To: T10 Technical Committee
From: Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)
Date: 26 July 2001
Subject: SPI-4 negotiation message rewrite

Revision History
Revision 0 (16 April 2001): first revision released to T10.
Revision 1 (30 June 2001): incorporated feedback from Brian Cockburn (Adva Optical) and the May Parallel SCSI WG. Added figures for initiator-originated and target-originated WDTR and SDTR.
Revision 3 (26 July 2001): incorporated more feedback from George Penokie and the July Parallel SCSI WG. Added recommendations for numerous editorial updates to match the core proposal.

Related Documents
spi4r05 – SCSI Parallel Interface – 4 revision 5 (core proposal based on this)
spi4r06 – SCSI Parallel Interface – 4 revision 6 (editorial updates based on this)

Overview
The PPR, WDTR, and SDTR message descriptions are full of duplications and contain some errors. This proposal rewrites the three sections, moving most of the text into a model section in clause 4 and simplifying the individual sections in clause 16. It attempts to eliminate duplications and fix errors. It only attempts to change behavior that needs to be changed.

Most of the errors are in the descriptions of error handling – parity errors and unexpected bus frees. The standard is not clear on when the initiator and target should maintain their negotiated settings and when they should reset them to asynchronous. This does not cause many problems in practice because errors are rare, and today’s software probably issues a bus resets when errors occur, which brings all the devices back to a known state. By providing better guidance, resorting to bus resets should be less necessary.
**Suggested Changes**

**Table of contents:**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Negotiation</td>
<td>3</td>
</tr>
<tr>
<td>4.1.1 Negotiation introduction</td>
<td>3</td>
</tr>
<tr>
<td>4.1.2 Negotiation algorithm</td>
<td>3</td>
</tr>
<tr>
<td>4.1.3 When to negotiate</td>
<td>4</td>
</tr>
<tr>
<td>4.1.4 Negotiable fields</td>
<td>5</td>
</tr>
<tr>
<td>4.1.4.1 Negotiable fields introduction</td>
<td>5</td>
</tr>
<tr>
<td>4.1.4.2 Transfer agreements</td>
<td>6</td>
</tr>
<tr>
<td>4.1.4.3 Transfer period factor</td>
<td>6</td>
</tr>
<tr>
<td>4.1.4.4 REQ/ACK offset</td>
<td>7</td>
</tr>
<tr>
<td>4.1.4.5 Transfer width exponent</td>
<td>9</td>
</tr>
<tr>
<td>4.1.4.6 Protocol options</td>
<td>9</td>
</tr>
<tr>
<td>4.1.4.6.1 Protocol options introduction</td>
<td>9</td>
</tr>
<tr>
<td>4.1.4.6.2 IU_REQ</td>
<td>9</td>
</tr>
<tr>
<td>4.1.4.6.3 DT_REQ</td>
<td>10</td>
</tr>
<tr>
<td>4.1.4.6.4 QAS_REQ</td>
<td>10</td>
</tr>
<tr>
<td>4.1.4.6.5 HOLD_MCS</td>
<td>10</td>
</tr>
<tr>
<td>4.1.4.6.6 WR_FLOW</td>
<td>10</td>
</tr>
<tr>
<td>4.1.4.6.7 RD_STRM</td>
<td>11</td>
</tr>
<tr>
<td>4.1.4.6.8 RTI</td>
<td>11</td>
</tr>
<tr>
<td>4.1.4.6.9 PCOMP_EN</td>
<td>11</td>
</tr>
<tr>
<td>4.1.5 Negotiable field combinations</td>
<td>11</td>
</tr>
<tr>
<td>4.1.6 Message restrictions</td>
<td>12</td>
</tr>
<tr>
<td>4.1.7 Negotiation message sequences</td>
<td>13</td>
</tr>
<tr>
<td>4.1.7.1 Negotiation message sequences overview</td>
<td>13</td>
</tr>
<tr>
<td>4.1.7.2 Initiator originated PPR negotiation</td>
<td>14</td>
</tr>
<tr>
<td>4.1.7.3 Initiator originated WDTR negotiation</td>
<td>16</td>
</tr>
<tr>
<td>4.1.7.4 Initiator originated SDTR negotiation</td>
<td>18</td>
</tr>
<tr>
<td>4.1.7.5 Target originated WDTR negotiation</td>
<td>20</td>
</tr>
<tr>
<td>4.1.7.6 Target originated SDTR negotiation</td>
<td>22</td>
</tr>
<tr>
<td>16.3.1 Link control message codes</td>
<td>24</td>
</tr>
<tr>
<td>16.3.12 PARALLEL PROTOCOL REQUEST message description</td>
<td>24</td>
</tr>
<tr>
<td>16.3.16 SYNCHRONOUS DATA TRANSFER REQUEST message description</td>
<td>24</td>
</tr>
<tr>
<td>16.3.18 WIDE DATA TRANSFER REQUEST message description</td>
<td>25</td>
</tr>
</tbody>
</table>
4.1 Negotiation

4.1.1 Negotiation introduction

PARALLEL PROTOCOL REQUEST (PPR), SYNCHRONOUS DATA TRANSFER REQUEST (SDTR), and WIDE DATA TRANSFER REQUEST (WDTR) messages are used to alter the transfer agreement between two ports (see 3.1.76). The transfer agreement defines the protocol used during DATA phases (e.g., transfer period, REQ/ACK offset, transfer width) and agreement on features not affecting DATA phases (e.g., QAS). All other information transfer phases (COMMAND, MESSAGE, and STATUS) use eight-bit asynchronous data transfers.

PPR, SDTR, and WDTR are called negotiation messages. When an initiator sends one of them, the message names are PPR OUT, SDTR OUT, and WDTR OUT. When a target sends one of them, the message names are PPR IN, SDTR IN, and WDTR IN. A negotiation sequence consists of at least one matching set of negotiation messages [e.g., PPR OUT and PPR IN].

A transfer agreement is maintained by each port for each other port on the SCSI bus. Each port (see 3.1.76) may be used as either a target port (see 3.1.xx) or an initiator port (see 3.1.xx). The same transfer agreement applies whether the port is being used as a target port or as an initiator port.

4.1.2 Negotiation algorithm

An initiator port and target port exchange negotiation messages to perform negotiation. The originating port is the one that sends the first negotiation message and the responding port is the one that replies.

Ports shall not set message fields to values they do not support. The originating port should set the fields in the originating negotiation message to the maximum values (e.g., fastest transfer period, largest REQ/ACK offset, etc.) it supports. If the responding port is able to support the requested values, it shall return the same values in the responding negotiation message. If the responding port requires different values (i.e., a subset of the originating port’s request), it shall return those values in the responding negotiation message (e.g., if the originating port asks for a REQ/ACK offset of 32 and the responding port only supports a REQ/ACK offset of 16, the originating message requests an offset of 32 and the responding message replies with an offset of 16).

If the responding negotiation message contains values the originating port does not support, the originating port shall respond with a MESSAGE REJECT message.

The valid error-free negotiation message sequences are shown in Figure 1. A description of all the possible message sequences is in section 4.1.7.
4.1.3 When to negotiate

A target port shall consider its transfer agreement invalid after:

a) a reset event (see 12.5); or
b) an error occurs while transmitting a responding negotiation message.

An initiator port shall consider its transfer agreement invalid after:

a) a reset event (see 12.5));
b) a unit attention status is received with an 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; TRANSCEIVER MODE CHANGED TO SINGLE-ENDED; or TRANSCEIVER MODE CHANGED TO LVD);

NOTE nn: These additional sense codes are never reported in a status information unit because SCSI devices default to information units disabled.
c) an unexpected COMMAND phase occurs when selecting without using attention condition (i.e., when selecting a target with information units enabled), (the initiator port detects an unexpected COMMAND phase (see 10.6.3.1). This may occur if the target has been hot-swapped; or
d) an error occurs while transmitting a responding negotiation message.

A logical unit reset (see 16.5.6) has no effect on a transfer agreement.

An initiator port shall originate negotiation before sending a command whenever it has an invalid transfer agreement. A target port shall originate negotiation before accepting a command whenever it has an invalid transfer agreement.

A port may originate negotiation even if it has a valid transfer agreement (e.g., to change the settings or as part of integrity checking procedures). Negotiation should not be originated after every selection as this may impact performance. Because ports remember their transfer agreements between connections, negotiation is unnecessary and performance impact from extra negotiations is likely.

[SCSI-2 had a note 43 in 6.6.1.1 SDTR section like this:]
NOTE nn: Target ports may have had their support for originating negotiation after power on disabled to support broken initiator software. If an initiator port sends a command to a target that has been powered on (e.g., after a hot plug) that results in a unit attention condition, the initiator port realizes the transfer agreement is invalid and originates negotiation before the next command. However, if the command is INQUIRY, REPORT LUNS, or REQUEST SENSE, a unit attention condition is not created. An invalid data phase may occur if the target port does not originate negotiation. If the initiator port always originates negotiation before sending those commands, the data phase runs correctly. When information units are disabled, an initiator port should-may originate negotiation with its currently negotiated settings before each INQUIRY, REPORT LUNS, or REQUEST SENSE command to avoid this problem. When information units are enabled, the selection without attention results in an unexpected COMMAND phase that notifies the initiator port that its transfer agreement is invalid, so extra negotiation is not needed.

