CIRCUITOS LOGICOS DIGITALES

UPC

Universidad Peruana de Ciencias Aplicadas

Laureate International Universities®

SISTEMAS DE NUMERACIÓN, CONVERSIONES, OPERACIONES Y REPRESENTACIÓN DE NÚMEROS CON SIGNO

CICLO ACADÉMICO: 2024-I

SISTEMAS DE NUMERACIÓN, CONVERSIONES, OPERACIONES Y REPRESENTACIÓN DE NÚMEROS CON SIGNO.

CONTENIDO:

- Sistemas de numeración y conversiones entre las bases 10, 2, 8 y 16.
- Representaciones y operaciones de números con signo.

SISTEMAS DE NUMERACIÓN

DEFINICIÓN:

- Un sistema de numeración se define como una metodología para representar números mediante un conjunto de símbolos capaces de representar valores numéricos.
- La base de los sistemas de numeración se define como la cantidad de símbolos diferentes que se emplean para representar los valores numéricos.
- Cada símbolo del sistema de numeración recibe el nombre de dígito.

SISTEMAS DE NUMERACIÓN

EJEMPLOS DE SISTEMAS DE NUMERACIÓN:

- **Sistema decimal o de base 10**

  Los números se representan usando diez dígitos:

  {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

- **Sistema binario o de base 2**

  Los números se representan usando dos dígitos:

  {0, 1}

SISTEMAS DE NUMERACIÓN

EJEMPLOS DE SISTEMAS DE NUMERACIÓN:

- **Sistema octal o de base 8**

  Los números se representan usando ocho dígitos :

  {0, 1, 2, 3, 4, 5, 6, 7}

- **Sistema hexadecimal o de base 16**

  Los números se representan usando dieciséis dígitos :

  {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

- Conversión de un número del sistema binario al sistema decimal - **método de la suma de pesos**.

$$1 \quad 1 \quad 0 \quad 1 \quad 1_2$$

$$2^4 + 2^3 + 0 + 2^1 + 2^0 = 16 + 8 + 2 + 1$$

$$= 27_{10}$$

MSB **LSB**

$$1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1_2 =$$

$$2^7 + 0 + 2^5 + 2^4 + 0 + 2^2 + 0 + 2^0 = 181_{10}$$

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

- Conversión de un número en sistema binario al sistema decimal - **método de la suma de pesos.**

$1011_2 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 11$

$110010_2 = 1x2^5 + 1x2^4 + 0x2^3 + 0x2^2 + 1x2^1 + 0x2^0$
$= 32 + 16 + 2 = 50$

$101.11_2 = 1x2^2 + 0x2^1 + 1x2^0 + 1x2^{-1} + 1x2^{-2}$
$= 4 + 1 + 0.5 + 0.25 = 5.75$

PUNTO BINARIO

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

- Conversión de un número binario al sistema decimal – Ejemplo

1. Convierta el número 100011011011_2 en su equivalente decimal.
2. ¿Cuál es el peso del MSB de un número de 16 bits?

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

- Conversión de un número decimal al sistema binario

Existen 2 métodos:

Método 1 – Suma de Pesos – Proceso inverso:

El numero decimal a convertir se expresará como una suma de potencias de 2 partiendo del peso del MSB y luego completando los demás pesos de forma descendente en las posiciones que correspondan en el número binario hasta llegar al LSB.

$$\begin{aligned} 45_{10} &= 32 + 8 + 4 + 1 = 2^5 + 0 + 2^3 + 2^2 + 0 + 2^0 \\ &= 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1_2 \end{aligned}$$

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

Método 2 – divisiones sucesivas entre 2:

El número decimal se dividirá de forma sucesiva entre 2 hasta obtener un cociente igual a 0. Los residuos obtenidos en cada operación de división formarán parte de los dígitos del número binario los cuales se ordenarán de manera ascendente (desde el LSB hasta el MSB).

