**To:**       T10 Technical Committee
**From:**     Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)
**Date:**     31 October 2000
**Subject:**  Reporting saved training configuration values

Revision 0, 25 Oct 2000: first revision
Revision 1, 30 Oct 2000: Renamed from "Reporting deskew values with ECP" to "Reporting saved training configuration values."  Merged functionality of 00-393r0, "reporting AAF values", into this proposal.  Changed fields to generic vendor-specific fields with 4 bytes per signal.  Made the proposal include both an ECP version for expanders and a mode page version for targets.

**Related documents**
00-378r0 (by George Penokie, Tivoli) is the proposed text for SPI-4 revision 1, which incorporates ECP.

00-392r0 was Reporting deskew values with ECP (merged into this document)

00-393r0 was Reporting AAF values with ECP (merged into this document)

00-396r1 Mode page equivalents for ECP commands

**Overview**
SPI-4 devices include deskew logic to adjust the skew between the clock and data signals.  For DATA IN phases, this is from REQ to data; for DATA OUT phases, it is from ACK to data.
The deskewed data signals are DB[15:0], DBP_1, and P_CRCA.  Deskew settings are maintained separately for each I_T nexus.  A standard way to return these values is desired for lab debug and diagnostic software.  This might be useful in judging how much margin a system has left – if the skew values are all being pushed to their maximum values, the system has less margin than one where skew values are all near nominal.

SPI-4 devices may include adjustable active filter (AAF) logic in their receivers that determines parameters like amplifier boost values during the training sequence for use during data transfers.
The filtered signals always include all the high speed signals - REQ or ACK, DB[15:0]. P1, P_CRCA.  Other receiver settings may apply to low speed signals.  There may be one setting per bit, or one setting per group of bits – this is vendor-specific.  Settings are maintained separately for each I_T nexus.  This might be useful in judging how much margin a system has left – if the boost values are all being pushed to their maximum values, the system has less margin than one where boost values are all near nominal.

A REPORT SAVED TRAINING CONFIGURATION VALUES subpage of the Port Control mode page is proposed for targets.  A REPORT SAVED TRAINING CONFIGURATION VALUES function using ECP is proposed for expanders.

We have no need yet to change the values with a standard command.  If wanted, a SET SAVED TRAINING CONFIGURATION VALUES outbound single function paralleling this function would work for ECP and changeable bits could be implemented for the mode page.

The meanings of the values are vendor-specific.  This proposal provides a standard way to read them, but not a standard way to interpret them.  Diagnostic software may display the values on-screen or log them to a file, but cannot interpret the values without knowing the device manufacturer and type.  Reserved space is left in the header in case some day we do want to define standard meanings for the values (requiring vendors to round to match as much as possible).  One of the reserved bytes could be set to a non-zero value indicating compliance to such a definition.

Four bytes are provided for each value.

Due to the large amount of data (4 x 27 bytes per port = 98 bytes), an inbound single function is used (for ECP) rather than an inbound multiple function.  Multiple function commands expect only 16 bytes of data per device.

**Change Table G.2 to add a new function code:**

**Table G.2 — Expander functions**

| EXPANDER FUNCTION CODE | Expander function | Type |
|---|---|---|
| C1h<br>C2h - EFh | REPORT SAVED TRAINING CONFIGURATION VALUES<br>Reserved | Inbound single function |

**Add this section:**

## G.6.x REPORT SAVED TRAINING CONFIGURATION VALUES

The REPORT SAVED TRAINING CONFIGURATION VALUES function is used to report the device's saved training configuration values.  These vendor-specific values are maintained by the device when Retain Training Information is negotiated enabled (see 18.8.4.2.1).

For expanders implementing this function, both near port and far port values shall be included. For initiators implementing this function, only the far port values shall be included.  The near port values are used during DATA OUT phases, while the far port values are used during DATA IN phases.

Only values for the I_T connection in progress are reported.

The meaning of the fields is vendor-specific.

The data structure for this function shall include function specific fields described in table G.12x.