4.1.4 Negotiable fields

4.1.4.1 Negotiable fields introduction

Table 1 lists the fields that may be negotiated and the effects of successful negotiation on those fields by each of the different negotiation messages. Ports shall implement a given message if they implement fields that are only negotiable with that message.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Negotiation message pair</th>
<th>PPR</th>
<th>WDTR</th>
<th>SDTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSFER PERIOD FACTOR</td>
<td>negotiated</td>
<td>(valid values: 08h-FFh)</td>
<td>No requirement</td>
<td>negotiated (valid values: 0Ah-FFh)</td>
</tr>
<tr>
<td>REQ/ACK OFFSET</td>
<td>negotiated</td>
<td>sets to 00h</td>
<td>negotiated</td>
<td></td>
</tr>
<tr>
<td>TRANSFER WIDTH EXPONENT</td>
<td>negotiated</td>
<td>negotiated</td>
<td>negotiated</td>
<td>unchanged</td>
</tr>
<tr>
<td>PROTOCOL OPTIONS</td>
<td>PCOMP_EN</td>
<td>negotiated</td>
<td>sets to zero</td>
<td>sets to zero</td>
</tr>
<tr>
<td></td>
<td>RTI</td>
<td>negotiated</td>
<td>sets to zero</td>
<td>sets to zero</td>
</tr>
<tr>
<td></td>
<td>RD_STRM</td>
<td>negotiated</td>
<td>sets to zero</td>
<td>sets to zero</td>
</tr>
<tr>
<td></td>
<td>WR_FLOW</td>
<td>negotiated</td>
<td>sets to zero</td>
<td>sets to zero</td>
</tr>
<tr>
<td></td>
<td>HOLD_MCS</td>
<td>negotiated</td>
<td>sets to zero</td>
<td>sets to zero</td>
</tr>
<tr>
<td></td>
<td>OAS_REQ</td>
<td>negotiated</td>
<td>sets to zero</td>
<td>sets to zero</td>
</tr>
<tr>
<td></td>
<td>DT_REQ</td>
<td>negotiated</td>
<td>sets to zero</td>
<td>sets to zero</td>
</tr>
<tr>
<td></td>
<td>IU_REQ</td>
<td>negotiated</td>
<td>sets to zero</td>
<td>sets to zero</td>
</tr>
</tbody>
</table>

When negotiating, the responding port shall respond with values that are a subset of the values in the originating message as indicated by the “Response shall be” column in Table 2 (e.g., if the originating message requests a REQ/ACK offset of 10h, the responding message has a REQ/ACK offset field set to 10h or lower).
### Table 2. Responding message requirements

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Message</th>
<th>Response shall be numerically</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSFER PERIOD FACTOR</td>
<td>PPR, SDTR</td>
<td>Greater than or equal</td>
</tr>
<tr>
<td>REQ/ACK OFFSET</td>
<td>PPR, SDTR</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>TRANSFER WIDTH EXPONENT</td>
<td>PPR, WDTR</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>PROTOCOL OPTIONS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCOMP_EN</td>
<td>PPR</td>
<td>Any value</td>
</tr>
<tr>
<td>RTI</td>
<td>PPR</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>RD_STRM</td>
<td>PPR</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>WR_FLOW</td>
<td>PPR</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>HOLD_MCS</td>
<td>PPR</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>QAS_REQ</td>
<td>PPR</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>DT_REQ</td>
<td>PPR</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>IU_REQ</td>
<td>PPR</td>
<td>Less than or equal</td>
</tr>
</tbody>
</table>

### 4.1.4.2 Transfer agreements

The valid transfer agreements that are in effect for various combinations of field values are described in Table 3.
### Table 3. Valid transfer agreements

<table>
<thead>
<tr>
<th>Transfer agreement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>valid</td>
<td>Port has a transfer agreement in effect.</td>
</tr>
<tr>
<td>invalid</td>
<td>Port has no transfer agreement in effect; negotiation is needed.</td>
</tr>
<tr>
<td>default</td>
<td>REQ/ACK offset set to 00h transfer width exponent set to 00h all protocol options set to zero</td>
</tr>
<tr>
<td>asynchronous</td>
<td>REQ/ACK offset set to 00h all protocol options set to zero</td>
</tr>
<tr>
<td>synchronous</td>
<td>REQ/ACK offset greater than or equal to 01h transfer period factor greater than or equal to 09h all protocol options set to zero</td>
</tr>
<tr>
<td>ST synchronous</td>
<td>REQ/ACK offset greater than or equal to 01h transfer period factor greater than or equal to 09h DT_REQ set to zero IU_REQ set to zero</td>
</tr>
<tr>
<td>DT synchronous</td>
<td>REQ/ACK offset greater than or equal to 01h transfer period factor greater than or equal to 09h DT_REQ set to one</td>
</tr>
<tr>
<td>paced</td>
<td>REQ/ACK offset greater than or equal to 01h transfer width exponent set to 01h transfer period factor set to 08h DT_REQ set to one IU_REQ set to one</td>
</tr>
<tr>
<td>wide</td>
<td>Transfer width exponent set to 01h</td>
</tr>
<tr>
<td>narrow</td>
<td>Transfer width exponent set to 00h</td>
</tr>
<tr>
<td>data group</td>
<td>REQ/ACK offset greater than or equal to 01h DT_REQ set to one IU_REQ set to zero</td>
</tr>
<tr>
<td>information unit</td>
<td>REQ/ACK offset greater than or equal to 01h DT_REQ set to one IU_REQ set to one</td>
</tr>
<tr>
<td>ST data</td>
<td>REQ/ACK offset greater than or equal to 01h DT_REQ set to zero</td>
</tr>
<tr>
<td>DT data</td>
<td>REQ/ACK offset greater than or equal to 01h DT_REQ set to one</td>
</tr>
</tbody>
</table>

[July WG requested section references be added, but to what? Most sentences defining the transfer agreements were removed. Description now reformatted one field per line to make them more readable. Added valid, invalid, ST synchronous, DT synchronous. Added IU_REQ set to one to paced description. Removed protocol options set to zero from the synchronous description.]

#### 4.1.4.3 Transfer period factor

The TRANSFER PERIOD FACTOR field selects the transfer period (see 3.1.101) and determines which transfer rate's timing values in table 32, table 33, table 34, and table 35 shall be honored. The field values are defined in Table 4. The TRANSFER PERIOD FACTOR field is negotiated with the PPR and SDTR messages.
### Table 4. Transfer Period Factor

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Message</th>
<th>Transfer rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h - 07h</td>
<td>Reserved. Faster transfer periods may be defined by future standards.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>08h</td>
<td>Transfer period equals 6,25 ns. <strong>Only valid for paced transfers.</strong></td>
<td>PPR</td>
<td>Fast-160</td>
</tr>
<tr>
<td>09h</td>
<td>Transfer period equals 12,5 ns. <strong>Only valid for DT data transfers.</strong></td>
<td>PPR</td>
<td>Fast-80</td>
</tr>
<tr>
<td>0Ah</td>
<td>Transfer period equals 25 ns</td>
<td>PPR, SDTR</td>
<td>Fast-40</td>
</tr>
<tr>
<td>0Bh</td>
<td>Transfer period equals 30,3 ns</td>
<td>PPR, SDTR</td>
<td>Fast-40</td>
</tr>
<tr>
<td>0Ch</td>
<td>Transfer period equals 50 ns</td>
<td>PPR, SDTR</td>
<td>Fast-20</td>
</tr>
<tr>
<td>0Dh - 18h</td>
<td>Transfer period equals the TRANSFER PERIOD FACTOR x 4</td>
<td>PPR, SDTR</td>
<td>Fast-20</td>
</tr>
<tr>
<td>19h - 31h</td>
<td>Transfer period equals the TRANSFER PERIOD FACTOR x 4</td>
<td>PPR, SDTR</td>
<td>Fast-10</td>
</tr>
<tr>
<td>32h - FFh</td>
<td>Transfer period equals the TRANSFER PERIOD FACTOR x 4</td>
<td>PPR, SDTR</td>
<td>Fast-5</td>
</tr>
</tbody>
</table>

Table 5 shows which transfer period factors may be used with different types of transfer agreements.

### Table 5. Transfer Period Factor relationships

<table>
<thead>
<tr>
<th>Value</th>
<th>Synchronous</th>
<th>Paced</th>
<th>Data group</th>
<th>Information unit</th>
<th>ST data</th>
<th>DT data</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h - 07h</td>
<td>reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08h</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>09h</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>0Ah</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>0Bh</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>0Ch</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>0Dh - 18h</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>19h - 31h</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>32h - FFh</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 9 defines valid combinations of transfer period factor and other fields.

**4.1.4.4 REQ/ACK offset**

The REQ/ACK OFFSET field determines the maximum number of REQs allowed to be outstanding before a corresponding ACK is received at the target during synchronous or paced transfers. The REQ/ACK OFFSET field is negotiated with the PPR and SDTR messages.

**Option A: wording based on original PPR text**

For ST **data synchronous** transfers (i.e., DT_REQ negotiated to zero) the REQ/ACK OFFSET is the maximum number of REQ assertions allowed to be outstanding before a corresponding ACK assertion is received at the target. The REQ/ACK offset represents the number of bytes if the transfer width is one byte or twice the number of bytes if the transfer width is two bytes.
For DT synchronous data transfers (i.e., DT_REQ negotiated to one and IU_REQ negotiated to zero) the REQ/ACK OFFSET is the maximum number of REQ transitions allowed to be outstanding before a corresponding ACK transition is received at the target. The REQ/ACK offset represents twice the number of bytes, since DT data transfers always use a transfer width of two bytes.

