Capítulo 3

Como se comentó en los capítulos anteriores, hay convenios para representar número con signo. Hemos visto las representaciones Signo y Magnitud, por un lado y Complemento a 2 por otro lado.

Hay una tercera forma de representar denominada Binario Desplazado, veamos con cuatro bits las 16 combinaciones:

	No Signado	Binario Desplazado
1111	15	+7
1110	14	+6
1101	13	+5
1100	12	+4
1011	11	+3
1010	10	+2
1001	9	+1
1000	8	0
0111	7	-1
0110	6	-2
0101	5	-3
0100	4	-4
0011	3	-5
0010	2	-6
0001	1	-7
0000	0	-8

Esta representación mueve el eje del cero hasta la mitad de las combinaciones, por lo cual, se puede generalizar de la siguiente forma:

1. Para pasar de decimal a binario desplazado, se suma al decimal signado el valor 2^{n-1}, siendo n la cantidad de bits (que tiene que estar definida como en los casos anteriores).
   a. Ejemplo 1: se quiere el +5, con lo cual sumo $+5+ 2^{4-1}= +5+2^3 = +5+8 = 13$-> 1101 (en binario).
   b. Ejemplo 2: se quiere el -6, con lo cual sumo $-6+ 2^{4-1}= -6+2^3 = -6+8 = 2$-> 0010 (en binario).
2. Para pasar de binario desplazado a decimal, se toma el binario a decimal y se le resta 2^{n-1}.
   a. Ejemplo 1: se quiere 1100, con lo cual representa el 12, le resto 2^{n-1}, o sea $12-2^{4-1} = 12-8 = +4$
   b. Ejemplo 2: se quiere el 0010, con lo cual representa el 2, le resto 2^{n-1}, o sea $2-2^{4-1} = 2-8 = -6$.

El rango de representación es similar a Complemento a 2, o sea, desde $-(2^{n-1}) \leq N_{signado} \leq +(2^{n-1}-1)$

Estas son las formas más utilizadas para representar números signados…

Códigos Numéricos

Se pueden representar solo los números decimales, y para ello se necesitan cuatro bits, pero con cuatro bits, se tienen 16 combinaciones, seis más que la cantidad de símbolos decimales, y eso permite una gran variedad de representaciones, vamos a ver tres y sus propiedades…

La primera de ellas es BCD Natural (BCD = **B**inary **C**ode **D**igit)

2^3	2^2	2^1	2^0	
8	4	2	1	
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**2**
0	0	1	1	**3**
0	1	0	0	**4**
0	1	0	1	**5**
0	1	1	0	**6**
0	1	1	1	**7**
1	0	0	0	**8**
1	0	0	1	**9**
1	0	1	0	--
1	0	1	1	--
1	1	0	0	--
1	1	0	1	--
1	1	1	0	--
1	1	1	1	--

Esta representación se forma con los pesos o ponderación de cada columna binaria, por eso al BCD Natural se lo denomina Código Pesado o Ponderado.

Otra representación es el código Aiken, que toma las primeras cinco combinaciones y las cinco últimas.

2^1	2^2	2^1	2^0	
2	4	2	1	
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**2**
0	0	1	1	**3**
0	1	0	0	**4**
0	1	0	1	**--**
0	1	1	0	**--**
0	1	1	1	**--**
1	0	0	0	**--**
1	0	0	1	**--**
1	0	1	0	**--**
1	0	1	1	**5**
1	1	0	0	**6**
1	1	0	1	**7**
1	1	1	0	**8**
1	1	1	1	**9**

En este caso, este código también es ponderado, pero con pesos distintos al anterior, en este caso 2 4 2 1 (ver la tabla), pero en este caso también es auto complementario o complementario a 9, esto es, los decimales que entre si suman 9 (ejemplo 4 + 5) sus representaciones tienen invertidos los uno por ceros y los ceros por uno. 4 = 0100 y 5 = 1011. Este código es ponderado y auto complementario.

Por último, vamos a ver el código Exceso 3, este código es solo auto complementario, no tiene peso…

0	0	0	0	--
0	0	0	1	--
0	0	1	0	--
0	0	1	1	**0**
0	1	0	0	**1**
0	1	0	1	**2**
0	1	1	0	**3**
0	1	1	1	**4**
1	0	0	0	**5**
1	0	0	1	**6**
1	0	1	0	**7**
1	0	1	1	**8**
1	1	0	0	**9**
1	1	0	1	--
1	1	1	0	--
1	1	1	1	--

Vamos a trabajar con BCD Natural:

Se pueden hacer operaciones aritméticas con estos códigos…

Hay que tener en cuenta que cada número es representado por el código BCDNat

Ejemplo: 45 se representa como 0100 0101

Para sumar utilizaremos los mismos principios que vimos en binario:

1. Sumar 45 + 32

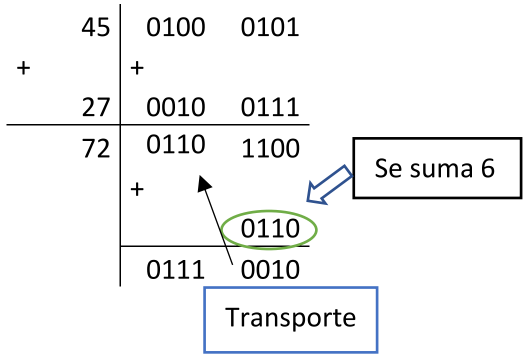
	45	0100	0101
+		+	
	32	0011	0010
	77	0111	0111

El resultado es correcto…

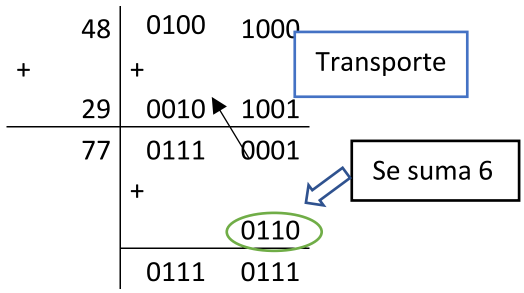
2. Sumar 45 + 27

	45	0100	0101
+		+	
	27	0010	0111
	72	0110	1100

El resultado no es el correcto, el segundo término es un número no BCD, pero se puede corregir sumando 6 a ese término…



Hay un último caso, sumamos 48 + 29



En caso de que haya un transporte del cuarto a quinto bit, también se suma seis…

Números fraccionarios.

Los números fraccionarios en el sistema decimal, tienen el siguiente formato:

Ejemplo:

14,25 se puede representar como

5 x	10^{-2}	5/100	0,05
2 x	10^{-1}	2/10	0,2
4 x	10^{0}	4	4
1 x	10^{1}	10	10
			14,25

Lo mismo ocurre cuando se trabaja en binario. Ejemplo 1110,01

1 x	2^{-2}	1/4	0,25
0 x	2^{-1}	0	0
0 x	2^{0}	0	0
1 x	2^{1}	2	2
1 x	2^{2}	4	4
1 x	2^{3}	8	8
			14,25

O sea, para pasar de binario fraccionario, a decimal es como anteriormente, multiplicado por las potencia de acuerdo con la posición.

Para pasar de decimal a binario, la parte entera se realiza como se había visto en el capitulo 1, y la parte fraccionaria se puede realizar de la siguiente forma:

Ejemplo 14,25

La parte entera 14 = 1110

La parte faccionaria se hace de la siguiente forma: se multiplica por dos, si el resultado incluye un valor entero mayor a cero, se escribe uno en la columna de la derecha, y se elimina ese entero y se sigue multiplicando.

0,25	0,	
0,50	0	
1,00 -> 0,00	1	Se resta el entero y queda 0,00
0,00	0	Acá se termina
		Se reescribe de arriba hacia abajo y queda 0,010

Sumando los términos, finalmente queda 1110 + 0,010 = 1110,010

Este es un ejemplo simple, que termina rápidamente, veamos que pasa si queremos pasar 14,23

Como antes 14 = 1110 y la parte fraccionaria se hace en forma similar

0,23	0,	
0,46	0	
0,92	0	
1,84 → 0,84	1	Se resta el entero y queda 0,84
1,68 → 0,68	1	
1,36 → 0,36	1	
0,72	0	
1,44 → 0,44	1	
0,88	0	
1,76 → 0,76	1	
1,52 → 0,52	1	
1,04 → 0,04	1	
0,08	0	…… Continua…

Como se ve puede continuar hasta el infinito, o hasta la cantidad de bits que se quieran utilizar… y esto trae errores, ese error se puede calcular, aunque no es parte de esta materia, sin embargo, podemos ver cual es el valor final del ejemplo con 12 bits.

0,	x 2^{0}	0,0
0	x 2^{-1}	0,0
0	x 2^{-2}	0,0
1	x 2^{-3}	0,125
1	x 2^{-4}	0,0625
1	x 2^{-5}	0,03125
0	x 2^{-6}	0,00
1	x 2^{-7}	0,0078125
0	x 2^{-8}	0,00
1	x 2^{-9}	0,001953125
1	x 2^{-10}	0,0009765625
1	x 2^{-11}	0,00048828125
0	x 2^{-12}	0,00
Total		0,22998046875

Debería ser 0,23… hay un error, a medida que se agreguen bits se reduce el error, pero no se llega al valor inicial

Ejercicios:

1. Representar los siguientes números base 10, en los distintos convenios de 8 bits

Num. Signado	Signo y Magnitud	Convenio a dos	Binario Desplazado
+ 104			
- 106			
+61			
-54			

2. Representar los siguientes números base 10, en los distintos convenios de 16 bits

Num. Signado	Signo y Magnitud	Convenio a dos	Binario Desplazado
+ 21104			
- 34106			
+361			
-4554			

3. Sumar números BCDNat

458596	
158675	

4. Pasar a binario fraccionario los siguientes números decimales (8 bits enteros y 12 bits fraccionarios).
   a. 41,45
   b. 58,69
   c. 47,36