Working Draft **T13 1226DT**

Revision 6

25 July, 1997

Information Technology - Enhanced BIOS Services For Disk Drives

This is an internal technical report of T13, a Technical Committee of Accredited Standards Committee NCITS. As such, this is not a completed technical report and has not been approved. The contents may be modified by the T13 Technical Committee. The contents are actively being modified by T13. This technical report is made available for review and comment only.

ASC T13 Technical Editor:

Curtis E. Stevens and Mike Christensen
Phoenix Technologies LTD
135 Technology
Irvine Ca. 92618

Tel: (714) 790-2000
Fax: (714) 790-2001
E-Mail: Curtis_Stevens@Phoenix.COM

Reference number
ANSI TR X3.*** - 199x
Printed July, 25, 1997 9:20PM

Other Points of Contact:

T13 Chair
Gene Milligan
Seagate Technology
OKM 251
10323 West Reno (West Dock)
P.O. Box 12313
Oklahoma City, OK 73157-2313
Tel: 405-324-3070
Fax: 405-324-3794
E-mail: gene_milligan@notes.seagate.com

T13 Vice-Chair
Pete McLean
Maxtor Corporation
2190 Miller Drive
Longmont, CO 80501
Tel: 303-678-2149
Fax: 303-682-4811
E-mail: pete_mclean@maxtor.com

NCITS Secretariat
Administrator Standards Processing
1250 Eye Street, NW Suite 200
Washington, DC 20005
Tel: 202-737-8888
Fax: 202-638-4922
Email: NCITS@ITIC.NW.DC.US

T13 Reflector
Internet address for subscription to the T13 reflector: majordomo@dt.wdc.com
Send email to above account and include in BODY of text, on a line by itself the following:
"subscribe t13 [your email address]"
Internet address for distribution via T13 reflector: T13@dt.wdc.com

T13 Anonymous FTP Site
fission.dt.wdc.com
T13 directory is: "/T13"

T13 mailings
Global Engineering
15 Inverness Way East
Englewood, CO 80112-5704
Tel: 303-792-2181 or 800-854-7179
Fax: 303-792-2192

Revision History		
Rev	**Date**	**Description**
0	October 23, 1996	Initial Release
1	November 12, 1996	Changed style to be more in line with ANSI standards
2	February 18, 1997	Made changes based on a line by line review on 14-NOV-96
3	May 19, 1997	Fixed spelling errors Changed the name of the report to: “Enhanced BIOS Services For Disk Drives” Changed Fixed Disk Parameter Table Extension to be Device Parameter Table Extension Changed formatting to match ANSI requirements as per Pete McLean Added a note regarding bit shifting and possible DOS compatibility issues
4	June 3, 1997	Converted document format and style Added clause 2.3 to explain support for drives greater than 7.9 GB Added a statement to INT 13 FN 48 requiring the drive geometry be removed when the drive capacity exceeds 7.9 GB Added bibliography Changed notes in Table 2 and Table 3 Modified table 3 to state all values in sectors
5	July 2, 1997	Added clause 3.5.9.11, 3.5.9.12 and 3.5.10 Removed clause 4.1.2 Renamed clause 4.2.11 to 4.3 Added ATA-3, ATA/ATAPI-4, SFF 8038i and IBM PS/2 technical reference to to the bibliography Added a table between table 8 and 9. Added table 11 Removed note at the end of table 6 General grammar corrections
6	July 25, 1997	Added changes based on feedback from the working group. These changes include: Updated reflector address Modifications to some of the wordings in the definition of terms Fixed conflicting references to Fixed Disk vs Device

		Minor re-work of the 528MB barrier description

This page left intentionally blank

ANSI®

X3.*-199x**

American National Standard

for Information Systems —

Enhanced BIOS Services for Disk Drives

Secretariat

Information Technology Industry Council

Approved mm dd yy

American National Standards Institute, Inc.

Abstract

This technical report describes new services provided by BIOS firmware to support ATA hard disks up to 16 mega-tera-bytes ($16x10^{18}$). Older BIOS services have a compatibility limit of 528 MB and a theoretical limit of 8.4 GB.

