Accredited Standards Committee*
National Committee for Information Technology Standards (NCITS)

Doc. No.:	T10/98-184r1
Date:	September 8, 1998
Reply to:	ljllamers@ieee.org

To: T10 Membership
From: Lawrence J. Lamers
Subject: Extended Modes for READ/WRITE BUFFER

BACKGROUND

This proposal outlines a basic host initiated validation for a SCSI domain. The host is the most likely device in the domain to have information regarding the attached peripherals and segment configuration. The method proposed is an extension to the READ/WRITE BUFFER commands that provides for logical buffer segment addressing.

The goal is to define a procedure whereby a host could interrogate a device and determine its ability to communicate with that device at the negotiated data rate. Domain validation verifies that two devices on the SCSI bus can communicate at the negotiated speed and width. It is a means of assuring that bus configurations work and making the entire process more user friendly.

Domain validation provides no guarantee of data integrity, so while it may detect some instances of marginal physical plant it is just as likely not too. A legacy CRC implementation is needed in addition to packet protocol and dual transition CRC to guarantee data integrity.

The concept of a logical buffer is introduced by an extension to the Mode field. The existing commands imply the use of physical buffers. A logical buffer is assigned by the target in a manner used when a write operation is requested.

The use of a logical buffer allows this process to take place at any point after power-on, during initialization or run periodically as system integrity check. This process is able to detect a legacy expander that does not operate with the dual transfer mode in SPI-3. A future extension for margining will allow checking and adapting to a variety of cable plant, terminators, and bus configurations.

The basic premise is that the host issues a WRITE BUFFER command with a known data pattern to the selected device; then issues a READ BUFFER command to retrieve that data from the target. The host can then examine the data pattern to determine its validity.

A simpler and quicker test of the system could be performed by just issuing the READ BUFFER command to each device. This allows for a quick check of communication paths. The power-on sequence would be:

1) BUS RESET
2) Select device
3) Issue INQUIRY command
4) Negotiate bus speed and width
5) Issue READ BUFFER command
6) If the command completes successfully, (i.e., no parity errors, CRC failures, or bus hangs) the bus speed and width are established.

*Operating under the procedures of The American National Standards Institute.

7) If the command fails the host performs another negotiation and retest.

The recommended data pattern is 128 bytes in length and transfers in less than 25 microseconds even at asynchronous speeds. The host provides a time-out to check for bus hangs due to REQ/ACK miscount.

The use of the domain validation is being considered for inclusion in DMTF. This effort would add the needed attributes to the component interface (CI) layer.

Support for margin testing has been added. This includes a redefinition of the buffer offset field when using logical and margin modes.

1. READ BUFFER command

The READ BUFFER command (see table 1) is used in conjunction with the WRITE BUFFER command as a diagnostic function for testing memory in the SCSI device and the integrity of the service delivery subsystem. This command shall not alter the medium.

Table 1: READ BUFFER command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (3Ch)							
1	Reserved				Mode			
2	Buffer ID							
3	(MSB)	Buffer offset						
4								
5								(LSB)
6	(MSB)	Allocation length						
7								
8								(LSB)
9	Control							

If reservations are active, they shall affect the execution of the READ BUFFER command as follows. A reservation conflict shall occur when a READ BUFFER command is received from an initiator other than the one holding a logical unit reservation. The READ BUFFER command shall not be affected by extent or element reservations.

The function of this command and the meaning of fields within the command descriptor block depend on the contents of the mode field.

The mode field is defined in table 1.

Table 2: READ BUFFER Mode Field.

Mode	Description	Type
0000b	Combined header and data	Optional
0001b	Vendor-specific	Vendor-specific
0010b	Data	Optional
0011b	Descriptor	Optional
0100b	Reserved	Reserved
0101b	Reserved	Reserved
0110b	Reserved	Reserved
0111b	Reserved	Reserved
1000b	Reserved	Reserved
1001b	Reserved	Reserved
1010b	Logical	128 Byte Data

Mode	Description	Type
1011b	Margin	
1100b	Reserved	Reserved
1101b	Reserved	Reserved
1110b	Reserved	Reserved
1111b	Reserved	Reserved
Mode	**Description**	**Type**

Table 3: Buffer Offset Alternate Definition

Bit Byte	7	6	5	4	3	2	1	0
3	Data Pattern Type							
4	Margin Adjustment							
5	Reserved							

The buffer offset field is redefined for the logical and margin modes.

