En las matemáticas puras un valor no tiene un límite de espacio para su representación, sin embargo, las computadoras generalmente trabajan con un número fijo de bits.

Contenido

[ocultar]

Bit [editar]

Artículo principal: Bit

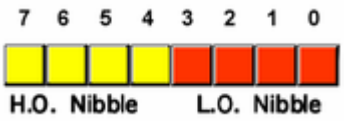
La unidad más pequeña de información en una computadora se le llama bit. Con un bit se puede representar solo un valor de dos posibles valores diferentes, ejemplo: cero o uno, falso o verdadero, rojo o azul, 56 o 2458, etc.

Nibble [editar]

Artículo principal: Nibble

Un **nibble** es una colección de 4 bits. No sería un tipo de dato interesante a excepción de que con un nibble se presenta un número BCD y también que un nibble puede representar un dígito hexadecimal.

Byte [editar]



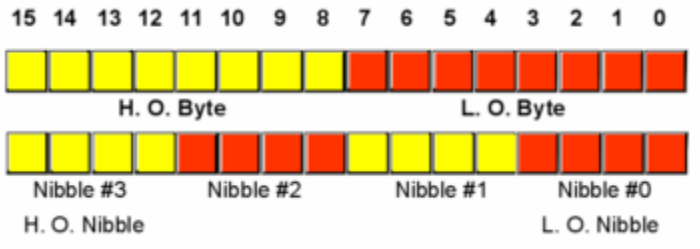
Nibbles de un byte.
Artículo principal: Byte

Sin lugar a dudas el tipo de datos más importante para los microprocesadores es el byte. Un byte consiste de 8 bits. Las referencias hacia cierta localidad de memoria en todos los microprocesadores nunca es menor que un byte, (la mayoría usan múltiplos de bytes), por lo tanto, se considera el dato localizable más pequeño.

Los bits de un byte normalmente se numeran desde 0 hasta 7. El bit 0 se le llama bit de más bajo orden o menos significante, el bit 7 se considera el bit de más alto orden o el más significante.

Un byte consta también de 2 nibbles, los bits 0, 1, 2 y 3 forman el llamado nibble de menor orden, y los bits 4, 5, 6 y 7 forman el nibble de mayor orden. Como un byte está formado de exactamente dos nibbles, es posible representar cualquier valor con dos dígitos hexadecimales.

Palabra [editar]



Nibbles y bytes de una palabra.
Artículo principal: Palabra (informática)

Una **palabra** es un grupo de 16 bits, el bit 0 es el bit de más bajo orden y el bit 15 es el de más alto orden. Una palabra se puede dividir en 2 bytes llamados igualmente de bajo y alto orden. También una palabra puede considerarse como un grupo de 4 nibbles.

Se considera una **palabra doble** a un grupo de 32 bits. Un grupo de mayor número de bits simplemente se nombra por su número de bits, ejemplo: palabra de 64 bits, palabra de 128 bits, etc.

Números enteros [editar]

Con un número fijo de bits podemos representar cierto número de objetos. Por ejemplo, con 8 bits podemos representar 256 objetos diferentes. Si se usara un esquema de números enteros positivos cada uno de éstos objetos se numerarían de 0 a 255:

Bits	Número decimal
00000000	0
00000001	1
00000010	2
00000011	3

00000100	4
...	...
11111110	254
11111111	255

También es posible usar un esquema de números enteros negativos, en dado caso se usa el sistema complemento a dos, donde el bit de mayor orden es el bit de signo, si tal bit es cero, el número es positivo, si es uno, el número es negativo. Si el número es positivo es almacenado en su valor binario estándar, si el número es negativo se almacena en su forma complemento a dos. Ejemplos:

Bits	**Número decimal**	**Bits**	**Número decimal**
00000000	0		
00000001	1	11111111	-1
00000010	2	11111110	-2
00000011	3	11111101	-3
00000100	4	11111100	-4
...	...	...	...
01111110	126	10000010	-126
01111111	127	10000001	-127
		10000000	-128

Números coma flotante [editar]

Representación binaria de números en coma flotante de doble precisión.
Artículo principal: Coma flotante

La forma en que la arquitectura de computadoras resuelve el problema de representar números reales es por medio de los números de coma flotante. Un número coma flotante se divide en 3 secciones de bits: **signo**, **mantisa** y **exponente con signo**.

Ejemplo de coma flotante de 8 bits							
b7	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
$\pm$	$\pm$	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}

Este ejemplo consta de un entero flotante hipotético de 8 bits donde el bit 7 corresponde al signo del número, el bit 6 al signo del exponente, los bits 5 y 4 al exponente y los bits 3,2,1 y 0 a la mantisa. Ejemplos de números para este caso serían:

$$01111010 = (1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4}) \times 2^{-3}$$
$$10011011 = -(1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}) \times 2^{1}$$

- Con los números punto flotante hay un rango limitado para representar cantidades, emplear números fuera del rango resultará en overflow o en underflow.
- Hay un número finito de números reales que puede ser representado dentro del rango.
- La mantisa se normaliza.
- La forma más común de usar puntos flotantes es como lo dicta el IEEE 754.

Obtenido de "http://es.wikipedia.org/wiki/Tipos_de_datos_m%C3%A1quina"