

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: ~~June 2~~May 19, 1998

0 Overview

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: DED 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:DECDI timing information, and 153r1 does not define Fast-10:DECDI, Fast-20:DECDI, and Fast-40:DECDI. 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 DECDI (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 SECSI DATA IN, SECSI DATA OUT, DECDI DATA IN, and DECDI DATA OUT (SECSI is Single-Edge Clocked, and DECDI is Double-Edge Clocked). DATA IN generically now refers to SECSI DATA IN and DECDI DATA IN; DATA OUT to SECSI DATA OUT and DECDI DATA OUT. SECSI DATA generically refers to SECSI DATA IN and SECSI DATA OUT; DECDI DATA to DECDI DATA IN and DECDI 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 DECDI data phase, but that could be changed rather easily if people wanted to allow information units during the traditional SECSI data phase.

I did not change the IUTRPPR message section (George Penokie was going to work on that). It is important to note that this is the negotiation that established asynchronous, SECSI synchronous, or DECDI 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-edged~~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~~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:DECDT 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:DECDT, Fast-20:DECDT, and Fast-40:DECDT, which hopefully will prove to be less controversial than Fast-80:DECDT 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).

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

1.23.2 ~~double-edged~~double transition clocking (DECDT): 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 DECDT data phases to perform data transfers on both edges of the REQ or ACK.

1.33.3 fast-10:DECDT: Negotiated to receive synchronous data at a transfer rate less than or equal to a transfer rate of 10 megatransfers per second using ~~double-edged~~double transition clocking.

1.43.4 fast-20:DECDT: 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-edged~~double transition clocking.

1.53.5 fast-40:DECDT: 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-edged~~double transition clocking.

1.63.6 fast-80:DECDT: 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-edged~~**double transition** clocking.

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

~~1.8Frame Check Sequence: The CRC bits for a block of data.~~

1.93.8 Intersymbol interference (ISI): The effect of ~~preceding~~**adjacent** symbols on the symbol currently being received.

1.103.9 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.-

1.113.10 Single-edged clocking (SECST): 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 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.

~~4.123.11~~ 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 byte plus the parity bit equal to an odd number (1, 3, 5, 7, or 9). See ~~xxx3-1-19~~, parity bit.

~~4.133.12~~ Make the following modification :

parity bit: A bit associated with a transfer on the DATA BUS a byte that is used to detect the presence of single-bit errors within the transfer byte. The parity bit is driven such that the number of logical ones in the transfer byte 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
DECDT	double-edged double transition clocked
FCS	Frame Check Sequence
ISI	Intersymbol Interference
MTps	Megatransfers per second
SECST	single-edged single transition or single-edge clocked

5 Clause 4.1.1 Data Transfer Modes

This clause should be modified as follows:

SCSI parallel interface devices default to 8-bit asynchronous transfer. The 8-bit asynchronous information transfer mode is always used for all information transfers except ~~the DATA IN phases and DATA OUT phases~~. DATA IN phases and DATA OUT phases may use asynchronous or synchronous transfers that may be 8-bits, 16-bits or 32-bits wide, if a synchronous transfer agreement or a wide transfer agreement is in effect. ~~DEC DT DATA phases shall use double-edge clocked synchronous transfers if a double-edge clocked synchronous transfer agreement is in effect not use asynchronous data transfers~~. These double-edge clocked synchronous transfers may be 8 bits wide, or, if a wide transfer agreement is in effect, 16 bits or 32 bits wide.

[This is an award paragraph that probably needs restructuring (we are combining a discussion of transfer protocol and transfer width in a potentially confusing manner). I would like to revisit why we allow asynchronous data transfers in the ST DATA phases but not in the DT data phases.]

6 Clause 4.1.2 Cables, Connectors, Signals, Transceivers

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

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

- 1) Insert a row between "LVD" and "HVD" labeled "~~DEC 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 ~~DEC DT~~ data phases as well as the MESSAGE phase. The signal will still be known as the MSG signal.]

