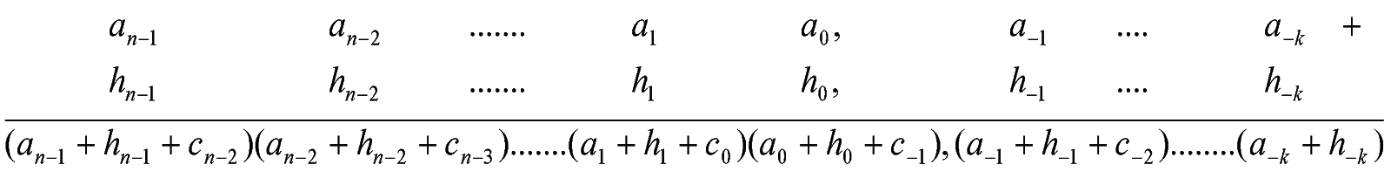
**SISTEMAS NUMÉRICOS**

**OPERACIONES ARITMÉTICAS DE LOS DISTINTOS SISTEMAS.**

Al igual que en el sistema decimal, también en otros sistemas de numeración, se pueden realizar operaciones aritméticas, tales como: suma, resta, multiplicación y división tomando como referencia la base del sistema dado.

**SUMA BINARIA, OCTAL Y HEXADECIMAL.**

En general, para realizar la suma se procede de la misma forma como se hace en el sistema decimal. Por ejemplo, si  https://lh4.googleusercontent.com/ttYtH_kqwatWC9OHqDnvx1WPVL77vbPbqSj9Kzxd0LOOdWNmT7-Yk4lBXq0FmNQ0_FCwebWLL93QEttZys0Wbj-pnyzbCChpHf6uu16lnfYu3KC5e2F58R9KV2JLT1UkV5Q4jjoSsJDQAqc65g es un número dado en una base ***b***  y https://lh5.googleusercontent.com/oIqEBnVlaEt7ozVmX711NSOJlDhYggJr1Mxr7_cVeAFj_c7jTh5z_BEqs22cxsnWJJ__bieSUNq25yNE80ilHEWzp0zaMkFMYEPg04pTe-bUecnrEdFpzc3z89AQNpCqb73JAJfOS6h4-3SuMQ es otro dado en la misma base entonces la suma se debe realizar de la siguiente forma:



Los dígitos ***mj=(aj+hj+cj-1)*** pertenecientes al resultado se forman sumando los dígitos de cada columna de los cosumandos, más el acarreo ***cj-1*** que viene de la columna anterior. Cada unidad de acarreo tiene el mismo valor de la base del sistema, por ejemplo, en la suma binaria es dos, en octal ocho y en hexadecimal dieciséis. Por ejemplo, llevar 2 en hexadecimal significa que el acarreo es el doble de la base y vale exactamente 32; de este mismo modo, en binario equivale a 4 veces y 16 en octal. Los acarreos aparecen cuando las semisumas de las columnas superan la base del sistema numérico.

**SUMA BINARIA:** Las operaciones de suma binaria se realizan de la siguiente forma:

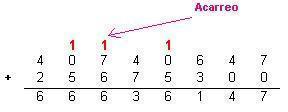
| 0 | + | 0 | = | 0 |  |
| --- | --- | --- | --- | --- | --- |
| 0 | + | 1 | = | 1 |  |
| 1 | + | 0 | = | 1 |  |
| 1 | + | 1 | = | 0 | Llevo 1 |

**Ejemplo:**  Dado los números binarios:  W=1111100012; T=11011101012;   Obtener W+T

| 0  + | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

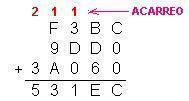
**SUMA OCTAL:** Se debe restar o dividir la semisuma de cada columna, cuando la misma exceda la base del sistema, y colocar en la columna inmediata del lado izquierdo, el valor del acarreo tantas veces se haya superado la base del sistema. De esta misma forma cada unidad que se acarrea equivale a ocho unidades de la columna anterior.

**Ejemplo:**  Dado los números binarios:  A. 40740647 y B. 25675300, Obtener A+B



**SUMA HEXADECIMAL:** Se debe restar o dividir la semisuma de cada columna, cuando la misma exceda la base del sistema, y colocar en la columna inmediata del lado izquierdo, el valor del acarreo tantas veces se haya superado la base del sistema. Cada unidad que se acarree equivale a dieciséis unidades de la columna anterior.

**Ejemplo:** Dado los números binarios:

****

**REPRESENTACIÓN NUMÉRICA EN COMPLEMENTO A DOS.**

En el sistema binario, la forma más utilizada para representar los números enteros con signo es la de complemento a dos. Los circuitos microprocesadores poseen internamente unidades de procesamiento aritmético que trabajan bajo este formato, el cual puede estar constituido por n bits múltiplos de la potencia de base dos. Por ejemplo, para representar los números positivos y negativos se definen datos con tamaño estándar: ocho bits, 16 bits, 32 bits, etc.