$$\frac{25}{2} = 12 + \boldsymbol{residuo\ 1}\ (\boldsymbol{LSB})$$

$$\frac{12}{2} = 6 + \boldsymbol{residuo\ 0}$$

$$\frac{6}{2} = 3 + \boldsymbol{residuo\ 0}$$

$$\frac{3}{2} = 1 + \boldsymbol{residuo\ 1}$$

$$\mathbf{1}\ (\boldsymbol{MSB})$$

$$\mathbf{25_{10} = 1\,1\,0\,0\,1_2}$$

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

Convertir 37 a binario por el método de la división entre 2 sucesiva.

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

Convertir 37 a binario por el método de la división entre 2 sucesiva.

$$\frac{37}{2} = 18 + \boldsymbol{residuo}\ \mathbf{1}(\boldsymbol{LSB})$$

$$\frac{18}{2} = 9 + \boldsymbol{residuo}\ \mathbf{0}$$

$$\frac{9}{2} = 4 + \boldsymbol{residuo}\ \mathbf{1}$$

$$\frac{4}{2} = 2 + \boldsymbol{residuo}\ \mathbf{0}$$

$$\frac{2}{2} = 1 + \boldsymbol{residuo}\ \mathbf{0}$$

$$\mathbf{1}\ (\boldsymbol{MSB})$$

$$\mathbf{37_{10} = 100101_2}$$

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

La parte entera **se dividirá de forma sucesiva entre 2** hasta obtener un cociente igual a 0. Los residuos obtenidos en cada división formarán parte de los dígitos de la parte entera del número binario cuyos pesos aparecerán de forma ascendente desde el LSB hasta el MSB de dicho número binario En el caso de la parte fraccionaria, **esta se multiplicará de forma sucesiva por 2** hasta obtener un valor igual a 0. Los dígitos de acarreo generados por las multiplicaciones son los dígitos que forman la parte fraccionaria del número binario cuyos pesos aparecerán de forma descendente desde el MSB hasta el LSB.

Convertir el número 69.8125 a base 2

PUNTO BINARIO

Parte entera

69	**1**	LSB
34	**0**	
17	**1**	
8	**0**	
4	**0**	
2	**0**	
1	**1**	MSB

Parte fraccionaria

0.8125x2 = 1 + 0.625		MSD
0.625x2 = 1 + 0.25		
0.25x2 = 0 + 0.5		
0.5x2 = 1 + 0.0		LSD

$69.8125 = 1000101.1101_2$

SISTEMAS DE NUMERACIÓN

ALCANCE DEL CONTEO:

Recuerde lo siguiente: Si empleamos un numero binario de *N* bits, podemos contar hasta 2^N números decimales diferentes que van desde 0_{10} hasta $[2^N-1]_{10}$

Por ejemplo, para $N = 4$ tenemos un numero binario de 4 bits y podemos contar en binario desde 0000_2 hasta 1111_2 o también en decimal desde 0_{10} hasta 15_{10} lo cual dará un total de 16 números decimales distintos.

En el ejemplo anterior el valor decimal de mayor magnitud es $[2^4-1]_{10} = 15_{10}$ y hay $2^4 = 16$ números decimales diferentes.

En general, podemos indicar que:

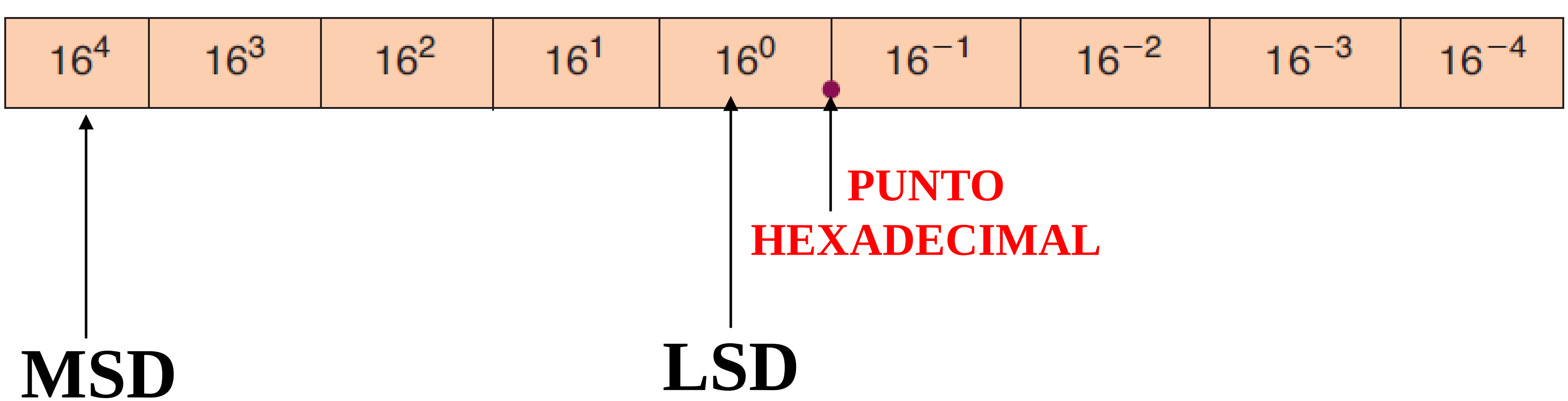
Si usamos un numero binario de N bits podemos representar números decimales que van desde 0_{10} hasta $[2^N-1]_{10}$
Es decir, un total de 2^N números diferentes.

SISTEMAS DE NUMERACIÓN

SISTEMA DE NUMERACIÓN HEXADECIMAL

Utiliza la base 16 y a consecuencia de ello posee 16 símbolos o dígitos. Utiliza los símbolos del 0_{16} al 9_{16} y las letras A_{16} (10_{10}), B_{16} (11_{10}), C_{16} (12_{10}), D_{16} (13_{10}), E_{16} (14_{10}) y F_{16} (15_{10}) que cubren los 16 dígitos.

Las posiciones de los dígitos se ponderan como potencias en base 16 a diferencia del sistema de numeración decimal o binario cuyas posiciones se ponderan como potencias en base 10 y 2 respectivamente.



SISTEMAS DE NUMERACIÓN

EQUIVALENCIAS ENTRE LOS SISTEMAS DE NUMERACIÓN BINARIO, DECIMAL Y HEXADECIMAL

Hexadecimal	Decimal	Binario
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

- Conversión de un número hexadecimal al sistema decimal - **método de suma de pesos**.

Es posible convertir un numero hexadecimal a su equivalente decimal gracias a que la posición de cada digito del numero hexadecimal tiene un peso que es equivalente a una potencia de la posición en base 16. El LSD tiene un peso de 16^0 =1; la siguiente posición de digito tiene un peso de 16^1=16; la siguiente tiene un peso de 16^2=256; y así sucesivamente hasta llegar al MSD.

$$\begin{aligned} 356_{16} &= 3 \times 16^2 + 5 \times 16^1 + 6 \times 16^0 \\ &= 768 + 80 + 6 \\ &= 854_{10} \end{aligned}$$

$$\begin{aligned} 2AF_{16} &= 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 \\ &= 512 + 160 + 15 \\ &= 687_{10} \end{aligned}$$

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

La parte entera se dividirá de forma sucesiva entre 16 hasta obtener un cociente igual a 0. Los residuos obtenidos en cada operación de división formarán parte de los dígitos de la parte entera del número binario siendo el 1er y último resto los LSB y MSB de dicho número binario.

