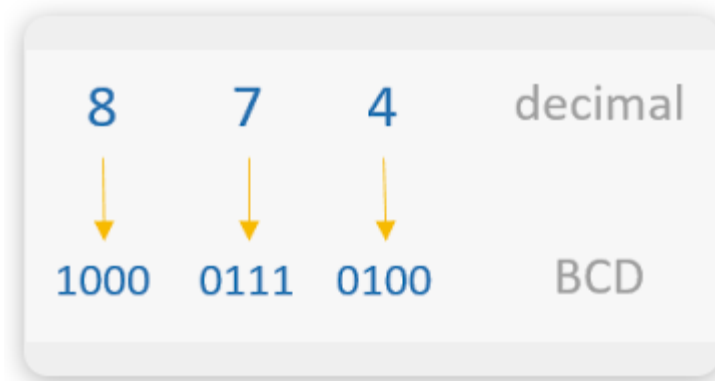


Decimal Empaquetado

Fuentes:

W. Stallings - Computer Organization and Architecture, 7° Edition, pág. 357

75039557 - Apunte de Sistemas de Numeración, pág. 16



Fuente de la imagen: <https://electronicaradical.blogspot.com/2014/08/codigo-bcd-decimal-codificado-en-binario.html>

Números

Todos los lenguajes máquina incluyen tipos de datos numéricos. Incluso en el procesamiento de datos no numéricos se necesitan números que actúen como contadores, longitudes de campos, etc. Una distinción importante entre los números utilizados en las matemáticas ordinarias y los almacenados en un computador es que estos últimos están limitados. Esto es cierto en dos sentidos. En primer lugar, hay un límite para la magnitud de los números representables en una máquina y, en segundo lugar, en el caso de números en coma flotante, su precisión está limitada. Por tanto, el programador debe ser consciente de las consecuencias del redondeo, el desbordamiento o el desbordamiento a cero.

En los computadores son usuales tres tipos de datos numéricos:

- Números enteros o en coma fija.
- Números en coma flotante.
- Números decimales.

Números decimales

Aunque todas las operaciones internas del computador son en esencia binarias, los usuarios del sistema utilizamos números decimales. Es necesario, pues, convertir de decimal a binario las entradas y de binario a decimal las salidas. Para aplicaciones en las que hay muchas entradas/salidas frente a pocos cálculos, y cálculos comparativamente simples, es preferible memorizar y operar con los números directamente en su forma decimal. La representación más común para ello es la de decimal empaquetado.

Los libros de texto la suelen denominar a decimal empaquetado como «decimal codificado en binario» (BCD). Hablando con propiedad:

BCD se refiere a la codificación de cada dígito decimal mediante una secuencia única de cuatro bits. Decimal empaquetado hace referencia al almacenamiento de dígitos codificados en BCD empleando un byte para cada pareja de dígitos.

En decimal empaquetado, cada dígito decimal se representa mediante un código de cuatro bits, según la manera habitual. Es decir, 0 0000, 1 0001, ..., 8 0100, y 9 1001. Observe que resulta un código bastante ineficiente ya que solo se emplean diez de las 16 posibles combinaciones de cuatro bits. Para construir números, se van encadenando códigos de 4 bits, normalmente formando múltiplos de 8 bits. Así, el código de 246 es 0000 0010 0100 0110. Este código es claramente menos compacto que la representación binaria directa 11110110, pero evita la necesidad de conversiones. Los números negativos pueden representarse incluyendo un dígito de signo de cuatro bits bien a la izquierda o a la derecha de la cadena de dígitos decimales empaquetados. Por ejemplo, el código 1111 podría representar el signo menos.

Muchas máquinas tienen instrucciones aritméticas destinadas a operar directamente con números en decimal empaquetado.

Los algoritmos correspondientes son bastante similares a los utilizados para enteros, coma fija y coma flotante pero deben tener en cuenta la operación de acarreo decimal.

BCD Empaquetado

Base = 16 Precisión = n Enteros positivos y negativos

Como almacenar un número

- 1) Pasar el nro a base 10
- 2) Colocar c/dígito en los nibbles dejando libre el último (el de la derecha)
- 3) Colocar en el último nibble el signo siendo
C, A, F o E para positivos
B o D para negativos

Ej. n=3 +123₁₀ --> 00123A₁₆ -456₁₀ --> 00456B₁₆

Como recuperar un número almacenado

- 1) Tomar cada dígito de los nibbles (excepto el último) para armar la cadena en base 10
- 2) Colocar el signo según el dígito del último nibble

Un Byte

4 bits 4 bits

Nibble Zone Nibble Numeric

Formato decimal empaquetado a partir de decimal

1. Colocar cada dígito decimal en un nibble, dejando libre el nibble menos significativo ya que en el mismo se almacena el signo.
2. Colocar en el nibble menos significativo el signo según:
 - C,A,F,E indican positivo.
 - B,D indican negativo.

Recordar que los símbolos en hexadecimal van de 0 a F, mientras que en BCD van de 0 a 9. Los símbolos desde A hasta F de hexadecimal se utilizan para representar el signo.

