

To: T10 Technical Committee
From: Bob Sheffield(robert.l.sheffield@intel.com)
Date: 12 July 2006
Subject: 06-216r2: SAT - Block Mapping Issues

Revision history

Revision 0 (1 May 2006) First revision

Revision 1 (22 June 2006) Added all changes for clause-9 with direct/indirect block mapping changes and all resolved LB comments in clause-9 integrated.

Revision 2 (12 July 2006) Incorporated feedback from 11 July 2006 SAT WG.

Related documents

SAT-r08 - SCSI / ATA Translation revision 08

06-121r1 SAT-r08_LB_Comment_Resolution.pdf

Overview

Letter ballot comments received for SAT (documented in 06-121r0) reflect diverging opinions about the basic mapping of ATA logical sectors to SCSI logical blocks. Two related aspects of this are:

- a) the mapping of logical block addresses between ATA and SCSI command sets; and
- b) the mapping of the number of bytes within each block between ATA and SCSI command sets.

Item b is the more crucial element, with the result there driving the resolution of item a.

One school of thought is that because most ATA disks are formatted fixed 512-byte logical sector sizes, a SATL should implement a one-for-one mapping and support only a fixed 512-byte logical block size without protection information.

The other school of thought says that many factors drive the need to provide more flexibility in the mapping of ATA sectors to SCSI logical blocks:

- a) recent modifications to ATA provide for larger physical sector sizes (e.g., 4096);
- b) various methods exist for providing data protection capabilities in a SATL, and many of those methods involve something other than a one-for-one mapping.

This proposal is intended to define a common direction regarding resolution of the SAT LB comments affecting the mapping between ATA logical sectors and SCSI logical blocks, as well as identifying which changes are required in which subclauses in the SAT standard to resolve those LB comments. The LB comment resolution for each LB comment listed in 06-121 related to block mapping will simply refer to this proposal as the resolution.

This proposal defines two models:

- a) A direct mapping model to translate between SCSI logical blocks and ATA logical sectors that provides a mapping of ATA logical sectors to SCSI logical blocks where SCSI logical blocks and ATA logical sectors contain the same number of bytes, and the ATA LBA for an ATA logical sector is the same value as the logical block address for the corresponding SCSI logical block. The reported capacity of the SCSI logical unit is less than or equal to the reported capacity of the ATA device.
- b) An indirect mapping model that allows ATA logical sector size to be different than SCSI logical block lengths, different addressing of SCSI logical blocks and ATA logical sectors, and skewing block boundaries between ATA logical sectors and SCSI logical blocks.

A SATL implements one or the other of these two models.

This proposal resolves all LB comments in clause 9.

Suggested changes

Add ATA/ATAPI-7 back into normative references as follows:

~~ISO/IEC 14776-971, AT Attachment – 7 with Packet Interface (ATA/ATAPI-7) [ANSI INCITS 397-2005]~~
[ISO/IEC 14776-971, AT Attachment with Packet Interface - 7 \(ATA/ATAPI-7\) \[ANSI INCITS 397-2005\]](#)

Add a definition for ATA Logical Sector Size

3.1.x ATA device capacity: The capacity of an ATA device, in bytes, is the ATA logical sector size (see 3.1.x) times one more than the ATA maximum LBA (see 3.1.x).

3.1.x ATA logical sector size: The size of an ATA logical sector in bytes, calculated as two times the value returned in ATA IDENTIFY DEVICE data words 118:117 (see ATA8-ACS) if the ATA device returns a value of one in ATA IDENTIFY DEVICE data word 106 bit 12 (i.e., the ATA device supports the Long Logical Sector feature set). The size of an ATA Logical Sector is 512 bytes if the ATA device returns a value of zero in ATA IDENTIFY DEVICE data word 106 bit 12 (i.e., the ATA device does not support the Long Logical Sector feature set).

NOTE <x> - The Logical Sector Size indicated by an ATA device is represented in words; therefore, the number of bytes in an ATA device logical sector is two times the value indicated in the Logical Sector Size.

3.1.x ATA maximum LBA: The maximum user LBA for 48-bit address feature set returned in ATA IDENTIFY DEVICE data words (103:100) minus one if the ATA device returns a value of one in ATA IDENTIFY DEVICE data word 86 bit 10 (i.e., the ATA device supports the 48-bit address feature set), or the total number of user addressable sectors returned in ATA IDENTIFY DEVICE data words (61:60) minus one if the ATA device returns a value of zero in ATA IDENTIFY DEVICE data word 86 bit 10 (i.e., the ATA device does not support the 48-bit address feature set) (see ATA8-ACS).

3.1.x direct block mapping: A SATL implementation that maps logical blocks on a logical unit one-for-one with ATA logical sectors on an ATA device, where the LBA of a logical block has the same value as the LBA of the corresponding ATA logical sector and the number of bytes in a logical block equals the number of bytes in an ATA logical sector (see 9.1.2).

3.1.x indirect block mapping: A SATL implementation that does not follow the constraints of direct block mapping (see 3.1.x and 9.1.3).

3.1.x logical unit capacity: The capacity of a logical unit in bytes calculated as length in bytes of each logical block times one more than the LBA of the last logical block on the logical unit.

Modify subclause 8.3.3.1 Mode parameter block descriptor fields as follows:

The SATL may support the direct-access [short LBA](#) mode parameter block descriptor ~~or The SATL may support the long LBA mode parameter block descriptor. The SATL shall not support the general mode parameter block descriptor. The DENSITY CODE (see SPC-3) is reserved for SATL devices.~~ Table 14 shows the fields in the [short LBA](#) mode parameter block ~~descriptors~~ [descriptor](#) and the long LBA mode parameter [block descriptor](#) supported by the SATL.

Table 14 — Mode parameter block descriptor fields

Field	Description or reference
NUMBER OF BLOCKS ^b	<p>The SATL shall not support changing the capacity of the logical unit by changing the NUMBER OF BLOCKS field using the MODE SELECT command, and the value in this field is ignored.^a</p> <p>Unspecified (see 3.4.3).</p>
BLOCK LENGTH ^b	<p>The SATL shall return the same block length for the entire LUN. For direct access devices, the SATL shall set this field to 200h.</p> <p>When processing a MODE SELECT command, if the SATL implements direct block mapping (see 3.x.y) and the value of the BLOCK LENGTH field is not the same as the ATA logical sector size (see 3.x.y), then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST. If the SATL supports indirect block mapping this field is unspecified (see 3.4.3).</p> <p>When processing the MODE SENSE command, if the SATL implements direct block mapping (see 3.1.x) then the SATL shall return the same block length for the entire logical unit and the BLOCK LENGTH field shall contain the ATA logical sector size (see 3.1.x). Otherwise the BLOCK LENGTH field is unspecified (see 3.4.3).</p>
<p>^a This is because the ATA device provides no method to modify the capacity</p> <p>^b The values reported in the NUMBER OF BLOCKS field AND THE BLOCK LENGTH field shall be such that the logical unit capacity (see 3.1.x) is less than or equal to the ATA device capacity (see 3.1.x).</p>	

Modify subclause 8.5.5 General mode parameter block descriptor fields as follows:

8.5.5 General-mode Mode parameter block descriptor fields

Table 19 describes the translation of [fields in the general short LBA mode parameter block descriptor](#) [and the long LBA mode parameter block descriptor supported by the SATL fields.](#)

Table 19 — General-mode Mode parameter block descriptor fields

Field	Description or reference
NUMBER OF BLOCKS ^a	Unspecified (see 3.4.3)
DENSITY CODE	For direct access devices, this field should be set to 00h.
BLOCK LENGTH ^a	<p>Describes the block length for the section of the LUN described by this mode parameter block descriptor. Since there is only one mode parameter block descriptor, this describes the block length of the entire LUN. For direct access devices, the block length is set to 200h or 512 bytes per block.</p> <p>When processing a MODE SELECT command, if the SATL implements direct block mapping (see 3.x.y) and the value of the BLOCK LENGTH field is not the same as the ATA logical sector size (see 3.x.y), then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST. If the SATL supports indirect block mapping this field is unspecified (see 3.4.3).</p> <p>When processing the MODE SENSE command, if the SATL implements direct block mapping (see 3.1.x) then the SATL shall return the same block length for the entire logical unit and the BLOCK LENGTH field shall contain the ATA logical sector size (see 3.1.x). Otherwise the BLOCK LENGTH field is unspecified (see 3.4.3).</p>
^a The values reported in the NUMBER OF BLOCKS field AND THE BLOCK LENGTH field shall be such that the logical unit capacity (see 3.1.x) is less than or equal to the ATA device capacity (see 3.1.x).	

~~SATL shall support only the mode parameter block descriptor formats for direct access devices.~~

Editor's Note 1: In the SAT-r08b draft, subclauses 8.3.3.1 and 8.5.5 are being combined into a subclause in clause-10 with a combined table with information from tables 14 and 19 shown in this proposal.

Remove subclause 8.5.6 mode sense block descriptor (8-byte format).

