To: T10 Technical Committee
From: Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)
Date: 16 April 2001
Subject: SPI-4 negotiation message rewrite

Revision History
Revision 0 (16 April 2001) first revision released to T10.

Related Documents
spi4r04 – SCSI Parallel Interface – 4 revision 4

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 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.

Suggested Changes
Editor's directions: Add the new model section to chapter 4. Remove sections 16.3.10 PPR, 16.3.14 SDTR, and 16.3.16 WDTR. Add a new Negotiation section 16.x parallel to the Link section with subsections for PPR, SDTR, and WDTR.

The text was originally pulled from SPI-4 revision 0; some changes made through revision 4 may have been missed.

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 negotiate a transfer agreement between two ports (see 3.1.76). The transfer agreement includes a data transfer agreement that defines the protocol used during DATA phases (e.g. transfer period, REQ/ACK offset, transfer width) and agreement on other features (e.g. QAS).

The default transfer agreement is eight-bit asynchronous data transfers with protocol options set to 00h. All other information transfer phases (COMMAND, MESSAGE, and STATUS) shall use eight-bit asynchronous data transfers.

When an initiator sends one of these messages, the message names are PPR OUT, SDTR OUT, and WDTR OUT. When a target sends one of these messages, the message names are PPR IN, SDTR IN, and WDTR IN.

A transfer agreement applies to the pair of ports that negotiate the agreement. 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). All communication that occurs though the port shall use the negotiated transfer agreement regardless of whether the port is acting as a target port or an initiator port (e.g. if port 7 is used as an initiator port and port 0 is used as a target port, the transfer agreement also applies when port 7 is used as a target port and port 0 is used as an initiator port).

Negotiations are maintained on a target port basis. All logical units in a target share the same transfer agreement.
[previous text said "A … agreement applies to all logical units of the two SCSI devices that negotiated agreement", but an initiator is not a "logical unit."]
[previous text said "That is, if SCSI device A acts as an initiator to negotiate a data transfer agreement with SCSI device B which acts as a target, then the same data transfer agreement applies even if SCSI device B changes to an initiator and SCSI device A changes to a target." This example was rewritten.]

A transfer agreement only applies to the two ports that negotiate the agreement. Separate transfer agreements may be negotiated for each pair of ports.

4.1.2 Negotiation algorithm

Negotiation is done using message pairs exchanged between an initiator port and target port prior to exchange of a command. The originating port is the one that sends the first 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 illegal values, the original port shall respond with a MESSAGE REJECT.

The valid error-free negotiation message sequences are shown in

Figure 1. A description of all possible message sequences is in section 4.1.7.

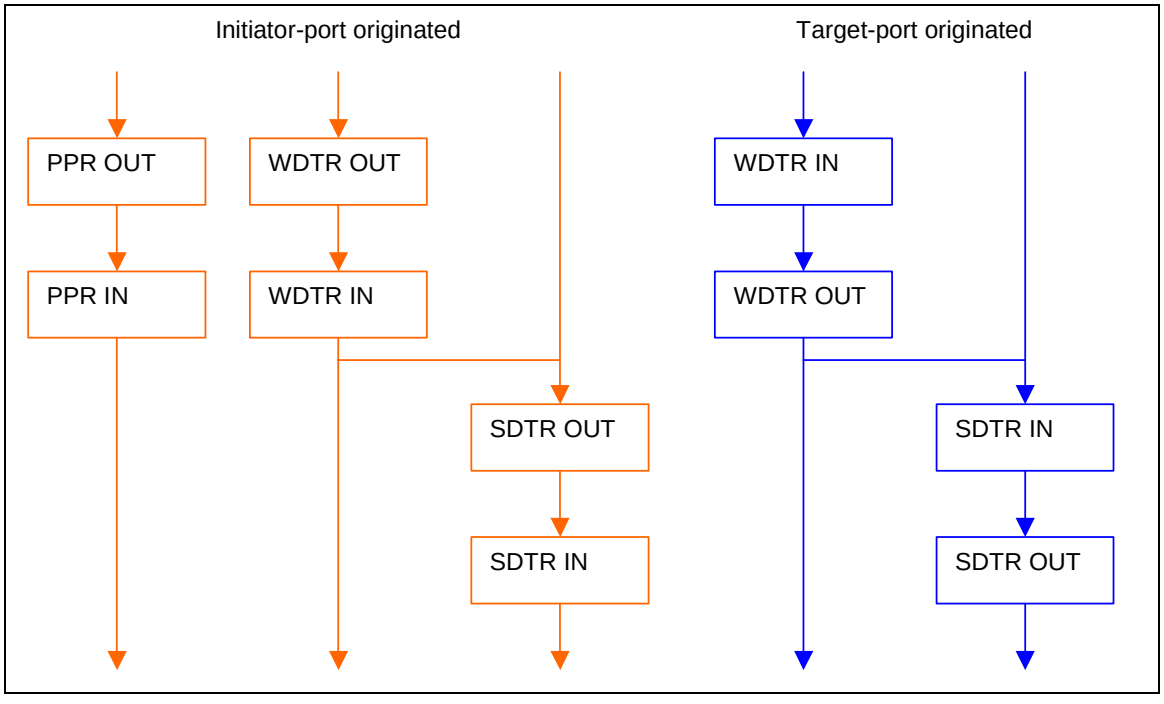

Figure 1. Error-free negotiation message sequences.

4.1.3 When to negotiate

A transfer agreement stays in effect until another successful negotiation message sequence occurs or the port detects that the agreement may have become invalid. The transfer agreement becomes invalid after a reset event or an error occurs while transmitting a responding negotiation message. A logical unit reset has no effect on a transfer agreement.

An initiator port shall originate negotiation before sending a command whenever it does not have a valid transfer agreement. A target port shall originate negotiation before accepting a command whenever it does not have a valid 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. Because ports remember their transfer agreements between connections, negotiation is unnecessary and performance impact from extra negotiations is likely.

[changed “initiated” to “originated” everywhere]
[previous text: Renegotiation after every selection is not recommended, since a significant performance impact is likely. This was a note in SCSI-2.]

4.1.4 Negotiable fields

4.1.4.1 Negotiable fields introduction

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

Table 1. Negotiable fields and side-effects

Field Name		Negotiation message pair		
		PPR	**WDTR**	**SDTR**
TRANSFER PERIOD FACTOR		negotiated (valid values: 07h-FFh)	Resets to FFh	negotiated (valid values: 09h-FFh)
REQ/ACK OFFSET		negotiated	Resets to 00h	negotiated
TRANSFER WIDTH EXPONENT		negotiated	negotiated	unchanged
PROTOCOL OPTIONS	PCOMP_EN	negotiated	Resets to 0	Resets to 0
	RTI	negotiated	Resets to 0	Resets to 0
	RD_STRM	negotiated	Resets to 0	Resets to 0
	WR_FLOW	negotiated	Resets to 0	Resets to 0
	HOLD_MCS	negotiated	Resets to 0	Resets to 0
	QAS_REQ	negotiated	Resets to 0	Resets to 0
	DT_REQ	negotiated	Resets to 0	Resets to 0
	IU_REQ	negotiated	Resets to 0	Resets to 0

[Editor’s note: It is not strictly necessary for WDTR to reset the transfer period factor, as long as it resets the REQ/ACK offset. The value won’t be used unless REQ/ACK offset is non-zero, and changing REQ/ACK offset back to non-zero also requires setting the transfer period factor. It could be marked as reset to 00h, FFh, unchanged, or don’t care. The value is observable through the Report Currently Negotiated Settings mode page.]

When negotiating, the responding port shall respond with values that are a subset of the originating port's values 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 is 10h or lower).

