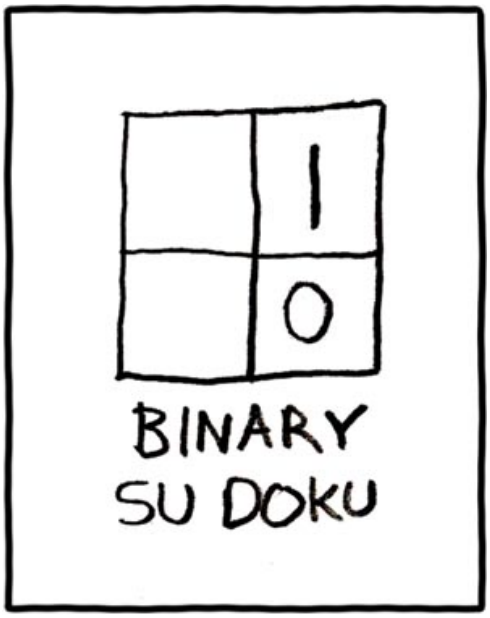
Suma y resta en binario

Trinchero Daniel

66.70 Estructura del Computador - Curso 002
Facultad de Ingeniería, Universidad de Buenos Aires



1. Introducción

Estamos acostumbrados a usar el símbolo «-» para representar un número negativo. Esto nos resulta muy práctico cuando trabajamos en el sistema numérico decimal, pero, por razones de diseño, simplicidad y eficiencia la computadora trabaja con otro sistema númerico que es el binario.

En dicho sistema, el símbolo «-» no es utilizado, en cambio, se representan los números negativos de formas que luego facilitarán las operaciones aritméticas. Las formas de representación mas conocidas son Módulo y Signo, Complemento al módulo, Complemento al módulo menos uno y Exceso.

En este documento haremos hincapié en las operaciones de suma y resta utilizando **Complemento al módulo** y **Complemento al módulo menos uno**.

Nota Este documento está escrito para uso de los alumnos del curso 002 de Estructura del Computador. Los temas que se tratarán, y otros temas de gran importancia para entender este documento se consideran sabidos.

2. Suma y resta en binario

En lo que respecta a la suma (con signo) de dos números positivos no hay mucho para decir. Los números positivos se representan igual que si se consideracen sin signo, por ende el resultado es el mismo.

Ejemplo 1 Se desea sumar los números 5 y 10 en binario. Utilizar 5 bits (5 dígitos) y, además, considerar que los números están representados en complemento al módulo y complemento al módulo menos uno.

Lo primero que hacemos es convertir 5 y 10 a base 2 utilizando 5 bits

$$(5)_{10} = (00101)_2$$
$$(10)_{10} = (01010)_2$$

Estos números se representan igual en ambos complementos que si se considerara que no tienen signo. Finalmente la suma es

$$\begin{array}{r} 00101 \\ +01010 \\ \hline 01111 \end{array}$$

ya sea que los números estén representados en complemento a la base o en complemento a la base menos uno.

Una resta siempre puede ser expresada como la suma de su complemento. La resta, desde el punto de vista de su implementación en hardware, conviene sea realizada como la suma del minuendo mas el complemento del sustraendo ya que ello evita tener que implementar un hardware específico para la resta (restador). Dicho de otra forma,

$$A - B = A + (-B)$$

2.1. Resta en complemento al módulo

Uno de los métodos, el más usado por las computadoras hoy en día, es representar el número que se desea restar, en complemento a dos[1] y luego sumarlo. La implementación de sumadores haciendo uso de esta convención es más conveniente que utilizando complemento a uno.

Representar un número A en complemento a la base (en el sistema binario se dice complemento a dos y se lo nota C_{A2}), como su nombre lo indica, consiste en complementar el número (i.e. lo que falta para llegar al módulo). Matematicamente sería

$$C_M(B) = M - B \text{ (ver nota al final[2])}$$

Teniendo en cuenta que A y B son números n bits, y que M (módulo) es un número de n + 1 bits, aplicamos esta representación a la resta

$$\begin{aligned} A - B &= A - B + M - M \\ &= A + (M - B) - M \\ &= A + C_M(B) - M \end{aligned}$$

Aquí podemos considerar dos posibles resultado:

- Que la suma $A + C_M(B)$ dé **carry**. En este caso, el carry ocuparía el bit n+1, por consiguiente se anularía con el módulo, dando como resultado

$$A - B = A + C_M(B)$$

- O el caso en que la suma $A + C_M(B)$ **no dé carry**

$$\begin{aligned} A - B &= A + C_M(B) - M \\ &= -M + A + C_M(B) \\ &= -(M - A - C_M(B)) \\ &= -(M - (A + C_M(B))) \end{aligned}$$

El término $M - (A + C_M(B))$ es el complemento al módulo de $A + C_M(B)$, finalmente

$$A - B = -C_M(A + C_M(B))$$

Ejemplo 2 Se desea restar los números 5 y 10 en binario. Utilizar 5 bits (5 dígitos) y considerar que los números están representados en complemento al módulo.

