

archive.is

からアーカイブ

検索

2013年10月6日 09:26:56 UTC

webpage capture

このURLからは、他のスナップショットはありません

すべてのスクリーンショット, ドメインから [web.inter.nl.net](http://web.inter.nl.net)

ウェブページ

スクリーンショット



シェア

ダウンロード .zip

エラー報告または乱用

# Harddisk Upgrade: The BIOS IDE Harddisk Limitations

by Jan Steunebrink  
Version 2, June 1999

This has become quite a story! Instead of reading the [HTML version below](#), you can also download the Microsoft Word 6.0/7.0 version of this article.

- [Download BIOSLIM2.DOC now](#) (14 KB ZIP-file)

The utilities mentioned in this article can also be downloaded here.

- [Download the WD Table Check utility](#) (56 KB)
- [Download the BIOS Int 13h Extensions detection utility](#) (6 KB ZIP-file)

## The BIOS IDE Harddisk Limitations

### Contents

- [Introduction](#)
- [The Interrupt 13h interface](#)
- [The 504 MB BIOS Limit](#)
- [The translating BIOS](#)
  1. [Bit-shifting translation](#)
  2. [LBA assisted translation](#)
- [Does my BIOS support translation?](#)
- [The 2 GB BIOS Limit](#)
- [The 4 GB Limit](#)
- [The Int 13h extensions – Breaking the 8 GB barrier](#)
- [More information on the net.](#)

### Introduction

In the last few years, harddisks have become larger and faster at an incredible rate.

Due to the favorable prices of the present large Enhanced IDE drives and the ever expanding software footprint, the whole class of 486 and Pentiums PCs can benefit of a harddisk upgrade.

This article targets at these PCs that have a system BIOS dated from 1992 to 1998.

These BIOSes can limit the usable capacity of your new (E)IDE drive.

Note: All MB values in this article are 1,048,576 Bytes (1024x1024).

## The Interrupt 13h Interface

When IBM engineered the AT they put the interface for the harddisk (Int 13h) in the system BIOS.

Whenever an application wants to read from/write to a drive, it calls DOS.

DOS knows the structure of the disk and where the target file is located. It then calculates a CHS (Cylinder, Head, Sector) address and calls the BIOS via Int 13h.

The BIOS then executes the read or write command at this CHS address by accessing the HD controller directly via its I/O-port addresses.

The result is passed back to DOS who passes it back to the application.

This scheme makes DOS (drive) hardware independent and leaves the hardware specifics to the BIOS.

The traditional BIOS Int 13h interface has the following limitations (when called from DOS):

1024 Cylinders, 256 Heads and 63 Sectors/track.

With 512 bytes/sector this counts up to 8 GB (8064 MB)!

## The 504 MB BIOS Limit

The (in)famous 504 MB (528 million bytes) limit comes from the fact that the ATA (IDE) specification has different limits than the BIOS. When the traditional BIOS Int 13h interface is used to control an (E)IDE harddisk, the limits are combined as illustrated below.

	BIOS	IDE	Combined Limit
Max. Sectors/track	63	255	63
Max. Heads	256	16	16
Max. Cylinders	1024	65536	1024
Max. Capacity	8 GB	127.5 GB	504 MB

If you do not have an Enhanced BIOS, you need one of these solutions to break the 504 MB barrier:

1. A translating (Enhanced) BIOS upgrade
2. Add-in card with an Enhanced BIOS (this takes over the Int 13h interface only)
3. Software like [Disk Manager from Ontrack](#) or [EZ-Drive from StorageSoft](#) (formerly Micro House).

## The translating BIOS

The traditional BIOS Int 13h interface passes the CHS address directly onto the harddisk controller, thereby creating the 504 MB barrier in case of an (E)IDE drive.

%

Nowadays we have the translating BIOS whose Int 13h interface can translate the CHS address to a different geometry or to an LBA (Logical Block Address).

The CHS which DOS uses to call the BIOS is now called the L-CHS (Logical CHS) and the CHS which the BIOS uses to control the drive is the P-CHS (Physical CHS).

Two translation methods can be distinguished:

1. The first method is normally used for direct L-CHS to P-CHS translation.

According the ATA-2 specification, all EIDE drives up to 8 GB should conform to the (BIOS) Sector limit of 63 sectors/track. On a >504 MB drive with a maximum of 16 Heads (IDE limit), only the number of Cylinders will be above the BIOS limit of 1024.

