



## Adaptec Technical Memorandum

NCITS T10-9-197r0

**To:** Improved SCSI Protocol Ad-Hoc Group

**Date:** June 4, 1997

**From:** Tak Asami (tasami@corp.adaptec.com)

**File:** 97-197r0.doc

**Subject:** CRC Attach proposal for LFP SCSI protocol

---

### Background:

Newer generation of SCSI is proposed to enable data transfers of 160MB/sec (or more) across 16-bit bus. Due to its high speed and low voltage signaling, such factors as signal attenuation, inter-symbolic interference and hot plugging are threatening the overall data integrity of this bus than ever before expected.

While it is not practical to prevent all of the problems from ever happening, it is vital to add a mechanism in the protocol to let the system detect the error situation to maintain the data integrity.

### Proposal:

After the initiator and target negotiated to support "LF Protocol" (to be proposed and discussed separately),

1. During the data transfer phase, the data source agent shall compute the 32-bit CRC and attach it at the end of every Logical Block worth of transmitted data.
2. Consequently, if CRC Attach is enabled, a target device shall disconnect only at LB boundary.
3. The data receiver agent shall compute the CRC for the incoming data, and respond to the data source agent in case the error was detected as soon as possible, and prior to the disconnection in any case.
4. If the receiving agent did not report the error prior to the disconnection, the data transmission up to that point is considered a success.
5. CRC errors are handled as follows:
  - 4a) If a CRC error is detected by the initiator during a Data In phase, then the initiator shall transmit the INITIATOR DETECTED ERROR message to the target. No new message needs to be defined.
  - 4b) If a CRC error is detected by the target during a Data Out phase, the target may retry the transmission immediately or disconnect without Save Data Pointer and retry at a later time.

### Justification:

#### *Main motivation*

The practice of attaching an error detecting code at the end of data transmission was proposed before in the process of developing SCSI-3 Parallel Interface document. It was rejected because there was no mechanism or justifiable reasons for a 10MHz system to have intermittent errors from which the data need to be protected to maintain its integrity. The system either always worked, or almost always failed.

During the past year of activity within SPI-2 Workgroup, it has been shown that:

- (1) Significant (over -10dB) signal level attenuation is expected at the 80MHz and above signals.
- (2) For the non-DC balanced mode of transmission such as NRZ that is traditionally used for parallel SCSI, a significant level of Intersymbolic Interference (ISI) is expected that can cause data pattern sensitive errors.
- (3) Class-4 hot plugging, while is judged to be workable under conventional Single Ended and High Voltage Differential bus, is yet to be proven on the Low Voltage Differential bus; it does have reasons for the concern.

Because LFP SCSI is expected to fill in the need of high-end storage systems where the system integrity is at least as important as the throughput performance, I feel that, at this point, we do have a strong reason to re-submit the mechanism to let the system detect the transmission error and let it deal with it.

**For Data Transfer Phases only**

This proposal covers only the data transfer phases (DATA IN, DATA OUT), because the system is most susceptible to errors during these phases due to their a) higher rate of transfer, in synchronous transfer mode, and b) longer length than any other transfer phases.

This proposal does not cover other information transfer phases such as MESSAGE IN/OUT, COMMAND OUT, STATUS IN phases, because I felt they are much less susceptible to the failure modes listed above. Reasons:

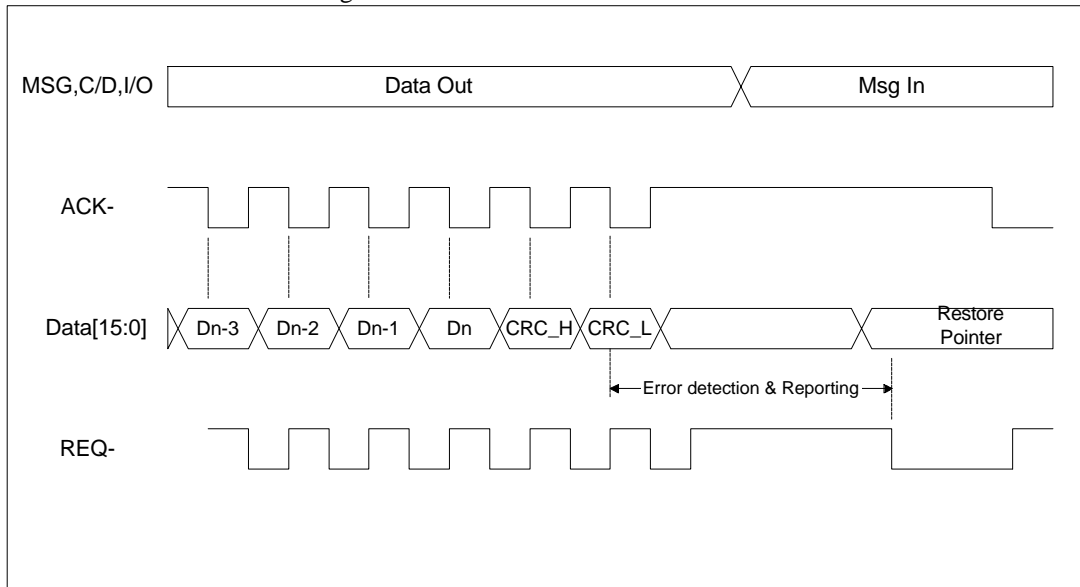
- a) they are generally a lot shorter in length (up to 12 bytes for command phase)
- b) the transfer takes place in asynchronous transfer mode, which adjusts its transfer timings to a given physical configuration automatically.

