

Codificación de Números Negativos

Sistemas de Procesamiento de Datos

UTN-FRA

Técnico Superior en Programación

2017



Formas de Representar Números Negativos

Dentro de un procesador, microprocesador, memoria o cualquier otro medio de almacenamiento de datos digitales que utiliza el sistema binario debemos encontrar alguna forma de representar números negativos. Existen varios códigos que pueden utilizarse a tal fin:

- Signo magnitud o bit de signo.
- Complemento a 1
- Complemento a 2

Signo magnitud o bit de signo

Se utiliza un bit para representar el signo. Este bit, normalmente es el bit más significativo y, por convención: un 0 denota un número positivo, y un 1 denota un número negativo.

Binario de 8 Bits

10000001

Sin Signo

129

Con signo

-1

Signo magnitud o bit de signo

Ventajas

- Simple de entender

Desventajas

- El número cero posee dos representaciones
- Pierde la virtud de permitir hacer operaciones aritméticas

Complemento a 1

Se trata de representar números negativos usando el complemento a la base menos uno. En el caso de los binarios, cuya base es 2, se trata de realizar el complemento a uno y su forma, es hacer un “NOT” (invertir bit a bit) al número, es decir, la inversión de unos por ceros y ceros por unos.

Binario de 8 Bits

10000001

Sin Signo

129

Con signo

-126

Complemento a 1

Ventajas

- Simple de aplicar (NOT)

Desventajas

- El número cero posee dos representaciones
- Pierde la virtud de permitir hacer operaciones aritméticas

Complemento a 2

Es representación que utilizan los microprocesadores y microcontroladores en la actualidad. Sin entrar en una definición académica rigurosa, diremos que el complemento a dos de un número cualquiera se obtiene de sumarle "1" al número convertido en complemento a 1.

Binario de 8 Bits

10000001

Sin Signo

129

Con signo

-127

Complemento a 2

Ventajas

- Simple de aplicar ($\text{NOT} + 1$)
- Permite hacer operaciones aritméticas
- El cero posee una representación única al igual que todos los números representables.

Desventajas

- -

Operaciones en Complemento a 2

Suma

Cuando se suman dos números binarios con signo pueden producirse cuatro casos:

1. Ambos números son positivos.
2. El número positivo es mayor que el negativo en valor absoluto.
3. El número negativo es mayor que el positivo en valor absoluto.
4. Ambos números son negativos.

Operaciones en Complemento a 2

Caso 1

Ambos números son positivos.

$$\begin{array}{r} + \quad 00000111 \quad 7 \\ \quad 00000100 \quad 4 \\ \hline \quad 00001011 \quad 11 \end{array}$$

- La suma es positiva y, por tanto, es un número binario real (no complementado).

Operaciones en Complemento a 2

Caso 2

El número positivo es mayor que el negativo en valor absoluto.

$$\begin{array}{r} 00001111 \quad 15 \\ + \quad 11111010 \quad -6 \\ \hline 100001001 \quad 9 \end{array}$$

- El bit de acarreo final no se tiene en cuenta. La suma es positiva y, por tanto es un número binario real (no complementado).

Operaciones en Complemento a 2

Caso 3

El número negativo es mayor que el número positivo en valor absoluto:

$$\begin{array}{r} 00010000 \quad 16 \\ + \quad 11101000 \quad -24 \\ \hline 11111000 \quad -8 \end{array}$$

-La suma es negativa y, por tanto, está en complemento a 2.

Operaciones en Complemento a 2

Caso 4

Ambos números son negativos:

$$\begin{array}{r} + 11111011 \quad -5 \\ 11110111 \quad -9 \\ \hline 111110010 \quad -14 \end{array}$$

-El bit de acarreo final no se tiene en cuenta. La suma es negativa y, por tanto, está en complemento a 2.

Operaciones en Complemento a 2

- **Condición de desbordamiento (overflow)**

Cuando se suman dos números y el número de bits requerido para representar la suma excede al número de bits de los dos números, se produce un desbordamiento que se indica mediante un bit de signo incorrecto.

Un desbordamiento se puede producir sólo cuando ambos números son positivos o negativos.