To: T10 Technical Committee
From: Luben Tuikov, Vitesse Semiconductor (ltuikov@vitesse.com)
Date: 08 June 2007
Subject: 07-130r4 SAS-2 CRC fixes

Revision history
Revision 0 (12 March 2007) First revision
Revision 1 (19 March 2007) Second revision
1) Table 112, row defining R'(x):
a) Define R'(x) explicitly.
b) Use M(x) instead of M'(x), as the latter is a designation of the former, which is what is received.
Revision 2 (20 March 2007) Third revision
Address editor's comments.
Revision 3 (23 May 2007) Fourth revision
Address editor's comments.
Revision 4 (08 June 2007) Fifth revision
Adress editor's comments.

Related documents
The iSCSI CRC32C Digest and the Simultaneous Multiply and Divide Algorithm - Tuikov&Cavanna, Jan 30, 2002, although using a different generator polynomial, the paper describes the algebra in great detail.

Overview

Fix the math.

Convert to Frame Maker's equations format.

Suggested changes

7.5 CRC

7.5.1 CRC overview

All frames include cyclic redundancy check (CRC) values to help detect transmission errors.

Frames transmitted in an STP connection shall include a CRC as defined by SATA (see ATA/ATAPI-7 V3). Address frames, SSP frames, and SMP frames shall include a CRC as defined by this standard.

Annex D contains information on CRC generation/checker implementation.

Arithmetic is modulo 2.

Table 112 defines notation used in the following text describing the CRC calculation.

Table 112 — CRC notation and definitions

Notation	Definition
$T(i)$	A transformation over the non-negative integers, $T(i) = i + 7 - 2(i \bmod 8), i \geq 0, i \in Z^{+}$.
$F(x)$	A polynomial representing the frame not including the CRC field. If the number of bits in the frame is k and the frame bits described by b_i, where the bit index i, only denotes that bit b_i is more significant than bit b_{i+1}, then, $F(x) = b_0x^{k-1} + b_1x^{k-2} + \ldots + b_{k-2}x + b_{k-1}$. For example, if the frame contains one dword set to 516F_3019h, then, $F(x) = x^{30} + x^{28} + x^{24} + x^{22} + x^{21} + x^{19} + x^{18} + x^{17} + x^{16} + x^{13} + x^{12} + x^4 + x^3 + 1$.
$F_t(x)$	A polynomial representing the frame not including the CRC field, but bit positions of each byte are transposed. That is bit index 7 is now bit index 0, bit index 6 is now bit index 1, etc. Defined as, $F_t(x) = b_{T(0)}x^{k-1} + b_{T(1)}x^{k-2} + \ldots + b_{T(k-2)}x + b_{T(k-1)}$. For example, if the frame contains one dword set to 516F_3019h, then, $F_t(x) = x^{31} + x^{27} + x^{25} + x^{23} + x^{22} + x^{21} + x^{20} + x^{18} + x^{17} + x^{11} + x^{10} + x^7 + x^4 + x^3$.
$L(x)$	The identity polynomial of degree 31. A polynomial with all of the coefficients set to one, $L(x) = x^{31} + x^{30} + \ldots + x + 1$. (i.e., $L(x) = FFFF\_FFFFh$).
$G(x)$	The CRC generator polynomial, the divisor polynomial, $G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$. (i.e., $G(x) = 1\_04C1\_1DB7h$).
$R(x)$	A remainder polynomial, always of degree less than 32.
$R_t(x)$	A remainder polynomial, but bit positions of each byte are transposed. That is, the bit index to each term of this polynomial is T(i), instead of i.
$Q(x)$	A quotient polynomial resulting from the CRC generation calculation. This value is discarded.
$Q'(x)$	A quotient polynomial resulting from the CRC generation calculation. This value is discarded..
$M(x)$	A polynomial of degree k+31 representing the frame including the CRC field, as presented to the 8b10b encoder for transmission.
$M'(x)$	A polynomial representing the received frame including the CRC field, after it has been 8b10b decoded. If the received frame has no errors, $M'(x) = M(x)$ and is of degree k+31.
$M_t'(x)$	M'(x) but bit positions of each byte are transposed.
$R'(x)$	The result of finding the remainder of an error free reception of $M(x)$. The remainder of $\frac{x^{32}L(x)}{G(x)}$, a unique constant polynomial, $R'(x) = x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1$. (i.e., $R'(x) = C704\_DD7Bh$). Note that $R'(x)$ transposed and inverted is 1CDF_4421h, (i.e. $R_t'(x) + L(x) = 1CDF\_4421h$).

