

Phoenix Technologies Ltd.



PhoenixBIOS 4.0

Revision 6

User's Manual

Phoenix Technologies Ltd., 411 E. Plumeria, San Jose, CA 95134

Copyright

PhoenixBIOS 4.0 User's Manual

22 June 2000

© 2000 Phoenix Technologies Ltd.

All Rights Reserved.

Disclaimer

The programs are provided "as is" without warranty of any kind either expressed or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. This publication could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of this publication. Phoenix Technologies Ltd. is without obligation to notify any person of such revisions or changes.

Purpose of Document

This guide explains how to configure your PC and optimize its performance using the Setup program. It also explains how to use the BIOS function calls in writing computer programs.

PB4.0 UM 06.22.00

Contents

About This Manual	1
Chapter 1 The Setup Guide	2
The Main Menu	2
The Menu Bar	3
The Legend Bar	3
The Field Help Window	4
The General Help Window	4
Main Menu Selections	4
Master and Slave Sub-Menus	5
Memory Cache	7
Memory Shadow	8
Boot Sequence	9
Keyboard Features	10
Boot Menu	11
The Advanced Menu	12
Advanced Chipset Control (No PCI)	13
Advanced Chipset Control Menu (PCI BIOS)	14
PCI Devices Menu	15
I/O Device Configuration Menu	16
The Security Menu	18
The Power Menu	19
The Exit Menu	21
Saving Values	21
Exit Discarding Changes	21
Load Setup Defaults	21
Discard Changes	22
Save Changes	22
PhoenixBIOS Messages	23
Chapter 2 Boot Utilities	26
Phoenix QuietBoot	26
Press <ESC>	26
Press <F2>	26
POST Error	26
Keyboard Input Request	27
Phoenix MultiBoot	27
The Boot First Menu	27
Chapter 3 Phoenix Phlash	28
Installation	28
Create the Crisis Recovery Diskette	28
Updating the Crisis Recovery Diskette	29
Executing Phoenix Phlash	29
Crisis Recovery Mode	30
Chapter 4 Programmer's Guide	31
What is a ROM BIOS?	31
ROM BIOS Functions	32
Initialize and Configure the computer	32
BIOS Services	32
System Hardware Requirements	32
Fixed Disk Tables	33
PhoenixBIOS Function Keys	34
POST Errors and Beep Codes	34
Recoverable POST Errors	34
Terminal POST Errors	34
Test Points and Beep Codes	34
PhoenixBIOS 4.0 Services	39

<i>BIOS32 Service Directory</i>	39
<i>Interrupt 10h—Video Services</i>	41
<i>Interrupt 11h—Return System Information</i>	43
<i>Interrupt 12h—Return Memory Size</i>	43
<i>Interrupt 13h—Diskette Services</i>	43
<i>Interrupt 13h—Fixed Disk Services</i>	46
<i>Interrupt 13h—Extended Fixed Disk Services</i>	49
<i>Interrupt 13h—Bootable CD-ROM Services</i>	50
<i>Interrupt 14h—Serial Services</i>	51
<i>Interrupt 15h—System Services</i>	53
<i>Interrupt 15h—APM Services</i>	56
<i>Interrupt 15h—Big Memory Services</i>	59
<i>Interrupt 15h—PS/2 Mouse Services</i>	60
<i>Interrupt 15h—EISA Services</i>	61
<i>Interrupt 16h—Keyboard Services</i>	63
<i>Interrupt 17h—Parallel Printer Services</i>	64
<i>Interrupt 17h—EPP Services</i>	64
<i>Interrupt 1Ah—Time of Day Services</i>	68
<i>Interrupt 1Ah—General PCI Services</i>	68
<i>PnP Run-Time Services</i>	70
<i>SMBIOS Services</i>	74
<i>MultiBoot III Run-Time Services</i>	76
<i>BIOS Data Area</i>	77
<i>Extended BIOS Data Area</i>	79
<i>Interrupt Vectors</i>	80
<i>Index</i>	81

4 Programmer's Guide

This chapter of the User's Manual gives application developers, programmers, and expert computer users a detailed description of the BIOS.

This chapter describes the following subjects:

- What is a ROM BIOS?
- System Hardware Requirements
- Fixed-Disk Tables
- PhoenixBIOS Function Keys
- POST Errors
- Run-Time Services

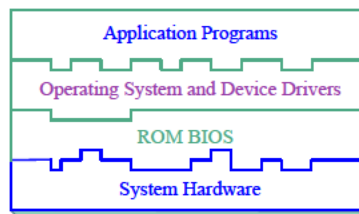
What is a ROM BIOS?

