ARITMÉTICA BINARIA

Operaciones elementales con números binarios

La Unidad Aritmético Lógica, en la CPU del procesador, es capaz de realizar operaciones aritméticas, con datos numéricos expresados en el sistema binario. Naturalmente, esas operaciones incluyen la adición, la sustracción, el producto y la división. Las operaciones se hacen del mismo modo que en el sistema decimal, pero debido a la sencillez del sistema de numeración, pueden hacerse algunas simplificaciones que facilitan mucho la realización de las operaciones.

Suma en binario

Para aprender a sumar, con cinco o seis años de edad, tuviste que memorizar las 100 combinaciones posibles que pueden darse al sumar dos dígitos decimales. La tabla de sumar, en binario, es mucho más sencilla que en decimal. Sólo hay que recordar cuatro combinaciones posibles:

+	0	1
0	0	1
1	1	0 + 1

Las sumas 0 + 0, 0 + 1 y 1 + 0 son evidentes:

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

Pero la suma de 1+1, que sabemos que es 2 en el sistema decimal, debe escribirse en binario con dos cifras (10) y, por tanto 1+1 es 0 y se arrastra una unidad, que se suma a la posición siguiente a la izquierda. Veamos algunos ejemplos:

010 + 101 = 111 $2_{10} + 5_{10} = 7_{10}$

001101 + 100101 = 110010 $13_{10} + 37_{10} = 50_{10}$

1011011 + 1011010 = 10110101 $91_{10} + 90_{10} = 181_{10}$

110111011 + 100111011 = 1011110110 $443_{10} + 315_{10} = 758_{10}$

Ejercicio 1:
Realiza las siguientes sumas de números binarios:
111011 + 110
111110111 + 111001
10111 + 11011 + 10111

Sustracción en binario

La técnica de la resta en binario es, nuevamente, igual que la misma operación 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**.

-	0	1
0	0	1

1	1 + 1	0

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

0 – 0 = 0

1 – 0 = 1

1 – 1 = 0

La resta 0 - 1 se resuelve, igual que en el sistema decimal, tomando una unidad prestada de la posición siguiente: 10 - 1, es decir, $2_{10} - 1_{10} = 1$. Esa unidad prestada debe devolverse, sumándola, a la posición siguiente. Veamos algunos ejemplos:

111 – 101 = 010 $7_{10} - 5_{10} = 2_{10}$

10001 – 01010 = 00111 $17_{10} - 10_{10} = 7_{10}$

11011001 – 10101011 = 00101110 $217_{10} - 171_{10} = 46_{10}$

111101001 – 101101101 = 001111100 $489_{10} - 365_{10} = 124_{10}$

Ejercicio 2:

Realiza las siguientes restas de números binarios y comprueba los resultados convirtiéndolos al sistema decimal:

111011 - 110

111110111 - 111001

1010111 - 11011 – 10011

A pesar de lo sencillo que es el procedimiento de restar, es facil confundirse. Tenemos interiorizado el sistema decimal y hemos aprendido a restar mecánicamente, sin detenernos a pensar en el significado del arrastre. Para simplificar las restas y reducir la posibilidad de cometer errores hay varias soluciones:

- **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

- **Calculando el complemento a dos del sustraendo**

i. **Complemento a dos**

El ***complemento a dos*** de un número N, compuesto por n bits, se define como:

$$C_{2N} = 2^n - N$$

Veamos un ejemplo: tomemos el número $N = 101101_2$, que tiene 6 bits, y calculemos su complemento a dos:

$N = 45_{10}$ $n = 6$ $2^6 = 64$ y, por tanto: $C_{2N} = 64 - 45 = 19 = 010011_2$

Ejercicio 3:

Calcula el complemento a dos de los siguientes números:

