Representación de Números Binarios con Signo:

Sistemas de Procesamiento de Datos - UTN

Prof. Lic Verónica Lourdes Tomich
Prof. TUP Rodrigo Soto
Prof. TUSI Leonardo Chiessa
Prof. Lic Eduardo Monaco
Prof. PDI Guillermo Gimenez

Representación de Números Binarios con Signo

INTRODUCCIÓN:

Hasta el momento hemos estado trabajando con Números Naturales lo que se conoce como Representación de Números Binarios sin Signo. En esta presentación vamos a ver cómo representamos **Números Binarios con Signo**.

Representación de Números Binarios con Signo

SIGNO-MAGNITUD:

- Este sistema sirve para representar tanto números positivos como negativos de una forma muy sencilla.
- Para representar los números se deben definir datos con tamaño estándar: 8 bits, 16 bits, 32 bits, etc.

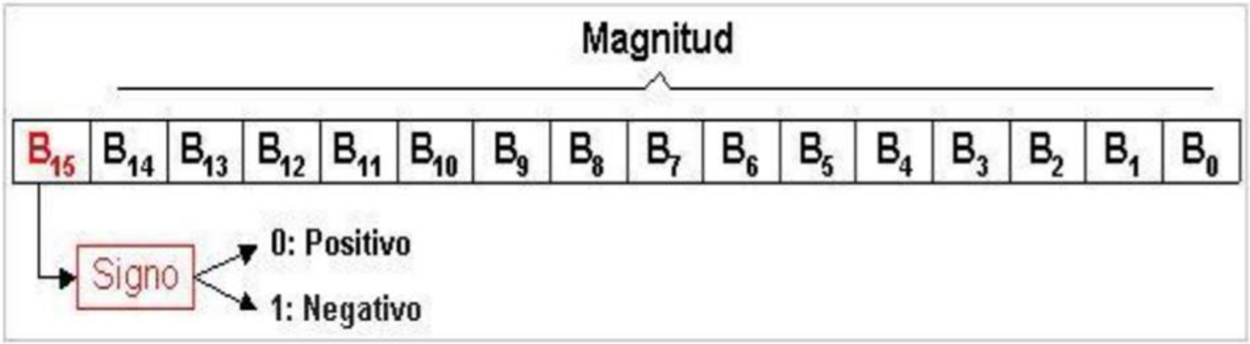
Representación de Números Binarios con Signo

SIGNO-MAGNITUD:

- En este formato, el bit más significativo (MSB) del dato se utiliza para indicar el signo y los bits restantes representan la magnitud del número.
- Posee dos representaciones del 0, el 0 y el -0.

Representación de Números Binarios con Signo

La siguiente imagen representa un número de 16 bits, donde B_{15} es el MSB(Bit Más Significativo) y representa el Signo, el resto de los bits van a representar la Magnitud del número.



Representación de Números Binarios con Signo

Ejemplo de representación del número 50 y -50 en Signo-Magnitud con 8 bits:

Número en Decimal	Número en Signo-Magnitud
50	**0** 0 1 1 0 0 1 $0_{S\text{-}M}$
-50	**1** 0 1 1 0 0 1 $0_{S\text{-}M}$

REPRESENTACIÓN NUMÉRICA EN SIGNO-MAGNITUD

Para n = 8 (8 bits) en Signo y Magnitud

Número Binario de 8 bits	Interpretado como número entero en Sistema Decimal	Interpretado como número natural en Sistema Decimal
00000000	0	0
00000001	1	1
00000010	2	2
...	...	...
01111110	126	126
01111111	127	127

Para n = 8 (8 bits) en Signo y Magnitud

Número Binario de 8 bits	Interpretado como número entero en Sistema Decimal	Interpretado como número natural en Sistema Decimal
10000000	-0	128
10000001	−1	129
10000010	−2	130
...	...	...
11111101	−125	253
11111110	−126	254
11111111	−127	255

Representación de Números Binarios con Signo

COMPLEMENTOS BINARIOS:

En aritmética decimal, cada número tiene un complemento, es decir, un valor que cuando se agrega al número original da como resultado un cero.

Por ejemplo, 5 y -5 son complementos porque 5 + (-5) = 0.

Representación de Números Binarios con Signo

COMPLEMENTO A UNO:

- Tiene muchas aplicaciones en los circuitos digitales y sistemas de computación.
- Sirve para representar tablas numéricas de cantidades positivas y negativas.
- Es utilizado como paso previo para hallar el complemento a dos.

