

# BIOS Enhanced Disk Drive Specification

Version 3.0

Rev 0.9b

June 24, 1999



Technical Editor:

Curtis E. Stevens  
Phoenix Technologies  
135 Technology Dr.  
Irvine, Ca. 92618  
Phone: (949) 790-2000  
Fax: (949) 790-2001  
Curtis\_Stevens@Phoenix.COM

Phoenix Technologies Ltd.

**THIS SPECIFICATION IS MADE AVAILABLE WITHOUT CHARGE FOR USE IN DEVELOPING COMPUTER SYSTEMS AND DISK DRIVES. PHOENIX MAKES NO REPRESENTATION OR WARRANTY REGARDING THIS SPECIFICATION OR ANY ITEM DEVELOPED BASED ON THIS SPECIFICATION, AND PHOENIX DISCLAIMS ALL EXPRESS AND IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND FREEDOM FROM INFRINGEMENT. WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, PHOENIX MAKES NO WARRANTY OF ANY KIND THAT ANY ITEM DEVELOPED BASED ON THIS SPECIFICATION WILL NOT INFRINGE ANY COPYRIGHT, PATENT, TRADE SECRET OR OTHER INTELLECTUAL PROPERTY RIGHT OF ANY PERSON OR ENTITY IN ANY COUNTRY. USE OF THIS SPECIFICATION FOR ANY PURPOSE IS AT THE RISK OF THE PERSON OR ENTITY USING IT.**

<b>Contributors</b>		
<b>Name</b>	<b>Company</b>	<b>E-Mail Address</b>
Mani Ayyer	Intel PCD	<a href="mailto:Mani_Ayyar@ccm.sc.intel.com">Mani_Ayyar@ccm.sc.intel.com</a>
Rick Bramley	Phoenix Technologies	<a href="mailto:Rick_Bramley@phoenix.com">Rick_Bramley@phoenix.com</a>
Mike Glass	Microsoft	<a href="mailto:Mglass@MICROSOFT.com">Mglass@MICROSOFT.com</a>
Don James	Compaq	<a href="mailto:Don.James@Compaq.com">Don.James@Compaq.com</a>
Robert Hale	Intel OPSD	<a href="mailto:Robert_P_Hale@ccm2.hf.intel.com">Robert_P_Hale@ccm2.hf.intel.com</a>
Dave Morrison	Microsoft	<a href="mailto:DaveMor@Microsoft.COM">DaveMor@Microsoft.COM</a>
Rajeev Nalawadi	Intel PCD	<a href="mailto:rajeev_k_nalawadi@CCM.FM.Intel.com">rajeev_k_nalawadi@CCM.FM.Intel.com</a>
Shaun Pierce	Microsoft	<a href="mailto:Shaunp@microsoft.com">Shaunp@microsoft.com</a>
Bob Riney	Microsoft	<a href="mailto:BobRi@Microsoft.COM">BobRi@Microsoft.COM</a>
Curtis Stevens	Phoenix Technologies	<a href="mailto:Curtis_stevens@phoenix.com">Curtis_stevens@phoenix.com</a>
Dong Wei	Hewlett Packard	<a href="mailto:dong_wei@hp.com">dong_wei@hp.com</a>
Mark Williams	Microsoft	<a href="mailto:Markwi@microsoft.com">Markwi@microsoft.com</a>

<b>Contents</b>	<b>Page</b>
1 Introduction .....	3
2 Scope.....	3
3 Overview .....	3
4 Calling conventions .....	3
4.1 Data Structure .....	4
4.2 Removable media .....	4
4.3 Int 13h interface subsets.....	5
4.3.1 Fixed disk access subset .....	5
4.3.2 Drive locking and ejecting subset .....	6
4.3.3 Enhanced disk drive (EDD) support subset .....	6
5 Int 13h extensions .....	6
5.1 Check extensions present.....	6
5.2 Extended read .....	7
5.3 Extended write.....	7
5.4 Verify sectors.....	7
5.5 Lock/unlock media.....	8
5.6 Eject removable media .....	8
5.7 Extended seek.....	9
5.8 Get drive parameters.....	9
5.8.1 Interface Path .....	11
5.8.2 Device Path.....	11
5.8.3 Device Parameter Table Extension (DPTE) .....	12
5.9 Get extended media change status.....	15
5.10 Set hardware configuration.....	15
5.11 Send Packet Command.....	16
6 Int 15h removable media eject .....	16

