

To: T10 Technical Committee
 From: Rob Elliott, HP (elliott@hp.com)
 Date: 11 November 2006
 Subject: 06-282r4 SAM-4 SPC-4 WRITE BUFFER clarifications

Revision history

Revision 0 (15 June 2006) First revision
 Revision 1 (6 September 2006) Incorporated comments from July 2006 CAP WG - added a download microcode model section, added to SAM-4 that a microcode change causes a hard reset condition
 Revision 2 (15 September 2006) Incorporated comments from September 2006 CAP WG.
 Revision 3 (24 October 2006) Incorporated comments from George Penokie (IBM).
 Revision 4 (11 November 2006) Incorporated comments from November 2006 CAP WG.

Related documents

spc4r05 - SCSI Primary Commands - 4 (SPC-4) revision 5
 sam4r04 - SCSI Architecture Model - 4 (SAM-4) revision 4

Overview

The WRITE BUFFER descriptions for the download and save modes (05h and 07h) are silent about when microcode actually starts being used - does it take effect when the WRITE BUFFER command is complete, or does it wait until the next hard reset?

There appear to be designs with different implementations in the field, so an application client should not assume any particular behavior.

This proposal adds an explicit statement that the new microcode "may or may not be activated after the WRITE BUFFER command completes" so applications know not to assume anything.

A variety of editorial changes are also included to normalize the descriptions of all the different download microcode modes:

- a) Delete the detailed BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field descriptions for mode 07h and 0Eh; just point to the mode 06h descriptions instead. Currently they are identical except for "send commands that conform to" vs. "shall conform to" in the BUFFER ID field description.
- b) Delete "until it is supplanted by..." wording following each mention of "effective after each hard reset." The current text in modes 5h and 7h doesn't mention the new modes Eh/Fh, so is incomplete. Rather than reformat the sentence into an a)b)c) list to add "download microcode with offsets and defer activate operation after the deferred downloaded code has replaced the current operational code" (as was done in modes Eh and Fh), just drop this phrase altogether from all the modes. It should be obvious that another download microcode overrides the previous one, just like writing to an LBA overwrites the data that was previously written.
- c) Change "[vendor-specific] microcode or control information" to just "microcode" everywhere.
- d) Change "code" to "microcode" in the Eh/Fh descriptions.
- e) Use the "activate" term in all the modes, replacing text like "transfer to the control memory space of the logical unit" or "replace operational code."
- f) In the introduction, merge all 3 downloading types into one. The offset types are not described separately (it is not a list of all the specific modes), so a more generic description suffices.
- g) Change "logical unit" to "device server" in a few places
- h) Change "download microcode with offsets and defer activation" to "download microcode with offsets and save and defer activate" to match wording in other modes
- i) Add hard reset into the list of events that cause deferred microcode to be activated
- j) Add microcode changes to the SAM-4 list of events that cause hard reset conditions
- k) Created a model section introducing the download microcode modes, comparing their options, and including some common rules. Revision 3 moved much of the material from the mode descriptions into the model itself.
- l) Mention download microcode functionality in READ BUFFER. The READ BUFFER Descriptor mode (03h) returns the offset boundaries that apply to the WRITE BUFFER download microcode with offsets modes.

In revision 3, removed this text from the description of mode 0Eh:

"All deferred microcode (i.e., microcode that has not been activated) shall be discarded if a WRITE BUFFER command is received with a:

- a) Download microcode and activate mode (04h);
- b) Download microcode, save, and activate mode (05h);
- c) Download microcode with offsets and activate mode (06h);
- d) Download microcode with offsets, save, and activate mode (07h); or
- e) Download microcode with offsets, save, and defer activation mode (0Eh) after the verification of the complete set of downloaded microcode is complete."

Assume microcode A was previously saved. If a complete microcode B is delivered (and saved) before a hard reset/power on occurs, image B will be used. That implies B becomes "the" normal saved image. However, if another microcode C starts to be delivered after B but is not complete and saved before a hard reset/power on occurs, this text was requiring the device to discard B and go back to A. I don't think such double buffering was intended by the deferred microcode proposal. The original wording was "All deferred microcode (i.e. microcode or control information that has not replaced operational code) shall be discarded if...". Interpreting "has not replaced operational code" as "has not been saved" makes more sense. Replaced the text with this in the model section (as of revision 5):

"If new microcode is saved before deferred microcode is activated, the deferred microcode is not activated and the new saved microcode is considered deferred."

Suggested changes to SAM-4

5.3.3 Status precedence

If a device server detects that more than one of the following conditions applies to a completed task, it shall select the condition to report based on the following precedence:

- 1) An ACA ACTIVE status;
 - 2) A CHECK CONDITION status for any of the following unit attention conditions (i.e., with a sense key set to UNIT ATTENTION and one of the following additional sense codes):
 - A) POWER ON, RESET, OR BUS DEVICE RESET OCCURRED;
 - B) POWER ON OCCURRED;
 - C) SCSI BUS RESET OCCURRED;
 - D) MICROCODE HAS BEEN CHANGED;
 - E) BUS DEVICE RESET FUNCTION OCCURRED;
 - F) DEVICE INTERNAL RESET; or
 - G) I_T NEXUS LOSS OCCURRED;
 - 3) A RESERVATION CONFLICT status;
- and
- 4) A status of:
 - A) CHECK CONDITION, for any reason not listed in 2);
 - B) GOOD;
 - C) CONDITION MET; or
 - D) TASK ABORTED.

