Long Logical Sector Size Proposal

To: **T13 Technical committee**
From: **Dan Colegrove**
Hitachi Global Storage Technologies
2903 Carmelo Dr.
Henderson, NV 89052
Phone: 702-614-6119
Fax: 702-614-7955
Email: dcolegro@us.ibm.com
Date: **June 19, 2003**

1 Introduction

The purpose of this proposal is to provide support for logical sectors longer than 256 words. Longer logical sectors are needed to support RAID applications. The main features of this proposal are:

1. Support for logical sectors longer than 256 words
2. The method used to format a drive with long logical sectors is vendor specific. It is not specified in this proposal.

Changes

r0: Derived from proposal e01138r2.
Only sectors equal to or greater than 256 words supported
Added Enable/Disable 256 word mode to Set Features
AV Commands allowed with >256 word sectors

r1: Removed Enable/Disable 256 word mode
Added Long Logical Sector Feature Set Section
Changed title to Long Logical Sector Proposal

R2: Added word TBD3
Made word TBD2 valid only if 106 bit 12 is 1
Added text to compare to large physical sector command set
Added 28 bit commands back in.

R3: Made words TBD2-TBD3 one 32 bit value.
Added more blather about Physical sector
Added Diagram of Physical, Logical and Both
Added Feature Set descriptions
Added Annex C for Reference

2 Definitions

logical sector: A uniquely addressable set of 256 words.
Long logical sector: A uniquely addressable set of >256 words.

3 Specification Changes

3.1 Identify Device

IDENTIFY DEVICE information

Word	O/M	F/V	Description
106	O		Physical Sector Size / Logical Sector Size
		F	15 Shall be cleared to zero
		F	14 Shall be set to one
		F	13 1 = Device has multiple logical sectors per physical sector
		F	12 1 = Device Logical Sector Longer than 256 Words.
		F	11-4 Reserved
		F	3-0 2^X Logical Sectors per Physical Sector
TBD2-TBD3	O	F	Words per Logical Sector

3.1.1 Word TBD1: Physical Sector Size / Logical Sector Size

Bit 12 of word TBD1 shall be set to 1 to indicate that the device has been formatted with a logical sector size larger than 256 words. Bit 12 of word TBD1 shall be cleared to 0 to indicate that word TBD2 is invalid and that the logical sector size is 256 words.

Bits (11:4) of word TBD1 are reserved.

Bits (3:0) of word TBD1 indicate the size of device physical sectors in power of two logical sectors.
See Proposal e01138

3.1.2 Words TBD2-TBD3: Logical Sector Size

Words TBD2-TBD3 indicate the size of device logical sectors in words. The value of words TBD2-TBD3 shall be equal to or greater than 256. The value in words TBD2-TBD3 shall be valid when word 106 bit 12 is set to 1.