Modify the clause-9 heading as follows:

9 SCSI Block Commands (SBC) ~~Mapping~~[mapping](#)

Modify subclause 9.1 Translating LBA and transfer length and ATA command use constraints as follows:

9.1 Translating LBA and transfer length and ATA command use constraints

~~A SATL emulates SCSI logical blocks. The SCSI BLOCK LENGTH IN BYTES field in the READ CAPACITY data (see 9.8.2 and 9.9.2) may not be equal to the Logical Sector Size of the ATA device (see ATA/ATAPI-7).~~

~~NOTE 6—The Logical Sector Size indicated by an ATA device is the number of words in a logical sector. The number of bytes in an ATA device logical sector is twice the value indicated in the Logical Sector Size.~~

9.1.1 Overview

A SATL may implement a direct block mapping of ATA logical sectors to SCSI logical blocks (see 9.1.2), or the SATL may implement indirect block mapping translation (see 9.1.3).

9.1.2 Direct block mapping model

If the SATL implements direct block mapping (see 3.1.x), the logical block size indicated by the BLOCK LENGTH IN BYTES field in the READ CAPACITY data (see 9.8.2 and 9.9.2) shall equal the ATA logical sector Size (see 3.1.x). The ATA LBA of an ATA logical sector shall equal the logical block address of the corresponding SCSI logical block.

9.1.3 Indirect block mapping model

If the SATL implements indirect block mapping (see 3.1.x), the constraints of the direct block mapping model do not apply. The logical block size indicated by the BLOCK LENGTH IN BYTES field in the READ CAPACITY data (see 9.8.2 and 9.9.2) may not equal the ATA logical sector size (see 3.1.x) (e.g., SCSI logical block size of 520 bytes with an ATA Logical Sector Size of 512 bytes). The SATL translates between the SCSI LOGICAL BLOCK ADDRESS field and the ATA LBA in a vendor-specific manner. The result of a logical block address translated in one direction and then translated in the reverse direction shall yield the original logical block address.

9.1.4 Selection of ATA block commands

The ATA commands the SATL may use to implement the functions specified by SCSI block commands depend upon:

- a) the value of the logical block address and transfer length fields specified in the SCSI CDB; and
- b) the capabilities of the ~~attached~~ ATA device and the ATA host within the SATL.

Table 34 relates selection conditions to allowable ATA commands used to implement SCSI block storage data transfer commands. ATA commands listed in the Allowed ATA commands column shall not be used in the emulation of a SCSI block command if the prerequisite conditions listed in Selection Prerequisites columns are not met (i.e., the word 'yes' in a Selection Prerequisites column means the prerequisite shall be met before the SATL may use an ATA command listed in that row, and the word 'no' indicates the prerequisite need not be met for the SATL to use the ATA command listed).

Table 34 — ~~Read and write type command translation selection~~ ATA commands used for SCSI block command translations (part 1 of 3)

Selection Prerequisites ^a					Allowed ATA commands
SCSI-CDB <u>Highest ATA logical sector accessed</u>	ATA feature sets <u>required to be supported and enabled</u> ^e				
(Transfer length + LBA) Required that the logical sector address is <math>2^{28}</math> ^{b,c}	48-bit Address ^c	DMA ^d	Overlap	SATA 2.5 NCQ	
n/ano	n/ano	n/ano	n/ano	n/ano	FLUSH CACHE ^g FLUSH CACHE-EXT ^h
yes ^{e,f}	n/ano	n/ano	n/ano	n/ano	READ MULTIPLE READ SECTOR(S) READ VERIFY SECTOR(S) WRITE MULTIPLE WRITE SECTOR(S)

^a ~~An ATA command may be used to implement a SCSI block command only if all the prerequisites in the prerequisite columns for that command marked as yes are satisfied.~~

^b ~~If the SATL implements the direct mapping model (see 9.1.1) between ATA logical sectors and SCSI logical blocks, then this represents the last logical block transferred. If the SATL implements the indirect block mapping model, then this constraint is vendor-specific.~~

^c ~~If the attached ATA device does not supports neither the 48-bit Address feature set (i.e., ATA IDENTIFY DEVICE data word 83, bit 10 is set to zero see ATA/ATAPI-7) nor NCQ (see SATA 2.5), and the SATL receives a request to access an LBA beyond the LBA of the logical sector is greater than $(2^{28}-1)$, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.~~

^d ~~The DMA prerequisite requires both the ATA host in the SATL and the attached ATA device to have the same DMA transfer mode enabled (i.e., ATA IDENTIFY DEVICE data word 49, bit 8 is set to one and at least one DMA mode is enabled in the ATA IDENTIFY DEVICE data word 63 or word 88 bit 8 of word 49 in the IDENTIFY DEVICE data is set to one and at least one DMA mode is enabled in word 63 or word 88 of the IDENTIFY DEVICE data).~~

^e See ATA/ATAPI-7 ATA8-ACS.

^f ~~The SATL may transfer the number of logical blocks requested in the TRANSFER LENGTH field of the SCSI CDB by sending multiple ATA commands, each time incrementing the ATA LBA by the ATA sector count ATA Sector Count transferred.~~

^g ~~The FLUSH CACHE command may be used if ATA IDENTIFY DEVICE data indicates the command is supported in word 83 bit 12, and the command is enabled in word 86 bit 12 (see ATA/ATAPI-7).~~

^h ~~The FLUSH CACHE-EXT command may be used if ATA IDENTIFY DEVICE data indicates the command is supported in word 83 bit 13, and indicates the command is enabled in word 86 bit 13 (see ATA/ATAPI-7).~~

Table 34 — ~~Read and write type command translation selection~~ [ATA commands used for SCSI block command translations](#) (part 2 of 3)

Selection Prerequisites ^a					Allowed ATA commands
SCSI CDB Highest ATA logical sector accessed	ATA feature sets required to be supported and enabled ^e				
(Transfer length + LBA) Required that the logical sector address is <math>2^{28}</math>	48-bit Address ^c	DMA ^d	Overlap	SATA 2.5 NCQ	
yes ^{e,f}	n/ano	yes	n/ano	n/ano	READ DMA WRITE DMA
yes ^{e,f}	n/ano	yes	yes	n/ano	READ DMA QUEUED WRITE DMA QUEUED
n/ano	yes	yes	n/ano	n/ano	FLUSH CACHE EXT READ DMA EXT WRITE DMA EXT WRITE DMA FUA EXT
n/ano	yes	yes	yes	n/ano	READ DMA QUEUED EXT WRITE DMA QUEUED EXT WRITE DMA QUEUED FUA EXT

^a An ATA command may be used to implement a SCSI block command only if all the prerequisites in the prerequisite columns for that command marked as yes are satisfied.

^b If the SATL implements the direct mapping model (see 9.1.1) between ATA logical sectors and SCSI logical blocks, then this represents the last logical block transferred. If the SATL implements the indirect block mapping model, then this constraint is vendor-specific.

^c If the ~~attached~~ ATA device ~~does not~~ supports ~~neither~~ the 48-bit Address feature set (i.e., [ATA IDENTIFY DEVICE data word 83, bit 10 is set to zero](#) ~~see ATA/ATAPI-7~~) ~~nor~~ NCQ (see SATA 2.5), and ~~the SATL receives a request to access an LBA beyond the LBA of the logical sector is greater than~~ $(2^{28}-1)$, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

^d The DMA prerequisite requires both the ATA host in the SATL and the ~~attached~~ ATA device to have the same DMA transfer mode enabled (i.e., [ATA IDENTIFY DEVICE data word 49, bit 8 is set to one and at least one DMA mode is enabled in the ATA IDENTIFY DEVICE data word 63 or word 88](#) ~~bit 8 of word 49 in the IDENTIFY DEVICE data is set to one and at least one DMA mode is enabled in word 63 or word 88 of the IDENTIFY DEVICE data~~).

^e See [ATA/ATAPI-7 ATA8-ACS](#).

^f The SATL may transfer the number of logical blocks requested in the TRANSFER LENGTH field [of the SCSI CDB](#) by sending multiple ATA commands, each time incrementing the ATA LBA by the ~~ATA sector count~~ [ATA Sector Count](#) transferred.

^g ~~The FLUSH CACHE command may be used if ATA IDENTIFY DEVICE data indicates the command is supported in word 83 bit 12, and the command is enabled in word 86 bit 12 (see ATA/ATAPI-7).~~

^h ~~The FLUSH CACHE EXT command may be used if ATA IDENTIFY DEVICE data indicates the command is supported in word 83 bit 13, and indicates the command is enabled in word 86 bit 13 (see ATA/ATAPI-7).~~

Table 34 — ~~Read and write type command translation selection~~ [ATA commands used for SCSI block command translations](#) (part 3 of 3)

Selection Prerequisites ^a					Allowed ATA commands
SCSI-CDB Highest ATA logical sector accessed	ATA feature sets required to be supported and enabled ^e				
(Transfer length + LBA) Required that the logical sector address is <math>\leq 2^{28}</math>^{b,c}	48-bit Address ^c	DMA ^d	Overlap	SATA 2.5 NCQ	
n/ano	yes	n/ano	n/ano	n/ano	READ MULTIPLE EXT READ SECTOR(S) EXT READ VERIFY SECTOR(S) EXT WRITE MULTIPLE EXT WRITE MULTIPLE FUA EXT WRITE SECTOR(S) EXT
n/ano	n/ano	n/ano	n/ano	yes	READ FPDMA QUEUED WRITE FPDMA QUEUED

^a ~~An ATA command may be used to implement a SCSI block command only if all the prerequisites in the prerequisite columns for that command marked as yes are satisfied.~~

^b ~~If the SATL implements the direct mapping model (see 9.1.1) between ATA logical sectors and SCSI logical blocks, then this represents the last logical block transferred. If the SATL implements the indirect block mapping model, then this constraint is vendor-specific.~~

