LSI LOGIC®

4420 ArrowsWest Drive
Colorado Springs, CO 80907

August 21, 2000

To: T10 Technical Committee
From: John Lohmeyer, LSI Logic Principal Member of T10
Subj: Expander Communication Protocol

Revision 2 changes:

1. Corrected error in Table X.2. Reserved range is now C1h - EFh.
2. Clarified that expanders alter the entire first byte of an expander descriptor block.
3. Changed 'initiator' to 'application client' and 'target' to 'device server' in several places.
4. Recommended that disconnects be disabled.
5. Changed 'the ECP protocol' to 'ECP' several places.
6. Clarified non-target port and series expander set definitions.
7. Changed the expander function signature to 7 bytes and made the eighth byte the initiator's address.
8. Deleted the expander reset functions and re-ordered the remaining expander functions.
9. Changed the DRIVER TIMING SKEW to RECEIVER TIMING SKEW in the margin control and margin report functions.
10. Defined address 0000000b as unassigned.
11. Increased the TARGET_ADRS field to 8 bits in the CONTROL data structure.
12. Encoded the far port control actions in the CONTROL data structure and specified when they take effect.
13. Deleted the R_EXP bit and the FES_FAR bit in the CONTROL data structure.
14. Added reserved byte at byte 4 of REPORT CAPABILITIES control/data structure.
15. Included the USED bit and the EXPANDER ADDRESS field in byte 0 of the expander INQUIRY data.
16. Changed nomenclature as follows:

old name	new name
relative addressing function	multiple function
direct addressing function	single function
expander control/data structure	short expander descriptor block (SEDB)
function-specific data structure	long expander descriptor block (LEDB)

I received a request to include the ability to report skew compensation and AAF settings. I have not yet done so, but expect that this could be done with an inbound single function. While the values for these settings are likely to be vendor specific, I think we could standardize the number of fields and the size of the fields. I am especially looking for input on how big to make these fields.

I have documented the ECP proposal in the form of an annex for SPI-4 on the following pages.

Annex X

(normative)

Expander Communication Protocol

X.1 Introduction

This annex describes a method of expander communication and topology discovery called Expander Communication Protocol (ECP). This protocol permits application clients to detect expanders that support the protocol. It also permits the application client to pass parameter settings to expanders and permits expanders to report settings and status information. No new hardware features are required of initiators or targets to implement this protocol.

ECP depends on the expander being able to monitor the data transfers associated with WRITE BUFFER and READ BUFFER commands (see SPC-2) and to alter specific portions of the data transferred as it passes through the expander. To simplify the expander implementation requirements, ECP is restricted to 8-bit asynchronous transfers.

X.2 Glossary and Definitions

X.2.1 Communicative expander: A simple expander (see Annex F) that has the additional capability to support the requirements of this annex and thus is capable of transmitting information beyond that received on its ports to specific other entities in the domain. In this annex, unless stated otherwise, the term expander means communicative expander.

X.2.2 Expander function signature: A specific sequence of data bytes that identifies an ECP function in the data phase of a WRITE BUFFER or READ BUFFER command.

X.2.3 Far port: For the current I/O process, an expander port that is not the near port.

X.2.4 First expander: In a series expander set, the expander that couples the bus segment containing the initiator to the next bus segment on the path to the target.

X.2.5 Last expander: In a series expander set, the expander that couples the bus segment containing the target to the next bus segment on the path to the initiator.

X.2.6 Near port: For the current I/O process, the expander port connected directly to the initiator through a bus segment or connected to the initiator through other expanders and bus segments.

X.2.7 Non-target port: A far port (X.2.7) that is not a target port (X.2.11). That is, all far ports that do not include the target for this I/O process.

X.2.8 nth expander: In a series expander set, the nth expander on the path from the initiator to the target.

X.2.9 Path: The set of all bus segments and expanders between an initiator and a target.

X.2.10 Series expander set: The set of one or more expanders that couple the bus segment containing an initiator to the bus segment containing a target.

X.2.11 Target port: For the current I/O process, a far port that is connected directly to the target through a bus segment or connected to the target through other expanders and bus segments.

X.3 Enabling ECP

Following a power cycle or bus reset, a communicative expander shall function as a simple expander for each initiator until the initiator enables ECP as follows:

1. Negotiate asynchronous transfer mode and a transfer width of 8 bits to some target device
2. Issue a WRITE BUFFER command to the same target device with the MODE field set to Echo buffer plus enable ECP (1Ah).