This makes a simple CHS translation scheme possible in which the number of Cylinders is divided by 2, 4, 8 or 16 and the number of Heads is multiplied with the same number.

The number of sectors/track will remain unchanged.

The maximum capacity depends on the number of sectors/track and is equal to the BIOS limit of 8064 MB if the drive has 63 sectors/track.

The following table illustrates this.

Actual Cylinders	Actual Heads	Altered Cylinders	Altered Heads	Max. Capacity
$1 < C \leq 1024$	$1 < H \leq 16$	$C=C$	$H=H$	504 MB
$1024 < C \leq 2048$	$1 < H \leq 16$	$C=C/2$	$H=H*2$	1008 MB
$2048 < C \leq 4096$	$1 < H \leq 16$	$C=C/4$	$H=H*4$	2016 MB
$4096 < C \leq 8192$	$1 < H \leq 16$	$C=C/8$	$H=H*8$	4032 MB
$8192 < C \leq 16384$	$1 < H \leq 16$	$C=C/16$	$H=H*16$	8064 MB

Example: A 2014MB drive has a CHS count of 4092x16x63. According to the above method, this will be translated into 1023x64x63, nicely within the Int 13h interface limit.

In other words; DOS (or any other OS which uses the BIOS Int 13h interface for that matter) thinks that it deals with a drive which has 1023 Cylinders, 64 Heads and 63 Sectors/track while the BIOS still controls the drive with its original geometry.

Note: A drive which has 16 Heads and more than 8192 Cylinders (> 4 GB) will be translated to 256 Heads. This is a problem because DOS can't handle 256 Heads, but there is an easy workaround for that. See [The 4 GB Limit](#) below.

Because the translation is done with powers of 2, the software inside the BIOS can accomplish this by simply shifting the bit patrons of the Cylinder and Head addresses one or more places to the right respectively to the left.

This translation is therefore known as the bit-shifting translation.

2. The second method is used with drives that can be accessed with LBA.

The translation here is from L-CHS (DOS) to LBA (drive).

With LBA, all sectors are numbered sequentially and the P-CHS is not used anymore.

The BIOS controls the drive by sending the required sector number (= LBA) to the drive instead of the P-CHS. DOS however, still works with the CHS address so the BIOS has to calculate an artificial L-CHS and present this geometry to DOS.

This L-CHS is calculated from the total capacity of the drive as indicated below.

Capacity X in bytes	Sectors	Heads	Cylinders
$1 < X \leq 504 \text{ MB}$	63	16	$X / (63 * 16 * 512)$
$504 \text{ MB} < X \leq 1008 \text{ MB}$	63	32	$X / (63 * 32 * 512)$
$1008 \text{ MB} < X \leq 2016 \text{ MB}$	63	64	$X / (63 * 64 * 512)$
$2016 \text{ MB} < X \leq 4032 \text{ MB}$	63	128	$X / (63 * 128 * 512)$
$4032 \text{ MB} < X \leq 8032.5 \text{ MB}$	63	255*	$X / (63 * 255 * 512)$

As you can see, the number of Sectors/track is fixed at 63 and the number of Heads is 16 or a multiple of that. Then the number of Cylinders is calculated by dividing the total capacity in bytes by the number of bytes per cylinder. This translation is called the LBA assisted method.

Note: The last row of the above table indicates that a drive over 4 GB should be translated to 255 i.s.o. 256 Heads. This to avoid problems with DOS which is unable to handle 256 Heads. For more information see [The 4 GB Limit](#) below.

The two translation methods produce similar geometries (L-CHS) in many cases. The difference becomes apparent when a drive reports less than 63 Sectors/track.

The LBA assisted method always assigns 63 Sectors/track.

The LBA assisted method is therefore more flexible than the bit-shifting translation and places no limits on the reported drive geometry (P-CHS).

The bit-shifting method however, has to be used if the drive cannot handle LBA.

Note that most translating BIOSes require you to set the P-CHS values in the BIOS setup.

With any translation method, the BIOS has to ensure that the sectors are accessed in the same order as with "untranslated" access.

This however, is NOT always the case. Therefore, it can be DANGEROUS to your data if you change to another translation mode on an already formatted drive!

