To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 23 February 2004
Subject: 03-348r1 SBC-2 4-byte LBA commands on 8 byte capable drives

Revision history
Revision 0 (11 October 2003) First revision
Revision 1 (23 February 2004) Incorporated comments from November 2003 CAP WG.

Related documents
sbc2r12 - SCSI Block Commands - 2 revision 12

Overview
In a logical unit supporting 8-byte LBAs, 4-byte LBA commands are still allowed to be used to access the lower part of the disk. If:

a) a 4 byte LBA command is used (e.g., READ (10));
b) the length field crosses the 4-byte LBA boundary (LBA > FFFFFFFFh; e.g., the starting LBA was near FFFFFFFFh and the transfer length was large enough to cross that boundary);
c) the (old) fixed format sense data format is being used; and
d) the command fails (e.g. an unrecoverable bad block),

then the LBA cannot be returned in the sense data INFORMATION field. This might confuse the application which issued the command. In many cases, it simply should not have done so; however, some commands use a length of zero to mean “the rest of the logical blocks on the medium” so they can easily overflow.

Table 1 shows affected commands and field names.

Table 1 — Possible affected commands

Command	Possibly affected field	Notes
LOCK UNLOCK CACHE (10)(16)	NUMBER OF BLOCKS	0 means “all remaining”
PRE-FETCH (10)(16)	PREFETCH LENGTH	0 means “all remaining”
READ (6)(10)(12)(16)	TRANSFER LENGTH	
READ LONG (10)(16)	(BYTE TRANSFER LENGTH)	Command limited to one block, so byte transfer length is not a concern.
SYNCHRONIZE CACHE (10)(16)	NUMBER OF BLOCKS	
VERIFY (10)(12)(16)	VERIFICATION LENGTH	
WRITE (6)(10)(12)(16)	TRANSFER LENGTH	
WRITE AND VERIFY (10)(12)(16)	TRANSFER LENGTH	
WRITE LONG (10)(16)	(BYTE TRANSFER LENGTH)	Command limited to one block, so byte transfer length is not a concern.
WRITE SAME (10)(16)	NUMBER OF BLOCKS	
XDREAD (10)(32)	TRANSFER LENGTH	
XDWRITE (10)(32)	TRANSFER LENGTH	
XDWRITEREAD (10)(32)	TRANSFER LENGTH	
XPWRITE (10)(32)	TRANSFER LENGTH	

The CAP WG in November 2003 felt that 4-byte LBA commands should be allowed to stray into that region, and the device server not be required or even allowed to check that boundary. It is incumbent on the application client not to issue such commands unless descriptor sense format is enabled if it wants to see errors reported. The WG requested a statement be added that application clients should enable descriptor format if they plan to access regions beyond the 4-byte LBA boundary.

Also, sentences are added in each command highlighting the existing capacity check that was buried in the model section.

Also, READ LONG and WRITE LONG are removed from the list of commands affected by the Block Limits VPD page. This was an error in 03-028, the proposal that defined that page. Those commands only transfer data for one logical block plus its additional data (e.g., ECC). The Block Limits VPD page fields are intended to restrict the number of blocks that are transferred by commands. They have no meaning when the number of blocks = 1 as it does for READ LONG and WRITE LONG.

Suggested changes

4.4 Logical blocks

Logical blocks are stored on the medium along with additional information that the medium controller uses to manage the storage and retrieval. The format of the additional information is defined by other standards or is vendor-specific and is hidden from the application client during normal read or write operations. This additional information may be used to identify the physical location of the blocks of data and the address of the logical block, and to provide protection against the loss of user data (e.g., ECC bytes).

The address of the first logical block is zero. The address of the last logical block is [n-1], where [n] is the number of logical blocks available to the application client on the medium. A READ CAPACITY command may be issued to determine the value of [n-1].

Logical block addresses are no larger than 8 bytes. Some commands support only 4 byte LOGICAL BLOCK ADDRESS fields (e.g., READ CAPACITY (10), READ (10), and WRITE (10)). The READ CAPACITY (10) command returns a capacity of FFFFFFFFh if the capacity exceeds that accessible with 4-byte logical block addresses, indicating that:

a) the application client should enable descriptor format sense data (see SPC-3) in the Control mode page (see SPC-3) and in any REQUEST SENSE commands (see SPC-3) it sends; and
b) the application client should use commands with 8-byte LOGICAL BLOCK ADDRESS fields (e.g., READ CAPACITY (16), READ (16), and WRITE (16)).

NOTE 1 If a command with a 4-byte LOGICAL BLOCK ADDRESS field accesses logical blocks beyond logical block address FFFFFFFFh and fixed format sense data is used, there is no field in the sense data large enought in which to report the logical block address of the error (see 4.12).

