Aritmética de punto fijo
Representación de números

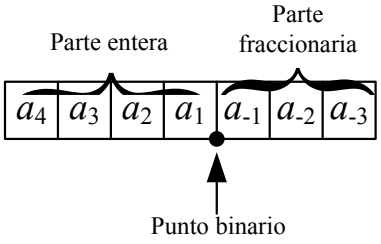
En general, un numero binario de *n* bits puede ser representado de la siguiente forma

$$N = a_{n-1}2^{n-1} + \cdots + a_1 2^1 + a_0 2^0 \quad (1)$$

Donde los coeficientes a_i, con $i = 0, 1, 2, \cdots$ son "0" o "1" y donde

$$\begin{cases} a_{n-1} : \text{MSB} \\ a_0 : \text{LSB} \end{cases}$$

Para representar números fraccionarios puede usarse una notación de punto fijo, donde un punto binario ficticio se usa para separar la parte entera de la parte fraccionaria.



$$N = a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0 + a_{-1} 2^{-1} + a_{-2} 2^{-2} + a_{-3} 2^{-3} \quad (2)$$

En los procesadores digitales de punto fijo actuales, se utiliza comúnmente el formato "Q_k" establecido en el standard IEEE 754 donde el subíndice k representa la cantidad de bits de la parte fraccionaria. En general un número den formato Q puede escribirse de la siguiente forma: Ejemplo con palabra de 16 bits

$$N_Q = b_{15-k} 2^{15-k} + \cdots \quad \cdots \quad (3)$$

En general:

$$N_Q = b_{(n-1)-k} 2^{(n-1)-k} + \cdots \quad \cdots \quad (4)$$

Por ejemplo, el número binario de 8 bits 01101.011 equivale al siguiente número decimal

$$\begin{aligned} N &= a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0 + a_{-1} 2^{-1} + a_{-2} 2^{-2} + a_{-3} 2^{-3} \\ &= 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 1/2 + 1 \cdot 1/4 + 1 \cdot 1/8 \\ &= 8 + 4 + 1 + 1/4 + 1/8 \\ &= 13.375 \end{aligned}$$

Hasta ahora no consideramos números negativos. Los procesadores digitales utilizan el bit MSB de la palabra de *n* bits para representar el signo:

si MSB=0 $\rightarrow$ Nº (+)

si MSB=1 $\rightarrow$ Nº (-)

Otra forma de representar número negativos es por el complemento a 2. Por ejemplo el número 120 en binario es:

	0	1 1 1 1 0 0 0	(120)
Complemento $\rightarrow$	1	0 0 0 0 1 1 1	
		+1	
	1	0 0 0 1 0 0 0	(-120)

Formato Q: Usando el MSB como bit de signo, con un procesador digital de tamaño de palabra de 16 bits, un número fraccionario representado en Q_{12} tendrá:

12 bits dedicados a la parte fraccionaria

4 bits dedicados a la parte entera

De los 4 bits de la parte entera el MSB se usa para el signo. Esto implica que los números representados en este formato deben estar en un intervalo $[-2^3 ; 2^3] = \pm 8$ tanto en régimen permanente como en régimen transitorio. Por otro lado, la presición o resolución que ofrece este formato "Q_{12}" es de $2^{-12} = 1/4096 = 0.00024414$. Esto significa que una tensión de 380 V con un valor base de 550V, dentro del procesador digital estará representado por

$$\frac{380}{550} = 0.69 \text{ p.u.}$$

con una presición de

$$380 \times \frac{1}{4096} = \pm 0.0927 \text{ V} \quad \text{ó} \quad \pm 92.77 \text{ mV}$$

Ejemplo: Nº $\pi = 3.141592654$

Para representar con buena presición al Nº π puedo usar el formato Q_{13}:

12 bits dedicados a la parte fraccionaria

3 bits dedicados a la parte entera

Rango dinámico: $\pm 2^2 = \pm 4$

Presición: $\frac{1}{2^{13}} = \frac{1}{8192} = 0.00012207$

El formato Q que ofrece la mejor precisión es el Q_{15}:

15 bits parte fraccionaria, resolución $\frac{1}{2^{15}} = 0.000030517$ y 1 bit parte entera. Ósea solo pueden ser representados números entre ± 1.