Modify the following as indicated:

- 1) ~~C/D (CONTROL/DATA). A signal sourced by a target that indicates whether control or data information (i.e. command, status, message) is on the DATA BUS. True indicates CONTROL.~~
- 2) ~~MSG (MESSAGE). A signal sourced by a target to indicate the MESSAGE phase. During the MESSAGE phase either 1) single-edge clocked data, command or status information is on the DATA bus, or 2) double-edge clocked data or message information is on the DATA BUS. True indicates DEC DATA or MESSAGE.~~

[It was suggested that the above be removed entirely because the individual signal lines do not have simple meanings anymore – there are other tables in the document that translate the combination of phase control line states to actual phases.]

[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).]

- 3) **DB(7-0,P) (8-bit DATA BUS)**. Eight data-bit signals, plus a parity-bit protection signal ~~that form that forms~~ the 8-bit DATA BUS. ~~DB(P) shall contain odd parity for DB(7-0).~~ Bit significance and priority during arbitration are shown in table 39.
- 4) **DB(15-0,P,P1) (16-bit DATA BUS)**. Sixteen data-bit signals, plus two parity-bit protection signals that form the 16-bit DATA BUS. ~~DB(P,P1) shall contain odd parity for DB(7-0) and DB(15-8), respectively.~~ Bit significance and priority during arbitration are shown in table 39.

- 5) **DB(31-0,P,P1,P2,P3) (32-bit DATA BUS)**. Thirty-two data-bit signals, plus four parity-bit protection signals that form the 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.~~ 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.]

~~[[??Question: is parity still specified for SELECTION and RESELECTION? SPI-3 appears to say it is for RESELECTION, but not for SELECTION ???]]~~

“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. DB(P) shall be used as the CRC_Available signal in DECDT DATA IN and DECDT DATA OUT phases.

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 DECDT DATA IN and DECDT DATA OUT phases DB(P) shall be used as the CRC_Available signal and DB(P1) shall contain odd parity for DB(15-0).

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 DECDT DATA IN and DECDT DATA OUT phases DB(P,P2) shall be used as CRC_Available signals and DB(P1,P3) shall contain odd parity for DB(15-0) and DB(31-16) respectively.

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 either 8-bit or 16 bit 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), and the secondary bus is always 16 bits wide.

[???Question: It has been suggested that CRC be restricted to just the 16 bit and 32 bit buses (i.e. no option for a narrow bus). I have written the proposal to include narrow – my worry is that there are too many low speed devices that would benefit from CRC that still use the narrow bus (e.g. CD-ROM/DVD). ???]

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 (~~a Frame Check Sequence~~) are transferred with ~~each block of~~ 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, ~~and has a 2⁻³² rate of undetected random error patterns.~~

8.2.2.2 Application

CRC is used in the DECDT DATA IN and DECDT 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 CRC bytespad and CRC fields -when requested by the target.

8.2.2.3 ~~Data Information Frame Group~~ -Format

~~A data frame~~A data group -consists of a data field followed by a ~~CRC field pad field and a CRC field~~. A framegroup -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 framegroup. This is the result of a frame 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 (two for an 8-bit bus, or four 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) -Any pad bytes necessary to create a frame with an even number of transfers are included at the end of the CRC field.

The pad field shall consist of 0 to 3 bytes as follows:

- 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
- 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

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

<u>Data Field</u>	
<u>Associated with REQ sent with CRC Available Negated</u>	<u>Associated with REQ sent with CRC Available Asserted</u>

<u>Pad Field</u>	<u>CRC Field</u>
------------------	------------------

<u>0 to N bytes</u>	<u>0 to 3 bytes</u>	<u>4 bytes</u>
<u>Even R/A Transitions</u>		<u>Even R/A Transitions</u>
<u>Four Byte Multiple</u>		<u>Four Byte Multiple</u>