^c ~~If the attached ATA device does not supports neither the 48-bit Address feature set (i.e., ATA IDENTIFY DEVICE data word 83, bit 10 is set to zero see ATA/ATAPI-7) nor NCQ (see SATA 2.5), and the SATL receives a request to access an LBA beyond the LBA of the logical sector is greater than $(2^{28}-1)$, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.~~

^d ~~The DMA prerequisite requires both the ATA host in the SATL and the attached ATA device to have the same DMA transfer mode enabled (i.e., ATA IDENTIFY DEVICE data word 49, bit 8 is set to one and at least one DMA mode is enabled in the ATA IDENTIFY DEVICE data word 63 or word 88 bit 8 of word 49 in the IDENTIFY DEVICE data is set to one and at least one DMA mode is enabled in word 63 or word 88 of the IDENTIFY DEVICE data).~~

^e ~~See ATA/ATAPI-7 ATA8-ACS.~~

^f ~~The SATL may transfer the number of logical blocks requested in the TRANSFER LENGTH field of the SCSI CDB by sending multiple ATA commands, each time incrementing the ATA LBA by the ATA sector count ATA Sector Count transferred.~~

^g ~~The FLUSH CACHE command may be used if ATA IDENTIFY DEVICE data indicates the command is supported in word 83 bit 12, and the command is enabled in word 86 bit 12 (see ATA/ATAPI-7).~~

^h ~~The FLUSH CACHE EXT command may be used if ATA IDENTIFY DEVICE data indicates the command is supported in word 83 bit 13, and indicates the command is enabled in word 86 bit 13 (see ATA/ATAPI-7).~~

The SATL may use the ATA commands listed in table 34 in the translation of SCSI read commands (see 3.1.64), SCSI write commands (see 3.1.68), SCSI write and verify commands (see 3.1.69), SCSI verify commands (see 3.1.67), and SCSI synchronize cache commands (see 3.1.65) if the prerequisites defined for the command as shown in table 34 are satisfied. The translations for specific SCSI block commands in clause 9 further constrain the use of the available ATA commands in implementing the translation.

The SATL emulation of ~~a six-byte SCSI read command, SCSI verify command, SCSI write command, or SCSI write and verify command~~ [the READ \(6\) command and the WRITE \(6\) command](#) in which the TRANSFER LENGTH field ~~or VERIFICATION LENGTH field~~ is set to zero, shall translate the transfer length or verification length to 256, and issue ATA commands that operate on the ATA logical sectors corresponding to the specified 256 SCSI logical blocks.

In all other cases, the SATL shall transfer or operate on the ATA logical sectors corresponding to the number of logical blocks specified by the SCSI command.

9.2 FORMAT UNIT command

9.2.1 FORMAT UNIT command overview

The FORMAT UNIT command verifies that all logical block addresses accessible to ~~SCSI an~~ application clients are formatted and ready for data transfers. [Table 35 shows the translation for fields specified in the FORMAT UNIT CDB.](#)

Table 35 — FORMAT UNIT ~~command~~ CDB fields [translations](#)

Field	Description or reference
OPERATION CODE	If no defect list header is provided or a defect list header is provided with the DCRT bit set to one, then the SATL shall return GOOD status without issuing any commands to the ATA device. If the SATL supports certification of media and a defect list header is provided with the DCRT bit set to zero, then the SATL shall certify the media as described in 9.2.5
DEFECT LIST FORMAT	If the DEFECT LIST FORMAT field is the mandatory format (000b) or the vendor specific format (110b) the defect list length shall be zero (see SBC-2). If the DEFECT LIST FORMAT field is any other value the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
CMPLIST	If a emplist complete list is specified , (i.e., the FMTDATA bit is set to one, and the CMPLIST bit is set to one) , then the SATL shall terminate the command with a CHECK CONDITION status with sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB.
FMTDATA	If set to zero, no data shall be transferred from the data-out-buffer application client . If set to one the FORMAT UNIT parameter list shall be transferred from the client's data-out-buffer application client . The SATL may accept a FORMAT UNIT parameter list specifying the IMMED bit and an initialization pattern (see 9.2.2) . The SATL shall ignore any defect list descriptors and any other fields provided in the FORMAT UNIT parameter list (see 9.2.2) .
LONGLIST	Unspecified (see 3.4.3)
FMTPINFO	Unspecified (see 3.4.3)
RTO_REQ	If the RTO_REQ bit is set to one and the SATL implements direct block mapping (see 3.1.x), then the SATL shall return CHECK CONDITION status with sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB. If the SATL implements indirect block mapping, the translation for this field is unspecified (see 3.4.3).
CONTROL	6.4

The SATL shall process commands received during the processing of the FORMAT UNIT command as specified in SBC-2.

9.2.2 FORMAT UNIT parameter list

If the FORMAT command CDB specifies a FMTDATA bit of one, the SATL shall accept a FORMAT UNIT parameter list consisting of a short or long defect list header and may accept an initialization pattern descriptor. The SATL shall ignore any defect descriptors provided. Table 36 defines the SATL handling of fields in the FORMAT UNIT defect list header.

Table 36 — SATL defect list header

Field	Description or reference
FOV	9.2.3 and 9.2.4
DPRY	The SATL shall ignore this field.
DCRT	9.2.3 and 9.2.5
STPF	Unspecified (see 3.4.3)
IP	9.2.3 and 9.2.6
IMMED	9.2.3
DEFECT LIST LENGTH	The SATL shall ignore any defect descriptors provided.

9.2.3 SATL defect list header field combinations

Table 37 describes the actions the SATL takes depending on the values set in the IMMED bit, the FOV bit, the DCRT bit, and the IP bit.

Table 37 — SATL defect list header field combinations

IMMED	FOV	DCRT	IP	Description of SATL processing
one 1	n/a	n/a	n/a	The SATL may complete the FORMAT UNIT command immediately with GOOD status.
n/a	zero 0	n/a	n/a	
n/a	one 1	one 1	zero 0	
zero 0	one 1	zero 0	zero 0	If the SATL does not support media certification, then the SATL may terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD PARAMETER LIST. Otherwise, the SATL shall issue the required ATA read commands and ATA write commands to certify and initialize the media as specified by DCRT bit and IP bit, and shall then return GOOD status if no unrecoverable write errors occur.
		zero 0	one 1	
		one 1	one 1	

9.2.4 FOV bit

The FOV bit may be set to one to include an initialization pattern descriptor and no defect descriptors, otherwise the SATL may terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD PARAMETER LIST.

9.2.5 DCRT bit

If the DCRT bit is set to zero and media certification is supported by the SATL, then the SATL shall issue ATA verify commands (see 3.1.17) to access all the logical sectors on [the medium of](#) the ATA device's ~~medium~~ that comprise every ~~SCSI~~ logical block emulated by the SATL. For every unrecoverable read error that is encountered, the SATL shall issue an ATA write command (see 3.1.18) to the defective logical sector to attempt to cause logical sector reallocation. The data written shall be the data pattern specified by the initialization pattern descriptor, if any, or vendor-specific. After writing the affected logical sector, the SATL shall again issue an ATA verify command to the same logical sector to verify the alternate logical sector is not defective. The process ([e.g., verify, write, verify, write, ... etc.](#)) shall repeat until the logical sector is verified

successfully or the disk returns a fatal error other than an unrecoverable read error (e.g., device fault). See 5.3 for a description of error handling for multiple ATA command sequences.

9.2.6 IP bit

If the SATL supports an IP bit value of one and the IP bit is set to one, the SATL shall process the command as follows:

- a) If the attached ATA device supports the SCT LBA Segment Access (see SCT) command and the value of the INITIALIZATION PATTERN LENGTH field in the initialization pattern descriptor is four, and the value of the IP MODIFIER field in the initialization pattern descriptor is zero, then the SATL should issue an SCT LBA Segment Access command to the attached ATA device with the Function Code field set to 0001b (i.,e., Repeat Write Pattern), with the Start field and the Count field set to initialize the area of the media accessible by the application client, and with the Pattern field set to the value of the INITIALIZATION PATTERN field from the FORMAT command initialization pattern descriptor; and
- b) if the SCT LBA Segment Access command is not used to write the initialization pattern, then the SATL shall write the specified pattern by issuing ATA write commands (see 3.1.18 and 9.1) to the attached ATA device.

If the IP bit is set to zero, then the SATL shall return GOOD status.

NOTE 7 - The SATL should reverse the order of the bytes between the Pattern field in the SCT LBA Segment Access command and the value stored in the INITIALIZATION PATTERN field in the FORMAT command initialization pattern descriptor to adjust for the translation from little-endian to big-endian byte ordering.

9.3 READ commands overview

9.3.1 READ commands operation code translation

This subclause applies to the translation of SCSI READ(6), READ(10), READ(12), and READ(16) commands.

The SATL shall issue ATA read commands (see 3.1.15) in accordance with the constraints specified in 9.1 to cause the ATA device to transfer the logical blocks specified in the SCSI read command (see 3.1.64).

If the SATL returns an error other than an ILLEGAL REQUEST while processing the command the SATL may transfer a vendor-specific amount of data before terminating the command.

