

Common CRC for DT and Packetized SCSI

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: January 25, 1999

0 Overview

Rev 0

This proposal is based on SPI-3, revision 2.

1 Introduction

I have received some feedback that it would be good for SPI-3 to use one common CRC protocol. Currently we have two: one for DT data phases using the current protocol and one for DT phases using the packetized protocol. The difference is not so much in the CRC itself (the polynomial is the same), but rather in the way in which it is transferred.

Thumbnail Descriptions

In the current protocol, the P0 line is used to separate data from CRC and pad. This allows for the sending of an arbitrary amount of data. It also allows the determination as to how much data to transfer to be made just before the target requests the CRC. It also allows for a simple implementation (e.g. just combinatorial logic using the REQ/ACK and P0 signals) to detect which bytes transferred or to be transferred are data, and which are not.

The packetized protocol currently uses an embedded CRC. This allows it to avoid the use of the P0 signal. It also allows for the sending of an arbitrary amount of data. But this amount must be determined ahead of time (it is indicated in the packet header). It also requires parsing the packet information, finding the appropriate length field, and then using a counter to detect the termination of data and the start of CRC.

Primary intent of this proposal

The primary thrust of this proposal is that these two approaches are one too many. This has been the feedback from many implementers (down to the ASIC level). I would appreciate feedback and discussion from the T10 committee.

Of the two approaches, the one defined for DT transfers with the existing protocol will work with packetized. The one designed for packetized will not work for the current protocol. Indeed, the history of the evolution of these protocols (the packetized one being developed first and only later the current protocol one) implies that both approaches were justified when they were presented. But now that a single approach can potentially satisfy both the current and packetized protocols, it makes some sense to see if they should be joined.

The only loss I can see is the use of the P0 signal in packetized. However, given our lack of proposals to use the P1 signal, this would not appear to be a big problem.

Secondary intent of this proposal

While the issue of duplicate implementations was the primary mover for this proposal, there are secondary ones. First the packetized CRC requires the target to send a packet to the initiator on WRITES after every packet from the initiator to the target. This is to insure that there is adequate buffer space in the target to allow it to receive the packet. In addition to the overhead of this packet, there is the overhead of turning around the direction of the SCSI bus signaling after each packet.

During data transfers the drive must balance this overhead (which can be minimized by the use of few long packets) with the ability to start data transfer ASAP (which can be done by using smaller packets). In general this makes for an awkward design when optimizing for low overhead during data transfer.

Secondly, the ability to detect the difference between user data and CRC at high speeds is very important. With the current protocol approach, this is a simple combinatorial circuit using the REA/ACQ and P0 signals as inputs. For the packetized approach, each packet must be parsed, the packet length (and/or the length of the next packet) identified, the appropriate registers loaded, and then countdown begun before the end of data transfer and the transfer of CRC. In general the former is easier to implement than the later.

Impact of the proposal

This proposal would have a significant impact on those planning on doing packetized but not planning on doing the DT phase with CRC for the current protocol. Note that by leveraging off the current protocol CRC mechanism, any software version of packetized may actually be easier (since it can use hardware CRC rather than software CRC).

For those implementing both protocols, this should prove a welcome reduction in complexity of design, testing, and verification.

2 Actual changes proposed

It turns out that the draft is written so well that the changes needed to accomplish this appear to be minor:

Section 8.1

In the P_CRCA paragraph, delete: "If information unit transfers are enabled the P_CRCA signal shall be negated by the target during any part of the DT DATA phase (i.e., all pad and CRC information is contained within the information units as defined in clause 12)."

Section 11.1.11

Restore the proposed deleted wording: "b) the P_CRCA, P1 (if any), P2 (if any), and P3 (if any) signal(s) shall be used to detect parity errors, and"

And delete: "c) the information units embedded CRC shall be used to detect CRC information data errors."

In section 12.2:

In **Table 35 - SPI command information unit**, remove bytes 20-23 from the table (CRC)

Delete "The crc field shall use the algorithm defined in clause 8.2.2."

In section 12.3:

In **Table 38 - SPI L_Q information unit**, remove bytes 16-19 from the table (CRC)

Delete "The crc field shall use the algorithm defined in clause 8.2.2."

In section 12.4

In **Table 40 - SPI data information unit**, remove bytes N=1 to N+4 from the table (CRC)

Delete "The crc field shall use the algorithm defined in clause 8.2.2."

In section 12.5

In **Table 41 - SPI status information unit**, remove bytes N=1 to N+4 from the table (CRC)

Delete "The crc field shall use the algorithm defined in clause 8.2.2."