**Tables**

**Page**

TABLE 1 – DEVICE ADDRESS PACKET.....	5
TABLE 2 – EXTENSION RESULT BUFFER .....	7
TABLE 3 – RESULT BUFFER.....	9
TABLE 4 – INTERFACE PATH DEFINITIONS.....	11
TABLE 5 – DEVICE PATH DEFINITIONS .....	11
TABLE 6 – DEVICE PARAMETER TABLE EXTENSION.....	12
TABLE 7 – TRANSLATION TYPE.....	14
TABLE 8 – HARDWARE CONFIGURATION SUB-FUNCTIONS.....	15
TABLE 9 – FORMATTED COMMAND PACKET.....	16

## 1 Introduction

In the past, DOS has accessed its mass storage devices using a BIOS provided INT 13 interface. This interface was designed in the early 1980's and upgraded in the late 1980's. The maximum theoretical capacity of this API is 8.4 giga-bytes. This INT 13 interface, now known as the legacy INT 13 interface, uses function numbers 1-15h and is Cylinder-Head-Sector (CHS) oriented. An extended INT 13 interface has been created, the purpose of these Int 13h extensions is to:

- Replace CHS addressing with Logical Block Addressing (LBA).
- Remove the current requirement of using interrupt 41h/46h to point at the Fixed Disk Parameter Table information.
- Give the BIOS better control over how this data is used.
- Make location and configuration information available to operating systems that do not use the BIOS to access mass storage devices.

Many BIOS, Option ROM, and OS vendors have already implemented the extensions defined in this document for ATA and SCSI style devices. EDD 3.0 builds on EDD 1.1 to enable other mass storage technologies, such as 1394, Fibre Channel, and USB.

## 2 Scope

This document builds on EDD 1.1, available free of charge at [WWW.PHOENIX.COM](http://WWW.PHOENIX.COM). The reader should be familiar with EDD 1.1, legacy INT 13, and other related technologies such as ATA, ATAPI, 1394 and Fibre Channel. It is the intention of the sponsors of this document to create a unified work environment for PC users that allows the operating system to provide the same drive letters that DOS provides. Other related documents include:

- El Torito CD-ROM Boot Specification
- ATAPI Removable Media BIOS Specification
- EDD 1.1
- ATA/ATAPI-5
- IEEE 1394-1995
- SBP-2
- USB Mass Storage Bulk Only
- USB Mass Storage Control Bulk Interrupt

## 3 Overview

EDD 3.0 provides a linkage between the BIOS device assignments on the operating system drive letter assignments. There are several benefits provided by full implementation of this specification as follows:

- DOS and other operating systems, such as Win '98 and Win NT, will provide the same drive letter assignments to the user. The result of this capability is that new drives can be added to an EDD 3.0 system, and the existing drive letters will not change.
- New technologies, such as 1394, blur the difference between fixed and removable media. Compatibility with this specification requires that operating systems allow removable media devices, such as the Iomega Zip, Fujitsu MO, and MKE LS-120/SuperDisk, work equally well as A:, B:, C:, as well as D: and above. The problem being solved is that the media can have data on it that renders it incompatible with certain drive letters.

## 4 Calling conventions

The extended Int 13h functions are numbered 41h-48h. These new functions are fundamentally different from the legacy Int 13h interface in the following ways:

- Register conventions have been changed to support the passing of data structures;
- All media addressing information is passed via a buffer, not registers;
- Flags are used to identify optional capabilities.

#### 4.1 Data Structure

The data structure for the Int 13h extensions is the disk address packet. Int 13h converts addressing information in the device address packet to physical parameters appropriate to the media. Table 1 defines the device address packet.

#### 4.2 Removable media

The distinction between "removable" disks numbered 0-7Fh and "fixed" disks numbered 80h-FFh differ from conventional Int 13h functions. Drives numbered 0-7Fh are not changed, they follow conventional Int 13h standards for floppy disk operation. Drives numbered 80h-FFh include traditional fixed disks, and now also include removable media devices that support media change notification as well as software locking and unlocking capabilities. Functions in this technical report support these devices. Return codes defined for the conventional Int 13h interface are still valid, and the following return codes have been added to support removable media:

- B0h - Media Not Locked In Drive;
- B1h - Media Locked In Drive;
- B2h - Media Not Removable;
- B3h - Media In Use;
- B4h - Lock Count Exceeded;
- B5h - Valid Eject Request Failed;
- B6h - Media Present but Read Protected.