## Does my BIOS support translation?

Now that the theory has been dealt with, it is time to answer this question.

First check your BIOS date. If it is 7/94 or later, chances are pretty good that you have a translating BIOS. With older BIOSes, your chances are reducing.

Secondly, you can check in the BIOS SETUP if you have User definable drive types and if there is a translation MODE selection. MODE indications like LARGE or ECHS are normally used for bit-shifting translation. If you can select LBA mode, your BIOS supports the LBA assisted translation method.

But to know for sure, we have to examine the Fixed Disk Parameter Table (FDPT).

The BIOS maintains such a (16 byte) table for each physical drive it supports.

When a BIOS supports translation it uses an enhanced version of this table, the so-called Enhanced Disk Parameter Table (EDPT)

The location of the FDPT or EDPT for the first drive is stored in Interrupt vector 41h and Interrupt vector 46h holds the location of the table for the

The location of the FDPT or EDPT for the first drive is stored in interrupt vector 13h and interrupt vector 13h holds the location of the table for the second drive.

(There are no pointers to the FDPTs for the third and subsequent drives because direct access to these tables is not used anymore. The pointers for the first two drives are however still there to remain compatible with older software. Nowadays, data from a FDPT/EDPT is provided by the BIOS via Interrupt 13h functions 8h and 48h.)

The table below indicates what is stored in the FDPT or EDPT.

Offset	Type	FDPT	EDPT
0-1	Word	Physical Cylinders	Logical Cylinders, limit 1024
2	Byte	Physical Heads	Logical Heads, limit 256
3	Byte	Reserved	A0h Signature, indicating EDPT
4	Byte	Reserved	Physical Sectors/Track
5-6	Word	Precompensation (Obsolete)	Precompensation (Obsolete)
7	Byte	Reserved	Reserved
8	Byte	Drive Control Byte	Drive Control Byte
9-10	Word	Reserved	Physical Cylinders, limit 65536
11	Byte	Reserved	Physical Heads, limit 16
12-13	Word	Landing Zone (Obsolete)	Landing Zone (Obsolete)
14	Byte	Physical Sectors/Track	Logical Sectors/Track, limit 63
15	Byte	Reserved	Checksum

If your BIOS builds an EDPT for a certain drive, it is a translating BIOS!

You can find an EDPT with the WDTBLCHK utility from Western Digital.

This very useful program can be downloaded here. It is a self extracting file called [CHKBIOSEX.E](#).

Or you can get it from <ftp://ftp.wdc.com/drivers/hdutil/chkbios.exe>. (56 KB)

There is a problem however. Most translating BIOSes only build an EDPT i.s.o. a FDPT if the drive is accessed in a translation mode. I therefore use the following trick.

This trick assumes a PC with one < 504 MB harddisk and a BIOS with user definable drive types.

- Enter the BIOS setup and select for the second (non-present) drive a CHS of 2048x16x63.
- If possible select a mode like ECHS, LARGE or LBA for this drive. You probably see the CHS numbers change, indicating the translation.
- Now exit the BIOS setup via "save and exit" and let the computer reboot.
- You probably get an error message because of the missing second drive. Ignore this and continue the boot to the DOS prompt. (If you have

Windows 95/98, hit the F8 key when you see "Starting Windows 95/98 ....." and select "Command prompt only" from the menu.)

- Now run the WDTBLCHK utility and check on screen 3-1 if the BIOS build an Enhanced table for the second drive. Look under "INT 46 DRIVE". On the same screen you can also check both the L-CHS and P-CHS as stored in the EDPT. This to confirm a correct translation.
- If this works, repeat the whole procedure with different CHS values to check the boundary conditions of the translation algorithm(s). Note that the **red** figures may bring you into problems as described in [The 2 GB BIOS Limit](#) below.  
Use 4095x16x63 / **4096**x16x63 (2 GB); **8191**x16x63 / **8192**x16x63 (4 GB) and **16320**x16x63 for the 8 GB limit. (The last figure should translate to 1024x255x63 in LBA mode.)

Restore the BIOS setup to its original settings after these tests.

**Caution:** To be on the safe side, (and to protect the data on your harddisk) it is best to use a boottable floppy with the WD Table Check utility on it. Reboot from this floppy, i.e. your harddisk, as long as the BIOS setup is in the 'test' mode.

