt/Ctrl
Alt/Ctrl/Shift</td><td>5.</td></tr>
</table>

LEGEND

1. PAUSE STATE IS ENDED (Bit 3 of Keyboard Shift Flags Byte 2 is reset). The scan codes normally associated with these keys do not have their usual effect on the shift and toggle flags. They are interpreted instead as a request to reset Bit 3 of Keyboard Shift Flags Byte 2 to zero, thus ending the Pause State.
2. PAUSE STATE IS NOT ENDED. The 8042 controller chip places the System Scan Code for (Ins) or (Shift Ins) into its internal buffer. The INT 09h ISR will act on these codes once Pause State is ended.
3. PAUSE STATE IS NOT ENDED. These key combinations are treated as valid requests to change Shift/Toggle states or update LEDs.
4. PAUSE STATE IS NOT ENDED. The keys which initiate the Pause State can not be used to end Pause State. When in Pause State, these keys are ignored.
5. PAUSE STATE IS NOT ENDED. When in Pause State, these keys are ignored.

INT 09h: Print Screen (<PrtSc> or <Shift> <PrtSc>)

Description

The CBIOS INT 09h ISR invokes INT 05h (the CBIOS Print Screen service) whenever it receives the combination of System Make Scan Codes corresponding to the left or the right Shift key and the Print Screen key.

The System Make Scan Codes necessary to cause INT 09h to invoke INT 05h can be generated in either of the ways listed below:

1. By pressing the Print Screen key alone, or
2. By pressing either of the Shift keys and the Print Screen key.

Input/Output

Input Case #1: <PrtSc> key only

Input Case #2: <Shift> <PrtSc> keys

System Scan Codes = E0h, 2Ah, E0h, 37h
(Left Shift and Print Screen both depressed)

Keyboard Shift Flags Byte (40:17h)
Bit 1 = 1 (Left Shift pressed)
or
Bit 0 = 2 (Right Shift pressed)
System Scan Codes = E0h, 37h

Output for both Case #1 and Case #2:

INT 09h invokes INT 05h BIOS Print Screen Service

INT 09h: System Request (<SysReq>)

Description

The CBIOS INT 09h ISR interprets the System Request key in either of two ways:

- When the System Request key is depressed, INT 09h interprets the resulting System Scan Code as an internal function request for INT 15h: function AH = 85h, subfunction AL 00h — Process System Request.
- When the System Request key is released, INT 09h interprets the resulting System Scan Code as an internal request for INT 15h, function AH = 85h, subfunction AL = 01h — Terminate System Request.

Process SysReq input/output

The inputs and outputs that result when the System Request key is pressed are listed below:

Input: Keyboard Shift Flags Byte 2 (40:18h)
Bit 2 = 1 (System Request pressed)

Output: AH = 85h
AL = 00h
INT 09h invokes INT 15h Process System Request

Terminate SysReq input/output

The inputs and outputs that result when the System Request key is released are listed below:

Input: Keyboard Shift Flags Byte 2 (40:18h)
Bit 2 = 0 (System Request key released)

Output: AH = 85h
AL = 01h INT 09h invokes INT 15h Terminate System Request

INT 16h: Keyboard Services

Introduction

The CBIOS keyboard services are invoked via software interrupt INT 16h. Individual function codes are selected via the AH register.

Summary of INT 16h functions

A table of INT 16h AH register values and their corresponding function is shown below.

Function	Description
00h	Read Keyboard Input
01h	Return Keyboard Status
02h	Return Shift Flag Status
03h	Set Typematic Rate and Delay
04h	Reserved
05h	Store Key Data
06h–09h	Reserved
10h	Read Extended Key Input
11h	Return Extended Keyboard Status
12h	Return Extended Shift Flags Status
13h–FFh	Reserved

INT 16h: AH = 00h Read Keyboard Input

Description

The Read Keyboard Input function executes as follows:

- If a character code is available in the keyboard buffer, this function reads the character code and returns its value in AX.
- If the keyboard buffer is empty, this function suspends program execution until a character code is inserted into the buffer.
- If the keyboard LEDs do not match the current flag settings, this function updates the LEDs.

Input/Output

Input: AH = 00h

Output: AH = Scan code or character ID for special character
AL = ASCII code or other translation of character.

84-key "filtering"

The read keyboard input function acts as an 84-key keyboard filter, selectively editing the buffer contents for 84-key keyboard compatibility.

When a 101-key keyboard is in place, INT 16h function AH = 00h:

- Returns all standard 84-key keyboard Character Codes as is,
- Adjusts 101-key keyboard duplicates of 84-key keyboard characters so that they take on the same two-byte key code as their 84-key counterparts, and
- Destroys any 101-key Character Codes not compatible with the 84-key keyboard.

Note: The data table labeled "Character Codes: AH = 00h/01h" lists the codes returned when using INT 16h AH = 00h and AH = 01h.

Reference

Function AH = 10h Read Extended Keyboard Input does not modify character codes for 84-key keyboard compatibility. For further details, see the Function: AH = 10h heading in this chapter.

INT 16h: AH = 01h Return Keyboard Status

Description

This function is useful for allowing programs to check for keyboard input while continuing to run if there is none. It executes as follows:

- Checks to see if a two-byte character code is available in the keyboard buffer.
- If a key is waiting, a copy of the key is returned. The buffer is not incremented to the next key.
- If the keyboard buffer contains at least one key, the Zero Flag is cleared.
- If the keyboard buffer does not contain a key, the Zero Flag is set.
- If the keyboard LEDs do not match the current flag settings, this function updates the LEDs.

Input/Output

Input:	AH	=	01h
Output:	AH	=	Scan code or character ID for special character
	AL	=	ASCII code or other translation of character.
	ZF	=	0 Character is ready
			1 No character is available

84-key "filtering"

The read keyboard input function acts as an 84-key keyboard filter, selectively editing the buffer contents for 84-key keyboard compatibility.

When a 101-key keyboard is in place, INT 16h function AH = 01h:

- Returns all standard 84-key keyboard Character Codes as is,
- Adjusts 101-key keyboard duplicates of 84-key keyboard characters so that they take on the same two-byte key code as their 84-key counterparts, and
- Destroys any 101-key Character Codes not compatible with the 84-key keyboard.

Note: The data table labeled "Character Codes: AH = 00h/01h" lists the codes returned when using INT 16h AH = 00h and AH = 01h.

continued

INT 16h: AH = 01h Return Keyboard Status, Continued

Reference

Function AH = 11h Return Extended Keyboard Input does not modify character codes for 84-key keyboard compatibility. For further details, see the Function: AH = 11h heading in this chapter.

INT 16h: AH = 02h Return Shift Flag Status

Description

The Return Shift Flag Status function returns the current shift status from the Keyboard Shift Flags byte (40:17h). The LED status is not updated.

Input/Output

Input: AH = 02h

Output: AL = Current shift status, where:
Bit 7 = 1 Insert active
Bit 6 = 1 Caps Lock active
Bit 5 = 1 Num Lock active
Bit 4 = 1 Scroll Lock active
Bit 3 = 1 Alt pressed
Bit 2 = 1 Ctrl pressed
Bit 1 = 1 Left Shift pressed
Bit 0 = 1 Right Shift pressed

Int 16: AH = 03h Set Typematic Rate and Delay

Description

The Set Typematic Rate and Delay function changes the typematic rate (make code per second) and the delay.

Input/Output

If the parameters are valid, the function transmits a Set Rate command (F3h) to the keyboard controller.

Input:	AH	=	03h
	AL	=	05h
	BH	=	Delay (00h - 03h)
	BL	=	Typematic rate (00h - 1Fh)
Output:	None		

INT 16h: AH = 04h Reserved

Function: AH = 05h Store Key Data

Description

The Store Key Data function stores program-generated data into the keyboard buffer just as if a key were pressed. The buffer pointer (40:1Ch) is adjusted to point to the next available location in the keyboard buffer.

If the keyboard buffer is full, AL is set to indicate a keyboard buffer full error.

Input/Output

Input: AH = 05h
CH = Scan code
CL = ASCII character

Output: AL = 00h No error
= 01h Keyboard buffer full

Note: Keyboard enhancers and other utilities can use this function to interpolate keys into the data stream viewed by applications programs.

INT 16h: AH = 06h – 09h Reserved

INT 16h: AH = 10h Read Extended Keyboard Input

Description

The Read Extended Keyboard Input function reads the next two-byte character code in the keyboard buffer and returns the value in AX. It is designed for use with the 101/102-key keyboard. Unlike function AH = 00h, this function does not modify character codes for 84-key keyboard compatibility.

Input/Output

Input: AH = 10h

Output: AH = Scan code or character ID for special character
AL = ASCII code or other translation of character

Note: The character codes returned by this function are found in the table "Character Codes: AH = 10h/11h" in this chapter.

F0h low byte filter

INT 09h places the value F0h in the low byte character code position for some Alt/character key combinations. The Return Extended Keyboard Input function strips F0h from the keyboard buffer image before returning to the caller. Keys with F0h are returned to AX with their low order byte set to 00h.

All other keys are returned to AX unmodified by this function.

Reference

Function AH = 00h Read Keyboard Input selectively filters keyboard buffer input for compatibility with 84-key keyboards. For details, see the Function: AH = 00h heading in this chapter.

INT 16h: AH = 11h Return Extended Keyboard Status

Description

This function is similar to function AH = 01h, except that it returns unique scan codes for all keys on the 101/102-key keyboard. It executes as follows:

- Checks to see if a character code is available in the keyboard buffer.
- If a key is waiting, a copy of the character code is returned in AX. The buffer is **not** incremented to the next key.
- If the keyboard buffer contains at least one key, the Zero Flag is cleared.
- If the keyboard buffer does not contain a key, the Zero Flag is set.
- If the keyboard LEDs do not match the current flag settings, this function updates the LEDs.

Input/Output

Input:	AH	=	11h
Output:	AH	=	Scan code or character ID for special character (if ZF is set)
	AL	=	ASCII code or other translation of character ID (ZF is set)
	ZF	=	0 No key in buffer
		=	1 Key waiting

F0h low byte filter

INT 09h places the value F0h in the low byte character code position of some Alt/character combinations. The Return Extended Keyboard Status function strips F0h from the keyboard buffer image before returning to the caller. Keys with F0h are returned to AX with their low order byte set to 00h.

All other keys are returned to AX unmodified by this function.

Note: The character codes returned by this function are found in the table "Character Codes: AH = 10h/11h" in this chapter.

Reference

Function AH = 01h Read Keyboard Status selectively filters keyboard buffer input for compatibility with 84-key keyboards. For details, see "Function: AH = 01h" in this chapter.

INT 16h: AH = 12 Return Extended Shift Flags Status

Description

This function is similar to function AH = 02h, except that it returns information on the shift keys provided on the 101/102-key keyboard. It also returns an additional byte of flags in the AH register.

The Return Extended Shift Flags Status function returns the shift flag status from Keyboard Shift Flags Byte (40:17) in the AL register, and returns the shift flag status from Keyboard Shift Flags Byte 2 (40:18) and 101-key Keyboard Status Byte (40:96) in the AH register.

Input/Output

Input: AH = 12h

Output: AH = More keyboard shift flags, where:
Bit 7 = 1 Sys Req pressed (40:18h Bit 2)
Bit 6 = 1 Caps Lock active (40:18h Bit 6)
Bit 5 = 1 Num Lock active (40:18h Bit 5)
Bit 4 = 1 Scroll Lock active (40:18h Bit 4)
Bit 3 = 1 Right Alt active (40:96h Bit 3)
Bit 2 = 1 Right Ctrl active (40:96h Bit 2)
Bit 1 = 1 Left Alt active (40:18h Bit 1)
Bit 0 = 1 Left Ctrl active (40:18h Bit 0)

AL = Keyboard shift flags (copy of Keyboard Shift Flags Byte 40:17), where:
Bit 7 = 1 Insert active
Bit 6 = 1 Caps Lock active
Bit 5 = 1 Num Lock active
Bit 4 = 1 Scroll Lock active
Bit 3 = 1 Alt pressed
Bit 2 = 1 Ctrl pressed
Bit 1 = 1 Left Shift pressed
Bit 0 = 1 Right Shift pressed

INT 16h: AH = 13 - FFh Reserved

How to Read Character Code Tables

Introduction

The tables on the following pages present data related to the IBM PS/2–compatible keyboard.

There are two character–code tables. They are labeled:

- Character Codes: AH = 00h/01h
- Character Codes: AH = 10h/11h

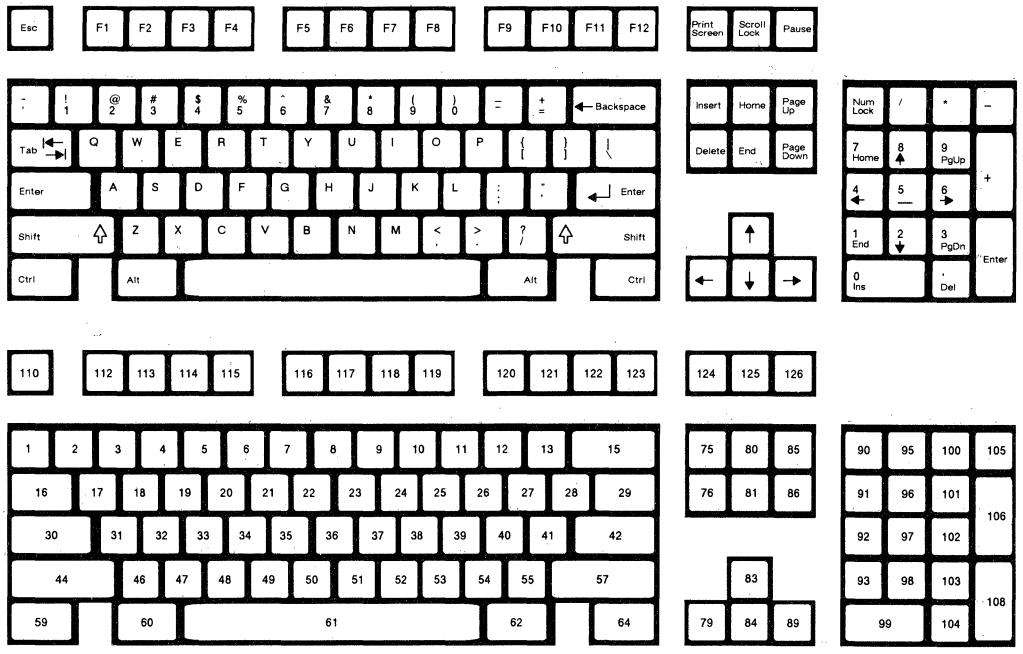
How to read the tables

- The data presented in each table is listed according to key number. These key numbers serve as a legend for identifying the keys. They have no relationship to Keyboard Scan Codes, System Scan Codes, or to Character Codes. See the keyboard layout figure on the next page for a map of these key numbers.
- The tables list Keyboard Scan and System Scan Codes by make code only. Break codes are found by adding a value of 80h to the make code in question.
- The two–byte number associated with each Character Code is listed in four–digit hex notation. The high order byte, which is placed by INT 16h into the AH register, is listed first. The low order byte, which is placed by INT 16h into the AL register, is listed second. The two bytes are separated with a slash. For example, the Character Code for the unshifted "n" key is listed on the table as 31/6E.
- Caps Lock plus any typewriter key yields the same character code as Shift plus any typewriter key.
- Num Lock plus any numeric keypad key yields the same character code as Shift plus any numeric keypad key.
- Alt plus the decimal value for an ASCII character input from the numeric keypad yields the ASCII character. Allowable values are 0 to 255. These instances are noted with "***" in the Alt column.

continued

Keyboard layout

The figure below depicts the key arrangement and key number system applied to the typical 101-key keyboard.



Character Codes: AH = 00h/01h

Introduction

The read keyboard input function acts as an 84-key keyboard filter, selectively editing the buffer contents for 84-key keyboard compatibility.

When a 101-key keyboard is in place, INT 16h subservice AH = 00h:

- Returns all standard 84-key keyboard Character Codes "as is,"
- Adjusts 101-key keyboard duplicates of 84-key keyboard characters so that they take on the same two-byte key code as their 84-key counterparts, and
- Destroys any 101-key Character Codes not compatible with the 84-key keyboard.

Table: AH = 00h/01h

The table below lists the character codes returned by functions AH = 00h and AH = 01h.

Key #	U.S. Keyboard Legend	Make Codes (hex)		Character Codes AH/AL (hex)			
		Keyboard	System	Normal	Shifted	Control	Alt
1	`~	0E	29	29/60	29/7E		
2	1!	16	02	02/31	02/21		78/00
3	2@	1E	03	03/32	03/40	03/00	79/00
4	3#	26	04	04/33	04/23		7A/00
5	4$	25	05	05/34	05/24		7B/00
6	5%	2E	06	06/35	06/25		7C/00
7	6	36	07	07/36	07/5E	07/1E	7D/00
8	7&	3D	08	08/37	08/26		7E/00
9	8*	3E	09	09/38	09/2A		7F/00
10	9(	46	0A	0A/39	0A/28		80/00
11	0)	45	0B	0B/30	0B/29		81/00
12	-_	4E	0C	0C/2D	0C/5F	0C/1F	82/00
13	=+	55	0D	0D/3D	0D/2B		83/00
15	Backspace	66	0E	0E/08	0E/08	0E/7F	
16	Tab	0D	0F	0F/09	0F/00		

continued

Table: AH = 00h/01h, cont'd

Key #	U.S. Keyboard Legend	Make Codes (hex)		Character Codes AH/AL (hex)			
		Keyboard	System	Normal	Shifted	Control	Alt
17	Q	15	10	10/71	10/51	10/11	10/00
18	W	1D	11	11/77	11/57	11/17	11/00
19	E	24	12	12/65	12/45	12/05	12/00
20	R	2D	13	13/72	13/52	13/12	13/00
21	T	2C	14	14/74	14/54	14/14	14/00
22	Y	35	15	15/79	15/59	15/19	15/00
23	U	3C	16	16/75	16/55	16/15	16/00
24	I	43	17	17/69	17/49	17/09	17/00
25	O	44	18	18/6F	18/4F	18/0F	18/00
26	P	4D	19	19/70	19/50	19/10	19/00
27	[{	54	1A	1A/5B	1A/7B	1A/1B	
28	] }	5B	1B	1B/5D	1B/7D	1B/1D	
29	(101-key only)	5D	2B	2B/5C	2B/7C	2B/1C	
30	Caps Lock	58	3A				
31	A	1C	1E	1E/61	1E/41	1E/01	1E/00
32	S	1B	1F	1F/73	1F/53	1F/13	1F/00
33	D	23	20	20/64	20/44	20/04	20/00
34	F	2B	21	21/66	21/46	21/06	21/00
35	G	34	22	22/67	22/47	22/07	22/00
36	H	33	23	23/68	23/48	23/08	23/00
37	J	3B	24	24/6A	24/4A	24/0A	24/00
38	K	42	25	25/6B	25/4B	25/0B	25/00
39	L	4B	26	26/6C	26/4C	26/0C	26/00
40	; :	4C	27	27/3B	27/3A		
41	' "	52	28	28/27	28/22		
42	(102-key only)	5D	2B	2B/5C	2B/7C	2B/1C	
43	Enter	5A	1C	1C/0D	1C/0D	1C/0A	

continued

Table: AH = 00h/01h, cont'd

Key #	U.S. Keyboard Legend	Make Codes (hex)		Character Codes AH/AL (hex)			
		Keyboard	System	Normal	Shifted	Control	Alt
44	L Shift	12	2A				
45	(102-Key only)	61	56				
46	Z	1A	2C	2C/7A	2C/5A	2C/1A	2C/00
47	X	22	2D	2D/78	2D/58	2D/18	2D/00
48	C	21	2E	2E/63	2E/43	2E/03	2E/00
49	V	2A	2F	2F/76	2F/56	2F/16	2F/00
50	B	32	30	30/62	30/42	30/02	30/00
51	N	31	31	31/6E	31/4E	31/0E	31/00
52	M	3A	32	32/6D	32/4D	32/0D	32/00
53	, <	41	33	33/2C	33/3C		
54	. >	49	34	34/2E	34/3E		
55	/ ?	4A	35	35/2F	35/3F		
57	R Shift	59	36				
58	L Ctrl	14	1D				
60	L Alt	11	38				
61	Space	29	39	39/20	39/20	39/20	39/20
62	R Alt	E0-11	E0-38				
64	R Ctrl	E0-14	E0-1D				
75	Insert	E0-70	E0-52	52/00	52/00		
76	Delete	E0-71	E0-53	53/00	53/00		
79	Left	E0-6B	E0-4B	4B/00	4B/00	73/00	
80	Home	E0-6C	E0-47	47/00	47/00	77/00	
81	End	E0-69	E0-4F	4F/00	4F/00	75/00	
83	Up	E0-75	E0-48	48/00	48/00		
84	Down	E0-72	E0-50	50/00	50/00		
85	Page Up	E0-7D	E0-49	49/00	49/00	84/00	
86	Page Down	E0-7A	E0-51	51/00	51/00	76/00	

continued

Table: AH = 00h/01h, cont'd

Key #	U.S. Keyboard Legend	Make Codes (hex)		Character Codes AH/AL (hex)			
		Keyboard	System	Normal	Shifted	Control	Alt
89	Right	E0-74	E0-4D	4D/00	4D/00	74/00	
90	Num Lock	77	45				
91	7 Home	6C	47	47/00	47/37	77/00	**
92	4 Left	6B	4B	4B/00	47/34	73/00	**
93	1 End	69	4F	4F/00	47/31	75/00	**
95	/	E0-4A	E0-35	35/2F	35/2F		
96	8 UP	75	48	48/00	58/38		**
97	5	73	4C	4C/00	4C/35		**
98	2 Down	72	50	50/00	50/32	91/00	**
99	0 Ins	70	52	52/00	52/30		
100	*	7C	37	37/2A	37/2A		
101	9 PgUp	7D	49	49/00	49/39	84/00	**
102	6 Right	74	4D	4D/00	4D/36	74/00	**
103	3 PgDn	7A	51	51/00	51/33	76/00	**
104	. Del	71	53	53/00	53/2E		
105	-	7B	4A	4A/2D	4A/2D		
106	+	79	4E	4E/2B	4E/2B		
108	Enter	E0-5A	E0-1C	0D/1C	0D/1C	0A/1C	
110	Esc	16	01	01/1B	01/1B	01/1B	
112	F1	05	3B	3B/00	54/00	5E/00	68/00
113	F2	06	3C	3C/00	55/00	5F/00	69/00
114	F3	04	3D	3D/00	56/00	60/00	6A/00
115	F4	0C	3E	3E/00	57/00	61/00	6B/00
116	F5	03	3F	3F/00	58/00	62/00	6C/00
117	F6	0B	40	40/00	59/00	63/00	6D/00
118	F7	83	41	41/00	5A/00	64/00	6E/00
119	F8	0A	42	42/00	5B/00	65/00	6F/00

** Alt plus the decimal value for an ASCII character input from the numeric keypad yields the ASCII character. Allowable values are 0 to 255.

continued

Table: AH = 00h/01h, cont'd

Key #	U.S. Keyboard Legend	Make Codes (hex)		Character Codes AH/AL (hex)			
		Keyboard	System	Normal	Shifted	Control	Alt
120	F9	01	43	43/00	5C/00	66/00	70/00
121	F10	09	44	44/00	5D/00	67/00	71/00
122	F11	78	57				
123	F12	07	58				
124	Print Screen	E0-12/ E0-7C	E0-2A/ E0-37 E0-37				
125	Scroll Lock	7E	46				
126	Pause	E1-14/ 77-E1/ F0-14/ F0-77	E1-1D/ 5-E1/ 9D/C5				

Character Codes: AH = 10h/11h

Introduction

INT 09h places the value F0h in the low byte character code position for some Alt/character key combinations. The Return Extended Keyboard Input function strips F0h from the keyboard buffer image before returning to the caller. Keys with F0h are returned to AX with their low-order byte set to 00h.

All other keys are returned to AX unmodified by this function.

Table: AH = 10h/11h

The table below lists the character codes returned by functions AH = 10h and AH = 11h.

Key #	U.S. Keyboard Legend	Make Codes (hex)		Character Codes AH/AL (hex)			
		Keyboard	System	Normal	Shifted	Control	Alt
1	`~	0E	29	29/60	29/7E		
2	1!	16	02	02/31	02/21		78/00
3	2@	1E	03	03/32	03/40	03/00	79/00
4	3#	26	04	04/33	04/23		7A/00
5	4$	25	05	05/34	05/24		7B/00
6	5%	2E	06	06/35	06/25		7C/00
7	6	36	07	07/36	07/5E	07/1E	7D/00
8	7&	3D	08	08/37	08/26		7E/00
9	8*	3E	09	09/38	09/2A		7F/00
10	9(	46	0A	0A/39	0A/28		80/00
11	0)	45	0B	0B/30	0B/29		81/00
12	-_	4E	0C	0C/2D	0C/5F	0C/1F	82/00
13	=+	55	0D	0D/3D	0D/2B		83/00
15	Backspace	66	0E	0E/08	0E/08	0E/7F	
16	Tab	0D	0F	0F/09	0F/00		

continued

Table: AH = 10h/11h, cont'd

Key #	U.S. Keyboard Legend	Make Codes (hex)		Character Codes AH/AL (hex)			
		Keyboard	System	Normal	Shifted	Control	Alt
17	Q	15	10	10/71	10/51	10/11	10/00
18	W	1D	11	11/77	11/57	11/17	11/00
19	E	24	12	12/65	12/45	12/05	12/00
20	R	2D	13	13/72	13/52	13/12	13/00
21	T	2C	14	14/74	14/54	14/14	14/00
22	Y	35	15	15/79	15/59	15/19	15/00
23	U	3C	16	16/75	16/55	16/15	16/00
24	I	43	17	17/69	17/49	17/09	17/00
25	O	44	18	18/6F	18/4F	18/0F	18/00
26	P	4D	19	19/70	19/50	19/10	19/00
27	[{	54	1A	1A/5B	1A/7B	1A/1B	1A/00
28	] }	5B	1B	1B/5D	1B/7D	1B/1D	1B/00
29	(101-key only)	5D	2B	2B/5C	2B/7C	2B/1C	2B/00
30	Caps Lock	58	3A				
31	A	1C	1E	1E/61	1E/41	1E/01	1E/00
32	S	1B	1F	1F/73	1F/53	1F/13	1F/00
33	D	23	20	20/64	20/44	20/04	20/00
34	F	2B	21	21/66	21/46	21/06	21/00
35	G	34	22	22/67	22/47	22/07	22/00
36	H	33	23	23/68	23/48	23/08	23/00
37	J	3B	24	24/6A	24/4A	24/0A	24/00
38	K	42	25	25/6B	25/4B	25/0B	25/00
39	L	4B	26	26/6C	26/4C	26/0C	26/00
40	; :	4C	27	27/3B	27/3A		27/00
41	' "	52	28	28/27	28/22		28/00
42	(102-key only)	5D	2B	2B/5C	2B/7C	2B/1C	
43	Enter	5A	1C	1C/0D	1C/0D	1C/0A	1C/00

continued

Table: AH = 10h/11h, cont'd

Key #	U.S. Keyboard Legend	Make Codes (hex)		Character Codes AH/AL (hex)			
		Keyboard	System	Normal	Shifted	Control	Alt
44	L Shift	12	2A				
45	(102-Key only)	61	56				
46	Z	1A	2C	2C/7A	2C/5A	2C/1A	2C/00
47	X	22	2D	2D/78	2D/58	2D/18	2D/00
48	C	21	2E	2E/63	2E/43	2E/03	2E/00
49	V	2A	2F	2F/76	2F/56	2F/16	2F/00
50	B	32	30	30/62	30/42	30/02	30/00
51	N	31	31	31/6E	31/4E	31/0E	31/00
52	M	3A	32	32/6D	32/4D	32/0D	32/00
53	, <	41	33	33/2C	33/3C		33/00
54	. >	49	34	34/2E	34/3E		34/00
55	/ ?	4A	35	35/2F	35/3F		35/00
57	R Shift	59	36				
58	L Ctrl	14	1D				
60	L Alt	11	38				
61	Space	29	39	39/20	39/20	39/20	39/20
62	R Alt	E0-11	E0-38				
64	R Ctrl	E0-14	E0-1D				
75	Insert	E0-70	E0-52	52/00	52/00	E0/92	A2/00
76	Delete	E0-71	E0-53	53/00	53/00	E0/93	A3/00
79	Left	E0-6B	E0-4B	4B/00	4B/00	73/00	9B/00
80	Home	E0-6C	E0-47	47/00	47/00	77/00	97/00
81	End	E0-69	E0-4F	4F/00	4F/00	75/00	9F/00
83	Up	E0-75	E0-48	48/00	48/00	E0/8D	98/00
84	Down	E0-72	E0-50	50/00	50/00	E0/91	A0/00
85	Page Up	E0-7D	E0-49	49/00	49/00	84/00	99/00
86	Page Down	E0-7A	E0-51	51/00	51/00	76/00	A1/00

continued

Table: AH = 10h/11h, cont'd

Key #	U.S. Keyboard Legend	Make Codes (hex)		Character Codes AH/AL (hex)			
		Keyboard	System	Normal	Shifted	Control	Alt
89	Right	E0-74	E0-4D	4D/00	4D/00	74/00	9D/00
90	Num Lock	77	45				
91	7 Home	6C	47	47/00	47/37	77/00	**
92	4 Left	6B	4B	4B/00	47/34	73/00	**
93	1 End	69	4F	4F/00	47/31	75/00	**
95	/	E0-4A	E0-35	35/2F	35/2F	95/00	A4/00
96	8 UP	75	48	48/00	58/38	8D/00	**
97	5	73	4C	4C/00	4C/35	8F/00	**
98	2 Down	72	50	50/00	50/32	91/00	**
99	0 Ins	70	52	52/00	52/30	92/00	
100	*	7C	37	37/2A	37/2A	96/00	37/00
101	9 PgUp	7D	49	49/00	49/39	84/00	**
102	6 Right	74	4D	4D/00	4D/36	74/00	**
103	3 PgDn	7A	51	51/00	51/33	76/00	**
104	. Del	71	53	53/00	53/2E	93/00	
105	-	7B	4A	4A/2D	4A/2D	8E/00	4A/00
106	+	79	4E	4E/2B	4E/2B	90/00	4E/00
108	Enter	E0-5A	E0-1C	0D/1C	0D/1C	0A/1C	A6/00
110	Esc	16	01	01/1B	01/1B	01/1B	
112	F1	05	3B	3B/00	54/00	5E/00	68/00
113	F2	06	3C	3C/00	55/00	5F/00	69/00
114	F3	04	3D	3D/00	56/00	60/00	6A/00
115	F4	0C	3E	3E/00	57/00	61/00	6B/00
116	F5	03	3F	3F/00	58/00	62/00	6C/00
117	F6	0B	40	40/00	59/00	63/00	6D/00
118	F7	83	41	41/00	5A/00	64/00	6E/00
119	F8	0A	42	42/00	5B/00	65/00	6F/00

** Alt plus the decimal value for an ASCII character input from the numeric keypad yields the ASCII character. Allowable values are 0 to 255.

continued

Table: AH = 10h/11h, cont'd

Key #	U.S. Keyboard Legend	Make Codes (hex)		Character Codes AH/AL (hex)			
		Keyboard	System	Normal	Shifted	Control	Alt
120	F9	01	43	43/00	5C/00	66/00	70/00
121	F10	09	44	44/00	5D/00	67/00	71/00
122	F11	78	57	85/00	57/00	89/00	8B/00
123	F12	07	58	86/00	88/00	8A/00	8C/00
124	Print Screen	E0-12/ E0-7C	E0-2A/ E0-37 E0-37				
125	Scroll Lock	7E	46				
126	Pause	E1-14/ 77-E1/ F0-14/ F0-77	E1-1D/ 45-E1/ 9D/C5				

9

Chapter 9
INT 10h Video Service

Overview

Description

The CBIOS Video Service provides I/O support for both color and monochrome analog video monitors and supports selection of display mode, cursor addressing, text display, scrolling, and writing pixels.

A significant advantage of the CBIOS Video Service is that most routines can be called by the user program regardless of the video mode being used (monochrome, CGA, EGA, or VGA). The CBIOS Video Service offers all the features and functions of the VGA BIOS. The video services themselves find out what the current type of display is and take care of whatever address translation is needed, including getting bit patterns from an in-memory table font if the user's program asks for text display in graphics mode.

How the services are invoked

INT 10h invokes the CBIOS Video Service.

The INT 10h vector resides at Interrupt Table Address 00:40h. To ensure compatibility, the CBIOS initializes the INT 10h Video Service vector to absolute address FF065h.

continued

How the services are invoked, cont'd

Individual Video Service functions are selected via the AH register. If the number in AH is outside the legal range, no action will be taken. Subfunctions are selected via the AL register or the BL register. The following general rules apply to these services:

- The character or pixel value that is to be written is normally passed in register AL.
- BX, CX, DX and segment registers are preserved through all service calls. Assume that all other register contents, particularly SI and DI, may be destroyed.
- The x coordinate (column number) is passed in CX (graphics functions) or DL (text functions).
- Any display page value is passed in BH. Display pages are zero-based (i.e., page 0 = first page, page 1 = second page, etc.).

continued

Summary of video service functions

The table below summarizes the Video Service function and subfunctions.

AH	Description
00h	Set Video Mode
01h	Set Cursor Type
02h	Set Cursor Position
03h	Read Current Cursor Position
04h	Read Light Pen Position
05h	Select New Video Page
06h	Scroll Current Page Up
07h	Scroll Current Page Down
08h	Read Character/Attribute to Screen
09h	Write Character/Attribute from Screen
0Ah	Write Character Only to Screen
0Bh	Set Color Palette
0Ch	Write Pixel
0Dh	Read Pixel
0Eh	Write Teletype to Active Page
0Fh	Return Video Status
10h	Set Palette Registers AL = 00h Set single palette * AL = 01h Set overscan register * AL = 02h Set all palette registers and overscan AL = 03h Toggle intensity/blinking bit AL = 04h–06h Reserved * AL = 07h Read individual palette register AL = 08h Read overscan register AL = 09h Read all palette registers and overscan register AL = 10h Set individual color register AL = 11h Reserved AL = 12h Set block of color registers * AL = 13h Select color paging mode (not valid for mode 13h) BL = 00h Select paging mode BL = 01h Select page * AL = 14h Reserved AL = 15h Read single DAC color register AL = 16h Reserved
* Not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware.	

continued

Summary of video service functions, cont'd

AH	Description
10h (cont'd)	Set Palette Registers, continued AL = 17h Read block of color registers AL = 18h–19h Reserved * AL = 1Ah Read color paging status AL = 1Bh Sum color values to gray shades
11h	Character Generator AL = 00h Load user text mode font AL = 01h Load ROM 8x14 text mode font AL = 02h Load ROM 8x8 double dot font AL = 03h Set block specifier (text modes only) AL = 04h Load ROM 8x16 text mode font * AL = 10h Load user text mode load (after mode set) * AL = 11h Load ROM 8x14 text mode font (after mode set) * AL = 12h Load ROM 8x8 double dot text mode font (after mode set) AL = 14h Load ROM 8x16 text mode font (after mode set) AL = 20h Set user graphics characters pointer at INT 1Fh (8x8 font) AL = 21h Set user graphics characters pointer at INT 43h * AL = 22h Load ROM 8x14 font for graphics AL = 23h Load ROM 8x8 double dot font for graphics AL = 24h Load ROM 8x16 font for graphics AL = 30h Get font pointer information
12h	Alternate Select * BL = 10h Return VGA information BL = 20h Switch to alternate print screen routine * BL = 30h Select scan lines for text modes BL = 31h Enable/disable default palette loading during set mode BL = 32h Enable/disable video BL = 33h Enable/disable summing to gray shades * BL = 34h Enable/disable cursor scaling BL = 35h Switch display * BL = 36h Video screen off/on
13h	Write String AL = 00h Cursor not moved AL = 01h Cursor is moved AL = 02h Cursor not moved (text modes only) AL = 03h Cursor is moved (text modes only)
14h–19h	Reserved
1Ah	Read/Write Display Combination Code AL = 00h Read display combination code AL = 01h Write display combination code
* Not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware.	

continued

Summary of video service functions, cont'd

AH	Description
1Bh	Return Functionality/State Information
1Ch	Save/Restore Video State* * AL = 00h Return Save/Restore buffer size needed * AL = 01h Save current video state * AL = 02h Restore current video state
1Dh-FFh	Reserved
* Not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware.	

In this chapter

This chapter focuses on the Video Service. The following topics are presented:

- Hardware Environment
- Video I/O Ports
- Video Modes
- Mode/Monitor Support
- ROM Character Generators
- System RAM Data
- Video Service Functions

Hardware Environment

Introduction

The CBIOS Video Service assumes the following hardware environment:

- IBM VGA-compatible VLSI chip or chip set
- INMOS G171 or compatible digital-to-analog converter (DAC) chip
- 256K of dynamic read/write video RAM configured as four 64K maps
- Monochrome or color direct drive analog monitor

VGA-compatible chip (or chip set)

The VGA chip (or chip set) provides all CRT control signals. It consists of four components: CRT Controller, Sequencer, Graphics Controller, and Attribute Controller.

In general, the VGA-compatible chip provides the interfacing between the system microprocessor and video memory, provides all of the CRT control signals, and outputs 8-bit digital data to the Digital-to-Analog converter for display.

Each of these components contains a fixed number of control registers that are accessed via 8-bit I/O ports. The CBIOS exchanges information with control registers by writing the register number to the appropriate I/O port and by subsequently reading from or writing to the specified register.

continued

Table of VGA component functions

The function of each VGA component is summarized in the table below.

Component	Function CRT Controller
CRT Controller	Generates horizontal and vertical CRT sync timings, cursor and underline timings, video buffer addressing, and refresh addressing.
Sequencer	Arbitrates system access to display RAM and fonts. The sequencer allows up to eight fonts with two fonts display at any one time.
Graphics Controller	Handles read/write operation on four parallel bit planes. Outputs data to Attribute Controller.
Attribute Controller	Converts incoming text mode attribute data or graphics mode pixel data into 8-bit indexes into the Digital-to-Analog Converter (DAC) color registers (see below).

Digital-to-Analog Converter (DAC)

The video DAC contains 256 individual color registers which can be accessed by the CBIOS as either four 64-color registers or sixteen 16-color registers.

Each DAC color register contains one 18-bit RGB analog value. Six bits of each register are allocated to each primary color. Thus, the color represented in each DAC color register may be any of 256K possible colors (i.e. 2^{3*6} = 256K).

continued

DAC, Attribute controller, BIOS initialization

The CBIOS initializes the 8-bit index values contained in the Attribute Controller and the 18-bit analog color values contained in the DAC color registers each time a video mode is set.

The table below outlines the DAC/Attribute Controller relationship for the three major types of video modes.

Mode interrupt/initialization table

Type	Description
EGA and 16-color VGA	**DAC** The CBIOS initializes access to the 256 DAC color registers in blocks, or pages, of 64 colors each. The first 64-color page is initialized to analog equivalents of the 64 EGA colors. The other 192 colors remain undefined. **Attribute Controller** To maintain color compatibility with existing EGA software, CBIOS initializes the Attribute Controller to the 16 commonly accepted default EGA colors.
CGA	**DAC** The CBIOS initializes access to the 256 DAC color registers as four blocks of 64 colors each in CGA modes. The first 64-color block is initialized as four 16-color sub-blocks. Each of these 16-color blocks is initialized to the 16 colors displayable in CGA mode. The remaining 192 DAC registers are undefined. **Attribute Controller** To maintain color compatibility with existing CGA software, the CBIOS initializes the Attribute Controller to the 16-color CGA palette. When applications software requests a CGA color, the Attribute Controller maps the requests through to its analog equivalent in the DAC controller.
13h	**DAC** The CBIOS initializes all 256 color registers to their IBM PS/2 default equivalents. **Attribute Controller** The attribute controller receives 8-bit color values directly from applications programs. It maps those values directly into the DAC color registers.

Note: For more information on the CBIOS Video Service video modes, refer to the Video Modes page heading in this chapter.

continued

Video RAM

The CBIOS Video Service requires 256K of read/write video RAM formatted into four banks (or maps) of 64K.

To maintain compatibility, display memory for each of the historical MDA, CGA, and EGA modes is exactly as it was before. The display memory organization for the new VGA modes is outlined in this chapter in the table found under the Video Modes heading.

Analog monitor support

To display all modes, the Video Service requires either a monochrome or a color direct drive analog monitor with a 31.5 KHz horizontal scan frequency.

The display's vertical gain is adjusted automatically by the VGA-compatible circuitry. Thus, video modes with 350, 400, and 480 horizontal scan lines can be displayed without requiring manual adjustment.

Video I/O Ports

INT 10h, Video Service, references the following I/O ports:

I/O Address	Read/Write Status	Description
0061h	W	System control port B, where: Bit 7 = 1 Reset timer 0 output latch (IRQ 0) Bits 6-4 = Reserved Bit 3 = 0 Enable channel check Bit 2 = 0 Enable parity check Bit 1 = 1 Speaker data enable Bit 0 = 1 Enable timer 2 gate to speaker
0064h	W	8042 Commands
0064h	R	8042 Status, where: Bit 7 = 1 Parity error Bit 6 = 1 General time out Bit 5 = 1 Auxiliary output buffer full Bit 4 = 1 Inhibit switch Bit 3 = 1 Command/data Bit 2 = System flag Bit 1 = 1 Input buffer full Bit 0 = 1 Output buffer full
03B4h	R/W	VGA CRT controller index register (mono)
03B5h	R/W	Other VGA CRT controller registers (mono)
03BAh	R	VGA input status register 1 (mono)
03BAh	W	VGA feature control register (mono)
03C0h	R/W	VGA attribute address register
03C0h	W	Other VGA attribute registers
03C1h	R	Other VGA attribute registers
03C2h	W	VGA miscellaneous output register
03C2h	R	VGA input status register 0
03C3h	R/W	VGA video subsystem enable
03C4h	R/W	VGA sequencer address register
03C5h	R/W	Other VGA sequencer registers
03C6h	R/W	Video DAC PEL mask
03C7h	W	Video DAC PEL address, read mode
03C7h	R	Video DAC state register
03C8h	R/W	Video DAC PEL address, write mode
03C9h	R/W	Video DAC PEL data register
03CAh	R	VGA feature control register

continued

Video I/O Ports, Continued

I/O Address	Read/Write Status	Description
03CCh	R	VGA miscellaneous output register
03CEh	R/W	VGA graphics registers, address register
03CFh	R/W	VGA and other graphics registers
03D4h	R/W	VGA CRT controller index register (color)
03D5h	R/W	Other VGA CRT controller registers (color)
03DAh	R	VGA input status register 1 (color)
03DAh	W	VGA feature control register (color)

Video Modes

Introduction

The CBIOS Video Service supports 17 video modes, providing backward compatibility with MDA, CGA, and EGA modes, as well as compatibility with all new VGA modes.

Models 25 and 30: IBM PS/2 Models 25 and 30 which utilize multicolor graphics array (MCGA) video subsystems support six video modes, which are outlined in a separate table.

Table of CBIOS video modes

The table below lists the 17 video service modes supported by the CBIOS Video Service.

Mode	Emul.	Res.	Type	Max. Colors	Scheme	Char. Box	Max. Pgs.	Buff. Start
0, 1	CGA*	320x200	Text	16/256K	40x25	8x8	8	B8000
0, 1	EGA*	320x350	Text	16/256K	40x25	8x14	8	B8000
0, 1	VGA+	360x400	Text	16/256K	40x25	9x16	8	B8000
2, 3	CGA*	640x200	Text	16/256K	80x25	8x8	8	B8000
2, 3	EGA*	640x350	Text	16/256K	80x25	8x14	8	B8000
2, 3!	VGA+	720x400	Text	16/256K	80x25	9x16	8	B8000
4, 5	CGA	320x200	Graphics	4/256K	40x25	8x8	1	B8000
6	CGA	640x200	Graphics	2/256K	80x25	8x8	1	B8000
7	MDA*	720x350	Text	MDA Mono	80x25	9x14	8	B0000
7!	VGA*	720x400	Text	VGA Mono	80x25	9x16	8	B0000
D	EGA	320x200	Graphics	16/256K	40x25	8x8	8	A0000
E	EGA	640x200	Graphics	16/256K	80x25	8x8	4	A0000
F	EGA	640x350	Graphics	Mono	80x25	8x14	2	A0000
10	EGA	640x350	Graphics	16/256K	80x25	8x14	2	A0000
11	VGA	640x480	Graphics	2/256K	80x30	8x16	1	A0000
12	VGA	640x480	Graphics	16/256K	80x30	8x16	1	A0000
13	VGA	320x200	Graphics	256/256K	40x25	8x8	1	A0000

Legend: "!" indicates power-on default mode
3! = color monitor is attached,
7! = monochrome monitor is attached.

"*" indicates that scan lines must be specified before mode set. (See AH = 12h BL = 30h Select Scan Line for Alphanumeric Codes for details.)

"+" indicates default mode

continued

Model 25 and 30 video modes

The table below lists the video modes supported by IBM Models 25 and 30 with MCGA.

Mode	Emul.	Res.	Type	Max. Colors	Scheme	Char. Box	Max. Pgs.	Buff. Start
0, 1	CGA†	320x400	Text	16/256K	40x25	8x16	8	B8000
2, 3	EGA†	640x400	Text	16/256K	80x25	8x16	8	B8000
4, 5	CGA	320x200	Graphics	4/256K	40x25	8x8	1	B8000
6	CGA	640x200	Graphics	2/256K	80x25	8x8	1	B0000
7	MDA	720x350	Text	MDA Mono	80x25	9x14	8	B0000
11	VGA	640x480	Graphics	2/256K	80x30	8x16	1	A0000
13	VGA	320x200	Graphics	256/256K	40x25	8x8	1	A0000

Legend: "†" indicates that 320x200 or 640x200 is used if hardware initialization detects a nonstandard monitor type.

continued

Video mode facts

The facts below apply to the Video Service modes listed on the previous page.

Fact	Description
Modes 0, 2, 4 = Modes 1, 3, 5	In CGA mode, modes hex 1, 3, and 5 have color burst turned on and modes hex 0, 2, and 4 have color burst turned off. The VGA hardware does not support color burst; modes hex 0, 2, and 4 are identical to modes hex 1, 3, and 5 respectively.
200-line modes double-scanned	All 200 scan line modes are double-scanned by the analog monitor. Each line of video is painted on the screen twice, one beneath the other, before the next new scan line is painted.
Number of text mode lines may vary	200, 350, or 400 scan lines can be displayed in all text modes. The number of scan lines to display is selected via Video Service function AH = 12h, BL = 30h Alternate Select. Refer to AH = 12h in this chapter for further details.
No cursor in graphics	The cursor is not displayed in graphics modes.
MDA, CGA, EGA Emulation	To insure compatibility with older software, the CBIOS Video Service supports all 200 and 350 scan line MDA, CGA, and EGA video modes. To activate a 200 or 350 scan line mode: ▪ Execute function AH = 12h, BL = 30h Alternate Scan Line Select with AL = 00h or 01h. (i.e. 200 or 350 scan lines) ▪ Execute function AH = 00h Set Mode **Note:** Unless otherwise indicated, the BIOS defaults to 400 scan lines for all text modes.

Color/Monochrome analog monitors display all modes

Both monochrome and color 31.5 KHz direct drive analog and multiscan monitors can display all of the modes listed in the table of video modes earlier in this chapter.

The following facts apply:

If...	Then...
A monochrome analog monitor is attached...	Colors are displayed as shades of gray, with the maximum number of shades equal to the maximum number of colors display in the respective mode. In mode 0Eh, for example, 16 shades of gray can be displayed. Mode 13h is an exception to this rule. Only 64 (and not 256) shades of gray can be displayed in mode 13h.
A color analog monitor is attached...	The colors displayed are selected from a BIOS-initialized palette of 64 color registers. The number of colors displayed in each mode is listed in the table of video modes above. In mode 0Eh, for example, 16 different colors can be displayed. The BIOS initializes the 64 colors in this palette to analog equivalents of the 64-color digital EGA palette. The remaining 192 color registers are undefined. Mode 13h is an exception to this rule. Mode 13h is capable of displaying all of the 256 colors stored in the DAC. The BIOS initializes these colors to IBM-compatible defaults at mode set. **Models 25 and 30:** In MCGA-based systems, the BIOS initializes the first 16 color registers, leaving 240 color registers undefined.
A 200 scan line mode is selected...	All 200 scan line modes are double-scanned. That is, each horizontal line is scanned twice, for a total of 400 scan lines. This is true for both color and monochrome analog monitors.
Power-on default video modes are in effect ...	The default power-on mode is mode 3+ (720x400, 16-color) when a color analog monitor is attached, or The default power-on mode is mode 7+ (720x400, monochrome) when a monochrome analog monitor is attached. **Models 25 and 30:** In MCGA-based systems, the default power-on mode is 3.

ROM-resident fonts

The ROM BIOS contains three BIOS-resident character generators (fonts):

- 8x8 dot
- 8x14 dot (not available in IBM Models 25 and 30 with MCGA)
- 8x16 dot

9x14 and 9x16 dot fonts

The BIOS is also capable of generating a 9x16 dot font in text modes 0+,1+, 2+, 3+, and 7+. However, the BIOS generates this 9x16 font by running the 8x16 character set through a ROM-resident table of corrections. The 9x16 font is not available to BIOS end users through the AH = 11h Character Generator function.

The 9x14 dot font is created in the same way that the 9x16 dot font is, except that it is based on the 8x14 character set. The 9x14 dot font is used in mode 7.

Text mode fonts and memory map 2

In text modes, the ASCII character number is written to video memory map 0 and the corresponding attribute is written to memory map 1.

Memory map 2 is used for storing character generator (font) data. Each font occupies one 8K block, making for a total of eight possible fonts. A maximum of two 256-character fonts (i.e. 512 characters) can be displayed at one time. The Video Service Set Block Specifier subfunction provides BIOS controls for selecting which two blocks are to be displayed. (See AH = 11h, AL = 03h in this chapter for details.)

continued

Default fonts

The font associated with a given video mode is directly related to the resolution that mode is capable of reproducing. The table below outlines the font/resolution relationship.

If the resolution is....	Then the default font is...
320x200	8x8
320x350	8x14
360x400	9x16
640x200	8x8
640x350	8x14
640x400	8x14
640x480	8x16
720x350	9x14
720x400	9x16

How fonts are loaded

Each time a video mode is set, the CBIOS automatically loads the font associated with the specified mode into block 0 in memory map 2.

Note: For 200 and 350 scan line text modes, the user must set the number of scan lines through the Video Service subfunction AH = 12h, BL = 30h, Select Scan Lines for Text Modes, before mode set.

Overriding default fonts and default memory blocks

The Video Service functions AH = 11h Character Generator and AH 12h Alternate Select contain subfunctions that enable the user to:

- Load a default font into a nondefault character block in memory map 2
- Specify a nondefault font for display in text or graphics modes
- Load user-defined font for display in text or graphics modes.
- Control the number of scan lines in text modes

Refer to the Character Generator and Alternate Select headings in this chapter for further details.

Video data definitions

The data definitions used by the INT 10h Video services are stored in system RAM in segment 40h and are presented below in offset order.

System RAM Offset (hex)	Size	Description
49h	1 Byte	Video mode setting.
4Ah	1 Word	Number of columns on screen.
4Ch	1 Word	Current page size.
4Eh	1 Word	Current page address.
50h	8 Words	Cursor position on each page. Two bytes/page. First byte (low order) of each pair is column, second is row. 0, 0 is upper left corner of screen.
60h	1 Word	Cursor type defined as 6845 video chip-compatible starting and ending scan lines. High order byte holds starting scan line; low order byte holds ending scan line.
62h	1 Byte	Current page number.
63h	1 Word	6845 video chip-compatible I/O port number for current mode. (Port 03D4h or 03B4h)
65h	1 Byte	Current mode select register.
66h	1 Byte	Current palette value.
84h	1 Byte	Number of rows on screen (24/25)
85h	1 Word	Character height (bytes/character)
87h	1 Byte	Video control bits, where: Bit 7 = Clear RAM Bit 6,5 = Memory on video hardware as follows: 00b = 64K 01b = 128K 10b = 192K 11b = 256K Bit 4 = Not used Bit 3 = 0 EGA-compatible mode active Bit 2 = 1 Wait for display enable Bit 1 = 0 color or ECD monitor attached to EGA-compatible adapter = 1 Monochrome monitor is attached to EGA-compatible adapter Bit 0 = 0 translate cursor video modes 0-3 when using ECD monitor in 350-line mode

continued

Video data definitions, cont'd

System RAM Offset (hex)	Size	Description
88h	1 Byte	EGA/VGA switch data where: Bits 7-4 = Feature connector bits 3-0, respectively Bits 3-0 = Option switches 3-0, respectively
89h	1 Word	VGA control bits, where: Bit 7 = 200 lines Bits 6-5 = Reserved Bit 4 = 400 lines Bit 3 = No palette load Bit 2 = Mono Monitor Bit 1 = Gray Scaling Bit 0 = Reserved
8Ah	1 Byte	Index to the Display Combination Code table
A8h	1 Word	Pointer to video parameter table and overrides (in segment:offset format).