Table 1 - Device address packet

Offset	Type	Description
0	Byte	Packet size in bytes. Shall be 16 (10h) or greater. If the packet size is less than 16 the request is rejected with CF=1h and AH=01h.
1	Byte	Reserved, must be 0
2	Byte	Number of blocks to transfer. This field has a maximum value of 127 (7Fh). A block count of 0 means no data is transferred. If a value greater than 127 is supplied the request is rejected with CF=1 and AH=01. If this field is set to FFh, then the transfer buffer address is found at offset 10h, the number of blocks to transfer is found at offset 18h, and the transfer buffer at offset 4 is ignored.
3	Byte	Reserved, must be 0
4	Double word	Address of transfer buffer. The buffer which Read/Write operations will use to transfer the data. This is a 32-bit address of the form Seg:Offset. If this field is set to FFFF:FFFF then the address of the transfer buffer is found at offset 10h
8	Quad word	Starting logical block address, on the target device, of the data to be transferred. This is a 64 bit unsigned linear address. If the device supports LBA addressing this value should be passed unmodified. If the device does not support LBA addressing the following formula holds true when the address is converted to a CHS value:  $LBA = (C_1 * H_0 + H_1) * S_0 + S_1 - 1$ Where: $C_1$ = Selected Cylinder Number $H_0$ = Number of Heads (Maximum Head Number + 1) $H_1$ = Selected Head Number $S_0$ = Maximum Sector Number $S_1$ = Selected Sector Number  For ATA compatible drives, with less than or equal to 15,482,880 logical sectors, the $H_0$ and $S_0$ values are supplied by WORDS 3 and 6 of the IDENTIFY DEVICE command.
10h	Quad word	64 bit flat address of the transfer buffer. This is the buffer that Read/Write operations will use to transfer the data if the data at offset 4 is set to FFFFh:FFFFh.
18h	Double Word	Total number of block to transfer when the data at offset 2 is set to FFh
1Ch	Double Word	Reserved, must be 0.

### 4.3 Int 13h interface subsets

It is permissible for a BIOS to support only certain subsets of the Int 13h extensions. These subsets are defined in this technical report. If a subset is supported then all functions within that subset shall be supported. The supported subsets shall be determined via the check extensions present function. If a function is not supported and that function is subsequently invoked, then the function rejects the request with CF=1, AH=01h. There are three subsets defined, at least one of these shall be supported

#### 4.3.1 Fixed disk access subset

These functions support basic access to devices using the disk address packet structure as follows:

- Check extensions present (41h);
- Extended read (42h);
- Extended write (43h);
- Verify sectors (44h);
- Extended seek (47h);
- Get drive parameters (48h).

### 4.3.2 Drive locking and ejecting subset

These functions support software control of media locking and ejecting as follows:

- Check extensions present (41h);
- Lock/unlock media (45h);
- Eject drive (46h);
- Get drive parameters (48h);
- Get extended disk change status (49h);
- The Int 15h removable media eject intercept.

### 4.3.3 Enhanced disk drive (EDD) support subset

These functions provide EDD support as follows:

- Check extensions present (41h);
- Get parameters with EDD extensions (48h);

## 5 Int 13h extensions

The INT 13 extensions define an API for accessing a variety of mass storage devices, up to 16 mega-tera sectors ( $2^{64}$  sectors) in size. The expected lifetime of this interface is in excess of 15 years,

### 5.1 Check extensions present

Entry:

AH - 41h  
 BX - 55AAh  
 DL - Drive number

Exit:

carry clear  
 AH - Version of extensions = 30h  
 AL - Internal use only  
 BX - AA55h  
 CX - Interface support bit map (see Table 2 )

carry set  
 AH - error code (01h, Invalid Command)

**Table 2 - Extension result buffer**

Bit	Description
0	1 – Fixed disk access subset
1	1 – Drive locking and ejecting subset
2	1 – Enhanced disk drive support subset
3	1 – 64 bit extensions are present. When the 64 bit extensions are present, the fields starting at offset 10h in the Device Address Packet are valid
4-15	Reserved, must be 0

This function is used to check for the presence of Int 13h extensions. If the carry flag is returned set, the extensions are not supported for the requested drive. If the carry flag is returned cleared, BX shall be checked for the value AA55h to confirm that the extensions are present. If BX is AA55h, the value of CX is checked to determine what subsets of this interface are supported for the requested drive. At least one subset must be supported. The version of the extensions is 30h. This indicates that the Int 13h extensions are compliant with this specification.