For paced transfers in DT DATA IN transfers-phase the REQ/ACK OFFSET is the maximum number of data valid state REQ assertions (see 10.8.4.3) allowed to be outstanding before a corresponding ACK assertion is received at the target. The REQ/ACK OFFSET represents four times the number of bytes.

For paced transfers in DT DATA OUT transfers-phase the REQ/ACK OFFSET is the maximum number of REQ assertions allowed to be outstanding before a corresponding data valid state ACK assertion is received at the target. The REQ/ACK OFFSET represents four times the number of bytes.

Option B: alternative wording adopted from spi4r06 section 4.11. This may be clearer than the previous wording.

For ST synchronous transfers the REQ/ACK offset is the number of REQ assertions that may be sent by the target in advance of the number of ACK assertions received from the initiator.

For DT synchronous transfers the REQ/ACK offset is the number of REQ transitions that may be sent by the target in advance of the number of ACK transitions received from the initiator.

For paced transfers in DT DATA IN phase the REQ/ACK offset is the number of data valid state REQ assertions (see 10.7.4.3) that may be sent by the target in advance of ACK assertions received from the initiator.

For paced transfers in DT DATA OUT phase the REQ/ACK offset is the number of REQ assertions that may be sent by the target in advance of the number of data valid state ACK assertions (see 10.7.4.3) received from the initiator.

[end of Editor’s choice]

See 4.8 for an explanation of the differences between ST and DT data transfers.

The REQ/ACK OFFSET value is chosen to prevent overflow conditions in the port’s receive buffer and offset counter. The values are defined in Table 6. Table 6 also indicates which timing values in table 32, table 33, table 34, and table 35 shall be honored.

**Table 6. REQ/ACK offset**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Timing values</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>Asynchronous transfer agreement. Transfer period factor and protocol options shall be ignored.</td>
<td>Asynch</td>
</tr>
<tr>
<td>01h-FEh</td>
<td>Synchronous or paced transfer agreement. Specified offset.</td>
<td>Determined by Transfer Period Factor, See Table 4.</td>
</tr>
<tr>
<td>FFh</td>
<td>Synchronous or paced transfer agreement. Unlimited offset.</td>
<td>Determined by Transfer Period Factor, See Table 4.</td>
</tr>
</tbody>
</table>
Table 9 defines valid combinations of REQ/ACK offset and other fields.

4.1.4.5 Transfer width exponent

The TRANSFER WIDTH EXPONENT field defines the transfer width to be used during DATA IN and DATA OUT phases during data transfers. The values are defined in Table 7. The TRANSFER WIDTH EXPONENT field is negotiated with the PPR and WDTR messages.

If any of the protocol options bits are set to one, the only valid transfer width is 16 bits (01h). If all the protocol options bits are set to zero, the valid transfer widths are 8 bits (00h) or 16 bits (01h). A TRANSFER WIDTH EXPONENT field value of 02h is obsolete and values greater than 02h are reserved.

If the transfer width is 8 bits a narrow transfer agreement is in effect. If the transfer width is 16 bits a wide transfer agreement is in effect.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>8 bit data bus (narrow), Narrow transfer agreement.</td>
</tr>
<tr>
<td>01h</td>
<td>16 bit data bus (wide), Wide transfer agreement.</td>
</tr>
<tr>
<td>02h</td>
<td>Obsolete</td>
</tr>
<tr>
<td>03h-FFh</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Table 9 defines valid combinations of transfer width exponent and other fields.

4.1.4.6 Protocol options

4.1.4.6.1 Protocol options introduction

The protocol options fields affect the protocol used between the ports. They may only be negotiated through PPR messages (which are originated only by initiator ports), and are set to zero by WDTR and SDTR messages.

The target port uses the protocol options bits to indicate to the initiator port if it agrees to enable the requested protocol options. Except for the PCOMP_EN bit, the target shall not enable any protocol options that were not enabled in the negotiation message received from the initiator.

Table 8 describes the protocol options bits.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCOMP_EN</td>
<td>Precompensation enable</td>
</tr>
<tr>
<td>RTI</td>
<td>Retain training information</td>
</tr>
<tr>
<td>RD_STRM</td>
<td>Read streaming and read flow control enable</td>
</tr>
<tr>
<td>WR_FLOW</td>
<td>Write flow control enable</td>
</tr>
<tr>
<td>HOLD_MCS</td>
<td>Hold margin control settings</td>
</tr>
<tr>
<td>QAS_REQ</td>
<td>Quick arbitration and selection (QAS) enable request</td>
</tr>
<tr>
<td>DT_REQ</td>
<td>Dual transition (DT) clocking enable request</td>
</tr>
<tr>
<td>IU_REQ</td>
<td>Information units (IU) enable request</td>
</tr>
</tbody>
</table>

4.1.4.6.2 IU_REQ

The initiator port shall set IU_REQ to one in the PPR OUT message to request that information unit transfers be enabled. In response, the target port shall set its IU_REQ to one if it agrees to use information unit transfers or zero if it does not.
The initiator port shall set IU_REQ to zero in the PPR OUT message to request that information unit transfers be disabled. In response, the target port shall set IU_REQ to zero in the PPR IN message.

If IU_REQ is one, an information unit transfer agreement is in effect. If IU_REQ is zero, an asynchronous, ST synchronous, or data group transfer agreement is in effect.

Table 9 defines valid combinations of IU_REQ and other fields.

Each time a negotiation results in the IU_REQ bit being changed from the previous agreement (i.e., zero to one or one to zero) the target shall go to a BUS FREE phase on completion of the negotiation. Additional requirements (see 14.1) shall be met if the IU_REQ bit is changed as a result of the negotiation.

4.1.4.6.3 DT_REQ

The initiator port shall set DT_REQ to one to request that DT DATA phases be enabled. In response, the target port shall set DT_REQ to one if it agrees to use DT DATA phases or zero if it does not.

The initiator port shall set DT_REQ to zero to request that information unit transfers be disabled. In response, the target port shall set DT_REQ to zero in the PPR IN message.

If DT_REQ is one, a DT data transfer agreement is in effect. If DT_REQ is zero, an ST data transfer agreement is in effect.

Table 9 defines valid combinations of DT_REQ and other fields.

4.1.4.6.4 QAS_REQ

The initiator port shall set QAS_REQ to one to request that QAS be enabled. In response, the target port shall set QAS_REQ to one if it supports QAS or zero if it does not.

The initiator port shall set QAS_REQ to zero to request that information unit transfers be disabled. In response, the target port shall set QAS_REQ to zero in the PPR IN message.

Table 9 defines valid combinations of QAS_REQ and other fields.

When QAS is enabled, the port shall participate in QAS arbitrations when attempting to connect to a port that has enabled QAS. When QAS is enabled and information unit transfers are enabled for a connected target port, that target port may issue a QAS REQUEST message to release the bus after a DT DATA phase. When QAS is enabled and information unit transfers are disabled for a connected target port, that target port shall not issue QAS REQUEST messages.

4.1.4.6.5 HOLD_MCS

The initiator port shall set HOLD_MCS to one to indicate that the target should hold any margin control settings set with the margin control subpage of the port control mode page (see 18.1.4). In response, the target port shall set HOLD_MCS to one if it is capable of retaining the settings and zero if it is not.

The initiator port shall set HOLD_MCS to zero to indicate that the target shall reset to their default values any margin control settings set with the margin control subpage of the port control mode page (see 18.1.4). In response, the target port shall set HOLD_MCS to zero.

Table 9 defines valid combinations of HOLD_MCS and other fields.
4.1.4.6.6  WR_FLOW

The initiator port shall set WR_FLOW to one to indicate that the target should enable write flow control during write streaming (see Table 29, 4.10.3.3 and 8.2). In response, the target port shall set WR_FLOW to one if it is capable of write flow control and zero if it is not.

The initiator port shall set WR_FLOW to zero to indicate that the target shall disable write flow control during write streaming. In response, the target port shall set WR_FLOW to zero.

Write streaming and write flow control only occurs during information unit transfers.

Table 9 defines valid combinations of WR_FLOW and other fields.

4.1.4.6.7  RD_STRM

The initiator port shall set RD_STRM to one to indicate that the target should enable read streaming and read flow control (see Table 29, 4.10.3.3, 8.2, and 14.3.4). In response, the target port shall set RD_STRM to one if it agrees capable of read streaming and read flow control and zero if it is not.

The initiator port shall set RD_STRM to zero to indicate that the target shall disable read streaming and read flow control. In response, the target port shall set RD_STRM to zero.

Read streaming and read flow control only occur during information unit transfers.

Table 9 defines valid combinations of RD_STRM and other fields.

4.1.4.6.8  RTI

The initiator port shall set RTI to one to indicate it is capable of request saving paced data transfer training information (see 10.8.4.2.1) and to indicate that the target should not need to retrain on each connection. In response, the target port shall set RTI to one if it is capable of saving paced data transfer training information and zero if it is not.

The initiator port shall set RTI to zero to indicate that the target shall retrain. In response, the target port shall set RTI to zero.

Table 9 defines valid combinations of RTI and other fields. For negotiated transfer periods slower than fast-160 the RTI bit shall be set to zero.

4.1.4.6.9  PCOMP_EN

The initiator port shall set PCOMP_EN to one to indicate that the target shall enable precompensation on all signals transmitted during DT DATA phases (see 4.9, 7.3.2, and 10.8.4.1). The initiator port shall set PCOMP_EN to zero to indicate that the target shall disable precompensation.