5.5 Task and command lifetimes

This subclause specifies the events delimiting the beginning and end (i.e., lifetime) of a task or pending command from the viewpoint of the device server and application client.

The device server shall create a task upon receiving a SCSI Command Received indication.

The task shall exist until:

- a) The device server sends a SCSI transport protocol service response for the task of TASK COMPLETE; or
- b) The task is aborted as described in 5.6.

The application client maintains an application client task to interact with the task from the time the Send SCSI Command SCSI transport protocol service request is invoked until it receives one of the following SCSI target device responses:

- a) A service response of TASK COMPLETE for that task;
- b) Notification of a unit attention condition with one of the following additional sense codes;
 - A) Any additional sense code whose ADDITIONAL SENSE CODE field contains 2Fh (e.g., COMMANDS CLEARED BY ANOTHER INITIATOR, or COMMANDS CLEARED BY POWER LOSS NOTIFICATION), if in reference to the task set containing the task; **or**
 - B) Any additional sense code whose ADDITIONAL SENSE CODE field contains 29h (e.g., POWER ON, RESET, OR BUS DEVICE RESET OCCURRED; POWER ON OCCURRED; SCSI BUS RESET OCCURRED; BUS DEVICE RESET FUNCTION OCCURRED; DEVICE INTERNAL RESET; or I_T NEXUS LOSS OCCURRED); or
 - C) MICROCODE HAS BEEN CHANGED;**
- c) Notification that the task manager has detected the use of a duplicate I_T_L_Q nexus (see 5.8.3);
- d) A service response of FUNCTION COMPLETE following an ABORT TASK task management function directed to the specified task;
- e) A service response of FUNCTION COMPLETE following an ABORT TASK SET or a CLEAR TASK SET task management function directed to the task set containing that task;
- f) A service response of FUNCTION COMPLETE following an I_T NEXUS RESET task management function delivered on the I_T nexus used to deliver that task; or
- g) A service response of FUNCTION COMPLETE in response to a LOGICAL UNIT RESET task management function directed to the logical unit.

6 SCSI events and event notification model

6.1 SCSI events overview

SCSI events may occur or be detected in either:

- a) The SCSI device;
- b) One or more SCSI ports within a SCSI device; or
- c) The application client, task manager, or device server.

The detection of any event may require processing by the object that detects it.

Events that occur in the SCSI device are assumed to be detected and processed by all objects within the SCSI device.

When a SCSI port detects an event, it shall use the event notification services (see 6.4) to notify SCSI application layer objects that the event has been detected.

The events detected and event notification services usage depends on whether the SCSI device is a SCSI target device (see figure 35) or a SCSI initiator device (see figure 36). SCSI target/initiator devices shall use the event notification services defined for both SCSI target devices and SCSI initiator devices.

Editor's Note 1: In figure 35 (Events and event notifications for SCSI target devices), add a yellow box "Microcode change" with a dashed black line feeding into the Hard Reset box

6.2 Establishing a unit attention condition subsequent to detection of an event

Table 32 shows the additional sense code that a logical unit shall use when a unit attention (see 5.8.7) is established for each of the conditions shown in figure 35 (see 6.1). A SCSI transport protocol may define a more specific additional sense code than SCSI BUS RESET OCCURRED for reset events. The most specific

condition in table 32 known to the logical unit should be used to establish the additional sense code for a unit attention.

Table 32 — Unit attention sense codes for events detected by SCSI target devices

Condition	Additional sense code	Specificity
Logical unit is unable to distinguish between the conditions	POWER ON, RESET, OR BUS DEVICE RESET OCCURRED	Lowest
Power on	POWER ON OCCURRED or DEVICE INTERNAL RESET	
Hard reset	SCSI BUS RESET OCCURRED or MICROCODE HAS BEEN CHANGED or protocol specific	
Logical unit reset	BUS DEVICE RESET FUNCTION OCCURRED	
I_T nexus loss	I_T NEXUS LOSS OCCURRED	Highest

A logical unit may use the I_T NEXUS LOSS OCCURRED additional sense code when establishing a unit attention condition if:

- The SCSI initiator port to which the sense data is being delivered is the SCSI initiator port that was associated with the I_T nexus loss, and the logical unit has maintained all state information specific to that SCSI initiator port since the I_T nexus loss; and
- The I_T nexus being used to deliver the sense data is the same I_T nexus that was lost, and the logical unit has maintained all state information specific to that I_T nexus since the I_T nexus loss.

Otherwise, the logical unit shall use one of the less specific additional sense codes (e.g., POWER ON OCCURRED) when establishing a unit attention condition.

6.3 Conditions resulting from SCSI events

6.3.1 Power on

Power on is a SCSI device condition resulting from a power on event. When a SCSI device is powered on, it shall cause a hard reset.

The power on condition applies to both SCSI initiator devices and SCSI target devices.

