Phoenix Technical Reference Series

System BIOS for IBM® PC/XT™/AT® Computers and Compatibles

The Complete Guide to ROM-Based System Software

Phoenix Technologies Ltd.

Phoenix Technical Reference Series

System BIOS for IBM® PC/XT™/AT® Computers and Compatibles

The Complete Guide to ROM-Based System Software

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

System BIOS for IBM PC/XT/AT computers and compatibles / Phoenix Technologies, Ltd.
p. cm. -- (Phoenix technical reference series)
ISBN 0-201-51806-6 :
1. IBM Personal Computer. 2. IBM Personal Computer XT. 3. IBM Personal Computer AT. 4. Operating systems (Computers) I. Phoenix Technologies. II. Title: BIOS. III. Series.
QA76.8.I2594S97 1989
004.4'469--dc20 89-17557
CIP

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

ISBN: 0-201-51806-6
EFGHIJ-AL-93210
Fifth Printing, August 1990

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.

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.

Hercules, Hercules Graphic Card, and Hercules Graphic Adapter are trademarks of Hercules Computer Technology.

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, XENIX, and Microsoft are registered trademarks of Microsoft Corporation.

NEC and Multisync are registered trademarks of Nippon Electric Corporation.

Quadram is a registered trademark of Quadram Corporation.

Dedication

To the IBM engineers and programmers who designed and coded the IBM PC, XT, and AT BIOSs.

Table of Contents

continued

Chapter 3 — CMOS RAM Data

Chapter 4 — ROM BIOS Data

Chapter 5 — I/O Port Addresses

Chapter 6 — Power-On Self Tests (POST)

continued

Chapter 7 — INT 02h Nonmaskable Interrupt

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

Chapter 9 — INT 10h Video Service

continued

Chapter 9 — INT 10h Video Service, Continued

Chapter 10 — INT 13h Diskette Service

continued

Chapter 10 — INT 13h Diskette Service, Continued

Chapter 11 — INT 13h Fixed Disk Service

continued

Chapter 11 — INT 13h Fixed Disk Service, Continued

Chapter 12 — INT 14h Serial Communications Service

Chapter 13 — INT 15h System Services

continued

Chapter 13 — INT 15h System Services, Continued

Chapter 14 — INT 17h Parallel Printer Service

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

continued

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

Chapter 16 — Single Function BIOS Services

Appendix A — AT BIOS Error Codes and Messages

continued

Appendix B — XT BIOS Error Messages

Foreword

The Micro Revolution

The personal computer has come a long way from being a hacker's hobby in basements and garages to a multibillion-dollar industry. In the early days, programming carried with it an aura of history-in-the-making, of a revolution that would alter not only the business world, but the fabric of society as well. In the personal computer, we had found a new vehicle for the exchange of ideas and the formation of a global village. Personal computing promised to democratize technology.

At that time, the structure of computing environments centered around the mainframe. Mainframes cost millions of dollars, filled up large air-conditioned rooms and required a host of programmers to appease and maintain them. Users who wished to supplicate the mainframe for a specific task were forced to request an audience through their Management Information Systems group. They then had to wait months and even years for their applications to be built and run, often sacrificing precious time and frustrating their creative spirits.

I was an undergraduate at M.I.T. when I first heard of the microprocessor. It was the key to the computer kingdom that all hackers dreamt about, a way to bring computing power to individuals without the intervention of bureaucracies. The microprocessor created a cult following among programmers.

In the mid-seventies, large computer companies, built around the proprietary technology of their mainframes and minicomputers, downplayed the micro's potential. As often happens in the business world, it took a group of visionaries and entrepreneurs to capitalize on that potential and to create an industry around it.

Once the market had been formed, though, the computer establishment realized there was gold in those silicon hills and valleys and rushed to enter. It was actually just what the fledgling industry needed: someone with the marketing clout to establish a standard platform. One cannot advance technology without a platform to serve as a basis for expansion and evolution into higher forms.

Thus, the IBM PC was born, and with it the era of DOS.

BIOS: The Missing Link

I founded Phoenix in 1979 as a supplier of programmers' tools to support the DOS standard. We were successful, but not a phenomenon until we broke through with the PC ROM BIOS in May 1984. It proved to be the missing link for manufacturers to build systems that were 100 percent compatible with the IBM PC and, just as important, 100 percent legal.

Our "clean room" methodology was responsible. "Contaminated" engineers studied the BIOS' functionality and passed on those functions to "virgin" engineers, who had never seen the BIOS and who then wrote the code that would accomplish specific tasks. I am proud to say the Phoenix ROM BIOS was one of the keys that unlocked the PC market and enabled it to grow to its present proportions.

Phoenix was your archetypal start-up. The first bus architecture that Phoenix supported was the VW platform as it was driven from one programmer's garage to another's basement, picking up products and orders for our initial line of programmers' tools. Our corporate meeting room was in a basement. I would lead visiting computer executives down a narrow stairwell to a dark basement, where they would share an old sofa with a bunch of cats, shedding hairs all over their pinstriped suits. That was three office buildings ago, and we're still growing.

Over 300 companies have licensed our ROM BIOS. Today, there are more installed Phoenix BIOSs than IBM's, with over 14 million systems booting up with the Phoenix name as the first item to appear on screen.

Our engineering ventures did not end with the BIOS. Since the PC standard is always changing, compatibility is a moving target. In fact, the BIOS has been an evolving piece of indispensable code which has been continually upgraded to maintain a system's compatibility with the advancements of the PC's capabilities.

Phoenix also helped to pioneer emulation technologies that enable technical and graphical workstations to become DOS compatible. Our software co-processor technology is one of the key reasons UNIX workstations are currently enjoying their acceptance in mainstream computing environments. Our latest product development, PhoenixPage, is a systems software suite that supports printer standards, such as PostScript-language compatibility.

Portrait of a Company as a Young BIOS

Phoenix is in the middle of the micro madness, but with a method. Phoenix's strategy is based on the revolution at the heart of the computer industry's dynamic growth. The emergence of open standards has drastically altered the way we do business with advanced technologies. Contrary to the conventional wisdom that looked for a market edge through the development of proprietary technologies, companies are realizing that industry-wide adoption of standards is a prerequisite for growth, both in technical and financial terms.

The diversification of the Phoenix product line revolves around a central concept: the promulgation of open standards. In the current proliferation of operating systems, CPUs, chip sets, system board configurations, and bus architectures, Phoenix's flexible systems software gives a manufacturer freedom of design and end users the assurance of quality compatibility.

The BIOS itself may serve as a perfect business model in understanding Phoenix's role in the microcomputer revolution. Just like systems-level components in the architecture of a computer, there are players in the computer industry that are mainly invisible to end users, but who possess roles that are crucial to the smooth execution of product cycles and advancing technologies.

Just as the BIOS sits at the heart of the PC, where software meets hardware, and interfaces communications between the operating system, CPU, and a computer's peripheral devices (monitor, printer, disk drives, etc.), so Phoenix is positioned at the center of the PC universe, working closely with semiconductor, hardware, and peripheral manufacturers; operating system and independent software developers; and system integrators, ensuring compatible designs among them all, peace of mind for purchasers and end users, and a stable and open platform for programmers.

PC Compatibility: The Only Game in Town

The PC standard is now entirely software-driven. It is not dependent on IBM's hardware architecture. The goal of a standard is not IBM hardware emulation, but a hardware-independent compatibility based on the ability to run PC applications. It was as much to achieve compatibility with each other as with IBM that caused most of the industry's largest manufacturers (some of whom once shunned PC compatibility) to jump on the bandwagon.

The value of compatibility with the PC standard from the end user's viewpoint is that it offers investment protection, portability, and access to the world's largest library of applications software packages. These considerations have made 100 percent compatibility the main customer-driven demand in the personal computer marketplace.

In the past few years, the PC industry has seen phenomenal growth. Now that the standard has been established, the platform is in place to launch an even more impressive and expansive future. The goal for software developers is to break away from hardware-imposed stagnation and push the PC environment to new heights. Phoenix's biggest challenge as a company will be to help manufacturers and software developers integrate compatibility into new products with smart, innovative engineering.

Open Sesame

Standards advance technology. The establishment of a common platform enables hardware and software developers to evolve more advanced technologies, since they now have a target and a customer base to make it worthwhile. The adoption of standards frees engineering resources to concentrate on differentiating products with expanded capabilities.

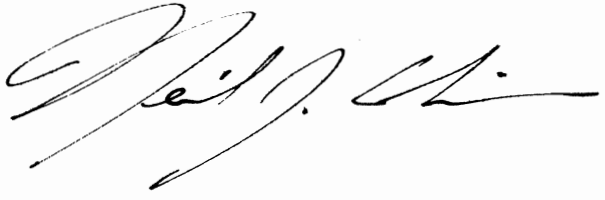
Phoenix's goal has always been to support standards in the micro environment. Open standards enable an industry to reap the rewards of advanced technology by attracting the widest possible customer base and applications. Phoenix provides a cost-effective engineering resource to solve the integration challenges the PC industry faces.

We are offering this series of BIOS technical references in the same spirit of "openness" that has characterized our research and development, from the days of our programmers' tools through the introduction of our PC BIOS and creation of the compatibles market.

Open systems need open communications. For personal computing to succeed in society, information must be readily available to allow individuals to exploit its capabilities. These manuals provide a complete and clear picture of the design of the PC standard. They describe the parameters of compatibility and how best to design a system based on the standard.

Programmers will find a treasure trove between these covers. They will have easy access to the PC architecture and will have a secure and stable platform to which to write. My hope is that these manuals will help put the personal back into computing.

Neil J. Colvin



Chairman
Phoenix Technologies Ltd.

About This Book

What this book is about

System BIOS for IBM PC/XT/AT Computers and Compatibles is a detailed technical reference that describes the BIOS, the component of PC, XT, and AT systems that insulates both applications software and single-tasking operating systems, such as MS-DOS, from direct manipulation of PC hardware. The information provided in this book is applicable to all PC, XT, and AT computers and compatibles.

Who should read this book

System BIOS for IBM PC/XT/AT Computers and Compatibles provides the most comprehensive source of information about IBM PC, XT, and AT and compatible BIOSs available today. It can be used by anyone interested in learning more about IBM or compatible computers.

Applications programmers and device driver developers will find a thorough discussion of each BIOS device service and function, including all levels of video BIOS, from MDA and CGA through EGA and VGA. In addition, *System BIOS for IBM PC/XT/AT Computers and Compatibles* defines, to the bit level, all IBM and compatible I/O port addresses and BIOS-related CMOS RAM and system RAM data.

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

Other volumes in this series

System BIOS for IBM PC/XT/AT Computers and Compatibles is one of several volumes about BIOS software in the Phoenix Technical Reference Series. 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.

CBIOS for IBM PS/2 Computers and Compatibles — a complete technical reference describing the portion of a PS/2 BIOS that supports single-tasking operating systems such as DOS.

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

How to find information

System BIOS for IBM PC/XT/AT Computers and Compatibles is organized into several major parts. Chapter 1 provides a general summary of BIOS concepts. Chapters 2–6 describe BIOS and system data definitions and I/O port usage. Chapters 7–16 describe the individual BIOS services and serve as a technical reference. Appendixes A and B provide comprehensive lists of potential error messages for AT and XT-compatible systems.

Each service-specific chapter (7–16) is organized in the same way. There is a description of the service and the theory of operations for this service, a list of all I/O ports and data definitions accessed by the 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.

A word about diversity

In the eight years since the original PC emerged, there have been countless manufacturers of PC, XT, and AT systems and BIOS products. Although all BIOSs in IBM-compatible microcomputers by definition have to be compatible with the IBM standard, there has been a good deal of diversity in the detail of BIOS architecture.

This diversity makes it difficult to precisely identify terms such as PC-, XT-, or AT-compatible. BIOS functions that appear only in IBM AT systems can be found in the Phoenix XT BIOS. And there are many different combinations of BIOS functions in various XT or AT clones. Whenever possible, we provide directions on how to find out if a particular function is supported.

What we assume you know

This book assumes a basic knowledge of 8086, 80286, and 80386 assembly language programming concepts, PC architecture, and operating system concepts. If you are new to these subjects, you may want to refer to the introductory texts listed at the end of this book.

Acknowledgments

First and foremost, we would like to acknowledge the vision and technical skill of Neil Colvin, founder, Chairman of the Board and Chief Technical Officer of Phoenix Technologies Ltd. We also acknowledge the expert guidance of Lance Hansche, President and CEO of Phoenix Technologies Ltd.

We especially want to acknowledge the contributions made by Jeff Bobzin, Dave Hirschman, and Bob Vandette — the Phoenix engineers who initially brought the BIOS to life, as well as the continuing efforts of Eric Enge, Paula Bishop, Jeff Kleiman, Debbie Schultz, Stan Lyness, Greg Honsa, Bruce Cairns, Ruth Brent, Jeff Wheeler, and the rest of the Phoenix staff, who keep the BIOS alive and compatible with the widest range of PC hardware.

This volume could not have been produced without the tireless efforts of Rich Levandov, Vice President of Marketing, and Henry Suwinsky, Project Manager.

We would like to thank Kathy Schiff, Manager of Technical Communications, who provided editorial guidance and direction in the production of this book and Jeff Evangelista, who contributed two well-written chapters. And last but not least, our special thanks goes to Sandie Zierak, Chief Production Coordinator, for her invaluable contributions in document design, graphics, and production.

The Authors

Mike Boston
Paul Narushoff

Phoenix Technologies Ltd.
Norwood, MA
May, 1989

Acronyms and Abbreviations

The following abbreviations and acronyms are used in this manual:

ACK	Acknowledge (keyboard and serial port commands)
ACR	Fixed disk adapter controller register
ALE	Address latch enable
ANSI	American National Standards organization
ARB	Access rights byte (part of 80286/80386 instruction)
ASCII	American Standard Code for Information Interchange
ASIC	Application-specific integrated circuit
ASR	Fixed disk adapter status register
AT	Advanced technology
b	Binary
BAT	Basic assurance test (keyboard diagnostic)
BCD	Binary coded decimal
BIOS	Basic input/output system
bps	Bits per second
CAS	Column address strobe
CCB	Command control block
CGA	Color graphics adapter
CLK	Clock line on a microprocessor
CMOS	Complementary metal oxide semiconductor
CRC	Cyclic redundancy check
CSB	Command specify block
CTS	Clear to send
DAC	Digital-to-analog converter
DCC	Display combination code
DIN	Deutsche Industrie Norm
DIP	Dual inline package
DMA	Direct memory access
DPL	Descriptor privilege level — part of 80286/80386 code
DSR	Data set ready
DSR	Device service routine
DTR	Data terminal ready
EA	Effective address

continued

EBCDIC	Extended Binary Coded Decimal Interchange
ECC	Error checking and correction
EGA	Enhanced graphics adapter
EIA	Electronic Industries Association
EISA	Extended industry standard architecture
EOI	End of interrupt
EPL	Extended privilege level
EPROM	Erasable programmable read-only memory
ESDI	Enhanced small device interface
ETB	End of transmission block
FCB	Format control block — disk I/O data structure
GDT	Global descriptor table
h	Hexadecimal
ICW	Interrupt control word
INT	Interrupt
I/O	Input/Output
IRQ	Interrupt request line or Interrupt request
IRET	Return from an interrupt
ISA	Industry standard architecture
ISR	Interrupt service routine
ISR	Interrupt status register
K	Kilobytes
Kbs	Kilobits per second
LDT	Local descriptor table
LDTR	Local descriptor table register
LED	Light-emitting diode
LID	Logical ID
LSB	Least significant byte (or bit)
LSI	Large scale integration
M, MB	Megabytes
MDA	Monochrome display adapter
MFM	Modified frequency modulation
MHz	Megahertz

continued

MMU	Memory management unit
MSB	Most significant byte (or bit)
NCB	Network control block
NMI	Nonmaskable interrupt
OCW	Operation control word
OS/2	Operating System/2
PC	Personal computer
PCLK	Peripheral clock
PEL	Picture element (pixel)
PGA	Professional graphics adapter
PIC	Programmable interrupt controller
PIO	Programmed input/output
PIT	Programmable interrupt timer
POR	Power-on reset
POST	Power-on self test
PTL	Phoenix Technologies Ltd.
RAM	Random access memory
RAS	Row address strobe
RGB	Red-Green-Blue (type of color generator)
RI	Ring indicator
RLL	Run length limited
ROM	Read-only memory
RS-232C	Recommended standard-232C (serial controller interface)
RTC	Real time clock
RTS	Request to send
SCSI	Small computer systems interface
SDLC	Synchronous data link communications
SIMM	Single inline memory module
SIP	Single inline package
SMD	Surface-mount device
SSB	Sense summary block — a disk I/O structure
TSS	Task state segment
TTL	Transistor-to-transistor logic

continued

UART	Universal asynchronous receiver/transmitter
VGA	Video graphics array
VLSI	Very large scale integration
XT	Extended technology

1

Chapter 1
The ROM BIOS

Overview

What is a BIOS?

A basic input output system (BIOS) is the lowest level software component of a computer's operating system. In personal computer systems, the BIOS usually consists of a routine designed to test system hardware and bootstrap load the operating system and a group of device drivers that service system I/O components, such as the keyboard, the display, fixed disk drive, diskette drives, real time clock, parallel ports, serial ports, and other components.

Two methods for implementing BIOS

There are two methods for implementing a BIOS. The BIOS can be incorporated directly into system software and delivered on diskette as an integral part of the operating system. Or, the BIOS can be tied directly to the system hardware, programmed into ROM on the motherboard or on an expansion card.

continued

Advantage of the ROM BIOS

As long as the interface between the operating system and the BIOS is standardized, programming the BIOS into ROM has the advantage of allowing system hardware to evolve independently from the operating system. When the BIOS is tied directly to the operating system, however, the operating system must be modified each time there is a change in system hardware.

PC ROM BIOS

The original IBM PC, a system based on the the 8/16-bit Intel 8088 microprocessor, was introduced in 1981. The IBM PC/XT, a fixed disk capable version of the PC, was introduced in 1983. The IBM PC/AT, a 16-bit system based on the Intel 80286 microprocessor, was introduced in 1984. Clone manufacturers have added Intel 80386-based AT-compatible systems. The BIOSs for all these systems are compatible, in the sense that the later AT BIOSs support all PC and XT BIOS functions in some way.

The BIOS in all models and versions of the IBM PC/XT/AT family of personal computers is programmed into ROM contained on the motherboard. The video BIOSs supporting the EGA and VGA PC video adapters are an exception to this rule. The EGA and VGA BIOSs are programmed into ROM contained on EGA and VGA adapter cards.

DOS: the PC operating system

PC-DOS, or MS-DOS — its Microsoft Corporation counterpart, is a single-tasking operating system capable of addressing the 1 megabyte real mode address range of IBM PC/XT/AT systems. Introduced along with the original IBM PC in 1981, PC-DOS/MS-DOS has emerged as the predominant PC operating system. Both PC- and MS-DOS interface with the ROM BIOS instead of directly with system hardware. Programs that interface with DOS services or directly with the BIOS are assured the highest degree of hardware independence.

continued

OS/2 and the ROM BIOS

In 1987, IBM introduced OS/2, a multitasking operating system designed for 16-bit PCs. When implemented on IBM PS/2 systems, OS/2 interfaces with the ABIOS portion of the PS/2 ROM BIOS. Although OS/2 can operate on AT class machines, it does not interface with the AT ROM BIOS. There is no BIOS support for OS/2 on an AT, so the level of support usually provided by the BIOS must be built into OS/2. Since OS/2 on an AT must interface directly with the computer hardware, there is a minimal level of portability from system to system with the AT version of OS/2.

Reference: For more information on the ABIOS, see *ABIOS for IBM PS/2 Computers and Compatibles* in this series.

Terminology used in this book

With the introduction of IBM PC/XT/AT compatible hardware and of IBM PC/XT/AT compatible ROM BIOSs (such as the Phoenix Technologies Ltd. 80x86 ROM BIOSs) came an entire PC "clone" industry. Like their IBM counterparts, the clones also contained a system board ROM BIOS.

Because of their sheer number, keeping track of PC clone model names can be confusing. In this book, "PC" refers to the IBM Personal Computer and compatible systems; "PC/XT" refers to the IBM Personal Computer/XT and compatibles, and "PC/AT" refers to the IBM Personal Computer/AT and compatibles. The terms PC BIOS, XT BIOS, and AT BIOS correspond to the ROM BIOS contained in each of those systems and their compatibles.

What this chapter is about

This chapter presents a general introduction to the concepts and design features of the IBM PC/XT/AT and compatible ROM BIOS.

The following topics are discussed:

- BIOS: Theory of Operation
- BIOS Memory Usage
- Interrupt Vector Table
- Unexpected Interrupt Handler
- System BIOS Functions and BIOS Diversity
- Summary: The BIOS Services

BIOS: Theory of Operation

Background

All IBM PC/XT/AT and compatible computer systems are based on the Intel 80x86 or compatible microprocessors. To understand the BIOS architecture, one must consider the interrupt-driven nature of the Intel 80x86 architecture.

In Intel 80x86-based systems, 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.

Where the ROM BIOS fits in

The majority of the ROM BIOS I/O drivers provide low level device services to the operating system and application programs running on the system. This kind of BIOS I/O driver is invoked exclusively by software interrupt.

The remaining few ROM BIOS I/O services provide direct support for system hardware; they are called via a hardware Interrupt.

BIOS services that are driven by software interrupt are referred to as ROM BIOS device service routines (DSR), or more commonly by the name of the device involved. The ROM BIOS Video DSR, for example, is commonly referred to as the ROM BIOS Video Service. BIOS hardware driven services are referred to as hardware interrupt service routines (ISR).

Understanding the ROM BIOS is largely a matter of understanding how the BIOS device services fit into the Intel 80x86 scheme of interrupts.

continued

Facts about interrupts

Be it microprocessor, hardware, or software, every 80x86 interrupt is assigned a unique interrupt number ranging from 00h to FFh. By convention and to maintain IBM compatibility, certain ranges of interrupt numbers are reserved for special use. The software interrupt numbers ranging from 20h - 3Fh, for example, are reserved for the operating system (usually PC- or MS-DOS). End user software interrupts fall in the ranges 60h - 67h and F1h - FFh.

Each interrupt number is associated with one interrupt vector. An interrupt vector is the double word, segment:offset address of the routine assigned to an 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, and so forth.

In a process called the power-on self test (POST), the portion of the ROM BIOS responsible for testing system components and booting the operating system writes the interrupt vector table to low memory and initializes the vector address of all those interrupts concerned with the BIOS. When the operating system is booted, it, in turn, initializes all those interrupt vectors concerned with DOS. Users calling the BIOS from applications programs must take care to initialize the vectors concerned with their own interrupts.

continued

Interrupt ranges, types, and uses

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 BIOS.
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. ▪ BIOS 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.

Reference: The contents of the interrupt vector table, as initialized by the ROM BIOS, are listed under the heading Interrupt Vector Table in this chapter. A discussion of how the ROM BIOS handles unexpected interrupts is found under the heading Unexpected Interrupt Handler.

continued

Calling ROM BIOS services/functions

Each ROM BIOS DSR is associated with one entry in the interrupt vector table. The ROM BIOS Video Service is vectored to INT 10h, the ROM BIOS Parallel Printer Service is vectored to INT 17h, and so on.

Most of the BIOS DSRs are capable of executing more than one device related routine, or function. Individual functions within a device service are identified by a hex number and are selected by specifying that number in the AH register when the service is invoked. (Some ROM BIOS DSRs also contain subfunctions. Subfunctions are usually selected via the AL or BL registers.)

In addition to function numbers, all other parameters are passed to and from BIOS functions via the microprocessor registers. All BIOS device services 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.

Reference: A complete list of the ROM BIOS DSRs, their associated interrupts, and functions is found under the heading Summary: The BIOS Services in this chapter.

Example program

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

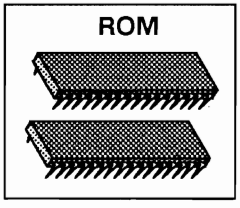
```
MOV AH, 2          ;Select "Set Cursor Position" function
MOV DH, 3          ;Input row parameter into DH register
MOV DL, 14         ;Input column parameter into DL register
INT 10H            ;Invoke INT 10h, BIOS Video Service
```

BIOS Memory Usage

Description

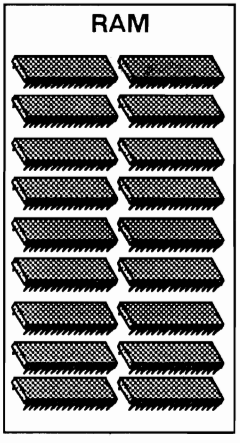
Aside from the interrupt vector table, a BIOS device service makes use of other information stored in System RAM, CMOS RAM, and the BIOS itself. The illustration below outlines the location and use of BIOS ROM, System RAM, and CMOS RAM. XT systems usually do not have any CMOS RAM, however, XT systems use ROM BIOS and System RAM data.

Reference: Additional information on System RAM data, CMOS RAM data, and ROM BIOS data, is found in Chapters 2, 3, and 4, respectively.



ROM BIOS DATA

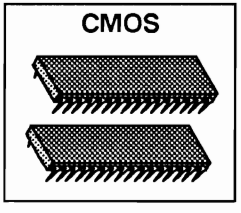
The BIOS contains routines that test and initialize system hardware and boot the operating system. It also contains routines that service the microprocessor, the fixed disk, keyboard, display, and other peripheral devices with which the microprocessor must communicate. Each of the routines is stored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



SYSTEM RAM DATA

The BIOS uses system RAM to store data that it needs to reference or update in the course of executing its device service routines. The BIOS 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 an interrupt vector table in low memory beginning at 00:00h and ending at 00:3FFh. The BIOS 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 BIOS stores a set of data definitions in system RAM in absolute memory location 400h-500h. In the process of executing device service routines, the BIOS refers to and updates this data. For example, the BIOS updates 40:50h, the location of the cursor on the video page, each time the BIOS routine "Set Cursor Position" is executed.



CMOS RAM DATA

The BIOS uses nonvolatile CMOS RAM to store real time clock, system configuration, and system diagnostic information. XT systems usually do not have any CMOS RAM.

- *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 BIOS make use of this information.

Interrupt Vector Table

Description

The table below identifies each interrupt by function and type. Where applicable, it lists the interrupt vector address initialized by the BIOS at POST. System software may revector an interrupt at or shortly after the boot process is completed, so these values may not be the same in every system on every occasion.

Note: The column labeled Ref. Chapter indicates the chapter to turn to 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
05h	Bounds Exception (80286, 80386)	Hardware		
06h	Invalid Op Code (80286, 80386)	Hardware		
06h	Reserved (PC only)	Hardware		
07h	Reserved (PC only)	Hardware		
07h	Math Coprocessor Not Present	Hardware		
08h	Double Exception Error (80286, 80386) (AT only)	Hardware		
08h	System Timer (IRQ 0)	Hardware	FFEA5h	15
09h	Keyboard	Hardware	FE987h	8
09h	Math Coprocessor Segment Overrun (80286, 80386) (AT only)	Hardware		
0Ah	IRQ 2 Cascade from Second Programmable Interrupt Controller	Hardware		
0Ah	Invalid Task Segment State (80286, 80386) (AT only)	Hardware		
0Ah	IRQ 2 (Reserved) (PC only)	Hardware		

continued

Interrupt vector table, cont'd

INT	Function	Type	Vector	Ref. Chap.
0Bh	Serial Communications (COM2)	Hardware		
0Bh	Segment Not Present (80286, 80386)	Hardware		
0Ch	Serial Communications (COM1)	Hardware		
0Ch	Stack Segment Overflow (80286, 80386)	Hardware		
0Dh	Parallel Printer (LPT2) (AT only)	Software		14
0Dh	IRQ 5 Fixed Disk (XT only)	Software		11
0Dh	General Protection Fault (80286, 80386)	Software		
0Eh	IRQ 6 Diskette	Software	FEF57h	10
0Eh	Page Fault (80386 only)	Software		
0Fh	Parallel Printer (LPT1) IRQ 7	Software		14
10h	Video	Software	FF065h	9
10h	Numeric Coprocessor Fault (80286, 80386)	Software		
11h	Equipment List	Software	FF84Dh	16
12h	Memory Size	Software	FF841h	16
13h	Fixed Disk/Diskette	Software	FE3FEh	10/11
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 (PC only)	Software	F1C90h	16
18h	Process Boot Failure (XT, AT)	Software	F1C90h	16
19h	Bootstrap Loader	Software	FE6F2h	16
1Ah	Time-of-Day	Software	FFE6Eh	15
1Bh	Keyboard Break	Software	FFF53h	8
1Ch	User Timer Tick	User	FFF53h	15
1Dh	Video Parameter Table	BIOS Table	FF0A4h	9
1Eh	Diskette Parameter Table	BIOS Table	FEFC7h	10

continued

Interrupt vector table, cont'd

INT	Function	Type	Vector	Ref. Chap.
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-59h	Reserved			
5Ah	Cluster Adapter			
5Bh-5Fh	Reserved			
60h-66h	Reserved for User Program Interrupts	User		1
67h	LIM EMS Driver (AT only)			
68h-6Fh	Reserved			
70h	Real Time Clock	Hardware	F5124h	15
71h	Redirect Cascade	Hardware	F5266h	
72h-73h	Reserved			
74h	Reserved	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		

Unexpected Interrupt Handler

Description

The BIOS initializes unused interrupt vectors to the BIOS unexpected interrupt handler routine. The unexpected interrupt handler processes all interrupts that are either special user processes (INT 1Ch or INT 4Ah), or not meaningful to the BIOS, such as INT 72h, 73h, or 74h, which are all reserved in PC, XT, and AT BIOSs.

Generally speaking the BIOS either ignores unexpected software interrupts, returning control to the interrupted program with registers preserved and the Carry Flag set or it revectors the interrupt to a caller-supplied routine. An example would be INT 1Ch, for which the caller would supply a routine that would access the BIOS INT 08h System Timer ISR.

An exception is INT 0Dh. If the microprocessor is an 80286, 80386SX, or 80386, the BIOS generates a screen message about a General Protection Fault.

continued

Interrupts initialized to BIOS unexpected interrupt handlers

The following interrupts are initialized by the BIOS to different unexpected interrupt handlers. Operating systems and user applications may replace these interrupt vectors, if they wish. The BIOS provides these routines to ensure complete BIOS error recovery.

INT	Description
00h	Divide by zero
01h	Single step
03h	Breakpoint
04h	Overflow
06h	Invalid op code
07h	Math coprocessor not present
0Ah	IRQ 2 Cascade from slave 8259A
0Bh	IRQ 3 Serial Communications hardware (COM2)
0Ch	IRQ 4 Serial Communications hardware (COM1)
0Dh	IRQ 5 Fixed disk controller
0Fh	Parallel printer IRQ 7
1Bh	Keyboard Ctrl/Break
1Ch	Timer tick
72h	Reserved, IRQ 10
73h	Reserved, IRQ 11
74h	Reserved, IRQ 12
76h	IRQ 14 Fixed disk controller
77h	Reserved, IRQ 15

System BIOS Functions and BIOS Diversity

Introduction

Throughout this manual, each ROM BIOS function description includes an explanation of what the function does, the function's required inputs and outputs, and any special programming considerations or hardware "gotcha's" associated the the function.

Although all BIOSs in IBM-compatible microcomputers by definition have to be compatible with the IBM standard, there has been a tremendous amount of diversity in the level of functions supported between BIOS versions.

This diversity makes it difficult to precisely identify functions as PC-, XT-, or AT-specific. BIOS functions that are AT-only in IBM systems can be found in the Phoenix XT BIOS. And, depending on the manufacturer of the BIOS or computer, there are variations on which function appears in which XT or AT clone.

How to determine level of support

Throughout this book, the commonly accepted level of support for each ROM BIOS function is identified by the legend [PC], [XT], and/or [AT] on the same line as the function title.

Legend	Description
[PC]	Function is supported by the original IBM PC BIOS only.
[XT]	Function is supported by all IBM PC/XT and compatible ROM BIOSs. **Note:** This level of support is contained in the ROM BIOS implemented in the majority of IBM PC and PC/XT compatible systems.
[AT]	Function is supported by all IBM PC/AT and compatible ROM BIOSs.

continued

Sample: Diskette Service function description

The sample description below is excerpted from Chapter 10, which describes the ROM BIOS Diskette Service functions. As the legend indicates, diskette function AH = 00h is supported by the system BIOS commonly implemented in IBM PC/XT/AT and compatible systems.

Function: AH = 00h Reset Diskette System [XT] [AT]

Description

This function resets both the diskette controller and the specified diskette drive (the R/W arm is moved to track 0). The diskette drive number (either 00h or 01h) is specified in DL.

Function failure

Call this function when a problem occurs in attempting to access the diskette subsystem with any other function, then retry the function that failed. The diskette subsystem will not react immediately. Instead, a reset flag forces the BIOS to recalibrate the diskette drive's read/write heads the next time they are used. The heads are relocated track 0 in order to start the next I/O operation from a known state.

Input/Output

Input: AH = 00h
DL = Drive number
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

Video BIOS Functions and BIOS Diversity

Introduction

Since the introduction of the IBM PC in 1981, four video adapter types have emerged: MDA, CGA, EGA, and VGA. MDA and CGA adapters rely exclusively on the Video Service contained in the system board ROM BIOS of all PC/XT/AT and compatible computers. EGA and VGA adapters contain a dedicated ROM BIOS installed on the adapter hardware itself.

The number and sophistication of the functions supported by the video BIOSs associated with each adapter varies. As a general rule, the more recent the adapter, the greater the number of functions its BIOS supports.

Note: The Hercules display adapter allows for the display of 720x350 monochrome graphics on standard TTL monochrome monitors. However, neither the system ROM BIOS nor the Hercules adapter itself contain a BIOS. Applications programs that support Hercules graphics must contain an application-specific device driver.

How to determine level of support

When a function applies exclusively to the VGA adapter, for example, its description will bear the legend [VGA]. When the function applies to all adapter types, it will bear the legend [MDA] [CGA] [EGA] [VGA].

The table below describes the meaning of each legend.

Legend	Description
[MDA]	Function is supported by IBM Monochrome Display Adapters (MDA), and compatibles.
[CGA]	Function is supported by IBM Color Graphics Adapter (CGA), and compatibles.
[EGA]	Function is supported by IBM Enhanced Graphics Adapter (EGA), and compatibles.
[VGA]	Function is supported by IBM Video Graphics Array (VGA) and compatible adapters.

continued

Sample: Video BIOS function description

The sample description below is excerpted from Chapter 9, which describes the BIOS Video Service functions. As the legend indicates, video function AH = 02h is supported by the video BIOS associated with MDA, CGA, EGA, and VGA adapters.

Function: AH = 02h Set Cursor Position [MDA][CGA][EGA][VGA]

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

Summary: The BIOS Services

Each BIOS service executes at least one function. When a BIOS 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 BIOS service and list each BIOS function and subfunction.

Print Screen Service		The BIOS 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 BIOS Video Service provides I/O support for the MDA and CGA adapters. BIOS support for EGA and VGA adapters is contained in adapter ROM.
INT	**PARAMETER**	**FUNCTION**
10h	AH = 00h	Set Video Mode
	AH = 01h	Set Text Mode Cursor Size
	AH = 02h	Set Cursor Position
	AH = 03h	Read Current Cursor Position
	AH = 04h	Read Light Pen Position
	AH = 05h	Select New Video Page
	AH = 06h	Scroll Current Page Up
	AH = 07h	Scroll Current Page Down
	AH = 08h	Read Character/Attribute from Screen
	AH = 09h	Write Character/Attribute to Screen

continued

Video Service, cont'd		The BIOS Video Service provides I/O support for the MDA and CGA adapters. BIOS support for EGA and VGA adapters is contained in adapter ROM.
INT	PARAMETER	FUNCTION
10h	AH = 0Ah	Write Character Only to Screen
	AH = 0Bh	Set Color Palette
	AH = 0Ch	Write Pixel
	AH = 0Dh	Read Pixel
	AH = 0Eh	Write Teletype To Active Page
	AH = 0Fh	Return Video Status
	AH = 10h	Set Palette/Color Registers:

Parameter	Subfunction
AL = 00h	Set single palette
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 (border color)
AL = 09h	Read all palette registers and overscan register (border color)
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
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

Video Service, cont'd

The BIOS Video Service provides I/O support for MDA and CGA. BIOS support for EGA and VGA adapters is contained in adapter ROM.

INT	PARAMETER	FUNCTION
10h	AH = 11h	Load Character Generator:

Parameter	Subfunction
AL = 00h	Load user text mode font
AL = 01h	Load ROM 8x14 text mode font
AL = 02h	Load ROM 8x8 double dot text mode font
AL = 03h	Set block specifier (text mode only))
AL = 04h	Load 8x16 ROM text mode font
AL = 10h	Load user text mode font (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 8x16 ROM text mode font (after mode set)
AL = 20h	Set user graphics characters pointer at INT 1Fh (8x8 font)
AL = 21h	Set user graphics font pointer at INT 43h
AL = 22h	Use ROM 8x14 font for graphics
AL = 23h	Use ROM 8x8 double dot font for graphics
AL = 24h	Use ROM 8x16 font for graphics
AL = 30h	Get font pointer information

INT	PARAMETER	FUNCTION
10h	AH = 12h	Alternate Select:

Parameter	Subfunction
BL = 10h	Return configuration 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

continued

Video Service, cont'd

The BIOS Video Service provides I/O support for the MDA and CGA adapters. BIOS support for EGA and VGA adapters is contained in adapter ROM.

INT	PARAMETER	FUNCTION
10h	AH = 13h	Write String:
	AH = 14h–19h	Reserved
	AH = 1Ah	Read/Write Display Combination Code:
	AH = 1Bh	Return Functionality/State Information
	AH = 1Ch	Save/Restore Video State
	AH = 1Dh–FFh	Reserved

Parameter	Subfunction
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)

Parameter	Subfunction
AL = 00h	Read display combination code
AL = 01h	Write display combination code

Equipment List Service

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

INT	PARAMETER	FUNCTION
11h	None	Read Equipment List

continued

Memory Size Service		The BIOS 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 BIOS 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 BIOS automatically revectors all 13h Diskette Service requests to INT 40h. 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

Fixed Disk Service		The BIOS 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 Diskette(s) and Fixed Disk
	AH = 01h	Read Fixed Disk Status
	AH = 02h	Read Sectors
	AH = 03h	Write Sectors
	AH = 04h	Verify Sectors
	AH = 05h	Format Cylinder
	AH = 06h	Format Bad Track
	AH = 07h	Format Drive
	AH = 08h	Read Drive Parameters
	AH = 09h	Initialize Drive Parameters
	AH = 0Ah	Read Long Sectors
	AH = 0Bh	Write Long Sectors
	AH = 0Ch	Seek to Cylinder
	AH = 0Dh	Alternate Fixed Disk Reset
	AH = 0Eh	Diagnostics 1: Read Test Buffer
	AH = 0Fh	Diagnostics 2: Write Test Buffer
	AH = 10h	Test for Drive Ready
	AH = 11h	Recalibrate Drive
	AH = 12h	Controller RAM Diagnostic
	AH = 13h	Controller Drive Diagnostic
	AH = 14h	Controller Internal Diagnostic
	AH = 15h	Read Fixed Disk Type
	AH = 16h–FFh	Reserved

continued

Serial Communication Service		The BIOS Serial Communication Service performs RS-232-C character I/O for IBM-compatible serial port adapters.
INT	PARAMETER	FUNCTION
14h	AH = 00h	Initialize Serial Communications Port
	AH = 01h	Send Character
	AH = 02h	Receive Character
	AH = 03h	Read Serial Port Status
	AH = 04h-FFh	Reserved

System Services		The BIOS 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	Turn Cassette Motor On
	AH = 01h	Turn Cassette Motor Off
	AH = 02h	Read Cassette
	AH = 03h	Write to Cassette
	AH = 04h-4Eh	Reserved
	AH = 4Fh	Keyboard Intercept
	AH = 50h-7Fh	Reserved
	AH = 80h	Device Open
	AH = 81h	Device Close
	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

continued

System Services, cont'd		The BIOS 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 = 89h	Switch Processor to Protected Mode
	AH = 8Ah-8Fh	Reserved
	AH = 90h	Device Busy
	AH = 91h	Interrupt Complete
	AH = 92h-BFh	Reserved
	AH = C0h	Return System Configuration Parameters

Keyboard Service		The BIOS Keyboard Service interfaces the operating system and application programs with the keyboard.
INT	PARAMETER	FUNCTION
16h	AH = 00h	Read Keyboard Input
	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 = 10h	Read Extended Keyboard Input
	AH = 11h	Return Extended Keyboard Status
	AH = 12h	Return Extended Shift Flag Status
	AH = 13h-FFh	Reserved

continued

Parallel Printer Service		The BIOS 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 Printer Status
	AH = 03h–FFh	Reserved

Time-of-Day Service		The BIOS 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 Time Counter
	AH = 01h	Set System Timer Time Counter
	AH = 02h	Read Real Time Clock Time
	AH = 03h	Set Real Time Clock Time
	AH = 04h	Read Real Time Clock Date
	AH = 05h	Set Real Time Clock Date
	AH = 06h	Set Real Time Clock Alarm
	AH = 07h	Reset Real Time Clock Alarm
	AH = 80h	Set Sound Source (PCjr only)

2

Chapter 2
System RAM Data

Overview

Introduction

The BIOS 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.
BIOS Data Area	400h to 4FFh	BIOS work area. Contains data definitions related to BIOS fixed disk, diskette, keyboard, video, and other BIOS features and functions.

In this chapter

This chapter presents the following information:

- Interrupt Vectors
- BIOS Data Area
- Power-On Self Test Definitions
- Data Definitions

Interrupt vector table

The table below identifies each interrupt by function and type. Where applicable, it lists the interrupt vector address initialized by the BIOS at POST. System software may revector an interrupt at or shortly after the boot process is completed, so these values may not be the same in every system on every occasion. Low-level hardware interrupts are handled differently on the PC and XT. The 80286 and 80386 architectures require the use of some of these interrupts.

Note: Further discussion of interrupts is limited to those interrupts directly related to the BIOS. 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
05h	Bounds Exception (80286, 80386)	Hardware		
06h	Invalid Op Code (80286, 80386)	Hardware		
06h	Reserved (PC only)	Hardware		
07h	Reserved (PC only)	Hardware		
07h	Math Coprocessor Not Present	Hardware		
08h	Double Exception Error (80286, 80386) (AT only)	Hardware		
08h	System Timer (IRQ 0)	Hardware	FFEA5h	15
09h	Keyboard	Hardware	FE987h	8
09h	Math Coprocessor Segment Overrun (80286, 80386) (AT only)	Hardware		
0Ah	IRQ 2 Cascade from Second Programmable Interrupt Controller	Hardware		
0Ah	Invalid Task Segment State (80286, 80386) (AT only)	Hardware		
0Ah	IRQ 2 (Reserved) (PC only)	Hardware		

continued

Interrupt vector table, cont'd

INT	Function	Type	Vector	Ref. Chap.
0Bh	Serial Communications (COM2)	Hardware		
0Bh	Segment Not Present (80286, 80386)	Hardware		
0Ch	Serial Communications (COM1)	Hardware		
0Ch	Stack Segment Overflow (80286, 80386)	Hardware		
0Dh	Parallel Printer (LPT2) (AT only)	Software		14
0Dh	IRQ 5 Fixed Disk (XT only)	Software		11
0Dh	General Protection Fault (80286, 80386)	Software		
0Eh	IRQ 6 Diskette	Software	FEF57h	10
0Eh	Page Fault (80386 only)	Software		
0Fh	Parallel Printer (LPT1) IRQ 7	Software		14
10h	Video	Software	FF065h	9
10h	Numeric Coprocessor Fault (80286, 80386)	Software		
11h	Equipment List	Software	FF84Dh	16
12h	Memory Size	Software	FF841h	16
13h	Fixed Disk/Diskette	Software	FE3FEh	10/11
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 (PC only)	Software	F1C90h	16
18h	Process Boot Failure (XT, AT)	Software	F1C90h	16
19h	Bootstrap Loader	Software	FE6F2h	16
1Ah	Time-of-Day	Software	FFE6Eh	15
1Bh	Keyboard Break	Software	FFF53h	8
1Ch	User Timer Tick	User	FFF53h	15
1Dh	Video Parameter Table	BIOS Table	FF0A4h	9
1Eh	Diskette Parameter Table	BIOS Table	FEFC7h	10

continued

Interrupt vector table, cont'd

INT	Function	Type	Vector	Ref. Chap.
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-59h	Reserved			
5Ah	Cluster Adapter			
5Bh-5Fh	Reserved			
60h-66h	Reserved for User Program Interrupts	User		1
67h	LIM EMS Driver (AT only)			
68h-6Fh	Reserved			
70h	Real Time Clock	Hardware	F5124h	15
71h	Redirect Cascade	Hardware	F5266h	
72h-73h	Reserved			
74h	Reserved	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		

Introduction

The BIOS uses system RAM for storage of data definitions specific to the BIOS. This data area is almost exactly the same for the PC, XT, and AT architectures. Differences are noted.

System RAM summary

The table below lists the system RAM data definitions in offset order. Offsets are from segment address 40:00h.

Location	BIOS Service	Description
00h	INT 14h	I/O address of up to 4 serial communications adapters
08h	INT 17h	I/O address of up to 4 parallel printer adapters
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	Manufacturing test port (AT only)
13h	INT 12h	Installed memory in Kilobytes
15h	POST	Manufacturing test port (AT only)

continued

5150 : # of 4 KB blocks on motherboard

5160/5170 : 14h = # of 256 KB blocks
13h = remainder
(example: 14h = 2 and 13h = 128 dec
equates to 640 KB)

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	Extended 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 = 1 Recalibrate drive 1 Bit 0 = 1 Recalibrate drive 0
3Fh	INT 13h	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 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 (video)
4Eh	INT 10h	Current page address (video)
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 (video)
63h	INT 10h	6845-compatible I/O port number for current mode (Port 03D4h or 03B4h) (video)
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	Timer count — number of ticks since midnight (four bytes)
70h	INT 1Ah	24 hour roll over flag

continued

System RAM summary, cont'd

Location	BIOS Service	Description
71h	INT 16h	Ctrl-Break flag (bit 7 = 1 <Ctrl><Break> pressed)
72h	POST	System Reset Flag (AT only), where: 1234h = Bypass memory test (also warm boot) 4321h = Preserve memory 64h = Burn in mode
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 0-3)
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) (VGA only)
85h	INT 10h	Character height (bytes/character) (VGA only)

continued

System RAM summary, cont'd

Location	BIOS Service	Description
87h	INT 10h	Video control bits (VGA only), 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 (VGA only), 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 (VGA only), 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	Index into DCC table (VGA only)
8Bh	INT 13h	Diskette data rate information Bits 7-6 = Last data rate set by controller, where: 00b = 500 Kilobits/second (Kbs) 01b = 300 Kbs 10b = 250 Kbs 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 Bits 1-0 = Reserved
8Ch	INT 13h	Fixed disk status register (AT only)
8Dh	INT 13h	Fixed disk error register (AT only)
8Eh	INT 13h	Fixed disk interrupt flag (AT only)

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	Keyboard 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	Keyboard LED 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 (AT only)
9Ah	INT 15h	User wait flag segment address (AT only)
9Ch	INT 15h	Least significant byte of wait count (AT only)
9Eh	INT 15h	Most significant byte of wait count (AT only)
A0h	INT 15h	Wait active flag, (AT only) 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) (VGA only)
B0h-B5h		Reserved
B6h-B8h	POST	Reserved for POST (AT only)
0C0h-0CDh		Reserved
CEh	INT 1Ah	Count of days since 1-1-80
CFh-FFh		Reserved
100h	INT 05h	Print screen status byte

POST data definitions

The data definitions used by the power-on self test (POST) routine are stored in system RAM at segment 40h and are presented below in offset order.