This section briefly explains the function of a BIOS in managing the special features of your system.

A **ROM BIOS (Basic Input/Output System)** is a set of programs permanently stored in a **ROM (Read-Only Memory)** chip located on the computer motherboard. These programs micro-manage the hardware devices installed on your computer. When you turn on your computer, the ROM BIOS initializes and tests these devices. During run-time, the ROM BIOS provides the Operating System and application programs with access to these devices. You can also use the **BIOS Setup** program to change your computer's hardware or behavior.

Software works best when it operates in layers, and the ROM BIOS is the bottom-most software layer in the computer. It functions as the interface between the hardware and the other layers of software, isolating them from the details of how the hardware works. This arrangement enables you to change hardware devices without having to install a new operating system.

The following diagram shows the function of the ROM BIOS as the interface between the hardware and other layers of software:



ROM BIOS Functions

The *PhoenixBIOS* software performs these functions:

The Setup Program	Using the Setup program, you can install, configure, and optimize the hardware devices on your system (clock, memory, disk drives, etc.).
Initialize Hardware at Boot	At power-on or reset, perform Power-On Self Test (POST) routines to test system resources and run the operating system.
Run-Time Routines	Basic hardware routines that can be called from DOS and Windows applications.

Initialize and Configure the computer

The first job of a ROM BIOS is to initialize and configure the computer hardware when you turn on your computer (system boot). The BIOS runs a series of complex programs called the **Power On Self Test (POST)**, which performs a number of tasks, including:

- Test Random Access Memory (RAM)
- Conduct an inventory of the hardware devices installed in the computer
- Configure hard and floppy disks, keyboard, monitor, and serial and parallel ports
- Configure other devices installed in the computer such as CD-ROM drives and sound cards
- Initialize computer hardware required for computer features such as Plug and Play and Power Management
- Run Setup if requested
- Load and run the Operating System such as DOS, OS/2, UNIX, or Windows 95 or NT.

BIOS Services

The second task of the ROM BIOS is to provide the Operating System, device drivers, and application programs with access to the system hardware. It performs this task with a set of program routines called **BIOS Services**, which are loaded into high memory at boot time.

The number of BIOS Services is always changing. The BIOS Services of PhoenixBIOS 4.05 provide precise control of hardware devices such as disk drives, which require careful management and exhaustive checking for errors. They also help manage new computer features such as Power Management, Plug and Play, and MultiBoot.

System Hardware Requirements

PhoenixBIOS 4.0 requires the following hardware components on the motherboard:

System Board Requirements
1. CPU (486 or later) 2. AT-compatible and MC146818 RTC-compatible chipset. 3. AT or PS/2-compatible Keyboard controller 4. At least 1 MB of system RAM

The Power On Self Test (POST) of the BIOS initializes additional ROM BIOS extensions (Option ROMs) if they are accessible in the proper format. The requirements are:

Adapter ROM Requirements

1. The code must reside in the address space between C0000H and F0000H.
2. The code must reside on a 2K boundary.
3. The first two bytes of the code must be 55H and AAH.
4. The third byte must contain the number of 512-byte blocks.
5. The fourth byte must contain a jump to the start of the initialization code.
6. The code must checksum to zero (byte sum).

NOTE: The address space from C0000H to C8000H is reserved for external video adapters (e.g. EGA, VGA). Part of the address space from D0000H to E0000H is typically used by expanded memory (EMS).

Fixed Disk Tables

PhoenixBIOS 4.0 supports up to four fixed-disk drives. For each drive, it supports 39 pre-defined drive types and four user-defined types (40-43). Below is a table of the pre-defined drive types and their default values.

End users can modify the user-defined drive type for each fixed disk listed in Setup by using the menus of the Setup program. This feature avoids the need for customized software for non-standard drives.

NOTE: Since most hard drives are autotyped (i.e., automatically determined by the BIOS or Operating System), there is usually no need to set the drive geometry manually.