Under these assumptions, the parity assigned to each byte should be an adequate protection already.

**Error Handling**

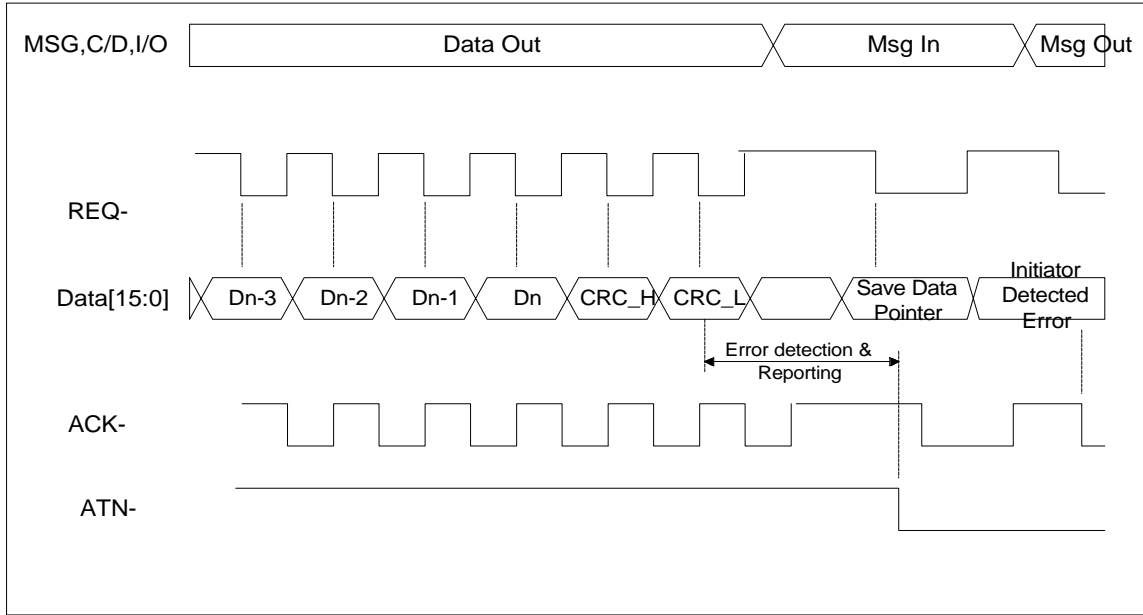
The CRC is transmitted at the end of every logical block (whose length is know prior to starting of the transfer) as well as at the end of nexus prior to disconnection. These check bytes are not included in the Transfer Count field of the CDB. The data receiving device checks for the data validity at each instance .

- If no error is detected, the target shall either (a) resume the current transfer, or (b) issue SDP-Disconnect messages and goes bus free.
- If it is the Target that detected an error (i.e., that was a WRITE command), then the target shall either:
  - a) Send a RESTORE POINTERS message to the initiator, then resend the same block of data
  - b) Send a DISCONNECT message to the initiator without saving Data Pointer; next reconnect will resume the data transfer at the original Data Pointer.



**Figure 1 Target Error Reporting at the end of Nexus**

- If it is the Initiator that detected an error (i.e., that was a READ command), then the initiator shall:
  1. Assert the ATN at the earliest opportunity, before acknowledging SDP or DISCONNECT message in any case.
  2. If the target role agent continues to transmit data, it shall continue transmitting ACK until the target goes to Message Out phase.
  3. Instead of going bus free, the target shall move into Message Out phase.
  4. Transmit INITIATOR DETECTED ERROR message.
  5. At this point, the target may take either action a) or b) listed above for the target detected error case.