7.5.2 CRC generation

The CRC computation process of a frame is as follows:

$$x^kL(x) + x^{32}F_t(x) = Q(x)G(x) + R(x).$$

That is, the frame is transposed, then the first 32 bits of the transposed frame are inverted, then 32 bits of 0 are appended to the end of the transposed frame, then this result is divided by the generator polynomial to find the remainder, R(x).

The transmitter shall present to the 8b10b encoder:

$$M(x) = x^{32}F(x) + L(x) + R_t(x).$$

That is, the inverted transposed remainder is appended to the end of the frame, then this result is presented to the 8b10b encoder for transmission.

For the purposes of CRC computation, inverting the first 32 bits of a frame may be performed by one of the following exclusive options:

a) invert the first 32 bits of the frame, F(x), and seed the CRC remainder register with 0000_0000h;
b) seed the CRC remainder register with FFFF_FFFFh; or
c) prepend the constant 62F5_2692h to F(x) and seed the CRC remainder register with 0000_0000h.

Figure 151, shows the CRC process for an address frame, an SSP frame and SMP frame.

Figure 151 — Address frame, SSP frame, and SMP frame CRC bit ordering

Dwords in STP frames are little-endian and feed into the STP CRC generator without swapping bits within each byte and inverting the output like the SAS CRC generator. Figure 152 shows the STP CRC bit ordering.

Figure 152 — STP frame CRC bit ordering

Since STP is little-endian, the first byte of a dword is in bits 7:0 rather than 31:24 as in SSP and SMP. Thus, the first byte contains the least-significant bit. In SSP and SMP, the first byte contains the most-significant bit.

See 7.7 for details on how the CRC generator fits into the dword flow along with the scrambler.

7.5.3 CRC checking

The CRC field of a received frame is generated by the receiver in the same manner that it is generated by the transmitter.

That is, the received frame including the CRC field is transposed, then the first 32 bits are inverted, then 32 bits of 0 are appended to the end, then this result is divided by the generator polynomial to find the remainder.

A received frame which has not incurred any CRC detectable errors during transmission generates a remainder equal to R'(x).

If there were no transmission errors, then the received frame equals M(x):

$$\begin{aligned} M'(x) &= M(x) \\ &= x^{32}F(x) + L(x) + R_t(x). \end{aligned}$$

The CRC, R'(x), is derived as follows:

$$\begin{aligned} x^{(k+32)}L(x) + x^{32}M_t'(x) &= x^{(k+32)}L(x) + x^{32}(x^{32}F_t(x) + L(x) + R(x)) \\ &= x^{32}Q(x)G(x) + x^{32}L(x). \end{aligned}$$

That is, the received frame, M'(x), is transposed, then the first 32 bits of the received transposed frame are inverted, then 32 bits of 0 are appended to the end of the received transposed frame, then this result is divided by the generator polynomial to find the remainder.

But $G(x)$ divides $x^{32}L(x)$:

$$x^{32}L(x) = Q'(x)G(x) + R'(x).$$

In the above equation, L(x) and G(x) are known and constant, thus R'(x) is known and constant. It is this R'(x) which the receiver expects after calculating the CRC of a received frame.

From the previous two results:

$$x^{k+32}L(x) + x^{32}(x^{32}F_t(x) + L(x) + R(x)) = (x^{32}Q(x) + Q'(x))G(x) + R'(x).$$

R'(x) is then transposed and inverted, in the same manner as is done by the transmitter, to obtain:

$$R_t'(x) + L(x) = 1CDF\_4421h.$$

Alternatively to this process, the receiver may check the CRC validity of the frame by stripping off the last 32 bits, to be left with $F(x)$ and computing the CRC as presented in section 7.5.2. The frame has a valid CRC if the result, $L(x) + R_t(x)$ equals the last 32 bits of the frame which were stripped.

See 7.7 for details on where the CRC checker fits into the dword flow along with the descrambler.