Table 2. Responding message requirements

Field Name		Response shall be numerically
TRANSFER PERIOD FACTOR		Greater than or equal
REQ/ACK OFFSET		Less than or equal
TRANSFER WIDTH EXPONENT		Less than or equal
PROTOCOL OPTIONS	PCOMP_EN	Any value
	RTI	Less than or equal
	RD_STRM	Less than or equal
	WR_FLOW	Less than or equal
	HOLD_MCS	Less than or equal
	QAS_REQ	Less than or equal
	DT_REQ	Less than or equal
	IU_REQ	Less than or equal

The transfer agreements that are in effect for various combinations of field values are described in Table 3. The terms are not exclusive; more than one may be in effect at the same time.

Table 3. Transfer agreements

Transfer agreement	Description
default	REQ/ACK offset set to 00h, transfer width exponent set to 00h, all protocol options set to 0.
asynchronous data	REQ/ACK offset set to 00h, all protocol options set to 0.
synchronous data	REQ/ACK offset set to 01h – FFh, all protocol options set to 0.
paced data	REQ/ACK offset set to 01h – FFh., transfer period factor set to 08h.
wide	Transfer width exponent set to 01h.
ST data	DT_REQ set to 0.
DT data	DT_REQ set to 1.

4.1.4.2 Transfer period factor

The TRANSFER PERIOD FACTOR field sets the transfer period (see 3.1.101) and determines which timing values in tables 31 and 32 shall be honored. The field values are defined in Table 4. A value of 08h implies that a paced data transfer agreement is in effect. A value of 09h implies that a data group transfer agreement is in effect.

Table 4. Transfer Period Factor

Value	Description	Timing Values
00h-07h	Reserved. Faster transfer periods may be defined by future standards.	Not applicable
08h	Transfer period equals 6,25 ns. Only valid for paced transfers.	Fast-160
09h	Transfer period equals 12,5 ns. Only valid for data group transfers.	Fast-80
0Ah	Transfer period equals 25 ns	Fast-40
0Bh	Transfer period equals 30,3 ns	Fast-40
0Ch	Transfer period equals 50 ns	Fast-20
0Dh-18h	Transfer period equals the TRANSFER PERIOD FACTOR x 4	Fast-20
19h-31h	Transfer period equals the TRANSFER PERIOD FACTOR x 4	Fast-10
32h-FFh	Transfer period equals the TRANSFER PERIOD FACTOR x 4	Fast-5

[Editor's note: previous table used "code" instead of "value" and "notes" instead of "timing values"]
[Editor's note: previous table used wordy notes to vaguely refer to the fast-XX timing values. I think the new column and the references to tables 31 and 32 is clearer.]

[Editor's note: the following table is a bit redundant but might make things clearer. It relates the transfer period factor field to the protocol options bits (IU_REQ and DT_REQ). Should it be kept here, moved to the protocol options section, or removed?]

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

Table 5. Transfer Period Factor relationships

Value	Data group transfers	Information unit transfers	Synchronous data transfers	Paced data transfers
00h-07h	reserved			
08h	no	yes	no	yes
09h	yes	yes	yes	no
0Ah	yes	yes	yes	no
0Bh	yes	yes	yes	no
0Ch	yes	yes	yes	no
0Dh-18h	yes	yes	yes	no
19h-31h	yes	yes	yes	no
32h-FFh	yes	yes	yes	no

4.1.4.3 REQ/ACK offset

[Editor's note: in the rest of 4.1.4, text in SPI-4 is shown here with strikeouts for comparison.]

[Editor's note: should the offset of FFh be obsoleted? It is useless except for lab debug.]

For ST data 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 DT DATA IN transfers 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 DT DATA OUT transfers 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.

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 31 shall be honored. A REQ/ACK OFFSET value of zero indicates asynchronous data transfer mode and that the PERIOD FACTOR field and the PROTOCOL OPTIONS field shall be ignored; a value of FFh indicates unlimited REQ/ACK offset. If the REQ/ACK offset is not zero either a synchronous or paced data transfer agreement is in effect.
[Editor's note: the last sentence defines "synchronous data transfer agreement" used elsewhere. The terminology of the synchronous, paced, etc. data transfer agreements does not seem well defined. Is a

paced agreement also a synchronous agreement? If not "or paced data transfer agreement" needs to be added.]

Table 6. REQ/ACK offset

Value	Description	Timing values
00h	Asynchronous data transfer agreement. Transfer period factor and protocol options shall be ignored.	Asynch
01h-FEh	Synchronous or paced data transfer agreement. Specified offset.	Determined by Transfer Period Factor. See Table 4.
FFh	Synchronous or paced data transfer agreement. Unlimited offset.	Determined by Transfer Period Factor. See Table 4.

4.1.4.4 Transfer width exponent

~~The TRANSFER WIDTH EXPONENT field defines the transfer width to be used during DATA IN phases, and DATA OUT phases. The transfer width that is established applies to both SCSI devices. Valid transfer widths are 8 bits (m = 00h) and 16 bits (m = 01h) if all the protocol options bits are zero. The only valid transfer width is 16 bits (m = 01h) if any of the protocol options bits are one. TRANSFER WIDTH EXPONENT field values greater than 01h are reserved.~~

The TRANSFER WIDTH EXPONENT field defines the transfer width to be used during DATA IN and DATA OUT phases during synchronous data transfers. The values are defined in Table 7. If any of the protocol options bits are one, the only valid transfer width is 16 bits (01h). If all the protocol options bits are zero, a valid transfer width is 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 data transfer agreement is in effect. If the transfer width is 16 bits a wide data transfer agreement is in effect.
[Editor's note: the last sentence defines "wide data transfer agreement" used elsewhere]

Table 7. Transfer Width Exponent

Value	Description
00h	8 bit data bus (narrow)
01h	16 bit data bus (wide)
02h	Obsolete
03h-FFh	Reserved

4.1.4.5 Protocol options

4.1.4.5.1 Protocol options introduction

The protocol options fields affect the protocol used between the ports. They may only be negotiated through PPR messages, and are reset 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 lists the names of each of the protocol options bits.

Table 8. Protocol options bits.

Acronym	Name
PCOMP_EN	Precompensation enable
RTI	Retain training information
RD_STRM	Read streaming enable
WR_FLOW	Write flow control enable
HOLD_MCS	Hold margin control settings
QAS_REQ	Quick arbitration and selection (QAS) enable request
DT_REQ	Dual transition (DT) clocking enable request
IU_REQ	Information units (IU) enable request

4.1.4.5.2 IU_REQ

~~An information units enable request bit (IU_REQ) of zero indicates that information unit transfers shall not be used (i.e., data group transfers shall be enabled) when received from the initiator and that information unit transfers are disabled when received from the target. An IU_REQ bit of one indicates that information unit transfers shall be used when received from the initiator and that information unit transfers are enabled when received from the target. Each time the IU_REQ bit is changed from the previous agreement (i.e., zero to one or one to zero) as a result of a negotiation the target shall go to a BUS FREE phase on completion of the negotiation. Additional requirements (see 14.1) must be met if the IU_REQ bit is changed as a result of this negotiation.~~

~~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) must be met if the IU_REQ bit is changed as a result of this negotiation.~~

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 supports 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.

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) must be met if the IU_REQ bit is changed as a result of this negotiation.

4.1.4.5.3 DT_REQ

~~A DT enable request bit (DT_REQ) of zero indicates that DT DATA phases are to be disabled when received from the initiator and that DT DATA phases are disabled when received from the target. An DT_REQ bit of one indicates that DT DATA phases are to be enabled when received from the initiator and that DT DATA phases are enabled when received from the target.~~

~~A DT enable request bit (DT_REQ) of zero indicates that ST DATA phases are to be used rather than DT DATA phases, and that a ST data transfer agreement is in effect. A DT enable request bit of one indicates that DT DATA phases are to be used rather than ST DATA phases, and that a DT data transfer agreement is in effect. DT_REQ shall be set to one only if the transfer width exponent is set to 01h.~~

The initiator port may 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 supports DT DATA phases or zero if it does not.