Video Service Pointer Tables

The Video Service initializes and maintains two system RAM pointer tables. Each table consists of seven double-word segment:offset pointers to video parameters and optional auxiliary character generator information.

The Set Mode function, AH = 00h, refers to the video service pointer tables (and to the information pointed to by the tables) on every mode set.

Default table locations

There are two video service pointer tables: the primary table and the secondary table. The locations of each of these tables is initialized by the CBIOS during the power-on self test and initialization (POST) procedure. The table locations are initialized as follows:

- This memory location (40:A8h in the BIOS Data Area) points to the first pointer table. It is initialized by the CBIOS to a default table in ROM.
- The fifth entry in the first pointer table points to the location of the second pointer table.

Note: Users who wish to override the CBIOS default locations may supply their own values for 40:A8h and entry 5 of Table #1. See next page.

continued

Structure of Video Service Pointer Table #1

The location of the first, or primary, pointer table must be stored in 40:A8h. This address is initialized by the CBIOS to a default location.

The table is structured as follows:

<table>
<tr><th>Table Location</th><th>Description</th></tr>
<tr><td>Pointer 1</td><td>POINTER TO VIDEO PARAMETER TABLE
Initialized at POST to the ROM BIOS video parameter table. This table entry is mandatory. If reinitialized, it must point to a valid video parameter table.</td></tr>
<tr><td>Pointer 2</td><td>POINTER TO DYNAMIC SAVE AREA
Optional table entry. Initialized at POST to 00:00h. When non-zero, this entry must point to a Dynamic Save Area in system RAM. The Dynamic Save Area must be at least 256 bytes long. It contains the 16 EGA palette register values as well as the overscan register value in bytes 0–16, respectively.</td></tr>
<tr><td>Pointer 3</td><td>POINTER TO TEXT MODE AUXILIARY CHARACTER GENERATOR
Optional table entry. Initialized by the BIOS to 00:00h. When nonzero, this entry must point to a table structured as below:
<table>
<tr><th>Size</th><th>Description</th></tr>
<tr><td>Byte</td><td>Bytes per character</td></tr>
<tr><td>Byte</td><td>Block to load, 0 = Normal operation</td></tr>
<tr><td>Word</td><td>Count to store, 256 = Normal operation</td></tr>
<tr><td>Word</td><td>Character offset, 0 = Normal operation</td></tr>
<tr><td>Dword</td><td>Pointer to a font table</td></tr>
<tr><td>Byte</td><td>Displayable rows, where:
0FFh = Maximum calculated value otherwise indicated number of rows displayed.</td></tr>
<tr><td>Byte</td><td>Consecutive bytes of mode values for font.
0FFh = Indicates end of stream</td></tr>
</table>
Note: Using this third pointer can cause unexpected cursor operation. See "Function: AH = 01h Set Cursor Type," in this chapter for details.</td></tr>
</table>

continued

Structure of Video Service Pointer Table #1, cont'd

<table>
<tr><th>Table Location</th><th>Description</th></tr>
<tr><td>Pointer 4</td><td>POINTER TO GRAPHICS MODE CHARACTER GENERATOR
Optional table entry. Initialized to 00:00h. When nonzero, this entry must point to a table structured as below:
<table>
<tr><th>Size</th><th>Description</th></tr>
<tr><td>Byte</td><td>Displayable rows</td></tr>
<tr><td>Word</td><td>Bytes per character</td></tr>
<tr><td>DWord</td><td>Pointer to font table</td></tr>
<tr><td>Byte</td><td>Consecutive bytes of mode values for font.
0FFh = Indicates end of stream</td></tr>
</table></td></tr>
<tr><td>Pointer 5</td><td>POINTER TO VIDEO SERVICE POINTER TABLE #2 (Secondary Save Pointer Data Area)
Initialized by the BIOS to the default Pointer Table #2 location. This table entry is mandatory. If reinitialized, it must point to a valid Pointer Table #2.</td></tr>
<tr><td>Pointer 6</td><td>Reserved. Set by the BIOS to 00:00h</td></tr>
<tr><td>Pointer 7</td><td>Reserved. Set by the BIOS to 00:00h</td></tr>
</table>

continued

Structure of Video Service Pointer Table #2

The location of second, or secondary, pointer table must be stored in Pointer 5 of pointer table #1. Pointer 5 of table #1 is initialized by the BIOS to a default location.

The table is structured as follows:

<table>
<tr><th>Table Location</th><th>Description</th></tr>
<tr><td>Pointer 1</td><td>TABLE LENGTH
Initialized by the BIOS to default table length.</td></tr>
<tr><td>Pointer 2</td><td>POINTER TO DISPLAY COMBINATION CODE (DCC)
Initialized by the CBIOS to the ROM BIOS DCC table. This table entry is mandatory. If reinitialized, it must point to a valid DCC table. The DCC table is structured as below:
<table>
<tr><th>Size</th><th>Description</th></tr>
<tr><td>Byte</td><td>Number of entries in the DCC table</td></tr>
<tr><td>Byte</td><td>DCC Table Version Number</td></tr>
<tr><td>Byte</td><td>Maximum display type code</td></tr>
<tr><td>Byte</td><td>Reserved</td></tr>
</table>
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
01,08 Entry 12 CVGA + MDPA
00,07 Entry 13 MVGA
02,07 Entry 14 MVGA + CGA
02,06 Entry 15 MVGA + PGC

Abbreviations
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</td></tr>
</table>

continued

Structure of Video Service Pointer Table #2, cont'd

<table>
<tr><th>Table Location</th><th>Description</th></tr>
<tr><td>Pointer 3</td><td>POINTER TO SECOND TEXT MODE AUXILIARY CHARACTER GENERATOR
Optional table entry. Initialized by the BIOS to 00:00h When nonzero, this entry must point to a table structured as below:
<table>
<tr><th>Size</th><th>Description</th></tr>
<tr><td>Byte</td><td>Bytes per character</td></tr>
<tr><td>Byte</td><td>Block to load, 0 = Normal operation</td></tr>
<tr><td>Word</td><td>Count to store, 256 = Normal operation</td></tr>
<tr><td>Word</td><td>Character offset, 0 = Normal operation</td></tr>
<tr><td>Dword</td><td>Pointer to a font table</td></tr>
<tr><td>Byte</td><td>Displayable rows, where:
0FFh = Maximum calculated value; otherwise indicated number of rows displayed.</td></tr>
<tr><td>Byte</td><td>Consecutive bytes of mode values for font.
0FFh = Indicates end of stream</td></tr>
</table>
Note: Bit 3 of the Attribute byte is used to switch between text mode fonts. It may be necessary to employ the user palette profile table to define a palette of colors independent of attribute bit 3.</td></tr>
</table>

continued

Structure of Video Service Pointer Table #2, cont'd

<table>
<tr><th>Table Location</th><th>Description</th></tr>
<tr><td>Pointer 4</td><td>Pointer to User Palette Profile Table

Optional table entry. Initialized by the BIOS to 00:00h. When nonzero, this entry must point to a table structured as below:

<table>
<tr><th>Size</th><th>Description</th></tr>
<tr><td>Byte</td><td>Underlining flag, where:
1 = On
0 = Ignore (Normal operation)
-1 = Off</td></tr>
<tr><td>Byte</td><td>Reserved</td></tr>
<tr><td>Word</td><td>Reserved</td></tr>
<tr><td>Word</td><td>Internal palette count (0-17), where:
17 = Normal operation</td></tr>
<tr><td>Word</td><td>Internal palette index (0-16), where:
0 = Normal operation</td></tr>
<tr><td>DWord</td><td>Pointer to internal palette</td></tr>
<tr><td>Word</td><td>External palette count (0-256), where:
0 = Normal operation</td></tr>
<tr><td>Word</td><td>External palette index (0-255), where:
0 = Normal operation</td></tr>
<tr><td>DWord</td><td>Pointer to external palette</td></tr>
<tr><td>Byte</td><td>Consecutive bytes of mode values for font, where:
0FFh = Indicates end of stream</td></tr>
</table>
</td></tr>
<tr><td>Pointer 5</td><td>Reserved</td></tr>
<tr><td>Pointer 7</td><td>Reserved</td></tr>
</table>

Function: AH = 00h Set Video Mode

Description

The Set Video Mode function sets the video mode registers for operation in any supported mode. It selects the active video mode if more than one is installed, clears the screen, positions the cursor at 0,0 and resets the color palette to default color values. (See "Video Modes" in this chapter for a description of each mode.)

Models 25 and 30: IBM PS/2 Models 25/30 (and compatibles) equipped with MCGA graphics do not support the full number of VGA video modes. See "Video Modes" for more information.

Input/Output

Input: AH = 00h
AL = Video mode, where:
00h–07h — Valid values
08h–0Ch — Reserved
0Dh–13h — Valid values

Output: AL = Video mode, where:
20h — If mode is greater than 7
30h — If mode is from 0–5 or 7
3Fh — If mode is mode 6

Additional information

- Resetting the same video mode can be used to clear the screen.
- To avoid resetting the palette when working with colors, use Video Service 06h rather than Service 00h to clear the screen.
- The cursor is not displayed in graphics modes.
- Modes 0, 2, and 5 are identical to modes 1, 3, and 4.
- The power-on default mode with a color analog monitor attached is 3.
- The power-on default mode with a monochrome analog monitor attached is 7.
- During mode set, if bit 7 of AL is set, the video buffer is not cleared.
- Refer to AH = 12h BL = 30h to select alpha mode scan lines (200, 350, or 400).
- For all modes except mode 13, the first 64 color registers are initialized and the values in the remaining 192 color registers are undefined.

Function: AH = 01h Set Cursor Type

Description

The Set Cursor Type function sets the size of the cursor that will appear in text modes. Cursor size and location within the character box is determined by the starting and ending scan lines indicated in bits 4:0 of registers CH and CL respectively.

The Set Cursor Type function stores cursor size parameters in the cursor type byte at 40:60h.

Models 25 and 30: In IBM PS/2 Models 25 and 30 (and compatibles) equipped with MCGA graphics, this function multiplies CH by 2, then it multiplies CL by 2 and increases the CL product by 1 before writing cursor values to MCGA hardware.

Input/Output

Input: AH = 01h
CH = Top scan line, where:
Bits 7–6 = 00h (must be 00h, otherwise cursor blinking becomes erratic)
Bit 5 = To shut cursor off
Bits 4–0 = Top scan line
CL = Bottom scan line, where:
Bit 7 = Undefined
Bits 6–5 = Show cursor
Bits 4–0 = Lower scan line

Output: None

continued

Additional information

- Only one cursor type is available for all video pages.
- The default text mode cursor occupies the bottom two scan lines of the character box associated with the given text mode.
- Setting register bit 5 of the CH register causes the text mode cursor to disappear completely. The cursor can also be eliminated by positioning it to a nondisplay address, e.g. (x,y) = (0,25)
- In graphics mode, bit 5 is set automatically, which prevents the cursor from being displayed. To simulate the cursor, use the solid block character DFh, or change background attributes.
- CGA mode can a display cursor on eight lines, numbered 0 to 7, top to bottom.
- MDA and EGA modes can display a cursor on 14 lines, numbered 0 to 13, top to bottom.
- MCGA and VGA modes can display a cursor on 16 lines, the default character size of this mode.

Default cursor settings

The default cursor settings for each mode type are listed below. Default cursor settings are established by function AH = 00h at mode set.

Mode type	Cursor settings
CGA	CH = 6, CL = 7
MDA, EGA	CH = 11, CL = 12
MCGA, VGA	CH = 13, CL = 14

Function: AH = 02h Set Cursor Position

Description

The Set Cursor Position function sets the cursor position (in terms of row-by-column coordinates) for the display page indicated in BL. It saves the position as a two-byte row-by-column table entry in the cursor coordinates byte at 40:50h. Row and column coordinates are indicated in registers DH and DL respectively.

The Set Cursor Position function applies to both text and graphics video modes. In text modes, if the display page selected in BH is the active display page, the screen cursor will move to the coordinates indicated in registers DH and DL. In graphics modes, the cursor is invisible, but it is used to define a position on the screen.

Input/Output

Input:	AH	=	02h
	BH	=	Display page number (see function AH = 05h)
	DH	=	Row (0 is top row of screen)
	DL	=	Column (0 is leftmost column)
Output:	AX	=	00h

Cursor Positioning

Positioning the cursor to coordinates (0,0) places it in the upper left corner of the screen in the 80x25 text mode. Selecting coordinates (79,24) in the 80x25 text mode will allow placement of the cursor in the lower right corner of the screen. The cursor can also be placed in the lower right corner of the screen using coordinates (39,24) in the 40x25 text mode.

The cursor can be turned off by moving it to an off-screen location or changing its coordinates to a position such as (0,25). However, if it is moved too far off-screen, the actual position if the cursor may become unpredictable.

Additional information

- The display page number must be set to 0 in CGA graphics mode.
- Multiple display pages in 16-color graphics as well as text modes are supported by both EGA and VGA.

Function: AH = 03h Read Current Cursor Position

Description

The Read Current Cursor Position function reads the cursor position for the given video page from the cursor coordinates byte at 40:50h. It reads the cursor type from 40:60h and returns the current cursor position in text coordinates. This function is useful for determining the exact cursor type before it is changed.

Input: AH = 03h
BH = Display page number (zero-based)

Output: AX = 00h
CH = Starting cursor scan line
CL = Ending cursor scan line
DH = Row number
DL = Column number

A different cursor is maintained for each display page. Each of these cursors can be looked at independently with this function, whatever page is currently active. How many display pages are available is defined by the display mode selected.

Function: AH = 04h Read Light Pen Position

Description

This function reads the light pen's status and position. PS/2 systems (including Models 25 and 30) do not support light pen input devices. They are not effective on monochrome monitors with long image-retention phosphor, and the light pen position is not accurate enough for use with high resolution devices.

Input: AH = 04h

Output: AH = 00h Light pen is not supported
AL = 00h

All registers except AX are altered on return.

Function: AH = 05h Select New Video Page

Description

The Select New Video Page function sets the active page for the video mode selected. Refer to the table under the Video Modes heading for a list of the maximum number of pages allowed for each video mode.

Input/Output

Input: AH = 05h
AL = New page number (zero-based)

Output: None

Additional information

- All page numbers are zero-based (i.e. page numbers begin at page 0).
- In text modes, page numbers range from 0-7.
- Page 0, located at the beginning of display memory, is used by default in all video modes.
- Switching between pages does not alter their contents. Also, no matter which page is active (currently displayed), text can be written to any video page using INT 10h functions 02h, 09h and 10h.
- In the CGA mode, no video paging is possible.
- In EGA and VGA modes, video pages are available up to the limits of video RAM in both alphanumeric and graphics modes.
- The current cursor position is maintained by the CBIOS for as many as eight video pages.
- An instantaneous screen change can be created by building a screen on an undisplayed page, then using this function to display it.

Function: AH = 06h Scroll Current Page Up
Function: AH = 07h Scroll Current Page Down

Description

The Scroll Current Page Up/Down functions employ the CH, DH, CL, and DL registers to define a screen window and allow the contents of the window to be scrolled either up or down by the number of character rows specified in register AL. If AL = 00h, the window is blanked instead of scrolled.

Information appearing on the screen but lying outside the parameters of the defined window remains on the screen. Only one window can be defined at any given time.

These two functions operate in both text and graphics modes.

Input/Output

Input:

AH	=	06h Scroll current page up 07h Scroll current page down
AL	=	Scroll distance in character rows (0 blanks entire scroll area)
BH	=	Attribute to use on blanked lines
CH	=	Top row of scroll window
CL	=	Left column of scroll window
DH	=	Bottom row of scroll window
DL	=	Right column of scroll window

Output: None

Use the cursor positioning and character writing services to fill the blank line with text.

Initializing a Window

Setting the subfunction register AL = 0 initializes a window on the display screen. Setting the AL register to zero blanks out the region specified by the CX and DX general purpose registers and fills them with the attribute in the BH register.

Function: AH = 08h Read Character/Attribute from Screen

Description

The Read Character/Attribute from Screen function reads the character at the current cursor location. For text modes, the attribute is also returned. In graphics modes, the character matrix at the cursor position is compared to the bit patterns in the current graphics character definition table to determine the character's ASCII value. Characters other than standard ASCII characters are returned as AL = 00h.

Input/Output

Input:	AH	=	08h
	BH	=	Display page (refer to "Video Modes" for maximum pages per mode)
Output:	AH	=	Attribute (text modes only)
	AL	=	Character read

Additional information

- To read a character from any valid display page (other than the active one), specify the display page number.
- Information about the screen is in screen memory and need not be stored in a program.
- This function can be used to read the screen for TSR spelling and thesaurus utilities.

Function: AH = 09h Write Character/Attribute to Screen

Description

The Write Character/Attribute function writes the character to the screen starting at the current cursor location for as many times as indicated in the CX register. The cursor is not moved even if more than one character is written, unless the same character is repeated.

Note: INT 10h functions AH = 09h and AH = 0Ah are similar. AH = 09h should be used for all graphics modes.

Input/Output

Input:

AH	=	09h
AL	=	ASCII character to write
BH	=	Display page (refer to "Video Modes" for maximum pages per mode)
	=	Background color (graphics mode 13h only)
BL	=	Character attribute (text modes)
	=	Foreground color (graphics modes) (i.e. color of character)
CX	=	Repeat count

Output: None

continued

Additional information

- In text modes, the number of repeats placed into CX may exceed the number of columns remaining in a given row. Characters will wrap around from row to row. In graphics modes, the number of repeats placed into CX cannot exceed the number of columns remaining in a given row. Characters will not wrap around from row to row.
- In graphics mode 13h, the color specified in BH determines the screen background color.
- In CGA graphics mode, the bit map used for ASCII characters 80–FFh is stored in a table that starts at 0:7Ch. This value is stored in the vector for INT 1Fh. By resetting the vector, the bit map table location can be changed to point to a different bit map.
- For EGA and VGA graphics modes, use the table whose pointer is stored in the vector for INT 43h. For further information refer to the INT 10h, function AH = 11h AL = 20 and AL = 21 headings in this chapter.
- Any value of AL will produce a display; this includes all control characters (e.g. bell, backspace, CR, LF). These control characters are not interpreted as special characters and do not change the cursor position.
- After a character has been written, the cursor has to be explicitly moved to the next position using INT 10h function 02h.
- To write a character without changing the attribute at the current cursor position, use INT 10h function 0Ah.
- When this function is used to write characters in graphics mode, and with bit 7 of BL set to 1, the character is XORed with the contents of the current display. This feature can be used to write characters and then erase them.

Function: AH = 0Ah Write Character Only to Screen

Description

The Write Character Only to Screen function operates identically to the Write Character/Attribute function, except that for text modes the attribute bytes corresponding to the characters remain unchanged. Use this function to write a character to the screen in text modes.

Note: Use AH = 09h in graphics modes.

Input/Output

Input:

AH	=	0Ah
AL	=	Character to write (ASCII codes)
BH	=	Display page (text modes only)
CX	=	Repeat count

Output: None

Additional information

- Any value of AL will produce a display; this includes all control characters (e.g. bell, backspace, CR, LF). These control characters are not interpreted as special characters and do not change the cursor position.
- After a character has been written, the cursor has to be explicitly moved to the next position using INT 10h function 02h.
- To write a character without changing the attribute at the current cursor position, use INT 10h function 0Ah.
- When this function is used to write characters in graphics mode, and with bit 7 of BL set to 1, the character is XORed with the contents of the current display. This feature can be used to write characters and then erase them.

Function: AH = 0Bh Set Color Palette

Description

This function selects colors for medium resolution graphics modes. Depending on the value placed in BH, the Set Color Palette function will perform any of four operations.

If BH = 00h, then the value in BL sets the:

- Background color for 320x200 graphics modes (modes 4, 5)
- Border color for 320x200 text modes (modes 0, 1, 2, 3)
- Foreground color for 640x200 graphics mode (mode 6)

If BH = 01h, then the value in BL sets the palette for 320x200 graphics modes (modes 4, 5).

Input/Output

Input: AH = 0Bh
BH = 00h
= if mode = 4, 5, set background color to value in BL
= If mode = 0, 1, 2, 3, set border color to value in BL
= If mode = 6, set foreground color (640x200 graphics) to value in BL
BL = (0–31), when colors (16–31 are high intensity background set)

or

BH = 01h select palette for modes 4, 5 (320x200 graphics)
BL = 00h; Palette = Green (1), Red (2), Brown (3)
= 01h; Palette = Cyan (1), Magenta (2), White (3)

Output: None

Additional information

- In CGA graphics modes, bit 4 of BL selects between normal and high intensity. EGA and VGA graphics modes emulate this by selecting a palette of high intensity colors when bit 4 of BL is set.
- A flashing display can be set by rapidly changing the palette.
- The background color of text is determined by the high order four bits of the attribute byte of each character.

Function: AH = 0Ch Write Pixel

Description

The Write Pixel function writes to video memory the pixel specified by row and column number in DX and CX. When a given video mode allows more than one page, the (0-based) page number must be indicated in BH.

For all graphics modes except mode 13h, bit 7 of AL acts as an inverter flag. If bit 7 of AH is set, then the color value in AL is exclusively ORed (XOR) with the current pixel. That pixel can be erased by writing it a second time.

Refer to "Video Modes" for a table listing video modes, resolutions, and maximum pages/mode.

Input/Output

Input:

AH	=	0Ch
AL	=	Color (Bit 7 is exclusive OR flag)
BH	=	Page number (Modes allowing more than one page)
DX	=	Pixel row number
CX	=	Pixel column number

Output: None

Pixel values

- In four-color graphics modes (display modes 04h and 05h), pixel values range from 0-3.
- In two-color graphics modes (display mode 06h), pixel values range from 0-1.

Function: AH = 0Dh Read Pixel

Description

The Read Pixel function returns the value of an addressed pixel to the low order bits of the AL register. This function can be used for collision detection in video games. It can also be used by advanced graphics programs to detect boundaries when moving a graphics object on the screen.

Refer to "Video Modes" for a table listing video modes, resolutions, and maximum pages/mode.

Input/Output

Input:	AH	=	0Dh
	BH	=	Page number (modes allowing more than one page)
	CX	=	Column number
	DX	=	Row number
Output:	AL	=	Color value of pixel read

Pixel values

Display modes 04h, 05h and 06h have valid pixel values in the ranges of 0–3, 0–3 and 0–1 respectively. Refer to "Video Modes" for a table listing video modes, resolutions, and maximum pages/mode.

Function: AH = 0Eh Write Teletype to Active Page

Description

The Write Teletype function makes the display appear as a serial terminal. Display characters are written to regeneration memory to the active page and the cursor is moved to the next character location (scrolling is necessary). Screen width is a function of the video mode currently in effect.

Input/Output

Input: AH = 0Eh
AL = Character to write
BL = Foreground color (graphics modes only)

Output: None

Special characters

The four ASCII characters listed below are not displayed but are interpreted instead as control characters. All other characters (including other control characters) are interpreted as display characters.

Character	ASCII Code	Function
Bell	07h	A beep is sounded.
Backspace	08h	If the cursor is already on column 0, nothing happens. Otherwise, the cursor moves back one column.
LF	0Ah	The cursor moves down one row. If done on the last row of the screen, the display is scrolled one row.
CR	0Dh	The cursor moves to column 0 on the current row.

Additional information

- When working in the active page, this function allows a character to be printed at the current cursor position. After printing the character, the function then moves the cursor to the right one space. If by chance the cursor is moved past the end of the line, the function will wrap the cursor to the next line. Scrolling the screen up one line requires that the cursor be moved past the lower right corner of the screen.

continued

Additional information, cont'd

- In CGA text mode, characters can be written to any legal display page, no matter which page is active.
- Compare this function with INT 10h, function 13h.
- This is the best function for simple output.

DOS uses

DOS uses this function in the console driver for putting operating system text and messages on the screen.

This function does not, however, allow the attribute of a text character to be selected. A way of defining an attribute for a character written to the screen may be defined by

1. writing in the ASCII blank character (20h) having the desired attribute to the current position using function 09h, and
2. writing the desired character using function 0Eh.

With this method, the user does not have to provide for line wrapping and screen scrolling but can turn these over to the BIOS.

Function: AH = 0Fh Return Video Status

Description

The Return Video Status function returns current display mode information. It gives the mode, screen width in characters, and the display page number.

Refer to "Video Modes" in this chapter for maximum pages per mode.

Input/Output

Input: AH = 0Fh

Output: AH = Number of columns on screen from the screen width byte (40:4Ah)
AL = Current mode from the video mode setting byte (40:49h)
BH = Active display page number from the display page byte (40:62h)

Examples of function's use

Some ways to use this function are

- for finding out the screen width in the current screen mode before clearing the screen.
- to determine the settings of the display system at program initialization so that they can be returned to when the program terminates.
- when writing TSR utilities that pop up on the screen while another application is running. The background application may be running in a different mode than the TSR.

Function: AH = 10h Set Palette/Color Registers

Description

The Set Palette/Color Register consists of 14 subfunctions that control operations on the color palette registers within EGA/VGA video controllers. These subfunctions allow control of color, blinking and video DAC.

The 14 subfunctions can be divided into two groups:

- Subfunctions that service the Attribute Controller's Internal Palette
- Subfunctions that service the DAC Color Registers.

Subfunctions

AL Value	Subfunction Name	Programs
00h	Set single palette register	Atrib. Contrl.
01h*	Set overscan register	Atrib. Contrl.
02h*	Set all palette registers and overscan	Atrib. Contrl.
03h*	Toggle intensify/blinking bit	Atrib. Contrl.
04h-06h	Reserved	Atrib. Contrl.
07h*	Read individual palette register	Atrib. Contrl.
08h	Read overscan register	Atrib. Contrl.
09h	Read all palette registers and overscan register	Atrib. Contrl.
10h	Set individual color register	DAC
11h	Reserved	DAC
12h	Set block of color registers	DAC
13h*	Select color paging mode (not valid for mode 13h) BL = 00h Select paging mode BL = 01h Select page	DAC
14h	Reserved	DAC
15h	Read single DAC color register	DAC
16h	Reserved	DAC
17h	Read block of color registers	DAC
18h-19h	Reserved	DAC
1Ah	Read color paging status	DAC
1Bh	Sum color values to gray shades	DAC
* Not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware.		

continued

Reference

For more information on the relationship between the DAC and the Attribute Controller, refer to the Hardware Environment heading in this chapter.

Error Handling

The CBIOS Video Service checks for valid parameters upon entry into each subfunction. If invalid parameters are detected the subfunction returns with all registers preserved.

Subfunction: AL = 00h Set single palette

This subfunction sets a single color value in the Attribute Controller's 16-value internal palette. Each palette register value points to one location among the 256 DAC color registers. This subfunction exits without processing if the current mode is 13h.

Models 25 and 30: In IBM PS/2 Models 25 and 30 (and compatibles) equipped with MCGA graphics, input BX = 0712h. This will define a color register set that results in eight consistent colors.

Input:
AH = 10h
AL = 00h set single palette
BL = Palette register
BH = New color value

Output: None

Subfunction: AL = 01h Set overscan register

This subfunction sets the overscan color (screen border) for the current video modes.

Input:
AH = 10h
AL = 01h
BH = Color value to set

Output: None

continued

Subfunction: AL = 02h Set all palette registers and overscan (border color)

This subfunction sets all 16 Attribute Controller internal palette registers, as well as the overscan register (11h), to the values pointed to by the 17-byte table by ES:BX. It exits without processing if the current mode is 13h.

Models 25 and 30: In IBM PS/2 Models 25 and 30 (and compatibles) with MCGA graphics, this subfunction is not supported.

Input: AH = 10h
AL = 02h
ES:DX = Pointer to 17-byte table, where:
Byte 16 = Overscan value (border color)
Bytes 15-0 = Palette values

Output: None

Subfunction: AL = 03h Toggle intensity/blink bit

This subfunction sets either the background intensity or the foreground blinking for CGA, EGA, and VGA modes.

Input: AH = 10h
AL = 03h
BL = 00h Enable background intensity
= 01h Enable foreground blinking

Output: None

Subfunctions: AL = 04h to 06h

These subfunctions are reserved.

---

continued

Subfunction: AL = 07h Read individual palette register

This subfunction reads the color value of the register input into BL and returns that value in BH.

Models 25 and 30: In IBM PS/2 Models 25 and 30 (and compatibles) with MCGA graphics, this subfunction is not supported.

Input: AH = 10h
AL = 07h
BL = Palette register to be read (Range 0–15)

Output: BH = Value read

Subfunction: AH = 08h Read overscan register (border color)

This subfunction reads the color value stored in the overscan register of the current color palette. This value defines the current border color.

Input: AH = 10h
AL = 08h

Output: BH = Value read

Subfunction: AH = 09h Read all palette registers and overscan register (border color)

This subfunction reads the contents of the current palette register and the overscan register and outputs the contents to a table pointed to by ES:DX. The table pointed to by ES:DX must be 17 bytes long.

Input: AH = 10h
AL = 09h

ES:DX: Pointer to 17-byte buffer for return values

Output: ES:DX = Pointer to 17-byte table destination, where:
Byte 16 = Overscan value (border color)
Bytes 15–0 = Register color values

---

continued

Subfunction: AL = 10h Set individual color register

This subfunction reads in a table of RGB color values from the area pointed to by ES:DX. The number of the first DAC color register to set is specified in BX. The total number of color registers to set is specified in CX. Each red, green, and blue entry must be one byte long.

Bits 7-6 are "don't care" bits. Bits 5-0 are set as 3Fh (most intense) to 00h (off).

Input:
AH = 10h
AL = 10h
BX = Color register to set
DH = Red value to set, where:
Bits 7-6 = Reserved
Bits 5-0 = 3Fh Most intense; 00h = Off (black)
CH = Green value to set, where:
Bits 7-6 = Reserved
Bits 5-0 = 3Fh Most intense; 00h = Off (black)
CL = Blue value to set, where:
Bits 7-6 = Reserved
Bits 5-0 = 3Fh Most intense; 00h = Off (black)

Output: None

With gray scale summing selected, the weighted sum gray shade value is calculated and saved to each of the three RGB parts of the color register. See INT 10h, function 10h, subfunction 1Bh for a description of this calculation.

Subfunction: AL = 11h

This subfunction is reserved.

---

continued

Subfunction: AL = 12h Set block of color registers

This subfunction sets a consecutive series of DAC color registers. Input the number of the first color register to set in BX, the number of color registers to set in CX, and the pointer to the table of color values in ES:DX.

Input:
AH = 10h
AL = 12h
BX = First color register to set
CX = Number of color registers to set
ES:DX = Pointer to table of color values. Table format is red, green, blue, red, green, blue

Output: None

With gray scale summing selected, the weighted sum gray shade value is calculated and saved to each of the three RGB parts of the color register. See INT 10h, function 10h, subfunction 1Bh for a description of this calculation.

continued

Subfunction: AL = 13h Select color paging mode

The DAC chip contains 256 color registers. In all modes except mode 13h (i.e., 256-color graphics), the DAC color registers can be logically divided into four blocks, each containing 64 color registers, or into 16 blocks, each containing 16 color registers.

Depending on the value placed into BL, this subfunction performs two operations:

- When BL = 00h, the value placed in BH selects whether the 256 DAC color registers will be logically divided into four blocks of 64 color registers or into 16 blocks of 16 color registers.
- When BL = 01h, the value placed in BL selects an individual block of color registers. When in four-block mode, the allowable range for BH is 00h–03h. When in 16-block mode, the allowable range for BH is 00h–0Fh.

Models 25 and 30: In IBM PS/2 Models 25 and 30 (and compatibles) with MCGA graphics, this subfunction is not supported.

Input:

AH	=	10h	
AL	=	13h	
BL	=	00h	Set block mode
BH	=	00h	Set 4 blocks of 64 registers
	=	01h	Set 16 blocks of 16 registers
or			
BL	=	01h	Select individual block
BH	=	00h–03h	for 4-block mode
		00h–0Fh	for 16-block mode

Output: None

The Set Mode functions (AH = 10h, AL = 00h) defaults to the 64 register/block mode, initializing only the first block of 64 color registers. Alternate blocks of color registers are initialized through the select paging mode subfunction (AH = 10h, AL = 13h, BL = 00h). Alternate blocks of color registers must be initialized for the "select page" subfunction (AH = 10h, AL = 13h, BL = 01h) to operate properly.

continued

Subfunction: AL = 14h

This subfunction is reserved.

Subfunction: AL = 15h Read single DAC color register

This subfunction reads an individual DAC color register.

Input: AH = 10h
AL = 15H
BX = Color register to read

Output: CH = Green value
CL = Blue value
DH = Red value

Subfunction: AL = 16h

This subfunction is reserved.

Subfunction: AL = 17h Read block of color registers

This subfunction reads a block of DAC color registers to the buffer area pointed to by ES:DX.

Input: AH = 10h
AL = 17h
BX = Which DAC color register begins block
CX = Total number of color registers to read
ES:DX = Pointer to buffer to save register information in. (3 bytes per color register). Format as red byte, green byte, blue byte...)

Output: ES:DX = Pointer to values read

Subfunctions: AL = 18h to 19h

These subfunctions are reserved.

---

continued

Subfunction: AL = 1Ah Read color paging status

This subfunction returns the current color paging mode as well as the current color page within this mode. See INT 10h, function 10h, subfunction 13h, Select color paging mode.

Models 25 and 30: In IBM PS/2 Models 25 and 30 (and compatibles) with MCGA graphics, this subfunction is not supported.

Input: AH = 10h
AL = 1Ah

Output: BH = Current page
BL = Current paging mode
= 00h (4 pages of 64 registers)
= 01h (16 pages of 16 registers)

Subfunction: AL = 1Bh Sum color values to gray shades

This subfunction reads the red, green, and blue values stored in the specified color registers and performs the following weighted sum:

Gray Shade = 30% red + 59% green + 11% blue

The resulting red, green, and blue values are written to the specified color registers. The original contents of each register are not retained.

Input: AH = 10h
AL = 1Bh
BX = Color register to start with
CX = Count of how many to sum

Function: AH = 11h Character Generator

Description

The Character Generator function consists of 14 subfunctions, all of which, in one way or another, permit the loading and/or enabling of text mode and graphics mode character generators (fonts).

For more information on text and graphics modes character generators, turn to the ROM Character Generators heading near the beginning of this chapter.

Character generator subfunctions

The Character Generator subfunctions are invoked by passing the proper parameter in the AL register. The subfunctions are listed in the table below:

AL Value	Subfunction Name
00h	Load user text mode font
01h	Load ROM 8x14 text mode font
02h	Load ROM 8x8 text mode font
03h	Set block specifier (text modes only)
04h	Load ROM 8x16 text mode font
10h*	Load user text mode font (after mode set)
11h*	Load ROM 8x14 text mode font (after mode set)
12h*	Load 8x8 double dot text mode font (after mode set)
14h*	Load ROM 8x16 text mode font (after mode set)
20h*	Set user graphics character pointer at INT 1Fh (8x8 font)
21h	Set user graphics character pointer at INT 43h
22h	Use ROM 8x14 font at INT 43h for graphics
23h	Use ROM 8x8 font at INT 43h for graphics
24h	Use ROM 8x16 font at INT 43h for graphics
30h	Get font pointer information
* Not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware.	

continued

Subfunction: AL = 00h Load user text mode font

This subfunction loads a user-defined font into the memory map 2 block specified in BL. The user font must be stored in a table pointed to by ES:BP, with the ASCII character ID or the first character specified in DX.This function is for fonts in text display mode. For fonts in graphics mode, see INT 10h, function 11h, subfunctions 20h-24h.

Input:

AH	=	11h
AL	=	00h
BH	=	Number of bytes per character
BL	=	Block to load (valid values are 0-7)
CX	=	Number of characters to store
DX	=	Character ID of first character in ES:BP table
ES:BP	=	Pointer to the user table

Output: None

Scan lines per character, number of character rows, buffer length, and cursor size are not recalculated by this subfunction. The user-defined font loaded here must occupy the same character box size as the mode the font will be displayed under.

continued

Subfunction: AL = 01h Load ROM 8x14 text mode font

This subfunction loads the ROM-resident 8x14 font into the memory map 2 block indicated in BL. Use this subfunction to override the CBIOS default character block location for 8x14 character video modes. For fonts in graphics mode, see INT 10h, function 11h, subfunctions 20h-24h.

Models 25 and 30: This function is not available on IBM PS/2 Models 25 and 30 with MCGA. If called, AL = 04h is executed.

Input: AH = 11h
AL = 01h
BL = Block to load (valid values are 0-7)

Output: None

Subfunction: AL = 02h Load 8x8 double dot text mode font

This subfunction loads the ROM-resident 8x8 font into the character block indicated in BL. Use this subfunction to override the CBIOS default character block location for 8x8 character video modes. For fonts in graphics mode, see INT 10h, function 11h, subfunctions 20h-24h.

Input: AH = 11h
AL = 02h
BL = Block to load (valid values are 0-7)

Output: None

---

continued

Subfunction: AL = 03h Set block specifier (text modes only)

VGA-compatible hardware allows the user to display up to 512 text mode characters at a time. However, in text modes, all character font bit patterns are stored in 8K blocks in video memory map 2. Each 8K block can store up to 256 characters. In order to display 512 text mode characters, then, special steps need to be taken to allow end user software to select between either of two possible 8K character blocks.

This subfunction sets up the video hardware so that bit 3 of each text mode attribute byte can select between either of two memory map 2 character blocks, thus allowing up to 512 characters to be displayed in any text mode. Characters are selected according to the rules below:

- The value of bits 4, 1, 0 of BL set by this subfunction indicates the first character block; the value of bits 5, 3, and 2 in BL indicates the second character block.
- The value of bit 3 of each character attribute byte determines from which character block a given ASCII character will be selected.
  - When bit 3 = 0, the character is selected from the block defined by bits 4, 1, and 0 that input into BL.
  - When bit 3 = 1, the character is selected from the block defined by bits 5, 3, and 2 of BL.
- If bits 4, 1, 0 = bits 5, 3, 2, then bit 3 of the character attribute byte toggles foreground intensity on or off. When two character blocks are selected, function call AX = 1000h BX = 0712h will ensure that eight consistent color planes are set.

Note: To make a loaded character block active, this subfunction must be run after any load character block subfunction.

Input: AH = 11h
AL = 03h
BL = Select character block (see description above)

Output: None

continued

Subfunction: AL = 03h Set block specifier (text modes only), cont'd

Models 25 and 30 — On IBM PS/2 Models 25 and 30 (and compatibles) equipped with MCGA hardware, the following rules apply:

- The values of Bits 1 and 0 in BL indicate the first character block; the values of bits 3 and 2 indicate the second character block.
- The value of bit 3 of each character attribute byte determines from which character block a given ASCII character will be selected. When bit 3 = 0, the character is selected from the block defined by BL bits 1 and 0. When bit 3 = 1, the character is selected from the block defined by BL bits 3 and 2.
- If BL bits 1 and 0 = bits 3 and 2, then bit 3 of the attribute byte turns foreground intensity on and off.

Subfunction: AL = 04h Load 8x16 ROM text mode font

This subfunction loads the ROM-resident 8x16 font into the character block indicated in BL. Use this subfunction to override the CBIOS default character block for 8x16 character block modes. For fonts in graphics mode, see INT 10h, function 11h, subfunctions 20h–24h.

Input: AH = 11h
AL = 04h
BL = Block to load (valid values are 0–7)

Output: None

---

continued

Description: Subfunctions: AL = 10h, 11h, 12h, 14h

Subfunctions AL = 10h, 11h, 12h, and 14h, respectively, are identical to subfunctions AL = 00h, 01h, 02h, and 04h except for the following:

- Page 0 is active
- Bytes per character (points) are recalculated
- Number of rows is recalculated
- Display buffer length is calculated
- CRTC registers below are reprogrammed:
  - Index 09h Maximum scan line index
  - 0Ah Cursor start index
  - 0Bh Cursor end index
  - 12h Vertical displacement end index
  - 14h Underline location (mode 07h only)

Models 25 and 30: These subfunctions are not available on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware. If called, AL = 10h executes AL = 00h; AL = 11h executes AL = 01h; AL = 12h executes AL = 02h, and AL = 14h executes AL = 04h.

Notes:

- The registers programmed by subfunctions 10h, 11h, 12h, and 14h are a subset of all VGA CRTC registers. Because of this, the font downloaded by each of these subfunctions must be relatively close in size to the default font associated with the current video mode. To prevent unpredictable results, these subfunctions should only be called after a mode set call (i.e., INT 10h AH = 00h Set Mode).
- These functions are for fonts in text display mode. For fonts in graphics mode, see INT 10h, function 11h, subfunctions 20h–24h.

continued

Subfunction: AL = 10h Load user text mode font (after mode set)

This subfunction loads the user character set defined in the table pointed to by ES:BP to the character block indicated BL.

Models 25 and 30: This subfunction is not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware. If called, subfunction AL = 10h executes AL = 00h.

Input:

AH	=	11h
AL	=	10h
BH	=	Number of bytes per character
BL	=	Block to load (valid values are 0–7)
CX	=	Number of characters to store
DX	=	Character ID of first character in ES:BP table
ES:BP	=	Pointer to table

Output: None

The registers programmed by subfunction 10h are a subset of all VGA CRTC registers. Because of this, the font downloaded by this subfunction must be relatively close in size to the default font associated with the current video mode. To prevent unpredictable results, these subfunctions should only be called after a mode set call (i.e., INT 10h AH = Set Mode).

continued

Subfunction: AL = 11h Load ROM 8x14 text mode font (after mode set)

This subfunction loads the ROM 8x14 font into the character block indicated in BL. Bytes/characters (point size), number of display rows, cursor and underline parameter, and display buffer length are recalculated. Use this subfunction to override the CBIOS-defined font for the text mode in question.

Models 25 and 30: This subfunction is not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware. If called, subfunction AL = 11h executes AL = 01h.

Input: AH = 11h
AL = 11h
BL = Block to load (valid values are 0-7)

Output: None

The registers programmed by subfunction 11h are a subset of all VGA CRTC registers. Because of this, the font downloaded by this subfunction must be relatively close in size to the default font associated with the current video mode. To prevent unpredictable results, these subfunctions should only be called after a mode set call (i.e., INT 10h AH = Set Mode).

continued

Subfunction: AL = 12h Load ROM 8x8 double dot text mode font

This subfunction loads the ROM 8x8 font into the character block indicated in BL. Bytes/character (point size), number of display rows, cursor and underline parameters, and display buffer length are recalculated. Use this subfunction to override the BIOS-defined font for the text mode in question.

Models 25 and 30: This subfunction is not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware. If called, subfunction AL = 12h executes AL = 02h.

Input: AH = 11h
AL = 12h
BL = Block to load

Output: None

The registers programmed by subfunction 12h are a subset of all VGA CRTC registers. Because of this, the font downloaded by this subfunction must be relatively close in size to the default font associated with the current video mode. To prevent unpredictable results, these subfunctions should only be called after a mode set call (i.e., INT 10h AH = Set Mode).

continued

Subfunction: AL = 14h load ROM 8x16 text mode font

This subfunction loads the ROM 8x16 font into the character block indicated in BL. Bytes/character (point size), number of display rows, cursor and underline parameters, and display buffer length are recalculated. Use this subfunction to override the BIOS-defined font for the text mode in question.

Models 25 and 30: This subfunction is not supported on IBM PS/2 Models 25 and 30 and compatibles with MCGA hardware. If called, subfunction AL = 14h executes AL = 04h.

Input: AH = 11h
AL = 14h
BL = Block to load

Output: None

The registers programmed by subfunction 14h are a subset of all VGA CRTC registers. Because of this, the font downloaded by this subfunction must be relatively close in size to the default font associated with the current video mode. To prevent unpredictable results, these subfunctions should only be called after a mode set call (i.e., INT 10h AH = Set Mode).

continued

Description: Subfunctions: AL = 20h, 21h, 22h, 23h, 24h

Subfunctions AL = 20h, 21h, 22h, 23h, and 24h all deal with fonts that are used in graphics modes. For functions that deal with fonts in text display mode, see INT 10h, function 11h, subfunctions 00h–14h.

To prevent unpredictable results, subfunctions AL = 20h, 21h, 22h, 23h, and 24h should be called only immediately after a mode set (INT 10h, AH = 00h, AL = mode to select).

Subfunction: AL = 20h Set user graphics font pointer at INT 1Fh (8x8 font)

This subfunction sets the graphics font pointer to the vector contained in INT 1Fh. The INT 1Fh vector is specified in ES:BP. The font table pointed to is used by applications programs running in graphics modes 04h–06h (CGA only) for character codes 80h–FFh.

Input: AH = 11h
AL = 20h
ES:BP = User graphics font pointer
(INT 1Fh is set to the vector contained in ES:BP)

Output: None

Notes:

- The character generators for all graphics modes are contained within the CBIOS. The ROM BIOS–based character generators for graphics modes 0Dh to 13h contain all 256 ASCII characters.
- The character generators for graphics modes 04h, 05h, and 06h, (CGA–compatible graphics modes), contain only the first 128 ASCII characters. The remaining 128 ASCII characters (extended ASCII characters 80h–FFh) may be supplied by the user. All user–supplied character sets must be vectored at a memory location pointed to by INT 1Fh. In general, these user–supplied character sets are built into an 8x8 character box.
- The DOS command GRAFTABL loads a graphic table which allows modes 04h, 05h, and 06h to display extended ASCII characters. Vectoring INT 1Fh to point at the DOS GRAFTABL graphics table is the usual method of providing user–supplied extended ASCII characters. For more information, refer to the MS–DOS technical reference manual.

---

continued

Subfunction: AL = 21h Set user graphics font pointer at INT 43h (EGA/VGA only)

This subfunction sets the graphics font pointer to the vector contained in INT 43h. To avoid unpredictable results, this function should be called only after a mode set (INT 10h, AH = 00h Set Video Mode). The INT 43h vector is specified in ES:BP.

Input:

AH = 11h
AL = 21h
BL = Rows on screen specifier
= 00h if user-supplied
01h = 14 (0Eh) rows
02h = 25 (19h) rows
03h = 43 (2Bh) rows
CX = Points (bytes/character)
DL = Rows per screen (if BL = 00h)
ES:BP = Pointer to user table

Output: None

continued

Subfunction: AL = 22h Use ROM 8x14 font for graphics

This subfunction specifies the use of the ROM-based 8x14 font in EGA and VGA graphics modes. Use it to display the 8x14 font in non-8x14 font graphics modes. To avoid unpredictable results, this function should be called only after a mode set (INT 10h, AH = 00h Set Video Mode). This Subfunction should be used in conjunction with subfunction AH = 11h, AL = 21h, Set User Graphics Font Pointer at INT 43h.

Models 25 and 30: This subfunction is not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware. If called, AL = 22h executes subfunction AL = 24h.

Input:
AH = 11h
AL = 22h
BL = Row specifier
= 00h if user-supplied
01h = 14 (0Eh) rows
02h = 25 (19h) rows
03h = 43 (2Bh) rows
DL = Rows per screen (if BL = 00h):

Output: None

continued

Subfunction: AL = 23h Use ROM 8x8 double dot font for graphics

This subfunction specifies the use of the ROM-based 8x8 font in EGA and VGA graphics modes. Use it to display the 8x8 font in non-8x8 font graphics modes. To avoid unpredictable results, this function should be called only after a mode set (INT 10h, AH = 00h Set Video Mode). This subfunction should be used in conjunction with subfunction AH = 11h, AL = 21h, Set User Graphics Font Pointer at INT 43h.

Input: AH = 11h
AL = 23h
BL = Row specifier
= 00h if user-supplied
01h = 14 (0Eh) rows
02h = 25 (19h) rows
03h = 43 (2Bh) rows
DL = Rows per screen (if BL = 00h)

Output: None

continued

Subfunction: AL = 24h Use ROM 8x16 font for graphics

This subfunction specifies the use of the ROM-based 8x16 font in EGA and VGA graphics modes. Use it to override to display the 8x16 font in non-8x16 font graphics modes. To avoid unpredictable results, this function should be called only after a mode set (INT 10h, AH = 00h Set Video Mode). This subfunction should be used in conjunction with subfunction AH = 11h, AL = 21h, Set User Graphics Font Pointer at INT 43h.

Input: AH = 11h
AL = 24h
BL = Row specifier
= 00h if user-supplied
01h = 14 (0Eh) rows
02h = 25 (19h) rows
03h = 43 (2Bh) rows
DL = Rows per screen (if BL = 00h)

Output: None

continued

Subfunction: AL = 30h Get font pointer information

This subfunction returns font pointer information. The pointer information desired is requested in the BH register. To avoid unpredictable results, this subfunction should be issued immediately after a mode set.

Models 25 and 30: IBM PS/2 Models 25 and 30 and compatibles equipped with MCGA hardware do not support 9x14 or 9x16 dot fonts. Calling this function with BH = 05h or 07h yields undefined results.

Input:
- AH = 11h
- AL = 30h
- BH = Font pointer, where:
  - = 00h Return current INT 1Fh pointer
  - = 01h Return current INT 43h pointer
  - = 02h Return ROM font 8x14 pointer
  - = 03h Return current ROM 8x8 font pointer
  - = 04h Return current ROM 8x8 font pointer (top)
  - = 05h Return current ROM 9x14 font alternate
  - = 06h Return current ROM 8x16 font pointer
  - = 07h Return current ROM 9x16 font alternate
  - = >07h Returns all registers (including AX) preserved

Output:
- CX = Bytes per character
- DL = Maximum number of rows on screen
- ES:BP = Pointer to character table

Function: AH = 12h Alternate Select

Description

The Alternate Select function contains several subfunctions. Each subfunction allows the user to enable or disable certain operations which are standard video mode defaults. For example, the Video Service automatically programs the DAC color registers each time the video mode is set. Alternate Select subfunction AL = 31h allows the user to selectively enable or disable DAC programming on mode set.

Alternate select subfunctions

The alternate select subfunctions are listed in the table below:

BL Value	Subfunction Name
10h*	Return VGA information
20h	Switch to alternate print screen routine
30h*	Select scan lines for text modes
31h	Enable/disable default palette loading during mode set
32h	Enable/disable video
33h	Enable/disable summing to gray shades
34h*	Enable/disable cursor scaling
35h	Switch display
36h*	Video screen on/off
* Not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware.	

Error handling

The Video Service checks for valid parameters upon entry into each alternate select subfunction. If an invalid parameter is detected, the subfunction returns with all registers preserved.

continued

Subfunction: BL = 10h Return VGA information

This subfunction returns the state of VGA hardware.

Models 25 and 30: This function is not available on IBM PS/2 Models 25 and 30 and compatibles with MCGA hardware.

Input: BL = 10h

Output: BH = 00h if VGA color mode
= 01h if VGA monochrome mode
BL = Memory available, where:
00h = 64K available
01h = 128K available
02h = 192K available
03h = 256K available
CH = Adapter bits
CL = Switch settings

The references to adapter bits and switch settings are retained only for historical reasons. The VGA hardware is not affected. VGA hardware simulates earlier adapter cards in software rather than in hardware. IBM-compatible VGA hardware only gives a software emulation of switches.

continued

Subfunction: BL = 10h Return VGA information, cont'd

Adapter bits are set from Input Status register 0, responding to output on a given Adapter Control register bit.

Adapter Bit No.	Adapter Control Output Bit Setting	Input Status Bit No.
0	0	5
1	0	6
2	1	5
3	1	6
4-7	unused	

The bits of the switch settings byte give the settings of the EGA's configuration DIP switch (where 1 = off, 0 = on). Color = 09h and monochrome = 0Bh.

Bit No.	Meaning
0	Configuration switch 1
1	Configuration switch 2
2	Configuration switch 3
3	Configuration switch 4
4-7	unused

Subfunction: BL = 20h Switch to alternate print screen routine

This subfunction sets the INT 05h Print Screen service to the proper alternate Print Screen vector. The default BIOS print screen routine assumes a screen length of 25 lines. This subfunction allows an alternative routine to be selected that supports a screen length other than 25 lines.

Input: AH = 12h
BL = 20h

Output: None

continued

Subfunction: BL = 30h Select scan lines for text modes (takes effect on next mode set)

This subfunction sets the number of scan lines to be displayed in text modes. By default, the BIOS sets all text modes to 400 scan lines as part of the power-on self test and initialization (POST) process.

Models 25 and 30: This subfunction is not available on IBM PS/2 Models 25 and 30 and compatibles with MCGA hardware.

Input: AH = 12h
AL = Number of scan lines, where
00h = 200 scan lines
01h = 350 scan lines
02h = 400 scan lines
BL = 30h

Output: AL = 12h Function is supported

Notes:

- The number of scan lines indicated in AL will take effect upon the next mode set.
- The BIOS loads the font associated with the mode/scan line combination upon the next mode set.
- 200-line scan modes are double-scanned. Each video line is painted twice before the next new line is begun.

Subfunction: BL = 31h Enable/Disable default palette loading during set mode

If Disable Default Palette Loading is selected, no updates are made to the overscan color register, the attribute color registers or the DAC color registers when a video mode is set.

