

Circuitos y Sistemas Digitales

Andrés Prieto-Moreno Torres

19th November 2002

1 Representación de números enteros

Una de las formas para representar números enteros dentro del ordenador es almacenar el valor del número entero en base 2 en una posición de memoria, la cual tiene reservado el bit de la izquierda (el de mayor peso) para indicar el signo del número en cuestión. El criterio adoptado para el signo de los números es:

Número positivo $\rightarrow$ bit de signo 0

Número negativo $\rightarrow$ bit de signo 1

Existen tres métodos para representar el número dentro del ordenador.

1.1 Representación con signo-magnitud

Es la misma representación que la utilizada para los números positivos excepto la inclusión de un bit de signo que indica si el número es positivo (0) o negativo (1). el rango de representación con n bits es:

$$[-(2^{n-1} - 1), (2^{n-1} - 1)]$$

Los inconvenientes de este sistema son que hay dos representaciones para el cero 0:00...0 0:10...0

1.2 Representación en complemento a 1 (Ca 1)

Los números positivos se representan como en el caso de signo-magnitud. Los negativos como el complementario a 1 del entero positivo de la misma magnitud. El complemento a 1 de un número se obtiene sustituyendo los 0's por 1's y viceversa. El rango de representación con n bits es :

$$[-(2^{n-1} - 1), (2^{n-1} - 1)]$$

En los números positivos el primer bit es un 0 y en los negativos es un 1. Entre los inconvenientes es que seguimos teniendo dos representaciones del 0: 000...00 y 0: 111...11

1.3 Representación en complemento a 2 (Ca 2)

Los números positivos se representan como en el caso de signo-magnitud. Los números negativos como el complemento a 2 del entero positivo de la misma magnitud. El complemento a dos de un número se obtiene sumando 1 a su Ca 1. El rango de representación con n bits es:

$$[(-2^{n-1}), (2^{n-1} - 1)]$$

Otra propiedad que se cumple es $\text{Ca2}(N) = 2^n - N$

2 Aritmética de los complementos

La utilidad de los complementos es permitir que las máquinas realicen operaciones aritméticas a partir de sumadores. Es decir realizar multiplicaciones, restas y divisiones empleando sólo sumadores y conversiones a Ca1 o Ca2. Veamos un ejemplo de resta utilizando el Ca1.

$A - B = R$ se efectúa como $A + (-B) = A + \overline{B}$ y se pueden dar dos posibilidades:

- $A > B$, por lo tanto $R > 0$, y el resultado es $R = A + \overline{B} + 1$
- $A < B$, por lo tanto $R < 0$, entonces el resultado es $|R| = \overline{A + \overline{B}}$

Por ejemplo, calculemos $A - B$ considerando que $A = 55$, $B = 34$. Lo primero que nos damos cuenta es que $A > B$, por lo que el resultado tendrá que ser positivo.

$$A = 55_{(10)} = 110111_{(2)}$$

$$B = 34_{(10)} = 100010_{(2)} \implies \overline{B} = 011101$$

$$\begin{array}{r} A \quad 110111 \\ \overline{B} \quad 011101 \\ + 1 \quad \overline{010100} \end{array}$$

El 1 situado a la izquierda es el bit de *overflow*, este bit es el que se suma al resultado para obtener el valor de la resta. Se dice que se produce *overflow* cuando el número de bits resultante es mayor que el de los operandos.