Location	Length	Description
12h	1 Byte	Reserved for manufacturer test
15h	1 Word	Reserved for manufacturer test
67h	1 Word	Pointer to reset code upon system reset with memory preserved.
6Bh	1 Byte	Last interrupt that occurred
72h	1 Word	Reset flag, where: 1234h = Bypass memory test 4321h = Preserve memory 64h = Burn in mode

Video data definitions

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

Location	Length	Description
10h	1 Byte	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
49h	1 Byte	Video mode setting.
4Ah	1 Word	Number of columns on screen (40 or 80).
4Ch	1 Word	Current page size (in bytes).
4Eh	1 Word	Offset of 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 the upper left corner of screen.
60h	1 Word	Cursor type defined as 6845-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, where: 03B4h = Monochrome 03D4h = Color
65h	1 Byte	Current mode select register.
66h	1 Byte	Current palette value.
84h	1 Byte	Number of rows on screen minus one (24 or 25)
85h	1 Word	Character height (bytes/character)

continued

Video data definitions, cont'd

Location	Length	Description
87h	1 Byte	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 If EGA-compatible adapter is active = 1 If EGA-compatible adapter is inactive Bit 2 = Wait for display enable Bit 1 = 0 If color or ECD monitor is attached to EGA-compatible monitor = 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 = 1 If inhibit cursor translation
88h	1 Byte	EGA switch data where: Bits 7-4 = Feature connector bits 3-0, respectively Bits 3-0 = Option switches 3-0, respectively
A8h	1 Word	Pointer to video parameter table and overrides (in segment:offset format).
100h	1 Byte	Print flag, where: Bit 7 = 1 Print screen in progress

INT 11h Equipment List Service Data Definitions

Equipment data definitions

The data definition used by INT 11h, Equipment List Service, is stored in system RAM in segment 40h and is presented in offset order below.

Location	Length	Description
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 disk drives, where: 00b = 1 Disk drive 01b = 2 Disk drives Bits 5-4 = Initial video mode, where: 00b = EGA or PGA 01b = 40x25 color 10b = 80x25 color 11b = 80x25 black and white Bit 2 = Pointing device Bit 1 = 1 If math coprocessor Bit 0 = Diskette available for boot

INT 12h Memory Size Service Data Definition

Memory data definition

The data definition used by INT 12h, Memory Size Service, is stored in system RAM in segment 40h.

Location	Length	Description
13h	1 Word	Installed memory in K minus 1K

Diskette data definitions

The data definitions used by INT 13h, Diskette Service, are stored in system RAM in segment 40h and are presented below in offset order.

Location	Length	Description
3Eh	1 Byte	Diskette drive recalibrate status, where: Bit 7 = 1 Diskette hardware interrupt has occurred Bits 6–4 = Not used Bits 3–2 = Reserved Bit 1 = 1 Recalibrate drive 1 Bit 0 = 1 Recalibrate drive 0
3Fh	1 Byte	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 Bit 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	1 Byte	Diskette motor time-out count
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 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

continued

Diskette data definitions, cont'd

Location	Length	Description
42h	7 Bytes	Disk controller status bytes
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
75h	1 Byte	Number of fixed disk drives
8Bh	1 Byte	Diskette data rate information Bits 7-6 = Last data rate set by controller, where: 00b = 500 Kilobits/second (Kbs) 01b = 300 Kbs 10b = 250 Kbs 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 Bits 1-0 = Reserved
8Ch	1 Byte	Fixed disk drive controller status
8Dh	1 Byte	Fixed disk controller error status
8Eh	1 Byte	Fixed disk interrupt flag

continued

Diskette data definitions, cont'd

Location	Length	Description
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
90h	1 Word	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/ 1.2M 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, 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	1 Word	Diskette work area. Each entry contains the first value tried. (One byte per drive. Drive 0 at 40:92h Drive 1 at 40:93h)
94h	1 Word	Current track number for both drives. (One byte per drive. Drive 0 at 40:94h Drive 1 at 40:95h)

INT 14h Serial Communication Service Data Definitions

Serial data definitions

The data definitions used by the INT 14h Serial Communications services are stored in system RAM in segment 40h and are presented below in offset order.

Location	Length	Description
00h	4 Words	I/O address of up to 4 asynchronous communications adapters
78h		Printer time-out table
7Ch	2 Words	Serial (RS232) time-out table for serial ports 0 through 3.

INT 16h Keyboard Data Service Definitions and INT 09h Keyboard Service Interrupt Data Definitions

Keyboard data definitions

The data definitions used by the INT 16h and INT 9h Keyboard Service are stored in system RAM in segment 40h. They are presented below in offset order.

Location	Length	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	Length	Description
18h	1 Byte	Extended 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 Word	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	Reset flag, where: 1234h = Bypass memory test 4321h = Preserve memory
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 17h Parallel Printer Service Data Definitions

The data definitions used by INT 17h Parallel Printer Service are stored in system RAM in segment 40h and are presented below in offset order.

Location	Length	Description
08h	4 Words	I/O address of up to 4 printer adapters
78h	2 Words	Printer time-out table for printers 1-3
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 disk drives, where: 00b = 1 Disk drive 01b = 2 Disk 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 Bit 1 = 1 If math coprocessor Bit 0 = Diskette available for boot

INT 19h Bootstrap Loader Service Data Definitions

System RAM data

The data definitions used by INT 19h are stored in system RAM in segment 40h and are presented below in offset order.

Location	Size	Description
12h	1 Byte	Reserved for manufacturers test, where: Bits 7-1 = Reserved Bit 0 = 1 Manufacturing test mode = 0 Non-manufacturing test mode
72h	1 Word	Reset flag, where: 1234h = Bypass memory test 4321h = Preserve memory 64h = Burn in mode

INT 1Ah Time-of-Day Service Data Definitions

Timer data definitions

The data definitions used by the INT 1Ah Time-of-Day Service and by the INT 08h Timer Interrupt are stored in system RAM in segment 40h and are presented below in offset order.

Location	Length	Description
6Ch	4 Bytes	Timer tick count (count of timer ticks since midnight)
70h	1 Byte	24 hour rollover flag
98h	1 Word	User wait flag offset address
9Ah	1 Word	User wait flag segment address
9Ch	1 Word	Least significant wait count
9Eh	1 Word	Most significant wait count
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

3

Chapter 3
CMOS RAM Data

Overview

Introduction

The AT BIOS uses nonvolatile CMOS RAM to store real time clock, system configuration, system diagnostic, and other information. In general, only AT and AT-compatible systems have CMOS RAM.

Where CMOS RAM data is located

The table below describes the CMOS RAM areas available to the BIOS. CMOS RAM data is available only in AT and AT-compatible systems.

Data Area	I/O Port Access	Size	Description
CMOS RAM Data Area	070h & 071h	64 bytes	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.

continued

In this chapter

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

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

CMOS RAM I/O Ports

The BIOS 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

MC146818A Real Time Clock

The AT BIOS supports a Motorola MC146818A or equivalent Real Time Clock. This chip is assumed to have at least 64 bytes of nonvolatile CMOS RAM available to store configuration data. The first nine bytes, at offset locations 00h – 09h) are used to store RTC information. The next four bytes (offsets 0Ah – 0Dh) store status information. The balance of CMOS RAM storage (0Eh – 3Fh) is used to store system configuration and other data.

Writing to CMOS RAM

The bit definitions for most of the standard CMOS RAM data definitions are provided in this chapter. To write data to CMOS RAM:

1. Inhibit interrupts.
2. Write the CMOS RAM address to which the data is to be written to I/O port 0070h.
3. Write the data to be written to I/O port 0071h.

Values can be written to CMOS RAM locations 00h – 09h much more quickly using INT 1Ah functions 02h – 07h.

Reading from CMOS RAM

The bit definitions for most of the standard CMOS RAM data definitions are provided in this chapter. To read from RTC CMOS RAM:

1. Inhibit interrupts.
2. Write the CMOS RAM address from which the data is to be read to I/O Port 0070h.
3. Read from I/O port 0071h.

Values can be read from CMOS RAM locations 00h – 09h much more quickly using INT 1Ah functions 02h – 07h.

CMOS RAM Data

Introduction

The BIOS 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.

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

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 (update cycle) = 1 Abort any update cycle in progress 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 frequency set in Status Register A Bit 2 = 1 Calendar is in binary format = 0 Calendar is in BCD format Bit 1 = 1 24-hour clock = 0 12-hour clock Bit 0 = 1 Enable Daylight Savings Time
0Ch	1 Byte	Status Register C, where: Bits 7-4 = IRQF, PF, AF, and UF flags, respectively. Only valid if alarm interrupt, update-ended interrupt or periodic interrupt is enabled. 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 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 time found invalid Bit 1 = 1 Reserved Bit 0 = 1 Reserved
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 doubleword pointer with EOI 06h = Protected tests pass 07h = Protected tests fail 08h = Memory size fail 09h = INT 15h Block Move 0Ah = JMP doubleword pointer without EOI 0Bh = Used by 80386
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
13h	1 Byte	Reserved

continued

Configuration data definitions, cont'd

Location	Size	Description
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 = CGA in 40-column mode 10b = CGA in 80-column mode 11b = CGA in monochrome mode Bits 3-2 = Reserved Bit 1 = 1 Math coprocessor 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	Drive type of first fixed disk
1Ah	1 Byte	Drive type of second fixed disk
1Bh - 2Dh		Reserved
2Eh	1 Byte	High byte checksum for 10h - 2Dh
2Fh	1 Byte	Low byte checksum for 10h - 2Dh
30h	1 Byte	Low byte of actual expansion memory size
31h	1 Byte	High byte of actual expansion memory size
32h	1 Byte	Century in BCD
33h	1 Byte	Information flag
34h - 3Fh	6 Words	Reserved

4

Chapter 4
ROM BIOS Data

Overview

Introduction

In addition to the code for the various BIOS services, the BIOS also contains tabular data that is used to initialize devices. This data, particularly for fixed disk and diskette parameter tables, may be different for PC, XT, and AT systems. The differences are described in this chapter.

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
- Diskette Parameter Table
- AT Fixed Disk Parameter Table
- XT Fixed Disk Parameter Table
- Baud Rate Initialization

ROM Address Compatibility Table

Description

The BIOS maintains compatibility between PC, XT, and AT versions by providing a list of jump instructions that insure that both table information and calls to specific BIOS services are located at fixed entry points.

ROM BIOS address map

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

continued

ROM BIOS address map, cont'd

Address	Description
FFEF3h	Initial Interrupt Vector Offsets Loaded by POST
FFF53h	IRET Instruction for Dummy Interrupt Handler
FFF54h	INT 05h Print Screen Service Entry Point
FFFF0h	Power-up Entry Point
FFFF5h	ASCII Date ROM was built — 8 characters xx/xx/xx, (for example: 05/31/88)
FFFFEh	System Model ID, where: FCh = AT FDh = IBM PCjr FEh = XT FFh = PC

System Configuration Data Table (AT only)

Description

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

Offset (hex)	Initial Value	Length	Description
00h	00h	2 Bytes	Number of bytes in this table (minimum = 8)
02h	02h	1 Byte	Model byte, where: PC = FFh PC XT = FEh PC XT = FBh PCjr = FDh AT = FCh XT-286 = FCh PC Convertible = F9h Unknown system board = FFh
03h	03h	1 Byte	Submodel byte, where: PC = 00h PC XT = 00h PCjr = 00h AT = 00h AT = 01h XT-286 = 02h PC Convertible = 00h Unknown system board = FFh
04h	04h	1 Byte	BIOS revision level (first release = 00b)
05h	05h	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 = Reserved Bit 1 = 0 PC-type I/O channel implemented Bit 0 = 0 Reserved
06h	06h	1 Byte	Feature information byte 2 (reserved, zeros)
07h	07h	1 Byte	Feature information byte 3 (reserved, zeros)
08h	08h	1 Byte	Feature information byte 4 (reserved, zeros)
09h	09h	1 Byte	Feature information byte 5 (reserved, zeros)

Diskette Parameter Table

Description

The diskette device service routine in PC/XT/AT systems contains a table of parameters used to manipulate diskette drives. For many AT systems, values have been modified so 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. This arrangement allows operating systems or application programs to change the INT 1Eh vector to point to a different set of diskette parameters.

Diskette parameter table structure

<table>
<tr><th>Offset</th><th>Description</th></tr>
<tr><td>00h</td><td>First data byte of the diskette Specify command, where:
<table>
<tr><th rowspan="2">Drive Type/
Transfer Rate</th><th colspan="6">Setting for Bits 7-4</th></tr>
<tr><th>F</th><th>E</th><th>D</th><th>C</th><th>B</th><th>A</th></tr>
<tr><td>360K/250 Kbs</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
<tr><td>Other/500 Kbs</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
<tr><td>Other/300 Kbs</td><td>1.7</td><td>3.3</td><td>5</td><td>6.7</td><td>8.3</td><td>10</td></tr>
<tr><td>Other/250 Kbs</td><td>2</td><td>4</td><td>6</td><td>8</td><td>10</td><td>12</td></tr>
</table>
Bits 3-0 = Head unload time. Default is 240 milliseconds (0Fh).</td></tr>
<tr><td>01h</td><td>Second data byte of the diskette Specify command, where:
Bits 7-1 = Head load time. Default is 01h (4 milliseconds).
Bit 0 = Non-DMA mode flag
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.</td></tr>
<tr><td>02h</td><td>Motor turn-off delay. 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. The default value for this field is 25h.</td></tr>
</table>

continued

Diskette parameter table structure, cont'd

Offset	Description
03h	Bytes per sector. This field is encoded in the following way to match the encoding used by the diskette controller: 00h = 128 bytes per sector 01h = 256 bytes per sector 02h = 512 bytes per sector (default value) 03h = 1024 bytes per sector
04h	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. The range of values is: 08h = 8 sectors per track (320K drive) 09h = 9 sectors per track (360K/720K 5.25-inch drive) 15h = 15 sectors per track (5.25-inch 1.2 MB drive) 18h = 18 sectors per track (3.5-inch drives)
05h	Gap length. The length of the gap between sectors. 1Bh for a 3.5-inch diskette drive and 2Ah for a 5.25-inch diskette drive.
06h	Data length. Since the bytes per sector field is nonzero, this field is meaningless and is set to FFh.
07h	Gap length for format. The length of the gap between sectors to maintain when formatting. The format gap length is 6Ch for a 3.5-inch drive and 50h for a 5.25-inch drive.
08h	Fill byte for format. The default is F6h.
09h	Head settle time. The amount of time in milliseconds the diskette DSR must wait for the heads to settle after doing a seek operation. For a 3.5-inch diskette drive, this field is 0Fh. 1.2 MB drives require 15 milliseconds and 360K drives, 20 - 25 milliseconds.
0Ah	Motor start time. The amount of time in eighths of a second that the diskette DSR must wait for the motor to come up to speed before doing an I/O operation. Most drives have a motor start time of one second so the default value is 08h for most operations, except Read and Verify, where the default value is either 04h or 05h.

AT Fixed Disk Parameter Table

Description

The AT fixed disk parameter 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 AT fixed disk parameter table. To maintain compatibility, the fixed disk parameter table is based at F000:E401h.

Parameter table structure

Each entry in the fixed disk parameter 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 Bytes	Reserved
05h	2 Bytes	Starting write precompensation cylinder
07h	1 Byte	Reserved
08h	1 Byte	Control byte, where: Bits 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 drive type table

The fixed disk drives that the AT BIOS supports are named in a ROM-based table. Table entries 1-47 are popular fixed disk drive types. In the Phoenix AT BIOS, table entries 48 and 49, however, are left for the end user to define. To define fixed disk drive types 48 and 49, the end user runs SETUP. Once specified, the drive characteristics remain stored in CMOS RAM. The end user can add any type of AT-compatible fixed disk drive to a system.

The fixed disk drive table appears as follows in ROM:

#	Drive Type	Cylinders	Heads	Write Precomp	Landing Zone	Sectors
1	IBM 5.25" 10 MB	306	4	128	305	17
2	IBM 5.25" 20 MB Seagate ST-225 CDC Wren II 9415-5-25 * Miniscribe 8438F	615	4	300	615	17
3	IBM 5.25" 30 MB	615	6	300	615	17
4	IBM 5.25" 62 MB	940	8	512	940	17
5	IBM 5.25" 46 MB	940	6	512	940	17
6	IBM 5.25" 20 MB Miniscribe MS 8425 Tandon TM 262 *** Tandon TM 702AT ***	615	4	-1	615	17
7	IBM 5.25" 30 MB	462	8	256	511	17
8	IBM 5.25" 30 MB Seagate ST-4038 *** CDC Wren II 9415-5-38 * *** Tandon TM 703AT ***	733	5	-1	733	17
9	IBM 5.25" 112 MB Priam IDED 130*	900	15	-1	901	17
10	IBM 5.25" 20 MB Priam IDED75, 100, 120, 150, 160, 230, 330*	820	3	-1	820	17

* The OEM may have to disable drive connector J1 pin 2 when used on the AT.
** Remove jumper W3.
*** Values for Write Precomp of -1, 0, or the last track are assumed to be equivalent.

continued

Fixed disk drive type table, cont'd

#	Drive Type	Cylinders	Heads	Write Precomp	Landing Zone	Sectors
11	IBM 5.25" 35 MB Priam IDED 40, 45, 45H Priam ID45T-S, ID45T-Q*	855	5	-1	855	17
12	IBM 5.25" 49 MB	855	7	-1	855	17
13	IBM 5.25" 20 MB	306	8	128	319	17
14	IBM 5.25" 42 MB	733	7	-1	733	17
15	Reserved					
16	IBM 5.25" 20 MB	612	4	0	663	17
17	IBM 5.25" 40 MB	977	5	300	977	17
18	IBM 5.25" 56 MB Priam ID 60, 62*	977	7	-1	977	17
19	IBM 5.25" 5 9MB	1024	7	512	1023	17
20	IBM 5.25" 30 MB	733	5	300	732	17
21	IBM 5.25" 42 MB	733	7	300	732	17
22	IBM 5.25" 30 MB	733	5	300	733	17
23	IBM 5.25" 10 MB	306	4	0	336	17
24	Priam IDED 62	1024	7	-1	1024	17
25	NEC 5126 20 MB	615	4	0	615	17
26	Micropolis 1323 *** Rodime 5040	1024	4	-1	1024	17
27	Micropolis 1323A *** Seagate ST-4077R Miniscribe 3053/6053 Priam IDED 45, 45H**	1024	5	-1	1024	17
28	Micropolis 1325 *** Miniscribe 6085 Maxtor XT-1085 ***	1024	8	-1	1024	17
29		512	8	256	512	17

* The OEM may have to disable drive connector J1 pin 2 when used on the AT.

** Remove jumper W3.

*** Values for Write Precomp of -1, 0, or the last track are assumed to be equivalent.

continued

Fixed disk drive type table, cont'd

#	Drive Type	Cylinders	Heads	Write Precomp	Landing Zone	Sectors
30	Syquest SQ312RD ** Miniscribe MS 8212 ***	615	2	615	615	17
31	CDC Wren II 9420-5-51 **	989	5	0	989	17
32		1020	15	-1	1024	17
33	Reserved					
34	Reserved					
35	Seagate ST-4096 Seagate ST-4144 Seagate ST-4144R	1024	9	1024	1024	17 26
36	Miniscribe MS 6053 Microscience HH1050	1024	5	512	1024	17
37		830	10	-1	830	17
38	NEC 67 MB	823	10	256	824	17
39	NEC D5126H 20 MB	615	4	128	664	17
40	IBM 5.25" 40 MB NEC D5146H	615	8	128	664	17
41	IBM 5.25" 114 MB	917	15	-1	918	17
42	IBM 5.25" 127 MB Priam IDED 130**	1023	15	-1	1024	17
43	NEC	823	10	512	823	17
44	Seagate ST-251 40 MB	820	6	-1	820	17
45	Reserved					
46	CDC Wren II (1) 9415-5-86	925	9	-1	925	17
47		699	7	256	700	17
48	Configurable by end user					
49	Configurable by end user					

* The OEM may have to disable drive connector J1 pin 2 when used on the AT.
** Remove jumper W3.
*** Values for Write Precomp of -1, 0, or the last track are assumed to be equivalent.

continued

Fixed disk drive type table, cont'd

If a table entry contains -1 for Write Precompensation, then there is no write precompensation for this fixed disk drive type. If the Write Precompensation is zero, then there is write precompensation for all cylinders.

Drive types 48 and 49 in a Phoenix BIOS can be edited by the Phoenix SETUP utility. Parameter values entered in SETUP for these drives are stored in CMOS RAM.

XT Fixed Disk Parameter Table

Description

The XT fixed disk parameter table defines the types of fixed disk drives that can be used in an XT system. As part of system configuration, the INT 41h vector is set up to point to the fixed disk parameter entry associated with drive 0. To maintain compatibility, the fixed disk parameter table is based at F000:E401h.

Parameter table structure

Each entry in the fixed disk parameter table occupies 16 bytes as defined below.

Offset	Bytes	Description
00h	2	Maximum number of cylinders
02h	1	Maximum number of heads
03h	2	Starting reduced write current cylinder
05h	2	Starting write precompensation cylinder
07h	1	Maximum ECC burst length
08h	1	Control byte, where: Bit 7 = 1 Disable fixed disk retries Bit 6 = 1 Disable ECC retries Bits 5-3= 1 Reserved Bits 2-0= Drive option
09h	1	Standard timeout value
0Ah	1	Time-out value for format drive command
0Bh	1	Time-out value for check drive command
0Ch	4	Reserved

continued

XT fixed disk drive type table

There are four fixed disk drive types supported by the XT-compatible Fixed Disk Service. The parameters for each of these XT-compatible drive types are:

Parameter	Type 0	Type 1	Type 2	Type 3
Maximum cylinders	306	612	615	306
Maximum head(s)	4	4	4	8
Reduced write current cyl.	306	612	615	306
Write precomp cylinder	0	0	300	128
ECC Data Burst Length	11	11	11	11
Control byte	5	5	5	5
Standard time-out	12	32	24	12
Format drive time-out	180	180	180	180
Check drive time-out	40	40	40	40
Reserved	0	0	0	0
Reserved	0	0	0	0
Reserved	0	0	0	0
Reserved	0	0	0	0

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

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:

$$\frac{1{,}843{,}200}{16} = \frac{115200}{\text{Baud Rate}} = \text{Divisor}$$

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

5

Chapter 5
I/O Port Addresses

Overview

I/O Ports

In the PC, XT, and AT architecture, the bus can address either system memory or I/O. Four bytes are used for the address formation, so I/O port addresses can range from 0000h to FFFFh for a total of 64K locations. The microprocessor can address as many I/O locations as it can address memory locations in any given memory segment. In an AT system, addresses 0000h to 00FFh are reserved for the system board and addresses 0100h - 03FFh are reserved for the I/O channel.

This I/O addressing capability gives the ISA architecture considerable power in that a system is able to access I/O devices through a broad range of addresses.

Standard I/O port protocols have been established for many common input devices. These standard I/O port address definitions are presented in the following pages. Where possible, we have provided exact bit definitions for I/O ports. Note that some IBM-compatible systems use nonstandard I/O devices or nonstandard I/O port definitions. For instance, the location of the third and fourth serial ports, given here as 3220h and 3228h, may vary from system to system.

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

continued

I/O Address	Read/Write Status	Description
000Ch	W	DMA Clear Byte Pointer
000Dh	W	DMA Master Clear Byte
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
0019h	R/W	DMA scratch register
0020h	R	Programmable Interrupt Controller (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 = 0 Only used in 80/85 mode Bit 4 = 1 Reserved Bit 3 = 0 Reserved 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

I/O Address	Read/Write Status	Description
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 = 0 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 = 0 Reserved Bit 4 = 0 No special fully-nested mode = 1 Special fully-nested mode Bits 3-2 = 00b Non-buffered mode = 01b Non-buffered mode = 10b Buffered mode/slave = 11b 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

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
0022h–003Fh	R/W	Reserved for use by Intel 8259 Programmable Interrupt Controller
0040h	R/W	Programmable Interrupt Timer — Read/write counter 0
0041h	R/W	Programmable Interrupt Timer register
0042h	R/W	Programmable Interrupt Timer — miscellaneous register

continued

I/O Address	Read/Write Status	Description
0043h	W	Programmable Interrupt Timer, 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	Programmable Interrupt Timer, miscellaneous register (AT only)
0047h	W	Programmable Interrupt Timer, control word register for counter 0 (AT only), where: Bits 7-6 = 00b Select counter 0 = 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
0048h-005Fh	R/W	Used by Programmable Interrupt Timer
0060h	W	Keyboard data input buffer (AT only)
0060h	R	Keyboard data output buffer (AT only)
0061h	R/W	8042 control register (AT only), 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

I/O Address	Read/Write Status	Description
0061h	W	8255 output register (XT only), where: Bit 7 = 1 Clear keyboard) Bit 6 = 0 Hold keyboard clock low Bit 5 = 0 Enable I/O check Bit 4 = 0 Enable RAM parity check Bit 3 = 0 Read low switches Bit 2 = Reserved Bit 1 = 1 Speaker data enable Bit 0 = 1 Enable timer 2 gate to speaker
0062h	R/W	8255 input register (XT only), where: Bit 7 = 1 RAM parity check Bit 6 = 1 I/O channel check Bit 5 = 1 Timer channel 2 out Bit 4 = Reserved Bit 3 = 1 System board RAM size type 1 Bit 2 = 1 System board RAM size type 0 Bit 1 = 1 Coprocessor installed Bit 0 = 1 Loop in POST
0063h	R/W	8255 Command mode register (XT only), where: Bits 7-6 = Number of diskette drives, where: 00b - 1 drive 01b - 2 drives 10b - 3 drives 11b - 4 drives Bits 5-4 = Type of display at power-on, where: 00b - Reserved 01b - 40x25 color (mono mode) 10b - 80x25 color (mono mode) 11b - MDA (80x25) Bits 3-2 = Memory on system board (256K chips), where: 00b - 256K 01b - 512K 10b - 576K 11b - 640K Bits 3-2 = Memory on system board 964K chips), where: 00b - 64K 01b - 128K 10b - 192K 11b - 256K Bits 1-0 = Reserved

continued

I/O Address	Read/Write Status	Description
0064h	W	8042 keyboard input buffer (AT only)
0064h	R	8042 Status (AT only), 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
0065h-006Fh		Reserved for use by 8255 (XT) or 8042 (AT)
0070h	W	CMOS RAM address register port (AT only), where: Bit 7 = 1 NMI disable Bits 6-0 = 0 CMOS RAM address
0071h	R/W	CMOS RAM data register port (AT only)
0080h	R	Manufacturing test port (AT only)
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, cascade to Channels 5-7
0093h-009Fh		Reserved for DMA Controller
00A0h	R/W	NMI mask register (XT)
00A0h	R/W	Programmable Interrupt Controller 2
00A1h	R/W	Programmable Interrupt Controller 2 mask (AT only), where: Bit 7 = 0 Reserved Bit 6 = 0 Enable fixed disk interrupt Bit 5 = 0 Enable coprocessor 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

I/O Address	Read/Write Status	Description
00C0h	R/W	DMA channel 0 (AT only), memory address register
00C2h	R/W	DMA channel 0 (AT only), transfer count register
00C4h	R/W	DMA channel 1 (AT only), memory address register
00C6h	R/W	DMA channel 1 (AT only), transfer count register
00C8h	R/W	DMA channel 2 (AT only), memory address register
00CAh	R/W	DMA channel 2 (AT only), transfer count register
00CCh	R/W	DMA channel 3 (AT only), memory address register
00CEh	R/W	DMA channel 3 (AT only), transfer address register
00D0h	R	DMA status register (AT only), 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
00D2h	W	DMA Write request register (AT only)
00D4h	R/W	DMA mask register (AT only), 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

I/O Address	Read/Write Status	Description
00D6h	R/W	DMA Mode Register (AT only), 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 (AT only)
00DAh	W	DMA Master Clear (AT only)
00DCh	W	DMA Clear Mask Register (AT only)
00DEh	W	DMA Write Mask Register (AT only), 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
00DF-00EFh		Reserved
00F0-00FFh	R/W	Math Coprocessor
0100-016Fh		Reserved
0170h	R/W	Fixed disk 1 data register (AT only)
0171h	R/W	Fixed disk 1 error register (AT only)
0172h	R/W	Fixed disk 1 sector count (AT only)
0173h	R/W	Fixed disk 1 sector number (AT only)
0174h	R/W	Fixed disk 1 cylinder low (AT only)
0175h	R/W	Fixed disk 1 cylinder high (AT only)
0176h	R/W	Fixed disk 1 drive/head register (AT only)
0177h	R/W	Fixed disk 1 status register (AT only)

continued

I/O Address	Read/Write Status	Description
01F0h	R/W	Fixed disk 0 data register (AT only)
01F1h	R/W	Fixed disk 0 error register (AT only)
01F2h	R/W	Fixed disk 0 sector count (AT only)
01F3h	R/W	Fixed disk 0 sector number (AT only)
01F4h	R/W	Fixed disk 0 cylinder low (AT only)
01F5h	R/W	Fixed disk 0 cylinder high (AT only)
01F6h	R/W	Fixed disk 0 drive/head register (AT only)
01F7h	R/W	Fixed disk 0 status register (AT only)
01F9-01FFh	R/W	Reserved
0200-020Fh		Game control port
0201h	R/W	Game port I/O data
0210-0217h	R/W	Expansion Unit (PC and XT only)
0278h	R/W	Parallel 3, data port (AT only)
0279h	R/W	Parallel 3, status port, (AT only), 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 (AT only), where: Bits 7-5 = Reserved 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
02B0-02DFh		Reserved
02E0h		EGA/VGA (alternate)
02E1h		GPIB (Adapter 0)
02E2h		Data Acquisition (Adapter 0)
02E3h		Data Acquisition (Adapter 0)
02E4-02F7h		Reserved

continued

I/O Address	Read/Write Status	Description
02F8h	W	Serial 2, transmitter holding register, which contains the character to be sent. Bit 0, the least significant bit, is sent first. Bits 7-0 = Contains data bits 7-0, respectively, when Divisor Latch Access Bit (DLAB) = 0
02F8h	R	Serial 2, receiver buffer register, which contains the received character. Bit 0, the least significant bit, is sent first. Bits 7-0 = Contains 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. Allows the four controller interrupts to enable the chip interrupt output signal. 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 ID register. Information about a pending interrupt is stored here. When ID register is addressed, the highest priority interrupt is held and no other interrupts are acknowledged until the CPU services that interrupt. Bits 7-3 = 0 Reserved Bits 2-1 = These bits identify the pending interrupt with the highest priority: 11b Receiver Line Status Interrupt; priority = highest. 10b Received Data Available; priority = second 01b Transmitter holding register; priority = third. 00b Modem status interrupt; priority = fourth. Bit 0 = 0 Interrupt pending, and contents of register can be used as a pointer to the appropriate interrupt service routine. = 1 No interrupt pending.

continued

I/O Address	Read/Write Status	Description
02FBh	R/W	Serial 2, 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. Serial output is forced to spacing state and remains there. Bit 5 = Stick Parity Bit 4 = Even Parity Select Bit 3 = Parity Enable = 1 Even number of ones are sent and checked in the data word bits and parity bit. = 0 Odd number of ones are sent and checked. Bit 2 = Number of stop bits per character: = 0 1 Stop bit = 1 0 Stop bits Bits 1-0 = Specify the number of bits per character: 00b 5 bit word length 01b 6 bit word length 10b 7 bit word length 11b 8 bit word length
02FCh	R/W	Serial 2, Modem Control Register, where: Bits 7-5 = 0 Reserved Bit 4 = 1 Loopback Mode for diagnostic testing of serial port. Output of Transmitter Shift register is looped back to Receiver Shift register input. When in this mode, transmitted data is received immediately so that the CPU can verify the transmit data/ receive data serial port paths. 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. The controller is ready to accept a new character to send. Bit 4 = 1 Break Interrupt. The received data input is held in the zero bit state longer than the time of start bit + data bits + parity bit + stop bits. Bit 3 = 1 Framing Error. The stop bit that follows the last parity or data bit is a zero bit. Bit 2 = 1 Parity error. Character has wrong even or odd parity. Bit 1 = 1 Overrun Error. A character was sent to the receiver buffer before the previous character in the buffer could be read. This destroys the previous character. Bit 0 = 1 Data Ready. A complete incoming character has been received and sent to the receiver buffer register.
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, Reserved
0300-031Fh	R/W	Prototype card
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 Bits 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

continued

I/O Address	Read/Write Status	Description
0322h	R	Fixed Disk Adapter Status Register, where: Bits 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 Bits 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
0325h-0347h		Reserved
0348h-0357h		DCA 3278
0360h-36Fh	W	PC Network (PC and XT only)
0372h	R	Diskette controller digital output register, where: Bits 7-6 = 0 Reserved Bit 5 = 1 Enable drive motor 1 Bit 4 = 1 Enable drive motor 0 Bit 3 = 0 Allow interrupts Bit 2 = 0 Controller reset Bit 1 = 0 Reserved Bit 0 = 0 Select drive 0 = 1 Select drive 1

continued

I/O Address	Read/Write Status	Description
0374h	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 Bits 3-2 = Reserved Bit 1 = 1 Drive 1 busy Bit 0 = 0 Drive 0 busy
0375-0376h	R/W	Diskette controller data registers
0377h	R	Diskette controller digital input register where: Bit 7 = Diskette change Bits 6-1 = Reserved Bit 0 = 0 High density select
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 7-5 = Reserved 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
380-38Fh		SDLC and BSC communications
390-393h		Cluster (adapter 0)
3A0-3AFh		BSC communications (primary)
03B0-03B3h*	R/W	Miscellaneous video registers
03B4h*	R/W	MDA, EGA, VGA CRTC index register
03B5h*	R/W	Other MDA, EGA, VGA CRTC registers
03B8h*	R/W	MDA mode control register
* For more information on video I/O ports, refer to the Video I/O Port List at the end of this chapter.		

continued

I/O Address	Read/Write Status	Description
03BAh*	R	VGA input status register 1
03BAh*	W	EGA, VGA feature control register
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 7-5 = Reserved 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
03BFh*	R/W	Hercules configuration switch register, where: Bits 7-2 = Unused Bit 1 = 0 Disables upper 32K of graphics mode buffer = 1 Enables upper 32K of graphics mode buffer at B800:0000h Bit 0 = 0 Prevents graphics mode = 1 Allows graphics mode
03C0h*	R/W	VGA attribute controller index register
03C0h*	W	Other video attribute controller registers
03C1h*	R/W	Other video attribute controller registers
03C2h*	W	EGA, VGA, CGA miscellaneous output register
03C2h*	R	EGA, VGA, CGA input status register 0
03C3h*	R/W	Video subsystem enable
03C4h*	R/W	CGA, EGA, VGA sequencer index register
03C5h*	R/W	Other CGA, EGA, VGA sequencer registers
03C6h*	R	VGA video DAC state register
03C6h*	R/W	VGA video DAC PEL mask
03C7h*	R	VGA video DAC state register
03C8h*	W	VGA video DAC PEL address, read mode

* For more information on video I/O ports, refer to the Video I/O Port List at the end of this chapter.

continued

I/O Address	Read/Write Status	Description
03C8h*	R/W	VGA video DAC PEL address, write mode
03C9h*	R/W	VGA video DAC registers
03CAh*	R/W	CGA/EGA, VGA Graphics 2 position register
03CAh*	R	CGA/EGA, VGA feature control register
03CCh*		VGA miscellaneous output register (R), VGA Graphics 1 position register (R/W), EGA memory mode register (W)
03CDh*	W	VGA feature control register (color)
03CEh*	R/W	VGA graphics controller index register
03CFh*	W	Other VGA graphics controller registers
03D0H*		Video register
03D1h*		Video register
03D2h*		Video register
03D3h*		Video register
03D4h*	R/W	Video CRTC index register
03D5h*	R/W	Other CGA, EGA, VGA registers
03D6h*		Video register
03D7h*		Video register
03D8h*	R/W	CGA, EGA, VGA mode control register
03D9h*	R/W	CGA, EGA, VGA palette register
03DAh*		CGA, EGA, VGA register
03DBh*		Video register
03DCh*		Video register
03DDh*		Video register
03DEh*		Video register
03DFh*		Video register
03D4h*	R	VGA CRT controller index register (color)
03D5h*		Other VGA CRT controller registers (color)
03DAh*	R	VGA input status register 1 (color)
03DAh*	W	VGA feature control register (color)
* For more information on video I/O ports, refer to the Video I/O Port List at the end of this chapter.		

continued

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 = 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: Bits 7-6 = 0 Reserved Bit 5 = Select drive Bit 4 = Write data Bit 3 = Read data Bit 2 = Write enable Bit 1 = 1 Enable drive motor 1 Bit 0 = 1 Enable drive motor 0
03F2h	W	Diskette controller digital output register, where: Bits 7-6 = 0 Reserved Bit 5 = 1 Enable drive motor 1 Bit 4 = 1 Enable drive motor 0 Bit 3 = 0 Allow interrupts Bit 2 = 0 Controller reset Bit 1 = 0 Reserved Bit 0 = 0 Select drive 0 = 1 Select drive 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 Bits 3-2 = Reserved Bit 1 = 1 Drive 1 busy Bit 0 = 0 Drive 0 busy
03F5-03F6h	R/W	Diskette controller data registers

continued

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, which contains the character to be sent. Bit 0, the least significant bit, is sent first. Bits 7-0 = Contains data bits 7-0, respectively, when Divisor Latch Access Bit (DLAB) = 0
03F8h	R	Serial 1, receiver buffer register, which contains the received character. Bit 0, the least significant bit, is sent first. Bits 7-0 = Contains data bits 7-0, respectively, when DLAB = 0
03F8h	R/W	Serial 1, divisor latch, low byte. Both divisor latch registers store the baud rate divisor. 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. Allows the four controller interrupts to enable the chip interrupt output signal. 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
03FAh	R	Serial 1 Interrupt ID register. Information about a pending interrupt is stored here. When ID register is addressed, the highest priority interrupt is held and no other interrupts are acknowledged until the CPU services that interrupt. Bits 7–3 = 0 Reserved Bits 2–1 = These bits identify the pending interrupt with the highest priority: 11b Receiver Line Status Interrupt; priority = highest. 10b Received Data Available; priority = second 01b Transmitter Holding Register; priority = third. 00b Modem Status Interrupt; priority = fourth. Bit 0 = 0 Interrupt pending, and contents of register can be used as a pointer to the appropriate interrupt service routine. = 1 No interrupt pending.
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. Serial output is forced to spacing state and remains there. Bit 5 = Stick Parity Bit 4 = Even Parity Select Bit 3 = Parity Enable = 1 Even number of ones are sent and checked in the data word bits and parity bit. = 0 Odd number of ones are sent and checked. Bit 2 = Specify number of stop bits per character: = 0 1 Stop bit = 1 0 Stop bits Bits 1–0 = Specify the number of bits per character: 00b 5 bit word length 01b 6 bit word length 10b 7 bit word length 11b 8 bit word length

continued

I/O Address	Read/Write Status	Description
03FCh	R/W	Serial 1, Modem Control Register, where: Bits 7-5 = 0 Reserved Bit 4 = 1 Loopback Mode for diagnostic testing of serial port. Output of Transmitter Shift register is looped back to Receiver Shift register input. When in this mode, transmitted data is immediately received so that the CPU can verify the transmit data/receive data serial port paths. 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
03FDh	R	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. The controller is ready to accept a new character to send. Bit 4 = 1 Break Interrupt. The received data input is held in the zero bit state longer than the time of start bit + data bits + parity bit + stop bits. Bit 3 = 1 Framing Error. The stop bit that follows the last parity or data bit is a zero bit. Bit 2 = 1 Parity error. Character has incorrect even or odd parity. Bit 1 = 1 Overrun Error. A character was sent to the receiver buffer before the previous character in the buffer could be read. This destroys the previous character. Bit 0 = 1 Data Ready. A complete incoming character has been received and sent to the receiver buffer register.
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/W	Serial 1, Reserved
06E2-06E3h	R/W	Data Aquisition (Adapter 1)

continued

I/O Address	Read/Write Status	Description
06E4-0790h		Reserved
0790-0793h	R/W	Cluster (Adapter 1)
0794-0AE1h		Reserved
0AE2-0AE3h	R/W	Cluster (Adapter 2)
0AE4-0B8Fh		Reserved
0B90-0B93h	R/W	Cluster (adapter 2)
0B94-138Fh		Reserved
1390-1393h	R/W	Cluster (adapter 3)
1394-238Fh		Reserved
2390-2393h	R/W	Cluster (adapter 4)
2394h-321Fh		Reserved
3220h-3227h	R/W	Serial Port 3 (see description for addresses 03F8h-03FFh for details).
3228h-322Fh	R/W	Serial Port 4 (see description for addresses 03F8h-03FFh for details).
3230h-42E0h		Reserved
42E1h	R/W	GPIB (Adapter 2)
42E2h-62E0h		Reserved
62E1h	R/W	GPIB (Adapter 3)
62E2h-82E0h		Reserved
82E1h	R/W	GPIB (Adapter 4)
82E2h-A2E0h		Reserved
A2E1h	R/W	GPIB (Adapter 5)
A2E2h-C2E0h		Reserved
A2E2h-AFFEh		Reserved
AFFFh	R/W	Plane 0-3 system latch (video register)
B000h-C2E0h		Reserved
C2E1h	R/W	GPIB (Adapter 6)
C2E2h-E2E0h		Reserved
E2E1h	R/W	GPIB (Adapter 7)
E2E2h-FFFFh		Reserved

Video I/O Port List

Introduction

The following pages contain tables that describe all the video I/O ports for each video adapter: MDA, CGA, EGA, and VGA. Because of the complexity of the way the video I/O ports are mapped, they are listed separately from the rest of the hardware I/O ports.

Table: MDA I/O ports

The 6845 CRTC's index register is mapped to I/O port 03B4h. The index value written to port 03B4h controls the register that appears at port 03B5h. The 6845 mode control register is accessed directly via I/O port 03B8h.

I/O Address	Read/Write	Index	Description
03B4h	R/W	—	CRTC Index Register
03B5h	W	01h	Horizontal Total
03B5h	W	02h	Horizontal Display Enable End
03B5h	W	03h	Start Horizontal Blanking
03B5h	W	04h	End Horizontal Blanking
03B5h	W	05h	Start Horizontal Retrace Pulse
03B5h	W	06h	End Horizontal Retrace
03B5h	W	07h	Vertical Total
03B5h	W	08h	Overflow
03B5h	W	09h	Preset Row Scan
03B5h	W	0Ah	Maximum Scan Line
03B5h	W	0Ah	Cursor Start
03B5h	W	0Bh	Cursor End
03B5h	W	0Ch	Start Address High
03B5h	W	0Dh	Start Address Low
03B5h	R/W	0Eh	Cursor Location High
03B5h	R/W	0Fh	Cursor Location Low
03B5h	R/W	10h	Light Pen High
03B5h	R/W	11h	Light Pen Low
03B8h	R/W	—	Mode Control Register

continued

Table: CGA I/O ports

The 6845 CRTC's index register is mapped to I/O port 03D4h. The value written to I/O port 03D4h controls the register that appears at port 03D5h. Mode control and palette registers are accessed via ports 03D8h and 03D9h.

I/O Address	Read/Write	Index	Description
03D4h	R/W	—	CRTC Index Register
03D5h	W	01h	Horizontal Total
03D5h	W	02h	Horizontal Display Enable End
03D5h	W	03h	Start Horizontal Blanking
03D5h	W	04h	End Horizontal Blanking
03D5h	W	05h	Start Horizontal Retrace Pulse
03D5h	W	06h	End Horizontal Retrace
03D5h	W	07h	Vertical Total
03D5h	W	08h	Overflow
03D5h	W	09h	Preset Row Scan
03D5h	W	0Ah	Maximum Scan Line
03D5h	W	0Ah	Cursor Start
03D5h	W	0Bh	Cursor End
03D5h	W	0Ch	Start Address High
03D5h	W	0Dh	Start Address Low
03D5h	R/W	0Eh	Cursor Location High
03D5h	R/W	0Fh	Cursor Location Low
03D5h	R/W	10h	Light Pen High
03D5h	R/W	11h	Light Pen Low
03D8h	R/W	—	Mode Control Register
03D9h	R/W	—	Palette Register

continued

Table: EGA I/O ports

The EGA I/O port address map is listed in the table below.

I/O Address	Read/Write	Index	Description
GENERAL REGISTERS			
03C2h	W	—	Miscellaneous Output
03C2h	R	—	Input Status 0
3?2h*	R	—	Input Status 1
03?Ah/3?Ah*	W	—	Feature Control
SEQUENCER REGISTERS			
03C4h	R/W	—	Sequencer Index Register
03C5h	W	00h	Reset
03C5h	W	01h	Clocking Mode
03C5h	W	02h	Map Mask
03C5h	W	03h	Character Map Select
03CCh	W	04h	Memory Mode
CRT CONTROLLER REGISTERS			
3?4h/3?4h*	R/W	—	CRTC Index Register
3?5h/3?5h*	W	01h	Horizontal Total
3?5h/3?5h*	W	02h	Horizontal Display Enable End
3?5h/3?5h*	W	03h	Start Horizontal Blanking
3?5h/3?5h*	W	04h	End Horizontal Blanking

* The value of ? depends on the value of bit 0 in the Miscellaneous Output Register
? = B in Monochrome Emulation Modes
? = D in Color Emulation Modes

continued

Table: EGA I/O ports, cont'd

I/O Address	Read/Write	Index	Description
CRT CONTROLLER REGISTERS, cont'd			
3?5h/3?5h*	W	05h	Start Horizontal Retrace Pulse
3?5h/3?5h*	W	06h	End Horizontal Retrace
3?5h/3?5h*	W	07h	Vertical Total
3?5h/3?5h*	W	08h	Overflow
3?5h/3?5h*	W	09h	Preset Row Scan
3?5h/3?5h*	W	0Ah	Maximum Scan Line
3?5h/3?5h*	W	0Ah	Cursor Start
3?5h/3?5h*	W	0Bh	Cursor End
3?5h/3?5h*	R/W	0Ch	Start Address High
3?5h/3?5h*	R/W	0Dh	Start Address Low
3?5h/3?5h*	R/W	0Eh	Cursor Location High
3?5h/3?5h*	R/W	0Fh	Cursor Location Low
3?5h/3?5h*	W	10h	Vertical Retrace Start
3?5h/3?5h*	R	10h	Light Pen High
3?5h/3?5h*	W	11h	Vertical Retrace End
3?5h/3?5h*	R	11h	Light Pen Low
3?5h/3?5h*	W	12h	Vertical Display Enable End
3?5h/3?5h*	W	13h	Offset
3?5h/3?5h*	W	14h	Underline Location
3?5h/3?5h*	W	15h	Start Vertical Blank
3?5h/3?5h*	W	16h	End Vertical Blank
3?5h/3?5h*	W	17h	CRTC Mode Control
3?5h/3?5h*	W	18h	Line Compare

* The value of ? depends on the value of bit 0 in the Miscellaneous Output Register
? = B in Monochrome Emulation Modes
? = D in Color Emulation Modes

continued

Table: EGA I/O ports, cont'd

I/O Address	Read/Write	Index	Description
ATTRIBUTE CONTROLLER REGISTERS			
03C0h	W	—	Attribute Controller Index Register
03C0h	W	00h–0Fh	Palette Registers 00h - 0Fh
03C0h	W	10h	Attribute Mode Control Register
03C0h	W	11h	Overscan Color Register
03C0h	W	12h	Color Plane Enable Register
03C0h	W	13h	Horizontal PEL Panning Register
GRAPHICS CONTROLLER REGISTERS			
03CAh	R/W	—	Graphics 2 Position Register
03CCh	R/W	—	Graphics 1 Position Register
03CEh	R/W	—	Graphics Controller Index Register
03CFh	R/W	00h	Set/Reset
03CFh	R/W	01h	Enable Set/Reset
03CFh	R/W	02h	Color Compare
03CFh	R/W	03h	Data Rotate
03CFh	R/W	04h	Read Map Select
03CFh	R/W	05h	Graphics Mode Register
03CFh	R/W	06h	Miscellaneous
03CFh	R/W	07h	Color Don't Care
03CFh	R/W	08h	Bit Mask

continued

Table: VGA I/O ports

The VGA I/O port address map is listed in the table below.