If a command is issued that requests access to a logical block not within the capacity of the medium, the command is terminated with CHECK CONDITION status ~~and~~with the sense key ~~is~~ set to ILLEGAL REQUEST ~~with~~ and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The command may be terminated before processing or after the device server has transferred some or all of the data.

The number of bytes of user data contained in a logical block is the block length. Each logical block has a block length associated with it. The READ CAPACITY data (see 5.12) describes the block lengths that are used on the medium. The block descriptor header used by the MODE SELECT command and the FORMAT UNIT command (see 5.3) are required to change the block length of block devices that support variable block lengths. The block length does not include the length of protection information and additional information, if any.

The location of a logical block on the medium is not required to have a relationship to the location of any other logical block. However, in a typical block device, the time to access a logical block at address [x+1] after accessing logical block [x] is often less than the time to access some other logical block. The time to access the logical block at address [x] and then the logical block at address [x+1] need not be less than time to access [x] and then [x+100]. The READ CAPACITY command issued with a PMI bit set to one may be useful in determining where longer access times occur.

4.12 Error reporting

If any of the conditions in table 4 occur during the processing of a command, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to the appropriate sense key with the appropriate additional sense code for the condition. Some errors may occur after the completion status has already been reported. For such errors, SPC-3 defines a deferred error reporting mechanism. Table 4 lists

some error conditions and the applicable sense keys. The list does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

...

When an invalid LBA is encountered, the first invalid LBA shall be returned in the INFORMATION field of the sense data. When a recovered read error is reported, the INFORMATION field of the sense data shall contain the LBA of the last recovered error during the transfer. When an unrecovered read error is reported, the INFORMATION field of the sense data shall contain the LBA of the unrecovered logical block

The sense data INCORRECT LENGTH INDICATION (ILI) bit indicates that the requested data length in a READ LONG or WRITE LONG command did not match the length of the data on the medium.

Direct-access devices compliant with this standard shall support both the ~~short and long~~ fixed and descriptor sense data formats (see SPC-3). If the ~~short~~ fixed sense data format is requested but the sense data contains an INFORMATION field value or COMMAND-SPECIFIC INFORMATION field value too large for the ~~short~~ fixed sense data format (e.g., an 8-byte LBA), the VALID bit shall be set to zero.

Table 5 summarizes use of the sense data fields.

...

5.3 LOCK UNLOCK CACHE (10) command

...

The NUMBER OF BLOCKS field specifies the ~~total~~ number of contiguous logical blocks within the range. A NUMBER OF BLOCKS field ~~of~~ set to zero ~~indicates~~ specifies that the range contains all remaining logical blocks on the medium~~on the block device shall be within the range~~. Any other value specifies the number of logical blocks within the range. If the logical block address plus the number of blocks exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

5.4 LOCK UNLOCK CACHE (16) command

...

See the LOCK UNLOCK CACHE (10) command (see 5.4) for a description of the fields in this command.

5.6 PRE-FETCH (10) command

...

The PREFETCH LENGTH field specifies the number of contiguous logical blocks of data that shall be transferred to the block device's cache memory from the medium. A PREFETCH LENGTH ~~of~~ field set to zero ~~indicates~~specifies that ~~the contiguous~~all remaining logical blocks ~~up to and including the last logical block of the block device~~ shall be transferred to the block device's cache memory. Any other value ~~indicates~~specifies the number of logical blocks that shall be transferred. If the logical block address plus the prefetch length exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The device server ~~may elect to not~~ is not required to transfer logical blocks that already are contained in the cache memory.

Editor's Note 1: Normally I prefer referring to multibit fields contain zero by clearly marking the width (e.g. 0000h). Since the PREFETCH LENGTH field is referenced by PRE-FETCH (16), where it is 4 bytes rather than 2, the generic "zero" is used. This avoids restating the paragraph there with wider 00<...>00h values. The same approach has been taken for the other commands.

5.7 PRE-FETCH (16) command

...

See the PRE-FETCH (10) command (see 5.6) for a description of the fields in this command.

5.8 READ (6) command

...

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data to be transferred. A TRANSFER LENGTH ~~of~~ field set to zero ~~indirectly indicates~~ specifies that 256 logical blocks shall be transferred. Any other value ~~directly~~ ~~indicates~~ specifies the number of logical blocks that shall be transferred. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

...

NOTE 2 For the READ (10) command, READ (12) command, and READ (16) command, a ~~transfer length of zero~~ TRANSFER LENGTH field set to zero specifies that no logical blocks are transferred.

5.9 READ (10) command

...

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be transferred. A TRANSFER LENGTH ~~of~~ field set to zero ~~indicates~~ specifies that no logical blocks shall be transferred~~. This~~; this condition shall not be considered an error. Any other value ~~indicates~~ specifies the number of logical blocks that shall be transferred. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

5.10 READ (12) command

...