The initiator port shall set IU_REQ to zero 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 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.

4.1.4.5.4 QAS_REQ

~~A QAS enable request bit (QAS_REQ) of zero indicates that QAS is to be disabled when received from the initiator and that QAS is disabled when received from the target. A QAS_REQ bit of one indicates that QAS is to be enabled when received from the initiator and that QAS is enabled when received from the target.~~

~~When QAS is enabled, the SCSI device shall participate in QAS arbitrations when attempting to connect to a SCSI device 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.~~

The initiator port may 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 IU_REQ to zero to request that information unit transfers be disabled. In response, the target port shall set IU_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.

4.1.4.5.5 HOLD_MCS

~~A hold margin control settings (HOLD_MCS) bit of zero received for by the target indicates 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) and that the target shall respond with the HOLD_MCS bit set to zero. A HOLD_MCS bit of one received by the target indicates that the target should hold any margin control settings set with the margin control subpage and that the target shall respond with the HOLD_MCS bit set to one if it retains the settings and zero if it does not.~~

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.

[Editor's note: SPI-4 currently excludes this from the "protocol options" bits because it doesn't affect the protocol. However, it's embedded among the other bits.]

4.1.4.5.6 WR_FLOW

A write flow control bit (WR_FLOW) of zero indicates that flow control during write streaming shall be disabled. A WR_FLOW bit of one indicates flow control during write streaming shall be enabled.

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

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

Write stream only occurs during information unit transfers.

Table 9 defines valid combinations of WR_FLOW and other fields.

4.1.4.5.7 RD_STRM

A read streaming bit (RD_STRM) of zero indicates read streaming shall be disabled. A RD_STRM bit of one indicates read streaming shall be enabled.

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

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

Read streaming only occurs during information unit transfers.

Table 9 defines valid combinations of RD_STRM and other fields.

4.1.4.5.8 RTI

A retain training information bit (RTI) of zero received by a target indicates the initiator does not support saving training information and the target shall respond with the RTI bit set to zero. For negotiated transfer periods greater than 6,25 ns the RTI bit shall be set to zero. A RTI bit of one received by a target indicate the initiator does support saving training information and the target may respond with the RTI bit set one if it supports saving training information.

The initiator port shall set RTI to one if it supports saving paced data transfer training information (see 10.8.4.2.1) to indicate that the target should not retrain. 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 if it does not support saving paced data transfer training information 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.

4.1.4.5.9 PCOMP_EN

For negotiated transfer periods equal to 6,25 ns a precompensation enabled bit (PCOMP_EN) of zero indicates to the SCSI device that receives the PPR message that it shall disable precompensation on all signals transmitted during DT DATA phases. For negotiated transfer periods greater than 6,25 ns the PCOMP_EN bit shall be set to zero. A PCOMP_EN bit of one indicates to the SCSI device that receives the PPR message that it shall enable precompensation on all signals transmitted during DT DATA phases. SCSI devices that support fast-160 shall support a the receipt of a PCOMP_EN bit set to zero or one.

~~Unlike other fields and bits in the PPR message the PCOMP_EN bit is not a negotiated value, rather 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 a successful PPR negotiation.~~

The initiator port shall set PCOMP_EN to one to indicate that the target shall enable precompensation (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 shall enable precompensation. The target port shall set PCOMP_EN to zero to indicate that the initiator shall disable precompensation.

When a port has precompensation enabled, precompensation drivers shall be used on all signals transmitted during DT DATA phases.

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.

4.1.5 Negotiable field combinations

Not all combinations of the 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 field combinations

Transfer period factor	REQ/ACK Offset	Transfer width exponent	Protocol options						Description
			RTI	RD_STRM	WR_FLOW	QAS_REQ	DT_REQ	IU_REQ	
0Ah – FFh	00h	01h	0	0	0	0	0	0	Use ST DATA IN and ST DATA OUT phases to transfer data with asynchronous data transfers
0Ah - FFh	01h - FFh	00h or 01h	0	0	0	0	0	0	Use ST DATA IN and ST DATA OUT phases to transfer data with synchronous data transfers
09h - FFh	01h - FFh	01h	0	0	0	0	1	0	Use DT DATA IN and DT DATA OUT phases with data group transfers
09h - FFh	01h - FFh	01h	0	0	0	1	1	0	Use DT DATA IN and DT DATA OUT phases with data group transfers, and participate in QAS arbitrations
09h – FFh	01h – FFh	01h	0	0 or 1	0 or 1	0	1	1	Use DT DATA IN and DT DATA OUT phases with information unit transfers
08h	01h - FFh	01h	0 or 1	0 or 1	0 or 1	0	1	1	Use DT DATA IN and DT DATA OUT phases with information unit transfers
09h – FFh	01h - FFh	01h	0	0 or 1	0 or 1	1	1	1	Use DT DATA IN and DT DATA OUT phases with information unit transfers, participate in QAS arbitrations, and use QAS for arbitration
08h	01h - FFh	01h	0 or 1	0 or 1	0 or 1	1	1	1	Use DT DATA IN and DT DATA OUT phases with information unit transfers, participate in QAS arbitrations, and use QAS for arbitration

4.1.6 Message restrictions

PPR shall not be originated by targets. Initiators should only use PPR when requesting values not attainable via WDTR and SDTR (selecting a transfer period factor < 0Ah or setting any protocol option bits to 1). If a target responds to PPR with values that are attainable via WDTR and SDTR, the initiator 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. Initiators shall only initiate PPR to targets which have reported DT support in their INQUIRY page CLOCKING fields (byte 56; see SPC-2).
[Editor's note: The last sentence is a new requirement. Is the second sentence new or can it be a shall? George wants should there.]

WDTR and SDTR may be originated by either targets or initiators. Since WDTR resets all the values that SDTR sets, it shall be sent first if both are being used.

Targets shall initiate negotiations only to request eight-bit asynchronous mode with protocol options set to 00h. *[this is a new requirement]* Negotiation for faster speeds is reserved for initiators, which can perform integrity checking to determine negotiated settings that work on the interconnect. Targets shall start negotiation by using a WDTR message with the TRANSFER WIDTH EXPONENT set to 00h. If that is rejected, they shall use an SDTR message with the transfer period factor and REQ/ACK offset set to 00h.

[FIXFIX now in the table The text below is repeated several times in SPI-3. The scenario is not clear. How can a device not respond? Unexpected Bus Free cases are covered above. Need any other cases be discussed?
"If an abnormal condition prevents the responding device from responding with a negotiation message or with a MESSAGE REJECT message then both SCSI devices shall use the eight-bit/asynchronous data transfer mode with all the protocol options bits set to zero."]

4.1.7 Negotiation message sequences

[Editor's note: this is the difficult section. The initiator and target each have different views of the state of negotiation. When one side sends a message, the other side may receive it, or may receive a parity error on any byte.

If the originating message went bad, the responding message won't be a valid negotiation message, so the originator knows the responder did not receive it correctly. For example, a PPR OUT won't be followed by a PPR IN unless the target was happy with it.

However, if the originating message is fine but the responding message is bad, the two sides start to have trouble determining whether to maintain their settings, invalidate them (and negotiate again with an older message), or go to async mode. It's not always safe to drop to async mode – the other side might not have done the same.

When an initiator detects a parity error, it creates an attention condition and sends a MESSAGE PARITY OUT on the resulting MESSAGE OUT phase.

When a target detects a parity error, it retries MESSAGE OUT one or more times and goes BUS FREE if it gives up.

Targets can tell which byte an initiator saw with bad parity by noting when the initiator creates an attention condition, but initiators cannot tell which byte a target saw with bad parity.]