The data pattern type field is defined in table 4.

Table 4: Data Pattern Type Field.

Value	Description	Type
00h	Reserved	00h
01h	Oscillating	AAh/55h
02h	Alternating	00h/FFh
03h	ISI	walking ones then zeros
04h	Counting	0,1,2,3,...3,2,1,0
05h to	Reserved	Reserved
0Fh	Reserved	Reserved

1.1 Combined header and data mode (000b)

In this mode, a four-byte header followed by data bytes is returned to the application client in the Data-In Buffer. The buffer ID and the buffer offset fields are reserved.

The four-byte READ BUFFER header (see table 5) is followed by data bytes from the buffer.

Table 5 — READ BUFFER header

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Buffer capacity							
3								

The buffer capacity field specifies the total number of data bytes available in the buffer. This number is not reduced to reflect the allocation length; nor is it reduced to reflect the actual number of bytes written using the WRITE BUFFER command. Following the READ BUFFER header, the device server shall transfer data from the buffer. The device server shall terminate filling the Data-In Buffer when allocation length bytes of header plus data have been transferred or when all available header and buffer data have been transferred to the application client, whichever is less.

1.2 Vendor-specific mode (001b)

In this mode, the meanings of the buffer ID, buffer offset, and allocation length fields are not specified by this standard.

1.3 Data mode (010b)

In this mode, the Data-In Buffer is filled only with logical unit buffer data. The buffer ID field identifies a specific buffer within the logical unit from which data shall be transferred. 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, additional buffer ID codes shall be assigned contiguously, beginning with one. Buffer ID code assignments for the READ BUFFER command shall be the same as for the WRITE BUFFER command. If an unsupported buffer ID code is selected, the device server shall return CHECK CONDITION status, shall set the sense key to ILLEGAL REQUEST, and set the additional sense code to INVALID FIELD IN CDB.

The device server shall terminate filling the Data-In Buffer when allocation length bytes have been transferred or when all the available data from the buffer has been transferred to the application client, whichever amount is less.

The buffer offset field contains the byte offset within the specified buffer from which data shall be transferred. The application client should conform to the offset boundary requirements returned in the READ BUFFER descriptor (see 1.4). If the device server is unable to accept the specified buffer offset, it shall return CHECK CONDITION status, shall set the sense key to ILLEGAL REQUEST, and set the additional sense code to INVALID FIELD IN CDB.

1.4 Descriptor mode (011b)

In this mode, a maximum of four bytes of READ BUFFER descriptor information is returned. The device server shall return the descriptor information for the buffer specified by the buffer ID (see the description of the buffer ID in 1.3). If there is no buffer associated with the specified buffer ID, the device server shall return all zeros in the READ BUFFER descriptor. The buffer offset field is reserved in this mode. The allocation length should be set to four or greater. The device server shall transfer the lesser of the allocation length or four bytes of READ BUFFER descriptor. The READ BUFFER descriptor is defined as shown in table 6.

The offset boundary field returns the boundary alignment within the selected buffer for subsequent WRITE BUFFER and READ BUFFER commands. The value contained in the offset boundary field shall be interpreted as a power of two.

Table 6 — READ BUFFER descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	Offset boundary							
1	Buffer capacity							
3								

The value contained in the buffer offset field of subsequent WRITE BUFFER and READ BUFFER commands should be a multiple of $2^{\text{offset boundary}}$ as shown in table 7.

Table 7 — Buffer offset boundary

Offset boundary	**$2^{\text{Offset boundary}}$**	**Buffer offsets**
0h	$2^0 = 1$	Byte boundaries
1h	$2^1 = 2$	Even-byte boundaries
2h	$2^2 = 4$	Four-byte boundaries
3h	$2^3 = 8$	Eight-byte boundaries
4h	$2^4 = 16$	16-byte boundaries
.	.	.
FFh	Not applicable	0 is the only supported buffer offset

The buffer capacity field shall return the size of the selected buffer in bytes.