There is another way to detect a translating BIOS but you need a > 504 MB (actually > 1024 cylinders) drive for that. Connect this drive as slave on the primary IDE-port and, if possible, have the BIOS setup autodetect this drive's geometry. After "save and exit" and reboot use the WD utility to check for an Enhanced table on the second drive.

Finally, if you have a BIOS which supports the IBM/Microsoft Int 13h extensions, you do not need all these tricks because all those BIOSes support translation.

These BIOSes started to appear in 1995. See [The Int 13h extensions](#) below.

## The 2 GB BIOS Limit

All major harddisk manufacturers have reported about a 2 GB BIOS limit.

It appears that a number of translating BIOSes, manufactured before May 1996 have problems translating drives with cylinder values over 4095. This limits the capacity to 2 GB (2015 MB).

To break this limitation, many harddisk manufacturers supply a software solution like Disk Manager or EZ-Drive with their large harddisks. Alternately, there may be a BIOS upgrade available for your system.

A nice article about this problem and its various solutions is [The other BIOS Limitation!](#) from Western Digital.

Details from one of the BIOS manufacturers can be found in Micro Firmware's article [Notes on Installing Hard Drives Larger Than 2 GB](#).

Reported problem scenarios for the affected BIOSes are:

- The BIOS can only see a maximum of 4095 cylinders thereby truncating the usable drive space to 2015 MB
- The BIOS uses only the lower 12 bits of the 16-bit cylinder word thereby losing 2015 MB of the drive's capacity when a drive is larger than 2 GB!
- When a cylinder count of over 4095 is entered in the BIOS setup, the BIOS will cause a system lock-up at boot time, making the entire system inaccessible.

Apart from the last scenario, I don't know any other reliable method to detect this 2 GB limit than by hooking up a drive with more than 4095 cylinders. However, you may have found something when using [the trick](#) for detecting a translating BIOS. (See above)

BIOSes dated May 1996 or later should be free of these problems and support translation up to the Int 13h interface limit of 8 GB.

Related to the 2 GB limit is the Award BIOS harddisk size display limit bug.

I've found that all Award BIOSes dated July 1994 or later correctly support the LBA assisted translation up to the 8 GB limit. There is however a bug in the BIOSes dated before January 1996 that limits the harddisk size display, on the BIOS Setup and boot screens, to 2015 MB. Whenever a drive is 2016 MB or larger, the display starts to count from zero again. The same happens at 4032 and 6048 MB. This looks a lot like the above mentioned 2 GB limit but is actually only a bug in the harddisk size calculation routine, and it doesn't affect the BIOS support for these large drives. For these Award v4.50(P)(G) BIOSes, just use the HDD AUTO DETECTION feature to Setup the drive, select the option with LBA at the end, and disregard the incorrect HD size display.

## The 4 GB Limit

Yes, there is another limit! This is actually an Operating System issue but the appropriate way to deal with this problem is to account for it in the system BIOS.

It appears that DOS and Windows 95/98 are limited to 255 Heads.

A translated geometry of 256 Heads will therefore create a problem.

This can happen when a drive has 16 Heads and more than 8192 Cylinders (> 4032 MB).

As indicated under [The translating BIOS](#) the LBA assisted method should translate to 255 i.s.o. 256 Heads when a drive is larger than 4032 MB. If this is not the case, then the bit-shifting translation (select ECHS or LARGE) should be used with the following workaround:

1. Enter the CHS values of the drive in CMOS setup or do a "HDD autodetect" and do not select a translation mode yet
2. Adjust the number of heads from 16 to 15
3. Multiply the number of cylinders by 16/15 (round down to a whole number)
4. Adjust the number of cylinders to this higher amount
5. Check or select a bit-shifting translation mode
6. Save and exit CMOS setup and partition and format the drive.

With this workaround, a translated geometry of 15 X 16 = 240 Heads will be used.

This limits the maximum L-CHS to 1024x240x63 which is equivalent to 7560 MB.

For more information read the excellent article [Issues with Hard Drives Over 4 GB](#).

## The Int 13h extensions – Breaking the 8 GB barrier

With the rate in which the harddisk technology is progressing, we are now faced with EIDE drives exceeding the 8064 MB limit of the Int 13h interface. The only way to break this limit is to ditch the CHS system in favor of a direct LBA interface.