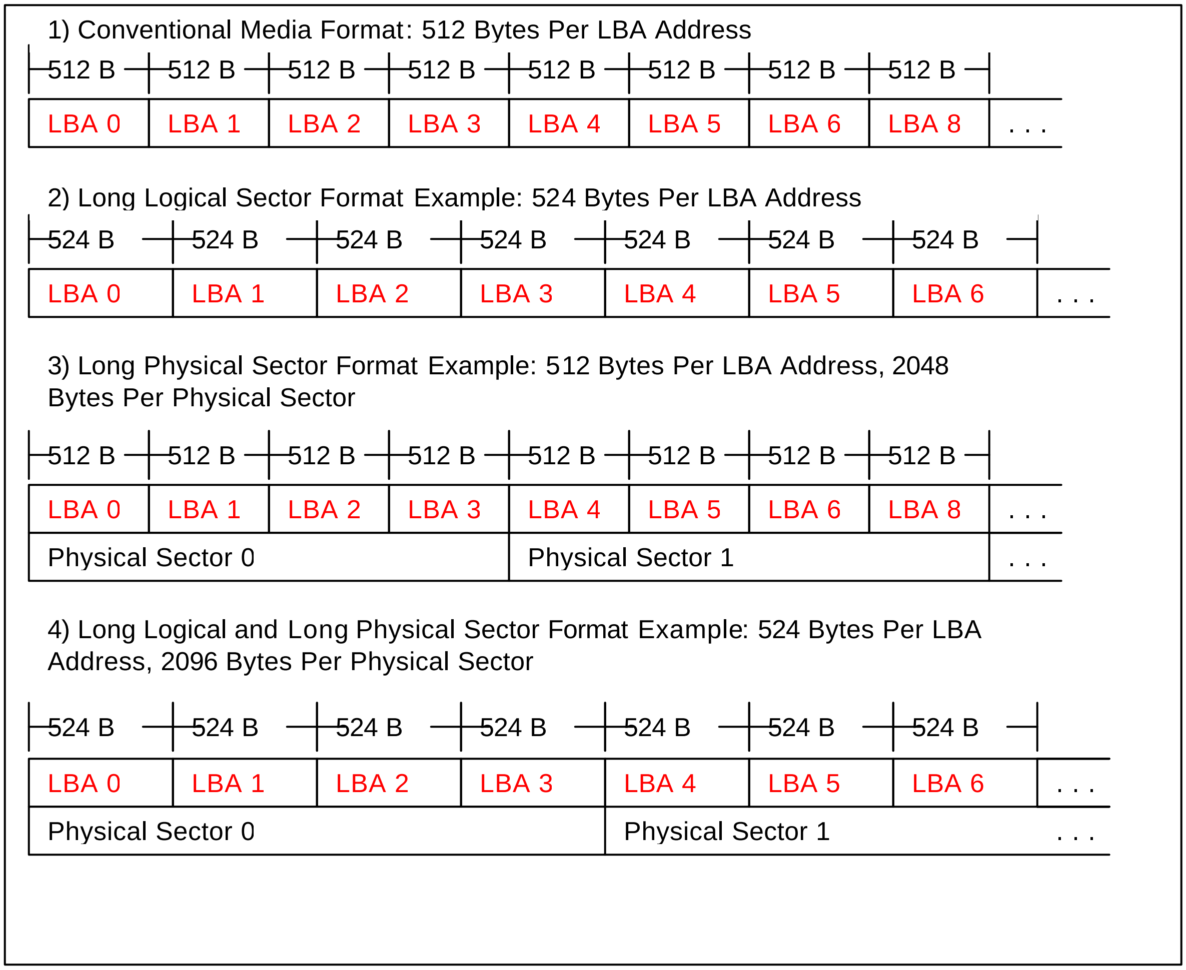

Figure ZZ. Long Logical and Long Physical Sector Examples

3.2 Long Physical Sector Feature Set for Non-Packet Devices

The purpose of the long physical sector feature set is to allow increased media format efficiency. During write operations devi ces calculate and error correction code, ECC, and write the ECC on the media following the data. ECC encoding is more efficient when used over a larger amount of data.
The long physical sector feature set allows a device to be formatted so that there are multiple logical sectors per physical sector on the media. Each physical sector has an ECC field. This allows, for example, a device to have 2048 word physical sectors each containing 8 logical sectors, or one ECC field per 8 256 word logical sectors, see figure ZZ example 3.
A performance penalty may be incurred when writing to devices that implement long physical sector feature set. A physical sector is read or written in a single operation. If a host system does not write all of the logical sectors in a physical sector during a single command the device may need to read the logical sectors that are not to be changed into memory and then write the entire physical sector, see Appendix C.

3.3 Long Logical Sector Feature Set for Non-Packet Devices

The purpose of the long logical sector feature set is to allow additional data words per sector for server applications. Sectors with 520 or 528 bytes are typical. Devices with long logical sectors set Identify Device word 106 bit 13 to 1. The Long Logical Sector length is described by Identify Device data words TBD2-TBD3.

Devices that implement the Long Logical Sector Feature set are not backward compatible with applications that use 256 word logical sectors, e.g. desktop and laptop system.