I/O Address	Read/Write	Index	Description
GENERAL REGISTERS			
03CCh	R	—	Miscellaneous Output
03C2h	W	—	Miscellaneous Output
03C2h	R	—	Input Status 0
3?Ah*	R	—	Input Status 1
03CAh	R	—	Feature Control
03?Ah/3?Ah*	W	—	Feature Control
03C3h	R/W	—	VGA Enable
03C7h	R	—	DAC State
SEQUENCER REGISTERS			
03C4h	R/W	—	Sequencer Index Register
03C5h	R/W	00h	Reset
03C5h	R/W	01h	Clocking Mode
03C5h	R/W	02h	Map Mask
03C5h	R/W	03h	Character Map Select
03C5h	R/W	04h	Memory Mode

* The value of ? depends on the value of bit 0 in the Miscellaneous Output Register
? = 03BAh in Monochrome Emulation Modes
? = 03DAh in Color Emulation Modes

continued

Table: VGA I/O ports, cont'd

I/O Address	Read/Write	Index	Description
CRT CONTROLLER REGISTERS			
3?4h/3?4h*	R/W	—	CRTC Index Register
3?5h/3?5h*	R/W	01h	Horizontal Total
3?5h/3?5h*	R/W	02h	Horizontal Display Enable End
3?5h/3?5h*	R/W	03h	Start Horizontal Blanking
3?5h/3?5h*	R/W	04h	End Horizontal Blanking
3?5h/3?5h*	R/W	05h	Start Horizontal Retrace Pulse
3?5h/3?5h*	R/W	06h	End Horizontal Retrace
3?5h/3?5h*	R/W	07h	Vertical Total
3?5h/3?5h*	R/W	08h	Overflow
3?5h/3?5h*	R/W	09h	Preset Row Scan
3?5h/3?5h*	R/W	0Ah	Maximum Scan Line
3?5h/3?5h*	R/W	0Ah	Cursor Start
3?5h/3?5h*	R/W	0Bh	Cursor End
3?5h/3?5h*	R/W	0Ch	Start Address High
3?5h/3?5h*	R/W	0Dh	Start Address Low
3?5h/3?5h*	R/W	0Eh	Cursor Location High
3?5h/3?5h*	R/W	0Fh	Cursor Location Low
3?5h/3?5h*	R/W	10h	Vertical Retrace Start
3?5h/3?5h*	R/W	11h	Vertical Retrace End
3?5h/3?5h*	R/W	12h	Vertical Display Enable End
3?5h/3?5h*	R/W	13h	Offset
3?5h/3?5h*	R/W	14h	Underline Location
3?5h/3?5h*	R/W	15h	Start Vertical Blank
3?5h/3?5h*	R/W	16h	End Vertical Blank
3?5h/3?5h*	R/W	17h	CRTC Mode Control
3?5h/3?5h*	R/W	18h	Line Compare

* The value of ? depends on the value of bit 0 in the Miscellaneous Output Register
? = 03BAh in Monochrome Emulation Modes
? = 03DAh in Color Emulation Modes

continued

Table: VGA I/O ports, cont'd

I/O Address	Read/Write	Index	Description
ATTRIBUTE CONTROLLER REGISTERS			
03C0h	R/W	—	Attribute Controller Index Register
03C1h	R/W	00h-0Fh	Palette Registers 00h - 0Fh
03C1h	R/W	10h	Attribute Mode Control Register
03C1h	R/W	11h	Overscan Color Register
03C1h	R/W	12h	Color Plane Enable Register
03C1h	R/W	13h	Horizontal PEL Panning Register
GRAPHICS CONTROLLER REGISTERS			
03CEh	R/W	—	Graphics Controller Index Register
03CFh	R/W	00h	Set/Reset
03CFh	R/W	01h	Enable Set/Reset
03CFh	R/W	02h	Color Compare
03CFh	R/W	03h	Data Rotate
03CFh	R/W	04h	Read Map Select
03CFh	R/W	05h	Graphics Mode Register
03CFh	R/W	06h	Miscellaneous
03CFh	R/W	07h	Color Don't Care
03CFh	R/W	08h	Bit Mask
AFFFh	R/W	00h	Plane 0 System Latch
AFFFh	R/W	01h	Plane 1 System Latch
AFFFh	R/W	02h	Plane 2 System Latch
AFFFh	R/W	03h	Plane 3 System Latch

continued

Table: VGA I/O ports, cont'd

I/O Address	Read/Write	Index	Description
DIGITAL TO ANALOG CONVERTER (DAC) REGISTERS			
03C8h	R/W	—	Pixel Address (Write Mode)
03C8h	W	—	Pixel Address (Read Mode)
03C6h	R	—	DAC State Register
03C6h	R/W	—	Pixel Mask*
03C9h	R/W	00h	Red value color 0
03C9h	R/W	00h	Green value color 0
03C9h	R/W	00h	Blue value color 0
03C9h	R/W	01h	Red value color 1
⋮	⋮	⋮	⋮
03C9h	R/W	FFh	Red value color FFh
03C9h	R/W	FFh	Green value color FFh
03C9h	R/W	FFh	Blue value color FFh
03C1h	R/W	14h	Color select register
* Applications should not write to this register. Unpredictable results will occur.			

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 ROM BIOS. It is commonly referred to by the acronym POST, for power-on self test and initialization procedure.

IBM BIOS POST and Phoenix BIOS POST

The process followed in performing the power-on self test is similar in every XT or AT BIOS, although the POST process as described in this chapter applies more specifically to Phoenix BIOS products.

continued

How POST is started

The BIOS POST process can be started in any of the following ways:

Start Method	BIOS Behavior
Apply power to system *(i.e. turn system on.)*	Jump to entry point indicated by the processor reset vector (F:0000h). All POST tests and initializations are executed. POST invokes BIOS INT 19h, 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 BIOS INT 19h, Operating System Bootstrap Loader.
Press <Ctrl> <Alt> <Del> *(warm boot)*	INT 09h keyboard hardware interrupt service routine transfers control to BIOS POST. The POST test and initialization of memory above 64K is **not** executed. All other POST tests and initializations are executed. POST invokes BIOS INT 19h, Bootstrap Loader.

Processor reset vector location

The processor reset vector points to address F000:0000h.

In this chapter

This chapter presents the following information:

- POST Procedures
- Rules for Positioning I/O Expansion ROM
- Re-Entering Real Mode (80286-based systems only)
- INT 19h, INT 18h, and System Boot
- POST Error Handling in AT Systems
- POST Error Handling in XT Systems

Introduction

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

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

How POST handles errors

In a Phoenix BIOS, a failure in the central hardware POST test generally results in a fatal system board error, which generates a beep code. A beep code is a series of three beep bursts, separated by short pauses. All AT beep codes are listed in Appendix A. The XT beep code is described in Appendix B.

Nonfatal beep codes may also be sounded. These are listed in Appendix A.

Note: See the POST error handling information in Appendix A.

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),
- the ROM BIOS (checksum),
- the CMOS RAM (AT only),
- the Intel 8237 Direct Memory Access (DMA) Controller,
- the keyboard controller,
- the base 64K System RAM,
- the Programmable Interrupt Controller,
- the Programmable Interrupt Timer, and, if present,
- the cache controller (generally present only on AT systems).

continued

Nonsystem hardware POST tests

Once the central hardware has been tested and initialized, POST verifies that the system configuration data stored in CMOS RAM 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,
- the CRT controllers,
- RAM memory above 64K,
- the keyboard,
- diskette drive A availability,
- the serial interface circuitry,
- the diskette controller,
- the fixed disk controller, and
- any additional hardware.

If an error occurs

A failure in the nonsystem hardware POST test generally results in a screen message. Both IBM and Phoenix AT BIOS messages are listed in Appendix A. XT BIOS messages are listed in Appendix B.

Rules for Positioning I/O Expansion ROM [PC] [XT] [AT]

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 BIOS, VGA video adapter).

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 three predefined address ranges, shown below:

Address Range	Description
C0000h - 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.
C8000h - 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.
E0000h	SYSTEM EXPANSION ROM POST searches for expansion ROM only at E0000h. If no expansion ROM is at E0000h, the search is ended. Any expansion ROM at this address must occupy the entire 64K.

continued

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 (80286-based systems only) [AT]

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 in an 80286-based system, 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

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"

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.

The PC BIOS initializes INT 18h to point to ROM Resident BASIC. However, in an XT BIOS, INT 18 points to a routine that displays a message indicating that there is no boot device available, and that the boot process has failed.

By default, the AT BIOS initializes INT 18h to point to a routine that displays the message NO BOOT DEVICE AVAILABLE, and transfers control to INT 18h if the INT 19h process fails.

POST Error Handling in AT Systems [AT]

Introduction

For AT and AT-compatible systems, POST reports test or initialization failures in one of the following ways:

- Beep codes (only Phoenix BIOS products)
- System failure messages
- Boot failure messages
- Informational messages

Reference

See Appendix A for a list of messages that can be generated by an AT BIOS. Separate lists are provided of Phoenix and IBM AT BIOS messages.

See Appendix B for a list of messages that can be generated by an XT BIOS. Separate lists are provided of Phoenix and IBM XT BIOS messages.

POST Error Handling in XT Systems [XT]

Introduction

For XT and XT-compatible systems, POST reports test or initialization failures in one of the following ways:

- Beep code
- System failure messages
- Boot failure messages
- Informational messages

Reference

See Appendix B for a more complete list of all messages that can be generated by a BIOS in an XT or XT-compatible system.

7

Chapter 7
INT 02h Nonmaskable Interrupt

Overview

Description

When the system hardware activates the Nonmaskable Interrupt (NMI) pin on the microprocessor, the BIOS INT 02h Nonmaskable Interrupt Service ISR is invoked. A nonmaskable interrupt is a hardware interrupt that cannot be masked. An NMI indicates a serious condition that usually cannot be shut off or disabled by any process. The microprocessor does not generate an interrupt acknowledge cycle when it receives an NMI (as it usually does when it receives an interrupt), since the NMI is handled by a software ISR.

If an NMI is received while a previous NMI is being serviced, the second NMI is not serviced until the first has completed.

Disabling NMIs

In AT systems, NMIs can be disabled by setting bit 7 of the CMOS RAM address register port, 0070h. This practice is not recommended.

continued

In this chapter

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

The following topics are presented:

- XT NMI Handling
- 8087 Coprocessor Exceptions
- AT NMI Handling
- 80x87 Coprocessor Exceptions in AT Systems

How to determine level of support

Throughout this book, the commonly accepted level of support for each ROM BIOS function is identified by the legend [PC], [XT], and/or [AT] on the same line as the function title.

Legend	Description
[PC]	Function is supported by original IBM PC BIOS only.
[XT]	Function is supported by all IBM PC/XT and compatible ROM BIOSs. **Note:** This level of support is contained in the ROM BIOS implemented in the majority of IBM PC and PC/XT compatible systems.
[AT]	Function is supported by all IBM PC/AT and compatible ROM BIOSs.

XT BIOS Nonmaskable Interrupt ISR

When the system hardware activates the NMI pin on the 8086 or 8088 microprocessor, the XT BIOS INT 02h Nonmaskable Interrupt Service ISR is invoked.

The XT BIOS Nonmaskable Interrupt Service determines the reason for the NMI and takes appropriate action. In general, this involves:

- displaying a run-time error message (shown below), and
- halting the processor.

NMIs can only be cleared by a system reboot. The BIOS supports NMI clearing by warm (Ctrl-Alt-Del) reboot or cold reboot (system reset button or power switch).

INT 02h vector

The XT BIOS initializes the INT 02h vector to F000:E2C3h. The INT 02h vector resides at address 00:08h in the BIOS interrupt vector table.

XT BIOS run-time error messages

The following messages may accompany an NMI in an XT system. Possible causes and solutions are listed. These messages apply only to systems with a Phoenix BIOS. Other BIOSs may generate different messages and may not let the user recover from the NMI without rebooting the system.

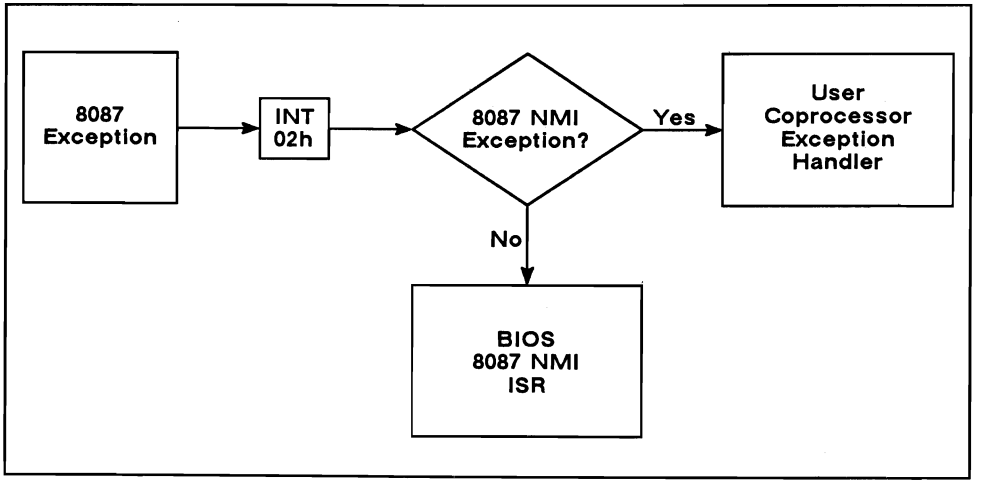
Message	Possible Cause	Solution
8087 NMI at *address*. Type (S)hut off NMI, (R)eboot, other keys to continue	An NMI occurred on access to the math coprocessor.	Type (S)hut off NMI. Temporarily continues processing, but the coprocessor must be replaced.
Memory parity NMI at *address*. Type (S)hut off NMI, (R)eboot, other keys to continue	A memory chip(s) has failed.	Type (S)hut off NMI. Temporarily continues processing. The bad memory chip(s) must be replaced.
I/O card NMI at *address*. Type (S)hut off NMI, (R)eboot, other keys to continue	An NMI occurred on access to an adapter card.	Turn the machine off and then on again. If that doesn't work, replace the adapter card.

Description

Systems based on the Intel 8086 and 8088 that incorporate the 8087 math coprocessor indicate math coprocessor exceptions by generating an NMI (i.e., INT 02h).

A user 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. The graphic below illustrates the process that the caller's 8087 NMI handling routine must follow.



AT BIOS Nonmaskable Interrupt Service (ISR)

The AT BIOS Nonmaskable Interrupt Service determines the reason for the NMI and takes appropriate action. In general, this involves displaying a run-time error message, and halting the processor.

NMIs can only be cleared by a system reboot. The AT BIOS supports NMI clearing by warm (Ctrl-Alt-Del) reboot or cold reboot (system reset button or power switch).

INT 02h vector

The AT BIOS initializes the INT 02h vector to F000:E2C3h. The INT 02h vector resides at address 00:08h in the BIOS interrupt vector table.

Handing NMI parity errors

Parity errors in system memory cause system hardware to activate the 80286, 80386SX, or 80386 processor's NMI pin.

There are two types of memory parity errors:

- system board memory failure, and
- I/O board memory failure.

continued

AT BIOS parity error messages

The following messages may accompany an NMI in an AT system. Possible causes and solutions are listed. These messages apply only to systems with a Phoenix BIOS. Other BIOSs may generate different messages and may not let the user recover from the NMI without rebooting the system.

Message	Possible Cause	Suggested Solution
Memory Parity Error at *xxxxx* (R)	Memory failed. If the BIOS 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, the message will read **"Memory Parity Error ????"**.	Type (S)hut off NMI to temporarily continue processing. Test the system board and all installed devices for proper operation. Replace if necessary.
I/O Card Parity Error at *xxxx* (R)	A peripheral card failed. If the BIOS NMI handler can determine the address of the failed card, it will be displayed in place of *xxxxx*. If the failed card is not found, then the message will read **"I/O Card Parity Error ????"**.	Type (S)hut off NMI to temporarily continue processing. Test the system board and all installed devices for proper operation. Replace if necessary.

DMA bus time-out errors

Time-out errors cause the system hardware to set the 80286 or 80386 processor's NMI pin. Direct Memory Access (DMA) bus time-out errors cause NMI time-out errors.

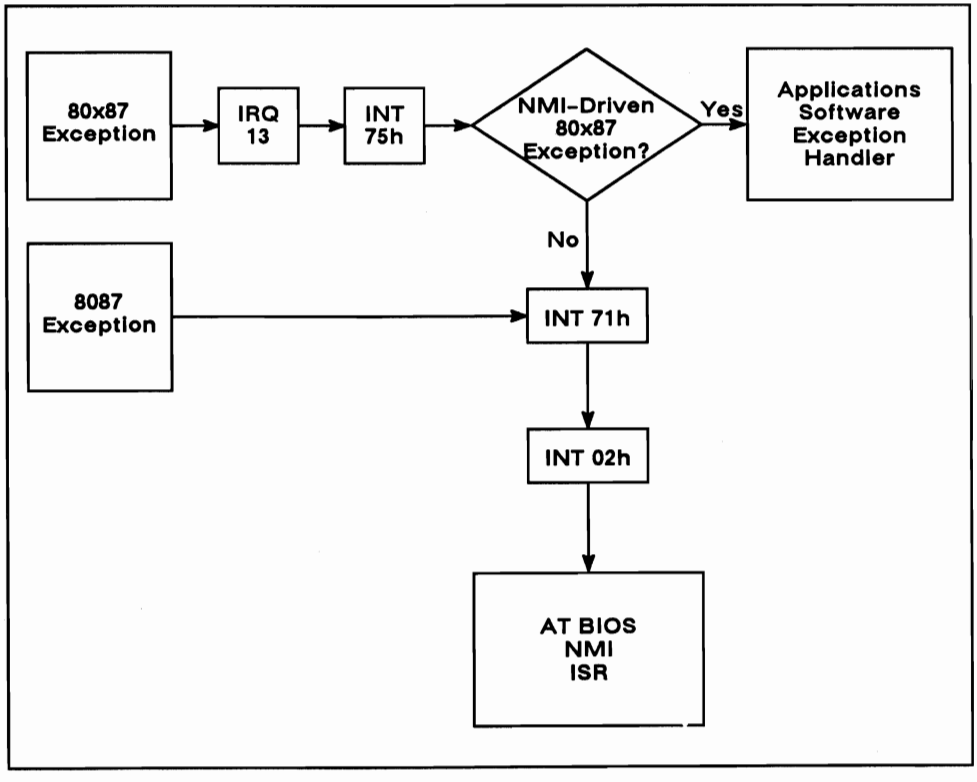
When a DMA driven device uses the bus longer that 7.8 microseconds, an NMI and the following error message is generated.

Message	Possible Cause	Suggested Solution
DMA Bus Time-out	A device has driven the -BURST signal line for more than 7.8 microseconds, causing the DMA Controller to generate a bus time-out.	Test the system board and all installed devices for proper operation. Replace if necessary.

80287/80387 coprocessor exceptions

Systems based on the 80286 or 80386 microprocessor that incorporate the 80287 or 80387 math coprocessor indicate math coprocessor exceptions by setting IRQ 13. IRQ 13 invokes INT 75h, the AT BIOS Coprocessor Exception Handling routine.

Software written to the 80287 or 80387 math coprocessor should handle coprocessor exceptions by intercepting INT 75h. The caller's coprocessor exception routine should determine if the NMI was generated by the 80287/80387 by following the process that is depicted in the graphic below. If the NMI was not coprocessor-related, the BIOS provides an INT 71h handler which redirects coprocessor exceptions to INT 02h.



continued

How AT BIOS maintains compatibility

Unless it is revectored by a user program, the AT BIOS INT 75h Coprocessor Exception ISR is initialized to the same entry point as INT 02h. This ensures that 80x87 software running under AT BIOS is given the opportunity to handle coprocessor exceptions by intercepting INT 02h as it does in XT systems.

The AT BIOS 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 AT BIOS interrupt vector table.

XT compatibility

An AT-compatible system revectors any expansion bus hardware interrupt that generates an INT 02h in 8088-based systems to INT 71h.

In an AT, hardware IRQ 9 generates an INT 71h. The AT-compatible BIOS provides an interrupt handler which redirects this interrupt to INT 0Ah, providing compatibility with programs that expect the INT 0Ah interrupt handler in response to a hardware interrupt (8088-based systems).

8

Chapter 8
INT 09h and INT 16h Keyboard Services

Overview

Introduction

The ROM BIOS contains two keyboard-related services, a Keyboard ISR and a Keyboard DSR:

- **Keyboard ISR**
  The BIOS Keyboard Interrupt Service routine (ISR) is invoked via hardware INT 09h each time a key is pressed. The BIOS Keyboard ISR processes keyboard data and converts it into information that is useful to the system. In most cases, this conversion results in the Keyboard ISR placing a two byte character code into a keystroke buffer located at 40:1Eh in low memory.

- **Keyboard DSR**
  The BIOS Keyboard Device Service routine (DSR) is invoked via software INT 16h. The BIOS Keyboard DSR provides an interface through which the operating system or application software can interface with the keystroke buffer. The Keyboard DSR contains functions that read the BIOS keyboard buffer, write to the buffer, return the status of the buffer, and so on.

continued

Vector values

Each time the system is booted, the ROM BIOS vectors INT 09h to F000:E987h, the Keyboard ISR entry point. In addition, it vectors INT 16h to F000:E82Eh, the Keyboard DSR entry point.

In this chapter

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

- Theory of Operation
- Keyboard I/O Ports
- System RAM Data
- INT 09h: Internal Function Requests
- INT 16h: Keyboard DSR
- How to Read Character Code Tables
- Character Codes: AH = 00h/01h
- Character Codes: AH = 10h/11h

How to determine level of support

Throughout this book, the commonly accepted level of support for each ROM BIOS function is identified by the legend [PC], [XT], and/or [AT] on the same line as the function title.

Legend	Description
[PC]	Function is supported by original IBM PC BIOS only.
[XT]	Function is supported by all IBM PC/XT and compatible ROM BIOSs. **Note:** This level of support is contained in the ROM BIOS implemented in the majority of IBM PC and PC/XT compatible systems.
[AT]	Function is supported by all IBM PC/AT and compatible ROM BIOSs.

Theory of Operation

Introduction

IBM PC/XT, PC/AT, and compatible keyboards add flexibility to IBM compatible PCs by tying the interpretation of keystrokes to system software, rather than directly to keyboard hardware.

Raw key data originates in hardware

Each time a key is pressed or released, the IBM PC keyboard hardware generates a one byte "make code" or "break code."

The keyboard generates a key's make code each time that key is pressed, and it generates the key's break code each time the key is released. The byte associated with a key's break code is identical the one associated with its make code except that bit 7 is set. That is, the numeric value its break code is 80h higher than the key's make code.

Make and break code conversion

In and of itself, a key's make or break code has no direct meaning to an operating system or applications software. The keyboard make and break codes generated by a U.S keyboard, for example, are identical to the keyboard codes generated by a German, French, or Italian keyboard.

Keyboard codes are converted from meaningless numbers into ASCII characters or other meaningful data through a series of transformations controlled by system software. The number of transformations involved depends directly on the kind of PC in use.

PC, PC/XT and the 8255A-5 controller chip

In IBM PC, PC/XT, and compatible systems, keyboard make and break codes are converted from serial to parallel form by an Intel 8255A-5 (or compatible) keyboard controller chip. The 8255A-5 chip then generates an interrupt request (IRQ1) which invokes the ROM BIOS INT 09h Keyboard Interrupt Service Routine (ISR).

continued

PC/AT and the 8042 controller chip

In IBM PC/AT and compatible systems, keyboard make and break codes are converted from serial to parallel form by an Intel 8042 (or compatible) keyboard controller chip.

Like its PC/XT counterpart, the 8042 chip generates IRQ1 which invokes the ROM BIOS INT 09h Keyboard Interrupt Service Routine. Because of hardware differences between PC and PC/AT keyboards, however, firmware resident in the 8042 controller must perform the additional step of converting AT key codes into System Scan Codes, values that are backwards compatible with those generated by PC/XT keyboards.

ROM BIOS INT 09h processing

The ROM BIOS Keyboard ISR analyzes each key code as follows:

- Codes generated by keyboard shift or toggle keys cause the Keyboard ISR to update the keyboard shift key and toggle key flags
- Shift key combinations, such as Ctrl–Alt–Del, Ctrl–Num Lock, and Ctrl–Scroll Lock, are converted into requests for keyboard ISR internal functions, such as Reset, Pause, and Break.
- Codes that correspond to ASCII or special key values (e.g., function or edit keys) are converted into two byte character codes and are placed in a 16 word keystroke buffer. Operating systems or application software can then access the keystroke via the ROM BIOS INT16h Keyboard Device Service Routine (DSR).

Character code format

The character codes generated by the INT 09h Keyboard ISR are made up of a low order byte, called the main byte, and of a high order byte, called the auxiliary byte. The ASCII value of a keystroke, if any, is always contained in the main byte; the auxiliary byte contains the keystroke's make code or system scan code. When a key, such as a function key, edit key, or function key combination, does not have an ASCII value, the Keyboard ISR sets the main byte to 00h and the auxiliary byte to a special key value.

Reference: A complete list of all character codes is found at the end of this chapter.

continued

Typematic Rate and Delay

A key's break code indicates that the key has been released. The way the INT 09h ISR handles a key's break code depends on which key is pressed.

When a key is held down continuously, the keyboard hardware delays for a brief period then sends repeat values of the key's make code to the keyboard controller chip. Upon receipt of each make code, the controller chip does its processing and generates an IRQ1 which invokes INT 09h, the keyboard ISR.

If the make code corresponds to an alphanumeric, arrow, backspace, enter, or tab key, the keyboard ISR sends repeat character codes to the keystroke buffer until it receives the key's break code. If the make code corresponds to a special key, such as a function key, or a shift/toggle key, then the keyboard ISR ignores subsequent make codes from the keyboard, toggling the status of the key when it receives the key's break code.

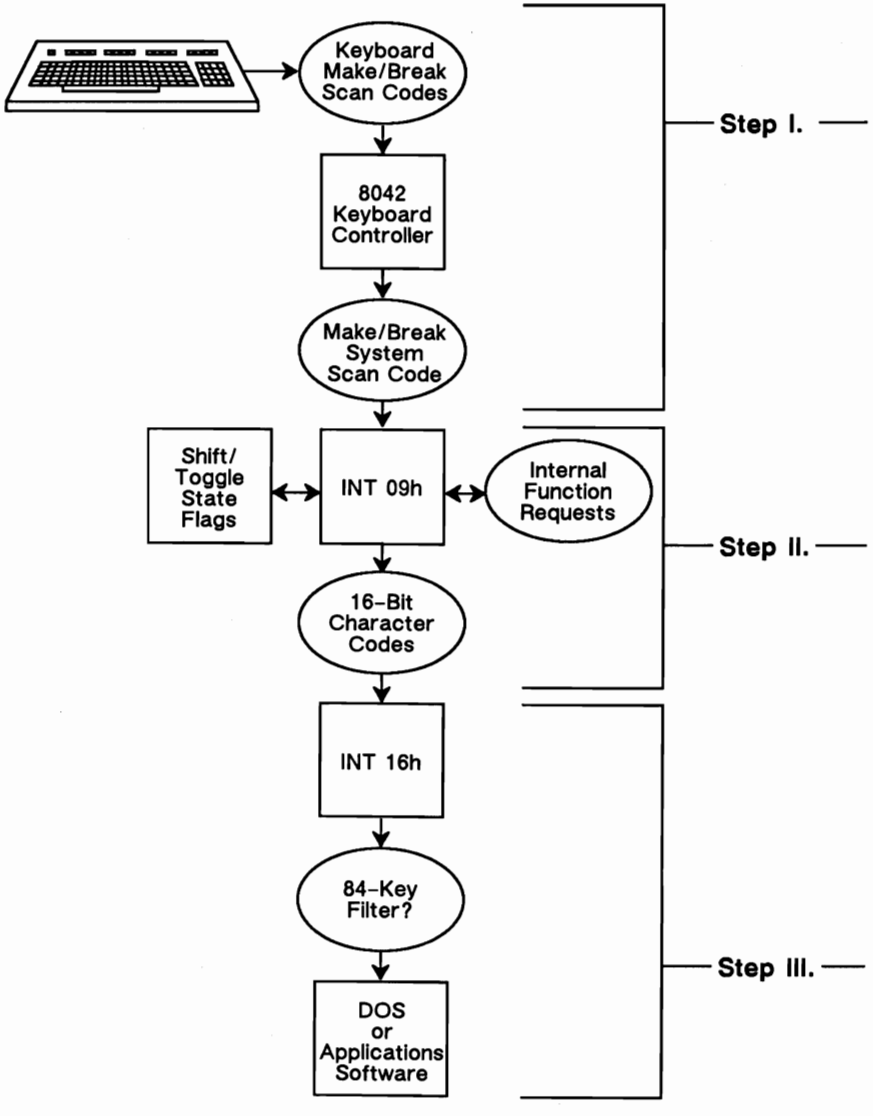
INT 09h, DOS, and foreign keyboard support

The BIOS Keyboard ISR can only convert key codes into their American English ASCII values. When a foreign keyboard is in place, the MS-DOS command **KEYBxx** must be invoked. ("**xx**" must be a supported two-character country code) The **KEYBxx** command replaces the ROM BIOS INT 09h Keyboard ISR with one which uses foreign keyboard translation tables. For more information on foreign keyboards, refer to your PC-DOS or MS-DOS reference guide.

continued

Keystroke data flow in PC/AT systems

Using the PC/AT model, the graphic below traces the path of a keystroke from keyboard to application program.



Step I.

The 8042 Keyboard Controller converts keyboard make or break codes from serial to parallel form, then it translates them into System Scan Codes, values that are backwards compatible with PC/XT keyboard hardware. In PC/AT and compatible systems, 8042 System Scan Codes are the only codes the INT 09h Keyboard ISR can interpret. In PC/XT systems, conversion of key codes into System Scan Codes is unnecessary; keyboard make or break codes are passed directly through to the INT 09h Keyboard ISR.

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 that 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 the 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 reads the Character Code from the keyboard buffer. If desired, 101/102-key keyboard character codes can be filtered for compatibility with application software that accepts only those character codes generated by 83- and 84-key keyboards.

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

I/O Address	Read/Write Status	Description
0060h	W	Keyboard data input buffer (AT-only)
0060h	R	Keyboard data output buffer (AT-only)
0062h	W	8042 keyboard input buffer (AT-only)
0061h	R/W	8255 output register (XT-only), where: Bit 7 = 1 Clear keyboard Bit 6 = 1 Hold keyboard clock low Bit 5 = 1 Enable I/O check Bit 4 = 1 Enable RAM parity check Bit 3 = 1 Read low switches Bit 2 = 0 Reserved Bit 1 = 1 Speaker data enable Bit 0 = 1 Enable timer 2 gate to speaker
0062h	R/W	8255 input register (XT-only), where: Bit 7 = 1 RAM parity check Bit 6 = 1 I/O channel check Bit 5 = 1 Timer channel 2 out Bit 4 = Reserved Bit 3 = 1 System board RAM size type 1 Bit 2 = 1 System board RAM size type 0 Bit 1 = 1 Coprocessor installed Bit 0 = 1 Loop in POST
0063h	R/W	8255 Command mode register (XT-only), where: Bits 7-6 = Number of diskette drives, where: 00b 1 drive 01b 2 drives 10b 3 drives 11b 4 drives Bits 5-4 = Type of display at power-on, where: 00b Reserved 01b 40x25 color (mono mode) 10b 80x25 color (mono mode) 11b MDA (80x25) Bits 3-2 = Memory on system board (256K chips), where: 00b 256K 01b 512K 10b 576K 11b 640K Bits 3-2 = Memory on system board (64K chips), where: 00b 64K 01b 128K 10b 192K 11b 256K Bits 1-0 = Reserved

continued

Keyboard I/O Ports, Continued

I/O Address	Read/Write Status	Description
0064h	W	8042 keyboard input buffer (AT-only)
0064h	W	8042 Commands (AT-only)
0064h	R	8042 Status (AT-only), 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

System RAM Data

Introduction

Throughout the course of operation, the BIOS 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	Extended 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 BIOS 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 15h Function: AH = 4Fh Keyboard Intercept [XT] [AT]

The INT 09h ISR calls INT 15h AH = 4Fh, the Keyboard Intercept function, each time a key is pressed. The Keyboard Intercept function can be used to create alternate keyboard layouts and/or to cause the system to ignore certain keystrokes.

The Keyboard Intercept function is supported only by later versions of the XT and AT BIOS. To determine if your BIOS supports this function, invoke INT 15h AH = C0h, Return System Configuration Parameters. If the Keyboard Intercept function is supported, bit 4 of Feature Information Byte 1 (40:05h) will be set to 1.

Input: AH = 4Fh
AL = Scan code input by INT 09h ISR

Output: AL = Scan code
CF = 0 Keystroke unchanged (original scan code in AL)
= 1 Keystroke changed (new scan code in AL)

Reference: See Chapter 13 for more information on INT 15h functions 4Fh and C0h.

INT 09h: System Reset (<Ctrl> <Alt> <Del>) [XT] [AT]

Description

The BIOS 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)
XT Make Code or
AT 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>) [XT] [AT]

Description

The BIOS 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)
XT Make Code or
AT 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>) [XT] [AT]

Description

The BIOS 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)
XT Make Code or
AT 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 either

- the make System Scan Code for any non-shift key except the SYS REQ key, or
- 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 Keyboard 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 keyboard 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. (LED updating is supported on AT-compatible keyboards only.)
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>) [XT] [AT]

Description

The BIOS INT 09h ISR invokes INT 05h (the BIOS 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 by pressing

- the Print Screen key alone, or
- 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>) [AT]

Description

On AT-compatible systems, the BIOS 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.

Note: The Sys Req is not available on 83-key PC/XT and compatible keyboards.

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 DSR

Introduction

The BIOS Keyboard DSR is 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	Platforms
00h	Read Keyboard Input	[XT] [AT]
01h	Return Keyboard Status	[XT] [AT]
02h	Return Shift Flag Status	[XT] [AT]
03h	Set Typematic Rate and Delay	[XT] [AT]
04h	Reserved	Reserved
05h	Store Key Data	[XT]* [AT]
06h-0Fh	Reserved	Reserved
10h	Read Extended Keyboard Input	[XT]* [AT]
11h	Return Extended Keyboard Status	[XT]* [AT]
12h	Return Extended Shift Flags Status	[XT]* [AT]
13h-FFh	Reserved	Reserved
* Functions 05h and 10h - 12h are not supported on all XT BIOSs. See *Determining level of support* below.		

Determining level of support

Functions 05h and 10h - 12h are not available on all XT BIOSs. To determine if your BIOS supports functions 05h and 10h - 12h;

- Use function 05h to write a value of FFFFh to the keyboard buffer. If function 05h returns with AL= 00h, the value was written and function 05h is supported. (See Function: AH = 05h Store Key Data in this chapter.)
- Use function 10h to read the keyboard buffer. If FFFFh is not returned in AX, retry at least 15 times (i.e. the buffer holds 16 characters). Functions 10h - 12h are supported only if FFFFh is returned to AX.

INT 16h: AH = 00h Read Keyboard Input [XT] [AT]

Description

If a character code is available in the keyboard buffer, the Read Keyboard Input function reads the character code, removes it from the keyboard buffer, and returns its value in AX. The way this function processes when no character code is available depends on the kind of PC in place.

PC/XT processing

In PC/XT and compatible systems, if the keyboard buffer is empty, this function suspends program operation until a character code is inserted into the buffer. Users who wish to avoid suspending the system are advised to use function AH = 01h Return Keyboard Status to test for the presence of a character code.

AT processing

In PC/AT and compatible systems, if the keyboard buffer is empty, this function invokes INT 15h function AH = 90h (Device Busy) with AL = 02h Keyboard. This allows the operating system to perform another task while the keyboard loop is occurring.

Once the INT 09h Keyboard ISR places a character code in the keyboard buffer, it invokes INT 15h function AH = 91h (Interrupt Complete) with AL = 02h Keyboard. The Read Keyboard Input function then reads the character code, removes it from the buffer, and returns to the caller with the character code in AX. In addition, if the keyboard LEDs do not match the current flag settings, function AH = 00h also updates the keyboard LEDs.

Input/Output

Input: AH = 00h

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

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

continued

Extended key code "filtering"

The new and duplicate keys (i.e. the "extended keys") present on 101/102-key PC/AT compatible keyboards generate System Scan Codes that are not useful to applications software written to interact exclusively with the traditional 83-key PC/XT or 84-key PC/AT keyboards.

The Read Keyboard Input function acts as an 83-key/84-key keyboard filter, selectively editing the keyboard buffer contents for 83-key/84-key keyboard compatibility. When a 101-key keyboard is in place, INT 16h function AH = 00h:

- returns all standard 83-key/84-key keyboard Character Codes as is,
- adjusts 101-key keyboard duplicate keyboard characters so that they take on the same two-byte key code as their 83-key/84-key counterparts, and
- destroys any 101-key Character Codes not compatible with the 83-key/84-key keyboards.

The "84th" key

"Sys Req," the 84th key on PC/AT compatible keyboards, is not present on PC/XT 83-key keyboards. The absence of this 84th key is transparent to the Read Keyboard Input function; the System Scan Code generated by the Sys Req key is trapped by INT 09h and does not result in character code being placed in the keyboard buffer. For a discussion of INT 09h Sys Req key processing see "INT 09h: System Request <SysReq>" in this chapter.

Extended key passthrough

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 [XT] [AT]

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.
- In PC/AT systems, 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

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

Extended key code "filtering"

The new and duplicate keys (i.e. the "extended keys") present on 101/102-key PC/AT compatible keyboards generate System Scan Codes that are not useful to applications software written to interact exclusively with the traditional 83-key PC/XT or 84-key PC/AT keyboards.

The Read Keyboard Input function acts as an 83-key/84-key keyboard filter, selectively editing the keyboard buffer contents for 83-key/84-key keyboard compatibility.

Note: A complete discussion of extended key code filtering is found under the heading "INT 16h: AH = 00h Read Keyboard Input" on the previous page.

---

continued

INT 16h: AH = 01h Return Keyboard Status, Continued [XT] [AT]

Extended key passthrough

Function AH = 11h Return Extended Keyboard Status 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 [XT] [AT]

Description

The Return Shift Flag Status function returns the current shift status from the Keyboard Shift Flags byte (40:17h).

Note: In PC/AT and compatible systems, 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 16h: AH = 03h Set Typematic Rate and Delay [AT]

Description

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

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 value (in milliseconds)

00h = 250	03h = 1000
01h = 500	04h to FFh — Reserved
02h = 750	

BL = Typematic rate (in characters per second)

00h = 30.0	0Bh = 10.9	16h = 4.3
01h = 26.7	0Ch = 10.0	17h = 4.0
02h = 24.0	0Dh = 9.2	18h = 3.7
03h = 21.8	0Eh = 8.6	19h = 3.3
04h = 20.0	0Fh = 8.0	1Ah = 3.0
05h = 18.5	10h = 7.5	1Bh = 2.7
06h = 17.1	11h = 6.7	1Ch = 2.5
07h = 16.0	12h = 6.0	1Dh = 2.3
08h = 15.0	13h = 5.5	1Eh = 2.1
09h = 13.3	14h = 5.0	1Fh = 2.0
0Ah = 12.0	15h = 4.6	20h to FFh — Reserved

Output: None

INT 16h: AH = 04h Reserved

INT 16h: AH = 05h Store Key Data [XT] [AT]

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

XT support: This function is only supported by later version XT BIOSs. See *Determining level of support* under the heading INT 16h: Keyboard DSR.

Input/Output

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

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

INT 16h: AH = 06h - 0Fh Reserved

INT 16h: AH = 10h Read Extended Keyboard Input [XT] [AT]

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.

XT support: This function is only supported by later version XT BIOSs. See *Determining level of support* under the heading INT 16h: Keyboard DSR.

Input/Output

Input: AH = 10h

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

Reference: 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.

Extended key "filtering"

Function AH = 00h Read Keyboard Input selectively filters keyboard buffer input for compatibility with PC/XT 83-key and PC/AT 84-key keyboards. For details, see the Function: AH = 00h heading in this chapter.

INT 16h: AH = 11h Return Extended Keyboard Status [XT] [AT]

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.

XT support: This function is only supported by later version XT BIOSs. See *Determining level of support* under the heading INT 16h: Keyboard DSR.

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

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

---

continued

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.

Extended key "filtering"

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 = 12h Return Extended Shift Flags Status [XT] [AT]

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.

XT support: This function is only supported by later version XT BIOSs. See *Determining level of support* under the heading INT 16h: Keyboard DSR.

continued

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 = 13h - FFh Reserved

How to Read Character Code Tables

Introduction

The tables on the following pages present data related to the IBM PC/XT/AT and compatible keyboards.

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

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

How to read the tables

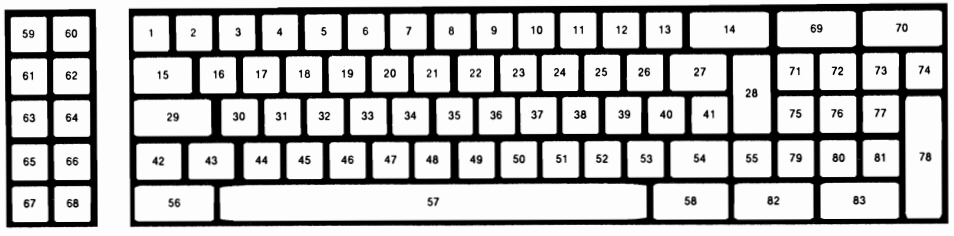
- The data presented in each table is listed by key number and U. S. keyboard key legend. Readers who are using foreign keyboards should refer to the keyboard layout figures on the next page to see how their keyboard legend maps into key numbers.
- The columns labeled "Scan Codes" list the codes offered to the INT 09h ISR by the keyboard controller chip. In PC/AT compatible systems these codes are generated by the 8042 controller chip and are more commonly referred to as System Scan Codes. Only Make Code values are listed. Break codes are found by adding a value of 80h to the make code in question. Scan Codes in E0-xxh form correspond to the 101/102-key keyboard extended keys. These keys are not present on 83- and 84-key keyboards.
- 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

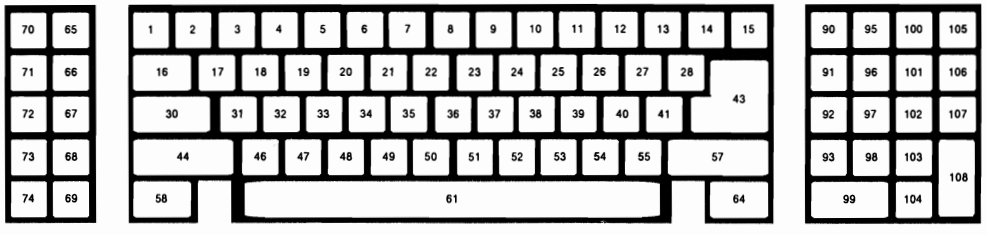
83-key, 84-key, and 101/102-key keyboard numbers

The key number systems applied to 83-key PC-compatible, 84-key AT-compatible, and 101/102-key AT-compatible keyboards are shown below.

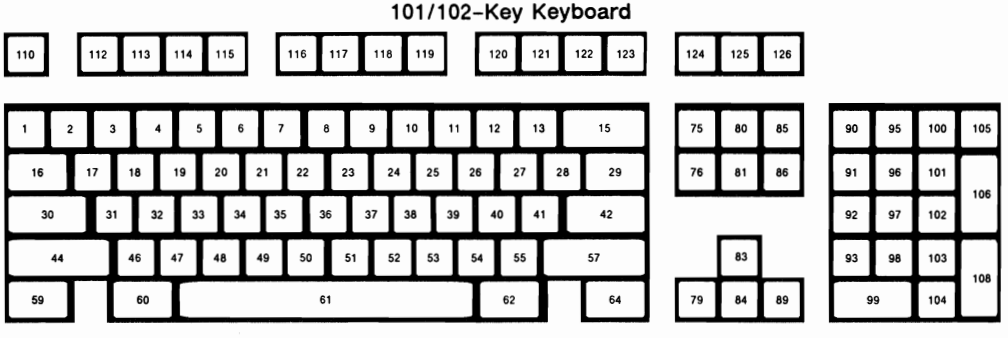
83-Key Keyboard



84-Key Keyboard



101/102-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.

