



MATEMÁTICAS PARA LA COMPUTACIÓN
CAPÍTULO 1. SISTEMAS NUMÉRICOS

SUMA DE DOS CANTIDADES EN COMPLEMENTO A 2.

AUTOR: JOSÉ ALFREDO JIMÉNEZ MURILLO

SUMA DE DOS CANTIDADES EN COMPLEMENTO A 2.

Realmente las operaciones que la computadora realiza internamente se llevan a cabo en forma muy diferente a la que nosotros estamos acostumbrados, se sabe de antemano que el sistema numérico utilizado es el binario y que la operación básica es la suma. En computación las cantidades se representa por un conjunto de bits (ceros y unos), usando un bit exclusivo para distinguir las cantidades negativas de las positivas, el cual recibe el nombre de "bit de signo". La convención más común para el signo es 0=positivo y 1=negativo.

Existen tres formas de representar cantidades, estas son: magnitud verdadera, complemento a 1 y complemento a 2. cada una de ellas tiene su utilidad, dentro de la computación.

Magnitud verdadera.

En la representación en magnitud verdadera se muestran los bits en forma real, una característica de este tipo de representación es que se puede saber fácilmente a cuanto equivale ese conjunto de bits en el sistema decimal, usando para ello la representación exponencial.

$$1 \ 110110101.011_{(2)} = -1 \times 2^8 + 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} = 437.44_{(10)}$$

Complemento a 1.

Como en el sistema binario solamente existen como dígitos válidos el 0 y el 1, por lo tanto se dice que el complemento de 0 es 1 y el complemento de 1 es 0. El complemento de un número en binario se obtiene complementado cada uno de los bits, sin considerar el signo.

1	1	0	1	0	1	1	1	0	0	1	0	0	1	.	0	1	(2)	Magnitud verdadera
1	0	1	0	1	0	0	0	1	1	0	1	1	0	.	1	0	(2)	Complemento a 1
0	1	0	0	0	1	0	0	1	1	.	1	0	0	(2)	Magnitud verdadera			
0	0	1	1	1	0	1	1	0	0	.	0	1	1	(2)	Complemento a 1			

Se puede observar que para obtener el complemento a 1 de una cantidad expresada en binario, solamente es suficiente con cambiar todos los ceros por unos y los unos por ceros, pero en ningún momento se cambia el bit de signo, que en este caso es el bit de la extrema izquierda.

Para resolver el problema de desbordamiento es conveniente definir a las variables con una capacidad mayor, por ejemplo suponer que las variables A, B y C son enteras pero ahora con una capacidad de 16 bits, que es lo que realmente ocurre en un programa cuando se definen variables con capacidad menor a la requerida.

$$\begin{array}{r}
 + 225_{(10)} = 00000000011100001_{(2)} \\
 + 76_{(10)} = 00000000001001100_{(2)} \\
 \hline
 + 301_{(10)} = 00000000100101101_{(2)}
 \end{array}$$

↙ Signo
↑ Magnitud

Se observa ahora que $+301_{(10)} = 0000000100101101_{(2)}$ es el resultado correcto, con lo cual se evita el desbordamiento.

Es importante mencionar que el desbordamiento solamente ocurre cuando las dos cantidades que se están sumando son del mismo signo, ya que son los únicos casos en que el resultado puede requerir mayor espacio. Cuando las cantidades a sumar son de signo contrario no se presenta el desbordamiento pues el valor absoluto del resultado siempre será menor al valor absoluto de alguna de las cantidades que se suman.

Ejemplo

Sumar $A = -225$ con $B = +76$

Cuando una cantidad es negativa, se deberá encontrar el complemento a 2 de esa cantidad y después realizar la suma.

$$\begin{array}{r}
 - 225_{(10)} = 111100001_{(2)} \text{ Magnitud verdadera} \\
 \phantom{- 225_{(10)}} = 100011110_{(2)} \text{ Complemento a 1} \\
 \phantom{- 225_{(10)}} = \frac{ + 1}{100011111_{(2)}} \text{ Complemento a 2}
 \end{array}$$

Note que para encontrar el complemento a 1 se cambian todos los bits por su complemento, pero el bit de signo no se cambia. Para encontrar el complemento a 2 se le suma 1 al bit menos significativo del complemento a 1.

