Annex KK

Implementation Guidelines for 1K/4K Sector Sizes

November 28, 2006

T13/e05122r5

Technical Editors:

Curtis E. Stevens
Western Digital
Phone: 949-672-7933
E-Mail: Curtis.Stevens@WDC.com

Dan Colegrove
Hitachi Global Storage Technologies

Contributors	
Name	E-Mail

Revision History		
Date	Revision	Description
2-Mar-05	0	Initial Draft
25-May-05	1	• Modified sections 4.2 and 4.3 to have the proper values.
1-June-05	2	• Updated the revision history for rev 1 • Added a section to call for reporting alignment.
19-Aug-05	3	• Changed Title • Added details for reporting alignment
28-Nov-06	4	• Added 4K/4K examples
		•

Table of Contents

LIST OF FIGURES

1 Introduction

The disk drive industry has been standardized on a 512 byte sector size for over 25 years. In the continual pursuit for size and performance, larger sector sizes are being considered.

Scope

This annex provides guidelines for implementing a media format that incorporates sector sizes greater than 512 bytes. The target sector sizes are 1k followed by 4k. This paper does not make a case for using larger sector sizes. Instead, this paper assumes that the move to larger sector sizes will happen and addresses both system and industry implications.

The information provided in this paper enables sector sizes that are a binary multiple greater than 512 bytes. ATA/ATAPI-7 also specifies methods to report sector sizes that are not a binary multiple. Common sector sizes that are not binary multiples include 520, 524, 528 and 532 byte sectors. Non-binary multiples are beyond the scope of this paper.

2 Overview

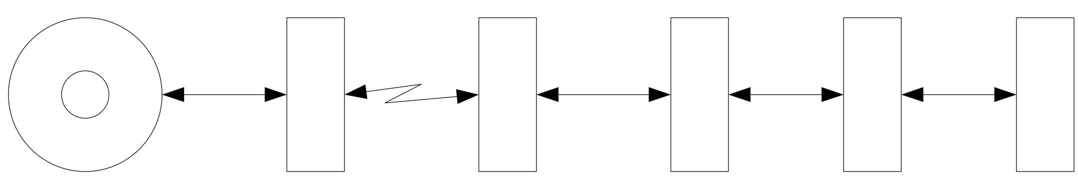
The disk drive industry is considering implementing drives with a media sector size larger than 512 bytes. The purpose of this change is to allow for greater format efficiency, greater error recovery capability, or both. Figure 1 shows major system components that are affected by a change in sector size.

Ed note, add yield option

Make first sentence non-disk drive

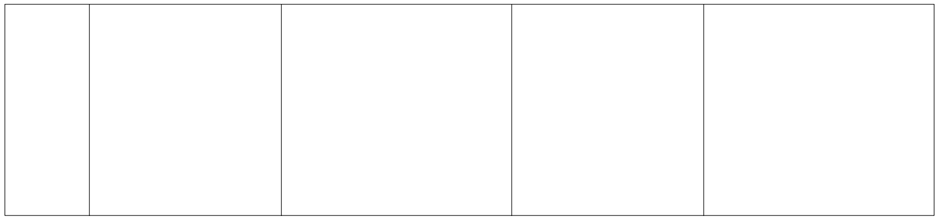
Just sector size >512

Figure 1 – System Food Chain



There are two competing possibilities for expanding the sector size on the media. One proposal would expand the sector size seen at the drive interface; the other would keep the 512 byte sector size at the drive interface. Both possibilities have drawbacks. Figure 2 illustrates the possibilities.

