Large Physical Sector Size Proposal

To: **T13 Technical committee**
From: **Dan Colegrove**
IBM
2903 Carmelo Dr.
Henderson, NV 89052
Phone: 702-614-6119
Fax: 702-614-7955
Email: dcolegro@us.ibm.com
Date: ~~**February 12**~~**July 31, 2002**

1 Introduction

~~2.~~The purpose of this proposal is to provide support for physical sector sizes that are a power of two times 512bytes long. Format efficiency ~~and ECC~~ efficiency increases with increasing physical sector size.

This proposal is based on Sector Globs (d99118r0) by Hale Landis

2 Definitions

physical sector: A group of contiguous logical sectors, which must be read from or written to the device media in a single operation.

logical sector: A uniquely addressable set of 256 words (512 bytes).

unaligned write: An unaligned write is a write command that does not start at the first logical sector of a physical sector or does not end at the last logical sector of a physical sector.

3 Specification Changes

3.1 Identify Device

IDENTIFY DEVICE information

Word	O/M	F/V		Description
TBD1	O			Physical 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~~	~~0 = Device does not support unaligned write operations~~
		F	~~11~~12-4	Reserved
		F	3-0	2^X Logical Sectors per Physical Sector

3.1.1 Word TBD1: Physical Sector Size

Bit 13 of word TBD1 shall be set to 1 to indicate that the device has more than one logical sector per physical sector.

Bit 12 of word TBD1 shall be cleared to 0 to indicate that write operations shall begin at the first logical sector of a physical sector and end at the last logical sector of a physical sector.

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

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

Bits (3:0): 0 = 2^0 = 1 logical sector per physical sector
Bits (3:0): 1 = 2^1 = 2 logical sectors per physical sector
Bits (3:0): 2 = 2^2 = 4 logical sectors per physical sector
Bits (3:0:) 3 = 2^3 = 8 logical sectors per physical sector

4 Appendix QQ2 Design and Programming Considerations for Large Physical Sector Devices

4.1 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.

4.2 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.

4.3 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 information (ref).

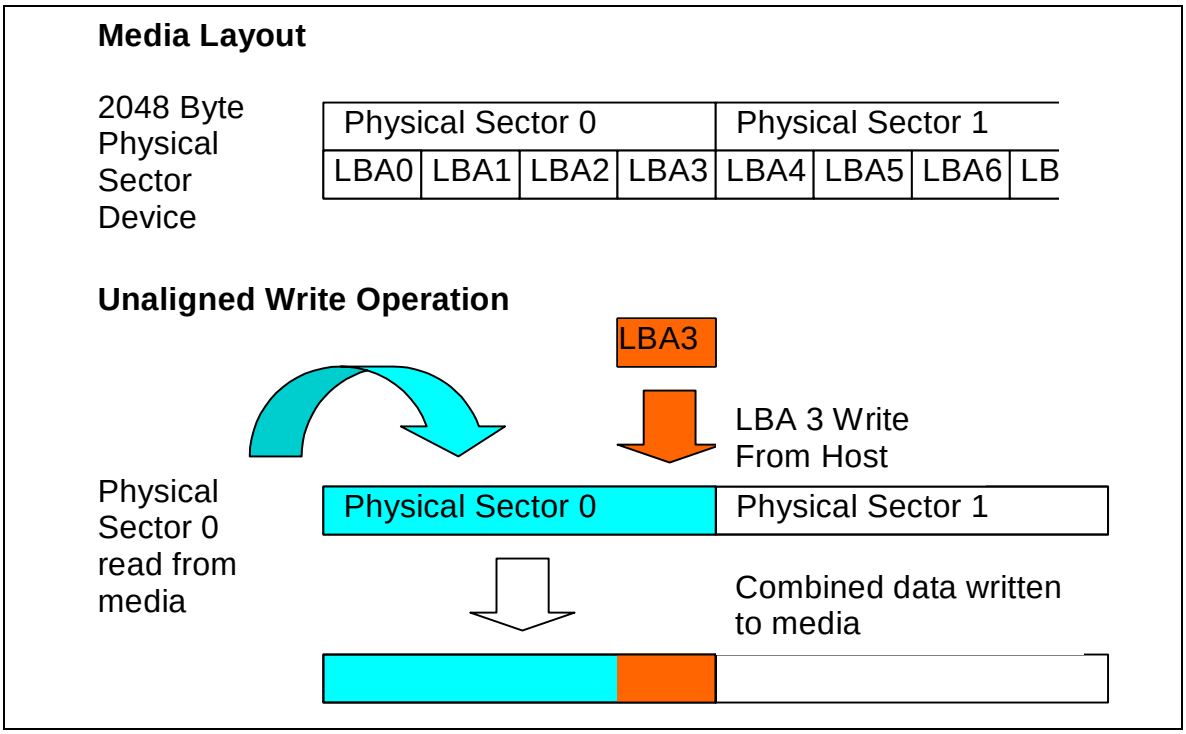

Figure <QQ.1> – Unaligned Write Example

Figure QQ.1 illustrates an unaligned write on a device with 2048 byte physical sectors. The First 4 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.

4.4 Set Max

Hosts which use the Set Max command should Set Max 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.

4.5 Software Compatibility

While the current specification allows devices to report up to 2^{15} or 32768 logical sectors per physical sector there are file system limitations in existing systems that restrict practical device implementations to 4096 bytes per physical sector.