TRABAJO DE INVESTIGACION SOBRE LAS CONVERCIONES DE LOS SISTEMAS NUMERICOS

JIMMY DADNOVER ROZO GUERRERO

UNISANGIL

LOGICA DE PROGRAMACION

INGENIERIA DE SISTEMAS

CHIQUINQUIRA – BOY

2015

TRABAJO DE INVESTIGACION SOBRE LAS CONVERCIONES DE LOS SISTEMAS NUMERICOS

JIMMY DADNOVER ROZO GUERRERO

Trabajo presentado para adquirir conocimiento sobre

Conversiones de los sistemas numéricos

LICENCIADO

William Alexander Matallana Porras

Docente

UNISANGIL

LOGICA DE PROGRAMACION

INGENIERIA DE SISTEMAS

CHIQUINQUIRA – BOY

2015

CONTENIDO

Pág.

LISTA DE FIGURAS

1. SISTEMAS DE NUMEROS BINARIOS

Este es el sistema numérico que utilizan los sistemas digitales para contar y es el código al que traduce todas las informaciones que recibe. Se dice "Binario" a todo aquello que tiene dos partes, dos aspectos, etc. Muchas cosas en los sistemas digitales son binarias: Los impulsos eléctricos que circulan en los circuitos son de baja o de alta tensión, los interruptores biestables están encendidos o apagados, abiertos o cerrados, etc.

A diferencia del sistema decimal al que estamos habituados, y que utiliza diez cifras, del 0 al 9, el sistema numérico binario utiliza solo dos cifras, el 0 y el 1. En el sistema binario las columnas no representan la unidad, la decena, la centena, como en el sistema decimal, sino la unidad (20), el doble (21), el doble (22), etc. De modo que al sumar en la misma columna 1 y 1, dará como resultado 0, llevándonos 1 a la columna inmediatamente a la izquierda. Para los sistemas digitales es fácil, hasta el punto que reduce todas las operaciones a sumas y restas de números binarios.

Figura 1: Sistema binario

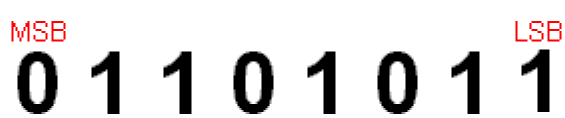

Figura 2: Números binarios

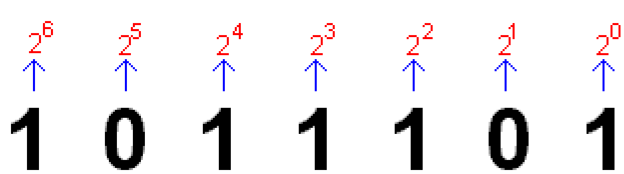

Figura 3: Valores de las posiciones de los números binarios

1.1 SUMA DE NÚMEROS BINARIOS

Suma de números binarios
La tabla de sumar para números binarios es la siguiente:

+	**0**	**1**
0	0	1
1	1	10

Figura 4: Tabla para sumar binarios

Las posibles combinaciones al sumar dos bits son:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10
Note que al sumar 1 + 1 es 102, es decir, llevamos 1 a la siguiente posición de la izquierda (acarreo). Esto es equivalente en el sistema decimal a sumar 9 + 1, que da 10: cero en la posición que estamos sumando y un 1 de acarreo a la siguiente posición.

Ejemplo

```
    1
  10011000
+ 00010101
__________
  10101101
```

Se puede convertir la operación binaria en una operación decimal, resolver la decimal, y después transformar el resultado en un (número) binario. Operamos como en el sistema decimal: comenzamos a sumar desde la derecha, en nuestro ejemplo, 1 + 1 = 10, entonces escribimos 0 en la fila del resultado y llevamos 1 (este "1" se llama acarreo o arrastre). A continuación se suma el acarreo a la siguiente columna: 1 + 0 + 0 = 1, y seguimos hasta terminar todas las columnas (exactamente como en decimal).

1.2 RESTA DE NÚMEROS BINARIOS

El algoritmo de la resta en sistema binario es el mismo que en el sistema decimal. Pero conviene repasar la operación de restar en decimal para comprender la operación binaria, que es más sencilla. Los términos que intervienen en la resta se llaman minuendo, sustraendo y diferencia.