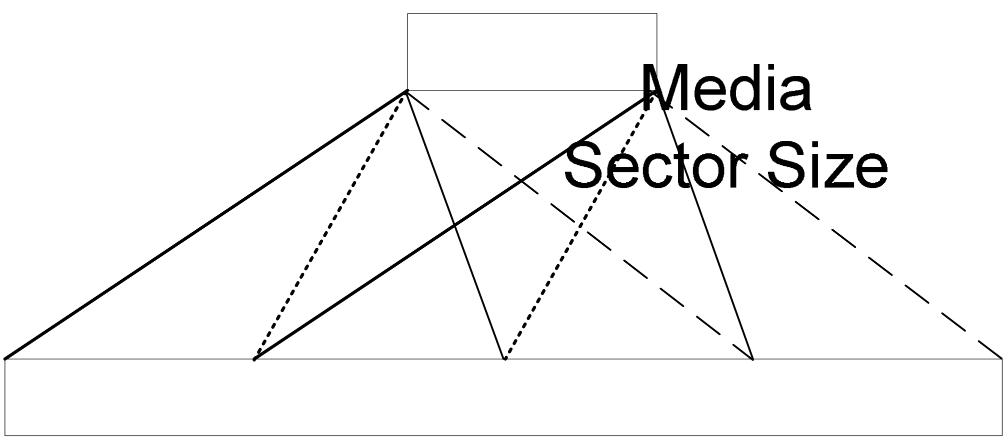
Figure 2 – Mapping Proposals



Using the 512 byte LBA mechanism, the Drive Interface, Host Interface, BIOS, OS, and Applications will still function. If the OS were modified to properly align the disk accesses then performance of the disk drive would be optimal. The 512 byte LBA mechanism also allows a drive manufacturer to ship a utility with the unit that can optimize performance. If the Physical Sector Size LBA mechanism is employed, the existing Drive Interface, Host Interface, BIOS, OS, and Applications may not function. The reason they may not function is that many components in the System Food Chain are hardwired to 512 bytes. These hardwired elements include hardware, firmware and software. If you attach a drive with 1K interface sectors to a system today it will not be able to boot using Windows 2000/XP. If the host interface is able to transfer the data, it is highly likely that the system BIOS is hardwired to 512 bytes. If the BIOS were able to launch Windows, the user would find that Windows was hardwired to 512 bytes and the system would hang. In the case where the BIOS or host interface are hardwired to 512 bytes, no utility can reasonably be used to fix the problem.

This standard provides a mechanism for describing media format and host LBA alignment requirements in the IDENTIFY DEVICE command and as a part of the Long Logical and Long Physical feature sets. Figure 3 illustrates an example of the capability documented in this standard.

Figure 3 – Logical to Physical mapping



In this example, the interface (logical) sector size is 512 bytes, and the media (physical) sector size is 2048 bytes. This mechanism allows an ATA device to both implement a larger physical sector and maintain compatibility with existing systems, interfaces, and software. One of the drawbacks of this method is that drive performance can suffer if the host writes data starting or ending on an LBA that is misaligned with respect to the physical sector boundaries. When misalignment occurs, the drive will be forced to perform a Read-Modify-Write (RMW) operation in order to satisfy the host request.

ATA also allows the Logical Sector size to be changed. This would allow a device to implement a 4KB sector on the media and require that the host transfer 4KB of data for each LBA requested. This type of implementation avoids the RMW issue noted above. The main drawback of this implementation is that existing systems, interfaces, BIOS and system software (OS and otherwise) would have to change in order to accommodate the device.

3 Implementation

3.1 1KB Sector Size Implementation

The 1KB sector size allows for greater format efficiency, and a slight increase in performance. The change to 1KB sectors will cause some issues regarding access alignment. These issues will not be seen in an environment that has been optimized for 4KB accessing.

The device indicates the 1KB sector size to the host by returning 6001h in word 106 of IDENTIFY DEVICE. This indicates that the device has 2 512 byte logical sectors to compose a 1KB physical sector. The host can use this information to know that transfers should start on even LBAs and end on odd LBAs for best performance.

Or

The device indicates the 1KB sector size to the host by returning 6000h in word 106 and 400h in words 117-118 of IDENTIFY DEVICE. This indicates that the device has 1 1024 byte logical sector per 1KB physical sector. The host can use this information to know that transfers require 1K bytes per logical block requested. .

3.2 4KB Sector Size Implementation

The 4KB sector size allows for greater format efficiency than the 1KB sector size; as well as a slight increase in performance. The change to 4KB sectors will cause additional issues regarding access alignment.

The device indicates the 4KB sector size to the host by returning 6003h in word 106 of IDENTIFY DEVICE. This indicates that the device has 8 512 byte logical sectors to compose a 4KB physical sector. The host can use this information to know that transfers should start with an LBA where the low order 3 bits are zero and the transfer ends on an LBA where the low order 3 bits are 1.

Or

The device indicates the 4KB sector size to the host by returning 6000h in word 106 and 1000h in words 117-118 of IDENTIFY DEVICE. This indicates that the device has 1 4096 byte logical sector per 4KB physical sector. The host can use this information to know that transfers require 4K bytes per logical block requested. .

3.3 Reporting Alignment (512 Byte LBA Only)

ATA/ATAPI-7 provides a mechanism for reporting both logical and physical sector sizes, but it does not currently provide a mechanism for reporting the alignment of LBA 0 within the first logical sector. ATA8-ACS has added the ability to report alignment by placing the sector number of the first alignment point in IDENTIFY DEVICE word 209.

