To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 26 May 2004
Subject: 04-176r0 SBC-2 New CRC figure and example C code

Revision history

Revision 0 (26 May 2004) First revision

Related documents

spc3r16 - SCSI Primary Commands - 3 revision 16
03-381r0 SBC-2 Data protection CRC seed (Rob Elliott, HP)

Overview

The endianness of the CRC figure in the data protection section has caused some confusion. It needs to be more clear that byte 0 and byte 1 are fed into the 16-bit CRC generator with byte 0 positioned to provide the most significant bit (MSB) and byte 1 to provide the LSB. Then, byte 2 and 3 are fed in with byte 2 containing the MSB and byte 3 containing the LSB, etc.

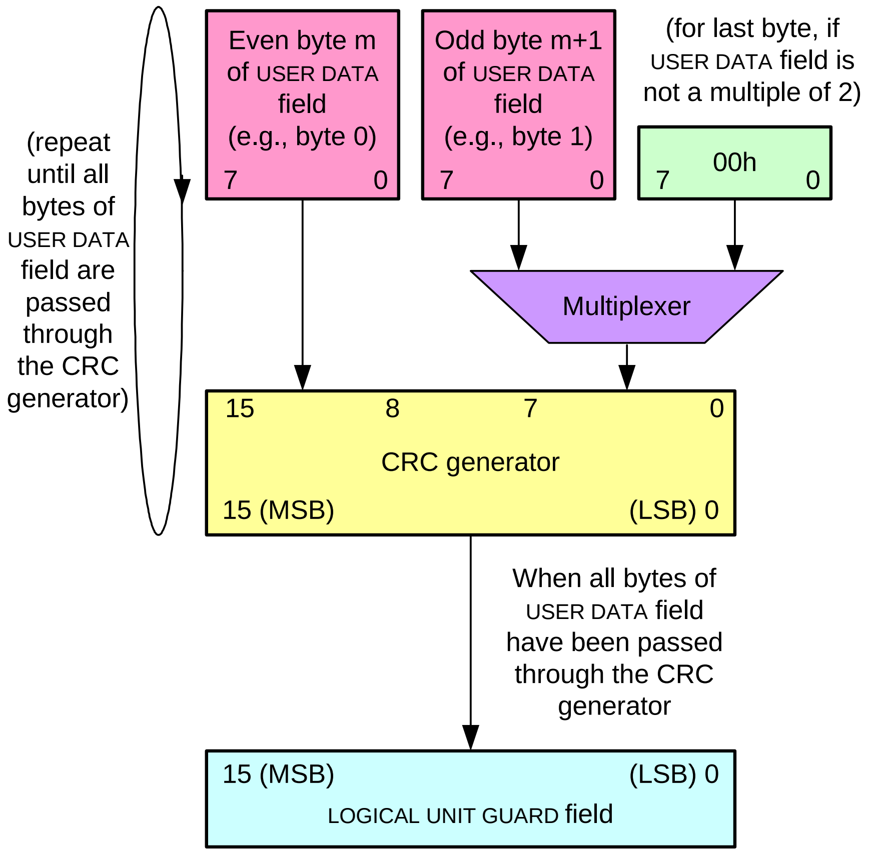
Also, the current text ignores how to handle uneven blocks of data. Although unlikely, uneven block sizes are not prohibited anywhere. If the last byte is not available, 00h should be fed into the CRC generator.

To help avoid confusion, sample C code is provided as well (as is done in SAS and SATA).

Suggested changes

~~The bit order of F(x) presented to the CRC function is two bytes at a time starting with byte zero bit seven of the USER DATA field until all the contents of the USER DATA field are processed. An example of an even byte transfer CRC generation is shown in figure 3.~~

The CRC generator processes the USER DATA field one word (i.e., two bytes) at a time, starting with byte 0. In the first word, byte 0 bit seven is the MSB and byte 1 bit 0 is the LSB. Figure 3 shows an example CRC generator.