Las restas básicas 0 - 0, 1 - 0 y 1 - 1 son evidentes:

0 - 0 = 0
1 - 0 = 1
1 - 1 = 0
0 - 1 = 1 (se transforma en 10 - 1 = 1) (en sistema decimal equivale a 2 - 1 = 1)

La resta 0 - 1 se resuelve igual que en el sistema decimal, tomando una unidad prestada de la posición siguiente: 0 - 1 = 1 y me llevo 1 (este valor se resta al resultado que obtenga, entre el minuendo y el sustraendo de la siguiente columna), lo que equivale a decir en el sistema decimal, 2 - 1 = 1.

Ejemplos

```
  10001          11011001
 -01010         -10101011
_______        __________
  00111          00101110
```

En sistema decimal sería: 17 - 10 = 7 y 217 - 171 = 46.

Para simplificar las restas y reducir la posibilidad de cometer errores hay varios métodos:

Dividir los números largos en grupos. En el siguiente ejemplo, vemos cómo se divide una resta larga en tres restas cortas:

```
  100110011101       1001    1001    1101
 -010101110010      -0101   -0111   -0010
______________  =  ______  ______  ______
  010000101011       0100    0010    1011
```

Utilizando el complemento a dos (C2). La resta de dos números binarios puede obtenerse sumando al minuendo el «complemento a dos» del sustraendo.

Ejemplo
La siguiente resta, 91 - 46 = 45, en binario es:

1011011		1011011
-0101110	el C2 de 0101110 es 1010010	+1010010
0101101		10101101

En el resultado nos sobra un bit, que se desborda por la izquierda. Pero, como el número resultante no puede ser más largo que el minuendo, el bit sobrante se desprecia.

Un último ejemplo: vamos a restar 219 - 23 = 196, directamente y utilizando el complemento a dos:

11011011		11011011
-00010111	el C2 de 00010111 es 11101001	+11101001
11000100		111000100

Y, despreciando el bit que se desborda por la izquierda, llegamos al resultado correcto: 11000100 en binario, 196 en decimal.

Utilizando el complemento a uno. La resta de dos números binarios puede obtenerse sumando al minuendo el complemento a uno del sustraendo y a su vez sumarle el bit que se desborda.

1.3 PRODUCTO DE NÚMEROS BINARIOS

La tabla de multiplicar para números binarios es la siguiente:

·	0	1
0	0	0
1	0	1

El algoritmo del producto en binario es igual que en números decimales; aunque se lleva a cabo con más sencillez, ya que el 0 multiplicado por cualquier número da 0, y el 1 es el elemento neutro del producto.

Por ejemplo, multipliquemos 10110 por 1001:

```
10110
     1001
_________
    10110
   00000
  00000
 10110
_________
 11000110
```

Figura 5: Ejemplo de multiplicación en binario

En sistemas electrónicos, donde suelen usarse números mayores, se utiliza el método llamado algoritmo de Booth.

```
       11101111
         111011
      __________
       11101111
      11101111
     00000000
    11101111
   11101111
  11101111
 _____________
 11011100010101
```

Figura 6: Ejemplo de multiplicación en binario

1.4 DIVISIÓN DE NÚMEROS BINARIOS

La división en binario es similar a la decimal; la única diferencia es que a la hora de hacer las restas, dentro de la división, éstas deben ser realizadas en binario.

Ejemplo

Dividir 100010010 (274) entre 1101 (13):

```
 100010010 /1101 = 010101
 -0000
_______
 10001
 -1101
_______
  01000
 - 0000
 _______
   10000
  - 1101
  _______
    00111
   - 0000
   _______
     01110
    - 1101
    _______
     00001
```

Figura 7: Ejemplo de división en binario

2. SISTEMA DE NUMERACION OCTAL

Este sistema consta de 8 símbolos desde el 0 hasta el 7, es muy poco utilizado en los computadores. La facilidad con que se pueden convertir entre el sistema Octal y el binario hace que el sistema Octal sea atractivo como un medio "taquigráfico" de expresión de números binarios grandes. Cuando trabajamos con una gran cantidad de números binarios de muchos bits, es más adecuado y eficaz escribirlos en octal y no en binarios. Sin embargo, recordemos los circuitos y sistemas digitales trabajan eléctricamente en binario, usamos el sistema Octal solo por conveniencia con los operadores del sistema.

Los números octales pueden construirse a partir de números binarios agrupando cada tres dígitos consecutivos de estos últimos (de derecha a izquierda) y obteniendo su valor decimal.