## 5.2 Extended read

Entry:

AH - 42h  
DL - Drive number  
DS:SI - Disk address packet

Exit:

carry clear  
AH - 0  
carry set  
AH - error code

This function transfer sectors from the device to memory. In the event of an error, the block count field of the disk address packet contains the number of good blocks read before the error occurred.

## 5.3 Extended write

Entry:

AH - 43h  
AL - 0 or 1, write with verify off  
2, write with verify on  
DL - Drive number  
DS:SI - Disk address packet

Exit:

carry clear  
AH - 0  
carry set  
AH - error code

This function transfer sectors from memory to the device. If write with verify is not supported, this function rejects the request with AH=01h, CF=1. Function 48h is used to detect if write with verify is supported. In the event of an error, the block count field of the disk address packet contains the number of blocks written before the error occurred. AL also contains the values 0, 1, or 2. This function rejects all other values with AH=01h, CF=1

## 5.4 Verify sectors

Entry:

AH - 44h  
 DL - Drive number  
 DS:SI - Disk address packet

Exit:

carry clear  
     AH - 0  
 carry set  
     AH - error code

This function verifies sectors without transferring data between the device and system memory. When an error is reported the block count field of the disk address packet is filled in with the number of blocks verified before the error occurred.

## 5.5 Lock/unlock media

Entry:

AH - 45h  
 AL - 0 - Lock media in drive  
     1 - Unlock media in drive  
     2 - Return lock/unlock status  
     3h-FFh - Invalid  
 DL - Drive number

Exit:

carry clear  
     AH - 0  
     AL - 1 if device is locked, 0 if not  
 carry set  
     AH - error code.

This function logically locks/unlocks removable media in a specific device. All removable media devices numbered 80h and above require this function. If a fixed disk (non-removable device) supports the media locking and ejecting subset, this function always returns with success, AH=0, CF=0. There must be support for up to 255 locks per device. A device shall not be unlocked until all locks to that device have been released with unlock commands. Excess unlock calls return with carry set and AH = B0h, "Device Not Locked". If the number of locks supported value is exceeded on a lock request, this function rejects the request with carry set and AH = B4h, "Lock Count Exceeded". Locking a device without media present is a valid operation. On return from a lock or unlock request, AL contains the lock state of the media as maintained by the BIOS. This provides for unlock requests when the lock count is greater than 0. In this case, the media remains locked. Any physical locking and unlocking of the media is implementation dependent, but system software operates on the assumption that locked media cannot be removed without an unlock request. After power-on, or a system reset, all devices automatically enter an unlocked state.

## 5.6 Eject removable media

Entry:

AH - 46h  
 AL - 0h  
 DL - Drive number

Exit:

carry clear  
     AH - 0  
 carry set  
     AH - error code

This function will eject media from the specified device. If a fixed disk (non-removable device) supports the media locking and ejecting interface subset, this function always returns CF=1, AH = B2h, "Volume Not Removable". An attempt to eject media locked in a device must return with CF=1, AH = B1h, "Media Locked In Device". This function represents a request to remove media from the selected device. Actual ejection is implementation

dependent, but system software that issues or observes this function should flush any buffers it is holding. If this function is issued for a device without media the request is returned with CF=1, AH = 31h, "No Media In Device". If this call is issued to an unlocked removable media device that has media present, an Int 15h, Fn 52h (removable media eject) is issued to determine if eject removable media may proceed with the ejection request. If Int 15h returns an error the ejection request is rejected. If the ejection request is accepted, followed by an unrecoverable error, this function returns with CF=1, AH = B5h "Valid Eject Request Failed".

## 5.7 Extended seek

Entry:

AH - 47h  
DL - Drive number  
DS:SI - Disk address packet

Exit:

carry clear  
AH - 0  
carry set  
AH - error code

This function allows the host to provide advanced notification that particular data may be requested by the host in a subsequent command. This command initiates a seek operation. The seek may not be complete when this function completes.

