Polynomial Arithmetic and Cyclic Redundancy Checks

The calculation of cyclic redundancy checks (CRCs) depends upon the arithmetic of modulo 2 polynomials. A modulo 2 polynomial is an expression of the form $a_0 + a_1x + a_2x^2 + ...+ a_nx^n$, where the coefficients a_0, a_1, ... ,a_n are integers modulo 2, that is, they can take values of 0 or 1. The coefficients obey regular modulo 2 arithmetic as follows:

Addition

+	0	1
0	0	1
1	1	0

Multiplication

×	0	1
0	0	0
1	0	1

Electrical engineers will recognize the addition operation as the XOR operation defined on binary states.

The following are examples:

- $(1+x^2+x^4) + (1+x+x^4) = x+x^2$
- $(1+x^2+x^4)\ (1+x+x^4) = 1+x+x^2+x^3+x^5+x^6+x^8$.

Division of modulo 2 polynomials is done in exactly the same way as it is for ordinary polynomials, remembering that the coefficients obey modulo 2 arithmetic as defined above.

A very important operation in the calculation of CRCs is the calculation of the remainder R(x) when one polynomial M(x) is divided by another G(x). R(x) is uniquely defined by M(x) = G(x)K(x) + R(x), where the degree of R(x) is less than the degree of G(x). The degree of a polynomial is defined as the largest value of n for which the coefficient a_n is nonzero, so, for example, $1+x^2+x^5$ has degree 5.

Although this sort of polynomial division may look formidable, there are very efficient means of calculation based on shift registers.

CRCs are calculated on a stream of data by assuming that the data represents the coefficients of some modulo 2 polynomial. So, given a stream of data n bits long, the first bit can be considered as the coefficient a_{n-1} of x^{n-1}, the second bit as the coefficient a_{n-2} of x^{n-2}, the $(n-1)$th bit as the coefficient a_1 of x and the nth bit as the constant term a_0.

Roughly speaking, the 32 bits of the CRC are defined to be the polynomial remainder R(x) when the polynomial defined by the data M(x) is divided by a standard polynomial:

$$G(x)= 1+x+x^2+x^4+x^5+x^7+x^8+x^{10}+x^{11}+x^{12}+x^{16}+x^{22}+x^{23}+x^{26}+x^{32}.$$

Slight modifications are made for implementation reasons, but as far as the error properties are concerned, this is what is calculated.

The CRC bits are appended onto the data. When the data is received, the CRC is calculated and compared with the received CRC. If there is a difference, the data is known to have been corrupted.

When a corruption occurs in transmission, a number of bits are inverted. Let the errored bits define the coefficients of a polynomial E(x); for example, if errors occur at bit positions p and q, the polynomial will be $x^{n-p} + x^{n-q}$. Since the operation of calculating CRCs is linear, the error is detected if and only if E(x) is not exactly divisible by G(x). CRC-generating polynomials such as G(x) are chosen precisely to detect as many polynomials like E(x) as possible.