9.3.2 READ commands with FUA

If the SATL does not support FUA and the FUA bit is set to one, the SATL shall terminate the READ (10), READ (12) or READ (16) command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

The SATL shall process a SCSI read command with the FUA bit set to one as follows depending on whether or not the attached ATA device supports NCQ:

- a) If the attached device supports NCQ (i.e., bit-8 in word 77 of ATA IDENTIFY DEVICE data is set to one) the SATL shall issue a READ FPDMA QUEUED command (see SATA 2.5) with the FUA bit in the Device field set to one;
- b) otherwise, the SATL shall,
 - 1) if the ATA device's write cache is enabled (see ~~ATA/ATAPI-7~~ [ATA8-ACS](#)), issue an ATA verify command (see 3.1.17); and,
 - 2) issue an ATA read command as specified in 9.3.1.

9.4 READ (6) command

The READ (6) command is used to request the device to transfer logical blocks of user data to the requester. Data may be read from ATA device's medium or, data may be read from the ATA device's cache if the most recent copy is in the ATA device's cache (see SBC-2).

The READ (6) command is used to request the device to transfer logical blocks of user data to the application client (see SBC-2).

Table 38 shows the translation for fields specified in the READ (6) CDB.

Table 38 — READ (6) ~~command~~ CDB fields translations

Field	Description or reference
OPERATION CODE	9.3.1
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11) <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA read command equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u>
TRANSFER LENGTH	The transfer length shall be used to set the ATA sector count <u>ATA Sector Count</u> (see 3.1.16) ^a , <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA read command (see 3.1.x) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u>
CONTROL	6.4
^a A transfer length of zero specifies to a transfer of 256 data <u>logical</u> blocks from the attached ATA device to the application client (see SBC-2).	

9.5 READ (10) command

The SATL shall process the READ (10) command the same as the READ (6) command (see 9.3.1), with the additional fields in the CDB implemented as described in the table 39 and 9.3.2.

Table 39 — READ (10) ~~command~~ CDB fields translations

Field	Description or reference
OPERATION CODE	9.3.1 and 9.3.2
RDPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
FUA	If the SATL supports the FUA bit set to one (see 8.5.3), then <u>The the SATL may support shall implement</u> the FUA bit as defined in SBC-2 (see 9.17.2).
FUA_NV	If the FUA_NV bit is set to one the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11) <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA read command (see 3.1.x) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u>
GROUP NUMBER	Unspecified (see 3.4.3)
TRANSFER LENGTH	The transfer length shall be used to set the ATA sector count <u>ATA Sector Count</u> (see 3.1.16) ^a . <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA read command (see 3.1.x) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3)</u> The SATL shall issue as many ATA read commands (see 3.1.15) as needed to satisfy the transfer length specified by the READ (10) command.
CONTROL	6.4
^a A transfer length of zero indicates <u>specifies</u> that a data transfer shall not take place.	

9.6 READ (12) command

The SATL shall process the READ (12) command the same as the READ (10) command (see 9.3.1), with the fields in the CDB implemented as described in table 40 and 9.3.2.

Table 40 — READ (12) ~~command~~ CDB fields translations

Field	Description or reference
OPERATION CODE	9.3.1 and 9.3.2
RDPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
FUA	If the SATL supports the FUA bit set to one (see 8.5.3), then the SATL may support <u>shall implement</u> the FUA bit as defined in SBC-2 (see 9.17.2).
FUA_NV	If the FUA_NV bit is set to one the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11) <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA read command (see 3.1.x) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u>
GROUP NUMBER	Unspecified (see 3.4.3)
TRANSFER LENGTH	The transfer length shall be used to set the ATA sector count <u>ATA Sector Count</u> (see 3.1.16) ^a , <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA read command (see 3.1.x) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u> The SATL shall issue as many ATA read commands (see 3.1.15) as needed to satisfy the transfer length specified by the READ (10 <u>12</u>) command.
CONTROL	6.4

^a A transfer length of zero ~~indicates~~ specifies that a data transfer shall not take place.

9.7 READ (16) command

The SATL shall process the READ (16) command the same as the READ (10) command (see 9.3.1), with the fields in the CDB implemented as described in table 41 and 9.3.2.

Table 41 — READ (16) ~~command~~ CDB fields translations

Field	Description or reference
OPERATION CODE	9.3.1 and 9.3.2
RDPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
FUA	If the SATL supports the FUA bit set to one (see 8.5.3), then the SATL may support <u>shall implement</u> the FUA bit as defined in SBC-2 (see 9.17.2).
FUA_NV	If the FUA_NV bit is set to one the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11) <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA read command (see 3.1.x) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u>
GROUP NUMBER	Unspecified (see 3.4.3)
TRANSFER LENGTH	The transfer length shall be used to set the ATA sector count <u>ATA Sector Count</u> (see 3.1.16) ^a , <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA read command (see 3.1.x) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u> The SATL shall issue as many ATA read commands (see 3.1.15) as needed to satisfy the transfer length specified by the READ (10 <u>16</u>) command.
CONTROL	6.4

^a A transfer length of zero ~~indicates~~ specifies that a data transfer shall not take place.

9.8 READ CAPACITY (10) command

9.8.1 READ CAPACITY (10) command overview

The READ CAPACITY (10) command (see SBC-2) shall requests that the device server transfer 8 bytes of parameter data describing information about the capacity and medium format of the direct-access block device being addressed to the application client.

Table 42 shows the translation for fields specified in the READ CAPACITY(10) CDB.

Table 42 — READ CAPACITY (10) command CDB fields translations

Field	Description or reference
OPERATION CODE	The SATL shall use ATA IDENTIFY DEVICE data information to compute the ATA device's maximum user addressable medium capacity of the ATA device.
LOGICAL BLOCK ADDRESS	If the LOGICAL BLOCK ADDRESS field is not set to zero, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
PMI	If the PMI bit is not set to zero the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
CONTROL	6.4

9.8.2 READ CAPACITY (10) parameter data

The SATL shall return READ CAPACITY (10) parameter data as defined by SBC-2. Table 45 Table 43 describes the translation of fields in the READ CAPACITY (10) parameter data.

Table 43 — READ CAPACITY (10) parameter data

Field	Description or reference
RETURNED LOGICAL BLOCK ADDRESS ^a	If the number of logical blocks exceeds the maximum value that is able to be specified in the RETURNED LOGICAL BLOCK ADDRESS field, the device server shall set the RETURNED LOGICAL BLOCK ADDRESS field to FFFFFFFFh. The application client should then issue a READ CAPACITY (16) command (see 9.9) to retrieve the READ CAPACITY (16) parameter data. This field is otherwise Unspecified unspecified (see 3.4.3).
BLOCK LENGTH IN BYTES ^a	If the SATL implements direct block mapping (see 3.1.x) then the BLOCK LENGTH IN BYTES field shall contain the ATA logical sector size (see 3.1.x). Otherwise the BLOCK LENGTH field is unspecified Unspecified (see 3.4.3).
^a The values reported in the RETURNED LOGICAL BLOCK ADDRESS field AND THE BLOCK LENGTH IN BYTES field shall be such that the logical unit capacity (see 3.1.x) is less than or equal to the ATA device capacity (see 3.1.x).	

9.9 READ CAPACITY (16) command

9.9.1 READ CAPACITY (16) command overview

The READ CAPACITY (16) command (see SBC-2) shall request that the device server transfer parameter data describing information about the capacity and medium format of the direct-access block device being addressed to the application client.

Table 44 shows the translation for fields specified in the READ CAPACITY(16) CDB.

Table 44 — READ CAPACITY(16) command CDB fields translations

Field	Description or reference
OPERATION CODE	Shall be set to 9Eh/10h. The SATL shall use ATA IDENTIFY DEVICE data information to compute the ATA device's maximum user addressable medium capacity of the ATA device.
SERVICE ACTION	
LOGICAL BLOCK ADDRESS	If the LOGICAL BLOCK ADDRESS field is not set to zero 0000000h the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
ALLOCATION LENGTH	Unspecified (see 3.4.3)
PMI	If the PMI bit is not set to zero the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
CONTROL	6.4

9.9.2 READ CAPACITY (16) parameter data

The SATL shall return READ CAPACITY (16) parameter data as defined by SBC-2. Table 45 describes the translation of fields in the READ CAPACITY (16) parameter data.

Table 45 — READ CAPACITY (16) data

Field	Description or reference
RETURNED LOGICAL BLOCK ADDRESS ^a	The maximum value that shall be returned in the RETURNED LOGICAL BLOCK ADDRESS field is FFFFFFFF FFFFFFFEh. This field is otherwise unspecified. Unspecified (see 3.4.3).
BLOCK LENGTH IN BYTES ^a	If the SATL implements direct block mapping (see 3.1.x) then the BLOCK LENGTH IN BYTES field shall contain the ATA logical sector size (see 3.1.x). Otherwise the BLOCK LENGTH field is unspecified. Unspecified (see 3.4.3).
RTO_EN	Unspecified (see 3.4.3)
PROT_EN	Unspecified (see 3.4.3)
^a The values reported in the RETURNED LOGICAL BLOCK ADDRESS field AND THE BLOCK LENGTH IN BYTES field shall be such that the logical unit capacity (see 3.1.x) is less than or equal to the ATA device capacity (see 3.1.x).	

9.10 REASSIGN BLOCKS command

9.10.1 REASSIGN BLOCKS command overview

The REASSIGN BLOCKS command requests that the SATL reassign defective logical blocks (see SBC-2). ATA devices do not support or have a direct translation for the REASSIGN BLOCKS command. The SATL shall emulate the ~~SCSI~~ REASSIGN BLOCKS command as defined in table 46.

