To: T10 Technical Committee
From: Luben Tuikov, Vitesse Semiconductor (ltuikov@vitesse.com)
Date: 19 March 2007
Subject: 07-130r1 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.

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.

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

Table 112 — CRC polynomials

Polynomial	Definition
F(x)	A polynomial representing the frame contents which are covered by the CRC. For the purposes of the CRC, the coefficient of the highest order term shall be the first bit transmitted. Let the number of bits in the message be k and the message bits described by b_i, then, $F(x) = b_0x^{k-1} + b_1x^{k-2} + \ldots + b_{k-2}x + b_{k-1}.$
$F_t(x)$	A polynomial representing the frame contents, but bit positions of each byte are transposed. I.e. bit 7 is now bit 0, bit 6 is now bit 1, etc.
L(x)	The identity polinomial 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, i.e. 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 (see below).
$R_t(x)$	A remainder polynomial, but bit positions of each byte are transposed. See definition of $F_t(x)$.
Q(x)	A quotient polynomial (see below).
Q'(x)	A quotient polynomial (see below).
M(x)	A polynomial of degree 31+k representing the transmitted frame followed by the transmitted CRC.
M'(x)	A polynomial representing the received frame followed by the received CRC. It is of degree 31+k, and equal M(x), for an error free reception.
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.

7.5.2 CRC generation

Arithmetic is modulo 2. The CRC, R(x), of a frame is generated as follows,

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

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

Note that inverting the first 32 bits of the message (i.e. seeding the CRC remainder register with FFFF_FFFFh) is equivalent to prepending an untransposed message with the constant 62F5_2692h and seeding the CRC register with 0000_0000h.

The sender shall transmit,

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

such that the CRC can be computed and the message sent in one go.

That is, the inverted transposed remainder is appended at end of the frame, then this result transmitted. Bit order is described in the following text and the inversion of R(x) explicitly shown.

The bit order of F(x) presented to the CRC function is the same order as the bit transmission order (i.e., the bits within each byte encoded into a data dword are transposed to match the implicit transposition in the 8b10b encoding process). This order is shown in figure 151.

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

CRC of a received frame is generated by the receiver in the same manner that it is generated by the transmitter. A received frame which has not incurred any CRC detectable errors during transmission, should generate a remainder equal to R′(x).

Denote a received frame by M′(x). If there were no transmission errors the received frame would equal M(x) and of degree k+32,

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

The CRC, R′(x), is derived as follows,

$$x^{k+32}L(x) + x^{32}(x^{32}F_t(x) + L(x) + R(x)) = \ldots$$

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

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 should get after calculating the CRC of a received message. From the previous two results,

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

The bit order of F(x) presented to the CRC checking function is the same order as the CRC generation bit order (see figure 151). See 7.7 for details on where the CRC checker fits into the dword flow along with the descrambler.