Por ejemplo, el numero binario para 74 (en decimal) es 1001010 (en binario), lo agruparíamos como 1 001 010. De modo que el número decimal 74 en octal es 112

$$N = d_n \dots d_1 d_0, d_{-1} \dots d_{-k} =$$

$$d_n \cdot 8^n + \dots + d_1 \cdot 8^1 + d_0 \cdot 8^0, + d_{-1} \cdot 8^{-1} + \dots + d_{-k} \cdot 8^{-k} =$$

$$N = \sum_{i=-k}^{n} d_i \cdot 8^i$$

Figura 8: Teorema fundamental aplicado al sistema octal

3. SISTEMAS DE NUMERACION HEXADECIMAL

El **sistema hexadecimal** (a veces abreviado como **Hex**, no confundir con *sistema sexagesimal*) es el sistema de numeración posicional que tiene como base el 16. Su uso actual está muy vinculado a la informática y ciencias de la computación, pues los computadores suelen utilizar el byte u octeto como unidad básica de memoria; y, debido a que un byte representa 2^8 valores posibles, y esto puede representarse como $2^8 = 2^4 \cdot 2^4 = 16 \cdot 16 = 1 \cdot 16^2 + 0 \cdot 16^1 + 0 \cdot 16^0$, que equivale al número en base 16 100_{16}, dos dígitos hexadecimales corresponden exactamente a un byte.

En principio, dado que el sistema usual de numeración es de base decimal y, por ello, sólo se dispone de diez dígitos, se adoptó la convención de usar las seis primeras letras del alfabeto latino para suplir los dígitos que nos faltan. El conjunto de símbolos sería, por tanto, el siguiente:

$$S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{F}\}$$

Figura 9: Símbolos del sistema hexadecimal.

Se debe notar que A = 10, B = 11, C = 12, D = 13, E = 14 y F = 15. En ocasiones se emplean letras minúsculas en lugar de mayúsculas. Como en cualquier sistema de numeración posicional, el valor numérico de cada dígito es alterado dependiendo de su posición en la cadena de dígitos, quedando multiplicado por una cierta potencia de la base del sistema, que en este caso es 16. Por ejemplo: $3E0A_{16}$ = $3 \times 16^3 + E \times 16^2 + 0 \times 16^1 + A \times 16^0$ = 3×4096 + 14×256 + 0×16 + 10×1 = 15882.

3.1 SUMA HEXADECIMAL

9 + 7 = 16 (16 - 16 = 0 nos llevamos 1 y es = 10

En este caso la respuesta obtenida, 16, no está entre el 0 y el 15, por lo que tenemos que restarle 16. Por lo tanto, la respuesta obtenida será 10 (sistema hexadecimal). Hay que tener cuidado de utilizar correctamente las letras, ya que operar a la vez con letras y números puede crear confusiones.

A + 6 = 16 (16 - 16 = 0 y nos llevamos 1)
Ocurre lo mismo que en el ejemplo anterior.

A + A = 20 (20 – 16 = 4 y nos llevamos 1)

La respuesta es 20 y no está entre el 0 y el 15, por lo que tenemos que restarle 16. Por lo tanto, la respuesta obtenida será 14 (sistema hexadecimal).

Hay que tener cuidado de utilizar correctamente las letras, ya que operar a la vez con letras y números puede crear confusiones.

F + E = 29 (29 – 16 = D y nos llevamos 1)

La respuesta es 29 y no está entre el 0 y el 15, por lo que tenemos que restarle 16. Por lo tanto, la respuesta obtenida será 1D (sistema hexadecimal).

Hay que tener cuidado de utilizar correctamente las letras, ya que operar a la vez con letras y números puede crear confusiones.

Ahora haremos una operación más complicada:

A + 2 = 12 (12 corresponde a C)

Ten en cuenta que puedes comprobar los resultados utilizando una calculadora científica.

3.2 RESTA HEXADECIMAL

3.2.1 Complemento C15

Como podemos hacer la resta de dos números hexadecimales utilizando el complemento a 15. Para ello tendremos que sumar al minuendo el complemento a quince del sustraendo, y finalmente sumarle el bit de overflow (bit que se desborda).

Para entender la resta en complemento a 15 lo analizaremos con un ejemplo. Ésta es la resta que tenemos que resolver:

```
  A4FC9
-   DE8
_________
 ¿?¿?¿?¿?
```

Figura 10: ejemplo de la resta C15

Primero tenemos que hacer que el minuendo y el sustraendo tengan la misma cantidad de números. Para ello, añadiremos ceros al sustraendo hasta que sean suficientes.

```
  A4FC9
- 00DE8
_________
 ¿?¿?¿?¿?
```

Figura 11: segundo ejemplo

Después, crearemos un nuevo número con la misma cantidad de números que el nuevo sustraendo. Como en el sistema hexadecimal el mayor número que tenemos es el 15, que corresponde a la letra F, tendremos que escribir la F tantas veces como números tiene el sustraendo.

```
  FFFFF
- 00DE8
_________
  FF217
```

Figura 12: tercer ejemplo

La resta se hace siguiendo las normas generales de la resta común. La diferencia obtenida se denomina el complemento a 15. Recuerda el valor correspondiente a cada letra al operar.