101/102 Key #	83 Key #	84 Key #	U.S. Keyboard Legend	Scan Codes (hex)	Character Codes AH/AL (hex)			
					Normal	Shifted	Control	Alt
1	41	1	`~	29	29/60	29/7E		
2	2	2	1!	02	02/31	02/21		78/00
3	3	3	2@	03	03/32	03/40	03/00	79/00
4	4	4	3#	04	04/33	04/23		7A/00
5	5	5	4$	05	05/34	05/24		7B/00
6	6	6	5%	06	06/35	06/25		7C/00
7	7	7	6^	07	07/36	07/5E	07/1E	7D/00
8	8	8	7&	08	08/37	08/26		7E/00
9	9	9	8*	09	09/38	09/2A		7F/00
10	10	10	9(	0A	0A/39	0A/28		80/00
11	11	11	0)	0B	0B/30	0B/29		81/00
12	12	12	-_	0C	0C/2D	0C/5F	0C/1F	82/00
13	13	13	=+	0D	0D/3D	0D/2B		83/00
15	14	15	Backspace	0E	0E/08	0E/08	0E/7F	
16	15	16	Tab	0F	0F/09	0F/00		

continued

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

101/102 Key #	83 Key #	84 Key #	U.S. Keyboard Legend	Scan Codes (hex)	Character Codes AH/AL (hex)			
					Normal	Shifted	Control	Alt
17	16	17	Q	10	10/71	10/51	10/11	10/00
18	17	18	W	11	11/77	11/57	11/17	11/00
19	18	19	E	12	12/65	12/45	12/05	12/00
20	19	20	R	13	13/72	13/52	13/12	13/00
21	20	21	T	14	14/74	14/54	14/14	14/00
22	21	22	Y	15	15/79	15/59	15/19	15/00
23	22	23	U	16	16/75	16/55	16/15	16/00
24	23	24	I	17	17/69	17/49	17/09	17/00
25	24	25	O	18	18/6F	18/4F	18/0F	18/00
26	25	26	P	19	19/70	19/50	19/10	19/00
27	26	27	[{	1A	1A/5B	1A/7B	1A/1B	
28	27	28	] }	1B	1B/5D	1B/7D	1B/1D	
29*	43	14	¦ \	2B	2B/5C	2B/7C	2B/1C	
30	58	64	Caps Lock	3A				
31	30	31	A	1E	1E/61	1E/41	1E/01	1E/00
32	31	32	S	1F	1F/73	1F/53	1F/13	1F/00
33	32	33	D	20	20/64	20/44	20/04	20/00
34	33	34	F	21	21/66	21/46	21/06	21/00
35	34	35	G	22	22/67	22/47	22/07	22/00
36	35	36	H	23	23/68	23/48	23/08	23/00
37	36	37	J	24	24/6A	24/4A	24/0A	24/00
38	37	38	K	25	25/6B	25/4B	25/0B	25/00
39	38	39	L	26	26/6C	26/4C	26/0C	26/00
40	39	40	; :	27	27/3B	27/3A		
41	40	41	' "	28	28/27	28/22		
42			(102-key only)	2B	2B/5C	2B/7C	2B/1C	
43	28	43	Enter↵	1C	1C/0D	1C/0D	1C/0A	

* 101-key keyboard only.

continued

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

101/102 Key #	83 Key #	84 Key #	U.S. Keyboard Legend	Scan Codes (hex)	Character Codes AH/AL (hex)			
					Normal	Shifted	Control	Alt
44	42	44	L Shift	2A				
45			(102-Key only)	56	56/5C	56/7C		
46	44	46	Z	2C	2C/7A	2C/5A	2C/1A	2C/00
47	45	47	X	2D	2D/78	2D/58	2D/18	2D/00
48	46	48	C	2E	2E/63	2E/43	2E/03	2E/00
49	47	49	V	2F	2F/76	2F/56	2F/16	2F/00
50	48	50	B	30	30/62	30/42	30/02	30/00
51	49	51	N	31	31/6E	31/4E	31/0E	31/00
52	50	52	M	32	32/6D	32/4D	32/0D	32/00
53	51	53	, <	33	33/2C	33/3C		
54	52	54	. >	34	34/2E	34/3E		
55	53	55	/ ?	35	35/2F	35/3F		
57	54	57	R Shift	36				
58	29	30	L Ctrl	1D				
60	56	58	L Alt	38				
61	57	61	Space	39	39/20	39/20	39/20	39/20
62			R Alt	E0-38				
64			R Ctrl	E0-1D				
75			Insert	E0-52	52/00	52/00		
76			Delete	E0-53	53/00	53/00		
79			Left	E0-4B	4B/00	4B/00	73/00	
80			Home	E0-47	47/00	47/00	77/00	
81			End	E0-4F	4F/00	4F/00	75/00	
83			Up	E0-48	48/00	48/00		
84			Down	E0-50	50/00	50/00		
85			Page Up	E0-49	49/00	49/00	84/00	
86			Page Down	E0-51	51/00	51/00	76/00	

continued

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

101/102 Key #	83 Key #	84 Key #	U.S. Keyboard Legend	Scan Codes (hex)	Character Codes AH/AL (hex)			
					Normal	Shifted	Control	Alt
89			Right	E0-4D	4D/00	4D/00	74/00	
90	69	95	Num Lock	45				
91	71	91	7 Home	47	47/00	47/37	77/00	**
92	75	92	4 Left	4B	4B/00	4B/34	73/00	**
93	79	93	1 End	4F	4F/00	4F/31	75/00	**
95			/	E0-35	35/2F	35/2F		
96	72	96	8 UP	48	48/00	48/38		**
97	76	97	5	4C	4C/00	4C/35		**
98	80	98	2 Down	50	50/00	50/32		**
99	82	99	0 Ins	52	52/00	52/30		
100	55	106	*	37	37/2A	37/2A		
101	73	101	9 PgUp	49	49/00	49/39	84/00	**
102	77	102	6 Right	4D	4D/00	4D/36	74/00	**
103	81	103	3 PgDn	51	51/00	51/33	76/00	**
104	83	104	Del	53	53/00	53/2E		
105	74	107	-	4A	4A/2D	4A/2D		
106	78	108	+	4E	4E/2B	4E/2B		
108			Enter	E0-1C	1C/0D	1C/0D	1C/0A	
110	1	90	Esc	01	01/1B	01/1B	01/1B	
112	59	70	F1	3B	3B/00	54/00	5E/00	68/00
113	60	65	F2	3C	3C/00	55/00	5F/00	69/00
114	61	71	F3	3D	3D/00	56/00	60/00	6A/00
115	62	66	F4	3E	3E/00	57/00	61/00	6B/00
116	63	72	F5	3F	3F/00	58/00	62/00	6C/00
117	64	67	F6	40	40/00	59/00	63/00	6D/00
118	65	73	F7	41	41/00	5A/00	64/00	6E/00
119	66	68	F8	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

101/102 Key #	83 Key #	84 Key #	U.S. Keyboard Legend	Scan Codes (hex)	Character Codes AH/AL (hex)			
					Normal	Shifted	Control	Alt
120	67	74	F9	43	43/00	5C/00	66/00	70/00
121	68	69	F10	44	44/00	5D/00	67/00	71/00
122			F11	57				
123			F12	58				
		105	SysReq					
124			Print Screen	E0-2A/ E0-37 E0-37			72/00	
125	70	100	Scroll Lock	46				
126			Pause	E1-1D/ 5-E1/ 9D/C5			00/00	

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.

101/102 Key #	U.S. Keyboard Legend	Scan Codes (hex)	Character Codes AH/AL (hex)			
			Normal	Shifted	Control	Alt
1	`~	29	29/60	29/7E		29/00
2	1!	02	02/31	02/21		78/00
3	2@	03	03/32	03/40	03/00	79/00
4	3#	04	04/33	04/23		7A/00
5	4$	05	05/34	05/24		7B/00
6	5%	06	06/35	06/25		7C/00
7	6^	07	07/36	07/5E	07/1E	7D/00
8	7&	08	08/37	08/26		7E/00
9	8*	09	09/38	09/2A		7F/00
10	9(	0A	0A/39	0A/28		80/00
11	0)	0B	0B/30	0B/29		81/00
12	-_	0C	0C/2D	0C/5F	0C/1F	82/00
13	=+	0D	0D/3D	0D/2B		83/00
15	Backspace	0E	0E/08	0E/08	0E/7F	0E/00
16	Tab	0F	0F/09	0F/00	94/00	A5/00

continued

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

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

continued

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

101/102 Key #	U.S. Keyboard Legend	Scan Codes (hex)	Character Codes AH/AL (hex)			
			Normal	Shifted	Control	Alt
44	L Shift	2A				
45	(102-Key only)	56	56/5C	56/7C		
46	Z	2C	2C/7A	2C/5A	2C/1A	2C/00
47	X	2D	2D/78	2D/58	2D/18	2D/00
48	C	2E	2E/63	2E/43	2E/03	2E/00
49	V	2F	2F/76	2F/56	2F/16	2F/00
50	B	30	30/62	30/42	30/02	30/00
51	N	31	31/6E	31/4E	31/0E	31/00
52	M	32	32/6D	32/4D	32/0D	32/00
53	, <	33	33/2C	33/3C		33/00
54	. >	34	34/2E	34/3E		34/00
55	/ ?	35	35/2F	35/3F		35/00
57	R Shift	36				
58	L Ctrl	1D				
60	L Alt	38				
61	Space	39	39/20	39/20	39/20	39/20
62	R Alt	E0-38				
64	R Ctrl	E0-1D				
75	Insert	E0-52	52/E0	52/E0	92/E0	A2/00
76	Delete	E0-53	53/E0	53/E0	93/E0	A3/00
79	Left	E0-4B	4B/E0	4B/E0	73/E0	9B/00
80	Home	E0-47	47/E0	47/E0	77/E0	97/00
81	End	E0-4F	4F/E0	4F/E0	75/E0	9F/00
83	Up	E0-48	48/E0	48/E0	8D/E0	98/00
84	Down	E0-50	50/E0	50/E0	91/E0	A0/00
85	Page Up	E0-49	49/E0	49/E0	84/E0	99/00
86	Page Down	E0-51	51/E0	51/E0	76/E0	A1/00

continued

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

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

101/102 Key #	U.S. Keyboard Legend	Scan Codes (hex)	Character Codes AH/AL (hex)			
			Normal	Shifted	Control	Alt
120	F9	43	43/00	5C/00	66/00	70/00
121	F10	44	44/00	5D/00	67/00	71/00
122	F11	57	85/00	87/00	89/00	8B/00
123	F12	58	86/00	88/00	8A/00	8C/00
124	Print Screen	E0-2A/ E0-37 E0-37			72/00	
125	Scroll Lock	46				
126	Pause	E1-1D/ 45-E1/ 9D/C5			00/00	

9

Chapter 9
INT 10h Video Service

Overview

Description

In general, all IBM PC/XT/AT and compatible video BIOSs contain a group of functions that allow the user to select video display mode, cursor addressing, text display, scrolling, and writing pixels. However, the physical location and number of functions supported by a PC/XT/AT or compatible video BIOS depends on the kind of adapter in place.

MDA and CGA adapters rely exclusively on the Video Service contained in the system board ROM BIOS of all PC/XT/AT and compatible computers. EGA and VGA adapters contain a dedicated ROM BIOS installed on the adapter hardware itself.

How the functions are invoked

Be they in system or adapter ROM, all video BIOS functions are invoked by INT 10h. The INT 10h vector resides at interrupt table address 00:40h and is initialized to absolute address FF065h.

continued

Determining video BIOS support **[MDA] [CGA] [EGA] [VGA]**

Since the introduction of the IBM PC in 1981, four video adapter types have emerged: the MDA, CGA, EGA, and the VGA. The number and sophistication of the functions supported by the video BIOSs associated with each adapter varies; the more recent the adapter, the greater the number of functions its BIOS supports.

In order to identify which function is supported by which adapter, we have adopted the legend above. When a function applies exclusively to the VGA adapter, for example, its description will bear the legend [VGA]. When the function applies to all adapter types, it will bear the legend [MDA] [CGA] [EGA] [VGA].

The table below describes the meaning of each legend.

Legend	Description
[MDA]	Function is supported by IBM Monochrome Display Adapters (MDA), and compatibles.
[CGA]	Function is supported by IBM Color Graphics Adapter (CGA), and compatibles.
[EGA]	Function is supported by IBM Enhanced Graphics Adapter (EGA), and compatibles.
[VGA]	Function is supported by IBM Video Graphics Array (VGA) and compatible adapters.

Note: The Hercules display adapter allows for the display of 720x350 monochrome graphics on standard TTL monochrome monitors. However, neither the system ROM BIOS nor the Hercules adapter itself contain a BIOS. Applications programs that support Hercules graphics must contain an application-specific device driver.

continued

In this chapter

This chapter defines the features and functions of all PC-compatible video and ROM BIOSs. The following topics are presented:

- Theory of Operation
- MDA Hardware Environment
- CGA Hardware Environment
- EGA Hardware Environment
- VGA Hardware Environment
- Video Modes
- System RAM Data
- Summary of INT 10h Functions
- How Functions Are Called
- Error Handling
- Video BIOS Function Descriptions

Theory of Operation

Introduction

A computer monitor is a specialized device designed to display digital data written into a video buffer by the computer's microprocessor. The signals driving the monitor originate in the computer's video subsystem. Computer monitors in themselves are not programmable, but the video subsystems controlling them are. The BIOS Video Service offers operating systems and applications programs a set of functions that eliminate the need to directly program video subsystem hardware.

How information is "painted" on the screen

The front surface on a computer monitor — the computer screen — is actually the end a large Cathode Ray Tube (CRT). The rearmost part of the CRT contains a cathode that, when heated to approximately 600 degrees Centigrade, emits a high voltage, narrowly focused, electron beam. When the electron beam strikes the phosphor coated inner surface of the computer screen, it causes an area of the phosphor to glow. When the voltage of the beam is modulated, so too is the intensity of the CRT's glow.

The computer's video subsystem generates signals that cause the CRT's electron beam to sweep across the screen in a left to right, top to bottom pattern of horizontal lines called a raster. The information written by the microprocessor to the video buffer is represented on the CRT screen as a pattern of illuminated dots, called pixels (picture elements). As the beam sweeps across the screen, the video subsystem turns the electron beam on and off, thus "painting" the buffer information onto the screen.

continued

Display resolution

A monitor's horizontal scan frequency determines the maximum number of lines the monitor is capable of painting per second. Its vertical scan frequency determines the frames per second the monitor is capable of refreshing.

- **Vertical resolution**
  A display's vertical resolution — the maximum number of scan lines it can display on the screen — is calculated by dividing horizontal scan frequency by vertical scan frequency.

  IBM PC (and compatible) monochrome displays, for example, have a horizontal scan frequency of 18.432 KHz and a vertical scan frequency of 50Hz. The maximum number of lines displayed on the screen is calculated as follows:

$$\frac{\text{18,432 KHz (lines per second)}}{\text{50 Hz (frames per second)}} = \text{368 lines per frame}$$

  To allow for vertical centering of the display image, approximately 5 percent of the maximum displayable lines must be subtracted from the total. Thus maximum vertical resolution is 350 lines.

- **Horizontal resolution**
  The dots, or pixels, that appear on each horizontal scan line are generated by the video subsystem controlling the monitor. Physically, the dots are generated by a crystal-driven device, called a dot clock. A display's horizontal resolution — the maximum number of dots per line — is calculated by dividing the monitor's horizontal scan frequency into the dot rate of the subsystem driving the monitor.

  The Monochrome Display Adapter, or MDA, (about which we will learn more later) has a dot clock rate of 16.257 MHz, or 16,257,000 dots per second. When driving an IBM-compatible monochrome display, maximum dots per line are calculated as follows:

$$\frac{\text{16.257 MHz (16,257,000 dots per second)}}{\text{18.432 KHz (18,432 lines per second)}} = \text{882 pixels per line}$$

  To allow for horizontal centering of the displayed image, 10 to 15 percent of the maximum displayable pixels must be subtracted from this total. Thus maximum horizontal resolution is 720 pixels.

continued

Monitor types

Computer monitors come in two varieties: color and monochrome.

- **Color monitors**
  Color computer monitors contain three electron guns: one red, one blue, and one green. The special phosphors coating the CRT surface glow accordingly: when struck by a red beam they glow red; when struck by a blue beam, they glow blue, and so on. Although each gun illuminates a separate dot, the dots are so tightly focused that they appear to the eye as one color.

- **Monochrome monitors**
  Monochrome monitors have only one electron gun. The color of the screen is a function of the phosphor used. In monochrome computer displays, common phosphor colors are amber, green, or white.

Signal types

Computer monitors are further distinguished from each other by the kind of input signal they are designed to accept:

- **Composite**
  Composite monitors accept a composite, analog signal — like the signal used to drive a television set. In a composite color signal, the red, green, and blue (and other control signals) are combined by the video subsystem and separated by the monitor. Composite monochrome signals are the same except that they contain only one "color" signal. Composite signals can be sent over a single line, but the difficulty involved in accurately separating colors limits the resolution achieved on composite monitors.

continued

Signal types, cont'd

- **Digital RGB and digital monochrome**
  Digital RGB monitors accept signals for red, green, or blue on separate lines. The monitor does not perform color signal separation. The number of colors displayed by a digital RGB monitor is a function of the number of color lines. A 64-color RGB monitor, such as the IBM Enhanced Color Display, is driven by two lines for each primary color. Since each pair of lines can combine into four possible intensities, 64 colors are possible (i.e., 4 x 4 x 4 = 64). A digital monochrome signal is identical to its RGB counterpart except that only one "color" line is required.

- **Analog RGB and monochrome**
  Analog RGB monitors accept signals for red, green, and blue on separate lines. The number of colors potentially displayable, however, is infinite, since the intensity of each line varies with the voltage applied to it. Video subsystems designed to drive analog color monitors must convert digital color information into analog voltages.

  The practical limit on the number of displayable colors is determined by the number of digital bits dedicated to each color line. A video subsystem that devotes 6 bits per color would be capable of generating signals for 262,144 different colors (i.e. $2^{(3*6)}$ = 262,144). Monochrome analog signals are identical to color analog signals save that they transmit only one "color" signal. A video subsystem that devotes six bits per "color" would be capable of generating 64 shades of that color (i.e., 2^6 = 64).

continued

Adapter types

IBM PC/XTs, ATs, and compatible systems are not manufactured with the video subsystem integrated in the motherboard; video hardware, instead, must be configured into the system bus on a video adapter board.

The table below describes monitors and resolutions supported by the major PC video adapters.

Adapter Type	Year Intro.	Dot Clock	Monitor Support	Horiz. Freq.	Vert. Freq.	Resolution Color
MDA	1981	16.257 MHz	Digital Mono	18.43 KHz	50 Hz	720x350
HGC	1982	16.257MHz	Digital Mono	18.43 KHz	50 Hz	720x350
CGA	1981	14.318 MHz	Composite Color Digital RGB*	15.75 KHz	60 Hz	640x200
EGA	1985	16.257 MHz 14.318 MHz 16.257 MHz	Digital RGB** Digital RGB* Digital Mono	21.85 KHz 15.75 KHz 18.43 KHz	60 Hz 60 Hz 50 Hz	640x350 640x200 720x350
VGA	1987	25.175 MHz 28.175 MHz 25.175 MHz 25.175 MHz	Analog*** Analog*** Analog*** Analog***	31.5 KHz 31.5 KHz 31.5 KHz 31.5 KHz	70 Hz 70 Hz 60 Hz 70 Hz	640x400 720x400 640x480 640x350

Legend:

MDA = Monochrome Display Adapter
HGC = Hercules Graphics Card
CGA = Color Graphics Adapter
EGA = Enhanced Graphics Adapter
VGA = Video Graphics Array

* IBM Color Display or compatible
** IBM Enhanced Color Display or compatible
*** IBM VGA Analog or compatible

continued

Video data format

Each location in the video adapter's display buffer maps to a location on the display screen. The first byte of buffer memory maps to the top, leftmost point on the screen. As memory addresses increase, the screen location mapped moves from left to right and top to bottom. The microprocessor writes information to the video hardware's data buffer in either of two formats: alphanumeric or graphic.

- **Alphanumeric format**
  When writing alphanumeric data to the video buffer, the microprocessor writes each symbol (e.g., letter, number, punctuation mark, etc.) as a series of two bytes. The first byte contains the ASCII code for the symbol, and the second byte contains the color attributes to be applied to the symbol.

  Alphanumeric characters are painted on the screen in a matrix of dots, called a character cell. The video subsystem's control circuitry reads one display line of ASCII data from the video buffer and converts it into the necessary series of dots. Simultaneously, the video hardware translates the ASCII character's accompanying attribute into the indicated foreground/background colors.

- **Graphic format**
  When writing graphical data to the video buffer, the microprocessor writes the individual color value of each pixel to be painted on the CRT. In order to do this, the microprocessor must address the display buffer as a memory map made up of a series of bits. When displaying data in any of 16 colors, for example, the microprocessor writes four bits per pixel (i.e., 2^4 =16). The video subsystem's control circuitry reads each pixel location from the buffer and generates the color and control signals necessary to write the dot to the correct screen location.

continued

Adapters, modes, and BIOSs

A video mode is characterized by:

- screen resolution,
- number of displayable colors, and the
- kind of data (alphanumeric or graphic) displayed.

Most PC video adapters support more than one video mode. The reason for this is largely historical: PC video adapters have evolved from low resolution, 16-color devices to relatively high resolution, 256K-color devices. In order to maintain compatibility with software written for the previous video adapters, each successive generation of video hardware had to be made mode-compatible with its predecessors. Thus the video modes supported by CGA adapters (1981) are also supported by the more current VGA adapters (1987).

Establishing a video mode is a matter of programming those components of the video adapter's hardware that control the adapter's dot clock, enable or disable the adapter hardware generation of alphanumeric dot patterns, and so on. Although it is possible to program these components directly, setting video modes is more easily accomplished through the video BIOS.

The physical location and number of functions supported by a video BIOS depends on the kind of adapter in place. MDA and CGA adapters rely exclusively on the Video Service contained in the system board ROM BIOS of all PC/XT/AT and compatible computers. EGA and VGA adapters contain a dedicated ROM BIOS installed on the adapter hardware itself.

continued

Compatibility and portability

In addition to simplifying mode sets, the video BIOS contains a group of other functions that further insulate applications software from direct manipulation of video hardware. For example, the Video Service contains functions that allow users to write a dot on the screen, write a string of ASCII characters to a specified screen location, to control the location of the cursor, and so on.

Application software can be made to execute faster if it writes directly to the video hardware. In fact, this is a common strategy used by the authors of games and other software. However, writing applications directly to the video hardware ties the application to the hardware, for there is never a guarantee that the next generation video adapter will be 100% hardware compatible with its predecessors. The BIOS that interfaces with each generation of hardware generally contains a superset of the functions supported by the previous version. Writing software to an adapter's video BIOS instead of directly to its hardware helps insure the application's longevity and portability.

Adapter hardware components

When viewed on the highest functional level, all PC video adapters share the same programmable components:

- **Video Buffer**
  An adapter's video buffer is a block of RAM that exists on the adapter but is mapped into the microprocessor's address space. Each location in the video buffer maps to a location on the video screen. The video buffer address map can range from segment B000:0000h – B000:BFFFh for MDA adapters to A000:0000h – B000:FFFFh for VGA adapters. (See Video Modes in this chapter.)

- **Cathode Ray Tube Controller (CRTC)**
  The CRTC generates the horizontal and vertical timing signals that control the path and duration of the electron beam sweeping the inner surface of the CRT. The CRTC also increments a video buffer address counter so that the data contained in the buffer will be synchronized with the timing signals. The CRTC controls the size/location of the text mode cursor, and selects the portion of the video buffer to be displayed on the screen.

continued

Adapter hardware components, cont'd

- **Hardware Alphanumeric Character Generator**
  In alphanumeric modes, characters are written to the video buffer as pairs of bytes. The first byte represents the character's ASCII value; the second byte represents the character's attribute (i.e. foreground/background color, blinking, intensity). The hardware character generator is responsible for translating the ASCII value of a character into a matrix of dots to display on the screen. To do this, the hardware character generator must reference a character table under the control of the video BIOS. Depending on adapter type, this character table can resides permanently at a fixed ROM location, or it can be written (on mode set) from ROM to a fixed location in RAM.

- **Attribute Decoder**
  An adapter's attribute decoder component uses data from the video buffer to control the color and brightness of the signals produced by the video signal generator.

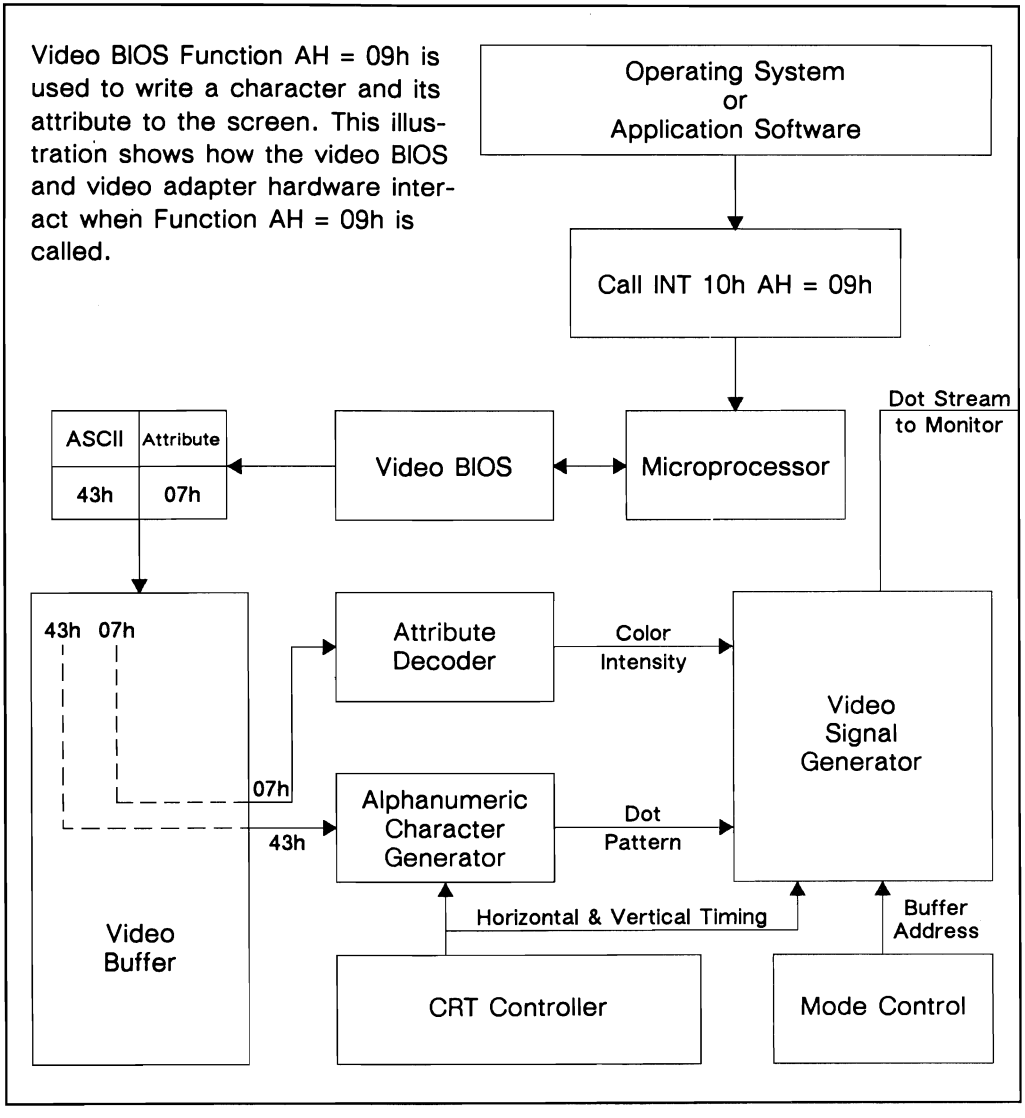
- **Video Signal Generator**
  An adapter's video signal generator component outputs the signals that control what appears on the monitor's screen. As we have seen there are three kinds of signals: composite, digital, and analog.

- **Mode Control Registers**
  Video mode control registers put the video adapter hardware in a state that will allow it to produce the indicated mode. For example, in graphics modes, the character generator must be disabled. The adapter's character clock, which determines the number of dots that are put up for each ASCII character in alphanumeric modes can vary from mode to mode. Characters can be 9 dots wide in some modes and 8 dots wide in others.

continued

Putting a character on the screen

Video BIOS Function AH = 09h is used to write a character and its attribute to the screen. This illustration shows how the video BIOS and video adapter hardware interact when Function AH = 09h is called.



MDA Hardware Environment

Introduction

The MDA BIOS resides in system ROM and not on the MDA adapter card. The MDA BIOS assumes the following hardware environment:

- Motorola 6845 (or compatible) CRTC
- Monochrome digital signal monitor
- 4K of video RAM

Motorola 6845 CRTC and the MDA

When implemented on a MDA adapter, the Motorola 6845 CRTC contains 20 programmable, 8-bit internal registers. Nineteen of these registers control horizontal and vertical timings. The remaining register, the mode control register, is used in setting the MDA video mode.

continued

MDA I/O ports

The 6845 CRTC's index register is mapped to I/O port 03B4h. The index value written to I/O port 03B4h controls which register will appear at I/O port 03B5h. The 6845 mode control register is accessed directly via I/O port 03B8h.

I/O Address	Read/Write	Index	Description
03B4h	R/W	—	CRTC Index Register
03B5h	W	01h	Horizontal Total
03B5h	W	02h	Horizontal Display Enable End
03B5h	W	03h	Start Horizontal Blanking
03B5h	W	04h	End Horizontal Blanking
03B5h	W	05h	Start Horizontal Retrace Pulse
03B5h	W	06h	End Horizontal Retrace
03B5h	W	07h	Vertical Total
03B5h	W	08h	Overflow
03B5h	W	09h	Preset Row Scan
03B5h	W	0Ah	Maximum Scan Line
03B5h	W	0Ah	Cursor Start
03B5h	W	0Bh	Cursor End
03B5h	W	0Ch	Start Address High
03B5h	W	0Dh	Start Address Low
03B5h	R/W	0Eh	Cursor Location High
03B5h	R/W	0Fh	Cursor Location Low
03B5h	R/W	10h	Light Pen High
03B5h	R/W	11h	Light Pen Low
03B8h	R/W	—	Mode Control Register

For more information

A full discussion of MDA hardware is beyond the scope of this book. For more information, see the documentation accompanying your MDA adapter or one of the excellent references listed at the end of this book.

CGA Hardware Environment

Introduction

The CGA BIOS resides in system ROM and not on the CGA adapter card. The CGA BIOS assumes the following hardware environment:

- Motorola 6845 (or compatible) CRTC
- Composite or Digital RGB monitor
- 16K of video RAM

Motorola 6845 CRTC and the CGA

When implemented on a CGA adapter, the Motorola 6845 CRTC contains 21 programmable, 8-bit internal registers. Nineteen of these registers control horizontal and vertical timings. The remaining two registers, the mode control and palette registers, are used to set video modes and program the CGA palette.

continued

CGA I/O ports

The 6845 CRTC's index register is mapped to I/O port 03D4h. The value written to I/O port 03D4h controls the register that appears at port 03D5h. Mode control and palette registers are accessed via ports 03D8h and 03D9h.

I/O Address	Read/Write	Index	Description
03D4h	R/W	—	CRTC Index Register
03D5h	W	01h	Horizontal Total
03D5h	W	02h	Horizontal Display Enable End
03D5h	W	03h	Start Horizontal Blanking
03D5h	W	04h	End Horizontal Blanking
03D5h	W	05h	Start Horizontal Retrace Pulse
03D5h	W	06h	End Horizontal Retrace
03D5h	W	07h	Vertical Total
03D5h	W	08h	Overflow
03D5h	W	09h	Preset Row Scan
03D5h	W	0Ah	Maximum Scan Line
03D5h	W	0Ah	Cursor Start
03D5h	W	0Bh	Cursor End
03D5h	W	0Ch	Start Address High
03D5h	W	0Dh	Start Address Low
03D5h	R/W	0Eh	Cursor Location High
03D5h	R/W	0Fh	Cursor Location Low
03D5h	R/W	10h	Light Pen High
03D5h	R/W	11h	Light Pen Low
03D8h	R/W	—	Mode Control Register
03D9h	R/W	—	Palette Register

For more information

A full discussion of CGA hardware is beyond the scope of this book. For more information, see your CGA adapter documentation.

EGA Hardware Environment

Introduction

The EGA BIOS resides in EGA adapter cards, completely superseding the system ROM video BIOS. The EGA BIOS assumes the following hardware environment:

- IBM EGA or compatible chip
- 64K of video RAM
- Digital RGB or digital monochrome monitor

EGA proprietary LSI chip

EGA adapters are controlled by an IBM (or compatible) LSI CRTC chip.

EGA I/O port addressing

The IBM EGA (or compatible) chip is a Large Scale Integration (LSI) device that incorporates the functionality of four general purpose adapter control registers, a CRTC, an Attribute Controller, an Address Sequencer, and a Graphics Controller onto one chip.

The limited I/O address space of PC/XT/AT and compatible computers, makes it impossible to assign each register its own I/O port. With the exception of the general registers, each major EGA register group is addressed via an index/access port pair. For example, the EGA Sequencer's index register is mapped to I/O port 03C4h. The value (index) written to port 03C4h determines which register appears at I/O port 03C5h.

continued

EGA I/O ports

The EGA I/O port address map is listed in the table below.

I/O Address	Read/Write	Index	Description
GENERAL REGISTERS			
03C2h	W	—	Miscellaneous Output
03C2h	R	—	Input Status 0
3?2h*	R	—	Input Status 1
03?Ah/3?Ah*	W	—	Feature Control
SEQUENCER REGISTERS			
03C4h	R/W	—	Sequencer Index Register
03C5h	W	00h	Reset
03C5h	W	01h	Clocking Mode
03C5h	W	02h	Map Mask
03C5h	W	03h	Character Map Select
03CCh	W	04h	Memory Mode
CRT CONTROLLER REGISTERS			
3?4h/3?4h*	R/W	—	CRTC Index Register
3?5h/3?5h*	W	01h	Horizontal Total
3?5h/3?5h*	W	02h	Horizontal Display Enable End
3?5h/3?5h*	W	03h	Start Horizontal Blanking
3?5h/3?5h*	W	04h	End Horizontal Blanking

* The value of ? depends on the value of bit 0 in the Miscellaneous Output Register
? = B in Monochrome Emulation Modes
? = D in Color Emulation Modes

continued

EGA I/O ports, cont'd

I/O Address	Read/Write	Index	Description
CRT CONTROLLER REGISTERS, cont'd			
3?5h/3?5h*	W	05h	Start Horizontal Retrace Pulse
3?5h/3?5h*	W	06h	End Horizontal Retrace
3?5h/3?5h*	W	07h	Vertical Total
3?5h/3?5h*	W	08h	Overflow
3?5h/3?5h*	W	09h	Preset Row Scan
3?5h/3?5h*	W	0Ah	Maximum Scan Line
3?5h/3?5h*	W	0Ah	Cursor Start
3?5h/3?5h*	W	0Bh	Cursor End
3?5h/3?5h*	R/W	0Ch	Start Address High
3?5h/3?5h*	R/W	0Dh	Start Address Low
3?5h/3?5h*	R/W	0Eh	Cursor Location High
3?5h/3?5h*	R/W	0Fh	Cursor Location Low
3?5h/3?5h*	W	10h	Vertical Retrace Start
3?5h/3?5h*	R	10h	Light Pen High
3?5h/3?5h*	W	11h	Vertical Retrace End
3?5h/3?5h*	R	11h	Light Pen Low
3?5h/3?5h*	W	12h	Vertical Display Enable End
3?5h/3?5h*	W	13h	Offset
3?5h/3?5h*	W	14h	Underline Location
3?5h/3?5h*	W	15h	Start Vertical Blank
3?5h/3?5h*	W	16h	End Vertical Blank
3?5h/3?5h*	W	17h	CRTC Mode Control
3?5h/3?5h*	W	18h	Line Compare

* The value of ? depends on the value of bit 0 in the Miscellaneous Output Register
? = B in Monochrome Emulation Modes
? = D in Color Emulation Modes

continued

EGA I/O ports, cont'd

I/O Address	Read/Write	Index	Description
ATTRIBUTE CONTROLLER REGISTERS			
03C0h	W	—	Attribute Controller Index Register
03C0h	W	00h-0Fh	Palette Registers 00h - 0Fh
03C0h	W	10h	Attribute Mode Control Register
03C0h	W	11h	Overscan Color Register
03C0h	W	12h	Color Plane Enable Register
03C0h	W	13h	Horizontal PEL Panning Register
GRAPHICS CONTROLLER REGISTERS			
03CAh	R/W	—	Graphics 2 Position Register
03CCh	R/W	—	Graphics 1 Position Register
03CEh	R/W	—	Graphics Controller Index Register
03CFh	R/W	00h	Set/Reset
03CFh	R/W	01h	Enable Set/Reset
03CFh	R/W	02h	Color Compare
03CFh	R/W	03h	Data Rotate
03CFh	R/W	04h	Read Map Select
03CFh	R/W	05h	Graphics Mode Register
03CFh	R/W	06h	Miscellaneous
03CFh	R/W	07h	Color Don't Care
03CFh	R/W	08h	Bit Mask

For more information

A full discussion of EGA hardware is beyond the scope of this book. For more information, see the documentation accompanying your EGA adapter or one of the excellent references listed at the end of this book.

VGA Hardware Environment

Introduction

The VGA BIOS resides in ROM on VGA adapter cards. The VGA BIOS 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
- Multisynchronous style monitor, such as the NEC MultiSync

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 interface 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 (DAC) for display.

Each of these components contains a fixed number of control registers that are accessed via 8-bit I/O ports. The BIOS 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.

Digital-to-Analog Converter (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., $2^{3*6} = 256K$).

continued

DAC, Attribute Controller, ands BIOS initialization

The BIOS 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.

Video RAM

The BIOS 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 heading Video Modes.

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. MultiSync-style monitors are also supported.

VGA I/O port addressing

The IBM VGA (or compatible) chip is a very large scale integration (VLSI) device that incorporates the functionality of six general purpose adapter control registers, a CRTC, an Attribute Controller, an Address Sequencer, and a Graphics Controller onto one chip.

The limited I/O address space of PC/XT/AT and compatible computers, makes it is impossible to assign each register its own I/O port. With the exception of the general registers, each major VGA register group is addressed via an index/access port pair. For example, the VGA Sequencer's index register is mapped to I/O port 03C4h. The value (index) written to port 03C4h determines which register appears at I/O port 03C5h.

continued

VGA I/O ports

The VGA I/O port address map is listed in the table below.

I/O Address	Read/Write	Index	Description
GENERAL REGISTERS			
03CCh	R	—	Miscellaneous Output
03C2h	W	—	Miscellaneous Output
03C2h	R	—	Input Status 0
3?Ah*	R	—	Input Status 1
03CAh	R	—	Feature Control
03?Ah/3?Ah*	W	—	Feature Control
03C3h	R/W	—	VGA Enable
03C7h	R	—	DAC State
SEQUENCER REGISTERS			
03C4h	R/W	—	Sequencer Index Register
03C5h	R/W	00h	Reset
03C5h	R/W	01h	Clocking Mode
03C5h	R/W	02h	Map Mask
03C5h	R/W	03h	Character Map Select
03C5h	R/W	04h	Memory Mode

* The value of ? depends on the value of bit 0 in the Miscellaneous Output Register
? = 03BAh in Monochrome Emulation Modes
? = 03DAh in Color Emulation Modes

continued

VGA I/O ports, cont'd

I/O Address	Read/Write	Index	Description
CRT CONTROLLER REGISTERS			
3?4h/3?4h*	R/W	—	CRTC Index Register
3?5h/3?5h*	R/W	01h	Horizontal Total
3?5h/3?5h*	R/W	02h	Horizontal Display Enable End
3?5h/3?5h*	R/W	03h	Start Horizontal Blanking
3?5h/3?5h*	R/W	04h	End Horizontal Blanking
3?5h/3?5h*	R/W	05h	Start Horizontal Retrace Pulse
3?5h/3?5h*	R/W	06h	End Horizontal Retrace
3?5h/3?5h*	R/W	07h	Vertical Total
3?5h/3?5h*	R/W	08h	Overflow
3?5h/3?5h*	R/W	09h	Preset Row Scan
3?5h/3?5h*	R/W	0Ah	Maximum Scan Line
3?5h/3?5h*	R/W	0Ah	Cursor Start
3?5h/3?5h*	R/W	0Bh	Cursor End
3?5h/3?5h*	R/W	0Ch	Start Address High
3?5h/3?5h*	R/W	0Dh	Start Address Low
3?5h/3?5h*	R/W	0Eh	Cursor Location High
3?5h/3?5h*	R/W	0Fh	Cursor Location Low
3?5h/3?5h*	R/W	10h	Vertical Retrace Start
3?5h/3?5h*	R/W	11h	Vertical Retrace End
3?5h/3?5h*	R/W	12h	Vertical Display Enable End
3?5h/3?5h*	R/W	13h	Offset
3?5h/3?5h*	R/W	14h	Underline Location
3?5h/3?5h*	R/W	15h	Start Vertical Blank
3?5h/3?5h*	R/W	16h	End Vertical Blank
3?5h/3?5h*	R/W	17h	CRTC Mode Control
3?5h/3?5h*	R/W	18h	Line Compare

* The value of ? depends on the value of bit 0 in the Miscellaneous Output Register
? = 03BAh in Monochrome Emulation Modes
? = 03DAh in Color Emulation Modes

continued

VGA I/O ports, cont'd

I/O Address	Read/Write	Index	Description
ATTRIBUTE CONTROLLER REGISTERS			
03C0h	R/W	—	Attribute Controller Index Register
03C1h	R/W	00h-0Fh	Palette Registers 00h - 0Fh
03C1h	R/W	10h	Attribute Mode Control Register
03C1h	R/W	11h	Overscan Color Register
03C1h	R/W	12h	Color Plane Enable Register
03C1h	R/W	13h	Horizontal PEL Panning Register
GRAPHICS CONTROLLER REGISTERS			
03CEh	R/W	—	Graphics Controller Index Register
03CFh	R/W	00h	Set/Reset
03CFh	R/W	01h	Enable Set/Reset
03CFh	R/W	02h	Color Compare
03CFh	R/W	03h	Data Rotate
03CFh	R/W	04h	Read Map Select
03CFh	R/W	05h	Graphics Mode Register
03CFh	R/W	06h	Miscellaneous
03CFh	R/W	07h	Color Don't Care
03CFh	R/W	08h	Bit Mask
AFFFh	R/W	00h	Plane 0 System Latch
AFFFh	R/W	01h	Plane 1 System Latch
AFFFh	R/W	02h	Plane 2 System Latch
AFFFh	R/W	03h	Plane 3 System Latch

continued

VGA I/O ports, cont'd

I/O Address	Read/Write	Index	Description
DIGITAL TO ANALOG CONVERTER (DAC) REGISTERS			
03C8h	R/W	—	Pixel Address (Write Mode)
03C8h	W	—	Pixel Address (Read Mode)
03C6h	R	—	DAC State Register
03C6h	R/W	—	Pixel Mask*
03C9h	R/W	00h	Red value color 0
03C9h	R/W	00h	Green value color 0
03C9h	R/W	00h	Blue value color 0
03C9h	R/W	01h	Red value color 1
⋮	⋮	⋮	⋮
03C9h	R/W	FFh	Red value color FFh
03C9h	R/W	FFh	Green value color FFh
03C9h	R/W	FFh	Blue value color FFh
03C1h	R/W	14h	Color select register
* Applications should not write to this register. Unpredictable results will occur.			

For more information

A full discussion of VGA hardware is beyond the scope of this book. For more information, see the documentation accompanying your VGA adapter or one of the excellent references listed at the end of this book.

MDA video mode

MDA adapters support only one video mode.

Mode	Type	Resolution	Max. Colors	Scheme	Char. Box	Max. Pages	Buff. Start
7	Text	720x350	mono	80x25	9x14	8	B0000h

CGA video modes

The table below lists the seven video modes supported by the CGA BIOS.

Mode	Type	Resolution	Max. Colors	Scheme	Char. Box	Max. Pages	Buff. Start
0	Text	320x200	16	40x25	8x8	8	B8000h
1	Text	320x200	16	40x25	8x8	8	B8000h
2	Text	640x200	16	80x25	8x8	8	B8000h
3	Text	640x200	16	80x25	8x8	8	B8000h
4	Graphics	320x200	4	40x25	8x8	1	B8000h
5	Graphics	320x200	4	40x25	8x8	1	B8000h
6	Graphics	640x200	2	80x25	8x8	1	B8000h

CGA video mode facts

The following facts apply to the CGA supported modes:

- No cursor is displayed in graphics modes
- Modes 1, 3, and 4 are the same as modes 0, 2,and 5 except that in modes 0, 2, and 5 color burst is enabled. Enabling color burst allows color information to be displayed on composite color displays.
- Digital RGB displays are not affected by enabling color burst.

continued

EGA video modes

The table below lists the sixteen video modes supported by the EGA BIOS.

Mode	Type	Disp.	Res.	Max. Colors	Scheme	Char. Box	Max. Pages	Buff. Start
0	Text	CD	320x200	16	40x25	8x8	8	B8000h
0	Text	ECD	320x350	16/64	40x25	8x14	8	B8000h
1	Text	CD	320x200	16	40x25	∂x8	8	B8000h
1	Text	ECD	320x350	16/64	40x25	8x14	8	B8000h
2	Text	CD	640x200	16	80x25	8x8	8	B8000h
2	Text	ECD	720x400	16/64	80x25	8x14	4/8	B8000h
3	Text	CD	640x200	16	80x25	8x8	4/8	B8000h
3	Text	ECD	640x200	16/64	80x25	8x14	4/8	B8000h
4	Graphics	CD/ECD	320x200	4	40x25	8x8	1	B8000h
5	Graphics	CD/ECD	320x200	4	40x25	8x8	1	B8000h
6	Graphics	CD/ECD	640x200	2	80x25	8x8	1	B8000h
7	Text	Mono	720x350	Mono	80x25	9x14	4	B0000h
D	Graphics	CD/ECD	320x200	16	40x25	8x8	2/4	A0000h
E	Graphics	CD/ECD	640x200	16	80x25	8x8	1/2	A0000h
F	Graphics	Mono	640x350	Mono	80x25	9x14	1	A0000h
10	Graphics	ECD	640350	4/16	80x25	8x14	1/2	A0000h

Legend:

CD = IBM Color Display, or compatible
ECD = IBM Enhanced Color Display, or compatible
Mono = IBM Monochrome Display, or compatible

1/2 = 64K/128K video memory installed
4/8 = 64K/128K video memory installed
4/8 = 64K/128K video memory installed

EGA video mode facts

The following facts apply to the EGA supported modes:

- No cursor is displayed in graphics modes
- Modes 1, 3, and 4 are the same as modes 0, 2,and 5 except that in modes 0, 2, and 5 color burst is enabled. Enabling color burst allows color information to be displayed on composite color displays.
- Digital RGB displays are not affected by enabling color burst.

continued

VGA video modes

The VGA BIOS supports seventeen video modes, providing backward compatibility with MDA, CGA, and EGA modes, as well as all new VGA modes.

The table below lists the modes supported by the VGA BIOS.

Mode	Emul.	Res.	Type	Max. Colors	Scheme	Char. Box	Max. Pages	Buff. Start
0, 1	CGA*	320x200	Text	16/256K	40x25	8x8	8	B8000h
0, 1	EGA*	320x350	Text	16/256K	40x25	8x14	8	B8000h
0, 1	VGA+	360x400	Text	16/256K	40x25	9x16	8	B8000h
2, 3	CGA*	640x200	Text	16/256K	80x25	8x8	8	B8000h
2, 3	EGA*	640x350	Text	16/256K	80x25	8x14	8	B8000h
2, 3!	VGA+	720x400	Text	16/256K	80x25	9x16	8	B8000h
4, 5	CGA	320x200	Graphics	4/256K	40x25	8x8	1	B8000h
6	CGA	640x200	Graphics	2/256K	80x25	8x8	1	B8000h
7	MDA*	720x350	Text	MDA Mono	80x25	9x14	8	B0000h
7!	VGA+	720x400	Text	VGA Mono	80x25	9x16	8	B0000h
D	EGA	320x200	Graphics	16/256K	40x25	8x8	8	A0000h
E	EGA	640x200	Graphics	16/256K	80x25	8x8	4	A0000h
F	EGA	640x350	Graphics	Mono	80x25	8x14	2	A0000h
10	EGA	640x350	Graphics	16/256K	80x25	8x14	2	A0000h
11	VGA	640x480	Graphics	2/256K	80x30	8x16	1	A0000h
12	VGA	640x480	Graphics	16/256K	80x30	8x16	1	A0000h
13	VGA	320x200	Graphics	256/256K	40x25	8x8	1	A0000h

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 lines for text modes for details.)
"+" Indicates default mode

continued

VGA video mode facts

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

Fact	Description
Modes 0, 2, 4 = modes 1, 3, 5	In CGA, modes 1, 3, and 5 have color burst turned on and modes 0, 2, and 4 have color burst turned off. The VGA hardware does not support color burst; modes 0, 2, and 4 are identical to modes 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.
No cursor in graphics	The cursor is not displayed in graphics modes.
MDA, CGA, EGA Emulation	To insure compatibility with older software, the BIOS 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 Select scan lines for text mode with AL = 00h or 01h. (i.e., 200 or 350 scan lines). ▪ Execute Function AH = 00h Set Video Mode. **Note:** Unless otherwise indicated, the BIOS defaults to 400 scan lines for all text modes.

Video Service data definitions

The data definitions used by the INT 10h Video Service 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 Service 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 [EGA] [VGA]

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.

AH = 00h Set Video Mode refers to the INT 10h Video Service pointer tables (and to the information pointed to by the tables) on every mode set.