Lo primero que hacemos es convertir 5 y 10 a base 2 utilizando 5 bits

$$\begin{aligned} (5)_{10} &= (00101)_2 \\ (10)_{10} &= (01010)_2 \end{aligned}$$

[1] Las demostraciones se harán para cualquier base, pero los ejemplos son para base 2

[2] Dijimos que se utiliza esta representación para evitar hacer restas, sin embargo, el proceso de conversión de un número A a su complemento $C_M(A)$ obliga (aparentemente) a hacer una resta $C_M(A) = M - A$. En la práctica, al trabajar con el sistema númerico binario, esta conversión es muy sencilla y fácil de implementar.

Luego operamos

$$\begin{aligned} 00101 - 01010 &= 00101 - 01010 + 100000 - 100000 \\ &= 00101 + (100000 - 01010) - 100000 \end{aligned}$$

El complemento al módulo de 01010 es 10110

$$00101 - 01010 = 00101 + 10110 - 100000$$

En este caso la suma 00101 + 10110 no da carry

$$\begin{aligned} 00101 - 01010 &= 11011 - 100000 \\ &= -100000 + 11011 \\ &= -(100000 - 11011) \end{aligned}$$

El complemento al módulo de 11011 es 00101 (5 en decimal), finalmente

$$00101 - 01010 = -00101 \textit{ (nos dió el resultado esperado -5)}$$

Ejemplo 3 Se desea restar los números 15 y 4 en binario. Utilizar 5 bits (5 dígitos) y considerar que los números están representados en complemento al módulo.

Lo primero que hacemos es convertir 10 y 4 a base 2 utilizando 5 bits

$$\begin{aligned} (15)_{10} &= (01111)_2 \\ (4)_{10} &= (00100)_2 \end{aligned}$$

Luego operamos

$$\begin{aligned} 01111 - 00100 &= 01111 - 00100 + 100000 - 100000 \\ &= 01111 + (100000 - 00100) - 100000 \end{aligned}$$

El complemento al módulo de 00100 es 11100

$$01111 - 00100 = 01111 + 11100 - 100000$$

En este caso la suma 01111 + 11100 **si** da carry

$$\begin{array}{r} 01111 \\ +11100 \\ \hline \boxed{1}\,01011 \end{array}$$

Cómo en este caso hay carry, y según lo visto anteriormente, el resultado será

$$\begin{aligned} 01111 - 00100 &= 01111 + 11100 \\ &= 01011 \textit{(nos dió el resultado esperado 11)} \end{aligned}$$

Este método teórico no es utilizado, en la práctica para realizar la operación A - B se complementa al módulo B y se lo suma directamente. Teniendo en cuenta, que si el número resultante empieza con un 1 (para el caso del binario) en su bit más significativo, el resultado será negativo y habrá que complementarlo nuevamente para obtener el número que corresponde (agregandole el signo menos adelante), en caso contrario será positivo y no se realiza ninguna otra operación.

Para el ejemplo 2, la operación era 5 - 10

$$\begin{array}{r} 00101 \\ +10110 \\ \hline 11011 \end{array}$$

Como el resultado dió negativo, se lo complementa nuevamente y se le agrega el signo menos $(-00101)_2 = (-5)_{10}$.

Para el ejemplo 3, la operación era 15 - 4

$$\begin{array}{r} 01111 \\ +11100 \\ \hline \boxed{1}\,01011 \end{array}$$

En este caso el resultado dió positivo, es decir, $(01011)_2 = (11)_{10}$.

2.2. Resta en complemento al módulo menos uno

Otra forma de representación es el complemento al módulo menos uno. No es tan habitual su uso pero es de importancia conocerla.

En este caso, se complementa al módulo menos uno (en binario también se dice complemento a 1 y se nota C_{A1}). Matematicamente sería:

$$C_{M-1}(A) = M - 1 - A$$

Teniendo en cuenta que A y B son números n bits, y que M (módulo) es un número de n + 1 bits, aplicamos esta representación a la resta

$$\begin{aligned} A - B &= A - B + M - M + 1 - 1 \\ &= A + (M - 1 - B) - M + 1 \\ &= A + C_{M-1}(B) - M + 1 \end{aligned}$$

Aquí, nuevamente, podemos considerar dos posibles resultado:

- Que la suma $A + C_{M-1}(B)$ dé **carry**. En este caso, el carry ocuparía el bit n+1, por consiguiente se anularía con el módulo, dando como resultado

$$A - B = A + C_{M-1}(B) + 1$$

- O el caso en que la suma $A + C_{M-1}(B)$ **no de carry**

$$\begin{aligned} A - B &= A + C_M(B) - M + 1 \\ &= -M + A + C_M(B) + 1 \\ &= -(M - A - C_M(B) - 1) \\ &= -(M - 1 - (A + C_M(B))) \end{aligned}$$

El término $M-1-(A+C_M(B))$ es el complemento al módulo menos uno de $A+C_{M-1}(B)$, finalmente

$$A - B = -C_{M-1}(A + C_M(B))$$

Ejemplo 4 Se desea restar los números 5 y 10 en binario. Utilizar 5 bits (5 dígitos) y considerar que los números están representados en complemento al módulo menos uno.

