

## SUPPORT FOR GREATER THAN 528MB ATA DRIVE

APRIL, 1992

REVISION 1.3

BY

SHISHIR SHAH

WESTERN DIGITAL CORPORATION

PURPOSE:

This document is written as public review comments against ATA Attachment Draft Revision 3.0 ( DSR x3.221, project 791-d). The purpose of this document is to break 528MB boundary limitations created by combination of BIOS and ATA interface.

PROBLEM:

Today, INT 13H interface defines 10 bit cylinder number (zero based), 8 bit head number (zero based, maximum 255) and 5 bit sector number (one based). ATA task file interface defines 16 bit cylinder number (zero based), 4 bit head number (zero based) and 8 bit sector number (one based). Minimum of these cylinder, head and sector numbers define the maximum capacity of 528 MB for AT IDE drive using standard BIOS and drivers from different OS vendors. Following table tabulates this information.

	BIOS	IDE	Limitation
Max Sectors/Track	63	255	63
Number of heads	255	16	16
Number of Cylinders	1024	65536	1024
Maximum Capacity	8.4 GB	136.9 GB	528 MB

During boot up sequence, System INT 13H BIOS ( AT IDE ) reads the drive type from CMOS and gets drive geometry parameters either from BIOS tables or from CMOS (user-defined) or by performing drive Identify command. It resets the drive and sets IDC parameter ( sectors/track, Heads and cylinders ) equal to that is mentioned in the drive table. Same parameters are presented to DOS via INT 13H interface. Drive uses the same parameters to translate Sectors/track, Head and cylinder number to compute the Logical Block Address (LBA). Based on this LBA drive will compute the true physical drive geometry parameters.

To support higher capacity than 528 MB ATA drive, BIOS must perform translation from ATA interface parameters to INT 13H interface drive geometry parameters. This means BIOS uses different geometry parameters for Set drive parameter command than actually reported at INT 13H interface. This poses the problem for the protected mode device drivers as they do not know what parameters they should use to communicate with the drive.

PROPOSED SOLUTION:

ATA Task file has total of 28 bits available for Sectors, Heads and Cylinder Numbers. If we were to change the definition of these registers as shown below we can access of 137 GB of data at this interface.

	Current	Proposed
Bits	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Sector Number	S S S S S S S S	LBA Byte 0
Cylinder Low	C C C C C C C C	LBA Byte 1
Cylinder High	C C C C C C C C	LBA Byte 2
SDH Port	1 0 1 D H H H H	1 L 1 D LBA 27_24

Where:

C = Cylinder Number  
D = Drive Select  
H = Head Number  
L = 1, Task File in LBA Format  
S = Sector Number

The selection of LBA vs Cylinder, Head and Sector in the task file is made by using a BIT 6 of SDH ( port address 1F6H/176H) register. Currently this bit is defined as 0 in ATA specification. This bit should be redefined as Logical Block Address Bit, when set to 1, task file will indicate Logical Block Address.

All op-codes defined in ATA specification Rev 3.0, remain the same. The host will write task file register with LBA, set Logical BIT in SDH port and issue the command. The task file registers Cylinder, Head and sector number, will be valid only when I/O is completed successfully OR error is detected during an I/O operation. When last sector/block is transferred, drive will update the task file with correct value of LBA in Cylinder, Head and Sector Registers.

Drive that supports LBA will maintain its geometry parameters through soft resets. Drive will maintain the last IDC parameters until the next IDC ( Set parameter) command is issued.

CHANGE TO IDENTIFY COMMAND:

Since new capability is added to this interface, BIT 9 of word 49 ( Drive capabilities ) of IDENTIFY Command data is defined as Logical Block support available. This bit is currently defined as 0 by ATA specifications.

Word 49 Drive Capabilities:  
Bit 15-10 = 0, Reserved

K3T9.2 / 4X-065R1

09 = 1, LBA Mode Supported  
 08 = 1, DMA Supported  
 07-00 = Vendor Unique

Word 60 = Maximum user addressable LBA, BITS 15\_0  
 Word 61 = Maximum user addressable LBA, BITS 31\_16

It is very important to note that when drive is accessed in LBA mode it will always access the LBA in linear mode irrespective of IDC geometry data.

#### RECOMMENDED DRIVE TYPE:

It would be nice to have a new drive type that indicates the drive has LBA support. BIOS can use this drive type and issue IDENTIFY command to the drive and auto configure the geometry parameters.

Since there will be a standard drive type defined, Device Drivers that communicate directly to AT IDE interface (Windows 3.1, OS/2, UNIX, Netware, etc) will determine Sectors/Track and Heads from BIOS INT 13H, Read Drive parameter (function =8) command. This geometry is required today to support FDISK partition table (LBA 0) and partitions on the drive.

#### COMPUTE LBA FROM SECTOR, HEAD, AND CYLINDER NUMBER AT INT 13H INTERFACE:

$LBA = [(Cylinder * (\#of Heads)) + Head] * Sectors/Track + Sector - 1$

Initial definition CHS ( 0, 0, 1) = LBA 0.

#### CHANGES REQUIRED AS FOLLOWS:

##### 1. Drive F/W:

When command is issued, Drive F/W will read SDH register and check if bit 6 is set. If bit 6 is set, host has programmed task file register with zero based LBA. Upon command completion, update ATA task file with last LBA processed.

Identify command data has to be modified to indicate that LBA support is present. i.e. BIT 9 of word 49 = 1, for LBA support & Word 60 and Word 61 is defined as Maximum user addressable LBA. Drive when accessed in LBA mode, will always access Linear LBAs irrespective of IDC data.

##### 2. INT 13H BIOS:

Changes to BIOS are as follows:

#### DURING POST:

- Send IDENTIFY Command and check if LBA capability present.
- IF ( LBA supported )  
     LBA\_Support = TRUE  
   ELSE  
     LBA\_Support = FALSE
- Get Drive capacity from word 60 - word 61 of IDENTIFY data and if drive type is "Auto Configure" word 1 through 6 indicates the Cylinder, head and sector number.

#### AT RUN TIME:

- IF ( LBA\_Support )  
   (  
     - Convert INT 13 parameter to LBA.  
     - Program ATA Task file with LBA.  
     - Set the BIT 6 of head/drive register to 1.  
     - Send Normal Op-code.  
     - Upon command completion, Read LBA from task file and convert it to INT 13H cylinder, head, sector number.  
   )

#### 3. OS Drivers ( Windows 3.1, OS/2, UNIX, Novell ):

To support LBA drives, OS drivers must be modified as follows:

#### Initialization time:

- Send Drive IDENTIFY COMMAND.
- IF ( LBA supported )  
     LBA\_Support = TRUE  
     Get Drive SPT and Heads from fn 8 / INT 13H.  
     Get total drive capacity from word 60 and word 61 of Identify data.  
     Compute Number of cylinders.  
   ELSE  
     LBA\_Support = FALSE

#### RUN TIME:

- IF ( LBA\_Support )  
   (  
     - Program ATA task file with LBA.  
     - Set BIT 6 of Drive/Head Register  
     - Write Normal Command Code in Command Register (1F7).  
     - Upon I/O completion, Read LBA from Task file to get last LBA transferred.  
   )

62

2