En este formato, el bit más significativo (MSB) del dato se utiliza para indicar el signo y los bits restantes representan la magnitud del número. En la figura 1.2 se puede apreciar la representación del formato utilizado para 16 bits, donde el más significativo (B15) indica que el signo es negativo si vale uno o positivo si vale cero. Las cantidades positivas se encuentran en binario normal mientras que los números negativos están en complemento a dos, esto significa que estos últimos, se deben complementar para poder hallar su verdadero valor.

**OPERACIONES ARITMÉTICAS EN COMPLEMENTO A DOS.**

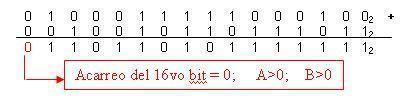
La suma y resta son las operaciones básicas realizadas por los microprocesadores, cualquiera otra operación, es consecuencia **recursiva** de éstas. A continuación se describen estas dos operaciones aritméticas, realizadas con números binarios en complemento a dos utilizando formato de signo y magnitud de 16 bits.

**SUMA EN COMPLEMENTO A DOS.**

Son cuatro casos que se presentan al sumar dos datos en formato con signo de complemento a dos:

**I)** **SUMA DE DOS NÚMEROS POSITIVOS.** El resultado debe ser positivo, y el bit más significativo de la suma, siempre dará cero.

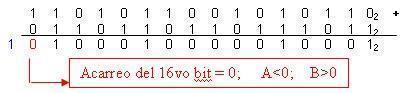
Ejemplo: A = 100011111000100**2**; B = 10010110111011**2.**

****

Antes de realizar la suma binaria se debe tener la precaución de sumar en decimal los números. De esta manera se puede chequear el resultado de la suma para tener la certeza de que no exceda el valor  +32767**10**  y por lo tanto no sobrepasar el formato de 16 bits (Esto se conoce como OVERFLOW). También el 16vo bit en uno señala el sobreflujo de la operación.

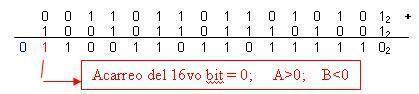
**II)** **SUMA DE UNO NEGATIVO Y OTRO POSITIVO.** El resultado debe poseer el signo del que tenga mayor valor absoluto. En este caso el resultado es positivo y el 16vo bit vale cero.

Ejemplo: A = 1101011001010110**2**; B = 110110110111011**2**

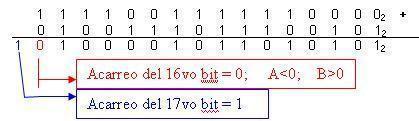
****

**III)** **SUMA DE UNO POSITIVO Y OTRO NEGATIVO.** El resultado debe poseer el signo del que tenga mayor valor absoluto. En este caso el resultado es negativo y el 16vo bit vale cero; del mismo modo no se debe tomar en cuenta el acarreo del 17vo bit.

Ejemplo: A = 11011011010101**2**; B = 1001011011101001**2**

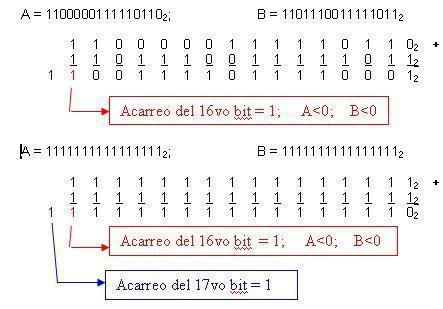
****

A = 1111001111110000**2**; B = 100111011100101**2**



Con dos números de distintos signos se dan los casos de acarreo en el 17vo bit. Si este acarreo es cero significa que el resultado es negativo y se debe complementar para hallar su verdadero valor de la otra forma, si el acarreo es uno, entonces el signo del resultado es mayor o igual a cero y se encuentra en verdadero valor.

**IV)** **SUMA DE DOS NÚMEROS NEGATIVOS.** El resultado debe ser negativo, por lo tanto el bit más significativo de la suma siempre dará uno.



Antes de realizar la suma binaria se debe tener la precaución de sumar en decimal los números. De esta manera se puede chequear el resultado de la suma para tener la certeza de que no exceda el valor -32767**10**  y por lo tanto no sobrepasar el formato de 16 bits (Esto se conoce como OVERFLOW). También el 16vo y/o 17vo bits en cero señalan el sobreflujo de la operación.