- Conversión de un número decimal al sistema hexadecimal - **método de la división sucesiva entre 16**

Convierta 423 a hexadecimal

Convierta 214 a hexadecimal

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

- Conversión de un número decimal al sistema hexadecimal(método de la división sucesiva entre 16)

Convierta 423 a hexadecimal

$$\frac{423}{16} = 26 + \boldsymbol{residuo\ 7\ (LSB)}$$

$$\frac{26}{16} = 1 + \boldsymbol{residuo\ 10}$$

$$\mathbf{1}\ (\boldsymbol{MSB})$$

$$\boldsymbol{423_{16} = 1A7_{16}}$$

Convierta 214 a hexadecimal

$$\frac{214}{16} = 13 + \boldsymbol{residuo\ 6\ (LSB)}$$

$$\mathbf{13}\ (\boldsymbol{MSB})$$

$$\boldsymbol{214_{16} = D6_{16}}$$

SISTEMAS DE NUMERACIÓN

CONVERSIÓN ENTRE SISTEMAS DE NUMERACIÓN

- Conversión de un número binario al sistema hexadecimal – **método de agrupación de dígitos**

$$11001101_2 = \underbrace{1100}\ \underbrace{1101} = CD_{16}$$

$$100111_2 = \underbrace{0010}\ \underbrace{0111} = 27_{16}$$

Para la conversión de números binarios enteros a hexadecimal, también enteros, se deberá generar agrupaciones de 4 dígitos comenzando desde el LSD.

ARITMETICA DE NUMEROS BINARIOS SIN SIGNO

SUMA BINARIA

- 4 reglas básicas
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 1 0

Bits de acarreo

- Ejemplo:

```
     1   1
     1   1   1      # binario 3-bits
+    0   0   1      # binario 3-bits
-------------------
 1   0   0   0      # binario 4-bits
```

ARITMETICA DE NUMEROS BINARIOS SIN SIGNO

SISTEMAS DE NUMERACIÓN

- Operaciones Aritméticas en base 2.

Suma:

$$\begin{array}{r} 11_2 (3) + \\ \underline{110_2} (6) \\ 1001_2 \end{array} \qquad \begin{array}{r} 1001_2 (9) + \\ \underline{1111_2} (15) \\ 11000_2 \end{array}$$

$$\begin{array}{r} 1+ \\ \underline{1} \\ 10+ \\ \underline{1} \\ 11 \end{array}$$

ARITMETICA DE NUMEROS BINARIOS SIN SIGNO

<u>MULTIPLICACIÓN BINARIA</u>

- 4 reglas básicas
  - 0 x 0 = 0   0 x 1 = 0
  - 1 x 0 = 0   1 x 1 = 1
- Ejemplo:

$$
\begin{array}{cccccccl}
 & & & 1 & 1 & 1 & & \text{\# binario 3-bits} \\
 & & \text{x} & 1 & 0 & 1 & & \text{\# binario 3-bits} \\
\hline
 & & & 1 & 1 & 1 & + & \\
 & & 0 & 0 & 0 & & & \\
 & 1 & 1 & 1 & & & & \\
\hline
1 & 0 & 0 & 0 & 1 & 1 & & \text{\# binario 6-bits}
\end{array}
$$

ARITMETICA DE NUMEROS BINARIOS SIN SIGNO

SISTEMAS DE NUMERACIÓN

- Operaciones Aritméticas en base 2.

Multiplicación:

$$
\begin{array}{r}
11\ \times \\
\underline{10} \\
00\ + \\
\underline{11} \\
110
\end{array}
\qquad\qquad
\begin{array}{r}
1101\times \\
\underline{110} \\
0000+ \\
1101 \\
\underline{1101} \\
1001110
\end{array}
$$

ARITMETICA DE NUMEROS BINARIOS SIN SIGNO

RESTA BINARIA

Regla de excepción: Para esta operación (0 – 1), al ser menor el minuendo que el sustraendo; el minuendo deberá prestarse un bit del digito contiguo la izquierda generándose un bit de acarreo negativo por lo que la operación quedará como (10 – 1 = 1).