The initiator may disable ECP by:

1. Negotiate asynchronous transfer mode and a transfer width of 8 bits to some target device
2. Issue a WRITE BUFFER command to the same target device with the MODE field bit set to Disable ECP (1Bh).

[Editor's Note: This proposal would expand the MODE field in the WRITE BUFFER command (SPC-2) from 4 bits to 5 bits adding the above two code values and reserving all other new code values. These values were selected so that enabling ECP and writing to the echo buffer can be combined in one operation with MODE 1Ah. Legacy device servers may interpret this value as either Echo buffer (0Ah) or Write data (02h), which should cause no harm. Legacy device servers may interpret the Disable ECP code value as either 0Bh or 03h, both of which are reserved and should cause no harm.**]**

This enabling and disabling of ECP is done on an initiator basis (i.e., each initiator issues a single WRITE BUFFER command to enable or disable ECP for that initiator for all communicative expanders on the bus). The enabling or disabling of ECP occurs regardless of the device server's response to the WRITE BUFFER command. (Legacy device servers may return CHECK CONDITION status and ILLEGAL REQUEST sense key because they implemented a smaller MODE field.)

Note: While the initiator has ECP enabled, it is responsible for not issuing WRITE BUFFER commands with the EXPANDER FUNCTION SIGNATURE in the first 7 bytes of the data buffer unless it is an expander function header (see X.4).

X.4 Communicative expander function structures

Communicative expander functions consist of outbound and inbound functions. The outbound functions are contained in the data of a WRITE BUFFER command with the MODE field set to Write data (00010b), Echo buffer mode (01010b) or Echo buffer plus enable ECP mode (11010b). The inbound functions return information in the data of a READ BUFFER command with the mode field set to Data (00010b), Echo buffer mode (01010b) or Echo buffer plus enable ECP mode (11010b).

It is recommended that the initiator not enable disconnects for these WRITE BUFFER and READ BUFFER commands. That is, the DISCPRIV bit in the IDENTIFY message (see 16.3.3) should be set to 0.

The outbound and inbound functions are further divided into multiple and single functions. For multiple functions, the data is transferred in a 172-byte data structure consisting of a 16-byte expander function header followed by ten 16-byte short expander descriptor blocks (SEDB). For single functions, the data is transferred in a data structure whose length depends on the function. This data structure consists of a 16-byte expander function header followed by a one long expander descriptor block (LEDB). In either case, the first 16 bytes of the data structure contain an expander function header as shown in table X.1.

Table X.1 — Expander function header

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB) EXPANDER FUNCTION SIGNATURE (B73384B8508F27h)							
1								
2								
3								
4								
5								
6								(LSB)
7	INITIATOR SCSI ADDRESS							
8	EXPANDER FUNCTION CODE							
9	Function specific							
10								
11								
12								
13								
14								
15								

The EXPANDER FUNCTION SIGNATURE contains a code of B73384B8508F27h that signifies this WRITE BUFFER or READ BUFFER data is an expander function.

The application client shall set the INITIATOR SCSI ADDRESS field to the SCSI address of the initiator. The expander shall check that this field matches the SCSI address for the initiator of the current I/O process.

If both the EXPANDER FUNCTION SIGNATURE is correct and the INITIATOR SCSI ADDRESS field matches the initiator's SCSI address, then this WRITE BUFFER or READ BUFFER data is an expander function and shall be processed by the expander. Otherwise, is a normal WRITE BUFFER or READ BUFFER command and shall be repeated by communicative expanders but otherwise shall be ignored.

The EXPANDER FUNCTION CODES are documented in table X.2.

The function-specific bytes are documented for single functions in X.5.2 and X.5.4. The function-specific bytes are not presently used for multiple functions and are reserved for future use.