Table 46 — REASSIGN BLOCKS ~~command~~ CDB fields ~~translations~~

Field	Description or reference
OPERATION CODE	9.10.2
LONGLBA	See SBC-2
LONGLIST	See SBC-2
CONTROL	6.4

The REASSIGN BLOCKS command parameter list ~~provided in the data-out buffer transferred from the application client~~ contains the LBAs ~~of logical blocks~~ to be reassigned. ~~The LBAs provided in the parameter list shall be utilized for the LBAs in any ATA verify commands (see 3.1.17) or ATA write commands (see 3.1.18) issued by the SATL.~~

If the SATL implements direct block mapping (see 9.1.2), then the values set by the SATL in the ATA LBA of the ATA verify command(s) and ATA write command(s) shall equal the value(s) of the LBAs in the parameter list. Otherwise, the mapping is unspecified (see 3.4.3).

The SATL shall support the LONGLBA ~~field~~ bit and the LONGLIST ~~field~~ bit (see SBC-2).

9.10.2 REASSIGN BLOCKS operation code

The SATL shall accept a parameter list specifying LBAs to of logical blocks be reassigned (see SBC-2).

~~For each LBA in the parameter list, the SATL shall:~~

- ~~1) issue an ATA verify command (see 3.1.17) to the specified LBA;~~
- ~~2) if the ATA verify command completes successfully, then the SATL shall return GOOD status for the REASSIGN BLOCKS command;~~
- ~~3) if the ATA verify command does not complete successfully, then the SATL shall issue an ATA write command (see 3.1.18) with vendor specific data to the LBA that failed the ATA verify command;~~
- ~~4) if the ATA write command fails, then the SATL shall terminate the REASSIGN BLOCKS command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to WRITE ERROR — AUTO REALLOCATION FAILED;~~
- ~~5) if the write command completes successfully, then the SATL shall issue a second ATA verify command to the same LBA;~~
- ~~6) if the second ATA verify command completes successfully, then the SATL shall return GOOD status for the REASSIGN BLOCKS command; and~~

~~if the second ATA verify command does not complete successfully, then the SATL shall terminate the REASSIGN BLOCKS command with CHECK CONDITION status with the sense key set to MEDIUM ERROR and the additional sense code set to UNRECOVERED READ ERROR — AUTO REALLOCATE FAILED.~~

The SATL shall process each ATA LBA corresponding to LBAs specified in the parameter list as shown in figure 8.

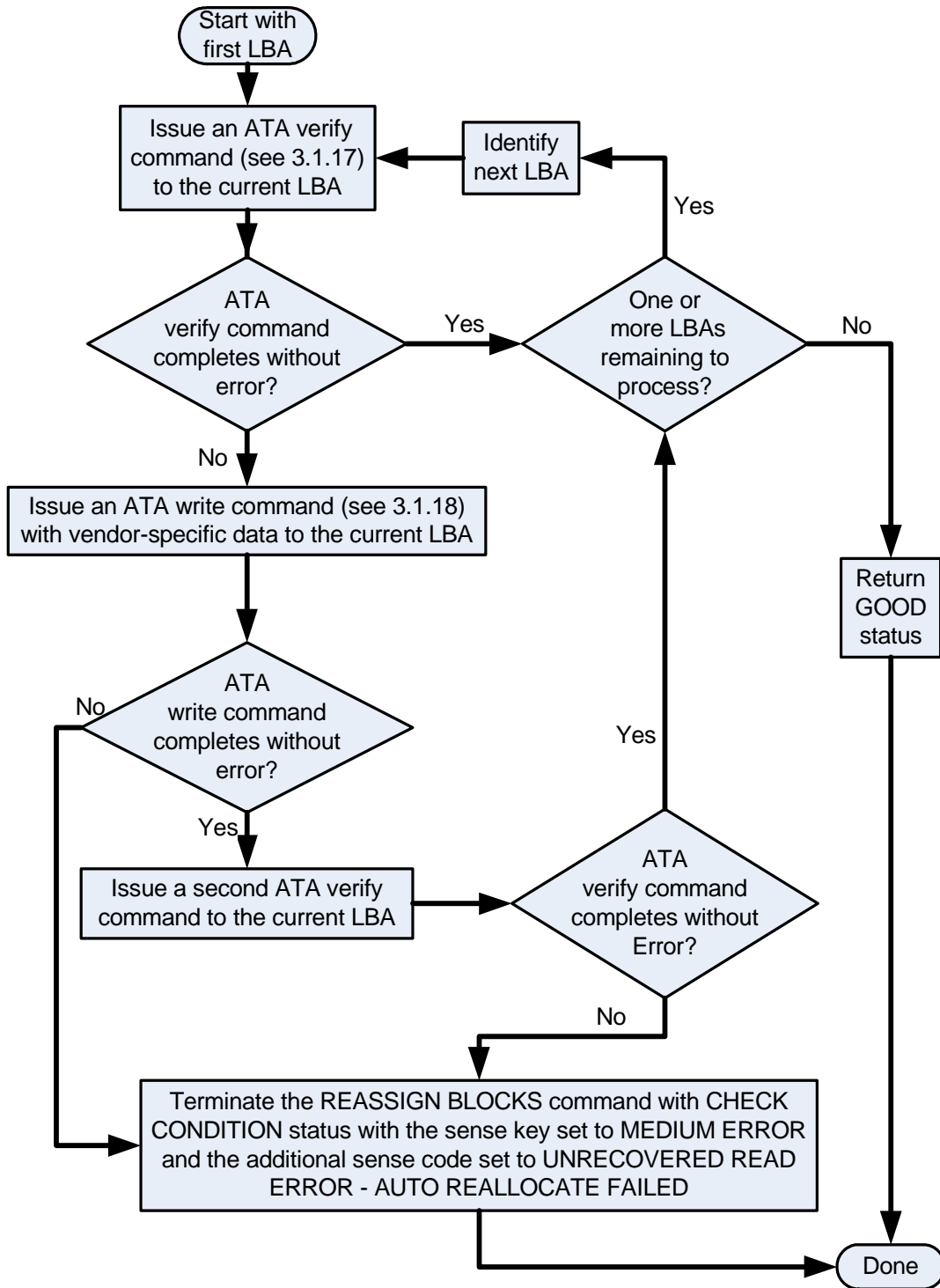


Figure 8 — REASSIGN BLOCKS command translation

9.11 START STOP UNIT command

9.11.1 START STOP UNIT command overview

The START STOP UNIT command provides a method for controlling the power state of a logical unit.

[Table 47 shows the translation for fields specified in the START STOP UNIT CDB.](#)

Table 47 — START/STOP UNIT ~~command~~ CDB fields ~~s~~ translations

Field	Description or reference
OPERATION CODE	Commands issued to the attached ATA device depend upon the other values in the CDB as described in 9.11.2.
IMMED	The SATL shall implement this field as defined in 9.11.2 and 9.11.3.
POWER CONDITIONS	The SATL shall ignore this field. If the POWER CONDITION field is not set to zero, the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
LOEJ	The SATL shall implement this field as defined in 9.11.3.
START	The SATL shall implement this field as defined in 9.11.3.
CONTROL	6.4

If ~~a~~ [the](#) SATL receives a command ~~other than a START STOP UNIT command that requires medium access while for a~~ [the](#) device ~~that~~ is in the Stopped state (see SBC-2), then the SATL shall return CHECK CONDITION status, with the sense key set to NOT READY and the additional sense code set to LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED.

9.11.2 ~~IMMED bit processing for the START STOP UNIT command~~ [Processing ending status if an error occurs](#)

9.11.2.1 ~~Before processing the START STOP UNIT command~~

The SATL shall:

- ~~1) return GOOD status if the IMMED bit is set to one, and~~
- ~~2) continue processing the START STOP UNIT command.~~

9.11.2.2 ~~After START STOP UNIT completes with no error~~

~~If no error occurs during the processing of the START STOP UNIT command and the IMMED bit is set to zero, then the SATL shall return GOOD status.~~

9.11.2.3 ~~After START STOP UNIT completes with an error~~

If an error occurs during the processing of the START STOP UNIT command and the IMMED bit is set to zero, [then](#) the SATL shall terminate the [START STOP UNIT](#) command with CHECK CONDITION status with a sense key of ABORTED COMMAND, and the additional sense code [set to the value](#) ~~for~~ specified for the error being reported ([see table 48](#)).

If an error occurs during the processing of the START STOP UNIT command and the IMMED bit is set to one, [then](#) the SATL shall terminate the [START STOP UNIT](#) command and return CHECK CONDITION status as a deferred error (see SPC-3) with a sense key of ABORTED COMMAND, and the additional sense code set to the value specified for the error being reported ([see table 48](#)).

9.11.3 START STOP UNIT START bit LOEJ bit combinations

The SATL shall perform the actions shown in table 48 in response to a START STOP UNIT command.