The target port shall set PCOMP_EN to one to indicate that the initiator port shall enable precompensation on all signals transmitted during DT DATA phases (see 4.9, 7.3.2, and 10.8.4.1). The target port shall set PCOMP_EN to zero to indicate that the initiator port shall disable precompensation.

Table 9 defines valid combinations of PCOMP_EN and other fields. Ports that support fast-160 shall support enabling and disabling precompensation of their drivers. For negotiated transfer periods slower than fast-160 the PCOMP_EN bit shall be set to zero.

NOTE nn: Unlike other fields and bits in the PPR message the PCOMP_EN bit is not a negotiated value; instead, it instructs the receiving SCSI device as to whether or not precompensation is to be disabled or enabled. Because of this, precompensation may be enabled on one of the SCSI devices and disabled on the other SCSI device at the completion of a successful PPR negotiation.
4.1.5 Negotiable field combinations

Not all combinations of the negotiable fields are valid. Only the combinations defined in Table 9 shall be allowed. All other combinations of the listed fields are reserved.

Table 9. Valid negotiable field combinations

<table>
<thead>
<tr>
<th>Transfer period factor</th>
<th>REQ/ACK Offset</th>
<th>Transfer width exponent</th>
<th>Protocol options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0Ah – FFh</td>
<td>00h</td>
<td>00h or 01h</td>
<td>0 0 0 0 0 0 0 0</td>
<td>Use ST DATA IN and ST DATA OUT phases to transfer data with asynchronous data transfers</td>
</tr>
<tr>
<td>0Ah - FFh</td>
<td>01h - FFh</td>
<td>00h or 01h</td>
<td>0 0 0 0 0 0 0 0</td>
<td>Use ST DATA IN and ST DATA OUT phases to transfer data with synchronous data transfers</td>
</tr>
<tr>
<td>09h - FFh</td>
<td>01h - FFh</td>
<td>01h</td>
<td>0 0 0 0 0 0 1 0</td>
<td>Use DT DATA IN and DT DATA OUT phases with data group transfers</td>
</tr>
<tr>
<td>09h - FFh</td>
<td>01h - FFh</td>
<td>01h</td>
<td>0 0 0 0 0 1 1 0</td>
<td>Use DT DATA IN and DT DATA OUT phases with data group transfers, and participate in QAS arbitrations</td>
</tr>
<tr>
<td>09h – FFh</td>
<td>01h – FFh</td>
<td>01h</td>
<td>0 0 0 0 0 1 1 1</td>
<td>Use DT DATA IN and DT DATA OUT phases with information unit transfers</td>
</tr>
<tr>
<td>08h</td>
<td>01h - FFh</td>
<td>01h</td>
<td>0 or 1 0 or 1 0 or 1 0 or 1 0 1 1</td>
<td>Use DT DATA IN and DT DATA OUT phases with information unit transfers</td>
</tr>
<tr>
<td>09h – FFh</td>
<td>01h - FFh</td>
<td>01h</td>
<td>0 0 0 or 1 0 or 1 0 or 1 1 0 1 1</td>
<td>Use DT DATA IN and DT DATA OUT phases with information unit transfers, participate in QAS arbitrations, and issue QAS_REQUEST messages to initiate QAS arbitrations</td>
</tr>
<tr>
<td>08h</td>
<td>01h - FFh</td>
<td>01h</td>
<td>0 or 1 0 or 1 0 or 1 0 or 1 1 1 1</td>
<td>Use DT DATA IN and DT DATA OUT phases with information unit transfers, participate in QAS arbitrations, and issue QAS_REQUEST messages to initiate QAS arbitrations</td>
</tr>
</tbody>
</table>

4.1.6 Message restrictions

PPR may be originated by initiator ports but shall not be originated by target ports.

If bus expanders are present, initiator ports should only use PPR when requesting values not attainable via WDTR and SDTR (e.g., setting any protocol option bits to one). If a target port responds to PPR with
values that are attainable via WDTR and SDTR, the initiator port should repeat negotiation starting with a WDTR and SDTR negotiation sequence. This ensures that bus expanders that do not support PPR are still able to handle the data phase correctly.

WDTR and SDTR may be originated by either target ports or initiator ports. Since WDTR resets all the values that SDTR sets (see 4.1.4.1), it shall be sent first if both are needed.

### 4.1.7 Negotiation message sequences

#### 4.1.7.1 Negotiation message sequences overview

An initiator originated negotiation sequence contains up to four steps:

1) Initiator port's originating message;
2) Target port response;
3) Initiator port response; and
4) Target port second response.

A target originated negotiation sequence contains up to four steps:

1) Target port's originating message;
2) Initiator port response;
3) Target port response; and
4) Initiator port second response.
4.1.7.2 Initiator originated PPR negotiation

Figure 2 shows how the initiator port shall respond to various target port responses to an originating PPR OUT. The initiator port shall maintain the previous transfer agreement unless otherwise indicated.

**Figure 2. Initiator originated PPR negotiation: initiator response**
Figure 3 shows how the target port shall respond to various initiator port responses to a responding PPR IN. The target port shall maintain the previous transfer agreement unless otherwise indicated.

<table>
<thead>
<tr>
<th>2. Target port's responding message</th>
<th>3. Initiator port response</th>
<th>4. Target port second response</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPR IN</td>
<td>Attention condition; MESSAGE OUT phase</td>
<td>Reason: Initiator port claims target provided incompatible values. Target port shall invalidate its transfer agreement and originate a WDTR IN.</td>
</tr>
<tr>
<td></td>
<td>MESSAGE REJECT OUT</td>
<td>Reason: Initiator port detected a parity error. Target port may retry the PPR IN. If parity errors persist, target port shall create a bus free condition (unexpected). Retry outcome: Successful retry: see other responses Bus free condition: Target port shall invalidate its transfer agreement and originate a WDTR IN on the next connection if the initiator does not create an attention condition and originate negotiation itself.</td>
</tr>
<tr>
<td></td>
<td>MESSAGE PARITY ERROR OUT</td>
<td>Reason: Initiator port response cannot be determined. Target port may repeat the MESSAGE OUT phase and create a bus free condition if it gives up. Retry outcome: Successful retry: see other responses Bus free condition: Target port shall invalidate its transfer agreement and originate a WDTR IN on the next connection if the initiator port does not originate negotiation first.</td>
</tr>
<tr>
<td></td>
<td>Another message</td>
<td>Reason: Successful negotiation. Target port shall update its transfer agreement. If IU_REQ was changed, target port shall ignore the attention condition and create a bus free condition. [tgt cannot bus free immed after PPR IN - there might be a MRO or MPEO message]</td>
</tr>
<tr>
<td></td>
<td>No attention condition</td>
<td>Reason: Successful negotiation. Target port shall update its transfer agreement. If IU_REQ was changed, target port shall create a bus free condition.</td>
</tr>
</tbody>
</table>

Figure 3. Initiator originated PPR negotiation: target response
### 4.1.7.3 Initiator originated WDTR negotiation

Figure 4 shows how the initiator port shall respond to various target port responses to an originating WDTR OUT. The initiator port shall maintain the previous transfer agreement unless otherwise indicated.

<table>
<thead>
<tr>
<th>1. Initiator port’s originating message</th>
<th>2. Target port response</th>
<th>3. Initiator port response</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDTR OUT</td>
<td>MESSAGE IN Phase</td>
<td>Reason: Successful negotiation message exchange. The initiator port shall update it transfer agreement. If any values are illegal, the initiator port shall issue a MESSAGE REJECT OUT and invalidate its transfer agreement. The target port then originates negotiation.</td>
</tr>
<tr>
<td>WDTR IN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MESSAGE REJECT IN</td>
<td></td>
<td>Reason: Target port does not support WDTR. Initiator port shall invalidate its transfer agreement, create an attention condition and originate an SDTR OUT.</td>
</tr>
<tr>
<td>Bad parity</td>
<td></td>
<td>Reason: Target port response cannot be determined. Initiator port shall create an attention condition and send MESSAGE PARITY ERROR OUT. Target port may retry and creates a bus free condition when it gives up. Retry outcome: see other responses.</td>
</tr>
<tr>
<td>A message other than WDTR IN or MESSAGE REJECT IN</td>
<td></td>
<td>Reason: Target port faulty. Initiator port shall invalidate its transfer agreement, create an attention condition and originate an SDTR OUT.</td>
</tr>
<tr>
<td>BUS FREE phase</td>
<td></td>
<td>Reason: Target port detected an unrecoverable parity error, or target port faulty. Initiator port shall invalidate its transfer agreement, rearbitrate and select the target port, create an attention condition and originate an SDTR OUT.</td>
</tr>
<tr>
<td>Another bus phase</td>
<td></td>
<td>Reason: Target port faulty. Initiator port shall invalidate its transfer agreement, create an attention condition and originate an SDTR OUT.</td>
</tr>
<tr>
<td>No response*</td>
<td></td>
<td>Reason: Target port faulty. Initiator port should create a bus reset condition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* No REQ transitions within a timeout. The timeout value is not specified by this standard.</td>
</tr>
</tbody>
</table>

**Figure 4. Initiator originated WDTR negotiation: initiator response**
Figure 5 shows how the target port shall respond to various initiator port responses to a responding WDTR IN. The target port shall maintain the previous transfer agreement unless otherwise indicated.

![Diagram of WDTR negotiation]

Figure 5. Initiator originated WDTR negotiation: target response
4.1.7.4 Initiator originated SDTR negotiation

Figure 6 shows how the initiator port shall respond to various target port responses to an originating SDTR OUT. The initiator port shall maintain the previous transfer agreement unless otherwise indicated.