NCITS's Technical Report Series

This Technical Report is one of a series produced by the American National Standards Committee, NCITS, Information Technology. The secretariat for X3 is held by the Computer and Business Equipment Manufacturers Association (CBEMA), 1250 Eye Street, NW Suite 200, Washington DC 20005.

As a by-product of the standards development process and the resources of knowledge devoted to it, NCITS from time to time produces Technical Reports. Such Technical Reports are not standards, nor are they intended to by used as such.

NCITS Technical Reports are produced in some cases to disseminate the technical and logical concepts reflected in standards already published or under development. In other cases, they derive from studies in areas where it is found premature to develop a standard due to still changing technology, or inappropriate to develop a rigorous standard due to the existence of a number of viable options, the choice of which depends on the users particular requirements. These Technical Reports, thus, provide guidelines, the use of which may result in greater consistency and coherence of information processing systems.

When the draft Technical Report is completed, the Technical Committee approval process is the same as for a draft standard. Processing by NCITS is also similar to that for a draft standard.

Published by

American National Standards Institute

11 West 42nd Street, New York, New York 10036

Contents **Page**

Tables **Page**

Foreword

(This foreward is not part of American National Standard X3.***-199x.)

This technical report describes new services provided by BIOS firmware to support ATA hard disks up to 16 mega-tera-bytes ($16x10^{18}$). Older BIOS services have a compatibility limit of 528 MB and a theoretical limit of 8.4 GB.

This technical report was developed by the ATA/ATAPI ad hoc working group of T13 during 1996-1997. The approval process started in 199x. This technical report includes an annex that is informative and is not considered part of the technical report.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the NCITS Secretariat, Information Technology Industry Council, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This technical report was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, NCITS. Committee approval of the technical does not necessarily imply that all committee members voted for approval. At the time it approved this technical, the NCITS Committee had the following members:

James D. Converse, Chairman

Donald C. Loughry, Vice-Chairman

Joanne M. Flanagan, Secretary

Organization Represented	Name of Representative
American Nuclear Society	Geraldine C. Main
	Sally Hartzell (Alt.)
AMP, Inc.	Edward Kelly
	Charles Brill (Alt.)
Apple Computer	Karen Higginbottom
Association of the Institute for Certification of Professionals (AICCP)	Kennath Zemrowski
AT&T/NCR	Thomas W. Kern
	Thomas F. Frost (Alt.)
Boeing Company	Catherine Howells
	Andrea Vanosdoll (Alt.)
Bull HN Information Systems, Inc.	William George
Compaq Computer Corporation	James Barnes
Digital Equipment Corporation	Delbert Shoemaker
	Kevin Lewis (Alt.)
Eastman Kodak	James D. Converse

Michael Nier (Alt.)
GUIDE International.....Frank Kirshenbaum
Harold Kuneke (Alt.)
Hewlett-Packard .....Donald C. Loughry
Hitachi America, Ltd.....John Neumann
Kei Yamashita (Alt.)
Hughes Aircraft Company.....Harold L. Zebrack
IBM Corporation.....Joel Urman
Mary Anne Lawler (Alt.)
National Communication Systems.....Dennis Bodson
National Institute of Standards and Technology.....Robert E. Roundtree
Michael Hogan (Alt.)
Northern Telecom, Inc. .....Mel Woinsky
Subhash Patel (Alt.)

Neville & Associates	Carlton Neville
Recognition Technology Users Association	Herbert P. Schantz
	G. Edwin Hale (Alt.)
Share, Inc.	Gary Ainsworth
	David Thewlis (Alt.)
Sony Corporation	Michael Deese
Storage Technology Corporation	Joseph S. Zajaczkowski
	Samuel D. Cheatham (Alt.)
Sun Microsystems	Scott Jameson
	Gary Robinson (Alt.)
3M Company	Eddie T. Morioka
	Paul D. Jahnke (Alt.)
Unisys Corporation	John L. Hill
	Stephen P. Oksala (Alt.)
U.S. Department of Defense	William C. Rinehuls
	C. J. Pasquariello (Alt.)
U.S. Department of Energy	Alton Cox
	Lawrence A. Wasson (Alt.)
U.S. General Services Administration	Douglas Arai
	Larry L. Jackson (Alt.)
Wintergreen Information Services	Joun Wheeler
Xerox Corporation	Dwight McBain
	Roy Peirce (Alt.)