Type	Cylinders	Heads	Sectors	Wrt Pre	Landing
1	306	4	17	128	305
2	615	4	17	300	615
3	615	6	17	300	615
4	940	4	17	512	940
5	940	6	17	512	940
6	615	4	17	-1	615
7	462	8	17	256	511
8	733	5	17	-1	733
9	900	15	17	-1	901
10	820	3	17	-1	820
11	855	5	17	-1	855
12	855	7	17	-1	855
13	306	8	17	128	319
14	733	7	17	-1	733
15	Reserved				
16	612	4	17	0	633
17	977	5	17	300	977
18	977	7	17	-1	977
19	1024	7	17	512	1023
20	733	5	17	300	732
21	733	7	17	300	732
22	733	5	17	300	733
23	306	4	17	0	336
24	612	4	17	305	663
25	612	2	17	300	612
26	614	4	17	-1	614
27	820	6	17	-1	820
28	977	5	17	-1	977
29	1218	15	36	-1	1218

Type	Cylinders	Heads	Sectors	Wrt Pre	Landing
30	1224	15	17	-1	1224
31	823	10	17	512	823
32	809	6	17	128	809
33	830	7	17	-1	830
34	830	10	17	-1	830
35	1024	5	17	-1	1024
36	1024	8	17	-1	1024
37	615	8	17	128	615
38	1024	8	26	-1	1024
39	925	9	17	-1	925
40	User def.				
41	User def.				
42	User def.				
43	User def.				

PhoenixBIOS Function Keys

The following are the special PhoenixBIOS function keys:

<F2>	Enter SETUP program during POST
Ctrl-Alt-<>	Switch to slow CPU speed
Ctrl-Alt-<+>	Switch to fast CPU speed

The speed switching keys are only operational when speed switching is available.

POST Errors and Beep Codes

Recoverable POST Errors

Whenever a recoverable error occurs during POST, *PhoenixBIOS* displays an error message describing the problem.

PhoenixBIOS also issues a beep code (one long tone followed by two short tones) during POST if the video configuration fails (no card installed or faulty) or if an external ROM module does not properly checksum to zero.

An external ROM module (e.g. VGA) can also issue audible errors, usually consisting of one long tone followed by a series of short tones.

Terminal POST Errors

There are several POST routines that issue a **POST Terminal Error** and shut down the system if they fail. Before shutting down the system, the terminal-error handler issues a beep code signifying the test point error, writes the error to port 80h, attempts to initialize the video, and writes the error in the upper left corner of the screen (using both mono and color adapters).

The routine derives the beep code from the test point error as follows:

1. The 8-bit error code is broken down to four 2-bit groups (Discard the most significant group if it is 00).
2. Each group is made one-based (1 through 4) by adding 1.
3. Short beeps are generated for the number in each group.

Example:

Test point 01Ah = 00 01 10 10 = 1-2-3-3 beeps

Test Points and Beep Codes

At the beginning of each POST routine, the BIOS outputs the test point error code to I/O address 80h. Use this code during trouble shooting to establish at what point the system failed and what routine was being performed.

PhoenixBIOS 4.0 Services

The ROM BIOS contains a number of useful run-time **BIOS Services** that are easily called by an outside program. As a programmer, you can execute these services, which are nothing more than subroutines, by invoking one of the BIOS interrupt routines (or, when specified, calling a protected-mode entry point and offset). Invoking a software interrupt causes the CPU to fetch an address from the **interrupt table** in low memory and execute the service routine. Some services return exit values in certain registers. All registers are preserved unless they return data or status.

Generally, a Carry flag set on exit indicates a failed service. A zero on exit in the AH register usually indicates no error; any other value is the service's **exit status code**.

BIOS32 Service Directory

While the standard BIOS services are accessed through the interrupt table, newer services are accessed by a FAR CALL to a service entry point. Programmers can determine the entry point by searching for a particular signature (such as "\$PnP") in the BIOS range and finding the entry point in the header.

The **BIOS32 Service Directory** (standard in PhoenixBIOS 4.0) provides a single entry point for all those services in the BIOS that are designed for BIOS clients running in a 32-bit code segment, such as 32-bit operating systems and 32-bit device drivers. The BIOS32 Service Directory itself is a 32-bit BIOS service that provides a single entry point for the other 32-bit services. For a full description of this service, see the **Standard BIOS 32-Bit Service Directory Proposal, Rev 0.4** published by Phoenix and available on the Phoenix Web site at:

<http://www.phoenix.com/products/specs.html>

Programs calling the 32-bit BIOS services should scan 0E0000h to 0FFFF0h on the 16-byte boundaries for the contiguous 16-byte data structure beginning with the ASCII signature "_32_".