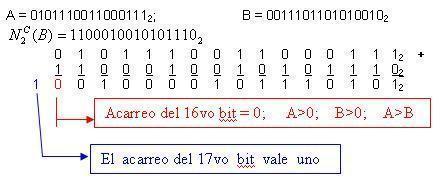
**RESTA EN COMPLEMENTO A DOS.**

La resta en complemento a dos resuelve el problema de esta operación con los signos. Por ejemplo, el sustraendo negativo y minuendo positivo produce un resultado positivo; la resta de dos números A y B negativos puede dar resultados positivos o negativos. Para realizarla se procede con la fórmula definida de la siguiente forma:

https://lh6.googleusercontent.com/Px4RvXJdG_HkKI-T5riwuGnEbJnhZ3P-EdO32XlKaIBvkDQI8xBJwbbGl53-qVgQxH5eIDXZK45dr5UMU8MGjxZ23WcfQ2Dxxk1wv1F-lx9a43rVkm2yOddlO3BWY_W-dongYlDGSp5tFCA8jg **(Ec.1.5)**; La diferencia de dos números, **A menos B** es equivalente a la suma de **A** más el complemento a dos de **B**.

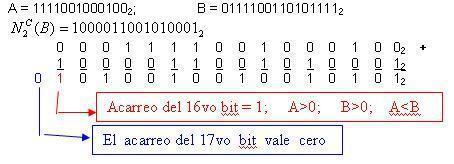
**I)** **Resta de dos números positivos.** El resultado puede presentar varias formas que se determinan con los siguientes casos:

**(A  mayor o igual que  B):**



De esta manera, el  resultado queda en forma binaria normal y es igual a valor del 17vo bit no se toma en cuenta para el resultado. En decimal A=23751**10** y  B=15186**10**; entonces  A-B=8565**10** = 0010000101110101**2**

**(A  menor que  B):**



De esta manera, el  resultado es negativo y queda en forma de complemento a dos, el acarreo del 17vo bit no se toma en cuenta. Sin embargo, para saber el verdadero valor, el resultado se debe complementar a dos. Este es un número binario negativo de 16 bits, lo cual tiene un valor de: https://lh4.googleusercontent.com/4xcwGfhGi3R0_8QjjuPxBp1ALPh2e5ztPMvyDPbPLi38BVGXGY8UIAuS6ox78TykJvPZniRXljZISa6C5_bRXJLBQ9Nc5r3w8xh-AtDWrH0rhaWnoZdjyNljoXfKZzkHUKVDTr1PowTHtVz8iA. En decimal la operación se efectúa: A = 7748**10**  y B = 31151**10**   entonces el resultado es   A-B = -23403**10**.

**II)** **RESTA DE DOS NÚMEROS NEGATIVOS Y DE DISTINTO SIGNO.** El resultado puede presentar varias formas que se determinan aplicando los mismos casos de la suma en formato de 16 bits.

**Tabla 1. 3. Resumen de las operaciones suma y resta binaria con los datos A y B, utilizando el formato de 16 bits.**

| ***Operación*** | ***Acarreo***  ***17vo bit*** | ***Acarreo 16vo bit*** | ***Resultado*** | ***Observaciones*** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **A+B**  A>0;  B>0 | 0 | 0 | Positivo en binario normal | Chequear para no exceder el formato de 16 bits. |
| **A+B**  A>0;  B<0  **(\*\*)** | 0 | 1 | Negativo en complemento a dos | Complementar los 16 bits para obtener el verdadero valor. |
| **A+B**  A<0;  B>0  **(\*\*)** | 1 | 0 | Positivo en binario normal | El  17vo bit no se toma en cuenta para el resultado. |
| **A+B**  A<0;  B<0 | 1 | 1 | Negativo en complemento a dos | Complementar los 16 bits para obtener el verdadero valor,  Chequear para no exceder el formato de 16 bits y el 17vo bit no se toma en cuenta. |
| **A-B**  A>0;  B>0  A>=B | 1 | 0 | Positivo en binario normal | El  17vo bit no se toma en cuenta para el resultado. |
| **A-B**  A>0;  B>0  A<B | 0 | 1 | Negativo en complemento a dos | Complementar los 16 bits para obtener el verdadero valor. |
| **A-B**  A>0;  B<0 | 0 | 0 | Positivo en binario normal | Chequear para no exceder el formato de 16 bits. |
| **A-B**  A<0;  B>0 | 1 | 1 | Negativo en complemento a dos | Complementar los 16 bits para obtener el verdadero valor,  Chequear para no exceder el formato de 16 bits y el 17vo bit no se toma en cuenta. |
| **A-B**  A<0;  B<0  **(\*\*)** | 0 | 1 | Negativo en complemento a dos o positivo normal | Complementar los 16 bits para obtener el verdadero valor o dejarlo igual. Todo depende de la magnitud de A  y B. |
|  |  |  | **(\*\*) Se producen resultados negativos o positivos dependiendo del mayor entre A y B.** |  |