Table 48 — Definition of **IMMED**, LOEJ, and START bits in the START STOP UNIT CDB (part 1 of 2)

START	LOEJ	Definition
0	0	<p>The SATL shall:</p> <ol style="list-style-type: none"> 1) Process If the IMMED bit is set to one, then return GOOD status (see 9.11.2.1); 2) Issue a FLUSH CACHE or FLUSH CACHE EXT an ATA flush command (see 3.1.9) to the attached ATA device; 3) If the FLUSH CACHE or FLUSH CACHE EXTENDED ATA flush command completes with any error, then process ending status according to the IMMED bit (see 9.11.2.3) with the additional sense code set to COMMAND SEQUENCE ERROR; 4) If the FLUSH CACHE or FLUSH CACHE EXTENDED ATA flush command completes without no error, then issue an ATA STANDBY IMMEDIATE command to the attached ATA device with the ATA zero in Sector Count set to zero; 5) If the ATA STANDBY IMMEDIATE command completes with any error, then process ending status according to the IMMED bit (see 9.11.2.3) with the additional sense code set to COMMAND SEQUENCE ERROR; and 6) If the ATA STANDBY IMMEDIATE command completes without no error^a and the IMMED bit is set to zero, then process return GOOD status according to the IMMED bit (see 9.11.2.2)(see 9.11.2)^b.
1	0	<p>The SATL shall:</p> <ol style="list-style-type: none"> 1) Process If the IMMED bit is set to one, then return GOOD status (see 9.11.2.1); 2) Issue an ATA verify command (see 3.1.17) to the attached ATA device with one in the ATA Sector Count set to one and a value in the LBA set to a value between from zero to and the maximum LBA supported by the ATA device in its current configuration^d; and 3) If the IMMED bit is set to one, then return GOOD status. When command completion is received for the ATA verify command (see 3.1.17) process GOOD status according to the IMMED bit (see 9.11.2.2)^c.
0	1	<p>If the attached ATA device supports the Removable Media feature set (see ATA/ATAPI-7), then the SATL shall:</p> <ol style="list-style-type: none"> 1) Process If the IMMED bit is set to one, then return GOOD status (see 9.11.2.1); 2) Issue an ATA MEDIA EJECT command to the attached ATA device; 3) If the ATA MEDIA EJECT command completes with any error, then process ending status according to the IMMED bit (see 9.11.2.3) with the additional sense code set to MEDIA LOAD OR EJECT FAILED; and 4) If the ATA MEDIA EJECT command completes without no error and the IMMED bit is set to zero, then process return GOOD status according to the IMMED bit (see 9.11.2.2). <p>If the attached ATA device does not support the Removable Media feature set, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.</p>
<p>^a An ATA device may return completion status for a STANDBY command before removal of the ATA device may be accomplished without damaging the ATA device.</p> <p>^b After returning GOOD status for a START STOP UNIT command with the START bit set to zero, the SATL shall consider the ATA device to be in the Stopped power state (see SBC-2).</p> <p>^c After returning GOOD status for a START STOP UNIT command with the START bit set to one, the SATL shall consider the ATA device to be in the Active power state (see SBC-2).</p> <p>^d An ATA device medium access occurs when an LBA is specified whose data is not contained in ATA device's cache memory. If a value in LBA is specified for an ATA verify command where the data is contained in ATA device's cache memory, then an ATA device may not be in the Active power mode (see ATA/ATAPI-7 ATA8-ACS) after completion of the ATA verify command.</p>		

Table 48 — Definition of ~~IMMED~~, LOEJ, and START bits in the START STOP UNIT CDB (part 2 of 2)

START	LOEJ	Definition
1	1	The SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, with the additional sense code set to INVALID FIELD IN CDB.
<p>^a An ATA device may return completion status for a STANDBY command before removal of the ATA device may be accomplished without damaging the ATA device.</p> <p>^b After returning GOOD status for a START STOP UNIT command with the START bit set to zero, the SATL shall consider the ATA device to be in the Stopped power state (see SBC-2).</p> <p>^c After returning GOOD status for a START STOP UNIT command with the START bit set to one, the SATL shall consider the ATA device to be in the Active power state (see SBC-2).</p> <p>^d An ATA device medium access occurs when an LBA is specified whose data is not contained in ATA device's cache memory. If a value in LBA is specified for an ATA verify command where the data is contained in ATA device's cache memory, then an ATA device may not be in the Active power mode (see ATA/ATAPI-7 ATA8-ACS) after completion of the ATA verify command.</p>		

9.12 SYNCHRONIZE CACHE (10) command

9.12.1 SYNCHRONIZE CACHE (10) command overview

The SYNCHRONIZE CACHE (10) command is used to flush the most recent data values in the ATA device's cache of the ATA device to physical medium. Unlike in SCSI, ATA does not provide a way to specify a particular LBA to start flushing the ATA device's cache.

Table 49 shows the translation for fields specified in the SYNCHRONIZE CACHE(10) CDB.

Table 49 — SYNCHRONIZE CACHE (10) command CDB fields translations

Field	Description or reference
OPERATION CODE	The SATL shall issue an ATA flush command (see 3.1.8) in accordance with the constraints described in 9.1
SYNC_NV	Unspecified (see 3.4.3)
IMMED	If the IMMED bit is set to one the SATL shall return GOOD status and then issue an ATA flush command. If the IMMED bit is set to zero the SATL shall issue an ATA flush command and return status upon completion.
LOGICAL BLOCK ADDRESS	The SATL shall ignore this field and shall process this command as though this field contained zero.
GROUP NUMBER	Unspecified (see 3.4.3)
NUMBER OF BLOCKS	The SATL shall ignore this field and shall process this command as though this field contained zero (see SBC-2) (i.e., synchronize all logical blocks starting with the one specified in the LOGICAL BLOCK ADDRESS field to the last logical block on the ATA device's medium).
CONTROL	6.4

9.13 SYNCHRONIZE CACHE (16) command

9.13.1 SYNCHRONIZE CACHE (16) command overview

The SYNCHRONIZE CACHE (16) command is used to flush the most recent data values in the ATA device's cache of the ATA device to physical medium. Unlike in SCSI, ATA does not provide a way to specify a particular LBA to start flushing the ATA device's cache.

[Table 50 shows the translation for fields specified in the SYNCHRONIZE CACHE\(10\) CDB.](#)

Table 50 — SYNCHRONIZE CACHE(160) command CDB fields translations

Field	Description or reference
OPERATION CODE	The SATL shall issue an ATA flush command (see 3.1.8) in accordance with the constraints described in 9.1
SYNC_NV	Unspecified (see 3.4.3)
IMMED	If the IMMED bit is set to one the SATL shall return GOOD status immediately and then issue an ATA flush command. If the IMMED bit is set to zero , the SATL shall issue an ATA flush command and return status upon completion.
LOGICAL BLOCK ADDRESS	The SATL shall ignore this field and shall process this command as though this field contained zero.
GROUP NUMBER	Unspecified (see 3.4.3)
NUMBER OF BLOCKS	The SATL shall ignore this field and shall process this command as though this field contained zero (see SBC-2) (i.e., synchronize all logical blocks starting with the one specified in the LOGICAL BLOCK ADDRESS field to the last logical block on the ATA device's medium).
CONTROL	6.4

9.14 VERIFY (10) command

The VERIFY (10) command is used to verify data on the ATA device's medium. Table 51 describes the translation of fields in the VERIFY (10) CDB.

Table 51 — VERIFY (10) command CDB fields [translations](#)

Field	Description or reference
OPERATION CODE	The SATL shall issue an ATA verify command (see 3.1.17) in accordance with the constraints defined in 9.1.
VRPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
BYTCHK	If the SATL supports a bytechk value of BYTCHK bit set to one and if the CDB specifies one in the BYTCHK bit is set to one field, then the SATL shall perform a byte-by-byte comparison of the data transferred from the SCSI application client to the SATL with data read from the ATA device by the SATL, and shall return completion status reflecting the results of that the comparison as described in (see SBC-2).
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11) as defined by 9.1 . If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA verify command (see 3.1.17) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).
GROUP NUMBER	Unspecified (see 3.4.3)
VERIFICATION LENGTH	The verification length shall be used to set the ATA sector count ATA Sector Count (see 9.1) as defined in 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.16) in the ATA verify command (see 3.1.x) equal to the value specified in the VERIFICATION LENGTH field. Otherwise, the mapping is unspecified (see 3.4.3).
CONTROL	6.4

9.15 VERIFY (12) command

Table 52 describes the translation of fields in the VERIFY (12) CDB.

Table 52 — VERIFY (12) ~~command~~ CDB fields [translations](#)

Field	Description or reference
OPERATION CODE	The SATL shall issue an ATA verify command (see 3.1.17) in accordance with the constraints defined in 9.1.
VRPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
BYTCHK	If the SATL supports a bytechk value of BYTCHK bit set to one and if the CDB specifies one in the BYTCHK bit is set to one field, then the SATL shall perform a byte-by-byte comparison of the data transferred from the SCSI application client to the SATL with data read from the ATA device by the SATL, and shall return completion status reflecting the results of that the comparison as described in (see SBC-2).
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11) as defined by 9.1 . If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA verify command (see 3.1.17) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).
GROUP NUMBER	Unspecified (see 3.4.3)
VERIFICATION LENGTH	The verification length shall be used to set the ATA sector count ATA Sector Count (see 9.1) as defined in 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.16) in the ATA verify command (see 3.1.x) equal to the value specified in the VERIFICATION LENGTH field. Otherwise, the mapping is unspecified (see 3.4.3).
CONTROL	6.4

9.16 VERIFY (16) command

Table 53 describes the translation of fields in the VERIFY (16) CDB.