If the drive reports a 4K physical sector and a 512 byte logical sector, the following word 209 values report the alignment:

1. Logical LBA0 is aligned to the beginning for the first physical sector – word 209 = 4000h
2. Logical LBA0 is offset from the start of the first physical sector by 512 bytes (1 sector) – word 209 = 4001h
3. Logical LBA0 is offset from the start of the first physical sector by 1024 bytes (1 sector) – word 209 = 4002h
4. Logical LBA0 is offset from the start of the first physical sector by 1536 bytes (1 sector) – word 209 = 4003h
5. Logical LBA0 is offset from the start of the first physical sector by 2048 bytes (1 sector) – word 209 = 4004h
6. Logical LBA0 is offset from the start of the first physical sector by 2560 bytes (1 sector) – word 209 = 4005h
7. Logical LBA0 is offset from the start of the first physical sector by 3072 bytes (1 sector) – word 209 = 4006h

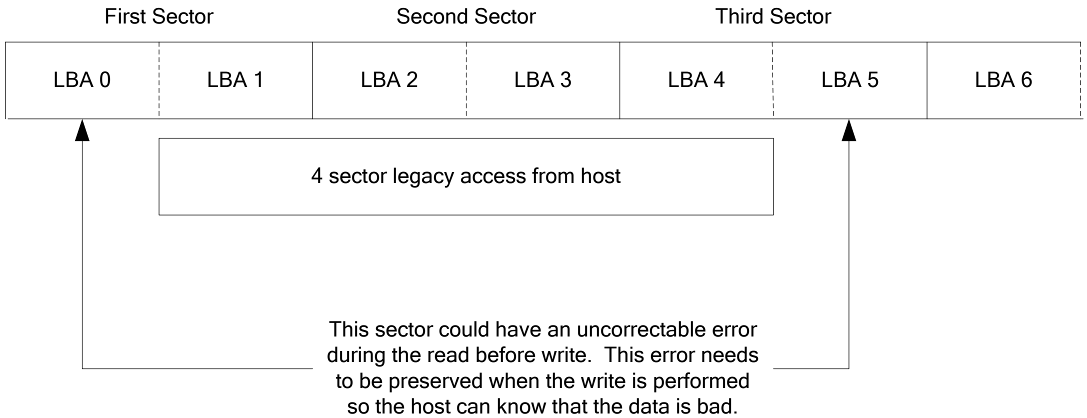
For systems that use Windows XP and earlier, and have drives formatted with a single partition, the optimal value is 4006h.

Windows 3.1, 95, 98, me, NT, 2000, 2003 and XP do not check the logical and physical sector size fields reported in IDENTIFY device. Therefore, it is recommended to optimize alignment to support the target applications required by the host system. It is believed that future operating systems which comply to ATA/ATAPI-7 and above will need the new alignment information in order to gain optimal performance from the drive.

3.4 Read-Modify-Write (RMW) (512 Byte LBA Only)

For devices with a logical sector size of 512 bytes, the drive will be forced to perform RMW when it receives an unaligned transfer. The ATA/ATAPI-7 WRITE commands do not provide a way to return an error other than an ABORT or a DEVICE FAULT. If there is an uncorrectable error encountered during the initial read operation, the WRITE command has no way to report the issue. Further, this error may affect sectors not accessed by the WRITE command. There are several possible solutions for drive vendors to choose from in providing the information to the host. Figure 4 illustrates the issue.