Default table locations [EGA] [VGA]

There are two pointer tables: the primary table and the secondary table. The locations of each of these tables is initialized by the BIOS 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 BIOS 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 BIOS default locations may supply their own values for 40:A8h and entry 5 of Table #1. See below.

continued

Structure of Video Service Pointer Table #1 [EGA] [VGA]

The location of the first, or primary, pointer table must be stored in 40:A8h. This address 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>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 Text Mode Cursor Size, in this chapter for details.</td></tr>
</table>

continued

Structure of Video Service Pointer Table #1, cont'd [EGA] [VGA]

<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 [VGA]

The location of the 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 BIOS 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 [VGA]

<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 [VGA]

<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 = 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>

Summary of Video Service functions

The physical location and number of functions supported by a given video BIOS depends on the kind of adapter in place. MDA and CGA adapters rely exclusively on the Video Service contained in the system board ROM BIOS of all PC/XT/AT and compatible computers. EGA and VGA adapters contain a dedicated ROM BIOS installed on the adapter hardware itself.

The table below lists the Video Service functions and subfunctions. The column labeled Adapter Type defines each function's level of support for each video adapter type.

AH	Description	Adapter Type
00h	Set Video Mode	[MDA] [CGA] [EGA] [VGA]
01h	Set Text Mode Cursor Size	[MDA] [CGA] [EGA] [VGA]
02h	Set Cursor Position	[MDA] [CGA] [EGA] [VGA]
03h	Read Current Cursor Position	[MDA] [CGA] [EGA] [VGA]
04h	Read Light Pen Position	[CGA] [EGA]
05h	Select New Video Page	[CGA] [EGA] [VGA]
06h	Scroll Current Page Up	[MDA] [CGA] [EGA] [VGA]
07h	Scroll Current Page Down	[MDA] [CGA] [EGA] [VGA]
08h	Read Character/Attribute from Screen	[MDA] [CGA] [EGA] [VGA]
09h	Write Character/Attribute to Screen	[MDA] [CGA] [EGA] [VGA]
0Ah	Write Character Only to Screen	[MDA] [CGA] [EGA] [VGA]
0Bh	Set Color Palette	[CGA] [EGA] [VGA]
0Ch	Write Pixel	[CGA] [EGA] [VGA]
0Dh	Read Pixel	[CGA] [EGA] [VGA]
0Eh	Write Teletype to Active Page	[MDA] [CGA] [EGA] [VGA]
0Fh	Return Video Status	[MDA] [CGA] [EGA] [VGA]

continued

Summary of Video Service functions, cont'd

AH	Description	Adapter Type
10h	Set Palette/Color Registers AL = 00h Set single palette AL = 01h Set overscan register AL = 02h Set all palette registers and overscan (border color) AL = 03h Toggle intensity/blinking bit AL = 04h–06h Reserved * AL = 07h Read individual palette register * AL = 08h Read overscan register (border color) * AL = 09h Read all palette registers and overscan register (border color) * 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 * 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	[EGA] [VGA]
11h	Load Character Generator AL = 00h Load user text mode font AL = 01h Load ROM 8x14 text mode font AL = 02h Load ROM 8x8 double dot text mode font AL = 03h Set block specifier(text mode only) * AL = 04h Load 8x16 ROM text mode font AL = 10h Load user text mode font (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 8x16 ROM text mode font (after mode set) AL = 20h Set user graphics characters pointer at INT 1Fh (8x8 font) AL = 21h Set user graphics font pointer at INT 43h AL = 22h Use ROM 8x14 font for graphics AL = 23h Use ROM 8x8 double dot font for graphics * AL = 24h Use ROM 8x16 font for graphics AL = 30h Get font pointer information	[EGA] [VGA]
* Supported on VGA adapters only.		

continued

Summary of Video Service functions, cont'd

AH	Description	Adapter Type
12h	Alternate Select BL = 10h Return configuration 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	[EGA] [VGA]
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)	[MDA] [CGA] [EGA] [VGA]
14h–19h	Reserved	
1Ah	Read/Write Display Combination Code AL = 00h Read display combination code AL = 01h Write display combination code	[VGA]
1Bh	Return Functionality/State Information	[VGA]
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	[VGA]
* Supported on VGA adapters only.		

How Functions Are Called [MDA] [CGA] [EGA] [VGA]

Description

Individual Video Service functions are selected via the AH register. Subfunctions are selected via the AL register or the BL register. The following general rules apply to these functions:

- 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 function 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.).

Error Handling [MDA] [CGA] [EGA] [VGA]

If the number in AH is outside the legal range, no action will be taken.

Function: AH = 00h Set Video Mode [MDA] [CGA] [EGA] [VGA]

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 the Video Modes heading in this chapter for a description of each mode.)

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 — Mode is greater than 7
30h — Mode is from 0–5 or 7
3Fh — 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 INT 10h Function 06h rather than Function 00h to clear the screen.
- The cursor is not displayed in graphics modes.
- On EGA and VGA adapters, 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 on VGA adapters, if bit 7 of AL is set, the video buffer is not cleared.
- Refer to AH = 12h BL = 30h to select the number of scan lines for text modes (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 Text Mode Cursor Size

[MDA] [CGA]
[EGA] [VGA]

Description

This function sets the size of the cursor that appears 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.

This function stores cursor size parameters in the cursor type byte at 40:60h.

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 the change background attributes.
- CGA modes can display a 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.
- 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 Set Video Mode 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 [MDA] [CGA] [EGA] [VGA]

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 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

[MDA] [CGA]
[EGA] [VGA]

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/Output

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 examined independently with this function, no matter which page is currently active. The number of available display pages is defined by the display mode selected.

Function: AH = 04h Read Light Pen Position [CGA] [EGA]

Description

This function reads the light pen's status and position. Light pen input devices are not supported on VGA adapters. Light pens 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/Output

Input: AH = 04h

Output: AH = 00h Light pen switch is not active
= 01h Light pen coordinate values
DH,DL = Row and column of character
CH = Raster line (0 -199)
CL = Raster line (0 - nnn) modes > 200 line resolution
BX = Pixel column (0 - 319,639)

Function: AH = 05h Select New Video Page [CGA] [EGA] [VGA]

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 modes, 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 BIOS 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 [MDA] [CGA]
Function: AH = 07h Scroll Current Page Down [EGA] [VGA]

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 functions to fill the blank line with text.

Initializing a window

Setting AL = 00h 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 the window with the attribute in the BH register.

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

[MDA] [CGA] [EGA] [VGA]

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 the Video Modes heading for the 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

[MDA] [CGA]
[EGA] [VGA]

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.

INT 10h Functions AH = 09h and AH = 0Ah are similar. AH = 09h should be used for all graphics modes and in all text modes where character by character control of foreground and background attributes is desired.

Input/Output

Input:

AH	=	09h
AL	=	ASCII character to write
BH	=	Display page (refer to the Video Modes heading for the 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 VGA graphics mode 13h, the color specified in BH determines the screen background color.
- In CGA graphics modes, 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 = 20h and AL = 21h 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 bit 7 of BL is 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

[MDA] [CGA] [EGA] [VGA]

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. This function is often used 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 bit 7 of BL is 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 [CGA] [EGA] [VGA]

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
= Mode = 4, 5, set background color to value in BL
= Mode = 0, 1, 2, 3, set border color to value in BL
= 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 [CGA] [EGA] [VGA]

Description

The Write Pixel function writes to video memory the pixel specified by row and column number in DX and CX. When a 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 the Video Modes heading for 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)
CX	=	Pixel column number
DX	=	Pixel row number

Output: None

Pixel values

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

Function: AH = 0Dh Read Pixel [CGA] [EGA] [VGA]

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 the Video Modes heading for a listing of 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.

Function: AH = 0Eh
Write Teletype to Active Page

[MDA] [CGA] [EGA] [VGA]

Description

This function makes the display appear as a serial terminal. The character in AL is written to video memory to be placed in the active page at the current cursor position 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)
BH = Active page

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.

continued

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, it moves the cursor to the right one space. It then wraps 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.
- 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 to use for simple output.

DOS uses

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

This function does not, however, allow the attribute of a text character to be selected. To define an attribute for a character written to the screen:

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

With this method, the user does not have to provide for line wrapping and screen scrolling, but can allow the BIOS to control line wrap and screen scroll.

Function: AH = 0Fh
Return Video Status

[MDA] [CGA]
[EGA] [VGA]

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 the Video Modes heading 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 use

Some ways to use this function are:

- To determine 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.
- To write 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 [EGA] [VGA]

Description

This function has fourteen subfunctions that control operations on the color palette registers in EGA/VGA video controllers. The subfunctions control color, blinking and the video DAC for VGA controllers. There are 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 (border color)	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 (border color)	Atrib. Contrl.
09h*	Read all palette registers and overscan register (border color)	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
* Supported on VGA adapters only.		

continued

Subfunction: AL = 00h Set single palette [EGA] [VGA]

This subfunction sets a single color value in the Attribute Controller's 16-value internal palette. For VGA mode 13h (i.e, 256-color graphics), this subfunction exits without processing.

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

Output: None

Subfunction: AL = 01h Set overscan register [EGA] [VGA]

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) [EGA] [VGA]

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. This function exits without processing if the current mode is VGA mode 13h (i.e., 256-color graphics).

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 [EGA] [VGA]

This subfunction sets either the background intensity or the foreground blinking for all color modes displayed by EGA and VGA adapters.

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 [VGA]

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

Input: AH = 10h
AL = 07h
BL = Palette register to be read (Range 00h –0Fh)

Output: BH = Value read

Subfunction: AL = 08h Read overscan register (border color) [VGA]

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: AL = 09h
Read all palette registers and overscan register (border color) [VGA]

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 [VGA]

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
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)
DH = Red 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 [VGA]

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 [VGA]

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 in 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.

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

Output: None

The Set single palette subfunction (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 [VGA]

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 [VGA]

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

Input: AH = 10h
AL = 17h
BX = DAC color register that 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 [VGA]

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.

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 [VGA]

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

Output: None

Function: AH = 11h Load Character Generator [EGA] [VGA]

Description

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

Load Character Generator subfunctions

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

AL Value	Subfunction Name
00h	Load user text mode font
01h	Load ROM 8x14 text mode font
02h	Load ROM 8x8 double dot text mode font
03h	Set block specifier (text modes only)
04h*	Load 8x16 ROM text mode font
10h	Load user text mode font (after mode set)
11h	Load ROM 8x14 text mode font (after mode set)
12h	Load ROM 8x8 double dot text mode font (after mode set)
14h*	Load 8x16 ROM 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 for graphics
23h	Use ROM 8x8 double dot font for graphics
24h*	Use ROM 8x16 font for graphics
30h	Get font pointer information
* Supported on VGA adapters only.	

continued

Subfunctions: AL = 00h, 01h, 02h, 03h, 04h [EGA] [VGA]

As a group, subfunctions 00h – 04h allow the user to load character font sets into memory map 2, the area of video memory where the BIOS references text mode character generators. There are slight differences in the way these functions are supported on EGA and VGA adapters.

The following facts apply:

- On EGA adapters, this subfunction performs a mode set, resetting the video environment but preserving the contents of the video buffer.
- Subfunction 04h is not supported on EGA subsystems.
- On VGA adapters, no mode set occurs. Scan lines per character, number of character rows, buffer length, and cursor size are not recalculated.
- To prevent unpredictable results, the user-defined font loaded here must occupy a character box size that is close in size to the one employed by the current mode.

Subfunction: AL = 00h Load user text mode font [EGA] [VGA]

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 subfunction 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 00h – 07h)
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

---

continued

Subfunction: AL = 01h Load ROM 8x14 text mode font **[EGA] [VGA]**

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

Input: AH = 11h
AL = 01h
BL = Block to load (valid values are 00h - 07h)

Output: None

Subfunction: AL = 02h Load 8x8 double dot text mode font **[EGA] [VGA]**

This subfunction loads the ROM-resident 8x8 font into the character block indicated in BL. Use this subfunction to override the BIOS 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 00h - 07h)

Output: None

---

continued

Subfunction: AL = 03h Set block specifier (text modes only) [EGA] [VGA]

EGA- and 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 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, and 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 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, and 0 = bits 5, 3, and 2, then bit 3 of the character attribute byte toggles foreground intensity on or off. When two character blocks are selected, Function 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 = 04h Load 8x16 ROM text mode font [VGA]

This subfunction loads the ROM-resident 8x16 font into the character block indicated in BL. Use this subfunction to override the BIOS 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

Subfunctions: AL = 10h, 11h, 12h, 14h [EGA] [VGA]

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,
- the bytes per character (points) are recalculated,
- the number of rows is recalculated,
- the display buffer length is calculated, and
- the following CRTC registers are reprogrammed:
  - Index 09h Maximum scan line index
  - Index 0Ah Cursor start index
  - Index 0Bh Cursor end index
  - Index 12h Vertical displacement end index
  - Index 14h Underline location (mode 07h only)

---

continued

Subfunction: AL = 10h
Load user text mode font (after mode set) [EGA] [VGA]

This subfunction loads the user character set defined in the table pointed to by ES:BP to the character block indicated BL.

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 EGA/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 call to INT 10h, AH = 00h Set Video Mode.

Subfunction: AL = 11h
Load ROM 8x14 text mode font (after mode set) [EGA] [VGA]

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 BIOS-defined font for the text mode in question.

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 EGA/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 call to INT 10h, AH = 00h Set Video Mode.

---

continued

Subfunction: AL = 12h
Load ROM 8x8 double dot text mode font [EGA] [VGA]

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.

Input: AH = 11h
AL = 12h
BL = Block to load

Output: None

The registers programmed by subfunction 12h are a subset of all EGA/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 = 00h Set Video Mode).

Subfunction: AL = 14h Load 8x16 ROM text mode font [VGA]

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.

Input: AH = 11h
AL = 14h
BL = Block to load

Output: None

The 8x16 text mode font is not available in EGA adapters. As such, 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 = 00h Set Video Mode).

---

continued

Subfunctions: AL = 20h, 21h, 22h, 23h, 24h [EGA] [VGA]

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 video mode is set (See INT 10h, AH = 00h Set Video Mode, AL = mode to select).

Subfunction: AL = 20h
Set user graphics font pointer at INT 1Fh (8x8 font) [EGA] [VGA]

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 on EGA or VGA adapters that are set to modes 04h – 06h, the CGA graphic modes. The graphics character set loaded can contain bit patterns for ASCII characters 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

- The character generators for all graphics modes are contained within the video BIOS. The ROM BIOS–based character generators for graphics VGA 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 to 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 to the DOS GRAFTABL graphics table is the usual method of providing user–supplied extended ASCII characters.

---

continued

Subfunction: AL = 21h
Set user graphics font pointer at INT 43h [EGA] [VGA]

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

Subfunction: AL = 22h Use ROM 8x14 font for graphics [EGA] [VGA]

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.

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 [EGA] [VGA]

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

Subfunction: AL = 24h Use ROM 8x16 font for graphics [VGA]

This subfunction specifies the use of the ROM-based 8x16 font in 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 [EGA] [VGA]

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.

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 [EGA] [VGA]

Description

The Alternate Select function contains several subfunctions. Each subfunction allows the user to enable or disable certain operations that are standard video mode defaults. For example, on VGA adapters the video BIOS 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 configuration 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
* Supported on VGA adapter only.	

continued

Subfunction: BL = 10h Return configuration information [EGA] [VGA]

This subfunction returns the state of EGA or VGA hardware.

Input: AH = 12h
BL = 10h

Output: BH = 00h EGA/VGA color mode
= 01h EGA/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

In VGA compatible subsystems, the references to adapter bits and switch settings are retained only for historical reasons. 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 configuration information, cont'd [EGA] [VGA]

On VGA adapters, 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	Reserved	

On VGA adapters, 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	Reserved

Subfunction: BL = 20h Switch to alternate print screen routine [EGA] [VGA]

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 [EGA] [VGA]

This subfunction sets the number of scan lines to be displayed in text modes. By default, the VGA BIOS sets all text modes to 400 scan lines as part of the power-on self test and initialization (POST) process.

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
= 00h VGA not active

- The number of scan lines indicated in AL will take effect upon the next mode set.
- The VGA 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 [VGA]

If AL = 01h 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 is supported

---

continued

Subfunction: BL = 32h Enable/disable video [VGA]

This subfunction enables/disables the video I/O port regenerator and the display buffer address decoder for the currently active video monitor.

Input: AH = 12h
AL = 00h Enable video
= 01h Disable video
BL = 32h

Output: AL = 12h Function is supported

Subfunction: BL = 33h Enable/disable summing to gray shades [VGA]

This subfunction enables/disables gray scale summing that occurs during the Set Video Mode Function (AH = 00h) and during the Set Palette Registers Function (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 [VGA]

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 Text Mode Cursor Size for more information on setting cursor type.

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 [VGA]

This subfunction permits switching between the system video and an adapter video ROM. When there is a conflict between the system video and an adapter video (i.e., overlapping usage of the BIOS 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 Enable/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. The user must establish a 128-byte save area pointed to by ES:DX
= 01h Turn on initial system board video
= 02h Disable active video. The user must have a save buffer pointer in ES:DX
= 03h Enable inactive video. The user must have a pointer in ES:DX to a previously filled save buffer
BL = 35h
ES:DX = Pointer to a 128-byte save buffer area

Output: AL = 12h Function is supported

When there is no conflict between the system video and an adapter video board, both video devices are active and use of this subfunction is not necessary.

Subfunction: BL = 36h Video screen on/off [VGA]

This subfunction turns the video screen on or off.

Input: AH = 12h
AL = 01h Screen off
= 00h Screen on
BL = 36h

Output: AL = 12h Function is supported

Function: AH = 13h Write String [MDA] [CGA] [EGA] [VGA]

Description

The Write String function operates similarly to Write Teletype to Active Page Function (AH = 0Eh), except that an entire string is handled with each call. The AL register contains two single-bit fields differentiated 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 = 00h Cursor not moved.
= 01h Cursor is moved.
= 02h Cursor not moved (text mode only)
= 03h Cursor is moved (text mode only)

Output: None

Functions: AH = 14h - 19h Reserved

Function: AH = 1Ah Read/Write Display Combination Code [VGA]

Description

The Read/Write Display Combination Code function has two subfunctions, one devoted to reading the display codes (AL = 00h) and one devoted to writing the display combination codes (AL = 01h). 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–FEh	Reserved
FFFFh	Display type unknown

Subfunction: AL = 00h Read display Combination Code [VGA]

This subfunction reads the display code in AL.

Input: AH = 1Ah
AL = 00h

Output: AL = 1Ah Function is supported
BH = Inactive display code
BL = Active display code

continued

Subfunction: AL = 00h Write Display Combination Code [VGA]

This subfunction writes the display code in AL.

Input: AH = 1Ah
AL = 01h
BH = Alternate display code (if any)
BL = Active display code

Output: AL = 1Ah — Function is supported

Function: AH = 1Bh Return Functionality/State Information [VGA]

Description

This 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 BIOS 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

This function 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 Mode heading in this chapter for supported video modes)
DI+05h	1 Word	Column/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	Number of scan lines in current video mode 00b = 200 01b = 350 10b = 400 11b = 480 100b - 11111111b = Reserved
DI+2Bh	1 Byte	Primary character block 00b = Block 0 01b = Block 1 10b = Block 2 11b - 11111111b = Blocks 3 to 255
DI+2Ch	1 Byte	Secondary character block 00b = Block 0 01b = Block 1 10b = Block 2 11b - 11111111b = Blocks 3 - 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. The table is located at E000:305Fh in the BIOS. 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 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 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, Bits 7-4 = 0 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 [VGA]

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 a 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

continued

Error handling

Upon entry into the Save/Restore Video State function, the BIOS 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 [VGA]

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 [VGA]

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 current video state subfunction immediately after saving the video state.

Subfunction: AL = 02h Restore current video state [VGA]

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, the current hardware state is stored to the system RAM buffer pointed to by 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	Palette Registers 00h - 0Fh
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 current video state subfunction is executed, the current video BIOS state is stored to the system RAM buffer pointed to by ES:BX. Executing the Restore current video state subfunction with the same CX and ES:BX values restores the video BIOS state information in its location in the ROM BIOS data area (0400h–0500h) in system RAM.

Video data definitions

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

Location	Size	Description
40:49h	1 Byte	Video mode setting.
40:4Ah	1 Word	Number of columns on screen.
40:4Ch	1 Word	Current page size.
40:4Eh	1 Word	Current page address.
40:50h	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.
40:60h	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.
40:62h	1 Byte	Current page number.
40:63h	1 Word	6845-compatible I/O port number for current mode. (Port 03D4h or 03B4h).
40:65h	1 Byte	Current mode select register.
40:66h	1 Byte	Current palette value.
40:84h	1 Byte	Number of rows on screen (24 or 25).
40:85h	1 Word	Character height (bytes/character).

continued

Video data definitions, cont'd

System RAM Offset (hex)	Size	Description
40:87h	1 Byte	Video control bits, where: Bit 7 = Clear RAM Bits 6-5 = Memory on video hardware , where: 00b = 64K 01b = 128K 10b = 192K 11b = 256K Bit 4 = Not used Bit 3 = 0 EGA-compatible mode is active Bit 2 = 0 Wait for display enable Bit 1 = 0 Color or ECD monitor is 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
40:88h	1 Byte	EGA/VGA switch data where: Bits 7-4 = Feature connector bits 3-0, respectively Bits 3-0 = Option switches 3-0, respectively
40:89h	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
40:8Ah	1 Byte	Index to the Display Combination Code table
40: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 current 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 current 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

Functions: AH = 1Dh - FFh Reserved

10

Chapter 10
INT 13h Diskette Service

Overview

Description

The BIOS Diskette Service performs read, write, format, diagnostic, initialization and other operations for up to two internal diskette drives.

PC diskette drive support history

The original PC provided support for single-sided, single-density 5.25-inch diskette drives. Soon, support for 360K diskette drives was added. The AT introduced the high-density 1.2 MB 5.25-inch diskette drive. Later versions of the AT introduced support for 3.5-inch diskette drives.

How the diskette service is invoked

The BIOS Diskette Service is invoked via software INT 13h. The BIOS initializes the INT 13h Diskette Service vector to F000:EC59h. The INT 13h vector resides at address 00:4Ch in the interrupt vector table.

The BIOS 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 BIOS automatically redirects all INT 13h Diskette Service requests to INT 40h. This redirection is transparent to BIOS users. Users should continue to invoke INT 13h for both diskette and fixed disk services.

For a discussion of the BIOS Fixed Disk Service, refer to Chapter 11.

In this chapter

This chapter focuses on the BIOS Diskette Service INT 13h. The following topics are discussed:

- Summary of Functions
- Theory of Operations
- Hardware Environment
- System RAM Data
- CMOS RAM Data
- ROM BIOS Data
- Diskette Service I/O Ports
- Error Handling
- Diskette Service Functions
- INT 0Eh Diskette Hardware Interrupt

Summary of Functions

Summary of INT 13h Diskette Service functions

Function	Description	Platforms
00h	Reset Diskette System	[XT] [AT]
01h	Read Diskette Status	[XT] [AT]
02h	Read Diskette Sectors	[XT] [AT]
03h	Write Diskette Sectors	[XT] [AT]
04h	Verify Diskette Sectors	[XT] [AT]
05h	Format Diskette Track	[XT] [AT]
06h–07h	Reserved	Reserved
08h	Read Drive Parameters	[AT]
09h–14h	Reserved	Reserved
15h	Read Drive Type	[AT]
16h	Detect Media Change	[AT]
17h	Set Diskette Type	[AT]
18h	Set Media Type for Format	[AT]

How to determine level of support

Throughout this book, the commonly accepted level of support for each ROM BIOS function is identified by the legend [PC], [XT], and/or [AT] on the same line as the function title.

Legend	Description
[PC]	Function is supported by original IBM PC BIOS only.
[XT]	Function is supported by all IBM PC/XT and compatible ROM BIOSs. **Note:** This level of support is contained in the ROM BIOS implemented in the majority of IBM PC and PC/XT compatible systems.
[AT]	Function is supported by all IBM PC/AT and compatible ROM BIOSs.

Introduction

Diskettes remain the fundamental media for transfer of information in the microcomputing world. New software is shipped on diskettes, and despite the proliferation of networks, software in the workplace is still primarily transferred by diskette.

This chapter contains general information about diskette drives and specific information on how the BIOS manipulates the diskette drives.

Ordinarily, programmers will (and should) use the DOS INT 21h diskette functions. INT 21h itself calls the BIOS Diskette Service functions.However, for certain types of applications (games programming, copy protection schemes), a programmer may want to use the BIOS Diskette Service directly, or may even want to bypass the BIOS and write directly to the diskette hardware.

Diskette sizes

The diskette drives commonly supported by PCs come in two sizes — 5.25 inch and 3.5 inch. The 5.25-inch diskette drive supports standard formats of 360K, 720K, or 1.2 MB. The 720K format is not as widely supported as the 360K and 1.2 MB standards.

3.5-inch diskette drives have been available for several years. Many IBM-compatibles support both 720K and 1.44 MB variants of the 3.5-inch diskette drive.

System configurations vary widely when it comes to diskette support. Most IBM-compatible systems that have a Phoenix BIOS support all of the various standards. The Phoenix BIOS even supports 720K quad-density 5.25-inch diskette drives.

continued

Diskette terms

The reader should be familiar with the following diskettes terms:

- **Head**
  The head contains an electromagnet, positioned on a movable assembly just above the surface of the diskette. By pulsing the electromagnet, the head reads or writes data to or from the diskette.

- **Track**
  While the head is positioned over a point on the diskette, ready to read or write, the diskette surface spins underneath it, tracing a full circle. This circle is a track. There may be 40 to 80 tracks per diskette surface, or side.

- **Sector**
  Diskette systems divide each track into short arcs (usually 9, 15, or 18 per track) called sectors. Each sector usually holds 512 bytes of data.

- **Sector size**
  Using Diskette Service function 03h, the caller can change the diskette sector size, but then must carefully control diskette use while the different sector size is used. The caller must also make sure that the sector size is set to 512 bytes for the use of other applications when his routine completes its functions.

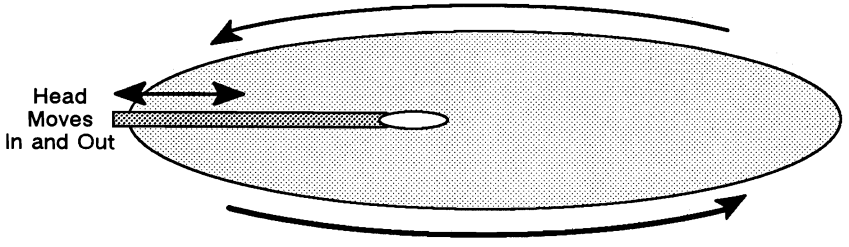
- **Change line**
  Some diskette drives support a disk change line signal that is set if the drive door is opened. The BIOS checks the change line signal before each diskette I/O operation.

continued

Heads and surfaces

Both sides of the diskette platter surface are covered with a material that can be magnetized and is used to store information.

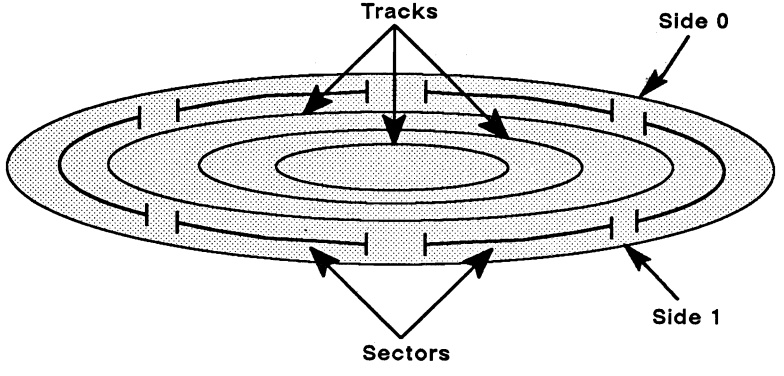
Diskette drives generally have two magnetic "heads" (the mechanisms that allow them to read/write the encoded bits on the diskette surface). One head is used for each side.



Tracks and sectors

The diskette head rapidly magnetizes areas on the diskette surface as the platter spins, representing binary zeros and ones. These areas are organized into concentric circles (tracks). There can be 40, 80, or even more tracks per diskette side.

Tracks are further subdivided into sectors. Sectors usually store 512 bytes, but the sector size can also be changed with INT 13h, AH = 05h. The bytes in a sector must be organized in a prescribed manner. There can be 9, 15, 18 or more sectors per track.



continued

How the diskette service works

The BIOS Diskette Service Read, Write, Verify, and Format functions all require data to be transferred to or from the diskette drive to system memory. This is always done using the DMA controller. DMA facilitates the movement of data directly from memory to I/O devices, freeing the microprocessor for other functions. The DMA controller is programmed by the BIOS before the command block is written to the diskette controller. All diskette operations use DMA controller channel 2.

The process of checking the diskette controller status, sending a byte, and waiting is repeated until all bytes in the command block are sent. Once the command block is received, the controller performs the function requested.

After the diskette controller has performed the command and transferred data, an interrupt is generated by the diskette controller. INT 0Eh, the diskette ISR, handles the interrupt, setting bit 7 of location 40:3Eh. INT 40, the diskette DSR, monitors location 40:3Eh for up to one second, waiting for the interrupt.

The interrupt indicates that one or more (a maximum of 7) result bytes may be waiting for INT 40h to read. If there is no interrupt, the time-out bit in 40:41h is set by INT 40h.

Hardware assumed

The NEC 765 diskette controller chip (or its equivalent) and some additional logic is used to control the diskette drives.

Diskette change line

Some diskette drives and diskette controllers support a change line signal that is set if the drive door of the selected drive is opened. The caller can use this signal to determine if the sector images in memory are still current. The BIOS checks the change line signal before each diskette I/O operation. Bit 7 of port 03F7h indicates the change line signal status: if not set, the drive door has not been opened and a diskette is probably in the drive. If set, the drive door has been opened and additional verification must take place to determine if the diskette has been changed.

Recalibrating a diskette drive

The following information is provided for programs that bypass BIOS diskette operations and write directly to the diskette hardware.

The diskette status information at location 40:3Eh contains a bit for each diskette drive that indicates if a drive should be recalibrated. The BIOS automatically instructs the diskette controller to recalibrate a drive, if necessary.

How diskette drives are identified

Each diskette drive must be identified to the system with a unique number. There are four valid diskette numbers in an XT system. AT systems support a maximum of two internal drives, and, with an external diskette driver (usually a DOS routine) and a Phoenix BIOS, may support a total of four diskette drives.

- Diskette drive 1 is number 0
- Diskette drive 2 is number 1
- Diskette drive 3 is number 2
- Diskette drive 4 is number 3

These values must be in DL for most Diskette Service functions.

continued

Phoenix-supported diskette drive types

AT and XT BIOSs support various combinations of diskette drives types in various configurations. XT systems support up to four diskette drives, but older XT systems may only support 360K 5.25-inch drives. Newer XT systems may support 720K 3.5-inch drives.

Many European models support 720K 5.25-inch diskette drives, but support for these types of drives is rare in the US (the Phoenix BIOS supports them).

Newer AT systems support both 3.5-inch and 5.25-inch diskette drives, but older AT systems may only support 5.25-inch drives.

The Phoenix BIOS Diskette Service supports six types of diskette drives. Not all BIOSs support all diskette drive types on all XT and AT systems.

Drive Type	Media type	Diskette Size	Tracks/Side	Sectors/Track
1.44 MB	1.44 MB	3.5	80	18
720K	720K	3.5	80	9
360K	360K	5.25	40	9
320K	360K	5.25	40	8
720K	720K	5.25	80	9
1.2 MB	1.2 MB	5.25	80	15

360K diskette format compatibility

The BIOS 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

External diskette drives

In some cases, BIOS support is extended to support external diskette drives. This support is usually provided by a DOS driver.

An external diskette driver, EXDSKBIO.DRV, is provided with most Phoenix BIOSs. This driver, operating through DOS, allows up to two additional diskette drives to be configured external to the system. EXDSKBIO.DRV supports all types of both 3.5-inch and 5.25-inch diskette drives.

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	1.2 MB drive	1.2 MB drives		1.2 MB drives

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
Kbs = Kilobits per second			

System RAM data area table

The BIOS Diskette Service references control data stored in the BIOS 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/VGA/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 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

continued

System RAM data area table, cont'd

Location	Size	Description
40:8Bh	2 Bytes	Diskette data rate information (AT only) Bits 7-6 = Last data rate set by controller, where: 00b = 500 Kilobits/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 (AT only), 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 (AT only), where: (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 the first media type 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	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 at POST Bit 4 = 1 Memory size compare error at POST Bit 3 = 1 Fixed disk or adapter fails initialization Bit 2 = 1 RTC 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

Description

The diskette device service routine in PC/XT/AT systems contains a table of parameters used to manipulate diskette drives. For many newer systems, values have been modified so the table is more suitable for use with 3.5-inch diskette drives. Both 3.5-inch and 5.25-inch values are provided below. The default 11-byte table is located in ROM at F000:EFC7h and is pointed to by the interrupt 1Eh vector. This arrangement allows operating systems or application programs to change the INT 1Eh vector to point to a different set of diskette parameters.

Diskette parameter table

<table>
<tr><th>Offset</th><th>Description</th></tr>
<tr><td>00h</td><td>First data byte of the diskette Specify command, where:
<table>
<tr><th rowspan="2">Drive Type/
Transfer Rate</th><th colspan="6">Setting for Bits 7-4</th></tr>
<tr><th>F</th><th>E</th><th>D</th><th>C</th><th>B</th><th>A</th></tr>
<tr><td>360K/250 Kbs</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
<tr><td>Other/500 Kbs</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
<tr><td>Other/300 Kbs</td><td>1.7</td><td>3.3</td><td>5</td><td>6.7</td><td>8.3</td><td>10</td></tr>
<tr><td>Other/250 Kbs</td><td>2</td><td>4</td><td>6</td><td>8</td><td>10</td><td>12</td></tr>
</table>
Bits 3-0 = Head unload time. Default is 240 milliseconds (0Fh).</td></tr>
<tr><td>01h</td><td>Second data byte of the diskette Specify command, where:
Bits 7-1 = Head load time. Default is 01h (4 milliseconds).
Bit 0 = Non-DMA mode flag
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.</td></tr>
<tr><td>02h</td><td>Motor turn-off delay. 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. The default value for this field is 25h.</td></tr>
</table>

continued

Diskette parameter table, cont'd

Offset	Description
03h	Bytes per sector. This field is encoded in the following way to match the encoding used by the diskette controller: 00h = 128 bytes per sector 01h = 256 bytes per sector 02h = 512 bytes per sector (default value) 03h = 1024 bytes per sector
04h	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. The range of values is: 08h = 8 sectors per track (320K drive) 09h = 9 sectors per track (360K/720K 5.25-inch drive) 15h = 15 sectors per track (5.25-inch 1.2 MB drive) 18h = 18 sectors per track (3.5-inch drives)
05h	Gap length. The length of the gap between sectors. 1Bh for a 3.5-inch diskette drive and 2Ah for a 5.25-inch diskette drive.
06h	Data length. Since the bytes per sector field is nonzero, this field is meaningless and is set to FFh.
07h	Gap length for format. The length of the gap between sectors to maintain when formatting. The format gap length is 6Ch for a 3.5-inch drive and 50h for a 5.25-inch drive.
08h	Fill byte for format. The default is F6h.
09h	Head settle time. The amount of time in milliseconds the diskette DSR must wait for the heads to settle after doing a seek operation. For a 3.5-inch diskette drive, this field is 0Fh. 1.2 MB drives require 15 milliseconds and 360K drives, 20 - 25 milliseconds.
0Ah	Motor start time. The amount of time in eighths of a second that the diskette DSR must wait for the motor to come up to speed before doing an I/O operation. Most drives have a motor start time of one second so the default value is 08h for most operations, except Read and Verify, where the default value is either 04h or 05h.

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, where: Bit 7 = 1 NMI disable = 0 NMI enabled Bits 6–0 = 0 CMOS RAM address
0071h	R/W	CMOS RAM data register port
0081h	R/W	DMA channel 2, page register
0372h	W	Diskette controller digital output register, where: Bits 7–6 = 0 Reserved Bit 5 = 1 Enable drive motor 1 Bit 4 = 1 Enable drive motor 0 Bit 3 = 0 Allow interrupts Bit 2 = 0 Controller reset Bit 1 = 0 Reserved Bit 0 = 0 Select drive 0 = 1 Select drive 1

continued

Diskette Service I/O Ports, Continued

I/O Address	Read/Write Status	Description
0374h	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 Bits 3-2 = Reserved Bit 1 = 1 Drive 1 busy Bit 0 = 0 Drive 0 busy
0375-0376h	R/W	Diskette controller data registers
0377h	R	Diskette controller digital input register where: Bit 7 = Diskette change Bits 6-1 = Reserved Bit 0 = 0 High density select
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
03F1h	R	Diskette controller status register B, where: Bit 7 = 0 Reserved Bit 6 = 0 Reserved Bit 5 = 0 Select drive 0 Bit 4 = Write data Bit 3 = Read data Bit 2 = 1 Write enable Bit 1 = 1 Enable drive motor 1 Bit 0 = 1 Enable drive motor 0
03F2h	W	Diskette controller digital output register, where: Bit 7 = 0 Reserved Bit 6 = 0 Reserved Bit 5 = 1 Enable drive 1 motor Bit 4 = 1 Enable drive 0 motor Bit 3 = 0 Enable diskette interrupts and DMA Bit 2 = 0 Controller reset Bit 1 = 0 Reserved Bit 0 = 0 Select drive 0 = 1 Select drive 1

continued

Diskette Service I/O Ports, Continued

I/O Address	Read/Write Status	Description
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	Fixed disk digital input register (AT only), where: Bit 7 = 1 Diskette change Bit 6 = 1 Write gate Bit 5 = Head select 3/reduced write current Bit 4 = Head select 2 Bit 3 = Head select 1 Bit 2 = Head select 0 Bit 1 = Select drive1 Bit 0 = Select drive 0 (Bits 6-0 apply to the fixed disk drive currently selected).
03F7h	W	Diskette data transfer rate select register (AT only), where: Bits 7-2 = Reserved Bits 1-0 = 00b 500 Kbs mode = 01b 300 Kbs mode = 10b 250 Kbs mode = 11b Reserved

Error Handling

Introduction

The BIOS Diskette Service returns the completion status of each function via the AH register, the Carry Flag, and the Diskette Status data definition, located at 40:41h.

Result bytes

The BIOS stores all result bytes returned by the diskette controller in the seven-byte location starting at 40:42h. Usually there are four to seven result bytes.

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 (an error code — see the following page)
- Diskette Status (40:41h) = Error code
- Carry Flag = 1 (set)

continued

Error code table

The BIOS 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

Function: AH = 00h Reset Diskette System [XT] [AT]

Description

This function resets both the diskette controller and the specified diskette drive (the R/W arm is moved to track 0). The diskette drive number (either 00h or 01h) is specified in DL.

Function failure

Call this function when a problem occurs in attempting to access the diskette subsystem with any other function, then retry the function that failed. The diskette subsystem will not react immediately. Instead, a reset flag forces the BIOS to recalibrate the diskette drive's read/write heads the next time they are used. The heads are relocated to track 0 in order to start the next I/O operation from a known state.

Input/Output

Input: AH = 00h
DL = Drive number
Bit 7 = 0 for a diskette
1 for a fixed disk

Output: AH = 00h No error
xxh Error (See error code table earlier in this chapter)
= Diskette Status (40:41h)
CF = 0 No error
= 1 Error

Function: AH = 01h Read Diskette Status [XT] [AT]

Description

The BIOS Diskette Service stores the error code associated with the last requested function in Diskette Status (40:41h). If the last function executed successfully, the Diskette Status and AH are set to 00h. If the last function was not successful, AH and the Diskette Status are set to the appropriate nonzero error code.

This function reads the value contained in Diskette Status and returns it in the AH register. CF is not set by this function.

Input/Output

Input:	AH	=	01h
	DL	=	00h or 01h (drive number)
Output:	AH	=	00h No error
	AL	=	Diskette Status from previous operation

Diskette status byte

The Diskette Status byte is saved after each read, write, verify or format diskette function. This allows error handling or error reporting routines to be written which are entirely independent of diskette operation routines.

Function: AH = 02h Read Diskette Sectors [XT] [AT]

Description

This function reads data from 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 specified in DH. Starting track and sector numbers are specified in CH and CL.

This function is useful for block move operations that require reading many individual sectors or a whole trackful of sectors. For example, the DOS DISKCOPY command uses this function.

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
ES:BX	=	Pointer to buffer

Output:

AL	=	Number of sectors read/written
AH	=	00h No error
		xxh Error (See error code table earlier in this chapter)
	=	Diskette Status (40:41h)
CF	=	0 No error
	=	1 Error

Wait for device to reach proper speed

In some BIOSs, an error may be caused by the diskette drive motor being off when the request is made. The BIOS may not wait for the device to reach proper speed before trying to read. In such a situation, the caller 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 [XT] [AT]

Description

This function writes data to the number of sectors specified in AL on the drive specified in DL from the buffer specified in ES:BX.

The diskette drive head number is specified in DH. Starting track and sector numbers are specified in CH and CL. Diskette 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
ES:BX	=	Pointer to buffer

Output:

AL	=	Number of sectors read/written
AH	=	00h No error
		xxh Error (See error code table earlier in this chapter)
	=	Diskette Status (40:41h)
CF	=	0 No error
	=	1 Error

Wait for device to reach proper speed

In some BIOSs, an error may be caused by the diskette drive motor being off when the request is made. The BIOS may not wait for the device to reach proper speed before trying to read. In such a situation, the caller 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 [XT] [AT]

Description

This function verifies the address fields of 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 specified in DH. Starting track and sector numbers are specified in CH and CL. No data is transferred from the diskette in this operation. Diskette data is not compared to data in memory.

The BIOS Diskette Service verifies diskette sectors by determining if the sectors can be found, read, and pass a Cyclic Redundancy Check (CRC).

Determining if a readable diskette is present

This function can be used to determine if a readable diskette is in the drive, although an error will occur if the diskette motor is off when this function is invoked.

First invoke function 00h, then invoke function 04h. If no diskette is present, AH will contain a nonzero value and the Carry Flag will be set. Retry three times to make sure that no other error condition is occurring.

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
ES:BX	=	Buffer with address field data

Output:

AL	=	Number of sectors actually transferred
AH	=	00h No error
	=	nnh Error (See error code table earlier in this chapter)
	=	Diskette Status (40:41h)
CF	=	0 No error
	=	1 Error

Function: AH = 05h Format Diskette Track [XT] [AT]

Description

This function formats a single diskette track on the drive specified in DL. The format operation consists of writing the diskette sector and track address field data on the specified track. The number of sectors is specified in AL, the head number in DH, and the track number in CH. Each call to function 05h of the Diskette Service can be verified by following it with a call to function 04h.

ES:BX points to a table defining the address fields for the track being formatted.

If the diskette drive in question supports more than one diskette format, the caller must invoke either Diskette Service Function AH = 17h Set Diskette Type or AH = 18h Set Media Type for Format before calling this function.

Use function 17h first if AT system

On an AT-compatible system, invoke INT 13h function 17h to select the type of media to be formatted before invoking function 05h.

Input/Output

Input:

AH	=	05h
AL	=	Number of sectors to be formatted (starts at one)
DH	=	Head number
DL	=	Drive number
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 code table 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	Sector Size indicator, where: 00h = 128 bytes/sector 01h = 256 bytes/sector 02h = 512 bytes/sector 03h = 1024 bytes/sector

Example: Address Field Table

For example, the address field table 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

Function 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.

Function: AH = 08h Read Drive Parameters [AT]

Description

This function returns the diskette parameters for the drive specified in DL. If successful, the Carry Flag is cleared and Diskette Status (40:41h) is not modified.

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.

Diskette change line support

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 1.2 MB diskette in a 1.2 MB drive are pointed to by ES:DI. The drive type returned in BL is zero.

---

continued

Function: AH = 08h Read Drive Parameters, Continued [AT]

Input/Output

Input:
AH = 08h
DL = Drive number

Output:
AX = 0000h
BH = 00h
BL = Bits 4–7 = 0
Bits 0–3 = Valid drive type value from CMOS RAM
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 Invalid parameter
CH = Maximum usable track number
CL = Maximum usable sector number
DH = Maximum usable head number
(always 1 if CMOS RAM value valid)
DL = Number of diskette drives installed (00h, 01h, or 02h)
ES:DI = Pointer to diskette drive parameter table for the maximum media type supported on the specified drive

Function: AH = 15h Read Drive Type [AT]

Description

This function returns information about the drive specified in DL.

Unlike most other BIOS 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
00h	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

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 [AT]

Description

This function determines if the drive door has been opened since the last time the change line was cleared. The change line signal is generally associated with 3.5-inch diskette drives.

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 specified diskette drive is not configured, AH and Diskette Status are set to 80h (time-out), CF is set, and control is returned to the caller.

Using this function

Invoke INT 13h function 15h first to determine whether the diskette drive hardware can sense when a diskette is changed. AH will be set to 02h if it can.

Diskette change line support

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
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 [AT]

Description

This function sets the transfer rate for the specified drives using the diskette types passed in AL. This function is provided for use with DOS 3.0 or 3.1. Function 18h replaces this function for DOS versions 3.2 and above.

1.44 MB drives are not handled by this function.

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 (40:41h) are set to 01h, and control is returned to the caller.

The media transfer rate is set according to the drive type specified. For diskette types 02h, 03h, and 04h, 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 = 00h Unused
= 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
DL = 00h or 01h Drive number

Output:
AH = 00h No error
= xxh Error (See error code table earlier in this chapter)
= Diskette Status (40:41h)
CF = 0 No error
= 1 Error

Function: AH = 18h Set Media Type for Format [AT]

Description

This function sets the media type in preparation for a format command.

If the drive number is invalid, the Carry Flag (CF) is set and control is returned to the caller with both AH and Diskette Status (40:41h) = 01h. 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, the BIOS selects a diskette parameter table 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.

Diskette change line support

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.

Using this function

Invoke this function before formatting a diskette with function 05h so the BIOS can set the correct data rate for the media.

---

continued

Input/Output

Input:
- AH = 18h
- CH = Maximum number of tracks
- CL = Maximum sectors per track
- DL = Drive number

Output:
- ES:DI = Pointer to drive parameter table
- AH = 00h track/sector combination is supported
  - = 0Ch Media unknown (CMOS RAM not valid, drive not configured in CMOS RAM, or not in the 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 BIOS Diskette Interrupt Service Routine (ISR) services interrupts from the diskette controller.

How BIOS Diskette ISR is invoked

The diskette controller generates an INT 0Eh, the BIOS Diskette ISR, to signal the completion of the last command issued by INT 13h.

The INT 0Eh entry point is located at address F000:EF57h. INT 0Eh resides at address 38h in the interrupt vector table.

BIOS multitasking support

Each time the BIOS 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. It then sets bit 7 of (40:3Eh) to signal the completion of the command.

continued

System RAM data area

INT 0Eh, Diskette ISR, references the following system RAM data area location:

Location	Length	Description
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 = 1 Recalibrate drive 1 Bit 0 = 1 Recalibrate drive 0

I/O Port table

INT 0Eh, Diskette ISR, references the following I/O port:

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

11

Chapter 11
INT 13h Fixed Disk Service

Overview

Description

The BIOS Fixed Disk Service performs read, write, format, diagnostic, initialization and other operations for up to two fixed disk drives.