6.3.2 Hard reset

Hard reset is a SCSI device condition resulting from:

- A power on condition (see 6.3.1);
- [Microcode change \(see SPC-4\)](#); or
- A reset event indicated by a Transport Reset event notification (see 6.4).

The definition of reset events and the notification of their detection is SCSI transport protocol specific.

Each SCSI transport protocol standard that defines reset events shall specify a SCSI target port's protocol specific actions in response to reset events. Each SCSI transport protocol standard that defines reset events should specify when those events result in the delivery of a Transport Reset event notification to the SCSI applications layer.

SCSI transport protocols may include reset events that have no SCSI effects (e.g., a Fibre Channel non-initializing loop initialization primitive).

The hard reset condition applies to both SCSI initiator devices and SCSI target devices.

A SCSI target port's response to a hard reset condition shall include a logical unit reset condition (see 6.3.3) for all logical units to which the SCSI target port has access. A hard reset condition shall not affect any other

SCSI target ports in the SCSI target device, however, the logical unit reset condition established by a hard reset may affect tasks that are communicating via other SCSI target ports.

Although the task manager response to task management requests is subject to the presence of access restrictions, as managed by ACCESS CONTROL OUT commands (see SPC-3), a hard reset condition shall not be prevented by access controls.

When a SCSI initiator port detects a hard reset condition, it should terminate all its outstanding Execute Command procedure calls with a service response of SERVICE DELIVERY OR TARGET FAILURE. A hard reset condition shall not affect any other SCSI initiator ports in the SCSI initiator device, however, the logical unit reset condition established in a SCSI target device by a hard reset may affect tasks that are communicating via other SCSI initiator ports.

A SCSI port's response to a hard reset condition shall include establishing an I_T nexus loss condition (see 6.3.4) for every I_T nexus associated with that SCSI port.

7.9 Task management function lifetime

The task manager shall create a task management function upon receiving a Task Management Request Received indication (see 7.10). The task management function shall exist until:

- a) the task manager sends a SCSI transport protocol service response for the task management function;
- b) an I_T nexus loss (see 6.3.4);
- c) a logical unit reset (see 6.3.3);
- d) a hard reset (see 6.3.2); or
- e) a power on condition (see 6.3.1).

The application client maintains an application client task to interact with the task management function from the time the Send Task Management Request SCSI transport protocol service request is invoked until it receives one of the following SCSI target device responses:

- a) A service response of FUNCTION COMPLETE, FUNCTION SUCCEEDED, FUNCTION REJECTED, or SERVICE DELIVERY OR TARGET FAILURE is received for that task management function;
- b) Notification of a unit attention condition with any additional sense code whose ADDITIONAL SENSE CODE field contains 29h (e.g., POWER ON, RESET, OR BUS DEVICE RESET OCCURRED; POWER ON OCCURRED; SCSI BUS RESET OCCURRED; BUS DEVICE RESET FUNCTION OCCURRED; DEVICE INTERNAL RESET; or I_T NEXUS LOSS OCCURRED); ~~or~~
- c) [Notification of a unit attention condition with an additional sense code of MICROCODE HAS BEEN CHANGED; or](#)
- d) Notification of a unit attention condition with an additional sense code of COMMANDS CLEARED BY POWER LOSS NOTIFICATION.

Suggested changes to SPC-4

5.xx Downloading and activating microcode [all new]

SCSI target device implementations may use microcode (e.g., firmware) that is stored in nonvolatile storage. Microcode may be changeable by an application client using the WRITE BUFFER command (see 7.xx). The WRITE BUFFER command provides multiple methods for downloading microcode to the SCSI target device and activating the microcode.

Downloading and activating microcode involves the following steps:

- 1) **Download:** The application client transfers microcode from the Data-Out buffer to the device server in one or more WRITE BUFFER commands;
- 2) **Save:** After receiving the complete microcode, if defined by the download microcode mode, the device server saves the microcode to nonvolatile storage; and
- 3) **Activate:** After receiving the complete microcode and after saving it to nonvolatile storage if defined by the download microcode mode, the SCSI target device begins using the new microcode for the first time after an event defined by the download microcode mode.

After power on or hard reset, the SCSI target device shall use the last microcode that was saved to nonvolatile storage.

Table 33 defines the WRITE BUFFER download microcode modes.

Table 33 — WRITE BUFFER download microcode modes

Mode	Download ^a	Save ^d	Activate ^e
Download microcode and activate (04h)	yes ^b	no	yes
Download microcode, save, and activate (05h)	yes ^b	yes	optional
Download microcode with offsets and activate (06h)	yes ^c	no	yes
Download microcode with offsets, save, and activate (07h)	yes ^c	yes	optional
Download microcode with offsets, save, and defer activate (0Eh)	yes ^c	yes	no
Activate deferred microcode (0Fh)	no	no	yes

^a For modes labeled “yes” in the “Download” column, the application client delivers microcode in the WRITE BUFFER command(s). For modes labeled “no,” the application client does not deliver microcode with the WRITE BUFFER command.

^b The application client shall deliver the entire microcode in one WRITE BUFFER command. The device server shall perform any required verification of the microcode prior to returning GOOD status for the command.

