Proposal for Parallel SCSI:
Increase Transfer Rate and Improve Error Detection

To: **T10 Technical committee**
From: **Jim McGrath**
Quantum Corporation
500 McCarthy Boulevard
Milpitas, CA USA 95035
Phone: 408-894-4019
Fax: 408-952-3620
Email: jim.mcgrath@quantum.com
Date: **September 11 ~~August 17~~,1998**

0   Overview

Rev 5

This revision incorporates changes suggested during the conference call to review revision 4, held on August 27, 1998. Representatives from Adaptec, Compaq, IBM, QLogic, Quantum, Seagate, Symbios, and WD attended.

- an error was made in the revision note for revision 4, and that change has been made
- in section 8.2.2.3, a (obvious) error was made, caught, and corrected.
- The transfer period times were made TDB – they will be speified in the Fast-80 proposal instead.

Otherwise all changes are insertions/deletions of appropriate wording to support decision(s) taken several issues. The issues surrounding these decisions are document in another proposal ("Outstanding Issues in Document 98-177, Proposal for Parallel SCSI: Increase Transfer Rate and Improve Error Detection"), and modifying/approving that proposal will make the corresponding changes in this proposal. This issues, and their identifying Letter, are:

A.   Degree of support for 8 bit transfers
B.   Degree of support for single ended
C.   Use of the P1 signal
D.   Value of the Pad bytes

For each issue there are a number of options. For issues A and B the options are very similar:

A1) 16 bit only, 8 bit forbidden
A2) 16 bit only, silence on 8 bit
A3) 16 bit recommended but and 8 bit allowed
A4) 16 bit and 8 bit allowed

and

       B1) LVD only, single ended and HVD forbidden
       B2) LVD only, silence on single ended, and HVD forbidden
       B3) LVD recommended but, single ended allowed, and HVD forbidden
       B4) LVD and single ended allowed, and HVD forbidden

For issues C and D the options are fewer:

       C1) the current wording indicating that P1 is 16 bit parity
       C2) new wording indicating that the P1 line shall not be driven by the device receiving data, and shall be released or driven at a rate no higher than the negotiated data transfer rate for the device sending the data. It is intended to reclaim this line for future use.

and

       D1) pad bytes are 00
       D2) pad bytes are anything

The text that is inserted/deleted is delimiated with a letter and number combination, e.g. A3, which would refer to issue A, option 3 ("16 bit recommended but and 8 bit allowed"). This information is noted by being enclosed in double square brackets ([[ ]]), making the changes easy to search for and subsequently edit (e.g. [[A3]]).

Rev 4

This revision incorporates changes suggested at the T1110 standards meeting in Portland, Maine on AugustJuly 14, 1998.

A general consensus emerged at that meeting that this document is close to being recommended for inclusion in the next revision of SPI-3. Towards that end, this revision is intended for review at a teleconference call before the next T10 meeting in September. At that time a final revision will be presented and (hopefully) meet full committee approval.

A reminder that notes included between square brackets are explanatory information for the committee, and will not be incorporated into SPI-3. If someone feels that some of that material (or material like that) is needed in the standard, then please advise me ASAP so I can draft appropriate wording.

All changes in this document are highlighted from revision 3. These include:
- An explanation of how the CRC protocol works in both 16 and 8 bit modes
- Inclusion of a diagram on how CRC is calculated approved at the earlier Orange Country meeting
- Removal in section 24 of the exception for BUS FREE phase (since it is covered elsewhere in SPI), but an enumeration of the other phases to insure there is no misunderstanding.
- Elimination in section 27 of timings other than those related to CRC_Valid in the Timing budget table

Rev 3

This revision notes the changes made in response to feedback at the special working group in Orange County June 19.

At various points table, figure, and clause numbers are identified by xxx. The SPI-3 editor will insert the appropriate numbers when the document is incorporated into SPI-3. Some figures may also have to be

modified for incorporation into the SPI-3 document. The SPI-3 editor may make the appropriate non-substantive changes required.

The note on using DT/CRC only with wide buses has been deleted. Although raised as an issue in the committee, the working groups opinion is to continue to allow narrow buses to be used with the new protocols. This issue may be raised again at a later date. Given its controversy and lack of solid support in the committee, I suggest that any attempt to restrict the new protocol to wide buses be offered as a separate proposal to the committee.

The section on CRC testing was modified with new tests included.

The sections on information transfer phases and DT transfer phase was clarified to require that the REQx/ACKx signals be deasserted before leaving the DT phase, even on error conditions.

Rev 2

This revision notes the changes made in response to the phone conference call reviewing the rev 1 document that took place on June 5. Individuals from Adaptec, Compaq, IBM, Qlogic, Quantum, Seagate, and Symbios Logic particiapated.

Rev 1

This revision notes the changes from revision 0. It is an outgrowth of the discussions at the T10 working group meeting on May 22, 1998 in Chicago. It will be reviewed on a conference call on June 5, with one or more subsequent revisions generated for the next working group meeting on June 19 in Irvine.

Name changes: DEC to DT, double-edged to double transition, SEC to ST, single-edged to single transition, IUTR to PPR, frame to group, and frame check sequence to group check sequence.

Section 5: reword to make asynchronous data transfers in DT data phase explicitly illegal.

Section 7: eliminate the enumeration of the phase lines, since they do not have simple meanings as individual lines.

Section 8: keep parity in SELECTION phase (it was accidentally dropped in the rev 0 of SPI-3). Note that DB(P) and DB(P2) are used for parity only in the ST DATA IN and ST DATA OUT phases. Eliminated the error rate information for the CRC algorithm. Restructured the description of a data group, and moved the pad bytes to be between the data bytes and the CRC bytes. CRC now covers the data field and the pad field. The CRC calculation section needs more work, and will be revised in the next revision.

Section 22: a note was added that SECTION still uses parity, and is assumed that will be fixed in SPI-3, rev 1.

Section 27: extensive rewording of how CRC errors are detected was made for clarity. The additional setup time for CRC_AVAILABLE assertion was changed from 12.5 to 10 ns in Fast-40:DT and Fast-80:DT. A couple of sections were eliminated, although they may just need rewording. Another 10 ns setup time for the subsequent deassertion is proposed for these speeds.

Rev 0

What follows beginning with Clause 1 is our first pass at a proposal for including double-edge clocked clocking and CRC into SPI.

General themes:

This proposal references the SPI-3 r0 draft. I have tried to note references to (SPI-2) rev 20a or the packetized SCSI proposal (T10/97-230r6) when possible, but the SPI-3 draft is a fairly simple merge of the two documents, so it should be straightforward for people to follow.

I avoided changes to sections 6 and 7, since they are the analog sections affected T10/98-153r1. Comments on that proposal are included, since there is some degree of content overlap. Specifically, other sections in the document do require Fast-80:DT timing information, and 153r1 does not define Fast-10:DT, Fast-20:DT, and Fast-40:DT. These items have been provided, but it should be noted that some change in terminology may be required for double edge clocking, and the exact timing values are subject to further discussion.

A whole bunch of new definitions were added.

Note that DT (double-edge clocked) synchronous data transfer must be defined. I was surprised to note that there are no synchronous data transfer timing diagrams in SPI (there are sections in the clauses that specify measurement points, but no overall timing diagram). I have supplied such a timing diagram (I am sure that some changes will be necessary to put it in "proper" form).

"Parity" has been replaced by "Protection" in all areas of SPI where the clause could deal with either Parity or CRC. In the clause where Parity was defined, I have added a clause defining CRC, as well as the 16-bit P1 parity (and an analogous 16-bit P3 parity for the secondary bus).

DATA phase now consists of ST DATA IN, ST DATA OUT, DT DATA IN, and DT DATA OUT (ST is Single-Edge Clocked, and DT is Double-Edge Clocked). DATA IN generically now refers to ST DATA IN and DT DATA IN; DATA OUT to ST DATA OUT and DT DATA OUT. ST DATA generically refers to ST DATA IN and ST DATA OUT; DT DATA to DT DATA IN and DT DATA OUT. I made corresponding changes throughout SPI. This change gives us terminology to be precise in direction and protocol for data transfer when that is needed, but to retain the existing data phase terminology throughout our standards where that level of detail is irrelevant.

Packetized SCSI was actually pretty easy to accommodate – basically all references to INFORMATION UNIT IN (or OUT) phases were replaced by information units or data phases, whichever was appropriate (sometimes both, as in data phases that can only contain information units). At this time I have kept the initial restriction that information units only be transferred in the DT data phase, but that could be changed rather easily if people wanted to allow information units during the traditional ST data phase.

I did not change the PPR message section (George Penokie was going to work on that). It is important to note that this is the negotiation that established asynchronous, ST synchronous, or DT synchronous/CRC transfer protocols; whether the information unit protocol is used or not; whether Domain Validation is to be attempted; and the resource requirements for Domain Validation. The Domain Validation proposal is T10/98-176.

Various tables have to be altered – rather than produce them, I listed instructions on changing them for the editor (since any change I make he has to redo in FRAMEMAKER anyway). The only exception is in the timing tables, since there is a lot of information in them.

I have identified high priority issues for us to focus on by [??? … ???]. These issues have been identified through discussions with a variety of people, and signify important areas where I do not believe there is a consensus.

This proposal marches sequentially through the SPI document. The "technical meat" of this proposal in sections 8 (on CRC) and 27 (double-edge clocking data transfers). I suggest that people focus primarily on these areas in their review.

I am looking into a couple of issues people raised privately to me recently, and will make whatever (minor) updates are needed for the meeting of March 22. Hardcopy markups for information purposes will be available at the working group meeting. I strongly suggest that people have available some form of SPI-2 and the packetized SCSI proposal, or the SPI-3 draft, since I have made little attempt to duplicate the text surrounding the proposed modifications.

1   Introduction

This document describes a proposal for the SCSI parallel interface to:

1)   define double transition clocking to reduce the frequency of the REQx/ACKx signals at a given transfer rate and to increase the maximum data rate to 80 megatransfers per second; and,
2)   Improve error detection through the use of CRC.