## 5.8 Get drive parameters

Entry:

AH - 48h  
DL - Drive number  
DS:SI - address of result buffer.

Exit:

carry clear  
AH - 0  
DS:SI - address of result buffer  
carry set  
AH - error code

This function returns physical device parameters. It is mandatory regardless of the interface subset that is supported. Table 3 defines the result buffer. On entry the first word of the result buffer must be the bufer length in bytes.

**Table 3 - Result buffer**

Offset	Type	Description						
0	Word	The caller sets this value to the maximum buffer size. If the length of this buffer is less than 30 bytes, this function does not return the pointer to DPT extension. If the buffer size is 30 or greater on entry, it shall be set to 30 on exit. If the buffer size is between 26 and 29, it shall be set to 26 on exit. If the buffer size is less than 26 on entry an error is returned.						
2	Word	Information Flags. A 1 bit indicates that the feature is available, a 0 bit indicates the feature is not available and will operate in a manner consistent with the conventional Int 13h interface.						
		<table border="0"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>DMA boundary errors are handled transparently</td> </tr> <tr> <td>1</td> <td>The geometry returned in bytes 4-15 is valid</td> </tr> </tbody> </table>	Bit	Description	0	DMA boundary errors are handled transparently	1	The geometry returned in bytes 4-15 is valid
Bit	Description							
0	DMA boundary errors are handled transparently							
1	The geometry returned in bytes 4-15 is valid							

Offset	Type	Description
		2 Media is removable. Bits 4-6 are not valid if this bit is 0 3 Device supports write verify 4 Device has media change notification 5 Media is lockable 6 Device geometry is set to maximum and no media is present when this bit is set to one 7-15 Reserved
4	Double word	Number of physical cylinders. This is 1 greater than the maximum cylinder number. Use Int 13h Fn 08h to find the logical number of cylinders.
8	Double word	Number of physical heads. This is 1 greater than the maximum head number. Use Int 13h Fn 08h to find the logical number of heads.
12	Double word	Number of physical sectors per track. This number is the same as the maximum sector number because sector addresses are 1 based. Use Int 13h Fn 08h to find the logical number of sectors per track.
16	Quad word	Number of physical sectors. This is 1 greater than the maximum sector number. If this field is greater than 15,482,880 then word 2, bit 1 is cleared to 0.
24	Word	Number of bytes in a sector.
26	Double word	Pointer to the Device Parameter Table Extension (DPTE). This field follows the seg:offset address format. The DPTE is only present if Int 13h, Fn 41h, CX register bit 2 is equal to 1. This field points to a temporary buffer that the BIOS may invalidate on subsequent Int 13h calls. A value of FFFFh:FFFFh in this field means that the pointer is invalid. If the length of this result buffer is less than 30, the DPTE is not present. This field is only used for legacy ATA hard drive configurations.
30	Word	0BEDDh – Key, indicates presence of Device Path Information
32	Byte	Length of Device Path Information including the key. = 44
33	Byte	Reserved = 0
34	Word	Reserved = 0
36	ASCII	Host bus type, 4 bytes PCI PCI Local Bus ISA Legacy 16 bit fixed bus
40	ASCII	Interface type, 8 bytes ATA ATA/ATAPI-4 compliant device using ATA commands ATAPI ATA/ATAPI-4 compliant device using ATAPI commands SCSI SCSI compliant device USB USB Mass Storage compliant device 1394 1394 Mass Storage device FIBRE Fibre Channel
48	Qword	Interface Path, 8 bytes. See below for format information
56	Double Qword	Device Path. See below for format information.
72	Byte	Reserved = 0
73	Byte	Checksum for Device Path Information includes the 0BEDDh signature. 2's complement of the sum of offset 30-72. The sum of offset 30-73 is 0.

