Operaciones en Binario Puro

Suma

Para empezar recordemos como es la suma en decimal

Si tenemos 3+5=8 en este caso la suma se puede representar en un dígito.

Si tenemos 9+7=16 en este caso no alcanza un dígito. Es una situación que todos sabemos resolver, pero estamos aplicando nuevamente el teorema de la numeración.

El dígito 1 es mas significativo en el número resultante. Con los símbolos posibles en el sistema de numeración decimal solo puedo llegar a 9 luego paso a 10 que es la base, ese 1 esta multiplicado por 10^1, en realidad decimos que es 1×10^1, solo que por simplicidad de notación lo representamos 10.

Esta misma idea se mantiene en binario, ¿Que quiere decir esto?

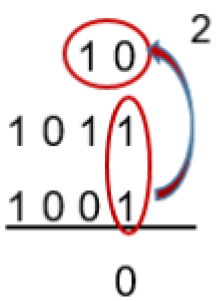
##0+0=0##
##1+0=1##
##0+1=1##
##1+1=10##, que es 2, o sea 1+1 es 2 como siempre, pero estoy representando en binario, 2 en binario es 10, porque lo que tenemos acá es $1\times2^1+0x2^0$

Con esta idea presente

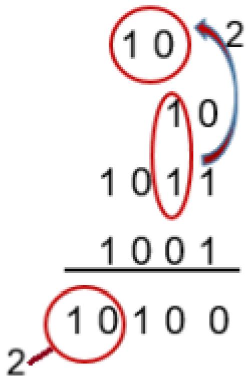
1011	11_{10}
100	4_{10}
1111	15_{10}

Hacemos la suma en binario y podemos comprobar en decimal si es correcta la operación

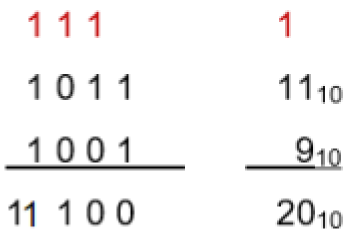
Vamos a un caso mas complejo



En este caso se presenta la situación de 1+1 que da 2, escribimos 10 o decimos que “nos llevamos 1”; esto denomina **acarreo (carry)**



En el segundo dígito a sumar vuelve a ocurrir lo mismo y de nuevo “nos llevamos 1” y hay un acarreo.



Una forma mas simplificada de escribirlo seria, como vemos también en decimal “nos llevamos 1”, u ocurrió un acarreo

sTWBhHE6uEo **Suma de binarios**

Resta

¿Cómo se comporta la resta?

##1-1=0##
##0-0=0##
##1-0=1##
##0-1= - 1##

Como vemos si restamos un numero mayor queda un valor negativo, como estamos en un sistema de numeración, esto lo representamos con un signo **-** . En una computadora no seria posible, es decir la única forma de manejar la información, en una computadora es con 0 y 1, no tiene forma de representar símbolos, como el **-** , todo hay que llevarlo de alguna forma a 0 y 1. Pero ahora solo estamos viendo como se resuelve una resta en binario

Por otro lado podríamos tener la siguiente situación:

11_2 - 1_2 , si lo pensamos en decimal estamos diciendo 3-1 debería dar 2.

Como seria una situación similar en el sistema decimal

115
− 9
106

0
1~~1~~5
− 9
106

Que estamos haciendo?

En este caso el primer dígito a restar es mayor el sustraendo (**9**), en esa situación decíamos le “pido prestado al de al lado”, y de esa forma resolvíamos la operación.
Cuando trabajamos en binario es exactamente lo mismo hay que “pedir prestado”, el problema es que en binario solo hay 0 o 1, si le pido prestado a un 1, este se queda en 0, pero cuanto me esta “dando”?, me esta “dando” una base o sea un 2, de la misma forma que si le pido prestado en decimal, lo que me “presta” es un 10 como se ve en el ejemplo que me quedo 15-9

Veamos entonces un ejemplo en binario

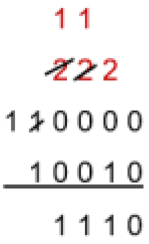
1 0 1 1
1 0
1 0 0 1

En este caso no hay dificultad (11-2=9)

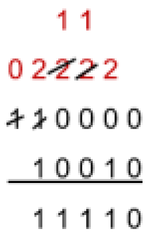
2
~~1~~ 0 1 0
1 0 0
0 1 1 0

Ocurrió que tuve que “prestar”, en un dígito (10-4=6)

Esta es una situación mas compleja (48-18=30)



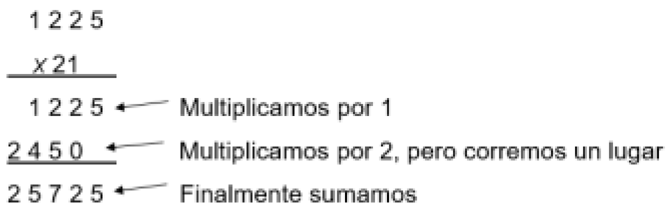
En la posición contigua a la que necesito hay un 0 por lo tanto es necesario pasar otra posición, esto se repite 2 veces como se ve en ejemplo, luego para los últimos dígitos vuelve a ocurrir que necesito "pedir"