^c the application client delivers microcode in multiple WRITE BUFFER commands, specifying a buffer ID and buffer offset in each command. If the device server does not receive the necessary WRITE BUFFER commands required to deliver the complete microcode before a logical unit reset occurs, an I_T nexus loss occurs, or a WRITE BUFFER command specifying a different download microcode mode is processed, the device server shall discard the new microcode. If the device server determines that it is processing the final WRITE BUFFER command (i.e., the WRITE BUFFER command delivering the last part of the microcode), it shall perform any required verification of the microcode prior to returning GOOD status for the command.

^d For modes labeled “yes” in the “Save” column, the device server shall save the microcode to nonvolatile storage for use after each subsequent power on or hard reset. For modes labeled “no,” the device server shall discard the microcode on the next power on or hard reset.

^e For modes labeled “yes” in the “Activate” column, the device server shall activate the microcode after completion of the final WRITE BUFFER command. For modes labeled “optional,” the device server may or may not activate the microcode image upon completion of the final WRITE BUFFER command. For modes labeled “no,” the device server shall not activate the microcode upon completion of the final WRITE BUFFER command.

When microcode is activated due to processing a WRITE BUFFER command with a mode that causes activation after processing, the device server shall establish a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus except the I_T nexus on which the WRITE BUFFER command was received with the additional sense code set to MICROCODE HAS BEEN CHANGED.

When microcode is activated due to power on or hard reset by a device server that queues unit attention conditions, the device server may establish a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus with the additional sense code set to MICROCODE HAS BEEN CHANGED queued behind the unit attention condition for the power on or hard reset.

When microcode is activated due to START STOP UNIT command or a FORMAT UNIT command that causes a unit attention condition by a device server that queues unit attention conditions, the device server may establish a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus with the additional sense code set to MICROCODE HAS BEEN CHANGED queued behind the unit attention condition for the power on or hard reset.

If new microcode is saved before deferred microcode is activated, the deferred microcode is not activated and the new saved microcode is considered deferred.

Table 34 defines how the WRITE BUFFER download microcode modes use the BUFFER ID field, the BUFFER OFFSET field, and the PARAMETER LIST LENGTH field in the WRITE BUFFER CDB.

Table 34 — WRITE BUFFER download microcode mode field usage

Mode	BUFFER ID field	BUFFER OFFSET field	PARAMETER LIST LENGTH field
Download microcode and activate (04h)	vendor specific	vendor specific	vendor specific
Download microcode, save, and activate (05h)			
Download microcode with offsets and activate (06h)	see 6.36.7	see 6.36.7	see 6.36.7
Download microcode with offsets, save, and activate (07h)			
Download microcode with offsets, save, and defer activate (0Eh)			
Activate deferred microcode (0Fh)	ignored	ignored	ignored

If the device server is unable to process a WRITE BUFFER command with a download microcode mode because of a vendor specific condition, it shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.

[\[End of all-new section: change marks resume\]](#)

6.14 READ BUFFER command

6.14.1 READ BUFFER command introduction

The READ BUFFER command (see table 127) is used in conjunction with the WRITE BUFFER command as a ~~diagnostic~~ function for ~~testing memory in the SCSI device and the integrity of a service delivery subsystem;~~

- a) [Testing logical unit buffer memory;](#)
- b) [Testing the integrity of the service delivery subsystem; and](#)
- c) [Downloading microcode \(see 5.xx\).](#)

This command shall not alter the medium.

...

6.35 WRITE BUFFER command

6.36.1 WRITE BUFFER command introduction

The WRITE BUFFER command (see table 195) is used in conjunction with the READ BUFFER command as a ~~diagnostic~~ function for ~~testing logical unit memory in the SCSI target device and the integrity of the service-delivery subsystem. Additional modes are provided for:~~

- a) [Testing logical unit buffer memory;](#)
- b) [Testing the integrity of the service delivery subsystem;](#)
- c) Downloading microcode [\(see 5.xx\); and](#)
- d) ~~Downloading and saving microcode;~~
- e) ~~Downloading microcode with deferred activation; and~~
- f) Downloading application logs (see 5.11).

Table 196 — WRITE BUFFER command

Byte/Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (3Bh)							
1	Reserved			MODE				
2	BUFFER ID							
3	(MSB) _____ (LSB)							
5	BUFFER OFFSET							
6	(MSB) _____ (LSB)							
8	PARAMETER LIST LENGTH							
9	CONTROL							

This command shall not alter any medium of the logical unit when the data mode or the combined header and data mode is specified.

The function of this command and the meaning of fields within the CDB depend on the contents of the MODE field. The MODE field is defined in table 197.