### 5.8.1 Interface Path

The Interface Path field at offset 48 allows software external to a system BIOS to locate mass storage device interface chips. The format of this field is dependent on the Host Bus type, offset 36-39 of the result buffer. The following formats are currently defined:

**Table 4 - Interface Path Definitions**

Host Bus Type	Offset	Type	Definition
Legacy	48	Word	16 bit base address
	50	Word	Reserved = 0
	52	Dword	Reserved = 0
PCI	48	Byte	Bus
	49	Byte	Slot
	50	Byte	Function
	51	Byte	Reserved = 0
	52	Dword	Reserved = 0

### 5.8.2 Device Path

The Device Path at offset 56 combined with the Interface Path allows software external to a system BIOS to locate a specific mass storage device. The Device Path field provides a path from an interface to a specific device. The format of the Device Path is dependent on the Interface type, offset 40-47. The following formats are currently defined:

**Table 5 - Device Path Definitions**

Host Bus Type	Offset	Type	Definition
ATA	56	Byte	0 = Master, 1 = Slave
	57	Byte	0 = Primary Channel, 1 = Secondary Channel
	58	Word	Reserved = 0
	60	Dword	Reserved = 0
	64	Qword	Reserved = 0
ATAPI	56	Byte	0 = Master, 1 = Slave
	57	Byte	0 = Primary Channel, 1 = Secondary Channel
	58	Byte	Logical Unit Number
	59	Byte	Reserved = 0
	60	Dword	Reserved = 0
SCSI	64	QWord	Reserved = 0
	56	Word	Physical Unit Number/SCSI ID
	58	Qword	Logical Unit Number
	66	Word	Reserved = 0
	68	Dword	Reserved = 0

USB	56	QWord	64 Bit Serial Number as defined in the USB Mass Storage specification
	64	QWord	Reserved = 0
1394	56	Qword	64 bit General Unique Identifier (GUID)
	64	QWord	Reserved = 0
FIBRE	56	QWord	64 bit Word Wide Number (WWN)
	64	QWord	Reserved = 0

### 5.8.3 Device Parameter Table Extension (DPTE)

The DPTE provides hardware configuration information to applications that bypass Int 13h for accessing a legacy ATA device.

**Table 6 - Device parameter table extension**

Offset	Type	Description
0-1	Word	I/O port base address
2-3	Word	Control port address
4	Byte	Head register upper nibble bit 0-3     0 bit 4     ATA DEV bit bit 5     1 bit 6     LBA enable (1 = enabled) bit 7     1
5	Byte	BIOS Vendor Specific.
6	Byte	IRQ information bits 0-3    IRQ for this drive bits 4-7    0
7	Byte	Block count for ATA READ/WRITE MULTIPLE commands
8	Byte	DMA information bits 0-3    DMA channel bits 4-7    DMA type
9	Byte	PIO information bits 0-3    PIO type bits 4-7    0
10-11	Word	BIOS selected hardware specific option flags bit 0     Fast PIO accessing enabled bit 1     DMA accessing enabled bit 2     ATA READ/WRITE MULTIPLE accessing enabled bit 3     CHS translation enabled bit 4     LBA translation enabled bit 5     Removable media bit 6     ATAPI device bit 7     32-bit transfer mode bit 8     ATAPI device uses command packet interrupt bits 9-10   Translation type bit 11    Ultra DMA accessing enabled bits 12-15   Reserved, shall be 0
12-13h	Word	Reserved, shall be 0
14	Byte	11h, revision level of this table.
15	Byte	Checksum, 2's complement of the 8 bit unsigned sum of bytes 0-14

#### 5.8.3.1 Offset 0-1 - I/O port base

This word is the address of the data register in the ATA Command Block. Any application that provides a proprietary interface to the device may use this base address.

#### 5.8.3.2 Offset 2-3 - control port base

This word is the address of the ATA Control Block register. Any application that provides a proprietary interface to the device may use this address.

#### 5.8.3.3 Offset 4 - head prefix

The upper nibble of this byte is logically ORed with the head number, or upper 4 bits of the LBA, each time the disk is addressed. It contains the ATA DEV bit and the LBA addressing bit which are preset, and makes these functions transparent to any software using this extension.

#### 5.8.3.4 Offset 5 - Internal use only

For BIOS use only.

#### 5.8.3.5 Offset 6 - IRQ number

Each ATA channel requires an interrupt. This byte identifies which IRQ is used by this device's channel.

#### 5.8.3.6 Offset 7 - READ/WRITE MULTIPLE command block count

If the drive was configured to use the READ/WRITE MULTIPLE command, then this field contains the block size of the transfer, in sectors, used by the BIOS.