NOTE 1 In a system employing multiple application clients, a buffer may be altered between the WRITE BUFFER and READ BUFFER commands by another application client. Buffer testing applications should insure that only a single application client is active. Use of reservations (to all logical units on the device) or linked commands may be helpful in avoiding buffer alteration between these two commands.

1.5 Read Data from logical buffer (1010b)

In this mode the data is read from the logical buffer assigned to domain validation. This buffer shall be the same buffer as used when the WRITE BUFFER command was issued. A maximum of 128 bytes of data are returned. The Buffer ID and Buffer Offset fields are ignored in this mode.

2. WRITE BUFFER command

The WRITE BUFFER command (see table 8) is used in conjunction with the READ BUFFER command as a diagnostic function for testing logical unit memory in the target SCSI device and the integrity of the service delivery subsystem. Additional modes are provided for downloading microcode and for downloading and saving microcode.

If reservations are active, they shall affect the execution of the WRITE BUFFER command as follows. A reservation conflict shall occur when a WRITE BUFFER command is received from an initiator other than the one holding a logical unit, extent, or element reservation.

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

Table 8: WRITE BUFFER command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (3Bh)							
1	Reserved (Margin Testing)				Mode			
2	Buffer ID							
3	(MSB)							
4		Buffer offset						
5								(LSB)
6	(MSB)							
7		Parameter list length						
8								(LSB)
9	Control							

The function of this command and the meaning of fields within the command descriptor block depend on the contents of the mode field.

The WRITE BUFFER Mode field is defined in table 9.

Table 9: WRITE BUFFER Mode field

Mode	Description	Implementation requirements
0000b	Write combined header and data	Optional
0001b	Vendor-specific	Vendor-specific
0010b	Write data	Optional
0011b	Reserved	Reserved
0100b	Download microcode	Optional
0101b	Download microcode and save	Optional
0110b	Download microcode with offsets	Optional
0111b	Download microcode with offsets and save	Optional
1000b	Reserved	Reserved
1001b	Reserved	Reserved
1010b	Write data to logical buffer	Required
1011b	Reserved	Reserved
1100b	Reserved	Reserved
1101b	Reserved	Reserved
1110b	Reserved	Reserved
1111b	Reserved	Reserved

NOTES
2 Modes 000b and 001b are not recommended.

3 When downloading microcode with buffer offsets, the WRITE BUFFER command mode should be 110b or 111b.

Table 10: Buffer Offset Alternate Definition

Bit Byte	7	6	5	4	3	2	1	0
3	Data Pattern Type							
4	Margin Adjustment							
5	Reserved							

The buffer offset field is redefined for the logical and margin modes.

The data pattern type field is defined in table 11.

Table 11: Data Pattern Type Field.

Value	Description	Type
00h	Reserved	00h
01h	Oscillating	AAh/55h
02h	Alternating	00h/FFh
03h	ISI	walking ones then zeros
04h	Counting	0,1,2,3,...3,2,1,0
05h to	Reserved	Reserved
0Fh	Reserved	Reserved

2.1 Combined header and data mode (000b)

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 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 (see 7.15.1) that is returned in the header of the READ BUFFER command (mode 00b). If the parameter list length exceeds the buffer capacity the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST.

2.2 Vendor-specific mode (001b)

In this mode, the meaning of the buffer ID, buffer offset, and parameter list length fields are not specified by this standard.

2.3 Data mode (010b)

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, additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is selected, the device server shall return

CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

Data are written to the logical unit buffer starting at the location specified by the buffer offset. The application client should conform to the offset boundary requirements returned in the READ BUFFER descriptor. If the device server is unable to accept the specified buffer offset, it shall return CHECK CONDITION status and it shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

The parameter list length 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 may be determined by the buffer capacity field in the READ BUFFER descriptor.) If the buffer offset and parameter list length fields specify a transfer in excess of the buffer capacity, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

2.4 Download microcode mode (100b)

If the logical unit cannot accept this command because of some device condition, the device server shall terminate each WRITE BUFFER command with this mode (100b) with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and shall set the additional sense code to COMMAND SEQUENCE ERROR.

In this mode, vendor-specific microcode or control information shall be transferred to the control memory space of the logical unit. After a power-cycle or reset, the device operation shall revert to a vendor-specific condition. 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. When the microcode download has completed successfully the device server shall generate a unit attention condition for all initiators except the one that issued the WRITE BUFFER command (see SAM). The additional sense code shall be set to MICROCODE HAS BEEN CHANGED.