The methods defined in this proposal allow initiators and targets to be backward compatible with fast-40, fast-20, fast-10 and fast-5 initiators and targets. This proposal requires no change in maximum cable lengths or number of devices from fast-20. In addition, there are no changes required for connectors, cables or terminators.

This proposal references clauses in SCSI Parallel Interface – 3 (SPI-3) rev 0. Where text should be modified I have tried to produce the original text and the new text with strikeouts and underscores to reflect the modifications. Changes in tables are much harder, and there I have been reduced to specifying the changes in English.

[For discussion purposes I have included comments in several sections. These comments are in brackets and can be deleted from the final proposal document.]

Special thanks to Mark Evans for his diligent review and tireless editing of this document. Without his outstanding help this document would barely have been possible, and certainly not in as good shape.

2   T10/98-153r1 comments

Elimination of HVD from SPI-3: I support this, but desire committee input before assuming we can eliminate it in our documentation.

Page 2 (clause 9) – change looks OK.

Page 3 (clause 9.1) – the Fast-80:DT timings are documented in this proposal in section 27, and there are some differences. A discussion based on the model for the timings would be helpful in clarifying some of the differences.

Page 11 - In addition, I have provided timings for Fast-10:DT Fast-20:DT, and Fast-40:DT, which hopefully will prove to be less controversial than Fast-80:DT timings for obvious reasons.

I believe that Board skew can be significantly reduced for most devices (such as disk drives) if some way can be found to allow a few devices on the bus (e.g. motherboard mounted SCSI chips) a looser specification due to their greater board layout challenges. One technique would be to allow these devices to have twice as much skew, under the assumption that they are also supplying termination (which limits the number of such devices to 2). Another is to create two classes of devices – Class 1 devices are help to the tighter specification, Class 2 to a looser one, and then restrict the number of Class 2 devices in configurations (afterall, there are not many systems with 16 hosts with motherboard

mounted SCSI chips). Note that Domain Validation can assist in insuring that very badly configured systems are detected and appropriate transfers rates used.

Page 19 – for the LVD capacitance loads, we support the 15 pF. This is an area where we agree, and can even suggest a lower value for difference between C1 and C2 (1 pF rather than 1.5 pF). Even more than board skew, reductions in this area should help overall margin.

Once again, while I believe that the capacitance can be significantly reduced for most devices (such as disk drives) if some way can be found to allow a few devices on the bus (e.g. motherboard mounted SCSI chips) a looser specification due to their greater board layout challenges. The same techniques discussed for board skew could be applied in this instance.

As far as I can see, all other portions of the proposal are identical with SPI-3, and so don't present any problems with this proposal.

3   Definitions

The following definitions should be added in clause 3.1 of SPI-3 (or SPI-2).

3.1 Cyclic Redundancy Check (CRC): An error detecting code used to detect the validity of data that has been transferred during the current data phase.

3.2 double transition clocking (DT): The method for transferring data into a register or latch on both polarity edges of the clock signal. Double-edged clocking is used in the **DT** data phases to perform data transfers on both edges of the REQ or ACK.

3.3 fast-10:DT: Negotiated to receive synchronous data at a transfer rate less than or equal to a transfer rate of 10 megatransfers per second using **double transition** clocking.

3.4 fast-20:DT: Negotiated to receive synchronous data at a transfer rate greater than 10 megatransfers per second and less than or equal to 20 megatransfers per second using **double transition** clocking**.**

3.5 fast-40:DT: Negotiated to receive synchronous data at a transfer rate greater than 20 megatransfers per second and less than or equal to 40 megatransfers per second using **double transition** clocking.

3.6 fast-80:DT: Negotiated to receive synchronous data at a transfer rate greater than 40 megatransfers per second and less than or equal to 80 megatransfers per second using **double transition** clocking.

3.7 Data Group: The complete sequence of data bytes, any pad bytes, and the four CRC bytes during a DT phase.

3.8 Intersymbol interference (ISI): The effect of adjacent symbols on the symbol currently being received.

3.9 Protection: Signal used to transfer parity information or control the transfer of CRC information. Both allow error detection for signals on the DATA BUS, and so provide protection for those signals.

3.10 Single-edged clocking (ST): The method for transferring data into a register or latch on one polarity edge of the clock signal. Single-edged clocking is used in the **ST** data phases to perform data transfers when REQ or ACK transition from negated to asserted**.**

The following definitions should be modified in clause 3.1 of SPI-3.

3.11 Make the following modification:

odd parity: Odd logical parity, where the parity bit is driven and verified to be that value that makes the number of assertions on the associated DATA BUS plus the parity bit equal to an odd number (1, 3, 5, 7,or 9). See xxx, parity bit.

3.12 Make the following modification :

parity bit: A bit associated with a transfer on the DATA BUS that is used to detect the presence of single-bit errors within the transfer. The parity bit is driven such that the number of logical ones in the transfer plus the parity bit is odd**.**

4   Abbreviations

The following abbreviations should be added in clause 3.2 of SPI-2.

> CRC   Cyclic Redundancy Check
> DT   double transition clocked
> ISI   Intersymbol Interference
> MTps   Megatransfers per second
> ST   single transition clocked

5   Clause 4.1.1 Data Transfer Modes

This clause should be modified as follows:

SCSI parallel interface devices default to 8-bit transfer. The 8-bit information transfer mode is always used for all information transfers except the DATA phases. DATA phases may use the 8-bit, 16-bit or 32-bit wide, if a wide transfer agreement is in effect.

SCSI parallel interface devices default to asynchronous transfer. The asynchronous information transfer mode is always used for all information transfers except the DATA phases. ST DATA phases may use either asynchronous or synchronous, if a synchronous transfer agreement is in effect. DT DATA phases shall use synchronous and a synchronous transfer agreement shall be effect if that phase is used.

6   Clause 4.1.2 Cables, Connectors, Signals, Transceivers

[This modification introduces the concepts of the fast-80 transfer rate and DT (double-edge clocked) LVD transfers. It restricts fast-80 to DT LVD and notes that fast-10:DT, fast-20:DT, and fast-40:DT are also allowed (note that asynchronous DT is not allowed).]

Modify Table 1 (Transceiver/speed support map) in the following manner:

1) Insert a row between "LVD" and "HVD" labeled "DT LVD"
2) In this row make the respective values (from left to right) "No", "No", "Yes", "Yes", and "Yes"
3) Then insert a column after "Fast-40" labeled "Fast-80"
4) In this column make the respective values (from top to bottom) "No", "No", "No", "Yes", "No"

7   Clause 8.1 Signal Descriptions

[The following two modifications simply note that the MSG signal is used to distinguish the DT data phases as well as the MESSAGE phase. The signal will still be known as the MSG signal.]

Eliminate the Control and Message line sections and put in a note to refer to the information phase definition in table xxx in SPI-3.

[The following modifications change "parity" to "protection" and remove the definition of parity for the DB(P, P1, P2, P3) signals (they are defined later).]

1) **DB(7-0,P) (8-bit DATA BUS).** Eight data-bit signals, plus a protection signal that forms the 8-bit DATA BUS. Bit significance and priority during arbitration are shown in table 39.
2) **DB(15-0,P,P1) (16-bit DATA BUS).** Sixteen data-bit signals, plus two protection signals that form the 16-bit DATA BUS. Bit significance and priority during arbitration are shown in table 39.
3) **DB(31-0,P,P1,P2,P3) (32-bit DATA BUS).** Thirty-two data-bit signals, plus four protection signals that form the 32-bit DATA BUS. Bit significance and priority during arbitration are shown in table 39.

8   Clause 8.2 Parity checking rules

[This is a major modification to the document. The parity nomenclature is replaced by the protection nomenclature -- this ripples throughout the document. This clause should specify two types of protection: parity and CRC. It should both define them and indicate how and when they are used.]

"Valid parity is determined by rules in table 40." Should be replaced by the following clauses [there are no "change" indications in the following as this is essentially an entire new "8.2" clause]:

8.2 Data Bus Protection

The data bus protection signals are used to generate parity or control the transfer of CRC information.

8.2.1 Parity

For an 8-bit data bus: DB(P) shall contain odd parity for DB(7-0) when in the SELECTION, RESELECTION, COMMAND, STATUS, MESSAGE IN, MESSAGE OUT, ST DATA IN, or ST DATA OUT phases. [[A3]][[A4]]DB(P) shall be used as the CRC_Available signal in DT DATA IN and DT DATA OUT phases.[[A3]][[A4]]

For a 16-bit data bus: DB(P,P1) shall contain odd parity for DB(7-0) and DB(15-8), respectively, when in the SELECTION, RESELECTION, COMMAND, STATUS, MESSAGE IN, MESSAGE OUT, ST DATA IN, or ST DATA OUT phases. In the DT DATA IN and DT DATA OUT phases DB(P) shall be used as the CRC_Available signal [[C1]] and DB(P1) shall contain odd parity for DB(15-0).[[C1]][[C2]]. In the DT DATA IN phase the initiator shall not drive P1, and the target shall either release it or drive it no faster than the negotiated data transfer rate. In the DT DATA OUT phase the target shall not drive P1, and the initiator may either release it or drive it no faster than the negotiated data transfer rate. In either case the initiator shall ignore the value of this signal[[C2]]

For a 32-bit data bus: DB(P,P1,P2,P3) shall contain odd parity for DB(7-0), DB(15-8), DB(23-16), and DB(31-24), respectively, when in the SELECTION, RESELECTION, COMMAND, STATUS, MESSAGE IN, MESSAGE OUT, ST DATA IN, or ST DATA OUT phases. In the DT DATA IN and DT DATA OUT phases DB(P,P2) shall be used as CRC_Available signals[[C2]] and DB(P1,P3) shall contain odd parity for DB(15-0) and DB(31-16) respectively.[[C1]][[C2]]. In the DT DATA IN phase the initiator shall not drive DB(P1,P3), and the target shall either release them or drive them no faster than the negotiated data transfer rate. In the DT DATA OUT phase the target shall not drive DB(P1,P3), and the initiator may either release them or drive them no faster than the negotiated data transfer rate. In either case the initiator shall ignore the value of this signal. [[C2]]

8.2.1.1 Parity Checking Rules

Valid parity is determined by rules in table 40.