If they do not find this data structure, then the platform does not support the BIOS32 Service Directory. The following chart describes the data structure.

Offset	Size	Description
0h	4 bytes	ASCII signature "_32_" Offset 0 = underscore Offset 1 = "3" Offset 2 = "2" Offset 3 = underscore
4h	4 bytes	Entry point for the BIOS32 Service Directory, a 32-bit physical address
8h	1 byte	Revision level. Currently 00h.
9h	1 byte	Length of this structure in 16-byte units. This structure is 16 bytes long, so the field = 01h.
0Ah	1 byte	Checksum of whole data structure. Result must be 0.
0Bh	5 bytes	Reserved. Must be zero.

Once the data structure is found and verified, the program can do a FAR CALL to the entry point specified in the above structure. The calling environment requires:

1. The CS code segment selector and the DS data segment selector must encompass the physical page of the entry point as well as the following page.
2. The SS stack segment selector must have available 1 kB of stack space.
3. Access to I/O space.

The BIOS32 Service Directory provides a single call that:

Interrupt 11h—Return System Information

This service returns the equipment installed as determined by the BIOS on power-up diagnostics and stored in the BIOS Data Area.

Interrupt 11 Return System Information	
Exit:	
AX	Equipment information:
Bit	Definition
0	Not used
1	Math coprocessor installed
2	PS/2 mouse installed
3	Not used
4,5	Initial video mode:
	00 = EGA/VGA
	01 = 40x25 CGA
	10 = 80x25 CGA
	11 = Monochrome
6,7	Diskette drives:
	00 = 1 drive
	01 = 2 drives
	10 = 3 drives
	11 = 4 drives
8	Not used
9-11	Number of serial adapters
12	Game Adapter installed
13	Not used
14,15	Number of parallel adapters

Interrupt 12h—Return Memory Size

Returns up to 640 kB of the amount of system memory determined by early POST diagnostics.

Interrupt 12 Return System Memory Size	
Exit:	
AX	Number of 1-kilobyte memory blocks

Interrupt 13h—Diskette Services

Interrupt 13 is the BIOS software interface for access to the 5- $\frac{1}{4}$ " and 3- $\frac{1}{2}$ " inch diskette drives. When there is a fixed disk in the system, the BIOS assigns Interrupt 13h to the fixed disk and routes diskette calls to Interrupt 40h.

The following table lists the AH error codes.

Int 13 Diskette Exit Status Codes	
AH	00h = No error
	If Carry = 1:
AH	01h = Illegal BIOS command
	02h = Bad address mark
	03h = Write-protect occurred
	04h = Sector not found
	06h = Media changed
	09h = DMA crossed 64K boundary
	08h = DMA failed
	0Ch = Media not found
	10h = CRC failed
	20h = NEC failed
	30h = Drive does not support media sense
	31h = No media in drive
	32h = Drive does not support media type
	40h = Seek failed
	80h = Time out occurred

The following table contains the combinations of drive types and media types supported by the INT 13 services 02h to 05h.

Media	Drive	Diskette Types	
		Sec/Trk	Tracks
360 kB	360 kB	8-9	40
360 kB	1.2 MB	8-9	40
1.2 MB	1.2 MB	15	80
720 kB	720 kB	9	80
720 kB	1.44 MB	9	80
1.44 MB	1.44 MB	18	80
720 kB	2.88 MB	9	80
1.44 MB	2.88 MB	18	90
2.88 MB	2.88 MB	36	80

The following describes the diskette services with their entry and exit values.

Interrupt 13h Diskette Services	
AH = 00h	Reset diskette system
AH = 01h	Return diskette status
Exit:	
AH	00h = No error 01h = Illegal BIOS command 02h = Address mark not found 03h = Write-protect error 04h = Sector not found 06h = Media has been changed 08h = DMA overrun 09h = DMA boundary error 0Ch = Media not found 10h = CRC error 20h = NEC error 40h = Seek error 80h = Time out occurred
AH = 02h	Read diskette sectors
Entry:	
ES:BX	Buffer address
DL	Drive number (0-1)
DH	Head number (0-1)
CH	Track number (0-79)
CL	Sector number (8-36)
AL	Number of sectors (1-15)
Exit:	
AL	Number of sectors transferred
AH = 03h	Write diskette sectors
Entry:	
ES:BX	Buffer address
DL	Drive number (0-1)
DH	Head number (0-1)
CH	Track number (0-79)
CL	Sector number (8-36)
AL	Number of sectors (1-15)
Exit:	
AL	Number of sectors transferred
AH = 04h	Verify diskette sectors
Entry:	
DL	Drive number (0-1)
DH	Head number (0-1)
CH	Track number (0-79)
CL	Sector number (8-36)
AL	Number of sectors (1-15)
Exit:	
AL	Number of sectors verified