Technical committee T13 on ATA interfaces, which reviewed this standard, had the following members:

G. E. Milligan, Chairman

Peter T. McLean, Vice-Chairman

Lawrence J. Lamers, Secretary

I. Dal Allan	Bill McFerrin	Wayne Baldwin (Alt.)
Darrin Bulik	Masa Morizumi	Carl Bonke (Alt.)
Joe Chen	Marc Noblitt	Les Cline (Alt.)
Dan Colegrove	Dennis Pak	Stephen Finch (Alt.)
Greg Elkins	Duncan Penman	Robin Freeze (Alt.)
Mark Evans	Paul Raikunen	Richard Harcourt (Alt.)
Tony Goodfellow	Yogi Schaffner	LeRoy Leach (Alt.)

Tom Hanan
Richard Kalish
Konichi Kasima
Hale Landis
Robert Liu
Alan Longo
J. R. Sims
Curtis Stevens
Tokuyuk Totani
Dennis Van Dalsen
Anthony Yang
John Masiewicz (Alt.)
James McGrath (Alt.)
Patrick Mercer (Alt.)
Ron Roberts (Alt.)
Devon Worrell (Alt.)

Other ad hoc participants were:

Michael Aarans
Lyle Adams
Michael Alexenko
Joe Bennett
John Brooks
Peter Brown
Ian Davies
Pat Edsall
David Evans
Mike Flora
Takayuk Fujioka
Parami Gill
Mark Gurkowski
Jon Haines
Jonathan Hanmann
Yas Hashimoto
Yoshihito Higashitsutsumi
Steve Horeff
Edward Hoskins
Stan Huyge
Bob Jackson
Jerry Kachlic
Kelvin Kao
Prakash Kamath
Yasu Kinoshita
Curtiss Krueger
Jesse Kup
Tony Kwan
Lane Lee
Min-Yi Li
Roger Li
Sam Lin
Marvin Lum
Kent Manabe
Gerald Marazas
Hisashi Nakamura
Kristin Nguyen
Michael Nguyen
Danny Ong
Charles Patton
Brett Philip
Anthony Pione
Doug Prins
Jim Randall
Steve Reames
Jeff Reid
David Roe
Richard Schnell
Karl Schuh
Mark Shipman
Randeep Sidhu
Neil Sugie
Steve Timm
Kevin Tso
Motoyas Tsunoda
Mark Vallis
Chi Wang
Keji Watanabe
Bill Willette
Tom Wood
John Wright
Chi-Che Wu
Daniel Wu
Steven Xu
Charles Yang
Mike Yokoyama

Introduction

This technical report encompasses the following:

Clause 1 describes the scope.

Clause 2 provides definitions, abbreviations, and conventions used within this technical report.

Clause 3 describes the parameter tables the BIOS makes available to the OS and application software.

Clause 4 describes the BIOS extensions.

Clause 5 describes compatibility issues.

This page left intentionally blank

AMERICAN NATIONAL STANDARD **X3.***-199x**

American National Standard

for Information Systems —

Information Technology —

Enhanced Bios Services For Disk Drives

1 Scope

This technical report assumes that the reader is familiar with the conventional Int 13h interface, the usage of the Device Parameter Table, and the basic operation of ATA devices. This technical report describes in detail extensions to the BIOS and extensions to the data maintained by the BIOS.

The disk-drive industry has increased the capacity and functionality of the ATA-compatible disk drive, finally surpassing the capability of some BIOSs to service these new capabilities. This specification addresses the following BIOS specific problems:

- The BIOS must support drives with a capacity greater than 528 MB. The conventional Int 13h interface has a limit of 1024 cylinders;
- The Int 13h interface allows more than two drives to be attached to a system but has no consistent method for storing the additional parameters;
- CHS-independent methods for accessing the drives have now been defined. These are drive-geometry independent and require a different method of address representation and operation;
- New methods of data transfer continue to be added to ATA devices. Capabilities such as, DMA modes, multi-sector data transfers and Fast PIO are not reported to the operating system;
- Systems require more than two disk drives, and with this requirement comes the requirement to assign the order in which the drives are to be accessed.