[Insert table 40 here]

8.2.2 CRC

The primary bus and secondary bus generate, transfer, and check CRC independently. This section describes the operation for the primary bus, which is [[A3]][[A4]]either 8-bit or [[A3]][[A4]]16 bits wide. The operation for the secondary bus is identical except that the secondary bus signals are used instead of the primary bus signals (DB(P2) for DB(P), DB(P3) for DB(P1), DB(31-16) for DB(15-0), REQQ for REQ, ACKQ for ACK)[[A3]][[A4]], and the secondary bus is always 16 bits wide.[[A3]][[A4]]

8.2.2.1 Error Detection Capabilities

The error detecting code is a 32 bit Cyclic Redundancy Check (CRC), referred to as CRC-32. It is also used by Fibre Channel, FDDI, and Ethernet. Four CRC bytes are transferred with data to increase the reliability of data transfers.

The CRC can detect all single bit errors, all double bit errors, all odd numbers of errors, and all burst errors up to 32 bits long.

8.2.2.2 Application

CRC is used in the DT DATA IN and DT DATA OUT phases only. CRC protects a group of information transferred during these phases. Each phase consists of a transfer of one or more of these data groups. A data group consists of a data field, followed by a pad field, and then followed by a CRC field. The initiator or target transmitting data sends the pad and CRC fields at a point in time determined by the target.

The target shall not change phases except at data group boundaries. Any phase change within a data group indicated either an error in logic or a data transmission error that can be alleviated in no other manner. In either event, all of the data transferred for that data group shall be considered to have been transferred incorrectly, and appropriate error reporting/recovery actions taken.

[Note: in the case of logic errors (changing phases "out of the blue") the error is usually a fatal hardware error. Of more interest are errors due to data transmission errors that generate REQ/ACK offset mismatches or an incorrect number of pad/CRC bytes (due to a problem with CRC_Available).

In the former case there the SCSI bus would hang today (since the REQ/ACK offset would never get to zero), forcing a BUS RESET to clear the condition. While it may be possible to improve on this for both ST and DT data phases, this problem is not unique to DT data phase and so is not solved as part of this proposal. In the latter case the detection/recovery is straightforward – if the rules on pad/CRC bytes are not followed, then an error is declared. Note that hardware can, for the sake of efficiency, assume that the rules are always followed (this simplifies the pipelining of data into the CRC generator/checker) and note the failure to follow the rules in parallel. As long as error is detected in real time, so that the device would act in the same manner as if the CRC checker detected an error, then the implementation is acceptable.]

8.2.2.3 DATA Group Format

A data group consists of a data field followed by a pad field and a CRC field. A group shall always have an even number of transfers, and shall be a multiple of four bytes (32 bits).

REQ and ACK are negated both before and after the transmission of a data group. This is the result of a group always being an even number of transfers. The combined data and pad fields shall always have an even number of transfers (since the CRC field always consists of an even ([[A3]][[A4]]two four for an 8-bit bus, or [[A3]][[A4]]four two for a 16 bit bus) number of transfers). The combined data and pad fields shall also always be a multiple of four bytes (since the CRC field always consists of a multiple (of one) of four bytes).

The pad field shall consist of 0 [[A3]][[A4]]to 3 [[A3]][[A4]][[A1]][[A2]]or 2 [[A1]][[A2]]bytes as follows:

    [[A3]][[A4]]8-bit bus: 0 bytes if the data field is a multiple of four bytes in length

            1 byte if the data field is a multiple of four bytes plus three extra bytes in length

            2 bytes if the data field is a multiple of four bytes plus two extra bytes in length

            3 bytes if the data field is a multiple of four bytes plus one extra byte in length[[A3]][[A4]]

    16-bit bus: 0 bytes if the data field is a multiple of four bytes in length

            2 bytes if the data field is a multiple of four bytes plus two extra bytes in length

    [If there are 0 pad bytes, then no pad bytes are sent, pad bytes may be non-zero]

If there are no 0 pad bytes, then no pad bytes are sent. If there are [[A3]][[A4]]1, 2, or 3[[A3]][[A4]][[A1]][[A2]]2[[A1]][[A2]] pad bytes, then that number of pad bytes are sent. [[D2]]Pad bytes may be of any value.[[D2]] [[D1]]Pad bytes shall have the value of zero. The receiving device is not required to check the value of the pad bytes.[[D1]]

The data field shall consist of any number of transfers (including zero), as determined by the target.

Data Field	Pad Field	CRC Field
Associated with REQ sent with CRC_Available Negated	Associated with REQ sent with CRC_Available Asserted	
0 to N bytes	0 to 3 bytes	4 bytes
Even R/A Transitions		Even R/A Transitions
Four Byte Multiple		Four Byte Multiple

           Increase Transfer Rate and Improve Error Detection Proposal

Figure xxx: Fields within a Data Group

Only one set of REQxs associated with CRC_Available for the pad field and the CRC field of a single data group shall be outstanding at any time.

[Note that these rules allow for a straightforward implementation of CRC. Since the data and pad field bytes are checked by the CRC, all of the data is put into the CRC checker as usual. Since all calculations are done using 32 bit (4 byte) data word, these words are sent to the CRC checker and are considered as a unit. For any word, if the first byte received by the device is associated with an asserted state of CRC_Available, then the entire word must be the CRC field (since pads of 4 bytes are illegal). For sending user data onto the device, the rule is simplier yet: after CRC checking, the bytes not associated with the asserted state of CRC_Available are the data field – the others are pad/CRC field bytes. Note that one implementation is to record the value of CRC-available associated with each 8 or 16 bit piece of data transferred on the bus until these determinations are made (similar to storing the parity bit for each word today). Other implementations are possible and legal as long as the observable ~~behaviour~~behavior is as documented in the standard.]
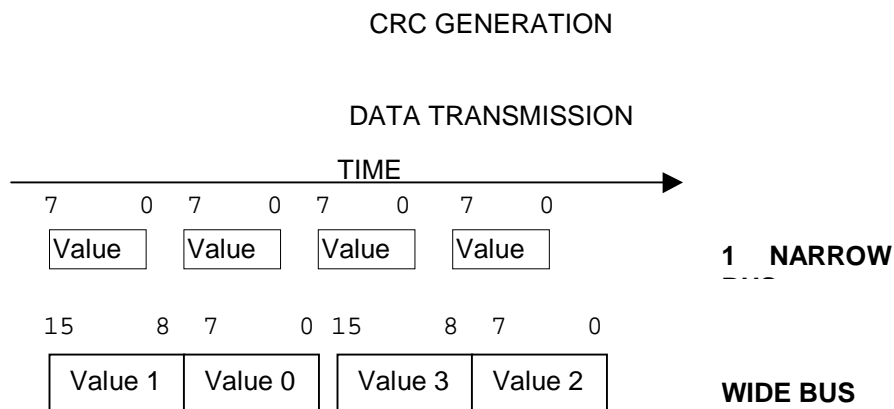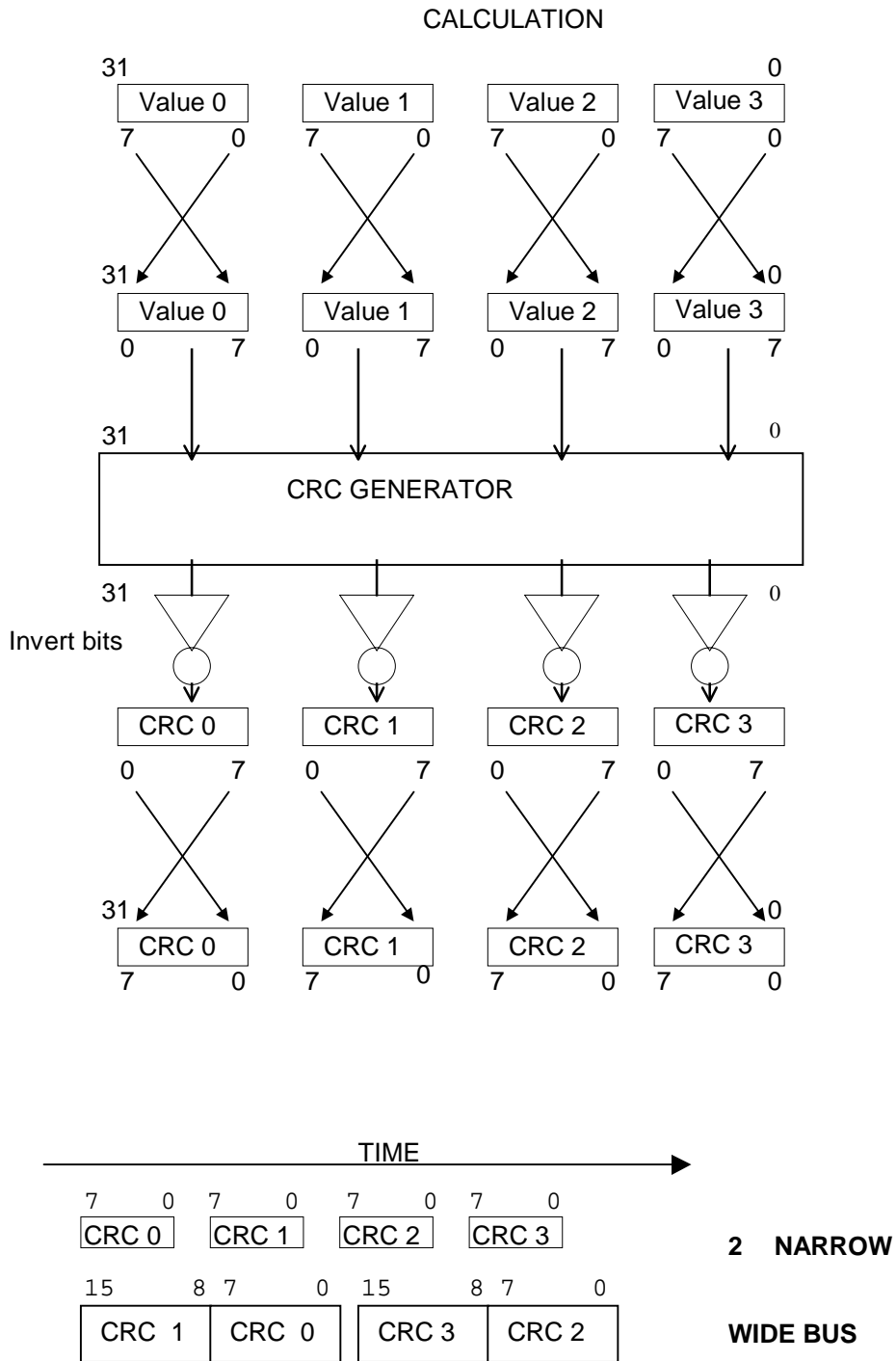
8.2.2.4 CRC_Available Signal

The CRC_Available signal is asserted by the target to indicate a REQ associated with byte(s) in a pad or CRC field of a data group.. The CRC_Available signal is negated to indicate a REQ associated with byte(s) in a data field of a data group. CRC_Available is transmitted on DB(P) for [[A3]][[A4]]an 8-bit or [[A3]][[A4]]16-bit DATA BUS; DB(P) and DB(P2) for a 32-bit DATA BUS.