Input: AH = 12h
AH = 00h
AL = 00h Enable default palette loading
= 01h Disable default palette loading
BL = 31h

Output: AL = 12h Function supported

---

continued

Subfunction: BL = 32h Enable/Disable video

This subfunction enables/disables the video I/O port regenerator and the display buffer address decode for the currently active video monitor.

Input: AH = 12h
AL = 00h Enable video
= 01h Disable video
BL = 32h

Output: AL = 12h Function supported

Subfunction: BL = 33h Enable/Disable summing to gray shades

This subfunction enables/disables gray scale summing that occurs during Mode Set (AH = 00h) and during Set Palette Registers (AH = 10h) for the display that is currently active.

Input: AH = 12h
AL = 00h Enable summing
= 01h Disable summing
BL = 33h

Output: AL = 12h Function is supported

Subfunction: BL = 34h Enable/Disable cursor scaling

When cursor scaling is enabled (which is the power-on default), cursor start and end information is scaled to the current character height. See INT 10h, AH = 01h Set Cursor Type for more information on setting cursor type.

Models 25 and 30: This function is not available on IBM PS/2 Models 25 and 30 and compatibles with MCGA hardware.

Input: AH = 12h
AL = 00h Enable cursor scaling
= 01h Disable cursor scaling
BL = 34h

Output: AL = 12h Function is supported

---

continued

Subfunction: BL = 35h Switch Display

When there is a conflict between the system video and an adapter video (i.e. overlapping usage of the CBIOS data area and/or hardware capabilities), the adapter video is the power-up default primary video. The system board video remains inactive until switching has been enabled.

To be able to switch, the Disable Video subfunction (AH = 12h, BL = 32H) must be supported by both the adapter and the system board.

- To enable video switching, first call with BL = 35h and AL = 00h to disable the adapter video, then call with AL = 01h to enable system board video.
- Once switching has been enabled, all subsequent switches are done through AL = 02h Disable Active Video and AL = 03h Enable Inactive Video.

Input: AH = 12h
AL = 00h Turn off initial video adapter, must have 128-byte save area pointed to by ES:DX
= 01h Turn on initial system board video
= 02h Disable active video, must have save buffer pointer in ES:DX
= 03h Enable inactive video, must have ES:DX pointer to previously filled save buffer
BL = 35h

Output: AL = 12h function supported

When there is no conflict between the system video and an adapter video board, both video devices are active and use of the display switch subfunction is not necessary.

Subfunction: BL = 36h Video screen ON/OFF

This subfunction turns the video screen on or off.

Models 25 and 30: This function is not available on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware.

Input: AH = 12h
AL = 01h Screen OFF
= 00h Screen ON
BL = 36h

Output: AL = 12h Function supported

Function: AH = 13h Write String

Description

The Write String function operates similarly to Write Teletype (AH = 0Eh), except that an entire string is handled with each call. The AL register contains two single-bit fields as follows:

If...	Then...
AL Bit 0 = 1	the cursor remains at the last character written.
AL Bit 0 = 0	the cursor is restored to where it was before the "write string" operation began.
AL Bit 1 = 1	each string character to be displayed is followed by its attribute.
AL Bit 1 = 0	carriage return, line feed, backspace, and bell are interpreted as commands rather than as printable characters. The string contains only display characters, and the attribute is taken from the BL register.

Input/Output

Input:

AH	=	13h
ES:BP	=	Pointer to start of string
BH	=	Page number (for text modes)
BL	=	Attribute for characters (graphics modes)
CX	=	Length of string (attributes don't count)
DX	=	Starting cursor position (DH = row, DL = column)
AL	=	Flags (see above)

Output: None

Functions: AH = 14h - 19h Reserved

Function: AH = 1Ah Read/Write Display Combination Code

Description

The Read/Write Display Combination Code service has two subfunctions, one devoted to reading the display codes and one devoted to writing the display combination codes. In either case, the display code for the active monitor is specified in BL. The display code for the inactive monitor (if any) is specified in BH.

The display codes corresponding to the monitors are listed below.

Code	Description
00h	No display attached
01h	Monochrome Display Adapter (MDA) with monochrome monitor
02h	Color Graphics Adapter (CGA) with color monitor
03h	Reserved
04h	Enhanced Graphics Adapter (EGA) with color monitor
05h	EGA with monochrome monitor
06h	Professional Graphics Adapter (PGA) with color monitor
07h	Video Graphics Array (VGA) with monochrome analog monitor
08h	VGA with color analog monitor
09h-0Ah	Reserved
0Bh	Multicolor Graphics Array (MCGA) with monochrome analog monitor
0Ch	MCGA with color analog monitor
0Dh-FEh	Reserved
FFFFh	Unknown

Subfunction: AL = 00h Read display combination code

Input: AH = 1Ah
AL = 00h

Output: AL = 1Ah Function supported
BH = Inactive display code
BL = Active display code

continued

Function: AH = 1Ah Read/Write Display Combination Code, Continued

Subfunction: AL = 01h Write display combination code

Input: AH = 1Ah
AL = 01h
BH = Alternate display code (if any)
BL = Active display code

Output: AL = 1Ah — Function supported

Function: AH = 1Bh Return Functionality/State Information

Description

The Return Functionality/State Information function outputs a table describing the current state of the video hardware to a 40h-byte buffer location pointed to by ES:DI.

Input/Output

Input: AH = 1Bh
BX = 0000h Implementation Type
ES:DI = Pointer to 40h byte buffer

Output: AL = 1Bh — Function successful

Functionality/state information is output to 40h-byte buffer (see description on the following page.

Error conditions

The CBIOS will accept only one implementation type: BX = 0000h. If a value other than 0000h is input into BX, this function returns the following values:

- AH = 1Bh
- AL = 00h
- All other registers are preserved.

continued

Structure of the Functionality/State Table

Function AH = 1Bh places the functionality and state of the current video environment in a 40h-byte table pointed to by ES:DI. The structure of the 40h-byte Functionality/State Table is detailed below:

Offset	Size	Description
DI+00h	Dword	Segment/offset of static functionality table fixed address
DI+04h	1 Byte	Video Mode (See the Video Modes table in this chapter for supported video modes)
DI+05h	1 Word	Col/row
DI+07h	1 Word	Display buffer length
DI+09h	1 Word	Starting address of buffer
DI+0Bh	1 Word	Array of 8 cursor positions
DI+1Bh	1 Word	Cursor type
DI+1Dh	1 Byte	Active page
DI+1Eh	1 Word	CRT controller base address
DI+20h	1 Byte	Current value of 3x8 register
DI+21h	1 Byte	Current value of 3x9 register
DI+22h	1 Byte	Rows on screen
DI+23h	1 Word	Character height (bytes/character)
DI+25h	1 Byte	Active display code
DI+26h	1 Byte	Inactive display code
DI+27h	1 Byte	Numbers of colors available in current mode
DI+29h	1 Byte	Display pages supported for current video mode

continued

Structure of the Functionality/State Table, cont'd

Offset	Size	Description
DI+2Ah	1 Byte	Scan lines in current video mode 00b = 200 scan lines 01b = 350 scan lines 10b = 400 scan lines 11b = 480 scan lines 100b - 11111111b Reserved
DI+2Bh	1 Byte	Primary character block 00b = Block 001b = Block 110b = Block 2 - 11111111b = Block 255
DI+2Ch	1 Byte	Secondary character block 00b = Block 001b = Block 110b = Block 2 - 11111111b = Block 255
DI+2Dh	1 Byte	Miscellaneous state information Bits 7-6 = Reserved Bit 5 = 0 Background intensity = 1 Blinking Bit 4 = 1 Cursor emulation is active Bit 3 = 1 Mode set auto palette loading disabled Bit 2 = 1 Mono display is attached Bit 1 = 1 gray scale summing is active Bit 0 = 1 All modes on all displays active
DI+2Eh	1 Byte	Reserved
DI+2Fh	1 Byte	Reserved
DI+30h	1 Byte	Reserved
DI+31h	1 Byte	Video memory available 00b = 64K 01b = 128K 10b = 192K 11b = 256K
DI+32h	1 Byte	Save pointer state information Bits 7-6 = Reserved Bit 5 = 1 Display Combination Code (DCC) extension active Bit 4 = 1 Palette override active Bit 3 = 1 Graphics font override active Bit 2 = 1 Text mode font override active Bit 1 = 1 Dynamic save area active Bit 0 = 1 512-character character set active
DI+33h-3Fh	13 Bytes	Reserved

continued

Structure of the Static Functionality Table

The first entry in the Functionality/State Table is a double word pointer to a 16-byte ROM-based Static Functionality Table.

The Static Functionality Table contains fixed video parameter information. To insure compatibility, the static functionality table is located at E000:305Fh in the CBIOS. The contents of the Static Functionality Table are defined below.

Offset	Value	Description
00h	FFh	Video modes supported Bit 7 = 1 Mode 7 supported Bit 6 = 1 Mode 6 supported Bit 5 = 1 Mode 5 supported Bit 4 = 1 Mode 4 supported Bit 3 = 1 Mode 3 supported Bit 2 = 1 Mode 2 supported Bit 1 = 1 Mode 1 supported Bit 0 = 1 Mode 0 supported
01h	F0h	Video modes supported, Continued Bit 7 = 1 Mode F supported Bit 6 = 1 Mode E supported Bit 5 = 1 Mode D supported Bit 4 = 0 Mode C supported Bit 3 = 0 Mode B supported Bit 2 = 0 Mode A supported Bit 1 = 0 Mode 9 supported Bit 0 = 0 Mode 8 supported
02h	0Fh	Video modes supported, Continued Bits 7-4 = Reserved Bit 3 = 1 Mode 13 supported Bit 2 = 1 Mode 12 supported Bit 1 = 1 Mode 11 supported Bit 0 = 1 Mode 10 supported
03h-06h	00h	Reserved
07h	07h	Scan line modes available for text modes Bits 7-3 = 00000b Reserved Bit 2 = 1 400 scan lines supported Bit 1 = 1 350 scan lines supported Bit 0 = 1 200 scan lines supported

continued

Structure of the static functionality table, cont'd

Offset	Value	Description
08h	02h	Number of character blocks available in text modes
09h		Maximum number of active character blocks available in text modes
0Ah	FFh	Miscellaneous Bit 7 = 1 Color paging supported Bit 6 = 1 Color palette supported Bit 5 = 1 EGA palette supported Bit 4 = 1 Cursor emulation supported Bit 3 = 1 Cursor emulation supported Bit 3 = 1 Default palette loading supported Bit 2 = 1 Character font loading supported Bit 1 = 1 gray scale summing supported Bit 0 = 1 All modes on all displays supported
0Bh	0Eh	Miscellaneous, Continued Bits 7-4 = 0000b Reserved Bit 3 = 1 Display combination codes (DCC) supported Bit 2 = 1 Background intensity/blinking control supported Bit 1 = 1 Save/restore supported Bit 0 = 0 Light pen not supported
0Ch-0Dh	00h	Reserved
0Eh	3Fh	Save pointer functions Bits 7-6 = 00b Reserved Bit 5 = 1 DCC extension Bit 4 = 1 Palette override Bit 3 = 1 Graphics font override Bit 2 = 1 Alpha font override Bit 1 = 1 Dynamic save area Bit 0 = 1 512 Character set supported
0Fh	00h	Reserved

Function: AH = 1Ch Save/Restore Video State

Introduction

At any time, the savable portion of the current video state consists of three discrete parts:

Video State	Description
Hardware State	The contents of the CRT, Attribute, and Graphics Controller registers
Video BIOS State	Those areas in the BIOS Data Area in system RAM containing video information
Digital/Analog Converter (DAC) State	The contents of the DAC control and color registers

Description

The Save/Restore Video State function consists of three subfunctions that enable the user to save (and subsequently restore) any or all of the three states making up the current video state.

Video state information is stored in system RAM buffer pointed to by ES:BX. The video states (i.e. hardware/BIOS/DAC) to save or restore are selected in CX by setting or clearing Bits 2–0.

Save/Restore video state subfunctions

Each of the three Save/Restore Video State subfunctions is selected via the AL register. The table below lists the three subfunctions:

AL Value	Subfunction Name
00h	Return Save/Restore buffer size needed
01h	Save current video state
02h	Restore current video state

Models 25 and 30: This function is not supported on IBM PS/2 Models 25 and 30 (and compatibles) with MCGA hardware. If called, no action is performed.

continued

Error handling

Upon entry into the Save/Restore Video State function, the CBIOS checks the parameter passed in the AL register. If it is out of range (i.e. greater than 02h), then the function returns with AL = 00h and AH preserved.

Buffer format for save/restore video state

The save buffer is composed of a 20h-byte Fixed Offset Area followed by one, two, or three optional areas, depending on which areas (hardware/ BIOS/DAC) were specified in CX.

The 20h-byte fixed offset area contains the following information:

Offset	Description
00h	Word offset of hardware save area, if saved
02h	Word offset of BIOS RAM save area, if saved
04h	Word offset of DAC save area, if saved
06h-1Fh	Reserved

Subfunction AL = 00h Return save/restore buffer size needed

This subfunction returns the buffer size (in 64-byte blocks) needed for any combination of the three video states that can be stored. The video state (or combination of video states) is specified in CX; the buffer size in 64-byte blocks is returned in BX.

Input: AH = 1Ch
AL = 00h
CX = Video state to store, where:
Bit 0 = 1 Save video hardware state
Bit 1 = 1 Save video BIOS data area
Bit 2 = 1 Save video DAC state and color registers
Bits 3-15 = 0 Reserved

Output: AL = 1Ch Function is supported
BX = Buffer size block count (1 block = 64 bytes)

continued

Subfunction: AL = 01h Save current video state

This subfunction stores the video states specified in CX to a system RAM buffer area pointed to by ES:BX.

Input: AH = 1Ch
AL = 01h
CX = Video state to store, where:
Bit 0 = 1 Save video hardware state
Bit 1 = 1 Save video BIOS data area
Bit 2 = 1 Save video DAC state and color registers
Bits 3–15 = 0 Reserved
ES:BX = Pointer to system RAM buffer

Output: AL = 1Ch Function is supported

> **Warning**
>
> Saving a video state alters the original contents of the registers or data areas involved. To maintain the current video state, execute the Restore Video State subfunction immediately after saving the video state.

Subfunction: AL = 02h Restore current video state

This subfunction restores the video states specified in CX from the system RAM buffer area pointed to by ES:BX.

Input: AH = 1Ch
AL = 02h
CX = Video state to restore, where:
Bit 0 = 1 Save video hardware state
Bit 1 = 1 Save video BIOS data area
Bit 2 = 1 Save video DAC state and color registers
Bits 3–15 = 0 Reserved
ES:BX = Pointer to previously saved system RAM buffer

Output: AL = 1Ch Function is supported

continued

Structure of hardware state save area

If Bit 0 of the CX register is set and the Save Video State subfunction is executed, then current hardware state is stored to the system RAM buffer pointed to in ES:BX. Executing the Restore Video State subfunction with the same CX and ES:BX values restores the hardware state information to the actual video hardware.

The structure of the hardware state save area is as follows:

Offset	I/O Address	Index	Description
00h	03C4h		Sequencer Index
01h	3B4h/3D4h*		CRT Controller Index
02h	03CEh		Graphics Controller Index
03h	03C0h		Attribute Controller Index
04h	03CAh		Feature Control
SEQUENCER REGISTERS			
05h	03C5h	01h	Clocking Mode
06h	03C5h	02h	Map Mask
07h	03C5h	03h	Character Map Select
08h	03C5h	04h	Memory Mode
09h	03CCh	00h	Miscellaneous Output
* 03B4h in Monochrome Emulation Modes 03D4h in Color Emulation Modes			

continued

Structure of hardware state save area, cont'd

Offset	I/O Address	Index	Description
CRT CONTROLLER REGISTERS			
0Ah	3B5h/3D5h*	01h	Horizontal Total
0Bh	3B5h/3D5h*	02h	Horizontal Display Enable End
0Ch	3B5h/3D5h*	03h	Start Horizontal Blanking
0Dh	3B5h/3D5h*	04h	End Horizontal Blanking
0Eh	3B5h/3D5h*	05h	Start Horizontal Retrace Pulse
0Fh	3B5h/3D5h*	06h	End Horizontal Retrace
10h	3B5h/3D5h*	07h	Vertical Total
11h	3B5h/3D5h*	08h	Overflow
12h	3B5h/3D5h*	09h	Preset Row Scan
13h	3B5h/3D5h*	0Ah	Maximum Scan Line
14h	3B5h/3D5h*	0Ah	Cursor Start
15h	3B5h/3D5h*	0Bh	Cursor End
16h	3B5h/3D5h*	0Ch	Start Address High
17h	3B5h/3D5h*	0Dh	Start Address Low
18h	3B5h/3D5h*	0Eh	Cursor Location High
19h	3B5h/3D5h*	0Fh	Cursor Location Low
1Ah	3B5h/3D5h*	10h	Vertical Retrace Start
1Bh	3B5h/3D5h*	11h	Vertical Retrace End
1Ch	3B5h/3D5h*	12h	Vertical Display Enable End
1Dh	3B5h/3D5h*	13h	Offset
1Eh	3B5h/3D5h*	14h	Underline Location
1Fh	3B5h/3D5h*	15h	Start Vertical Blank
20h	3B5h/3D5h*	16h	End Vertical Blank
21h	3B5h/3D5h*	17h	CRTC Mode Control
22h	3B5h/3D5h*	18h	Line Compare
* 03B5h in Monochrome Emulation Modes 03D5h in Color Emulation Modes			

continued

Structure of hardware state save area, cont'd

Offset	I/O Address	Index	Description
ATTRIBUTE CONTROLLER REGISTERS			
23h-32h	03C1h	00h-0Fh	00-0F Palette Registers
33h	03C1h	10h	Attribute Mode Control Register
34h	03C1h	11h	Overscan Color Register
35h	03C1h	12h	Color Plane Enable Register
36h	03C1h	13h	Horizontal PEL Panning Register
GRAPHICS CONTROLLER REGISTERS			
37h	03CFh	00h	Set/Reset
38h	03CFh	01h	Enable Set/Reset
39h	03CFh	02h	Color Compare
3Ah	03CFh	03h	Data Rotate
3Bh	03CFh	04h	Read Map Select
3Ch	03CFh	05h	Graphics Mode Register
3Dh	03CFh	06h	Miscellaneous
3Eh	03CFh	07h	Color Don't Care
3Fh	03CFh	08h	Bit Mask
40h	40:63h		CRTC Base Address Low
41h	40:64h		CRTC Base Address High
42h	AFFFFh	00h	Plane 0 System Latch
43h	AFFFFh	01h	Plane 1 System Latch
44h	AFFFFh	02h	Plane 2 System Latch
45h	AFFFFh	03h	Plane 3 System Latch

continued

Structure of video BIOS state save area

If Bit 1 of the CX register is set and the Save Video State subfunction is executed, then the current video BIOS state is stored to the system RAM buffer pointed to in ES:BX. Executing the Restore Video State subfunction with the same CX and ES:BX values restores the video BIOS state information in its location in the BIOS Data Area (0400h–0500h) in system RAM .

Video data definitions

The data definitions used by the INT 10h Video services are stored in system RAM in segment 40h and are presented below.

Location	Size	Description
49h	1 Byte	Video mode setting.
4Ah	1 Word	Number of columns on screen.
4Ch	1 Word	Current page size.
4Eh	1 Word	Current page address.
50h	8 Words	Cursor position on each page. Two bytes/page. First byte (low order) of each pair is column, second is row. 0, 0 is upper left corner of screen.
60h	1 Word	Cursor type defined as 6845 video chip-compatible starting and ending scan lines. High-order byte holds starting scan line; low-order byte holds ending scan line.
62h	1 Byte	Current page number.
63h	1 Word	6845-compatible I/O port number for current mode. (Port 03D4h or 03B4h)
65h	1 Byte	Current mode select register.
66h	1 Byte	Current palette value.
84h	1 Byte	Number of rows on screen (24 or 25)
85h	1 Word	Character height (bytes/character)

continued

Video data definitions, cont'd

System RAM Offset (hex)	Size	Description
87h	1 Byte	Video control bits, where: Bit 7 = Clear RAM Bit 6,5 = Memory on video hardware as follows: 00b = 64K 01b = 128K 10b = 192K 11b = 256K Bit 4 = Not used Bit 3 = 0 If EGA-compatible mode is active Bit 2 = 0 Wait for display enable Bit 1 = 0 If color or ECD monitor is attached to EGA-compatible adapter = 1 If Monochrome monitor is attached to EGA-compatible adapter Bit 0 = 0 If translate cursor video modes 0-3 when using ECD monitor in 350 line mode
88h	1 Byte	EGA/VGA switch data where: Bits 7-4 = Feature connector bits 3-0, respectively Bits 3-0 = Option switches 3-0, respectively
89h	1 Word	VGA control bits, where: Bit 7 = 200 lines Bits 6-5 = Reserved Bit 4 = 400 lines Bit 3 = No palette load Bit 2 = Mono monitor Bit 1 = Gray scalling Bit 0 = Reserved
8Ah	1 Byte	Index to the Display Combination Code table
A8h	1 Word	Pointer to video parameter table and overrides (in segment:offset format).

continued

Structure of DAC state save area

If Bit 2 of the CX register is set and the Save Video State subfunction is executed, then the current state of the DAC is stored to the system RAM buffer pointed to in ES:BX. Executing the Restore Video State subfunction with the same CX and ES:BX values restores the DAC information to the actual DAC hardware.

The structure of the DAC state save area is as follows:

Offset	I/O Address	Index	Description
00h	03C7h		Read/Write mode DAC
01h	03C8h		Pixel address
02h	03C6h		Pixel mask
03h	03C9h	00h	Red value color 0
04h	03C9h	00h	Green value color 0
05h	03C9h	00h	Blue value color 0
06h	03C9h	01h	Red value color 1
300h	03C9h	FFh	Red value color FFh
301h	03C9h	FFh	Green value color FFh
302h	03C9h	FFh	Blue value color FFh
303h	03C1h	14h	Color select register

Function: AH = 1Dh - FFh Reserved

10

Chapter 10
INT 13h Diskette Service

Overview

Description

The CBIOS Diskette Service performs BIOS-level read, write, format, diagnostic, initialization and other operations for up to two internal diskette drives.

How diskette services are invoked

The CBIOS Diskette Service is invoked via software INT 13h.

The CBIOS initializes the INT 13h Diskette Service vector to F000:E3FEh, the IBM-compatible entry point. The INT 13h vector resides at address 00:4Ch in the Interrupt Vector Table.

The CBIOS Diskette Service has eleven functions. Individual functions are selected via the AH register.

continued

When a fixed disk is present

When a fixed disk is installed, the CBIOS automatically redirects all INT 13h Diskette Service requests to INT 40h. This redirection is transparent to end users. End users should continue to invoke INT 13h for both diskette and fixed disk services.

For a discussion of the CBIOS Fixed Disk Service, refer to Chapter 11 of this volume.

Summary of diskette functions

The following table summarizes Diskette Service functions:

Function	Description
00h	Reset Diskette System
01h	Read Diskette Status
02h	Read Diskette Sectors
03h	Write Diskette Sectors
04h	Verify Diskette Sectors
05h	Format Diskette Track
06h–07h	Reserved
08h	Read Drive Parameters
09h–14h	Reserved
15h	Read Drive Type
16h	Detect Media Change
17h	Set Diskette Type
18h	Set Media Type for Format
19h–FFh	Reserved

continued

In this chapter

This chapter focuses on the CBIOS Diskette Service INT 13h. The following topics are discussed:

- Hardware Environment
- Diskette Service I/O Ports
- System RAM Data
- CMOS RAM Data
- ROM BIOS Data
- Error Handling
- Rules for Using CBIOS and ABIOS Diskette Services
- Diskette Service Functions
- INT 0Eh Diskette Hardware Interrupt

Hardware Environment

Hardware assumed

The NEC 765 diskette controller chip (or its equivalent) is used to control the internal and external diskette drives.

How diskette drives are identified

Each diskette drive must be identified to the system with a unique number. There are only two valid diskette numbers:

- Diskette drive 0 is number 0
- Diskette drive 1 is number 1

These values must be in DL for most Diskette Service functions.

Supported drive types

The CBIOS Diskette Service supports six types of diskette drives:

Drive Type	Media type	Disk Size	Tracks/Side	Sectors/Track
1.44 MB	1.44 MB	3.5	80	18
720K	720K	3.5	40	18
360K	360K	5.25	40	9
320K	360K	5.25	40	8
720K	720K	5.25	80	8
1.2 MB	1.2 MB	5.25	80	15

360K diskette format compatibility

CBIOS supports 8 or 9 sectors per track and either single-sided or double-sided diskette drives. 360K is the maximum data storage capability for standard double-density diskette drives. 320K, 160K, or 180K diskette capabilities are also supported.

continued

5.25 inch diskette compatibility

5.25 inch diskette media can be high density (1.2 MB) or double density (360K). Diskettes written on one type of 5.25 inch drive may or may not be written on or read from using the other type. The following table outlines the possible read/write combinations of 5.25 inch diskette media and drive types.

Media Type	If diskette was formatted on...	Then it can be read on...	or...	And it can be written to by...
360K	360K drive	360K drives	1.2 MB drives	360K drives only
1.2 MB	360K drive	360K drives	1.2 MB drives	360K drives only

If a high density (1.2 MB) diskette is formatted on a 1.2 MB drive, then it can be read on and written to by 1.2 MB drives only.

The DOS command, format/4, can be used to format a 360K diskette in a 1.2 MB drive. Both 1.2 MB and 360K drives can generally read and write these diskettes.

3.5 inch diskette compatibility

The following outlines the possible read/write combinations for 3.5 inch drives and media:

- If a 720K media type diskette is formatted on a 720K drive, then it can be read on either 720K or 1.44 MB drives, but it can only be written to by 720K drives.
- If a 1.44 MB media type diskette is formatted on a 1.44 MB drive, then it can be read on and written to by 1.44 MB drives only.

continued

Data transfer rates

All supported formats use a 512-byte sector size. Data transfer rates are:

Transfer Rate	Diskette Capacity	Drive Capacity	Drive Size
250 Kbs	360K	360K	5.25
250 Kbs	720K	720K	3.5
500 Kbs	1.2 MB	1.2 MB	5.25
300 Kbs	360K	1.2 MB	5.25
250 Kbs	720K	1.44 MB	3.5
500 Kbs	1.44 MB	1.44 MB	3.5
Note: Kbs = Kilobytes per second			

Diskette Service I/O Ports

I/O Address	Read/Write Status	Description
0004h	R/W	DMA channel 2, memory address register
0005h	R/W	DMA channel 2, transfer count register
000Ah	R/W	DMA channel 0–3, mask register, where: Bits 7–3 = 0 Reserved Bit 2 = 0 Clear mask Bit = 1 Set mask Bit Bits 1–0 = 00b Select channel 0 = 01b Select channel 1 = 10b Select channel 2 = 11b Select channel 3
000Bh	W	DMA channel 0–3, mode register, where: Bits 7–6 = 00b Demand mode = 01b Signal mode = 10b Block mode = 11b Cascade mode Bit 5 = 0 Address increment select = 1 Address decrement select Bit 4 = 0 Autoinitialization disable = 1 Autoinitialization enable Bits 3–2 = 00b Verify operation = 01b Write to memory = 10b Read from memory = 11b Reserved Bits 1–0 = 00b Channel 0 select = 01b Channel 1 select = 10b Channel 2 select = 11b Channel 3 select
000Ch	W	DMA Clear Byte Pointer
0070h	W	CMOS RAM address register port
0071h	R/W	CMOS RAM data register port
0081h	R/W	DMA channel 2, page register
03F0h	R	Diskette controller status register A, where: Bit 7 = 1 Interrupt pending Bit 6 = 0 Second drive installed Bit 5 = 1 Step Bit 4 = 1 Track 0 Bit 3 = 1 Head 1 select Bit 2 = 0 Index Bit 1 = 0 Write protect Bit 0 = 1 Direction

continued

Diskette Service I/O Ports, Continued

I/O Address	Read/Write Status	Description
03F1h	R	Diskette controller status register B, where: Bit 7 = 0 Reserved Bit 6 = 0 Reserved Bit 5 = 0 Drive 0 select Bit 4 = Write data Bit 3 = Read data Bit 2 = 1 Write enable Bit 1 = 1 Motor enable 1 Bit 0 = 1 Motor enable 0
03F2h	W	Diskette controller digital output register, where: Bit 7 = 0 Reserved Bit 6 = 0 Reserved Bit 5 = 1 Motor enable 1 Bit 4 = 1 Motor enable 0 Bit 3 = 0 Reserved Bit 2 = 0 Controller reset Bit 1 = 0 Reserved Bit 0 = 0 Drive select 0 = 1 Drive select 1
03F4h	R	Diskette controller status register, where: Bit 7 = 1 Data register is ready for transfer Bit 6 = 1 Transfer is from controller to system = 0 Transfer is from system to controller Bit 5 = 1 Non-DMA mode Bit 4 = 1 Diskette controller busy Bit 3 = 0 Reserved Bit 2 = 0 Reserved Bit 1 = 1 Drive 1 busy Bit 0 = 0 Drive 0 busy
03F5h	R/W	Diskette controller data registers
03F7h	R	Digital input register, where: Bit 7 = 1 Diskette change Bits 6-1 = Reserved Bit 0 = 0 High density select
03F7h	W	Diskette controller configuration control register, where: Bits 7-2 = Reserved Bits 1-0 = 00b 500 Kbs mode = 01b 300 Kbs mode = 10b 250 Kbs mode = 11b Reserved

System RAM data area table

The CBIOS Diskette Service references control data stored in the CBIOS Data Area (400h–500h). All Diskette Service items are defined in the table below.

Location	Size	Description
40:10h	2 Bytes	Number of devices installed, where: Bits 15–14 = Number of printer adapters Bits 13,12 = Reserved Bits 11–9 = Number of asynchronous adapters (RS232) Bit 8 = Reserved Bits 7–6 = Number of diskette drives, where: 00b= 1 diskette drive 01b= 2 diskette drives Bits 5–4 = Initial video mode, where: 00b= EGA or PGA 01b= 40x25 color 10b= 80x25 color 11b= 80x25 black and white Bit 3 = Reserved Bit 2 = Pointing device installed Bit 1 = 1 Math coprocessor installed Bit 0 = Diskette available for boot
40:3Eh	1 Byte	Diskette drive recalibration status, where: Bit 7 = 1 Diskette hardware interrupt has occurred Bits 6–4 = Not used Bits 3,2 = Reserved Bit 1 = Recalibrate drive 1 Bit 0 = Recalibrate drive 0
40:3Fh	1 Byte	Diskette drive motor status, where: Bit 7 = 1 Current operation is a write or format = 0 Current operation is a read or verify Bit 6 = Reserved Bit 5,4 = Drive select status where: 00b= Drive 0 selected 01b= Drive 1 selected 10b= Reserved 11b= Reserved Bit 1 = 1 Drive 1 motor is on Bit 0 = 1 Drive 0 motor is on
40:40h	1 Byte	Diskette motor time-out count

continued

System RAM data area table, cont'd

Location	Size	Description
40:41h	1 Byte	Diskette status return code, where: Bit 7 = 1 Drive not ready Bit 6 = 1 Seek error occurred Bit 5 = 1 Diskette controller failed Bits 4-0 = Error codes, where: 01h= Illegal function requested 02h= Address mark not found 03h= Write protect error 04h= Sector not found 06h= Drive door was opened 08h= DMA overrun error 09h= DMA boundary error 0Ch = Media type unknown 10h= CRC failed on diskette read
40:42h	7 Bytes	Diskette controller status bytes
40:74h	1 Word	Status from last fixed disk operation, where: 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 EOh = Status error/error register is 0 FFh = Sense operation failed
40:75h	1 Byte	Number of fixed disk drives

continued

System RAM data area table, cont'd

Location	Size	Description
40:8Bh	2 Bytes	Diskette data rate information Bits 7,6= Last data rate set by controller, where: 00b = 500 Kilobytes/second (Kbs) 01b = 300 Kbs 10b = 250 Kbs 11b = Reserved Bits 5,4 = Last diskette drive step rate selected Bits 3,2 = Data transfer rate at operation start, where: 00b = 500 Kbs 01b = 300 Kbs 10b = 250 Kbs 11b = Reserved Bits 1,0 = Reserved
40:8Fh	1 Byte	Diskette controller information, where: Bit 7 = Reserved Bit 6 = 1 Drive determined for drive 1 Bit 5 = 1 Drive 1 is multirate Bit 4 = 1 Drive 1 supports change line Bit 3 = Reserved Bit 2 = 1 Drive determined for drive 0 Bit 1 = 1 Drive 0 is multirate Bit 0 = 1 Drive 0 supports change line
40:90h-91h	2 Bytes	Media Type of Both Drives: (One byte per drive. drive 0 at 40:90h; drive 1 at 40:91h) Bits 7,6 = Data Transfer Rate, where: 00b = 500 Kbs 01b = 300 Kbs 10b = 250 Kbs Bit 5 = 1 Double stepping required (360K media/1.2 MB drive) Bit 4 = 1 Known media in drive Bit 3 = Reserved Bits 2-0 = Definitions on return to user: 111b = 720K media in 720K or 1.44 MB drive; or 1.44 MB media in 1.44 MB drive 101b = Known 1.2 MB media in 1.2 MB drive 100b = Known 360K media in 1.2 MB drive 011b = Known 360K media in 360K drive 010b = Trying 1.2 MB media in 1.2 MB drive 001b = Trying 360K media in 1.2 MB drive 000b = Trying 360K media in 360K drive

continued

System RAM data area table, cont'd

Location	Size	Description
40:92h	2 Bytes	Diskette Service work area. Each entry is first FDMED value tried. One byte per drive. Drive 0 at 92h, Drive 1 at 93h.
40:94h	2 Bytes	Current track number for both drives. One byte per drive. Drive 0 at 94h, drive 1 at 95h.
40:A0h	2 Bytes	Wait active flag, where: Bit 7 = 1 Wait time elapsed Bits 6-1 = Reserved Bit 0 = 1 INT 15h, AH = 86h occurred

CMOS RAM Data

Introduction

The Diskette Service routine makes use of control information located in CMOS RAM data areas 0Eh through 10h. The table below provides detailed information about the areas used.

CMOS RAM data area table

The following table describes all CMOS RAM data areas used by this interrupt.

CMOS RAM Offset (hex)	Size	Description
0Eh	1 Byte	Diagnostic status, where: Bit 7 = 1 Real time clock lost power Bit 6 = 1 CMOS checksum is bad Bit 5 = 1 Invalid configuration information at POST Bit 4 = 1 Memory size compare error at POST Bit 3 = 1 Fixed disk or adapter fails initialization Bit 2 = 1 CMOS time found invalid Bit 1 = 1 Adapters do not match configuration Bit 0 = 1 Time-out in reading an adapter ID
10h	1 Byte	Type of diskette drives: Bits 7-4 = Drive type of drive 0, where: 0000b = No drive 0001b = 360K drive 0010b = 1.2 MB drive 0011b = 720K 0100b = 1.44 MB Bits 3-0 = Drive type of drive 1, where: 0000b = No drive 0001b = 360K drive 0010b = 1.2 MB drive 0011b = 720K 0100b = 1.44 MB

ROM BIOS Data

Introduction

The diskette device service routine in PC XT/AT systems supplied a table of parameters used to manipulate diskette drives. For compatibility, the CBIOS Diskette service must still provide the same table, although some of its values are modified so that the table is more suited to the default 3.5-inch 1.44 MB diskette drives. The default 11-byte table is located in ROM at F000:EFC7h and is pointed to by the interrupt 1Eh vector.

Parameter table

Byte	Description
00h	The format for the first data byte of the diskette specify command is: Bits 7-4 = Stepping rate Bits 3-0 = Head unload time The default stepping rate is 0Ah for the 1.44 MB drive with a 1.44 MB diskette inserted. The default head unload time is the maximum value of 240 ms (0Fh).
01h	The format for the second data byte of the diskette specify command is: Bits 7-1 = Head load time Bit 0 = Non-DMA mode flag The default head load delay time is set to the minimum value of 4 ms (01h). The heads are loaded at the same time as the motor is started, but the motor delay is much longer so the head load time delay is not really needed. The non-DMA mode flag is always set to zero to indicate that DMA is being used.
02h	Motor turn-off delay. This is the amount of time in timer ticks that the diskette device service routine waits before turning off an inactive diskette drive motor. Timer ticks occur 18.2 times per second and the routine waits about two seconds; therefore the default value for this field is 25h.
03h	Bytes per sector. This field is encoded in the following way to match the encoding used by the diskette controller: 00 = 128 bytes per sector 01 = 256 bytes per sector 02 = 512 bytes per sector 03 = 1024 bytes per sector The standard sector size is 512 bytes per sector, so the default value for this field is 02h.
04h	End of track. Since the sector numbering is one relative, this field is actually the number of sectors per track. For a 1.44 MB diskette in the 1.44 MB drive, this field is 18 sectors per track.

continued

Parameter table, cont'd

Byte	Description
05h	Gap length. This field is the length of the gap between sectors. The default value for this field is 1Bh for a 1.44 MB diskette.
06h	Data Length. Since the bytes per sector field is non-zero, this field is meaningless and is set to FFh.
07h	Gap length for format. This field is the length of the gap between sectors to maintain when formatting. The default format gap length for a 1.44 MB diskette is 6Ch.
08h	Filler byte. This byte is the format filler byte and is F6h.
09h	Head seek delay time. This is the amount of time in milliseconds the diskette device service routine must wait for the heads to settle after doing a seek operation. For the 1.44 MB diskette drive, this field is 0Fh.
0Ah	Motor start time. This is the amount of time in eighths of a second that the diskette device service routine must wait for the motor to come up to speed before doing an I/O operation. The 1.44 MB drive has a motor start time of one second so the default value for this field is set to 08h.

Error Handling

Introduction

The CBIOS Diskette Service returns the completion status of each function via the AH register, the Carry Flag, Diskette Status (40:41h) and in Disk Controller Status (40:42h).

Successful functions

Successful functions return as follows:

- AH = 00h (No error)
- Diskette Status (40:41h) = AH
- Carry Flag = 0 (cleared)

Unsuccessful functions

Unsuccessful functions return as follows:

- AH = xxh (Error code)
- Diskette Status (40:41h) = AH
- Carry Flag = 1 (set)

Note: If the error was related to the diskette controller hardware, the contents of AH are also copied to Disk Controller Status (40:42h).

continued

Table: Error Codes

The CBIOS Diskette Service error codes are listed in the table below. The Carry Flag is set if any of the following errors occur.

Error code (in AH)	Description
00h	No error
01h	Invalid function request
02h	Address mark not found
03h	Write attempted on write-protected disk
04h	Sector not found
06h	Diskette change line active
08h	DMA overrun
09h	DMA boundary error
0Ch	Media type not available
10h	Bad CRC
20h	Diskette controller failed
40h	Seek failed
80h	Time-out

Rules for Using CBIOS and ABIOS Diskette Services

Introduction

Because they interface with two different operating systems, either the CBIOS or the ABIOS Diskette Services may put the diskette hardware into an unknown state.

Diskette hardware is not updated

The CBIOS does not inform the ABIOS of the diskette hardware's state when control is passed from the CBIOS to the ABIOS. Conversely, the ABIOS does not update the CBIOS diskette service data areas when control is passed back to the CBIOS from the ABIOS.

As a result, it is possible for one part of the PS/2 ROM BIOS to put the diskette hardware in a state that will not be recognized by the other. To avoid potential confusion, care must be taken to reset the diskette system each time the CBIOS diskette service is re-entered after any ABIOS service call.

Rules

The matter is resolved by following the rules below:

1. Do not invoke a CBIOS function if there is already an outstanding ABIOS function request.
2. Do not invoke an ABIOS function if there is already an outstanding CBIOS function request.
3. If the last function call was to ABIOS, invoke CBIOS function INT 13h, AH = 00h, Reset Diskette System, before invoking another CBIOS function.
4. If the last function call was to CBIOS, invoke ABIOS Diskette Service function 05h Reset Diskette System before invoking another ABIOS function.
5. ABIOS Diskette Service function 05h, Reset Diskette System, must be the first ABIOS function called after ABIOS is initialized.

Function: AH = 00h Reset Diskette System

Description

The Reset Diskette System function resets both the diskette controller and the indicated diskette drive (the R/W arm is moved to cylinder 0). The diskette drive number (either 0 or 1) is indicated in DL.

Using CBIOS Diskette Services and ABIOS Diskette Services

- Do not invoke a CBIOS Diskette Service function if there already is an outstanding ABIOS function request.
- Invoke INT 13h, AH = 00h, Reset Diskette System, before invoking any other CBIOS Diskette Service I/O function if the last Diskette Service access was via ABIOS.

Switching between ABIOS and CBIOS

Invoke this function after switching from CBIOS to ABIOS and before invoking any other ABIOS Diskette Service functions. For more information, see Rules for Using CBIOS and ABIOS Diskette Services in this chapter.

Input/Output

Input: AH = 00h
DL = Drive number (0 or 1)
Bit 7 = 0 for a diskette
1 for a fixed disk

Output: AH = 00h No error
xxh Error (See "Error Codes" earlier in this chapter)
= Diskette Status (40:41h)
CF = 0 No error
= 1 Error

Call this function when a problem occurs in attempting to access the diskette system, then retry the function that failed. The disk system will not react immediately. Instead, a reset flag forces the BIOS to recalibrate the disk drive's read/write heads the next time they are used. The heads are pulled to track 0 in order to start the next I/O operation from track 0.

Function: AH = 01h Read Diskette Status

Description

The CBIOS Diskette Service stores the error code associated with the last requested function in Diskette Status (40:41h). If the last function executed successfully, Diskette Status is set to 00h. If the last function was not successful, Diskette Status is set to the appropriate non-zero error code.

The Read Diskette Status function reads the value contained in Diskette Status and returns it in the AH register. The Carry Flag (CF) is set if the value returned in AH is nonzero. Otherwise the Carry Flag is cleared.

Input/Output

Input:	AH	=	01h
	DL	=	0 or 1 (drive number)
Output:	AH	=	00h No error
			xxh Error (See "Error Codes" earlier in this chapter)
		=	Diskette Status (40:41h)
	CF	=	0 No error
		=	1 Error

The Diskette Status byte is saved after each read, write, verify or format disk function. This allows error handling or error reporting routines to be written which are entirely independent of disk operation routines.

Function: AH = 02h Read Diskette Sectors

Description

The Read Diskette Sectors function:

- Reads the number of sectors specified in AL,
- From the drive specified in DL,
- To the buffer specified in ES:BX

The diskette drive head number is indicated in DH. Starting cylinder and sector numbers are indicated in CH and CL.

This service is useful for bulk operations that require reading of individual or a whole trackful of sectors. For example, DISKCOPY in DOS uses this service.

Input/Output

Input:

AH	=	02h
AL	=	Number of sectors (1–18, depending on drive media type)
CH	=	Track number (0–79, depending on drive media type)
CL	=	Sector number (1–18, depending on drive media type)
DH	=	Head number (0–1)
DL	=	Drive number (0 or 1)
ES:BX	=	Pointer to buffer

Output:

AL	=	Number of sectors read/written
AH	=	00h No error
		xxh Error (See "Error Codes" earlier in this chapter)
	=	Diskette Status (40:41h)
CF	=	0 No error
	=	1 Error

Wait for device to reach proper speed

With diskette drives, an error may be caused by the drive motor being off when the request is made. The CBIOS does not wait for the device to reach proper speed before trying to read. The calling program should reset the diskette drive (INT 13h function 00h) and retry three times to make sure that an error is real.

Function: AH = 03h Write Diskette Sectors

Description

The Write Diskette Sectors function:

- Writes the number of sectors specified in AL,
- From the drive specified in DL,
- To the buffer specified in ES:BX

The diskette drive head number is indicated in DH. Starting cylinder and sector numbers are indicated in CH and CL. Disk sectors must be formatted before they can be written to.

Input/Output

Input:

AH	=	03h
AL	=	Number of sectors (1–18, depending on drive media type)
CH	=	Track number (0–79, depending on drive media type)
CL	=	Sector number (1–18, depending on drive media type)
DH	=	Head number (0–1)
DL	=	Drive number (0 or 1)
ES:BX	=	Pointer to buffer

Output:

AL	=	Number of sectors read/written
AH	=	00h No error
		xxh Error (See "Error Codes" earlier in this chapter)
	=	Diskette Status (40:41h)
CF	=	0 No error
	=	1 Error

Wait for device to reach proper speed

With diskette drives, an error may be caused by the drive motor being off when the request is made. The CBIOS does not wait for the device to reach proper speed before trying to write. The calling program should reset the diskette drive (INT 13h function 00h) and retry three times to make sure that an error is real.

Function: AH = 04h Verify Diskette Sectors

Description

The Verify Diskette Sectors function:

- verifies the number of sectors specified in AL,
- from the drive specified in DL,
- to the buffer specified in ES:BX.

The diskette drive head number is indicated in DH. Starting cylinder and sector numbers are indicated in CH and CL. No data is transferred from the disk in this operation. Disk data is not compared to data in memory.

The CBIOS Diskette Service verifies diskette sectors by determining if the sectors can be found, read, and pass the Cyclic Redundancy Check (CRC).

Input/Output

Input:	AH	=	04h
	AL	=	Number of sectors (1–18, depending on drive media type)
	CH	=	Track number (0–79, depending on drive media type)
	CL	=	Sector number (1–18, depending on drive media type)
	DH	=	Head number (0–1)
	DL	=	Drive number (0 or 1)
Output:	AL	=	Number of sectors actually transferred
	AH	=	00h No error
			nnh Error (See "Error Codes" earlier in this chapter)
		=	Diskette Status (40:41h)
	CF	=	0 No error
		=	1 Error

Function: AH = 05h Format Diskette Track

Description

The Format Diskette Track function formats a single diskette track on the drive specified in DL. The number of sectors is specified in AL, the head number in DH, and the track number in CH. Each call to service 05 can be verified by following it with a call to service 04h.

ES:BX points to a table defining the address fields for the track being formatted.

Note: If the diskette drive in question supports more than one diskette format, the CBIOS user must call either Diskette Service function AH = 17h Set Diskette Type or AH = 18h Set Media Type for Format before calling this function.

Input/Output

Input:

AH	=	05h
AL	=	Number of sectors to be formatted (starts at one)
DL	=	Drive number (0 or 1)
DH	=	Head number (0 or 1)
CH	=	Track number (starts at 0)
ES:BX	=	Address field buffer

Output:

AH	=	00h No error
	=	80h Specified diskette drive does not exist
		xxh Error (See "Error Codes" earlier in this chapter)
	=	Diskette Status (40:41h)
CF	=	0 No error
	=	1 Error

continued

Address Field Table

The Address Field Table must contain one entry for each sector on the track to be formatted. Each entry consists of four bytes as defined in the table below.

Byte	Description
0	Track number
1	Head number — zero based
2	Sector number
3	Bytes/sector number, where: 02h = 512 bytes/sector

Example: Address Field Table

For example, the address field to format track 3, head 0 of a 9 sector/track diskette with 512-byte sectors, would be:

```
db 03,00,01,02,03,00,02,02,03,00,03,02
db 03,00,04,02,03,00,05,02,03,00,06,02
db 03,00,07,02,03,00,08,02,03,00,09,02
```

Copy protection

Service 05h can be used for copy protection by:

- squeezing more sectors onto a track.
- rearranging the order of the sectors.
- leaving out a sector number.
- specifying 1 or more sectors to be an unconventional size.
- adding a sector with an unconventional address mask.

Functions: AH = 06h - 07h Reserved

Function: AH = 08h Read Drive Parameters

Description

The Read Drive Parameters function returns the diskette parameters for the drive specified in DL. (Valid values are 0 and 1.)

- The drive type stored in CMOS RAM determines the parameters returned. If the drive type is not stored in CMOS RAM, all registers return a value of zero, except DL which contains the number of diskette drives installed.
- If the media type is known, the maximum media capacity is returned in registers BL, CH, CL, and DH. A pointer to the diskette parameter table is returned in ES:DI.
- If a diskette drive does not support the diskette change line, the parameters for a 360K, nine sector/track diskette in a 360K drive are pointed to by the value in ES:DI. The drive type returned in BL is zero.
- If a change line is supported, the parameters for a 1.44 MB diskette in a 1.44 MB drive are pointed to by ES:DI. The drive type returned in BL is zero.
- If successful, the Carry Flag is cleared. Diskette Status (40:41h) is not modified unless the drive type input into DL is a value reserved for fixed disks only. (Valid values for fixed disk drives are 80h and 81h.)

continued

Function: AH = 08h Read Drive Parameters, Continued

Input/Output

Input:	AH	= 08h
	DL	= 0 or 1 (drive number)
Output:	AX	= 0
	BH	= 0
	BL	= Bits 4–7 = 0 Bits 0–3 = Valid drive type value from CMOS 01h = 5.25", 360K, 40 track 02h = 5.25", 1.2 MB, 80 track 03h = 3.5", 720K, 80 track 04h = 3.5", 1.44 MB, 80 track
	CF	= 0 No error
		= 1 Illegal parameter
	CH	= Maximum usable track number
	CL	= Maximum usable sector number
	DH	= Maximum usable head number (always 1 if CMOS value valid)
	DL	= Number of diskette drives installed (0, 1, or 2)
	ES:DI	= Pointer to diskette drive parameter table for the maximum media type supported on the specified drive

Functions: AH = 09h – 14h Reserved

Function: AH = 15h Read Drive Type

Description

The Read Drive Type function returns information about the drive specified in the DL register. (Valid values are 0 and 1.)

Unlike most other CBIOS Diskette Service functions, the value returned in the AH register by this function is **not** an Error Code. Instead, the value returned in AH corresponds to one of the four indicators described in the table below.

AH Value	Meaning
0h	No drive installed
01h	Diskette drive that cannot detect media change (360K, 40-track diskette)
02h	Diskette drive that can detect media change (1.2 MB, 80-track diskette)
03h	Fixed disk installed

Input/Output

Input: AH = 15h
DL = Drive number (0 or 1)

Output: AH = 00h Drive number is valid
= 01h Diskette drive with no change line
= 02h Diskette drive with a change line
= 03h Fixed disk installed
CF = 0 No error
= 1 Invalid drive number

Function: AH = 16h Detect Media Change

Description

The Detect Media Change function determines if the drive door has been opened since the last time the change line was cleared.

- Use INT 13h function 15h first to determine whether the diskette drive hardware can sense when a diskette is changed.
- If the drive number is not valid, the Carry Flag is set, AH is set to 01h, and control is returned to the caller.
- If the drive is not configured, AH and Diskette Status are set to 80h (time-out), CF is set, and control is returned to the caller.
- When this function returns with CF set, it does not necessarily mean that a diskette has been changed. It only means that the diskette drive door has been opened and closed since the last time the change line was cleared.
- If the change line is not supported, AH and Diskette Status are set to 06h (media change), CF is cleared, and control is returned to the caller.

Input/Output

Input:	AH	=	16h
	DL	=	Drive number (0 or 1)
Output:	AH	=	00h Diskette change line signal not active
		=	01h Invalid drive number
		=	06h Either change line not supported or diskette change line signal is active
		=	80h Diskette drive not ready or no drive present
		=	Diskette Status (40:41h)
	CF	=	0 No error
		=	1 Error

Function: AH = 17h Set Diskette Type

Description

The Set Diskette Type function sets the transfer rate for the specified drives using the diskette types passed in AL. 1.44 MB drives are not handled by this function. This function is provided for compatibility with DOS 3.0 or 3.1. Function 18h supersedes it for DOS 3.2 and up. If the diskette type in AL is invalid, AH and Diskette Status are set to 01h, CF is set, and control is returned to the caller. If the drive number is invalid, CF is set, AH and Diskette Status are set to 01h, and control is returned to the caller.

The media transfer rates are set according to the drive type specified. For diskette types 2, 3, and 4, the change line determines if there is a diskette in the drive. If there is no diskette in the drive, AH and Diskette Status are set to 80h (time-out), CF is set, and control is returned to the caller. If a diskette is present, Diskette Status is set to 00h and the following values are set according to the diskette type in AL.

Diskette Type	Transfer Rate	Double Step	Known Media
02h	300 Kbs (01b)	Yes	Yes
03h	500 Kbs (00b)	No	Yes
04h	250 Kbs (10b)	No	Yes

Input/Output

Input: AH = 17h
AL = 01h 360K diskette in 360K drive
= 02h 360K diskette in 1.2 MB drive
= 03h 1.2 MB diskette in 1.2 MB drive
= 04h 720K diskette in 720K drive

Output: AH = 00h No error
= xxh Error (See "Error Codes" earlier in this chapter)
= Diskette Status (40:41h)
CF = 0 No error
= 1 Error

Function: AH = 18h Set Media Type for Format

Description

The Set Media Type for Format function sets the media type in preparation for a format command.