Table 1 ? Long Logical Sector Function

Command	Function	Words Per Sector Transferred
CFA ERASE SECTORS	Command aborted	-
CFA REQUEST EXTENDED ERROR CODE	Command aborted	-
CFA TRANSLATE SECTOR	Command aborted	-
CFA WRITE MULTIPLE WITHOUT ERASE	Command aborted	-
CFA WRITE SECTORS WITHOUT ERASE	Command aborted	-
CHECK MEDIA CARD TYPE	Command aborted	-
CHECK POWER MODE	Executable	-
CONFIGURE STREAM	Executable	256
DEVICE CONFIGURATION	Executable	256
DEVICE RESET	Executable	-
DOWNLOAD MICROCODE	Executable	256
EXECUTE DEVICE DIAGNOSTIC	Executable	-
FLUSH CACHE	Executable	-
FLUSH CACHE EXT	Executable	-
GET MEDIA STATUS	Executable	-
IDENTIFY DEVICE	Executable	256
IDENTIFY PACKET DEVICE	Command aborted	-
IDLE	Executable	-
IDLE IMMEDIATE	Executable	-
MEDIA EJECT	Executable	-
MEDIA LOCK	Executable	-
MEDIA UNLOCK	Executable	-
NOP	Executable	-
PACKET	Command aborted	-
READ BUFFER	Executable	256
READ DMA	Executable	Identify Word TBD2
READ DMA EXT	Executable	Identify Word TBD2
READ DMA QUEUED	Executable	Identify Word TBD2
READ DMA QUEUED EXT	Executable	Identify Word TBD2
READ LOG EXT	Executable	256
READ MULTIPLE	Executable	Identify Word TBD2
READ MULTIPLE EXT	Executable	Identify Word TBD2
READ NATIVE MAX ADDRESS	Executable	-
READ NATIVE MAX ADDRESS EXT	Executable	-
READ SECTORS	Executable	Identify Word TBD2
READ SECTORS EXT	Executable	Identify Word TBD2
READ STREAM DMA	Executable	Identify Word TBD2
READ STREAM PIO	Executable	Identify Word TBD2-
READ VERIFY SECTORS	Executable	Identify Word TBD2
READ VERIFY SECTORS EXT	Executable	-
SECURITY DISABLE PASSWORD	Executable	256
SECURITY ERASE PREPARE	Executable	-
SECURITY ERASE UNIT	Executable	256
SECURITY FREEZE LOCK	Executable	-
SECURITY SET PASSWORD	Executable	256
SECURITY UNLOCK	Executable	256
SEEK	Command aborted	-
SERVICE	Executable	-
SET FEATURES	Executable	-
SET MAX ADDRESS	Executable	-
SET MAX ADDRESS EXT	Executable	-
SET MULTIPLE MODE	Executable	-
SLEEP	Executable	-
SMART DISABLE OPERATIONS	Executable	-
SMART ENABLE/DISABLE AUTOSAVE	Executable	-

(continued)

Table 0 ? Non-256 Word Logical Block Function *(continued)*

Command	Function	Words Per Sector Transferred
SMART ENABLE OPERATIONS	Executable	-
SMART EXECUTE OFF-LINE IMMEDIATE	Executable	-
SMART READ DATA	Executable	256
SMART READ LOG	Executable	256
SMART RETURN STATUS	Executable	-
SMART WRITE LOG	Executable	256
STANDBY	Executable	-
STANDBY IMMEDIATE	Executable	-
WRITE BUFFER	Executable	256
WRITE DMA	Executable	Identify Word TBD2
WRITE DMA EXT	Executable	Identify Word TBD2
WRITE DMA QUEUED	Executable	Identify Word TBD2
WRITE DMA QUEUED EXT	Executable	Identify Word TBD2
WRITE LOG EXT	Executable	256
WRITE MULTIPLE	Executable	Identify Word TBD2
WRITE MULTIPLE EXT	Executable	Identify Word TBD2
WRITE SECTORS	Executable	Identify Word TBD2
WRITE SECTORS EXT	Executable	Identify Word TBD2
WRITE STREAM DMA	Executable	Identify Word TBD2
WRITE STREAM PIO	Executable	Identify Word TBD2

(concluded)

Table 1 describes the command behavior of drives that have been manufactured with long logical sectors. Data transfer commands transfer either the long logical sector length or 256 words depending on the command. For example, Read and Write Extended commands transfer data in long logical sectors while Read and Write Log Extended commands transfer 256 words per sector, regardless of the logical sector length. Figure ZZ example 2 shows a diagram of a device formatted with long logical sectors.

3.3.1 Devices Implementing the Long Physical Sector Feature Set and the Long Logical Feature Sector Set

The long physical sector feature set and the long logical sector feature set are not exlusive. Figure XX example 4 illustrates a device implementing both the Long Physical Sector and Long Logical Sector feature sets.

Editors Note: Add, "Logical Block Length is 256 words" to the prerequisites for the commands marked command aborted in the table.

Annex C

(informative)

Design and programming considerations for large physical sector devices

C.1 Introduction

Since the inception of the ATA interface the smallest addressable unit of data has been the 512 byte sector. In hard disk drives each sector has an associated error correcting code field to allow detection and correction of read errors. Over time, error correcting code fields have been lengthened to provide greater detection and correction capability. As a result, the proportion of device media devoted to ECC fields has risen. Increasing the length of data sectors on the media increases the efficiency of ECC by enabling better error detection and correction using a smaller proportion of media.

C.2 Physical sectors

Because the 512 byte sector has been a constant since the beginning of ATA many software changes would be required if device logical sectors were made larger. To preserve the legacy software that assumes a 512 byte sector, logical addressing based on 512 byte sectors has been retained. Larger physical sectors are implemented as power of two multiples of 512 byte logical sectors, 1,2,4,8,16, etc. For example, devices may have physical sectors that are 8 logical sectors long or 4096 bytes total.