Lo primero que hacemos es convertir 5 y 10 a base 2 utilizando 5 bits

$$(5)_{10} = (00101)_2$$
$$(10)_{10} = (01010)_2$$

Luego operamos

$$\begin{aligned} 00101 - 01010 &= 00101 - 01010 + 10000 - 100000 + 00001 - 00001 \\ &= 00101 + (10000 - 01010 - 00001) - 100000 + 00001 \end{aligned}$$

El complemento al módulo menos uno de 01010 es 10101

$$00101 - 01010 = 00101 + 10101 - 100000 + 000001$$

En este caso la suma 00101 + 10101 no da carry

$$\begin{aligned} 00101 - 01010 &= 11010 - 100000 + 00001 \\ &= -100000 + 11010 + 00001 \\ &= -(100000 - 11010 - 00001) \end{aligned}$$

El complemento al módulo menos uno de 11010 es 00101 (5 en decimal), finalmente

$$00101 - 01010 = -00101 \textit{ (nos dió el resultado esperado -5)}$$

Ejemplo 5 Se desea restar los números 15 y 4 en binario. Utilizar 5 bits (5 dígitos) y considerar que los números están representados en C_{A1}.

Lo primero que hacemos es convertir 15 y 4 a base 2 utilizando 5 bits

$$(15)_{10} = (01111)_2$$
$$(4)_{10} = (00100)_2$$

Luego operamos

$$\begin{aligned} 01111 - 00100 &= 01111 - 00100 + 100000 - 100000 + 00001 - 00001 \\ &= 01111 + (100000 - 00100 - 00001) - 100000 + 00001 \end{aligned}$$

El complemento al módulo menos uno de 00100 es 11011

$$01111 - 00100 = 01111 + 11011 - 100000 + 00001$$

En este caso la suma 01111 + 11011 **si** da carry

$$\begin{array}{r} 01111 \\ +11011 \\ \hline \boxed{1}\,01010 \end{array}$$

Cómo en este caso hay carry, y según lo visto anteriormente, el resultado será

$$\begin{aligned} 01111 - 00100 &= 01111 + 11011 + 00001 \\ &= 01011 \textit{(nos dió el resultado esperado 11)} \end{aligned}$$

En la práctica, para realizar la operación A - B directamente se complementa B y se lo suma a A, si la cuenta da carry, se le suma un uno[3]. al final. Para el caso del ejemplo 5

$$\begin{array}{r} 01111 \\ +11011 \\ \hline \boxed{1}\,01010 \end{array}$$

sumamos un 1

$$\begin{array}{r} 01010 \\ +00001 \\ \hline 01011 \end{array}$$

Como se puede ver, para obtener el resultado correcto hay que sumar el carry. Este es uno de los motivos por los cuales no se utiliza esta convención, su implementación es más costosa. Otro de los motivos es que en complemento al módulo menos uno hay dos representaciones del cero.

Por otra parte, si el número resultante empezara con un 1 (para el caso del binario) en su bit más significativo, significaría que el resultado es negativo y habría que complementarlo nuevamente para obtener el número que corresponde (agregandole el signo menos adelante), en caso contrario sería positivo y no se realizaría ninguna otra operación.

Para el ejemplo 4, al hacer 5 - 10

$$\begin{array}{r} 00101 \\ +10101 \\ \hline 11010 \end{array}$$

Como el resultado es negativo, se lo complementa nuevamente y se le agrega el signo menos $(-00101)_2 = (-5)_{10}$

3. Conclusión

Hasta aquí, hemos visto el método teórico para restar dos números utilizando complemento al módulo y complemento al módulo menos uno. Se han dado además, un par de ejemplos utilizando base dos, resueltos por el método teórico y por el método práctico. Pero todavía queda una interrogante ¿Cómo trabaja la computadora?.

La unidad aritmético lógica (ALU) realiza una única operación: **la suma**. Suma dos operandos A y B sin importar el signo, incluso sin tener en cuenta la convención en la que están representados. Quedará a criterio del programador interpretar el resultado como corresponde. El método práctico que utilizamos en los ejemplos y en clase para resolver ejercicios es muy similar a la forma de operar de una ALU.

[3]La computadora, al hacer una operación, nos brinda además del resultado un conjunto de banderas (Carry, Overflow, Signo, Cero y Paridad) que nos permiten analizar el resultado de la cuenta. En este caso, que luego de hacer la suma, se le adiciona uno, las banderas (flags) no se recalculan.