- If the drive number is not valid, the Carry Flag (CF) is set and control is returned to the caller with both AH and Diskette Status (40:41h) equal to 01h.
- If the drive number is valid, this function determines if the drive supports a change line. If the drive supports a change line, this function determines if there is a diskette in the drive. If there is no diskette, AH and Diskette Status are set to 80h (time-out), the Carry Flag (CF) is set, and control is returned to the caller.
- If the drive type can't be determined, AH and Diskette Status are set to 0Ch (unknown media), CF is set, and control is returned to the caller.
- If there are no problems with the values returned by this function, a diskette table is chosen based on the data input to this function. ES:DI points to the chosen diskette parameter table, AH and Diskette Status are set to 00h, CF is cleared, and control is returned to the caller.

Input/Output

Input:

AH	=	18h
CH	=	Maximum number of tracks
CL	=	Maximum sectors per track
DL	=	Drive number (0 or 1)

Output:

ES:DI	=	Pointer to drive parameter table
AH	=	00h track/sector combination is supported
	=	0Ch Media unknown (CMOS not valid, drive not configured in CMOS, or not in diskette parameter table)
	=	80h No diskette in drive
	=	Diskette Status (40:41h)
CF	=	0 No error
	=	1 Error

INT 0Eh Diskette Hardware Interrupt

Introduction

The CBIOS Diskette Interrupt Service Routine (ISR) services interrupts from the diskette controller.

How CBIOS Diskette ISR is invoked

The Diskette controller generates an INT 0Eh, the CBIOS Diskette ISR, to signal the completion of the last command issued by INT 13h.

To maintain compatibility, the INT 0Eh entry point is located at address F000:EF57h. INT 0Eh resides at address 40h in the Interrupt Vector Table.

CBIOS multitasking support

Each time the CBIOS Diskette Service issues a diskette controller command, it also invokes INT 15h with AH = 90h Device Busy Loop.

Calling INT 15h allows other tasks in the system to process while diskette I/O is in progress. However, this multitasking must be implemented by user-supplied programs executing in the microprocessor's real address mode.

INT 0Eh and multitasking return

When the diskette controller completes a command, it issues an INT 0Eh interrupt and enables interrupts.

INT 0Eh sets Bit 7 of the Diskette Drive Recalibrate Status byte (40:3Eh) and calls INT 15h with AH = 91h and AL = 01h Interrupt Complete. Then it sets Bit 7 of (40:3Eh) to signal the completion of the command.

continued

I/O Port table

INT 0Eh, Diskette ISR, references the following I/O ports:

I/O Address	Read/Write Status	Description
03F0h	R	Diskette controller status register A, where: Bit 7 = 1 Interrupt pending Bit 6 = 0 Second drive installed Bit 5 = 1 Step Bit 4 = 0 Track 0 Bit 3 = 1 Select head 1 Bit 2 = 0 Index Bit 1 = 0 Write protect Bit 0 = 1 Direction is from controller to diskette

System RAM data area

INT 0Eh, Diskette ISR, references the following system RAM data area locations:

Location	Length	Description
40:3Eh	1 Byte	Diskette drive recalibrate status, where: Bit 7 = 1 Diskette hardware interrupt has occurred Bit 6-4 = Not used Bits 3,2 = Reserved Bit 1 = Recalibrate drive 1 Bit 0 = Recalibrate drive 0

11

Chapter 11
INT 13h Fixed Disk Service

Overview

Description

The CBIOS Fixed Disk Service performs BIOS-level read, write, format, diagnostic, initialization and other operations for up to two fixed disk drives.

How the service is invoked

INT 13h invokes the CBIOS Fixed Disk Service.

The INT 13h vector resides at the Interrupt Vector Table address 00:4Ch. The CBIOS initializes the INT 13h Fixed Disk Service vector to address F000:E3FEh, the IBM-compatible BIOS entry point.

continued

Summary of fixed disk service functions

The CBIOS Fixed Disk Service contains several functions. Individual functions are selected via AH. The table below summarizes these functions.

Function	Description
00h	Reset Disk System
01h	Read Disk Status
02h	Read Disk Sectors
03h	Write Disk Sectors
04h	Verify Disk Sectors
05h	Format Disk Cylinder
06h–07h	Reserved
08h	Read Drive Parameters
09h	Initialize Drive Parameters
0Ah	Read Long Disk Sectors
0Bh	Write Long Disk Sectors
0Ch	Seek to Cylinder
0Dh	Alternate Reset Fixed Disk
0Eh	Diagnostics 1, Read Test Buffer
0Fh	Diagnostics 2, Write Test Buffer
10h	Test for Drive Ready
11h	Recalibrate Drive
12h–14h	Reserved
15h	Read Disk Type
16h–18h	Reserved
19h	Park Heads
1Ah	Format ESDI Drive
1Bh–FFh	Reserved

continued

In this chapter

This chapter focuses on the CBIOS Fixed Disk Service. The following topics are presented:

- Hardware Environment
- Fixed Disk Service I/O Ports
- System RAM Data
- CMOS RAM Data
- ROM BIOS Data
- Fixed Disk Service Error Handling
- Rules for Using CBIOS and ABIOS Fixed Disk Services
- Fixed Disk Service Functions

Support for two fixed disk drives

The CBIOS Fixed Disk Service will physically support up to two fixed disk drives. An ST412/506 interface fixed disk adapter or equivalent is assumed present. CBIOS uses information from the drive table which may include support for RLL and ESDI drives. ESDI drives generally require an option ROM.

How fixed disks are identified

Each fixed disk must be identified to the system with a unique number.

There are only two valid fixed disk identification numbers — 80h and 81h. The numbers must be assigned as per below:

- Fixed Disk Drive 0 = 80h
- Fixed Disk Drive 1 = 81h

Note: The fixed disk drive values 80h or 81h are required DL register inputs for many of the INT 13h fixed disk services.

CBIOS diskette support

When a fixed disk is installed, the CBIOS **automatically** redirects all INT 13h Diskette Service requests to INT 40h. This redirecting is transparent to end users, however. End users should continue to invoke INT 13h for **both** diskette and fixed disk services.

For discussion of CBIOS Diskette Services, refer to Chapter 10.

Fixed Disk Service I/O Ports

INT 13h, Fixed Disk Service, references the following I/O ports:

I/O Address	Read/Write Status	Description
0018h	W	DMA Extended function register where: Bits 7-4 = Progress command, where: 00h I/O address register 01h Reserved 02h Memory address register write 03h Memory address register read 04h Transfer count register write 05h Transfer count register read 06h Status register read 07h Mode register 08h Arbus register 09h Mask register set single bit 0Ah Mask register reset single bit 0Bh-0Ch Reserved 0Dh Master clear 0Eh-0Fh Reserved Bits 3-0 = 0 Reserved
001Ah	R/W	DMA Extended function execute register
0020h	R	Interrupt request/in-service registers programmed by Operation Command Word 3 (OCW3): Interrupt request register, where: Bits 7-0 = 0 No active request for the corresponding interrupt line = 1 Active request for the corresponding interrupt line Interrupt in-service register, where: Bits 7-0 = 0 The corresponding interrupt line is not currently being serviced = 1 The corresponding interrupt line is currently being serviced
0020	W	Initialization Command Word 1 (ICW1) (Bit 4 is one), where: Bits 7-5 = 000 Only used in 80/85 mode Bit 4 = 1 Reserved Bit 3 = 0 Edge triggered mode = 1 Level triggered mode Bit 2 = 0 Successive interrupt vectors are separated by eight bytes = 1 Successive interrupt vectors are separated by four bytes Bit 1 = 0 Cascade mode = 1 Single mode - no ICW3 needed Bit 0 = 0 No ICW4 needed = 1 ICW4 needed

continued

Fixed Disk Service I/O Ports, Continued

I/O Address	Read/Write Status	Description
0021h	W	ICW2, ICW3, or ICW4 in sequential order after ICW1 written to Port 0020h ICW2, where: Bits 7-3 = Address lines A0-A3 of base vector address for interrupt controller Bits 2-0 = 000 Reserved ICW3, where: Bits 7-0= 0 Slave controller not attached to corresponding interrupt pin = 0 Slave controller attached to corresponding interrupt pin ICW4, where: Bits 7-5= 000 Reserved Bit 4 = 0 No special fully-nested mode = 1 Special fully-nested mode Bits 3-2= 00 Nonbuffered mode = 01 Nonbuffered mode = 10 Buffered mode/slave = 11 Buffered mode/master Bit 1 = 0 Normal EOI = 1 Auto EOI Bit 0 = 0 80/85 mode = 1 8086/8088 mode
0021h	R/W	Interrupt mask register (OCW1), where: Bit 7 = 0 Enable parallel printer interrupt Bit 6 = 0 Enable diskette interrupt Bit 5 = 0 Enable fixed disk interrupt Bit 4 = 0 Enable serial interrupt Bit 3 = 0 Reserved Bit 2 = 0 Enable video interrupt Bit 1 = 0 Enable keyboard/pointing device/RTC interrupt Bit 0 = 0 Enable timer interrupt
0021h	W	OCW2 (Bit 4 is zero, Bit 3 is zero), where: Bits 7-5 = 000 Rotate in automatic EOI mode (clear) = 001 Non-specific EOI = 010 No operation = 011 Specific EOI = 100 Rotate in automatic EOI mode (set) = 101 Rotate on non-specific EOI command = 110 Set priority command = 111 Rotate on specific EOI command Bit 4 = 0 Reserved Bit 3 = 0 Reserved Bits 2-0 = Interrupt request to which the command applies

continued

Fixed Disk Service I/O Ports, Continued

I/O Address	Read/Write Status	Description
0020h	W	OCW3 (Bit 4 is zero, Bit 3 is one), where: Bit 7 = 0 Reserved Bits 6-5 = 00 No operation = 01 No operation = 10 Reset special mask = 11 Set special mask Bit 4 = 0 Reserved Bit 3 = 1 Reserved Bit 2 = 0 No poll command = 1 Poll command Bits 1-0 = 00 No operation = 01 No operation = 10 Read interrupt request register on next read at Port 0020h = 11 Read interrupt inservice register on next read at Port 0020h
0070h	W	CMOS RAM address register port, where: Bit 7 = 1 NMI disable = 0 NMI enabled Bits 6-0 = 0 CMOS address
0071h	R/W	CMOS RAM data register port
0074h	W	Extended CMOS RAM address register port, least significant byte
0075h	W	Extended CMOS RAM address register port, most significant byte
0076h	R/W	Extended CMOS RAM data register port
0092h	R/W	System control port A, where: Bit 7 = 1 Fixed disk activity light bit A is on Bit 6 = 1 Fixed disk activity light bit B is on Bit 5 = 0 Reserved Bit 4 = 1 Watchdog time-out occurred Bit 3 = 1 Security lock latch is locked Bit 2 = 0 Reserved Bit 1 = 1 Alternate gate A20 active Bit 0 = 1 Alternate hot reset
00A0h	W	Programmable Interrupt Controller 2
00A1h	R/W	Programmable Interrupt Controller 2 mask, where: Bit 7 = 0 Reserved Bit 6 = 0 Enable fixed disk interrupt Bit 5 = 0 Enable 80387 exception interrupt Bit 4 = 0 Enable mouse interrupt Bit 3 = 0 Reserved Bit 2 = 0 Reserved Bit 1 = 0 Enable redirect cascade Bit 0 = 0 Enable real time clock interrupt

continued

Fixed Disk Service I/O Ports, Continued

I/O Address	Read/Write Status	Description
0320h	R/W	Fixed Disk Adapter Register (8 or 16 bit)
0322h	W	Fixed Disk Adapter Control Register, where: Bit 7 = 1 Reset Bit 6 = 1 Reserved (except during reset) Bit 5 = 1 16-bit mode (must match bit 2) = 0 8-bit mode Bit 4-3 = 0 Reserved Bit 2 = 1 16-bit mode (must match bit 5) = 0 8-bit mode Bit 1 = 1 Enable interrupt through Programmable Interrupt Controller (hardware interrupt) = 0 Enable interrupt through Interrupt Status Register (port 324h) Bit 0 = 1 DMA mode = 0 PIO mode
0322h	R	Fixed Disk Adapter Status Register, where: Bit 7-6 = 0 Reserved Bit 5 = 1 16-bit mode = 0 8-bit mode Bit 4 = 1 Data transfer requested by adapter Bit 3 = 1 Direction is adapter to system = 0 Direction is system to adapter Bit 2 = 1 Busy Bit 1 = 1 Interrupt request (notification) Bit 0 = 1 Transfer in progress
0324h	W	Fixed Disk Adapter Attention Register, where: Bit 7 = 1 Command control block Bit 6 = 1 Command specify block Bit 5 = 1 Sense summary block Bit 4 = 1 Data transfer requested by system Bit 3 = 0 Reserved Bit 2 = 0 Drive 0 select = 1 Drive 1 select Bit 1 = 0 Reserved Bit 0 = 1 Abort current command
0324h	R	Fixed Disk Adapter Interrupt Status Register where: Bit 7 = 1 Termination error, bits 0-6 indicate what the error is Bit 6 = 1 Invalid command Bit 5 = 1 Command reject Bit 4-3 = 0 Reserved Bit 2 = 0 Drive 0 selected = 1 Drive 1 selected Bit 1 = 1 Error recovery procedure invoked Bit 0 = 1 Equipment check

System RAM Data

Introduction

The Fixed Disk Service makes use of System RAM Data located in the CBIOS Data area and in the CBIOS Extended Data area.

The CBIOS Data area (absolute hex address 400–500) contains data definitions that are referred to by the CBIOS Fixed Disk Service.

Location	Size	Description
40:74h	1 Word	Status from last fixed disk operation, where: 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
40:75h	1 Byte	Number of fixed drives
40:8Ch	1 Byte	Fixed disk controller status
40:8Dh	1 Byte	Fixed disk controller error status
40:8Eh	1 Byte	Fixed disk interrupt flag
40:8Fh	1 Byte	Diskette controller information where: Bit 7 = Reserved Bit 6 = 1 If drive determined for drive 1 Bit 5 = 1 If drive is multirate Bit 4 = 1 If drive supports change line Bit 3 = Reserved Bit 2 = 1 If drive 0 is multirate Bit 0 = 1 If drive 0 supports change line
40:Eh	1 Word	Extended CBIOS data area segment

continued

BIOS Extended Data Area

The CBIOS Extended Data area is located in the top 1K of system RAM; the segment of the extended data area is defined in location 40:0Eh of the CBIOS Data area. The Fixed Disk Service uses three locations in the CBIOS Extended Data area. Those locations are defined in the table below:

<table>
<tr><th>Location</th><th>Size</th><th>Description</th></tr>
<tr><td>EDA:3Dh</td><td>16 Bytes</td><td>Fixed disk parameter table for drive 0, where:
<table>
<tr><th>Offset</th><th>Size</th><th>Description</th></tr>
<tr><td>00h</td><td>Word</td><td>Max number of cylinders</td></tr>
<tr><td>02h</td><td>Byte</td><td>Max. number of heads</td></tr>
<tr><td>03h</td><td>Word</td><td>Reserved</td></tr>
<tr><td>05h</td><td>Word</td><td>Starting write precompensation cylinder</td></tr>
<tr><td>07h</td><td>Byte</td><td>Not used</td></tr>
<tr><td>08h</td><td>Byte</td><td>Control byte</td></tr>
<tr><td>09h</td><td>Byte</td><td>Reserved</td></tr>
<tr><td>0Ah</td><td>Byte</td><td>Reserved</td></tr>
<tr><td>0Bh</td><td>Byte</td><td>Reserved</td></tr>
<tr><td>0Ch</td><td>Word</td><td>Landing zone</td></tr>
<tr><td>0Eh</td><td>Byte</td><td>Number of sectors/track</td></tr>
<tr><td>0Fh</td><td>Byte</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>EDA:4Dh</td><td>16 Bytes</td><td>Fixed disk parameter table for drive 1, where:
<table>
<tr><th>Offset</th><th>Size</th><th>Description</th></tr>
<tr><td>00h</td><td>Word</td><td>Max number of cylinders</td></tr>
<tr><td>02h</td><td>Byte</td><td>Max. number of heads</td></tr>
<tr><td>03h</td><td>Word</td><td>Reserved</td></tr>
<tr><td>05h</td><td>Word</td><td>Starting write precompensation cylinder</td></tr>
<tr><td>07h</td><td>Byte</td><td>Reserved</td></tr>
<tr><td>08h</td><td>Byte</td><td>Control byte</td></tr>
<tr><td>09h</td><td>Byte</td><td>Reserved</td></tr>
<tr><td>0Ah</td><td>Byte</td><td>Reserved</td></tr>
<tr><td>0Bh</td><td>Byte</td><td>Reserved</td></tr>
<tr><td>0Ch</td><td>Word</td><td>Landing zone</td></tr>
<tr><td>0Eh</td><td>Byte</td><td>Number of sectors/track</td></tr>
</table></td></tr>
<tr><td>EDA:6Ch</td><td>1 Byte</td><td>Fixed Disk arbitration level and channel number, where:
Bits 7–4 = Channel number
Bits 3–0 = DMA arbitration level</td></tr>
<tr><td colspan="3">Note: EDA = Extended Data Area</td></tr>
</table>

CMOS RAM Data

Introduction

The Fixed Disk Service makes use of data stored in conventional CMOS RAM, and for those MCA-based systems that include more than four expansion slots, in Extended CMOS RAM.

Table of CMOS RAM data used

The CMOS RAM data referenced by the Fixed Disk Service is listed below.

CMOS RAM Offset (hex)	Size	Description
0Eh	1 Byte	Diagnostic status, where: Bit 7 = 1 Real time clock lost power Bit 6 = 1 CMOS checksum is bad Bit 5 = 1 Invalid configuration information at POST Bit 4 = 1 Memory size compare error at POST Bit 3 = 1 Fixed disk or adapter fails initialization Bit 2 = 1 CMOS time found invalid Bit 1 = 1 Adapters do not match configuration Bit 0 = 1 Time-out in reading an adapter ID
11h	1 Byte	Type of fixed disk drive 0
12h	1 Byte	Type of fixed disk drive 1
19h	1 Byte	Adapter ID for channel 0, low byte
1Ah	1 Byte	Adapter ID for channel 0, high byte
1Bh	1 Byte	Adapter ID for channel 1, low byte
1Ch	1 Byte	Adapter ID for channel 1, high byte
1Dh	1 Byte	Adapter ID for channel 2, low byte
1Eh	1 Byte	Adapter ID for channel 2, high byte
1Fh	1 Byte	Adapter ID for channel 3, low byte
20h	1 Byte	Adapter ID for channel 3, high byte
21h	1 Byte	POS 2 configuration byte for channel 0
22h	1 Byte	POS 3 configuration byte for channel 0
25h	1 Byte	POS 2 configuration byte for channel 1
26h	1 Byte	POS 3 configuration byte for channel 1
29h	1 Byte	POS 2 configuration byte for channel 2
2Ah	1 Byte	POS 3 configuration byte for channel 2
2Dh	1 Byte	POS 2 configuration byte for channel 3
2Eh	1 Byte	POS 3 configuration byte for channel 3

continued

Extended CMOS RAM data

The CBIOS uses an additional 2K of CMOS RAM to store Programmable Option Select (POS) data for MCA-based systems that include more than four expansion slots. Fixed disk parameter data for fixed disks other than those listed in the ROM BIOS Fixed Disk Parameter Table is also stored in this extra CMOS RAM.

Table of Extended CMOS data used

The table below outlines the extended CMOS RAM data used by the Fixed Disk Service.

CMOS RAM Offset (hex)	Size	Description
0000h	1 Byte	LSB of adapter ID for channel 0
0001h	1 Byte	MSB of adapter ID for channel 0
0003h	1 Byte	POS 2 for channel 0
0004h	1 Byte	POS 3 for channel 0
0023h	1 Byte	LSB of adapter ID for channel 1
0024h	1 Byte	MSB of adapter ID for channel 1
0026h	1 Byte	POS 2 for channel 1
0027h	1 Byte	POS 3 for channel 1
0046h	1 Byte	LSB of adapter ID for channel 2
0047h	1 Byte	MSB of adapter ID for channel 2
0049h	1 Byte	POS 2 for channel 2
0069h	1 Byte	LSB of adapter ID for channel 3
006Ah	1 Byte	MSB of adapter ID for channel 3
006Ch	1 Byte	POS 2 for channel 3
008Ch	1 Byte	LSB of adapter ID for channel 4
008Dh	1 Byte	MSB of adapter ID for channel 4
008Fh	1 Byte	POS 2 for channel 4
0090h	1 Byte	POS 3 for channel 4
00AFh	1 Byte	LSB of adapter ID for channel 5
00B0h	1 Byte	MSB of adapter ID for channel 5

continued

Extended CMOS data definitions, cont'd

CMOS RAM Offset (hex)	Size	Description
00B2h	1 Byte	POS 2 for channel 5
00B3h	1 Byte	POS 3 for channel 5
00D2h	1 Byte	LSB of adapter ID for channel 6
00D3h	1 Byte	MSB of adapter ID for channel 6
00D5h	1 Byte	POS 2 for channel 6
00D6h	1 Byte	POS 3 for channel 6
00F5h	1 Byte	LSB of adapter ID for channel 7
00F6h	1 Byte	MSB of adapter ID for channel 7
0166 - 0175h	8 Words	Fixed parameter table for drive 0
0176 - 0185h	8 Words	Fixed parameter table for drive 1

Description

The Fixed Disk Parameters Table defines the types of fixed disk drives that can be used. The address of the correct entry within the Fixed Disk Parameters Table is contained in the INT 41h vector for drive 0 and in the INT 46h vector for drive 1. To maintain compatibility, the Fixed Disk Parameters Table is based at hex F000:E401.

Parameter table structure

Each entry in the Fixed Disk Parameters Table occupies 16 bytes.

The 16-byte entry structure is defined in the table below:

Offset	Size	Description
00h	2 Bytes	Maximum number of cylinders
02h	1 Byte	Maximum number of heads
03h	2 Byte	Reserved
05h	2 Bytes	Starting write precompensation cylinder
07h	1 Byte	Reserved
08h	1 Byte	Control byte, where: Bit 7,6 = 1 Disable retries (either bit disables) Bit 5 = 1 Defect map present at max, cylinder + 1 Bit 4 = 0 Reserved Bit 3 = 1 More than 8 heads Bits 2-0 = 0 Reserved
09h	3 Bytes	Reserved
0Ch	2 Bytes	Landing zone
0Eh	1 Byte	Number of sectors per track
0Fh	1 Byte	Reserved

continued

Fixed disk parameter entries

The CBIOS portion of the ROM BIOS defines only the first 32 entries in the Fixed Disk Table. Fixed disk type 15 is not used. Definitions for fixed disk types 1–32 are located in a 16-byte table starting at F000:E401h.

Table of fixed disk definitions

The defined entries into the Fixed Disk Parameters table are listed below. Wherever possible, the manufacturer name and model number associated with a given drive type are listed in the column, "Manufacturer."

Type	Manufacturer	Cyl.	Heads	Write Precomp	Landing Zone	Sectors/ Track	Defect Map
1	IBM 10 MB	306	4	128	305	17	No
2	IBM 20 MB Seagate ST-225 CDC Wren II 9415-5-25 Miniscribe 8438F	615	4	300	615	17	No
3	IBM 30 MB	615	6	300	615	17	No
4	IBM 62 MB	940	8	512	940	17	No
5	IBM 46 MB	940	6	512	940	17	No
6	IBM 20 MB Miniscribe MS 8425 Seagate ST4026 Tandon TM 262 Tandon TM 702AT	615	4	0FFFFh	615	17	No
7	IBM 30 MB	462	8	256	511	17	No
8	IBM 30 MB Seagate ST-4038 CDC Wren II 9415-5-38 Tandon TM 703AT	733	5	0FFFFh	733	17	No
9	IBM 112 MB	900	15	0FFFFh	901	17	No
10	IBM 20 MB	820	3	0FFFFh	820	17	No
11	IBM 35 MB	855	5	0FFFFh	855	17	No
12	IBM 49 MB	855	7	0FFFFh	855	17	No
13	IBM 20 MB	306	8	128	319	17	No

Note: If a table entry contains 0FFFFh for Write Precompensation, then there is no write precompensation for this disk. If the write precompensation is zero, then there is write precompensation for all cylinders.

continued

Table of fixed disk definitions, cont'd

Type	Manufacturer	Cyl.	Heads	Write Precomp	Landing Zone	Sectors/ Track	Defect Map
14	IBM 42 MB	733	7	0FFFFh	733	17	No
15	Not used						
16	IBM 20 MB	612	4	0	663	17	No
17	IBM 40 MB	977	5	300	977	17	No
18	IBM 56 MB	977	7	0FFFFh	977	17	No
19	IBM 59 MB	1024	7	512	1023	17	No
20	IBM 30 MB	733	5	300	732	17	No
21	IBM 42 MB	733	7	300	732	17	No
22	IBM 30 MB	733	5	300	733	17	No
23	IBM 10 MB	306	4	0	336	17	No
24	IBM 20 MB	612	4	305	663	17	No
25	IBM 10 MB	306	4	0FFFFh	340	17	No
26	IBM 20 MB	612	4	0FFFFh	670	17	No
27	IBM 40.5 MB	698	7	300	732	17	Yes
28	IBM 40.5 MB	976	5	488	977	17	Yes
29	IBM 10 MB	306	4	0FFFFh	340	17	No
30	IBM 20 MB	611	4	306	663	17	Yes
31	IBM 42.5 MB	732	7	300	732	17	Yes
32	IBM 42.5 MB	1023	5	0FFFFh	1023	17	Yes

Note: If a table entry contains 0FFFFh for Write Precompensation, then there is no write precompensation for this disk. If the write precompensation is zero, then there is write precompensation for all cylinders.

Fixed Disk Service Error Handling

Introduction

Upon return from each function, the Fixed Disk Service indicates the result of the operation with a numeric error code. This error code is returned in AH and is stored in The Fixed Disk Status byte (40:74h).

Error code and successful functions

Successful functions return with

- AH and Fixed Disk Status (40:74h) set to 00h (i.e. Error Code = 00h, no error)
- Carry Flag (CF) cleared

Error codes and unsuccessful functions

Functions which have not executed successfully return with

- Carry Flag set
- AH and Fixed Disk Status (40:74h) set to one of 23 possible error codes

---

continued

Table of error codes

The table below lists the Fixed Disk Service Error Codes.

Error Code	Description
00h	No error
01h	Invalid function passed in AH or invalid parameter
02h	Address mark not found
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 ECC or CRC error
11h	ECC corrected data error
20h	Controller failure
40h	Seek operation failed
80h	Time-out
AAh	Drive not ready or not selected
BBh	Undefined error occurred
CCh	Write fault on selected drive
E0h	Status error/error register = 0
FFh	Sense operation failed

Rules for Using CBIOS and ABIOS Fixed Disk Services

Introduction

Because they interface with two different operating systems, either the CBIOS or the ABIOS Fixed Disk Services may put the fixed disk hardware into an unknown state.

Fixed disk hardware is not updated

The CBIOS does not inform the ABIOS of the fixed disk hardware's state when control is passed from the CBIOS to the ABIOS. Conversely, the ABIOS does not update the CBIOS fixed disk service data areas when control is passed back to the CBIOS from the ABIOS. As a result, it is possible for one part of the PS/2 ROM BIOS to put the fixed disk hardware in a state that will not be recognized by the other.

To avoid potential confusion, care must be taken to reset the fixed disk system each time the CBIOS Fixed Disk Service is re-entered after **any** ABIOS service call.

Rules

The matter is resolved by following the rules below:

1. Do not invoke a CBIOS function if there is already an outstanding ABIOS function request.
2. Do not invoke an ABIOS function if there is already an outstanding CBIOS function request.
3. If the last function call was to ABIOS, invoke CBIOS function INT 13h, AH = 0Dh Alternate Reset Fixed Disk System before invoking another ABIOS function.
4. If the last function call was to CBIOS, invoke Fixed Disk Service function 05h Reset Fixed Disk System before invoking another ABIOS function.
5. ABIOS Fixed Disk function 05h Reset Fixed Disk System must be the first ABIOS function called after ABIOS is initialized.

Function: AH = 00h Reset Diskette(s) and Fixed Disk

Description

The Reset Diskette(s) and Fixed Disk function resets both the diskette and the fixed disk controllers. It places both the diskette and the fixed disk systems in a known state by reinitializing the fixed disk and diskette drive parameters and by recalibrating the read/write heads positions of both devices.

The fixed disk drive number (either 80h or 81h) is specified in DL. If successful, this function returns with AH set to 00h and Carry Flag cleared.

Note: The diskette system is reset regardless of the drive number specified in DL. To reset the fixed disk controller only, use function AH = 0Dh.

Input/Output

Input:	AH	=	00h
Output:	AH	=	00h No error
		=	Nonzero, Fixed Disk Service Error Code
		=	Fixed Disk Status (40:74h)
	CF	=	0 No error
		=	1 Error

Error conditions

If not successful, the Reset Diskette(s) and Fixed Disk function returns with a nonzero error code in AH.

Differentiating diskette from fixed disk errors

The status of the last Diskette Service operation is stored in Diskette Status (40:41h), and the status of the last Fixed Disk Service operation is stored in Fixed Disk Status (40:74h).

- If the error code returned in AH by this function pertains to the diskette system, then the value of Diskette Status will be equal to the value of AH.
- If the error code in AH pertains to the fixed disk, then the value in Fixed Disk Status will be equal to the value in AH.

Function: AH = 01h Read Disk Status

Description

The Read Disk Status function returns Fixed Disk Status, the error code from the last operation, in the AL register. Before returning to the caller this function sets Fixed Disk Status and AH to zero and clears the Carry Flag.

Input/Output

Input: AH = 01h
DL = Drive number
80h Fixed disk 1
81h Fixed disk 2

Output: AH = 00h
= Fixed Disk Status (40:74h)
AL = Error code for last operation
CF = 0 No error
= 1 Error

Error conditions

If not successful, the Read Disk Status function returns with a nonzero Error Code in AH and the Carry Flag set.

Function: AH = 02h Read Disk Sectors

Description

The Read Disk Sectors function reads the number of sectors specified in AL from the drive specified in DL to a buffer area defined by ES:BX.

Fixed disk head number is indicated in DH. Starting cylinder and sector number are indicated in CH and CL as shown in the Input/Output section below.

If successful, the Read Disk Sectors function returns with the Carry Flag cleared and AH = 00h.

Note: The number of sectors specified in AL must not be zero or greater than 128. Numbers greater than 128 cause a transfer of greater than 64K, and thus force a DMA boundary error.

Input/Output

Input:

AH	=	02h
AL	=	Number of sectors to read
CH	=	Cylinder number (low 8 bits, zero-based)
CL	=	Cylinder/sector number, where:
		Bits 7-6 = cylinder number (high 2 bits)
		Bits 5-0 = sector number
DH	=	Head number (zero-based)
DL	=	Drive number
		80h fixed disk 1
		81h fixed disk 2
ES:BX	=	Pointer to buffer

Output:

AH	=	00h No error
	=	Nonzero, error code
	=	Fixed disk status (40:74h)
CF	=	0 No error
	=	1 Error

continued

Error conditions

If it is unsuccessful, the Read Disk Sectors function returns with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

For example, if an invalid number of sectors is input into AL, then the Read Disk Sectors function returns with the CF set and AH and Fixed Disk Status (40:74h) set to 09h, the Error Code that indicates a DMA boundary error has occurred.

Note: Since the disk drive motor takes a certain amount of time to reach a working speed, a program should try this function three times when an error condition occurs. It should also use the reset function between tries.

Function: AH = 03h Write Disk Sectors

Description

The Write Disk Sectors function writes the number of sectors specified in AL to the drive specified in DL from a buffer area defined by ES:BX.

The fixed disk head number is indicated in DH. Starting cylinder and sector number are indicated in CH and CL as shown in "Input/Output" below.

If successful, the Write Disk Sectors function returns with the Carry Flag cleared and AH = 00h.

Note: The number of sectors specified in AL must not be zero or greater than 128. Numbers greater than 128 cause a transfer of greater than 64K, and thus force a DMA boundary error.

Input/Output

Input:

AH	=	03h
AL	=	Number of sectors to write
CH	=	Cylinder number (low 8 bits, zero-based)
CL	=	Cylinder/sector number, where: Bits 7-6 = cylinder number (high 2 bits) Bits 5-0 = sector number
DH	=	Head number (zero-based)
DL	=	Drive number 80h fixed disk 1 81h fixed disk 2
ES:BX	=	Disk transfer address

Output:

AH	=	Status of operation — Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

Error conditions

If unsuccessful, the Write Disk Sectors function returns with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and Fixed Disk Status (40:74h). For example, if an invalid number of sectors is input into AL, then the Read Disk Sectors function returns with the CF set and AH and Fixed Disk Status (40:74h) set to 09h, the Error Code that indicates a DMA boundary error has occurred.

Function: AH = 04h Verify Disk Sectors

Description

The Verify Disk Sectors function verifies the number of sectors specified in AL on the cylinder, head, and drive specified in CH, CL, and DH respectively. (See "Input/Output" below.)

If successful, the Verify Disk Sectors function returns with the Carry Flag cleared and AH and Fixed Disk Status (40:74h) set to 00h.

Note: This function does not compare data on disk with data in memory. It merely verifies that the specified sectors can be read and that the cyclical redundancy check (CRC) is correct.

Input/Output

Input:

AH	=	04h
AL	=	Number of sectors to verify
CH	=	Cylinder number (low 8 bits, zero-based)
CL	=	Cylinder/sector number, where: Bits 7-6 = cylinder number (high 2 bits) Bits 5-0 = sector number
DH	=	Head number (zero-based)
DL	=	Drive number 80h fixed disk 1 81h fixed disk 2

Output:

AH	=	00h No error
	=	nonzero, Fixed Disk Service Error Code
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

Error conditions

If unsuccessful, the Verify Disk Sectors function returns with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and Fixed Disk Status (40:74h).

Refer to the Error Conditions heading in this chapter for a table of Fixed Disk Service Error Codes.

Function: AH = 05h Format Disk Cylinder

Description

The Format Cylinder function formats the cylinder specified in CH and CL using the head specified in DL.

The caller must provide a table of address markers pointed to by ES:BX. There must be one two-byte table entry for each sector on the cylinder. Table entries must be formatted as shown below:

Byte 1 = good/bad flag
00h is good
80h is bad
Byte 2 = sector number

Input/Output

Input:
AH = 05h
CH = Cylinder number (low 8 bits)
CL = Cylinder/sector number, where:
Bits 7-6 = cylinder number (high 2 bits, zero-based)
Bits 5-0 = sector number
DH = Head number (zero-based)
DL = Drive number
80h fixed disk 1
81h fixed disk 2
ES:BX = Pointer to table of 2 byte address markers

Output:
AH = 00h No error
= nonzero, Fixed Disk Service Error Code
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If unsuccessful, the Format Disk Cylinder function returns with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h). For example, if a time-out error occurs during the execution of this function an error code of 80h is returned in both AH and Fixed Disk Status.

Refer to "Error Conditions" in this chapter for a table of Fixed Disk Service Error Codes.

Functions: AH = 06h - 07h Reserved

Function: AH = 08h Read Drive Parameters

Description

The Read Drive Parameters function returns parameters associated with the fixed disk drive whose number is specified in DL. (Valid numbers are 80h and 81h.)

For drive 80h, the Read Disk Parameters function references the fixed disk parameter table pointed to by the Interrupt 41h vector. For drive 81h, the function references the Interrupt 46h vector.

If successful, the Read Drive Parameters function returns with AL and Fixed Disk Status set to zero. Maximum cylinder number is returned in CH, maximum sector number in CL, maximum head number in DH, and number of fixed disk drives contained in the system is returned in DL.

Input/Output

Input:

AH	=	08h
DL	=	Drive number 80h fixed disk 1 81h fixed disk 2

Output:

AH	=	07h If drive number in DL is invalid
	=	00h If operation is successful
AL	=	00h
CF	=	0 No error
	=	1 Error
CH	=	Maximum usable cylinder number (low 8 bits, zero-based) (00h if AH = 07h)

continued

Function: AH = 08h Read Drive Parameters, Continued

Input/Output, cont'd

Output: CL = Cylinder/sector number, where:
(Bits 7-0 = 00h if AH = 07h)
Bits 7-6 = Maximum usable cylinder number (high 2 bits)
Bits 5-0 = Maximum usable sector number
DH = Maximum usable head number
(zero-based, 00h if AH = 07h)
DL = Number of drives (zero-based, 00h if AH = 07h)

Error conditions

If the drive number specified in DL is not valid, the Read Drive Parameters function returns with AH and Fixed Disk Status (40:74h) set to 07h to indicate that the drive parameter activity failed. AL, CX, and DX are set to zero, the Carry Flag is set, and control is returned to the caller.

Function: AH = 09h Initialize Drive Parameters

Description

The Initialize Drive Parameters function initializes the controller associated with the fixed disk drive whose number is specified in DL. (Valid values are 80h and 81h.)

For drive 80h, the Initialize Drive Parameters function references the fixed disk parameter table pointed to by the Interrupt 41h vector. For drive 81h, the function references the Interrupt 46h vector.

If successful, the Initialize Drive Parameters function returns with AH and Fixed Disk Status (40:74h) set and the Carry Flag cleared.

Input/Output

Input: AH = 09h
DL = Drive number
80h fixed disk 1
81h fixed disk 2

Output: AH = 00h No error
= 07h Operation has failed
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If unsuccessful, the Initialize Drive Parameters function returns with AH and Fixed Disk Status (40:74h) equal to 07h. This indicates that drive parameter activity has failed.

Note: The Initialize Drive Parameters function generates only two valid return codes:

AH = Fixed Disk Status = 00h = Function successful
AH = Fixed Disk Status = 07h = Function unsuccessful, drive parameter activity has failed.

Function: AH = 0Ah Read Long Disk Sectors

Description

The Read Long Disk Sectors function reads one sector from the location on the fixed disk specified in DL. This function also reads the two words of Error Correction Code (ECC) associated with that sector. Cylinder number, sector number, and head number are specified in CH, CL, and DH, respectively (see "Input/Output," below).

If successful, the Read Long Disk Sectors function returns with the Carry Flag cleared and AH and Fixed Disk Status (40:74h) equal to zero.

Notes: This function is normally reserved for diagnostics and should not be used without good reason.

Input/Output

Input:

AH	=	0Ah
AL	=	01h (must always be 1)
CH	=	Cylinder number (low 8 bits, zero-based)
CL	=	Cylinder/sector number, where: Bits 7-6 = cylinder number (high 2 bits) Bits 5-0 = sector number
DH	=	Head number (zero-based)
DL	=	Drive number 80h fixed disk 1 81h fixed disk 2
ES:BX	=	Disk transfer address

Output:

AH	=	00h No error
	=	Nonzero, (Fixed Disk Service Error Code)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

continued

Function: AH = 0Ah Read Long Disk Sectors, Continued

Error conditions

If unsuccessful, the Read Long Disk Sectors function returns with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

For example, if the number of sectors in AL is not equal to 01h, then the Read Long Sector function returns with AH and Fixed Disk Status (40:74h) equal to 01h (the invalid parameter error code) and the Carry Flag set.

Refer to the Error Conditions heading in this chapter for a table of Fixed Disk Service Error Codes.

Function: AH = 0Bh Write Long Disk Sectors

Description

The Write Long Disk Sectors function writes one sector to the location on the fixed disk specified in DL from the buffer pointed to by ES:BX. The function also writes the two words of Error Correction Code (ECC) associated with the specified sector.

Cylinder number, sector number, and head number are specified in CH, CL, and DH, respectively (see Input/Output below).

If successful, the Write Long Sectors function returns with the Carry Flag cleared and AH and Fixed Disk Status (40:74h) equal to zero.

Additional information

- The caller must indicate the number of long sectors to read in AL. Since the Read Long Disk Sectors function is limited to one sector, any AL value other than 01h results in an invalid parameter error condition.
- Along with function AH = 0Ah, this function is normally reserved for diagnostics and should not be used without good reason.

continued

Function: AH = 0Bh Write Long Disk Sectors, Continued

Input/Output

Input:

AH	=	0Bh
AL	=	01h (must always be 1)
CH	=	Cylinder number (low 8 bits, zero-based)
CL	=	Cylinder/sector number, where: Bits 7-6 = cylinder number (high 2 bits) Bits 5-0 = sector number
DH	=	Head number (zero-based)
DL	=	Drive number 80h fixed disk 1 81h fixed disk 2
ES:BX	=	Disk transfer address

Output:

AH	=	00h No error
	=	nonzero, (Fixed Disk Service Error Code)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

Error conditions

If unsuccessful, the Write Long Disk Sectors function returns to the caller with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

For example, if the number of sectors in AL is not equal to 01h, then the Read Long Sector function returns with AH and Fixed Disk Status (40:74h) equal to 01h (the invalid parameter error code) and the Carry Flag set.

Refer to the Error Conditions heading in this chapter for a table of Fixed Disk Service Error Codes.

Function: AH = 0Ch Seek to Cylinder

Description

The Seek to Cylinder function positions the disk read/write head over the cylinder specified in CH and CL (see Input/Output below). The fixed disk drive number must be specified in DL.

If successful, the Seek to Cylinder function returns with the Carry Flag cleared and AH and Fixed Disk Status (40:74h) equal to zero.

Note: The Read Disk Sectors, Write Disk Sectors, Read Long Disk Sectors and Write Long Disk Sectors functions have a seek operation implicitly built into them and do not require a prior call of this function.

Input/Output

Input:

AH	=	0Ch
CH	=	Cylinder number (low 8 bits, zero-based)
CL	=	Cylinder/sector number, where:
		Bits 7-6 = cylinder number (high 2 bits)
		Bits 5-0 = sector number
DH	=	Head number (zero-based)
DL	=	Drive number
		80h fixed disk 1
		81h fixed disk 2

Output:

AH	=	00h No error
	=	Nonzero, (Fixed Disk Service Error Code)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

Error conditions

If unsuccessful, the Seek to Cylinder function returns to the caller with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

Refer to "Error Conditions" in this chapter for a table of Fixed Disk Service Error Codes.

Function: AH = 0Dh Alternate Reset Fixed Disk

Description

The Alternate Reset Fixed Disk is identical to the Reset Diskette(s) and Fixed Disk function (see AH = 00h) except that the diskette system is not reset.

The caller must specify the fixed disk drive to reset in DL. The Alternate Reset Fixed Disk function reinitializes the fixed disk controller and places the specified fixed disk drive in a known state by resetting the fixed disk parameters and by recalibrating the read/write head positions.

If successful, the Alternate Reset Fixed Disk function returns with the Carry Flag cleared and AH and Fixed Disk Status (40:74h) equal to zero.

Input/Output

Input:	AH	=	0Dh
	DL	=	Drive number
			80h fixed disk 1
			81h fixed disk 2
Output:	AH	=	00h No error
		=	Nonzero, (Fixed Disk Service Error Code)
		=	Fixed Disk Status (40:74h)
	CF	=	0 No error
		=	1 Error

A special case

The ABIOS does not inform the CBIOS of the fixed disk hardware's state when control is passed from the ABIOS to the CBIOS. As a result, it is possible for the ABIOS to put the fixed disk hardware into a state not recognized by the CBIOS.

To avoid confusion, then, care must be taken to reset the fixed disk system each time the CBIOS Fixed Disk Service is re-entered after **any** ABIOS call. The following rules apply:

1. Do not invoke a CBIOS function if there is already an outstanding ABIOS function request.
2. If the last function call was to ABIOS, invoke CBIOS Fixed Disk Service function AH = 0Dh Alternate Fixed Disk Reset.

---

continued

Reference

For more information on the relationship between the CBIOS and ABIOS fixed disk services, turn to the heading "Rules for Using CBIOS and ABIOS Fixed Disk Services" in this chapter.

Error conditions

If it is not successful, the Alternate Reset Disk function returns to the caller with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

Refer to the Error Conditions heading in this chapter for a table of Fixed Disk Service Error Codes.

Function: AH = 0Eh Diagnostics 1: Read Test Buffer

Description

The Diagnostics 1, Read Test Buffer, function reads a test buffer from the fixed disk adapter into the diagnostics buffer specified in ES:BX. Data is not read from the actual physical disk drive.

If successful, the Diagnostics 1 function returns with the Carry Flag (CF) cleared and AH and Fixed Disk Status (40:74h) set to zero.

Input/Output

Input:

AH	=	0Eh Diagnostics 1, read test buffer
DL	=	Drive number
		80h fixed disk 1
		81h fixed disk 2
ES:BX	=	Pointer to diagnostic buffer

Output:

AH	=	00h No error
	=	Nonzero, (Fixed Disk Service Error Code)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

Error conditions

If it is not successful, the Diagnostics 1 function returns to the caller with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

Refer to the Error Conditions heading in this chapter for a table of Fixed Disk Service Error Codes.

Function: AH = 0Fh Diagnostics 2: Write Test Buffer

Description

The Diagnostics 2, Write Test Buffer, function writes a test buffer to the adapter from the diagnostics buffer specified in ES:BX. Data is not written to the actual physical disk drive.

If successful, the Diagnostics 2 function returns with the Carry Flag (CF) cleared and AH and Fixed Disk Status (40:74h) set to zero.

Input/Output

Input: AH = 0Fh Diagnostics 2, write test buffer
DL = Drive number
80h fixed disk 1
81h fixed disk 2
ES:BX = Pointer to diagnostic buffer

Output: AH = 00h No error
= Nonzero, (Fixed Disk Service Error Code)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If it is not successful, the Diagnostics 2 function returns to the caller with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

Refer to the Error Conditions heading in this chapter for a table of Fixed Disk Service Error Codes.

Function: AH = 10h Test for Drive Ready

Description

The Test for Drive Ready function determines if the drive specified in DL is ready and can process a command.

If successful, the Test for Drive Ready function returns to the caller with the Carry Flag (CF) cleared and AH and Fixed Disk Status (40:74h) set to zero.

Input/Output

Input: AH = 10h
DL = Drive number
80h fixed disk 1
81h fixed disk 2

Output: AH = 00h No error
= Nonzero, (Fixed Disk Service Error Code)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If it is not successful, the Test for Drive Ready function returns to the caller with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

Refer to the Error Conditions heading in this chapter for a table of Fixed Disk Service Error Codes.

Function: AH = 11h Recalibrate Drive

Description

The Recalibrate Drive function repositions head 0 over cylinder 0 of the drive specified in DL.

If successful, the Recalibrate Drive function returns with the Carry Flag (CF) cleared and AH and Fixed Disk Status (40:74h) set to zero.

Input/Output

Input: AH = 11h
DL = Drive number
80h fixed disk 1
81h fixed disk 2

Output: AH = 00h No error
= Nonzero, (Fixed Disk Service Error Code)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If it is not successful, the Recalibrate Drive function returns to the caller with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

Refer to the Error Conditions heading in this chapter for a table of Fixed Disk Service Error Codes.

Function: AH = 12h - 14h Reserved

Function: AH = 15h Read Disk Type

Description

The Read Disk Type function returns the number of 512-byte blocks on the fixed disk if the drive specified in DL is valid. (Valid values are 80h and 81h.)

If successful, the high word 512-byte block amount is returned in CX, and the low word amount is returned in DX. Fixed Disk Status (40:74h) and the Carry Flag (CF) are set to zero, and control is returned to the caller.

Input/Output

Input: AH = 15h
DL = Drive number
80h fixed disk 1
81h fixed disk 2

Output: AH = 00h No drive installed
= 03h Fixed disk was accessed
= Nonzero (Fixed Disk Service Error Code)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

AH reports drive number validity

For drive 80h, the Read Disk Type function references the fixed disk parameter table pointed to by the Interrupt 41h vector. For drive 81h, the function references the Interrupt 46h vector.

Unlike most other Fixed Disk Service functions, in the Read Disk Type function the value stored in AH is not equal to the value stored in Fixed Disk Status (40:74h). Instead, AH reports whether or not the drive number input into DL is valid (i.e., whether or not the drive is installed).

Error conditions

If the drive number specified in DL is not valid, the Read Disk Type function sets AH, AL, CX, DX, the Carry Flag (CF), and Fixed Disk Status (40:74h) to zero, and returns control to the caller.

Functions: AH = 06h - 07h Reserved

Function: AH = 19h Park Heads

Description

The Park Heads function positions the heads of the fixed disk read/write mechanism on the landing zone of the disk.

For drive 80h, the landing zone cylinder is specified in the fixed disk parameter table pointed to by the Interrupt 41h vector. For drive 81h, the landing zone cylinder is specified in the parameter table pointed to by the Interrupt 46h vector.

Input/Output

Input: AH = 19h
DL = Drive number
80h fixed disk 1
81h fixed disk 2

Output: AH = 00h No error
= Nonzero (Fixed Disk Service Error Code)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If it is not successful, the Test for Drive Ready function returns to the caller with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

Refer to the Error Conditions heading in this chapter for a table of Fixed Disk Service Error Codes.

Function: AH = 1Ah Format ESDI Drive

Description

The Format ESDI Drive function initializes the disk sector and track address fields on a drive attached to ESDI Fixed Disk Drive Adapter A. Normally, ESDI drives require an option ROM.

This function is a low level format that prepares the fixed disk for physical read/write operations at the sector level. The disk then has to be partitioned with the FDISK command and the FORMAT command used to give the disk a high level format and install a file system on it.

This function is not implemented in every BIOS. The Phoenix CBIOS supports ESDI low-level format through the Phoenix Reference Diskette.

> **Warning**
>
> The formatting process will destroy all information on the fixed disk.

Input/Output

Input:

AH = 1Ah
AL = Relative block address (RBA) defect table block count
= 0 No RBA table
> 0 RBA table is used
CL = Format modifier bits, where
Bit 0 = ignore primary defect map
Bit 1 = ignore secondary defect map
Bit 2 = update secondary defect map
Bit 3 = do extended surface analysis
Bit 4 = generate periodic interrupt
Bits 5–7 = reserved (= 0)
DL = Drive number
80h fixed disk 1
81h fixed disk 2
ES:BX = Pointer to RBA table

Output:

AH = 00h No error
= Nonzero (Fixed Disk Service Error Code)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

continued

Error conditions

If it is not successful, the Format ESDI Drive function returns to the caller with the Carry Flag (CF) set and the appropriate Fixed Disk Service error code in both AH and in Fixed Disk Status (40:74h).

Refer to the Error Conditions heading in this chapter for a table of Fixed Disk Service Error Codes.

Additional information

- When bit 4 of CL is set, INT 15h function AH = 0Fh, subfunction AL = phase code is called after each cylinder is formatted or analyzed. The phase codes are:

  0 = reserved
  1 = surface analysis
  2 = formatting

  Refer to INT 15h function AH = 0Fh for further information.
- When bit 2 of CL is set, the secondary defect map for the drive is updated so it reflects any errors found during surface analysis. The secondary defect map is replaced when both bit 1 and bit 2 are replaced.
- To do an extended surface analysis, first format the disk by calling this function with bit 3 cleared, then do the analysis by calling the function with bit 3 set.
- For other types of fixed disk adapters, this function is reserved.

12

Chapter 12

INT 14h Serial Communications Service

Overview

Description

The CBIOS Serial Communications Service performs RS-232-C character I/O on IBM-compatible serial port adapters.

How the Serial Communications Service is invoked

Software INT 14h invokes the CBIOS Serial Communications Service.

The INT 14h vector resides at address 00:50h in the Interrupt Vector Table. The CBIOS initializes the INT 14h vector to address F000:E739h, the IBM-compatible entry point.

continued

Summary of Serial Communications functions

The CBIOS Serial Communications Service contains six functions. Individual functions are selected via the AH register.

The table below summarizes the Serial Communications Service functions.

Function	Description
00h	Initialize Serial Communications Port
01h	Send Character
02h	Receive Character
03h	Read Serial Port Status
04h	Extended Initialize
05h	Extended Serial Port Control
06h-FFh	Reserved

In this chapter

This chapter focuses on the CBIOS Serial Communications Service. The following topics are discussed:

- Hardware Environment
- Serial Communications I/O Ports
- System RAM Data
- ROM BIOS Data
- Error Conditions
- Serial Communications Functions

NS 16550 Serial Communications Controller

The CBIOS Serial Communications Service assumes a National Semiconductor NS 16550 serial communications controller (or compatible) is in place. The NS 16550 contains a programmable baud rate generator that supports baud rates from 50 baud to 19,200 baud. The NS 16550 also supports 5-, 6-, 7-, and 8-bit characters with 1-, 1.5-, or 2-stop bits operating in even, odd, and no parity modes.

Serial port addresses/interrupt levels

The following table lists the serial port base addresses.