8.2.2.5 Order of Bytes in the CRC Field

[[A1]][[A2]] [Note to the editor: that the following tables will be modififed to remove 8 bit.][[A1]][[A2]]

The data is transmitted, used to calculate the CRC, and the CRC is transmitted in the following manner:

CRC GENERATION

DATA TRANSMISSION

TIME →

7	0	7	0	7	0	7	0
Value		Value		Value		Value	

1   NARROW BUS

15	8	7	0	15	8	7	0
Value 1		Value 0		Value 3		Value 2	

WIDE BUS

CALCULATION

31   0

| Value 0 | | Value 1 | | Value 2 | | Value 3 |
| 7 | 0 | 7 | 0 | 7 | 0 | 7 | 0 |

31   0

| Value 0 | | Value 1 | | Value 2 | | Value 3 |
| 0 | 7 | 0 | 7 | 0 | 7 | 0 | 7 |

31   0

| CRC GENERATOR |

31   0

Invert bits

| CRC 0 | | CRC 1 | | CRC 2 | | CRC 3 |
| 0 | 7 | 0 | 7 | 0 | 7 | 0 | 7 |

31   0

| CRC 0 | | CRC 1 | | CRC 2 | | CRC 3 |
| 7 | 0 | 7 | 0 | 7 | 0 | 7 | 0 |

TIME

| 7 | 0 | 7 | 0 | 7 | 0 | 7 | 0 |
| CRC 0 | | CRC 1 | | CRC 2 | | CRC 3 |

2   NARROW

| 15 | 8 | 7 | 0 | 15 | 8 | 7 | 0 |
| CRC 1 | | CRC 0 | | CRC 3 | | CRC 2 |

WIDE BUS

8.2.2.6 CRC Generation and Checking

The 32 bit generator polynomial used is:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

This equals 0x104C11DB7 in hexadecimal.

The remainder is generated by dividing the bytes in the data and pad fields (which are a multiple of four bytes) by the generator polynomial, modulo two. The remainder is generated 32 bits at a time.

The remainder is initialized to all ones (0xFFFFFFFF). This is the seed value. It is reloaded at the beginning of each DT DATA phase and after each CRC is generated/checked.

Bytes are bit reversed prior to generating the remainder and the bytes of the remainder are bit reversed prior to forming the CRC field.

The CRC Field is the ones complement of the bit reversed remainder.

A unique remainder is generated by an error free group . The unique remainder polynomial of an error free group is:

$$x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1$$

This equals 0xC704DD7B in hexadecimal.

8.2.2.7 Test Cases

For a 32 byte transfer of all 0's, the CRC transferred across the SCSI bus:

Narrow:	0xad, 0x55, 0x0a, 0x19
Wide:	0x55ad, 0x190a

For a 32 byte transfer of all 1's:

Narrow:	0x0b, 0xab, 0x6c, 0xff
Wide:	0xab0b, 0xff6c

For a 32 byte transfer of an incrementing patter from 0x00 to 0x1F:
(Note that the last test case used to be 0x01 to 0x20, I didn't have
any reason for changing it, I just didn't check before running the simulation)

Narrow:	0x8a, 0x7e, 0x26, 0x91
Wide:	0x7e8a, 0x9126

9 Clause 8.4 OR-Tied Signals

Modify the third sentence in the second paragraph as follows: "Protection bits shall not be driven false during the ARBITRATION phase but may be driven false in other phases."

10 Clause 8.5 Signal Sources

[The packetized SCSI proposal (T10/97-230r6) added two rows to table 41 (Signal Sources).]

1) In the first column: replace INFORMATION UNIT IN with DT DATA IN, and INFORMATION UNIT OUT with DT DATA OUT.
2) Split column 6 (headed by "DB7-0, DB(P)") into two columns (headed by "DB7-0" and "DB(P)" respectively). The contents of the two columns shall be the same as the original column, except for the row marked "DT DATA OUT". In that row, the value in the "DB(P)" column shall be "Targ", not "Init."
3) Split column 10 (headed by "DB31-16, DB(P2), DB(P3)") into two columns (headed by"DB31-16, DB(P3)" and "DB(P2)" respectively). The contents of the two columns shall be the same as the original column, except for the row marked "DT DATA OUT". In that row the value in the "DB(P2)" column shall be "Targ", not "Init."
4) In the footnotes for Table 41, modify the following: "S ID: A unique data bit (the SCSI ID) shall be driven by each SCSI device that is actively arbitrating; the other data bits shall be released (i.e., not driven) by this SCSI device. The protection bit(s) may be released or driven to the true state, but shall not be driven to the false state during this phase."

[Note: the SPI editor may make whatever formatting changes in Table 41 that make the text more usable without loss of the content specified above.]

11 Clause 9 SCSI parallel bus timing

[This modification provides the timing tables for Fast-10:DT, Fast-20:DT, Fast-40:DT, and Fast-80:DT. The actual times are TBD, and will be filled in by the Fast-80:DT proposal.]

1) **Table 42 (SCSI Bus Timing Values)** needs to add columns for Fast-10:DT, Fast-20:DT, Fast-40:DT, and Fast-80:DT.
2) The timing values for these new entries are TBD, and will be supplied in the Fast-80:DT proposal.
3) A footnote should be added to remind people that the CRC_Available signal must obey the same constraints as any other signal on the data bus (since it is mapped to DB(P) and BD(P2)).

[Note: the SPI editor may make whatever formatting changes in Table 42 that make the text more usable without loss of the content specified above.]

12 Clause 9.2 Measurement Points

Modify as follows: "The measurements points for ST and differential ACK, REQ, DATA, and PROTECTION signals are defined in this clause."

13 Clause 9.2.1 ST fast-10 data transfer rates

Modify the second sentence in the first paragraph as follows: "The rise and fall times for the ST REQ/ACK signals shall be nominally the same as for the ST DATA/PROTECTION signals."

14 Clause 9.2.2 ST fast-20 data transfer rates

Modify the second sentence in the first paragraph as follows: "The rise and fall times for the ST REQ/ACK signals shall be nominally the same as for the ST DATA/PROTECTION signals."

15 Clause 9.2.3 LVD

1) Modify the second sentence in the first paragraph as follows: "The rise and fall times for the LVD REQ/ACK signals shall be nominally the same as for the LVD DATA/ PROTECTION signals."
2) In **Figure 45 -- LVD timing measurement points**, replace "PARITY" with "PROTECTION"

16 Clause Section 9.2.4 HVD

[Until the committee officially removes HVD from SPI-3, I felt it prudent to continue to include it in the modifications.]

1) Modify the second sentence in the first paragraph as follows: "The rise and fall times for the HVD REQ/ACK signals shall be nominally the same as for the HVD DATA/ PROTECTION signals."
2) In **Figure 46 -- HVD timing measurement points**, replace "PARITY" with "PROTECTION"

17 Clause 9.3.1 Fast-10 data transfer rates

1) Modify the second paragraph as follows: "The receiver skew is the maximum difference in propagation time between any two receivers on the REQ, REQQ, ACK, ACKQ, DATA BUS or protection signals of the same SCSI bus when external receivers are used."
2) Modify the third paragraph as follows: "The transmitter skew is the maximum difference in propagation time between any two transmitters on the REQ, REQQ, ACK, ACKQ, DATA BUS or protection signals of the same SCSI bus when external transmitters are used."

18 Clause 9.3.2 Fast-20 data transfer rates

1) Modify the third paragraph as follows: "The receiver skew is the maximum difference in propagation time between any two receivers on the REQ, REQQ, ACK, ACKQ, DATA BUS or protection signals of the same SCSI bus when external receivers are used."
2) Modify the fourth paragraph as follows: "The transmitter skew is the maximum difference in propagation time between any two transmitters on the REQ, REQQ, ACK, ACKQ, DATA BUS or protection signals of the same SCSI bus when external transmitters are used."

19 Clause 9.3.4 DT data transfer rates

[This modification provides the system level timing budget for Fast-10:DT, Fast-20:DT, Fast-40:DT, and Fast-80:DT. The actual times for Fast-80:DT are actually TBD, and are subject to change by the Fast-80:DT proposal.]

[This new section should be added to compliment the previous sections. I suggest using the same format as section 9.3.3, but limited to only Integrated Transmitters and Receivers, and, for completeness, specifying the budget for the various defined DT speeds (Fast-10:DT, Fast-20:DT, Fast-40:DT, and Fast-80:DT. The actual times for Fast-80:DT are actually TBD, and are subject to change by the Fast-80:DT proposal.]

20 Clause 11.1 SCSI Bus Phases

[These modifications are to SPI-3, which is identical in wording to that in the packetized SCSI proposal, T10/97-230r6.]

20.1 Delete: "g) INFORMATION UNIT phase"

20.2 Modify the second paragraph as follows:

"The COMMAND phase, DATA phase, , STATUS phase, and MESSAGE phase are collectively termed the information transfer phases."