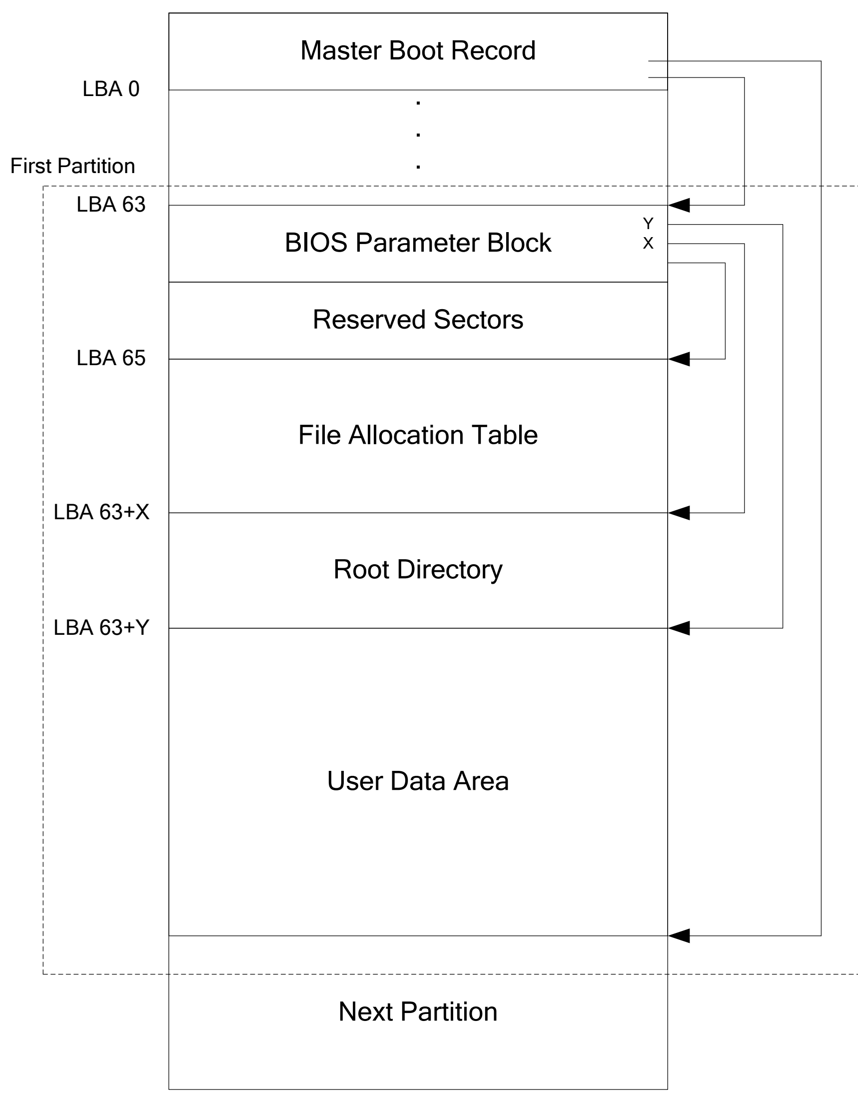
Figure 4 – Uncorrectable Error Handling



4 Implementation Issues (512 Byte LBA Only)

Although the implementation described here allows a drive to function in a legacy system without modification, there are some issues that are critical in allowing the drive to perform at peak efficiency. Figure 5 describes a typical device media layout showing the positions of the Master Boot Record (MBR), BIOS Parameter Block (BPB), and the remainder of a FAT based file system. This layout varies based on the type of FAT file system used, but all the elements described here are generally present. The sector numbers on the left hand side of the figures show typical and/or legacy locations for the various data structures on the media. The following sections describe alignment issues associated with current media layout.

Figure 5 – Typical HDD Layout



4.1 Drive Partitioning

In 1993 when the HDD industry was still dealing in cylinders heads and sectors, an important milestone was reached which caused drive manufacturers to standardize on 63 sectors per track. The norm for disk partitioning software was to place the Master Boot Record (MBR) at Cylinder 0, Head 0, sector 1 (or LBA 0). The MBR contains a pointer to the first partition. The common practice was to place the first partition at Cylinder 0 Head 1, sector 1. This meant that the LBA value of the first sector in the first partition could vary. Once the sectors per track standardized on 63, the LBA value of the first sector in the first partition standardized on LBA 63. Today, there are some applications that check to make sure that partitions start on a track boundary, even though there is no meaning for cylinders heads and sectors.

As we move forward and create larger sectors, partition alignment becomes an important issue. In the case of a 1KB sector device, the partitions should start on an even numbered sector and end on an odd numbered sector. If the drive implements a 4KB sector on the media, then the partition should start on an LBA where the low order 3 bits are zero.

For drives that use 512 byte LBA, all partitions should start on a LBA that is aligned with the start of a physical sector on the media.. This effects some applications that check to make sure the first partition starts on sector 63, but a change is required to implement larger sectors on the media.

4.2 File System Formatting

There are many file systems that cluster sectors together to create an allocation unit larger than a single 512 byte sector. These file systems generally implement a table to associate clusters with files, commonly called a File Allocation Table (FAT). A typical cluster size is 4KB or 8 512 byte sectors. Even if the Partition is properly aligned, there is an issue where the size of the FAT can cause the individual clusters in the user data area to be unaligned relative to the physical sectors on the media. This would also result in performance degradation.

If the clusters in the file system are properly aligned, file accesses will naturally be aligned in many cases and performance will not be degraded.

4.3 Virtual Memory accessing

Once the clusters in the file system are aligned, the OS memory manager needs to be modified to prevent unaligned accesses. When a drive has alignment requirements, disk performance tests may show acceptable performance, but if the virtual memory activity is not aligned, CPU performance tests may provide unacceptable results.

4.4 Booting

The drives with alignment requirements should not show significant performance degradation on unaligned reads. Since booting is mainly a reading process, an impact on system boot times in an unaligned environment is not expected.