Serial Port Number	Base Address	Interrupt Level
1	03F8h	4
2	02F8h	3
3	3220h	3
4	3228h	3
5	4220h	3
6	4228h	3
7	5220h	3
8	5228h	3

I/O Address	Read/Write Status	Description
03F8h	W	Serial 1, transmitter holding register, where: Bits 7-0 = Data bits 7-0, respectively, when Divisor Latch Access Bit (DLAB) = 0
03F8h	R	Serial 1, receiver buffer register, where: Bits 7-0 = Data bits 7-0, respectively, when DLAB = 0
03F8h	R/W	Serial 1, divisor latch, low byte, where: Bits 7-0 = Bits 7-0 of divisor, when DLAB = 1
03F9h	R/W	Serial 1, divisor latch, high byte, where: Bits 7-0 = Bits 15-8 of divisor, when DLAB = 1
03F9h	R/W	Serial 1 Interrupt ID Register, where: Bits 7-3 = 0 Reserved Bits 2-1 = 00b Modem Status Interrupt 01b Transmitter Holding Register Empty Interrupt 10b Received Data Available Register Interrupt 11b Receiver Line Status Interrupt Bit 0 = 0 Interrupt Pending
03FAh	R	Serial 1 Interrupt ID Register, where: Bits 7-3 = 0 Reserved Bits 2-1 = 00b Modem Status Interrupt 01b Transmitter Holding Register Empty Interrupt 10b Received Data Available Register Interrupt 11b Receiver Line Status Interrupt Bit 0 = 0 Interrupt Pending

continued

I/O Address	Read/Write Status	Description
03FAh	W	Serial 1, FIFO Control Register, where: Bits 7-6 = Receiver FIFO register trigger 00b = 1 byte 01b = 4 bytes 10b = 8 bytes 11b = 14 bytes Bits 5-3 = 0 Reserved Bit 2 = Transmitter FIFO register reset 1 = transmit FIFO register cleared, counter cleared, bit is self-clearing Bit 1 = Receiver FIFO register reset 1 = receiver FIFO register cleared, counter cleared, bit is self-clearing Bit 0 = FIFO enable 1 = receiver and transmitter FIFOs enabled, must be 1 to program FIFO registers 0 = clears receive and transmit FIFO registers, enters character mode
03FBh	R/W	Serial 1, Line Control Register, where: Bit 7 = 0 Receiver Buffer, Transmitter Holding, or Interrupt Enable Registers Access = 1 Divisor Latch Access Bit 6 = 1 Set Break Enabled Bit 5 = Stick Parity Bit 4 = Even Parity Select Bit 3 = Parity Enable Bit 2 = 0 1 Stop Bit = 1 0 Stop Bits Bits 1-0 = 00b 5 Bit Word Length 01b 6 Bit Word Length 10b 7 Bit Word Length 11b 8 Bit Word Length
03FCh	R/W	Serial 1, Mode Control Register, where: Bits 7-5 = 0 Reserved Bit 4 = 1 Loopback Mode Bit 3 = 1 Enable OUT2 Interrupt Bit 2 = 1 Force OUT1 Active Bit 1 = 1 Force Request-To-Send Active Bit 0 = 1 Force Data-Terminal-Ready Active

continued

I/O Address	Read/Write Status	Description
03FDh	R/W	Serial 1, Line Status Register, where: Bit 7 = 0 Reserved Bit 6 = 1 Transmitter Shift and Holding Registers Empty Bit 5 = 1 Transmitter Holding Register Empty Bit 4 = 1 Break Interrupt Bit 3 = 1 Framing Error Bit 2 = 1 Trailing Edge Ring Indicator Bit1 = 1 Overrun Error Bit 0 = 1 Data Ready
03FEh	R	Serial 1, Modem Status Register, where: Bit 7 = 1 Data Carrier Detect Bit 6 = 1 Ring Indicator Bit 5 = 1 Data Set Ready Bit 4 = 1 Clear To Send Bit 3 = 1 Delta Data Carrier Detect Bit 2 = 1 Trailing Edge Ring Indicator Bit 1 = 1 Delta Data Set Ready Bit 0 = 1 Delta Clear To Send
03FFh	R	Serial 1, Scratch Register
02F8h	W	Serial 2, transmitter holding register, where: Bits 7-0 = Data bits 7-0, respectively, when DLAB = 0
02F8h	R	Serial 2, receiver buffer register, where: Bits 7-0 = Data bits 7-0, respectively, when DLAB = 0
02F8h	R/W	Serial 2, divisor latch, low byte, where: Bits 7-0 = Bits 7-0 of divisor, when DLAB = 1
02F9h	R/W	Serial 2, divisor latch, high byte, where: Bits 7-0 = Bits 15-8 of divisor, when DLAB = 1
02F9h	R/W	Serial 2, interrupt enable register, where: Bits 7-4 = 0 Reserved Bit 3 = 1 Modem-status interrupt enable Bit 2 = 1 Receiver-line-status interrupt enable Bit 1 = 1 Transmitter-holding-register empty interrupt enable Bit 0 = 1 Received-data-available interrupt enable when DLAB = 0

continued

I/O Address	Read/Write Status	Description
02FAh	R	Serial 2, interrupt identification register, where: Bits 7-3 = 0 Reserved Bits 2-1 = 00b Modem status interrupt = 01b Transmitter holding register empty interrupt = 10b Received data available register interrupt = 11b Receiver line status interrupt Bit 0 = 0 Interrupt pending
02FAh	W	Serial 2, FIFO control register, where: Bits 7-6 = Receiver FIFO register trigger 00b = 1 byte 01b = 4 bytes 10b = 8 bytes 11b = 14 bytes Bits 5-3 = 0 Reserved Bit 2 = Transmitter FIFO register reset 1 = transmit FIFO register cleared, counter cleared, bit is self-clearing Bit 1 = Receiver FIFO register reset 1 = receiver FIFO register cleared, counter cleared, bit is self-clearing Bit 0 = FIFO enable 1 = receiver and transmitter FIFOs enabled, must be 1 to program FIFO registers 0 = clears receiver and transmits FIFO registers, enters character mode
02FBh	R/W	Serial 2, line control register, where: Bit 7 = 1 Divisor latch access = 0 Receiver buffer, transmitter holding, or interrupt enable registers access Bit 6 = 1 Set break enable Bit 5 = Stick parity Bit 4 = Even parity select Bit 3 = Parity enable Bit 2 = Number of stop bits Bits 1-0 = 00b Word length is 5 bits = 01b Word length is 6 bits = 10b Word length is 7 bits = 11b Word length is 8 bits
02FCh	R/W	Serial 2, modem control register, where: Bits 7-5 = 0 Reserved Bit 4 = 1 Loopback mode Bit 3 = 1 Enable OUT2 interrupt Bit 2 = 1 Force OUT1 active Bit 1 = 1 Force request-to-send active Bit 0 = 1 Force data-terminal-ready active

continued

I/O Address	Read/Write Status	Description
02FDh	R	Serial 2, line status register, where: Bit 7 = 0 Reserved Bit 6 = 1 Transmitter shift and holding registers empty Bit 5 = 1 Transmitter holding register empty Bit 4 = 1 Break interrupt Bit 3 = 1 Framing error Bit 2 = 1 Parity error Bit 1 = 1 Overrun error Bit 0 = 1 Data ready
02FEh	R	Serial 2, modem status register, where: Bit 7 = 1 Data carrier detect Bit 6 = 1 Ring indicator Bit 5 = 1 Data set ready Bit 4 = 1 Clear to send Bit 3 = 1 Delta to carrier detect Bit 2 = 1 Trailing edge ring indicator Bit 1 = 1 Delta data set ready Bit 0 = 1 Delta clear to send
02FFh	R/W	Serial 2, scratch register

System RAM Data

Introduction

The Serial Communications Services make use of control information located in the BIOS data area (address 40:00 - 40:100). The table below describes those areas.

Location	Length	Description
00h	4 Words	I/O address of up to 4 asynchronous communications adapters. One word for each asynchronous communications adapter
7Ch	4 Bytes	Serial (RS232) time-out table for serial ports 0 through 3. One byte per adapter.

How the Serial Port Table is initialized

As it identifies each RS-232-C communications line, the CBIOS power-on self test (POST) places its corresponding base port address into the Serial Port Table *(40:00h)*. In systems containing less than four serial ports, POST will initialize the base address for nonexistent ports to 0. All entries into the Serial Port Table are sequential. POST never writes a 0 into the Serial Port Table between two valid base address entries.

Baud rate initialization table

The Baud Rate Initialization Table is located at F000:E729h in the ROM BIOS. The table is structured as follows:

Baud Rate	Divisor
110	0417h
150	0300h
300	0180h
600	00C0h
1200	0060h
2400	0030h
4800	0018h
9600	000Ch
19200	0006h

How baud rate divisor is calculated

The input frequency to the device is 1.8432 MHz. The values in the table are calculated as follows:

1,843,200 / 16 = 115200 / Baud Rate = Divisor

For example, a baud rate of 2400 has a divisor of 115200/2400, which equals 48 decimal, 30 hex.

Error Conditions

Introduction

The Serial Communications Service detects two kinds of errors:

- Parameter-related errors
- Time-out errors

Parameter-related errors

The following parameter-related conditions are checked upon entry into each Serial Communications Service function.

- The function number specified in AH must fall within the range 0-5.
- The subfunction number specified in AL (if any) must fall within the range 0-1.
- The serial port specified in DX must fall within the range of 0-3.
- The serial port specified in DX must exist in hardware.

If any of the conditions above are not true, the Serial Communications Service does not perform the requested function and returns with all registers preserved.

Time-out errors

A time-out error occurs when either a read or a write of a specified communications line was unable to occur.

The Serial Communications Service read and write functions test the Line Status register and the Modem Status register. When a time-out error occurs, the contents of the status register being tested are returned with bit 7 "time-out error" set.

Function: AH = 00h Initialize Serial Communications Port

Description

The Initialize Serial Communications Port function initializes the selected adapter card from the Baud Rate, Parity, Stop Bit, and Word Length parameters specified in the AL register. The function returns with the Modem Status register and the line Status register in AL and AH, respectively. All other registers are preserved.

9600 baud limit

The Initialize Serial Communications Port function is unable to initialize a baud rate above 9600 baud because of input parameter size limitations. To initialize the adapter card at a higher rate, use function 04h.

Input/Output

Input: AH = 00h
AL = Serial port initialization parameters, where:
Bits 7–5 = Baud rate, where:
000b = 110 baud
001b = 150 baud
010b = 300 baud
011b = 600 baud
100b = 1200 baud
110b = 4800 baud
111b = 9600 baud
Bits 4–3 = Parity, where:
00b = None
01b = Odd
10b = None
11b = Even
Bit 2 = Number of stop bits, where:
0b = 1 Stop bit
1b = 2 Stop bits
Bits 1–0 = Character size, where:
10b = 7-bit characters
11b = 8-bit characters
DX = Serial port number (valid values are 0, 1, 2, 3, where 0 = COM1, 1 = COM2, etc.)

continued

Input/Output, cont'd

Output: AH = Line Status, where:
- Bit 7 = 0 Reserved
- Bit 6 = 1 Transmitter shift and holding register empty
- Bit 5 = 1 Transmitter holding register empty
- Bit 4 = 1 Break interrupt
- Bit 3 = 1 Framing error
- Bit 2 = 1 Parity error
- Bit 1 = 1 Overrun error
- Bit 0 = 1 Data ready

AL = Modem Status, where:
- Bit 7 = 1 Data carrier detect
- Bit 6 = 1 Ring indicator
- Bit 5 = 1 Data set ready
- Bit 4 = 1 Clear to send
- Bit 3 = 1 Delta data carrier select
- Bit 2 = 1 Trailing edge ring indicator
- Bit 1 = 1 Delta data set ready
- Bit 0 = 1 Delta clear to send

Function: AH = 01h Send Character

Description

The Send Character function transmits the character supplied in AL over the communication line specified in DX.

If successful, this function returns with the contents of the Line Status Register in AH, Bit 7 = 0. If unsuccessful, AH, Bit 7 = 1.

Input/Output

Input:

AH = 01h
AL = Character
DX = Serial Port Number (valid values are 0, 1, 2, 3, where 0 = COM1, 1 = COM2, etc.

Output:

AH = Line Status Register, where:
Bit 7 = 0 No time-out error has occurred
Bit 6 = 1 Transmitter shift and holding register empty
Bit 5 = 1 Transmitter holding register empty
Bit 4 = 1 Break interrupt
Bit 3 = 1 Framing error
Bit 2 = 1 Parity error
Bit 1 = 1 Overrun error
Bit 0 = 1 Data ready
AL = Character sent (unchanged)

Error conditions

When a time-out error occurs during the execution of the Send Character function, the value returned in AH can reflect the state of either the Modem Status Register or of the Line Status Register. However, since bit 7 of AH is used to report that an error has occurred, it is not available to report a time-out error. For this reason, if this function or INT 14h function 02h reports an error, it is preferable to use INT 14h function 03h, which gives a complete status report, rather than the less-complete status bits returned with the error bit in functions 01h and 02h.

Function: AH = 02h Receive Character

Description

The Receive Character function receives one character from the serial port specified in the DX register.

If successful, this function returns with the character received in the AL register. The contents of the Line Status Register is returned in the AH register.

Input/Output

Input: AH = 02h
DX = Serial port number (Valid values are 0, 1, 2, 3, where 0 = COM1, 1 = COM2, etc.)

Output: AH = Line Status Register, where:
Bit 7 = 0 No time-out error has occurred
Bit 6 = 1 Transmitter shift and holding register empty
Bit 5 = 1 Transmitter holding register empty
Bit 4 = 1 Break interrupt
Bit 3 = 1 Framing error
Bit 2 = 1 Parity error
Bit 1 = 1 Overrun error
Bit 0 = 1 Data ready
AL = Character received

Error conditions

If a time-out error occurs during the execution of the Receive Character function, the value returned in AH can reflect the state of either the Modem Status Register or of the Line Status Register. However, since bit 7 of AH is used to report that an error has occurred, it is not available to report a time-out error. For this reason, if this function or INT 14h function 01h reports an error, it is preferable to use INT 14h function 03h, which gives a complete status report, rather than the less-complete status bits returned with the error bit in functions 01h and 02h.

Function: AH = 03h Read Serial Port Status

Description

The Read Serial Port Status function returns the current Modem Status in the AL register and the current Line Status in the AH register. All other registers are preserved.

Input/Output

Input: AH = 03h
DX = Serial port number (Valid values are 0, 1, 2, 3, where 0 = COM1, 1 = COM2, etc.)

Output: AH = Line Status, where:
Bit 7 = 0 Reserved
Bit 6 = 1 Transmitter shift and holding register empty
Bit 5 = 1 Transmitter holding register empty
Bit 4 = 1 Break interrupt
Bit 3 = 1 Framing error
Bit 2 = 1 Parity error
Bit 1 = 1 Overrun error
Bit 0 = 1 Data ready

AL = Modem status, where:
Bit 7 = 1 Data carrier detect
Bit 6 = 1 Ring indicator
Bit 5 = 1 Data set ready
Bit 4 = 1 Clear to send
Bit 3 = 1 Delta data carrier detect
Bit 2 = 1 Trailing edge ring indicator
Bit 1 = 1 Delta data set ready
Bit 0 = 1 Delta clear to send

Function: AH = 04h Extended Initialize

Description

The Extended Initialize function does an extended initialization of the adapter card selected in the DX register based on the Break, Parity, Stop Bit, Word Length, and Baud Rate values placed in the BH, BL, CH, and CL registers. This function is a superset of function 00h.

The function returns with the Modem Status Register and the Line Status Register in the AL and AH registers, respectively.

This function must be used to initialize baud rates greater than 9600.

Input/Output

Input:

AH = 04h
AL = Break, where:
- 00h = No break
- 01h = Break

BH = Parity, where:
- 00h = None
- 01h = Odd
- 02h = Even
- 03h = Stick parity odd
- 04h = Stick parity even

BL = Stop Bit, where:
- 00h = One
- 01h = Two if 6-, 7-, or 8- bit word length
  One-and-one-half if 5-bit word length

CH = Word length, where:
- 00h = 5 Bits
- 01h = 6 Bits
- 02h = 7 Bits
- 03h = 8 Bits

continued

Input/Output, cont'd

Input: (cont'd)

CL = Baud rate, where:
- 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

DX = RS-232-C communications line to use (Valid values are 0, 1, 2, 3, where 0 = COM1, 1 = COM2, etc.)

Output:

AH = Line Status
- Bit 7 = 0 Reserved
- Bit 6 = 1 Transmitter shift and holding register empty
- Bit 5 = 1 Transmitter holding register empty
- Bit 4 = 1 Break interrupt
- Bit 3 = 1 Framing error
- Bit 2 = 1 Parity error
- Bit 1 = 1 Overrun error
- Bit 0 = 1 Data ready

AL = Modem Status
- Bit 7 = 1 Time-out error
- Bit 6 = 1 Ring indicator
- Bit 5 = 1 Data set ready
- Bit 4 = 1 Clear to send
- Bit 3 = 1 Delta data carrier detect
- Bit 2 = 1 Trailing edge ring indicator
- Bit 1 = 1 Delta data set ready
- Bit 0 = 1 Delta clear to send

Function: AH = 05h Extended Serial Port Control

Introduction

The Modem Control Register is an 8-bit register that controls data exchange with the modem, data set, or peripheral device emulating a modem.

The Extended Serial Port Control function will either read from or write to the Modem Control Register depending on the parameter input into the AL register. The two subfunctions and their AL register parameter values are listed below:

- AL = 00h Read the Modem Control Register
- AL = 01h Write to the Modem Control Register

continued

Subfunction: AL = 00h read the modem control register

This subfunction reads the NS 16550 (or compatible) Modem Control Register and returns the value in BL. The Modem Status Register is returned in AL.

Input/Output

Input: AH = 05h Extended serial port control
AL = 00h Read the Modem Control Register
DX = RS-232-C communications line to use (Valid values are 0, 1, 2, 3, where 0 = COM1, 1 = COM2, etc.)

Output: AH = Line Status, where:
Bit 7 = 0 Reserved
Bit 6 = 1 Transmitter shift and holding register empty
Bit 5 = 1 Transmitter holding register
Bit 4 = 1 Break interrupt
Bit 3 = 1 Framing error
Bit 2 = 1 Parity error
Bit 1 = 1 Overrun error
Bit 0 = 1 Data ready

AL = Modem status, where:
Bit 7 = 1 Data carrier detect
Bit 6 = 1 Ring indicator
Bit 5 = 1 Data set ready
Bit 4 = 1 Clear to send
Bit 3 = 1 Delta data carrier detect
Bit 2 = 1 Trailing edge ring indicator
Bit 1 = 1 Delta data set ready
Bit 0 = 1 Delta clear to send

BL = Modem Control Register, where:
Bits 7-5 = 0 Reserved
Bit 4 = 1 Loopback mode
Bit 3 = 1 Enable OUT2 interrupt
Bit 2 = 1 Force OUT1 active
Bit 1 = 1 Force request-to-send active
Bit 0 = 1 Force data-terminal-ready active

All other registers are preserved.

continued

Subfunction: AL = 01h write modem control register

This subfunction writes the value placed in BL to the NS 16550 Modem Control Register. The subfunction returns with the value written to the Modem Control Register in BL, the Modem Status register in AL, and the Line Status register in AH.

Input/Output

Input:
AH = 05h Extended Serial Port Control
AL = 01h Write Modem Control Register
BL = Modem Control Register, where:
Bits 7–5 = 0 Reserved
Bit 4 = 1 Loopback mode
Bit 3 = 1 Enable OUT2 interrupt
Bit 2 = 1 Force OUT1 active
Bit 1 = 1 Force request-to-send active
Bit 0 = 1 Force data-terminal-ready active
DX = RS-232-C communications line to use (Valid values are 0, 1, 2, 3, where 0 = COM1, 1 = COM2, etc.)

Output:
AH = Line Status, where:
Bit 7 = 0 Reserved
Bit 6 = 1 Transmitter shift and holding register empty
Bit 5 = 1 Transmitter holding register
Bit 4 = 1 Break interrupt
Bit 3 = 1 Framing error
Bit 2 = 1 Parity error
Bit 1 = 1 Overrun error
Bit 0 = 1 Data ready
AL = Modem status, where:
Bit 7 = 1 Data carrier detect
Bit 6 = 1 Ring indicator
Bit 5 = 1 Data set ready
Bit 4 = 1 Clear to send
Bit 3 = 1 Delta data carrier detect
Bit 2 = 1 Trailing edge ring indicator
Bit 1 = 1 Delta data set ready
Bit 0 = 1 Delta clear to send

---

continued

Function: AH = 05h Extended Serial Port Control, Continued

Subfunction: AL = 01h, cont'd

Output: (cont'd)

BL = Modem Control Register, where:

Bits 7–5	=	0 Reserved
Bit 4	=	1 Loopback mode
Bit 3	=	1 Enable OUT2 interrupt
Bit 2	=	1 Force OUT1 active
Bit 1	=	1 Force request-to-send active
Bit 0	=	1 Force data-terminal-ready active

All other registers are preserved.

Functions: AH = 06h – FFh Reserved

13

Chapter 13
INT 15h System Services

Overview

Introduction

INT 15h, System Services, provides several services:

- **ABIOS initialization routines**

  The CBIOS System Services contain two functions that are called during ABIOS initialization. These services are reserved exclusively for operating systems that employ the ABIOS.

- **Multitasking hooks**

  The CBIOS provides six hooks for use by multitasking operating systems. In a DOS environment (which is not a multitasking operating system) these six functions do nothing except return to the caller. A multitasking operating system would normally intercept these function calls by replacing the INT 15h vector and processing these function calls itself.

- **Joystick support**

  Function 84h supports up to two joysticks. Function 84h has two subfunctions: read current switch settings and read resistive inputs.

continued

Introduction, cont'd

- **Wait routines**

  INT 15h provides two wait functions: Function 83h, Set Event Wait Interval, and Function 86h, Wait. Function 86h does not return control to the caller until a specified interval is completed. Function 83h returns control to the caller.

- **Protected mode support**

  The CBIOS System Services provides limited protected mode support through two functions: Function 87h, Move Block, and function 89h, Switch Processor to Protected Mode.

  The ABIOS provides full protected mode, bimodal, and real mode addressing support.

- **Pointing device interface**

  INT 15h function C2h provides complete mouse support.

- **System information**

  Function C0h, Return System Configuration Parameters, Function C1h, Return Extended BIOS Data Area Segment Address, Function C3h, Watchdog time-out, and Function C4h, Programmable Option Select, provide detailed system services.

How INT 15h services are invoked

INT 15h invokes the CBIOS System Services.

The INT 15h vector resides at Interrupt table address 00:3Ch. To insure compatibility, the CBIOS initializes the INT 15h vector to F000:F859h.

Individual System Services functions are selected via the AH register. Subfunctions are selected via the AL, DX, or BH registers.

---

continued

Summary of INT 15h system services

The table below lists the System Services functions and subfunctions.

Note: Functions AH = 80h, 81h, 82h, 85h, 90h, and 91h (i.e. those marked with an asterisk in the table below) are multitasking hooks. A multitasking operating system would normally intercept these function calls by replacing the INT 15h vector and processing these function calls itself. In DOS (which is not a multitasking operating system) these six functions do nothing except return immediately to the caller with the Carry Flag cleared.

Function	Description
00h–03h	Reserved
04h	Build System Parameters Table
05h	Build Initialization Table
06h–0Eh	Reserved
0Fh	Format ESDI Drive Periodic Interrupt
10h–20h	Reserved
21h	Power-On Self Test Error Log AL = 00h Read POST Error Log AL = 01h Write Error Code to POST Error Log
22h–4Eh	Reserved
4Fh	Keyboard Intercept
50h–7Fh	Reserved
80h	Device Open*
81h	Device Close*
82h	Program Termination*
83h	Set Event Wait Interval AL = 00h Set Wait AL = 01h Cancel Wait
84h	Joystick Support DX = 00h Read Current Switch Settings DX = 01h Read Resistive Inputs
85h	System Request Key*
* Multitasking hooks (described in note above)	

continued

Summary of INT 15h System Services, cont'd

AH Value	Description
86h	Wait
87h	Move Block
88h	Read Extended Memory Size
89h	Switch Processor to Protected Mode
8Ah to 8Fh	Reserved
90h	Device Busy Loop*
91h	Interrupt Complete*
92h to BFh	Reserved
C0h	Return System Configuration Parameters
C1h	Return Extended BIOS Data Area Segment Address
C2h	Pointing Device Interface AL = 00h Enable/Disable Pointing Device AL = 01h Reset Pointing Device AL = 02h Set Sample Rate AL = 03h Set Resolution AL = 04h Read Device Type AL = 05h Pointing Device Interface Initialization AL = 06h Extended Commands BH = 00h Return Status BH = 01h Set Scaling Factor to 1=1 BH = 02h Set Scaling Factor to 2=1 AL = 07h Device Drive Far Call Initialization
C3h	Watchdog Time-out AL = 00h Disable watchdog time-out AL = 01h Enable watchdog time-out
C4h	Programmable Option Select AL = 00h Return Base Position Adapter Register Address AL = 01h Enable Cursor for Setup AL = 02h Adapter Enable
C5h to FFh	Reserved
* Multitasking hooks (described in note above)	

continued

In this chapter

The following topics are discussed in the rest of the INT 15h System Services chapter:

- Hardware Environment
- System Services I/O Ports
- System RAM Data
- Extended System RAM Data
- CMOS RAM Data
- ROM BIOS Data
- System Services Functions

Hardware Environment

The INT 15h System Services routines assume the existence of two 8259A Programmable Interrupt Controllers, an 8237 DMA controller, an 8254 programmable interval timer/counter chip, and an 8042/8242 keyboard controller or equivalents thereof.

These routines deal with input/output devices, including keyboards, mouse pointing devices, joysticks, touch pads, track balls, RS-232 devices, and other serial devices.

System Services I/O Ports

INT 15h, System Services, references the following I/O ports:

I/O Address	Read/Write Status	Description
0020h	W	Programmable interrupt controller 1
0021h	R/W	Programmable interrupt controller 1 mask
0040h	R/W	Programmable Interrupt Timer - Read/write counter 0
0043h	W	Control byte for timers 0 and 2, where: Bits 7-6 = timer select, where 00b = select timer 0 01b = reserved 10b = select timer 2 Bits 5-4 = timer action where: 00b = counter latch command 01b = read/write counter bits 0-7 only 10b = read/write counter bits 8-15 only 11b = Read/write counter bits 0-7 first, then bits 8-15 Bits 3-1 = timer mode where 000b = mode 0 select 001b = mode 1 select X10b = mode 2 select X11b = mode 3 select 100b = mode 4 select 101b = mode 5 select Bit 0 = 1 binary coded decimal counter = 0 binary counter 16 bits

continued

I/O Address	Read/Write Status	Description
0044h	R/W	Timer 3 counter register
0047h	W	Control byte for timer three where: Bits 7-6 = 00b select timer 3 Bits 5-4 = 00b counter latch command = 01b read/write counter bit 0-7 only Bits 3-0 = 0 reserved
0060h	R/W	Keyboard/auxiliary data port
0061h	R	System control port B, where: Bit 7 = 1 Parity check Bit 6 = 1 Channel check Bit 5 = 1 Timer 2 output Bit 4 = 1 Toggle with each refresh request Bit 3 = 0 Channel check enabled Bit 2 = 0 Parity check enabled Bit 1 = 1 Speaker data enabled Bit 0 = 1 Timer 2 gate to speaker enabled
0061h	W	System control port B, where: Bit 7 = 1 Reset timer 0 output latch (IRQ 0) Bits 6-4 = Reserved Bit 3 = 0 Enable channel check Bit 2 = 0 Enable parity check Bit 1 = 1 Speaker data enable Bit 0 = 1 Enable timer 2 gate to speaker
0064h	R	8042 Status, where: Bit 7 = 1 Parity error Bit 6 = 1 General time out Bit 5 = 1 Auxiliary output buffer full Bit 4 = 1 Inhibit switch Bit 3 = 1 Command/data Bit 2 = System flag Bit 1 = 1 Input buffer full Bit 0 = 1 Output buffer full
0064h	W	8042 Commands
0070h	W	CMOS RAM address register port, where: Bit 7 = 1 NMI disable Bits 6-0 = 0 CMOS address
0071h	R/W	CMOS RAM data register port
0092h	R/W	System control port A, where: Bit 7 = 1 Fixed disk activity light bit A is on Bit 6 = 1 Fixed disk activity light bit B is on Bit 5 = 0 Reserved Bit 4 = 1 Watchdog time-out occurred Bit 3 = 1 Security lock latch locked Bit 2 = 0 Reserved Bit 1 = 1 Alternate gate A20 active Bit 0 = 1 Alternate hot reset

continued

System Services I/O Ports, Continued

I/O Address	Read/Write Status	Description
0094h	W	System board setup enable register, where: Bit 7 = 0 Enable system board setup Bit 6 = 1 Reserved Bit 5 = 0 Enable VGA setup Bits 4-0 = 1 Reserved
0096h	R/W	Channel position select register, where: Bit 7 = 1 Channel reset Bits 6-4 = Reserved (written as 0, read as 1) Bit 3 = 1 Channel select Bits 2-0 = Channel number
00A0h	W	Programmable Interrupt Controller 2
00A1h	R/W	Interrupt Controller 2 mask, where: Bit 7 = 0 Reserved Bit 6 = 0 Enable fixed disk interrupt Bit 5 = 0 Enable 80387 exception interrupt Bit 4 = 0 Enable mouse interrupt Bit 3 = 0 Reserved Bit 2 = 0 Reserved Bit 1 = 0 Enable redirect cascade Bit 0 = 0 Enable real time clock interrupt
0100h	R	System board or adapter ID (low byte)
0101h	R	System board or adapter ID (high byte)
0201h	R/W	Joystick port

System RAM Data

System RAM data table

The following table shows the data definitions used by INT 15h. The data definitions are stored in system RAM in segment 40h.

Location	Size	Description
40:0Eh	2 Words	Extended BIOS data area segment
40:67h	2 Words	Pointer to reset code (Offset:segment)
40:98h	1 Word	User wait flag offset address
40:9Ah	1 Word	User wait flag segment address
40:9Ch	1 Word	Least significant wait count
40:9Eh	1 Word	Most significant wait count
40:A0h	1 Word	Wait active flag, where: Bit 7 = 1 Wait time elapsed Bits 6-1 = Reserved Bit 0 = 1 INT 15h, AH = 86h occurred

Extended System RAM Table

The following table provides the name, length, location, and description of all extended System RAM data areas used by the auxiliary device service:

Location	Size	Description
EDA:17h	1 Byte	Number of error codes in the POST error log
EDA:18h	5 Words	POST error log
EDA:39h	1 Byte	Watchdog timer initial count
EDA:22h	1 Word	Device drive offset
EDA:24h	1 Byte	Device driver segment
EDA:26h	1 Byte	Pointing device flag (Flag 1) Bit 7 = 1 Command in progress Bit 6 = 1 Resend received from auxiliary device Bit 5 = 1 Acknowledge received from auxiliary device Bit 4 = 1 BAT failure received from auxiliary device Bits 3-0 = Data package size counter (in bytes)
EDA:27h	1 Byte	Pointing device flag (Flag 1) Bit 7 = 1 Device driver far call installed Bit 6 = 1 Pointing device interface installed Bit 5 = Reserved Bit 4 = Reserved Bits 3-0 = Data package size (in bytes)
Note: EDA = Segment of Extended System RAM Data Area.		

CMOS RAM data area table

The following table describes all CMOS RAM locations used by the INT 15h System Services routines:

Location	Size	Description
0Bh	1 Byte	Status register B, where: Bit 7 = 0 Run = 1 Halt Bit 6 = 1 Enable periodic interrupt Bit 5 = 1 Enable alarm interrupt Bit 4 = 1 Enable update-ended interrupt Bit 3 = 1 Enable square wave interrupt Bit 2 = 1 Calendar is in binary format = 0 Calendar is in BCD format Bit 1 = 1 24-hour mode = 0 12-hour mode Bit 0 = 1 Enable Daylight Savings Time
0Fh	1 Byte	Reason for shutdown, where: 00h = Power on or soft reset 01h = Memory size pass 02h = Memory test pass 03h = Memory test fail 04h = POST end; boot system 05h = JMP dword pointer with end-of-interrupt 06h = Protected tests pass 07h = Protected tests fail 08h = Memory size fail 09h = INT 15h Block Move 0Ah = JMP dword pointer without end-of-interrupt 0Bh = Used by ABIOS
34h	1 Byte	Miscellaneous information Bits 7-4 = Actual number of RS-232 ports installed Bits 3-0 = Block move status before reset to real mode
35h	1 Byte	Low byte of actual expansion memory size
36h	1 Byte	High byte of actual expansion memory size

System configuration table

The System Configuration table is located in the CBIOS ROM at F000:E6F5h. This table can be called into RAM via INT 15h function AH = C0h Return System Configuration Parameters.

Offset	Initial value	Length	Description
00h	0	1 Word	Number of bytes in this table (minimum = 8)
02h	1	1 Byte	Model byte, where: Model 80 = F8h Model 70 = F8h Model 60 = FCh Model 50 = FCh Model 30 = FAh Model 25 = FAh Unknown system board = FFh
03h	1	1 Byte	Submodel byte, where: Model 80 = 00h or 01h Model 70 = 09h Model 60 = 05h Model 50 = 04h Model 30 = 00h Model 25 = 01h Unknown system board = FFh
04h	1	1 Byte	BIOS revision level (first release = 00b)
05h	1	1 Byte	Feature information byte, where: Bit 7 = 1 Fixed disk BIOS uses DMA channel 3 Bit 6 = 1 Second interrupt chip present Bit 5 = 1 Real time clock present Bit 4 = 1 Keyboard intercept (INT 15h, function AH = 4Fh) called by keyboard interrupt service (INT 09h) Bit 3 = 1 Wait for external event supported Bit 2 = 1 Extended BIOS data area is allocated Bit 1 = 1 Micro Channel supported = 0 PC-type I/O channel implemented Bit 0 = 0 Reserved
06h	0	1 Byte	Feature information byte 2 (reserved, zeros)
07h	1	1 Byte	Feature information byte 3 (reserved, zeros)
08h	1	1 Byte	Feature information byte 4 (reserved, zeros)
09h	0	1 Byte	Feature information byte 5 (reserved, zeros)

Functions: AH = 00h - 03h Reserved

Function: AH = 04h Build System Parameters Table

Description

The System Parameters table lists the system stack requirements, the number of devices installed in the system, and entry points of the ABIOS Common Entry routines. An operating system (e.g. MS OS/2) calls this CBIOS function during the first step in initializing ABIOS. This function preserves the interrupt flag and the contents of the DS register.

This function should only be called by operating systems that make use of the ABIOS. Calling this function from within DOS will give unpredictable results.

Input:
AH = 04h
ES:DI = Pointer to location in RAM where the table is to be built.
DS = Segment of the Extended System RAM data area. (Offset is assumed to be 0000h)

Output:
AH = 00h If function was successful
CF = 1 Exception error has occurred

Note: All registers except AX and flags are restored.

For more information

For more information on ABIOS initialization, refer to the volume, *ABIOS for IBM PS/2 Computers and Compatibles*, in this series.

Function: AH = 05h Build Initialization Table

Description

The ABIOS Initialization Table describes the initialization information for each device ABIOS is to support. An operating system (e.g. MS OS/2) calls this CBIOS function during the second step in initializing ABIOS. This function preserves the interrupt flag and the contents of the DS register.

This function should only be called by operating systems that make use of the ABIOS. Calling this function from within DOS will give unpredictable results.

Input:
AH = 05h
ES:DI = Pointer to location in RAM where the table is to be built.
DS = Segment of the Extended System RAM data area. (Offset is assumed to be 0000h)

Output:
AH = 00h If function was successful
CF = 0 No error
= 1 Exception error has occurred

Note: All registers except AX and flags are restored.

For more information

For more information on ABIOS initialization, refer to the volume, *ABIOS for IBM PS/2 Computers and Compatibles*, in this series.

Functions: AH = 06h–0Eh Reserved

Function: AH = 0Fh Format ESDI Drive Periodic Interrupt

Description

The BIOS uses this function on an ESDI Fixed Disk Drive Adapter/A when doing a format or surface analysis operation and after each cylinder is completed. Refer to INT 13h, function AH = 1Ah Format ESDI Drive. These devices generally require an option ROM.

Input: AH = 0Fh
AL = Phase code
= 0 Reserved
= 1 Surface analysis
= 2 Formatting

Output: CF = 0 Formatting or analysis should continue
= 1 Formatting or analysis should be stopped

Additional information

- A program can intercept this function call so it will be informed as each cylinder is formatted or analyzed. The program could count the interrupts for each phase to get the current cylinder number.
- The BIOS default handler for this function will return with the carry flag set.
- For other types of fixed disk adapters, this function is reserved. Also, not every BIOS implements this function.

Function: AH = 10h - 20h Reserved

Function: AH = 21h Power-On Self Test Error Log

Description

This function has two subfunctions which actually read and write to and from the POST error log. This error log holds one word (16 bit) error codes which are used by the reference diskette to select the appropriate diagnostic message. These subfunctions are called by entering either 00h or 01h in register AL. If any other value is in AL, AH is set to two, one is subtracted from AL, CF is set to one, and control is returned to the caller.

The error log is stored at 40:27h in the extended system RAM data area. Up to five errors can be logged into the array. The number of error codes logged is stored in 40:26h.

Models 25 and 30: This function is not available on IBM PS/2 Models 25 and 30 (and compatibles).

Input: AH = 21h
AL = 00h Read POST error log
= 01h Write error code to POST error log

Output: CF = 0 No error
= 1 Invalid input

Subfunction: AL = 00h Read POST error log

This subfunction returns the number of logged error codes and a pointer to the error log. The information returned in this function reflects the results of the last POST that was performed. 40:26h contains the number of error codes. 40:27h is the pointer to the POST error log, which is made up of single word entries. The first byte is the device error code. The second byte is the device identifier.

Input: AH = 21h
AL = 00h
CF = 0

Output: AH = 00h
BH = 00h
BL = Number of POST error codes stored
ES:DI = Pointer to POST error log.

---

continued

Function: AH = 21h Power-On Self Test Error Log, Continued

Subfunction: AL = 01h Write error code to POST error log

This subfunction stores the device and error code in BX into the POST error log. If the maximum number of errors (5) is already stored, this code is not stored.

Input:
AH = 21h
AL = 01h
BH = Device code
BL = Device error

Output:
AH = 00h Successful
= 01h Error code; location full
AL = Index into error log, if successful
= 05h Error code; location full
CF = 0 No error
= 1 Error code location full (AH = 01h; AL = 05h)

Error conditions

No error codes are stored if the maximum number of codes is already stored. If the maximum number of error codes is already stored, AH is set to one, AL to 5, CF to 1, and control is returned to the caller.

Functions: AH = 22h - 4Eh Reserved

Function: AH = 4Fh Keyboard Intercept

Description

The keyboard interrupt function is called by INT 09h when any key is pressed. This function returns the scan code in AL with CF = 1. This function can be used to scan codes to create alternate keyboard layouts, and to cause systems to ignore certain keystrokes.

An operating system or resident utility can intercept this function in order to filter the raw keyboard data stream. A new handler can be written to substitute a different scan code, return the existing scan code, or return the carry flag cleared (thus causing the keystroke to be discarded).

Input: AH = 4Fh

Output: AL = Scan code
CF = 0 Keystroke unchanged (original scan code in AL)
= 1 Keystroke changed (new scan code in AL)

INT 15h function C0h can be called to see if a machine's BIOS supports the keyboard intercept function.

Functions: AH = 50h - 7Fh Reserved

Function: AH = 80h Device Open

Description

This function is used by multitasking operating systems. A multitasking operating system would point INT 15h to itself, intercept this function, and process the request as necessary. This is a way of capturing control of a logical device so as to arbitrate its usage by multiple processes.

Input: AH = 80h
BX = Device ID
CX = Process ID

Output: AX = 0080h

Function: AH = 81h Device Close

Description

This function is used by multitasking operating systems. A multitasking operating system would point INT 15h to itself, intercept this function, and process the request as necessary. This function releases control of a logical device for a process. See also INT 15 functions 80h and 82h.

Input: AH = 81h
BX = Device ID
CX = Process ID

Output: AX = 0081h

Function: AH = 82h Program Termination

Description

This function is used by multitasking operating systems. A multitasking operating system would point INT 15h to itself, intercept this function, and process the request as necessary. This function releases control of all logical devices for a process that will soon terminate. See also INT 15h functions 80h 81h.

Input: AH = 82h
BX = Device ID

Output: AX = 0082h

Function: AH = 83h Set Event Wait Interval

Description

INT 15h provides two wait functions: Function 83h, Set Event Wait Interval, and Function 86h, Wait. Function 86h does not return control to the caller until a specified interval is completed. Function 83h returns control to the caller.

Models 25 and 30: This function is not available on IBM PS/2 Models 25 and 30 (and compatibles).

Input: AH = 83h
AL = 00h Set wait function
= 01h Cancel wait function

Output: CF = 0 No error
= 1 Invalid input

Error conditions

If the subfunction code in AL is not 00h or 01h, one is subtracted from AL, CF is set to one, and control is returned to the caller.

Additional Information

- An event wait's duration is an integral multiple of 976 microseconds. This function is implemented with CMOS date/clock interrupts.
- Use this function to get programmed, hardware-independent delays of finer granularity than provided with the MS-DOS Get Time function (INT 21h function 2Ch). The latter returns time only in hundredths of a second.
- See also INT 15h function 86h, Wait, which halts the calling program for an interval specified in milliseconds.

continued

Subfunction: AL = 00h Set interval

This subfunction starts the wait requested the calling routine if no wait has already been started. If a wait has already been started, control is returned to the caller with CF set to one.

Input:
AH = 83h
AL = 00h
CX = Microseconds until posting (high byte)
DX = Microseconds until posting (low byte)
ES:BX = Pointer to byte in caller's memory that will have bit 7 set when the interval expires

Output:
AH = 83h
AL = A value written to CMOS register B, if successful
= 00h Function busy
CF = 0 No error
= 1 Error, function busy

Subfunction: AL = 01h Cancel set interval

This function cancels the interval in progress.

Input:
AH = 83h
AL = 01h

Output:
AX = 8300h
CF = 0 No error
= 1 Error

Function: AH = 84h Joystick Support

Description

This function controls a joystick connected to the game control adapter. It has two subfunctions: read current joystick switch settings and read resistive inputs.

Input: AH = 84h
DX = 00h Read current switch setting subfunction
= 01h Read resistive inputs subfunction

Output: CF = 0 No error
= 1 Error

Error conditions

CF is set to one if the value in DX is not 00h or 01h.

Subfunction: DX = 00h Read current switch settings

This subfunction returns the switch settings read from the joystick in bits 4 through 7 of AL. The switch settings are at port 0201h. Bits 3–0 of AL are set to zero.

Input: AH = 84h
DX = 00h

Output: AH = DH on entry to this function
AL = Switch settings in bits 7–4, Bits 3–0 = 0

continued

Function: AH = 84h Joystick Support, Continued

Subfunction: DX = 01h Read resistive inputs

This subfunction retrieves the relative position of the X and Y coordinates of the two possible joysticks. Control is returned to the caller and CF is set to zero.

Input: AH = 84h
DX = 01h

Output: AX = Joystick A X-coordinate value
BX = Joystick A Y-coordinate value
CH = Joystick B X-coordinate value
DX = Joystick B Y-coordinate value

Note: With a 250K Ohm joystick connected, potentiometer values will be around 0-416 (0000-01A0h).

Function: AH = 85h System Request Key

Description

This function is used by multitasking operating systems. A multitasking operating system would point INT 15h to itself, intercept this function, and process the request as necessary. The function is called by the BIOS keyboard driver when the SysReq key is pressed.

Input: AH = 85h
AL = 00h Key make
= 01h Key break

Output: AX = 0085h

Function: AH = 86h Wait

Description

INT 15h provides two wait functions: Function 83h, Set Event Wait Interval, and Function 86h, Wait. Function 86h does not return control to the caller until a specified interval is completed. Function 83h returns control to the caller.

Input: AH = 86h
CX = High byte of wait interval (in microseconds)
DX = Low byte of wait interval (in microseconds)

Output: AH = 86h
AL = Mask written to interrupt controller 2 (if successful)
= Unmodified if function busy
CF = 0 No error
= 1 Function busy

Error conditions

If a wait is already in progress, CF is set to one and control is returned to the caller.

CBIOS Function	Effects on CBIOS RTC ISR
INT 15h: AH = 83h Event Wait Interval INT 15h: AH = 86h Wait	These functions activate the CBIOS Real Time Clock ISR's Periodic Interrupt component.
INT 1Ah: AH = 06h Set User Alarm	This function activates the CBIOS real time clock ISR's Alarm Interrupt component.

Additional information

- The wait's duration is an integral multiple of 976 microseconds.
- Use this function to get programmed, hardware-independent delays of finer granularity than provided with the MS-DOS Get Time function (INT 21h function 2Ch). The latter returns time only in hundredths of a second.

Function: AH = 87h Move Block

Description

This function moves a block of memory for a real mode program to or from extended memory by switching to protected mode. Conventional memory is located at addresses below 640K. Only this area can be accessed by MS-DOS and its application programs. Extended memory is located at addresses above 1 MB, and has to be accessed by an 80286 or 80386 operating in protected mode. The caller must build a Global Descriptor Table with the specified settings as input to this function.

Models 25 and 30: This function is not available on IBM PS/2 Models 25/30 (and compatibles).

Global Descriptor Table

The caller supplies a pointer in ES:SI to a global descriptor table (GDT) that contains six entries. These entries must be:

- Dummy (must be zero).
- Global descriptor table (must be zero): points to the data segment where the GDT is located; can be modified by the CBIOS.
- Source address (initialized by caller): points to the source block that is to be moved.
- Destination address (initialized by caller): points to the address where the block is to be moved.
- BIOS code segment (must be zero): the CBIOS uses this entry to create a protected mode code segment.
- Caller stack segment (must be zero): the CBIOS creates a protected mode stack segment here.

Location	Initial Value (hex)	Size	Description
00h	00000000h	4 Words	Dummy entry
08h	00000000h	4 Words	Location of GDT
10h	User-supplied address	4 Words	Address of source GDT
18h	User-supplied address	4 Words	Source address
20h	00000000h	4 Words	Destination address
28h	00000000h	4 Words	Caller's stack segment

continued

Function processing

ES:SI points to a global descriptor table (GDT) that the caller builds before calling this function. The descriptors in this table perform a block move in protected mode. The data access right byte should be set to 93h in order to permit access to the data. The 24-bit address must be set to the target/source.

The Timer 0 latch is not cleared by this function. Bits 0-3 of CMOS location 34h are cleared upon completion of this function.

Interrupts are disabled while the block move is performed. For this reason, using this function may interfere with software such as communications programs and network drivers that depend on prompt servicing of hardware interrupts.

Processor-specific information

On 80386-based systems, 32-bit moves are performed. The CBIOS uses I/O port 0092h to reset the processor on 80286-based systems.

Input/Output

Input:	AH	= 87h
	CX	= Block size to be moved (0-8000h words).
	ES:SI	= Pointer to global descriptor table
Output:	AH	= 00h Successful move
		= 01h RAM Parity error occurred
		= 02h Other exception interrupt error
	CF	= 0 No error
		= 1 Error
	ZF	= 0 Unsuccessful move
		= 1 Successful move

Error conditions

If an exception interrupt occurs during the block move, AH is set to 02h. CF is set to one, ZF is set to zero, and control is returned to the caller.

If a parity error occurs, AH is set to 01h, CF is set to one, ZF is set to zero, and control is returned to the caller.

Function: AH = 88h Read Extended Memory Size

Description

The Read Extended Memory Size function reads the size of the extended memory stored in CMOS locations 35h and 36h (beginning at address (100000h) and stores the combined contents as a word in the AX register.

Models 25 and 30: This function is not available on IBM PS/2 Models 25 and 30 (and compatibles).

Input: AH = 88h

Output: AX = Number of contiguous 1K blocks of extended memory starting at address 1024K (100000h)

Additional information

- The system may not be able to use the extended memory unless the system board is fully populated.
- Extended memory is located at addresses above 1 MB, and has to be accessed by an 80286 or 80386 operating in protected mode. As MS-DOS is a real mode operating system, extended memory cannot be used for executing its programs. Extended memory can, however, be used to store volatile data.

Function: AH = 89h Switch Processor to Protected Mode

Description

The Switch Processor to Protected Mode function sets the processor into protected mode and passes control to the code segment pointed to in the GDT (global descriptor table).

Models 25 and 30: This function is not available on IBM PS/2 Models 25 and 30 (and compatibles).

Input:
- AH = 89h
- BH = Offset within IDT that defines where first 8 hardware interrupts are based
- BL = Offset within IDT that defines where second 8 hardware interrupts are based
- ES:SI = Pointer to GDT built by user

Output:
- AH = 00h successful (CF = 0). Returns in protected mode
  = FFh Unsuccessful
- CF = 0 No error
  = 1 Error

Global Descriptor Table requirements

The GDT as described below defines the memory management environment that will be in effect upon return to the caller.

The entry requirements are:

- ES:SI points to a global descriptor table, which must be built before calling this function.
- The GDT entries are used by this function to 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 by the caller.
- BH must contain an index into the IDT (interrupt descriptor table) to point to where the first eight hardware interrupts begin.
- BL must contain an index into the IDT to point to where the second eight hardware interrupts begin.

---

continued

Global Descriptor Table format

The following table describes the information contained in the global descriptor table:

Offset from ES:SI	Table Entry #	Size	Description
0	00h	4 Words	Dummy. Must be initialized to 0
1	08h	4 Words	Points to this GDT
2	10h	4 Words	Points to caller-defined Interrupt Descriptor Table
3	18h	4 Words	Points to caller's data segment
4	20h	4 Words	Points to caller's extra segment
5	28h	4 Words	Points to caller's stack segment
6	30h	4 Words	Points to code segment to which the function will return
7	38h	4 Words	Used during this function to map BIOS code segment
Note: The user-supplied GDT starting address must be in register ES:SI.			

GDT Processing

All of the descriptors are initialized by the caller except the BIOS code segment descriptor. The access rights byte is set to 09Bh, the address to E0000h, and the data segment limit to FFFFh. The DS, ES, and SS selectors are loaded with 0081h, 0020h, and 0028h, respectively.

CBIOS interrupt vectors are not written to operate in protected mode. For this reason a program that enters protected mode must construct its own IDT which does not overlap the CBIOS real mode table, and which handles all I/O interrupts while the program is in protected mode.

Error conditions

If not successful, the Carry Flag is set and AH is set to FFh.

Reference

For information about the descriptor definition, see function AH = 87h earlier in this chapter.

Functions: AH = 8Ah - 8Fh Reserved

Function: AH = 90h Device Busy

Description

This function is used by multitasking operating systems. A multitasking operating system would point INT 15h to itself, intercept this function, and process the request as necessary. This would allow other tasks to be dispatched while I/O is in progress.

Input: AH = 90h
AL = Type code (hex), where:
= 00h Disk time-out
= 01h Diskette time-out
= 02h Keyboard
= 03h Pointing device time-out
= 21h Waiting for keyboard input
= 80h Network.
= FCh Fixed disk reset time-out
= FDh Diskette drive motor start
= FEh Printer time-out
ES:BX = Points to a network control block (if AL = 80h)

Output: CF = 0 Wait time not satisfied
= 1 Minimum wait time satisfied

Function: AH = 91h Interrupt Complete

Description

This function is used by multitasking operating systems. A multitasking operating system would point INT 15h to itself, intercept this function, and process the request as necessary. This would allow the operating system to be informed when I/O is done so the requesting task could be reactivated.

Input: AH = 91h
AL = Type code, where:
= 00h Disk time-out
= 01h Diskette time-out
= 02h Keyboard
= 03h Pointing device time-out
= 80h Network
= FCh Fixed disk reset time-out
= FDh Diskette drive motor start
= FEh Printer time-out
ES:BX = Points to a network control block (if AL = 80h)

Output: None

Functions: AH = 92h - BFh Reserved

Function: AH = C0h Return System Configuration Parameters

Description

The Return Configuration Parameters function returns a pointer in ES:BX to the system configuration parameter table, located in ROM at F000:E6F5h, which gives information about the system.

Input: AH = C0h

Output: AH = 00h Successful
= 86h System model could not be determined
CF = 0 No error (AH = 00h)
= 1 System model could not be determined (AH = 86h)
ES:BX = Address of system configuration table

System configuration table

Offset	Initial value	Length	Description
00h	0	1 Word	Number of bytes in this table (minimum = 8)
02h	1	1 Byte	Model byte, where: Model 80 = F8h Model 70 = F8h Model 60 = FCh Model 50 = FCh Model 30 = FAh Model 25 = FAh Unknown system board = FFh
03h	1	1 Byte	Submodel byte, where: Model 80 = 00h or 01h Model 70 = 09h Model 60 = 05h Model 50 = 04h Model 30 = 00h Model 25 = 01h Unknown system board = FFh
04h	1	1 Byte	BIOS revision level (first release = 00b)

continued

System configuration table, cont'd

Offset	Initial value	Length	Description
05h	1	1 Byte	Feature information byte, where: Bit 7 = 1 Fixed disk BIOS uses DMA channel 3 Bit 6 = 1 Second interrupt chip present Bit 5 = 1 Real time clock present Bit 4 = 1 Keyboard intercept (INT 15h, function AH = 4Fh) called by keyboard interrupt service (INT 09h) Bit 3 = 1 Wait for external event supported Bit 2 = 1 Extended BIOS data area is allocated Bit 1 = 1 Micro Channel supported = 0 PC-type I/O channel implemented Bit 0 = 0 Reserved
06h	0	1 Byte	Feature information byte 2 (reserved, zeros)
07h	1	1 Byte	Feature information byte 3 (reserved, zeros)
08h	1	1 Byte	Feature information byte 4 (reserved, zeros)
09h	0	1 Byte	Feature information byte 5 (reserved, zeros)

Determining model type: another method

The machine model type on which the BIOS is running can be determined by reading the system board ID from POS Register 0 (I/O Port 0100h) and POS Register 1 (I/O Port 0101h):

- Model 50 = FBFFh
- Model 60 = F7FFh
- Model 80 = FEFFh

Function: AH = C1h Return Extended BIOS Data Area Segment Address

Description

This function returns the address of the data segment of the extended BIOS data area (EBDA) in ES. The extended BIOS data area address is at 40:0Eh.

