Códigos CRC

Electrónica Digital 2020

Docente: Guido Sanchez

Inst. de Investigación en Señales, Sistemas e Inteligencia Computacional
Facultad de Ingeniería y Ciencias Hídricas
Universidad Nacional del Litoral

¿Qué vamos a ver en esta clase?

- Cómo generar y verificar mensajes con CRC
- Ejercicio tipo de la guía de problemas

El misterio de CRC

Depende de quién nos hable de CRC, tendremos diferentes respuestas...

- los diseñadores de hardware nos dirán que CRC es una repetición de la operación XOR a medida que el flujo de datos pasa...
- los ingenieros de software nos dirán que CRC es algo con lo que hay que lidiar byte a byte utilizando una *lookup table*...
- los matemáticos prefieren explicarlo como el resto de una división polinomial...
- los criptografos murmurarán en voz baja acerca de operaciones en el campo de Galois de orden 2...

Breve descripción de CRC

- Es un código de detección de errores que se basa en la división
- El contenido del mensaje es interpretado como una cadena de bits (dividendo) que es dividido por otro número binario fijo (divisor)
- El resto de esta división es el valor del *checksum*

Aritmética módulo-2

- Es un sistema aritmético donde cada resultado se toma módulo-2.
- Resumiendo, el resultado de la operación es 1 si el operando es impar y 0 si el operando es par.

Dado que en nuestra codificación de datos estamos tratando cadenas de bits, **todos** nuestros operandos serán 1 o 0. Mientras que nuestros operandos sean 1 o 0 y nuestros resultados sean módulo-2, **todos** los números que escribiremos serán 1 o 0

Operaciones módulo-2 sobre bits: Suma

Para sumar dos números, se toma el módulo-2 del resultado. La tabla de verdad para la operación suma es:

+	0	1
0	0	1
1	1	0

Recordando la operaciones de la lógica booleana, al observar esta tabla podemos ver que la suma es idéntica a la operación `xor`. Por este motivo, utilizaremos los términos "suma" y "xor" de forma indistinta.

Operaciones módulo-2 sobre bits: Multiplicación

Para multiplicar dos números en módulo-2 no se necesita definir un nuevo operador. La tabla de verdad para la operación multiplicación es:

$\times$	0	1
0	0	0
1	0	1

Donde se puede ver que exactamente igual a la tabla de verdad de la operación booleana `and`. Por este motivo, utilizaremos los términos "multiplicación" y "and" de forma indistinta.

Algoritmo de CRC

Definimos

M : Mensaje de m bits.

C : divisor de c bits

M puede variar, pero se debe cumplir que $m > c - 1$. Nuestro campo CRC consiste de una cadena R de $c - 1$ bits.

Algoritmo de CRC

1. Agrego $c-1$ "0" al final de M. Estos determinan el lugar en el que se guardará el CRC. Llamamos M' a este mensaje extendido: M' = M + R.
2. Dividir M' por C utilizando **aritmética de módulo-2**. Prestar atención que módulo-2 no equivale a base-2. Recordemos que estas cantidades no son números en el sentido usual (cada vez que se divide C (de *c* bits) por otra cantidad Z de c-bits, decimos que C "entra" en Z una vez si el primer bit es 1). Llamaremos R al resto de C-1 bits.
3. Reemplazar los C-1 '0' que agregamos por el resto. Este es nuestro mensaje con CRC.
4. Enviar nuestro mensaje con CRC al receptor.

Generación del mensaje

M=110101; C=101

1	1	0	1	0	1	0	0	Mensaje con 2 ceros agregados
1	0	1						Alineo el polinomio generador
	1	1	1					XOR con los datos
	1	0	1					Desplazamiento a la derecha
		1	0	0				XOR con los datos
		1	0	1				Desplazamiento a la derecha
				1	1	0		XOR con los datos
				1	0	1		Alineo el polinomio generador
					1	1	0	XOR con los datos
					1	0	1	Desplazamiento a la derecha
						1	1	<- Resultado luego del último XOR

El mensaje a enviar es 11010111

Comprobación del mensaje

El mensaje recibido es 11010111

1	1	0	1	0	1	1	1	Datos con CRC recibidos
1	0	1						Alineo el polinomio generador
	1	1	1					XOR con los datos
	1	0	1					Desplazamiento a la derecha
		1	0	0				XOR con los datos
		1	0	1				Desplazamiento a la derecha
				1	1	1		XOR con los datos
				1	0	1		Alineo el polinomio generador
					1	0	1	XOR con los datos
					1	0	1	Desplazamiento a la derecha
						0	0	<- Resultado luego del último XOR

El mensaje no contiene errores

¡Manos a la obra! Especificación del problema

Deseamos transmitir el mensaje "Hola" utilizando ASCII y CRC-3-GSM. ¿Cómo procedemos?

1. Buscamos la especificación CRC-3-GSM[1]. El polinomio de CRC-3-GSM es $x^3 + x + 1$.
2. Convertimos el mensaje a código ASCII[2]. De la tabla ASCII podemos obtener la codificación hexadecimal y/o binaria de cada caracter. Recordar que ASCII utiliza 7 bits para representar cada caracter.