How the service is invoked

INT 13h invokes the BIOS Diskette and Fixed Disk Service. If a fixed disk drive is present, the Fixed Disk Service redirects the INT 13h vector to the interrupt vector table address 00:4Ch. The Phoenix AT and XT BIOSs initialize the INT 13h Fixed Disk Service vector to F000:E3FEh.

BIOS diskette support

When a fixed disk is installed, the BIOS automatically redirects all INT 13h Diskette Service requests to INT 40h. This redirecting 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 BIOS Diskette Services, refer to Chapter 10.

continued

In this chapter

This chapter focuses on the BIOS Fixed Disk Service. The following topics are presented:

- Fixed Disk Service Function Summary
- Theory of Operations
- Hardware Environment
- System RAM Data
- CMOS RAM Data
- ROM BIOS Data
- Fixed Disk Service I/O Ports
- Error Handling
- Fixed Disk Service Functions

How to determine level of support

Throughout this book, the commonly accepted level of support for each ROM BIOS function is identified by the legend [PC], [XT], and/or [AT] on the same line as the function title. In this chapter, the designation [XT] refers to XT-type fixed disk controller support. [AT] refers to AT-type fixed disk controller support.

Legend	Description
[PC]	Function is supported by original IBM PC BIOS only.
[XT]	Function is supported by all IBM PC/XT and compatible ROM BIOSs. **Note:** This level of support is contained in the ROM BIOS implemented in the majority of IBM PC and PC/XT compatible systems.
[AT]	Function is supported by all IBM PC/AT and compatible ROM BIOSs.

Fixed Disk Service Function Summary

Summary of fixed disk service functions

The BIOS Fixed Disk Service contains twenty-three functions. Individual functions are selected via AH. The table below summarizes these functions. The notation in the Controller Type column refers specifically to the type of fixed disk controller in the system, not necessarily the type of system.

Function	Description	Controller type
00h	Reset Diskette(s) and Fixed Disk	[XT] [AT]
01h	Read Fixed Disk Status	[XT] [AT]
02h	Read Sectors	[XT] [AT]
03h	Write Sectors	[XT] [AT]
04h	Verify Sectors	[XT] [AT]
05h	Format Cylinder	[XT] [AT]
06h	Format Bad Track	[XT]
07h	Format Drive	[XT]
08h	Read Drive Parameters	[XT] [AT]
09h	Initialize Drive Parameters	[XT] [AT]
0Ah	Read Long Sectors	[XT] [AT]
0Bh	Write Long Sectors	[XT] [AT]
0Ch	Seek to Cylinder	[XT] [AT]
0Dh	Alternate Fixed Disk Reset	[XT] [AT]
0Eh	Diagnostics 1: Read Test Buffer	[XT]
0Fh	Diagnostics 2: Write Test Buffer	[XT]
10h	Test for Drive Ready	[XT] [AT]
11h	Recalibrate Drive	[XT] [AT]
12h	Controller RAM Diagnostic	[XT] [AT]
13h	Controller Drive Diagnostic	[XT] [AT]
14h	Controller Internal Diagnostic	[XT] [AT]
15h	Read Fixed Disk Type	[XT] [AT]

History

Fixed disks were not supported in the original IBM PC. The IBM XT, introduced later, contained the first fixed disk controller, supporting only one fixed disk drive. XT-type fixed disk controllers are still in use today, even in some AT-compatible systems. Today, most XT-type fixed disk controllers support two or more fixed disk drives.

Although the ST506 interface standard permits support of up to four drives per controller, most AT-type fixed disk controllers support up to two fixed disk drives. Almost all AT BIOSs support a maximum of two fixed disk drives. Most systems have a combination diskette/fixed disk controller.

Fixed disk basics

A fixed disk is an electromechanical device. The electronic part includes the read/write head(s), which are essentially electromagnets that convert rapidly pulsing digital electronic signals from the computer to semi-permanent magnetic fields on a specific location on the disk surface. There are other electronic components that control the mechanical parts. They properly align the magnetic storage and help to locate the information on the disk surface.

The mechanical part of a fixed disk drive includes a spindle (a shaft that connects directly to the disk drive motor), and disk platters (upon which data is actually written). The motor spins the disk platter at a precisely controlled speed. The only other mechanical part is the arm, upon which the read/write heads are mounted. The arm allows the heads to move freely just above the disk surface so that the heads can be placed anywhere between the inner and outermost part of the surface. Fixed disks usually have many read/write heads and both sides of the disk platters are used. There is one read/write heads positioned just above the surface of each side of the disk platter. The graphic on the following page describes these terms.

continued

Fixed disk terms

Some basic terms necessary to the understanding of a fixed disk subsystem include the following:

- **Head**
  The head contains an electromagnet, positioned on a movable assembly just above the surface of the diskette. By pulsing the electromagnet, the head reads or writes data from or to the disk platter surface.

- **Track**
  When the head is positioned over a point on the fixed disk, ready to read or write, the disk platter surface spins underneath it, tracing a full circle. This circle is a track. There may be 40 to 150 tracks per disk surface.

- **Sector**
  Fixed disk systems divide each track into short arcs (usually 17, 26, or 34 per track) called sectors. Each sector usually holds 512 bytes of data.

- **Cylinder**
  Each head on the fixed disk drive (there may be many heads) traces out a separate circle (track) across the platter of the fixed disk it rides above. The combination of all of the tracks traced out by the head at a given read/write head position forms the outline of a solid cylinder, thus such a vertical stack of tracks is called a cylinder.

continued

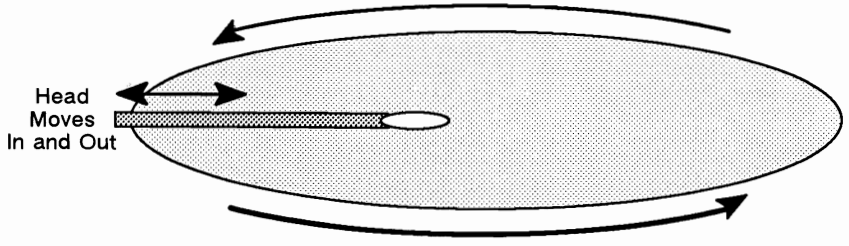
Fixed disk drive organization

The following graphic explains fixed disk tracks and sectors.

Heads and surfaces

Fixed disk drives generally have several magnetic "heads" (the mechanisms that allow them to read/write the encoded bits on the fixed disk platter surfaces). One head is used for each side of each disk platter. A fixed disk can have many disk platters.

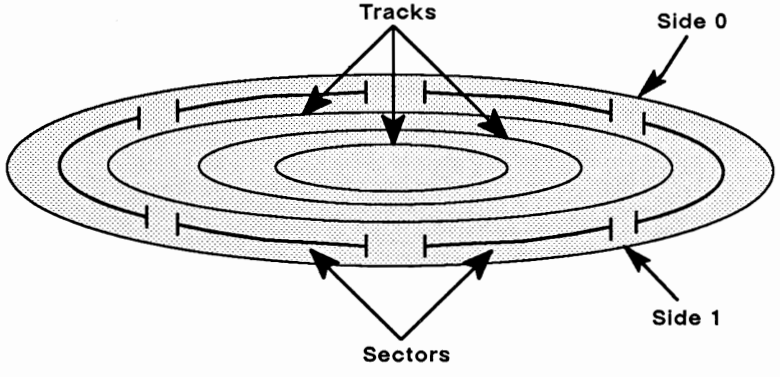
All sides of the fixed disk platter surfaces are covered with a material that can be magnetized and is used to store information.



Tracks and sectors

The fixed disk head rapidly magnetizes areas on the surface of the disk platter as the platter spins, representing binary zeros and ones. These areas are organized into concentric circles (tracks). There can be 40, 80, or even more tracks per fixed disk side.

Tracks are further subdivided into sectors. Sectors usually store 512 bytes, but the sector size can also be changed with INT 13h, AH = 05h. The bytes in a sector must be organized in a prescribed manner. There can be 17, 26, 34 or more sectors per track.



continued

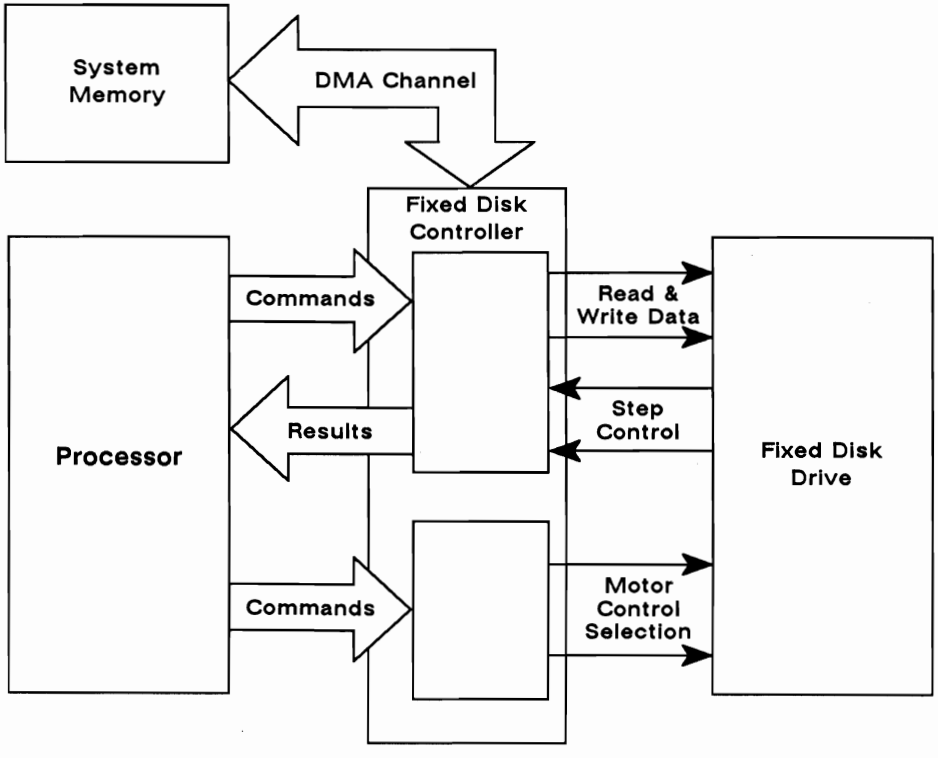
Fixed disk processing

Like the Diskette Service, the BIOS Fixed Disk Service programs the fixed disk controller directly. The BIOS never writes directly from the processor to the fixed disk drive. Sometimes data and commands are sent directly to the controller from the microprocessor, but more often, the BIOS uses the DMA controller.

The BIOS uses the DMA controller to move data directly to and from system memory without the intervention of the microprocessor.

The fixed disk controller communicates directly between the fixed disk drive, the processor, and system memory. It uses DMA channel 2 to access memory.

The following graphic depicts the relationship between the processor, system memory, the DMA channel, the fixed disk controller, and the fixed disk drive.



continued

Fixed disk controller interfaces

The computer does not access the fixed disk directly. The computer communicates with the fixed disk controller which controls the fixed disks and often also the diskettes. The fixed disk controller uses a predetermined standard for formatting instructions and data (called an interface) for accessing the fixed disk. There are three widely used fixed disk controller standards (or interfaces):

- ST506/ST412 — the standard interface
- ESDI (enhanced small device interface), and
- SCSI (small computer systems interface).

Fixed disk data encoding methods

On the ST506 interface, the disk-generated pulses that are used to represent the actual data are passed directly to the controller and the way these pulses are arranged can make a big difference in fixed disk drive performance.

There are several methods of data encoding used by ST506 fixed disk controllers. The most frequently used methods are:

- MFM — modified frequency modulation,
- RLL — run length limited, and
- ARLL — advanced run length limited

Encoding methodologies and the BIOS

The BIOS Fixed Disk Service does not need to worry about different types of encoding or interface methods for fixed disk drives. The BIOS uses the fixed disk drive parameters stored in the fixed disk drive type table in the ROM BIOS data area. See the ROM BIOS Data heading later in this chapter.

continued

Fixed disk drive types and the user

XT fixed disk controllers support a limited number of fixed disk drives, so an XT system user has a limited choice of fixed disk drive types. Only four drive types are available. See the ROM BIOS Data heading in this chapter for a description of the XT fixed disk drive types.

In an AT system, the BIOS or computer manufacturer changes the drive types (1–47) and fixed disk parameter table entries to correspond to popular types of fixed disk drives that use the various interfaces and encoding methods. Therefore, many popular types of fixed disk drives that use MFM, RLL, ARLL, or ESDI are available to the end user. The user can access the fixed disk drive type table in the ROM BIOS Data Area and select a fixed disk drive type that corresponds to the parameters of the system's fixed disk drive.

The end user can also use a BIOS utility program, such as the Phoenix SETUP utility, to change the fixed disk drive type table to ensure that the fixed disk drive is supported.

Setting up a fixed disk drive

In systems with XT fixed disk controllers, installing a fixed disk drive requires only physical installation and no software setup.

In AT systems, a SETUP utility must be used. The user needs to know several facts about the fixed disk drive, including the:

- cylinder write precompensation value,
- number of heads,
- number of cylinders,
- number of sectors per track,
- landing zone value, and
- capacity.

The user can look up the fixed disk drive type table stored in the system BIOS (usually the SETUP help facility provides this information) and find the drive type that matches the characteristics of his fixed disk drive.

Some BIOSs provide editable drive types. Phoenix BIOSs provide types 48 and 49, which the user can modify by entering the parameter values of the fixed disk drive with the SETUP utility. The data is automatically stored in CMOS RAM.

Support for two fixed disk drives

The BIOS Fixed Disk Service supports up to two fixed disk drives. An ST412/506 interface fixed disk controller or equivalent is assumed present. The BIOS uses information from the drive table which may include support for RLL and ESDI drives. ESDI drives generally require an option ROM, which replaces part of the ROM BIOS data.

Older XT-type fixed disk controllers may support only one fixed disk drive. These controllers may be found in both AT and XT systems.

How fixed disks are identified

There are only two valid fixed disk identification numbers — 80h and 81h. The numbers must be assigned as follows:

- 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.

BIOS allows processing while disk activity is performed

The BIOS issues an INT 15h, AH = 90h Device Busy (AL = 00h fixed disk) call while waiting for fixed disk interrupts, informing the operating system that the BIOS is waiting for the fixed disk hardware, so processing can go on. When the fixed disk interrupt occurs, the BIOS issues an INT 15h, AH = 91h Interrupt Complete (AL = 00h fixed disk), indicating that the fixed disk operation is complete.

System RAM Data

Introduction

The BIOS data area (address 400h – 500h) contains data definitions that are referred to by the BIOS 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 (AT only)
40:76h	1 Byte	Control byte (when to write to I/O port 03F6h).
40:77h	1 Byte	Fixed disk port offset
40:8Ch	1 Byte	Fixed disk controller status (AT only)
40:8Dh	1 Byte	Fixed disk controller error status (AT only)
40:8Eh	1 Byte	Fixed disk interrupt flag (AT only)

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 RAM 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 controller fails initialization Bit 2 = 1 RTC time invalid Bit 1 = 1 Controllers do not match configuration Bit 0 = 1 Time-out in reading an controller ID
11h	1 Byte	Type of fixed disk drive 0
12h	1 Byte	Type of fixed disk drive 1
19h	1 Byte	Drive type of fixed disk 1
1Ah	1 Byte	Drive type of fixed disk 2

AT fixed disk parameter table structure

The 16-byte AT fixed disk parameter table defines the types of fixed disk drives that can be used in an AT-compatible system. The address of the correct entry within the fixed disk parameter table is contained in the INT 41h vector for drive 0 and in the INT 46h vector for drive 1. The INT 41h fixed disk parameter table is usually located at F000:E401h. The default location of the INT 46h table is often the same as the INT 41h vector. By invoking INT 13h AH = 08h Read Drive Parameters, the caller can find out where the fixed disk parameter table is located.

Offset	Size	Description
00h	2 Bytes	Maximum number of cylinders
02h	1 Byte	Maximum number of heads
03h	2 Bytes	Reserved
05h	2 Bytes	Starting write precompensation cylinder
07h	1 Byte	Reserved
08h	1 Byte	Control byte, where: Bit 7 = 1 Disable retries Bit 6 = 1 Disable retries Bit 5 = 1 Defect map present at maximum 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 cylinder
0Eh	1 Byte	Number of sectors per track
0Fh	1 Byte	Reserved

continued

XT fixed disk parameter table structure

The XT fixed disk parameter table defines the types of fixed disk drives that can be used in an XT-compatible system. The address of the correct entry within the fixed disk parameter table is contained in the INT 41h vector. The fixed disk parameter table is usually located at F000:E401h. By invoking INT 13h, AH = 08h Read Drive Parameters, the caller can find out where the fixed disk parameter table is located.

Offset	Size	Description
00h	2 Bytes	Maximum number of cylinders
02h	1 Byte	Maximum number of heads
03h-04h	2 Bytes	Starting reduced write current cylinder
05h-06h	2 Bytes	Starting write precompensation cylinder
07h	1 Byte	Maximum ECC burst length
08h	1 Byte	Control byte, where: Bit 7 = 1 Disable retries Bit 6 = 1 Disable ECC retries Bits 5-3 = 0 Reserved Bits 2-0 = Drive option
09h	2 Bytes	Standard timeout value
0Ah	1 Byte	Timeout value for format drive command
0Bh	1 Byte	Timeout value for check drive command
0Ch	4 Bytes	Reserved

continued

AT fixed disk drive type table

The fixed disk drives that the AT BIOS supports are stored in a ROM-based table. In the Phoenix AT BIOS, table entries 48 and 49 are left for the end user to define by running SETUP. Once specified, the drive characteristics remain stored in CMOS RAM. The end user can add any type of AT-compatible fixed disk drive to a system in this manner. This list is by no means inclusive, but is merely the default drive types provided by Phoenix. Some manufacturers define many more default drive types; others provide means for user definition of drive types.

The Phoenix default drive type table is shown below. We have included the names of many popular fixed disk drive types for easy identification.

#	Drive Type	Cylinders	Heads	Write Precomp	Landing Zone	Sectors
1	IBM 5.25" 10 MB	306	4	128	305	17
2	IBM 5.25" 20 MB Seagate ST-225 CDC Wren II 9415-5-25 * Miniscribe 8438F	615	4	300	615	17
3	IBM 5.25" 30 MB	615	6	300	615	17
4	IBM 5.25" 62 MB	940	8	512	940	17
5	IBM 5.25" 46 MB	940	6	512	940	17
6	IBM 5.25" 20 MB Miniscribe MS 8425 Tandon TM 262 *** Tandon TM 702AT ***	615	4	-1	615	17
7	IBM 5.25" 30 MB	462	8	256	511	17
8	IBM 5.25" 30 MB Seagate ST-4038 *** CDC Wren II 9415-5-38 * *** Tandon TM 703AT ***	733	5	-1	733	17
9	IBM 5.25" 112 MB Priam IDED 130*	900	15	-1	901	17

* The OEM may have to disable drive connector J1 pin 2 when used on the AT.
** Remove jumper W3.
*** Values for Write Precomp of -1, 0, or the last track are assumed to be equivalent.

continued

AT fixed disk drive type table, cont'd

#	Drive Type	Cylinders	Heads	Write Precomp	Landing Zone	Sectors
10	IBM 5.25" 20 MB Priam IDED75, 100, 120, 150, 160, 230, 330*	820	3	-1	820	17
11	IBM 5.25" 35 MB Priam IDED 40, 45, 45H Priam ID45T-S, ID45T-Q*	855	5	-1	855	17
12	IBM 5.25" 49 MB	855	7	-1	855	17
13	IBM 5.25" 20 MB	306	8	128	319	17
14	IBM 5.25" 42 MB	733	7	-1	733	17
15	Reserved					
16	IBM 5.25" 20 MB	612	4	0	663	17
17	IBM 5.25" 40 MB	977	5	300	977	17
18	IBM 5.25" 56 MB Priam ID 60, 62*	977	7	-1	977	17
19	IBM 5.25" 5 9MB	1024	7	512	1023	17
20	IBM 5.25" 30 MB	733	5	300	732	17
21	IBM 5.25" 42 MB	733	7	300	732	17
22	IBM 5.25" 30 MB	733	5	300	733	17
23	IBM 5.25" 10 MB	306	4	0	336	17
24	Priam IDED 62	1024	7	-1	1024	17
25	NEC 5126 20 MB	615	4	0	615	17
26	Micropolis 1323 *** Rodime 5040	1024	4	-1	1024	17
27	Micropolis 1323A *** Seagate ST-4077R Miniscribe 3053/6053 Priam IDED 45, 45H**	1024	5	-1	1024	17

* The OEM may have to disable drive connector J1 pin 2 when used on the AT.
** Remove jumper W3.
*** Values for Write Precomp of -1, 0, or the last track are assumed to be equivalent.

continued

AT fixed disk drive type table, cont'd

#	Drive Type	Cylinders	Heads	Write Precomp	Landing Zone	Sectors
28	Micropolis 1325 *** Miniscribe 6085 Maxtor XT-1085 ***	1024	8	-1	1024	17
29		512	8	256	512	17
30	Syquest SQ312RD ** Miniscribe MS 8212 ***	615	2	615	615	17
31	CDC Wren II 9420-5-51 **	989	5	0	989	17
32		1020	15	-1	1024	17
33	Reserved					
34	Reserved					
35	Seagate ST-4096 Seagate ST-4144 Seagate ST-4144R	1024	9	1024	1024	17 26
36	Miniscribe MS 6053 Microscience HH1050	1024	5	512	1024	17
37		830	10	-1	830	17
38	NEC 67 MB	823	10	256	824	17
39	NEC D5126H 20 MB	615	4	128	664	17
40	IBM 5.25" 40 MB NEC D5146H	615	8	128	664	17
41	IBM 5.25" 114 MB	917	15	-1	918	17
42	IBM 5.25" 127 MB Priam IDED 130**	1023	15	-1	1024	17
43	NEC	823	10	512	823	17
44	Seagate ST-251 40 MB	820	6	-1	820	17
45	Reserved					
46	CDC Wren II (1) 9415-5-86	925	9	-1	925	17

* The OEM may have to disable drive connector J1 pin 2 when used on the AT.
** Remove jumper W3.
*** Values for Write Precomp of -1, 0, or the last track are assumed to be equivalent.

continued

AT fixed disk drive type table, cont'd

#	Drive Type	Cylinders	Heads	Write Precomp	Landing Zone	Sectors
47		699	7	256	700	17
48	Configurable by end user					
49	Configurable by end user					

* The OEM may have to disable drive connector J1 pin 2 when used on the AT.
** Remove jumper W3.
*** Values for Write Precomp of -1, 0, or the last track are assumed to be equivalent.

If a table entry contains -1 as the entry for the beginning write precompensation cylinder, then there is no write precompensation for this fixed disk drive type. If the write precompensation is zero, then there is write precompensation for all cylinders.

Drive types 48 and 49 in a Phoenix BIOS can be edited by the Phoenix SETUP utility. Parameter values entered in SETUP for these drives are stored in CMOS RAM.

continued

XT fixed disk drive types

Systems with an XT-type fixed disk controller support only four fixed disk drive types. The fixed disk parameter values for each of these XT drive types are defined below. Some AT systems may have XT-type fixed disk controllers.

Parameter	Type 0	Type 1	Type 2	Type 3
Maximum cylinders	306	612	615	306
Maximum head(s)	4	4	4	8
Reduced write current cyl.	306	612	615	306
Write precomp cylinder	0	0	300	128
ECC Data Burst Length	11	11	11	11
Control byte	5	5	5	5
Standard time-out	12	32	24	12
Format drive time-out	180	180	180	180
Check drive time-out	40	40	40	40
Reserved	0	0	0	0
Reserved	0	0	0	0
Reserved	0	0	0	0
Reserved	0	0	0	0

Fixed Disk Service I/O Ports

The INT 13h Fixed Disk Service references the following I/O ports:

I/O Address	Read/Write Status	Description
0020h	R	Programmable Interrupt Controller — 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
0020h	R	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	Programmable Interrupt Controller — 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

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 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

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 in-service 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
00A0h	R/W	Programmable Interrupt Controller 2
00A1h	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
0170h	R/W	Fixed disk 1 data register (AT only)
0171h	R/W	Fixed disk 1 error register (AT only)
0172h	R/W	Fixed disk 1 sector count (AT only)
0173h	R/W	Fixed disk 1 sector number (AT only)
0174h	R/W	Fixed disk 1 cylinder low (AT only)
0175h	R/W	Fixed disk 1 cylinder high (AT only)
0176h	R/W	Fixed disk 1 drive/head register (AT only)
0177h	R/W	Fixed disk 1 status register (AT only)
01F0h	R/W	Fixed disk 0 data register (AT only)
01F1h	R/W	Fixed disk 0 error register (AT only)

continued

Fixed Disk Service I/O Ports, Continued

I/O Address	Read/Write Status	Description
01F2h	R/W	Fixed disk 0 sector count (AT only)
01F3h	R/W	Fixed disk 0 sector number (AT only)
01F4h	R/W	Fixed disk 0 cylinder low (AT only)
01F5h	R/W	Fixed disk 0 cylinder high (AT only)
01F6h	R/W	Fixed disk 0 drive/head register (AT only)
01F7h	R/W	Fixed disk 0 status register (AT only)
0320h	R/W	Fixed Disk controller Register (8 or 16 bit)

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 codes

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

Functions which have not executed successfully return with

- AH and Fixed Disk Status (40:74h) set to one of 23 possible error codes
- Carry Flag set

continued

Table of 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 failed
80h	Time-out
AAh	Drive not ready or not selected
BBh	Undefined error
CCh	Write fault on selected drive
E0h	Status error/error register is 0
FFh	Sense operation failed

Function: AH = 00h Reset Diskette(s) and Fixed Disk [XT] [AT]

Description

This 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 to cylinder 0.

The fixed disk drive number (either 80h or 81h) is specified in DL. If successful, this function returns with AH set to 00h and the 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
= xxh Error (See error codes earlier in this chapter)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Differentiating diskette from fixed disk errors

In AT systems, 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 Fixed Disk Status [XT] [AT]

Description

This 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 00h and clears the Carry Flag.

Error code 11h, ECC Data Error, indicates that a recoverable error was detected during a preceding call to Function 02h Read Sectors. See the heading Error Handling earlier in this chapter for a complete list of all codes that may be returned in AL.

Input/Output

Input:	AH	=	01h
	DL	=	Drive number
			80h Fixed disk 1
			81h Fixed disk 2
Output:	AH	=	00h No error
		=	xxh Error (See error codes earlier in this chapter)
		=	Fixed Disk Status (40:74h)
	AL	=	Error code for last operation

Function: AH = 02h Read Sectors [XT] [AT]

Description

This function reads the number of sectors specified in AL from the drive specified in DL to a buffer area defined by ES:BX. This function also reads the two words of Error Correction Code (ECC) associated with that sector and automatically corrects ECC errors. Multisector transfers are terminated after any sector that has a read error. Cylinder number, sector number, and head number are specified in CH, CL, and DH, respectively (see Input/Output below).

If successful, this function returns with the Carry Flag cleared and AH = 00h.

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

10 bits = 1024 CYL

MAX 64 SECTORS

MAX 256 HEADS

Output:

AH	=	00h No error
	=	xxh Error (See error codes earlier in this chapter)
	=	Fixed Disk Status (40:74h)
AL	=	Number of sectors of data transferred
CF	=	0 No error
	=	1 Error

continued

1024 × 64 × 256

= 16,777,216

= 8.4 GB

Error conditions

If it is unsuccessful, this 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 in AL, this 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.

The caller should retry this function at least three times, issuing the Reset Function (AH = 00h) between retries, when an error condition occurs to ensure that the error is not just a hardware timing problem.

Function: AH = 03h Write Sectors [XT] [AT]

Description

This 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 under the Input/Output heading below.

This function does not require a prior call to the Seek Function (AH = 0Ch).

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.

If an error occurs when this function is invoked, issue a Function 00h to reset the fixed disk controller, and retry this function at least three times.

If successful, this function returns with the Carry Flag cleared and AH = 00h.

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	=	00h No error
	=	xxh Error (See error codes earlier in this chapter)
	=	Fixed Disk Status (40:74h)
AL	=	Number of sectors of data transferred
CF	=	0 No error
	=	1 Error

continued

Error conditions

If unsuccessful, this 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).

See the Error Handling heading earlier in this chapter for a list of possible errors.

For example, if an invalid number of sectors is in AL, this function returns with 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 Sectors [XT] [AT]

Description

This 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.)

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. This function does not cause any data to be transferred from memory to fixed disk or vice versa.

If successful, this function returns with the Carry Flag cleared and AH and Fixed Disk Status (40:74h) set to 00h.

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
	=	xxh Error (See error codes earlier in this chapter)
	=	Fixed Disk Status (40:74h)
AL	=	Number of sectors verified
CF	=	0 No error
	=	1 Error

Error conditions

If unsuccessful, this 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 Handling heading in this chapter for a table of error codes.

Function: AH = 05h Format Cylinder [XT] [AT]

Description

This function formats the cylinder specified in CH and CL using the head specified in DL.

If the system uses an XT-type fixed disk controller, INT 13h, AH = 0Fh should be called before invoking this function.

Input/Output

Input:

AH	=	05h
AL	=	Interleave factor (XT-type fixed disk controllers only)
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 field entries

Output:

AH	=	00h No error
	=	xxh Error (See error codes earlier in this chapter)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

continued

Buffer table contents

The caller must provide a 512-byte 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

Example: Address Field Table

For example, the address field to format a track on a fixed disk drive that has an interleave factor of two and that is formatted for 17 sectors per track would be:

```
db 00, 01, 00, 0A, 00, 02, 00, 0B, 00, 03, 00, 0C, 00, 04, 00, 0D
db 00, 05, 00, 0E, 00, 06, 00, 0F, 00, 07, 00, 10, 00, 08, 00, 11
db 00, 09
```

Error conditions

If unsuccessful, this 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 a time-out error occurs during the execution of this function, error code 80h is returned in both AH and Fixed Disk Status.

If an error occurs when this function is invoked, issue a Function 00h to reset the fixed disk controller and retry this function at least three times.

Refer to the Error Handling heading in this chapter for a table of error codes.

Function: AH = 06h Format Bad Track [XT]

Description

This function initializes a track, writing fixed disk address fields and data sectors and setting bad sector flags. It is designed for use with an XT-type fixed disk controller.

For drive 80h, this function references the fixed disk parameter table pointed to by the Interrupt 41h vector. If successful, this function returns with AH and 40:74h set to zero.

Input/Output

Input:

AH	=	06h
AL	=	Interleave factor
CH	=	Cylinder number
DH	=	Head number
DL	=	Drive number
		80h fixed disk 1
		81h fixed disk 2

Output:

AH	=	07h Drive number in DL is invalid
	=	01h Invalid function request
	=	00h If operation is successful
	=	xxh Error (See error codes earlier in this chapter)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

Error conditions

If an error occurs when this function is invoked, issue a Function 00h to reset the fixed disk controller, and retry this function at least three times.

If the drive number specified in DL is not valid, this function returns with AH and location 40:74h set to 07h to indicate that the drive parameter activity failed. AL, CX, and DX are cleared, the Carry Flag is set, and control is returned to the caller.

Function: AH = 07h Format Drive [XT]

Description

This function formats the entire fixed disk drive, writing disk address fields and data sectors, starting at the specified cylinder. This function works only with XT-type fixed disk controllers.

If successful, this function returns with AL and location 40:74h set to 00h.

Input/Output

Input: AH = ~~08h~~ 07
AL = Interleave factor
CH = Cylinder
DL = Drive number
80h fixed disk 1
81h fixed disk 2

Output: AH = 07h If drive number in DL is invalid
= 00h No error
= xxh Error (See error codes earlier in this chapter)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If an error occurs when this function is invoked, issue a Function 00h to reset the fixed disk controller, and retry this function at least three times.

If the drive number specified in DL is not valid, this function returns with AH and location 40:74h set to 07h to indicate that the drive parameter activity failed. AL, CX, and DX are cleared, the Carry Flag is set, and control is returned to the caller.

Function: AH = 08h Read Drive Parameters [XT] [AT]

Description

This function returns parameters associated with the fixed disk drive (either 80h or 81h) specified in DL.

For drive 80h, this function references the fixed disk parameter table pointed to by the INT 41h vector. For drive 81h, the function references the INT 46h vector. See the ROM BIOS Data heading in this chapter for the structure and contents of the fixed disk parameter table.

If successful, this function returns with AL and Fixed Disk Status set to 00h. The maximum usable cylinder number is returned in CH/CL, maximum usable sector number in CL, maximum usable head number in DH, the number of fixed disk drives contained in the system is returned in DL, and the address of the fixed disk parameter table is returned in ES:DI.

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 = xxh Error (See error codes earlier in this chapter) = Fixed Disk Status (40:74h)
AL	=	00h
CF	=	0 No error = 1 Error
CH	=	Maximum usable cylinder number (low 8 bits, zero-based) (00h if AH = 07h)
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
DL	=	Number of drives (zero-based, 00h if AH = 07h)
ES:DI	=	Address of fixed disk parameter table

continued

Error conditions

If the drive number specified in DL is not valid, this 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 cleared, the Carry Flag is set, and control is returned to the caller.

Function: AH = 09h Initialize Drive Parameters [XT] [AT]

Description

This function initializes the controller associated with the fixed disk drive (either 80h or 81h) specified in DL.

For drive 80h, this function references the fixed disk parameter table pointed to by the Interrupt 41h vector. For drive 81h, the function uses the interrupt 46h vector. See the ROM BIOS Data heading in this chapter for the structure and contents of the fixed disk parameter table.

If successful, this 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
= xxh Error (See error codes earlier in this chapter)
= 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 Sectors [XT] [AT]

Description

This function reads one or more sectors from the fixed disk specified in DL. This function also reads the four to seven bytes of Error Correction Code (ECC) associated with that sector, but does not automatically correct ECC errors (Function 02h automatically corrects ECC errors). Multisector transfers are terminated after any sector that has a read error. Cylinder number, sector number, and head number are specified in CH, CL, and DH, respectively.

This function does not require a prior call to the Seek Function (AH = 0Ch).

This function is normally reserved for diagnostics and should not be used for normal reads from the fixed disk.

If successful, this function returns with the Carry Flag cleared and AH and Fixed Disk Status (40:74h) equal to zero.

Input/Output

Input:

AH	=	0Ah
AL	=	Number of sectors (usually one)
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
	=	xxh Error (See error codes earlier in this chapter)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

continued

Error conditions

If unsuccessful, this 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 01h, this function returns with AH and Fixed Disk Status (40:74h) set to 01h (the invalid parameter error code) and the Carry Flag set.

Refer to the Error Handling heading in this chapter for a table of error codes.

Function: AH = 0Bh Write Long Sectors [XT] [AT]

Description

This function writes one or more sectors to the location on the fixed disk specified in DL from the buffer pointed to by ES:BX. This function also writes four to seven bytes of Error Correction Code (ECC) associated with the specified sectors.

Cylinder number, sector number, and head number are specified in CH, CL, and DH, respectively (see Input/Output below).

This function does not require a prior call to Function 0Ch, Seek to Cylinder.

Along with Function AH = 0Ah, this function is normally reserved for diagnostics and should not be used for normal write to fixed disk operations.

If successful, this function returns with the Carry Flag cleared and AH and Fixed Disk Status (40:74h) equal to zero.

Input/Output

Input:

AH	=	0Bh
AL	=	Number of sectors to write (usually one)
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
	=	xxh Error (See error codes earlier in this chapter)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

continued

Error conditions

If unsuccessful, this 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 Handling heading in this chapter for a table of error codes.

Function: AH = 0Ch Seek to Cylinder [XT] [AT]

Description

This 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, this function returns with the Carry Flag cleared and AH and Fixed Disk Status (40:74h) equal to zero.

The Read Disk Sectors (AH = 02h), Write Disk Sectors (AH = 03h), Read Long Sectors (AH = 0Ah), and Write Long Sectors (AH = 0Bh) functions have a seek operation implicitly built into them and do not require a prior call to 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
	=	xxh Error (See error codes earlier in this chapter)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Error

Error conditions

If unsuccessful, this 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 Handling heading in this chapter for a table of error codes.

Function: AH = 0Dh Alternate Fixed Disk Reset [XT] [AT]

Description

This function is identical to INT 13h AH = 00h Reset Diskette(s) and Fixed Disk except that the diskette system is not reset.

The caller must specify the fixed disk drive to reset in DL. This 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. This process resets the read/writer head to track 0.

If successful, this 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
= xxh Error (See error codes earlier in this chapter)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If not successful, this 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 Handling heading in this chapter for a table of error codes.

Function: AH = 0Eh Diagnostics 1: Read Test Buffer [XT]

Description

This function reads a test buffer from the fixed disk controller into the diagnostics buffer specified in ES:BX. Data is not read from the actual physical disk drive.

If successful, this 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
= xxh Error (See error codes earlier in this chapter)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If not successful, this 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 Handling heading in this chapter for a table of error codes.

Function: AH = 0Fh Diagnostics 2: Write Test Buffer [XT]

Description

This function writes a test buffer to the controller from the diagnostics buffer specified in ES:BX. Data is not written to the actual physical disk drive.

This function should be called to initialize the sector buffer contents before formatting an XT-type fixed disk drive using INT 13h AH = 05h.

If successful, this 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
		= xxh Error (See error codes earlier in this chapter)
		= Fixed Disk Status (40:74h)
	CF	= 0 No error
		= 1 Error

Error conditions

If not successful, this 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 Handling heading in this chapter for a table of error codes.

Function: AH = 10h Test for Drive Ready [XT] [AT]

Description

This function determines if the fixed disk drive specified in DL is ready and can process a command.

If successful, this 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
= xxh Error (See error codes earlier in this chapter)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If not successful, this 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 Handling heading in this chapter for a table of error codes.

Function: AH = 11h Recalibrate Drive [XT] [AT]

Description

This function repositions head 0 over cylinder 0 of the fixed disk drive specified in DL.

If successful, this 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
= xxh Error (See error codes earlier in this chapter)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If not successful, this 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 Handling heading in this chapter for a table of error codes.

Function: AH = 12h Controller RAM Diagnostic [XT]

Description

This function causes the fixed disk controller to carry out a built-in diagnostic test on its internal sector buffer, indicating whether the test was passed by a status indicator in AH.

Input/Output

Input:

AH	=	12h
AL	=	Number of sectors
CH	=	Cylinder
CL	=	Sector
DH	=	Head
DL	=	Drive number
		80h fixed disk 1
		81h fixed disk 2

Output:

AL	=	00h
AH	=	00h No error
	=	xxh Error (See error codes earlier in this chapter)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Invalid parameter

Error conditions

If not successful, this function returns with the appropriate Fixed Disk Service error code in both AH and location 40:74h and the Carry Flag (CF) set.

Refer to the Error Handling heading in this chapter for a table of error codes.

Function: AH = 13h Controller Drive Diagnostic [XT]

Description

This function causes the fixed disk controller to run internal diagnostic tests of the attached fixed disk drive. The value in AH indicates whether the test was passed, and, if not, gives an appropriate error code in AH. If there is an error the Carry Flag is set.

Input/Output

Input:

AH = 13h
AL = Number of sectors
CH = Cylinder
CL = Sector
DH = Head
DL = Drive number
80h fixed disk 1
81h fixed disk 2

Output:

AL = 00h
AH = 00h No error
= xxh Error (See error codes earlier in this chapter)
= Fixed Disk Status (40:74h)
CF = 0 No error
= 1 Error

Error conditions

If not successful, this function returns with the appropriate Fixed Disk Service error code in both AH and location 40:74h and the Carry Flag (CF) set.

Refer to the Error Handling heading in this chapter for a table of error codes.

Function: AH = 14h Controller Internal Diagnostic [XT] [AT]

Description

This function causes the fixed disk controller to carry out a built-in diagnostic self-test, indicating whether the test was passed by a status code in AH.

If an illegal parameter is passed in one of the input registers, the Carry Flag is set and control is returned to the caller with registers preserved.

Input/Output

Input:

AH	=	14h
AL	=	Number of sectors
CH	=	Cylinder
CL	=	Sector
DH	=	Head
DL	=	Drive number
		80h fixed disk 1
		81h fixed disk 2

Output:

AL	=	00h
AH	=	00h No error
	=	xxh Error (See error codes earlier in this chapter)
	=	Fixed Disk Status (40:74h)
CF	=	0 No error
	=	1 Invalid parameter

Error conditions

If not successful, this function returns with the appropriate Fixed Disk Service error code in both AH and location 40:74h and the Carry Flag (CF) set.

Refer to the Error Handling heading in this chapter for a table of error codes.

Function: AH = 15h Read Fixed Disk Type [AT]

Description

This function returns a status code in AH which indicates the type of diskette or fixed disk drive for the drive specified in DL. This function also 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 of the number of 512-byte sectors on the fixed disk is returned in CX, and the low word of the number of 512-byte sectors is returned in DX. The Fixed Disk Status (40:74h) and AH are set to 00h, the Carry Flag is cleared, 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
		=	xxh Error (See error codes earlier in this chapter)
		=	Fixed Disk Status (40:74h)
	CF	=	0 No error
		=	1 Error
	CX	=	Number of 512-byte sectors on the fixed disk (valid only of AH = 03h)

continued

AH reports drive number validity

For drive 80h, this 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 this 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 in DL is valid (AH will equal 00h if there is no drive for this drive number), or the fixed disk/diskette status of the drive specified in DL.

Error conditions

If the drive number specified in DL is not valid, this function clears AH, AL, CX, DX, and the Fixed Disk Status (40:74h), clears the Carry Flag, and returns control to the caller.

12

Chapter 12

INT 14h Serial Communications Service

Overview

Description

The BIOS Serial Communications Service performs RS-232C character I/O on IBM-compatible serial port adapters.

How the Serial Communications Service is invoked

Software INT 14h invokes the Serial Communications Service.

The INT 14h vector resides at address 00:50h in the interrupt vector table. The BIOS initializes the INT 14h vector to address F000:E739h.

continued

Summary of Serial Communications functions

The Serial Communications Service contains four functions. Individual functions are selected via the AH register. The table below summarizes the Serial Communications Service functions.

Function	Description	Platforms
00h	Initialize Serial Communications Port	[XT] [AT]
01h	Send Character	[XT] [AT]
02h	Receive Character	[XT] [AT]
03h	Read Serial Port Status	[XT] [AT]
04h-FFh	Reserved	Reserved

In this chapter

This chapter focuses on the BIOS Serial Communications Service. The following topics are discussed:

- Theory of Operation
- System RAM Data
- ROM BIOS Data
- Serial Communications I/O Ports
- Error Conditions
- Serial Communications Functions

How to determine level of support

Legend	Description
[PC]	Function is supported by original IBM PC BIOS only.
[XT]	Function is supported by all IBM PC/XT and compatible ROM BIOSs. **Note:** This level of support is contained in the ROM BIOS implemented in the majority of IBM PC and PC/XT compatible systems.
[AT]	Function is supported by all IBM PC/AT and compatible ROM BIOSs.

Theory of Operation

Introduction

The asynchronous serial port on an IBM PC, PC/XT, and PC/AT converts data from parallel form to serial form for communication between the computer and an external device, usually a serial printer or an external modem. The physical connection consists of an RS-232C-standard cable linking the two devices. The ROM BIOS Serial Communications Service, which is invoked via INT 14h, performs various operations that make serial transmission and reception possible.

Parallel and serial transmission

In a parallel connection, all the individual bits of a data byte are simultaneously transmitted over the parallel port. This kind of connection exists between a computer and a parallel printer. When data is transmitted serially, however, all the bits of a data byte are sequentially transmitted over the serial port. For example, during serial transmission, a parallel data byte of 8 bits would be converted to a serial data byte of 8 consecutive data bits, delivered one at a time. The receiving device then assembles these individual bits into a parallel data byte.

continued

Data sent as packets of frames

Since asynchronous serial transmission sends individual data bits along the I/O path, there has to be some way for the receiver to distinguish these bits and assemble them into separate characters. This is achieved through the serial controller, which packages each data byte into a separate frame. Each frame consists of a start bit, the actual data byte, an optional parity bit, and a stop bit(s).

The start bit is automatically inserted by the controller before it sends a character. This is the first bit that the receiving device gets, and it tells the device that a character is about to follow. The data byte, which can be a 5-, 6-, 7-, or 8-bit character, follows the start bit. If the controller is programmed to add an error-detecting parity bit, this comes next. Finally, the stop bits complete the frame. These tell the receiving device that this is the end of a character and another one is about to begin. The controller can be programmed to insert 1, 1.5, or 2 stop bits.

Sending characters as packets of frames allows the receiving device to recognize when a particular character begins and ends, so that these characters can be distinguished and joined into words.

NS 16450 IC Controller

The Serial Communications Service assumes a National Semiconductor NS 16450 IC serial communications controller (or compatible) is in place. The controller contains a programmable baud rate generator that supports baud rates from 50 baud to 9600 baud. The controller 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.

continued

RS-232C interface standard

All IBM and compatible serial controllers use a predetermined standard (called an interface) for sending and receiving serial data. This interface standard is called the RS-232C. The RS-232C interface standard defines various connection parameters, including the connector type, I/O signal levels, line names, and pin numbers.

The cable that links the computer's serial port to the external device must conform to EIA RS-232C (or equivalent) standards.

Programming the controller

The serial communications controller is responsible for performing a host of operations on the incoming or outgoing serial data stream. For one, the controller adds or removes bits as described above. In addition, the controller controls the speed at which data gets sent across the the data path. Also, the controller communicates with the external device through signals that inform the two devices about their readiness to send and receive data (see I/O signals on the next page).

All of these operations can be modified by programming the controller. The controller can be programmed by manipulating the serial communications I/O ports.

Reference: A complete list of the serial I/O ports can be found under the heading Serial Communications I/O Ports in this chapter.

continued

I/O signals

The controller communicates with the external device through input/output signals. The input signals are generated by the external device, and monitored by the modem status register. The output signals are generated by the controller, and controlled by bits 3-0 in the modem control register.

These signals can be set active or inactive by programming their respective register bits. The following table describes these signals.

Signal Name	Input/Output Status	Description
Clear to send	I	1 = The external device is ready for the serial port to transmit data.
Data set ready	I	1 = The external device is ready to establish the communications link and transfer data with the controller.
Ring indicator	I	1 = The external device detected a telephone ringing signal.
Data carrier detect	I	1 = The external device detected a data carrier.
Data terminal ready	I	1 = The controller is ready to communicate.
Request to send	O	1 = The controller is ready to send data.
Output 1	O	1 = User-designated output. This is a spare signal that can be used.
Output 2	O	1 = User-designated output. Spare signal that controls interrupts to the system.

continued

INT 14h and the serial port

The ROM BIOS Serial Communications Service is invoked via INT 14h, and its functions perform various serial I/O operations on IBM-compatible serial port adapters and controllers. This service contains functions that initialize the serial port, transmit a character across the I/O path, receive a character, and read the status of the serial port.