![Diagram of Initiator originated SDTR negotiation](image)

1. Initiator port's originating message
   - SDTR OUT

2. Target port response
   - MESSAGE IN Phase
     - SDTR IN
   - MESSAGE REJECT IN
     - Bad parity
     - A message other than SDTR IN or MESSAGE REJECT IN

3. Initiator port response
   - Reason: Successful negotiation message exchange. The initiator port shall update its transfer agreement. If any values are illegal, the initiator port shall issue a MESSAGE REJECT OUT and invalidate its transfer agreement. The target port then originates negotiation.
   - Reason: Target port does not support SDTR. The initiator port shall set its transfer agreement to the default transfer agreement.
   - Reason: Target port response cannot be determined. Target port does not support SDTR.
     - Initiator port shall create an attention condition and send MESSAGE PARITY ERROR OUT. Target port may retry and create a bus free condition when it gives up.
     - Retry outcome: see other responses.
   - Reason: Target port faulty. The initiator port shall set its transfer agreement to the default transfer agreement.
   - Reason: Target port detected an unrecoverable parity error, or target port faulty.
     - Initiator port shall set its transfer agreement to the default transfer agreement.
   - Reason: Target port faulty. The initiator port shall set its transfer agreement to the default transfer agreement.
   - Reason: Target port faulty. The initiator port should create a bus reset condition.

* no REQ transitions within a timeout. The timeout value is not specified by this standard.

Figure 6. Initiator originated SDTR negotiation: initiator response
Figure 7 shows how the target port shall respond to various initiator port responses to a responding SDTR IN. The target port shall maintain the previous transfer agreement unless otherwise indicated.

Figure 7. Initiator originated SDTR negotiation: target response
4.1.7.5 Target originated WDTR negotiation

Figure 8 shows how the target port shall respond to various initiator port responses to an originating WDTR IN. The target port shall maintain the previous transfer agreement unless otherwise indicated.

Figure 8. Target originated WDTR negotiation: target response
Figure 9 shows how the initiator port shall respond to various target port responses to a responding WDTR OUT. The initiator port shall maintain the previous transfer agreement unless otherwise indicated.

Figure 9. Target originated WDTR negotiation: initiator response
4.1.7.6 Target originated SDTR negotiation

Figure 10 shows how the target port shall respond to various initiator port responses to an originating SDTR IN. The target port shall maintain the previous transfer agreement unless otherwise indicated.

Figure 10. Target originated SDTR negotiation: target response
Figure 11 shows how the initiator port shall respond to various target port responses to a responding SDTR OUT. The initiator port shall maintain the previous transfer agreement unless otherwise indicated.

Figure 11. Target originated SDTR negotiation: initiator response

10.5 SELECTION phase
[change bars relative to SPI-4 revision 5 in the next sections. Revisions 1 and 2 of this proposal had incorrect section numbers.]

10.5.3 Selection without using attention condition
Information unit transfers enabled or disabled

If an initiator, when selecting without using an attention condition, detects an unexpected COMMAND phase it **should** invalidate all prior negotiations with the selected target. In this case, the initiator shall create an attention condition and on the corresponding MESSAGE OUT phase shall issue an ABORT
TASK message. On the next selection of the target that received the ABORT TASK message the initiator should do a selection using the attention condition and negotiate to enable information unit transfers.

16.3.1 Link control message codes

[In Table 54 - Link message codes, change the Yes description as follows:

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes=When sending the message in a MESSAGE OUT phase, the initiator shall clear the attention condition before the last ACK of the MESSAGE OUT phase. Init=initiator, Targ=target</td>
<td></td>
</tr>
</tbody>
</table>

16.3.12 PARALLEL PROTOCOL REQUEST message description

PARALLEL PROTOCOL REQUEST (PPR) messages (see Table 10) are used to negotiate the transfer period factor, REQ/ACK offset, transfer width exponent, and protocol options between two SCSI devices.

<table>
<thead>
<tr>
<th>Table 10. PARALLEL PROTOCOL REQUEST message format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
</tr>
<tr>
<td>Byte</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>PCOMP_EN</td>
</tr>
</tbody>
</table>

PPR messages shall be supported by ports supporting transfer period factors less than 0Ah or supporting any of the protocol options. PPR messages shall be supported by target ports with a CLOCKING field indicating DT support, IUS set to one, or QAS set to one in the INQUIRY page of all their logical units (see SPC-2).

Usage of this message is defined in 4.1. Fields are defined in 4.1.4.

16.3.16 SYNCHRONOUS DATA TRANSFER REQUEST message description

SYNCHRONOUS DATA TRANSFER REQUEST (SDTR) messages (see Table 11) are used to negotiate the transfer period factor and REQ/ACK offset between two SCSI devices.

<table>
<thead>
<tr>
<th>Table 11. SYNCHRONOUS DATA TRANSFER REQUEST message format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

SDTR messages shall be supported by devices supporting synchronous data transfers (i.e., non-zero REQ/ACK offsets). SDTR messages shall be supported by target ports with SYNC set to one in the INQUIRY page of all their logical units (see SPC-2).

Usage of this message is defined in 4.1. Fields are defined in 4.1.4.
16.3.18 WIDE DATA TRANSFER REQUEST message description

WIDE DATA TRANSFER REQUEST (WDTR) messages (see Table 12) are used to negotiate the transfer width exponent between two SCSI devices.

Table 12. WIDE DATA TRANSFER REQUEST message format

<table>
<thead>
<tr>
<th>Byte</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Extended Message (01h)</td>
</tr>
<tr>
<td>1</td>
<td>EXTENDED MESSAGE LENGTH (02h)</td>
</tr>
<tr>
<td>2</td>
<td>Wide Data Transfer Request (03h)</td>
</tr>
<tr>
<td>3</td>
<td>TRANSFER WIDTH EXPONENT [&quot;m&quot; removed]</td>
</tr>
</tbody>
</table>

WDTR messages shall be supported by ports supporting wide data transfers (i.e., non-zero transfer width exponents). WDTR messages shall be supported by target ports with WBUS16 set to one in the INQUIRY page of all their logical unit (see SPC-2).

Usage of this message is defined in 4.1. Fields are defined in 4.1.4.
Suggested changes to SPI-4 revision 6 to match this proposal

All uses of the following phrases not already addressed by the core proposal are excerpted below with suggested changes to bring terminology in line with the core proposal.

- agreement
- 16.3.12
- 16.3.16
- 16.3.18
- data transfer (not all uses are excerpted)

3.1.71 **pCRC protection**: The use of CRC to detect DT DATA phase data transmission errors during **parallel-data group** transfers.

4.8 Clocking methods for data transfers

Figure 6 - **ST synchronous ST-data** transfer example
Figure 7 - **DT synchronous DT-data** transfer example
Figure 8 - Paced **DT-data** transfer example

4.10.1 Data transfer modes

[change bars relative to SPI-4 revision 5 in the next sections. Revisions 1 and 2 of this proposal had incorrect section numbers.]

4.10.1.1 Asynchronous transfers

SCSI parallel interface devices default to 8-bit asynchronous transfers.

The 8-bit asynchronous transfers are used for all **COMMAND, STATUS, and MESSAGE phases**, information transfers except **DATA phases**.

ST DATA phases **may use 8-bit or 16-bit asynchronous transfers** when a **wide transfer agreement (see 10.8.5)** is in effect. Asynchronous transfers are not permitted when in **DT DATA phases are enabled**.

4.10.1.2 Synchronous transfers

Synchronous transfers shall only be used in **DATA phases** when a synchronous transfer agreement is in effect (see **10.8.5**).

ST DATA phases shall **transfer data using-use** synchronous transfers when a **ST-DATA synchronous phase enabled-transfer** agreement is in effect. ST DATA phases may use 8-bit or 16-bit synchronous transfers when a **wide transfer agreement (see 10.8.5)** is in effect.

DT DATA phases shall **transfer data using-use** synchronous transfers when a **DT-DATA synchronous phase enabled-transfer** agreement (see **10.8.5**) is in effect. DT DATA phases shall only use 16-bit **synchronous transfers**.

4.10.1.3 Paced transfers

Paced transfers shall only be used in **DT DATA phases** when a **fast-160-paced transfer** agreement is in effect (see **10.8.5**). DT DATA phases shall only use 16-bit **paced**-transfers.

4.10.3.1 DT DATA phase parallel transfers format

The format of the data transmitted during DT DATA phases is dependent on the negotiated protocol **options**.

If **data group transfers** are enabled then all data and protection are transmitted in data groups. If information unit transfers are enabled then all nexus, task management, task attribute, command, data, and protection are transmitted in SPI information units.
4.10.3.3 Information unit transfers
[Lots of red and blue here so excerpt not shown. Change “negotiated agreement” to “negotiated transfer agreement”]

4.11 Negotiation
Editors Note 1 - GOP: Put all the rules for target/initiator negotiations in this section including the REQ(ACK) offset definition from 99-295r5.

REQ/ACK offset: For ST DATA transfers the REQ/ACK offset is the number of REQ assertions that may be sent by the target in advance of the number of ACK assertions received from the initiator.

For DT DATA transfers not using paced transfers the REQ/ACK offset is the number of REQ transitions that may be sent by the target in advance of the number of ACK transitions received from the initiator.

For paced DT DATA IN transfers the REQ/ACK offset is the number of data valid state REQ assertions (see 10.7.4.3) that may be sent by the target in advance of ACK assertions received from the initiator.