Table 53 — VERIFY (16) ~~command~~ CDB fields [translations](#)

Field	Description or reference
OPERATION CODE	The SATL shall issue an ATA verify command (see 3.1.17) in accordance with the constraints defined in 9.1.
VRPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
BYTCHK	If the SATL supports a bytechk value of BYTCHK bit set to one and if the CDB specifies one in the BYTCHK bit is set to one field, then the SATL shall perform a byte-by-byte comparison of the data transferred from the SCSI application client to the SATL with data read from the ATA device by the SATL, and shall return completion status reflecting the results of that the comparison as described in (see SBC-2).
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11) as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA verify command (see 3.1.17) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).
GROUP NUMBER	Unspecified (see 3.4.3)
VERIFICATION LENGTH	The verification length shall be used to set the ATA sector count ATA Sector Count (see 9.1) as defined in 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.16) in the ATA verify command (see 3.1.x) equal to the value specified in the VERIFICATION LENGTH field. Otherwise, the mapping is unspecified (see 3.4.3).
CONTROL	6.4

9.17 WRITE commands overview

9.17.1 WRITE commands [operation code](#) translation

[This subclause applies to the translation of SCSI WRITE\(6\), WRITE\(10\), WRITE\(12\), and WRITE\(16\) commands.](#)

The SATL shall transfer the logical blocks specified in the ~~SCSI~~ write command (see 3.1.68) from the ~~SCSI~~ application client [to the ATA device, and The SATL](#) shall issue ATA write commands (see 3.1.18) in accordance with the constraints specified in 9.1 ~~to transfer the specified logical blocks through the ATA host in the SATL to the ATA device.~~

~~Data blocks specified in the LOGICAL BLOCK ADDRESS field shall be transferred to the specified ATA device, and the ATA device may transfer the data to its cache or medium.~~

9.17.2 WRITE commands with FUA

This subclause applies to the translation of SCSI WRITE (10), WRITE (12), and WRITE (16) commands.

If the FUA bit is [set to zero in the SCSI write command CDB, then](#) the SATL shall process this command as described in 9.17.1.

If the FUA bit is [set to one in the SCSI write command CDB](#), then the SATL shall issue [the following](#), ~~an ATA write FUA command sequence (see 3.1.19) to the attached ATA device~~ in accordance with the constraints described in 9.1.:

- a) two ATA commands as follows:
 - 1) an ATA write command (see 3.1.xx) excluding WRITE DMA FUA EXT, WRITE DMA QUEUED FUA EXT, WRITE MULTIPLE FUA EXT, and WRITE FPDMA QUEUE; and
 - 2) an ATA verify command (see 3.1.xx);
 - b) one of the following ATA commands (see ATA8-ACS):
 - A) WRITE DMA FUA EXT;
 - B) WRITE DMA QUEUED FUA EXT; or
 - C) WRITE MULTIPLE FUA EXT;
- or
- c) an ATA WRITE FPDMA QUEUED command (see SATA 2.5) with the FUA bit in the Device register set to one.

See 5.3 for a description of multiple command sequence error handling.

9.18 WRITE (6) command

The WRITE (6) command is used to request the ~~ATA device~~ [SATL](#) to transfer user data [from the application client](#) to the ATA device's ~~medium or to the ATA device's cache~~. Data may be written to the ~~ATA device's~~ medium or to [the cache of the](#) ATA device's ~~cache~~.

[Table 54 shows the translation for fields specified in the WRITE\(6\) CDB.](#)

Table 54 — WRITE (6) ~~command~~ CDB fields [translations](#)

Field	Description or reference
OPERATION CODE	9.17.1
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11), as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.18) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).
TRANSFER LENGTH ^a	The transfer length shall be used to set the ATA sector count ATA Sector Count (see 3.1.16) ^a , as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA write command (see 3.1.18) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).
CONTROL	6.4
^a A TRANSFER LENGTH of field set to zero specifies to a transfer of 256 data blocks logical blocks from the application client to the attached ATA device (see SBC-2).	

9.19 WRITE (10) command

The WRITE (10) command is used to request the ~~ATA device~~ [SATL](#) to transfer user data [from the application client](#) to the ATA device's ~~medium or to the ATA device's cache~~. Data may be written to the ~~ATA device's~~ medium or to [the cache of the](#) ATA device's ~~cache~~.

[Table 55 shows the translation for fields specified in the WRITE\(10\) CDB.](#)

Table 55 — WRITE (10) ~~command~~ CDB fields [translations](#)

Field	Description or reference
OPERATION CODE	9.17.1 and 9.17.2
WRPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
FUA	If the SATL supports the FUA bit set to one (see 8.5.3), then The the SATL may support shall implement the FUA bit as defined in SBC-2 (see 9.17.2).
FUA_NV	If the FUA_NV bit is set to one the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11), as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.18) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).
GROUP NUMBER	Unspecified (see 3.4.3)
TRANSFER LENGTH ^a	The transfer length shall be used to set the ATA sector count ATA Sector Count (see 3.1.16) ^a , as defined by 9.1. . If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA write command (see 3.1.18) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3). The SATL shall issue as many ATA write commands (see 3.1.18) as needed to satisfy the transfer length specified by the WRITE (10) command.
CONTROL	6.4
^a A transfer length of zero indicates specifies that a data transfer shall not take place.	

9.20 WRITE (12) command

The WRITE (12) command is used to request the ~~ATA device~~ [SATL](#) to transfer user data [from the application client](#) to the ATA device's ~~medium or to the ATA device's cache~~. Data may be written to the ~~ATA device's~~ medium or to [the cache of the](#) ATA device's ~~cache~~.

[Table 56 shows the translation for fields specified in the WRITE\(12\) CDB.](#)

Table 56 — WRITE (12) ~~command~~ CDB fields [translations](#)

Field	Description or reference
OPERATION CODE	9.17.1 and 9.17.2
WRPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
FUA	If the SATL supports the FUA bit set to one (see 8.5.3), then The the SATL may-support shall implement the FUA bit as defined in SBC-2 (see 9.17.2).
FUA_NV	If the FUA_NV bit is set to one the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11), as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.18) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).
GROUP NUMBER	Unspecified (see 3.4.3)
TRANSFER LENGTH ^a	The transfer length shall be used to set the ATA sector count ATA Sector Count (see 3.1.16) ^a , as defined by 9.1. . If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA write command (see 3.1.18) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3). The SATL shall issue as many ATA write commands (see 3.1.18) as needed to satisfy the transfer length specified by the WRITE (12) command.
CONTROL	6.4
^a A transfer length of zero indicates specifies that a data transfer shall not take place.	

9.21 WRITE (16) command

The WRITE (16) command is used to request the ~~ATA device~~ [SATL](#) to transfer user data [from the application client](#) to the ATA device's ~~medium or to the ATA device's cache~~. Data may be written to the ~~ATA device's~~ [medium or to the cache of the](#) ATA device's ~~cache~~.

[Table 57 shows the translation for fields specified in the WRITE\(16\) CDB.](#)

Table 57 — WRITE (16) ~~command~~ CDB fields [translations](#)

Field	Description or reference
OPERATION CODE	9.17.1 and 9.17.2
WRPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
FUA	If the SATL supports the FUA bit set to one (see 8.5.3), then <u>The the SATL may support shall implement</u> the FUA bit as defined in SBC-2 (see 9.17.2).
FUA_NV	If the FUA_NV bit is set to one the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. <u>The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2.</u>
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11), <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.18) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u>
GROUP NUMBER	Unspecified (see 3.4.3)
TRANSFER LENGTH ^a	The transfer length shall be used to set the ATA sector count <u>ATA Sector Count</u> (see 3.1.16) ^a , <u>as defined by 9.1. . If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA write command (see 3.1.18) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u> The SATL shall issue as many ATA write commands (see 3.1.18) as needed to satisfy the transfer length specified by the WRITE (16) command.
CONTROL	6.4
^a A transfer length of zero indicates <u>specifies</u> that a data transfer shall not take place.	

9.22 WRITE AND VERIFY commands overview

This subclause applies to the translation of ~~SCSI~~ WRITE AND VERIFY (10), WRITE AND VERIFY (12), and WRITE AND VERIFY (16) commands.

The SATL shall issue:

- 1) an ATA write command (see 3.1.18) in accordance with the constraints defined in 9.1; and
- 2) an ATA verify command (see 3.1.17).

9.23 WRITE AND VERIFY (10) command

The WRITE AND VERIFY (10) command ~~requests that the SATL is used to~~ transfer the specified logical blocks from the application client data to the ATA device's ~~medium~~, and then ~~to~~ verify that the data was written correctly to the medium of the ATA device.

[Table 58 shows the translation for fields specified in the WRITE AND VERIFY \(10\) CDB.](#)

Table 58 — WRITE AND VERIFY (10) ~~command~~ CDB fields translations

Field	Description or reference
OPERATION CODE	9.22
WRPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
BYTCHK	<p>If the application client specifies a value other than zero in this field, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.</p> <p><u>If the SATL supports a BYTCHK bit set to one and the BYTCHK bit is set to one, then after writing the data to the medium the SATL shall read the data from the medium and perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device, and then shall return completion status reflecting the results of the comparison (see SBC-2).</u></p>
LOGICAL BLOCK ADDRESS	<p>The logical block address shall be used to set the ATA LBA (see 3.1.11), <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.18) and the ATA verify command (see 3.1.17) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u></p>
GROUP NUMBER	Unspecified (see 3.4.3)
TRANSFER LENGTH ^a	<p>The transfer length shall be used to set the ATA-sector-count <u>ATA Sector Count</u> (see 3.1.16)^a, <u>as defined by 9.1. . If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA write command (see 3.1.18) and the ATA verify command (see 3.1.17) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u> The SATL shall issue as many ATA write commands (see 3.1.18) as needed to satisfy the transfer length specified by the WRITE AND VERIFY (10) command.</p>
CONTROL	6.4
<p>^a A transfer length of zero indicates <u>specifies</u> that a data transfer shall not take place.</p>	

9.24 WRITE AND VERIFY (12) command

The WRITE AND VERIFY (12) command ~~requests that the SATL is used to~~ transfer the specified logical blocks from the application client data to the ATA device's ~~medium~~, and then ~~to~~ verify that the data was written correctly to the medium of the ATA device.