Table 197 — WRITE BUFFER MODE field

MODE Code	Description	Reference
00h	Combined header and data ^a	6.36.2
01h	Vendor specific ^a	
02h	Data	6.36.3
03h	Reserved	
04h	Download microcode <u>and activate</u>	<u>5.xx and</u> 6.36.4
05h	Download microcode and , <u>save, and activate</u>	<u>5.xx and</u> 6.36.5
06h	Download microcode with offsets <u>and activate</u> ^b	<u>5.xx and</u> 6.36.6
07h	Download microcode with offsets and , <u>save and activate</u> ^b	<u>5.xx and</u> 6.36.7
08h - 09h	Reserved	
0Ah	Echo buffer	6.36.8
0Bh - 0Dh	Reserved	
0Eh	Download microcode with offsets, <u>save,</u> and defer activation <u>activate</u> ^b	<u>5.xx and</u> 6.36.9
0Fh	Activate deferred microcode	<u>5.xx and</u> 6.36.10
1Ah	Enable expander communications protocol and Echo buffer	6.36.11
1Bh	Disable expander communications protocol	6.36.12
1Ch	Download application log	<u>5.11 and</u> 6.36.13
1Dh - 1Fh	Reserved	
^a Modes 00h and 01h are not recommended ^b When download microcode with buffer offsets, the WRITE BUFFER command mode should be 06h, 07h, or 0Eh.		

6.36.2 Combined header and data mode (00h)

In this mode, data to be transferred is preceded by a four-byte header. The four-byte header consists of all reserved bytes.

The BUFFER ID and the BUFFER OFFSET fields shall be [set to zero](#).

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer. This number includes four bytes of header, so the data length to be stored in the device server's buffer is parameter list length minus four. The application client should attempt to ensure that the parameter list length is not greater than four plus the BUFFER CAPACITY field value (see 6.14.2) that is returned in the header of the READ BUFFER command [combined header and data mode \(mode 0h\)](#). If the parameter list length exceeds the buffer capacity, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

6.36.3 Vendor specific mode (01h)

In this mode, the meaning of the BUFFER ID, BUFFER OFFSET, and PARAMETER LIST LENGTH fields are not specified by this standard.

6.36.4 Data mode (02h)

In this mode, the Data-Out Buffer contains buffer data destined for the logical unit.

The BUFFER ID field identifies a specific buffer within the logical unit. The vendor assigns buffer ID codes to buffers within the logical unit. Buffer ID zero shall be supported. If more than one buffer is supported, then additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is selected, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

~~The BUFFER OFFSET field specifies the location in the buffer to which the data is written. Data are written to the logical unit buffer starting at the location specified by the BUFFER OFFSET field.~~ The application client should conform to the offset boundary requirements returned in the READ BUFFER descriptor ([see 6.14.5](#)). If the device server is unable to process the specified buffer offset, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should attempt to ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. The capacity of the buffer is indicated by the BUFFER CAPACITY field in the READ BUFFER descriptor (see 6.14.5). If the BUFFER OFFSET and PARAMETER LIST LENGTH fields specify a transfer in excess of the buffer capacity, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

6.36.5 Download microcode [and activate](#) mode (04h)

In this mode, ~~vendor-specific microcode or control information~~ shall be [transferred to the device server and activated \(see 5.xx\)](#). ~~After activation, a hard reset, the device operation shall revert to a vendor-specific condition the previous microcode upon a power on or hard reset.~~

~~The meanings of the BUFFER ID, BUFFER OFFSET, and PARAMETER LIST LENGTH fields are not specified by this standard and are not required to be zero filled set to zero.~~

[The BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field are vendor-specific.](#)

~~When the microcode download has completed successfully downloaded microcode is activated, the device server shall establish a unit attention condition (see SAM-3) for the initiator port associated with every I_T nexus except the I_T nexus on which the WRITE BUFFER command was received, with the additional sense code set to MICROCODE HAS BEEN CHANGED.~~

~~If the logical unit is unable to process this command because of some device condition, each WRITE BUFFER command with this mode (04h) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.~~

6.36.6 Download microcode ~~and~~, save, and activate mode (05h)

In this mode, ~~vendor specific microcode or control information~~ shall be transferred to the logical unit device server and, ~~if the WRITE BUFFER command is completed successfully, also shall be saved in a~~ nonvolatile memory space (e.g., semiconductor, disk, or other) storage (see 5.xx).

The downloaded microcode may or may not be activated after the WRITE BUFFER command completes and shall then be effective ~~activated~~ when one of the following occurs:

- a) power on; or
- b) ~~after each~~ hard reset,-

~~until it is supplanted in another download microcode and save operation or download microcode with offsets and save operation.~~

~~The meanings of the BUFFER ID, BUFFER OFFSET, and PARAMETER LIST LENGTH fields are not specified by this standard and are not required to be zero filled set to zero.~~

The BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field are vendor-specific.

~~When the download microcode and save command has completed successfully downloaded microcode is activated, the device server shall establish a unit attention condition (see SAM-3) for the initiator port associated with every I_T nexus except the I_T nexus on which the WRITE BUFFER command was received with the additional sense code set to MICROCODE HAS BEEN CHANGED.~~

~~If the logical unit is unable to process this command because of some device condition, each WRITE BUFFER command with this mode (05h) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.~~

6.36.7 Download microcode with offsets and activate mode (06h)

In this mode, microcode shall be transferred to the device server using one or more WRITE BUFFER commands and activated (see 5.xx).

