Cyclic Redundancy Check Codes Used in the SNS Real-Time Data Link

E. Bjorklund

Abstract

The Cyclic Redundancy Check (CRC) is an efficient and powerful tool for detecting errors in transmitted messages. The SNS Real-Time Data Link (RTDL) system employs two levels of CRC checking. Each RTDL frame is transmitted with an 8-bit CRC generated in hardware by the V105S RTDL Encoder module. In addition, the entire RTDL message contains a 24-bit software-generated CRC which is sent as the last RTDL frame (frame number 255) in the message. This tech-note discusses the details of the RTDL CRC implementation, with particular emphasis on the 24-bit CRC RTDL frame, since its implementation is a recent addition.

Introduction: Some CRC Basics

The basic concept behind CRCs is to treat the message as one giant binary number and divide it by a fixed value using modulo-2 polynomial arithmetic. The remainder of that division becomes the CRC "check-sum". The reasons for using modulo-2 polynomial arithmetic are:

- The long-division problem is reduced to a simple series of shift and XOR operations, making it efficient to implement in hardware.
- The mathematical properties of modulo-2 polynomial arithmetic can be used to determine and prove certain desirable characteristics of the polynomial chosen to be the divisor (also referred to as the "generator polynomial").

One useful property of modulo-2 polynomial arithmetic is that the string obtained by appending the CRC to the end of the message is guaranteed to be an exact multiple of the generator polynomial. Consequently, if you feed the entire message plus the CRC to a CRC-checker circuit, the final result in the shift register (the remainder) should always be zero if there were no errors.

Two other bits of "CRC-jargon" need to be briefly touched on before we proceed.

Initial Value: This is the value that you start with (or preset into the shift register) before beginning the CRC calculation. For a straight, unadulterated, division problem, this value would be zero. However, many CRC algorithms start with a non-zero initial value (which is equivalent to pre-pending a non-zero value to the beginning of the message) in order to catch errors such as when the transmitter is only outputting zeroes. Since anything divided into zero will produce a zero remainder, there is no generator polynomial that can detect a "string of zeroes" error as long as the initial value is zero.

Reflection: This term refers to the practice of reversing (or reflecting) the bit-order of each byte during the checksum calculation. This is frequently done to compensate for UARTs, which transmit bytes in reverse (least-significant-bit-first) order. Since the RTDL hardware transmits everything most-significant-bit-first, reflection is not used in the SNS RTDL CRCs.

The 8-Bit Frame CRC

Each RTDL frame consists of an 8-bit frame number, 24 bits of data, and an 8-bit CRC (plus some start and stop bits that don't figure into the CRC calculation). The CRC is hardware generated in the V105S RTDL Encoder module.

The generator polynomial is: $x^8 + x^7 + x^5 + x^4 + x + 1$, with an initial value of 0.

Using standard techniques from the literature [1,2,3] and a little "brute force," we can show that this polynomial has the following error-detecting characteristics:

- Detects all single-bit errors.
- Detects all two-bit errors.
- Detects any error involving an odd number of bits.
- Detects any consecutive run of errors of length 8 or less.
- The probability of an undetected error is approximately 3.91×10^{-3} (`1/256`).

Note that since the initial value is zero, "empty frame" errors (errors in which the entire frame is zero) will not be detected. However, since zero is not a valid RTDL frame number, there is another mechanism for detecting these errors.

The 24-bit Message CRC

An "RTDL Message" is a sequence of RTDL frames. A new RTDL message is sent out prior to the beginning of each SNS machine cycle. The last frame of an RTDL message is frame number 255 and contains a 24-bit "Message-CRC". This CRC is generated by software running in the timing master IOC.

24 bits was chosen because it was the largest CRC that would fit in an RTDL frame. The 24-bit generator polynomial we use in the RTDL system comes from "open PGP" [6].

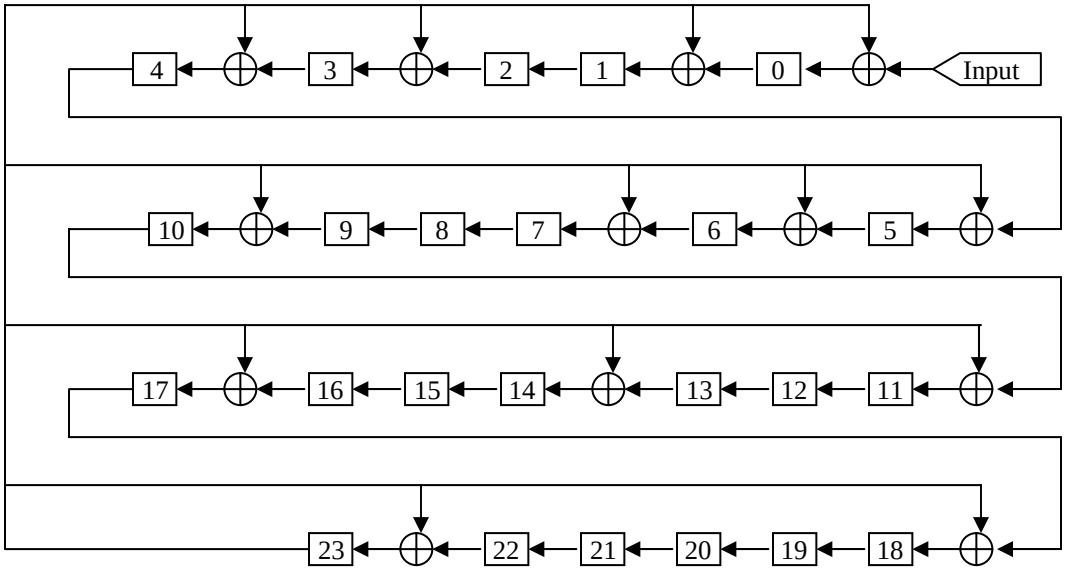
The generator polynomial is:
$x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$
with an initial value of 0xffffff.

It has the following properties:

- Detects all single-bit errors.
- Detects all two-bit errors.
- Detects any error involving an odd number of bits.
- Detects any consecutive run of errors of length 23 or less.
- Detects "run of zeroes" errors.
- The probability of an undetected error is approximately 5.96×10^{-8} (`1/16777215`).

Only the frame number and the 24-bit data portion of each RTDL frame are included in the message CRC. The hardware CRC and the start/stop bits are left out.

Efficient, table-driven, algorithms for computing a CRC in software are given in [4] and [5]. The following circuit can be used at the hardware-level to check the message CRC:



Notes:

⊕ = Exclusive OR

Initial value of shift register is all 1's

It should be noted that the circuit given above is a slightly different form than the per-frame CRC circuit implemented in the V105S module. The two circuit formats are functionally identical as long as the initial condition is 0. If you wish to implement the message CRC using the V105S-style circuit, you will need to set the initial condition to the hex value "edf8ce" (the result of shifting twenty four 1's through the circuit).

References

[1] A.S.Tanenbaum, “Computer Networks”, Prentice Hall, 1981.
[2] W.W.Peterson, E.J.Weldon, “Error Correcting Codes”, MIT Press, 1972,
[3] D.T.Tang, R.T.Chien, “Coding for Error Control”, IBM Systems Journal, Vol. 8 No. 1, 1969.
[4] D.V.Sarwate, “Computation of Cyclic Redundancy Checks via Table Look-Up”, Communications of the ACM, Vol. 31 No. 8, 1988.
[5] R.N.Williams, “A Painless Guide to CRC Error Detection Algorithms” http://www.ross.net/crc/crcpaper.html
[6] RFC2440 (Open PGP). http://www.gnupg.org/rfc2440.html