4.1.7.1 Initiator originated negotiation

When an initiator port originates a negotiation message, the target port may respond in these ways:

a) reply with the same negotiation message;
b) reply with MESSAGE REJECT IN;
c) reply with at least 3 valid message bytes indicating the same negotiation message, then provide a message byte indicating bad parity;
d) reply with one of the first 3 message bytes indicating bad parity;
e) reply with the first message bytes indicating some other message, then a message byte indicating bad parity
f) reply with some other message;
g) create a bus free condition; or
h) change bus phase.

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

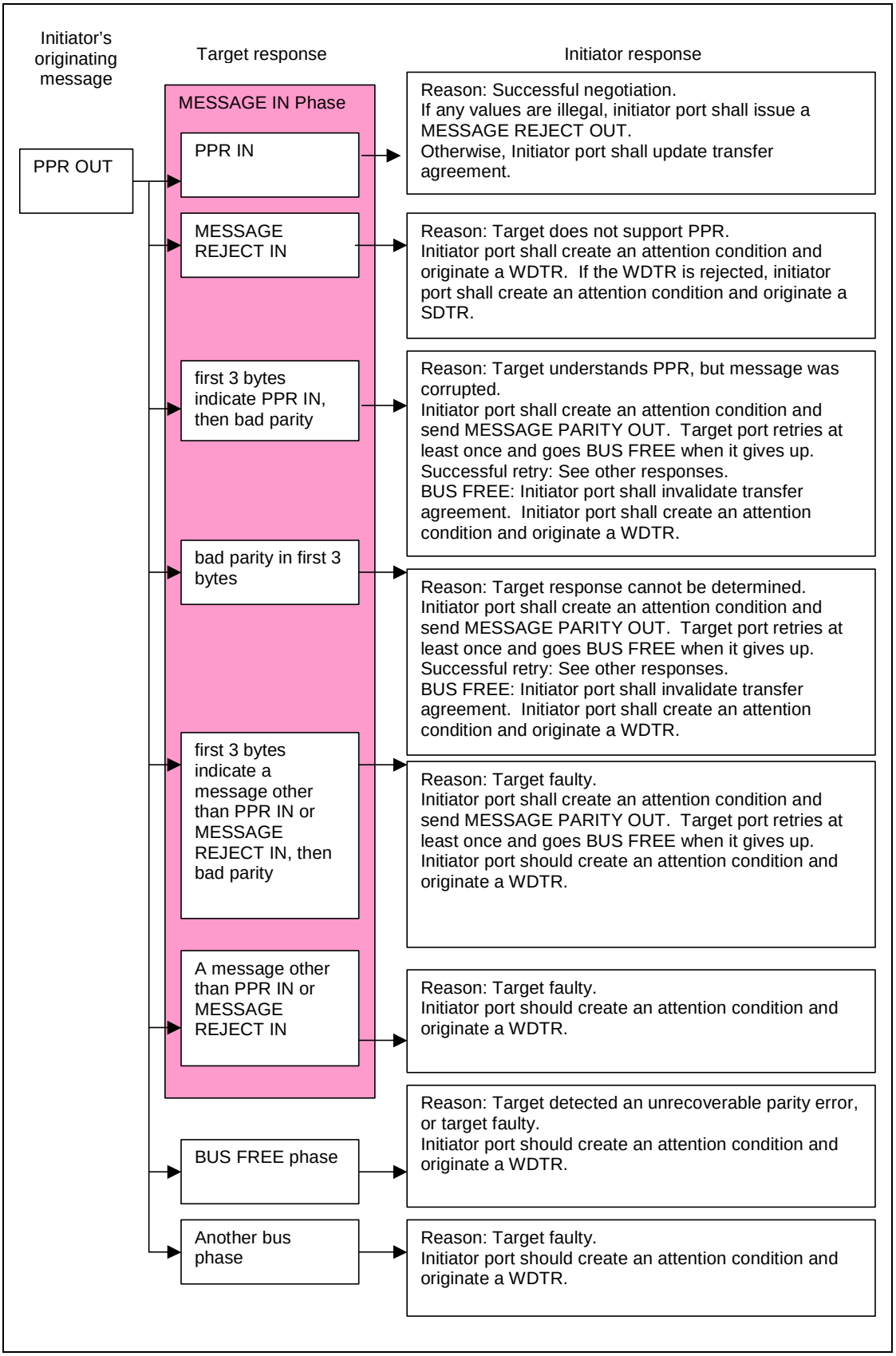

Figure 2. Initiator originated PPR negotiation: initiator response to target responses to PPR OUT

When the target port replies with the same negotiation message, the initiator may respond in these ways:

a) create an attention condition and send MESSAGE REJECT OUT;
b) create an attention condition and send MESSAGE PARITY OUT;
c) create an attention condition and send a message other than MESSAGE REJECT OUT or MESSAGE PARITY OUT;
d) create an attention condition and send a message with bad parity; or
e) not create an attention condition.

Figure 3 shows how the target shall respond to various initiator responses. The initiator port shall maintain the previous transfer agreement unless otherwise indicated.

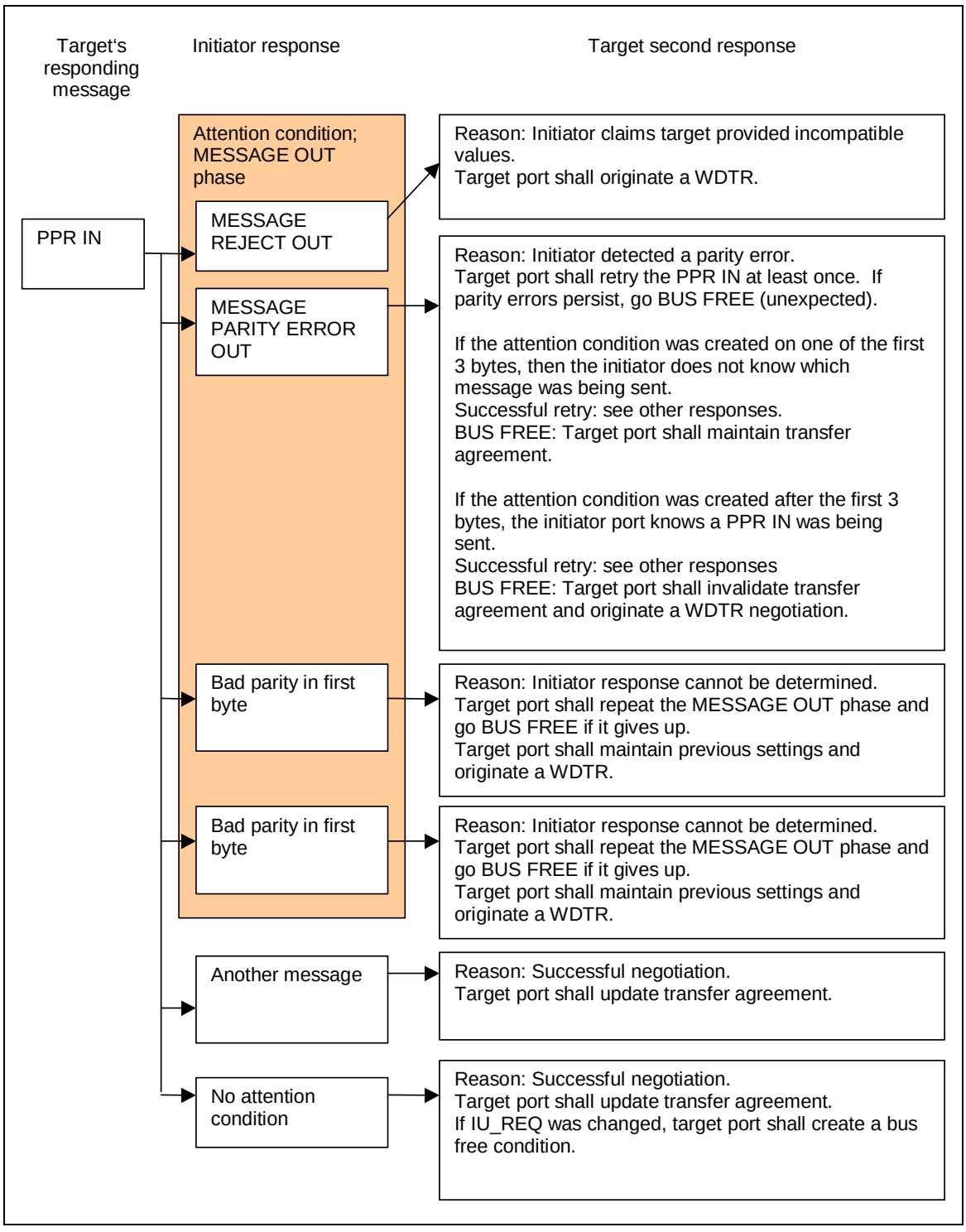

Figure 3. Initiator originated PPR negotiation: target response to initiator responses