#### 5.8.3.7 Offset 8 - DMA channel/Multiword DMA Type

If the BIOS has configured the system to perform multi-word DMA data transfers in place of the normal PIO transfers, this field specifies the DMA mode in the upper nibble, as per the ATA-2 or later definition, and the DMA Channel in the lower nibble. ATA Channels that conform to SFF-8038i set the DMA channel to 0. Note that the DMA Type field does not follow the format of the data returned by the drive. The value of the DMA mode is not limited to 2.

#### 5.8.3.8 Offset 9 - PIO type

If the BIOS has configured the system to perform PIO data transfers other than mode 0, this field specifies the PIO mode as per the ATA-2 or later definition.

#### 5.8.3.9 Offset 10-11 - BIOS selected hardware specific option flags

These bytes specify the current system configured enabled by the BIOS. They have a bit for each of the options listed below.

##### 5.8.3.9.1 Bit 0 - fast PIO

If the system is configured for a PIO mode greater than 0, this bit is set to 1 and byte 9 (PIO Type) shall be used to configure the system. If this bit is 0, the PIO-Type field shall be ignored.

##### 5.8.3.9.2 Bit 1 - fast DMA

If the system is configured for DMA, this bit is set to 1 and byte 8 (DMA Channel/DMA Type) should be used to configure the system. If this bit and bit 11, section 3.5.9.11, are 0, then the DMA Channel/DMA Type field shall be ignored.

**5.8.3.9.3 Bit 2 - ATA READ/WRITE MULTIPLE**

If the system is configured for multi-sector transfers, this bit is set to 1 and byte 7 (sector count) specifies the number of sectors used for each data transfer. If block PIO is disabled, ignore the block count field.

**5.8.3.9.4 Bit 3 - CHS translation**

If the disk-drive reports more than 1024 cylinders in the IDENTIFY DEVICE command data, this bit is set to 1.

**5.8.3.9.5 Bit 4 - LBA translation**

If the system is configured for LBA type addressing, this bit is set to 1. When LBA translation is on, the Extended Int 13h interface (Fn 41h-48h) pass LBA values directly to the device. The conventional Int 13h interface ignores this bit and always uses CHS. LBA-type addressing is available on drives with less than 1024 cylinders, and therefore bit 3 (CHS translation) is independent from bit 4 (LBA translation).

**5.8.3.9.6 Bit 5 - removable media**

If the device supports removable media, this bit is set to 1 and the extended Int 13h drive locking and ejecting subset shall also be supported.

**5.8.3.9.7 Bit 6 - ATAPI device**

If this ATA device uses the packet interface (ATAPI) as defined in ATA/ATAPI-4, this bit is set to 1.

**5.8.3.9.8 Bit 7 - 32-bit transfer mode**

If the BIOS has configured the host adapter to perform 32-bit wide data transfers, this bit is set to 1.

**5.8.3.9.9 Bit 8 - ATAPI device uses command packet interrupt**

If bit 6 is set to zero, then this field is ignored and must be 0. If bit 6 is set to one, this bit indicates how the ATAPI devices signals it is ready to receive a packet command. When this bit is 1, it indicates that the ATAPI device returns an interrupt, and sets DRQ, when it is ready for a packet. When this bit is 0, it indicates that the ATAPI device sets DRQ, without an interrupt, when it is ready for a packet.

**5.8.3.9.10 Bits 9-10 - translation type**

If bit 3 is zero then this field is ignored and must be 0. If bit 3 is 1 then this field identifies the geometric translation shown in Table 7 .

**Table 7 - Translation type**

Bits 9-10	Description
00	Bit-shift translation
01	LBA assisted translation
10	Reserved
11	Vendor specific translation

**5.8.3.9.11 Bit 11 - Ultra DMA**

If the system is configured for Ultra DMA, this bit is set to 1 and byte 8 (DMA Channel/DMA Type) should be used to configure the system. If this bit and bit 1, section 3.5.9.2, are 0, then the DMA Channel/DMA Type field shall be ignored.

**5.8.3.9.12 Bits 12-15 - Reserved**

Shall be set to 0.

#### 5.8.3.10 Offset 12-13 - Reserved

Shall be set to 0.

#### 5.8.3.11 Offset 14 - table revision

This is set to 11h and represents the version of this table.

#### 5.8.3.12 Offset 15 - checksum

This is the two's complement of the 8 bit unsigned sum of bytes 0 through 14. Adding bytes 0 through 15 shall in all cases produce an 8 bit result of 0.