2.5 Download microcode and save mode (101b)

If the logical unit cannot accept this command because of some device condition, the device server shall terminate each WRITE BUFFER command with this mode (101b) with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and shall set the additional sense code to COMMAND SEQUENCE ERROR.

In this mode, vendor-specific microcode or control information shall be transferred to the logical unit and, if the WRITE BUFFER command is completed successfully, also shall be saved in a non-volatile memory space (semiconductor, disk, or other). The downloaded code shall then be effective after each power-cycle and reset until it is supplanted in another download microcode 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. When the download microcode and save command has completed successfully the device server shall generate a unit attention condition (see SAM) for all initiators except the one that issued the WRITE BUFFER command. When reporting the unit attention condition, the device server shall set the additional sense code to MICROCODE HAS BEEN CHANGED.

2.6 Download microcode with offsets (110b)

In this mode, the application client may split the transfer of the vendor-specific microcode or control information over two or more WRITE BUFFER commands. If the logical unit cannot accept this command because of some device condition, the device server shall terminate each WRITE BUFFER command with this mode (110b) with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and shall set the additional sense code to COMMAND SEQUENCE ERROR.

If the last WRITE BUFFER command of a set of one or more commands completes successfully, the microcode or control information shall be transferred to the control memory space of the logical unit. After a power-cycle or reset, the device shall revert to a vendor-specific condition. 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 detects the last download microcode with offsets and save mode WRITE BUFFER command has been received, the device server 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 the device server shall generate a unit attention condition (see SAM) for all initiators except the one that issued the set of WRITE BUFFER commands. When reporting the unit attention condition, the device server shall set the additional sense code to MICROCODE HAS BEEN CHANGED.

If the complete set of WRITE BUFFER commands required to effect a microcode or control information change (one or more commands) are not received before a reset or power-on cycle occurs, the change shall not be effective and the new microcode or control information shall be discarded.

The buffer ID field identifies a specific 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, additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is identified, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of 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. The application client shall send commands that conform to the offset boundary requirements (see 7.15.4). If the device server is unable to accept the specified buffer offset, it shall return CHECK CONDITION status and it shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

The parameter list length 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 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 may be determined by the buffer capacity field in the READ BUFFER descriptor.) If the buffer offset and parameter list length fields specify a transfer in excess of the buffer capacity, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

2.7 Download microcode with offsets and save mode (111b)

In this mode, the initiator may split the transfer of the vendor-specific microcode or control information over two or more WRITE BUFFER commands. If the logical unit cannot accept this command because of some device condition, the device server shall terminate each mode 111b WRITE BUFFER command with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and shall set the additional sense code to COMMAND SEQUENCE ERROR.

If the last WRITE BUFFER command of a set of one or more commands completes successfully, the microcode or control information shall be saved in a non-volatile memory space (semiconductor, disk, or other). The saved downloaded microcode or control information shall then be effective after each power-cycle and 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 detects the last download microcode with offsets and save mode WRITE BUFFER command has been received, the device server 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 the device server shall generate a unit attention condition (see SAM) for all initiators except the one that issued the set of WRITE BUFFER commands. When reporting the unit attention condition, the device server shall set the additional sense code to MICROCODE HAS BEEN CHANGED.

If the complete set of WRITE BUFFER commands required to effect a microcode or control information change (one or more commands) are not received before a reset or power-on cycle occurs, the change shall not be effective and the new microcode or control information shall be discarded.

The buffer ID field identifies a specific 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, additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is identified, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of 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. The application client shall conform to the offset boundary requirements. If the device server is unable to accept the specified buffer offset, it shall return CHECK CONDITION status and it shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

The parameter list length 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 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 may be determined by the buffer capacity field in the READ BUFFER descriptor.) If the buffer offset and parameter list length fields specify a transfer in excess of the buffer capacity, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

2.8 Write data to logical buffer

In this mode the target transfers up to 128 bytes of data from the initiator and stores it in a logical buffer to be retrieved by the complimentary READ BUFFER command. The Buffer ID, and Buffer Offset fields are ignored in this mode.