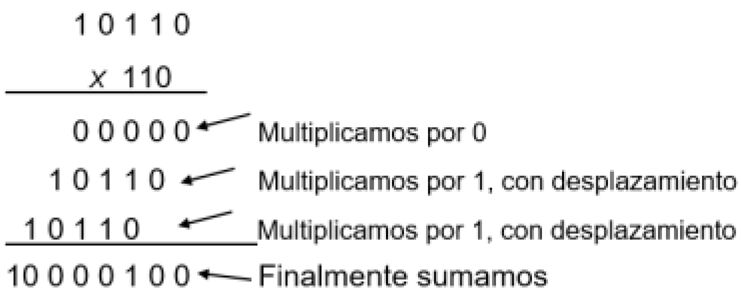
cnNInSjC6Gs **Resta de binarios**

Multiplicación

Como hicimos para la suma y la resta, recordemos como se resuelve una multiplicación en decimal.



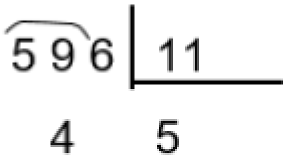
Este es el proceso que hacemos al multiplicar, en binario es mucho mas sencillo porque solo multiplicamos por 0 o por 1, pero no debemos olvidarnos desplazar un lugar por cada dígito que multiplicamos, no importa si es 0 o 1. Luego de multiplicar solo nos quedaría sumar. Entonces dado que la multiplicación por 0 por 1 es trivial, todo se reduce a la suma.



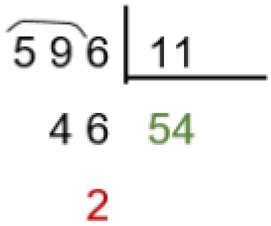
VtW2CWmvxio **Producto de binarios**

División

Para entender la división, debemos recordar la forma en que realizábamos nuestras primeras operaciones, y vamos a resolver una división entera, es decir no vamos a seguir dividiendo con decimales sino que nos vamos a detener y nuestro resultados van a ser un cociente y un resto.

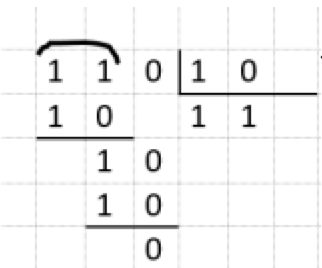


La forma de obtener el cociente, era tomar la porción del dividendo lo suficientemente grande para contener al divisor, la cantidad de dígitos que se toman inicialmente, entonces puede variar. Una vez obtenido el primer cociente, lo multiplicábamos por el divisor,en este caso seria 5×11 =55 y luego haríamos 59-55 de donde surge el 4

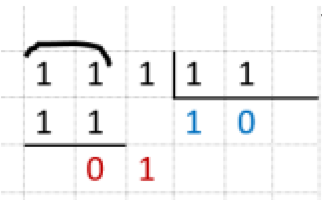


En este paso ya continuamos con la operación, y obtuvimos el cociente y resto. El cociente seria 54 y el resto 2

¿Cómo será esta operación en binario? Tengamos presente que los únicos dígitos posibles en el cociente son 1 y 0, de forma similar a la multiplicación el problema acá sera restar y no dividir. Veamos un caso simple.

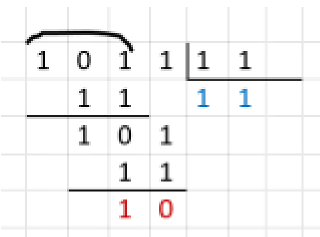


Tenemos 110_2 /10_2.
Tomamos 2 dígitos del dividendo, dado que es suficiente para contener al divisor 11_2>10_2, esto haría el primer dígito del cociente sea 1.
Entonces tenemos que restar 11_2 -10_2, esto da 1 luego "bajamos el 0", y nuevamente tendriamos cociente 1, le restamos 10_2, y nos queda resto 0.
El resultado de la operación es cociente 11_2 y resto 0_2. Sugerencia realizar la operación efectuando las restas en la división.



Veamos otro caso 111_2/11_2. El primer paso es similar al anterior, pero ahora si restamos 11_2 y 11_2 nos quedaría 0_2 y "bajamos el 1_2", pero 01_2 es menor que 11_2 o sea el cociente obtenido en este caso es 0, por eso el cociente nos queda 10_2 y el resto 1_2

Vemos un ultimo caso



En este caso tenemos 1011_2/11_2. Como se ve no alcanza con tomar solo 2 dígitos ya 10_2 es menor que 11_2, entonces debemos tomar 3 dígitos. Y procedemos nuevamente como en el anterior, al restar nos queda 10_2 y al bajar el ultimo dígito 101_2. El resultado final es cociente 11_2 y resto 10_2

Ql-Dl1_u4Zg **Cociente de binarios**

— *Martha*

Volver

(119)