Bits de acarreo Negativo

- 4 reglas básicas
  - 0 - 0 = 0
  - 1 - 1 = 0
  - 1 - 0 = 1
  - $\boxed{1\ 0 - 1 = 1}$
- Ejemplo:

```
  0 ~~1~~ 1 0   1      # binario 3-bits
-     0   1   1      # binario 3-bits
------------------
      0   1   0      # binario 3-bits
```

ARITMETICA DE NUMEROS BINARIOS SIN SIGNO

SISTEMAS DE NUMERACIÓN

❑ Operaciones Aritméticas en base 2.

Resta:

$$\begin{array}{r} 1011_2 (11) - \\ \underline{100_2} (4) \\ 111_2 \end{array} \qquad \begin{array}{r} 1001_2 (9) - \\ \underline{1111_2} (15) \\ ?? \end{array}$$

En este enunciado el minuendo es menor que el sustraendo. Por tal motivo, el resultado de esta operación dará un número negativo así que para ejecutar una operación con estas características se deberá representar el signo de dichos números.

ARITMETICA DE NUMEROS BINARIOS CON SIGNO

NÚMEROS CON SIGNO

- Hasta ahora sólo hemos utilizado binarios enteros positivos
  - con n bits $\Rightarrow 0 \leq N \leq 2^n-1$
- ¿Cómo representar números negativos?
  - 2 alternativas
    - *Signo-Magnitud*
    - *Complemento*

REPRESENTACION DE NUMEROS CON SIGNO

Para la representación del signo de un número binario se emplea <u>un bit extra</u> donde dicho bit extra, que puede ser 0 o 1, representan el signo (+) y (-), respectivamente. A continuación, se muestran algunas de las formas más básicas de representar un numero binario con signo:

1. Representación con signo (+) o (-) y magnitud (valor absoluto del número): **<u>signo-magnitud</u>**.
2. Representación con **<u>complemento a 1</u>**.
3. Representación con **<u>complemento a 2</u>**.
4. Representación por **<u>exceso</u>**.

REPRESENTACION DE NUMEROS CON SIGNO
SIGNO-MAGNITUD

- El primer bit representa el signo

  0 → signo positivo, 1 → signo negativo

- El resto de la cifra representa la magnitud (valor absoluto)

Signo	Magnitud			
S	MMM	...	MM	
n-1	n-2	...		0

- Ejemplos:

7_{10} Representación sin signo → 111_2

$+7_{10}$ Representación con signo positivo → $\boxed{0}111_2$

-7_{10} Representación con signo negativo → $\boxed{1}111_2$

Regla: El número de bits del número representado en signo-magnitud debe superar al número de bits del número que se desea representar.

P.e., $7_{10}=111_2$ representado en signo-magnitud tendrá 4 bits.

REPRESENTACION DE NUMEROS CON SIGNO
SIGNO-MAGNITUD

❖ Representación con signo y magnitud

Por ejemplo:

Representación con signo positivo ⇨ $(+17_{10})$

0	$10001_{S\text{-}M}$
signo	magnitud

Representación con signo negativo ⇨ (-17_{10})

1	$10001_{S\text{-}M}$
signo	magnitud

Aquí se ingresa el bit de signo

REPRESENTACION DE NUMEROS CON SIGNO
SIGNO-MAGNITUD

- Rango de números representables con *n* bits:

$$\text{sin signo}: \quad [2^n\text{-}1 \;..\; 0] \rightarrow \text{total} = 2^n$$

$$\text{signo-magnitud}: [-(2^{n-1}-1) \;..\; (2^{n-1}-1)] \rightarrow \text{total} = 2^n$$

- Ejemplo: 3 bits

Representación	Signo-Magnitud	Sin signo
000_2	0_{10}	0_{10}
001_2	1_{10}	1_{10}
010_2	2_{10}	2_{10}
011_2	3_{10}	3_{10}
100_2	-0_{10}	4_{10}
101_2	-1_{10}	5_{10}
110_2	-2_{10}	6_{10}
111_2	-3_{10}	7_{10}

Representación con signo positivo

Representación con signo negativo

Doble representación para el 0

OPERACIONES ARITMÉTICAS DE NUMEROS CON SIGNO SIGNO-MAGNITUD

DESBORDAMIENTO. Es una situación que se da cuando el resultado de una operación excede el rango de valores que puede representarse con el numero de bits disponibles para almacenar dicho resultado lo que ocasiona que se obtenga un resultado incorrecto.