[1] `https://en.wikipedia.org/wiki/Cyclic_redundancy_check`
[2] https://en.wikipedia.org/wiki/ASCII

¡Manos a la obra! Codificación a ASCII

La tabla de conversión para cada caracter de nuestro mensaje es la siguiente

Caracter	ASCII (hex)	ASCII (bin)
H	072	1001000
o	111	1101111
l	108	1101100
a	097	1100001

Mensaje en binario

1001000110111111011001100001

¡Manos a la obra! Calculamos el CRC

- El polinomio de CRC-3-GSM es $x^3 + x + 1$. Es decir que sus coeficientes son 1011.
- Deberemos agregar 000 a nuestro mensaje para calcular la división módulo 2 utilizando XOR.

¡Manos a la obra! Calculamos el CRC

```
10010001101111110110011000001 000
1011
00100001101111110110011000001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
       1011
00000000100111111011001100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
       1011
00000000100111111011001100001 000
        1011
00000000001111111011001100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
       1011
00000000100111111011001100001 000
        1011
00000000001111111011001100001 000
          1011
00000000000100110111001100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
       1011
00000000100111111011001100001 000
        1011
00000000001111111011001100001 000
          1011
00000000000100111011001100001 000
           1011
00000000000001010111001100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
       1011
00000000100111111011001100001 000
        1011
00000000001111111011001100001 000
          1011
00000000000100111011001100001 000
           1011
00000000000000101011001100001 000
             1011
00000000000000000111001100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
       1011
00000000100111111011001100001 000
        1011
00000000001111111011001100001 000
          1011
00000000000100111011001100001 000
           1011
00000000000000101011001100001 000
             1011
00000000000000000111001100001 000
                1011
00000000000000000010101100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
       1011
00000000100111111011001100001 000
        1011
00000000001111111011001100001 000
          1011
00000000000100111011001100001 000
           1011
00000000000000101011001100001 000
             1011
00000000000000000111001100001 000
                1011
00000000000000000010101100001 000
                 1011
00000000000000000000011100001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
       1011
00000000100111111011001100001 000
        1011
00000000001111111011001100001 000
          1011
00000000000100111011001100001 000
           1011
00000000000000101011001100001 000
             1011
00000000000000000111001100001 000
               1011
00000000000000000010101100001 000
                 1011
00000000000000000000011100001 000
                   1011
00000000000000000000001010001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
       1011
00000000100111111011001100001 000
        1011
00000000001111111011001100001 000
          1011
00000000000100111011001100001 000
           1011
00000000000000101011001100001 000
             1011
00000000000000000111001100001 000
                1011
00000000000000000010101100001 000
                 1011
00000000000000000000011100001 000
                   1011
00000000000000000000001010001 000
                    1011
00000000000000000000000001001 000
```

¡Manos a la obra! Calculamos el CRC

```
10010001101111111011001100001 000
1011
00100001101111111011001100001 000
  1011
00001101101111111011001100001 000
    1011
00000110101111111011001100001 000
     1011
00000011001111111011001100001 000
      1011
00000001111111111011001100001 000
       1011
00000000100111111011001100001 000
        1011
00000000001111111011001100001 000
          1011
00000000000100111011001100001 000
           1011
00000000000000101011001100001 000
             1011
00000000000000000111001100001 000
               1011
00000000000000000010101100001 000
                1011
00000000000000000000011100001 000
                  1011
00000000000000000000001010001 000
                   1011
00000000000000000000000001001 000
                     1011
00000000000000000000000000010 000
                       10 11
00000000000000000000000000000 100
```

¡Manos a la obra! Calculamos el CRC

Algoritmo finalizado

Checksum CRC: 100

Se envía el mensaje 10010001101111110110011000011100

Referencias

1. Apuntes de la cátedra "Códigos de verificación por redundancia cíclica"
2. Data Coding Theory/Modulo-2 Arithmetic: https://en.wikibooks.org/wiki/Data_Coding_Theory/Modulo-2_Arithmetic
3. Cyclic redundancy check: https://en.wikipedia.org/wiki/Cyclic_redundancy_check
4. Verificación de redundancia cíclica: https://es.wikipedia.org/wiki/Verificaci%C3%B3n_de_redundancia_c%C3%ADclica
5. Polynomial long division: https://en.wikipedia.org/wiki/Polynomial_long_division
6. CRC Generating and Checking http://ww1.microchip.com/downloads/en/appnotes/00730a.pdf

Referencias

7. Understanding and implementing CRC (Cyclic Redundancy Check) calculation `http://www.sunshine2k.de/articles/coding/crc/understanding_crc.html`
8. CRC Basics `https://www.digikey.com/eewiki/display/microcontroller/CRC+Basics`
9. Tutorial: Cyclic Redundancy Check (CRC) computation `https://netfuture.ch/tutorials/crc/`