To: T10 Technical Committee
From: Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)
Date: 25 October 2000
Subject: Controlling targets with ECP

Revision 0: first revision

00-257r2, the Expander Communication Protocol (ECP) proposal, defined communication with initiators, expanders, and targets using data embedded in WRITE BUFFER/READ BUFFER data. 00-257r3, which was accepted by T10, dropped target support – it allows expanders and initiators to support ECP commands, but not targets. (the target still must support WRITE BUFFER/READ BUFFER but does not interpret or change the data). This proposal restores target support.

The primary reason for ECP-capable targets is to provide consistent access to commands like MARGIN CONTROL, REPORT DESKEW VALUES (proposed in 00-392), and REPORT AAF VALUES (proposed in 00-393). In a target, these could be implemented in mode pages, log pages, new SCSI commands, new functions in old SCSI commands, or in messages. However, using the same method for expanders and targets is easier for software. All the data structures are compatible by default, error-handling is the same, capabilities stay in sync, etc.

Mode pages are typically not maintained on a per-initiator basis in parallel SCSI disk drives. Margin controls typically are. Targets are allowed to declare all their mode pages as shared or all as per-initiator. Making a MARGIN CONTROL page per-initiator would mean all the other pages would also become per-initiator to follow SPC-2 rules.

Changes to 00-378r0 (proposed SPI-4 revision 1) are listed below. A few editorial changes not directly related to this concept are also interspersed.

G.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 *and targets* that support the protocol. It also permits the application client to pass parameter settings to *initiators,* expanders*, and targets* and permits ~~expanders~~*them* to report settings and status information. No new hardware or firmware features are required of initiators or targets to implement this protocol *for expander communication*. *Initiators and targets that use this protocol for setting parameters and reporting settings do require new hardware or firmware features.*

G.2.x Communicative device: *A communicative expander, initiator, or target.*

G.2.1 Communicative expander: A simple expander (see Annex F) that has the additional capability to support the requirements of this annex and is capable of *interpreting and modifying data in ECP functions*~~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.

G.2.x Communicative initiator: *An initiator that has the additional capability to support the requirements of this annex and is capable of interpreting and modifying data in ECP functions*

G.2.y Communicative target: *A target that has the additional capability to support the requirements of this annex and is capable of interpreting data in ECP outbound functions and modifying data for ECP inbound functions.*

G.2.3 Far port: ***In an expander, the expander port that is not the near port f*** ~~F~~or the current I/O process~~, an expander port that is not the near port~~. *In an initiator, the port being used for the current I/O process.*

G.2.6 Near port: ***In an expander*** ~~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 *for the current I/O process*. *In a target, the port being used for the current I/O process.*

G.2.7 Non-target port: A far port that is not a target port (i.e., all far ports that do not include the target for this I/O process) *Only defined for expanders.* ~~.~~

G.2.10 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. *Only defined for expanders. (The port in use for the current I/O process on a target is a near port, not a target port).*

G.3 Symbols and abbreviations

SEDB Short Expander Descriptor Block~~s~~

G.4 Enabling ECP

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

…

The ~~initiator~~ *application client* may disable ECP by:

G.5 Communicative ~~expander~~ *device* function structures

Communicative ~~expander~~ *device* functions consist of outbound and inbound functions.

…

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 that shall be processed by the ~~expander.~~*communicative devices.* Otherwise, the WRITE BUFFER or READ BUFFER command shall be ignored by the communicative ~~expanders~~*devices*. In all cases the communicative expanders shall repeat the WRITE BUFFER or READ BUFFER data.

…

[In table G.2, consider adding three columns - Initiator, Target, Expander – indicating which expander functions may be implemented by communicative devices of each type. Expander gets a Yes for every command. Initiator and Targets get a Yes for MARGIN CONTROL only.]

…

Table G.4 — Device class

D_CLASS	Device class
000b	Reserved
001b	Communicative expander device
010b	~~SCSI~~ *Communicative i*Initiator device
011b	*Communicative target device*
~~011b~~ *100b* - 111b	Reserved

G.6.1.1 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 values specified in table G.2. The initiator shall transfer an expander function header followed by ten SEDBs. The application client may use an SEDB to ~~indicate~~ *set*

the initiator's parameters or settings. *The application client may use an SEDB to set the target's parameters or settings.* If the application client uses an SEDB *for the initiator*, it shall use the first SEDB and shall set the USED bit in the first SEDB to 1. *If the application client uses an SEDB for the target, it shall use the first unused SEDB.* The application client shall set the USED bit to 0 in all unused SEDBs.