Representación de Números Binarios con Signo

COMPLEMENTO A UNO:

- Se puede determinar que el COMPLEMENTO A UNO se obtiene invirtiendo el estado o nivel de los bits que conforman la cifra en todas sus posiciones.
- No proporciona muchos beneficios para buscar el complemento ya que cuenta con dos representaciones del 0.

Representación de Números Binarios con Signo

Ejemplo de representación del número 50_{10} y -50_{10} en Complemento a 1 con 8 bits:

Número en Decimal	Número en Complemento a 1
50	**0** 0 1 1 0 0 1 0_{C-1}
-50	**1** 1 0 0 1 1 0 1_{C-1}

REPRESENTACIÓN EN COMPLEMENTO A UNO

Para *n* = 8 (8 bits) en Complemento a uno

Valores de 8 bits	Interpretado en Complemento a uno en decimal	Interpretado como Entero sin signo en decimal
00000000	0	0
00000001	1	1
00000010	2	2
...	...	...
01111110	126	126
01111111	127	127

Para *n* = 8 (8 bits) en Complemento a uno

Valores de 8 bits	Interpretado en Complemento a uno en decimal	Interpretado como Entero sin signo en decimal
10000000	−127	128
10000001	−126	129
10000010	−125	130
...	...	...
11111101	−2	253
11111110	−1	254
11111111	−0	255

Representación de Números Binarios con Signo

COMPLEMENTO A DOS:

- En el sistema binario, la forma más utilizada para representar los números enteros con signo es la de COMPLEMENTO A DOS.
- Los circuitos microprocesadores poseen internamente unidades de procesamiento aritmético que trabajan bajo este formato.

Representación de Números Binarios con Signo

Para obtener el número en COMPLEMENTO A DOS se pueden utilizar dos métodos:

- Uno de los métodos consiste en obtener el complemento a uno de la cifra y luego sumarle uno.
- Para realizar esta operación primero vamos a ver la Suma Binaria.

Representación de Números Binarios con Signo

SUMA BINARIA:

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 2_{10}$ en binario es = 10_2(son dos dígitos), por lo tanto,

el resultado es 0 y llevo 1a la columna de la izquierda.

Representación de Números Binarios con Signo

Ejemplo de Suma Binaria con 4 bits:

$5_{10} = 0101_2$ $4_{10} = 0100_2$

$5_{10} + 4_{10} = 9_{10}$

$\mathbf{1001_2 = 9_{10}}$

1
+ 0 1 0 1_2
0 1 0 0_2
1 0 0 1_2

Representación de Números Binarios con Signo

Ejemplo de representación del número 50 y -50 en Complemento a 2 con 8 bits:

$50_{10} = 00110010_{C-2}$

$-50_{10} = \mathbf{11001110}_{\mathbf{C-2}}$

Binario positivo	$0\ 0\ 1\ 1\ 0\ 0\ 1\ 0_2$
Complemento a 1	$1\ 1\ 0\ 0\ 1\ 1\ 0\ 1_{C-1}$
+	1_2
Complemento a 2	$1\ 1\ 0\ 0\ \ 1\ 1\ 1\ 0_{C-2}$

Representación de Números Binarios con Signo

COMPLEMENTO A DOS:

- Otro método de conversión consiste en ir seleccionando y colocando de derecha a izquierda los dígitos binarios hasta conseguir el primer bit en uno, de allí en adelante se cambian de estado todos los bits restantes.

Representación de Números Binarios con Signo

Ejemplo de representación del número 50_{10} y -50_{10} en Complemento a 2 con 8 bits:

Número en Decimal	Número en Complemento a 2
50	$0\,0\,1\,1\,0\,0\,1\,0_{C\text{-}2}$
-50	$1\,1\,0\,0\,1\,1\,1\,0_{C\text{-}2}$

REPRESENTACIÓN EN COMPLEMENTO A DOS

Para n = 8 (8 bits) en Complemento a dos

Valores de 8 bits	Interpretado en Complemento a dos en decimal	Interpretado como Entero sin signo en decimal
00000000	0	0
00000001	1	1
00000010	2	2
...	...	...
01111110	126	126
01111111	127	127

Para n = 8 (8 bits) en Complemento a dos

Valores de 8 bits	Interpretado en Complemento a dos en decimal	Interpretado como Entero sin signo en decimal
10000000	−128	128
10000001	−127	129
10000010	−126	130
...	...	...
11111101	−3	253
11111110	−2	254
11111111	−1	255

Representación de Números Binarios con Signo

CONVERSIÓN DE SIGNO-MAGNITUD A DECIMAL:

Para poder determinar qué número en decimal representa un número que se encuentra en Signo-Magnitud se debe especificar la cantidad de bits que lo componen y realizar lo siguiente:

- Tomar el MSB y si este es 0 el número es positivo si es 1 es negativo.
- Tomar el resto de los bits y convertirlo a Binario sin Signo.