[Editor's note: The WDTR OUT sequences that follow are shown in table format. Which is preferred, table or figure? I think the figures are clearer.]
[Editor's note: All these cases need to be carefully reviewed. WDTR failure implies narrow and resetting the protocol options, but does not always trash the SDTR settings.]

Table 10 shows how the initiator shall respond to various target responses to a WDTR OUT.

Table 10. Initiator-originated WDTR negotiation: initiator response to target responses

Target response	Initiator response
WDTR IN	Reason: Successful negotiation. If any values are illegal, initiator port shall issue a MESSAGE REJECT OUT. Otherwise, initiator port shall update transfer agreement.
MESSAGE REJECT IN	Reason: Target does not support WDTR. Initiator port shall set its transfer width exponent to 00h and maintain other negotiated settings. Initiator port may originate SDTR negotiation.
first three bytes indicate WDTR IN, but then bad parity	Reason: Target understands WDTR, but message was corrupted. Initiator port shall create an attention condition and send MESSAGE PARITY OUT. Target port retries at least once and goes BUS FREE when it gives up. Successful retry: See other responses. BUS FREE: Initiator port shall reset its transfer agreement. *[SPI-4 16.3.16.3 If during the responding WDTR message the initiator creates an attention condition and the first message of the MESSAGE OUT phase is either a MESSAGE PARITY ERROR or MESSAGE REJECT message the wide data transfers shall be considered to be negated by both SCSI devices. In this case, both SCSI devices shall use the eight-bit data transfer mode for data transfers between the two devices.]* *[Reason for change: everything WDTR is capable of setting should be cleared, not just the width. Both sides know support a WDTR and know a WDTR was attempted, so they can both clear everything]*
bad parity in first three bytes	Reason: Target response cannot be determined. Initiator port shall create an attention condition and send MESSAGE PARITY OUT. Target port retries at least once and goes BUS FREE when it gives up. Successful retry: See other responses. BUS FREE: Initiator port shall maintain its negotiated settings..
first 3 bytes indicate another message, then bad parity	Reason: Target faulty. Initiator port shall create an attention condition and send MESSAGE PARITY OUT. Target port retries at least once and goes BUS FREE when it gives up. Initiator port shall maintain previous negotiated settings. Initiator port should create an attention condition and originate an SDTR. *[SPI-4 16.3.16.3 If an abnormal condition prevents the target from responding with a WDTR message or with a MESSAGE REJECT message then both SCSI devices shall go to eight-bit transfer mode for data transfers between the two SCSI devices.]*
Another message	Reason: Target faulty. Initiator port shall maintain previous negotiated settings. Initiator port should create an attention condition and originate an SDTR.
BUS FREE phase	Reason: Target detected an unrecoverable parity error, or target faulty. Initiator port shall maintain previous negotiated settings. Initiator port should create an attention condition and originate an SDTR. *[SPI-4 16.3.16.1 If there is an unexpected bus free on the initial WDTR message the initiating SCSI device shall retain its previous data transfer mode.]* *[SPI-4 16.3.16.1 If there is an unrecoverable parity error on the initial WDTR message (see 10.12.2 and 10.12.4) the initiating SCSI device shall retain its previous data transfer mode.]*
Another bus phase	Reason: Target faulty. Initiator port shall maintain previous negotiated settings. Initiator port should create an attention condition and originate a WDTR. *[SPI-4 16.3.16.3 If an abnormal condition prevents the target from responding with a WDTR message or with a MESSGE REJECT message then both SCSI devices shall go to eight-bit transfer mode for data transfers between the two SCSI devices.]*

Table 11 describes how the target shall respond to various initiator responses.

Table 11. **Initiator originated WDTR negotiation: target response to initiator responses**

Initiator responses	Target response
(either no attention condition, or attention condition but the message is not a MESSAGE REJECT OUT or MESSAGE PARITY OUT message)	Reason: Successful negotiation. Target port shall update transfer agreement. If IU_REQ was changed, target port shall create a bus free condition.
(attention condition) MESSAGE REJECT OUT	Reason: Initiator claims target provided incompatible values. Target port shall maintain previous settings. [What should the target width be? Should the target retry the WDTR IN in case the cause was a double-bit error? Is the initiator required to clear the attention condition, prohibiting it from retrying the WDTR OUT?] Target shall retry at least once. If rejects persist, the target shall go BUS FREE and both target and initiator shall set their data transfer agreements to eight-bit asynchronous with protocol options set to 00h. *[SPI-4 16.3.16.3 If during the responding WDTR message the initiator creates an attention condition and the first message of the MESSAGE OUT phase is either a MESSAGE PARITY ERROR or MESSAGE REJECT message the wide data transfers shall be considered to be negated by both SCSI devices. In this case, both SCSI devices shall use the eight-bit data transfer mode for data transfers between the two devices.]*
(attention condition) MESSAGE PARITY ERROR OUT	Reason: Initiator detected a parity error. Target port shall retry the WDTR IN at least once. If parity errors persist, go BUS FREE (unexpected). If the attention condition was created on one of the first 3 bytes, then the initiator does not know which message was being sent. Successful retry: see other responses. BUS FREE: Target port shall maintain its transfer agreement. If the attention condition was created after the first 3 bytes, the initiator port knows a WDTR IN was being sent. Successful retry: see other responses BUS FREE: Target port shall set its transfer width exponent to 00h. *[SPI-4 16.3.16.3 If during the responding WDTR message the initiator creates an attention condition and the first message of the MESSAGE OUT phase is either a MESSAGE PARITY ERROR or MESSAGE REJECT message the wide data transfers shall be considered to be negated by both SCSI devices. In this case, both SCSI devices shall use the eight-bit data transfer mode for data transfers between the two devices.]* *[Reason for change: everything WDTR is capable of setting should be cleared, not just the width.]*

(attention condition) MESSAGE PARITY ERROR OUT	Initiator detected parity error and doesn't know what the message is. Initiator shall consider the nexus unusable.
Unexpected Bus Free	Target detected unrecoverable parity errors. Both target and initiator shall set their data transfer agreements to eight-bit asynchronous with protocol options set to 00h. *[should sync and protocol options be cleared?]*

[Editor's note: until a format is chosen, SDTR OUT is not described. The difference from PPR and WDTR is that if SDTR fails, both sides assume the default transfer agreement. There is no lesser message to try.]

4.1.7.2 Target-initiated negotiation

[Editor's note: until the format is chosen, WDTR IN and SDTR IN are not described]

4.1.7.3 Another table format

[Editor's note: the following five tables were the original format used by the document. They are too confusing.]

Table xx1 defines the possible negotiation message sequences when the initiator starts with PPR OUT.