Ahora si se procede a sumar el complemento a 2 de la cantidad negativa y la otra cantidad positiva.

$$\begin{array}{r}
 - 225_{(10)} = 100011111_{(2)} \\
 + 76_{(10)} = 001001100_{(2)} \\
 \hline
 - 149_{(10)} = 101101011_{(2)}
 \end{array}$$

↑ Signo
↑ Magnitud

El resultado obtenido es negativo, como se esperaba pero la magnitud obtenida no es la correcta ya que $101101011_{(2)} = -107_{(10)}$ es diferente a $-149_{(10)}$. En forma general se puede decir que si el resultado de la suma es negativo, se deberá complementar a 2 el resultado.

$$\begin{array}{r}
 101101011_{(2)} \text{ Resultado negativo} \\
 = 110010100_{(2)} \text{ Complemento a 1} \\
 + 1_{(2)} \\
 \hline
 - 149_{(10)} = 110010101_{(2)} \text{ Complemento a 2}
 \end{array}$$

Ejemplo

Sumar A= +225 con B=-76

Complementando a 2 la cantidad negativa.

$$\begin{array}{r}
 - 76_{(10)} = 101001100_{(2)} \text{ Magnitud verdadera} \\
 = 110110011_{(2)} \text{ Complemento a 1} \\
 + 1_{(2)} \\
 \hline
 = 110110100_{(2)} \text{ Complemento a 2}
 \end{array}$$

Sumando el complemento encontrado a la cantidad positiva se obtiene.

$$\begin{array}{r}
 + 225_{(10)} = 011100001_{(2)} \\
 - 76_{(10)} = 110110100_{(2)} \\
 \hline
 + 149_{(10)} = 1010010101_{(2)}
 \end{array}$$

↑ Acarreo
↑ Signo
↑ Magnitud

Se puede observar en la suma anterior que el resultado es positivo, por lo tanto ya es el resultado correcto, ya que solamente se complementan a 2 los resultados negativos. También se obtiene un acarreo, el cual se debe despreciar en todos los casos.

Ejemplo.

Sumar A= -225 con B=-76

Cuando una cantidad es negativa, se deberá encontrar el complemento a 2 de esa cantidad. En este caso las dos cantidades a sumar son negativas, por lo tanto se tienen que encontrar el complemento a 2 de ambas antes de realizar la suma. Pero también se debe tomar en cuenta que la suma produce un desbordamiento de tal forma que en el complemento ya se debe trabajar con los bits correctos.

Complementando a dos ambas cantidades y considerando el desbordamiento.

$$\begin{array}{r}
 - 225_{(10)} = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ (2) \text{ Magnitud verdadera} \\
 \phantom{- 225_{(10)}} = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ (2) \text{ Complemento a 1} \\
 \phantom{- 225_{(10)}} = \underline{} \\
 \phantom{- 225_{(10)}} = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ (2) \text{ Complemento a 2}
 \end{array}$$

$$\begin{array}{r}
 - 76_{(10)} = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ (2) \text{ Magnitud verdadera} \\
 \phantom{- 76_{(10)}} = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ (2) \text{ Complemento a 1} \\
 \phantom{- 76_{(10)}} = \underline{} \\
 \phantom{- 76_{(10)}} = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ (2) \text{ Complemento a 2}
 \end{array}$$

Sumando:

$$\begin{array}{r}
 - 225_{(10)} = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ (2) \\
 - 76_{(10)} = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ (2) \\
 \hline
 - 301_{(10)} = 11\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ (2)
 \end{array}$$

↑
↑
↑

Acarreo
Signo
Magnitud

Si se convierte a decimal el resultado obtenido es posible observar que no es el esperado de $-301_{(10)}$. Sin embargo se sabe que cuando el resultado de la suma es negativo se deberá complementar a 2. En este caso también se tiene acarreo, el cual se desprecia.

$$\begin{array}{r}
 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \quad (2) \text{ Resultado negativo} \\
 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \quad (2) \text{ Complemento a 1} \\
 \hline
 - \ 3 \ 0 \ 1 = \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \quad (2) \text{ Complemento a 2}
 \end{array}$$