Table X.2 — Expander functions

EXPANDER FUNCTION CODE	Expander function	Type
00h	ASSIGN ADDRESS	Outbound multiple function
01h	MARGIN CONTROL	
02h - 2Fh	Reserved	
30h - 3Fh	Vendor specific	
40h	CONTROL	Outbound single function
41h - 6Fh	Reserved	
70h - 7Fh	Vendor specific	
80h	Reserved	Inbound multiple function
81h	MARGIN REPORT	
82h	REPORT CAPABILITIES	
83h - AFh	Reserved	
B0h - BFh	Vendor specific	
C0h	EXPANDER INQUIRY	Inbound single function
C1h - EFh	Reserved	
F0h - FFh	Vendor specific	

The outbound multiple functions are documented in X.5.1 and the outbound single functions are documented in X.5.2. The inbound multiple functions are documented in X.5.3 and the inbound single functions are documented in X.5.4.

For outbound and inbound multiple functions, the expander function header is followed by ten short expander descriptor blocks as shown in table X.3. For outbound and inbound single functions, the expander function header is followed by one long expander descriptor block documented in X.5.2 and X.5.4.

Table X.3 — Short expander descriptor block (SEDB)

Bit Byte	7	6	5	4	3	2	1	0
0	USED	Reserved						
1	Function-specific fields							
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

The USED bit is documented in X.5.1.1, X.5.2.1, X.5.3.1, and X.5.4.1.

The remaining fields in the short expander descriptor block are specific to the expander function and are documented in X.5.

X.5 Expander functions

X.5.1 Outbound multiple functions

X.5.1.1 Outbound multiple function data transfer rules

Outbound multiple functions shall be performed during a WRITE BUFFER command with the MODE field set to one of the three values specified in X.4. The initiator shall transfer an expander function header followed by ten short expander descriptor blocks (SEDB). The first SEDB shall have the USED bit set to 1 and may be used by the initiator to indicate its parameters or settings for the specified expander function. The application client shall set the USED bit to 0 in the remaining nine SEDBs. The last SEDB may be used to transfer or report the target's parameters or settings for the specified expander function.

Each expander in the domain shall repeat the entire data structure (expander function header plus ten SEDBs) without alteration to its non-target port or ports, if any. Each expander in the domain shall repeat the entire data structure without alteration to its target port, if any, except the expander shall alter the first SEDB encountered with a USED bit of 0. In this SEDB, the expander shall change the USED bit to 1 and shall output 0 bits in the reserved field of the first byte. The remaining 15 bytes of this SEDB shall be repeated without alteration. The expander shall interpret the other fields of this altered SEDB as described below for the EXPANDER FUNCTION CODE.

An expander that receives a reserved or unimplemented vendor-specific multiple EXPANDER FUNCTION CODE shall follow all of the rules in this sub-clause, but shall ignore the contents of the altered SEDB.

X.5.1.2 ASSIGN ADDRESS

The ASSIGN ADDRESS expander function is used to assign an expander address to one or more expanders. The SEDB for this expander function is shown in table X.7.

Table X.4 — ASSIGN ADDRESS SEDB

Bit Byte	7	6	5	4	3	2	1	0
0	USED	Reserved						
1	ASSIGN	EXPANDER ADDRESS						
2	Reserved							
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

The ASSIGN bit of 1 indicates that the expander shall respond to the expander address specified in the EXPANDER ADDRESS field for single functions. The address assignment shall remain in effect until changed by another ASSIGN ADDRESS function or until the next reset condition or power cycle. An ASSIGN bit of 0 indicates that the expander shall not change its expander address assignment.

Assigning the expander address 0000000b to an expander shall indicate that it has no expander address assigned. The application client shall not use expander address 0000000b in single functions.

X.5.1.3 MARGIN CONTROL

The MARGIN CONTROL expander function sets various parameter settings in the initiator, expander, or target for usage between the initiator-target pair on subsequent synchronous and paced transfers. These parameter settings shall remain in effect until changed by another MARGIN CONTROL expander function or by a reset condition.

The MARGIN CONTROL SEDB is shown in table X.5.