En general, un número representado por una palabra de "*n*" bits con un bit de señal y "*k*" bits de parte fraccionaria debe encontrase en el intervalo dado por:

$$\left[-\left(2^{n-1}-1\right)2^{-k}\,;\left(2^{n-1}-1\right)2^{-k}\right]$$

donde la presición del número es dada por 2^{-k}. Ósea si queremos representar números en Q_{14}, los mismos deben encontrarse o variar como máximo entre (palabra de 16 bits):

$$\left[-\left(2^{16-1}-1\right)2^{-14}\,;\left(2^{16-1}-1\right)2^{-14}\right]$$

$$\left[-1.99993\,;1.99993\right]$$

En el caso de Q_{12}, habíamos dicho que era entre ± 8; pero seria adecuado el intervalo:

$$\left[-\left(2^{16-1}-1\right)2^{-12}\,;\left(2^{16-1}-1\right)2^{-12}\right]$$

$$\left[-7.99975\,;7.99975\right]$$

En un procesador digital para representar un número entero o fraccionario en un formato Q_k determinado, se multiplica el número por 2^k. Por ejemplo:

$$\pi \text{ en } Q_{13} \rightarrow 3.141592654 \times 2^{13} = 25735.92$$
$$\cong 25736$$

Otro ejemplo:

$$0.00125 \text{ en } Q_{15} \rightarrow 0.00125 \times 2^{15} = 40.96$$
$$\cong 41$$

Operaciones aritméticas

La adición y sustracción deben ser realizadas en el mismo formato. Por ejemplo la suma o resta de dos números en Q_{10} resulta en un número también en Q_{10}.

Multiplicación:

Para la multiplicación vale la misma regla que para los números en base decimal, ósea:

$$10^3 \times 10^5 = 10^8$$

Un número en formato Q_k (2^k) por un número en formato Q_p (2^p) resulta en un número en formato $Q_{(k+p)}$ ($2^{(k+p)}$). Por ejemplo, la multiplicación de 2 números en formato Q_{12} da un número en Q_{24}. Esto significa que el punto decimal virtual es desplazado a la

izquierda en el acumulador de 32 bits y ahora, los 24 bits menos significativos de este acumulador representan la parte fraccionaria del resultado.

Ejemplo:

$$x = -1.125 \rightarrow \text{en } Q_{12} = 4608$$
$$y = 1.375 \rightarrow \text{en } Q_{12} = 5632$$
$$N_Q = 4608 \times 5632 = 25952256 \ \ (Q_{24})$$
$$\frac{25952256}{2^{24}} = -1.546875$$

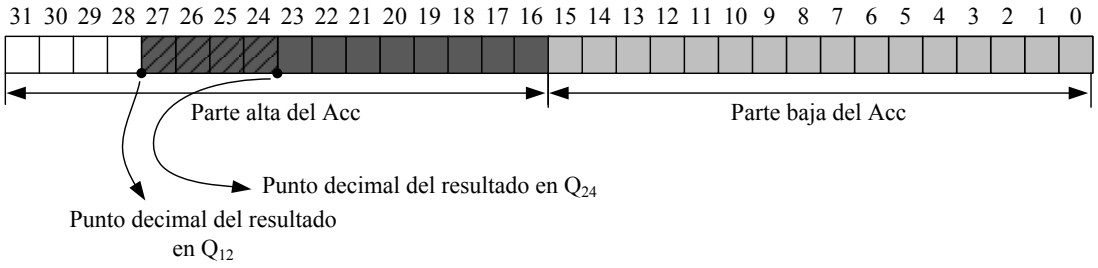
El resultado en Q_{24} en el acumulador debe guardarse en una posición de memoria en "Q_{12}". Para esto se desplaza (shift) hacia la izquierda el contenido del cumulador 4 veces y se almacena la parte alta del acumulador.

N_Q tiene 24 bits (Acc=32 bits)

24 – 16 bits parte baja=8 bits

8 bits + 4(shift) = 12

Shift de 4 bits a la izquierda equivale a multiplicar por $2^4 = 16$



Para procesadores de palabra de 16 bits el máximo resultado se da con 2 números en Q_{15}, ósea Q_{30} (30 bits de la parte fraccionaria).

Los procesadores digitales poseen un bit en un registro de la CPU denominado: Sign Extensión mode que permite propagar el signo del resultado a los 2 bits (MSB) mas significativos del acumulador y poder trabajar con valores positivos y negativos.