21 Clause 11.1.2 ARBITRATION phase

Modify the last paragraph as follows: "Protection is not valid during the ARBITRATION phase. During the ARBITRATION phase, DB(P), DB(P1) (if present), DB(P2) (if present), and DB(P3) (if present), may be released or asserted, but shall not be actively driven false."

22 Clause 11.1.3 SELECTION phase

[This modification is extensive since the packetized SCSI proposal, T10/97-230r6 restructured this clause in SPI-2. This wording is a modification of the wording for the section in SPI-3, which is identical to that in the packetized SCSI proposal. The intent is to make the information unit phases into references to information unit protocol.]

[Note that parity during SELECTION was deleted in the packetized SCSI proposal, but will be reinstated. The text below is still the original text from the packetized SCSI and SPI-3 rev (the SPI-3 editor has agreed to fix the wording in the next revision of SPI-3).]

The following are modifications that should be made to SPI-3 (or the corresponding sections documented in T10/97-230r6):

11.3.1.1 Selection with ATN

The initiator shall set the DATA BUS to a value that is the OR of its SCSI ID bit and the target's SCSI ID bit and assert the ATN signal (indicating that a MESSAGE OUT phase is to follow the SELECTION phase). The initiator shall then wait at least two system deskew delays and release the BSY signal. The initiator shall then wait at least a bus settle delay before looking for an assertion of the BSY signal from the target.

If the information protocol is disabled for the connecting initiator the target shall follow the phase sequences defined in clause 11.3.1.

If the information protocol is enabled for the connecting initiator the target shall indicate a protocol error by performing an unexpected bus free.

11.3.1.2 Selection without ATN

The initiator shall set the DATA BUS to a value that is the combination of its SCSI ID bit and the target's SCSI ID bit and negate the ATN signal (indicating that a DT DATA OUT phase is to

follow the SELECTION phase). The initiator shall then wait at least two deskew delays and release the BSY signal. The initiator shall then wait at least a bus settle delay before looking for an assertion of the BSY signal from the target.

If the information protocol is disabled for the connecting initiator the target shall follow the phase sequences defined in clause 11.3.3.

If the information protocol is enabled for the connecting initiator the target shall follow the phase sequences defined in clause 11.3.2.

If an initiator, when selecting without ATN, detects an unexpected COMMAND phase it should invalidate any prior IUTR with the selected target. In this case, the initiator shall assert the ATN signal and on the corresponding MESSAGE OUT phase shall issue an ABORT TASK message. On the next selection with the invalidated target the initiator should do a selection with ATN and negotiate to enable the information protocol.

23 Clause 11.1.4.1 RESELECTION

23.1   Modify the third sentence of the first paragraph as follows:

 "The winning SCSI device shall also set the DATA BUS to a value that is the logical OR of its SCSI ID bit and the initiator's SCSI ID bit and the appropriate protection bit."

23.2   Modify the fourth sentence of the second paragraph as follows:

 "The initiator shall not respond to a RESELECTION phase if there is a data protection error (see 8.2)."

24 Clause 11.1.5 Information transfer phases

Modify the first paragraph as follows: [new sentence was added since it is the current SPI-3 wording]

11.1.5 Information transfer phases

The COMMAND, DATA, STATUS, and MESSAGE phases are all grouped together as the information transfer phases because they are all used to transfer data or control information via the DATA BUS. The actual content of the information is beyond the scope of this section.

[The following was added to complement the changes made in section 24, clause 11.1.5.2.2 – please see that section for details.]

Insert this paragraph at the end of the section:

The target shall not transition into an information transfer phase unless the REQx/ACKx signals are in their deasserted state. The target shall not transition from an information transfer phase into another information transfer phase unless the REQx/ACKx signals are in their deasserted state.

25 Clause 11.1.5 Information transfer phases

[This wording is a modification of the wording for the section in SPI-3, which is identical to that in the packetized SCSI proposal. The intent is to remove the INFORMATION UNIT phases and replace them with the DT DATA phases.]

1) In Table 43 (Information Transfer Phases), in the 1st and 2nd row replace the phrase "Data phase" in the Comment column with the phrase "Single-edge clocked Data phase".
2) In Table 43 (Information Transfer Phases), in the 5th row (with the first three entries 1, 0, 0), replace the phrase "INFORMATION UNIT OUT" in the Phase column with the phrase "DT DATA OUT".
3) In Table 43 (Information Transfer Phases), in the 6th row (with the first three entries 1, 0, 1), replace the phrase "INFORMATION UNIT IN" in the Phase column with the phrase "DT DATA IN".
4) In Table 43 (Information Transfer Phases), in the 5th and 6th row replace the phrase "Information unit phase" in the Comment column with the phrase "Double-edge clocked Data phase".

26 Clause 11.1.9 Information transfer phases

[This modification is extensive since the packetized SCSI proposal, T10/97-230r6 restructured this clause in SPI-2. This wording is a modification of the wording for the section in SPI-3, which is identical to that in the packetized SCSI proposal. The intent is to make the information unit phases into references to information unit protocol.]

The following are modifications that should be made to the document T10/97-230r6 (all references are to the clause numbers in that document):

11.1.9 Information unit protocol

The information unit protocol is optional and is only used in DT DATA IN and DT DATA OUT phases. It shall be used in a data phase if an information unit has been established (see xxx).

11.1.9.1 INFORMATION UNIT IN protocol exception condition handling

The initiator shall not assert the ACK for the last byte of the CRC of any information unit until the CRC has been verified to be correct. If the initiator detects a parity error on any byte or a CRC error in any information unit it receives, the initiator shall create an attention condition by asserting the ATN signal before the ACK signal is released for the last byte of CRC. When the target switches to a MESSAGE OUT phase the initiator shall send an INITIATOR DETECTED ERROR message (see xxx) to the target. This message notifies the target that data in that information unit was invalid. If the target does not retry transmitting that information unit, or it exhausts its retry limit, it shall send a SPI L_Q/Status information unit pair to the initiator with a CHECK CONDITION status and a sense key set to ABORTED COMMAND and an additional sense code set to INITIATOR DETECTED ERROR MESSAGE RECEIVED for the task associated with the received INITIATOR DETECTED ERROR message. (48/00)

11.1.9.2 INFORMATION UNIT OUT protocol exception condition handling

The target shall only respond to a parity error or CRC error after all the data in an information unit has been received. If the nexus has been fully identified (i.e., an I_T_L_Q nexus has been established) and the target detects a parity error on any byte or a CRC error in any information

unit it receives, and the target does not retry transmitting that information unit, or it exhausts its retry limit, the target shall send a SPI L_Q/SPI status information unit pair to the initiator with a CHECK CONDITION status and a sense key set to ABORTED COMMAND and the additional sense code set to SCSI PARITY ERROR for the task associated with the parity error or CRC error. (47/00) If the target is receiving a SPI L_Q information unit and the target detects a parity error (i.e., the nexus identification fails) on any byte or a CRC error the target shall cause an unexpected bus free by generating a BUS FREE phase (see xxx).

27 Clause 11.1.5.2 Synchronous data transfer

Replace clause 11.1.5.2 Synchronous data transfer with the following clause with four sub-clauses. The first subclause defines single transition clocked synchronous data transfer; the second defines double-edge clocked synchronous data transfer; the third defines transfer of CRC; and, the fourth defines timings:

[changes are noted in 11.1.5.2.1, since it is a rewrite of the existing 11.1.5.2, but the latter sections are all new and so have no changes explicitly noted.]

11.1.5.2 Synchronous data transfer

11.1.5.2.1 Single-edge [transition] clocked synchronous data transfer

[The following modification is to define single transition clocked synchronous data transfer. What was clause 11.1.5.2 has been demoted and slightly modified for single transition clocked. The changes from the original clause are underlined. The SPI-3 editor may find an easier way to combine this section with the latter double-edge clocked section, but for the moment this seemed the approach that would insure the greatest degree of clarity.]

Single-edge clocked synchronous data transfer is optional and is only used in the ST DATA IN and ST DATA OUT data phases. It shall be used in a data phase if a single transition clocked synchronous data transfer agreement has been established (see xxx). The agreement specifies the REQx/ACKx offset and the minimum transfer period.

The REQx/ACKx offset specifies the maximum number of REQx assertions that shall be sent by the target in advance of the number of ACKx assertions received from the initiator, establishing a pacing mechanism. If the number of REQx assertions exceeds the number of ACKx assertions by the REQx/ACKx offset, the target shall not assert the REQx signal until after the next ACKx assertion is received. For successful completion of the data phase is that the number of ACKx and REQx assertions shall be equal.

The target shall assert the REQx signal for a minimum of an assertion period. The target shall then wait at least the greater of a transfer period from the last transition of the REQx signal to true or a minimum of negation period from the last transition of the REQx signal to false before again asserting the REQx signal.

The initiator shall assert the ACKx signal for each REQx assertion received. The ACKx signal may be asserted as soon as the corresponding REQx assertion has been received. The initiator shall assert the ACKx signal for a minimum of an assertion period. The initiator shall wait at least the greater of a transfer period from the last transition of the ACKx signal to true or for a minimum of a negation period from the last transition of the ACKx signal to false before asserting the ACKx signal.

[Although I left them in, the following two paragraphs are subject to the same concern as the similar sections in the DT section. – reword for setup and hold time]

If the I/O signal is true (transfer to the initiator), the target shall first drive the DB(7-0,P), DB(15-0,P,P1), or DB(31-0,P,P1,P2,P3) signals to their desired values, wait at least the setup time, then assert the REQx signal. The DB(7-0,P), DB(15-0,P,P1), or DB(31-0,P,P1,P2,P3) signals shall be held valid for a minimum hold time after the assertion of the REQx signal. The target shall assert the REQx signal for a minimum of an assertion period. The target may then negate the REQx signal and change or release the DB(7-0,P), DB(15-0,P,P1), or DB(31-0,P,P1,P2,P3) signals. The initiator shall read the value on the DB(7-0,P), DB(15-0,P,P1), or DB(31-0,P,P1,P2,P3) signals within one hold time of the transition of the REQx signal to true. The initiator shall then respond with an ACKx assertion.