11001, 10001011, 110011010

ii. **Complemento a uno**

El ***complemento a uno*** de un número N, compuesto por n bits es, por definición, una unidad menor que el complemento a dos, es decir:

$$C_{1N} = C_{2N} - 1$$

y, por la misma razón:

$$C_{2N} = C_{1N} + 1$$

Calculemos el ***complemento a uno*** del mismo número del ejemplo anterior:

siendo $N = 101101$, y su complemento a dos $C_{2N} = 010011$

$$C_{1N} = C_{2N} - 1 = 010011 - 000001 = 010010$$

$$C_{1N} = 010010$$

Da la sensación de que calcular el complemento a uno no es más que una forma elegante de comlicarse la vida, y que no va a ser más sencillo restar utilizando el complemento a dos, porque el procedimiento para calcular el complemento a dos es más difícil y laborioso que la propia resta. Pero es mucho más sencillo de lo que parece.

En realidad, el ***complemento a uno*** de un número binario es el número resultante de invertir los UNOS y CEROS de dicho número. Por ejemplo si:

$$N = 110100101$$

obtenemos su complemento a uno invirtiendo ceros y unos, con lo que resulta:

$$C_{1N} = 001011010$$

y su complemento a dos es:

$$C_{2N} = C_{1N} + 1 = 001011011$$

¡es muy fácil!

Veamos otro ejemplo de cálculo de complementos. Sea:

$$N = 0110110101$$

El complemento a uno es:

$$C_{1N} = 1001001010$$

y el complemento a dos es:

$$C_{2N} = 1001001011$$

iii. Restar en binario usando el complemento a dos

Y, por fin, vamos a ver cómo facilita la resta el complemento. La resta binaria de dos números puede obtenerse ***sumando al minuendo el complemento a dos del sustraendo***. Veamos algunos ejemplos:

Primer ejemplo:

Hagamos la siguiente resta, **91 – 46 = 45**, en binario:

$$1011011 - 0101110 = 0101101$$

Tiene alguna dificultad, cuando se acumulan los arrastres a la resta siguiente. Pero esta misma resta puede hacerse como una suma, utilizando el complemento a dos del sustraendo:

$$1011011 + 1010010 = 0101101$$

En el resultado de la suma 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.

Segundo ejemplo:

Hagamos esta otra resta, **219 – 23 = 196**, utilizando el complemento a dos:

$$219_{10} = 11011011_2,$$
$$23_{10} = 00010111_2$$
$$C_{223} = 11101001$$

El resultado de la resta será: $11011011 + 11101001 = 111000100$

Y, despreciando el bit que se desborda por la izquierda, llegamos al resultado correcto:

$$\boldsymbol{11000100_2 = 196_{10}}$$

¡Qué fácil!

Ejercicio 4:

Haz las siguientes restas binarias utilizando la técnica del complemento a dos. Al terminar, comprueba los resultados haciendo la resta en el sistema decimal:

11010001101 – 1000111101

10110011101 - 1110101

Multiplicación binaria

La multiplicación en binario es más fácil que en cualquier otro sistema de numeración. Como los factores de la multiplicación sólo pueden ser CEROS o UNOS, el producto sólo puede ser CERO o UNO. En otras palabras, las tablas de multiplicar del cero y del uno son muy fáciles de aprender:

x	0	1
0	0	0
1	0	1

En un ordenador, sin embargo, la operación de multiplicar se realiza mediante sumas repetidas. Eso crea algunos problemas en la programación porque cada suma de dos UNOS origina un arrastre, que se resuelven contando el número de UNOS y de arrastres en cada columna. Si el número de UNOS es par, la suma es un CERO y si es impar, un UNO. Luego, para determinar los arrastres a la posición superior, se cuentan las parejas de UNOS.

Veamos, por ejemplo, una multiplicación:

```
      110100010101
x             1101
  ----------------
      110100010101
     000000000000
    110100010101
   110100010101
  ----------------
  1010101000010001
```

Para comprobar que el resultado es correcto, convertimos los factores y el resultado al sistema decimal:

3349 * 13 = 43537

¡correcto!