Ahora tendremos que sumar el minuendo y el complemento a 15 utilizando la suma en sistema hexadecimal, mencionada anteriormente.

```
  A4FC9
+ FF217
_________
 1A41E0
```

Figura 13: Cuarto ejemplo, utilizando suma.

Con la suma obtenemos el resultado 1A41E0, pero no es la respuesta final. Te habrás dado cuenta que este nuevo número tiene más cifras que los números iniciales que teníamos que restar. Tenemos que quitar el número de la izquierda (en este caso, el 1) y sumarlo.

```
  A41E0
+     1
_________
  A41E1
```

Figura 14: Ejemplo resuelto

La respuesta es A41E1.

Ten en cuenta que puedes comprobar los resultados utilizando una calculadora científica.

3.2.2 Complemento C16

También podemos hacer la resta de dos números hexadecimales utilizando el complemento a 16, siguiendo un proceso similar que en el caso del complemento a 15. Para resolver la resta, tendremos que sumar al minuendo el complemento a dieciséis del sustraendo.

Para entender la resta en complemento a 16 lo analizaremos con el ejemplo anterior. Ésta es la resta que tenemos que resolver:

```
  A4FC9
-   DE8
_________
 ¿?¿?¿?¿?
```

Figura 15: Ejemplo C16

Primero tenemos que hacer que el minuendo y el sustraendo tengan la misma cantidad de números, al igual que ocurre en el proceso del complemento a 15.

Para ello, añadiremos ceros al sustraendo hasta que sean suficientes.

```
  A4FC9
- 00DE8
_________
 ¿?¿?¿?¿?
```

Figura 16: Proceso para la resta C16

Después, crearemos un nuevo número con la misma cantidad de números que el nuevo sustraendo.

Como en el sistema hexadecimal el mayor número que tenemos es el 15, que corresponde a la letra F, tendremos que escribir la F tantas veces como números tiene el sustraendo.

```
  FFFFF
- 00DE8
_________
  FF217
```

Figura 17: Segundo ejemplo resta C16

La resta se hace siguiendo las normas generales de la resta común.

Ahora tenemos que **sumarle 1 a la diferencia obtenida**. Este paso es muy importante, ya que es la diferencia entre hacer la resta en complemento a 15 ó 16, y se suele olvidar fácilmente. Además, recuerda que estás sumando en sistema hexadecimal, siguiendo el mismo proceso explicado anteriormente.

```
  FF217
+     1
_________
  FF218
```

Figura 18: Suma en C16

A la diferencia obtenida y sumarle uno le denominaremos el complemento a 16.

Ahora tendremos que sumar el minuendo y el complemento a 16

```
  A4FC9
+ FF218
_________
 1A41E1
```

Figura 19: Ejemplo resuelto

Con la suma obtenemos el resultado 1A41E1.

Te habrás dado cuenta que este nuevo número tiene más cifras que los números iniciales que teníamos que restas, cosa imposible en una resta (que la diferencia sea mayor que el minuendo y el sustraendo). Por eso, y estando en complemento a 16, tendremos que despreciar (eliminar) el número de la izquierda. En este caso es el 1.

La respuesta, por lo tanto, es **A41E1**.

En ambos casos la respuesta obtenida deberá ser la misma, ya que hemos resuelto la misma resta en sistema hexadecimal. Por lo tanto, podremos comprobar que hemos operado bien comparando las respuestas obtenidas en complemento a 15 y en complemento a 16 para una misma resta.

Además, ten en cuenta que puedes comprobar los resultados utilizando una calculadora científica.

BIBLIOGRAFIA

Información obtenida de:

- http://definicion.de/sistema-binario/
- http://es.wikipedia.org/wiki/Sistema_binario
- http://es.wikipedia.org/wiki/Sistema_hexadecimal
- http://www.ladelec.com/teoria/electronica-digital/147-sistemas-de-numeracion
- http://www.ladelec.com/teoria/electronica-digital/148-conversiones-de-sistemas-de-numeracion