If the I/O signal is false (transfer to the target), the initiator shall perform a data transfer for each REQx assertion received. After detecting a REQx assertion, the initiator shall first drive the DB(7-0,P), DB(15-0,P,P1), or DB(31-0,P,P1,P2,P3) signals to their desired values, delay at least one setup time, then assert the ACKx signal. The initiator shall hold the DB(7-0,P), DB(15-0,P,P1), or DB(31-0,P,P1,P2,P3) signals valid for at least one hold time after the assertion of the ACKx signal. The initiator shall assert the ACKx signal for a minimum of an assertion period. The initiator may then negate the ACKx signal and may change or release the DB(7-0,P), DB(15-0,P,P1), or DB(31-0,P,P1,P2,P3) signals. The target shall read the value of the DB(7-0,P), DB(15-0,P,P1), or DB(31-0,P,P1,P2,P3) signals within one hold time of the transition of the ACKx signal to true.

11.1.5.2.2 Double-edge [transition] clocked synchronous data transfer

[The following addition is to define double-edge clocked synchronous data transfer. The SPI-3 editor may find an easier way to combine this subclause with the earlier single transition clocked clause, but for the moment this seemed the approach that would insure the greatest degree of clarity.]

Double-edge clocked synchronous data transfer is optional and is only used in the DT DATA IN and DT DATA OUT data phases. It shall be used in a data phase if a double-edge clocked synchronous data transfer agreement has been established (see xxx). The agreement specifies the REQx/ACKx offset and the minimum transfer period.

[[A1]] The DT DATA IN and DT DATA OUT phases are valid when using 16 or 32 bit wide buses. These phases are invalid when using 8 bit wide buses. [[A1]]

[[A2]] The DT DATA IN and DT DATA OUT phases are valid when using 16 or 32 bit wide buses. [[A2]]

[[A3]] The DT DATA IN and DT DATA OUT phases are recommended only when 16 or 32 bit buses. These phases are valid for 8 bit wide buses. [[A3]]

[[A4]] The DT DATA IN and DT DATA OUT phases are valid when using either 8, 16, or 32 bit buses. [[A4]]

[[B1]] The DT DATA IN and DT DATA OUT phases are valid when using LVD signaling. These phases are invalid when using single ended or HVD signaling. [[B1]]

[[B2]] The DT DATA IN and DT DATA OUT phases are valid when using LVD signaling. These phases are invalid when using HVD signaling. [[B2]]

[[B3]] The DT DATA IN and DT DATA OUT phases are recommended only when using LVD signaling. These phases are valid for single ended signaling. These phases are invalid when using HVD signaling. [[B3]]

Increase Transfer Rate and Improve Error Detection Proposal

[[B4]] The DT DATA IN and DT DATA OUT phases are valid when using either LVD or single ended signaling. These phases are invalid when using HVD signaling. [[B4]]

Since data shall be clocked on both assertion and negation of the REQ and ACK signal lines, a REQx/ACKx transition is either an assertion or a negation of the REQx or ACKx signal.

[At the Orange County meeting a controversy arose over the next section. It appears that the key concern was to insure that REQx/ACKx are both deasserted whenever we transition out of the DT data phases into another phase. This was sometimes phrased as requiring REQx/ACKx transition to always appear as an assertion/deassertion pair. While this would satisfy the key concern, it was felt by others in the audience that such wording would be excessively restrictive on the run time implementation of the REQx/ACKx flow control in order to handle a concern that would only occur under error conditions. A straw poll was taken, with no decisive mandate from the working group.

I have attempted to resolve this issue for the committee by the wording in the following section. It explicitly requires that REQx/ACKx by deasserted whenever the target changes out of DT data phase. The only exception is the transition to the BUS FREE phase. This exception is made for two reasons. First, the original concern over the state of REQx/ACKx is not relevant when there is a transition to the BUS FREE phase (in the BUS FREE, ARBITRATION, SELECTION, and RE-SELECTION phases the REQx/ACKx signals are not used and their state is irrelevant). If the target is prevented from transitioning into an information transfer phase until REQx/ACKx are deasserted, then we insure that the DT phases cannot inappropriately interfere with the operation of the other phases. This behavior is specified in section 24, clause 11.1.5. Note that a persistent assertion of REQx/ACKx will eventually result in an upper level timeout and SCSI bus reset, but it is clearly expected that the devices will use the phases other than the information transfer phases to deassert REQx/ACKx without side effects.

Second, there must be some legal way to allow the target to get out of the DT phase if the initiator simply refuses (for whatever reason) to deassert ACKx. Otherwise a SCSI bus hang condition would occur, which would be resolved by an upper level time out and a subsequent SCSI bus reset (which people have indicated in the past is not desirable). Going to BUS FREE is traditionally accepted as an action of last resort for the target to indicate an unrecoverable SCSI bus failure.

Given these additions, I believe the key concerns have been satisfied with a minimal restriction on implementation innovation. The secondary issue is what constitutes an "offset." In previous discussions the key concern expressed itself in the form of focus on whether an "offset" meant one bus transfer or two. Given the previous resolution of the key concern, I have worded the document to indicate that "offset" refers to one bus transfer. To a large degree this is simply a matter of nomenclature – I hope that the committee will treat it as such, and identify any operational concerns as we previously did at the Orange County meeting and address them directly and with a minimal impact on implementation freedom.]

The REQx/ACKx offset specifies the maximum number of REQx transitions that shall be sent by the target in advance of the number of ACKx transitions received from the initiator, establishing a pacing mechanism. If the number of REQx transitions exceeds the number of ACKx transitions by the REQx/ACKx offset, the target shall not transition the REQx signal until after the next ACKx transition is received. For successful completion of the data phase is that the number of ACKx and REQx transitions shall be equal, and both REQx and ACKx are negated at the end of the data phase.

After the target has transitioned the REQx signal, the target shall wait for a minimum of a transition period before transitioning the REQx signal again

The initiator has transitioned the ACKx signal for each REQx transition received. The ACKx signal may be transitioned as soon as the corresponding REQx transition has been received. After the initiator has transitioned the ACKx signal, the initiator shall then wait a minimum of a transition period.

11.1.5.2.3 CRC transfer

[The following new subclause defines how CRC is transferred.]

11.1.5.2.3.1 DT DATA IN

If the I/O signal is true (transfer to the initiator), the target shall first drive the DB(7-0,P), DB(15-0,P[[C1]],P1[[C1]]), or DB(31-0,P[[C1]],P1[[C1]],P2[[C1]],P3[[C1]]) signals to their desired values. If the target wishes to send pad field or CRC field byte(s) for the current data group (see clause xxx), then the CRC_AVAILABLE signals (P and P2) shall be asserted, the DATA BUS (DB7-0, DB15-0[[C1]], DB(P1)[[C1]], DB31-0[[C1]], DB(P3)[[C1]]) shall contain the pad field or CRC field information. Otherwise the CRC_ AVAILABLE signal shall be negated.

The initiator shall read the value on the DB(7-0,P), DB(15-0,P[[C1]],P1[[C1]]), or DB(31-0,P[[C1]],P1[[C1]],P2[[C1]],P3[[C1]]) signals within one hold time of the transition of the REQx signal. If the CRC_AVAILABLE signal is asserted, then the initiator shall not pass the byte(s) transferred to the ULP. Instead, it shall continue to use the byte(s) for computing the CRC if they are in the pad field, or if they are in the CRC field it shall assemble the bytes(s) for checking against the computed CRC for this data group (see section xxx on the rules for distinguishing a pad field from CRC field). Upon receipt of the last byte of the CRC field, the received CRC and computed CRC shall be compared. If they do match (no CRC error), then the initiator shall then respond with an ACKx transition.

If they do not match (a CRC error is detected), or if an improperly formatted data group is transferred, then the initiator shall establish an attention condition (see 11.2.1) by asserting the ATN signal before the ACKx signal is transitioned for the last byte(s) of the CRC field. When the target switches to a MESSAGE OUT phase the initiator shall send an INITIATOR DETECTED ERROR message (see 12.7.2.6) to the target. This message notifies the target that data group contained was invalid.

11.1.5.2.3.2 DT DATA OUT

If the I/O signal is false (transfer to the target), then after detecting a REQx transition, the initiator shall first drive the DB(7-0), DB(15-0[[C1]],P1[[C1]]), or DB(31-0[[C1]],P1[[C1]],P3) signals to their desired values. As byte(s) are transferred from initiator to target, the REQx transition that was received in order to allow the data transfer shall be the first transition received and not already used to allow a previous data transfer from the initiator to the target. The initiator shall remember whether each REQx transition was received with CRC_AVAILABLE asserted or with CRC_AVAILABLE negated. When a REQx transition that was received with CRC_AVAILABLE asserted is used to allow a data transfer from the initiator to the target, the information transferred shall be from the pad field or CRC field bytes for the current data group.

The target shall read the value of the DB(7-0), DB(15-0[[C1]],P1[[C1]]), or DB(31-0[[C1]],P1,P3[[C1]]) signals within one hold time of the transition of the ACKx signal. If the target expected a pad field or CRC field byte(s), then it shall not pass the byte(s) transferred to the ULP. Instead, it shall continue to use the byte(s) for computing the CRC if they are in the pad field, or if they are in the CRC field it shall assemble the bytes(s) for checking against the computed CRC for this data group (see section xxx on the rules for distinguishing a pad field from CRC field). Upon receipt of the last byte of the CRC field, the received CRC and computed CRC shall be compared.

If they do not match (an error is detected), or if an improperly formatted data group is transferred, then the associated data group shall be considered invalid. The target may request a retry of the data transfer by sending a RESTORE POINTERS message (12.7.2.11). The target may also attempt to disconnect without first sending a SAVE DATA POINTER message (12.7.2.11). The target may go into STATUS phase and send a CHECK CONDITION status and a sense key set to ABORTED COMMAND and the additional sense code set to SCSI PARITY ERROR for the task associated with the parity error or CRC error.