For paced DT DATA OUT transfers the REQ/ACK offset is the number of REQ assertions that may be sent by the target in advance of the number of data valid state ACK assertions (see 10.7.4.3) received from the initiator.

7.2.2 SE output characteristics
While active-negation drivers may be used on any non-OR-tied signal (see 8.4), their usage is particularly valuable on the ACK and REQ signals, because these signals are vulnerable to glitches on the transition that could lead to false ACK or REQ detection. Additional benefit may be achieved by using active-negation drivers on the DATA BUS, P_CRCA, and DB(P1) signals when operating in fast synchronous data transfer mode by reducing the skews between the first group of signals (ACK and REQ) and the DATA BUS, P_CRCA, and DB(P1).

8 SCSI bus signals
8.1 SCSI bus signals overview
Information transfer on the SCSI bus is allowed between only two SCSI devices at any given time except during MESSAGE IN phase when QAS is enabled (see 16.3.12). All SCSI devices that have QAS enabled are required to monitor messages during a MESSAGE IN phase for a QAS REQUEST message. The maximum number of SCSI devices is determined by the width of the data path implemented and restrictions in clause 7. The SCSI devices may be any combination of initiators and targets, provided there is at least one of each.

8.2 Signal descriptions

P_CRCA (PARITY/CRC AVAILABLE) . A signal indicating either parity or CRC available based on bus phase and negotiated settings.

When referred to as DB(P_CRCA) it is a signal sourced by the SCSI device driving the DATA BUS during the SELECTION phase, RESELECTION phase, ST DATA phase, COMMAND phase, MESSAGE phase, and STATUS phase. This signal is referred to as DB(P_CRCA) and is sourced by the SCSI device driving the DATA BUS. The DB(P_CRCA) signal is associated with the DB(7-0) signals and is used to detect the presence of an odd number of bit errors within the byte. The DB(P_CRCA) bit is driven such that the number of logical ones in the byte plus the parity bit is odd.

When referred to as P_CRCA and data group transfers are enabled (see 16.3.12) it is a signal sourced by a target during DT DATA phases when data group transfers are enabled this signal is referred to as P_CRCA and is sourced by the target to control whether a data group field is a pad field, pCRC field, or data field (see 10.7.3.3.5). When asserted the data group field shall be pad or pCRC fields that shall not be transferred to the application client. When negated the data group field shall be a data field that shall be transferred to the application client.
During DT DATA phases when referred to as P_CRCA and information unit transfers are enabled it this signal is referred to as P_CRCA and is a signal sourced by the target. Depending on the negotiated condition of the read streaming and write flow control the initiator and target usage for P_CRCA is different. When information unit transfers are enabled the target and initiator shall use the P_CRCA signal as indicated in table 29.

P1 (PARITY 1). A signal normally sourced by the SCSI device driving the DATA BUS. The P1 signal is associated with the DB(15-8) signals and is used to detect the presence of an odd number of bit errors within the byte. The P1 bit is driven such that the number of logical ones in the byte plus the P1 bit is odd.

During the ST DATA phase with transfer length set for 8 bit transfers, when narrow transfers, and during COMMAND phase, MESSAGE phase, and STATUS phase the P1 signal shall not be driven by any SCSI device.

During the SELECTION phase and the RESELECTION phase on a 16-bit wide bus the P1 signal shall be sourced by the SCSI device driving the DATA BUS.

During DT DATA phases when data group transfers are enabled (see 16.3.12) the P1 signal shall be continuously negated by the SCSI device driving the DB(15-0) signals and shall be ignored by the SCSI device receiving the DB(15-0) signals during DT DATA phases.

When information unit transfers and DT synchronous transfers are enabled the P1 signal shall be continuously negated by the SCSI device driving the DB(15-0) signals and shall be ignored by the SCSI device receiving the DB(15-0) signals during DT DATA phases.

When information unit transfers and paced transfers are enabled the P1 signal shall be sourced by the SCSI device driving the DATA BUS. The P1 signal is used to indicate whether the data valid or data invalid state during paced transfers (see 10.7.4.3).

9.1 SCSI parallel bus timing values
The “Timing values” columns in tables 32, 33, 34, and 35 should be renamed “Timing values for transfer rate”.

9.2.16 pCRC receive hold time
The minimum time required at the receiver between the transition of the REQ signal and the transition of the P_CRCA signal while pCRC protection is enabled (see 16.3.12) during data group transfers.

9.2.17 pCRC receive setup time
The minimum time required at the receiver between the transition of the P_CRCA signal and the transition of the REQ signal while pCRC protection is enabled (see 16.3.12) during data group transfers.

Specified to ease receiver timing requirements and ensure that this signal, which is outside CRC protection, is received correctly.

9.2.18 pCRC transmit hold time
The minimum time provided by the transmitter between the transition of the REQ signal and the transition of the P_CRCA signal while pCRC protection is enabled (see 16.3.12) during data group transfers.

9.2.19 pCRC transmit setup time
The minimum time provided by the transmitter between the transition of the P_CRCA signal and the transition of the REQ signal while pCRC protection is enabled (see 16.3.12) during data group transfers.

9.2.24 QAS arbitration delay
The minimum time a SCSI device with QAS enabled (see 16.3.12) shall wait from the detection of the MSG, C/D, and I/O signals being false to start QAS until the DATA BUS is examined to see if QAS has been won (see 10.4).
9.2.28 **Receive assertion period**
The minimum time provided at a SCSI device receiving a REQ signal for the signal to be asserted while using synchronous transfers, provided P_CRCA is not transitioning with pCRC protection enabled (see 16.3.12)during data group transfers. Also, the minimum time required at a SCSI device receiving an ACK signal for the signal to be asserted while using synchronous transfers. For SE fast-5 and fast-10 operation, the time period is measured at the 0.8 V level. For SE fast-20 operation the period is measured at the 1.0 V level. For LVD see figure 53 and figure 54 for signal measurement points.

9.2.29 **Receive hold time**
For ST data transfers the minimum time provided at the receiving SCSI device between the assertion of the REQ signal or the ACK signals and the changing of the DATA BUS, DB(P_CRCA), and/or DB(P1) signals while using synchronous transfers, provided P_CRCA is not transitioning with pCRC protection enabled (see 16.3.12)during data group transfers.

For DT data transfers the minimum time required at the receiving SCSI device between the transition (i.e. assertion or negation) of the REQ signal or the ACK signals and the changing of the DATA BUS, DB(P_CRCA), and/or DB(P1) signals while using synchronous transfers.

9.2.35 **Receive REQ assertion period with P_CRCA transitioning**
The minimum time required at a SCSI device receiving a REQ signal for the signal to be asserted while using synchronous data group transfers with P_CRCA transitioning with pCRC protection enabled (see 16.3.12).

Specified to ensure that the assertion period is longer than the receive hold time plus the receive setup time.

9.2.36 **Receive REQ negation period with P_CRCA transitioning**
The minimum time required at a SCSI device receiving a REQ signal for the signal to be negated while using synchronous data group transfers with P_CRCA transitioning with pCRC protection enabled (see 16.3.12).

9.2.39 **REQ (ACK) period**
The REQ (ACK) period during synchronous transfers, specified in table 32 for ST DATA phases and in table 33 for DT DATA phases, is the nominal time between adjacent assertion edges of the REQ or ACK signal for the fastest negotiated data transfer rate. For the purpose of calculating the actual REQ (ACK) period tolerance the REQ (ACK) period should be measured without interruptions (e.g., offsets pauses). To minimize the impact of cross-talk and ISI the measurements should be made by averaging the time between edges during long (i.e., greater than 512 bytes) all zero or all ones data transfers and by ignoring the first and last 10 transitions.

9.2.53 **Transmit assertion period**
The minimum time that a target shall assert the REQ signal while using synchronous transfers, provided it is not transitioning P_CRCA with pCRC protection enabled (see 16.3.12)during data group transfers. Also, the minimum time that an initiator shall assert the ACK signal while using synchronous transfers.

9.2.56 **Transmit negation period**
The minimum time that a target shall negate the REQ signal while using synchronous transfers, provided it is not transitioning P_CRCA with pCRC protection enabled (see 16.3.12)during data group transfers. Also, the minimum time that an initiator shall negate the ACK signal while using synchronous transfers.

9.2.59 **Transmit REQ assertion period with P_CRCA transitioning**
The minimum time that a target shall assert the REQ signal during a synchronous data group transfer DT DATA phase while transitioning P_CRCA with pCRC protection enabled (see 16.3.12).

Specified to provide the increased receive REQ assertion period, subject to loss on the interconnect.

9.2.60 **Transmit REQ negation period with P_CRCA transitioning**
The minimum time that a target shall negate the REQ signal during a synchronous-data-group transfer DT DATA phase while transitioning P_CRCA with pCRC protection enabled (see 16.3.12).

Specified to provide the increased receive REQ negation period, subject to loss on the interconnect.

9.6.2 DT data transfer calculations
[In figure 66:] Consequently transmit setup and transmit hold time should be measured with an alternating pattern at the negotiated data rate-transfer period with all data signals other than that being measured held at a continual assertion or negation.

10.3 Expected and unexpected bus free phases
...

10.4.1 Arbitration and QAS overview
Arbitration allows one SCSI device to gain control of the SCSI bus to allow that SCSI device to initiate or resume a task.