Table X.5 — MARGIN CONTROL SEDB

Bit Byte	7	6	5	4	3	2	1	0
0	USED	Reserved						
1	DRIVER STRENGTH (near port)				RECEIVER TIMING SKEW (near port)			
2	SIGNAL GROUND BIAS (near port)				DRIVER PRECOMPENSATION (near port)			
3	SLEW RATE (near port)				Reserved			
4	Reserved				Reserved			
5	Reserved				Reserved			
6	Reserved				Reserved			
7	Reserved				Vendor specific (near port)			
8	Reserved							
9	DRIVER STRENGTH (target port)				RECEIVER TIMING SKEW (target port)			
10	SIGNAL GROUND BIAS (target port)				DRIVER PRECOMPENSATION (target port)			
11	SLEW RATE (target port)				Reserved			
12	Reserved				Reserved			
13	Reserved				Reserved			
14	Reserved				Reserved			
15	Reserved				Vendor specific (target port)			

Two sets of margin control fields (DRIVER STRENGTH, RECEIVER TIMING SKEW, SIGNAL GROUND BIAS, DRIVER PRECOMPENSATION, and SLEW RATE) are provided, one set for the near port and another set for the target port. The first SEDB is used for the initiator settings, in which only the target port fields are used; the near port fields are reserved. The last SEDB is used for the target settings, in which only the near port fields are used; the target port fields are reserved.

The margin control fields shall be implemented as two's-complement values with 0000b being the nominal value. The maximum supported setting for each field shall be 0111b and the minimum supported setting for each field shall be 1111b. Up to 16 distinct values are available for each field. Expanders that support fewer than 16 distinct values for a field should round intermediate settings to a supported value.

In the case of the SIGNAL GROUND BIAS fields, values 0000b through 0111b shall enable the bias cancellation circuit and values 1000b through 1111b shall disable the bias cancellation circuit, if disabling of this circuit is supported.

X.5.2 Outbound single functions

X.5.2.1 Outbound single function data transfer rules

Outbound single functions shall be performed during a WRITE BUFFER command with the MODE field set to one of the three values specified in X.4. The initiator shall transfer an expander function header followed by a long expander descriptor block (LEDB).

Each expander in the domain shall repeat the entire data structure (expander function header plus the LEDB) without alteration to its non-target port or ports, if any. Each expander in the domain shall repeat the entire data structure without alteration to its target port, if any, except if the EXPANDER ADDRESS field in the first byte of the LEDB matches its currently assigned expander address (see X.5.1.5) and the USED bit is 0. In this case, the expander shall change the USED bit to 1 and shall output its currently assigned expander address in the EXPANDER ADDRESS field. The expander shall interpret the other fields of this altered LEDB as described below for the EXPANDER FUNCTION CODE.

An expander that receives a reserved or unimplemented vendor-specific single EXPANDER FUNCTION CODE shall follow all of the rules in this sub-clause, but shall ignore the contents of the altered LEDB.

X.5.2.2 CONTROL

The CONTROL function is used to set or clear parameters on the addressed expander. The function-specific bytes in the expander function header are reserved for this function. The SEDB for the CONTROL function is shown in table X.6.

Table X.6 — CONTROL data structure

Bit Byte	7	6	5	4	3	2	1	0
0	USED	EXPANDER ADDRESS						
1	TARGET_ADRS							
2	Reserved					FAR_CTL		
3	Reserved							
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

The TARGET_ADRS field shall be set to the SCSI address of a target on one of the expander's far ports. This identifies which far port is to be controlled on expanders that have multiple far ports. If the specified TARGET_ADRS does not match a known target SCSI address on one of the expander's far ports, then the expander shall perform no far port control action on any port.

The FAR_CTL field is defined as shown in table X.7.

Table X.7 — FAR_CTL field values

FAR_CTL	Far port control action
000b	No operation
001b	Disable far port
010b	Enable far port
100b	Reset far port
all other vlaues	Reserved

A far port control action of no operation shall have no effect on the specified far port.

A far port control action of disable far port shall cause the expander to stop repeating signals to the specified far port and shall cause the expander to ignore signals from the specified far port upon the next BUS FREE phase.

A far port control action of enable far port shall cause the expander to resume repeating signals to the specified far port and shall cause the expander to resume responding to signals from the specified far port upon the next BUS FREE phase.

A far port control action of reset far port shall cause the expander to create a hard reset condition on the specified far port upon the next BUS FREE phase (i.e., the expander creates a pulse on the RST signal). The expander shall not propagate this hard reset condition to any other of its ports.

X.5.3 Inbound multiple functions

X.5.3.1 Inbound multiple function data transfer rules

The application client shall set the USED bit of the first short expander descriptor block (SEDB) to 1 and shall set the used bit in the remaining nine SEDBs to 0. The data structure containing an inbound multiple function is then placed in the target's buffer using a WRITE BUFFER command with the MODE field set to one of the three values specified in X.4. Expanders shall not alter the data structure during the WRITE BUFFER command if the EXPANDER FUNCTION CODE is 80h to FFh. The inbound multiple function is then performed during a subsequent READ BUFFER command with the MODE field set to one of the three values specified in X.4.