11.1.5.2.4 Timing

11.1.5.2.4.1 Data and CRC_Available Transfer Timing Budgets with LVD Transceivers

Table x - Data and CRC_Available Transfer Timing Budgets with LVD Transceivers

Description	Name	Timing Values (in ns)			
		Fast-10:DT	Fast-20:DT	Fast-40:DT	Fast-80:DT
Transfer Period during DT Synchronous Data Transfer Phases (note 1)	t_{DA}	TBD~~100 ns~~	TBD~~50 ns~~	TBD~~25 ns~~	TBD~~12.5 ns~~
Transmitter skew (note 3)	t_{TS}	TBD	TBD	TBD	TBD
Transmit Setup Time	t_{DST}	TBD~~48.0 ns~~	TBD~~23.0 ns~~	TBD~~10.5 ns~~	~~4.75 ns~~TBD
Transmit Hold Time	t_{DHT}	TBD~~48.0 ns~~	TBD~~23.0 ns~~	TBD~~10.5 ns~~	~~4.75 ns~~TBD
Receive Setup Time	t_{DSR}	TBD~~40.0 ns~~	TBD~~18.0 ns~~	TBD~~5.5 ns~~	TBD~~1.5 ns~~
Receive Hold Time	t_{DHR}	TBD~~40.0 ns~~	TBD~~18.0 ns~~	TBD~~5.5 ns~~	TBD~~1.5 ns~~
Signal timing skew (note 4)	t_{STS}	TBD~~8 ns~~	TBD~~5 ns~~	TBD~~5 ns~~	TBD~~3.25 ns~~
CRC_Available Transmit Setup Time (Note 5)	t_{CST}	48.0 ns	23.0 ns	23.0 ns	14.75 ns
CRC_Available Transmit Hold Time	t_{CHT}	48.0 ns	23.0 ns	10.5 ns	4.75 ns
CRC_Available Receive Setup Time (Note 5)	t_{CSR}	40.0 ns	18.0 ns	18.0 ns	11.5 ns
CRC_Available Receive Hold Time	t_{CHR}	40.0 ns	18.0 ns	5.5 ns	1.5 ns

Notes:
1) The transfer period is measured from a transition of REQ/REQQ (ACK/ACKQ) signal to the next transition edge of the signal
2) Skew is between any of the transmitter data lines and the transmitting REQ or ACK line.
3) Transmitter skew includes all clock asymmetry, IC skew, board skew and connector skew.
4) Signal timing skew includes cable skew, loading skew, and signal distortion skew.
5) In fast-40:DT and fast-80:DT modes: the CRC_Available signal is given 10.0 ns more setup time than the data signals.
6) Data setup and hold times are specified at the transmitter or receiver connector.

[Note: Fast-80:DT timing are for illustrative purposes and are subject to change by subsequent Fast-80:DT proposals.]

[Note: For comparison, T10/98-153r1 has 4.55 ns for DT Transmit Setup and DT Transmit Hold vs this proposal's 4.75 ns; 3.1 ns for Signal timing skew vs 3.25 ns in this proposal; and 1.45 ns DT Receive Setup and DT Receive Hold vs 1.5 ns in this proposal. There may be some difference in actual timings, but there also may be some difference in definition of the timings. The setup and hold timings above are taken at the transmitter and receiver, with items such as board skew put into the transmitter skew budget, while the 153r1 timings partition the skew elements differently. I suggest that a common framework for the timings be established before diving into the numbers proper.]

CRC_Available Receive hold time (CHR)

The minimum time required at the receiver between the transition of the REQx or ACKx signals and the transition of the CRC_AVAILABLE signal while using double transition clocking synchronous data transfers.

CRC_Available Receive setup time (CSR)

The minimum time required at the receiver between the transition of CRC_AVAILABLE signal and the transition of the REQx or ACKx signal while using double transition clocking synchronous data transfers.

CRC_Available Transmit hold time (CHT)

The minimum time provided by the transmitter between the transition of the REQx or ACKx signal and the transition of the CRC_AVAILABLE signal while using double transition clocking synchronous data transfers.

CRC_Available Transmit setup time (CST)

The minimum time provided by the transmitter between the transition of CRC_AVAILABLE signal and the transition of the REQ or ACKx signal while using double transition clocking synchronous data transfers.

DT Receive hold time (DHR)

The minimum time required at the receiver between the transition of the REQx ACKx signals and the changing of the DATA BUS while using double-edge clocking synchronous data transfers.

DT Receive setup time (DSR)

The minimum time required at the receiver between the changing of DATA BUS and the transition of the REQx or ACKx signal while using double-edge clocking synchronous data transfers.

DT Transmit hold time (DHT)

The minimum time provided by the transmitter between the transition of the REQx or ACKx signal and the changing of the DATA BUS while using double-edge clocking synchronous data transfers.

DT Transmit setup time (DST)

The minimum time provided by the transmitter between the changing of DATA BUS and the transition of the REQx or ACKx signal while using double-edge clocking synchronous data transfers.

11.1.5.2.4.2 Signal source timing

On a change in the signal source, delays for driving of the data or parity lines shall be in accordance with SPI-2. For example, on a change from Command phase to DT Data Out phase, the target may not drive the parity lines until after a Bus Settle Delay plus a Data Release Delay.

11.1.5.2.4.3 Timing conventions

╱ or ╲ - signal transition (asserted or negated)

⟨ or ⟩ - data transition (asserted or negated)

▭▭▭ - undefined but not necessarily released

░░░ - asserted, negated or released

·········· - the "other" condition if a signal is shown with no change
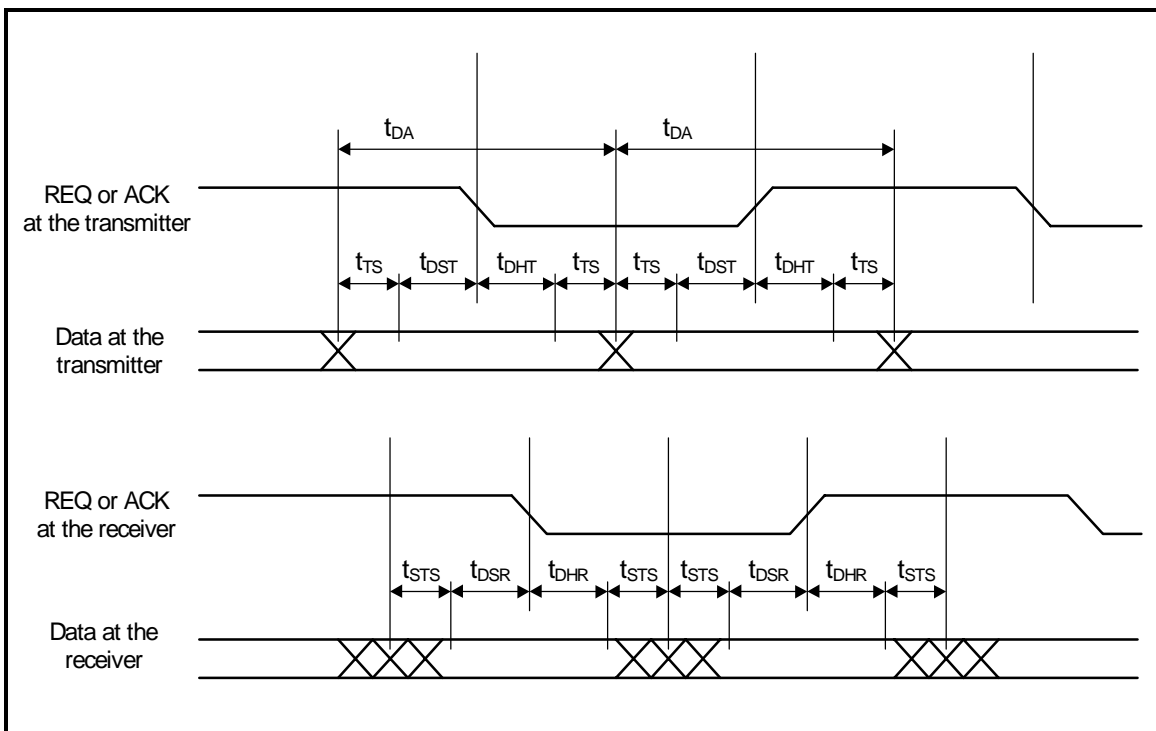
11.1.5.2.4.4 Data transfer timing

Figure 1 -- Data Transfer Timing

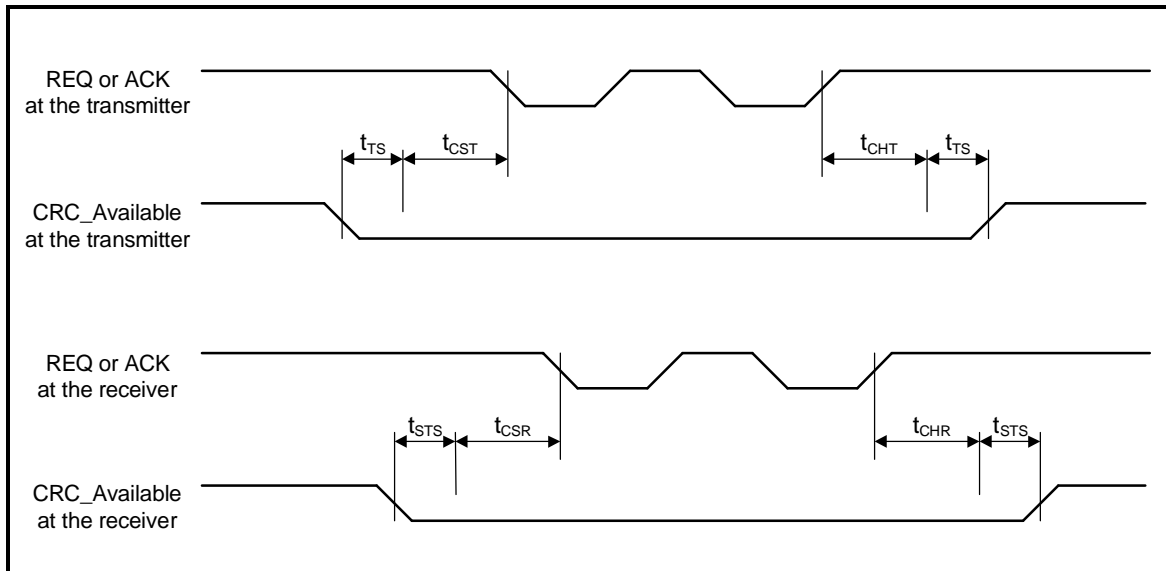### 11.1.5.2.4.5 CRC_AVAILABLE timing

Figure 2 -- CRC_AVAILABLE timing

[Note: in the second t_{STS}, there should be additional setup time to allow CRC_AVAILABLE to transition. Following the precedence for setup to CRC_AVAILABLE, it is suggested that an extra 10 ns setup be added here. – CC_AVALAIBLE setup time is 10 ns additional on either assert or deassert with request to REQ.]

