To: **T-10/SPI-4 Working Group**
From: **Richard Moore**
QLogic Corp.
Date: **August 18, 2000**

Subject: SPI-4 Clarifications

1. QAS Clarifications

Section 10.5.4: The sequence of events for a target to indicate it wants to release the bus needs some rewording. Item (2) says, "After the initiator negates the ACK signal for the QAS REQUEST message and if the initiator did not create an attention condition then the initiator shall release all SCSI signals within two system deskew delays after detecting MSG, C/D, and I/O signals false." Note that there are two "after" clauses here. The second "after" happens after item (3), which says "After detection of the last ACK signal being true and if there is no attention condition, the target shall negate REQ. After detection of the last ACK signal being false, the target shall release all SCSI signals except the BSY, MSG, C/D, I/O, and REQ signals. Then the target shall negate the MSG, C/D, and I/O signals within two system deskew delays. The target shall wait two system deskew delays after negating the C/D, I/O, and MSG signals before releasing the REQ signal."

To clean up this wording, replace items 2-3 with the following:

> 2) After detection of the REQ signal being true for the QAS REQUEST message, the initiator shall assert ACK.
> 3) After detection of the last ACK signal being true the target shall negate REQ.
> 4) After detection of the last REQ signal being false, the initiator shall negate ACK.
> 5) After detection of the last ACK signal being false and if the initiator did not create an attention condition, the target shall release all SCSI signals except the BSY, MSG, C/D, I/O, and REQ signals. Then the target shall negate the MSG, C/D, and I/O signals within two system deskew delays. The target shall wait two system deskew delays after negating the C/D, I/O, and MSG signals before releasing the REQ signal.
> 6) If the initiator did not create an attention condition then the initiator shall release all SCSI signals within two system deskew delays after detecting MSG, C/D, and I/O signals false.

Then, relabel items 4-6 to 7-9.

2. CRC Byte Swapping

The diagram in Figure 57, and the accompanying text in 11.3.4, are confusing. The figure shows the bytes being re-ordered before the bits are flipped but the text only mentions the bit flipping. Swapping two bytes and then reversing the bits in each byte (as shown in the figure) is equivalent to simply reversing the bits across all 16 bits:

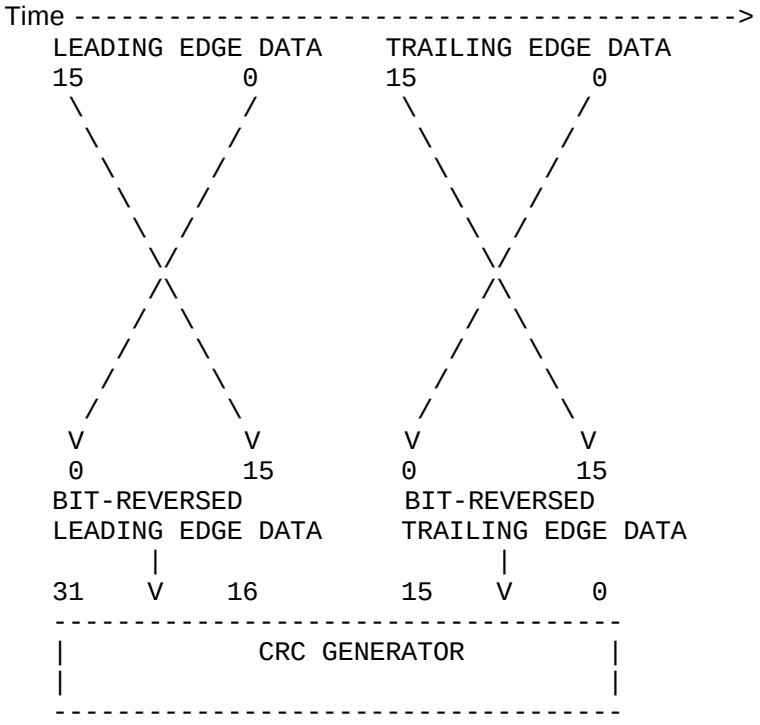
> Swapping two bytes B1[7:0], B0[7:0] gives B0[7:0], B1[7:0]
> Reversing bits on each byte then gives B0[0:7], B1[0:7] which is the exact reversal of the original 16 bits.

The two paragraphs in 11.2.3, beginning with "Bytes are bit reversed...", could be rewritten for greater clarity as follows:

> "The data transferred on the leading edge of a REQ or an ACK is bit-reversed and becomes the most significant 16 bits of the CRC generator's input. The data transferred on the trailing edge of a REQ or an ACK is bit-reversed and becomes the least significant 16 bits of the CRC generator's input.

> "The most significant 16 bits of the CRC generator's output are bit reversed and the one's complement of this result forms the portion of the CRC field that is transferred on the leading edge of the CRC REQ or ACK. The least significant 16 bits of the CRC generator's output are bit reversed and the one's complement of this result forms the portion of the CRC field that is transferred on the trailing edge of the CRC REQ or ACK."

The figure itself is overly complicated. The figure could be redrawn along the following lines to make it easier to understand:



with similar changes at the output of the CRC generator. The present complex form of this diagram seems to be a result of an earlier version of the CRC proposal which allowed CRC for narrow as well as wide transfers; since CRC is only allowed for wide transfers the narrow representation is not necessary.