**Table G.12x — REPORT SAVED TRAINING CONFIGURATION VALUES data structure**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | (MSB) | | | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | EXPANDER FUNCTION SIGNATURE | | | | |
| 4 | | | | (B73384B8508F27h) | | | | |
| 5 | | | | | | | | |
| 6 | | | | | | | | (LSB) |
| 7 | INITIATOR SCSI ADDRESS | | | | | | | |
| 8 | RESERVED | | | | | | | |
| 9 | RESERVED | | | | | | | |
| 10 | RESERVED | | | | | | | |
| 11 | RESERVED | | | | | | | |
| 12 | NEAR PORT DB[0] VALUE | | | | | | | |
| 13 | | | | | | | | |
| 14 | | | | | | | | |
| 15 | | | | | | | | |
| … | | | | … | | | | |
| 72-75 | NEAR PORT DB[15] VALUE | | | | | | | |
| 76-79 | NEAR PORT P_CRCA VALUE | | | | | | | |
| 80-83 | NEAR PORT P1 VALUE | | | | | | | |
| 84-87 | NEAR PORT BSY VALUE | | | | | | | |
| 88-91 | NEAR PORT SEL VALUE | | | | | | | |
| 92-95 | NEAR PORT RST VALUE | | | | | | | |
| 96-99 | NEAR PORT REQ VALUE | | | | | | | |
| 100-103 | NEAR PORT ACK VALUE | | | | | | | |
| 104-107 | NEAR PORT ATN VALUE | | | | | | | |
| 108-111 | NEAR PORT CD VALUE | | | | | | | |
| 112-115 | NEAR PORT IO VALUE | | | | | | | |
| 116-119 | NEAR PORT MSG VALUE | | | | | | | |
| 120-123 | RESERVED | | | | | | | |
| 124-127 | FAR PORT DB[0] VALUE | | | | | | | |
| … | | | | … | | | | |
| 184-187 | FAR PORT DB[15] VALUE | | | | | | | |
| 188-191 | FAR PORT P_CRCA VALUE | | | | | | | |
| 192-195 | FAR PORT P1 VALUE | | | | | | | |
| 196-199 | FAR PORT BSY VALUE | | | | | | | |
| 200-203 | FAR PORT SEL VALUE | | | | | | | |
| 204-207 | FAR PORT RST VALUE | | | | | | | |
| 208-211 | FAR PORT REQ VALUE | | | | | | | |
| 212-215 | FAR PORT ACK VALUE | | | | | | | |
| 216-219 | FAR PORT ATN VALUE | | | | | | | |
| 220-223 | FAR PORT CD VALUE | | | | | | | |
| 224-227 | FAR PORT IO VALUE | | | | | | | |
| 228-231 | FAR PORT MSG VALUE | | | | | | | |
| 232-235 | RESERVED | | | | | | | |

### 18.1.4.4 Saved Training Configuration Values subpage

The Saved Training Configuration Values subpage is used to report the device's saved training configuration values. These vendor-specific values are maintained by the device when Retain Training Information is negotiated enabled (see 18.8.4.2.1).

Only values for the I_T connection in progress are reported.

The meaning of the fields is vendor-specific.

The data structure for this subpage shall include fields described in table xx.

**Table xx — Saved Training Configuration subpage (3h)**

| Bit Offset | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | RESERVED | | | | | | | |
| 1 | RESERVED | | | | | | | |
| 2 | RESERVED | | | | | | | |
| 3 | RESERVED | | | | | | | |
| 4 | DB[0] VALUE | | | | | | | |
| 5 | | | | | | | | |
| 6 | | | | | | | | |
| 7 | | | | | | | | |
| … | … | | | | | | | |
| 64-67 | DB[15] VALUE | | | | | | | |
| 68-71 | P_CRCA VALUE | | | | | | | |
| 72-75 | P1 VALUE | | | | | | | |
| 76-79 | BSY VALUE | | | | | | | |
| 80-83 | SEL VALUE | | | | | | | |
| 84-87 | RST VALUE | | | | | | | |
| 88-91 | REQ VALUE | | | | | | | |
| 92-95 | ACK VALUE | | | | | | | |
| 96-99 | ATN VALUE | | | | | | | |
| 100-103 | CD VALUE | | | | | | | |
| 104-107 | IO VALUE | | | | | | | |
| 108-111 | MSG VALUE | | | | | | | |
| 112-115 | RESERVED | | | | | | | |
| 116-227 | RESERVED | | | | | | | |

[Editor's note: 8 extra reserved bytes could be added at the top so this lines up over the full LEDB.  This does not seem to be needed.]

[The reserved bytes at the end are there to match the full LEDB data structure.  They could be used in the future to report the device's corresponding initiator values.]