Continued

*Interrupt 13h—Diskette Services, Continued***AH = 05h Format diskette track**

Entry:

ES:BX Buffer address

DL Drive number (0-1)

DH Head number (0-1)

CH Track number (0-79)

AL Number of sectors (1-15)

Exit:

AL Number of sectors formatted

AH = 08h Read drive parameters

Entry:

DL Drive number

Exit:

ES:DI Pointer to parameter table

DH Maximum head number

DL Number of diskette drives present

CH Maximum track number

CL Drive capacity:

Bits 0-5 Maximum sector number

Bits 6-7 Maximum track number

BL Diskette drive type from CMOS:

Bits 0-3:

00 = CMOS not present or invalid

01 = 360 kB

02 = 1.2 MB

03 = 720 kB

04 = 1.44 MB

06 = 2.88 MB

Bits 4-7: 0

AH = 15h Read drive type

Entry:

DL Drive number

Exit:

AH 00 = Drive not present

01 = Drive cannot detect media change

02 = Drive can detect media change

03 = Fixed disk

AH = 16h Detect media change

Entry:

DL Drive Number (0-1)

Exit:

If Carry = 0:

AH 00 = Disk change not active

01 = Invalid drive number

06 = Either disk change line active or
change line not supported80h = Drive not ready or no drive present:
(timeout)**AH = 17h Set diskette type**

Entry:

AL Format:

00 = Invalid Request

01 = 360kB floppy in 360kB drive

02 = 360kB floppy in 1.2MB drive

03 = 1.2MB floppy in 1.2MB drive

04 = 720kB floppy in 720kB (1.44MB not supported)

DL Drive Number (0-1)

AH = 18h Set media type for format

Entry:

CH Maximum track number

CL Diskette parameters:

Bits 0-5: Maximum sector number

Bits 6-7: Maximum track number

DL Drive Number (0-1)

Exit:

ES:DI Pointer to parameter table

Continued

Interrupt 13h—Diskette Services, Continued**AH = 20h** Get media type

Entry:
 DL Drive number (0-1)
 Exit:
 AL Type of media installed:
 00h = 720 kB diskette
 01h = 1.44 MB diskette
 02h = 2.88 MB diskette
 03h = 1 MB diskette
 04h = 2 MB diskette
 06h = 4 MB diskette

Interrupt 13h—Fixed Disk Services

Interrupt 13h accesses these Services:

- Standard Fixed-Disk Services, 00h-15h
- Enhanced Disk Drive Services, 41h -48h
- Bootable CD-ROM Services, 4Ah-4Dh

The following box lists the error codes:

Int 13h Fixed-Disk Exit Codes

AH 00h = No error
 If Carry = 1:
 AH 01 = Bad command or parameter
 02h = Address mark not found
 04h = Sector not found
 05h = Reset failed
 07h = Drive parameter activity failed
 0Ah = Bad sector flag detected
 10h = ECC data error
 11h = ECC data corrected
 20h = Controller failure
 40h = Seek failed
 80h = Time out occurred
 AAh = Drive not ready
 BBh = Undocumented controller error
 CCh = Controller write fault
 E0h = Unrecognized controller error

The following describes the Standard Fixed-Disk services of PhoenixBIOS 4.0:

Interrupt 13 Standard Fixed Disk Services**AH = 00** Reset diskette and fixed-disk systems**AH = 01h** Read disk status

Entry:
 DL Drive number (80h-81h)
 Exit:
 AH 001h = Bad command
 002h = Bad address mark
 004h = Record not found
 005h = Controller reset error
 007h = Drive initialization error
 00Ah = Bad sector
 010h = ECC data error
 020h = Controller failed
 040h = Seek error
 0AAh = Drive not ready
 0BBh = Invalid controller error
 0CCh = Controller write fault
 0E0h = Unrecognized controller error

Continued

*Interrupt 13h—Fixed Disk Services, Continued***AH = 02h Read disk sectors**

Entry:

ES:BX Buffer address

DL Drive number (80h-81h)

DH Head number (0-15)

CH Track number (0-1023)