The BIOS uses the EBDA for transient data storage. For example, a program passes the address of a pointing device interface subroutine to the BIOS, which stores this address in its extended data area.

The EBDA is allocated by the POST routines, which adjust the amount of free memory to allow for it. Refer to INT 12h to determine the amount of free memory available.

Input: AH = C1h

Output: ES = Extended BIOS data area segment address

Function: AH = C2h Pointing Device Interface

Introduction

Function AH = C2h Pointing Device Interface contains a set of subfunctions designed to support pointing devices. Although the pointing device most commonly attached to a system is a mouse, any type of serial input device that is compatible with the Phoenix or IBM PS/2 8042 keyboard controller firmware can be used.

For more information on IBM-compatible 8042 controller firmware, please refer to Appendix B.

Purpose of Pointing Device Interface subfunctions

In general, the Pointing Device Interface subfunctions offer mouse driver software (or other pointing device software) a way to initialize, enable, disable, and reset the mouse, as well as control such device parameters as sample rate, resolution, and scaling factor.

INT 74h ISR, function C2h, and data transmission

It is important to note that data package transmission from the pointing device to the system is not handled directly by Function C2h. Pointing device data transmission is handled mainly through INT 74h, the Interrupt Service Routine (ISR) associated with pointing devices.

The flow of data from pointing devices to the system is as follows:

- Pointing device transmits data package to 8042 chip.
- 8042 interrupts systems on IRQ 12.
- IRQ 12 is handled by INT 74h ISR.
- INT 74h ISR collects data package, storing the information in the CBIOS Extended Data Area.
- INT 74h pushes pointing device data onto the stack and calls the pointing device's device driver.
- The address of the pointing device's device driver must be stored in the CBIOS Extended Data Area by Function C2h, Subfunction 07h Device Driver Far Call Initialization.

continued

Table of subfunctions

There are several Pointing Device Interface subfunctions. The subfunctions are selected through either the AL register alone or through a combination of both the AL and BH registers.

AL value	BH value	Description
00h		Enable/Disable pointing device
01h		Reset pointing device
02h		Set sample rate
03h		Set resolution
04h		Read device type
05h		Pointing device interface initialization
06h	00h	Return status
06h	01h	Set scaling to 1:1
06h	02h	Set scaling to 2:1
07h		Device driver far call initialization

Error codes

The error status of each Pointing Device Interface subfunction is returned in the AH register. When an error condition occurs, the Carry Flag is set and control is returned to the caller.

AH Value	Description
00h	No error
01h	Invalid function call
02h	Invalid input
03h	Interface error
04h	Resend
05h	No far call installed

continued

Subfunction: AL = 00h Enable/Disable pointing device

This subfunction either enables a pointing device to allow the device to be used, or disables the device so it can no longer be used until re-enabled.

Input:
AH = C2h
AL = 00h
BH = 00h Disable
= 01h Enable

Output:
AH = 00h No error
= 01h Invalid function
= 03h Interface error
= 04h Reset
= 05h No far call installed
CF = 0 No error
= 1 Error

Subfunction: AL = 01h Reset pointing device

This subfunction resets a pointing device, and stores the device ID in BH and the initial byte transmitted from the pointing device in BL. The Pointing Device Flag (40:28h) is cleared before this function terminates.

If this subfunction completes successfully, the pointing device state will be:

- disabled,
- the sample rate will be 100 reports per second,
- the resolution will be 4 counts per millimeter.
- the scaling will be 1:1, and
- the data package size will be the same.

Input:
AH = C2h
AL = 01h

Output:
AH = 00h No error
= 03h Interface error
= 04h Resend
BH = 00h Device ID
BL = First byte of device response
CF = 0 No error
= 1 Error

continued

Subfunction: AL = 02h Set sample rate

This subfunction establishes a sample rate for the auxiliary device. If the sample rate is out of range, an invalid input condition (AH = 02h) is set and the subfunction is terminated. The rate sent to the device is derived from the value in BH, as defined in the following table:

BH Input Value	Reports per Second Sent to Auxiliary Device
00h	0Ah = 10 reports/second
01h	14h = 20 reports/second
02h	28h = 40 reports/second
03h	3Ch = 60 reports/second
04h	50h = 80 reports/second
05h	64h = 100 reports/second
06h	C8h = 200 reports/second

Input/Output

Input: AH = C2h
AL = 02h
BH = Sample rate value (00h to 06h)

Output: AH = 00h No error
= 02h Invalid input
= 03h Interface error
= 04h Resend
CF = 0 No error
= 1 Error

continued

Subfunction: AL = 03h Set resolution

This subfunction sets the resolution of the auxiliary device. If the resolution is out of range, this subfunction terminates and sets an invalid input indicator.

Input:
AH = C2h
AL = 03h
BH = Resolution value (00h to 03h counts per millimeter), where:
00h = 1 count/millimeter
01h = 2 counts/millimeter
02h = 4 counts/millimeter
03h = 8 counts/millimeter

Output:
AH = 00h No error
= 02h Invalid input
= 03h Interface error
= 04h Resend
CF = 0 No error
= 1 Error

Subfunction: AL = 04h Read device type

This subfunction reads a byte from the auxiliary device and stores it in BH. If the first byte from the auxiliary device is not an Acknowledge, the device is re-enabled and a Resend condition is set.

Input:
AH = C2h
AL = 04h

Output:
AH = 00h No error
= 03h Interface error
= 04h Resend
BH = 00h Device ID
CF = 0 No error
= 1 Error

---

continued

Subfunction: AL = 05h Pointing device interface initialization

This subfunction initializes the auxiliary device. If the data package size is out of range, this subfunction stops and sets CF on and AH to 02h.

The auxiliary device reset is attempted three times. If all attempts fail, the data package size bits are reset, AH is set to 03h, and CF is set.

If this subfunction completes normally, the pointing device state is:

- disabled,
- the sample rate will be 100 reports/second,
- the resolution will be 4 counts/millimeter, and
- the scaling will be 1:1.

Input: AH = C2h
AL = 05h
BH = Data package size (01h–08h, in bytes)

Output: AH = 00h No error
= 02h Invalid input
= 03h Interface error
= 04h Resend
CF = 0 No error
= 1 Error

Subfunction: AL = 06h Extended commands

This subfunction jumps to a subfunction based on the value in BH. If BH is out of range, this subfunction terminates and sets an invalid subfunction indicator.

Input: AH = C2h
AL = 06h
BH = 00h Return status
= 01h Set scaling factor to 1 = 1
= 02h Set scaling factor to 2 = 1

Output: AH = 00h No error
= 01h Invalid subfunction
CF = 0 No error
= 1 Error

---

continued

Subfunction: AL = 06h BH = 00h Return status

This subfunction reads and returns the status of the auxiliary device. The device sends three status bytes that are stored in registers BL, CL, and DL.

Input: AH = C2h
AL = 06h
BH = 00h

Output: AH = 00h No error
= 03h Interface error
= 04h Resend

BL = Status byte 1, where:
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

CF = 0 No error
= 1 Error

CL = Status byte 2, where:
00h = 1 count per millimeter
01h = 2 counts per millimeter
02h = 4 counts per millimeter
03h = 8 counts per millimeter

DL = Status byte 3, where:
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

continued

Subfunction: AL = 06h BH = 01h Set scaling factor to 1:1

Input: AH = C2h
AL = 06h
BH = 01h

Output: AH = 00h No error
= 03h Interface error
= 04h Resend
CF = 0 No error
= 1 Error

Subfunction: AL = 06h BH = 02h Set scaling factor to 2:1

Input: AH = C2h
AL = 06h
BH = 02h

Output: AH = 00h No error
= 03h Interface error
= 04h Resend
CF = 0 No error
= 1 Error

Subfunction: AL = 07h Device driver far call initialization

This subfunction stores the device driver segment address in the CBIOS data area at 22h and the device driver offset address at 24h so that the device may be accessed.

There must be a routine that receives control when the pointing device data is available. The device driver far call initialization gives the address of this routine to the CBIOS. Whenever pointing device data is available, the pointing device interrupt handler calls the routine, with the following arguments on the stack:

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

---

continued

Subfunction: AL = 07h Device driver far call initialization, cont'd

These words will be in the following format:

Word	Byte	Bit number	Description
1	Low	7	= 1 Y data overflow
		6	= 1 X data overflow
		5	= 1 Negative Y data sign
		4	= 1 Negative X data sign
		3	= 1 Reserved (must be 1)
		2	= 0 Reserved (must be 0)
		1	= 1 Right button pressed
		0	= 1 Left button pressed
1	High	7-0	= 0 Reserved
2	Low	7	= X data — most significant bit
		0	= X data — least significant bit
2	High	7-0	= 0 Reserved
3	Low	7	= Y data — most significant bit
		0	= Y data — least significant bit
3	High	7-0	= 0 Reserved
4	Low	7-0	= 0 Reserved
4	High	7-0	= 0 Reserved

When transferring control to the user routine, the pointing device interrupt handler uses a far call. The user routine should be coded as a far procedure and should not move the parameters off the stack before returning.

Input: AH = C2h
AL = 07h
ES:BX = Device drive segment and offset address.

Output: AH = 00h No error
= Error status (40:41h)

Function: AH = C3h Watchdog Time-Out

Description

The watchdog time-out function has two subfunctions: enable and disable. The watchdog timer ensures that a program does not turn interrupts off for too long a period. If interrupts are turned off for too long, the timer interrupt may not be serviced.

The watchdog time-out decrements each timer tick that the timer interrupt is not serviced. At the end of the watchdog period, an NMI is generated. POST disables the watchdog timer. The user must enable it.

Models 25 and 30: This function is not available on IBM PS/2 Models 25 and 30 (and compatibles).

Error conditions

If the subfunction code is not valid, one is subtracted from AL, CF is set to one, and control is returned to the caller.

Subfunction: AL = 00h Disable watchdog time-out

This subfunction disables the watchdog timer.

Input: AH = C3h
AL = 00h

Output: CF = 0 No error
= 1 Error

Subfunction: AL = 01h Enable watchdog time-out

This subfunction enables the watchdog time-out with the time-out timer tick count (from 1-255) in BX.

Input: AH = C3h
AL = 01h
BX = Watchdog timer count (1-255)

Output: CF = 0 No error
= 1 Timer count is out of range

Function: AH = C4h Programmable Option Select

Description

This function allows the caller to access the Programmable Option Select (POS) features of the system's adapter cards. Three subfunctions are provided and are selected through the AL register: return base address, enable channel for setup, and enable adapter.

Models 25 and 30: This function is not available on IBM PS/2 Models 25 and 30 (and compatibles).

Subfunction: AL = 00h Return base POS adapter register address

Input: AH = C4h
AL = 00h

Output: CF = 0 No Error
= 1 Error
DX = Base POS adapter register address

Subfunction: AL = 01h Enable channel for setup

This subfunction enables the channel specified in BL. Channel numbers must be between 1 and 8, if not, CF is set and control is returned to the caller. Channel numbers for specific models are not checked. This subfunction does not modify any registers.

Input: AH = C4h
AL = 01h
BL = Channel number (1–8)

Output: CF = 0 No error
= 1 Invalid channel number

---

continued

Function: AH = C4h Programmable Option Select, Continued

Subfunction: AL = 02h Adapter enable

This subfunction disables the setup for all channels, thus enabling the adapters on those channels.

Input: AH = C4h
AL = 02h

Output: CF = 0 No Error
= 1 Error

Functions: AH = C5h - FFh Reserved

14

Chapter 14
INT 17h Parallel Printer Service

Overview

Description

The CBIOS Parallel Printer Service provides BIOS-level support for up to three parallel printer ports.

How the Parallel Printer Service is invoked

Software INT 17h invokes the CBIOS Parallel Printer Service.

The INT 14h vector resides at address 005Ch in the Interrupt Vector Table. The CBIOS initializes the INT 17h vector to F000:EFD2h, the IBM-compatible entry point.

continued

Summary of Parallel Printer Service functions

The CBIOS Parallel Printer Service contains three functions. Individual functions are selected via the AH register.

Function	Description
00h	Print Character
01h	Initialize Printer
02h	Read Printer Status
03h-FFh	Reserved

In this chapter

This chapter focuses on the INT 17h Parallel Printer Service. The following topics are discussed:

- Hardware Environment
- Parallel Port I/O Addresses
- System RAM Data
- Error Handling
- Parallel Printer Service Functions

Hardware Environment

Parallel printer ports assumed

The CBIOS Parallel Printer Service assumes three parallel printer ports.

Parallel Port I/O Addresses

The characteristics of each printer port are defined in the following table.

I/O Address	Read/Write Status	Description
0278h	R/W	Parallel 3 data port
0279h	R/W	Parallel 3, status port, where: Bit 7 = 0 Busy Bit 6 = 0 Acknowledge Bit 5 = 1 Out of paper Bit 4 = 1 Printer is selected Bit 3 = 0 Error Bit 2 = 0 IRQ has occurred Bits 1-0 = Reserved
027Ah	R/W	Parallel 3, control port, where: Bits 7-6 = Reserved Bit 5 = 0 Direction is write to port = 1 Direction is read from port Bit 4 = 1 Enable IRQ Bit 3 = 1 Select printer Bit 2 = 0 Initialize printer Bit 1 = 1 Automatic line feed Bit 0 = 1 Strobe
0378h	R/W	Parallel 2 data port
0379h	R/W	Parallel 2, status port, where: Bit 7 = 0 Busy Bit 6 = 0 Acknowledge Bit 5 = 1 Out of paper Bit 4 = 1 Printer is selected Bit 3 = 0 Error Bit 2 = 0 IRQ has occurred Bits 1-0 = Reserved

continued

Parallel Port I/O Addresses, Continued

I/O Address	Read/Write Status	Description
037Ah	R/W	Parallel 2, control port, where: Bits 7–6 = Reserved Bit 5 = 0 Direction is write to port = 1 Direction is read from port Bit 4 = 1 Enable IRQ Bit 3 = 1 Select printer Bit 2 = 0 Initialize printer Bit 1 = 1 Automatic line feed Bit 0 = 1 Strobe
03BCh	R/W	Parallel 1, data port
03BDh	R/W	Parallel 1, status port, where: Bit 7 = 0 Busy Bit 6 = 0 Acknowledge Bit 5 = 1 Out of paper Bit 4 = 1 Printer is selected Bit 3 = 0 Error Bit 2 = 0 IRQ has occurred Bits 1–0 = Reserved
03BEh	R/W	Parallel 1, control port, where: Bits 7–6 = Reserved Bit 5 = 0 Direction is write to port = 1 Direction is read from port Bit 4 = 1 Enable IRQ Bit 3 = 1 Select printer Bit 2 = 0 Initialize printer Bit 1 = 1 Automatic line feed Bit 0 = 1 Strobe

System RAM Data

Introduction

The CBIOS Parallel Printer Service makes use of information located in the BIOS Data Area of system RAM (address 40:00h through 40:100h). The table below describes the BIOS Data Area definitions referred to by the Printer Device Service.

Location	Size	Description
08h	3 words	I/O address of up to 3 printer adapters.
10h	1 word	Number of devices installed, where: Bits 15-14 = Number of printer adapters Bits 13-12 = Reserved Bits 11-9 = Number of asynchronous adapters (RS232) Bits 8 = Reserved Bits 7-6 = Number of diskette drives, where: 00b = 1 diskette drive 01b = 2 diskette drives Bits 5-4 = Initial video mode, where: 00b = EGA/VGA or PGA 01b = 40x25 color 10b = 80x25 color 11b = 80x25 black and white Bit 3 = Reserved Bit 2 = Pointing device Bit 1 = 1 if math coprocessor Bit 0 = Diskette available for boot
78h	3 bytes	One byte per adapter. Time-out values for printer number 1 through 3.

Error Handling

Before entering any functions, this service ensures that the DX index is not out of range, that the function number is not out of range, and that the printer is at the specified index. There are only three printer ports.

If any of the above conditions are not met, the service returns to the caller with all registers restored except AH. When DX is out of range or the printer does not exist, AX is not modified. When the function number is out of range, AH is returned, and is decremented by 2.

Function: AH = 00h Print Character

Description

The Print Character function prints the character in the AL register to the printer specified in the DX register. The printer status is returned in the AH register.

Input/Output

Input: AL = Character to print
DX = Printer number (0 = LPT1, 1 = LPT2, or 2 = LPT3) used as an index to the printer base port address table at 40:08h.

Output: AH = Printer status, where:

Bit 7	=	1 Printer not busy
Bit 6	=	1 Acknowledgment
Bit 5	=	1 Out of paper
Bit 4	=	1 Printer selected
Bit 3	=	1 I/O error
Bit 2	=	Reserved
Bit 1	=	Reserved
Bit 0	=	1 Time-out

Function: AH = 01h Initialize Printer

Description

The Initialize Printer function initializes the printer selected in the DX register. The printer status is returned in register AH.

Input/Output

Input: DX = Printer number (0 = LPT1, 1 = LPT2, or 2 = LPT3) used as an index to the printer base port address table at 40:08h.

Output: AH = Printer status, where:

Bit 7	=	1 Printer not busy
Bit 6	=	1 Acknowledgment from printer
Bit 5	=	1 Out of paper
Bit 4	=	1 Printer selected
Bit 3	=	1 I/O error
Bit 2	=	Reserved
Bit 1	=	Reserved
Bit 0	=	1 Time-out error

Function: AH = 02h Read Printer Status

Description

The Read Printer Status function reads and returns the status of the printer selected by the DX register.

Input/Output

Input: DX = Printer number (0 = LPT1, 1 = LPT2, or 2 = LPT3) used as an index to the printer base port address table at 40:08h.

Output: AH = Printer status, where:

Bit 7	=	1 Printer not busy
Bit 6	=	1 Acknowledgment from printer
Bit 5	=	1 Out of paper
Bit 4	=	1 Printer selected
Bit 3	=	1 I/O error
Bit 2	=	Reserved
Bit 1	=	Reserved
Bit 0	=	1 Time-out error

Functions: AH = 03h - FFh Reserved

15

Chapter 15
INT 1Ah Time-of-Day Service

Overview

Introduction

The CBIOS provides the following timer-related support:

- **INT 1Ah Time-of-Day Service**
  Software INT 1Ah invokes the CBIOS Time-of-Day Service. The Time-of-Day Service provides CBIOS users with 12 system time and date functions.

- **INT 08h System Timer ISR**
  The Intel 82284 clock generator chip, or its equivalent, "ticks" roughly 18.2 times per second. Each tick generates hardware INT 08h, System Timer Interrupt.

  The CBIOS System Timer routine increments the system time count, then invokes software INT 1Ch. CBIOS users may revector INT 1Ch to their own routines. If not revectored, INT 1Ch returns immediately to the caller with all registers preserved.

continued

Introduction, cont'd

- **INT 70h Real Time Clock ISR**
  The Motorola MC146818A real time clock chip, or its equivalent, generates a hardware interrupt approximately 1024 times per second.

  The CBIOS Services contain three functions that determine how the CBIOS Real Time clock hardware ISR will handle interrupts from the Real Time clock chip.

CBIOS Function	Effects on CBIOS RTC ISR
INT 15h: AH = 83h Event Wait Interval INT 15h: AH = 86h Wait	These functions activate the CBIOS Real Time Clock ISR's Periodic Interrupt component.
INT 1Ah: AH = 06h Set User Alarm	This function activates the CBIOS real time clock ISR's Alarm Interrupt component.

In this chapter

This chapter discusses each of the CBIOS timer-related services. The following topics are presented:

- INT 1Ah Time-of-Day Service
- INT 08h System Timer ISR
- INT 70h Real Time Clock ISR

INT 1Ah Time-of-Day Service

Introduction

The CBIOS Time-of-Day service provides 12 system time-related functions.

How the CBIOS Time-of-Day service is invoked

The CBIOS Time-of-Day service is invoked via software interrupt INT 1Ah. Individual Time-of-Day functions are selected via the AH register. Subfunctions are selected via the AL register.

The INT 1Ah vector resides at address 00:68h in the Interrupt Vector Table. The CBIOS initializes the INT 1Ah vector to address F000:FE6Eh, the IBM-compatible entry point.

Summary of Time-of-Day service functions

The following table lists the Time-of-Day functions.

Functions	Description
00h	Read System Timer Time Counter
01h	Set System Timer Time Counter
02h	Read Real Time Clock Time
03h	Set Real Time Clock Time
04h	Read Real Time Clock Date
05h	Set Real Time Clock Date
06h	Set Real Time Clock Alarm
07h	Reset Real Time Clock Alarm
08h	Set Real Time Clock Activated Power-On Mode
09h	Read Real Time Clock Alarm Time and Status*
0Ah	Read System Timer Day Counter
0Bh	Set System Timer Day Counter
0Ch - FFh	Reserved
* For IBM PS/2 Models 25 and 30 only	

continued

Hardware Environment

The CBIOS Time-of-Day Service routine assumes the existence of a CMOS real time clock that contains 64 bytes of nonvolatile RAM and is compatible with a Motorola MC146818A chip. Also, the Intel 82284 clock generator chip, or its equivalent, is assumed.

Time-of-Day Service I/O Ports

INT 1Ah, Time-of-Day Service, references the following I/O ports:

I/O Address	Read/Write Status	Description
0070h	W	CMOS RAM address register port, where: Bit 7 = 1 NMI disable = 0 NMI enabled Bits 6-0 = 0 CMOS address
0071h	R/W	CMOS RAM data register port
00A1h	R/W	Programmable Interrupt Controller 2 mask, where: Bit 7 = 0 Reserved Bit 6 = 0 Enable fixed disk interrupt Bit 5 = 0 Enable 80387 exception interrupt Bit 4 = 0 Enable mouse interrupt Bit 3 = 0 Reserved Bit 2 = 0 Reserved Bit 1 = 0 Enable redirect cascade Bit 0 = 0 Enable real time clock interrupt

System RAM data

The following table lists all system RAM data areas, beginning at segment 40h, used by the Time-of-Day service.

Location	Size	Description
40:6Ch	1 Word	Least significant timer 'tick' count
40:6Eh	1 Word	Most significant timer 'tick' count
40:70h	1 Byte	24 hour rollover flag
40:CEh	1 Word	Count of days since 1/1/80

continued

CMOS RAM Data

The following table presents the CMOS RAM data areas used by the Time-of-Day service routine.

Location	Size	Description
00h	1 Byte	Current second in binary coded decimal (BCD)
01h	1 Byte	Second alarm in BCD
02h	1 Byte	Current minute in BCD
03h	1 Byte	Minute alarm in BCD
04h	1 Byte	Current hour in BCD
05h	1 Byte	Hour alarm in BCD
06h	1 Byte	Current day of week in BCD
07h	1 Byte	Current Date in BCD
08h	1 Byte	Current month in BCD
09h	1 Byte	Current year in BCD
0Ah	1 Byte	Status Register A, where: Bit 7 = 1 Update in progress Bits 6-4 = Divider that identifies the time-based frequency Bits 3-0 = Rate-selection bits that define output frequency and periodic interrupt rate
0Bh	1 Byte	Status Register B, where: Bit 7 = 0 Run = 1 Halt Bit 6 = 1 Enable periodic interrupt Bit 5 = 1 Enable alarm interrupt Bit 4 = 1 Enable update-ended interrupt Bit 3 = 1 Enable square wave interrupt Bit 2 = 1 Calendar is in binary format = 0 Calendar is in BCD format Bit 1 = 1 24-hour mode = 0 12-hour mode Bit 0 = 1 Enable Daylight Savings Time
0Ch	1 Byte	Status register C, where: Bits 7-4 = IRQF, PF, AF, and UF flags, respectively Bits 3-0 = Reserved
0Dh	1 Byte	Status register D, where: Bit 7 = 1 Real time clock has power Bits 6-0 = Reserved
37h	1 Byte	Century in BCD

continued

Time-of-Day Error Handling

The CBIOS Time-of-Day service generates one error that is applicable to all functions of this service. The Carry Flag is set if an invalid function is requested. The function number is stored in AH if there is an invalid function request.

Otherwise, each function handles all other errors uniquely. Refer to the description for each function for specific information about error codes.

Function: AH = 00h Read System Timer Time Counter

The Read System Timer Time Counter function reads and returns the system tick count from the system RAM location 40:6Eh (high word) and 40:6Ch (low word).

The values stored in these locations are defined by the frequency of the timer interrupt, which is approximately 18.2 ticks per second. The AL register will be 0 if the timer has not exceeded 24 hours. Otherwise, RAM data area location 40:70h has the same value that the AL register has.

Interrupts are disabled while reading RAM data memory locations because a timer tick update may occur.

Execution of this function causes the Timer Overflow Flag at 40:70h to be reset to 0.

Input: AH = 00h

Output: AH = 00h
AL = Timer overflow value, where:
= 0 Timer count is less than 24 hours since the last power-on or system reset. Any value greater than 0 Timer count is more than 24 hours since last power-on, system reset, last system-timer time counter read, or last system-timer time counter set.
CF = 0 No error
= 1 Error
CX = High word of tick count
DX = Low word of tick count

continued

Function: AH = 01h Set System Timer Time Counter

The Set system timer time counter function stores values from CX and DX to RAM data areas 40:6Eh (high word) and 40:6Ch (low word).

Execution of this function causes the Timer Overflow Flag at 40:70h to be reset to 0.

Input: AH = 01h
CX = High word of tick count
DX = Low word of tick count

Output: AH = 00h
CF = 0 No error
= 1 Error

Function: AH = 02h Read Real Time Clock Time

This function first finds out if the CMOS RAM is currently updating its clock value.

If an update is not in progress, the appropriate values are taken from CMOS RAM and returned to the caller, with the Carry Flag cleared and AH = 00h.

Input/Output

Input: AH = 02h

Output: AH = 00h
AL = Same as CH
CF = 0 Clock operating
= 1 Clock not operating
CH = Hours in BCD
CL = Minutes in BCD
DH = Seconds in BCD
DL = 00h — No Daylight Savings Time option
01h — Daylight Savings Time option

Error conditions

If a time-out occurs, which indicates an update is in progress, CF is set to one, AH is cleared, and control is returned to the caller.

---

continued

Function: AH = 03h Set Real Time Clock Time

This function first finds out if the CMOS RAM is currently updating its clock value. If an update is in progress, the real time clock is initialized.

The values defined in the input section are stored in their respective CMOS RAM storage locations. Hours are stored in CMOS RAM location 04h, minutes in location 02h, and seconds in location 00h.

Input/Output

Input:

AH	=	03h
CH	=	Hours in BCD
CL	=	Minutes in BCD
DH	=	Seconds in BCD
DL	=	00h No DST option
		01h DST option

Output:

AH	=	00h
AL	=	Value written to CMOS 0Bh register
CF	=	0 No error
	=	1 Error

Error conditions

If a time-out occurs, which indicates that an update is in progress, the clock is initialized and the routine continues as if an update were not in progress.

continued

Function: AH = 04h Read Real Time Clock Date

This function first finds out if the CMOS RAM is currently updating its clock value.

If no time-out occurs, the century, year, month, and day are extracted from CMOS RAM and are returned to the caller as described in the output section. Upon return, the Carry Flag is cleared and AH = 00h.

Input/Output

Input:	AH	=	04h
Output:	AH	=	00h
	CF	=	0 No error
		=	1 Clock update in progress
	CH	=	Century in BCD (either 19 or 20)
	CL	=	Year in BCD
	DH	=	Month in BCD
	DL	=	Day in BCD

Error conditions

If a time-out occurs, which indicates that an update is in progress, the Carry Flag is set, AH is cleared, and control is returned to the caller.

---

continued

Function: AH = 05h Set Real Time Clock Date

This function first finds out if the CMOS RAM is currently updating its clock value.

The values defined in the input section are stored in their respective CMOS RAM storage locations. Century is stored in location 37h, year in 09h, month in location 08h, and day in location 06h. This function then returns to the caller with the Carry Flag cleared and AH = 00h.

Input/Output

Input:	AH	=	05h
	CH	=	Century in BCD (either 19 or 20)
	CL	=	Year in BCD
	DH	=	Month in BCD
	DL	=	Day in BCD
Output:	AH	=	00h
	AL	=	Value written to CMOS 0Bh register
	CF	=	0 No error
		=	1 Error

Error conditions

If a time-out occurs, which indicates that an update is in progress, the clock is initialized.

continued

Function: AH = 06h Set Real Time Clock Alarm

This function first tests if an alarm is in progress. If an alarm is in progress, control is returned to the caller with the Carry Flag set and AH cleared.

If an alarm is currently not enabled, this test finds out if the CMOS RAM is updating its clock value. If a time-out occurs, which indicates that an update is in progress, the clock is initialized. This function then sets the alarm with the values from registers CH, CL and DH.

The alarm interrupt will take place at the hour, minute, and second specified in registers CH, CL, and DH, respectively. You can only have one alarm function in an active state at any one time. The alarm interrupt, once set, will be activated every 24 hours until reset.

The values defined in the input section are stored in their respective CMOS RAM storage locations.

Note: This function clears AX, not just AH.

Input/Output

Input:	AH	=	06h
	CH	=	Hours in BCD
	CL	=	Minutes in BCD
	DH	=	Seconds in BCD
Output:	AH	=	00h
	AL	=	00h
	CF	=	0 No error
		=	1 Alarm already set

Error conditions

CF is set to one if an alarm has already been set.

---

continued

Function: AH = 07h Reset Real Time Clock Alarm

This function stops the real time clock alarm interrupt.

Input: AH = 07h

Output: AH = 00h
CF = 0 No error
= 1 Error

Function: AH = 09h Read Real Time Clock Alarm and Status

This function reads the real time clock alarm's time and status.

Models 25 and 30: This function applies only to PS/2 Models 25 and 30. Other PS/2 models will return CF = 1, invalid function request.

Input: AH = 09h

Output: CH = Hours in BCD
CL = Minutes in BCD
DH = Seconds in BCD
DL = Alarm status
00h = Alarm is not installed

Function: AH = 0Ah Read System Timer Day Counter

The Read System Timer Day Counter function reads the count of days since 1/1/80 from the System RAM data area 40:CEh, and returns it to the caller in the CX register.

Models 25 and 30: On the IBM PS/2 Models 25 and 30, POST initializes the count of days since 1/1/80 to zero.

Input: AH = 0Ah

Output: AH = 00h
CF = 0 No error
= 1 Error
CX = Count of days since 1/1/80

---

continued

Function: AH = 0Bh Set System Timer Day Counter

The Set System Timer Day Counter function sets the count of days since 1/1/80 in the ROM BIOS data field 40:CEh from the parameter passed in the CX register.

Models 25 and 30: On the IBM PS/2 Models 25 and 30 (and compatibles), POST initializes the count of days since 1/1/80 to zero.

Input: AH = 0Bh
CX = Count of days since 1/1/80

Output: AH = 00h
CF = 0 No error
= 1 Error

AH = 0Ch to FFh Reserved

Functions AH = 0Ch to FFh are reserved.

INT 08h System Timer ISR

Description

The Intel 82284 clock generator chip, or its equivalent, "ticks" roughly 18.2 times per second. Each tick generates hardware INT 08h, System Timer Interrupt.

The CBIOS System Timer routine increments the double word, system time count. The system time count is stored in the CBIOS system RAM data area at locations 40:6Ch (low word) and 40:6Eh (high word).

If the count in these location exceeds 24 hours, then the contents of location 40:70h is set to 1 and the date is incremented. When 40:70h is set, the system time count (40:6Ch and 40:6Eh) is cleared.

Diskette motor count

The processing of INT 08h includes the decrement of the diskette motor counts at 40:35h. If the count reaches zero, the diskette motors are turned off.

INT 1Ch

Once it has serviced the clock, INT 08h issues a software INT 1Ch. CBIOS users may revector INT 1Ch to their own routines.

If not revectored, INT 1Ch returns immediately to the caller with all registers preserved.

How CBIOS System-Timer ISR is invoked

The CBIOS System-Timer hardware interrupt service routine is invoked via hardware INT 08h.

The INT 08h vector resides at address 00:20h in the Interrupt Vector Table. The CBIOS initializes the INT 08h vector to F000:FEA5h, the IBM-compatible entry point.

---

continued

System RAM data

The CBIOS system timer ISR references the following system RAM data:

Location	Size	Description
40:0Eh	1 Byte	Extended BIOS data area
40:3Fh	1 Byte	Diskette drive motor status
40:40h	1 Byte	Diskette motor time-out count
40:6Ch	1 Word	Least significant timer count
40:6Eh	1 Word	Most significant timer count
40:70h	1 Byte	24 hour rollover flag
40:CEh	1 Byte	Count of days since 1/1/80

Extended BIOS data area

The CBIOS system timer ISR references the following item in the CBIOS extended data area:

Location	Size	Description
EDA:35h	1 Byte	Watchdog timer initial count
Note: EDA = Extended Data Area segment		

Hardware Environment

The CBIOS System Timer routine assumes the existence of a CMOS real time clock with 64 bytes of nonvolatile RAM, compatible with a Motorola MC146818A chip. Also, the Intel 82284 clock generator chip or its equivalent is assumed.

continued

System Timer I/O Ports

INT 08h, System Timer ISR, references the following I/O ports:

I/O Address	Read/Write Status	Description
0020h	R/W	Base port of 8259 interrupt
0040h	R/W	Read/write counter 0
0042h	R/W	Read/write counter 2
0043h	W	Control word register for counters 0 and 2, where: Bits 7-6 = 00b Select counter 0 = 01b Reserved = 10b Select counter 2 Bits 5-4 = 00b Counter latch command = 01b Read/write counter bits 0-7 only = 10b Read/write counter bits 8-15 only = 11b Read/write counter bits 0-7 first, then bits 8-15 Bits 3-0 = 000b Mode 0 select = 001b Mode 1 select = X10b Mode 2 select = X11b Mode 3 select = 100b Mode 4 select = 101b Mode 5 select Bit 0 = 0 Binary counter 16 bits = 1 Binary coded decimal counter
0044h	W	Read/write counter 3
0047h	W	Control word register for counter 3, where: Bits 7-6 = 00b Select counter 3 = 01b Reserved = 10b Reserved = 11b Reserved Bits 5-4 = 00b Counter latch command select counter 0 = 01b Read/write counter bits 0-7 only = 10b Reserved = 11b Reserved Bits 3-0 = 00b
0061h	R	System control port B
03F2h	W	Diskette controller digital output register, where: Bit 7-6 = 0 Reserved Bit 5 = 1 Motor enable 1 Bit 4 = 1 Motor enable 0 Bit 3 = 0 Allow interrupts Bit 2 = 0 Controller reset Bit 1 = 0 Reserved Bit 0 = 0 Drive select 0 = 1 Drive select 1

INT 70h Real Time Clock ISR

Introduction

The Motorola MC146818A real time clock chip, or its equivalent, generates a hardware interrupt approximately 1024 times per second. The CBIOS real time clock hardware interrupt service routine (ISR) is invoked on each real time clock interrupt.

How the real time clock ISR is invoked

The CBIOS real time clock ISR is invoked via hardware interrupt INT 70h.

The INT 70h vector resides at address 00:1Ch in the Interrupt Vector Table. The CBIOS initializes the INT 70h vector to address F000:5124h, the IBM-compatible entry point.

Real Time clock ISR processing

The CBIOS Services contain three functions that determine how the CBIOS Real Time clock hardware ISR will handle interrupts from the Real Time clock chip.

CBIOS Function	Effects on Real Time Clock Processing
INT 15h: AH = 83h Event Wait Interval INT 15h: AH = 86h Wait These functions activate the CBIOS Real Time Clock ISR's Periodic Interrupt component.	PERIODIC INTERRUPT PROCESSING The CBIOS Real Time clock ISR decrements a counter (set to a user-defined period of time) by 1/1024 second for each RTC interrupt. When the count is less than or equal to zero, bit 7 of a user Wait flag is set.
INT 1Ah: AH = 06h Set User Alarm This function activates the CBIOS real time clock ISR's Alarm Interrupt component.	ALARM INTERRUPT PROCESSING The CBIOS real time clock ISR decrements a counter (set to a user-specified period of time) by 1/1024 second for each RTC interrupt. When the count reaches zero, the CBIOS real time clock ISR issues a software INT 4Ch. CBIOS users may revector INT 4Ch to their own routines. If not revectored, INT 4Ch returns immediately to the caller with all registers preserved.

continued

Hardware Environment

The CBIOS Real Time Clock Interrupt Service routine assumes the existence of a CMOS real time clock with 64 bytes of nonvolatile RAM, compatible with a Motorola MC146818A chip. Also, the Intel 82284 clock generator chip or its equivalent is assumed.

System RAM data area

The CBIOS real time clock ISR references the following system RAM data:

Location	Size	Description
40:6Ch	1 Word	Least significant timer count (incremented by INT 08h)
40:6Eh	1 Word	Most significant timer count
40:70h	1 Byte	24 hour rollover flag
40:98h	1 Word	User wait flag offset address
40:9Ah	1 Word	User wait flag segment address
40:9Ch	1 Word	Least significant wait count
40:9Eh	1 Word	Most significant wait count
40:A0h	1 Byte	Wait active flag, where: Bit 7 = 1 If wait time elapsed Bits 6–1 = Reserved Bit 0 = 1 If INT 15h, AH = 86h (wait) has occurred

continued

Real Time Clock I/O Ports

INT 70h, Real Time Clock ISR, references the following I/O ports:

I/O Address	Read/Write Status	Description
0020h	R/W	Master Programmable Interrupt Controller
0070h	W	CMOS RAM address register port, where: Bit 7 = 1 NMI disable = 0 NMI enabled Bits 6-0 = 0 CMOS address
0071h	R/W	CMOS RAM data port
00A0h	R/W	Slave Programmable Interrupt Controller

CMOS RAM data table areas

The CBIOS real time clock ISR references the following CMOS RAM data:

Location	Size	Description
0Bh	1 Byte	Status register B, where: Bit 7 = 0 Run = 1 Halt Bit 6 = 1 Enable periodic interrupt Bit 5 = 1 Enable alarm interrupt Bit 4 = 1 Enable update-ended interrupt Bit 3 = 1 Enable square wave interrupt Bit 2 = 1 Calendar is in binary format = 0 Calendar is in BCD format Bit 1 = 1 24-hour mode = 0 12-hour mode Bit 0 = 1 Enable Daylight Savings Time
0Ch	1 Byte	Status register C, where: Bits 7-4 = IRQF, PF, AF, and UF flags, respectively Bits 3-0 = Reserved
0Dh	1 Byte	Status register D, where: Bit 7 = 1 Real time clock has power Bits 6-0 = Reserved

16

Chapter 16
Single Function CBIOS Services

Overview

Introduction

Most of the CBIOS device services contain a group of functions intended to service a given device. For example, in the CBIOS Parallel Printer service, invoking INT 17h with AH = 00h causes the service to execute its print character function. On the other hand, invoking INT 17h with AH = 01h causes the service to execute its Initialize Printer Port function.

The single function services

In addition to its multifunction device services, the CBIOS also contains four single function device services. Since these services perform only one function, it is not necessary to specify a function number in the AH register.

continued

In this chapter

Because each single function service can be described on one or two pages, they are grouped into this one chapter.

This chapter presents the following topics:

- INT 05h Print Screen Service
- INT 11h Equipment List Service
- INT 12h Memory Size Service
- INT 19h Bootstrap Loader Service

INT 05h Print Screen Service

Description

The Print Screen Service:

- prints the contents of the entire current video screen to printer number 0,
- moves the cursor across the screen as the lines on the screen are printed,
- repositions the cursor position to its original position, and
- returns all registers to their original values.

Since the Print Screen Service takes a considerable amount of time to execute, interrupts are enabled throughout the routine.

How the service is invoked

INT 05h invokes the CBIOS Print Screen Service.

The INT 05h vector resides at the Interrupt Vector Table address 00:14h. The CBIOS initializes the INT 05h Print Screen Service vector to address F000:FF54h, the IBM-compatible BIOS entry point.

Note: INT 05h is usually invoked by the keyboard interrupt handler (INT 09h) when the PrtSc key is pressed. However, CBIOS users may invoke INT 05h independently of the INT 09h interrupt handler. For more information on INT 09h, refer to Chapter 8 of this volume.

System RAM Data

The Print Screen Service uses two system RAM data definitions. Both definitions are located in the CBIOS Data Area (absolute hex address 400h to 500h) and are defined in the table below:

Location	Size	Description
40:84h	1 Byte	Number of rows on screen (24/25)
40:100h	1 Byte	Print Screen Status Byte, where: 00h = Print Screen Service inactive, or Print Screen Service successful upon return 01h = Print Screen in progress FFh = Error occurred

continued

Input/Output

Input: None

Output: None (Registers are preserved)

Error conditions

The Print Screen Service recognizes three kinds of error conditions.

The table below summarizes the Print Screen Service error handling procedures.

If...	Then the Print Screen Service...
the printer is busy	▪ Sets location 40:100h = FFh ▪ Repositions cursor to original position ▪ Preserves all registers ▪ Returns to the caller
the printer is out of paper	▪ Repositions cursor to original position ▪ Preserves all registers ▪ Returns to the caller ▪ Sets location 40:100h = FFh
recursive print screen requests are made	▪ Preserves all registers ▪ Returns to the caller ▪ Does not change location 40:100h (it was set to 01 by previous INT 05h request)

INT 11h Equipment List Service

Description

The Equipment List Service returns the contents of system RAM location 40:10h in the AX register. This memory location contains the system equipment list as determined by the CBIOS power-on self test and initialization (POST) routine.

How the service is requested

INT 11h requests the Equipment List Service of the CBIOS.

The INT 11h vector resides at the Interrupt Vector Table address 0:44h. The CBIOS initializes the INT 11h Equipment List Service vector to address F000:F84Dh, the IBM-compatible entry point.

Input/Output

Input: None

Output: Equipment List

AX = Contents of system RAM location 40:10h, where:

- Bits 15–14 = Number of printer adapters
- Bits 13–12 = Reserved
- Bits 11–9 = Number of RS-232-C asynchronous adapters
- Bit 8 = Reserved
- Bits 7–4 = Number of diskette drives, (if Bit 0 = 1), where:
  - 00b = 1 diskette drive
  - 01b = 2 diskette drives
- Bit 5–4 = Initial video mode, where:
  - 00b = EGA or PGA
  - 01b = 40x25 color
  - 10b = 80x25 color
  - 11b = 80x25 black and white
- Bit 3 = Reserved
- Bit 2 = 1 If pointing device installed
- Bit 1 = 1 If math coprocessor installed
- Bit 0 = 1 If diskette available for boot

INT 12h Memory Size Service

Description

The Memory Size Service returns the contents of system RAM location 40:13h in the AX register. This location contains the amount of available base memory in kilobytes as it is determined by the CBIOS power-on self test and initialization routine (POST).

POST determines the amount of available base memory according to the formula below:

Available base memory	= Installed base memory - Memory reserved for Extended Data Area - Upper 1K reserved for BIOS Extended Data Area

Note: The CBIOS Extended Data Area generally occupies 1K of upper memory. In systems with 640K of installed base memory, the Memory Size Service will report 639K of available base memory.

How the service is invoked

INT 12h invokes the CBIOS Memory Size Service.

The INT 12h vector resides at the Interrupt Vector Table address 0:48h. The CBIOS initializes the INT 12h memory size service vector to address F000:F841h, the IBM-compatible entry point.

Input/Output

Input: None

Output: Available Base Memory
AX = Contents of location 40:13h in binary form

INT 19h Bootstrap Loader Service

Description

The Bootstrap Loader Service attempts to load the 512-byte boot sector code from either floppy or fixed disk to address 0:7C00h, transferring control to that code segment.

The boot code is located in the first sector of the floppy diskette (track 0, head 0, sector 1) or in the first sector of the fixed disk (cylinder 0, head 0, sector 1).

Input/Output

Input: = None

Output: = None (registers are preserved)

How the service is invoked

INT 19h invokes the CBIOS Bootstrap Loader Service.

The INT 19h vector resides at the Interrupt Vector Table address 0:64h. The CBIOS initializes the INT 19h Bootstrap Loader Service vector to address F000:E6F2h, the IBM-compatible entry point.

Note: INT 19h is usually invoked by the power-on self test and initialization (POST) routine at the end of all POST processing. CBIOS users are free to invoke INT 19h independently of POST. Users should be aware, however, that invoking INT 19h does not reset or reinitialize the system but simply causes the system to reboot. For more information on POST and INT 19h, refer to Chapter 6 of this volume.

---

continued

Bootstrap processing

Once INT 19h has been invoked, the Bootstrap Loader Service executes as described below:

Step	Description
1. Locate and load boot code	The boot code, located in the first sector of the floppy, undertakes the following processing: diskette (track 0, head 0, sector 1) or in the first sector of the hard disk (cylinder 0, head 0, sector 1). The Bootstrap Loader attempts to locate the boot sector and load it into memory at address 0:7C00h.
2. Check for power-on password	The Bootstrap Loader Service tests for the presence of a system power-on password before it attempts to transfer control to the boot code. End users enable or disable the power-on password via the Reference Diskette.
3. If power-on password enabled	If the Bootstrap Loader finds the power-on password is enabled, it provides the user with three opportunities to enter the password correctly. After three attempts the system halts and must be turned off, then turned on again before INT 19h may be invoked again. The Bootstrap Loader recalculates the CRC if the power-on password has been changed.
4. Process boot code, if found	Once power-on password issues have been resolved, the Bootstrap Loader Service transfers control to the boot code if it has been located.
5. If no boot code found, invoke INT 18h.	If the Bootstrap Loader Service does not find the boot sector code, it invokes INT 18h. By default, INT 18h displays the message "No boot device available, press F1 to continue." (See INT 18h DSR handler below.)

INT 18h DSR handler

If no boot sector is found either on the diskette in drive A: or on the fixed disk, Bootstrap Loader Service invokes software interrupt INT 18h. INT 18h can be vectored to a "no boot device" routine which takes over the boot process. For example, INT 18h can be vectored to a routine that would allow the system to be booted over a network.

By default, the CBIOS initializes INT 18h to point to a routine that displays the message: "NO BOOT DEVICE AVAILABLE, PRESS F1 TO CONTINUE."

continued

INT 19h and the power-on password

The Bootstrap Loader Service maintains a flag which is set when the power-on password has been input correctly. Once the power-on password is input correctly, INT 19h may be invoked again (either directly or through <Ctrl> <Alt> <Del> warm boot) without requiring user re-entry of the password.

Password Messages

In PS/2 MCA-compatible systems, security is implemented primarily by password control. There are two types of password control:

- sign-on, which controls access to the system, and
- keyboard security, which controls access to the keyboard.

Several messages relate to password control. They are described in the following table.

Message	Cause	Solution
Enter password	A password has been installed and now controls access to the system.	Enter the correct 1-7 character password.
Password Ok	A correct password has been entered.	No action.
Password has been removed	The old password has been deleted. Now there is no password protection until a new password is entered.	No action.
New password has been installed	A new password is now entered.	Use the new password to access the system.
System halted! Must power down	An incorrect password was entered. The system stops and must be rebooted.	Enter the correct password.
Password is incorrect	Incorrect password entered.	Enter the correct password.

Models 25 and 30: Password control is not available in PS/2 compatible systems (such as Models 25 and 30) which are not MCA-based.

continued

I/O Ports Used

INT 19h, Bootstrap Loader Service, references the following I/O ports:

I/O Address	Read/Write Status	Description
0060h	R/W	Keyboard/auxiliary data port
0064h	W	8042 Commands
0064h	R	8042 Status, where: Bit 7 = 1 Parity error Bit 6 = 1 General time out Bit 5 = 1 Auxiliary output buffer full Bit 4 = 1 Inhibit switch Bit 3 = 1 Command 0 Data Bit 2 = System flag Bit 1 = 1 Input buffer full Bit 0 = 1 Output buffer full
0070h	W	CMOS RAM address register port, where: Bit 7 = 1 NMI disable = 0 NMI enabled Bits 6–0 = 0 CMOS address
0071h	R/W	CMOS RAM data register port
0090h	W	Arbitration register, where: Bit 7 = 1 Enable system microprocessor cycle Bit 6 = 1 Arbitration mask Bit 5 = 1 Enable extended arbitration Bits 4–0 = 0 Reserved
0090h	R	Arbitration register, where: Bit 7 = 1 System microprocessor cycles enabled Bit 6 = 1 Arbitration mask by NMI Bit 5 = 1 Bus timeout Bit 4 = 0 Reserved Bits 3–0 = Arbitration level
0092h	R/W	System control port A, where: Bit 7 = 1 Fixed disk activity light bit A on Bit 6 = 1 Fixed disk activity light bit B on Bit 5 = Reserved Bit 4 = 1 Watchdog time-out occurred Bit 3 = 1 Security lock latch is locked Bit 2 = Reserved Bit 1 = 1 Alternate gate A20 active Bit 0 = 1 Alternate hot reset
0094h	R/W	System board setup enable register, where: Bit 7 = 0 Enable system board setup = 1 Disable system board setup Bit 6 = 1 Reserved Bit 5 = 0 Enable VGA setup 1 Disable VGA setup Bits 4–0 = 1 Reserved

continued

System RAM data

The Bootstrap Loader Service uses three system RAM data definitions. The definitions are located in the BIOS Data Area (absolute hex address 400h to 500h) and are defined in the table below.

Location	Size	Description
40:12h	1 Byte	Reserved for manufacturer's test, where: Bits 7-1 = Reserved Bit 0 = 1 Manufacturing test mode = 0 Nonmanufacturing test mode
40:72h	1 Word	System Reset Flag Where: 1234h = Bypass memory text 4321h = Reserved for ABIOS
40:75h	1 Byte	Number of fixed disk drives

CMOS RAM data

The Bootstrap Loader Service uses four CMOS RAM data areas. The definitions are located in the standard CMOS RAM Data Area and are defined in the table below.

Location	Size	Description
0Eh	1 Byte	Diagnostic status, where: Bit 7 = 1 Real time clock has lost power Bit 6 = 1 CMOS checksum is bad Bit 5 = 1 Invalid configuration information found at POST Bit 4 = 1 Memory size compare error found at POST Bit 3 = 1 Fixed disk or adapter fails initialization Bit 2 = 1 CMOS time found invalid Bit 1 = 1 Adapters do not match configuration Bit 0 = 1 Time-out in reading an adapter ID
13h	1 Byte	Password Configuration Bits 7-2 = 0 Reserved Bit 1 = 1 Network password installed = 0 Network password not installed Bit 0 = 1 Power-on password installed = 0 Power-on password not installed
38h-3Eh	1 Byte	Power-on password
39h	1 Byte	Power-on password checksum

A

Appendix A
CBIOS Error Messages

Overview

Introduction

The CBIOS can generate a variety of messages. Messages can be merely informational, or can alert the user to a hardware or software failure.

A Phoenix CBIOS generates both beep codes (useful for error conditions when POST is not completed and the monitor is not yet initialized by the system) and screen error messages.

In this appendix

This appendix lists all CBIOS error messages and beep codes. The following information is presented:

- POST: Error Handling
- POST: Beep Codes
- POST: System Failure Messages
- POST: Boot Failure Messages
- POST: Informational Messages
- NMI: Run Time Messages
- INT 19h: Password Messages
- Model 25 and 30 Beep Codes
- Model 25 and 30 Error Messages

POST: Error Handling

Introduction

The Phoenix POST reports test or initialization failures in one of the following ways:

- Beep codes
- System failure messages
- Boot failure messages
- Informational messages

Beep codes

When a POST test fails before the video screen is available for display, POST halts the processor and notifies the user by sounding a series of beeps. Beep codes are written to a hardware test I/O port. The default value for the hardware test I/O port is 0680h, but it may be different in some systems. Hardware test equipment external to the system can be used to read this information.

Beep codes are generated by the Phoenix CBIOS only.

System failure messages

POST notifies the user of a test or initialization failure by displaying a message on the screen. Each message is preceded by an error code. The error code is also stored in the POST error log.

End users use the reference diskettes supplied with their systems to run a utility program that provides a detailed explanation of each error stored in the POST error log.

The IBM CBIOS displays only the numeric error code for a given system failure.

continued

Boot failure messages

Once the POST test and initialization procedures are completed, POST boots the system by issuing software interrupt, INT 19h System Boot load. If this process fails, POST displays a screen message or graphic. System boot messages are not preceded by a numeric error code and are not logged into the POST error log.

Informational messages

In addition to error messages, the CBIOS POST also generates a set of messages that inform the user of the progress of POST memory testing.

