CRC (cyclic-redundancy checking) allows you to verify the integrity of transmitted data frames. It finds use in many transmission protocols, both bit-oriented (High Level Data Link Control, Advanced Data Communications Control Procedure, CRC-CCITT, for example) and frame-oriented (asynchronous transfer mode, Ethernet, Fiber Distributed Data Interface, for example). CRC protection constitutes adding redundancy bits (n–k bits) to a frame (k bits) of data to transmit a coded frame of n bits. Express the data as a polynomial data frame of k bits:

\[ X_k = \{ x_{k-1}, x_{k-2}, \ldots, x_0 \} \], \hspace{1em} \text{(Equation 1)}

which is written as:

\[ X_k(D) = x_{k-1}D^{k-1} + x_{k-2}D^{k-2} + \ldots + x_1D + x_0 \], \hspace{1em} \text{(Equation 2)}

where \( D \) is the variable and bits \( x_i \) (1,0) are the coefficients of the polynomial. Starting from \( X_k(D) \) to generate a code word of n bits (n>k), you use a so-called generator polynomial:

\[ G_{n-k}(D) = g_{n-k-1}D^{n-k-1} + g_{n-k-2}D^{n-k-2} + \ldots + g_1D + g_0 \], \hspace{1em} \text{(Equation 3)}

Multiplying \( x_k(D) \) by \( D^{n-k} \), you obtain a polynomial of grade n. This operation shifts the data \( X_k(D) \) to the left by n-k bits, attaching n-k zeros to the least significant bits. Now, consider:

\[ D^{n-k}X_k(D) = Q_k(D)G_{n-k}(D) + R_{n-k-1}(D) \], \hspace{1em} \text{(Equation 4)}

where the original data multiplied by \( D^{n-k} \) is divided by the generator polynomial, yielding the quotient \( Q_k(D) \) and the remainder \( R_{n-k-1}(D) \), which always has the grade n-k-1 (n-k bits). The transmitted frame consists of the original data followed by n-k bits of remainder. The polynomial arithmetic, in this case, is cast in a binary algebraic field. Addition and subtraction are identical and equivalent to an XOR operation without the need of a carry. Thus, it is possible to rewrite Equation 4 as follows:

\[ D^{n-k}X_k(D) + R_{n-k-1}(D) = Q_k(D)G_{n-k}(D) \], \hspace{1em} \text{(Equation 5)}

Equation 5 shows that the transmitted frame is divisible by the generator polynomial. At the receiving side it is divided by the same generator polynomial, and the quotient, \( Q_k(D) \) is discarded. If
no errors exist in the transmitted data, the remainder should always be zero. Both the transmitter and the receiver side must perform the same division by $G_{n-k}(D)$. The linear-feedback shift registers in Figure 1 implement the division. When the $k$th bit loads in, the shift-register value is the remainder. For more information about how the CRC scheme works, see Reference 1. An AND function performs multiplication by one or zero, and an XOR function performs the sum (subtraction) operation. Listing 1 shows simple synthesizable VHDL code that generates the divider schematic.

The code includes constants that allow division by the generator polynomial. Note that Listing 1 includes common polynomials from Listing 2, used as examples (see commented Constants) and also includes the relative initial value of the CRC shift register. Some standards dictate reversing the incoming bit stream (LSB first), the calculated CRC, or both. The signal load_frame, active high, defines the dimension of the serially loaded data frame. For example, the simulation in Figure 2 has a data-frame size of 20 bits. At the 20th bit, load_frame goes low, reloading the initial value for the CRC shift registers in readiness for the next frame. The calculated CRC value is CC16. (DI #2553)

REFERENCE

1. Ross, N, "A painless guide to CRC error detection algorithms,"