Representación de Números Binarios con Signo

Ejemplo de conversión de Signo-Magnitud a Decimal con 8 bits:

- $\underline{0}\,0110010_{S\text{-}M} = 0*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 0*2^0 =$

  bit de signo $= 0 + 32 + 16 + 0 + 0 + 2 + 0 = \mathbf{50_{10}}$

- $\underline{1}\,0110010_{S\text{-}M} = 0*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 0*2^0 =$

  bit de signo $= 0 + 32 + 16 + 0 + 0 + 2 + 0 = \mathbf{-50_{10}}$

Representación de Números Binarios con Signo

CONVERSIÓN DE COMPLEMENTO A DOS A DECIMAL:

Para poder determinar qué número en decimal representa un número que se encuentra en Complemento a 2 se debe especificar la cantidad de bits que lo componen, el MSB nos va a indicar el signo, si es 0 (positivo) se procede como Signo-Magnitud, si es 1 (negativo) se puede obtener de dos formas:

Representación de Números Binarios con Signo

Ejemplo de conversión de Complemento a 2 a Decimal con 8 bits:

$$\underline{0}\,0110010_{S\text{-}M} = 0*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 0*2^0 =$$

bit de signo $= 0 + 32 + 16 + 0 + 0 + 2 + 0 = \mathbf{50_{10}}$

Representación de Números Binarios con Signo

CONVERSIÓN DE COMPLEMENTO A DOS NEGATIVO A DECIMAL:

- Una opción es obtener el complemento del complemento, de esta manera llegaremos al número en su signo opuesto y luego lo convertimos a decimal, por ultimo, le agregamos el signo negativo ya que obtuvimos el número positivo.

Representación de Números Binarios con Signo

Ejemplo de conversión del número 11001110_2 en Complemento a 2 Negativo a Decimal con 8 bits:

$00110010_2 = 50_{10}$

Se agrega el Signo = **-50_{10}**

Complemento a 2	$1\ 1\ 0\ 0\ 1\ 1\ 1\ 0_{C\text{-}2}$
Complemento a 1	$0\ 0\ 1\ 1\ 0\ 0\ 0\ 1_{C\text{-}1}$
+	1_2
Binario positivo	$0\ 0\ 1\ 1\ 0\ 0\ 1\ 0_2$

Representación de Números Binarios con Signo

CONVERSIÓN DE COMPLEMENTO A DOS NEGATIVO A DECIMAL:

- Otra opción es representar el valor del MSB agregando el signo negativo, para luego, sumar el resto de los valores tal como se realiza con Binario sin Signo.

Representación de Números Binarios con Signo

Ejemplo de conversión de Complemento a 2 Negativo a Decimal con 8 bits:

$$\underline{1}1001110_{C\text{-}2} = -\,\mathbf{1}\mathbf{*2^7} + 1*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0 =$$

$$= \mathbf{-128} + 64 + 0 + 0 + 8 + 4 + 2 + 0 = \mathbf{-50_{10}}$$

Representación de Números Binarios con Signo

Comparación de las diferentes representaciones de números binarios con 8 bits:

Decimal	Signo-Magnitud	Complemento a 1	Complemento a 2
50	00110010	00110010	00110010
-50	10110010	11001101	11001110

Representación de Números Binarios con Signo

MíNIMOS Y MÁXIMOS:

- Cuando se utiliza un número finito de posiciones de bits para almacenar información, es necesario poder determinar los valores mínimos y máximos que cada representación binaria puede manejar.
- Un fallo al hacer esto podría producir errores en el software que se crea.

Representación de Números Binarios con Signo

Mínimos y Máximos: Representación de Binarios sin Signo.

El valor más pequeño que se puede representar con binario sin signo se produce cuando todos los bits son iguales a cero. Conversión desde binario a decimal en 0 + 0 + ... + 0 = 0. Por lo tanto, para un número de n bits:

Número binario sin signo de n-bit mínimo = **0**

Representación de Números Binarios con Signo

Mínimos y Máximos: Representación en Binarios sin Signo.