28 11.1.5.3 Wide data transfer

Modify the first paragraph as follows: "Wide data transfer is optional and may be used in data phases only if a non-zero wide data transfer agreement is in effect (see 12.7.2.16 or 12.7.2.5). The agreement specifies a data path width to be used during the data phase."

Modify this clause in the list following the seventh paragraph of the section as follows: "a) The REQQ and ACKQ signals shall only be asserted during DATA phases. These signals shall not be asserted during other phases;"

29 Clause 11.1.7 Data phase

[The modifications in the following define new nomenclature for the DATA phases used throughout the document]

The following clauses should be modified or added:

11.1.7 Data phase

The data phase is a term that encompasses the DT DATA IN phase, the ST DATA IN phase, the DT DATA OUT phase, and the ST DATA OUT phase. The data phase is a term that encompasses both the DATA IN phase and the DATA OUT phase. THE ST DATA phase is a term that encompasses both the ST DATA IN phase and the ST DATA OUT phase. The DT data phase is a term that encompasses both the DT DATA IN phase and the DT DATA OUT

phase. The data in phase is a term that encompasses both the ST DATA IN phase and the DT DATA IN phase. The data out phase is a term that encompasses both the ST DATA OUT phase and the DT DATA OUT phase.

11.1.7.1 ST DATA IN phase

The ST DATA IN phase allows the target to request that data be sent to the initiator from the target. The target shall assert the I/O signal and negate the C/D and MSG signals during the REQx/ACKx handshake(s) of this phase.

11.1.7.2 ST DATA OUT phase

The ST DATA OUT phase allows the target to request that data be sent from the initiator to the target. The target shall negate the C/D, I/O, and MSG signals during the REQx/ACKx handshake(s) of this phase.

11.1.7.3 DT DATA IN phase

The DT DATA IN phase allows the target to request that data be sent to the initiator from the target using the double edge synchronous data transfer protocol. The target shall assert the I/O and MSG signals and negates the C/D signal during the REQx/ACKx handshake(s) of this phase.

11.1.7.4 DT DATA OUT phase

The DT DATA OUT phase allows the target to request that data be sent from the initiator to the target using the double edge synchronous data transfer protocol. The target shall assert the MSG signal and negate the C/D and I/O signals during the REQx/ACKx handshake(s) of this phase.

30 Clause 11.2.1 Attention condition

[This modification is to the SPI-2 document as modified by the packetized SCSI proposal, T10/97-230r6.]

Modify item (g) in paragraph five as follows:

g) If the ATN signal becomes true during a –DATA phase while the information unit protocol is in effect, the target shall enter MESSAGE OUT phase at the completion of the current SPI information unit. If The ATN signal becomes true between SPI information units the target shall enter MESSAGE OUT phase at the completion of the next SPI information unit.

31 Clause 11.3 SCSI bus phase sequences

[This modification is extensive since the packetized SCSI proposal, T10/97-230r6 restructured this clause in SPI-2. This wording is a modification of the wording for SPI-3, similar to the packetized SCSI proposal. The intent is to make the information unit phases into references to information unit protocol.]

The following are modifications that should be made to the document T10/97-230r6 (all references are to the clause numbers in that document):

Clause 11.3.1 **Phase sequences for reselection and selection with ATN and information unit protocol disabled**

The allowable sequences for a selection with ATN and reselection while a data transfer agreement is in effect that disabled information unit protocol shall be as shown in figure 53.

If a data transfer agreement is in effect that disables information unit protocol (see 12.7.2.5), the normal progression for selection with ATN (see 11.1.1.1) is from the BUS FREE phase to ARBITRATION, from ARBITRATION to SELECTION or RESELECTION, and 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 COMMAND COMPLETE message is transferred, followed by the BUS FREE phase.

In Figure 53 -- **Phase sequences for selection with ATN/reselection and information unit phases Disabled (this is figure 52 in SPI-2):**

Replace the phrase "DATA IN or DATA OUT" in the box in the figure with: "DATA".

Modify the title of the figure as follows: "Phase sequences for selection with ATN/reselection and information unit protocol Disabled".

Clause **3.3.2 Phase sequences for selection without ATN and information unit ~~protocol~~ disabled**

Modify the clause as follows:

The additional sequences for a selection without ATN while a data transfer agreement is in effect that disables information unit protocol shall be as shown in figure 54.

If a data transfer agreement is in effect that disables information unit protocol (see 12.7.2.5), the normal progression for selection without ATN (see 11.1.1.2) is from the BUS FREE phase to ARBITRATION, from ARBITRATION to SELECTION, from SELECTION to COMMAND phase, from COMMAND phase to DATA phase, from DATA phase to STATUS phase, and from STATUS phase to MESSAGE IN phase where a COMMAND COMPLETE message is transferred, followed by the BUS FREE phase.

31.1 In **Figure 54 -- Phase sequences for selection without ATN and information unit phases disabled**

Replace the phrase "DATA IN or DATA OUT" in the box in the figure with: "DATA".

Modify the title of the figure as follows: "Phase sequences for selection without ATN and information unit protocol disabled"

31.2 Clause **11.3.3 Phase sequences for selection without ATN/reselection and information unit phases enabled**

Modify the title of the clause follows: **11.3.3 Phase sequences for selection without ATN/reselection and information unit protocol enabled**

Modify the text of the clause to be as follows:

> The sequences for a selection without ATN and reselection while a data transfer agreement is in effect that enables information unit protocol shall be as shown in figure 55.

> If a data transfer agreement is in effect that enables information unit protocol (see 12.7.5.2), the normal progression for selection without ATN (see 11.1.1.2) is from the BUS FREE phase to ARBITRATION, from ARBITRATION to SELECTION or RESELECTION, and from SELECTION or RESELECTION to one or more data phases where information units are transferred. The final data phase is followed by the BUS FREE phase.

31.3 In **Figure 55 -- Phase sequences for selection without ATN/reselection and information unit phases enabled**

Replace the phrase "INFORMATION UNIT IN or INFORMATION UNIT OUT (note)" in the box in the figure with "DATA (note)"

Modify the title of the figure as follows: "Phase sequences for selection without ATN/reselection and information unit protocol enabled"

Modify the note in this figure as follows: "Note: Only information units shall be transferred within these data phases. See figure 4, figure 5, and figure 6 for the sequencing of SPI information units within the data phases"

31.4 Clause **12.1 SPI information unit sequences**

The following modifications should be made to this clause:

The information unit protocol transfers data in SPI information units. The order in which SPI information units are transferred within the information unit protocol follows a prescribed sequence.

The SPI information unit sequences shall be as shown in figure 56, figure 57, and figure 58. See figure 55 and figure 54 for the sequencing rules between the data phases and the other phases.

The normal progression is from SPI L_Q information unit/SPI command information unit pair(s), to SPI L_Q information unit/SPI data information unit pair(s), to SPI L_Q information unit/SPI status information unit pair(s).

NOTE 2 - An initiator may force a BUS FREE phase by asserting the ATN signal and sending a DISCONNECT message on the corresponding MESSAGE OUT phase. This allows an initiator to break up a long sequence of SPI L_Q information unit/SPI data information unit pairs into smaller sequences.

When a data transfer agreement is in effect that enables information unit protocol there is no option equivalent to the 'disconnect without sending a SAVE DATA POINTERS message' option for write operations. During a read operation the initiator should assert the ATN signal on a parity error or CRC error and not move the data pointers. This allows a target to do a RESTORE POINTERS message and retry the last SPI information unit. On a write, SPI protocol does not allow the target to inform the initiator if the data was properly received, therefore, the initiator shall assume the data was properly received and save the data pointers as soon as the last CRC byte is sent. To retry a write operation a target shall send a MODIFY DATA POINTERS message then request that the SPI information unit be sent again.

The target shall not start an information unit transfer until all REQ(s)/ACK(s) have been responded to by an equal number of ACK(s)/REQ(s).

31.5 In **Figure 56 -- SPI information unit sequence during initial connection**

Replace the phrase "INFORMATION UNIT OUT" with "DATA OUT" and the phrase "INFORMATION UNIT IN" with "DATA IN"

Add the note that: "Only information units shall be transferred within these data phases."

31.6 In **Figure 57 -- SPI information unit sequence during data transfers**

Replace the phrase "INFORMATION UNIT OUT" with "DATA OUT" and the phrase "INFORMATION UNIT IN" with "DATA IN"

Add the note that: "Only information units shall be transferred within these data phases."

31.7 In **Figure 58 --** SPI L_Q information unit sequence during status transfers

Replace the phrase "INFORMATION UNIT OUT" with "DATA OUT" and the phrase "INFORMATION UNIT IN" with "DATA IN"

Add the note that: "Only information units shall be transferred within these data phases."

SPI L_Q information unit sequence during status transfers

32 Clause **11.5.2.12 SYNCHRONOUS DATA TRANSFER REQUEST**

Modify paragraph 9 in this clause in SPI-2 as follows:

The SDTR message exchange establishes the permissible transfer periods and the REQ/ACK offsets for all logical units on the two SCSI devices. This agreement only applies to ST DATA phases. The IUTR message exchange establishes a similar agreement the only applies to the DT DATA phases (see section xxx). All other phases shall use asynchronous transfers.