POST: Beep Codes (Phoenix CBIOS Only)

When a POST test fails before the video screen is available for display, the Phoenix CBIOS POST notifies the user by sounding a series of beeps, and halts the processor. Beep codes are also written to a hardware test I/O port. The default value for this port is in I/O port 680h, but it may vary. Hardware test equipment external to the system can be used to read this information.

Beep Code	Contents Port 680h	Description
none	01h	CPU register test in progress
1-1-3	02h	CMOS write/read failure
1-1-4	03h	ROM BIOS checksum failure
1-2-1	04h	Programmable interval timer failure
1-2-2	05h	DMA initialization failure
1-2-3	06h	DMA page register write/read failure
1-3-1	08h	RAM refresh verification failure
none	09h	First 64K RAM test in progress
1-3-3	0Ah	First 64K RAM chip or data line failure multibit
1-3-4	0Bh	First 64K RAM odd/even logic failure
1-4-1	0Ch	Address line failure first 64K RAM
1-4-2	0Dh	Parity failure first 64K RAM
2-1-1	10h	Bit 0 first 64K RAM chip or data line failure
2-1-2	11h	Bit 1 first 64K RAM chip or data line failure
2-1-3	12h	Bit 2 first 64K RAM chip or data line failure
2-1-4	13h	Bit 3 first 64K RAM chip or data line failure
2-2-1	14h	Bit 4 first 64K RAM chip or data line failure
2-2-2	15h	Bit 5 first 64K RAM chip or data line failure
2-2-3	16h	Bit 6 first 64K RAM chip or data line failure
2-2-4	17h	Bit 7 first 64K RAM chip or data line failure
2-3-1	18h	Bit 8 first 64K RAM chip or data line failure
2-3-2	19h	Bit 9 first 64K RAM chip or data line failure
2-3-3	1Ah	Bit A first 64K RAM chip or data line failure

continued

POST: Beep Codes (Phoenix CBIOS Only), Continued

Beep Code	Contents Port 680h	Description
2-3-4	1Bh	Bit B first 64K RAM chip or data line failure
2-4-1	1Ch	Bit C first 64K RAM chip or data line failure
2-4-2	1Dh	Bit D first 64K RAM chip or data line failure
2-4-3	1Eh	Bit E first 64K RAM chip or data line failure
2-4-4	1Fh	Bit F first 64K RAM chip or data line failure
3-1-1	20h	Slave DMA register failure
3-1-2	21h	Master DMA register failure
3-1-3	22h	Master interrupt mask register failure
3-1-4	23h	Slave interrupt mask register failure
none	25h	Interrupt vector loading in progress
3-2-4	27h	Keyboard controller test failure
none	28h	CMOS power failure and checksum calculation in progress
none	29h	CMOS configuration validation in progress
3-3-4	2Bh	Screen memory test failure
3-4-1	2Ch	Screen initialization failure
3-4-2	2Dh	Screen retrace test failure
none	2Eh	Search for video ROM in progress
none	30h	Screen operable
none	30h	Screen running with video ROM
none	31h	Monochrome monitor operable
none	32h	Color monitor (40 column) operable
none	33h	Color monitor (80 column) operable

POST: System Failure Messages

Error Code	Message	Possible Cause	Suggested Solution
102	System Board Failure	Timer 0 or 2 failed during POST.	Replace the system board.
103	System Board Failure	The timer interrupt failed while testing timer 0 during POST.	Replace the system board.
104	System Board Failure	An unexpected interrupt has occurred while the processor was in protected mode.	Replace the system board.
108	System Board Failure	Timer 2 failed during POST.	Replace the system board.
114	Option ROM Checksum Failure — *xxxx:0*	The checksum calculated for an option ROM does not match the checksum stored in the option ROM. *XXXX* indicates the segment address of the failing option ROM.	Replace the failing option ROM or adapter card containing the option ROM.
130	System Board Failure	A shutdown failure has occurred during POST.	Replace the system board.
131	System Board Failure	Gating of the A20 address line failed when entering/exiting protected mode.	Replace the system board.
161	Real Time Clock Failure (battery) Please Reconfigure and Retry	The real time clock's bad battery flag bit is set.	Replace the battery. If the problem persists, replace the system board.
162	Real Time Clock Failure (CRC) Please Reconfigure and Retry	The checksum for the CMOS in the real time clock is incorrect.	Run the Reference Diskette to update CMOS. If the problem persists, replace the battery. if the problem still occurs, replace the system board.

continued

POST: System Failure Messages, Continued

Error Code	Message	Possible Cause	Suggested Solution
163	Time and Date Not Set	The date and time in the real time clock has not been set or has been corrupted.	Run the Reference Diskette to update time and date. If the problem persists, replace the battery. If the problem still occurs, replace the system board.
164	Invalid Configuration Information (Memory) Please Reconfigure and Retry	The amount of memory found in the system does not match the amount of memory specified in CMOS.	Run the Reference Diskette to correctly configure the system. If the problem persists, replace the system board.
165	Invalid Configuration Information (Adapter) Please Reconfigure and Retry	An adapter configured by the Reference Diskette program does not exist in the correct slot in the system.	Run the Reference Diskette to correctly configure the system.
166	Invalid Configuration Information (Adapter) Please Reconfigure and Retry	An adapter configured by the Reference Diskette either ▪ does not exist in the slot specified by the system, or ▪ is not responding.	Run the Reference Diskette to correctly configure the system. If the problem persists, replace the adapter.
167	Invalid Configuration Information (diskette)	Diskette adapter does not exist or is not responding.	Run the Reference Diskette to configure the system. Replace the adapter if problem persists.
168	Invalid configuration information (System board)	Math coprocessor either not operating or missing.	Either replace coprocessor or reconfigure.
201	Memory Failure at xxxx:yyyy, Read aa, Expecting bb	A read/write error has occurred in memory at the address specified by 'xxxx:yyyy.' The test value written to the memory location is indicated by 'bb' and the value read from the memory location is indicated by 'aa.'	Replace the memory in the system. If the problem persists, replace the system board.

continued

POST: System Failure Messages, Continued

Error Code	Message	Possible Cause	Suggested Solution
202	Memory Failure at xxxx:yyyy, Read aa, Expecting bb	A memory parity error has occurred in memory at the address specified by 'xxxx:yyyy.' The test value written to the memory location is indicated by 'bb' and the value read from the memory location is indicated by 'aa.'	Replace the memory in the system. If the problem persists, replace the system board.
203	Memory Failure at xxxx:yyyy, Read aa, Expecting bb	An address line failure has occurred in memory at the address specified by 'xxxx:yyyy.' The test value written to the memory location is indicated by 'bb' and the value read from the memory location is indicated by 'aa.'	Replace the memory in the system. If the problem persists, replace the system board.
301	Keyboard Failure	The keyboard controller has failed during POST.	Check the keyboard to make sure it is functioning correctly and is installed correctly. If the problem persists, replace the system board.
304	Keyboard Failure	The clock line on the keyboard controller has failed during POST.	Check the keyboard to make sure it is functioning correctly and is installed correctly. If the problem persists, replace the system board.
306	Keyboard Failure	The data line on the keyboard controller has failed during POST.	Check the keyboard to make sure it is working correctly and is installed correctly. If the problem persists, replace the system board.

continued

POST: System Failure Messages, Continued

Error Code	Message	Possible Cause	Suggested Solution
307	Keyboard Failure	A stuck key has been detected on the keyboard.	Check the keyboard to make sure it is working correctly and is installed correctly. If the problem persists, replace the system board.
601	Diskette Drive Failure	The reset function has failed on the diskette subsystem.	The diskette controller has failed and should be replaced.
602	Diskette Drive x Failure	The seek function has failed on diskette drive x.	The drive is malfunctioning and should be replaced.
901	Parallel Port Failure	Parallel printer data port failed.	Use an alternate printer port or replace the system board.
1101	Serial Port Failure	The serial port on the system board has failed.	Use an alternate serial port or replace the system board.
1780	Fixed Disk Drive 0 Failure	Fixed disk drive 0 is not ready, or the recalibration function failed during initialization.	Check the fixed disk controller. Check all connections. If the problem persists, test the drive using the Reference Diskette.
1781	Fixed Disk Drive 1 Failure	Fixed disk drive 1 is not ready, or the recalibration function failed during initialization.	Check the fixed disk controller. Check all connections. If the problem persists, test the drive using the Reference Diskette.
1782	Fixed Disk Reset Failure	Fixed disk failed reset.	Check the fixed disk controller. Check all connections. If the problem persists, test the drive using the Reference Diskette.
1790	Fixed Disk Drive 0 Failure	The verify function has failed on fixed disk drive 0.	Run Reference Diskette to determine if the disk is usable.

continued

POST: System Failure Messages, Continued

Error Code	Message	Possible Cause	Suggested Solution
1791	Fixed Disk Drive 1 Failure	The verify function has failed on fixed disk drive 1.	Run the Reference Diskette to determine if the disk is usable.
2401	Video Failure	The system board video controller has failed during POST.	Replace the system board.
3001	Video Failure	The system board video controller has failed during POST.	Replace the system board.
7000*	Chip Set CMOS Failure	Bad chip set battery, or bad CMOS RAM.	Replace the chip set battery.
7001*	Chip Set Shadow RAM Failure	ROM not shadowed to RAM.	Enable RAM shadowing again. If still generated, bad or nonexistent RAM.
7002*	Bad Configuration Information in CMOS	Chip set CMOS has incorrect information.	Reconfigure with the Reference Diskette. If error persists, replace CMOS battery or CMOS RAM.
8601	Pointing Device Interface Failure	The pointing device (mouse) was detected but did not accept a command.	Check the pointing device for proper operation. If the problem persists, replace the device. If the problem still occurs, replace the system board.
8602	Pointing Device Interface Failure	The pointing device (mouse) was detected but an interface error occurred while testing.	Check the pointing device for proper operation. If the problem persists, replace the device. If the problem still occurs, replace the system board.
8603	Pointing Device Failure	The pointing device (mouse) was detected but a time-out occurred while attempting to access it.	Check the pointing device for proper operation. If the problem persists, replace the device. If the problem still occurs, replace the system board.
* Indicates an error code that occurs only in the Phoenix BIOS.			

POST: Boot Failure Messages

Description

Once the POST test and initialization procedures are completed, POST boots the system by issuing a software interrupt, INT 19h System Bootload. If there is a failure in this process, POST displays a message or graphic on the screen.

Unlike those associated with failures in POST test or initialization procedures, system boot messages are not preceded by a numeric error code and are not logged into the POST error log.

The table below lists the POST system boot error messages, their possible cause, and suggested solution.

Message	Possible Cause	Suggested Solution
Not a Boot Diskette	The diskette in the diskette drive does not contain a valid bootstrap sector. Either the diskette is defective or has not been formatted.	Remove diskette and retry boot. If no fixed disk is available, insert a diskette known to be bootable, then retry boot.
No Boot Device Available	A read of diskette drive 0 timed out, and no fixed disk is available from which to boot.	Retry boot. If the problem persists, replace the diskette or fixed disk.
Fixed Disk Read Failure	A read of diskette drive 0 timed out, and a read of the fixed disk failed. The fixed disk is probably defective.	Retry boot. If the problem persists, replace the fixed disk.
No Boot Sector on Fixed Disk	A read of diskette drive 0 timed out, and the bootstrap sector on the fixed disk is invalid. The fixed disk is either defective, has not been formatted, and/or has never had FDISK (or a comparable utility) run on it.	Retry boot with a diskette known to be bootable. Try formatting the fixed disk and/or running FDISK, then retry boot. If the problem persists, replace the fixed disk.

POST: Informational Messages

Description

The following messages are for informational purposes only and do not report error conditions.

Message	Meaning
xxxK Base Memory	The amount of base memory in kilobytes that tested successfully.
xxxK Expanded Memory	The amount of expanded memory in kilobytes that tested successfully.
xxxK Extended Memory	The amount of extended memory in kilobytes that tested successfully.
xxxK Extra Memory	The amount of extra memory in kilobytes that tested successfully.
xxxK Standard Memory	The amount of standard memory in kilobytes that tested successfully.
Decreasing available memory	This message immediately follows any memory error messages and indicates that the memory chips are failing.
Memory tests terminated by keystroke	This message indicates that the spacebar was pressed while the memory tests were running, which stops the memory tests.
Strike the F1 key to continue	This message indicates that an error was found during POST. Pressing the F1 key allows the system to reboot.

Phoenix CBIOS NMI error messages

The Phoenix CBIOS NMI service generates the following run time error messages:

Error Code	Message	Possible Cause	Suggested Solution
110	Memory Parity Error at *xxxxx*	Memory is failing. If the NMI handler can determine the address of the failing memory, it will be displayed in place of *xxxxx*. If the failing memory is not found, then the message will read **"Memory Parity Error ????"**.	Check the system board and all installed devices for proper operation. Replace if necessary.
111	I/O Card Parity Error at *xxxx*	The peripheral card has failed. If the NMI handler can determine the address of the failing memory, it will be displayed in place of *xxxxx*. If the failing memory is not found, then the message will read **"I/O Card Parity Error ????"**.	Check the system board and all installed devices for proper operation. Replace if necessary.
112	Watchdog Time-out	The watchdog timer has detected that the timer 0 interrupt has not been serviced. A program or device has probably failed while interrupts were off.	Reboot the system. If the problem persists, check all installed devices for proper operation. Replace if necessary. If the problem still occurs, examine all programs for errors which cause the interrupts to remain off.
113	DMA Bus Time-out	A device has driven the -BURST signal line for more than 7.8 microseconds, causing the central arbiter device (on the motherboard) to generate a bus time-out.	Check system board and all installed devices for proper operation. Replace if necessary.

continued

IBM CBIOS NMI error messages

IBM PS/2 systems have a different set of error messages. The messages are followed by the segment address of the bad memory location or ????? if it cannot be found. These error codes and messages are listed in the following table:

Error Code	Message
110	System Board Memory Failure
111	I/O Channel Check Activated
112	Watchdog Time-out
113	Direct Memory Access (DMA) Bus Time-out

Models 25 and 30: IBM PS/2 Models 25 and 30 support memory parity error in the same way as other PS/2 systems but substitute the message PARITY CHECK1 for message 110 and PARITY CHECK2 for message 111.

Time-out error handling is not supported on IBM PS/2 Model 25 and 30 systems.

INT 19h: Password Messages

Description

In MCA-based PS/2 and compatible systems, The CBIOS INT 19h Bootstrap Loader Service performs password checking after the boot sector is loaded.

Several messages relate to password control. They are described in the following table.

Message	Cause	Solution
Enter password	A password has been installed and now controls access to the system.	Enter the correct 1-7 character password.
Password Ok	A correct password has been entered.	No action.
Password has been removed	The old password has been deleted. Now there is no password protection until a new password is entered.	No action.
New password has been installed	A new password is now entered.	Use the new password to access the system.
System halted! Must power down	An incorrect password was entered. The system stops and must be rebooted.	Re-enter the correct password.
Password is incorrect	Incorrect password entered.	Re-enter the correct password.

Models 25 and 30: Password control is not available in PS/2-compatible systems (such as Models 25 and 30) which are not MCA-based.

Model 25 and 30 Beep Codes

Beep Code	Contents Port 90h	Description
none	01h	CPU register test in progress
1-1-4	03h	ROM BIOS checksum failure
1-2-2	05h	DMA initialization failure
1-2-3	06h	DMA page register write/read failure
1-3-1	08h	RAM refresh verification failure
1-3-2	09h	First 64K RAM test in progress
1-3-3	0Ah	First 64K RAM chip or data line failure multi-bit
1-4-1	0Ch	Address line failure first 64K RAM
1-4-2	0Dh	Parity failure first 64K RAM
2-1-1	10h	Bit 0 first 64K RAM chip or data line failure
2-1-2	11h	Bit 1 first 64K RAM chip or data line failure
2-1-3	12h	Bit 2 first 64K RAM chip or data line failure
2-1-4	13h	Bit 3 first 64K RAM chip or data line failure
2-2-1	14h	Bit 4 first 64K RAM chip or data line failure
2-2-2	15h	Bit 5 first 64K RAM chip or data line failure
2-2-3	16h	Bit 6 first 64K RAM chip or data line failure
2-2-4	17h	Bit 7 first 64K RAM chip or data line failure
2-3-1	18h	Bit 8 first 64K RAM chip or data line failure
2-3-2	19h	Bit 9 first 64K RAM chip or data line failure
2-3-3	1Ah	Bit 10 first 64K RAM chip or data line failure
2-3-4	1Bh	Bit 11 first 64K RAM chip or data line failure
2-4-1	1Ch	Bit 12 first 64K RAM chip or data line failure
2-4-2	1Dh	Bit13 first 64K RAM chip or data line failure
2-4-3	1Eh	Bit 14 first 64K RAM chip or data line failure
2-4-4	1Fh	Bit 15 first 64K RAM chip or data line failure
none	21h	Master DMA register initialization
3-1-3	22h	Master interrupt mask register failure
none	25h	Interrupt vector loading in progress
3-2-4	27h	Keyboard failure
none	2Ch	Screen initialization failure
3-4-3	2Eh	Search for video ROM in failure

Model 25 and 30 POST error message table

Message	Possible Cause	Solution
Diskette configuration error	The specified configuration is not supported.	Change the configuration.
Diskette drive reset failed	The diskette adapter has failed.	Check the diskette adapter.
Diskette drive 1 seek failure	The B: drive failed or is missing.	Check the B: drive.
Diskette drive 0 seek failure	The A: drive has either failed or is missing.	Check the A: drive.
Diskette drive reset failed	The diskette adapter has failed.	Check the diskette adapter.
Diskette read failure — strike F1 to retry boot	The diskette is either not formatted or defective.	Replace the diskette with a bootable diskette and retry boot.
Display adapter failed; using alternate	▪ The color/monochrome switch is incorrect. ▪ The primary video adapter failed.	▪ Change the switch to the correct setting. ▪ Check the primary video adapter.
Errors found disk X: Failed Initialization	POST reports fixed disk configuration information is incorrect.	Rerun SETUP and enter correct fixed disk information.
Errors found incorrect configuration information memory size miscompare	POST reports the size of base or expansion memory does not agree with configuration information.	Rerun SETUP and enter correct memory size.
Fixed disk configuration error	The specified configuration is not supported.	Correct the fixed disk configuration.
Fixed disk controller failure	The controller card has failed.	Replace the controller card.
Fixed disk failure	Bad disk.	Retry boot. If that doesn't work, replace the fixed disk.
Fixed disk read failure — strike F1 to retry boot	The fixed disk is defective.	Retry boot. If that doesn't work, replace the fixed disk.
Invalid configuration information — please run SETUP program	▪ Memory size is incorrect. ▪ Display adapter is configured incorrectly. ▪ Wrong number of diskette drives.	Run the SETUP utility.

continued

Model 25 and 30 POST error message table, cont'd

Message	Possible Cause	Solution
Keyboard clock line failure Keyboard data line failure	Either the keyboard or the keyboard cable connection is defective.	Make sure the keyboard cable and keyboard are connected properly.
Keyboard controller failure	The keyboard controller firmware has failed.	Check the keyboard controller.
Keyboard is locked — please unlock	The keyboard lock located at the front of the computer is activated.	Unlock the keyboard.
Keyboard stuck key failure	A key(s) is jammed.	Try pressing the key(s) again.
Memory address line failure at *hex-value*, read *hex-value*, expecting *hex-value*	Circuitry associated with the memory chips has failed.	Check the circuitry.
Memory data line failure at *hex-value*, read *hex-value*, *hex-value*	One of the memory chips or associated circuitry has failed.	Replace the memory chips.
Memory high address line failure at *hex-value*, read *hex-value*, expecting *hex-value*	Circuitry associated with the memory chips has failed.	Check the circuitry.
Memory double word logic failure at *hex-value*, read *hex-value*, expecting *hex-value*	Memory chip circuitry failed.	Replace the memory chip.
Memory odd/even logic failure at *hex-failure*, read *hex-value*, expecting *hex-value*	Circuitry associated with the memory chips has failed.	Check the circuitry.
Memory parity failure at *hex-value*, read *hex-value*, expecting *hex-value*	One of the parity memory chips has failed.	Try replacing the memory chips.
Memory write/read failure at *hex-value*, read *hex-value*, expecting *hex value*	One of the memory chips has failed.	Try replacing the memory chips.

continued

Model 25 and 30 POST error message table, cont'd

Message	Possible Cause	Solution
No boot device available — strike F1 to retry boot	Either diskette drive A:, the fixed disk, or the diskette itself is defective.	Retry boot. If that doesn't work, replace the floppy diskette or the fixed disk.
No boot sector on fixed disk — strike F1 to retry boot	The C: drive is not formatted or is not bootable.	Format the C: drive, make it bootable.
Not a boot diskette — strike F1 to retry boot	The diskette in drive A: is not formatted as a bootable diskette.	Replace the diskette with a bootable diskette and retry boot.
No timer tick interrupt	The timer chip has failed.	Check the timer chip on the system board.
Hex-value optional ROM bad checksum = *hex-value*	The peripheral card contains a defective ROM.	Replace the peripheral card.
Shutdown failure	The keyboard controller or its associated logic has failed.	Check the keyboard controller.
Time-of-day clock stopped	The Time-of-day clock chip has failed.	Run the SETUP utility.
Time-of-day not set — please run SETUP program	Clock not set.	Run the SETUP utility.
Timer chip counter 2 failed	Chip failed.	Check the timer chip system board.
Timer or interrupt controller bad	Either the timer chip or the interrupt controller is defective.	Check the timer chip or the interrupt controller on the system board.
Unexpected interrupt in protected mode	The nonmaskable interrupt (NMI) port can't be disabled.	Check the system board, particularly the logic associated with the nonmaskable interrupt.

continued

Model 25 and 30 POST Informational Messages

Informational messages require no action but provide valuable information.

Message	Meaning
xxxK Base Memory	The amount of base memory in kilobytes that tested successfully.
xxxK Expanded Memory	The amount of expanded memory in kilobytes that tested successfully.
xxxK Extended Memory	The amount of extended memory in kilobytes that tested successfully.
xxxK Extra Memory	The amount of extra memory in kilobytes that tested successfully.
xxxK Standard Memory	The amount of standard memory in kilobytes that tested successfully.
Decreasing available memory	This message immediately follows any memory error messages and indicates that the memory chips are failing.
Memory tests terminated by keystroke	This message indicates that the spacebar was pressed while the memory tests were running, which stops the memory tests.
Strike the F1 key to continue	This message indicates that an error was found during POST. Pressing the F1 key allows the system to reboot.

continued

Model 25 and 30 Run Time Messages

The Model 25/30 bus may generate error messages if I/O or RAM errors occur. The possible errors are:

Error Message	Suggested Solution
Memory address line failure at *hex-value* ... expecting *hex-value*	Failed memory chip
Memory data line failure at *hex-value* ... expecting *hex-value*	Failed memory chip
Memory double word logic failure at *hex-value* ... expecting *hex-value*	Failed memory chip
Memory high address line failure at *hex-value* ... expecting *hex-value*	Failed memory chip
Memory odd/even logic failure at *hex-value* ... expecting *hex-value*	Failed memory chip
Memory parity failure at *hex-value* ... expecting *hex-value*	Failed memory chip
Memory parity error at *xxxxx*	Failed memory chip
Memory read/write failure at *hex-value* ... expecting *hex-value*	Failed memory chip
hex-value Optional ROM bad checksum = *hex-value*	I/O card failed
I/O card parity error at *xxxx*	I/O card failed

B

Appendix B
Keyboard/Mouse Controller

Overview

Introduction

The overall CBIOS hardware environment is described in Chapter 2. However, because it is so integral to the proper functioning of MCA-based PS/2 computers, the 8042 peripheral controller chip, as well as its accompanying keyboard/mouse control firmware, are discussed in greater detail here.

In this appendix

This appendix presents the following information:

- 8042 Hardware
- 8042 Firmware
- 80x86/8042 Device Interface
- System Commands to 8042
- 8042-to-System Replies
- System Commands to Keyboard
- Keyboard Replies to System
- System Commands to the Mouse
- Mouse-to-System Replies

8042 Hardware

Introduction

The Intel 8042 peripheral controller is a single chip microcomputer that can be programmed to allow bidirectional communication between the master microprocessor and up to two auxiliary serial input devices. The 8042 chip is generally mounted on the system motherboard. 8042 programs reside as firmware in the 8042 chip itself.

Note: Here the master processor, i.e. the 80286 or 80386, is referred to as the "system."

Device support

The kind of devices a given 8042 chip supports are dependent on how the 8042 is programmed.

On IBM MCA-compatible systems, the 8042 is programmed to allow bi-directional communication between the system and the keyboard, as well as between the system and one other auxiliary serial device, such as a mouse, joystick, or trackball.

8042 Keyboard/Mouse Controller functions

The 8042-based Keyboard/Mouse Controller in PS/2 MCA-compatible systems supports the following functions:

- Bidirectional keyboard interface
- Key code to 8042 scan code translation
- Keyboard password support
- 8042 keyboard inhibit switch
- Access to processor RESET signal
- Controlling gate A20 signal
- Bidirectional auxiliary device interface
- INT 74h interface to BIOS and auxiliary device driver
- Keyboard and auxiliary device error checking

continued

8042 microprocessor features

The 8042 peripheral controller chip contains the following features:

Feature	Description
Data Bus Input Register	8-bit write-only register accessed by the system at I/O addresses 064h and 060h.
Data Bus Output Register	8-bit read-only register accessed by the system at I/O address 060h.
Status Register	8-bit read-only register accessed by the system at I/O address 064h.
2KB ROM	Contains the 8042 firmware
256 Bytes RAM	Locations 20h to 3Fh contain 8042 firmware data definitions and can be read and written to by the system. The remaining RAM is reserved as 8042 firmware work area.
8-Bit Timer	Controls 8042 scan rate. (See "Timing Facts" below)
8-Bit Input Port	Data/Commands received from the keyboard or auxiliary device are transmitted through the 8042 input port.
8-Bit Output Port	Data/Commands sent from the 8042 to the keyboard or auxiliary device are transmitted from the 8042 output port.
Keyboard Clock Input Pin	Clock input for keyboard.
Aux. Device Input Pin	Clock input for auxiliary device.

8042 timing facts

The 8042 operates at a crystal frequency of 10 MHz. This gives a minimum count for the timer of 50 microseconds. When the 8042's 8-bit timer is started, it will increment to its maximum count (FFh), overflow to zero and continue to count. Incrementing from maximum to zero (overflow) sets the 8042 timer flag and generates an interrupt request to the 8042.

Note: Operating 8042 chips at a crystal rate other than 10 MHz may cause incompatibility problems.

continued

8042 Status Register definition

The 8042 Status register is an 8-bit read-only register that is accessed by the system at I/O address 064h. The values of the Input Buffer Flag, Output Buffer Flag, Auxiliary Output Buffer Flag, and Command/Data Flag help coordinate the bidirectional transfer of information between the 8042 and the system.

Status Register bit definitions

The bit definitions of the status register are as follows:

Bit #	Definition
Bit 7	PARITY ERROR. When this bit is a 1, it indicates that the last byte received from the keyboard or auxiliary device had the incorrect parity.
Bit 6	GENERAL TIME-OUT. When this bit is a 1, it indicates that a time-out has occurred during the transmission of data between the 8042 and the keyboard or the auxiliary device.
Bit 5	AUXILIARY DEVICE OUTPUT BUFFER FULL. When this bit is a 1, data received from the auxiliary device is present in the output buffer.
Bit 4	INHIBIT SWITCH. Since there is no inhibit switch in the PS/2, this bit must be a 1, to indicate that the keyboard is not inhibited.
Bit 3	COMMAND/DATA (F1). When this bit is a 1, the byte in the input buffer is a command for the 8042. When this bit is a 0, the byte in the input buffer is either a data byte for the 8042 or a command or data byte for the keyboard.
Bit 2	SYSTEM FLAG (F0). The 8042 replicates this bit from the system bit in the command byte.
Bit 1	INPUT BUFFER FULL. When this bit is a 1, a byte is present in the input buffer.
Bit 0	OUTPUT BUFFER FULL. When this bit is a 1, a byte is present in the output buffer.

continued

I/O Port definitions

The table below lists the bit definitions of the 8042 Input Port, Output Port, and Test Inputs.

Port	Bit Definition
Input	Bit 0 = Keyboard data input Bit 1 = Auxiliary device data input Bit 2 = Undefined Bit 3 = Fuse status of +5V dc line 0 = Fuse is open 1 = Fuse is closed Bit 4 = Undefined Bit 5 = Undefined Bit 6 = Undefined Bit 7 = Undefined
Output	Bit 0 = System reset Bit 1 = Gate A20 Bit 2 = Auxiliary device data output Bit 3 = Auxiliary device clock output Bit 4 = Output buffer full Bit 5 = Auxiliary output buffer full Bit 6 = Keyboard clock output Bit 7 = Keyboard data output
Test	T0 = Keyboard clock input T1 = Auxiliary device clock input

8042 Firmware

Description

On MCA-based, IBM PS/2 and compatible systems, the 8042 peripheral controller is programmed with firmware that supports the keyboard and one auxiliary device, usually a mouse.

The 8042 firmware:

- processes commands received by the 8042 from the system,
- passes through those commands intended for the keyboard or the auxiliary device,
- coordinates input from both the keyboard and the auxiliary device,
- outputs keyboard or auxiliary device data to the data bus,
- pulses IRQ 1 to initiate INT 09h, the CBIOS Keyboard Service ISR, and
- pulses IRQ 12 to initiate INT 74h, the Auxiliary Device ISR.

Phoenix scan code support

IBM-compatible keyboards that are attached to MCA-based PS/2 systems generate IBM keyboard scan code set 2, as defined in Chapter 8. Phoenix's 8042 firmware supports keyboard scan code set 2 only.

continued

8042 firmware data definitions

The 8042 Peripheral Controller chip contains 256 bytes of internal RAM. Firmware for the 8042 has the ability to read/write to 8042 RAM locations 020h to 03Fh. The 8042 firmware data is defined in the table below.

Location	Initial Value	Description
20h	30h	Command byte The command byte is a byte that is stored in the 8042 at RAM location 20h. It contains 8 flag bits that are used by the program. The command byte can be written or read by the system. The bit definition of the command byte is as follows: Bit 7 = Reserved, must be 0. Bit 6 = IBM PC Compatibility Mode. When this bit is a 1, the 8042 translates the scan codes received from the keyboard to those used by the IBM PC. When this bit is a 0, the 8042 passes the scan codes to the system without translation. Bit 5 = Disable Auxiliary Device. When this bit is a 1, the 8042 disables the auxiliary device by setting the auxiliary device clock line low. Bit 4 = Disable Keyboard. When this bit is a 1, the 8042 disables the keyboard by setting the keyboard clock line low. Bit 3 = Reserved Bit 2 = System Flag. The value of this bit is written to the system flag bit (F0) of the status register. Bit 1 = Enable Auxiliary Output Buffer Full Interrupt. When this bit is a 1, the 8042 generates an interrupt to the system by setting the Auxiliary OBF port bit (output port 2, bit 5) high, when it places a byte in the 8042 output buffer. Bit 0 = Enable Output Buffer Full Interrupt. When this bit is a 1, the 8042 generates an interrupt to the system by setting the OBF port bit (output port 2, bit 4) high, when it places a byte in the 8042 output buffer.
21h	01h	Number of times 8042 will send Resend command to device, if data received with error.
22h	0Bh	If not 0, device is read for a response to a transmission.
23h	00h	Running count of number of Resends sent to keyboard since power-up.
25h	02h	Initialized, but never used.

continued

8042 firmware data definitions, cont'd

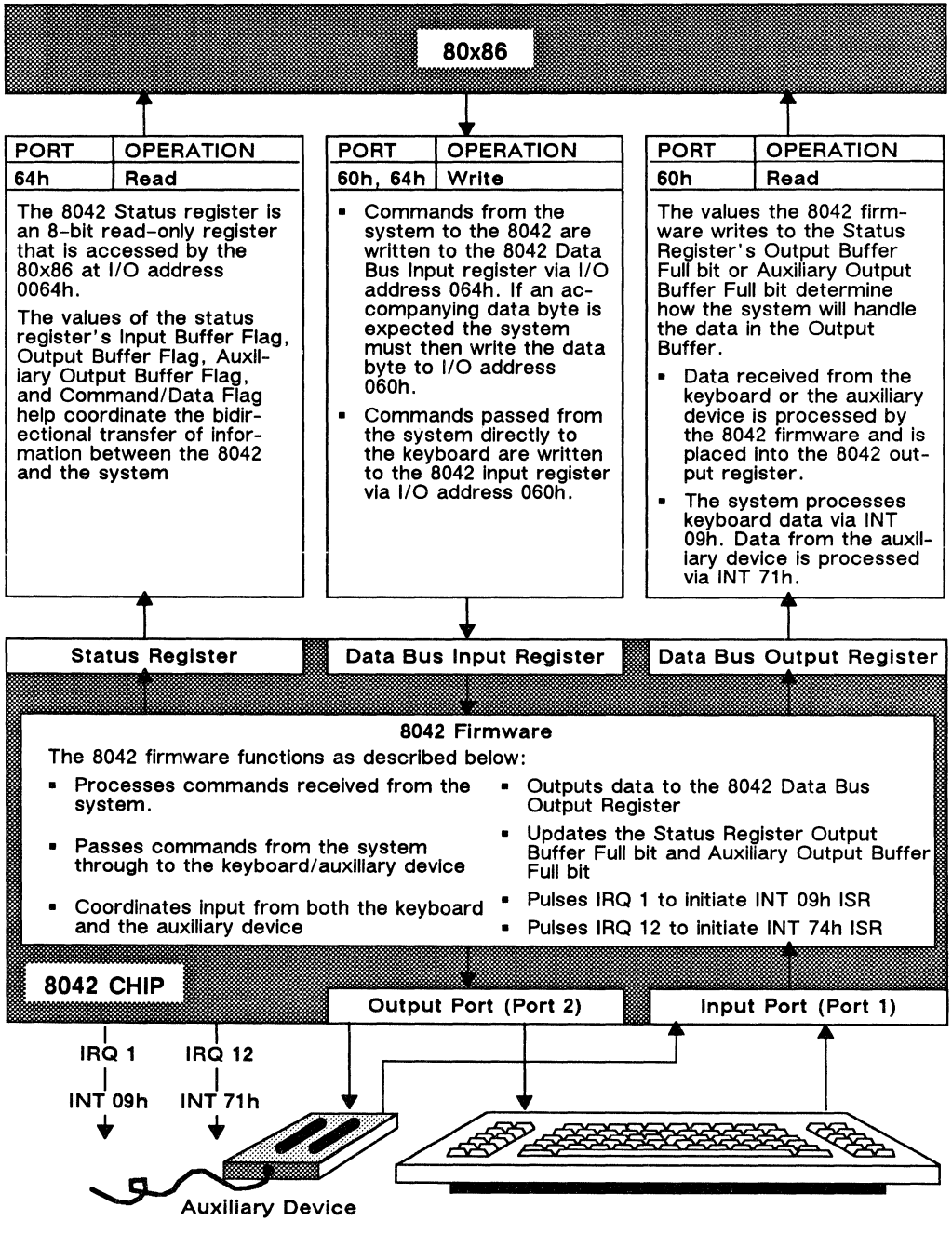
Location	Initial Value	Description
27h	F8h	Timer value.
28h	CEh	Timer value.
29h	0Bh	Initialized, but never used.
2Ah	10h	Initialized, but never used.
2Ch	15h	Initialized, but never used.
2Dh	00h	= 80h when break code (F0h) received from keyboard = 00h after adding 80h to key scan code
30h	0Bh	If not 0, device is read for a response to a transmission.
31h	00h	Running count of number of Resends sent to auxiliary device since power-up.
33h	00h	Code for null (FFh).
34h	00h	Code for null (FFh).
36h	00h	Code for left shift (2Ah).
37h	00h	Code for right shift (36h).
Note: The initial values shown here are set by the 8042 firmware initialization at run time. They may be changed by the CBIOS POST or other routines at a later time.		

80x86/8042 Device Interface

Introduction

The figure below illustrates the relationship between the system, the 8042 peripheral controller chip, the keyboard, and the auxiliary device.

continued


80x86
PORT
OPERATION
64h
Read
The 8042 Status register is an 8-bit read-only register that is accessed by the 80x86 at I/O address 0064h.
The values of the status register's Input Buffer Flag, Output Buffer Flag, Auxiliary Output Buffer Flag, and Command/Data Flag help coordinate the bidirectional transfer of information between the 8042 and the system
PORT
OPERATION
60h, 64h
Write
Commands from the system to the 8042 are written to the 8042 Data Bus Input register via I/O address 064h. If an accompanying data byte is expected the system must then write the data byte to I/O address 060h.
Commands passed from the system directly to the keyboard are written to the 8042 input register via I/O address 060h.
PORT
OPERATION
60h
Read
The values the 8042 firmware writes to the Status Register's Output Buffer Full bit or Auxiliary Output Buffer Full bit determine how the system will handle the data in the Output Buffer.
Data received from the keyboard or the auxiliary device is processed by the 8042 firmware and is placed into the 8042 output register.
The system processes keyboard data via INT 09h. Data from the auxillary device is processed via INT 71h.
Status Register
Data Bus Input Register
Data Bus Output Register
8042 Firmware
The 8042 firmware functions as described below:
Processes commands received from the system.
Passes commands from the system through to the keyboard/auxiliary device
Coordinates input from both the keyboard and the auxiliary device
Outputs data to the 8042 Data Bus Output Register
Updates the Status Register Output Buffer Full bit and Auxiliary Output Buffer Full bit
Pulses IRQ 1 to initiate INT 09h ISR
Pulses IRQ 12 to initiate INT 74h ISR
8042 CHIP
Output Port (Port 2)
Input Port (Port 1)
IRQ 1
IRQ 12
INT 09h
INT 71h
Auxiliary Device

System Commands to 8042

Introduction

Commands from the system to the 8042 are written to the Data Bus Input Register, an 8-bit write-only register that is accessed by the system at I/O addresses 064h. If an accompanying data byte is expected, the system must then write the data byte to I/O address 060h.

8042 commands

All 8042 commands are shown in the following table.

Command	Description
20h	READ THE 8042 COMMAND BYTE. Instructs the 8042 to send the contents of location 20h, the command byte, to the system.
21h - 3Fh	READ THE 8042 RAM. Instructs the 8042 to send the contents of the RAM location, defined by bits 5-0 of the command, to the system.
60h	WRITE THE 8042 COMMAND BYTE. Instructs the 8042 to write the data byte following the command to the location of the command byte (20h).
61h - 7Fh	WRITE THE 8042 RAM. Instructs the 8042 to write the data byte following the command to the RAM location defined by bits 0-5 of the command.
A4h	TEST PASSWORD INSTALLED. Instructs the 8042 to check whether there is a password currently installed. If there is no password, the contents of the first location where the password would be stored will be zero. If there is a password installed, FAh is placed in the 8042 output buffer, if not, F1h is placed in the buffer.
A5h	LOAD SECURITY. Instructs the 8042 to read password data from the 8042 input buffer and store it until a null (0) is detected. The null is stored as the last byte of the password.
A6h	ENABLE SECURITY. Instructs the 8042 to check the installed password against the incoming keystrokes for a match.
A7h	DISABLE AUXILIARY DEVICE INTERFACE. Instructs the 8042 to set bit 5 of the command byte. This disables the auxiliary device by driving the auxiliary clock low.
Note: The 8042 firmware will ignore any commands written to port 064h other than those listed.	

continued

8042 commands, cont'd

Command	Description
A8h	ENABLE AUXILIARY DEVICE INTERFACE. Instructs the 8042 to clear bit 5 of the command byte. This enables the auxiliary device by driving the auxiliary clock high.
A9h	AUXILIARY INTERFACE TEST. Instructs the 8042 to test the auxiliary device clock and data lines. The result of the test is placed in the output buffer as follows: 00 — No error detected. 01 — Auxiliary device clock line is stuck low. 02 — Auxiliary device clock line is stuck high. 03 — Auxiliary device data line is stuck low. 04 — Auxiliary device data line is stuck high.
AAh	SELF TEST. Instructs the keyboard 8042 to perform internal diagnostics tests. A 55h is placed in the output buffer if no errors are detected.
ABh	KEYBOARD INTERFACE TEST. Instructs the 8042 to test the keyboard clock and data lines. The result of the test is placed in the output buffer as follows: 00 — No error detected. 01 — Keyboard device clock line is stuck low. 02 — Keyboard device clock line is stuck high. 03 — Keyboard device data line is stuck low. 04 — Keyboard device data line is stuck high.
ACh	Reserved
ADh	DISABLE KEYBOARD INTERFACE. Instructs the 8042 to set bit 4 of the command byte. This disables the keyboard by driving the keyboard clock low.
AEh	ENABLE KEYBOARD INTERFACE. Instructs the 8042 to clear bit 4 of the command byte. This enables the keyboard by driving the keyboard clock line high.
C0h	READ INPUT PORT. Instructs the 8042 to read the 8042 input port (port 1) and place the data in the 8042 output buffer.
C1h	POLL INPUT PORT LOW. Instructs the 8042 to continuously read input port 1 bits 0–3 into the status register, bits 4–7, until IBF goes high, when the next command is executed.
Note: The 8042 firmware will ignore any commands written to port 064h other than those listed.	

continued

8042 commands, cont'd

Command	Description
C2h	POLL INPUT PORT HIGH. Instructs the 8042 to continuously read input port 1 bits 4–7 into the status register, bits 4–7, until IBF goes high, when the next command is executed.
D0h	READ OUTPUT PORT. Instructs the 8042 to read the 8042 output port (port 2) and place the data in the 8042 output buffer.
D1h	WRITE OUTPUT PORT. Instructs the 8042 that the next byte of data received should be sent to the 8042 output port.
D2h	WRITE KEYBOARD OUTPUT BUFFER. Instructs the 8042 that the next byte of data received should be sent to the output buffer. The controller will generate an interrupt to the system if the interrupt bit is enabled in the command byte.
D3h	WRITE AUXILIARY DEVICE OUTPUT BUFFER. Instructs the 8042 that the next byte of data received should be sent to the output buffer. The 8042 generates an interrupt to the system if the interrupt bit is enabled in the command byte.
D4h	WRITE TO AUXILIARY DEVICE. Instructs the 8042 that the next byte of data received should be transmitted to the auxiliary device.
E0h	READ TEST INPUTS. Instructs the 8042 to read the 8042 T0 and T1 inputs and place them in the output buffer. Bit 0 represents T0 and bit 1 represents T1.
F0 - FF	PULSE OUTPUT PORT. Instructs the 8042 to pulse bits 0–3 of the output port (port 2) low for 6 microseconds. Bits 0 through 3 of the command indicate the bits to be pulsed. If the bit is a 0 then the corresponding bit on port 2 should be pulsed. If the bit is a 1 then the corresponding bit should not be pulsed.
Note: The 8042 firmware will ignore any commands written to port 064h other than those listed.	

8042-to-System Replies

Command list

There are four 8042-to-system replies.

Command	Description
F1h	NO PASSWORD INSTALLED.
FAh	PASSWORD INSTALLED.
FEh	RESEND COMMAND.
55h	NO ERROR DETECTED ON 8042 FIRMWARE SELF-TEST.

System Commands to Keyboard

Introduction

Commands passed from the system directly to the keyboard are written to I/O address 0060h. The value of the Status Register's Command/Data Flag must be 0.

Command list

Command	Description
EDh	SET/RESET LED TOGGLE STATUS INDICATORS. The Num Lock, Caps Lock, and Scroll Lock LED indicators can be turned on or off by a command from the system. The EDh command byte is written to port 060h and the keyboard responds with FAh (ACK). The system then writes the option byte to port 60h. A value of 1 means turn the LED on. The option byte takes the format below: Bit Description 7-3 Reserved (must be 00000b) 2 Caps Lock LED 1 Num Lock LED 0 Scroll Lock LED
EEh	ECHO. The system uses this command to test the keyboard. The keyboard issues an EEh (Echo) in reply to this command.
EFh	INVALID COMMAND. The keyboard does not acknowledge this command.
F0h	CHOOSE ALTERNATE SCAN CODE SET. The 8042 APCF supports scan code set 2 only.
F1h	INVALID COMMAND. The keyboard does not acknowledge this command
F2h	READ KEYBOARD ID BYTES. The keyboard acknowledges the command and sends the two keyboard ID bytes.
F3h	SET TYPEMATIC REPEAT RATE AND DELAY PERIOD. The system may set typematic rate and delay.
F4h	ENABLE. Commands the keyboard to clear its output buffer and begin scanning.
F5h	DEFAULT DISABLE. Resets all conditions within the keyboard to their power-on default state and disables scanning. The keyboard responds with ACK, clears its output buffers, and waits for the next instruction from the system.

continued

System Commands to Keyboard, Continued

Command list, cont'd

Command	Description
F6h	SET DEFAULT Resets all conditions within the keyboard to their power-on default state and enables scanning. The keyboard responds with ACK and clears all output buffers.
F7h - FDh	RESERVED
FEh	RESEND. If the system detects an error in transmission, it issues the resend command. The keyboard responds by sending the previous output.
FFh	RESET. System issues this command to invoke the keyboard's internal self test.

Keyboard Replies to System

Introduction

Replies from the keyboard to the system are read by the system at port 060h. The keyboard to system replies are listed in the table below.

Reply list

Reply	Description
00h (Code Sets 2 and 3)	OVERRUN CHARACTER When the keyboard exceeds its buffer capacity, it places 00h (the overrun character) into the bottom of the keyboard buffer. When the overrun character reaches the top of the buffer, the keyboard sends it to the system.
83ABh	KEYBOARD ID The two-byte keyboard ID is defined as 83ABh. The keyboard replies to the keyboard ID command with ACK, then sends the low ID byte followed by the high ID byte.
AAh	BAT COMPLETION CODE At power-on-reset the keyboard logic initiates a keyboard Basic Assurance Test (BAT). If the BAT was successful, then the keyboard replies with AAh, the BAT Completion Code.
FCh	BAT FAILURE CODE When the BAT is not completed successfully, the keyboard responds with FCh, the BAT Failure Code. The keyboard will then discontinue scanning, and it will wait for a Reset command or some other response from the system.
EEh	ECHO This is the keyboard's reply to EEh, the system-to-keyboard echo command.
FAh	ACKNOWLEDGE (ACK) Keyboard reply to any valid system command other than Echo or Resend. The keyboard will discard ACK if it receives another system command while it is attempting to reply with ACK.
FEh	RESEND Keyboard receipt of invalid input or input with incorrect parity causes the keyboard to issue FEh, the Resend command.
FF (Code Set 1)	OVERRUN CHARACTER When the keyboard exceeds its buffer capacity, it places 00h (the overrun character) into the bottom of the keyboard buffer. When the overrun character reaches the top of the buffer, the keyboard sends it to the system.

System Commands to the Mouse

Introduction

The command D4h, Write to Auxiliary Device, tells the 8042 firmware to transmit the next byte it receives to the auxiliary device. All commands written to the mouse must therefore be written according to the process described below:

1. D4h, Write to Auxiliary Device, must be written to port 064h.
2. nnh, the system-to-mouse command, must be written to port 060h.

Note: Unless it receives D4h in port 064h first, the 8042 firmware will pass all commands written to port 060h directly to the keyboard.

Command list

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>E6h</td><td>RESET SCALING
Resets scaling 1:1</td></tr>
<tr><td>E7h</td><td>SET SCALING 2:1
This command can only be used when the mouse is in Stream Mode. In Stream Mode, the current X/Y coordinate values are converted to new values each time the sample period expires. In 2:1 scaling, the relationship between the input and output values is as follows:
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>0
1
2
3
4
5
N (>=6)</td><td>0
1
1
3
6
9
2.0 x N</td></tr>
</table></td></tr>
<tr><td>E8h</td><td>SET RESOLUTION
This is a two-byte command. The second byte written to port 060h is interpreted as a resolution in counts per mm. There are four possible resolutions. The relationship between second byte and resolution is listed below:
<table>
<tr><th>Byte</th><th>Resolution</th></tr>
<tr><td>00h
01h
02h
03h</td><td>1 count per mm
2 counts per mm
4 counts per mm
8 counts per mm</td></tr>
</table></td></tr>
</table>

continued

Command list, cont'd

Command	Description
E9h	STATUS REQUEST This command elicits a 3-byte mouse status report. The format of the mouse status report is shown below: Byte 3 = Sampling Rate, where: Bit 7 = Most significant bit Bit 0 = Least significant bit Byte 2 = Resolution, where: Bit 7 = Most significant bit Bit 0 = Least significant bit Byte 1 = Mouse Status, where: Bit 7 = Reserved Bit 6 = 0 Stream mode = 1 Remote mode Bit 5 = 0 Disabled = 1 Enabled Bit 4 = 0 Scaling 1:1 = 1 Scaling 2:1 Bit 3 = Reserved Bit 2 = 1 Left mouse button pressed Bit 1 = Reserved Bit 0 = 1 Right mouse button pressed
EAh	SET STREAM MODE In Stream mode, the mouse transmits data to the system each time a mouse button is pressed or released, or each time the mouse detects a unit of movement. The mouse data sample rate determines the maximum number of times per second mouse data can be transmitted to the system. If no button is pressed or if the mouse is not moved, then no data is transmitted to the system. The Set Stream Mode command turns the mouse stream mode on.
EBh	READ DATA The Read Data command forces the transmission of one mouse data packet. The Read Data command is valid in both stream mode and in remote mode.
ECh	RESET WRAP MODE This command resets the mouse to normal operation.
EEh	SET WRAP MODE This command sets wrap mode, the mouse "echo" mode. With the exception of ECh, Reset Wrap Mode, and FFh, Reset Mouse, the mouse will echo back all data and commands received from the system.

continued

Command list, cont'd

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>F0h</td><td>SET REMOTE MODE
In Remote mode, the mouse data can only be transmitted in reply to a Read Data command.</td></tr>
<tr><td>F2h</td><td>READ DEVICE TYPE
The mouse ID byte is 00h. The mouse returns 00h in reply to the Read Device Command.</td></tr>
<tr><td>F3h</td><td>SET SAMPLING RATE
Sampling Rate is defined as the number of times per second the system checks for mouse data. This is a two-byte command. F3h, the Set Sampling Rate command, must be followed by a second byte, representing the hex value of the sample rate. The values listed below are allowable sample rates.
<table>
<tr><th>Hex Value (2nd Byte)</th><th>Sample Rate</th></tr>
<tr><td>0Ah
14h
28h
3Ch
50h
64h
C8h</td><td>10/second
20/second
40/second
60/second
80/second
100/second
200/second</td></tr>
</table></td></tr>
<tr><td>F4h</td><td>ENABLE
Enables data transmissions if the mouse has been set to stream mode. Has no effect in remote mode.</td></tr>
<tr><td>F5h</td><td>DISABLE
Disables mouse data transmissions if the mouse is in stream mode. Has no effect in remote mode.</td></tr>
<tr><td>F6h</td><td>SET DEFAULT
The Set Default command reinitializes the mouse to its power-on default state. The mouse power-on default state is as follows:
Sampling Rate = 100/second
Scaling = Linear Scaling
Mode = Stream mode
Resolution = 4 counts/mm
Transmissions = Disabled</td></tr>
<tr><td>FEh</td><td>RESEND
Errors in transmissions from the mouse cause the system to send the Resend command. The mouse responds by retransmitting its last data packet.</td></tr>
<tr><td>FFh</td><td>RESET
Reset mode instructs the mouse to run its internal self test routine. This command puts the mouse into reset mode.</td></tr>
</table>

Mouse-to-System Replies

Reply list

There are only two mouse-to-system replies. Both replies are related to command processing and are read by the system at port 060h.

Reply	Description
FAh	ACK (Acknowledge) The mouse replies with FAh, ACK, anytime it receives a valid command from the system. Unlike mouse serial data packets, the ACK reply is not stored in a buffer in internal memory, but is discarded immediately after it is transmitted. If a new command is received while the mouse is in the ACK reply process, the mouse discards the ACK reply and begins processing the new command immediately. **Note:** The commands ECh, Reset Wrap Mode, and FFh, Reset, are exceptions to mouse ACK behavior. The mouse does not respond with ACK to either of these commands.
FEh	RESEND The mouse replies with FEh, Resend, anytime it receives an invalid command from the system. Two invalid commands in succession cause the mouse to send the error code FCh to the system. A single, isolated invalid command, however, does not affect mouse processing in any way. The mouse simply ignores single invalid commands, maintaining its present operational state.

C

Appendix C
Hardware Environment for 8086-Based Systems

Overview

In this appendix

This appendix describes the hardware environment needed to implement the ROM BIOS on PS/2 systems (e.g. IBM Models 25 and 30) and compatibles based on the 8086 microprocessor. These systems are not MCA-compatible.

The following topics are discussed:

- Hardware Address Locations
- Microprocessor
- 8087 Math Coprocessor
- System Support Gate Array
- I/O Support Gate Array
- System Time-Related Logic
- Interrupt Controller Extensions
- System Timer
- DMA Controller
- 8259 Programmable Interrupt Controller
- Miscellaneous Additional Logic

Table: Hardware I/O Port Definitions

The following table gives the locations of the various I/O ports and their functions.