There are two methods that a SCSI device may use to arbitrate for the SCSI bus: normal arbitration and QAS. Normal arbitration is mandatory and requires the detection of a BUS FREE phase on the SCSI bus before starting. QAS is optional and, when enabled (see 16.3.12), requires the detection of a QAS REQUEST message (see 16.3.13) before starting.

10.4.3 QAS protocol
QAS allows a target that has information unit transfers enabled (see 16.3.12) with an information unit transfer agreement in effect and QAS enabled (see 16.3.12) that is currently connected to an initiator that has QAS enabled to transfer control of the bus to another SCSI device that has QAS enabled without an intervening BUS FREE phase. SCSI devices that participate in QAS arbitration shall report that capability in the INQUIRY command.

Before an initiator may use QAS that initiator shall negotiate, using the PPR message, the use of the QAS phase with each target that has indicated support of QAS. Any time the data transfer agreement is in an indeterminate state (see 16.3.12) an initiator has an invalid transfer agreement, that initiator shall renegotiate to enable QAS before the initiator may use QAS that initiator shall renegotiate to enable QAS. SCSI devices that support QAS shall implement the fairness algorithm (see Annex B) during all QAS arbitrations. SCSI devices shall negotiate the use of QAS with a particular SCSI device before using QAS to select or reselect that SCSI device. Also, targets shall have negotiated the use of QAS with a particular initiator before using QAS REQUEST message to do a physical disconnect from that initiator, and initiators shall have negotiated the use of QAS with a particular target before accepting a QAS REQUEST message from that target. If an initiator receives a QAS REQUEST message from a target that has not negotiated the use of QAS, then the initiator shall create an attention condition for the QAS REQUEST message, and shall report MESSAGE REJECT on the following MESSAGE OUT phase.
...

10.5.2 Selection using attention condition
10.5.2.2 Information unit transfers disabled
If information unit transfers are disabled (see 16.3.12) for the connecting initiator the target shall follow the phase sequences defined in 13.22.1.

10.5.2.3 Information unit transfers enabled
If information unit transfers are enabled (see 16.3.12) for the connecting initiator the target shall proceed to a MESSAGE OUT phase (see 13.53.1). On detecting the MESSAGE OUT phase the initiator shall begin a PPR negotiation. On completion of the PPR negotiation the target shall proceed to a BUS FREE phase. If the first message received by the target during the MESSAGE OUT phase is not a task
management message or a PPR message the target shall change to a MESSAGE IN phase and issue a MESSAGE REJECT message followed by a WDTR message with TRANSFER WIDTH EXPONENT field set to 00h. If the target does not support the WDTR message it shall follow the MESSAGE REJECT message with a SDTR message with the REQ/ACK OFFSET field set to 00h.

10.5.3 Selection without using attention condition

10.5.3.1 Information unit transfers disabled or enabled

If information unit transfers are enabled (see 16.3.12) for the connecting initiator the target shall follow the phase sequences defined in 13.4.3.2.

If information unit transfers are disabled (see 16.3.12) for the connecting initiator the target shall follow the phase sequences defined in 13.3.2.2.

If an initiator, when selecting without using an attention condition, detects an unexpected COMMAND phase it should invalidate all prior negotiations its transfer agreement with the selected target. In this case, the initiator shall create an attention condition and on the corresponding MESSAGE OUT phase shall issue an ABORT TASK message. On the next selection of the target that received the ABORT TASK message the initiator should do a selection using the attention condition and negotiate to enable information unit transfers.

10.7.1 Information transfer phases overview

The information transfer phases use one or more REQ or ACK handshakes to control the information transfer. Each REQ/ACK handshake allows the transfer of 8- or 16-bits of information depending on the negotiated data transfer width exponent (see 16.3.18). During the information transfer phases the BSY signal shall remain true and the SEL signal shall remain false, except to indicate a paced transfer training pattern is going to occur (see 10.7.4.2). Additionally, during the information transfer phases, the target shall continuously envelope the REQ or ACK handshake(s) with the C/D, I/O, and MSG signals in such a manner that these control signals are valid for one bus settle delay before the assertion of the REQ signal of the first handshake and remain valid until after the negation of the ACK signal at the end of the handshake of the last transfer of the phase.

10.7.3 Synchronous transfer

10.7.3.1 Synchronous transfer overview

Synchronous transfer is optional and is only used in DATA phases. It shall be used in a DATA phase if a synchronous transfer agreement has been established (see 16.3.16). The transfer agreement specifies the REQ/ACK offset and the minimum transfer period.

10.7.3.2 ST synchronous transfer

When a ST data transfer agreement has been established (see 4.1.4.2) the target shall only use the ST DATA IN phase and ST DATA OUT phase for data transfers.

10.7.3.3 DT synchronous transfer overview

When a DT data transfer agreement has been established (see 4.1.4.2) the target shall only use the DT DATA IN phase and DT DATA OUT phase for data transfers.

10.7.3.3.2 Information unit transfer

When an information unit transfer agreement has been established (see 16.3.12); the target shall use the following phases for transfer.

10.7.3.3.3 Data Group Pad field and pCRC field transfer to target

If the I/O signal is false (transfer to the target), the initiator determines the data field transfer is complete by detecting an assertion of the P_CRC signal. If the REQ signal is asserted (i.e., pad field required) the initiator shall first transfer the two pad bytes, then the four pCRC bytes. If the REQ signal is negated (i.e., no pad field required) the initiator shall transfer the four pCRC bytes.

Pad field data and pCRC field data are transferred using the same negotiated values as the data field data.
The target may continue to send REQs, up to the negotiated REQ/ACK offset, for the next data group. The target shall not transition REQ with P_CRCA asserted until the initiator has responded with all ACK transitions for the previous data group.

10.7.4 Paced transfer
10.7.4.1 Paced transfer overview
If a paced transfer agreement has been established it shall be used in DT DATA phase and information unit transfers shall be used. The transfer agreement also specifies the REQ/ACK offset and the transfer period (see 16.3.12).

10.7.4.2.1 Training pattern overview
If the retain training information is disabled a training pattern shall be transferred at the start of the first DT data phase for each data transfer direction after each physical connect and physical reconnect. The training pattern shall not be transferred again until after a physical disconnection occurs.

If the retain training information is enabled a training pattern shall be transferred at the start of the first DT data phase for each data transfer direction after the retain training information is enabled. The SCSI device shall save training configuration values for each I_T nexus that has negotiated to retain training information. The SCSI device shall use the saved training configuration values for all paced transfers. The target may retrain an I_T nexus if it determines the training configuration values are invalid, without having to renegotiate the retain training information protocol option.

10.7.4.2.3 DT DATA OUT phase training pattern
... the target shall begin asserting and negating REQ to indicate to the initiator valid data may be sent. The number of REQ assertions shall not exceed the negotiated REQ/ACK offset.

10.7.5 Wide data transfer
Wide data transfers shall be used for DT DATA phases. Wide data transfers may shall be used in the ST DATA phase only if a non-zero wide data transfer agreement is in effect (see 16.3.18 or 16.3.124.1.4.5). These messages determine the use of wide mode by both SCSI devices and establish a data path width to be used during the ST DATA phase.

A wide data transfer of 16 bits may be established. All SCSI devices shall support 8-bit narrow data transfers.

During 8-bit narrow transfers, all information shall be transferred in bytes across the DB(7-0) and DB(P_CRCA) signals on the SCSI bus. At the receiving SCSI device the DB(15-8) (if present) and DB(P1) (if present) signals are undefined.

During 16-bit wide transfers, the first and second information bytes for each DATA phase shall be transferred across the DB(7-0) and DB(15-8) signals, respectively, on the SCSI bus. Subsequent pairs of information bytes are likewise transferred in parallel across the SCSI bus (see table 38).

The IGNORE WIDE RESIDUE message may be used to indicate that the last byte of a data field or the data byte of an information unit is undefined.

11.2 ST DATA BUS protection (parity)
For ST DATA phases the DB(P_CRA) signal shall indicate odd parity for DB(7-0). If 8-bit narrow transfers are enabled the DB(P1) signal shall not be checked. If 16-bit data wide transfers are enabled the DB(P1) signal shall indicate odd parity for DB(15-8). If 16-bit wide transfers are enabled and the last information byte transferred does not fall on the DB(15-8) signals the DB(P1) signal shall be valid for whatever data is placed on the bus.
12.4 Hard reset
Any SCSI device that detects a hard reset shall also set its transfer agreement to the default transfer agreement (see 4.1.4.2):

a) set the data transfer width to eight-bit transfer mode,
b) set the data transfer mode to asynchronous transfer mode, and
c) set to zero all the protocol options bits (see 16.3.12).

13 SCSI bus phase sequences
13.1 SCSI bus phase sequences overview

13.2 Phase sequences with information units disabled

13.2.1 Phase sequences for physical reconnection and selection using attention condition with information unit transfers disabled
The allowable sequences for either physical reconnection or a selection using attention condition and physical reconnection while a transfer agreement is in effect that disables information unit transfers are not in effect shall be as shown in figure 70.

If a data transfer agreement is in effect that disables information unit transfers (see 16.3.12), the normal progression for selection using attention condition (see 10.5.2) is from:
1) from the BUS FREE phase to ARBITRATION;
2) from ARBITRATION to SELECTION or RESELECTION; and
3) from SELECTION or RESELECTION to one or more of the information transfer phases (i.e., COMMAND, DATA, STATUS, or MESSAGE).

The final information transfer phase is normally the MESSAGE IN phase where a DISCONNECT, or TASK COMPLETE message is transferred, followed by the BUS FREE phase.