2 Definitions and abbreviations

2.1 Enhanced BIOS

All Enhanced BIOSs shall adhere to this technical report. Enhanced BIOSs should support at least one of the following:

- PIO Mode 3 or greater;
- DMA Mode 1 or greater.

2.2 Enhanced ATA Device

An Enhanced ATA Device is a hard disk or other device which interfaces to the system via Integrated Drive Electronics (IDE). These devices must conform to ATA-2(X3.279-1996)/ATAPI(SFF-8020 rev 2.6) or later and shall support:

- PIO Mode 3 or greater and;
- DMA Mode 1 or greater.

2.3 Enhanced ATA Channel

An Enhanced ATA Channel (or Chip or interface card) provides a communications port with an Enhanced ATA Device. These channels must be at least ATA-2(X3.279-1996)/ATAPI(SFF-8020 rev 2.6) compliant and have one of the following capabilities:

- PIO Mode 3 or greater;
- DMA Mode 1 or greater.

2.4 Conventional vs enhanced

When a word, term, or phrase is modified by the word “conventional” it refers to the legacy style, or method of operation which is limited to addressing ATA devices which have a 528 MB capacity or below. When a word, term, or phrase is modified by the word “enhanced” it means there is a “conventional” and an “enhanced” method of operation, the “enhanced” method is defined by this technical report.

2.5 Logical Address

A logical address is used to access a device by an application, such as DOS, using the Int 13 interface. Int 13 Fn 8 returns the logical geometry of the device.

2.6 Physical Address

A physical address is used to access the drive using the ATA command block registers. Words 1, 3, and 6 of the IDENTIFY DEVICE data define the physical geometry of a device.

3 Device Parameter Tables (DPT)

This clause provides a description of the parameter tables that the BIOS makes available to OS and application software. Register calling conventions limit the conventional Int 13h interface, resulting in the limitation of OS and application software as well. The Int 13h interface operates directly from the DPT and therefore limits the size of the cylinder field in this table.

This technical report defines a compatible version of the DPT, that has been implemented by some operating system (OS) developers, that allows the BIOS to change the way the geometry is represented and then to translate the changed ("logical") geometry back to actual ("physical") geometry when a disk access is performed. This Enhanced FDPT is identified by a signature byte (Axh) that indicates to the system that some form of translation is taking place.

The DPT is directly accessible only for Int 13h device numbers 80h and 81h. Int 41h and Int 46h provide pointers directly to the DPT for drive 80h and 81h respectively. These pointers are maintained for backward compatibility with older DOS applications only. Geometry information for drive numbers 82h and above is available only through Int 13h function (Fn) 8h and Fn 48h.

3.1 Device Parameter Table (DPT) Extensions

It has become necessary for the BIOS to store information representing the type of translation currently in operation, as well as any information relating to the current operation of the drive. One purpose of this technical report is to define a standard format for this extension area and to document its fields and functions so that other Operating System/Application software may easily use this information. This new information, called the "Device Parameter Table Extension" is 16 bytes long and is accessed via the extended Int 13h functions described in this technical report. See Table 4 , Table 5 , and Table 6 for a layout of the Device Parameter Table with its Extension.

3.2 The 528-megabyte barrier

BIOSs provide Int 13h services for accessing ATA drives from DOS. For conventional Int 13h the Cylinder-Head-Sector (CHS) values supplied to the Int 13h interface were passed to the drive without modification. This method of access allows “ill-behaved” applications to successfully access the drive, bypassing the BIOSs Int 13h interface. ATA drives support more than 1024 cylinders but the Int 13h interface is limited to 1024, this prevents the BIOS from accessing the full media by passing CHS values directly to the drive. Table 1 illustrates the limitations caused by the differences between the Int 13h and ATA maximum geometries.