Put the two high-order bits (8 and 9)
in the high-order bits of CL

CL Sector number (1-17)

AL Number of sectors (1-80h for read)
(1-79h for long read, includes ECC)

Exit:

AL Number of sectors transferred

AH = 03h Write disk sectors

Entry:

ES:BX Buffer address

DL Drive number (80H-81H)

DH Head number (0-15)

CH Track number (0-1023)

Put the two high-order bits (8 and 9)
in the high-order bits of CL

CL Sector number (1-17)

AL Number of sectors (1-80h for write)
(1-79h for long write, includes ECC)

Exit:

AL Number of sectors transferred

AH = 04h Verify disk sectors

Entry:

ES:BX Buffer address

DL Drive number (80h-81h)

DH Head number (0-15)

CH Track number (0-1023)

Put the two high-order bits (8 and 9)
in the high-order bits of CL

CL Sector number (1-17)

AL Number of sectors (1-80h for write)
(1-79h for long write, includes ECC)

Exit:

AL Number of sectors verified

AH = 05h Format disk cylinder

Entry:

ES:BX Pointer to table containing the
following byte pair for each sector
in the track:Byte 0: 00h if sector is good
80h if sector is bad

Byte 1: Sector Number (0-255)

DL Drive number (80H-81H)

DH Head number (0-15)

CH Track number (0-1023)

Put the two high-order bits (8 and 9)
in the high-order bits of CL

CL Sector number (1-17)

AL Number of sectors (1-80h for write)
(1-79h for long write, includes ECC)

Exit:

AL Number of sectors formatted

Continued

*Interrupt 13h—Fixed Disk Services, Continued***AH = 08h Read drive parameters**

Entry:
 DL Drive number (80H-81H)
 Exit:
 CL Maximum sector number
 CH Maximum cylinder number
 (High bits in CL)
 DH Maximum head number
 DL Number of responding drives (0-2)
 If Carry - 1:
 AH 07h = Invalid drive number
 AL 0 = Error
 CX 0 = Error
 DX 0 = Error

AH = 09h Initialize drive parameters

Entry:
 DL Drive number (80H-81H)

AH = 0Ah Read long sector

Entry:
 ES:BX Buffer address
 DL Drive number (80H-81H)
 DH Head number
 CH Cylinder number
 CL Sector number/Cyl high
 AL Number of sectors
 Exit:
 AL Number of sectors transferred

AH = 0Bh Write long sector

Entry:
 ES:BX Buffer address
 DL Drive number (80H-81H)
 DH Head number
 CH Cylinder number
 CL Sector number/Cyl high
 AL Number of sectors
 Exit:
 AL Number of sectors transferred

AH = 0Ch Seek drive

Entry:
 ES:BX Buffer address
 DL Drive number (80H-81H)
 DH Head number
 CH Cylinder number
 CL Cylinder high

AH = 0Dh Alternate disk reset

Entry:
 DL Drive number (80H-81H)

AH = 10h Test drive ready

Entry:
 DL Drive number (80H-81H)

AH = 11h Recalibrate drive

Entry:
 DL Drive number (80H-81H)

AH = 14h Controller diagnostic

Entry:
 DL Drive number (80H-81H)

AH = 15h Read drive type

Entry:
 DL Drive number (80H-81H)
 Exit:
 AH 00 = Drive not present
 01 = Drive cannot detect media change
 02 = Drive can detect media change
 03 = Fixed disk
 CX High word of number of 512-byte blocks
 DX Low word of number of 512-byte blocks

Interrupt 13h—Extended Fixed Disk Services

The following describes the Interrupt 13h Extended Fixed Disk Services, including the *PhoenixBIOS Enhanced Disk Drive (EDD)* services:

Int 13h Extended Fixed Disk Services	
AH = 41h	Check Extensions Present
Entry:	
BX	55AAh
DL	Drive Number
Exit:	
AH	Major version number (20h)
AL	Internal use only
BX	55AAh = Extensions present
CX	Feature support map:
	Bit 0: 1 = Extended disk access
	Bit 1: 1 = Removable drive control
	Bit 2: 1 = Enhanced Disk Drive Extensions
	Bits 3-7, Reserved, must be 0
AH = 42h	Extended Read
Entry:	
DL	Drive Number
DS:SI	Disk address packet
AH = 43h	Extended Write
Entry:	
AL	Verify Bits:
	Bit 0: 0 = Write with verify off
	1 = Write with verify on
	Bits 1-7 Reserved, set to 0
DL	Drive number
DS:SI	Disk address packet
AH = 44h	Verify Sectors
Entry:	
DL	Drive number
DS:SI	Disk address packet
AH = 47h	Extended Seek
Entry:	
DL	Drive number
DS:SI	Disk address packet
AH = 48h	Get Drive Parameters
Entry:	
DL	Drive Number
DS:SI	Address of Result Buffer
Exit:	
DS:SI	Pointer to Result Buffer:
	info_size dw 30 ;size of this buffer
	flags dw ? ;info flags (See below)
	cylinders dd ? ;cylinders on disk
	heads dd ? ;heads on disk
	sec_per_track dd ? ;sectors per track
	sectors dq ? ;sectors on disk
	sector_size dw ? ;bytes per sector
	extended_table dd ? ;extended table ptr
	; (See below)
	info flags:
	Bit 0 0 = DMA boundary errors possible
	1 = DMA errors handled
	Bit 1 0 = CHS info not supplied
	1 = CHS info valid
	Bit 2 0 = Drive not removable
	1 = Drive removable
	Bit 3 0 = No write with verify
	1 = Write with verify
	Bit 4 0 = No change-line support
	1 = Change-line support
	Bit 5 0 = Drive not lockable
	1 = Drive lockable
	Bit 6 0 = CHS values for installed media
	1 = Maximum CHS values for drive (media absent)

Continued

Interrupt 13h—Extended Fixed Disk Services, Continued

Extended Fixed Disk Parameter Table		
Byte	Type	Description
0-1	Word	I/O port address
2-3	Word	Control port address
4		Bit 0-3 Reserved, must be 0
		Bit 4 0 = Master, 1 = Slave
		Bit 5 Reserved, must be 0
		Bit 6 1 = LBA enabled
		Bit 7 Reserved, must be 1
5		Bits 0-3 Phoenix Proprietary
		Bits 4-7 Reserved, must be 0
6		Bits 0-3 IRQ for this drive
		Bits 4-7 Reserved, must be 0
7	Byte	Sector count for multi-sectored transfers
8		Bits 0-3 DMA channel
		Bits 4-7 DMA type
9		Bits 0-3 PIO type
		Bits 1-7 Reserved, must be 0
Byte	Type	Description
10-11		Bit 0 1 = Fast PIO access enabled
		Bit 1 1 = DMA access enabled
		Bit 2 1 = Block PIO access enabled
		Bit 3 1 = CHS translation enabled
		Bit 4 1 = LBA translation enabled
		Bit 5 1 = Removable media
		Bit 6 1 = CD ROM
		Bit 7 1 = 32-bit transfer mode
		Bit 8 1 = ATAPI Device uses Interrupt DRQ
		Bits 9-10 CHS Translation Type
		Bits 11-15 Reserved, must be 0
12-13	Byte	Reserved, must be 0
14	Byte	Extension Revision number
15	Byte	Checksum, 2s complement of the sum of bytes 0-14

Interrupt 13h—Bootable CD-ROM Services

Bootable CD-ROM Services 4Ah-4Ch use a pointer to the **Specification Packet**, described here:

Bootable CD-ROM Specification Packet		
Offset	Type	Description
0h	Byte	Packet size, currently 13h
1h	Byte	Boot media type: Bits 0-3: 00h = No emulation 01h = 1.2 MB diskette 02h = 1.44 MB diskette 03h = 2.88 MB diskette 04h = Hard disk (drive C:) Bits 05h-07h: Reserved Bit 6: 01h = System has ATAPI driver with 8 & 9 below describing IDE interface. Bit 7: 01h = System has SCSI drivers with 8 & 9 below describing SCSI interface
2h	Byte	Drive number: 00h = Floppy image 80 = Bootable hard disk 81h -FFh = "Non-bootable" or "No emulation"

Continued

Interrupt 13h—Bootable CD-ROM Services, Continued

Offset	Type	Description
3h	Byte	Controller index of CD drive
4h-7h	Dword	Logical Block Address
8h-9h	Word	Device specification: For SCSI: Byte 8: LUN and PUN of CD drive Byte 9: Bus number For IDE: Byte 8 LSB: 0 = Master, 1 = Slave
Ah-Bh	Word	User buffer segment
Ch-Dh	Word	Load segment (only for Int 13h 4Ch): 00h = 7C0h
Eh-Fh	Word	Virtual sector count (only for Int 13h 4Ch)
10h	Byte	Low-order bits (0-7) of the cylinder count (Matches returned CH of Int 13h 08h)
11h	Byte	Bits 0-5: Sector count Bits 6-7: High order 2 bits of cylinder count (Matches returned CL of Int 13h 08h)
12h	Byte	Head count (Matches returned DH of Int 13h 0h)