The SCSI drive technology uses this kind of LBA interface already for years.

LBA works with a 64-bit sector address, so we can (theoretically) access the staggering amount of 8,796,093,022,208 GB! Due to limitations of the ATA interface, only the lower 28 bits of the LBA address can be used by an EIDE drive resulting in a, still formidable, limit of 128 GB.

Windows 95/98 is ready for this because it already uses LBA internally.

Except in Safe or Compatibility Mode, the protected mode diskdriver takes over the BIOS Int 13h interface and can access a drive directly in LBA.

To remain compatible with all the Operating Systems that still use the CHS system to call the BIOS (including Windows 95/98 at boottime), we need an extension on, rather than a replacement of, the Int 13h BIOS interface.

This is where the IBM / Microsoft Int 13h extensions come in



THIS IS WHERE THE IBM/MICROSOFT INT 13H EXTENSIONS COME IN.

This specification adds new functions to the Int 13h interface. These new functions are fundamentally different from the conventional Int 13h interface and allow the BIOS to be called directly with the required LBA.

A BIOS with the IBM/Microsoft Int 13h extensions supports both LBA capable and LBA non-capable drives. To do this, such a BIOS must support all of the following translations.

— Traditional Int 13h functions:

- direct CHS addressing (Normal mode)
- L-CHS to P-CHS translation
- L-CHS to LBA translation

Extended Int 13h functions:

- direct LBA addressing
- LBA to P-CHS translation

Because BIOSes with the IBM/Microsoft Int 13h extensions also support at least four (sometimes eight) drives, including removable drives, you will find them in most modern PCs.

Most BIOSes dated January 1998 or later will support the Int 13h extensions. To detect such a BIOS, you can download my [EXTBIOS](#) utility. (6 KB ZIP-file)

Just as with the 504 MB barrier, you need one of these solutions to break the 8 GB barrier if you do not have the Int 13h extensions:

1. A BIOS upgrade
2. Add-in card with an Int 13h extension BIOS (this takes over the Int 13h interface only)
3. Software like Disk Manager or EZ-Drive.

%

Now that the BIOS is ready for it, what else do we need to break the 8 GB barrier?

Well, an Operating System capable of addressing a drive in LBA, either directly in protected mode or via the extended Int 13h BIOS interface.

Windows 95 (all versions), Windows 98, Windows NT 4.0 (with SP4), Linux, and OS/2 Warp 3 and 4 (on HPFS) are fully equipped for this.

(So DOS 5 / 6.x, Windows 3.x, and Windows NT 3.5x are limited to 8 GB.)

Last but not least, we need a partition utility capable of creating partitions above the 8 GB boundary which the OS can see. The version of FDISK supplied with these Operating Systems is able to do this with the aid of new partition types 0Eh and 0Fh (FAT16) or 0Bh and 0Ch (FAT32; Win95b-OSR2 / Win98 only). Except for primary partitions below the 8 GB boundary, these new FDISK versions automatically select the new partition types, i.e. the old ones, when a BIOS with Int 13h extensions is detected.

Microsoft Knowledge Base article [Q69912](#) gives a brief description of the various partition types.

A nice article about the various versions of FDISK and their limitation is [Notes on DOS FDISK Command](#).

Third party partition managers like "Partition Magic" from [PowerQuest](#) or "Partition Commander" from [V Communications](#) can also go above the 8 GB boundary, and make the job of creating, re-sizing, and moving partitions, and selecting partition types much easier. And all this without losing data!

Needless to say that the new partition types are invisible to DOS and that compatibility has to be sacrificed in the name of progress.

More information on the net

More information on the net

Possibly the most extensive source of information on this subject is  
The Enhanced IDE/Fast-ATA/ATA-2 FAQ by John Wehman and Peter den Haan.  
[Peter den Haan's EIDE storage page](#) is the place to look for this FAQ.  
— This page contains loads of links to other sites as well.

A good starting point for finding a BIOS upgrade is [Wim's BIOS page](#).  
Start with the FAQ and then work your way through the other pages.



---

Any comments? Please send a message to [J.Steunebrink@net.HCC.nl](mailto:J.Steunebrink@net.HCC.nl)



[Back to the Index page](#)