~~The the application client may split the transfer of the vendor specific microcode or control information over two or more WRITE BUFFER commands. If the last WRITE BUFFER command of a set of one or more commands completes successfully, then the microcode or control information shall be activated. After activation, a hard reset, the device operation shall revert to a vendor specific condition the previous microcode upon a power on or hard reset. In this mode, the Data Out Buffer contains vendor specific, self-describing microcode or control information.~~

~~Since the downloaded microcode or control information may be sent using several commands, when the logical unit device server detects that the last WRITE BUFFER command with this mode the last download microcode with offsets WRITE BUFFER command has been received, the device server it shall perform any logical unit required verification of the complete set of downloaded microcode or control information prior to returning GOOD status for the last command. After the last command completes successfully When the downloaded microcode is activated the device server shall establish a unit attention condition (see SAM-3) for the initiator port associated with every I_T nexus except the I_T nexus on which the set of WRITE BUFFER commands was received, with the additional sense code set to MICROCODE HAS BEEN CHANGED.~~

~~If the complete set of WRITE BUFFER commands required to effect a microcode or control information change (i.e., one or more commands) are not received before a logical unit reset or I_T nexus loss occurs, the change shall not be effective and the new microcode or control information shall be discarded.~~

The BUFFER ID field specifies a buffer within the logical unit. The vendor assigns buffer ID codes to buffers within the logical unit. A buffer ID value of zero shall be supported. If more than one buffer is supported, then additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is specified, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The BUFFER OFFSET field specifies the location in the buffer to which the ~~The~~ microcode or control information are ~~is~~ written to the logical unit buffer starting at the location specified by the BUFFER OFFSET field. The application client shall send commands that conform to the offset boundary requirements returned in the READ BUFFER descriptor (see 6.14.5). If the device server is unable to process the specified buffer offset,

the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be present in the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. The capacity of the buffer is indicated by the BUFFER CAPACITY field in the READ BUFFER descriptor (see 6.14.5). If the BUFFER OFFSET and PARAMETER LIST LENGTH fields specify a transfer in excess of the buffer capacity, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

~~If the logical unit is unable to process this command because of some device condition, each WRITE BUFFER command with this mode (06h) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.~~

6.36.8 Download microcode with offsets ~~and~~, save, and activate mode (07h)

In this mode, microcode shall be transferred to the device server using one or more WRITE BUFFER commands and saved to nonvolatile storage (see 5.xx).

~~The the application client may split the transfer of the vendor specific microcode or control information over two or more WRITE BUFFER commands. If the last WRITE BUFFER command of a set of one or more commands completes successfully, then the microcode or control information shall be saved in a nonvolatile memory space (e.g., semiconductor, disk, or other) storage.~~

The ~~saved~~ downloaded microcode ~~or control information~~ may or may not be activated after the WRITE BUFFER command completes and shall ~~then~~ be ~~effective~~ activated when one of the following occurs:

- a) power on; or
- b) ~~after each~~ hard reset,-

~~until it is supplanted by another download microcode with save operation or download microcode with offsets and save operation. In this mode, the Data-Out Buffer contains vendor specific, self-describing microcode or control information.~~

~~Since the downloaded microcode or control information may be sent using several commands, when the logical unit/device server detects that the last WRITE BUFFER command with this mode the last download microcode with offsets and save mode WRITE BUFFER command has been received, the device server it shall perform any logical unit required verification of the complete set of downloaded microcode or control information prior to returning GOOD status for the last command. After the last command completes successfully When the downloaded microcode is activated, the device server shall establish a unit attention condition (see SAM-3) for the initiator port associated with every I_T nexus except the I_T nexus on which the set of WRITE BUFFER commands was received, with the additional sense code set to MICROCODE HAS BEEN CHANGED.~~

~~If the complete set of WRITE BUFFER commands required to effect a microcode or control information change (i.e., one or more commands) are not received before a logical unit reset or I_T nexus loss occurs, the change shall not be effective and the new microcode or control information shall be discarded.~~

~~The BUFFER ID field specifies a buffer within the logical unit. The vendor assigns buffer ID codes to buffers within the logical unit. A buffer ID value of zero shall be supported. If more than one buffer is supported, then additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is specified, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.~~

~~The microcode or control information are written to the logical unit buffer starting at the location specified by the BUFFER OFFSET field. The application client shall conform to the offset boundary requirements. If the device server is unable to process the specified buffer offset, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.~~

~~The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be present in the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should~~

~~ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. The capacity of the buffer is indicated by the BUFFER CAPACITY field in the READ BUFFER descriptor (see 6.14.5). If the BUFFER OFFSET and PARAMETER LIST LENGTH fields specify a transfer in excess of the buffer capacity, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.~~

The BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field are defined in the download microcode with offsets mode (06h)(see 6.36.7).

~~If the logical unit is unable to process this command because of some device condition, each WRITE BUFFER command with this mode (07h) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.~~

6.36.9 Write data to echo buffer mode (0Ah)

In this mode the device server transfers data from the application client and stores it in an echo buffer. An echo buffer is assigned in the same manner by the device server as it would for a write operation. Data shall be sent aligned on four-byte boundaries.

The BUFFER ID and BUFFER OFFSET fields ~~are~~shall be ignored in this mode.

NOTE 32 - It is recommended that the logical unit assign echo buffers on a per I_T nexus basis to limit the number of exception conditions that may occur when I_T nexuses are present.