Table 1 – Disk drive min/max

	BIOS	ATA	Limit
Max sectors/track	63	255	63
Max heads	256	16	16
Max cylinders	1024	65536	1024
Capacity	8.4 GB	136.9 GB	528 MB

This table illustrates how the conventional Int 13h interface with an 8.4 GB limit is restricted to 528 MB (63 * 16 * 1024 * 512). One solution to this problem is to address the drive using the Int 13h Extensions described

in this technical report. Another solution is to create a false geometry that "fits" within Int 13h limitations, and also uses the full capacity of the drive. This capability is called geometric or drive translation. The translated geometry is applied in a manner that causes all sectors to maintain the same physical location on the media as when the drive is used in an untranslated environment. The Int 13h interface only has 10 bits for the cylinder, therefore Int 13h Fn 08h always returns the altered geometry information. This allows all DOS applications to function normally. Windows™ 3.11 and below functions normally when 32-bit disk access mode is disabled. A Windows™ driver which supports the geometry reported by Int 13h Fn 08h is required for 32-bit protected disk access mode.

Due to the lack of a recognized standard, several independent drive translation methods have been implemented which do not conform to this technical report, thus causing a drive interchange problem. The root of the problem is in the boot code and Partition Table stored at CHS=0,0,1. The boot sequence assumes that the geometry stored in the Partition Table matches the geometry returned by Int 13h Fn 08h. When a drive is moved from the original system where the partitions were defined, to a new system with a different translation method, the boot sequence will use Int 13h with boundaries defined by the Partition Table in the boot sector. Boot failures are then caused because the geometry in the partition table does not match the BIOS geometry.

A simple bit-shift mapping scheme may create altered drive geometries. This method has the advantage of working with all ATA drives, including those drives which do not support LBA. A second advantage is that operation is fast and the code is small. The disadvantage of this method is that it lacks the flexibility to translate all geometries reported by a drive with a capacity less than 8.4 GB. However, drives which are ATA-2 (X3.279-1996) and above compatible will report geometries that may be translated. Annex D of ATA-2 or Annex B of ATA-3 and ATA/ATAPI-4 place limits on geometries for drives with less than an 8.4 GB capacity. The bit-shift method of translation manipulates the head and cylinder part of the geometry, but not the sectors per track. Table 2 describes the bit-shift translation capability:

Table 2 – Bit Shift Translation

Actual cylinders	Actual heads	Altered cylinder	Altered heads (see note)	Approx. size
1<C≤1024	1<H≤16	C=C	H=H	528 MB
1024<C≤2048	1<H≤16	C=C/2	H=H*2	1 GB
2048<C≤4096	1<H≤16	C=C/4	H=H*4	2.1 GB
4096<C≤8192	1<H≤16	C=C/8	H=H*8	4.2 GB
8192<C≤16384	1<H≤16	C=C/16	H=H*16	8.4 GB
16384<C≤32768	1<H≤8	C=C/32	H=H*32	8.4 GB
32768<C≤65536	1<H≤4	C=C/64	H=H*64	8.4 GB
NOTE – Value can not be greater than 255 in some Operating Systems.				

Another method of translation exists which is flexible and works well on drives that support logical block addressing (LBA). This method places no limits on the reported drive geometry. The disadvantage of this method is that it does not function well on drives that do not support LBA. Table 3 describes the LBA assisted translation method:

Table 3 – LBA assist translation

Range	Sectors	Heads	Cylinders
1<X≤1,032,192	63	16	X/(1,008)
1,032,192<X≤2,064,384	63	32	X/(2,016)
2,064,384<X≤4,128,768	63	64	X/(4,032)
4,128,768<X≤8,257,536	63	128	X/(8,064)
8,257,536<X≤16,450,560	63	255	X/(16,065)
NOTE – X is the capacity of the drive, calculated by multiplying words 1, 3, and 6 of the IDENTIFY DEVICE data. This number may be different than the drive size reported by IDENTIFY DEVICE words 60 and 61.			

These two translation methods yield similar geometries in many cases. The difference between the two translations methods becomes apparent when a drive reports less than 63 sectors per track. The LBA assisted method always assigns a geometry with 63 sectors per track. The bit-shift method uses the sectors returned by the drive. This technical report provides a way to report the type of translation employed by the BIOS. Currently there is no solution for the interchange problem.