Each expander in the SCSI 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 set the USED bit to 1, shall set the D_CLASS field as described in table G.4, and shall output a zero 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 ~~defined in the ASSIGN ADDRESS (see G.6.1.2), and MARGIN CONTROL (see G.6.1.3)~~ *indicated by the* expander function code~~s~~.

Each expander in the domain shall either repeat the data received on the near port or shall repeat the data output to the target port on all *non-target ports (i.e.* far ports that are not part of the I_T nexus*)*.

A~~n expander~~ *communicative device* 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.

G.6.1.3 MARGIN CONTROL

The MARGIN CONTROL expander function sets parameter settings in the initiator*, target,* or expander*(s)* for usage between the I_T nexus on subsequent synchronous *(??)* transfers and paced transfers. These parameter settings shall remain in effect until changed by another MARGIN CONTROL expander function or by a reset condition.

[Editor's note: should margin control settings apply only to paced data transfers, DT data transfers, and/or ST data transfers?]

…

Two duplicate sets of margin control fields (i.e., DRIVER STRENGTH, SIGNAL GROUND BIAS, DRIVER PRECOMPENSATION, and SLEW RATE) are provided, one set for the near port and another set for the far port. If the first SEDB is used for the initiator settings, only the far port fields are used and the near port fields are reserved. *If an SEDB is used for target settings, only the near port fields are used and the far port fields are reserved.*

`…

For 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. *Interpretation of the other fields is vendor-specific.*

G.6.2.1 Outbound single function data transfer rules

…

Each expander in the SCSI domain shall repeat the entire data structure without alteration to its target port, if any, except if the USED bit is 0 and the contents of the EXPANDER ADDRESS field in the first byte of the LEDB matches the expanders currently assigned expander address for this initiator (see G.6.1.2). In this LEDB the expander shall change the USED bit to 1 and shall output its currently assigned expander address for this initiator in the EXPANDER ADDRESS field. The expander shall interpret the other fields of this altered LEDB as ~~defined in the CONTROL (see G.6.2.2)~~ *indicated by the* expander function code.

…

A~~n expander~~ *communicative device* 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.

G.6.3.1 Inbound multiple function data transfer rules

…

If an SEDB D_TYPE field indicates the communicative target is being addressed, the communicative target shall change the USED bit to 1 and output the remaining bytes of the SEDB as indicated by the expander function code.

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, shall set the D_CLASS field as described in table G.4, and shall output zero in the reserved field of the first byte. The remaining 15 bytes of this SEDB shall be output as ~~described in the MARGIN REPORT (see G.6.3.2) and REPORT CAPABILITES (see G.6.3.3)~~*indicated by the* expander function code~~s~~.

…

An ~~expander~~ *communicative device* 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.

G.6.3.2 MARGIN REPORT

The MARGIN REPORT expander function is used to report the current margin settings for the initiator*, target,* or expander*(s)*. The MARGIN REPORT SEDB is shown in table G.9.

Two duplicate sets of margin report fields (i.e., DRIVER STRENGTH, SIGNAL GROUND BIAS, DRIVER PRECOMPENSATION, and SLEW RATE) are provided, one set for the near port and another set for the far port. *If an SEDB is used to report target settings, only the near port fields are used while the far port fields are reserved.* If the last SEDB is used to communicate initiator settings to the application client, only the far port fields are used while the near port fields are reserved.

G.6.4.1 Inbound single function data transfer rules

…

If the LEDB D_TYPE field indicates the communicative target is being addressed, the communicative target shall change the USED bit to 1 and output the remaining bytes of the LEDB as indicated by the expander function code.

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 USED bit is 0 and the EXPANDER ADDRESS field in the first byte of the LEDB matches its currently assigned expander address (see G.6.1.2). In this LEDB, 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 for the EXPANDER INQUIRY~~*indicated by the* expander function code.

…

A~~n expander~~ *communicative device* 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.

G.7 Data Transfer Requirements

The communicative ~~expander~~ *device* 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 *and communicative initiators and targets shall not respond to ECP functions*.