Upon successful completion of a WRITE BUFFER command the data shall be preserved in the echo buffer unless there is an intervening command to any logical unit in which case the data may be changed.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer to be stored in the echo buffer. The application client should ensure that the parameter list length does not exceed the capacity of the echo buffer. The capacity of the echo buffer is indicated by the BUFFER CAPACITY field in the READ BUFFER echo buffer descriptor (see 6.14.7). If the PARAMETER LIST LENGTH field specifies a transfer in excess of the buffer capacity, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

6.36.10 Download microcode with offsets, save, and defer ~~activation~~activate mode (0Eh)

In this mode, microcode shall be transferred to the device server using one or more WRITE BUFFER commands, saved to nonvolatile storage (see 5.xx), and considered deferred.

~~The the application client may split the transfer of the vendor specific microcode or control information over two or more WRITE BUFFER commands and activation of microcode is deferred. If the last WRITE BUFFER command of a set of one or more commands completes successfully, then the microcode or control information shall be saved in to nonvolatile memory space (e.g., semiconductor, disk, or other)storage that is not associated with the current operational code and shall not be activated.~~

~~Since the downloaded microcode or control information may be sent using several commands, when the logical unit/device server detects that the last WRITE BUFFER command with this mode the last download microcode with offsets and save and defer activation mode WRITE BUFFER command has been received, the device server it shall perform any logical unit required verification of the complete set of downloaded microcode or control information prior to returning GOOD status for the last command.~~

The deferred ~~downloaded~~ microcode shall be activated and no longer considered deferred ~~replace the current operational code~~ when one of the following occurs:

- a) A power on;
- b) A hard reset;
- c) A START STOP UNIT command is processed (see SBC-3);
- d) A FORMAT UNIT command is processed (see SBC-3); or
- e) A WRITE BUFFER command with the activate deferred microcode mode (0Fh) is processed (see 6.36.11).

~~If the event that caused the deferred microcode to replace the current operational code be activated establishes a unit attention condition, then that unit attention condition shall be established as defined in the~~

applicable command standard (see 3.1.17) and SAM-4. If the event that caused the deferred microcode to replace the current operational code be activated does not establish a unit attention condition or the device server queues unit attention conditions, then after the deferred downloaded microcode replaces the current operational code is activated, the device server shall establish a unit attention condition for the initiator port associated with every I_T nexus with the additional sense code set to MICROCODE HAS BEEN CHANGED.

If the complete set of WRITE BUFFER commands required to effect a microcode or control information change (i.e., one or more commands) are not received before a logical unit reset or I_T nexus loss occurs, the save shall not be effective and the new microcode or control information shall be discarded.

All deferred microcode or control information (i.e., microcode or control information that has not replaced operational code been activated) shall be discarded if a WRITE BUFFER command is received with a:

- a) Download microcode and activate mode (04h);
- b) Download microcode and save, and activate mode (05h);
- c) Download microcode with offsets and activate mode (06h);
- d) Download microcode with offsets and save, and activate mode (07h); or
- e) Download microcode with offsets and save, and defer activation mode (0Eh) after the verification of the complete set of downloaded microcode or control information is complete.

After the deferred downloaded microcode replaces the current operational code is activated, the downloaded microcode or control information shall be effective after each hard reset until it is supplanted in another:

- a) Download microcode and save operation;
- b) Download microcode with offsets and save operation; or
- c) Download microcode with offsets and defer activation operation after the deferred downloaded code has replaced the current operational code.

The BUFFER ID field specifies a buffer within the logical unit. The vendor assigns buffer ID codes to buffers within the logical unit. A buffer ID value of zero shall be supported. If more than one buffer is supported, then additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is specified, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The microcode or control information are written to the logical unit buffer starting at the location specified by the BUFFER OFFSET field. The application client shall conform to the offset boundary requirements. If the device server is unable to process the specified buffer offset, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be present in the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. The capacity of the buffer is indicated by the BUFFER CAPACITY field in the READ BUFFER descriptor (see 6.14.5). If the BUFFER OFFSET field and PARAMETER LIST LENGTH field specify a transfer in excess of the buffer capacity, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

[The BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field are defined in the download microcode with offsets mode \(06h\)\(see 6.36.7\).](#)

If the logical unit is unable to process this command because of some device condition, every WRITE BUFFER command with this mode (0Eh) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.

6.36.11 Activate deferred microcode mode (0Fh) [\[with changes shown\]](#)

In this mode, deferred microcode or control information that has been processed in one or more download microcode with offsets, save, and defer activation mode WRITE BUFFER commands [that has been saved using the download microcode with offsets, save, and defer activate mode \(0Eh\)\(see 6.36.10\)](#), if any, shall replace the current operational code be activated (see 5.xx) and shall no longer be considered deferred. The microcode or control information shall then be effective activated after each hard reset until it is supplanted in

~~another download microcode and save operation or download microcode with offsets and save operation. Any deferred microcode or control information shall be discarded.~~

~~After the deferred microcode or control information replaces the current operational microcode or control information, the previously deferred (i.e., current operational) microcode or control information shall be effective after each hard reset until it is supplanted in another:~~

- ~~a) Download microcode and save operation;~~
- ~~b) Download microcode with offsets and save operation; or~~
- ~~c) Download microcode with offsets and defer activation operation after the deferred downloaded code has replaced the current operational code.~~

The BUFFER ID field, the BUFFER OFFSET field, and PARAMETER LIST LENGTH field shall be ignored in this mode.