Ejemplo:

Formatear el número decimal -127 [10] como decimal empaquetado de 3 bytes.

0	0	1	2	7	D
---	---	---	---	---	---

Ejemplos de formato empaquetado

Ejemplo 1:

Dado A = 561196 que representa la configuración decimal de un número almacenado en formato empaquetado de 3 bytes, indicar qué número es en base 10.

$$A = 561196 \text{ [10] Empaq.}$$

1. Como el empaquetado es BCD (Decimal codificado en binario), con cada dígito en un nibble, lo convertimos a base 16:

$$A = 561196 \text{ [10] Empaq.} = 8902C \text{ [16] Empaq.}$$

2. Cómo BCD tiene los dígitos decimales 0 a 9, y el empaquetamiento pone en el nibble menos significativo el signo, interpretamos en base 10 al número 8902C [16]:

$$A = + 8902 \text{ [10]}$$

Ejemplo 2:

B = 2013868923 es la configuración decimal de un número B empaquetado de 4 bytes. Indicar qué número es en decimal.

1. Para ver qué dígitos BCD tiene en cada nibble del empaquetado, lo convierto a hexadecimal, pero el número que va a aparecer en cada nibble va a ser un número decimal (BCD):

$$B = 2013868923 \text{ [10] Empaq.} = 7809337B \text{ [16] Empaq.}$$

2. Como en el empaquetado el nibble menos significativo es el signo, interpretamos al número decimal como:

$$B = -7809337 \text{ [10]}$$

Ejemplo 3:

B = 2013868923 es la configuración decimal de un número B empaquetado de 4 bytes. Indicar qué número es en:

- a. Configuración de binario de punto fijo con signo de 32 bits.
- b. Configuración hexadecimal de B representado como binario de punto fijo con signo de 32 bits.

a.

1. Para ver qué dígitos BCD tiene en cada nibble del empaquetado, lo convierto a hexadecimal, pero el número que va a aparecer en cada nibble va a ser un número decimal (BCD):

$$B = 2013868923 \text{ [10] Empaq.} = 7809337B \text{ [16] Empaq.}$$

2. Como en el empaquetado el nibble menos significativo es el signo, interpretamos al número decimal como:

$$B = -7809337 \text{ [10]}$$

3. Para interpretarlo como binario de punto fijo, lo convertimos a binario:

$$B = -0111\ 0111\ 0010\ 1001\ 1100\ 1001 \text{ [2]}$$

4. Para incluir el signo en el número hallamos el complemento a la base:

$$B = 1000\ 1000\ 1101\ 0110\ 1100\ 0111 \text{ [2]}$$

5. Como son 3 bytes (6 nibbles) y pide 4, extendemos el signo:

$$B = 1111\ 1111\ 1000\ 1000\ 1101\ 0110\ 1100\ 0111 \text{ [2]}$$

binario de punto fijo con signo de 32 bits.

- b. Para la configuración hexadecimal de B representado como binario de punto fijo con signo de 32 bits, lo convertimos en hexadecimal:

$$B = FF\ 88\ D6\ C7 \text{ [16]}$$

Es la configuración hexadecimal del número decimal B = 2013868923 empaquetado de 4 bytes interpretado como binario de punto fijo con signo de 32 bits.

Ejemplo 4:

Dado $A = 2013868923$ en configuración decimal de un número empaquetado de 4 bytes, expresarlo en configuración octal representado como binario de punto flotante IEEE 754 de precisión simple.

1. Para ver qué dígitos BCD tiene en cada nibble del empaquetado, lo convierto a hexadecimal, pero el número que va a aparecer en cada nibble va a ser un número decimal (BCD):

$$A = 2013868923 \text{ [10] Empaq.} = 7809337B \text{ [16] Empaq.}$$

2. Como en el empaquetado el nibble menos significativo es el signo, interpretamos al número decimal como:

$$A = -7809337 \text{ [10]}$$

3. Para representar a A en punto flotante:

- a. Lo pasamos a binario:

$$A = -11101110010100100111001$$

- b. Corremos la coma hacia la izquierda 22 lugares hasta que quede como parte entera el "1" del bit implícito y aumentamos el exponente de la base 2 en la misma cantidad (22):

$$A = 1,1101110010100100111001 \times 2^{22}$$

- c. Tomamos como mantisa M desde el primer bit después de la coma en adelante y completamos con 23 bits:

$$M = 11011100101001001110010$$

- d. Expresamos el exponente E en código Exceso 127 y lo convertimos a binario:

$$E = 22 + 127 = 149 \text{ [10]} = 10010101 \text{ [2]}$$

- e. Armamos el número IEEE754 poniendo signo S , exponente E y mantisa M en 1, 8 y 23 bits respectivamente en la grilla SEM:

$$A = 1 \ 10010101 \ 11011100101001001110010$$

Es la representación IEEE 754 del decimal 2013868923 empaquetado en 4 bytes.

4. Para expresarlo en configuración octal lo convertimos a base 8:

$$A = 31273451162 \text{ [8]}$$

Es la representación como binario de punto flotante IEEE 754 de precisión simple configurado como octal del decimal 2013868923 empaquetado en 4 bytes.