See the READ (10) command (see 5.9) for a description of the fields in this command.

5.11 READ (16) command

...

See the READ (10) command (see 5.9) for a description of the fields in this command.

5.16 READ LONG (10) command

...

The BYTE TRANSFER LENGTH field specifies the number of bytes of data that ~~should~~ shall be transferred. If a non-zero BYTE TRANSFER LENGTH does not match the available data length, the device server shall terminate the command with CHECK CONDITION status ~~and~~ with the sense key ~~shall be~~ set to ILLEGAL REQUEST ~~with~~ and the additional sense code set to INVALID FIELD IN CDB. In the sense data (see SPC-3), the ~~The~~ VALID and ILI bits (~~see SPC-3)~~ shall be set to one and the INFORMATION field shall be set to the difference (residue) of the requested length minus the actual length in bytes. Negative values shall be indicated by two's complement notation.

A BYTE TRANSFER LENGTH ~~of~~ field set to zero ~~indicates~~ specifies that no bytes shall be transferred ~~and~~. This condition shall not be considered an error.

If the logical block address exceeds the capacity of the medium the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

~~The BYTE TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).~~

Editor's Note 2: The byte transfer length is in units of bytes and is always going to be slightly

bigger than one block. The Block Limits VPD page field is in units of blocks. It was a mistake to include READ LONG and WRITE LONG in the list affected by the Block Limits VPD page.

5.17 READ LONG (16) command

...

See the READ LONG (10) command (see 5.16) for a description of the fields in this command.

5.20 SYNCHRONIZE CACHE (10) command

...

The NUMBER OF BLOCKS field specifies the ~~total~~ number of contiguous logical blocks within the range. A ~~number of blocks of zero indicates~~ NUMBER OF BLOCKS field set to zero specifies that the range contains all remaining logical blocks on the medium~~on the block device shall be within the range~~. If the logical block address plus the number of blocks exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

A logical block within the specified range that is not in cache memory is not considered an error.

5.21 SYNCHRONIZE CACHE (16) command

...

See the SYNCHRONIZE CACHE (10) command (see 5.20) for a description of the fields in this command.

5.22 VERIFY (10) command

...

The VERIFICATION LENGTH field specifies the number of contiguous logical blocks of data ~~or blanks~~ that shall be verified. A VERIFICATION LENGTH ~~of~~field set to zero ~~indicates~~ specifies that no logical blocks shall be verified~~. This~~; this condition shall not be considered as an error. Any other value ~~indicates~~ specifies the number of logical blocks that shall be verified. If the logical block address plus the verification length exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The VERIFICATION LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

5.23 VERIFY (12) command

...

See the VERIFY (10) command (see 5.22) for a description of the fields in this command.

5.24 VERIFY (16) command

...

See the VERIFY (10) command (see 5.22) for a description of the fields in this command.

5.25 WRITE (6) command

...

The LOGICAL BLOCK ADDRESS field specifies the logical block where the write operation shall begin.

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be transferred.

A TRANSFER LENGTH ~~of~~field set to zero ~~indirectly indicates~~ specifies that 256 logical blocks shall be transferred. Any other value ~~indicates~~ specifies the number of logical blocks that shall be transferred. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

NOTE 3 For the WRITE (10) command, WRITE (12) command, and WRITE (16) command, a TRANSFER LENGTH ~~of~~field set to zero indicates that no logical blocks are transferred.

5.2.31 WRITE (10) command

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be transferred.

A TRANSFER LENGTH ~~of~~field set to zero specifies that no logical blocks shall be transferred~~. This~~; this condition shall not be considered an error and no data shall be written. Any other value specifies the number of logical blocks that shall be transferred. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

NOTE 4 For the WRITE (6) command, a TRANSFER LENGTH ~~of~~field set to zero specifies that 256 logical blocks are transferred.

5.27 WRITE (12) command

...

See the WRITE (10) command (see 5.26) for a description of the fields in this command.

5.28 WRITE (16) command

...

See the WRITE (10) command (see 5.26) for a description of the fields in this command.

5.29 WRITE AND VERIFY (10) command

...

See the LOCK UNLOCK CACHE (10) command (see 5.4) for a definition of the LOGICAL BLOCK ADDRESS field.

See the WRITE (10) command (see 5.26) for a definition of the TRANSFER LENGTH field and the WRPROTECT field. See the READ (10) command (see 5.9) for a description of the DPO bit. See the WRITE (10) command (see 5.26) for a description of the EBP bit.

5.30 WRITE AND VERIFY (12) command

...

See the WRITE AND VERIFY (10) command (see 5.29) for a description of the fields in this command.

5.31 WRITE AND VERIFY (16) command

...

See the WRITE AND VERIFY (10) command (see 5.29) for a description of the fields in this command.