Table xx1. Initiator-originated PPR message sequences

Responding message or bus state	**Subsequent message or bus state**	**Description** ***[prev "implied agreements"]***
PPR IN	(no attention condition, or attention condition with a message other than MESSAGE REJECT OUT or MESSAGE PARITY ERROR OUT)	Successful negotiation. Both devices update their data transfer agreements.
PPR IN	(attention condition) MESSAGE REJECT OUT	Target provided incompatible values, or initiator faulty. Initiator and target shall maintain previous settings. Target shall originate a WDTR negotiation (see table xx3). *[SPI-4 16.3.10.2 If during the PARALLEL PROTOCOL REQUEST message the initiator creates an attention condition and the first message of the MESSAGE OUT phase is either a MESSAGE PARITY ERROR or MESSAGE REJECT message the data transfers shall be considered to be negated (sic) by both SCSI devices. In this case, both SCSI devices shall use the eight-bit/asynchronous data transfer mode with all the protocol options bits set zero to indicate ST DATA IN and ST DATA OUT phases for data transfers between the two SCSI devices.]* *[Reason for change: Both sides saw the PPR OUT and know*

		something is wrong. Initiator must drop attention after sending a MESSAGE REJECT OUT. Target should try another negotiation in case the initiator is faulty.]
PPR IN (first three bytes received successfully)	(attention condition) MESSAGE PARITY ERROR OUT	Initiator detected a parity error. Target shall retry the MESSAGE IN at least once. If parity errors persist, the target shall go BUS FREE and both target and initiator shall set their data transfer agreements to eight-bit asynchronous with protocol options set to 00h. Initiator shall originate a WDTR negotiation (see table xx2). *[SPI-4 16.3.10.2 If during the PARALLEL PROTOCOL REQUEST message the initiator creates an attention condition and the first message of the MESSAGE OUT phase is either a MESSAGE PARITY ERROR or MESSAGE REJECT message the data transfers shall be considered to be negated (sic) by both SCSI devices. In this case, both SCSI devices shall use the eight-bit/asynchronous data transfer mode with all the protocol options bits set zero to indicate ST DATA IN and ST DATA OUT phases for data transfers between the two SCSI devices.]* *[Reason for change: Both sides saw the PPR OUT and know they're having trouble exchanging a PPR IN. Initiator must drop attention after sending a MESSAGE REJECT OUT. Target should try another negotiation in case the initiator is faulty*
1st, 2nd, or 3rd byte parity error	(attention condition) MESSAGE PARITY ERROR OUT	Initiator detected parity error and doesn't know what the message is. Initiator shall consider the nexus unusable.
PPR IN	Expected Bus Free	Successful negotiation where IU_REQ was changed. Both devices update their data transfer agreements.
PPR IN	Unexpected Bus Free	Target detected unrecoverable parity errors or target faulty. Target may have received a MESSAGE PARITY ERROR OUT and retried the MESSAGE IN phase multiple times. Both target and initiator shall set their data transfer agreements to eight-bit asynchronous with protocol options set to 00h.
MESSAGE REJECT IN	(no parity error detected by initiator)	The target does not support PPR or target faulty. The initiator shall create an attention condition and originate a WDTR negotiation (see table xx2). *[SPI-3 said initiator goes to async 8-bit]* *[see T10/00-337]*
Unexpected Bus Free		Target detected an unrecoverable parity error or target faulty. Target may have retried the MESSAGE OUT phase multiple times. Initiator shall maintain previous settings. The initiator should create an attention condition and originate a WDTR negotiation (see table xx2). *[SPI-4 16.3.10.1 If there is an unrecoverable parity error on the initial PARALLEL PROTOCOL REQUEST message (see 10.12.2 and 10.12.4) the initiator shall retain its previous data transfer mode and protocol options.]* *[SPI-4 16.3.10.1 If there is an unexpected bus free on the initial PARALLEL PROTOCOL REQUEST message the initiator shall retain its previous data transfer mode and protocol options.]* *[SPI-4 16.3.10.2 If an abnormal condition prevents the target from responding with a PARALLEL PROTOCOL REQUEST message or with a MESSAGE REJECT message then both SCSI devices shall use the eight-bit/asynchronous data transfer mode with all the protocol options bits set zero to indicate ST DATA IN and ST DATA OUT phases between the two SCSI devices.]*

Target leaves MESSAGE phase or sends a message other than PPR IN or MESSAGE REJECT IN		Target faulty. Initiator shall maintain previous settings. The initiator should create an attention condition and originate a WDTR negotiation (see section table xx2). *[SPI-4 16.3.10.2 If an abnormal condition prevents the target from responding with a PARALLEL PROTOCOL REQUEST message or with a MESSAGE REJECT message then both SCSI devices shall use the eight-bit/asynchronous data transfer mode with all the protocol options bits set zero to indicate ST DATA IN and ST DATA OUT phases between the two SCSI devices.]*

Table xx2 defines the possible negotiation message sequences when the initiator starts with WDTR OUT. FIXFIX table 50 disagrees with 16.3.7 which talks about an attention condition still pending after a target sends a message reject

Table xx2. Initiator-originated WDTR message sequences

Responding message or bus state	**Subsequent message or bus state**	Description *[prev "implied agreements"]*
WDTR IN	(either no attention condition, or attention condition but the message is not a MESSAGE REJECT OUT or MESSAGE PARITY OUT message)	Successful negotiation. Both devices update their data transfer agreements.
WDTR IN	(attention condition) MESSAGE REJECT OUT	Target provided incompatible values, or initiator faulty. Initiator and target shall maintain previous settings. Target shall originate a SDTR negotiation (see table xx5). [What should the target width be? Should the target retry the WDTR IN in case the cause was a double-bit error? Is the initiator required to clear the attention condition, prohibiting it from retrying the WDTR OUT?] Target shall retry at least once. If rejects persist, the target shall go BUS FREE and both target and initiator shall set their data transfer agreements to eight-bit asynchronous with protocol options set to 00h. *[SPI-4 16.3.16.3 If during the responding WDTR message the initiator creates an attention condition and the first message of the MESSAGE OUT phase is either a MESSAGE PARITY ERROR or MESSAGE REJECT message the wide data transfers shall be considered to be negated by both SCSI devices. In this case, both SCSI devices shall use the eight-bit data transfer mode for data transfers between the two devices.]*
WDTR IN (first three bytes received successfully)	(attention condition) MESSAGE PARITY ERROR OUT	Initiator detected parity error. Target shall retry at least once. If parity errors persist, the target shall go BUS FREE and both target and initiator shall set their data transfer agreements to eight-bit asynchronous with protocol options set to 00h.

		[SPI-4 16.3.16.3 If during the responding WDTR message the initiator creates an attention condition and the first message of the MESSAGE OUT phase is either a MESSAGE PARITY ERROR or MESSAGE REJECT message the wide data transfers shall be considered to be negated by both SCSI devices. In this case, both SCSI devices shall use the eight-bit data transfer mode for data transfers between the two devices.] *[Reason for change: everything WDTR is capable of setting should be cleared, not just the width.]*
1st, 2nd, or 3rd byte parity error	(attention condition) MESSAGE PARITY ERROR OUT	Initiator detected parity error and doesn't know what the message is. Initiator shall consider the nexus unusable.
WDTR IN	Unexpected Bus Free	Target detected unrecoverable parity errors. Both target and initiator shall set their data transfer agreements to eight-bit asynchronous with protocol options set to 00h. *[should sync and protocol options be cleared?]*
MESSAGE REJECT IN		The target does not support a wide bus. Initiator shall set its transfer width exponent to 00h. Initiator may originate SDTR negotiation.
Unexpected Bus Free		Initiator shall maintain previous settings. Initiator should attempt SDTR negotiation (see table xx4). *[SPI-4 16.3.16.1 If there is an unexpected bus free on the initial WDTR message the initiating SCSI device shall retain its previous data transfer mode.]* *[SPI-4 16.3.16.1 If there is an unrecoverable parity error on the initial WDTR message (see 10.12.2 and 10.12.4) the initiating SCSI device shall retain its previous data transfer mode.]*
Target leaves MESSAGE phase or sends a message other than WDTR IN or MESSAGE REJECT IN		Target faulty. Initiator shall maintain previous settings. The initiator should create an attention condition and perform a SDTR negotiation (see table xx4). *[SPI-4 16.3.16.3 If an abnormal condition prevents the target from responding with a WDTR message or with a MESSGE REJECT message then both SCSI devices shall go to eight-bit transfer mode for data transfers between the two SCSI devices.]*

Table xx3 defines the possible negotiation message sequences when the target starts with WDTR IN.