[figure]

Figure 70 - Phase sequences for physical reconnection or selection using attention condition with information unit transfers disabled

13.2.2 Phase sequences for selection without using attention condition with information unit transfers disabled
The additional sequences for a selection without using attention condition while an information unit transfer agreement is not in effect that disables information unit transfers shall be as shown in figure 71.

If a data transfer agreement is in effect that disables information unit transfers (see 16.3.12), the normal progression for selection without using attention condition (see 10.5.3) is:
1) from the BUS FREE phase to ARBITRATION;
2) from ARBITRATION to SELECTION;
3) from SELECTION to COMMAND phase;
4) from COMMAND phase to DATA phase;
5) from DATA phase to STATUS phase; and
6) from STATUS phase to MESSAGE IN phase, where a TASK COMPLETE message is transferred; and
7) from MESSAGE IN to , followed by the BUS FREE phase.

[figure]

Figure 71 - Phase sequences for selection without using attention condition with information unit transfers disabled

13.3 Phase sequences with information units enabled
13.4 Phase sequences for physical reconnection or selection without using attention condition with information unit transfers enabled
The sequences for physical reconnection or a selection without using attention condition and physical reconnection while an data information unit transfer agreement is in effect that enables information unit transfers shall be as shown in figure 72.

If a data transfer agreement is in effect that enables information unit transfers (see 16.3.12), the normal progression, if QAS is disabled, for selection without using attention condition (see 10.5.3) if QAS is disabled is:
1) from the BUS FREE phase to ARBITRATION;
2) from ARBITRATION to SELECTION or RESELECTION;
3) from SELECTION or RESELECTION to one or more DT DATA phases; and
4) the final DT DATA phase is followed by the BUS FREE phase.

Figure 72 - Phase sequences for physical reconnection or selection without using attention condition/physical reconnection and with information unit transfers enabled

13.5.3.2 Phase sequences for physical selection using attention condition with information unit transfers enabled
The sequences for a selection with attention condition and physical reconnection while an data information unit transfer agreement is in effect that enables information unit transfers shall be as shown in figure 73.

If a data transfer agreement is in effect that enables information unit transfers (see 16.3.12), the normal progression, if QAS is disabled, for selection using attention condition (see 10.5.2.3) if QAS is disabled is:
1) from the BUS FREE phase to ARBITRATION;
2) from ARBITRATION to SELECTION;
3) from SELECTION to MESSAGE OUT;
4) from MESSAGE OUT to MESSAGE IN; and
5) from MESSAGE IN to BUS FREE phase.

Figure 73 - Phase sequences for selection with attention condition/physical reconnection and with information unit transfers enabled

14.1 SPI information unit overview
After any negotiation message (i.e., PPR or WDTR or SDTR) that results in the information unit transfer agreement being changed, the target shall abort all tasks, except the current task, for the initiator participating in the negotiation.

When an data information unit transfer agreement is in effect that enables information unit transfers there is no option equivalent to the "physical disconnect without sending a SAVE DATA POINTERS message."
The initiator shall save the data pointers as soon as the last byte of the last iuCRC for a SPI information unit is transferred. The save shall occur even if the initiator detects an error in the SPI data information unit.

16.2.5 Extended messages
The EXTENDED MESSAGE ARGUMENTS are specified within the extended message descriptions (see 16.3.9, 16.3.10, 16.3.12, 16.3.16, and 16.3.18).

16.5.6 LOGICAL UNIT RESET
A logical unit reset has no effect on the parallel protocol agreement, wide transfer agreement, or data transfer agreement.
17.2 Asynchronous event notification
The asynchronous event notification protocol shall be used only with SCSI devices that return processor
device type with an AERC bit of one in response to an INQUIRY command. The INQUIRY command
should be sent to logical unit zero of each SCSI device responding to selection. This procedure shall be
conducted prior to the first asynchronous event notification and shall be repeated whenever the SCSI
device deems it appropriate or when an event occurs that may invalidate the current information. (See
SYNCHRONOUS DATA TRANSFER REQUEST message (16.3.16) for examples of these events.)

18.1.4 Port control mode page
18.1.4.1 Port control mode page overview
The port control mode page (see table 80 and table 81) contains those parameters that affect SPI SCSI
device port operation options. The page shall be implemented by LUN 0 of all SPI SCSI devices. The
page shall not be implemented by logical units other than LUN 0. The implementation of any bit and its
associated functions is optional. The page follows the MODE SENSE / MODE SELECT rules specified by
SCSI Primary Commands-2 standard.

The target shall maintain an independent set of port control mode page parameters for each initiator. The
parameters saveable bit in the mode page format header returned with MODE SENSE command shall be
set to zero if the long mode page format is being used (i.e., LONG bit set to one), indicating the parameters
are not saved through resets.

After a MODE SELECT command, parameter settings shall remain in effect until either:
a) settings are changed by another MODE SELECT command,
b) a logical unit reset of LUN 0 occurs,
c) an SDTR negotiation successfully completes,
d) a WDTR negotiation successfully completes, or
e) a PPR negotiation successfully completes with the MAINTAIN MARGIN CONTROL SETTINGS
   HOLD_MCS bit set to zero.

18.1.4.2 Saved training configuration values subpage
The saved training configuration values subpage is used to report the SCSI device’s saved training
configuration values. These vendor specific values are maintained by the SCSI device when the retain
training information option is enabled (see 16.3.12.1 4.1.4.6.8). The fields are listed in table 83 however
the content of the fields is vendor specific.

18.1.4.3 Negotiated settings subpage
The TRANSFER PERIOD FACTOR field indicates the negotiated transfer period factor (see
16.3.12.1 4.1.4.3) for the current I_T nexus.

The REQ/ACK OFFSET field indicates the negotiated REQ/ACK offset (see 16.3.12.1 4.1.4.4) for the
current I_T nexus.

The TRANSFER WIDTH EXPONENT field indicates the negotiated transfer width exponent (see
16.3.12.1 4.1.4.5) for the current I_T nexus.

The PROTOCOL OPTIONS BITS field contain the negotiated protocol options (see 16.3.12.1 4.1.4.6) for
the current I_T nexus.

The RECEIVED PCOMP_EN bit contains the value of the PCOMP_EN bit (see 16.3.12.1 4.1.4.6.9)
received by the target for the current I_T nexus.

The SENT PCOMP_EN bit contains the value of the PCOMP_EN bit (see 16.3.12.1 4.1.4.6.9) sent by the
target for the current I_T nexus.
C.7 SCSI ID selection

Table C.6 - SCSI device ID selection signals

Addresses in the range of 8 to 15 are only supported by SCSI devices implementing wide-a 16-bit data transfers DATA BUS (see 8.2).

F.9 Special performance considerations for SCSI domains with simple expanders

Table F.3

Rounded up to the next multiple of 4 - see 16.3.12.1.4.4

G.4 Enabling ECP

Following a hard reset (see 12.4), 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 the default transfer agreement to with some target; and
2) issue a WRITE BUFFER command to that same target with the MODE field set to echo buffer plus enable ECP (see SCSI Primary Commands-2 standard).

The initiator may disable ECP by:

1) negotiating to asynchronous transfer mode with a transfer width of 8 bits to the default transfer agreement with any some target; and
2) issuing a WRITE BUFFER command to that same target with the MODE field bit set to disable ECP (see SCSI Primary Commands-2 standard).

G.6.3.4 REPORT CURRENT STATUS SEDB

The NEAR RECEIVED PCOMP_EN bit and the FAR RECEIVED PCOMP_EN contain the last received value for PCOMP_EN bit on the corresponding ports. The NEAR SENT PCOMP_EN bit and the FAR SENT PCOMP_EN bit contain the last sent values for PCOMP_EN bit on the corresponding ports. For initiators, only the far port values are defined. The values returned are from the most recent PPR negotiation that resulted in a synchronous or paced data transfer agreement.

G.6.3.3 REPORT CAPABILITIES

The MINIMUM TRANSFER PERIOD FACTOR field shall be set to the smallest value of the TRANSFER PERIOD FACTOR (see 16.3.12.14.1.4.3) supported by the expander.

The MAXIMUM REQ/ACK OFFSET field shall be set to the largest value of the REQ/ACK OFFSET (see 16.3.12.14.1.4.4) supported by the expander.

The MAXIMUM TRANSFER WIDTH EXPONENT field shall be set to the largest value of the TRANSFER WIDTH EXPONENT (see 16.3.12.14.1.4.5) 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 16.3.12.14.1.4.6).

G.6.6 REPORT SAVED TRAINING CONFIGURATION VALUES

The REPORT SAVED TRAINING CONFIGURATION VALUES function is used to report the SCSI device’s saved training configuration values. These vendor specific values are maintained by the SCSI device when the retain training information option is enabled (see 16.3.124.1.4.6.8).

G.7 Data Transfer Requirements
The communicative expander functions shall only be performed when the default transfer agreement is in effect when the data transfer agreement is 8-bit asynchronous. For any other data transfer agreement, the communicative expander shall operate as a simple expander.

**N.1.2 Integrity checking:** The act of verifying that the physical layer is able to transfer test data at the negotiated speed-transfer rate and transfer width between the initiator and target (i.e., a quick check for physical domain validation). For example, two wide SCSI devices connected with a narrow cable will discover that the cable does not support wide transfers during this checking. These SCSI devices will then re-negotiate to narrow transfers.

... b) expander errors (e.g., expanders not capable of the negotiated data-transfer rate);
...