Figure xxx: Fields within a Data Group

~~The data field consists of data bytes and any data pad bytes required in the last transfer with valid data bytes (e.g., one data pad byte would be transferred when transferring an odd number of data bytes over a 16-bit wide bus; up to three data pad bytes could be transferred at the end of a transfer over a 32-bit wide bus). The recommended size of a data field is the device block size. Data pad bytes shall be all zeros. The data field is transferred with CRC_Available negated.~~

~~The CRC field consists of four CRC bytes, followed by any CRC pad bytes necessary to result in the even number of transfers required to complete a frame (e.g., a frame transfer on an 8-bit bus may require one CRC pad byte to be transferred to result in an even number of transfers; a frame transfer on an 16-bit bus may require two CRC pad bytes to be transferred to result in an even number of transfers). CRC pad transfers shall be all zeros. The CRC field is transferred with CRC_Available asserted.~~

~~[???Question: It has been suggested the pad bytes be transferred before the CRC bytes rather than afterward. It has also been suggested that instead of transferring enough pad bytes to insure an even number of transfers, the number of pad bytes transferred be enough to insure a multiple of four bytes is transferred during a data frame. There is no difference for 16 or 32 bit buses, but for a 8 bit bus this could result in three pad bytes rather than one. The argument is that this allows the explicit transfer of all the data needed to insure 32 bit CRC calculation, without the need for the receiving device to pad out the received data for the purposes of CRC generation and checking (see 8.2.2.5). ???]~~

~~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. Only one CRC field may be requested or transmitted in any REQ/ACK offset period.~~

~~A zero length data transfer shall not transfer a CRC field.~~

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~~for the CRC field~~. The CRC_Available signal is negated to indicate a REQ associated with byte(s) in a data field of a data group~~for data~~. CRC_Available is transmitted on DB(P) for an 8-bit or 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

The 32-bit CRC is transmitted in the following byte order: 8-bit DATA BUS - bits [7:0] [15:8], [23:16], [31:24]); 16 bit DATA BUS: bits [15:0], then [31:16].

[Need to be altered – still under investigation]

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 data bytes, (which are as a multiple of four bytes), by the generator polynomial, modulo two. The remainder is generated 32 bits at a time. ~~If the data bytes are not a multiple of four bytes, the data is padded with zeros to a multiple of four bytes for the purposes of the remainder generation. These pad bytes are not transferred on the bus.~~

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

~~Data B~~bytes are bit reversed prior to generating the remainder and the bytes of the remainder are bit reversed prior to forming the CRC field~~Frame Check Sequence (FCS).~~

The CRC Field~~Frame Check Sequence~~ is the ones complement of the bit reversed remainder.

~~The four CRC bytes of the CRC field are the Frame Check Sequence.~~

A unique remainder is generated by an error free group frame consisting of the data field and CRC field. The unique remainder polynomial of an error free group frame 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

A data field of 32 bytes of all zeros shall generate a CRC of 0xAD550A19.

A data field of 32 bytes of all ones shall generate a CRC of 0x0BAB6CFF.

A data field of 32 bytes counting from 0x01 to 0x20 shall generate a CRC of 0x25ECE687.

9 Clause 8.4 OR-Tied Signals

Modify the third sentence in the second paragraph as follows: “Parity 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 **DECDI** DATA IN, and INFORMATION UNIT OUT with **DECDI** 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 “**DECDI** 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 “**DECDI** 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 parity 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:**DECDI**, Fast-20:**DECDI**, Fast-40:**DECDI**, and Fast-80:**DECDI**. The actual times are TBD, and will be filled in by the Fast-80:**DECDI** proposal.]

- 1) **Table 42 (SCSI Bus Timing Values)** needs to add columns for Fast-10:**DECDI**, Fast-20:**DECDI**, Fast-40:**DECDI**, and Fast-80:**DECDI**.
- 2) The timing values for these new entries are TBD, and will be supplied in the Fast-80:**DECDI** 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 **SECSI** and differential ACK, REQ, DATA, and **PARITY PROTECTION** signals are defined in this clause.”

