UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ORGANIZACIÓN COMPUTACIONAL

SECCIÓN “A”

TAREA 1

CODIGOS BINARIOS Y COMPLEMENTOS EN BASE “M” Y “M-1”

KARLA JULISSA AJTÚN VELÁSQUEZ

201700565

* Códigos Binarios

La codificación es el proceso en el que se traduce o hace conversión de cierta información a algún conjunto de símbolos ya creados.

* Códigos Numéricos

**BCD**

El binario decimal codificado, es una forma directa asignada a un equivalente binario. Es una conversión directa para números decimales que sean enteros entre 0 y 9, para los que son mayores a 9 la codificación es totalmente diferentes. Los códigos binarios para números decimales requieren un mínimo de cuatro bits. Con esto se puede obtener varios códigos combinados con cuatro o mas bits. En esta tabla se ven algunos de las posibles combinaciones:

Digital decimal BDC

0 0000

1 0001

2 0010

3 0011

La diferencia entre una conversión de un decimal a binario y una codificación de un decimal a binario es que los bits que se obtiene en una conversión son dígitos y los que se obtiene en una codificación son combinaciones de 1’s y 0’s que están arregladas de acuerdo con ciertas reglas definidas por el código usado.

Un ejemplo de esto es, que el número 51 en número binario sería 110011 mientras que en código binario seria 1011.

**REFLEJADO**

Se caracteriza porque al cambiar en un numero a otro solo cambia un bit y es a prueba de errores. Este código también es conocido como código Gray. Un ejemplo de esto es el siguiente: dado el numero 0001 (uno en decimal), el siguiente en binario seria 0010, pero con el código gray, ya que solo se puede cambiar un bit pasando de un numero al siguiente, este quedaría así 0011, ya que solo se cambió el segundo 1.

La siguiente tabla muestra el código reflejado en cuatro bits

Código Reflejado Su equivalente en Decimal

0011 2

0010 3

0110 4

0111 5

0101 6

**Exceso “3”**

Es un código auto complementario. Este código se caracteriza por el 3 que se le suma a cada digito decimal, para luego pasarlo a binario, es decir, si queremos pasar el 4 a exceso 3, le sumamos 3, quedaría el número 7, eso en binario sería 0111. Ahora para usar ya en sí, el código exceso “3” para números decimales (que ya no sean dígitos) se hace lo siguiente, supongamos que tenemos el número decimal 56, se le suma 3 a la unidad y 3 a la decena, esto nos da como resultado 89, y luego ya se pasa en BCD, que sería 1000 1001.

* CÓDIGOS ALFA-NUMÉRICOS

Consiste en letras y números. Es un código binario de un grupo de elementos que consiste en 10 números decimales, las 26 letras del alfabeto y algunos símbolos especiales.

**Código ASCII**

Se diseñó este código por la necesidad de representar más de 64 caracteres. Sus siglas en ingles American Standard Code for Information Interchange, que significa código normalizado americano para el intercambio de información. Este código consiste en siete bits, para efectos de paridad se agrega un octavo bit. Este permite escribir cualquier símbolo solo con números.

* Códigos Binarios de error

**Paridad**

Un bit de paridad es un bit extra, que está incluido con el mensaje para convertir el número total de 1’s par o impar. Se detecta el error si la paridad no corresponde a la adoptada. Por otra parte, una combinación par de errores no se puede detectar. Este método no es muy seguro del todo, se podrían dar casos en los que hay error y estos no ser detectados. Este es limitado por encontrar un error en un conjunto de bits, como máximo.

**Comprobación de redundancia cíclica (CRC)**

Su objetivo es poder detectar algún cambio accidental hecha en algún dato. El proceso de esto consiste en asignar a un bloque un segmento de datos. Este contiene datos redundantes con el bloque de manera que halle una solución al error que se haya detectado.

* Complemento en base “m” y “m-1”

**Complemento r**

Este es reconocido como complemento a 2, dicho complemento de n bits será 2^n-N. Este puede obtenerse sumando 1 al complemento. Este se caracteriza por que los bits más significativos se invierten. Por ejemplo: el complemento de 2 . En caso que no existiera un numero entero, solo decimal, este sería 2 elevado a la 0, por lo tanto, es uno, entonces, solo se hace una resta de 1 con el decimal que es.

**Complemento a 10**

El complemento de n bits será 10^n, en este caso 10^n representa un 1 seguido por n 0’s, otra forma de verlo es sumando uno al complemento de 9. Por ejemplo: el complemento de 10 de . El número de dígitos del número es n=5. Por lo tanto, es 10 elevado a la 5.

**Complemento r-1**

Complemento a 1

El complemento de n bits será 2^(n-1) por ejemplo, n sea igual a 6, entonces 2 elevada a la potencia 6 es igual a 1000000(64 en decimal) aplicando el complemento quedaría 64-1 en forma binaria 111111. Este complemento es como la representación de un doble cero, este complemento es considerado un paso intermedio para la complementación de la base. Por ejemplo: el complemento de 1 de .

**Complemento a 9**

10^n-1 representa un número que está formado por n nueves, en resumen, el complemento a 9 se obtiene restando 9 a cada digito del número decimal que se desee. Por ejemplo: el complemento de 0 de (52520) es .