Problemas sobre representación de la información (Ejemplos resueltos)

March 17, 2023

Problema 1. Utiliza las representaciones módulo signo, complemento a 1 y complemento a 2 para realizar las siguientes sumas en una celda de 7 bits:

- $-33 - 20$.
- $-40 - 24$.
- $50 - 12$.
- $64 - 10$.

Proof. Solución

Debemos comenzar por calcular el rango en cada una de las representaciones que vamos a utilizar. Para módulo signo y complemento a 1 el rango es:

$$R = (-(2^{n-1} - 1), 2^{n-1} - 1)$$

Con 7 bits: $R_{MS} = R_{C1} = (-63, 63)$.

En cambio el rango para complemento a 2, viene dado por:

$$R = (-2^{n-1}, 2^{n-1} - 1)$$

Por tanto, en este caso: $R_{C2} = (-64, 63)$.

Comencemos por la primera operación. Examinamos si los operandos están en los rangos y vemos que están en el rango de todas las representaciones. Ahora traducimos los operandos a cada representación sobre la celda de 7 bits.

En módulo signo: $-33 = 1100001$ y $-20 = 1010100$. Ahora operamos los bits mediante la suma binaria.

C	0	0	0	0	0	0	
O1	1	1	0	0	0	0	1
O2	1	0	1	0	1	0	0
C=1	0	1	1	0	1	0	1

Ahora convertimos el resultado a decimal, bajo la interpretación de módulo signo:

$$S = (0110101)_{MS} = +53$$

Como vemos, el resultado no concuerda con el valor correcto de la operación suma. Así pues, la suma binaria no coincide con la suma en el caso de que la representación sea módulo signo.

En complemento a uno: $-33 = 1011110$, $-20 = 1101011$. Entonces la suma binaria es:

C	1	1	1	1	1	0	
O1	1	0	1	1	1	1	0
O2	1	1	0	1	0	1	1
C=1	1	0	0	1	0	0	1

Ahora traducimos el resultado a decimal bajo la interpretación de complemento a 1.

$$S = (1001001)_{C1} = -(0110110)_2 = -54$$

. Como vemos no coincide con el resultado de la suma. En complemento a 1, la suma binaria no coincide con la suma, cuando hay un carry final. Este carry puede añadirse al resultdo e el bit menos significativo para obtener el resultado correcto.

Como vemos, el resultado coincide con la operación suma. Mientras el resultado esté en el rango de la representación (no se produce overflow), el valor de la suma binaria coincide con el de la suma.

En complemento a dos: $-33 = 1011111$, $-20 = 1101100$. Entonces la suma binaria es:

C	1	1	1	1	0	0	
O1	1	0	1	1	1	1	1
O2	1	1	0	1	1	0	0
C=1	1	0	0	1	0	1	1

Ahora traducimos el resultado a decimal bajo la interpretación de complemento a 2.

$$S = (1001011)_{C2} = -(0110101)_2 = -53$$

Como vemos, el resultado coincide con la operación suma. Mientras el resultado esté en el rango de la representación (no se produce overflow), el valor de la suma binaria coincide con el de la suma. No es necesario ninguna operación adicional.

□

Problema 2. Realiza en complemento a 2 las siguientes sumas sobre una celda de 5 bits y obtén el bit de overflow en cada caso:

- $-9 - 7$
- $9 + 7$
- $14 - 4$

Proof. Solución

En primer lugar hemos de encontrar el rango.

$$R = (-16, 15)$$

Todos los operandos están en el rango, por lo tanto podemos realizar todas las conversiones a binario.

$$-9 = (10111)_{C2}$$
$$-7 = (11001)_{C2}$$

La suma binaria es

C	1	1	1	1	
O1	1	0	1	1	1
O2	1	1	0	0	1
C=1	1	0	0	0	0

Como vemos los tres bits de signos (operandos y solución) son iguales, con lo que el bit de overflow es 0. Veremos que efectivamente la solución es correcta:

$$S = (10000)_{C2} = -(10000)_2 = -16$$

$$9 = (01001)_{C2}$$
$$7 = (00111)_{C2}$$

La suma binaria es

C	1	1	1	1	
O1	0	1	0	0	1
O2	0	0	1	1	1
C=0	1	0	0	0	0

Como vemos los tres bits de signos (operandos y solución) no son iguales, con lo que el bit de overflow es 1. Hay un desbordamiento porque la suma está fuera del rango. Veremos que efectivamente la solución es incorrecta:

$$S = (10000)_{C2} = -(10000)_2 = -16$$

□

Problema 3. Dadas las siguientes cadenas de bits, calcula el número en base decimal que representan bajo una interpretación de complemento a 2.

- $(101001)_{C2}$
- $(010101)_{C2}$
- $(11001.01)_{C2}$

Proof. 3

En el primer caso, tenemos que el bit se signo es 1, por lo tanto se trata de un número negativo. Así que debemos aplicar la operación NEG a toda la celda para calcular una representación que interpretaremos en binario natural para obtener el módulo.

$$|(101001)_{C2}| = NEG(101001)$$
$$NEG(101001) = (010111)_2 = (23)_{10}$$

Por lo tanto:

$$(101001)_{C2} = -23$$

El segundo ejemplo, se trata de un número positivo, así que sólo hemos de aplicar la regla de la representación posicional, interpretándolo en binario natural.

$$(010101)_{C2} = 21$$

Finalmente, el último número es un numero fraccionario negativo (ver bit de signo). Debemos aplicar la operación NEG, por tanto:

$$|(11001.01)_{C2}| = NEG(11001.01)$$
$$NEG(11001.01) = (00110.11)_2 = 6.75$$

Por tanto:

$$(11001.01)_{C2} = -6.75$$

□