Ejercicios:

Realizar la suma en complemento a 2, en cada uno de los siguientes incisos. Considerar que las cantidades que se están sumando se definen como enteras y que ocupan 2 bytes de memoria. Agregar bytes en caso de ser necesario, para evitar el desbordamiento.

a)
$$\begin{array}{r}
 + \ 6 \ 5 \ 5 \ 0 \ 8 \quad (10) \\
 + \ \underline{\quad \quad \quad 1 \ 0 \ 3 \quad (10)}
 \end{array}$$

b)
$$\begin{array}{r}
 - \ 6 \ 5 \ 5 \ 0 \ 8 \quad (10) \\
 + \ \underline{\quad \quad \quad 1 \ 0 \ 3 \quad (10)}
 \end{array}$$

c)
$$\begin{array}{r}
 + \ 6 \ 5 \ 5 \ 0 \ 8 \quad (10) \\
 - \ \underline{\quad \quad \quad 1 \ 0 \ 3 \quad (10)}
 \end{array}$$

c)
$$\begin{array}{r}
 - \ 6 \ 5 \ 5 \ 0 \ 8 \quad (10) \\
 - \ \underline{\quad \quad \quad 1 \ 0 \ 3 \quad (10)}
 \end{array}$$

Después de llevar a cabo las operaciones los resultados para cada uno de los incisos son:

a) $0 \ 00000010000000001001011_{(2)}$ b) $1 \ 1111111101111101_{(2)}$
c) $0 \ 1111111101111101_{(2)}$ d) $1 \ 00000010000000001001011_{(2)}$

Considerar que la parte entera en cada una de las cantidades que se suman ocupan 8 bits y que la parte fraccionaria ocupa 4 bits. Realizar la suma en complemento a 2. Agregar bytes en caso de ser necesario, para evitar el desbordamiento.

a)
$$\begin{array}{r}
 + \ 5 \ 4 \ . \ 2 \ 3 \quad (10) \\
 + \ \underline{\quad \quad \quad 2 \ 8 \ . \ 5 \ 6 \quad (10)}
 \end{array}$$

b)
$$\begin{array}{r}
 - \ 5 \ 4 \ . \ 2 \ 3 \quad (10) \\
 + \ \underline{\quad \quad \quad 2 \ 8 \ . \ 5 \ 6 \quad (10)}
 \end{array}$$

c)
$$\begin{array}{r}
 + \ 5 \ 4 \ . \ 2 \ 3 \quad (10) \\
 - \ \underline{\quad \quad \quad 2 \ 8 \ . \ 5 \ 6 \quad (10)}
 \end{array}$$

d)
$$\begin{array}{r}
 - \ 5 \ 4 \ . \ 2 \ 3 \quad (10) \\
 - \ \underline{\quad \quad \quad 2 \ 8 \ . \ 5 \ 6 \quad (10)}
 \end{array}$$

Después de llevar a cabo las operaciones es posible comprobar que los resultados son los siguientes:

a) $0 \ 01010010.1011_{(2)}$

c) $0 \ 00011001.1011_{(2)}$

b) $1 \ 00011001.1011_{(2)}$

d) $1 \ 01010010.1011_{(2)}$

Conclusiones de las sumas en complemento a 2.

De la misma manera en que se sumaron dos cantidades enteras en complemento a 2, también es posible sumar dos cantidades con parte entera y fraccionaria, teniendo en cuenta que el desbordamiento solamente puede darse en la parte entera, ya que las operaciones se llevan a cabo de derecha a izquierda y que para evitar el desbordamiento se deberá trabajar con el número de bits suficiente para evitarlo. Para encontrar el complemento a 1 se cambian cada bit por su complemento, independientemente si se encuentra en la parte entera o en la parte fraccionaria. Para encontrar el complemento a 2 se suma un 1 en el bit menos significativo (el que esté más a la derecha) independientemente si pertenece a la parte entera o bien a la parte fraccionaria. Que solamente se complementan a 2, las cantidades negativas y los resultados de las suma negativos.