13 Clause 9.2.1 **SECSI** fast-10 data transfer rates

Modify the second sentence in the first paragraph as follows: “The rise and fall times for the **SECSI** REQ/ACK signals shall be nominally the same as for the **SECSI** DATA/**PARITY PROTECTION** signals.”

14 Clause 9.2.2 **SECSI** fast-20 data transfer rates

Modify the second sentence in the first paragraph as follows: “The rise and fall times for the **SECSI** REQ/ACK signals shall be nominally the same as for the **SECSI** DATA/**PARITY 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/~~PARITY~~ 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/~~PARITY~~ 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 ~~parity~~ 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 ~~parity~~ 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 ~~parity~~ 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 ~~parity~~ protection signals of the same SCSI bus when external transmitters are used.”

19 Clause 9.3.4 ~~DECDT~~ data transfer rates

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

~~4.120.1~~ Delete: “g) INFORMATION UNIT phase”

~~4.220.2~~ Modify the second paragraph as follows:

“The COMMAND phase, DATA phase, ~~INFORMATION UNIT 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: “~~Parity Protection~~ is not valid during the ARBITRATION phase. During the ARBITRATION phase, DB(P), DB(P1) (if present), ~~and~~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 0 (I am assuming that the editor will fix the wording in rev 1).]

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 unit phases are protocol is disabled for the connecting initiator the target shall follow the phase sequences defined in clause 11.3.1.

If the information unit phases are 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 ~~INFORMATION UNIT OUT~~ DEGDT 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 ~~unit-phases-are-protocol-is~~ disabled for the connecting initiator the target shall follow the phase sequences defined in clause 11.3.3.

If the information ~~unit-phases-are-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 ~~unit-phases-are-protocol-is~~.

23 Clause 11.1.4.1 RESELECTION

~~4.123.1~~ 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 parity~~ bit.”

~~4.223.2~~ Modify the fourth sentence of the second paragraph as follows:

“The initiator shall not respond to a RESELECTION phase if ~~bad parity is detected~~ there is a data protection error (see 8.2).”

24 Clause 11.1.5 Information transfer phases

Modify the first paragraph as follows:

11.1.5 Information transfer phases

The COMMAND, DATA, ~~INFORMATION UNIT~~, 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.

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 ~~DEQDI~~ 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 “~~DEQDI~~ 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 “~~DEQDI~~ 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 ~~phase-protocol~~

~~The information unit phase is a term that encompasses both the INFORMATION UNIT IN phase and the INFORMATION UNIT OUT phase.~~

The information unit protocol is optional and is only used in ~~DEGDT DATA IN~~ and ~~DEGDT 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 phase~~

~~The INFORMATION UNIT IN phase allows the target to request that information units be sent to the initiator from the target.~~

~~The target shall assert the I/O and MSG signals and negate the C/D signal during the REQx/ACKx handshake(s) of this phase.~~

~~11.1.9.1.1~~ 11.1.9.1 INFORMATION UNIT IN ~~phase-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, ~~while in the INFORMATION UNIT IN phase~~ 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 ~~the INFORMATION UNIT IN phase that information unit~~ was invalid. If the target does not retry ~~the INFORMATION UNIT IN phase 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 phase~~

~~The INFORMATION UNIT OUT phase allows the target to request that information units be sent from the initiator to the target.~~

~~The target shall negate the C/D and I/O signals and assert the MSG signal during the REQx/ACKx handshake(s) of this phase.~~

~~4.4.9.2.4~~ **11.1.9.2 INFORMATION UNIT OUT phase 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, ~~while in the INFORMATION UNIT OUT phase~~ and the target does not retry ~~the INFORMATION UNIT OUT phase 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-edgesingle 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 clocked synchronous data transfer

[The following modification is to define single-edgesingle transition clocked synchronous data transfer. What was clause 11.1.5.2 has been demoted and slightly modified for single-edgesingle 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-edgesingle transition clocked synchronous data transfer agreement has been established (see xxx 6-6-21). 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.]