Bootable CD-ROM Service 4Dh uses a pointer to the **Command Packet**, described here:

Bootable CD-ROM Command Packet

Offset	Type	Description
0h	Byte	Packet size in bytes, currently 08h
1h	Byte	Count of sectors in boot catalog to transfer
2-h	Dword	Pointer to destination buffer for boot catalog
6-7h	Word	Beginning sector to transfer, relative to start of the boot catalog. Int 14 4Dh should set this value to 00h.

The following describes the Interrupt 13 Bootable CD-ROM Services of PhoenixBIOS 4.0:

Int 13 Bootable CD-ROM Services

AH = 4Ah	Initiate disk emulation
Entry:	
AL	00
DS:SI	Pointer to Specification Packet (See above)
CF	0 = Specified drive emulating 1 = System not in emulation mode
AH = 4Bh	Terminate disk emulation
Entry:	
AL	00h = Return status and terminate emulation 01h = Return status only, do not terminate
DL	Drive number to terminate 7Fh = Terminate all
DS:SI	Empty Specification Packet
Exit:	
DS:SI	Completed Specification Packet (See above)
AX	Exit status codes
CF	0 = System released 1 = System not in emulation mode
AH = 4Ch	Initiate disk emulation and boot
Entry:	
AL	00h
DS:SI	Specification Packet (See above)
AH = 4Dh	Return boot catalog
Entry:	
AL	00h
DS:SI	Point to Command Packet (See above)

Interrupt 14h—Serial Services

The INT 14 software interrupt handles serial I/O service requests. Use the AH register to specify the service to invoke. This describes the UART Modem and Line Status returned by these services. It also includes two services, 04h and 05h, that support the extended communication capabilities of PS/2.

Interrupt Vectors

The following table describes the AT system interrupt vectors. Status indicates whether the BIOS supports the interrupt.

INT	Description	Status
00	Divide by zero	Not Supported
01	Single step	Not Supported
02	Non-Maskable interrupt	Supported
03	Breakpoint	Not Supported
04	Overflow	Not Supported
05	Print Screen Interrupt	Supported
06	286 LoadAll Handler	Supported
07	Reserved	Not Supported
08	IRQ0 - System Timer Interrupt	Supported
09	IRQ1 - Keyboard Interrupt	Supported
0A	IRQ2 - Reserved	Not Supported
0B	IRQ3 - COM2: Interrupt	Supported
0C	IRQ4 - COM1: Interrupt	Supported
0D	IRQ5 - LPT2: Interrupt	Supported
0E	IRQ6 - Floppy Disk Interrupt	Supported
0F	IRQ7 - LPT1: Interrupt	Supported
10	BIOS Video Interface	Supported
11	BIOS Equipment Check	Supported
12	BIOS Memory Request	Supported
13	BIOS Fixed Disk/Diskette Interface	Supported
14	BIOS Serial Interface	Supported
15	BIOS System Functions Interface	Supported
16	BIOS Keyboard Interface	Supported
17	BIOS Parallel Printer Interface	Supported
18	BIOS Secondary Boot Request	Supported
19	BIOS Primary Boot Request	Supported
1A	BIOS System Timer Interface	Supported
1B	BIOS Control Break Interrupt	Supported
1C	BIOS User System Timer Interrupt	Supported
1D	BIOS Video Init Parameters	Supported
1E	BIOS Diskette Parameters	Supported
1F	BIOS Video Graphic Characters	Supported
40	BIOS Diskette (when fixed disk present)	Supported
41	BIOS Fixed disk 0 parameters	Supported
46	BIOS Fixed disk 1 parameters	Supported
70	IRQ8 - Real time clock interrupt	Supported
71	IRQ9 - IRQ2 redirection	Supported
72	IRQ10 - Reserved	Not Supported
73	IRQ11 - Reserved	Not Supported
74	IRQ12 - Available/PS/2 Mouse	Supported
75	IRQ13 - Math coprocessor	Supported
76	IRQ14 - Primary IDE HDD	Supported
77	IRQ15 - Available/Secondary IDE HDD	Supported