Device servers that implement ECP should alter the first SEDB to report capabilities or margin settings.

During the data transfer phase of the READ BUFFER command, each expander with a target port shall repeat the entire data structure without alteration to its near port, except the expander shall alter the first SEDB encountered with a USED bit of 0. In this SEDB, the expander shall change the USED bit to 1 and shall output 0 bits in the reserved field of the first byte. The remaining 15 bytes of this SEDB shall be output as described below for the EXPANDER FUNCTION CODE.

An expander that receives a reserved or unimplemented vendor-specific multiple EXPANDER FUNCTION CODE shall follow all of the rules in this sub-clause, but shall output 00h in bytes 1 through 15 of the altered SEDB.

X.5.3.2 MARGIN REPORT

The MARGIN REPORT expander function is used to report the current margin settings for the initiator, expander, or target. The MARGIN REPORT SEDB is shown in table X.11.

Table X.8 — MARGIN REPORT SEDB

Bit Byte	7	6	5	4	3	2	1	0
0	USED	Reserved						
1	DRIVER STRENGTH (near port)				RECEIVER TIMING SKEW (near port)			
2	SIGNAL GROUND BIAS (near port)				DRIVER PRECOMPENSATION (near port)			
3	SLEW RATE (near port)				Reserved			
4	Reserved				Reserved			
5	Reserved				Reserved			
6	Reserved				Reserved			
7	Reserved				Vendor specific (near port)			
8	Reserved							
9	DRIVER STRENGTH (target port)				RECEIVER TIMING SKEW (target port)			
10	SIGNAL GROUND BIAS (target port)				DRIVER PRECOMPENSATION (target port)			
11	SLEW RATE (target port)				Reserved			
12	Reserved				Reserved			
13	Reserved				Reserved			
14	Reserved				Reserved			
15	Reserved				Vendor specific (target port)			

Two sets of margin report fields (DRIVER STRENGTH, RECEIVER TIMING SKEW, SIGNAL GROUND BIAS, DRIVER PRECOMPENSATION, and SLEW RATE) are provided, one set for the near port and another set for the target port. The first SEDB is used for the target settings, in which only the near port fields are used; the target port fields are reserved. The last SEDB is used for the initiator settings, in which only the target port fields are used; the near port fields are reserved. The initiator may or may not set the USED bit to 1 in the last SEDB before returning the data buffer to the application client. It is recommended that initiators that use this SEDB to report the initiator margin settings set the USED bit to 1. Initiators that do not use this SEDB for this purpose should not alter this bit (i.e., leave it set to 0).

The margin report fields shall return the current settings for the initiator-target pair. Fields that are not implemented shall be reported as 0000b. Otherwise, the current setting for the field, possibly rounded as described in X.5.1.3, shall be returned.

X.5.3.3 REPORT CAPABILITES

The REPORT CAPABILITES function is used to determine domain topology and report expander characteristics. The REPORT CAPABILITES SEDB is shown in table X.9.

Table X.9 — REPORT CAPABILITES SEDB

Bit Byte	7	6	5	4	3	2	1	0
0	USED	Reserved						
1	(MSB)	FAR SCSI ID LIST						
2								(LSB)
3	MINIMUM TRANSFER PERIOD FACTOR							
4	Reserved							
5	MAXIMUM REQ/ACK OFFSET							
6	MAXIMUM TRANSFER WIDTH EXPONENT							
7	PROTOCOL OPTION BITS SUPPORTED							
8	PORTS				Reserved		TARG_MODE	
9	Reserved							
10								
11								
12								
13								
14								
15								

The FAR SCSI ID LIST field contains the inclusive OR of all SCSI IDs known to be located on the target port of the expander. For example, if SCSI devices with IDs 0, 1, and 12 were previously accessed on the target port, the expander sets this field to 1003h.

The MINIMUM TRANSFER PERIOD FACTOR field shall be set to the smallest value of the TRANSFER PERIOD FACTOR (see 16.3.10.1) supported by the expander.