- Suma y resta:

Caso 1: los números son de signo contrario.

- Se resta el mayor del menor. El signo resultante será el del mayor.

Representación S-M | Representación sin Signo | Representación S-M

- $4 - 3 \rightarrow \boxed{0}100 + \boxed{1}011 \rightarrow +\boxed{(100 - 011)} = + \quad \rightarrow \boxed{0}001$
- $-4 + 3 \rightarrow \boxed{1}100 + \boxed{0}011 \rightarrow -\boxed{(100 - 011)} = - \quad \rightarrow \boxed{1}001$

Caso 2: los números son del mismo signo

- Se suman ambos. El signo resultante será el de los operandos.
- Puede producirse $\boxed{\text{desbordamiento}}$
- $-2-3 \rightarrow \boxed{1}010 + \boxed{1}011 \rightarrow -(010 + 011) = -(101)$ => **correcto**
- $4 + 5 \rightarrow \boxed{0}100 + \boxed{0}101 \rightarrow +(100 + 101) = +(1001)$ => **DESBORDAMIENTO**

IMPORTANTE: EN CASO DE DESBORDAMIENTO, LA SITUACIÓN SE SOLUCIONA ADICIONANDO UN DIGITO A CADA UNO DE LOS OPERANDOS (UN 1 SI SE TRATA DE OPERACIONES CON NÚMEROS NEGATIVOS O UN 0 SI SE TRATA DE NUMEROS POSITIVOS).

REPRESENTACION DE NUMEROS CON SIGNO
SIGNO-MAGNITUD

❖ Representación mediante signo-magnitud

Características:

- Su representación decimal es fácil de interpretar.
- Negar un número o cambiar su signo supone cambiar también el bit de signo (0 ⇨ +, 1 ⇨ -).
- Con n bits el rango es: $-(2^{n-1}-1)$ a $+(2^{n-1}-1)$.
- Existen el +0 y el -0.
- Es incómodo para realizar operaciones aritmeticas.

REPRESENTACION DE NUMEROS CON SIGNO
COMPLEMENTO A 1 & COMPLEMENTO A 2

Características

- No es necesario distinguir entre la suma y la resta

- Existen dos tipos de representaciones:
  - Complemento a *1*
  - Complemento a *2*

REPRESENTACION DE NUMEROS CON SIGNO
COMPLEMENTO A 1

- Los números positivos se representan igual que en signo-magnitud. Los negativos se representan complementando o cambiando cada uno de los dígitos binarios.

- Rango : $[-(2^{n-1}-1) .. (2^{n-1}-1)] \rightarrow$ total $= 2^n$

- Ejemplos: n = 3 bits

$+3$ Representación con signo positivo ➡ $\boxed{0}11$ **Representación S-M**

-3 Representación con signo negativo ➡ $(\boxed{0}11)' = 100_{C1}$ **Representación C1**

Regla: El número de bits del número representado en C1 debe superar al número de bits del número negativo que se desea representar.
P.e., $-3_{10}=-11_2$ representado en C1 tendrá 3 bits (011)'.