C.3 Unaligned write

While retaining the 512 byte logical sector maintains software compatibility it introduces a potential performance issue, unaligned write, which must be avoided. A physical sector must be written to the media in a single operation. To complete a write command that writes a fraction of a physical sector the device must read the entire physical sector into buffer memory update the buffer memory with the write data and then write the entire physical sector to the media. This will incur a performance penalty of at least a drive revolution.

Write commands can begin mid physical sector and end mid physical sector resulting in two unaligned writes. In this case the device has to read both the beginning and ending physical sector of the write into the buffer.

To avoid the performance penalty from an unaligned write all write operations must begin with the first sector of a physical sector and end with the last sector of a physical sector.

The first logical sector must be the first 512 bytes of the first physical sector on the device. This allows a host to align write operations with the physical sectors.

Supporting unaligned write operations is optional, but highly recommended to maintain backward compatibility with software. See IDENTIFY DEVICE **Error! Reference source not found.**.

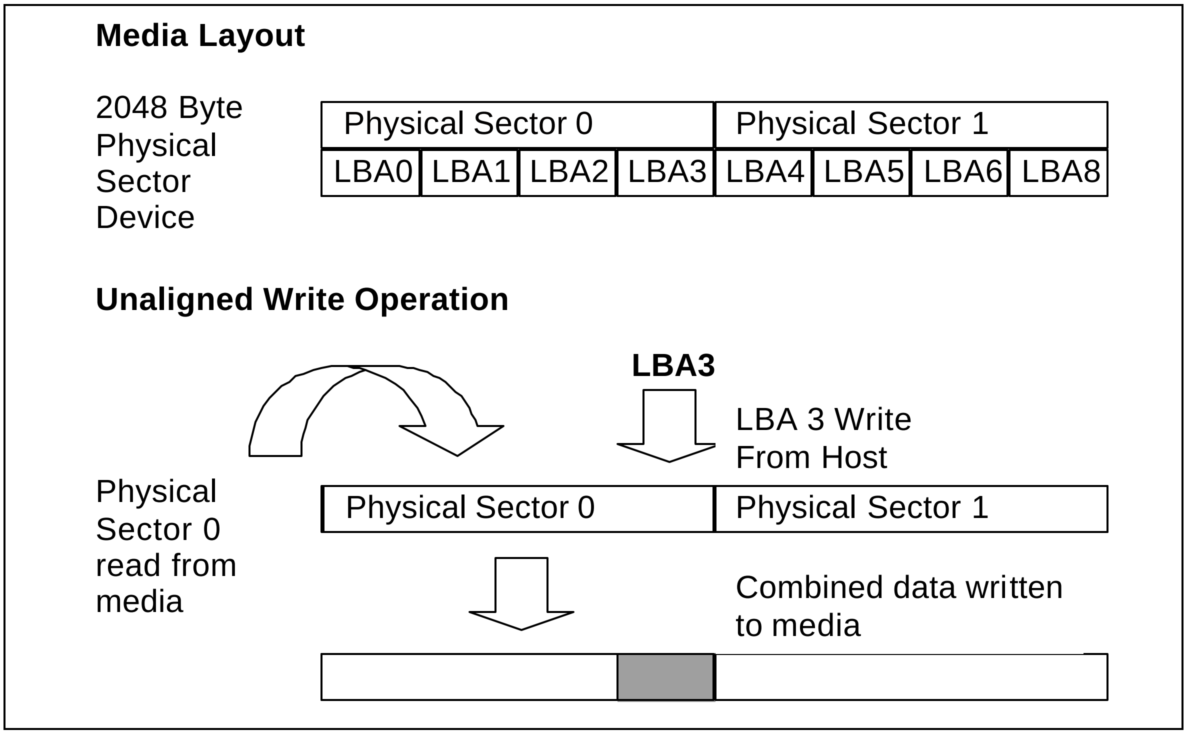

Figure C.1 – Unaligned Write Example

Figure C.1 illustrates an unaligned write on a device with 2048 byte physical sectors. The first four logical sectors, LBA0 – LBA3, reside on physical sector 0. To write only LBA3 the host sends a conventional write command and the data for LBA3. On receipt of the write command the device seeks to the physical sector that contains LBA3, which is physical sector 0. Physical sector 0 is read into the device buffer. Then the new write data for LBA3 is placed in the buffer, overwriting a segment of the buffer. The buffer data is then written to the media, physical sector 0.

C.4 SET MAX

Hosts which use the SET MAX command should set a value to the last logical sector of a physical sector to allow writes to the end of the user area without requiring an unaligned write. Devices should accommodate a SET MAX setting to any LBA address to maintain compatibility.

C.5 Software compatibility

While the current specification allows devices to report up to 2^{15} or 32,768 logical sectors per physical sector there are file system limitations in existing systems that restrict practical device implementations to 4096 bytes per physical sector.