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 one system deskew delay plus one cable skew, 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 of one system deskew delay plus one cable skew plus one 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 ~~one byte~~ 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 system deskew delay plus one cable skew, 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 system deskew delay plus one cable skew plus 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 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-edgesingle 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 DECDT DATA IN and DECDT 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.

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.

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 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 a transfer period from the last transition of the REQx signal before again transitioning the REQx signal.

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. The initiator shall wait at least the greater of a transfer period from the last transition of the ACKx signal or for a minimum of a transition period from the last transition of the ACKx signal to false before transitioning the ACKx signal.

11.1.5.2.3 CRC transfer

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

11.1.5.2.3.1 ~~DECDT~~ 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,P1), or DB(31-0,P,P1,P2,P3) signals to their desired values. If the target wishes to send pad field or CRC field byte(s) for the current data ~~group frame~~ (see clause xxx), then the CRC_AVAILABLE signals (P and P2) shall be asserted, the DATA BUS (DB7-0, DB15-0, DB(P1), DB31-0, DB(P3)) shall contain the pad field or CRC field information. Otherwise the CRC_AVAILABLE signal shall be negated.

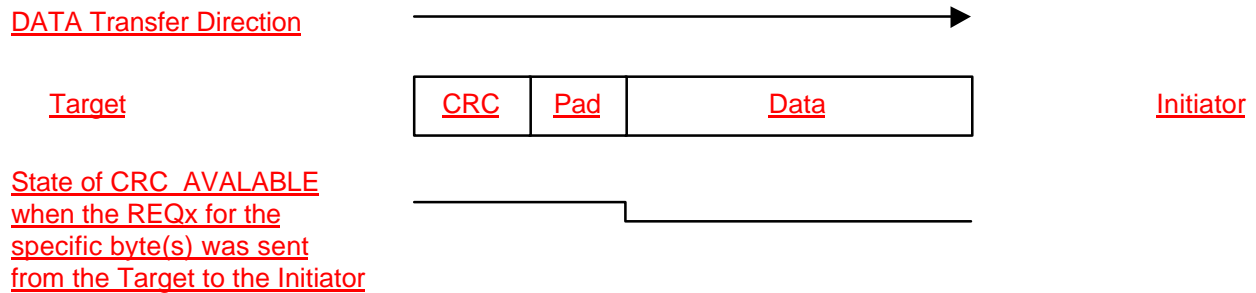


Figure xxx: CRC_AVAILABLE and Data Group fields for DATA IN

[The following section is currently in SPI, and was eliminated due to comments from the T10 members a at the Chicago meeting.]

~~The target shall then wait at least one system deskew delay plus one cable skew, then transition 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 of one system deskew delay plus one cable skew plus one hold time after the transition of the REQx signal. The target shall transition the REQx signal for a minimum of a transition period. The target may then transition 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. 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), check the CRC and parity (if any) for an error. If no error is detected, then the initiator shall then respond with an ACKx transition.

If they do not match (a CRC~~n~~ error is detected), 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 frame-contained was invalid.

11.1.5.2.3.2 DECDT 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,P1), or DB(31-0,P1,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 groupframe.

