TCOM 370

NOTES 99-9B

Further Notes on Cyclic Codes

Cyclic Shift of Length N Binary Words as Multiplication "mod(X^N+1)"

Consider the polynomial representation T(X) for a length-N codeword.

$$T(X) = c_{N-1}X^{N-1} + c_{N-2}X^{N-2} + \ldots + c_1X + c_0$$

- Now XT(X) is a degree-N polynomial in general (is monic if $c_{N-1} = 1$).

  (monic polynomial of degree N → coefficient of X^N is 1)

- Dividing XT(X) by (X^N+1) will produce the result c_{N-1} with a ***remainder term.*** This remainder term will be the polynomial corresponding to the ***one-unit cyclic left shift*** of the original codeword!

Thus, a cyclic left-shift of a codeword produces codeword corresponding to

remainder$\left\{\dfrac{XT(X)}{X^N+1}\right\}$ which is usually written as **XT(X) "mod(X^N+1)"**

Generator Polynomial

- Let G(X) be a monic polynomial of degree (N-k), of the form $X^{N-k} + \ldots + 1$
- Let G(X) divide X^N+1 without remainder.

Then G(X) generates a linear cyclic (N,K) block code with codewords (using our previous notation)

Q(X)=G(X)M(X)

Here M(X) is a message polynomial of max. degree (k-1).

Proof:

If $Q(X)=G(X)M(X)$ is a codeword generated this way, consider

$$XQ(X) = c_{N-1}(X^N+1) + S(X)$$

where $S(X)$ is the remainder in dividing the left side $XQ(X)$ by (X^N+1), and hence is a cyclic left-shift of the codeword $Q(X)$. But the left side above is divisible by $G(X)$ without remainder, and so is (X^N+1) on the right side. Therefore $S(X)$ must be divisible by $G(X)$, hence it is a codeword.

The linearity is obvious from the definition of the codewords as multiplications.

Systematic Code

The cyclic code generated above is not necessarily systematic, because codewords $G(X)M(X)$ do not necessarily produce the message bits in the first k positions of the word. However, we will see that the set of codewords generated this way contain all possible combinations of k bits in their first k positions, and so it is possible to re-assign codewords in a systematic way to message words.

Consider polynomial $Q_1(X) = X^{N-k} M_1(X) + R_1(X)$ where $R_1(X)$ is the remainder upon dividing $X^{N-k} M_1(X)$ by $G(X)$. This polynomial consists of the message bits in the first k positions. We claim this is a codeword of the cyclic code, for which we have to show it is divisible by $G(X)$ But this is easy to see, because of the way $R_1(X)$ is defined as a remainder. Thus $Q_1(X)=G(X)M_2(X)$ for some $M_2(X)$. We find therefore that the distinct 2^k codewords of the form $Q_1(X)$ are codewords generated through the operation $G(X)M(X)$ defining our original cyclic code.

CRC Codes

Note that the generator polynomials G(X) of degree (N-k) (usually 12, 16, or 32) used for the common CRC codes for error detection correspond to cyclic codes for specific values of $N=N_0$; remember that X^N+1 has to be divisible by G(X), and these CRC polynomials correspond to very large values of such N_0 (by design, to have good error detection capability). However, in practice they can be used with shorter codeword lengths N so that they are not exactly cyclic codes. Nonetheless, their implementation remains simple and they inherit the error detection characteristics of the cyclic codes.

A definition: G(X) is a primitive polynomial of degree m means that G(X) divides $X^N +1$ for $N=2^m-1$ and not for any smaller value of N.

The CRC polynomials are often of the form (1+X) times a primitive polynomial; for example, the 12-bit CRC polynomial is $(X^{11}+X^2 +1)(X+1)$. The first factor is a primitive polynomial and produces a cyclic code with $N=2^{11}-1=2047$.
The extra factor (X+1) in the polynomial makes the FCS of length 12 rather than 11, with the same N=2047 so that k=2035. The 12-bit CRC polynomial also exactly divides $X^N +1$ for N=2047, because (X+1) always divides any X^N+1.