3.3 Device capacities greater than 7.9 GB

The apparent capacity of the INT 13 interface is 16,514,064 sectors. However, there is a bug is some Operating Systems which prevents using the full INT 13 capacity. Devices that report more than 15,481,935 sectors in words 60 and 61 of the IDENTIFY DEVICE command data require additional support for addressing their full capacity. The limit of the conventional INT 13 interface is described in Table 1 . The Enhanced INT 13 functions described in this technical report allow full accessing of the device, up to 2^{64} sectors. Int 13 Extensions will allow access beyond 15,482,880 sectors.

3.3.1 Bit shift

The limit of bit shift addressing is 15,481,935 sectors-. Devices with more than 15,481,935 sectors report the following in ID DEVICE command: word 1 is 16,383, word 3 should be 15, and word 6 is 63. If the device happens to report word 3 as 16, the host issues INITIALIZE DEVICE PARAMETERS command with a value of 15 heads and 63 sectors per track. These rules allow legacy systems to access the first 15,481,935 sectors, the full device is then addressed by device drivers.

3.3.2 LBA assist

The limit of LBA Assist addressing is 16,450,560 sectors. Devices that follow the IDENTIFY DEVICE command recommendation in 3.3.1 will report 15,481,935 sectors, this guarantees compatibility with both translation methods. If a drive reports the following IDENTIFY DEVICE command data: word 1 between 8,192 and 16,383, and word 3 equal to 16; LBA Assist can access the full space described. However, systems that follow the recommendations in 3.3.1 will use a different geometry to address the drive, and the drive capacity is still limited to 15,481,935 sectors.

3.4 Device Parameter Table (DPT) definitions

Table 4 and Table 5 describe two DPTs. The standard DPT, described in Table 4 , is used for devices with 1024 or fewer cylinders.Table 5 , the translated DPT, is used for devices with more than 1024 cylinders (devices which require translation). These tables are only available for drives 80h and 81h.

3.4.1 Physical values

Physical values are the values returned by the IDENTIFY DEVICE command in words 1, 3 and 6. In Table 4 these values are limited to 1024 cylinders, 16 heads and 63 sectors. In Table 5 the limits are 65536 cylinders, 16 heads, and 63 sectors.

3.4.2 Logical values

Logical values only appear in the translated table. These values represent the geometry which the BIOS is receiving from applications that use Int 13h to address the device. Int 13h will then convert the Logical Values it receives to Physical values when it accesses the device. Logical values are limited to 1024 Cylinders, 256 Heads, and 63 Sectors per Track.

3.4.3 Obsolete fields

The pre-compensation and Landing Zone fields are obsolete because they are handled internally by today's ATA devices. These two fields are documented for legacy reasons only.

Table 4 – Standard device parameter table

Byte	Type	Description
0-1	Word	Physical number of cylinders
2	Byte	Physical number of heads
3	Byte	Not Axh signature, indicates untranslated table
4	Byte	Reserved
5-6	Word	Precompensation (obsolete)
7	Byte	Reserved
8	Byte	Drive control byte
9-10	Word	Reserved
11	Byte	Reserved
12-13	Word	Landing zone (obsolete)
14	Byte	Sectors per track
15	Byte	Reserved

Table 5 – Translated device parameter table

Byte	Type	Description
0-1	Word	Logical cylinders, limit 1024
2	Byte	Logical heads, limit 256 (see note)
3	Byte	Axh signature, indicates translated table
4	Byte	Physical sectors per track, limit 63
5-6	Word	Precompensation (obsolete)
7	Byte	Reserved
8	Byte	Drive control byte
9-10	Word	Physical cylinders, limit 65536 (see note)
11	Byte	Physical heads , limit 16 (see note)
12-13	Word	Landing zone (obsolete)
14	Byte	Logical sectors per track, limit 63
15	Byte	Checksum, 2's complement of the 8 bit unsigned sum of bytes 0-14
NOTE – 0 indicates the maximum value		

Table 6 – Device parameter table extension

Byte	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

3.5 Device Parameter Table (DPT) Extension

The DPT Extension provides hardware configuration information to applications that bypass Int 13h for accessing an ATA device. An application receives a pointer to the DPT Extension by issuing Int 13h Fn 48h.