INT 14h and data transmission

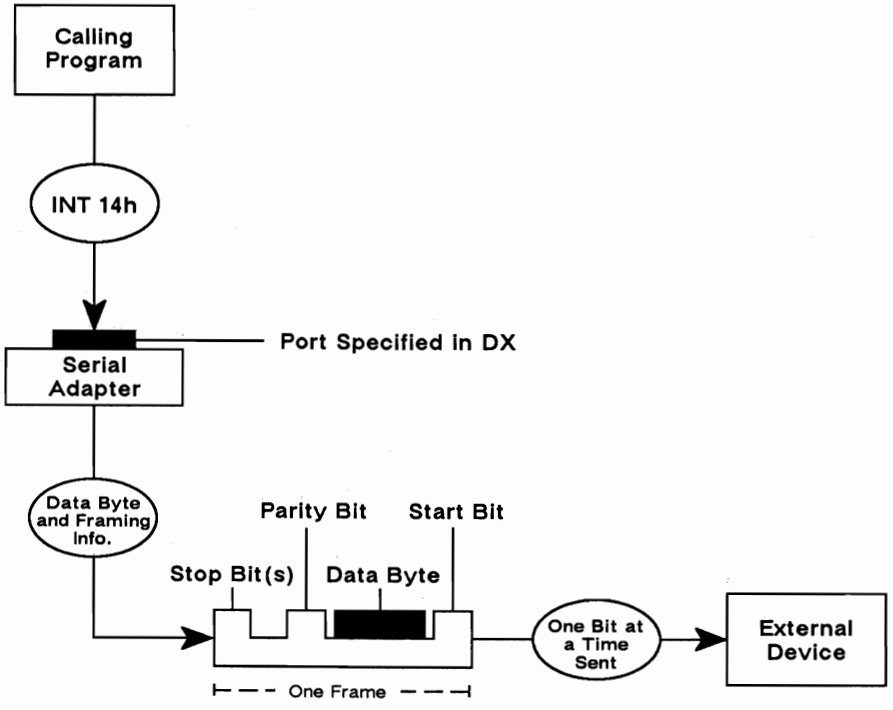
The process whereby an INT 14h request causes data to be transmitted across the RS-232C I/O path is:

1. A program places the data byte to be sent in AL, puts Function 01h Send Character in AH, and performs an INT 14h.
2. The BIOS transfers the data in AL to the serial port specified in DX. The serial controller and the external device communicate through a series of signals. These signals, such as Data terminal ready (DTR) and Request to send (RTS), relay control information across the I/O path. When the external device signals that it is ready to begin the data transfer, the controller inserts the appropriate bits before and after each data byte and sends each frame across the I/O path.
3. The external device receives each character, removes the start, parity, and stop bits, and assembles the characters.

continued

Serial data flow

The graphic below traces the path of serial data from the controller to the receiving device.



System RAM Data

Introduction

The Serial Communications Service makes use of control information located in the BIOS data area of system RAM (address 40:00h through 40:100h). 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 BIOS power-on self test (POST) places its corresponding base port address into the Serial Port Table (40:00h). In systems containing fewer 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

How baud rate divisor is calculated

The controller's clock input runs at 1.8432 MHz. The values in the table are calculated as follows:

$$\frac{1{,}843{,}200}{16} = \frac{115200}{\text{Baud Rate}} = \text{Divisor}$$

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

Serial Communications I/O Ports

INT 14h references the following I/O ports. These registers are used to monitor the controller's operations, and to transmit and receive data.

I/O Address	Read/Write Status	Description
03F8h	W	Serial 1, transmitter holding register, which contains the character to be sent. Bit 0, the least significant bit, is sent first. Bits 7-0 = Contains data bits 7-0, respectively, when Divisor Latch Access Bit (DLAB) = 0
03F8h	R	Serial 1, receiver buffer register, which contains the received character. Bit 0, the least significant bit, is sent first. Bits 7-0 = Contains data bits 7-0, respectively, when DLAB = 0
03F8h	R/W	Serial 1, divisor latch, low byte. Both divisor latch registers store the baud rate divisor. 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. Allows the four controller interrupts to enable the chip-interrupt output signal. 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
03FAh	R	Serial 1 interrupt ID register. Information about a pending interrupt is stored here. When ID register is addressed, the highest priority interrupt is held and no other interrupts are acknowledged until the CPU services that interrupt. Bits 7-3 = 0 Reserved Bits 2-1 = These bits identify the pending interrupt with the highest priority: 11b Receiver Line Status Interrupt; priority = highest. 10b Received Data Available; priority = second 01b Transmitter Holding Register; priority = third. 00b Modem Status Interrupt; priority = fourth. Bit 0 = 0 Interrupt pending, and contents of register can be used as a pointer to the appropriate interrupt service routine. = 1 No interrupt pending.
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. Serial output is forced to spacing state and remains there. Bit 5 = Stick Parity Bit 4 = Even Parity Select Bit 3 = Parity Enable = 1 Even number of ones are sent and checked in the data word bits and parity bit, if Bit 3 is 1. = 0 Odd number of ones are sent and checked, if Bit 3 is 1. Bit 2 = Specify number of stop bits per character: = 0 1 Stop Bit = 1 0 Stop Bits Bits 1-0 = Specify the number of bits per character: 00b 5 Bit Word Length 01b 6 Bit Word Length 10b 7 Bit Word Length 11b 8 Bit Word Length

continued

I/O Address	Read/Write Status	Description
03FCh	R/W	Serial 1, modem control register, where: Bits 7–5 = 0 Reserved Bit 4 = 1 Loopback Mode for diagnostic testing of serial port. Output of transmitter shift register is looped back to receiver shift register input. When in this mode, transmitted data is immediately received so that the CPU can verify the transmit data/receive data serial port paths. 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
03FDh	R	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. The controller is ready to accept a new character to send. Bit 4 = 1 Break interrupt. The received data input is held in the zero bit state longer than the time of start bit + data bits + parity bit + stop bits. Bit 3 = 1 Framing error. The stop bit that follows the last parity or data bit is a zero bit. Bit 2 = 1 Parity error. Character has wrong even or odd parity. Bit 1 = 1 Overrun error. A character was sent to the receiver buffer before the previous character in the buffer could be read. This destroys the previous character. Bit 0 = 1 Data ready. A complete incoming character has been received and sent to the receiver buffer register.
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

continued

I/O Address	Read/Write Status	Description
03FFh	R/W	Serial 1, Reserved
02F8h	W	Serial 2, transmitter holding register, which contains the character to be sent. Bit 0, the least significant bit, is sent first. Bits 7-0 = Contains data bits 7-0, respectively, when Divisor Latch Access Bit (DLAB) = 0
02F8h	R	Serial 2, receiver buffer register, which contains the received character. Bit 0, the least significant bit, is first to be sent. Bits 7-0 = Contains 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. Allows the four controller interrupts to enable the chip-interrupt output signal. 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 ID register. Information about a pending interrupt is stored here. When ID register is addressed, the highest priority interrupt is held and no other interrupts are acknowledged until the CPU services that interrupt. Bits 7-3 = 0 Reserved Bits 2-1 = These bits identify the pending interrupt with the highest priority: 11b Receiver Line Status Interrupt; priority = highest. 10b Received Data Available; priority = second 01b Transmitter Holding Register; priority = third. 00b Modem Status Interrupt; priority = fourth. Bit 0 = 0 Interrupt pending, and contents of register can be used as a pointer to the appropriate interrupt service routine. = 1 No interrupt pending.

continued

I/O Address	Read/Write Status	Description
02FBh	R/W	Serial 2, 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. Serial output is forced to spacing state and remains there. Bit 5 = Stick Parity Bit 4 = Even Parity Select Bit 3 = Parity Enable = 1 Even number of ones are sent and checked in the data word bits and parity bit, if Bit 3 is 1. = 0 Odd number of ones are sent and checked, if Bit 3 is 1. Bit 2 = Specify number of stop bits per character: = 0 1 Stop Bit = 1 0 Stop Bits Bits 1-0 = Specify the number of bits per character: 00b 5 Bit Word Length 01b 6 Bit Word Length 10b 7 Bit Word Length 11b 8 Bit Word Length
02FCh	R/W	Serial 2, modem control register, where: Bits 7-5 = 0 Reserved Bit 4 = 1 Loopback mode for diagnostic testing of serial port. Output of transmitter shift register is looped back to receiver shift register input. When in this mode, transmitted data is immediately received so that the CPU can verify the transmit data/receive data serial port paths. 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. The controller is ready to accept a new character to send. Bit 4 = 1 Break interrupt. The received data input is held in the zero bit state longer than the time of start bit + data bits + parity bit + stop bits. Bit 3 = 1 Framing error. The stop bit that follows the last parity or data bit is a zero bit. Bit 2 = 1 Parity error. Character has wrong even or odd parity. Bit 1 = 1 Overrun error. A character was sent to the receiver buffer before the previous character in the buffer could be read. This destroys the previous character. Bit 0 = 1 Data ready. A complete incoming character has been received and sent to the receiver buffer register.
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 Data 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, Reserved
3220-3228h	R/W	Often used for serial port 3 (see I/O address descriptions for 03F8h-03FFh for details).
3228-322Fh	R/W	Often used for serial port 3 (see I/O address descriptions for 03F8h-03FFh for details).

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 0-3.
- 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 [XT] [AT]

Description

This 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.

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
  - 101b = 2400 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 = 1 Time-out error
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 [XT] [AT]

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 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 [XT] [AT]

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 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 [XT] [AT]

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

Functions: AH = 04h - FFh Reserved

13

Chapter 13
INT 15h System Services

Overview

Introduction

INT 15h, System Services, provides several services:

- **Multitasking hooks**
  The BIOS 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.
- **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 immediately but sets a bit when a predetermined interval is over.

continued

Introduction, cont'd

- **Protected mode support**
  The BIOS System Services provides limited protected mode support through two functions: Function 87h, Move Block, and Function 89h, Switch Processor to Protected Mode.
- **System information**
  Function C0h, Return System Configuration Parameters, and Function 88h, Read Extended Memory Size, provide information about the system.

How INT 15h services are invoked

INT 15h invokes the BIOS System Services.

The INT 15h vector resides at interrupt table address 00:54h. The BIOS initializes the INT 15h vector to F000:F859h.

Individual INT 15h System Services functions are selected via the AH register. Subfunctions are selected via the AL, DX, or BH registers.

In this chapter

The following topics are discussed in the rest of the INT 15h System Services chapter:

- Summary of Functions
- Hardware Environment
- System RAM Data
- CMOS RAM Data
- ROM BIOS Data
- System Services I/O Ports
- System Services Functions

Summary of INT 15h System Services functions

The table below lists the System Services functions and subfunctions.

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	Platforms
00h	Turn Cassette Motor On	[PC]
01h	Turn Cassette Motor Off	[PC]
02h	Read Cassette	[PC]
03h	Write to Cassette	[PC]
4Fh	Keyboard Intercept	[AT]
80h	Device Open*	[AT]
81h	Device Close*	[AT]
82h	Program Termination*	[AT]
83h	Set Event Wait Interval AL = 00h Set Wait AL = 01h Cancel Wait	[AT]
84h	Joystick Support DX = 00h Read Current Switch Settings DX = 01h Read Resistive Inputs	[AT]
85h	System Request Key*	[AT]
86h	Wait	[AT]
87h	Move Block	[AT]
88h	Read Extended Memory Size	[AT]
89h	Switch Processor to Protected Mode	[AT]
90h	Device Busy*	[AT]
91h	Interrupt Complete*	[AT]
C0h	Return System Configuration Parameters	[AT]
* Multitasking hooks		

Hardware Environment

INT 15h System Services supports the following system components:

- an 8259 Programmable Interrupt Controller (two, if an AT system),
- an 8237 DMA controller,
- an 8254 (or 8253 in an XT) programmable interval timer/counter chip,
- an MC146818A real time clock chip (AT only), and
- a game port controller.

Equivalent logic is also supported.

INT 15h, Function 84h Joystick Support supports all devices that can be attached to a game port, such as a mouse pointing device, joystick, touch pads, and track balls. It will support any device that can be attached to the game 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
40:67h	2 Words	Pointer (offset:segment) to reset code
40:98h	1 Word	User wait flag offset address
40:9Ah	1 Word	User wait flag segment address
40:9Ch	1 Word	Least significant byte of wait count
40:9Eh	1 Word	Most significant byte of 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

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 = Reserved
30h	1 Byte	Low byte of actual expansion memory size
31h	1 Byte	High byte of actual expansion memory size

System configuration table

The system configuration table is located in the BIOS ROM at F000:E6F5h. This table can be called into RAM via INT 15h 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: PC = FFh XT = FEh XT = FBh PCjr = FDh AT = FCh XT-286 = FCh PC Convertible = F9h
03h	1	1 Byte	Submodel byte, where: PC = 00h XT = 00h or 01h PCjr = 00h AT = 00h or 01h XT-286 = 02h PC Convertible = 00h
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 = Reserved Bit 1 = 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)

I/O Address	Read/Write Status	Description
0020h	W	PIC, Initialization Command Word 1 (ICW1) (Bit 4 is one), where: Bits 7-5 = 0 Only used in 80/85 mode Bit 4 = 1 Reserved Bit 3 = 0 Reserved 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 = 0 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 = 0 Reserved Bit 4 = 0 No special fully-nested mode = 1 Special fully-nested mode Bits 3-2 = 00b Non-buffered mode = 01b Non-buffered mode = 10b Buffered mode/slave = 11b 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

I/O Address	Read/Write Status	Description
0020h	W	Programmable Interrupt Controller, Interrupt request/In-service registers, 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
0040h	R/W	Programmable Interrupt Timer — Read/write counter 0
0041h	R/W	Programmable interrupt timer register
0042h	R/W	Programmable Interrupt Timer — Read/write counter 2
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
0044h	R/W	Timer 3 counter register (AT only)
0045h-0046h	R/W	Timer registers (AT only)
0047h	W	Control byte for timer three (AT only) 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 (AT only)

continued

I/O Address	Read/Write Status	Description
0061h	R	8255 output register (XT only), where: Bit 7 = 1 Clear keyboard) Bit 6 = 0 Hold keyboard clock low Bit 5 = 0 Enable I/O check Bit 4 = 0 Enable RAM parity check Bit 3 = 0 Read low switches Bit 2 = Reserved Bit 1 = 1 Speaker data enable Bit 0 = 1 Enable timer 2 gate to speaker
0062h	R/W	8255 input register (XT only), where: Bit 7 = 1 RAM parity check Bit 6 = 1 I/O channel check Bit 5 = 1 Timer channel 2 out Bit 4 = Reserved Bit 3 = 1 System board RAM size type 1 Bit 2 = 1 System board RAM size type 0 Bit 1 = 1 Coprocessor installed Bit 0 = 1 Loop in POST
0063h	R/W	8255 Command mode register (XT only), where: Bits 7-6 = Number of diskette drives, where: 00b = 1 drive 01b = 2 drives 10b = 3 drives 11b = 4 drives Bits 5-4 = Type of display at power-on, where: 00b = Reserved 01b = 40x25 color (mono mode) 10b = 80x25 color (mono mode) 11b = MDA (80x25) Bits 3-2 = Memory on system board (256K chips), where: 00b = 256K 01b = 512K 10b = 576K 11b = 640K Bits 3-2 = Memory on system board (64K chips), where: 00b = 64K 01b = 128K 10b = 192K 11b = 256K Bits 1-0 = Reserved

continued

I/O Address	Read/Write Status	Description
0064h	W	8042 Status (AT only), 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 (AT only), where: Bit 7 = 1 NMI disable Bits 6-0 = 0 CMOS address
0071h	R/W	CMOS RAM data register port (AT only)
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
0201h	R/W	Joystick port

How to determine level of support

The commonly accepted level of support for each ROM BIOS function is identified by the legend [PC], [XT], and/or [AT] on the function title line.

Legend	Description
[PC]	Function is supported by original IBM PC BIOS only.
[XT]	Function is supported by all IBM PC/XT and compatible ROM BIOSs. **Note:** This level of support is contained in the ROM BIOS implemented in the majority of IBM PC and PC/XT compatible systems.
[AT]	Function is supported by all IBM PC/AT and compatible ROM BIOSs.

Function: AH = 00h Turn Cassette Motor On [PC]

Description

Older PC BIOSs support the Turn Cassette Motor On function. On an older PC, this function turns on the tape cassette motor if a cassette drive is attached to the system. The cassette connector is generally located near the keyboard connector on the back of the PC.

XT and AT BIOSs support this function to the extent that the BIOS does not generate an error and returns with registers preserved if this function is invoked.

Functions 00h, 01h, 02h, and 03h may be useful to those who need to attach a low-voltage analog device to a PC or to generate a signal to a signalling device. These functions control the operation of a relay within the PC. This relay is useful for controlling low-voltage and low-current devices.

Input/Output

Input: AH = 00h

Output: AH = 00h No error
86h Cassette not present.
= Status
CF = 0 No error
= 1 Error

Function: AH = 01h Turn Cassette Motor Off [PC]

Description

The Turn Cassette Motor Off function is supported in older PC BIOSs. In an older PC it turns the tape cassette motor off.

This function is supported by XT and AT BIOSs only to the extent that the BIOS does not generate an error and returns with registers preserved.

Functions 00h, 01h, 02h, and 03h may be useful to those who need to attach a low-voltage analog device to a PC or to generate a signal to a signalling device. These functions control the operation of a relay within the PC. This relay is useful for controlling low-voltage and low-current devices.

Input/Output

Input:	AH	=	01h
Output:	AH	=	00h No error
			86h Cassette not present.
		=	Status
	CF	=	0 No error
		=	1 Error

Function: AH = 02h Read Cassette [PC]

Description

The Read Cassette function is supported in older PCs. Invoking this function causes the system to read data from the cassette.

This function is supported for compatability purposes only on XT and AT systems. An XT or AT BIOS preserves register values, does not generate an error, and returns to the caller when this function is invoked.

Input/Output

Input:

AH = 02h
CX = Number of bytes to read
ES:BX = Segment:Offset of buffer

Output:

AH = 00h No error
01h CRC error
= 02h Bit signals scrambled
= 04h No data found
= 80h Invalid command
= 86h Cassette not present
= Status
CF = 0 No error
= 1 Error
DX = Number of bytes actually read
EX:BX = Segment:Offset + 1 of last byte read

Function: AH = 03h Write to Cassette [PC]

Description

The Write to Cassette function is supported in older PCs. Invoking this function causes the system to write data from the system to the cassette.

This function is supported for compatability purposes only on XT and AT systems. An XT or AT BIOS preserves register values, does not generate an error, and returns to the caller when this function is invoked.

Input/Output

Input:

AH	=	03h
CX	=	Number of bytes to read
ES:BX	=	Segment:Offset of buffer

Output:

AH	=	00h No error
		80h Invalid command
	=	86h Cassette not present
	=	Status
CF	=	0 No error
	=	1 Error
CX	=	00H
EX:BX	=	Segment:Offset + 1 of last byte written

Function: AH = 4Fh Keyboard Intercept [XT] [AT]

Description

The INT 09h ISR calls this function, each time a key is pressed.

This function can be used to create alternate keyboard layouts and/or to cause the system to ignore certain keystrokes. For example, an operating system or resident utility can intercept Function 4Fh 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 discard the scan code altogether before returning to the caller.

If this function is not intercepted by an alternate routine, the BIOS returns to the caller with the scan code in AL (unchanged) and the Carry Flag clear. Otherwise, it processes as described below.

Input/Output

Input: AH = 4Fh
AL = Scan code input by INT 09h ISR

Output: AL = Scan code
CF = 0 Keystroke unchanged (original scan code in AL)
= 1 Keystroke changed (new scan code in AL)

Note: This function is supported only by later versions of the XT and AT BIOS. To determine if your BIOS supports this function, invoke INT 15h AH = C0h Return System Configuration Parameters. If the Keyboard Intercept function is supported, bit 4 of feature information byte 1 (40:05h) will be set to 1.

Function: AH = 80h Device Open [AT]

Description

This function is used by multitasking operating systems. A multitasking operating system would intercept this function and process the request as necessary. Normally, the BIOS simply returns with AH set to 00h and the Carry Flag clear. The caller must clear the Carry Flag before invoking this function.

A multitasking operating system could use this function to capture control of a logical device so as to arbitrate its use by multiple processes.

Input/Output

Input:
AH = 80h
BX = Device ID
CF = 0
CX = Process ID

Output:
AX = 0080h

Function: AH = 81h Device Close [AT]

Description

This function is used by multitasking operating systems. A multitasking operating system would intercept this function and process the request as necessary. Normally, the BIOS simply returns with AH set to 00h and the Carry Flag clear. The caller must clear the Carry Flag before invoking this function.

This function could be used by a multitasking operating system to release control of a logical device for a process. See also INT 15h, AH = 80h and 82h.

Input/Output

Input:
AH = 81h
BX = Device ID
CF = 0
CX = Process ID

Output:
AX = 0081h

Function: AH = 82h Program Termination [AT]

Description

This function is used by multitasking operating systems. A multitasking operating system would intercept this function and process the request as necessary. Normally, the BIOS simply returns with AH set to 00h and the Carry Flag clear. The caller must clear the Carry Flag before invoking this function.

A multitasking operating system could use this function to release control of all logical devices for a process that will soon terminate. See also INT 15h, AH = 80h and 81h.

Input/Output

Input: AH = 82h
BX = Device ID
CF = 0

Output: AX = 0082h

Function: AH = 83h Set Event Wait Interval [AT]

Description

INT 15h provides two wait functions: Function 83h Set Event Wait Interval and Function 86h Wait. The difference between these two functions is that processing continues when INT 15h Function 83h is invoked, but processing stops for the calling program when INT15h Function 86h is invoked.

Function 83h, subfunction AL = 00h causes the BIOS to set a signal in a user-defined memory location after a specified interval passes. Function 83h, subfunction AL = 01h cancels the wait.

This function accesses INT 70h, the Real Time Clock ISR. INT 70h has two components: an alarm interrupt service, and a periodic interrupt service. INT 15h, Function 83h sets and cancels the periodic interrupt service of INT 70h, as does INT 15h, Function 86h.

With this function, the caller must specify a location in ES:BX where the BIOS will set the high-order bit when the time period specified in CX and DX expires. The caller is responsible for clearing this bit after the time interval expires and before invoking this function again.

The input and output register settings are shown on the following page.

Additional Information

- An event wait's duration is an integral multiple of 976 microseconds, since the MC146818 RTC chip is usually programmed to tick 1024 times per second, or once every 976 microseconds.
- This function provides hardware-independent wait periods of finer granularity than provided with the DOS Get Time Function (INT 21h, Function 2Ch), or the INT 08h System Timer ISR, which ticks only 18.2 times per second. MS-DOS INT 21h, Function 2Ch returns time only in hundredths of a second.

continued

Subfunction: AL = 00h Set interval

This subfunction starts the alarm process. If a wait has already been started, control is returned to the caller with CF set.

Input/Output

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	= Bit 6 of CMOS RAM location 0Bh is set, if successful
		= 00h Function busy
	CF	= 0 No error
		= 1 Error, function busy

continued

Subfunction: AL = 01h Cancel interval

This function cancels the interval in progress.

Input/Output

Input:	AH	=	83h
	AL	=	01h
Output:	AX	=	8300h
	CF	=	0 No error
		=	1 Error

Error conditions

No action is taken and the registers are preserved if CX and DX contain zero values. If a wait is already in progress, CF is set and control is returned to the caller.

If the subfunction code in AL is not 00h or 01h, one (1) is subtracted from AL, the Carry Flag is set, and control is returned to the caller.

Reference

Refer to Chapter 15, INT 1Ah Time-of-Day Service, for more information about INT 15h, Functions 83h and 86h.

Function: AH = 84h Joystick Support [AT]

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, which are described below.

Subfunction: DX = 00h Read current switch settings

This subfunction returns the switch settings read from the joystick in bits 4 through 7 of AL, while bits 3–0 of AL are cleared. The switch settings are read at port 0201h. If no game adapter is installed, AL is cleared.

Input/Output

Input: AH = 84h
DX = 00h

Output: AH = DH on entry to this function
AL = Switch settings in bits 7–4, bits 3–0 = 0000b
CF = 0 No error
= 1 Incorrect value in DX

---

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 cleared. If no game adapter is installed, AX, BX, CX, and DX are all cleared.

Input/Output

Input: AH = 84h
DX = 01h

Output: AX = Joystick A X-coordinate value
BX = Joystick A Y-coordinate value
CF = 0 No error
= 1 Incorrect value in DX
CX = Joystick B X-coordinate value
DX = Joystick B Y-coordinate value

With a 250K Ohm joystick connected, potentiometer values will be around 0–416 (0000–01A0h).

Function: AH = 85h System Request Key [AT]

Description

The keyboard interrupt function is called by INT 09h when the Sys Req key is pressed. When called, this function returns to the caller with the scan code in AL (unchanged) and the Carry Flag clear.

This function is used by multitasking operating systems. A multitasking operating system would intercept this function and process the request as necessary. Normally, the BIOS simply returns with AH set to 00h and the Carry Flag clear. The caller must clear the Carry Flag before invoking this function.

Input/Output

Input:	AH	=	85h
	AL	=	00h Key make
		=	01h Key break
	CF	=	0
Output:	AX	=	0085h
	CF	=	0 No error
		=	1 Invalid number in AL

Function: AH = 86h Wait [AT]

Description

This function causes the BIOS to immediately suspend activity for the calling program for the amount of time specified in CX and DX. It then sets bit 7 of location 40:A0h when the specified interval is completed.

The values entered in CX and DX must be in microseconds. However, an event wait's duration will always be an integral multiple of 976 microseconds. Any value input into CX and DX will be rounded to a multiple of 976 microseconds because the RTC, which is used by this function, is usually programmed to tick 1024 times per second, or once every 976 microseconds.

INT 15h provides two wait functions: Function 83h Set Event Wait Interval and Function 86h Wait. The difference between these two functions is that processing continues when INT 83h is invoked, but processing stops for the calling programs when INT 86h is invoked.

No action is taken and the registers are preserved if CX and DX contain zero values. If a wait is already in progress, CF is set and control is returned to the caller.

Input/Output

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, wait not performed

Time delay granularity

Using this function, the caller can obtain hardware-independent delays of finer granularity than provided by the MS-DOS Get Time function (INT 21h, Function 2Ch), or the INT 08h System Timer ISR, which ticks only 18.2 times per second. MS-DOS INT 21h, Function 2Ch returns time only in hundredths of a second.

Function: AH = 87h Move Block [AT]

Description

This function copies a block of memory for a program that operates in real address mode from anywhere in the system address space to anywhere else in the system's address space.

A program operating in real address mode can indirectly access extended memory by using this function to copy data located in a range of extended memory to conventional memory, where the program can access the data. By invoking this function, a program can also perform the reverse operation — copying data from a conventional memory range of addresses to an extended memory address range.

Types of memory

Conventional memory is located at addresses below 640K. Only this area can be accessed by DOS and DOS application programs. Extended memory is located at addresses above 1 MB, and has to be accessed by an 80286, 80386SX, or 80386 microprocessor operating in the protected address mode.

Accessing extended memory

Addresses above 1 MB must be accessed through 80286/80386/80386SX data structures called descriptors. Descriptors are organized into structures called local descriptor tables (LDT) or global descriptor tables (GDT). Programs accessing protected mode must build descriptor tables that organize access to extended memory and allow translation of 32-bit virtual addresses into corresponding 24-bit physical addresses.

Input to this function: building part of a descriptor table

The caller must help build a descriptor table before invoking this function. However, the caller need only supply the source and destination addresses of the copy, the BIOS constructs the rest of the descriptor table.

---

continued

Input table description

ES:SI points to global descriptor table (GDT) that the caller partially builds before calling this function. The caller need only enter the information required at offsets 12h and 1Ah in the table; the BIOS supplies the rest of the information. The descriptors in this table allow the BIOS to perform a block move in protected address mode. The BIOS will set the access right byte as indicated in the table below. The caller must set the 24-bit addresses to the target and source locations. A 24-bit address permits access to addresses up to 16 MB.

The input table format is:

Offset	Initial contents	Size	Description
00h	All zeros	8 Bytes	Dummy entry
08h	N/A	8 Bytes	Completed by the BIOS
10h	N/A	2 Bytes	Completed by the BIOS
12h	24-bit address	3 Bytes	Pointer to source address for copy
15h	N/A	5 Bytes	Completed by the BIOS
1Ah	24-bit address	3 Bytes	Pointer to destination address
1Dh	N/A	19 Bytes	Completed by the BIOS

Input/Output

Input:
AH = 87h
CX = Number of 16-bit words to move (0-8000h words)
ES:SI = Pointer to a 30h-byte table allocated by the caller

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

continued

24-bit address

The 24-bit address provided by the caller is a physical base address that is the beginning address of the memory segment specified by the descriptor. This is the beginning address of the segment of code or data to be moved.

80286 descriptors

Although the caller is not concerned with this part of the table format, each 8 bytes of the table (except the first eight bytes, which must be zero) is a descriptor that references a memory segment.

Intel 80286 code forms protected mode addresses to access addresses up to 16 MB differently, using a descriptor format. The segment:offset address form is modified so that the segment address is a 24-bit address located at the descriptor starting address + 2 and the offset is a 16-bit value located at the descriptor starting address + 0. The 80286 data segment descriptor format is shown in the following table. The 80386 data segment descriptor format is slightly different, but is not shown here. See the *Intel 80386 Programmer's Reference* for details about the 80386 protected mode instruction format.

Offset	Size	Description
00h	2 Bytes	The size of the memory segment minus one.
02h	3 Bytes	The start address of the memory segment.
05h	1 Byte	Access flags, where: 9Ah or 9Bh = Code segment 92h or 93h = Writable data segment 90h or 91h = Read only data segment ▪ If an 80286-based system, 9Bh, 93h, or 91h will be the required values. ▪ In an 80386-based system, any of the values listed above may occur.
06h	2 Bytes	Reserved in an 80286 instruction

continued

Interrupts disabled

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 BIOS uses the keyboard controller to reset the processor on 80286-based systems.

Error conditions

If an exception interrupt occurs during the block move, AH is set to 02h. CF is set, ZF is cleared, and control is returned to the caller.

If a parity error occurs, AH is set to 01h, CF is set, ZF is cleared, and control is returned to the caller.

Function: AH = 88h Read Extended Memory Size [AT]

Description

The Read Extended Memory Size function reads the size of the extended memory (memory beginning at address 100000h) from CMOS locations 30h and 31h and stores the combined contents as a word in the AX register.

Extended memory is located at addresses above 1 MB, and has to be accessed by an 80286 or 80386 operating in protected mode. Since DOS is a real mode operating system, extended memory cannot be used to execute DOS programs. Extended memory can, however, be used to store data that can be indirectly accessed through an Extended Memory Manager (EMM).

The system may not be able to use the extended memory unless the system board is fully populated.

Input/Output

Input: AH = 88h

Output: AX = Number of contiguous 1K blocks of extended memory starting at address 1024K (100000h)

Function: AH = 89h Switch Processor to Protected Mode [AT]

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) provided by the caller.

Input/Output

Input:	AH	= 89h
	BH	= Index into the interrupt descriptor table (IDT) that defines where the first 8 hardware interrupts are based
	BL	= Index into the 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 global descriptor table (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) descriptor and the extra segment (ES) descriptor are initialized by the caller.
- BH must contain an index into the IDT 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. The user-supplied global descriptor table starting address must be in register ES:SI.

Table Entry #	Offset from ES:SI	Size	Description
0	00h	8 Bytes	Dummy. Must be initialized to 0
1	08h	8 Bytes	Descriptor referencing this GDT
2	10h	8 Bytes	Descriptor referencing caller-defined interrupt descriptor table
3	18h	8 Bytes	Descriptor referencing caller's data segment
4	20h	8 Bytes	Descriptor referencing caller's extra segment
5	28h	8 Bytes	Descriptor referencing caller's stack segment
6	30h	8 Bytes	Descriptor referencing code segment to which the function will return
7	38h	8 Bytes	Descriptor used during this function to map BIOS code segment

Global descriptor table 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 0018h, 0020h, and 0028h, respectively, so that the descriptors built at these GDT offsets describe the segments that these registers will reference after the BIOS returns to the caller.

BIOS interrupt vectors do not operate in protected mode. For this reason, a program that enters protected mode must construct its own IDT which does not overlap the BIOS real mode table, and which handles all interrupts while the program is in protected mode.

Function: AH = 90h Device Busy [AT]

Description

This function is used by multitasking operating systems. A multitasking operating system would intercept this function and process the request as necessary. Normally, the BIOS simply returns with AH set to 00h and the Carry Flag clear. The caller must clear the Carry Flag before invoking this function.

In a multitasking operating system, this function could be invoked by the BIOS fixed disk, diskette, parallel printer, serial communications, or keyboard services to perform a programmed wait for I/O completion, which would allow other tasks to be dispatched while I/O is in progress.

Input/Output

Input:

AH = 90h
AL = Device Type, where:
= 00h Fixed 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
CF = 0
ES:BX = Points to a network control block (if AL = 80h or any address above 80h)

Output:

AH = 00h Wait time not satisfied
CF = 0 Wait time not satisfied (the service must use its own wait routine)
= 1 Minimum wait time satisfied (wait performed)

Device types

Serially reusable devices must be given device types from 00h – 7Fh; reentrant devices must have types 80h – BFh; and wait only calls, which have no corresponding function in BIOS POST, must have device types C0h – FFh.

Function: AH = 91h Interrupt Complete [AT]

Description

This function is used by multitasking operating systems. A multitasking operating system would intercept this function and process the request as necessary. Normally, the BIOS simply returns with AH set to 00h and the Carry Flag clear. The caller must clear the Carry Flag before invoking this function.

In a multitasking operating system, this function could be invoked by the BIOS fixed disk, diskette, parallel printer, serial communications, or keyboard services so the service can be notified when I/O is completed. This function could also be used to allow the operating system to be informed when I/O is done so the requesting task could be reactivated.

The multitasking operating system printer service would not invoke this function, since printer output is not interrupt-driven.

Input/Output

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
CF	=	0
ES:BX	=	Points to a network control block (if AL = 80h)

Output: None

Function: AH = C0h Return System Configuration Parameters [AT]

Description

This 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/Output

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

continued

System configuration table

The system configuration table is located in the BIOS ROM at F000:E6F5h.

Offset (hex)	Initial value	Length	Description
00h	00h	2 Bytes	Number of bytes in this table (minimum = 8)
02h	02h	1 Byte	Model byte, where: PC = FFh PC XT = FEh PC XT = FBh PCjr = FDh AT = FCh XT-286 = FCh PC Convertible = F9h Unknown system board = FFh
03h	03h	1 Byte	Submodel byte, where: PC = 00h PC XT = 00h PCjr = 00h AT = 00h AT = 01h XT-286 = 02h PC Convertible = 00h Unknown system board = FFh
04h	04h	1 Byte	BIOS revision level (first release = 00b)
05h	05h	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 = 0 PC-type I/O channel implemented Bit 0 = 0 Reserved
06h	06h	1 Byte	Feature information byte 2 (reserved, zeros)
07h	07h	1 Byte	Feature information byte 3 (reserved, zeros)
08h	08h	1 Byte	Feature information byte 4 (reserved, zeros)
09h	09h	1 Byte	Feature information byte 5 (reserved, zeros)

14

Chapter 14
INT 17h Parallel Printer Service

Overview

Description

The Parallel Printer Service provides BIOS-level support for up to three parallel printer ports. This service provides three printer-related functions.

Invoking the Parallel Printer Service

Software INT 17h invokes the BIOS Parallel Printer Service.

The INT 17h vector resides at address 005Ch in the interrupt vector table. The BIOS initializes the INT 17h vector to address F000:EFD2h.

continued

Summary of Parallel Printer Service functions

The following table lists the Parallel Printer functions. Individual functions are selected via the AH register.

Function	Description	Platforms
00h	Print Character	[XT] [AT]
01h	Initialize Printer	[XT] [AT]
02h	Read Printer Status	[XT] [AT]
03h-FFh	Reserved	Reserved

In this chapter

This chapter focuses on the INT 17h Parallel Printer Service. The following topics are discussed:

- Theory of Operation
- System RAM Data
- Parallel Printer I/O Ports
- Error Handling
- Parallel Printer Functions

How to determine level of support

Throughout this book, the commonly accepted level of support for each ROM BIOS function is identified by the legend [PC], [XT], and/or [AT] on the same line as the function title.

Legend	Description
[PC]	Function is supported by original IBM PC BIOS only.
[XT]	Function is supported by all IBM PC/XT and compatible ROM BIOSs. **Note:** This level of support is contained in the ROM BIOS implemented in the majority of IBM PC and PC/XT compatible systems.
[AT]	Function is supported by all IBM PC/AT and compatible ROM BIOSs.

Theory of Operation

Introduction

The parallel printer port on an IBM PC, PC/XT, and PC/AT is an 8-bit-wide device that can be used to drive various IBM PC and compatible printers. The BIOS Parallel Printer Service provides support for up to three parallel printer ports.

Parallel transmission of data

In a parallel transmission connection, all the data bits of a data byte are transmitted simultaneously over the parallel printer port. Similarly, the external device (typically a parallel printer) receives all these data bits simultaneously. This is in contrast to a serial connection, where the individual bits of a data byte are transmitted sequentially, or one at a time.

Printer adapter and interface

Typically the parallel printer adapter is used to connect a printer to the parallel port interface, but any device that accepts 8 bits of parallel data can be attached.

The 25-pin parallel connector at the rear of the adapter is the interface to an attached printer. The parallel port provides various input/output signals for communication with the printer. In particular, it uses the strobe signal to write data to the printer. The parallel port I/O signals are described on the next page.

continued

I/O signals

The parallel port makes available two types of signals: control output signals which are stored in the printer control port, and status input signals which are stored in the printer status port. These signals can be read by the microprocessor to determine the success or failure of an operation. The following table describes these signals.

Signal Name	Input/Output Status	Description
+IRQ Enable	O	1 = An interrupt occurs when -ACK changes from true to false
+SELECT IN	O	1 = Select the printer
-INIT	O	0 = Start the printer
+AUTO FEED	O	1 = The printer produces a line feed after a line has been printed
+STROBE	O	1 = Write data to the printer
-BUSY	I	1 = The printer is busy and cannot print
-ACK	I	0 = The printer received the character and is ready to accept another character
+PE	I	1 = The printer is out of paper
+SELECT	I	1 = The printer is selected
-ERROR	I	0 = An error condition has occurred

continued

INT 17h and the parallel port

The Parallel Printer Service is invoked via software interrupt 17h. This service contains three BIOS functions that interact with the parallel port: one to print the character in AL to the printer specified in DX, one to initialize that printer, and one to read and return that printer's status.

For example, when the caller wants to send a print request to the microprocessor, INT 17h Function 00h Print Character is invoked. This writes a character to the printer specified in DX.

The INT 17h functions reference three parallel printer I/O ports: the data port, the printer control port, and the printer status port.

Reference: A description of the parallel printer I/O ports can be found under the heading Parallel Printer I/O Ports in this chapter.

INT 17h and printing data

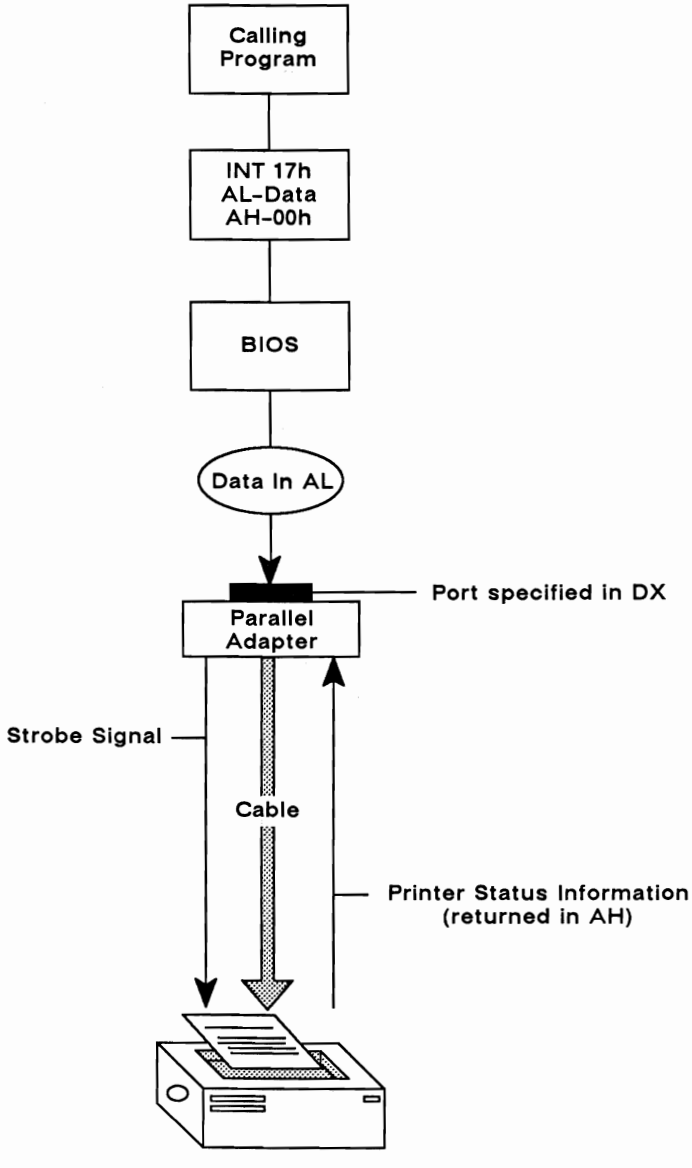
The process whereby an INT 17h request causes data to be printed is:

1. A program places the data byte to print in AL, puts Function 00h Print Character in AH, and performs an INT 17h.
2. The BIOS transfers the data in AL to the printer port specified in DX. The port's strobe signal is then activated, causing the data to be written to the printer. Printer status information, such as "out of paper" or "busy," is returned in AH and can be read by the microprocessor to check if the operation was successful.

continued

Printing a character

The graphic below illustrates how the INT 17h Print Character function (00h) writes data to an attached parallel printer.



Parallel printer data definitions

The INT 17h Parallel Printer Service uses certain locations in the BIOS data area of system RAM (address 40:00h through 40:100h). These data definitions are presented below in offset order.

Location	Size	Description
08h	4 words	I/O address of up to 4 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 4.

Parallel Printer I/O Ports

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 7–5 = Reserved 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
037Ah	R/W	Parallel 2, control port, where: Bits 7–5 = Reserved 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

continued

Parallel Printer I/O Ports, Continued

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: Bits 7-5 = Reserved 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

Error Handling

Before entering any functions, the Parallel Printer Service ensures that the index in DX is not out of range, that the function number is not out of range, and that the printer is at the specified index.

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 [XT] [AT]

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:
AH = 00h
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 [XT] [AT]

Description

The Initialize Printer function initializes the printer selected in the DX register. The printer status is returned in register AH.

Input/Output

Input: AH = 01h
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 [XT] [AT]

Description

The Read Printer Status function reads and returns the status of the printer selected by the DX register.

Input/Output

Input: AH = 02h
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

BIOS time services and timer chips

A PC, XT, or AT BIOS may provide up to five time-related interrupts: INT 1Ah, INT 08h, INT 70h, INT 1Ch, and INT 4Ah. The time-related services supported by the BIOS depend on the timer chips that are available.

AT systems have an Intel 8254 or equivalent timer chip. XT systems have an Intel 8253 or equivalent timer chip. Both the 8253 and 8254 are usually programmed to generate a timer tick 18.2 times a second. INT 1Ah, Functions 00h and 01h program the 8254. INT 1Ah is not available in most XT systems, and therefore, does not program the 8253. INT 08h and INT 1Ch also access the 8254 in both XT and AT systems.

AT systems have a Motorola MC146818A Real Time Clock, or equivalent, as well as an 8254. An MC146818A is usually programmed to generate a timer tick 1024 times per second, and contains 64 bytes of nonvolatile RAM. The first 12 bytes of this RAM are used to store time-related data, which INT 1Ah, Functions 02h - 05h program. INT 1Ah, Functions 06h and 07h and INT 15h System Services Functions 83h and 86h also set the RTC. INT 70h is the RTC ISR and INT4Ah is the User Alarm Interrupt that is invoked by INT 70h.

continued

INT 1Ah functions

The AT BIOS Time-of-Day Service provides eight system time-related functions. One additional INT 1Ah function is supported only by the PCjr BIOS and is documented here for historical purposes.

CMOS RAM access

The Time-of-Day Service provides an easy and quick way to modify CMOS RAM data, since the first 13 bytes of CMOS RAM contain time-related information (time, day, date, alarm settings, etc). The standard method of accessing CMOS RAM involves writing information to and reading I/O ports 0070h and 0071h multiple times. Invoking an INT 1Ah functions is much faster.

Invoking the BIOS Time-of-Day Service

The BIOS 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 BIOS initializes the INT 1Ah vector to address F000:FE6Eh.

How to determine level of support

Legend	Description
[PC]	Function is supported by original IBM PC BIOS only.
[XT]	Function is supported by all IBM PC/XT and compatible ROM BIOSs. **Note:** This level of support is contained in the ROM BIOS implemented in the majority of IBM PC and PC/XT compatible systems.
[AT]	Function is supported by all IBM PC/AT and compatible ROM BIOSs.

continued

Summary of Time-of-Day Service functions

Function	Description	Platforms
00h	Read System Timer Time Counter	[AT]
01h	Set System Timer Time Counter	[AT]
02h	Read Real Time Clock Time	[AT]
03h	Set Real Time Clock Time	[AT]
04h	Read Real Time Clock Date	[AT]
05h	Set Real Time Clock Date	[AT]
06h	Set Real Time Clock Alarm	[AT]
07h	Reset Real Time Clock Alarm	[AT]
80h	Set Sound Source (PCjr only)	[PCjr]

In this chapter

This chapter discusses each of the BIOS timer-related services. The following topics are presented:

- Theory of Operations
- System RAM Data
- CMOS RAM Data
- Time-of-Day Service I/O Ports
- Error Handling
- Time-of-Day Service Functions
- INT 08h System Timer ISR
- INT 70h Real Time Clock ISR
- INT 1Ch Timer Tick ISR
- INT 4Ah Alarm ISR

The BIOS provides the following timer-related support:

- **INT 1Ah Time-of-Day Service**
  AT-only software INT 1Ah invokes the BIOS Time-of-Day Service. The Time-of-Day Service provides access to the RTC CMOS RAM in locations 00h - 0Ch, and sets the INT 70h Real Time Clock ISR alarm function.

- **INT 08h System Timer ISR**
  Channel 0 of the Intel 8253 or 8254 Timer/Counter (or its equivalent), is usually programmed to interrupt 18.2 times per second. Each such interval generates hardware INT 08h, System Timer ISR. This interrupt is generated on PCs, XTs, and AT systems. INT 08h maintains a count of how many ticks have elapsed since midnight; this count is used for system timing purposes.