Note: An *x* before a bit setting description indicates that this bit setting has no meaning when the condition stated after the *x* is true.

I/O Address	Read/Write Status	Description
0000h	R/W	DMA channel 0, memory address register
0001h	R/W	DMA channel 0, transfer count register
0002h	R/W	DMA channel 1, memory address register
0003h	R/W	DMA channel 1, transfer count register
0004h	R/W	DMA channel 2, memory address register
0005h	R/W	DMA channel 2, transfer count register
0006h	R/W	DMA channel 3, memory address register
0007h	R/W	DMA channel 3, transfer count register
0008h	R	DMA channel 0-3, status register, where: Bit 7 = 1 Channel 3 request Bit 6 = 1 Channel 2 request Bit 5 = 1 Channel 1 request Bit 4 = 1 Channel 0 request Bit 3 = 1 Terminal count on channel 3 Bit 2 = 1 Terminal count on channel 2 Bit 1 = 1 Terminal count on channel 1 Bit 0 = 1 Terminal count on channel 0
0008h	W	DMA write command, where: Bit 7 = 0 DACK sense active low = 1 DACK sense active high Bit 6 = 0 DREQ sense active low = 1 DREQ sense active high Bit 5 = 0 Late write selection = 1 Extended write selection = x Bit 3 is one Bit 4 = 0 Fixed priority = 1 Rotating priority Bit 3 = 0 Normal timing = 1 Compressed timing = x Bit 0 is one Bit 2 = 0 Enable controller = 1 Disable controller Bit 1 = 0 Disable channel 0 address hold = 1 Enable channel 0 address hold = x Bit 0 is zero Bit 0 = 0 Disable memory to memory = 1 Enable memory to memory

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0009h	W	DMA request register, where: Bits 7-3= x Reserved Bit 2 = 0 Reset request bit = 1 Set request bit Bits 1-0= 00b Select channel 0 = 01b Select channel 1 = 10b Select channel 2 = 11b Select channel 3
000Ah	R/W	DMA single mask register, where: Bits 7-3= x Reserved Bit 2 = 0 Clear mask bit Bits 1-0= 00b Select channel 0 mask bit = 01b Select channel 1 mask bit = 10b Select channel 2 mask bit = 11b Select channel 3 mask bit
000Bh	W	DMA mode register, where: Bits 7-6= 00b Demand mode select = 01b Single mode select = 10b Block mode select = 11b Cascade mode select Bit 5 = 0 Address increment select = 1 Address decrement select Bit 4 = 0 Disable auto initialization = 1 Enable auto initialization Bits 3-2= 00b Verify transfer = 01b Write transfer = 10b Read transfer = 11b Illegal value = xx if Bits 7 and 6 on Bits 1-0= 00b Select channel 0 = 01b Select channel 1 = 10b Select channel 2 = 11b Select channel 3
000Ch	W	DMA clear byte pointer. Sets a low byte/high byte flip-flop for the dual read/write ports to a known state, where the next access will be to the low byte register.
000Dh	R	DMA read temporary register. Holds data during memory-to-memory transfers.
000Dh	W	DMA master clear (Resets the 8237. The command, status, request, temporary, and internal first/last flip-flop registers are cleared, and the mask register is set. The 8237 is in the idle state).
000Eh	W	DMA clear mask register (Clears all four channel masks so they can accept DMA requests)

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
000Fh	W	DMA write all mask register bits, where: Bits 7-4 = x Reserved Bit 3 = 0 Clear channel 0 mask bit = 1 Set channel 0 mask bit Bit 2 = 0 Clear channel 1 mask bit = 1 Set channel 1 mask bit Bit 1 = 0 Clear channel 2 mask bit = 1 Set channel 2 mask bit Bit 0 = 0 Clear channel 3 mask bit = 1 Set channel 3 mask bit
0010h - 001Fh		Reserved
0020h	R	PIC Interrupt request/In-service registers programmed by Operation Command Word 3 (OCW3): Interrupt request register, where: Bits 7-0 = 0 No active request for the corresponding interrupt line = 1 Active request for the corresponding interrupt line Interrupt in-service register, where: Bits 7-0 = 0 The corresponding interrupt line is not currently being serviced = 1 The corresponding interrupt line is currently being serviced
0020h	W	PIC Initialization Command Word 1 (ICW1) (Bit 4 is one), where: Bits 7-5 = 000 Only used in 80/85 mode Bit 4 = 1 Reserved Bit 3 = 0 Edge triggered mode = 1 Level triggered mode Bit 2 = 0 Successive interrupt vectors are separated by eight bytes = 1 Successive interrupt vectors are separated by four bytes Bit 1 = 0 Cascade mode = 1 Single mode - no ICW3 needed Bit 0 = 0 No ICW4 needed = 1 ICW4 needed
0021h	W	PIC ICW2, ICW3, or ICW4 in sequential order after ICW1 written to Port 0020h ICW2, where: Bits 7-3 = Address lines A0-A3 of base vector address for interrupt controller Bits 2-0 = 000 Reserved

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0021h (cont'd)		PIC ICW3, where: Bits 7-0= 0 Slave controller not attached to corresponding interrupt pin = 0 Slave controller attached to corresponding interrupt pin ICW4, where: Bits 7-5= 000 Reserved Bit 4 = 0 No special fully-nested mode = 1 Special fully-nested mode Bits 3-2= 00 Nonbuffered mode = 01 Nonbuffered mode = 10 Buffered mode/slave = 11 Buffered mode/master Bit 1 = 0 Normal EOI = 1 Auto EOI Bit 0 = 0 80/85 mode = 1 8086/8088 mode
0021h	R/W	PIC Interrupt mask register (OCW1), where: Bit 7 = 0 Enable parallel printer interrupt Bit 6 = 0 Enable diskette interrupt Bit 5 = 0 Enable fixed disk interrupt Bit 4 = 0 Enable serial interrupt Bit 3 = 0 Reserved Bit 2 = 0 Enable video interrupt Bit 1 = 0 Enable keyboard/pointing device/RTC interrupt Bit 0 = 0 Enable timer interrupt
0021h	W	PIC OCW2 (Bit 4 is zero, Bit 3 is zero), where: Bits 7-5 = 000 Rotate in automatic EOI mode (clear) = 001 Nonspecific EOI = 010 No operation = 011 Specific EOI = 100 Rotate in automatic EOI mode (set) = 101 Rotate on nonspecific EOI command = 110 Set priority command = 111 Rotate on specific EOI command Bit 4 = 0 Reserved Bit 3 = 0 Reserved Bits 2-0 = Interrupt request to which the command applies

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0020h	W	PIC OCW3 (Bit 4 is zero, Bit 3 is one), where: Bit 7 = 0 Reserved Bits 6-5 = 00 No operation = 01 No operation = 10 Reset special mask = 11 Set special mask Bit 4 = 0 Reserved Bit 3 = 1 Reserved Bit 2 = 0 No poll command = 1 Poll command Bits 1-0 = 00 No operation = 01 No operation = 10 Read interrupt request register on next read at Port 0020h = 11 Read interrupt in-service register on next read at Port 0020h
0024h-003Fh		Reserved
0040h	R/W	System timer counter 0 (Current 16-bit count for timer 0)
0041h	R/W	System timer counter 1 (Current 16-bit count for timer 1)
0042h	R/W	System timer counter 2 (Current 16-bit count for timer 2)
0043h	W	System timer control register, where: Bits 7-6 = 00 Select counter 0 = 01 Select counter 1 = 10 Select counter 2 = 11 Illegal Bits 5-4 = 00 Counter latching operation = 01 Read/load least significant byte only = 10 Read/load most significant byte only = 11 Read/load least significant byte, then most significant byte Bits 3-1 = 000 Mode 0 = 001 Mode 1 = x10 Mode 2 = x11 Mode 3 = 100 Mode 4 = 101 Mode 5 Bit 0 = 0 Binary counter (16 bits) = 1 BCD counter (four digits)
0044h-005Fh		Reserved
0060h	R	Keyboard/Pointing device output buffer

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0061h	W	System control port, where: Bit 7 = x Reserved Bit 6 = x Reserved Bit 5 = 0 Enable I/O channel check Bit 4 = 0 Enable RAM parity check Bit 3 = x Reserved Bit 2 = x Reserved Bit 1 = x Beeper data Bit 0 = 1 Enable timer 2 gate to speaker
0062h	R	System board status, where: Bit 7 = 1 Parity error occurred Bit 6 = 1 I/O channel check active Bit 5 = x Timer 2 output status Bit 4 = x Reserved Bit 3 = x Reserved Bit 2 = 0 Disk installed Bit 1 = 1 Coprocessor installed Bit 0 = x Reserved
0063h	R	Software vector number, where: Bits 7-3= Software vector number of interrupt vector for hardware interrupt 1 minus one
0064h	W	Keyboard status port
0065h	R/W	Planar control register, where: Bit 7 = 1 Enable parallel port output Bit 6 = 0 Reserved Bit 5 = 0 Reserved Bit 4 = 1 Select serial chip Bit 3 = 1 Select diskette chip Bit 2 = 1 Select video chip Bit 1 = 1 Select parallel port chip Bit 0 = 1 Select fixed disk chip
0066h	R/W	Keyboard/pointing device control port 1, where: Bit 7 = 0 Enable keyboard/pointing device command start Bit 6 = 0/1/0 System ready for keyboard/auxiliary data Bit 5 = 0 Enable keyboard/auxiliary command start Bit 4 = 0/1/0 System ready for keyboard/pointing device data Bit 3 = 1 Keyboard/pointing device attached to Port 0067h Bit 2 = 0 Keyboard unlocked = 1 Keyboard locked Bit 1 = 0 Port 0068h enabled Bit 0 = 0 Port 0067h enabled

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Device Description
0067h	R	Device 2 data port (data from keyboard input here)
0067h	W	Device 2 command port
0068h	R	Device 1 data port (data from mouse come here)
0068h	W	Device 1 command port (commands to the mouse sent from here)
0069h	W	Device 1 and 2 control port 2, where: Bits 7-5 = x Reserved Bit 4 = I/O pointing device/keyboard command start Bit 3 = I/O pointing device/ keyboard command start Bit 2 = 1 POST completion Bit 1-0 = Reserved
006Ah	R/W	Undefined system port, where: Bits 7-6= 00 Reserved Bit 5 = 1 Device 0 data ready Bits 4-3= 00 Reserved Bit 2 = 1 Device 1 data ready Bit 1 = 0 Reserved Bit 0 = x Diskette related to rate
006Bh	R/W	Planar RAM control/status register, where: Bit 7 = 0 Upper 512K failed (Parity check pointer) = 1 Lower 128K failed Bit 6 = 1 Enable RAM, 90000h - 9FFFFh Bit 5 = 1 Enable RAM, 80000h - 8FFFFh Bit 4 = 1 Enable RAM, 70000h - 7FFFFh Bit 3 = 1 Enable RAM, 60000h - 6FFFFh Bit 2 = 1 Enable RAM, 50000h - 5FFFFh Bit 1 = 1 Enable RAM, 40000h - 4FFFFh Bit 0 = 1 Remap low memory
006Ch-0076h		Reserved
0077h	W	Undefined system port
0078h-0079h		Reserved
0080h	R/W	DMA page register channel 2
0081h	R/W	DMA page register channel 3
0082h	R/W	DMA page register channel 1
0083h-0086h		Reserved
0087h	R/W	DMA page register channel 0

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Address	Description
0088h-008Fh		Reserved
0090h	R/W	Manufacturing port register (flags various stages of POST testing in progress)
0091h-009Fh		Reserved
00A0h	R	Interrupt controller extension request register, where: Bit 7 = 1 Enable NMI = 0 Disable NMI Bits 6-4 = x Reserved Bit 3 = 1 Device 1 (Port 0068h) interrupt Bit 2 = 1 Device 0 (Port 0067h) interrupt Bits 1-0 = Reserved
00A1h	R/W	Interrupt controller extension mask register, where: Bits 7-1 = Reserved Bit 0 = 1 Enable mask for RTC alarm interrupt
00A2h-00AFh		Reserved
00B0h	R	Real time clock alarm clear register (clears the alarm interrupt condition of the clock chip when read)
00B1h	W	Real time clock alarm enable register, where: Bits 7-1 = x Reserved Bit 0 = 1 Alarm enabled
00B2h		Reserved
00B3h	R/W	Real time clock enable register, where: FFh = Enable clock operation
00B4h	R/W	Real time clock status register, where: Bits 7-1 = 0 Reserved Bit 0 = 1 Clock update in progress
00B5h-00DFh		Reserved
00E0h	R/W	Real time clock 100 microsecond counter, where: Bits 7-4 = 100 microsecond count in BCD Bits 3-0 = 0 Reserved
00E1h	R/W	Real time clock millisecond counter, where: Bits 7-4 = 1 100 millisecond counter in BCD Bits 3-0 = 0 10 millisecond counter in BCD
00E2h	R/W	Real time clock seconds register (in BCD)

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Device Description
00E3h	R/W	Real time clock minutes register (in BCD
00E4h	R/W	Real time clock hours register (in BCD)
00E5h	R/W	Real time clock day of week (not used by the BIOS), where: Bits 7-3= 0 Reserved Bits 2-0= Day of week
00E6h	R/W	Real time clock day register (day of the month in BCD)
00E7h	R/W	Real time clock month register (in BCD)
00E8h		Real time clock month encode register 1 Bit 7 = 1 Reserved Bit 6 = 1 Reserved Bit 5 = Encoded month (bit 3) Bit 4 = Encoded month (bit 2) Bits 3-0= 0000 Reserved
00E9h	R/W	Real time clock year register. Year (module 100) in BCD format.
00EAh	R/W	Real time clock alarm seconds register (alarm time in seconds in BCD; set to FFh when alarm is not set)
00EBh	R/W	Real time clock alarm minutes register (alarm time in minutes in BCD; set to FFh when alarm is not set)
00ECh	R/W	Real time clock alarm hours register (alarm time in hours in BCD; set to FFh when alarm is not set)
00EDh	R/W	Real time clock month encode register 2, where: Bit 7 = 1 Reserved Bit 6 = 1 Reserved Bit 5 = 0 Reserved Bit 4 = 0 Reserved Bit 3 = 0 Reserved Bit 2 = 0 Reserved Bit 1 = Encoded month (bit 1) Bit 0 = Encoded month (bit 0)

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
00EEh	R/W	Real Time Clock checksum register 1, where: Bit 7 = 1 Reserved Bit 6 = 1 Reserved Bit 5 = 1 Bit 5 of checksum Bit 4 = 1 Bit 4 of checksum inverted Bit 3 = 1 Reserved Bit 2 = 1 Reserved Bit 1 = 0 Reserved Bit 0 = 0 21st century = 1 20th century
00EFh	R/W	Real Time Clock checksum register 2, where: Bit 7 = 1 Reserved Bit 6 = 1 Reserved Bit 5 = Bit 3 of checksum inverted Bit 4 = Bit 2 of checksum Bit 3 = 1 Reserved Bit 2 = 1 Reserved Bit 1 = Bit 1 of checksum Bit 0 = Bit 0 of checksum inverted
00F0h-00FEh		Reserved
00FFh	W	Undefined system port
0100h-0277h		Reserved
278h	R/W	Parallel 3 data port
279h	R/W	Parallel 3 status port (see 0379h)
27Ah	R/W	Parallel 3 control port (see 037Ah)
27Bh-2F1h		Reserved
02F2h		Adapter interrupt register level 2
02F3h		Adapter interrupt register level 3
02F4h		Adapter interrupt register level 4
02F5h		Adapter interrupt register level 5
02F6h		Adapter interrupt register level 6
02F7h		Adapter interrupt register level 7
2F8-2FFh		Serial register 2 (see 3F8h-3FFh)
031Fh		Reserved
0320h	R/W	Fixed disk adapter control register
0321h		Reserved

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0322h	W	Fixed disk adapter control register, where: Bit 7 = 1 Reset Bit 6 = 1 Reserved (except during reset) Bit 5 = 0 Force 8-bit programmed I/O (PIO) mode Bit 4 = 0 Reserved Bit 3 = 0 Reserved Bit 2 = 0 Force 8-bit PIO mode Bit 1 = 0 Interrupt through adapter status register (324) = 1 Interrupt through programmable interrupt controller Bit 0 = 0 Programmed I/O (PIO) mode = 1 DMA mode
0322h	R	Fixed disk adapter status register, where: Bit 7 = 0 Reserved Bit 6 = 0 Reserved Bit 5 = 0 Force 8-bit PIO mode Bit 4 = 1 Data transfer requested by adapter Bit 3 = 0 Data direction is system to adapter = 1 Data direction is adapter to system Bit 2 = 1 Adapter busy Bit 1 = 1 Interrupt occurred flag Bit 0 = 1 Transfer in progress
0323h		Reserved
0324h	W	Fixed disk adapter attention register, where: Bit 7 = 1 System request to send CCB (command control block) Bit 6 = 0 Reserved Bit 5 = 1 System request to send sense summary block Bit 4 = 1 Data transfer requested by system Bit 3 = 0 Reserved Bit 2 = 0 Reserved (Select drive 0) Bit 1 = 0 Reserved Bit 0 = 0 Abort current command
0324h	R	Fixed disk adapter interrupt status register, where: Bit 7 = 1 Termination error, error in bits 6-0 Bit 6 = 1 Invalid command Bit 5 = 1 Command reject Bit 4 = 0 Reserved Bit 3 = 0 Reserved Bit 2 = 0 Reserved (Select drive 0) Bit 1 = 1 Error recovery procedure invoked Bit 0 = 1 Equipment check
0325h-0377h		Reserved
0378h	R/W	Parallel 2 data port

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
0379h	R/W	Parallel 2 status port, where: Bit 7 = 0 Busy Bit 6 = 0 Acknowledge Bit 5 = 1 Out of paper Bit 4 = 1 Printer selected Bit 3 = 0 Error Bit 2 = 0 IRQ occurred Bit 1 = 0 Reserved Bit 0 = 0 Reserved
037Ah	R/W	Parallel 2, control port, where: Bits 7-6= 0 Reserved Bit 5 = 0 Direction is write to port = 1 Direction is read to port Bit 4 = 1 Enable IRQ Bit 3 = 1 Select printer Bit 2 = 0 Initialize printer Bit 1 = 1 Automatic line feed Bit 0 = 1 Strobe
037Bh-03B3h		Reserved
03B4h	R/W	VGA CRT controller index register (mono)
03B5h	R/W	Other VGA CRT controller registers (mono)
03BAh	R	VGA input status register 1 (mono)
03BAh	W	VGA feature control register (mono)
3BCh	R/W	Parallel 1 data port
3BDh	R/W	Parallel 1 status port (see 379h)
3BEh	R/W	Parallel 1 control port 37Ah)
3BFh		Reserved
03C0h	R/W	VGA attribute address register
03C0h	W	Other VGA attribute registers
03C1h	R	Other VGA attribute registers
03C2h	W	VGA miscellaneous output register
03C2h	R	VGA input status register 0
03C3h	R/W	VGA video subsystem enable
03C4h	R/W	VGA sequencer address register
03C5h	R/W	Other VGA sequencer registers
03C6h	R/W	Video DAC PEL mask

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
03C7h	W	Video DAC PEL address, read mode
03C7h	R	Video DAC state register
03C8h	R/W	Video DAC PEL address, write mode
03C9h	R/W	Video DAC PEL data register
03CAh	R	VGA feature control register
03CCh	R	VGA miscellaneous output register
03CEh	R/W	VGA graphics registers, address register
03CFh	R/W	VGA and other graphics registers
03D4h	R/W	VGA CRT controller index register, where: Bits 7-6= Reserved Bit 5 = Index5 Bit 4 = Index4 Bit 3 = Index3 Bit 2 = Index2 Bit 1 = Index1 Bit 0 = Index0
03D5h	R/W	VGA Data register
03D6h-03D7h		VGA registers
03D8h	R/W	CGA mode control register, where: Bit 7 = 0 Reserved Bit 6 = 0 Reserved Bit 5 = 0 No blink = 1 Blink Bit 4 = 1 Mode 6, 640 x 200 Bit 3 = 1 Enable display image Bit 2 = 0 Palette address 00h and CGA border register select color = 1 Palette address 00h-07h are the two colors selected Bit 1 = 1 Select mode 4 and 5, 320 x 200 Bit 0 = 1 Select 80 x 25 mode
03D9h	R/W	CGA border control register, where: Bit 7 = 0 Reserved Bit 6 = 0 Reserved Bit 5 = 320 x 200 palette select Bit 4 = Alternate intensity Bits 3-0= Border color

continued

Table: Hardware I/O Port Definitions, cont'd

Address	Read/Write Status	Description
03DAh	R	VGA status register, where: Bits 7-4= 0 Reserved Bit 3 = Vertical sync Bit 2 = 0 Reserved Bit 1 = 0 Reserved Bit 0 = Display enable
03DBh-03DCh		Reserved
03DDh	R/W	VGA extended mode control, where: Bit 7 = Readable DAC installed Bits 6-3= 0 Reserved Bit 2 = 256 colors Bit 1 = 0 Reserved Bit 0 = 0 Reserved
03DEh-03EFh		Reserved
03F0h	R	Diskette controller port A register, where: Bit 7 = 1 Interrupt pending Bit 6 = 0 Second drive installed Bit 5 = 1 Step (latched) Bit 4 = 1 Track 0 Bit 3 = 1 Head 1 selected Bit 2 = 1 Index Bit 1 = 0 Write protect Bit 0 = 0 Direction
03F1h	R	Diskette controller port B register, where: Bit 7 = 0 Reserved Bit 6 = 1 Select drive 1 Bit 5 = 1 Select drive 0 Bit 4 = 0 Write data (latched) Bit 3 = 0 Read data (latched) Bit 2 = 0 Write enable (latched) Bit 1 = 1 Select drive 3 Bit 0 = 1 Select drive 2
03F2h	W	Diskette controller digital output register, where: Bit 7 = 0 Enable motor drive 3 Bit 6 = 0 Enable motor drive 2 Bit 5 = 0 Enable motor drive 1 Bit 4 = 0 Enable motor drive 0 Bit 3 = 0 Enable DMA and interrupt Bit 2 = 1 Reset controller Bits 1-0= 00 Select drive 0 = 01 Select drive 1 = 10 Select drive 2 = 11 Select drive 3
03F3h		Reserved

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
03F4h	R	Diskette main status register, where: Bit 7 = Request for master Bit 6 = Stat input/output Bit 5 = Non-DMA mode Bit 4 = Diskette controller busy Bit 3,2 = Reserved Bit 1 = Drive 1 busy Bit 0 = Drive 0 busy
03F5h	R/W	Diskette controller data registers
03F6h		Reserved
03F7h	R	Diskette controller digital input register, where: Bit 7 = Diskette change Bits 6-4 = Reserved Bit 3 = DMA enable Bit 2 = No write precompensation Bit 1 = Select 250 rate Bit 0 = Reserved
03F7h	W	Diskette controller configuration control register, where: Bits 7-3 = Reserved Bit 2 = No write precompensation Bit 1 = 0 500 Kbs mode = 1 250 Kbs mode Bit 0 = Reserved
03F8h	W	Serial register 1, transmitter holding register, where: Bits 7-0 = Data bits 7 through 0, respectively, when Divisor Latch Access Bit (DLAB) = 0
03F8h	W	Serial register 1, receiver buffer register, where: Bits 7-0 = Data bits 7-0, respectively, when DLAB = 0
03F8h	R/W	Serial register 1, divisor latch, low byte, where: Bits 7-0 = Bits 7-0 of the divisor when DLAB = 1
03F9h	R/W	Serial register 1, divisor latch, high byte, where: Bits 7-0 = Bits 15-8 of the divisor when DLAB = 1

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
03F9h	R/W	Serial register 1, interrupt enable register, where: Bits 7-4 = 0 Reserved Bit 3 = 1 Enable modem status interrupt Bit 2 = 1 Enable receiver line status interrupt Bit 1 = 1 Enable transmitter holding register empty interrupt Bit 0 = 1 Enable received data available interrupt when DLAB = 0
03FAh	R	Serial register 1, interrupt identification register, where: Bits 7-3 = 0 Reserved Bits 2-1 = 00 Modem status interrupt = 01 Transmitter holding register empty interrupt = 10 Received data available register interrupt = 11 Receiver line status interrupt Bit 0 = 0 Interrupt pending
03FBh	R/W	Serial register 1, line control register, where: Bit 7 = 1 Divisor latch access = 0 Receiver buffer, transmitter holding or enable registers access Bit 6 = 1 Set break enabled Bit 5 = 1 Stick parity, if parity is enabled Bit 4 = 1 Even parity select, if parity is enabled Bit 3 = 1 Parity enable Bit 2 = 0 1 stop bit is specified if word length >5 = 1 2 stop bits are specified if word length > 5; 1.5 stop bits are specified if word length = 5 Bit 1-0 = 00b Word length is 5 bits = 01b Word length is 6 bits = 10b Word length is 7 bits = 11b Word length is 8 bits
03FCh	R/W	Serial register 1, modem control register, where: Bit 7-5 = 0 Reserved Bit 4 = 1 Loop back mode Bit 3 = 1 Enable OUT2 interrupt Bit 2 = 1 Force OUT1 active Bit 1 = 1 Force request-to-send active Bit 0 = 1 Force data-terminal-ready active

continued

Table: Hardware I/O Port Definitions, cont'd

I/O Address	Read/Write Status	Description
03FDh	R	Serial register 1, Line status register, where: Bit 7 = 0 Reserved Bit 6 = 1 Transmitter shift and holding registers empty Bit 5 = 1 Transmitter holding register empty Bit 4 = 1 Break interrupt Bit 3 = 1 Framing error Bit 2 = 1 Parity error Bit 1 = 1 Overrun error Bit 0 = 1 Data ready
03FEh	R	Serial register 1, modem status register, where: Bit 7 = 1 Data carrier detect Bit 6 = 1 Ring indicator Bit 5 = 1 Data set ready Bit 4 = 1 Clear to send Bit 3 = 1 Delta data carrier detect Bit 2 = 1 Trailing edge ring indicator Bit 1 = 1 Delta data set ready Bit 0 = 1 Delta clear to send
03FFh	R/W	Serial register 1, scratch register

Microprocessor

Requirement

The 8086 PS/2 BIOS must be implemented in a system that uses the Intel 8086-2 or an equivalent microprocessor.

Processor speeds supported

The 8086 PS/2 BIOS supports microprocessor speeds of from 4.77 to 10 MHz.

Wait states supported

The 8086 PS/2 BIOS supports either zero or one wait state.

Address modes

The 8086 PS/2 BIOS supports real address mode only. Real address mode supports 20-bit addresses, made up of 16-bit segments and 4-bit offsets, with a maximum addressable memory area of 1 MB. All segments are 64K.

8087 Math Coprocessor

Introduction

The 8086 PS/2 BIOS can be implemented in systems that include an Intel 8087 math coprocessor or its equivalent.

Description

The 8087 math coprocessor performs high speed arithmetic, logarithmic, and trigonometric floating point arithmetic calculations. This permits much speedier processing for mathematically-intensive processing applications.

The math coprocessor works in parallel with the microprocessor, allowing both to process instructions separately. See the user manual for the Intel 8087 math coprocessor for details about the extended data types, registers, and instructions available with this chip.

Math coprocessor hardware interface

The coprocessor operates in an asynchronous mode and can use the same clock generator as the microprocessor. It functions as an I/O device and its presence is indicated if Bit 1 of port 0062h is on.

The coprocessor BUSY signal tells the microprocessor that the coprocessor is operating. The WAIT signal means that it is executing and forces the microprocessor to wait until the coprocessor is done.

The coprocessor operates in the real address mode.

System Support Gate Array

Gate array contents

A system such as the PS/2 Model 30 and compatibles has a system support gate array. The system support gate array contains the following:

- a bus controller,
- a memory controller and parity checker,
- bus conversion logic,
- a wait-state generator, and
- a system clock generator.

Bus controller

The 8086 PS/2 BIOS supports a bus controller that has three bus masters. The bus masters are:

- a microprocessor master
- a coprocessor master, and
- a system support gate array bus master.

The 8086 PS/2 BIOS supports the XT I/O bus, so all XT-compatible add-on boards are supported.

Memory controller

The 8086 PS/2 BIOS supports a memory controller/parity checker function.

Bus conversion logic

The system support gate array also contains bus conversion logic and a wait-state generator.

System clock generator

The system support gate array contains the system clock function. The 8086 PS/2 BIOS expects an Intel 82284 clock generator chip, or equivalent chip, to be available.

I/O support

The 8086 PS/2 BIOS I/O support function includes support of all of the following I/O features and capabilities:

- a serial port
- a parallel port
- a diskette controller
- a fixed disk controller, and
- a VGA video controller.

Each of these capabilities is discussed in the following sections.

Serial port

The 8086 PS/2 BIOS supports either a National Semiconductor 16550 serial port controller, or equivalent logic. The serial port can be addressed as Serial 1 or Serial 2. IRQ 4 is the line for Serial 1 and IRQ 3 for Serial 2. The serial ports are at I/O port addresses 03F8h and 02F8h.

NS16550 characteristics

The NS16550 supports:

- Characters of 5, 6, 7, or 8 bits,
- 1, 1.5, or 2 stop bits, and
- even, odd, or no parity modes.

Note: The NS16550 is functionally compatible with the NS16450 and the NS8250.

Programmable baud rate generator

The serial port controller can operate at speeds of from 110 to 19,200 bps.

continued

Parallel port

The 8086 PS/2 BIOS supports a parallel port that can transfer 8 bits of data at standard TTL levels. The parallel port can be called port 1, 2, or 3, is AT-compatible, and has a bidirectional mode, supporting both input and output. It also supports level sensitive interrupts and a readable interrupt pending status.

Parallel port addresses

The following table lists the parallel port addresses:

Parallel Port Number	Data Address	Status Address	Control Address
1	03BCh	03BDh	03BEh
2	0378h	0379h	037Ah
3	0278h	0279h	027Ah

Diskette controller

The 8086 PS/2 BIOS supports an NEC 765, or equivalent diskette controller. Diskette support from other PS/2 BIOS products may exist only for 720K 3.5 inch diskette drives. Four types of diskette drives are supported by the Phoenix 8086 PS/2 BIOS:

- 720K 3.5 inch,
- 1.44 MB 3.5 inch,
- 360K 5.25 inch, and
- 1.2 MB 5.25 inch drives.

The Phoenix 8086 PS/2 BIOS also will work with a system with no diskette drives, as long as the BIOS options are configured to support the configuration error message suppression option.

continued

Diskette drive configuration

The IBM PS/2 8086 BIOS supports one or two internal diskette drives.

The Phoenix 8086 PS/2 BIOS supports up to two internal diskette drives and one external diskette drive. These drives can be any combination of zero, one or two internal diskette drives (which can be a mix of any supported drive types), and one external drive (must be a 3.5 inch drive).

Fixed disk controller

The 8086 PS/2 BIOS supports zero, one, or two fixed disks. The BIOS assumes that a Western Digital ST506 fixed disk controller, or equivalent, is available.

Fixed disk I/O channels

The 8086 PS/2 BIOS supports a dedicated I/O channel for fixed disk support at I/O address 0320h. The fixed disk adapter status register and control registers are at I/O address 0322h, the fixed disk adapter attention register is at 0324h, and the fixed disk adapter interrupt status register is at 0324h.

The DMA controller is programmed to perform the read/write fixed disk functions. DMA channel three is used for fixed disk data transfers. The system board controls the chip select feature of the fixed disk controller through the Planar Control Register (Port 0065h).

Video controller

The Phoenix 8086 PS/2 BIOS supports the Video Graphics Array (VGA) controller. The IBM 8086 PS/2 BIOS supports the MCGA standard.

---

continued

DAC

The video DAC (digital-to-analog converter chip) contains 256 individual color registers which can be accessed by the BIOS as either four 64-color registers or sixteen 16-color registers.

Each DAC color register contains one 18-bit RGB (red-green-blue) analog value. Six bits of each register are allocated to each primary color. Thus, the color represented in each DAC color register may be any of 256K possible colors (i.e. 2^{3*6} = 256K).

Video RAM

The Video Service requires 256K of read/write video RAM formatted into four banks (or maps) of 64K each.

To maintain compatibility, display memory for each of the historical MDA, CGA, and EGA modes is exactly as it was before. The display memory organization for the new VGA modes is outlined in the INT 10h Video Service Section.

Analog monitor support

To display all modes, the Video Service requires either a monochrome or a color direct drive analog monitor with a 31.5 KHz horizontal scan frequency.

The display's vertical gain is adjusted automatically by the VGA-compatible circuitry. Thus, video modes with 350, 400, and 480 horizontal scan lines can be displayed without requiring manual adjustment.

Multiscan monitor support

In addition to 31.5 KHz direct drive analog monitors, the Video Service also supports multiscan rate monitors capable of operating in analog modes (e.g. NEC Multisync monitor). Monitors of this type require an adapter cable that matches the signal assignments and monitor ID circuitry of the DAC external video controller.

continued

Keyboard controller

The 8086 PS/2 BIOS supports an intelligent keyboard subsystem which is referred to as the keyboard support hardware in this book.

Pointing device interface

The pointing device in the 8086 PS/2 BIOS is controlled by the keyboard support hardware. INT 73h handles interrupts from the pointing device. The INT 15h pointing device services control the device operations.

System Time-Related Logic

Chip requirements

The 8086 PS/2 BIOS expects the following time-handling chips or their equivalents to be available:

- Intel 82284 Clock Generator
- Intel 8254A Programmable Timer
- Real time clock (described later in the Interrupt Controller Extensions discussion).

Intel 82284 Clock Generator

The 8086 PS/2 BIOS expects an Intel 82284 clock generator chip, or equivalent chip, to be available.

8254A Programmable Timer

The 8086 PS/2 BIOS expects an Intel 8254A Programmable Timer chip, or equivalent, to be available. The 8254A Programmable Timer is a counter and timer that provides three channel timers. All channels are driven by a 1.19 MHz oscillator signal.

Timer Counters

There are some differences between the three timer channels:

Counters 0 and 2 are:

- independent 16-bit counters,
- can be preset, and
- can count in BCD (binary coded decimal) or in binary.

Counter 3 is:

- 8 bits only,
- can be preset,
- counts in binary only, and
- can count only downward.

Interrupt Controller Extensions

The 8086 PS/2 BIOS assumes the existence of the interrupt controller extension feature, which controls the:

- keyboard,
- pointing device, and
- real time clock interfaces.

See the INT 15h System Services section for a discussion of the interrupt controller extension.

Keyboard subsystem

The 8086 PS/2 BIOS supports a keyboard subsystem. 101-key keyboards are supported.

Pointing device interface

The pointing device (mouse) is controlled by INT 71h, which determines whether an interrupt is a keyboard, pointing device, or RTC interrupt.

INT 73h handles all interrupts from pointing devices. The INT 15h pointing devices functions control the mouse operation.

Real Time Clock

The 8086 PS/2 BIOS assumes a Real Time Clock function that is controlled by the interaction of I/O ports 00B0h – 00EFh. The ROM BIOS assumes that no CMOS RAM is available in systems based on the 8086.

System Timer

Chip requirements

The 8086 PS/2 BIOS expects an Intel 8253 Programmable Timer chip, or equivalent, to be available. The 8253 Programmable Timer is a counter and timer that works as an arrangement of four 16-bit I/O ports at addresses 0040h-0043h. The 1.19 MHz clock signal is received from the I/O support gate array. Ports 0040h-0042h are counters, and 0043h is a control register for timer mode programming.

Timer Channels

There are some differences between the three timer channels:

Channel 0:

- is a general-purpose timer,
- provides a constant time base for a time-of-day clock, and
- can count in binary coded decimal or in binary.

Channel 1:

- is for internal diagnostics,
- can be preset,
- counts in binary only, and
- can count only downward.

Channel 2:

- used for tone generation,
- outputs to the beeper data of the I/O support gate array, and
- is controlled by Bit 0 of port 0061h.

continued

System timer modes

The system timer has six modes. These modes are described in the following table:

Mode	Name
0	Interrupt on Terminal Count
1	Hardware Retriggerable One-Shot
2	Rate Generator
3	Square Wave
4	Software-Triggered Strobe
5	Hardware Retriggerable Strobe

Common timer mode operations

All modes have the following operations in common:

- All control logic resets when control bytes are written to a counter.
- Counters do not stop when they reach zero.
- In Modes 0, 1, 4, and 5 the counter wraps to the highest possible count, and continues to count.
- In Modes 2 and 3, the counter reloads the initial count and continues to count.

DMA Controller

DMA requirements

The 8086 PS/2 BIOS expects the DMA controller to be compatible with an Intel 8237, or equivalent controller. It is also expected that the DMA controller is capable of 8-bit wide data access between 8-bit I/O adapters and 8-bit system memory. The Direct Memory Access (DMA) controller allows I/O devices to transfer data directly to and from memory, significantly decreasing I/O processing by the microprocessor.

The 8086 PS/2 BIOS supports one 8237 DMA controller, supporting 4 channels (channels 0-3), each capable of 20-bit wide data access, operating at 4 MHz, and uses I/O port addresses 0000h to 001Fh.

Microprocessor access to DMA

The microprocessor can address the DMA controller and access the DMA registers. The microprocessor can control the DMA modes, transfer addresses, transfer counts, channel masks, and page registers.

On a 10 MHz system, the DMA controller requires

- 1 wait state to complete processing of any command from the system,
- at least three clock cycles to complete an I/O read or write transfer, and
- at least two clock cycles for a memory read or write transfer.

The system microprocessor must access the DMA with 8-bit I/O instructions.

DMA data transfer

After the DMA controller is programmed to service a data transfer request, a transfer can take place. Channels 0-3 are 8-bits wide and can transfer up to 64K to/from any address in system memory. DMA transfers take six clock cycles of 250ns each, or 1.5 microseconds. More wait states may be added for slower devices by making the I/O channel inactive.

continued

DMA page registers

The following table shows the DMA page register characteristics:

I/O Port Address	Read/Write Status	Length	Description
0081h	R/W	8 bits	Channel 2
0082h	R/W	8 bits	Channel 3
0083h	R/W	8 bits	Channel 1
0087h	R/W	8 bits	Channel 0

8259 Programmable Interrupt Controller

Description

The Intel 8259 Programmable Interrupt Controller handles all interrupts. One 8259 Programmable Interrupt Controller, or equivalent, is supported by the 8086 PS/2 BIOS. Compatibility with interrupts expected in existing software is maintained in the PS/2-compatible BIOS.

Levels of interrupts

There are eight levels of interrupts. Several devices may share the same hardware interrupt, which reduces the interrupt controller's sensitivity to a transient signal on the I/O bus.

Interrupt uses

The following table shows how interrupts are used in 8086-based PS/2-compatible systems:

Interrupt Level	System Board Availability	I/O Channel Availability	Comments
NMI	Parity check and coprocessor	I/O channel check	Only available on system board
IRQ 0	Timer channel 0	Not available	Called by INT 71h
IRQ 1	Keyboard, pointing device, and real time clock	Not available	
IRQ 2	Video	Available	
IRQ 3	Reserved	Available	
IRQ 4	Serial port	Available	
IRQ 5	Fixed disk	Available	
IRQ 6	Diskette drives	Available	
IRQ 7	Parallel port	Available	

continued

Interrupt Sharing

Multiple adapters may share an interrupt level. Adapters sharing interrupts allow the IRQ line to float high, and each adapter causes an interrupt by pulsing the line low. Adapters sharing interrupts must monitor the IRQ line, and must not issue interrupts until they are rearmed after an interrupt by another adapter.

Interrupt sharing provides a way to:

- link an adapter's interrupt handler to a chain of interrupt handlers,
- share interrupt levels among active adapters, and
- unlink the interrupt handler from the chain when the task is deactivated.

NMI mask

The NMI (nonmaskable interrupt) input to the 8086 is masked off at power-on reset. Bit 7 of I/O port address 00A0h is set to one to enable the NMI. A power-on reset clears this bit.

System control port (0061h)

Port 0061h contains information used for system control.

The following table shows Port 0061h write operations:

Bit	Description
7	Reserved
6	Reserved
5	Enable channel check
4	Enable parity check
3	Reserved
2	Reserved
1	Beeper data
0	Enable timer 2 gate to beeper

System board status port (0062h)

The following table shows Port 0062h read operations:

Bit	Description
7	Parity error occurred
6	I/O channel check active
5	Timer 2 output status
4	Reserved
3	Reserved
2	Disk installed
1	Coprocessor installed
0	Reserved

Glossary

ABIOS

Advanced BIOS. The BIOS designed to support multitasking operating systems such as OS/2. It comes packaged with a traditional Compatibility BIOS (CBIOS).

Adapter Card

A circuit board that can be installed into one of the expansion slots inside a PS/2-compatible computer in order to expand the capabilities of the computer.

Adapter Description Files (ADFs)

Text files supplied on a diskette by manufacturers of PS/2-compatible adapter cards. The ADFs contain information such as what resources are needed to use the card. The ADFs must be copied to a working copy of the Reference Diskette after the adapter card is installed.

Adapter ROM

The read-only memory on the adapter, which contains code to control the adapter device. An adapter is a peripheral card that extends the operation of the system. For example, a fixed disk drive controller is an adapter that may have an adapter ROM. Adapter ROM code may include an ABIOS ROM extension.

continued

Address Bus

One or more lines (conductors) that carry address codes from the microprocessor to all parts of the system.

Arbitration

Arbitration is a process through which devices compete for possession of the channel on a prioritized basis.

Arbitration Level

Arbitration levels are the levels of priority assigned to devices that compete for possession of the channel.

Bimodal Operation

Refers to the ability of a program to operate in both the real address mode and the protected address mode of the 80286, 80386SX, or 80386 microprocessor. ABIOS services are bimodal; they operate in either real or protected mode.

BIOS

Basic Input/Output System. Systems software that interfaces between the operating system and hardware.

Boot

Process of starting the computer.

Burst Mode

Burst mode is a method of DMA transfer that allows a device to remain inactive for long periods and then send large amounts of data in a short time.

Bus

One or more lines (conductors) that carry signals or power.

Byte

Eight contiguous bits; a bit is the smallest item of information that a computer can process.

Cache

Method of using a fast device to speed up access to a slow device.

continued

CBIOS

Compatibility BIOS. The traditional single-tasking portion of a PS/2-compatible BIOS.

CBIOS Service

A software routine that services a given peripheral device, and provides an interface between the operating system and the hardware. These services are single task, call/return functions, as opposed to the device services offered by ABIOS.

CMOS

Acronym for Complementary Metal Oxide Semiconductor. In PS/2 compatibles, it is low-power memory that is battery-backed and is therefore not lost when the computer is turned off.

Configuration

The process of setting up all the parts of the computer so they run effectively.

Cyclic Redundancy Check (CRC)

A method of redundancy check where the check key is produced by a cyclic or repeating algorithm. A common means of error checking.

Default

A value, setting, or option that is assigned by the program or system.

Direct Memory Access (DMA)

Direct Memory Access is a means for I/O devices to transfer data directly to and from system memory without the intervention of the microprocessor. This significantly decreases I/O processing by the microprocessor.

DMA Controller

A DMA controller is a device which gives addresses and control signals to the device that has won the bus through arbitration. The controller does not enter into the arbitration itself.

DMA Device

A DMA device enters into arbitration for the channel. If it wins, it receives addresses and control signals from the DMA controller so it can read or write data.

continued

DOS

Acronym for Disk Operating System and short for PC-DOS and MS-DOS. DOS, like other operating systems, organizes the files and memory for other programs.

Error Handler

An invisible program on the Reference Diskette that reads the POST error log. If an error is found, a cause and solution type message about the error is displayed.

Expanded Memory

For AT-compatible systems, up to 32 MB of additional "paged" memory above the DOS 640K limit. Application programs written according to LIM EMS or AST EEMS specifications can use this type of memory. Examples of such programs are Lotus 1-2-3, Symphony and Framework.

Expanded Memory Specification (EMS)

For AT-compatible systems, a specification and protocol established by a consortium of computer manufacturers, principally Lotus, Intel, and Microsoft (LIM), which establishes a set of rules for organizing and accessing expanded memory.

Extended Expanded Memory Specification (EEMS)

A specification and protocol established by a consortium of computer manufactures, principally AST, Quadram, and Ashton-Tate, which establishes a set of rules for organizing and accessing expanded memory.

Extended Memory

The memory above 1 MB. XENIX and IBM's VDISK can use this memory, but DOS and almost all application programs cannot, since use of the protected mode of the Intel 80386 or 80386 microprocessor is required.

Fixed Disk (Hard Disk)

A magnetic storage device consisting of a drive mechanism with permanently installed metallic disks; a "filing cabinet" for the computer.

Hardware

The physical equipment and components in the computer system.

continued

Interrupt

The suspending of microprocessor program execution by a demand for attention coming from a peripheral device. After the interrupt has been serviced, the suspended microprocessor task can be resumed where it was broken off by the interrupt.

Kilobytes (K)

1024 bytes.

Known State

When a device is initialized or reset, and then set to a particular pre-established condition, it is said to be in a known state.

Low Level Format

Electronic equivalent of drawing a detailed street map on the fixed disk. The electronic markings tell the system at what points to start and end reads and writes.

Main Memory

The memory between 0 and 1 MB.

Megabyte (MB)

One million, or 1024 x 1024 bytes.

Memory

A device that can store data recorded in it and from which the data can be retrieved.

Micro Channel

Information is exchanged between the computer system board and the adapter cards which are plugged into it by means of the "bus." Micro Channel refers to the particular bus design in a PS/2-compatible computer. It is also referred to as MCA, for Micro Channel Architecture.

Microprocessor

Central processing unit, or "brain" of the computer.

continued

Multitasking

Multitasking programs execute multiple program modules simultaneously. Information input into one module does not need to be processed completely before information can be input into another module. ABIOS supports multiple concurrent requests.

Offset

A method of addressing that defines an address as relative to the beginning of a memory segment.

Operating System

Generic systems software which controls the execution of applications software.

Option Diskette

The diskette provided by manufacturers of adapter cards that contains adapter description files (ADFs), which are written to the Reference Diskette and used by it to configure the system.

Parameter

Value, option, or setting that can be set in two or more ways.

Power-on Self Test (POST)

A program that tests all parts of the computer every time you turn on the computer.

Program

A set of instructions defining the operations of a computer in order to achieve the desired results.

Programmable Option Select (POS)

A way of setting up peripheral devices on a PS/2 machine via the power-on self test (POST), in which values are placed in registers on the devices. This software set up routine replaces the traditional switches and jumpers on devices.

continued

Protected Virtual Address Mode (Protected Mode)

One of two 80286 or three 80386 memory addressing modes. In protected virtual address mode, the 80386/80286 uses all address lines. This allows addressing of up to 16 megabytes of physical memory in an 80286 and 4 GB in an 80386. The 80286 processor's internal memory management allows addressing of an additional 1 gigabyte of virtual memory in protected mode (the 80386 up to 64 Terabytes). Protected mode addresses are specified in selector:offset format. ABIOS data structures support protected mode access of all ABIOS services.

PS/2-Compatible Computer

Any computer that can run software programs written for an IBM PS/2 computer. IBM PS/2 systems come in two basic varieties: Models 25 and 30 versus Models 50, 60, 70, and 80. The former systems do not implement the Micro Channel Architecture (MCA).

Real Address Mode

One of two 80286 memory addressing modes (the 80386 has three). In real address mode, the 80286 and 80386 microprocessors use 20 address lines, thus allowing memory addressing of up to 1 megabyte of physical memory (2^{20}). Real address mode does not support virtual memory addressing. Real mode addresses are specified in segment:offset format. ABIOS data structures support real mode access of all ABIOS services.

Reference Diskette

In PS/2-compatible MCA-based systems, POST error recovery, access to system utilities, and system configuration are all controlled by a utility diskette, the Reference Diskette. The Reference Diskette can automatically configure a system, resolving conflicts between adapter card settings, and optimally configuring the system.

Segment

A unit of contiguous, one-dimensional address space. In real mode, these address blocks are 64K in size, referenced by one byte. In protected mode, programs can allocate segments of any size they require up to 64K.

continued

Selector

A value contained in a segment register (such as the CS, DS, SS, or ES segment registers) when in protected mode. This value determines what segment is currently being used; e.g., with CS, what segment is being used for executing code.

Single-Tasking

Single-tasking operating systems can only execute one program module or routine at a time. Information input into a module or routine must be processed completely before information can be input into another module.

Software

A comprehensive term used to identify all of the nonhardware components of a computer. Software includes computer programs and data.

System Board

A large circuit board that holds most of the main electronic parts of the computer.

System Board ROM

Read-only memory chips that reside on the system board and provide control information for various system components.

Task

In an 80386, a task is the execution of a single process or set of instructions to perform a particular function. It is not the same as an operating system task.

Time-Out

When the interval of time expected for a certain process (an interrupt) to occur is exceeded.

Virtual 8086 Mode (80386 only)

A way of emulating the 8086 or 8088 microprocessors on the 80386. The 8086 program runs in protected mode as a task that can run with multiple 8086 virtual tasks, as well as alongside other multiprogrammed 80386 tasks.

Additional Resources

The following books provide additional material related to the CBIOS:

Dettman, Terry R. *DOS Programmer's Reference.* Carmel, IN: Que® Corporation, 1988.

Duncan, Ray. *Advanced MS-DOS Programming.* Redmond, WA: Microsoft Press, 1986.

Duncan, Ray. *IBM® ROM BIOS.* Redmond, WA: Microsoft Press, 1988.

International Business Machines Corporation. *IBM® Personal System/2™ and Personal Computer BIOS Interface Technical Reference*. Boca Raton, FL: IBM, 1987.

International Business Machines Corporation. *IBM® Personal System/2™ Model 30 Technical Reference*. Boca Raton, FL: IBM, 1987.

International Business Machines Corporation. *IBM Personal System/2™ Model 50 and 60 Technical Reference*. Boca Raton, FL: IBM, 1987.

Norton, Peter and Wilton, Richard. *The New Peter Norton Programmer's Guide to the IBM® PC and PS/2®*. Redmond, WA: Microsoft Press, 1988.

Wadlow, Thomas A. *Memory Resident Programming on the IBM PC.* Reading, MA: Addison-Wesley Publishing Co., Inc., 1987.

Wilton, Richard. *Programmer's Guide to PC® and PS/2™ Video Systems.* Redmond, WA: Microsoft Press, 1987.

Index

A

B, cont'd

C

D

E, cont'd

F

J

K

N

N, cont'd

O

P

R

S

U

V

W

W, cont'd

Y

The Phoenix Technical Reference Series

Phoenix Technologies Ltd.

CBIOS for IBM® PS/2® Computers and Compatibles is an indispensable guide to the CBIOS, the part of the PS/2 BIOS that supports single-tasking operating systems such as DOS. It is written for anyone who needs a detailed understanding of the system software resident in their IBM PS/2 or compatible computer—applications programmers, operating system developers, hardware manufacturers, and students of PC architecture. This reference book is the most complete source of information available on ROM-based system software for DOS.

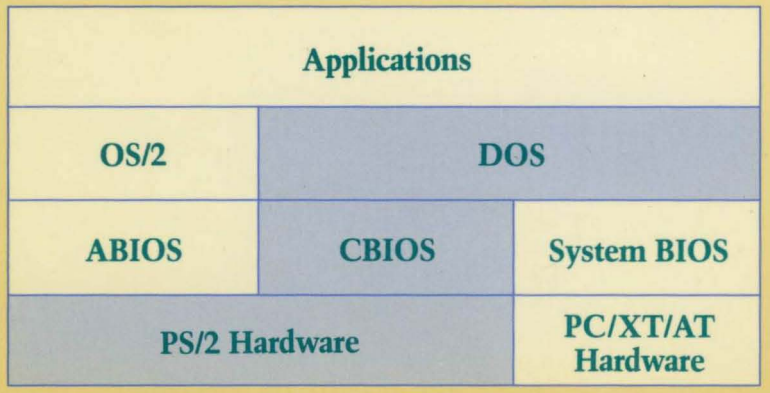
Key information includes:

- complete bit-level descriptions of I/O port addresses, CMOS RAM data areas, ROM BIOS and Extended ROM BIOS data areas, and all system configuration tables
- descriptions of undocumented IBM CBIOS features
- programming tips for using BIOS functions more productively
- complete documentation of the power-on self test (POST) routine
- complete beep code and error message documentation
- descriptions of key hardware commands, including DMA, POS, and the Micro Channel.™

The information in this book is applicable to all IBM PS/2 and compatible computers, ranging from IBM PS/2 Models 25 through Model 80.

Phoenix Technologies Ltd. of Norwood, Massachusetts is the industry leader in providing ROM BIOS products for IBM PC and PS/2 compatibles.

The Phoenix Technical Reference Series includes two other technical reference volumes on BIOS software—*ABIOS for IBM PS/2 Computers and Compatibles: The Complete Guide to ROM-Based System Software for OS/2* and *System BIOS for IBM PC/XT/AT Computers and Compatibles: The Complete Guide to ROM-Based System Software.*



Addison-Wesley Publishing Company, Inc.

52695

9 780201 518047

ISBN 0-201-51804-X