[Table 59 shows the translation for fields specified in the WRITE AND VERIFY \(12\) CDB.](#)

Table 59 — WRITE AND VERIFY (12) ~~command~~ CDB fields translations

Field	Description or reference
OPERATION CODE	9.22
WRPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
BYTCHK	<p>If the application client specifies a value other than zero in this field, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.</p> <p><u>If the SATL supports a BYTCHK bit set to one and the BYTCHK bit is set to one, then after writing the data to the medium the SATL shall read the data from the medium and perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device, and then shall return completion status reflecting the results of the comparison (see SBC-2).</u></p>
LOGICAL BLOCK ADDRESS	<p>The logical block address shall be used to set the ATA LBA (see 3.1.11), <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.18) and the ATA verify command (see 3.1.17) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u></p>
GROUP NUMBER	Unspecified (see 3.4.3)
TRANSFER LENGTH ^a	<p>The transfer length shall be used to set the ATA-sector-count <u>ATA Sector Count</u> (see 3.1.16)^a, <u>as defined by 9.1. . If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA write command (see 3.1.18) and the ATA verify command (see 3.1.17) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u> The SATL shall issue as many ATA write commands (see 3.1.18) as needed to satisfy the transfer length specified by the WRITE AND VERIFY (12) command.</p>
CONTROL	6.4
<p>^a A transfer length of zero indicates <u>specifies</u> that a data transfer shall not take place.</p>	

9.25 WRITE AND VERIFY (16) command

The WRITE AND VERIFY (16) command ~~requests that the SATL is used to~~ transfer the specified logical blocks from the application client data to the ATA device's ~~medium~~, and then ~~to~~ verify that the data was written correctly to the medium of the ATA device.

[Table 60 shows the translation for fields specified in the WRITE AND VERIFY \(16\) CDB.](#)

Table 60 — WRITE AND VERIFY (16) ~~command~~ CDB fields translations

Field	Description or reference
OPERATION CODE	9.22
WRPROTECT	Unspecified (see 3.4.3)
DPO	Unspecified (see 3.4.3)
BYTCHK	<p>If the application client specifies a value other than zero in this field, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.</p> <p><u>If the SATL supports a BYTCHK bit set to one and the BYTCHK bit is set to one, then after writing the data to the medium the SATL shall read the data from the medium and perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device, and then shall return completion status reflecting the results of the comparison (see SBC-2).</u></p>
LOGICAL BLOCK ADDRESS	<p>The logical block address shall be used to set the ATA LBA (see 3.1.11), <u>as defined by 9.1. If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.18) and the ATA verify command (see 3.1.17) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u></p>
GROUP NUMBER	Unspecified (see 3.4.3)
TRANSFER LENGTH ^a	<p>The transfer length shall be used to set the ATA-sector-count <u>ATA Sector Count</u> (see 3.1.16)^a, <u>as defined by 9.1. . If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the ATA Sector Count (see 3.1.x) in the ATA write command (see 3.1.18) and the ATA verify command (see 3.1.17) equal to the value specified in the TRANSFER LENGTH ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).</u> The SATL shall issue as many ATA write commands (see 3.1.18) as needed to satisfy the transfer length specified by the WRITE AND VERIFY (16) command.</p>
CONTROL	6.4
<p>^a A transfer length of zero indicates <u>specifies</u> that a data transfer shall not take place.</p>	

9.26 WRITE SAME (10) command

9.26.1 WRITE SAME (10) command overview

The WRITE SAME (10) command (see table 61) requests that the SATL transfer a single logical block from the ~~data-out-buffer~~ [application client](#) and write the contents of that single logical block, with modifications based on the LBDATA bit and the PBDATA bit, to the specified range of logical block addresses [on the ATA device](#).

Table 61 — WRITE SAME (10) ~~command~~ CDB fields [translations](#)

Field	Description or reference
OPERATION CODE	If the ATA device supports SCT LBA Segment Access (see SCT) the ATA SCT Write Same command (see ATA8-ACS) , the SATL should issue SCT LBA Segment Access the ATA SCT Write Same command to repeatedly write the data block transferred from the application client in the buffer to the ATA device . If the ATA device does not implement SCT the ATA SCT Write Same command then the SATL shall issue ATA write commands as defined in 9.17.1.
WRPROTECT	Unspecified (see 3.4.3)
LBDATA	9.26.2
PBDATA	9.26.2
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11), as defined by 9.1 . If the SATL implements direct block mapping (see 3.1.x) , then the SATL shall set the Start field in the ATA SCT Write Same command or the ATA LBA in the ATA write command equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).
GROUP NUMBER	Unspecified (see 3.4.3)
NUMBER OF BLOCKS	A NUMBER OF BLOCKS of zero indicates that the data-out-buffer SATL shall be repeatedly written from the specified logical block address through the last user-addressable logical sector on the media. A NUMBER OF BLOCKS field set to zero specifies that the SATL shall repeatedly write the logical block transferred from the application client to the range of ATA logical sectors corresponding to the range of logical block addresses specified by the LOGICAL BLOCK ADDRESS field through the LBA of the last logical block on the logical unit. If the NUMBER OF BLOCKS field is set to a value other than not zero , the SATL shall repeatedly write the data-out-buffer data block transferred from the application client to the medium of the ATA device for the number of logical blocks sectors specified to the corresponding logical sectors on the ATA device . The SATL shall send as many ATA commands as required to satisfy the number of logical blocks specified by the WRITE SAME (10) command.
CONTROL	6.4

9.26.2 LBDATA bit and PBDATA bit

The SATL shall write data to the specified logical block addresses according to the values in the LBDATA and PBDATA bits as shown in table 62.

Table 62 — LBDATA and PBDATA fields

LBDATA	PBDATA	Description
0	0	The SATL shall transfer the single block of data transferred from the data-output buffer application client to the range of blocks specified in LOGICAL BLOCK ADDRESS field and NUMBER OF BLOCKS field , repeatedly, on the media medium of the ATA device . If the ATA device supports the SCT LBA Segment Access capability ATA SCT Write Same command , then the SATL should use the ATA SCT Write Same command with the Function Code set to 002h or 004h for writing the data this should be used for the data-transfer . Otherwise, the SATL shall use ATA write commands shall be used as documented defined in 9.12.2 9.17.2 . S (see SBC-2).
1	0	The SATL shall replace the first four bytes of the logical block received from the data-out buffer application client with the least significant four bytes of the LBA of the logical block being written to the media, ending with the least significant byte (e.g., if the LBA is 77665544_33221100h, 33221100h is written with 33h written first and 00h written last). The SATL shall use ATA write commands as defined in 9.17.2. S (see SBC-2).
0	1	The SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
1	1	

9.27 WRITE SAME (16) command

The WRITE SAME (16) command (see table 63) requests that the SATL transfer a single logical block from the ~~data-out-buffer~~ [application client](#) and write the contents of that single logical block, with modifications based on the LBDATA bit and the PBDATA bit, to the specified range of logical block addresses [on the ATA device](#).

Table 63 — WRITE SAME (16) ~~command~~ CDB fields [translations](#)

Field	Description or reference
OPERATION CODE	If the ATA device supports SCT LBA Segment Access (see SCT) the ATA SCT Write Same command (see ATA8-ACS) , the SATL should issue SCT LBA Segment Access the ATA SCT Write Same command to repeatedly write the data block transferred from the application client in the buffer to the ATA device . If the ATA device does not implement SCT the ATA SCT Write Same command then the SATL shall issue ATA write commands as defined in 9.17.1
WRPROTECT	Unspecified (see 3.4.3)
LBDATA	9.26.2
PBDATA	9.26.2
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.11), as defined by 9.1 . If the SATL implements direct block mapping (see 3.1.x), then the SATL shall set the Start field in the ATA SCT Write Same command or the ATA LBA in the ATA write command equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.3).
GROUP NUMBER	Unspecified (see 3.4.3)
NUMBER OF BLOCKS	A NUMBER OF BLOCKS of zero indicates that the data-out-buffer SATL shall be repeatedly written from the specified logical block address through the last user-addressable logical sector on the media. A NUMBER OF BLOCKS field set to zero specifies that the SATL shall repeatedly write the logical block transferred from the application client to the range of ATA logical sectors corresponding to the range of logical block addresses specified by the LOGICAL BLOCK ADDRESS field through the LBA of the last logical block on the logical unit. If the NUMBER OF BLOCKS field is set to a value other than not zero , the SATL shall repeatedly write the data-out-buffer data block transferred from the application client to the medium of the ATA device for the number of logical blocks sectors specified to the corresponding logical sectors on the ATA device . The SATL shall send as many ATA commands as required to satisfy the number of logical blocks specified by the WRITE SAME (16) command..
CONTROL	6.4