Cyclic Redundancy Checks (CRC)

Cyclic Redundancy Checks fall into a class of codes called Algebraic Codes; more specifically, CRC codes are Polynomial Codes. These are error-detecting codes, not error-correcting codes, but they have the ability to detect large numbers of errors in a single codeword with relative little redundancy, that is, with a reasonable number of check bits.

CRC codes are generally used when transmitting data over some medium which may be noisy. The corruption characteristics of a noisy channel is that errors don’t occur singly or doubly, but they occur in bursts. Consider the following two streams of data bits. The first was sent over a channel and the second is the same data stream as received at the other end of the channel.

```
0101010101010101010101010101010101010101010101010101010
0101010101010101010101010101011100111100111011011011100
```

Notice that the strings are identical except for a burst of 10 bits starting with the 17th bit. Although not every one of the 10 bits is in error, we consider the entire length of the string from the first erroneous bit to the last erroneous bit as the burst length.

A CRC code would break such an input string as shown above into equal-sized segments and add a set of CRC bits onto the end of each segment, forming a set of codewords. We will now discuss how these CRC bits are generated and their error-detecting capability.

Assume that the follow string of bits constitutes one of the segments we which to transmit.

```
101110001
```

The size of the error burst, b, which we desire to detect is the same as the number of CRC bits we add to the data to form the codeword. We will use division by a generator string and use the remainder of that division as the CRC bits. In preparation for this division we append b zeros to the data word. For the data string above assume that we wish to guarantee detection of a burst of up to 5 errors (b=5). Therefore, we add 5 low order zeros to the dataword to create the dividend:

```
10111000100000
```

CRC codes are, as previously mentioned, called polynomial codes. This is because we
can consider the 1’s and 0’s to be the coefficients of a polynomial. Applying this to our current example we can represent the above string as

\[ 1X^{13}+0X^{12}+1X^{11}+1X^{10}+1X^9+0X^8+0X^7+0X^6+1X^5+0X^4+0X^3+0X^2+0X^1+0X^0 \]

which we simplify by dropping all terms with a coefficient of zero and leaving out the ‘1’s to give us

\[ X^{13}+X^{11}+X^{10}+X^9+X^5 \] [E7]

The CRC code is generated with the use of another polynomial, such as

\[ X^5+X^4+X^2+1 \] [E8]

which is the polynomial representation of 110101. Note that the generator string has 6 bits, one more than the number of check bits we need to generate.

The highest power in such a generator polynomial indicates the largest burst of errors which is guaranteed to be detected by the code\(^9\). In the current example, the resulting code will detect all burst errors of 5 bits or less.

The actual CRC bits to be added to a data work such as E7 is produced by dividing the data polynomial (E7) by the generator polynomial (E8) and using the coefficients of the remainder polynomial as the CRC bits.\(^{10}\)

\[
(X^{13}+X^{11}+X^{10}+X^9+X^5)/X^5+X^4+X^2+1 = X^8+X^7+X^3 \quad \text{with a remainder of } X^3
\]

The details of this division are given below. It is important to note that all the subtractions in the division are done module 2. That is, subtraction at each bit is the same as addition with no carries being carried over. Specifically, in modulo 2 arithmetic,

\[
0-0 = 0+0 = 01-1 = 1+1 = 00-1 = 0+1 = 11-0 = 1+0 = 1
\]

The actual division is

---

\(^9\) In fact, a very high percentage of bursts greater than the power of the generator polynomial will also be detected.

\(^{10}\) You should recognize this as a modulo operation.
Since the generator is 6 bits long, we must use a remainder of 5 bits, so $X^3$ gives us a CRC code for the given data of 01000. The complete codeword is then

$$10111000101000$$ [E9]

Errors are detected by, at the receiving end of the transmission medium, dividing the received codeword by the same generator polynomial as was used to generate the CRC code at the sending end; if the remainder is zero then no error occurred in transmission, otherwise errors were detected (and the sender is asked to retransmit the codeword.)

Using the current example, here is the receiver’s operation when no error has occurred:

$$X^5 + X^4 + X^2 + 1 \quad / \quad X^8 + X^7 + X^3 + X^1 + X^3$$

Now let’s see what happens in our example if there is an error of burst length 5. That is, instead of receiving E9 suppose we receive

$$10111001111100$$

where the error burst started at the 8th bit (01010 has become 11111).

Repeating the division shown in E10 yields the following.
The remainder, $X^2$, is non-zero, so an error was detected.

Choosing the generator polynomial is not difficult, using the following rules:

1. The highest power in the generator polynomial is the same as the maximum burst size we wish to be able to detect.
2. The generator polynomial is not divisible by $X$. That is, there is always a ‘+1’ in the polynomial.
3. The generator is divisible by $X+1$.

The following characteristics are true of CRC codes developed using these rules:

a. All bursts affecting an odd number of bits are detected.
b. All bursts less then or equal to the degree of the generator polynomial are detected.
c. A high percentage of burst errors greater than the degree of the generator polynomial are also detected.
### Practice Problems

1. Which of the following polynomials is a valid generator polynomial?

   a. $X+1$  
   b. $X^2+1$  
   c. $X^2+X+1$  
   d. $X^{10}+X^8+X$  
   e. $X^9+X^7+X^3+X^2+X+1$  
   f. $X^3+X^2+1$  

2. What is the largest burst error size guaranteed to be detected by each of the following generator polynomials?

   a. $X^4+1$  
   b. $X^{16}+X^{15}+X^2+1$  
   c. $X^{12}+X^{11}+X^3+X+1$  
   d. $X^{16}+X^{12}+X^5+1$  

3. Using the generator polynomial $X^7+X^6+X^4+X^2+X+1$, find the CRC bits for the following data words.

   - a. 10110100101  
   - b. 1111010010  

   CRC = 1011001  
   CRC = 0001111

   - a. 10110100101  
   - b. 1111010010  

   CRC = 1010  
   CRC = 0100

4. Using the generator polynomial $X^4+X^3+X+1$, determine which of the following received codewords has errors.

   - a. 11110101010  
   - b. 1100100110100

   CRC = 1010  
   CRC = 0100