1) Completar la siguiente tabla con las equivalencias numéricas correspondientes:

Binario	Decimal	Hexadecimal
1010000	80	50
1111000	**120**	78
111101	61	**3D**
1101	13	D
10010110	150	**96**
10100100101	1317	525
101111110011000001011010	12529754	**BF305A**

Métodos usados para el pasaje de sistemas:

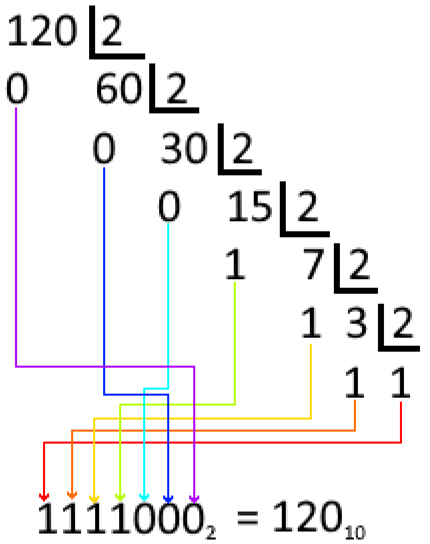
- Para pasar un número en binario o hexadecimal a decimal, usé el teorema fundamental de la numeración. Este dice que, en un sistema cíclico y posicional, un número es igual a la sumatoria de cada uno de sus dígitos multiplicados por la base del sistema elevada a la posición del dígito.
- Para pasar un número en binario a hexadecimal, tomé cada grupo de 4 dígitos binarios como un dígito hexadecimal. Esto se debe a que $2^4 = 16$, y cuando una base es potencia de otra, la cantidad de dígitos del exponente en el sistema de la base menor es igual a un dígito del sistema de la base mayor. Cuando la cantidad de bits no fuese múltiplo de 4, agregué ceros a la izquierda. Esto mismo se podría aplicar a otros dos sistemas cualquiera que cumplan con esas condiciones; por ejemplo, 3 dígitos binarios equivalen a un dígito octal ($2^3=8$).
- Para pasar un número en hexadecimal a binario, usé el método anterior pero invertido. Pasé cada dígito hexadecimal a 4 dígitos binarios.
- Para pasar un número en decimal a binario, dividí el número entre 2 continuamente, tomando los restos de las divisiones como los dígitos binarios. Repetí esto hasta obtener 1 como cociente de la división. Este proceso termina siendo lo mismo que ir sacándole al cociente la parte que sobra de las potencias de 2. Por ejemplo, si se toma un número decimal impar, al dividir por 2 se saca el 1 que sobra y este queda en el resto. Este mismo proceso se podría aplicar para pasar a cualquier otro sistema, siempre que se divida entre la base, y hasta que el cociente sea menor a la misma.

1010000_2:

$0*2^0 + 0*2^1 + 0*2^2 + 0*2^3 + 1*2^4 + 0*2^5 + 1*2^6 =$ $\underline{80_{10}}$

$0101_2\ (5_{16})\ 0000_2\ (0_{16}) =$ $\underline{50_{16}}$

120_{10}:



$120_{10} =$ $\underline{1111000_2}$

$0111_2\ (7_{16})\ 1000_2\ (8_{16}) =$ $\underline{78_{16}}$

$3D_{16}$:

$13*16^0 + 3*16^1 =$ $\underline{61_{10}}$

$3_{16}\ (0011_2)\ D_{16}\ (1101_2) =$ $\underline{111101_2}$

1101_2:

$1*2^0 + 0*2^1 + 1*2^2 + 1*2^3 =$ $\underline{13_{10}} = \underline{D_{16}}$

96_{16}:

$6*16^0 + 9*16^1 =$ $\underline{150_{10}}$

$9_{16}\ (1001_2)\ 6_{16}\ (0110_2) =$ $\underline{10010110_2}$

10100100101_2:

0101_2 (5_{16}) 0010_2 (2_{16}) 0101_2 (5_{16}) = 525_{16}

$5*16^0 + 2*16^1 + 5*16^2 = 1317_{10}$

$BF305A_{16}$:

$10*16^0 + 5*16^1 + 0*16^2 + 3*16^3 + 15*16^4 + 11*16^5 = 12529754_{10}$

B_{16} (1011_2) F_{16} (1111_2) 3_{16} (0011_2) 0_{16} (0000_2) 5_{16} (0101_2) A_{16} (1010_2) = $101111110011000001011010_2$

2) Realizar las siguientes sumas:

Sumar en binario y hexadecimal:

Para sumar en binario, el procedimiento es muy sencillo. Se suma cada dígito de ambos números, yendo de derecha a izquierda. Cuando el resultado es mayor que 1 (es decir, sumando 1+1) el resultado es 0 y se acarrea 1. Esto quiere decir que se suma 1 al dígito de la izquierda (uno más significativo). Este método es igual que en decimal y que en cualquier otra base, variando en cuándo se hace el acarreo en cada sistema, que es cuando el resultado de la suma de dos dígitos supera a la base.

$$\begin{array}{r} 1010_2 \\ +\ 0101_2 \\ \hline 1111_2 \end{array} \qquad \begin{array}{r} 1001_2 \\ +\ 0110_2 \\ \hline 1111_2 \end{array} \qquad \begin{array}{r} {\scriptstyle 1\,1\,1} \\ 1110_2 \\ +\ 1010_2 \\ \hline c\,1\,1000_2 \end{array}$$

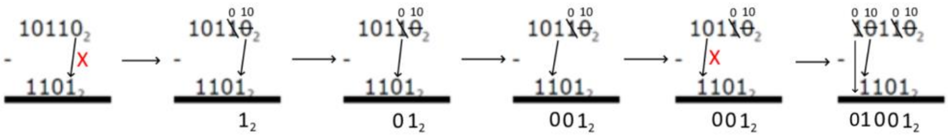
$$\begin{array}{r} {\scriptstyle 1\ \ 1} \\ 10110_2 \\ +\ 10101_2 \\ \hline c\,1\,01011_2 \end{array} \qquad \begin{array}{r} {\scriptstyle 1\,1\,1\,1} \\ 11011_2 \\ +\ 00110_2 \\ \hline c\,1\,00001_2 \end{array} \qquad \begin{array}{r} {\scriptstyle 1\ \ 1\,1} \\ 10010_2 \\ +\ 10110_2 \\ \hline c\,1\,01000_2 \end{array}$$

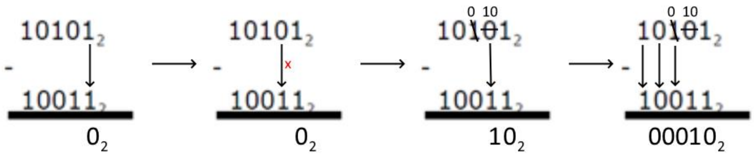
$$\begin{array}{r} 7354_{16} \\ +\ 1123_{16} \\ \hline 8477_{16} \end{array} \qquad \begin{array}{r} {\scriptstyle 1\ \ 1}\phantom{E5_{16}} \\ F1E5_{16} \\ +\ ABC1_{16} \\ \hline c\,1\,9DA6_{16} \end{array} \qquad \begin{array}{r} 3231_{16} \\ +\ 2123_{16} \\ \hline 5354_{16} \end{array}$$

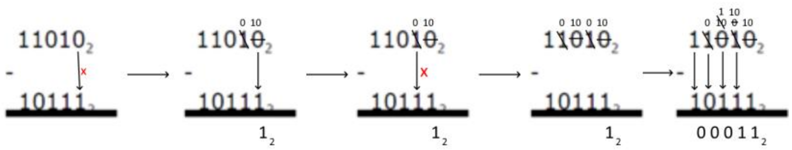
3) Realizar las siguientes restas

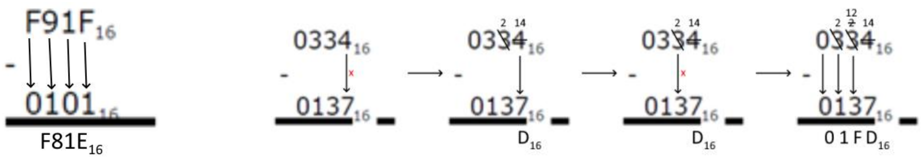
Restar en binario y hexadecimal:

Para restar en cualquier sistema de numeración, el procedimiento es casi igual. Yendo de derecha a izquierda (del dígito menos al más significativo), se va restando cada dígito con el de la misma posición en el otro número. Cuando el sustraendo es mayor que el minuendo tenemos que “pedirle” la base al dígito siguiente del minuendo; esto quiere decir que le vamos a restar uno a un dígito más significativo, y se le va a sumar la base al dígito actual al que le estamos restando. Por ejemplo, si tengo que hacer $32_{10} - 8_{10}$, como 8 es mayor que 2, le tengo que “pedir” la base (10) al 3, por lo que ese 3 pasa a ser un 2, y el 2 pasa a ser un 12. Ahora, como ya tengo un minuendo mayor que 8, puedo hacer la resta, y me da como resultado 4 para ese dígito (quedando 24 como resultado final). Para que esto se entienda mejor en las imágenes, los dígitos tachados en diagonal son los que cambian porque les “piden” la base (o sea, se les resta 1) y los que están tachados horizontalmente son los que cambian porque se les suma la base.









En esta última resta, el resultado es un número negativo. Pongo una F (15_{10}) porque se pide la base al dígito siguiente (y queda 10_{16}-1_{16}=F_{16}), pero como no hay dígito siguiente (son todos 0), se estaría pidiendo la base constantemente a 0. Como la palabra tiene 4 dígitos hexadecimales terminé obteniendo el resultado $F8EA_{16}$ que está en complemento a 2.

4) Utilizando una “palabra” de 3 bits de ancho, listar todos los números binarios signados y sus equivalencias decimales posibles representables en:

a) signo y magnitud b) Complemento a 1 c) Complemento a 2

Decodificación de palabras binarias:

- Para interpretar en decimal una palabra codificada en binario puro, solo se deben sumar sus dígitos multiplicados por su valor, que son las potencias de 2. El dígito menos significativo (de la derecha) vale 1, el próximo 2, 4 y 8.
- Para interpretar una palabra codificada en signo y magnitud, debemos reservar el dígito más significativo para el signo, si el de la izquierda es un 0, el signo es un +, y si es un 1, es un -. Luego, se lee el resto de la palabra en binario puro.
- Para interpretar una palabra codificada en complemento a 1, el primer dígito también indica el signo como en signo y magnitud. Para los números positivos (con 0 en el dígito de la izquierda), se lee normalmente el resto del número como en binario puro. Para los números con 1 en el dígito de la izquierda, se toma el resto de la palabra, se invierten todos sus dígitos (los 0 pasan a ser 1 y viceversa) y se escribe la traducción a binario puro de esa nueva palabra con un -.
- Para interpretar una palabra codificada en complemento a 2, el procedimiento es muy parecido al complemento a 1. Si el dígito más significativo es 0, se pone un + y se lee el resto de la palabra en binario puro. Si este dígito es un 1, se pone un -, se invierten todos los dígitos y a esa nueva palabra binaria se le suma 1 y se traduce a binario puro.

Al solo tener que decodificar palabras binarias de solo 3 y 4 bits, todos estos procedimientos los hice mentalmente.

Palabra	Binario Puro	Signo y Magnitud	Complemento a 1	Complemento a 2
000	0	+0	+0	+0
001	1	+1	+1	+1
010	2	+2	+2	+2
011	3	+3	+3	+3
100	4	-0	-3	-4
101	5	-1	-2	-3
110	6	-2	-1	-2
111	7	-3	-0	-1

5) Utilizando una “palabra” de 4 bits de ancho, listar todos los números binarios signados y sus equivalencias decimales posibles representables en:

a) signo y magnitud b) Complemento a 1 c) Complemento a 2

Palabra	Binario Puro	Signo y Magnitud	Complemento a 1	Complemento a 2
0000	0	+0	+0	+0
0001	1	+1	+1	+1
0010	2	+2	+2	+2
0011	3	+3	+3	+3
0100	4	+4	+4	+4
0101	5	+5	+5	+5
0110	6	+6	+6	+6
0111	7	+7	+7	+7
1000	8	-0	-7	-8
1001	9	-1	-6	-7
1010	10	-2	-5	-6
1011	11	-3	-4	-5
1100	12	-4	-3	-4
1101	13	-5	-2	-3
1110	14	-6	-1	-2
1111	15	-7	-0	-1