Table xx3. Target-originated WDTR message sequences

Responding message or bus state	**Subsequent message or bus state**	Description *[prev "implied agreements"]*
(attention condition) WDTR OUT	(either no attention condition, or attention condition but the message is not a MESSAGE REJECT OUT or MESSAGE PARITY OUT	Successful negotiation. Both devices update their data transfer agreements.

	message)	
(attention condition) WDTR OUT	MESSAGE REJECT IN	Target does not support the initiator's requested transfer width exponent, or target faulty. Both devices shall clear protocol options and set their transfer width exponents to 00h. *[SPI-4 16.3.16.2 If the target does not support the initiator's responding TRANSFER WIDTH EXPONENT the target shall switch to a MESSAGE IN phase and the first message shall be a MESSAGE REJECT message. In this case the implied agreement shall be considered to be negated and both SCSI devices shall use the eight-bit data transfer mode for data transfers between the two SCSI devices. Any prior synchronous data transfer agreement shall remain intact.]*
(attention condition) WDTR OUT (may be retried multiple times)	Unexpected Bus Free	Target detected unrecoverable parity error. Both target and initiator shall set their transfer width exponents to 00h. *[SPI-4 16.3.16.2 If a parity error occurs, the implied agreement shall be reinstated if a retransmission of a subsequent pair of messages is successfully accomplished. After a vendor-specific number of retry attempts (greater than zero), if the target continues to receive parity errors, it shall terminate the retry activity. This is done by the target causing an unexpected bus free. The initiator shall accept such action as aborting the WDTR negotiation, and both SCSI devices shall go to eight-bit data transfer mode for data transfers between the two SCSI devices. Any prior synchronous data transfer agreement shall remain intact.]* *[should sync and protocol options be cleared?]*
(attention condition) MESSAGE REJECT OUT		The initiator does not support a wide bus. Target shall set its transfer width exponent to 00h. Target may attempt SDTR negotiation.
(attention condition) MESSAGE PARITY ERROR OUT		Initiator detected parity error. Target shall retry at least once. If parity errors persist, target shall set its data transfer agreement to eight-bit asynchronous mode with protocol options set to 00h. Data errors are likely to follow because the initiator settings were not changed. *[SPI-4 16.3.16.1 If there is an unrecoverable parity error on the initial WDTR message (see 10.12.2 and 10.12.4) the initiating SCSI device shall retain its previous data transfer mode.]* *[SPI-4 16.3.16.2 If a parity error occurs, the implied agreement shall be reinstated if a retransmission of a subsequent pair of messages is successfully accomplished. After a vendor-specific number of retry attempts (greater than zero), if the target continues to receive parity errors, it shall terminate the retry activity. This is done by the target causing an unexpected bus free. The initiator shall accept such action as aborting the WDTR negotiation, and both SCSI devices shall go to eight-bit data transfer mode for data transfers between the two SCSI devices. Any prior synchronous data transfer agreement shall remain intact.]*
Unexpected Bus Free		Target faulty. Initiator shall maintain previous settings. Initiator may originate SDTR negotiation. *[SPI-4 16.3.16.1 If there is an unexpected bus free on the initial WDTR message the initiating SCSI device shall retain its previous data transfer mode.]*
Initiator does		Initiator faulty.

not assert attention condition or sends some other message		Target shall set its transfer width exponent to 00h and originate SDTR negotiation. *[SPI-4 16.3.16.2 If an abnormal condition prevents the initiator from responding with a WDTR message or with a MESSAGE REJECT message then both SCSI devices shall go to eight-bit data transfer mode for data transfers between the two SCSI devices.]*

Table xx4 defines the possible negotiation message sequences when the initiator starts with SDTR OUT.

Table xx4. Initiator-originated SDTR message sequences

Responding message or bus state	**Subsequent message or bus state**	Description *[prev "implied agreements"]*
SDTR IN	(no attention condition)	Successful negotiation. Both devices update their data transfer agreements.
SDTR IN	(attention condition) MESSAGE REJECT OUT	Target provided incompatible values, or initiator faulty. Target shall set its transfer period factor to 00h, REQ/ACK offset to 00h, and protocol options to 00h. Target shall send an SDTR IN with transfer period factor of 00h and REQ/ACK offset of 00h. *[SPI-3 16.3.14.3 If during the SDTR message the initiator creates an attention condition and the first message out is either MESSAGE PARITY ERROR or MESSAGE REJECT the synchronous operation shall be considered to be negated by both the initiator and the target. In this case, both SCSI devices shall go to asynchronous data transfer mode and any protocol options shall no longer be in effect (see 16.3.10) for data transfers between the two SCSI devices.]* *[Reason for change: initiator cannot be expected to do anything since it sent a reject. Target should send the simplest SDTR in a last-ditch effort to notify the initiator, and go to async mode]*
SDTR IN (first three message bytes received successfully)	(attention condition) MESSAGE PARITY ERROR OUT	Initiator detected parity error but knows the message is an SDTR IN. Target shall retry the SDTR IN at least once. If parity errors persist, both target and initiator shall set their transfer period factors to 00h, REQ/ACK offsets to 00h, and protocol options to 00h. *[SPI-3 16.3.14.3 If during the SDTR message the initiator creates an attention condition and the first message out is either MESSAGE PARITY ERROR or MESSAGE REJECT the synchronous operation shall be considered to be negated by both the initiator and the target. In this case, both SCSI devices shall go to asynchronous data transfer mode and any protocol options shall no longer be in effect (see 16.3.10) for data transfers between the two SCSI devices.]* *[no change]*
1st, 2nd, or 3rd byte parity error	(attention condition) MESSAGE PARITY ERROR OUT	Initiator detected parity error and doesn't know what the message is. Initiator shall consider the nexus unusable.
SDTR IN	Unexpected Bus Free	Target faulty. Both initiator and target shall set their transfer period factors to 00h, REQ/ACK offsets to 00h, and protocol options to 00h.
MESSAGE REJECT IN		The target does not support synchronous data transfers. Initiator shall set its transfer period factor to 00h, REQ/ACK offset to 00h, and protocol options to 00h. Initiator should send an SDTR OUT with transfer period factor of

		00h and REQ/ACK offset of 00h.
Unexpected Bus Free		Target detected unrecoverable parity errors, or target faulty. Initiator shall maintain its previous settings. Initiator should originate SDTR negotiation with transfer period factor of 00h and REQ/ACK offset of 00h during the next connection. *[SPI-3 16.3.14.1 If there is an unrecoverable parity error on the initial SDTR message (see 10.12.2 and 10.12.4) the initiating SCSI device shall retain its previous data transfer mode.]* *[SPI-3 16.3.14.1 If there is an unexpected bus free on the initial SDTR message the initiating SCSI device shall retain its previous data transfer mode.]* *[Reason for change: parity error may go away over time. No reason to automatically drop to async mode]*
Target leaves MESSAGE phase or sends a message other than SDTR IN or MESSAGE REJECT IN		Target faulty. Initiator shall maintain its previous settings. If still in message phase, the initiator should create an attention condition and send an SDTR message with transfer period factor of 00h and REQ/ACK offset of 00h, and set its data transfer agreement to eight-bit asynchronous with protocol options set to 00h. If no longer in MESSAGE phase, the initiator should originate an SDTR negotiation during the next connection with transfer period factor of 00h and REQ/ACK offset of 00h. *[SPI-3 16.3.14.3 If an abnormal condition prevents the target from responding with a SDTR message or with a MESSAGE REJECT message then both SCSI devices shall go to asynchronous data transfer mode and any protocol options shall no longer be in effect (see 16.3.10) for data transfers between the two SCSI devices.]* *[Reason for change: target cannot be expected to do anything, so just spec the initiator. Initiator should send the simplest SDTR to try to get agreement if it can. It might be better for the initiator to retain its settings and try to negotiate async at the next opportunity.]*

Table xx5 defines the possible negotiation message sequences when the target starts with SDTR IN.