**Figure 2 Initiator Error Reporting at the end of Nexus**

This requires that the data receiving device must complete the CRC evaluation before the target asserts REQ- to validate the Message In phase after completing the data transfer phase, before it starts affecting the throughput. An initiator may hold off asserting ACK- until it completes the task, and a target may hold off entering Message In phase. So this process time requirement is not for the system integrity; rather it is for the implementation guideline to prevent any performance degradation when no error is detected.

- For an initiator, this means it has one bus settle delay time (400nsec; going from Data In to Message In) to be ready to assert ATN.
- For a target, this means it has data release delay plus bus settle delay (800nsec, going from Data Out to Message In) to take actions listed above.

## Code Definition:

I propose to attach the 32-bit CRC at the end of every Logical Block. I recommend 32-bit CRC because the data block length is expected to be relatively long (typically 4KB), a good coverage for error detection is needed. A simple LRC (Longitudinal Redundancy Code) fails in that regard.

The particular code selected is the same code specified in Fibre Channel and Serial Storage Array and FDDI documents. I shall quote the definition here from Fiber Distributed Data Interface (FDDI) Media Access Control (MAC) document (ISO/IEC 9314-2:1989):

### 4.3.6.1 Definitions

$F(x)$  - A degree  $k-1$  polynomial which is used to represent the  $k$  bits of the frame covered by the FCS sequence. For the purposes of the FCS, the coefficient of the highest order term shall be the first bit transmitted.

$L(x)$  - A degree 31 polynomial with all of the coefficients equal to one, i.e.,  

$$L(x) = X^{31} + X^{30} + X^{29} + \dots + X^2 + X + 1$$

$G(x)$  - The standard generator polynomial  

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

$R(x)$  - The remainder polynomial which is of degree less than 32

$P(x)$  - The remainder polynomial on the receive checking side which is of degree less than 32

$Q(x)$  - The greatest multiple of  $G(x)$  in  
 $[X^{32}F(x) + X^kL(x)]$

$Q^*(x) - X^{32}Q(x)$

$M(x)$  - The sequence which is transmitted

$M^*(x)$  - The sequence which is received

$C(x)$  - A unique polynomial remainder produced by the receiver upon reception of an error free sequence. This polynomial has the value

$$C(x) = X^{32}L(x)/G(x)$$

$$C(x) = X31 + X30 + X26 + X25 + X24 + X18 + X15 + X14 + X12 + X11 + X10 + X8 + X6 + X5 + X4 + X3 + X + 1$$

#### 4.3.6.2 FCS Generation equations

The equations which are used to generate the FCS sequence from  $F(x)$  are as follows:

$$FCS = L(x) + R(x) = R\$(x) \quad (1)$$

where  $R\$(x)$  is the one's complement of  $R(x)$

$$[X^{32}F(x) + X^kL(x)] / G(x) = Q(x) + R(x)/G(x) \quad (2)$$

$$M(x) = X^{32}F(x) + FCS \quad (3)$$

NOTE - All arithmetic is modulo 2.

In equation (1), note that adding  $L(x)$  (all ones) to  $R(x)$  simply produces the one's complement of  $R(x)$ ; this equation is specifying that the  $R(x)$  is inverted before it is sent out.

Equation (3) simply specifies that the FCS is appended to the end of  $F(x)$ .

#### 4.3.6.3 FCS checking

The received sequence  $M^*(x)$  may differ from the transmitted sequence  $M(x)$  if there are transmission errors. The process of checking the sequence for validity involves dividing the received sequence by  $G(x)$  and testing the remainder. Direct division, however, does not yield a unique remainder because of the possibility of leading zeros. Thus a term  $L(x)$  is prepended to  $M^*(x)$  before it is divided. Mathematically, the received checking is shown in equation (4).

$$X^{32} [M^*(x) + X^kL(x)] / G(x) = Q^*(x) + P(x)/G(x) \quad (4)$$

In the absence of errors, the unique remainder is the remainder of the division

$$P(x)/G(x) = X^{32}L(x)/G(x) = C(x) \quad (5)$$

## Document Changes:

In order to incorporate the proposed change to the existing SCSI-3 documents, the physical interface document and SCSI Interlocked Protocol documents need to be modified. Ultimately, it also needs to be included in the unified SCSI document (SPUD), so my recommendation is to mark this proposal as "to be included in SPUD" and avoid working on its inclusion twice.