~~When the download deferred microcode has successfully replaced the current operational code is activated, the device server shall establish a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus except the I_T nexus on which the set of WRITE BUFFER commands was received, with the additional sense code set to MICROCODE HAS BEEN CHANGED.~~

If ~~the logical unit is unable to process this command because~~ there is no deferred microcode ~~or control information, it that has been saved using the download microcode with offsets, save, and defer activate mode (0Eh), the WRITE BUFFER command~~ shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.

6.36.12 Enable expander communications protocol and Echo buffer mode (1Ah)

Receipt of a WRITE BUFFER command with this mode ~~(1Ah)~~ causes a communicative expander (see SPI-5) to enter the expanded communications protocol mode.

Device servers in SCSI target devices that receive a WRITE BUFFER command with this mode shall process it as if it were a WRITE BUFFER command with [the write data to Echo buffer mode \(0Ah\)](#) (see 6.36.9).

6.36.13 Disable expander communications protocol mode (1Bh)

Receipt of a WRITE BUFFER command with this mode ~~(1Bh)~~ causes a communicative expander (see SPI-5) to exit the expanded communications protocol mode and return to simple expander operation.

Device servers in SCSI target devices that receive a WRITE BUFFER command with this mode 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.

6.36.14 Download application log mode (1Ch)

In this mode the device server transfers data from the application client and stores it in an application log (see 5.11). The format of the application log data is as specified in table 198.

The BUFFER ID field and BUFFER OFFSET field ~~are~~[shall be](#) ignored in this mode.

Upon successful completion of a WRITE BUFFER command the data shall be appended to the application log.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer to be stored in the application log. If the PARAMETER LIST LENGTH field specifies a transfer that exceeds the application log's capacity, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

...

1 Download microcode mode descriptions with changes absorbed

Editor's Note 2: Due to the quantity of changes, it may be easier to just review the surviving text

6.36.5 Download microcode and activate mode (04h)

In this mode, microcode shall be transferred to the device server and activated (see 5.xx).

The BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field are vendor-specific.

6.36.6 Download microcode, save, and activate mode (05h)

In this mode, microcode shall be transferred to the device server and saved to nonvolatile storage (see 5.xx).

The downloaded microcode may or may not be activated after the WRITE BUFFER command completes and shall be activated when one of the following occurs:

- a) power on; or
- b) hard reset.

The BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field are vendor-specific.

6.36.7 Download microcode with offsets and activate mode (06h)

In this mode, microcode shall be transferred to the device server using one or more WRITE BUFFER commands and activated (see 5.xx).

The BUFFER ID field specifies a buffer within the logical unit. The vendor assigns buffer ID codes to buffers within the logical unit. A buffer ID value of zero shall be supported. If more than one buffer is supported, then additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is specified, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The BUFFER OFFSET field specifies the location in the buffer to which the microcode is written. The application client shall send commands that conform to the offset boundary requirements returned in the READ BUFFER descriptor (see 6.14.5). If the device server is unable to process the specified buffer offset, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be present in the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. The capacity of the buffer is indicated by the BUFFER CAPACITY field in the READ BUFFER descriptor (see 6.14.5). If the BUFFER OFFSET and PARAMETER LIST LENGTH fields specify a transfer in excess of the buffer capacity, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

6.36.8 Download microcode with offsets, save, and activate mode (07h)

In this mode, microcode shall be transferred to the device server using one or more WRITE BUFFER commands and saved to nonvolatile storage (see 5.xx).

The downloaded microcode may or may not be activated after the WRITE BUFFER command completes and shall be activated when one of the following occurs:

- a) power on; or
- b) hard reset.

The BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field are defined in the download microcode with offsets mode (06h)(see 6.36.7).

6.36.10 Download microcode with offsets, save, and defer activate mode (0Eh)

In this mode, microcode shall be transferred to the device server using one or more WRITE BUFFER commands, saved to nonvolatile storage (see 5.xx), and considered deferred.

The deferred microcode shall be activated and no longer considered deferred when one of the following occurs:

- a) A power on;
- b) A hard reset;
- c) A START STOP UNIT command is processed (see SBC-3);
- d) A FORMAT UNIT command is processed (see SBC-3); or
- e) A WRITE BUFFER command with the activate deferred microcode mode (0Fh) is processed (see 6.36.11).

The BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field are defined in the download microcode with offsets mode (06h)(see 6.36.7).

6.36.11 Activate deferred microcode mode (0Fh)

In this mode, deferred microcode that has been saved using the download microcode with offsets, save, and defer activate mode (0Eh)(see 6.36.10), if any, shall be activated (see 5.xx) and shall no longer be considered deferred.

The BUFFER ID field, the BUFFER OFFSET field, and PARAMETER LIST LENGTH field shall be ignored in this mode.

If there is no deferred microcode that has been saved using the download microcode with offsets, save, and defer activate mode (0Eh), the WRITE BUFFER command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.