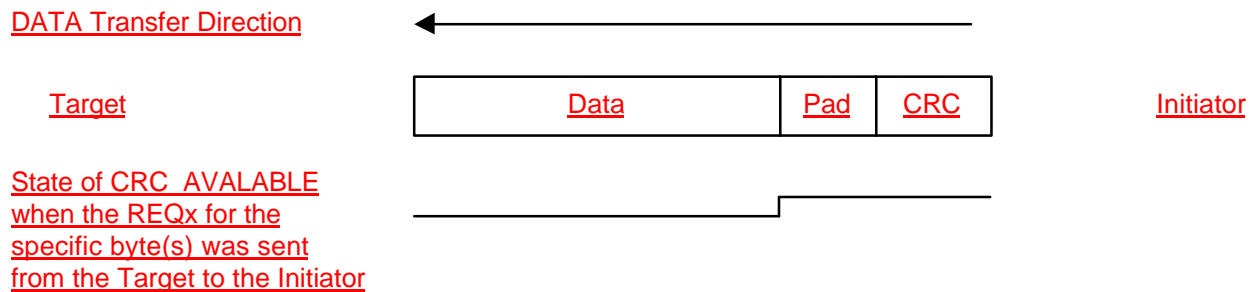


Figure xxx: CRC_AVAILABLE and Data Group fields for DATA OUT

[The following section is currently in SPI, and was eliminated due to comments from the T10 members a at the Chicago meeting.]

~~The initiator shall then delay at least one system deskew delay plus one cable skew, then transition the ACKx signal. The initiator shall hold the DB(7-0), DB(15-0,P1), or DB(31-0,P1,P3) signals valid for at least one system deskew delay plus one cable skew plus one hold time after the transition of the ACKx signal. The initiator shall transition the ACKx signal for a minimum of a~~

~~transition period. The initiator may then transition the ACKx signal and may change or release the DB(7-0), DB(15-0,P1), or DB(31-0,P1,P3) signals.~~

The target shall read the value of the DB(7-0), DB(15-0,P1), or DB(31-0,P1,P3) 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. ~~then the target shall check the CRC and parity (if any) for an error.~~

If they do not match (an error is detected), 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: DEC <u>DT</u>	Fast-20: DEC <u>DT</u>	Fast-40: DEC <u>DT</u>	Fast-80: DEC <u>DT</u> <u>I</u>
Transfer Period during DEC <u>DT</u> Synchronous Data Transfer Phases (note 1)	t _{DA}	100 ns	50 ns	25 ns	12.5 ns
Transmitter skew (note 3)	t _{TS}	2.0 ns	2.0 ns	2.0 ns	1.5 ns
Transmit Setup Time	t _{DST}	48.0 ns	23.0 ns	10.5 ns	4.75 ns
Transmit Hold Time	t _{DHT}	48.0 ns	23.0 ns	10.5 ns	4.75 ns
Receive Setup Time	t _{DSR}	40.0 ns	18.0 ns	5.5 ns	1.5 ns
Receive Hold Time	t _{DHR}	40.0 ns	18.0 ns	5.5 ns	1.5 ns
Signal timing skew (note 4)	t _{STS}	8 ns	5 ns	5 ns	3.25 ns
CRC_Available Transmit Setup Time (Note 5)	t _{CST}	48.0 ns	23.0 ns	23.0 ns	14.75 <u>7.25</u> 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	11.53 <u>7.5</u> 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:~~DEC~~DT and fast-80:~~DEC~~DT modes: the CRC_Available signal is given ~~10.02~~5 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:~~DEC~~DT timing are for illustrative purposes and are subject to change by subsequent Fast-80:~~DEC~~DT proposals.]

[Note: For comparison, T10/98-153r1 has 4.55 ns for ~~DEC~~DT Transmit Setup and ~~DEC~~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 ~~DEC~~DT Receive Setup and ~~DEC~~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-edged~~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-edged~~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-edged~~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-edged~~double transition clocking synchronous data transfers.

~~DECDT~~ 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.

~~DECDT~~ 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.

~~DECDT~~ 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.