3.5.1 Bytes 0-1 - I/O port base

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

3.5.2 Bytes 2-3 - control port base

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

3.5.3 Byte 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.

3.5.4 Byte 5 - Internal use only

For BIOS use only.

3.5.5 Byte 6 - IRQ number

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

3.5.6 Byte 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.

3.5.7 Byte 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 which 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.

3.5.8 Byte 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.

3.5.9 Byte 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.

3.5.9.1 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.

3.5.9.2 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.

3.5.9.3 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.

3.5.9.4 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.

3.5.9.5 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).

3.5.9.6 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.

3.5.9.7 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.

3.5.9.8 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.

3.5.9.9 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.

3.5.9.10 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

3.5.9.11 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.

3.5.9.12 Bits 12-15 - Reserved

Shall be set to 0.

3.5.10 Bytes 12-13 - Reserved

Shall be set to 0.

3.5.11 Byte 14 - table revision

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

3.5.12 Byte 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.

4 BIOS extensions

For compatibility with current and future software several extended Int 13h functions are defined to deal with the changes in system architecture. The purpose of these Int 13h Extensions is to remove the current requirement of using interrupt 41h/46h to point at the FDPT information, give the BIOS better control over how this data is used, and to make the DPT extension available to application and OS software.

4.1 Extended Int 13h calling conventions

The extended Int 13h functions are numbered 41h-48h. These new functions are fundamentally different from the conventional 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.

4.1.1 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 9 defines the device address packet.

4.1.2 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 9 – 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. Packet sizes greater than 16 are not rejected, the additional bytes beyond 16 shall be ignored.
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.
3	Byte	Reserved, must be 0
4	Double word	Address of transfer buffer. The is the buffer which Read/Write operations will use to transfer the data. This is a 32-bit address of the form Seg:Offset.
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, the CHS geometry is supplied by WORDS 1, 3 and 6 of the IDENTIFY DEVICE command. Note: The total logical sectors can also be read directly from WORD 60 and 61 of the IDENTIFY DEVICE command.

4.1.4 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

4.1.4.1 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).

4.1.4.2 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.

4.1.4.3 Enhanced disk drive (EDD) support subset

These functions provide EDD support as follows:

- Check extensions present (41h);
- Get parameters with EDD extensions (48h);
- Set hardware configuration (4Eh).

4.2 Int 13h extensions

4.2.1 Check extensions present

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

Exit:
carry clear
AH - Version of extensions
AL - Internal use only
BX - AA55h
CX - Interface support bit map (see Table 10)

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

Table 10 – 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
3-15	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 21h. This indicates that the Int 13h extensions are compliant with this technical report.

4.2.2 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.

4.2.3 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

4.2.4 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.

4.2.5 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 will Logically lock/unlock removable media in a specified 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.

4.2.6 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”.

4.2.7 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.

4.2.8 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 which is supported. Table 12 defines the result buffer. On entry the first word of the result buffer must be the bufer length in bytes.

Table 12 – 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. **Bit** – **Description** 0 – DMA boundary errors are handled transparently 1 – The geometry returned in bytes 4-15 is valid 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 then 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 DPT extension of the form seg:offset. This field 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.

4.2.9 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 notificiation 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.

4.2.10 Set hardware configuration

Entry:

AH - 4Eh
AL - Hardware configuration sub-function (see Table 13)
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 13 – 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

4.3 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.

5 Compatibility issues

Adding new features to the BIOS, or changing the functionality of an existing BIOS, may impact software performance. The purpose of this section is to discuss how to handle some of the anticipated issues.

5.1 Int 41h/46h

There may be system-specific issues associated with Int 41h and Int 46h. When a system supports Int 41h/46h, the BIOS uses Int 41h to point to an FDPT for drive 80h, and Int 46h to point to an FDPT for drive 81h. These pointers are maintained for backward compatibility purposes only. DOS/Windows applications should use Int 13 to get geometry information. The BIOS maintains pointers to drives greater than 81h internally. If the system does not support Int 41h/46h, the BIOS maintains these tables internally. In either case, Int 13h Fns 8h and 48h always return the requested information.