5.32 WRITE LONG (10) command

...

The BYTE TRANSFER LENGTH field ~~should~~ specif~~y~~ies the number of bytes of data that the device server ~~would~~ returns for the READ LONG command. If a non-zero BYTE TRANSFER LENGTH does not exactly match the data length the device server ~~would~~ returns for the READ LONG command, then the device server shall terminate the command with CHECK CONDITION status ~~and~~with the sense key ~~shall be~~ set to ILLEGAL REQUEST ~~with~~and the additional sense code set to INVALID FIELD IN CDB. In the sense data (see SPC-3), the ~~The~~ ILI and VALID bits shall be set to one and the INFORMATION field shall be set to the difference (residue) of the requested length minus the actual length in bytes. Negative values shall be indicated by two's complement notation.

A BYTE TRANSFER LENGTH ~~of~~ field set to zero indicates that no bytes shall be transferred~~ and~~. This condition shall not be considered an error. ~~The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).~~

If the logical block address exceeds the capacity of the medium the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

5.33 WRITE LONG (16) command

...

See the WRITE LONG (10) command (see 5.32) for a description of the fields in this command.

5.34 WRITE SAME (10) command

...

The NUMBER OF BLOCKS field specifies the number of contiguous logical blocks to be written. A NUMBER OF BLOCKS field set to ~~0000h~~ zero specifies that the device server write all the remaining logical blocks on the medium. If the logical block address plus the number of blocks exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

5.35 WRITE SAME (16) command

...

See the WRITE SAME (10) command (see 5.34) for a description of the fields in this command.

5.2.40 XDREAD (10) command

...

The XOR data transferred is identified by the LOGICAL BLOCK ADDRESS field and the TRANSFER LENGTH field.

The LOGICAL BLOCK ADDRESS field and TRANSFER LENGTH field shall be the same as, or a subset of, those specified in a prior XDWRITE command. If a match is not found the command is terminated with a CHECK CONDITION status ~~and~~ with the sense key ~~shall be~~ set to ILLEGAL REQUEST ~~with~~and the additional sense code set to INVALID FIELD IN CDB. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

5.37 XDREAD (32) command

...

See the XDREAD (10) command (see 5.36) and SPC-3 for a description of the fields in this command.

5.38 XDWRITE (10) command

...

The LOGICAL BLOCK ADDRESS field specifies the starting logical block address of the data on which an XOR operation shall be performed with the data from the medium.

The TRANSFER LENGTH field specifies the number of logical blocks that shall be transferred from the application client and the number of logical blocks on which an XOR operation shall be performed with the data from the medium. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

5.39 XDWRITE (32) command

...

See the XDWRITE (10) command (see 5.38) and SPC-3 for a description of the fields in this command.

5.40 XDWRITEREAD (10) command

...

See the XDWRITE (10) command (see 5.38) and XDREAD (10) command (see 5.36) for a description of the fields in this command.

5.41 XDWRITEREAD (32) command

...

See the XDWRITEREAD (10) command (see 5.40) and SPC-3 for a description of the fields in this command.

5.42 XPWRITE (10) command

...

The LOGICAL BLOCK ADDRESS field specifies the starting LBA where the target shall read data from its medium. It also specifies the starting LBA where the XOR result data shall be written to the medium.

The TRANSFER LENGTH field specifies the number of logical blocks that shall be read from the medium. It also specifies the number of logical blocks that shall be written to the medium. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

5.43 XPWRITE (32) command

...

See the XPWRITE (10) command (see 5.42) and SPC-3 for a description of the fields in this command.

6.4.2 Block Limits VPD page

...

The OPTIMAL TRANSFER LENGTH GRANULARITY field specifies the optimal transfer length granularity in blocks for a single PRE-FETCH, READ, ~~READ LONG,~~ VERIFY, WRITE, WRITE AND VERIFY, ~~WRITE LONG,~~ XDREAD, XDWRITE, XDWRITEREAD, or XPWRITE command. Transfers with transfer lengths not equal to a multiple of this value may incur significant delays in processing.

The MAXIMUM TRANSFER LENGTH field specifies the maximum transfer length in blocks that the target accepts for a single PRE-FETCH, READ, ~~READ LONG,~~ VERIFY, WRITE, WRITE AND VERIFY, ~~WRITE LONG,~~ XDREAD, XDWRITE, XDWRITEREAD, or XPWRITE command. Requests for transfer lengths exceeding this limit result in CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB. A value of zero means there is no reported limit on the transfer length.

The OPTIMAL TRANSFER LENGTH field specifies the optimal transfer length in blocks for a single PRE-FETCH, READ, ~~READ LONG,~~ VERIFY, WRITE, WRITE AND VERIFY, ~~WRITE LONG,~~ XDREAD, XDWRITE, XDWRITEREAD, or XPWRITE command.