~~DECDT~~ 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 ~~DECDT~~ 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)
- XXXX - undefined but not necessarily released
- XXXX - asserted, negated or released
- - the "other" condition if a signal is shown with no change

11.1.5.2.4.4 Data transfer timing

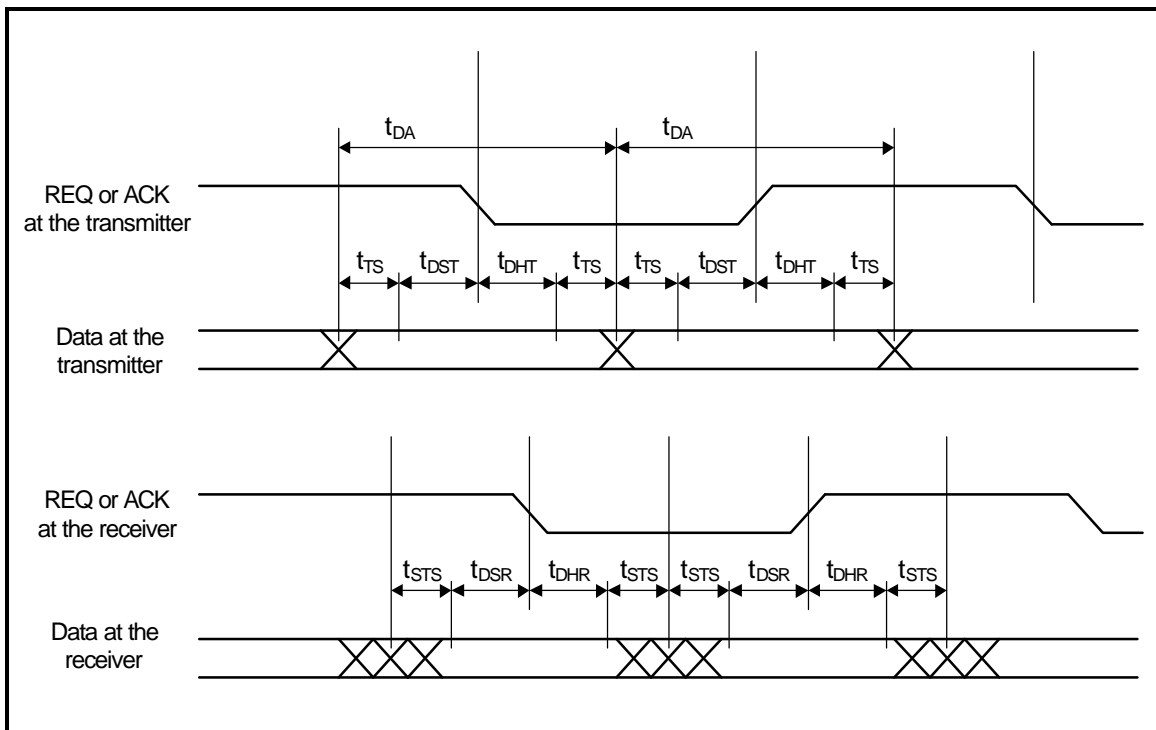


Figure 1 -- Data Transfer Timing

11.1.5.2.4.5 CRC_AVAILABLE timing

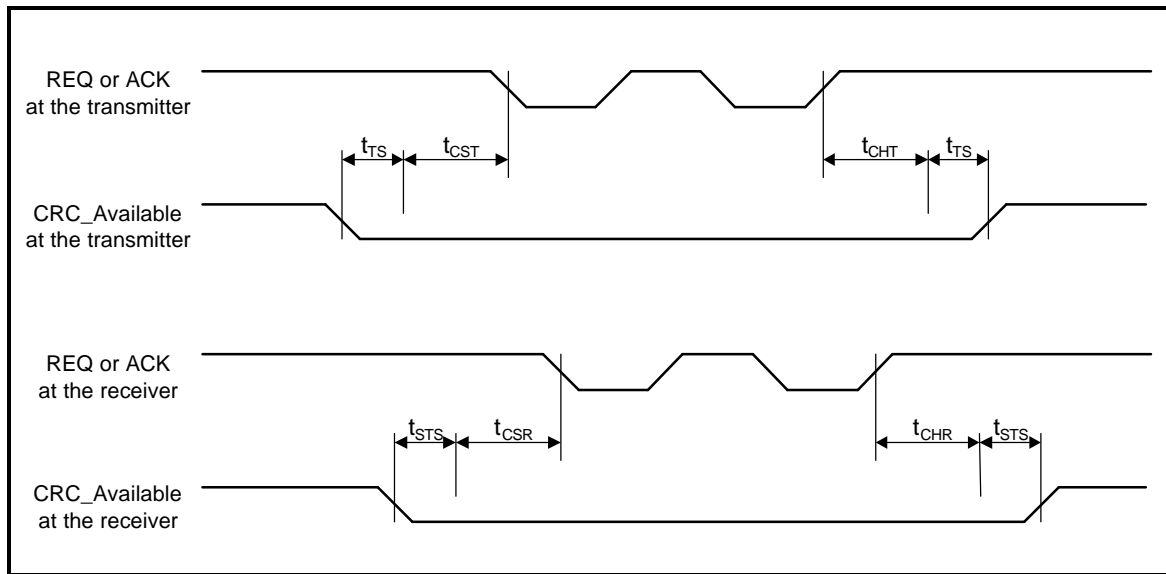


Figure 2 -- CRC_AVAILABLE timing

[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.]

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 **and information 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 **and the information 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 **IN and DATA OUT** 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 **both** the DECDT DATA IN phase, the ST DATA IN phase, the DECDT 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 DECDT data phase is a term that encompasses both the DECDT DATA IN phase and the DECDT DATA OUT phase. The data in phase is a term that encompasses both the ST DATA

[IN phase and the ~~DECDT~~ DATA IN phase. The data out phase is a term that encompasses both the ~~ST~~ DATA OUT phase and the ~~DECDT~~ DATA OUT phase.](#)

11.1.7.1 ~~SEGST~~ DATA IN phase

The ~~SEGST~~ 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 ~~SEGST~~ DATA OUT phase

The ~~SEGST~~ 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 ~~DECDT~~ DATA IN phase

~~The ~~DECDT~~ 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 ~~DECDT~~ DATA OUT phase

~~The ~~DECDT~~ 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 an [INFORMATION UNIT 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 ~~phases protocol~~ disabled**

The allowable sequences for a selection with ATN and reselection while a data transfer agreement is in effect that disables information unit [phases-protocol](#) shall be as shown in figure 53.

If a data transfer agreement is in effect that disables information unit [phases-protocol](#) (see 5.0.1.1 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, ~~excluding information unit 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 [phases-protocol](#) Disabled".