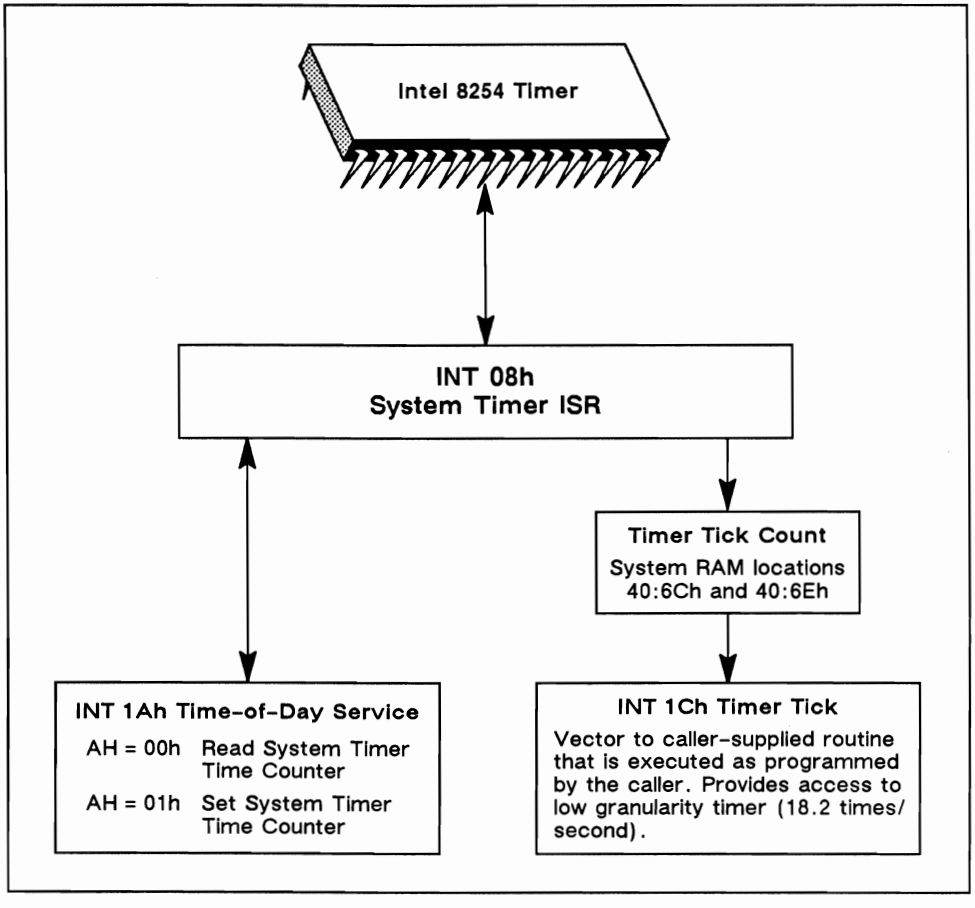
- **INT 1Ch Timer Tick ISR**
  Software INT 08h generates INT 1Ch 18.2 times per second. The user can revector INT 1Ch to a user-supplied routine to be executed on every system clock tick. No action is taken by the BIOS when INT 1Ch is invoked unless this vector is modified by the caller. Any caller modifying the vector associated with this interrupt is responsible for saving and restoring all registers upon return to the BIOS. Users should return control to the BIOS through an IRET.

continued

INT 08h System Timer ISR and the Intel 8254

The Intel 8254 provides timing services that can be accessed by the caller from INT 1Ch and INT 1Ah.

The following graphic describes how the Intel 8254, the INT 08h ISR, the INT 1Ah ISR, the INT 1Ch ISR, and the user interface are related.



continued

- **INT 70h Real Time Clock ISR**
  The Motorola MC146818A real time clock chip, or its equivalent, is usually programmed to generate hardware interrupt 70h approximately 1024 times per second. The real time clock chip usually appears only in AT systems.

  There are three functions that determine how the BIOS real time clock hardware ISR will handle interrupts from the real time clock chip.

BIOS Function	Effects on BIOS RTC ISR
Periodic function INT 15h: AH = 83h Event Wait Interval INT 15h: AH = 86h Wait	These functions activate the BIOS Real Time Clock ISR's periodic interrupt component. For INT 15h AH = 83h Event Wait, the caller must provide a designated location where the BIOS will set a bit when the time for the Event Interval period is expired. For INT 15h AH = 86h Wait, the BIOS sets bit 7 of location 40:A0h after the Wait is completed.
Alarm function INT 1Ah: AH = 06h Set User Alarm INT 1Ah: AH = 07h Reset User Alarm	This function activates the BIOS Real Time Clock ISR's alarm interrupt component.

- **INT 4Ah User Alarm**
  INT 70h generates an INT 4Ah every timer tick. Usually, INT 4Ah returns immediately to the BIOS. But a user can revector INT 4Ah to a user-supplied routine that will be executed every tick of the RTC. The user must restore registers and return to the BIOS with an IRET when the user-supplied routine completes processing.

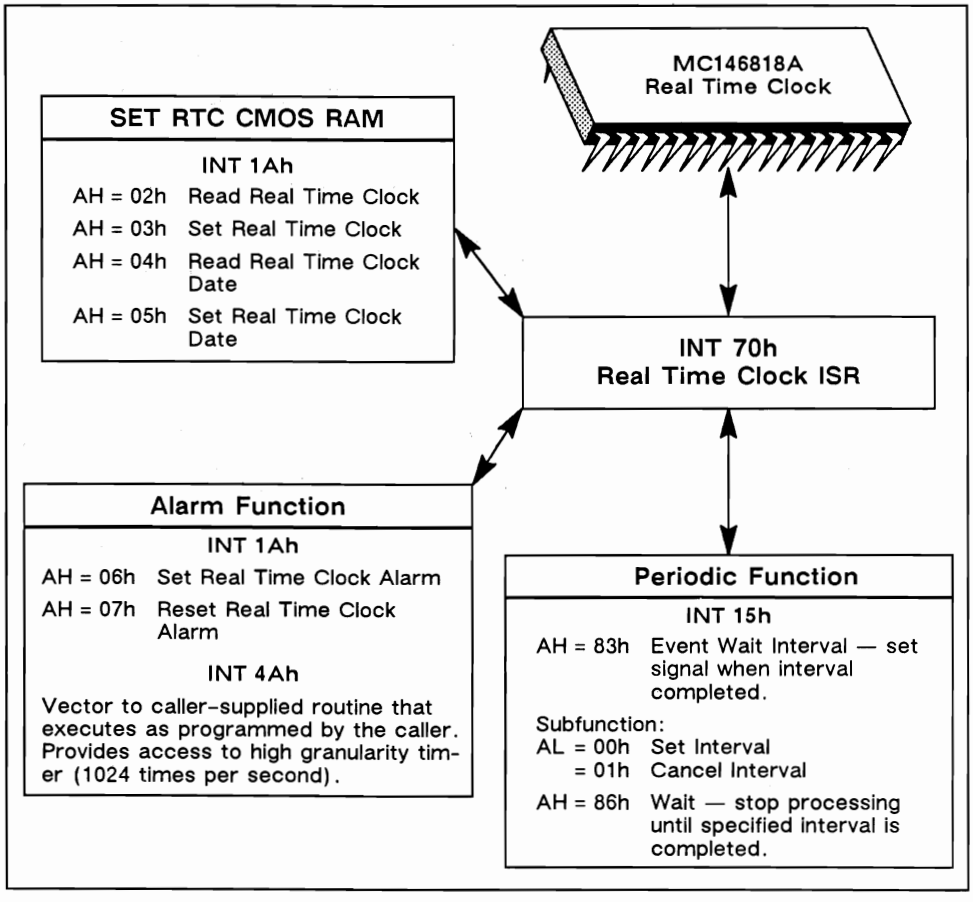
- **INT 15h System Services**
  Functions 83h and 86h of INT 15h set the RTC ISR (INT 70h) periodic function. See Chapter 13 for a more complete description of these two functions.

continued

INT 70h and the Real Time Clock

The MC146818A is accessed by the RTC DSR, INT 70h.

The following graphic shows how the MC146818A, the BIOS interrupts, and the user interface are related.



System RAM Data

Description

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	4 Bytes	32-bit count of timer ticks since midnight
40:70h	1 Byte	Timer Overflow Flag

CMOS RAM Data

Description

The Time-of-Day Service accesses the first 12 bytes of the RTC CMOS RAM data. Time data is stored in these locations. It also accesses location 32h, Century in BCD. Location 32h is updated by software access, and not internally by the chip.

The CMOS RAM configuration data definitions begin at address 0Eh. See Chapter 3 for a detailed description of the contents of CMOS RAM.

continued

RTC CMOS RAM Data

The following table lists the MC146818A Real Time Clock data locations used by the BIOS Time-of-Day Service.

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
32h	1 Byte	Century in BCD

I/O Address	Read/Write Status	Description
0020h	R/W	Programmable Interrupt Controller, 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	R/W	PIC, Initialization Command Word 1 (ICW1) (Bit 4 is one), where: Bits 7-5 = 0 Only used in 80/85 mode Bit 4 = 1 Reserved Bit 3 = 0 Reserved 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
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 port
00A0h	R/W	Slave Programmable Interrupt Controller (AT only)

Error Handling

Description

All functions of the BIOS Time-of-Day Service set the Carry Flag if an invalid function is requested. The function number remains in AH if there is an invalid function request.

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 [AT]

Description

This function reads and returns the system tick count from the system RAM location 40:6Eh (high word) and 40:6Ch (low word). The value returned is the cumulative number of Intel 8254 clock ticks since midnight. This operation sets the timer tick count maintained by INT 08h.

The values stored in these locations are defined by the frequency of the INT 08h System Timer interrupt, which is usually programmed to interrupt approximately 18.2 ticks per second. The contents of 40:70h, the Timer Overflow Flag, are returned in AL. This value will be 0 if the timer has not exceeded 24 hours.

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/Output

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. The timer count is more than 24 hours since the last power-on, system reset, last system-timer time counter read, or the last time the system-timer time counter was set.
CF = 0 No error
= 1 Error
CX = High word of tick count
DX = Low word of tick count

Function: AH = 01h Set System Timer Time Counter [AT]

Description

This function stores values from CX and DX to RAM data areas 40:6Eh (high word) and 40:6Ch (low word). This operation sets the INT 08h timer tick count.

Execution of this function causes the Timer Overflow Flag at 40:70h to be reset to 0.

Input/Output

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 [AT]

Description

This function first finds out if the RTC is currently updating its clock value. If so, the Carry Flag is set, the function is not performed, and control is returned to the caller.

If an update is not in progress, the appropriate values are read from the RTC CMOS RAM and returned to the caller in AL, CX, and DX, as specified below, with the Carry Flag cleared and AH = 00h.

Input/Output

Input: AH = 02h

Output: AH = 00h
AL = Hours in BCD
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

Function: AH = 03h Set Real Time Clock Time [AT]

Description

This function first finds out if the RTC is currently updating its clock value. If an update is in progress, the real time clock is initialized, the Carry Flag is set, the function is not performed, and control is returned to the caller.

The values placed by the caller in CX and DX 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 Daylight Savings Time option
		01h Daylight Savings Time option

Output:

AH	=	00h
AL	=	Value written to CMOS 0Bh register
CF	=	0 No error
	=	1 Error

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.

Function: AH = 04h Read Real Time Clock Date [AT]

Description

This function first finds out if the RTC is currently updating its clock value. If it is, the Carry Flag is set, this function is not performed, and control is returned to the caller.

If no time-out occurs, the century, year, month, and day are extracted from CMOS RAM and are returned to the caller as described below. 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

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.

Function: AH = 05h Set Real Time Clock Date [AT]

Description

This function first finds out if the RTC is currently updating its clock value. If it is, the Carry Flag is set, this function is not performed, and control is returned to the caller.

The values placed in CX and DX, as defined below, are stored in their respective CMOS RAM storage locations. Century is stored in location 32h, 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

If a time-out occurs, which indicates that an update is in progress, the clock is initialized and the Carry Flag is set.

Function: AH = 06h Set Real Time Clock Alarm [AT]

Description

This function sets the alarm function of INT 70h. The BIOS 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, the BIOS determines if the RTC 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. Only one alarm function can be in an active state at any one time. The alarm interrupt, once set, will be activated every 24 hours until reset. The caller must place the address of an interrupt handling routine for the alarm in the vector for INT 4Ah. The entry for INT 4Ah in the interrupt vector table is 128h.

This function also enables INT 70h (IRQ 8) when invoked.

The values defined below are stored in their respective CMOS RAM storage locations. Unlike all other functions in this service, this function clears both AH and AL.

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

Function: AH = 07h Reset Real Time Clock Alarm [AT]

Description

This function stops the real time clock alarm interrupt, canceling any pending alarm request stored in the date/time location in CMOS RAM. This function does not disable the IRQ 8 generated by the real time clock, but it does reset the alarm function of INT 70h.

Input/Output

Input: AH = 07h

Output: AH = 00h
CF = 0 No error
= 1 Error

Function: AH = 80h Set Sound Source [PCjr]

Description

The Set Sound Source function is used only by the PCjr to set the tones for the Audio Out or RF modulator on the PCjr. This function is not supported in any other PC, XT, or AT environment. If invoked, it will be treated as an invalid function (the Carry Flag is set and the registers are preserved).

Input/Output

Input: AH = 80h
AL = Sound source (PCjr only), where:
= 00h 8253 Programmable Timer, channel 2
= 01h Cassette input
= 02h Audio In line on I/O channel
= 03h Sound generator chip

Output: None

INT 08h System Timer ISR

Description

The Intel 8254 timer/counter chip, or its equivalent, "ticks" roughly 18.2 times per second. Each tick generates hardware INT 08h. PC, XT, and AT BIOSs support this interrupt.

The BIOS System Timer ISR increments the double word, system time count. The system time count is stored in the BIOS 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 Timer Overflow Flag at location 40:70h is set 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 and INT 08h

Once it has serviced the clock, INT 08h issues a software INT 1Ch Timer Tick ISR. BIOS users may revector INT 1Ch to their own routines. PC, XT, and AT BIOSs support this interrupt.

If not revectored, INT 1Ch returns immediately to the caller with all registers preserved.

How the BIOS System Timer ISR is invoked

The BIOS System Timer ISR is invoked via hardware INT 08h.

The INT 08h vector resides at address 00:20h in the interrupt vector table. The BIOS initializes the INT 08h vector to F000:FEA5h.

---

continued

System RAM data

The BIOS System Timer ISR references the following system RAM data:

Location	Size	Description
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

continued

System Timer I/O ports

The BIOS 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	Programmable Interrupt Timer — read/write counter 0
0042h	R/W	Programmable Interrupt Timer — 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
03F2h	W	Diskette controller digital output register, where: Bits 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, can be programmed to generate the real time clock hardware interrupt (INT 70h) approximately 1024 times per second. The BIOS Real Time Clock ISR is invoked on each real time clock interrupt. Only AT BIOSs support the Real Time Clock ISR.

How the Real Time Clock ISR is invoked

The BIOS 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 BIOS initializes the INT 70h vector to address F000:5124h.

Real Time Clock ISR processing

Three AT BIOS functions interface with the INT 70h Real Time Clock ISR. They are:

BIOS Function	Effects on Real Time Clock Processing
INT 15h AH = 83h Event Wait Interval INT 15h AH = 86h Wait These functions activate the BIOS Real Time Clock ISR's Periodic Interrupt component.	PERIODIC INTERRUPT PROCESSING The BIOS 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 location 40:A0h is set.
INT 1Ah AH = 06h Set User Alarm This function activates the BIOS Real Time Clock ISR's Alarm Interrupt component.	ALARM INTERRUPT PROCESSING The BIOS 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 BIOS Real Time Clock ISR issues a software INT 4Ah. BIOS users may revector INT 4Ah to their own routines. If not revectored, INT 4Ah returns immediately to the BIOS.

continued

System RAM data area

The BIOS Real Time Clock ISR references the following system RAM data:

Location	Size	Description
40:6Ch	2 Words	32-bit count of timer ticks since midnight.
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 byte of wait count
40:9Eh	1 Word	Most significant byte of wait count
40:A0h	1 Byte	Wait active flag, where: Bit 7 = 1 Wait time elapsed Bits 6-1 = Reserved Bit 0 = 1 INT 15h, AH = 86h (wait) has occurred

continued

CMOS RAM data table areas

The BIOS 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

Real Time Clock I/O Ports

The 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 RAM address
0071h	R/W	CMOS RAM data port
00A0h	R/W	Slave Programmable Interrupt Controller

INT 1Ch Timer Tick ISR

Once INT 08h has serviced the clock, it issues a software INT 1Ch. BIOS users may revector INT 1Ch to their own routines.

Software INT 1Ch points to a user routine to be executed on every system clock tick. The vector associated with this interrupt is invoked by the BIOS. Users should return control to the BIOS through an IRET. No action is taken by the BIOS when this interrupt is invoked unless this vector is modified by the caller. Any caller modifying the vector associated with this interrupt is responsible for saving and restoring all registers.

If not revectored, INT 1Ch returns immediately to the caller with all registers preserved.

INT 4Ah Alarm ISR

When the alarm function is activated, this interrupt will occur at the time specified when INT 1Ah Function 06h was last invoked. When the alarm time occurs, an INT 4Ah is generated. The caller must vector INT 4Ah to an alarm processing routine before invoking INT 1Ah AH = 06h Real Time Clock Alarm.

Only AT BIOSs support this interrupt. It is not supported by PC or XT BIOSs.

16

Chapter 16
Single Function BIOS Services

Overview

Introduction

Most of the BIOS device services contain a group of functions intended to service a given device. For example, in the BIOS 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 BIOS also contains several 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

How to determine level of support

Throughout this book, the commonly accepted level of support for each ROM BIOS function is identified by the legend [PC], [XT], and/or [AT] on the same line as the function title.

Legend	Description
[PC]	Function is supported by original IBM PC BIOS only.
[XT]	Function is supported by all IBM PC/XT and compatible ROM BIOSs. **Note:** This level of support is contained in the ROM BIOS implemented in the majority of IBM PC and PC/XT compatible systems.
[AT]	Function is supported by all IBM PC/AT and compatible ROM BIOSs.

INT 05h Print Screen Service [XT] [AT]

Description

The Print Screen Service:

- prints the contents of the entire current video screen to printer number 0 (LPT1),
- 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.

The original PC BIOS did not support INT 05h, however, many PC-compatible BIOSs do. Nearly all XT and AT BIOSs support INT 05h. As a rule, if the system has a keyboard with a PrintScreen key, the system BIOS will support INT 05h.

How the service is invoked

INT 05h invokes the BIOS Print Screen Service.

The INT 05h vector resides at the interrupt vector table address 00:14h. The BIOS initializes the INT 05h Print Screen Service vector to F000:FF54h.

INT 05h is usually invoked by the keyboard interrupt handler (INT 09h) when the PrtSc key is pressed. However, BIOS users may invoke INT 05h independently. For more information on INT 09h, refer to Chapter 8.

---

continued

System RAM Data

The Print Screen Service uses two system RAM data definitions. Both definitions are located in the BIOS Data Area (absolute 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

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 [XT] [AT]

Description

The Equipment List Service returns the contents of system RAM location 40:10h in AX. This memory location contains the system equipment list as determined by the BIOS POST routine.

How the service is requested

INT 11h requests the Equipment List Service of the BIOS.

The INT 11h vector resides at the interrupt vector table address 0:44h. The BIOS initializes the INT 11h Equipment List Service vector to address F000:F84Dh.

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 = VGA/EGA/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 [XT] [AT]

Description

The Memory Size Service returns the contents of system RAM location 40:13h in AX. This location contains the amount of available base memory in kilobytes as it is determined by the BIOS POST routine. The value returned reflects the amount of conventional memory in the system and does not report extended memory.

How the service is invoked

INT 12h invokes the BIOS Memory Size Service.

The INT 12h vector resides at the interrupt vector table address 0:48h. The BIOS initializes the INT 12h memory size service vector to address F000:F841h.

Input/Output

Input: None

Output: Available Base Memory
AX = Contents of location 40:13h in binary form

INT 19h Bootstrap Loader Service [XT] [AT]

Description

The Bootstrap Loader Service attempts to load the 512-byte boot sector code from either diskette (usually drive A: only) or fixed disk to address 0:7C00h, transferring control to the code segment at that address. The boot code is usually used to initialize the operating system.

The boot code is located in the first sector of the 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 BIOS Bootstrap Loader Service.

The INT 19h vector resides at the interrupt vector table address 0:64h. The BIOS initializes the INT 19h Bootstrap Loader Service vector to address F000:E6F2h.

INT 19h is usually invoked by the POST routine at the end of all POST processing. BIOS 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.

---

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 is located in the first sector of the diskette (track 0, head 0, sector 1) or in the first sector of the fixed 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	On those systems that have password protection, 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 system's password control program. Power-on passwords are not supported in most XTs and ATs.
3. If power-on password enabled	If there is password support and the power-on password is enabled, it allows 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. This step is ignored in most PC, XT, and AT systems.
4. Process boot code, if found	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 INT 19h 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

In XT and AT systems, if no boot sector is found either on the diskette in drive A: or on the fixed disk, the 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 BIOS initializes INT 18h to point to a routine that displays a message such as "No boot device available, press F1 to continue."

In a PC BIOS, INT 18h loads the ROM-based BASIC interpreter, if one exists.

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
12h	1 Byte	Reserved for manufacturer's test, where: Bits 7-1 = Reserved Bit 0 = 1 Manufacturing test mode = 0 Non-manufacturing test mode
72h	1 Word	System Reset Flag Where: 1234h = Bypass memory test 4321h = Preserve memory 0064h = Burn-in mode
75h	1 Byte	Number of fixed disk drives

CMOS RAM data

The Bootstrap Loader Service references the following location in the CMOS RAM data area:

Location	Size	Description
0Eh	1 Byte	Diagnostic status, where: Bit 7 = 1 RTC has 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 found at POST Bit 3 = 1 Fixed disk or adapter fails initialization Bit 2 = 1 RTC time invalid Bit 1 = 1 Adapters do not match configuration Bit 0 = 1 Time-out in reading an adapter ID

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 (AT only)
0064h	W	8042 Commands (AT only)
0064h	R	8042 Status (AT only), 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 (AT only), where: Bit 7 = 1 NMI disable = 0 NMI enabled Bits 6-0 = 0 CMOS address
0071h	R/W	CMOS RAM data register port (AT only)
0063h	R/W	8255 Command mode register (XT only), where: Bits 7-6 = Number of diskette drives, where: 00b - 1 drive 01b - 2 drives 10b - 3 drives 11b - 4 drives Bits 5-4 = Type of display at power-on, where: 00b - Reserved 01b - 40x25 color (mono mode) 10b - 80x25 color (mono mode) 11b - MDA (80x25) Bits 3-2 = Memory on system board (256K chips), where: 00b - 256K 01b - 512K 10b - 576K 11b - 640K Bits 3-2 = Memory on system board (64K chips), where: 00b - 64K 01b - 128K 10b - 192K 11b - 256K Bits 1-0 = Reserved

A

Appendix A

AT BIOS Error Codes and Messages

Introduction

This appendix lists, separately, the error messages generated by the IBM AT BIOS and the Phoenix AT-compatible BIOS.

In this appendix

The tables on the following pages present all Phoenix BIOS error messages, their causes and corrective actions. The following types of messages are documented:

- IBM BIOS POST and Boot Messages
- Phoenix BIOS POST and Boot Messages
- Phoenix BIOS Run-Time Messages
- Phoenix BIOS Beep Codes
- SETUP Messages

IBM AT BIOS error messages

Error Number	Message	Possible Cause
101	System Board Error	System interrupt failed.
102	System Board Error	System timer failed.
103	System Board Error	System timer interrupt failed.
104	System Board Error	Protected mode operation failed.
105	System Board Error	Keyboard communication failure.
106	System Board Error	POST logic test problem.
107	System Board Error	NMI test failed.
108	System Board Error	Failed system timer test.
109	System Board Error	Problem with first 64K of RAM.
161	System Options Not Set	Possible bad battery.
162	System Options Not Set	Invalid checksum.
163	Time and Date Not Set	RTC failed.
164	Memory Size Error — (Run SETUP)	CMOS RAM checksum error.
201	Memory Error	RAM failed test.
202	Memory Address Error	RAM failed test.
203	Memory Address Error	RAM failed test.
301	Keyboard Error	Keyboard not connected.
302	System Unit Keylock Is Locked	Keyboard is locked.
303	Keyboard or System Unit Error	Keyboard chord or keyboard itself is bad.
304	Keyboard or System Unit Error	Keyboard chord or keyboard itself is bad.
401	CRT Error	Monitor or video adapter bad.
501	CRT Error	Monitor or video adapter bad.
601	Diskette Error	Diskette drive or controller bad.
602	Diskette Boot Record Error	No boot sector on diskette.
1780	Disk 0 Failure	Primary fixed disk filed.
1781	Disk 1 Failure	Secondary fixed disk failed.
1782	Disk Controller Failure	Fixed disk controller failed.

continued

IBM AT BIOS error messages, cont'd

Error Number	Message	Possible Cause
1790	Disk 0 Error	Error on fixed disk number 0.
1791	Disk 1 Error	Error on fixed disk number 1.
	ROM Error	Error reading ROM BIOS code.
	RESUME = "F1" KEY	Error. Press F1 to continue.
	Unlock System Unit Keylock	Keyboard is locked. Unlock keyboard.

AT BIOS error messages

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.
Display adapter failed; using alternate	▪ The color/monochrome switch is set wrong. ▪ 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 hard disk configuration information is incorrect.	Rerun SETUP and enter correct hard 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.
Gate A20 failure	Protected mode cannot be enabled.	Check the system board.
Hard disk configuration error	The specified configuration is not supported.	Correct the fixed disk configuration.
Hard disk controller failure	The controller card has failed.	Replace the controller card.
Hard disk failure	Bad disk.	Retry boot. If that doesn't work, replace the hard disk.
Hard disk read failure — strike F1 to retry boot	Defective fixed disk.	Retry boot. If that doesn't work, replace the hard disk.
FDD controller failure	Disk and diskette controller failed.	Replace the controller card.
FDD A is not installed	Can not find diskette controller for diskette drive A	Either install or replace the controller card.

continued

AT BIOS error messages, cont'd

Message	Possible Cause	Solution
FDD B is not installed	Can not find diskette controller for diskette drive B	Either install or replace the controller card.
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.
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 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*, expecting *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-value*, 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.

continued

AT BIOS error messages, cont'd

Message	Possible Cause	Solution
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.
No boot device available — strike F1 to retry boot	Either diskette drive A:, the hard disk, or the diskette itself is defective.	Retry boot. If that doesn't work, replace the diskette or the hard disk.
No boot sector on hard 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 CMOS real time 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 non-maskable interrupt.
Last boot incomplete	Incorrect configuration of Intel 82335 chip set.	Reconfigure Intel 82335.

continued

Informational messages

Message	Meaning
nnnK Base Memory	The amount of base memory that tested successfully.
nnnK Expanded Memory	The amount of expanded memory that tested successfully.
nnnK Extended Memory	The amount of extended memory that tested successfully.
nnnK Extra Memory	The amount of extra memory that tested successfully.
nnnK Standard Memory	The amount of standard memory that tested successfully.
Decreasing available memory	This message immediately follows any memory error message, and informs you that the memory chips are failing.
Memory tests terminated by keystroke	This message indicates that you have pressed the Spacebar while the memory tests were running. This 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 attempt to boot.
Base Memory size = 64K	Used in reporting base memory for Extended Features.
Extended Memory size = 00000K	Used in reporting extended memory size for Extended Features.

Run-time messages

Run-time messages are displayed if an error occurs after the boot procedure is complete.

Message	Possible Cause	Solution
I/O card parity interrupt at *address*. Type (S)hut off NMI, (R)eboot, other keys to continue	The peripheral card has failed.	Type (S)hut off NMI. **Note:** This will only temporarily allow the user to continue. The user must replace the peripheral card.
Memory parity interrupt at *address*. Type (S)hut off NMI, (R)eboot, other keys to continue	A memory chip(s) has failed.	Type (S)hut off NMI. **Note:** This will only temporarily allow the user to continue. The user must replace the memory chip(s).
Unexpected HW interrupt *interrupt* at *address*. Type (R)eboot, other keys to continue	This could be any hardware-related problem. **Note:** Not displayed if the extended interrupt handler is not enabled.	Check the hardware.
Unexpected SW interrupt *interrupt* at *address*. Type (R)eboot, other keys to continue	There is an error(s) in the software program. **Note:** Not displayed if the extended interrupt handler is not enabled.	Try turning the machine off and then on again. If that doesn't work, check the program.
Unexpected type 02 interrupt at address. Type (S)hut off NMI, (R)eboot, other keys to continue	There is an error(s) in the software program. **Note:** Not displayed if the extended interrupt handler is not enabled.	Try turning the machine off and then on again. If that doesn't work, check the program.

Phoenix BIOS Beep Codes

Description

Sometimes POST errors cannot be reported on the screen. When an error occurs before the screen is initialized or when the system is set to loop on the system board tests, then the screen cannot display the error message. Systems with a Phoenix AT-compatible BIOS will then generate beep codes, which may be used to identify a POST error that occurs when the screen is not available.

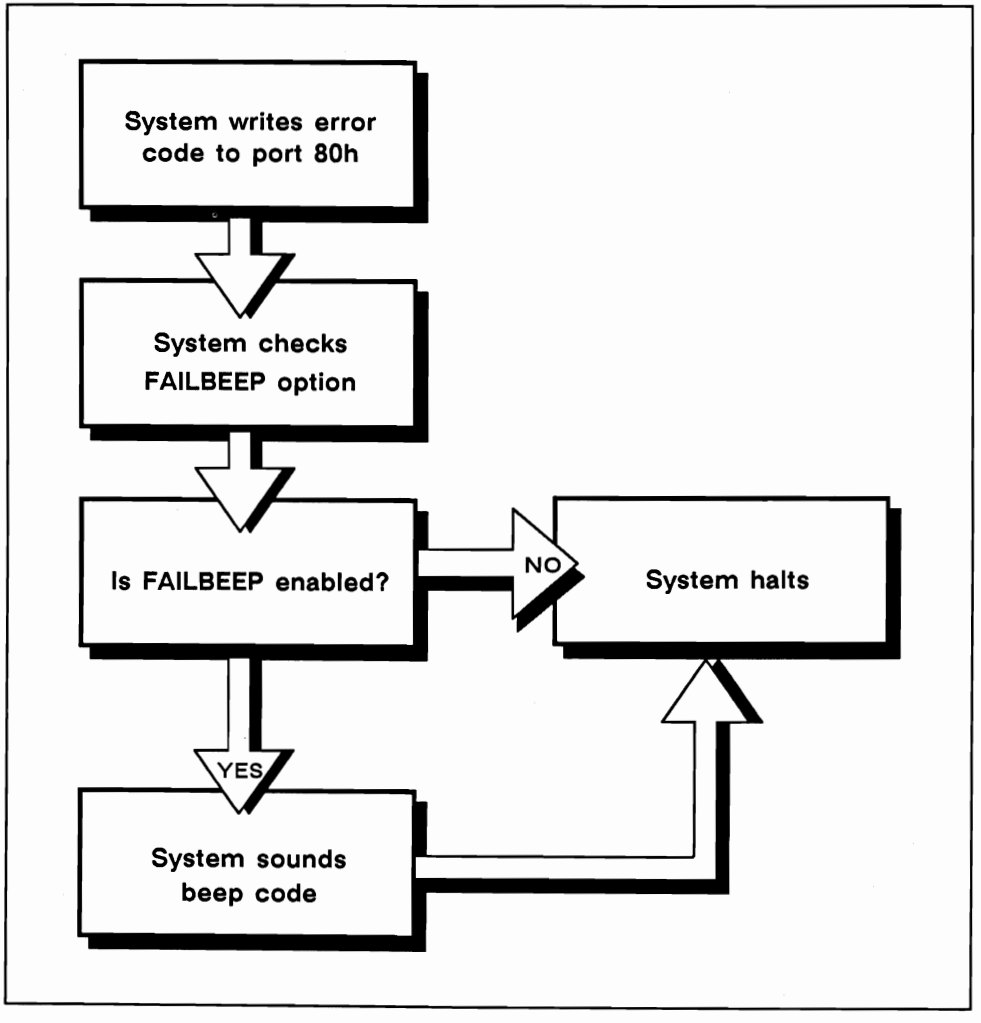
Example

A 2-1-4 beep code (a burst of 2 beeps, a single beep, and a burst of 4 beeps) indicates a failure of bit 3 in the first 64K of RAM.

continued

Diagram

The diagram below shows the process performed by the system if an error occurs and the screen is unavailable.



continued

Using Beep Codes

The table below shows the errors for which beep codes and screen messages are used:

	System Board Failure		Off board Failure
	Fatal	Non-fatal	
Looping on system board tests (MANLOOP set to True)	Beep and halt	Beep and halt	Not applicable
Normal power on (MANLOOP set to False)	Beep and halt	Screen message and prompt to 'Press F1 to continue'	Screen message and prompt to 'Press F1 to continue'

List of Beep Codes

The Beep Codes for fatal and non-fatal system board errors are listed separately on the following pages.

Note: No beep code is sounded if a test is aborted while in progress. The contents of port 80h can be read to identify the area of failure.

continued

Fatal system board errors

Beep Code	Contents Port 80h	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, multi-bit
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 failure
2-1-2	11h	Bit 1 first 64K RAM failure
2-1-3	12h	Bit 2 first 64K RAM failure
2-1-4	13h	Bit 3 first 64K RAM failure
2-2-1	14h	Bit 4 first 64K RAM failure
2-2-2	15h	Bit 5 first 64K RAM failure
2-2-3	16h	Bit 6 first 64K RAM failure
2-2-4	17h	Bit 7 first 64K RAM failure
2-3-1	18h	Bit 8 first 64K RAM failure
2-3-2	19h	Bit 9 first 64K RAM failure
2-3-3	1Ah	Bit 10 first 64K RAM failure
2-3-4	1Bh	Bit 11 first 64K RAM failure
2-4-1	1Ch	Bit 12 first 64K RAM failure
2-4-2	1Dh	Bit 13 first 64K RAM failure
2-4-3	1Eh	Bit 14 first 64K RAM failure

continued

Phoenix BIOS Beep Codes, Continued

Fatal system board errors, cont'd

Beep Code	Contents Port 80h	Description
2-4-4	1Fh	Bit 15 first 64K RAM 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 initialization failure
3-4-1	2Ch	Screen retrace test failure
3-4-2	2Dh	Search for video ROM in progress
none	2Eh	Screen running with video ROM
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

continued

Non-fatal system board errors

A failure in add-on boards or memory is reported on the monitor. These error messages help isolate the failed subsystem.

Beep codes 4-2-1 through 4-4-3 are only reported through the speaker and sent to the diagnostic port if the manufacturing loop option switch MANLOOP is set to TRUE and the manufacturing jumper indicator is on in POST. Otherwise, these errors are reported via the screen.

The following table describes the beep codes and error codes that are written to Port 80h for non-fatal system board errors.

Beep Code	Contents Port 680h	Description
4-2-1	34h	Timer tick interrupt test in progress or failure
4-2-2	35h	Shutdown test in progress or failure
4-2-3	36h	Gate A20 failure
4-2-4	37h	Unexpected interrupt in protected mode
4-3-1	38h	RAM test in progress or address failure > FFFFh
4-3-3	3Ah	Interval timer channel 2 test or failure
4-3-4	3Bh	Time-of-Day clock test or failure
4-4-1	3Ch	Serial port test or failure
4-4-2	3Dh	Parallel port test or failure
4-4-3	3Eh	Math coprocessor test or failure
low-1-1-2*	41h	System board select failure
low-1-1-3*	42h	Extended CMOS RAM failure
* "Low" means a lower-pitched beep precedes the three bursts.		

SETUP Messages

Configuration messages

All error messages that can be displayed by the SETUP utility are listed below. Each error message is followed by a description of the problem and a suggested action you can take to correct the problem.

Table of SETUP Error Messages

SETUP first prints the message:

```
Errors have been found during the power-on self test in your
computer. These errors were:
```

Message	Meaning	Required Action
Clock chip lost power *(hit any key to continue)*	The clock chip is not working.	Replace battery. Run SETUP; check all configuration settings.
CMOS checksum invalid *(hit any key to continue)*	There is a configuration error.	Run SETUP; check all configuration settings.
Incorrect configuration data in CMOS *(hit any key to continue)*	Configuration data entered in SETUP is not the same as the actual system components.	Run SETUP. Verify all settings are correct. If error persists, call the manufacturer's Technical Support Department.
Memory size in CMOS invalid *(hit any key to continue)*	Wrong base memory set.	Run SETUP; correct error in Base Memory field.
Disk C: failed initialization *(hit any key to continue)*	Incorrect fixed disk type.	Run SETUP; correct the fixed disk setting.
Time or date in CMOS is invalid *(hit any key to continue)*	The system clock time/date is not the same as that entered in the SETUP program.	Run SETUP; correct the date or time field.

Note: SETUP may not generate an error message if the incorrect diskette drive information is supplied.

B

Appendix B
XT BIOS Error Messages

Introduction

This appendix describes the power-on self test and boot messages that may be generated by both the IBM and the Phoenix XT BIOS.

These messages may be divided into the following types:

- Beep Code
- IBM XT BIOS POST Messages
- Phoenix XT BIOS POST and Boot Messages
- Phoenix XT BIOS Run-Time Messages

Each of these types of messages are discussed below.

Beep Code

Both the IBM and the Phoenix BIOS may generate a beep code under the following conditions. If a power-on self test message occurs before the video monitor testing is completed, there is no way for the BIOS to write an error message to the screen. Therefore, the BIOS will sound one long beep and then one short beep to indicate a POST error, and will stop all testing at the point of the error.

IBM XT BIOS POST Messages

The following messages may be generated by the IBM XT BIOS POST routine:

Message	Possible Cause
101	System board error.
201	Memory error.
ROM	Error reading ROM code.
1801	Expansion I/O box error.
Parity Check 1	NMI parity error found in I/O adapter card.
Parity Check 2	NMI parity error found in memory.
?????	Unknown NMI error.

Phoenix XT BIOS POST error messages

Power-on self test (POST) and boot messages are displayed during the power-on process. Within POST, there are three kinds of messages:

- system failure messages — failure in hardware, software, or firmware.
- boot failure messages — failure in the boot process.
- informational messages — require no action.

The following tables list these messages and describe their possible causes and suggested corrective actions.

Phoenix XT BIOS system failure messages

System failure messages generated by a Phoenix BIOS in an XT system are described in the following table.

Message	Possible Cause	Solution
Bad DMA port = *hex-value*	System board bad.	Test system board.
Disk bad	Fixed disk failure.	Check fixed disk controller cable, if not bad, controller failure possible.
Diskette drive 0 seek failure	The A: drive failed or is missing.	Possible bad or unformatted diskette. Format diskette, or use a different diskette.
Expansion Box Bad	Adapter cards may be malfunctioning.	Test adapter cards and all plugs. Replace if necessary.
No scan code from keyboard	Keyboard may be disconnected or malfunctioning.	Check keyboard plug and keyboard. Replace if necessary.
ROM Bad Sum =	Data in a ROM could not be read.	Replace bad ROM chip.
ROM bad checksum = *hex-value*	Read failed on ROM access.	Retry. If error persists, replace ROM.
Hex-value = Scancode, check keyboard	An erroneous scan code was received from the keyboard.	Key may be stuck on keyboard. Keyboard connector may be bad, or keyboard controller must be replaced.

continued

Phoenix XT BIOS system failure messages, cont'd

Message	Possible Cause	Solution
Stuck key scancode = *hex-value*	Key stuck on keyboard.	Loosen stuck key. Check scan code table to identify key.
Timer or Interrupt Controller Bad	Interrupt controller or timer chip inoperative.	Replace system board.
Timer chip counter 2 failed	Timer bad.	Replace system board or timer chip.

Phoenix XT BIOS boot failure messages

Boot failure messages generated by a Phoenix BIOS in an XT system are described in the following table.

Message	Possible Cause	Solution
128K not OK, parity disabled	First 128K of RAM failed parity test.	Reboot. If message repeats, replace bad RAM chip(s).
Keyboard bad	Keyboard failed POST tests.	Retry. If message persists, replace keyboard.
Error. Press F1 key to continue.	A recoverable error occurred.	Press F1 to continue (there may be errors), or reboot.
Non-System disk or disk error. Press a key to continue.	No boot diskette or disk available.	Use a system disk to boot the system.
Disk boot error. Replace and strike key or retry.	Boot data not readable from boot device.	Use a boot diskette/disk to boot system.
RAM bad	Memory failed POST tests.	Replace bad RAM chips.

continued

Phoenix XT BIOS informational messages

Informational messages generated by a Phoenix BIOS in an XT system are described in the following table.

Message	Possible Cause	Solution
RAM test	Information only.	No action required.
0h optional	Information only.	No action required.
Hard disk wait complete	Information only.	No action required.
Memory space preserved	Information only.	No action required.

Run-time message table

Run-time messages generated by a Phoenix BIOS are displayed if an error occurs after the boot procedure is complete.

Message	Possible Cause	Solution
8087 NMI at *address*. Type (S)hut off NMI, (R)eboot, other keys to continue.	An NMI occurred on access to the math coprocessor.	Type (S)hut off NMI. **Note:** This will only temporarily allow you to continue. You must replace the peripheral card.
Memory parity NMI at *address*. Type (S)hut off NMI, (R)eboot, other keys to continue.	A memory chip(s) has failed.	Type (S)hut off NMI. **Note:** This will only temporarily allow you to continue. You must replace the memory chip(s).
Unexpected HW interrupt *interrupt* at *address*. Type (R)eboot, other keys to continue.	This could be any hardware-related problem. **Note:** Not displayed if the extended interrupt handler is not enabled.	Check the hardware devices for loose cables, malfunctions, and other problems.
Unexpected SW interrupt *interrupt* at *address*. Type (R)eboot, other keys to continue.	There is an error(s) in the software program. **Note:** Not displayed if the extended interrupt handler is not enabled.	Turn the machine off and then on again. If that doesn't work, check the program.
I/O card NMI at *address*. Type (S)hut off NMI, (R)eboot, other keys to continue.	An NMI occurred on access to an adapter card.	Turn the machine off and then on again. If that doesn't work, replace the adapter card.

Glossary

Adapter Card

A circuit board that can be installed into one of the expansion slots inside a PC, XT, AT, or IBM-compatible computer in order to expand the capabilities of the computer.

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.

Address Bus

One or more lines (conductors) that carry address codes from the microprocessor to all parts of the system.

American National Standard Code for Information Exchange (ASCII)

A standard code, consisting of character, control, and graphic codes of 7 bits each (8 bits counting the parity bit).

Analog

Data transmitted in a continuously varying physical quality.

Arbitration

Arbitration is a process through which devices compete for possession of the channel on a prioritized basis.

continued

Arbitration Level

Arbitration levels are the levels of priority assigned to devices that compete for possession of the channel.

AT-Compatible Computer

Any computer that can run software programs written for an IBM AT computer.

Basic Input/Output System (BIOS)

Systems software that interfaces between the operating system and hardware.

BIOS 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.

Bits per second (BPS)

A unit of measure that represents the number of discrete bits transmitted by a device in one second.

Block Check Character (BCC)

An element in the checksum calculations in cyclic redundancy checking.

Boot

Process of starting the computer.

Buffer

An area of storage temporarily reserved for I/O operations.

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.

Bus High Enable (BHE)

A line on the 80286 processor that produces a signal, used in conjunction with the address 0 signal, to specify if data is transferred in words or bytes.

continued

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.

Central Processing Unit (CPU)

The main processing component of the microprocessor.

Channel

A path constructed specifically for moving data.

Command Control Block (CCB)

Disk I/O data structure.

Command Specify Block (CSB)

Disk I/O data structure.

Complementary Metal Oxide Semiconductor (CMOS)

In PC, XT, or AT 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.

Conventional memory

Memory located between addresses 0 and 1 MB. The memory that can be addressed by DOS.

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.

Cylinder

Tracks on a diskette or fixed disk that can be accessed without moving the read/write heads.

Default

A value, setting, or option that is assigned by the program or system.

continued

Digital-to-Analog Converter (DAC)

Used in VGA hardware to convert command and data when interfacing between digital computer hardware and an analog monitor.

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.

Disk Operating System (DOS)

Short for PC-DOS and MS-DOS. DOS, like other operating systems, organizes the files and memory for other programs.

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.

Dual Inline Package (DIP)

A method of organizing computer switches.

Erasable Programmable Read-Only Memory (EPROM)

A ROM device that can be erased and reprogrammed at anytime by a special machine.

Error Checking and Correction (ECC)

The process of detecting and correcting all single-bit errors using an algorithm that tests the accuracy of the transmitted data.

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.

continued

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 80286, 80386SX, 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.

Head

The read/write head on a diskette or fixed disk drive. The device that provides access to the data on the media.

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.

continued

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. Another term for conventional memory.

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.

Microprocessor

Central processing unit, or "brain" of the computer.

Modified Frequency Modulation

Traditional method of formatting data on a fixed disk.This technique involves a varying the amplitude and frequency of a signal to a fixed disk.

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.

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.

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.

continued

Program

A set of instructions defining the operations of a computer in order to achieve the desired results.

Protected 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 GB of virtual memory in protected mode (the 80386 up to 64 Terabytes). Protected mode addresses are specified in selector:offset format.

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.

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.

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.

continued

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.

Track

A segment of a diskette or fixed disk that is parallel to the edge of the media and that can be accessed by the drive read/write head without the head moving.

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.

Write Precompensation

A procedure where the timing of the head current on a diskette or fixed disk drive is varied depending on the head's position (anywhere from outer tracks to the inner tracks) in order to keep the Write signal constant. Often given as a cylinder number, as in the cylinder number where the write proecompensation procedure is to begin.

Additional Resources

The following books provide additional material related to the PC, XT, and AT BIOSs:

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.

Intel Corporation. *Introduction to the 80286.* Santa Clara, CA: Intel Corporation, 1982.

Intel Corporation. *80286 Programmer's Reference Manual.* Santa Clara, CA: Intel Corporation, 1983.

Intel Corporation. *80287 Programmer's Reference Manual.* Santa Clara, CA: Intel Corporation, 1984.

Intel Corporation. *Introduction to the 80386.* Santa Clara, CA: Intel Corporation, 1985.

Intel Corporation. *80386 Programmer's Reference Manual.* Santa Clara, CA: Intel Corporation, 1986.

Intel Corporation. *80387 Programmer's Reference Manual.* Santa Clara, CA: Intel Corporation, 1987.

continued

International Business Machines Corporation. *IBM Personal Computer AT Technical Reference*. Boca Raton, FL: International Business Machines Corporation, 1984.

International Business Machines Corporation. *IBM Personal Computer XT and Portable Personal Computer Technical Reference*. Boca Raton, FL: International Business Machines Corporation, 1986.

Kliewer, Bradley Dyck. *EGA/VGA Programmer's Reference Guide*. New York, NY: McGraw-Hill Book Company, 1988

Norton, Peter. *Inside the IBM® PC*. New York, NY: Brady Books, a division of Simon and Schuster, 1986.

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

C, cont'd

D

H, cont'd

I

J

K

P, cont'd

S, cont'd

T

W

X

Y

$26.95 FPT USA

The Phoenix Technical Reference Series

Phoenix Technologies Ltd.

System BIOS for IBM® PC/XT™/AT® Computers and Compatibles is an indispensable guide for anyone who needs a detailed understanding of the ROM-based system software resident in their IBM PC, XT, AT, or compatible computer. Applications programmers, operating system developers, hardware manufacturers, and students of PC architecture will find this handy reference book the most complete source of information available.

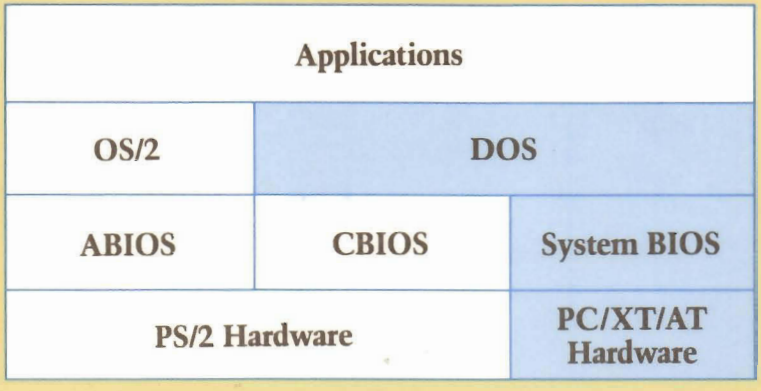
Key information includes:

- bit-level descriptions of I/O port addresses, CMOS RAM data areas, ROM BIOS data areas, and all system configuration tables
- descriptions of undocumented IBM BIOS 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 components, including video, keyboard, diskette, and fixed disk subsystems.

The information in this book is applicable to all IBM PC, XT, AT computers and compatibles.

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 *CBIOS for IBM PS/2 Computers and Compatibles: The Complete Guide to ROM-Based System Software for DOS.*



52695
9 780201 518061
Addison-Wesley Publishing Company, Inc.
ISBN 0-201-51806-6