### 5.9 Get extended media change status

Entry:

AH - 49h  
DL - Drive number

Exit:

carry clear  
AH - 00, change-line inactive  
carry set  
AH - 06, change-line active

This function returns media change status. If it returns with carry flag set, the media has not necessarily been changed; the media change notification may be activated by simply unlocking and locking the device door without removing the media. This function corresponds to Int 13h Function 16h, but explicitly allows any drive number to be passed in. If a non-removable device supports the Drive Locking and Ejecting interface subset, this function always returns with success, AH=0h, CF=0h. This function clears the media change notification on exit.

### 5.10 Set hardware configuration

Entry:

AH - 4Eh  
AL - Hardware configuration sub-function (see 9 )  
DL - Drive Number.

Exit:

carry clear  
AH - 0  
AL - 0 if command was safe  
1 if other devices are affected  
carry set  
AH - error code

**Table 8 - Hardware configuration sub-functions**

AL	Sub-function description
0h	Enable prefetch
1h	Disable prefetch
2h	Set maximum PIO transfer mode.
3h	Set PIO mode 0. Clear to the minimum PIO transfer rate.
4h	Return to default PIO transfer mode. Return the system to the PIO mode enabled by the BIOS setup utility.

5h	Enable Int 13h DMA maximum mode. Set the maximum rate allowed by both the host adapter and the device.
6h	Disable Int 13h DMA

The purpose of this function is to allow non-hardware-specific software to configure host adapter and devices for optimal operation. ATA channels may have 2 devices attached, but this function operates on a single-device basis. This is accommodated by the value that is returned in AL. If the host adapter supports the requested sub-function on a device basis, AL is set to 0. If the host adapter only supports the setting on an ATA channel basis, AL is set to 1. Once this function has been invoked, all subsequent Int 13h device-access functions use the mode specified by this invocation. This means that if “DMA Maximum” is enabled, Int 13h Fn 02h reads from the device using DMA transfers. The DMA/PIO selections are mutually exclusive. When “DMA Maximum” is enabled, “PIO Maximum” is disabled. If the requested mode change is not supported this function returns with CF=1 and AH=1

## 5.11 Send Packet Command

Entry:

AH - 50h  
 AL - 01  
 DL – Drive Number  
 ES:BX – Pointer to formatted command packet, (see Table 9).

Exit:

carry clear  
 AH - 0  
 carry set  
 AH - error code

**Table 9 - Formatted Command Packet**

Offset	Type	Description
0	Word	Key = 0B055hh
2	Byte	Length of this record in bytes
3	Byte	Reserved = 0
4-n	Byte	Formatted Packet

This function defines a service that the system BIOS will call for sending data to and from a serial packet oriented device. The BIOS will provide this service before the OS is loaded. When an operating system takes control of the serial device controller it takes the hook for this service to provide a seamless transfer of control from the BIOS to the operating system. This service allows several BIOS level services to continue functioning, even after the OS has taken control of the 1394 controller, as follows:

- The INT 13 mass storage interface
- Power Management
- Suspend to disk

The BIOS is a single threaded, master device. This means that the BIOS will not process asynchronous requests from other devices. The BIOS will send commands to devices and wait for responses. This means that the operating system can take control of the serial interface with no hand-off information from the system BIOS. The operating system will reconfigure the interface and hook the service described above. The system BIOS may provide INT 13h Fn 50h for the 1394, USB, and any other serial bus.

The format of the packet is determined by the requirements of the target bus and is beyond the scope of this document. In the case of USB, the packet format is determined by the USB specification. In the case of 1394, IEEE 1394-1995 determines the format of a packet with payload information defined in SBP-2.

## 6 Int 15h removable media eject

Entry:

AH - 52h  
DL - Drive number

Exit:

carry clear  
    AH - 0, ejection may proceed  
carry set  
    AH - error code, B1h or B3h, ejection is rejected

This function is called in response to a software request (Int 13h, AH=46h, Eject drive) to eject media from a removable media device.

Typically a user will press an eject button or use a software command to request that a particular media be ejected. By default the Int 15h handler returns with ejection accepted status. A disk cache program could hook this Int 15h call and return acceptance or rejection based on the state of its buffers for this disk. It may also be used by operating system software as a media change request