Table xx5. Target-originated SDTR message sequences

Responding message or bus state	**Subsequent message or bus state**	Description *[prev "implied agreements"]*
(attention condition) SDTR OUT	(no attention condition)	Successful negotiation. Both devices update their data transfer agreements.
(attention condition) SDTR OUT	MESSAGE REJECT IN	Target does not support the initiator's requested values, or initiator faulty. Both devices shall set their transfer period factors to 00h, REQ/ACK offsets to 00h, and protocol options to 00h. *[SPI-4 16.3.14.2: In this case the implied agreement shall be considered to be negated and both SCSI devices shall use the asynchronous data transfer mode and any protocol options shall no longer be in effect (see 16.3.10) for data transfers between the two SCSI devices.]* *[No change]*
(attention condition) SDTR OUT	Unexpected Bus Free	Target detected unrecoverable parity errors. Both target and initiator shall set their data transfer agreements to asynchronous mode.

(attention condition) MESSAGE REJECT OUT		The initiator does not support synchronous data transfers. Target shall set its transfer period factor to 00h, REQ/ACK offset to 00h, and protocol options to 00h.
(attention condition) MESSAGE PARITY ERROR OUT		Initiator detected parity error. Target shall retry the SDTR IN at least once. If parity errors persist, target shall go to BUS FREE and set its transfer period factor to 00h, REQ/ACK offset to 00h, and protocol options to 00h. Data errors are likely to follow because the initiator settings were not changed. *[SPI-4 16.3.14.1 If there is an unrecoverable parity error on the initial SDTR message (see 10.12.2 and 10.12.4) the initiating SCSI device shall retain its previous data transfer mode.]* *[16.3.14.2: if the target continues to receive parity errors, it shall terminate the retry activity. This is done by the target causing an unexpected bus free. The initiator shall accept such action as aborting the SDTR negotiation, and both SCSI devices shall go to asynchronous data transfer mode and any protocol options shall no longer be in effect (see 16.3.10) for data transfers between the two SCSI devices.]*
Unexpected Bus Free		Target faulty. Initiator shall maintain its previous settings. *[SPI-4 16.3.14.1 If there is an unexpected bus free on the initial SDTR message the initiating SCSI device shall retain its previous data transfer mode.]* *[Reason for change: the target should not have caused the bus free anyway, any target requirements are suspect.]*
Initiator does not assert attention condition or sends some other message		Initiator faulty. Target shall set its transfer period factor to 00h, REQ/ACK offset to 00h, and protocol options to 00h. *[SPI-4 16.3.14.2: If an abnormal condition prevents the initiator from responding with a SDTR message or with a MESSAGE REJECT message then both SCSI devices shall return to asynchronous data transfer mode and any protocol options shall no longer be in effect (see 16.3.10) for data transfers between the two SCSI devices.]* *[Reason for change: initiator cannot be expected to do anything here]*

16.1 Negotiation messages

[Editor's note: create a new section 16.x parallel to the Link control messages section.]

16.1.1 Negotiation message codes

[Editor's note: Move PPR, WDTR, and SDTR out of the Link control message codes table into a new table here.]

Table 12. Negotiation message codes

Extended Message, Length, and	Init	Targ	Message Name	In	Out	Clear Attention Condition

Code						
01h, 06h, 04h	O	O	PARALLEL PROTOCOL REQUEST (PPR) OUT		Out	Yes
01h, 06h, 04h	O	O	PARALLEL PROTOCOL REQUEST (PPR) IN	In		
01h, 03h, 01h	O	O	SYNCHRONOUS DATA TRANSFER REQUEST (SDTR) OUT		Out	Yes
01h, 03h, 01h	O	O	SYNCHRONOUS DATA TRANSFER REQUEST (SDTR) IN	In		
01h, 02h, 03h	O	O	WIDE DATA TRANSFER REQUEST (WDTR) OUT		Out	Yes
01h, 02h, 03h	O	O	WIDE DATA TRANSFER REQUEST (WDTR) IN	In		
Key: M=Mandatory support, O=Optional support In=Target to initiator, Out=Initiator to target Yes=When sending the message, the initiator shall clear the attention condition before the last ACK of the MESSAGE OUT phase. Init=initiator, Targ=target						

16.1.2 PARALLEL PROTOCOL REQUEST message description

PARALLEL PROTOCOL REQUEST (PPR) messages (see Table 13) are used to negotiate the transfer period factor, REQ/ACK offset, transfer width exponent, and protocol options between two SCSI devices.

Table 13. PARALLEL PROTOCOL REQUEST message format

Bit Byte	7	6	5	4	3	2	1	0
0	Extended Message (01h)							
1	Extended Message Length (06h)							
2	Parallel Protocol Request (04h)							
3	TRANSFER PERIOD FACTOR							
4	RESERVED							
5	REQ/ACK OFFSET							
6	TRANSFER WIDTH EXPONENT *["m" removed]*							
7	PROTOCOL OPTIONS							
	PCOMP_EN	RTI	RD_STRM	WR_FLOW	HOLD_MCS	QAS_REQ	DT_REQ	IU_REQ

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 1, or QAS set to 1 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.

16.1.3 SYNCHRONOUS DATA TRANSFER REQUEST message description

SYNCHRONOUS DATA TRANSFER REQUEST (SDTR) messages (see Table 14) are used to negotiate the transfer period factor and REQ/ACK offset between two SCSI devices.

Table 14. SYNCHRONOUS DATA TRANSFER REQUEST message format

Byte	
0	Extended Message (01h)
1	Extended Message Length (03h)

2	Synchronous Data Transfer Request (01h)
3	TRANSFER PERIOD FACTOR
4	REQ/ACK OFFSET

SDTR messages shall be supported by devices supporting synchronous data transfers (i.e. non-zero REQ/ACK offsets). SDTR messages shall be supported by targets with SYNC set to 1 in the INQUIRY page of all their logical units (see SPC-2).

Only transfer period factors greater than or equal to 0Ah shall be negotiated with SDTR. PPR shall be used for transfer period factors less than 0Ah.

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

16.1.4 WIDE DATA TRANSFER REQUEST message description

WIDE DATA TRANSFER REQUEST (WDTR) messages (see Table 15) are used to negotiate the transfer width exponent between two SCSI devices.

Table 15. WIDE DATA TRANSFER REQUEST message format

Byte	
0	Extended Message (01h)
1	Extended Message Length (02h)
2	Wide Data Transfer Request (03h)
3	TRANSFER WIDTH EXPONENT *["m" removed]*

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 1 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.

Additional changes

References to sections 16.3.12 (PPR), 16.3.16 (SDTR), and 16.3.18 (WDTR) need to be redirected to the new section numbers or the 4.x model section.

All references to the various field names need to be reviewed. (e.g. "if information units are enabled")

All references to "transfer agreement" need to be reviewed. (e.g. paced data transfer agreement vs. paced transfer agreement)