Ejercicio 5:

Haz las siguientes multiplicaciones binarias. Al terminar, comprueba los resultados haciendo las multiplicaciones en el sistema decimal:

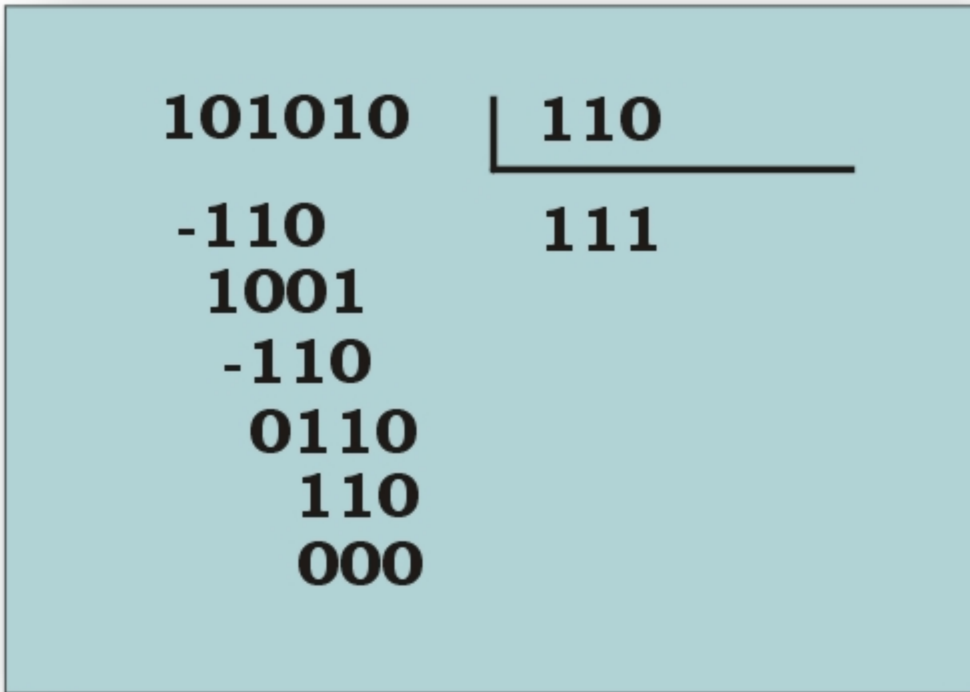
10110101000101 x 1011

10100001111011 x 10011

División binaria

Igual que en el producto, la división es muy fácil de realizar, porque no son posibles en el cociente otras cifras que UNOS y CEROS.

Consideremos el siguiente ejemplo, **42 : 6 = 7**, en binario:



Se intenta dividir el dividendo por el divisor, empezando por tomar en ambos el mismo número de cifras (100 entre 110, en el ejemplo). Si no puede dividirse, se intenta la división tomando un dígito más (1001 entre 100).

Si la división es posible, entonces, el divisor sólo podrá estar contenido **una vez** en el dividendo, es decir, la primera cifra del cociente es un UNO. En ese caso, el resultado de multiplicar el divisor por 1 es el propio divisor. Restamos las cifras del dividendo del divisor y bajamos la cifra siguiente.

El procedimiento de división continúa del mismo modo que en el sistema decimal.

Ejercicio 5:

Haz las siguientes divisiones binarias. Al terminar, comprueba los resultados haciendo las divisiones en el sistema decimal:

10110101000101 : 1011

10100001111011 : 10011

Luis González
Profesor de Tecnologías de la Información
I.E.S. Santa Eugenia (Madrid)

EJERCICIOS adicionales

1. Realiza las siguientes sumas de números octales:

   365 + 23
   2732 + 1265
   65 + 1773

2. Suma los siguientes números hexadecimales:

   17A + 3C
   20F5 + 31B
   2E70C + 1AA7F

3. Resta los siguientes números octales:

   365 - 23
   2732 - 1265
   1773 – 65

4. Realiza las siguientes restas de números hexadecimales:

   17A - 3C
   20F5 - 31B
   2E70C – 1AA7F

Arriba

Anterior **Inicio** **Siguiente**
Tecnologías de la información
Sistemas binarios