Cyclic Redundancy Check (CRC) Example, Part 2

Sean E. O'Connor

artifex@seanerikoconnor.freeservers.com

1. An Example with CRC-CCITT

The CRC-CCITT generator polynomial is

In[289]:= **g = x^16 + x^12 + x^5 + 1**

Out[289]= 1 + x^5 + x^12 + x^16

Let's assume we are encoding the message bit stream $00\,050\,400\,10_{16}$. The binary digits will be the coefficients of our message polynomial i(x), with the least significant bit being the constant term of the polynomial:

In[290]:= **i = x^26 + x^24 + x^18 + x^4 + x**

Out[290]= x + x^4 + x^18 + x^24 + x^26

Enter the code's blocklength n and message length k, and compute the the number of parity bits n-k.

In[291]:= **n = 2^15 - 1**

Out[291]= 32 767

In[292]:= **k = n - 16**

Out[292]= 32 751

In[293]:= **n - k**

Out[293]= 16

For systematic encoding, the parity is p(x) = $[-x^{n-k}\, i(x)]$ mod g(x) where we do modulo 2 arithmetic on the polynomial coefficients.

In[294]:= **p = PolynomialMod[x^(n-k) i, {g, 2}]**

Out[294]= x + x^2 + x^4 + x^5 + x^6 + x^7 + x^9 + x^12 + x^14

Writing the polynomial coefficients in binary we get the parity,
$0101\ 0011\ 1111\ 0110_{\ 2} = 53F5_{\ 16}$

Compute the systematically encoded codeword
c(x) = $x^{n-k}\, i(x)$ + p(x).

In[295]:= **c = Expand[x^(n-k) i + p]**

Out[295]= x + x^2 + x^4 + x^5 + x^6 + x^7 + x^9 + x^12 + x^14 + x^17 + x^20 + x^34 + x^40 + x^42

Compute the shifted syndrome s'(x) = $[x^{n-k}\, c(x)]$ mod g(x), modulo 2 on the polynomial coefficients. We should get zero.

In[296]:=
```
p = PolynomialMod[x^(n-k) c, {g, 2}]
```

Out[296]= 0

Add error to the codeword.

In[297]:=
```
cerror1 = c + x^11
```

Out[297]= $x + x^2 + x^4 + x^5 + x^6 + x^7 + x^9 + x^{11} + x^{12} + x^{14} + x^{17} + x^{20} + x^{34} + x^{40} + x^{42}$

Compute the shifted syndrome s'(x) = $[x^{n-k}\, c(x)]$ mod g(x) modulo 2 on the polynomial coefficients. We should get a non-zero answer.

In[298]:=
```
s = PolynomialMod[x^(n-k) cerror1, {g, 2}]
```

Out[298]= $1 + x^3 + x^5 + x^7 + x^8 + x^{11} + x^{15}$

On the other hand, if we add a multiple of a codeword, we won't see the error.

In[299]:=
```
cerror2 = c + x^2 g
```

Out[299]= $x + x^2 + x^4 + x^5 + x^6 + x^7 + x^9 + x^{12} + x^{14} + x^{17} + x^{20} + x^{34} + x^{40} + x^{42} + x^2 \left(1 + x^5 + x^{12} + x^{16}\right)$

In[300]:=
```
s = PolynomialMod[x^(n-k) cerror2, {g, 2}]
```

Out[300]= 0