(repeat until all bytes of USER DATA field are passed through the CRC generator)
Even byte m of USER DATA field (e.g., byte 0)
7
0
Odd byte m+1 of USER DATA field (e.g., byte 1)
7
0
(for last byte, if USER DATA field is not a multiple of 2)
00h
7
0
Multiplexer
15
8
7
0
CRC generator
15 (MSB)
(LSB) 0
When all bytes of USER DATA field have been passed through the CRC generator
15 (MSB)
(LSB) 0
LOGICAL UNIT GUARD field

~~Figure 1 — Even byte CRC generator bit order~~CRC generator example [new figure too]

Add new annex

Annex C

(informative)

CRC example

The following is an example C program that generates the value for the LOGICAL BLOCK GUARD field in protection information (see 4.15).

```
// dpcrc.cpp : SCSI SBC-2 Data Protection CRC generator
#include "stdafx.h"
#include <stdio.h>
#include <malloc.h>

/* return crc value */
unsigned short calculate_crc(unsigned char *frame, unsigned long length) {
unsigned short const poly = 0x8BB7L;  /* Polynomial */
unsigned const int poly_length = 16;
unsigned short crc_gen;
unsigned short x;
unsigned int i, j, fb;
unsigned const int invert = 0;/* 1=seed with 1s and invert the CRC */

crc_gen = 0x0000;
crc_gen ^= invert? 0xFFFF: 0x0000;   /* seed generator */

for (i = 0; i < length; i += 2) {
      /* assume little endian */
      x = (frame[i] << 8) | frame[i+1];

      /* serial shift register implementation */
        for (j = 0; j < poly_length; j++) {
            fb = ((x & 0x8000L) == 0x8000L) ^ ((crc_gen & 0x8000L) ==
0x8000L);
            x <<= 1;
            crc_gen <<= 1;
            if (fb)
                  crc_gen ^= poly;
      }
}

return crc_gen ^ (invert? 0xFFFF: 0x0000); /* invert output */
} /* calculate_crc */

/* function prototype */
unsigned short calculate_crc(unsigned char *, unsigned long);

void main (void) {
unsigned char *buffer;
unsigned long buffer_size = 32;
unsigned short crc;
unsigned int i;

/* 32 0x00 */
buffer = (unsigned char *) malloc (buffer_size);
for (i = 0; i < buffer_size; i++) {
      buffer[i] = 0x00;
}
crc = calculate_crc(buffer, buffer_size);
printf ("Example CRC all-zeros is %04x\n", crc);
free (buffer);

/* 32 0xFF */
buffer = (unsigned char *) malloc (buffer_size);
for (i = 0; i < buffer_size; i++) {
      buffer[i] = 0xFF;
}
crc = calculate_crc(buffer, buffer_size);
printf ("Example CRC all-ones is %04x\n", crc);
free (buffer);

/* 0x00 incrementing to 0x1F */
buffer = (unsigned char *) malloc (buffer_size);
for (i = 0; i < buffer_size; i++) {
      buffer[i] = i;
}
crc = calculate_crc(buffer, buffer_size);
printf ("Example CRC incrementing is %04x\n", crc);
free (buffer);

/* 0xFF 0xFF then 30 zeros */
```

```
buffer = (unsigned char *) malloc (buffer_size);
buffer[0] = 0xff;
buffer[1] = 0xff;
for (i = 2; i < buffer_size; i++) {
      buffer[i] = 0x00;
}
crc = calculate_crc(buffer, buffer_size);
printf ("Example CRC FF FF then 30 zeros is %04x\n", crc);
free (buffer);

/* 0xFF decrementing to 0xE0 */
buffer = (unsigned char *) malloc (buffer_size);
for (i = 0; i < buffer_size; i++) {
      buffer[i] = 0xff - i;
}
crc = calculate_crc(buffer, buffer_size);
printf ("Example CRC FF decrementing to E0 is %04x\n", crc);
free (buffer);

} /* main */
```