Clause 3.3.2 Phase sequences for selection without ATN and information unit ~~phases-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 [phases-protocol](#) shall be as shown in figure 54.

If a data transfer agreement is in effect that disables information unit [phases-protocol](#) (see 5.0.1.1 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.

4.431.1 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 [phases-protocol](#) disabled"

4.231.2 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 [phases-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 [phases-protocol](#) shall be as shown in figure 55.

If a data transfer agreement is in effect that enables information unit ~~phases-protocol~~ (see ~~5.0.1.4~~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 ~~information unit phases (INFORMATION UNIT OUT or INFORMATION UNIT IN)~~ data phases where information units are transferred. The final data phase ~~information unit phase~~ is followed by the BUS FREE phase.

4.331.3 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 ~~phases-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 ~~INFORMATION UNIT IN and INFORMATION UNIT OUT~~data phases"

4.431.4 Clause 12.1 SPI information unit sequences

The following modifications should be made to this clause:

The information unit ~~phase-protocol~~ transfers data in SPI information units. The order in which SPI information units are transferred within the information unit ~~phase-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 ~~INFORMATION UNIT IN or INFORMATION UNIT OUT~~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 ~~phase-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).

4.531.5 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."

4.631.6 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."

4.731.7 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 SEGST DATA IN-phases ~~and DATA OUT-phases~~. The IUTR message exchange establishes a similar agreement the only applies to the DECDT DATA phases (see section xxx). All other phases shall use asynchronous transfers.