5.2 Disk drive mapping

The system needs a way to set the relationship between the BIOS references (80h, 81h, 82h, ...) and the physical drives. The DPT extension allows this capability by storing the physical port addresses and control bits used for setting device 0/1 as well as LBA accessing. One side effect of this implementation is that any device may be the boot device, and device 1 may be at any BIOS reference number. There shall be no gaps in the BIOS reference numbers. The following are some examples of allowable configurations:

Standard Configuration:

80h - Primary device 0 [Int 41h]

81h - Primary device 1 [Int 46h]

Alternate Configurations:

80h - Primary device 0 [Int 41h]
81h - Secondary device 0 [Int 46h]
82h - Secondary device 1

80h - Secondary device 0 [Int 41h]
81h - Primary device 0 [Int 46h]
82h - Secondary device 1

80h - Secondary device 1 [Int 41h]
81h - Secondary device 0 [Int 46h]
82h - Primary device 0
83h - Primary device 1

5.3 Geometric translations

Applications that obtain device geometry information reading the tables which are accessed via the Int 41h/46h pointers and do not call Int 13h Fn 08h or 48h are ill-behaved. Ill-behaved applications fall into two categories:

1. Those that read the Int 41h data and then use the conventional Int 13h interface for accessing the device. These are “compatible” ill-behaved applications.
2. Those that read the Int 41h/46h data and then access the drive in a proprietary manner. These are incompatible ill-behaved applications.

5.3.1 Compatible ill-behaved applications

Compatible ill-behaved applications require that offset 0, 2, and 14 (cylinder, head, and sector) information, in the FDPT, be identical to the information returned in Int 13h Fn 08h.

5.3.2 Incompatible ill-behaved applications

Incompatible ill-behaved applications require that offset 0, 2, and 14 information, in the FDPT, have the geometry returned by IDENTIFY DEVICE command data words 1, 3, and 6, a requirement that may violate restrictions placed on standard FDPTs. Further, these incompatible ill-behaved applications may not check for the translated FDPT signature (Axh at byte 3).

5.3.3 Resolving geometric translation compatibility issues

The BIOS may only accomodate one class of these ill-behaved applications each time the system boots. This presents the BIOS and the user with a compatibility problem. A solution is to add a setup field which allows the user to select which ill-behaved applications will function correctly. An example menu item reads “Large Disk Access Mode”. This field may default to “DOS”, which creates a translated FDPT. Compatible ill-behaved applications may operate correctly when “DOS” is selected.

Another selection for “Large Disk Access Mode” may be “OTHER”. Incompatible ill-behaved applications may function correctly with “OTHER,” which creates a standard FDPT.

Because this format uses only physical geometries, “OTHER” creates problems for the compatible ill-behaved applications by generating an illegal standard FDPT with more than 1024 cylinders. The conventional Int 13h interface, however, continues to use a translated FDPT, which is maintained internally by the BIOS, and is accessible only through Int 13h Fn 08h. BIOS setup should never change the method of translation used by itself. Well behaved DOS and windows applications continue to function normally because they use Int 13h Fn 08h, which returns translated geometry.

Annex A

(informative)

Bibliography

AT Attachment Interface (ATA-1), ANSI X3.221-1994

AT Attachment Interface with Extensions (ATA-2), ANSI X3.279-1996

AT Attachment Interface with Extensions (ATA-3), ANSI X3.298-1997

AT Attachment Interface with Packet Interface Extensions (ATA/ATAPI-4), ANSI X3.***-199x

Small Form Factors (SFF) 8038i

System BIOS for IBM PCs, Compatibles and EISA Computers, Addison Wesley, ISBN 0-201-57760-20, First printing May, 1991

IBM Personal Computer AT Technical Reference, IBM Boca Raton, FL, 1984

IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference, S68X-2341 and S10G-6467.

El Torito CD-ROM Boot Specification, Phoenix Technologies Limited, Irvine, CA, January 1995