The MAXIMUM REQ/ACK OFFSET field shall be set to the largest value of the REQ/ACK OFFSET (see 16.3.10.1) supported by the expander.

The MAXIMUM TRANSFER WIDTH EXPONENT field shall be set to the largest value of the TRANSFER WIDTH EXPONENT (see 16.3.10.1) supported by the expander.

The PROTOCOL OPTIONS BITS SUPPORTED field shall set the corresponding bit to one for each supported protocol option bit in byte 7 of the PPR message (see 16.3.10.1).

The PORTS field shall contain the number of ports on the expander not including the near port. A value of 0 in this field indicates that the expander is not capable of reporting this information.

The TARG_MODE field specifies the current bus mode for the target port as defined in table X.10.

Table X.10 — Far port bus mode

TARG_MODE	Target port bus mode
00b	Unknown (expander not capable of reporting bus mode)
01b	Single ended
10b	Low Voltage Differential
11b	High Voltage Differential

X.5.4 Inbound single functions

X.5.4.1 Inbound single function data transfer rules

The application client shall set the USED bit in the long expander descriptor block (LEDB) to 0. The data structure containing an inbound single function is then placed in the target's buffer using a WRITE BUFFER command with the MODE field set to one of the three values specified in X.4. Expanders shall not alter the data structure during the WRITE BUFFER command if the EXPANDER FUNCTION CODE is 80h to FFh. The inbound single function is then performed during a subsequent READ BUFFER command with the MODE field set to one of the three values specified in X.4.

During the data transfer phase of the READ BUFFER command, each expander with a target port shall repeat the entire data structure without alteration to its near port, except if the EXPANDER ADDRESS field in the first byte of the LEDB matches its currently assigned expander address (see X.5.1.5) and the USED bit is 0. In this case, the expander shall change the USED bit to 1 and shall output its currently assigned expander address in the EXPANDER ADDRESS field. The expander shall output the remaining bytes of this LEDB as described below for the EXPANDER FUNCTION CODE.

An expander that receives a reserved or unimplemented vendor-specific single EXPANDER FUNCTION CODE shall follow all of the rules in this sub-clause, but shall output 00h in remaining bytes of the altered LEDB.

X.5.4.2 EXPANDER INQUIRY

The EXPANDER INQUIRY function is used to report information about the specified expander in a manner similar to the SCSI INQUIRY command documented in SPC-2. The expander function header for this function shall include function specific fields as described in table X.11.

Table X.11 — EXPANDER INQUIRY expander function header

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	EXPANDER FUNCTION SIGNATURE (B73384B8508F27h)						
1								
2								
3								
4								
5								
6								(LSB)
7	INITIATOR SCSI ADDRESS							
8	EXPANDER INQUIRY FUNCTION CODE (C0h)							
9	Reserved							EVPD
10	PAGE CODE							
11	Reserved							
12	ALLOCATION LENGTH							
13	Reserved							
14								
15								

An enable vital product data (EVPD) bit of one specifies that the expander shall return the optional vital product data specified by the PAGE CODE field (see the vital product data parameters documentation in SPC-2). If the expander does not support the optional vital product data, then it shall return all zero bytes for the specified allocation length. If the EVPD bit is zero, then the expander shall return Expander INQUIRY data as documented in table X.12.

Table X.12 — EXPANDER INQUIRY data

Bit Byte	7	6	5	4	3	2	1	0
0	USED	ADDRESS						
1	Reserved							
2								
3								
4	ADDITIONAL LENGTH (33h)							
5	Reserved							
6								
7								
8	(MSB)	VENDOR IDENTIFICATION						
15								(LSB)
16	(MSB)	PRODUCT IDENTIFICATION						
31								(LSB)
32	(MSB)	PRODUCT REVISION LEVEL						
35								(LSB)
36	Vendor-specific							
55								

The VENDOR IDENTIFICATION, PRODUCT IDENTIFICATION, and PRODUCT REVISION LEVEL fields shall return data as documented in SPC-2 for the Standard INQUIRY data format.

X.6 Data Transfer Requirements

The communicative expander functions shall only be performed when the data transfer agreement is 8-bit asynchronous. For any other data transfer agreement, the communicative expander shall operate as a simple expander.

When altering data, communicative expanders shall construct correct parity for the altered data on the outgoing port.