El valor más grande que se puede representar con binario sin signo se alcanza cuando todos los n bits son iguales. Cuando convertimos este valor de binario a decimal, obtenemos $2^{n-1} + 2^{n-2} + ... + 2^{0}$.

Por lo tanto, para un número binario sin signo de n-bit, el máximo es:

Número binario máximo n-bit sin signo = $(2^{n})-1$

Representación de Números Binarios con Signo

Para determinar la magnitud de un valor en Signo-Magnitud, se ignora el MSB y se utilizan los bits n-1 restantes para convertir a decimales como si estuvieran en representación sin signo. Esto significa que los valores más grandes y más pequeños representado con un número de Signo-Magnitud de n-bit es igual al positivo y valores negativos de un número binario sin signo (n-1)-bit.

- Número de Signo-Magnitud mínimo de n-bit = $-(2^{(n-1)}- 1)$
- Número de Signo-Magnitud máximo de n-bit = $(2^{(n-1)}- 1)$

Representación de Números Binarios con Signo

Mínimos y Máximos: Representación en Complemento a 2.

- A diferencia del caso sin signo, el más bajo valor decimal que se puede representar con n-bits en representación de complemento a 2 no es obvio.
- Resulta que el valor más bajo posible de Complemento a 2 es el MSB de 1 seguido de todos los ceros.

Número mínimo de complemento a 2 de n-bit = $\mathbf{-2^{(n-1)}}$

Representación de Números Binarios con Signo

Mínimos y Máximos: Representación en Complemento a 2.

- El valor máximo es un poco más fácil de encontrar. Es un número positivo, es decir, un MSB de 0. Los restantes n-1 bits se tratan como representación de Signo-Magnitud. Por lo tanto, para n bits:

Número máximo del Complemento a 2 de n-bit = $(2^{(n-1)} - 1)$

Representación de Números Binarios con Signo

Comparación de representación para números binarios de 8 bits.

Representación	Mínimo	Máximo	Número de Enteros Representados
Binario Sin Signo	0	255	256
Signo-Magnitud	-127	127	255
Complemento a 2	-128	127	256

SUMA BINARIA COMPLEMENTO A 2

En la suma binaria en complemento a 2 existen 4 posibles escenarios:

CASO 1: Sumar 2 números positivos.

CASO 2 y 3: Sumar un positivo y un negativo o de un negativo y un positivo.

CASO 4: Sumar 2 números negativos.

SUMA BINARIA COMPLEMENTO A 2

<u>CASO 1: SUMA DE 2 NÚMEROS POSITIVOS:</u>

- El resultado debe ser siempre un número positivo por lo que el MSB debe ser siempre 0.
- Es importante saber el rango de números (decimales) que admite la cantidad de bits que estamos utilizando.
- Si el número alcanzado estuviera fuera del rango permitido se producirá un OVERFLOW que derivará en un resultado erróneo por sobrecarga de datos

SUMA BINARIA COMPLEMENTO A 2

Ejemplo 1 Suma de 2 números positivos:

Dado los números binarios: W=00010001_2; T=00010101_2;

Obtener W+T (Trabajamos con 8 bits por lo que el 8vo bit debe ser 0)

			1				1	
	0	0	0	1	0	0	0	1
+	0	0	0	1	0	1	0	1
	0	0	1	0	0	1	1	0

8vo bit en 0

SUMA BINARIA COMPLEMENTO A 2

CASO 2 y 3: SUMA DE 1 NÚMERO POSITIVO Y 1 NÚMERO NEGATIVO:

- El resultado debe poseer el signo del que tenga mayor valor absoluto.

- Veremos 2 ejemplos uno que nos dará un resultado positivo y uno de valor negativo.

SUMA BINARIA COMPLEMENTO A 2

Caso 2 Suma de un número positivo y un número negativo:
Dado los números:

- A. 88_{10} = $01011000_{c\text{-}2}$
- B. -10_{10} = $11110110_{c\text{-}2}$

Obtener A+B

Como el valor absoluto de 88 es 88 y el valor absoluto de -10 es 10 esperamos que el resultado de esta suma arroje un valor **POSITIVO**. (>0)

Resolvemos:

SUMA BINARIA COMPLEMENTO A 2

- A. 88_{10} = $01011000_{c\text{-}2}$
- B. -10_{10} = $11110110_{c\text{-}2}$

1	1	1	1					
	0	1	0	1	1	0	0	0
+	1	1	1	1	0	1	1	0
	0	1	0	0	1	1	1	0

El 8vo bit da 0 como era de esperarse ya que el valor absoluto del número positivo era mayor al del número negativo

NOTA: Si quisiéramos agregar bits para resolver el acarreo seguiríamos llevándonos uno hasta q nos quedáramos sin bits

SUMA BINARIA COMPLEMENTO A 2

Caso 3 Suma de un número negativo y uno positivo:
Dado los números:

- A. -88_{10} = $10101000_{c\text{-}2}$
- B. 10_{10} = $00001010_{c\text{-}2}$

Obtener A+B

Como el valor absoluto de -88 es 88 y el valor absoluto de 10 es 10 esperamos que el resultado de esta suma arroje un valor **NEGATIVO**. (< 0)

Resolvemos:

SUMA BINARIA COMPLEMENTO A 2

- A. -88_{10} = $10101000_{c\text{-}2}$
- B. 10_{10} = $00001010_{c\text{-}2}$

				1				
	1	0	1	0	1	0	0	0
+	0	0	0	0	1	0	1	0
	1	0	1	1	0	0	1	0

El 8vo bit da 1 como era de esperarse ya que el valor absoluto del número negativo era mayor al del número positivo

NOTA: Si quisiéramos agregar bits para resolver el acarreo seguiríamos llevándonos uno hasta q nos quedáramos sin bits

SUMA BINARIA COMPLEMENTO A 2

<u>CASO 4: SUMA DE 2 NÚMEROS NEGATIVOS:</u>

- El resultado debe ser siempre un número negativo por lo que el MSB debe ser siempre 1.

- Es importante saber el rango de números (decimales) que admite la cantidad de bits que estamos utilizando.

- Si el número alcanzado estuviera fuera del rango permitido se producirá un OVERFLOW que derivará en un resultado erróneo por sobrecarga de dato

SUMA BINARIA COMPLEMENTO A 2

Ejemplo Suma de dos números negativos:

Dado los números:

- A. -88_{10} = 10101000_{c-2}
- B. -10_{10} = 11110110_{c-2}

Obtener A+B

Resolvemos:

SUMA BINARIA COMPLEMENTO A 2

- A. -88_{10} = $10101000_{c\text{-}2}$
- B. -10_{10} = $11110110_{c\text{-}2}$

1	1	1						
	1	0	1	0	1	0	0	0
+	1	1	1	1	0	1	1	0
1	1	0	0	1	1	1	1	0

Acarreo que se descarta

El 8vo bit da 1 como era de esperarse ya que al sumar un negativo con otro negativo el resultado debe ser negativo también

NOTA: Si quisiéramos agregar bits para resolver el acarreo seguiríamos llevándonos uno hasta q nos quedáramos sin bits

ARITHMETIC OVERFLOW

- Hasta ahora en las sumas que hemos hecho no hemos tenido problemas con los acarreos ya que siempre tuvimos cuidado de que los resultados estuvieran dentro de los límites que nos impone la cantidad de bits que disponemos.

- Cuando esto no se cumple nos encontramos con una sobrecarga aritmética o un **ARITHMETIC OVERFLOW**

- A continuación veremos 2 ejemplos en donde el acarreo cambia el resultado o incluso el signo de la operación por estar trabajando en la cantidad incorrecta de bits

ARITHMETIC OVERFLOW
Suma de números binarios sin signo

- A. 200_{10} = 11001000_2
- B. 175_{10} = 10101111_2

1				1				
	1	1	0	0	1	0	0	0
+	1	0	1	0	1	1	1	1
	0	1	1	1	0	1	1	1

El resultado de 200 + 175 es 375. Este último no puede ser representado en con 8 bits en binario sin signo y por este error el resultado ha sido:

$$\mathbf{01110111_2 = 119_{10}}$$

ARITHMETIC OVERFLOW
Suma de números binarios en C-2

- A. 99_{10} = $01100011_{C\text{-}2}$
- B. 53_{10} = $00110101_{C\text{-}2}$

	1	1			1	1	1	
	0	1	1	0	0	0	1	1
+	0	0	1	1	0	1	0	1
	1	0	0	1	1	0	0	0

El resultado de 99 + 53 es 152. Este último no puede ser representado con 8 bits en complemento a 2 y por este error el resultado ha sido:

$$\mathbf{10011000_{C\text{-}2} = -104_{10}}$$