Representación		C1
000		0_{10}
001		1_{10}
010		2_{10}
011		3_{10}
100	=(011)'	-3_{10}
101	=(010)'	-2_{10}
110	=(001)'	-1_{10}
111	=(000)'	-0_{10}

Doble representación para el 0

REPRESENTACION DE NUMEROS CON SIGNO
COMPLEMENTO A 1

❖ Representación con complemento a 1

Por ejemplo:

$(+17_{10})$

0	**10001**$_{S-M}$
signo	magnitud

(-17_{10})

1	**01110**$_{C1}$
signo	magnitud

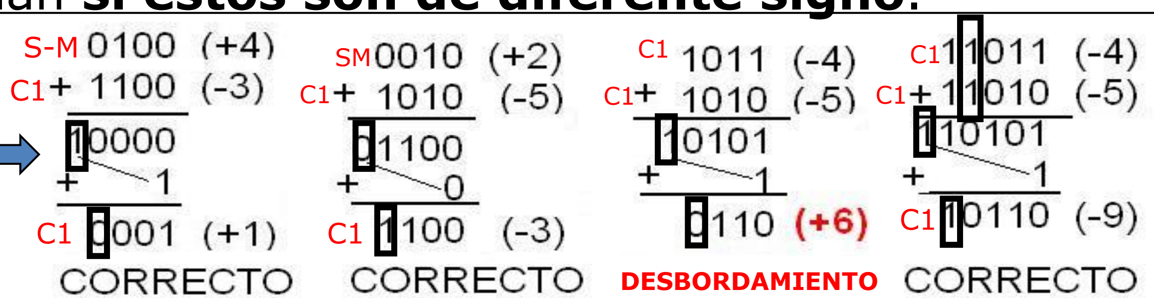
Aquí se complementa o cambian los dígitos del número binario con signo positivo para obtener su C1.

OPERACIONES ARITMÉTICAS DE NUMEROS CON SIGNO COMPLEMENTO A 1

<u>DESBORDAMIENTO</u>. Es una situación que se da cuando el resultado de una operación excede el rango de valores que puede representarse con el numero de bits disponibles para almacenar dicho resultado lo que ocasiona que se obtenga un resultado incorrecto.

- Suma y resta:
  - Indistintamente de la operación aritmética que se realizará, para obtener su resultado en C1 <u>los números binarios siempre se suman, y **TAMBIÉN** lo harán (se sumaran) los bits de acarreo los cuales se comportan como bits de **DESPLAZAMIENTO RECIRCULAR**.</u>
  - El <u>signo</u> de la suma resultante de los números enteros será:
    - **El (signo) de los números binarios** que se suman **si estos son del mismo signo**. Si el resultado de la operación excede el rango de valores con que puede representarse el numero, entonces ocurre **desbordamiento**.
    - **El (signo) del mayor de los dos números binarios** que se suman **si estos son de diferente signo**.

Giro con desplaza-
miento circular
del bit de
acarreo.



OPERACIONES ARITMÉTICAS DE NUMEROS CON SIGNO: COMPLEMENTO A 1

- Más casos de DESBORDAMIENTO: Como se aprecia en los 2 casos de **DESBORDAMIENTO**, el resultado de esta operación es incorrecta.

```
C1   1011  (-4)
C1   1010  (-5)
    ------
    10101
  +     1
    ------
     0110  (+6)
```

Giro con desplazamiento circular del bit de acarreo.

DESBORDAMIENTO

```
S-M    0100  (+4)
S-M +  0101  (+5)
      ------
      0 1001
   +       0
      ------
       1001  (-6)
```

DESBORDAMIENTO

<u>IMPORTANTE</u>: EN CASO DE <u>DESBORDAMIENTO</u>, LA SITUACIÓN SE SOLUCIONA ADICIONANDO A LA IZQUIERDA UN <u>1</u> SI SE SUMAN NÚMEROS NEGATIVOS O UN <u>0</u> SI SE SUMAN NÚMEROS POSITIVOS.

```
C1   1 1 0 1 1 +
C1   1 1 0 1 0
--------------------
   1 1 0 1 0 1 +
             1
--------------------
C1   1 0 1 1 0 = (-9)
```

```
   00100  (+4)
+  00101  (+5)
  001001
+      0
  ------
   01001  (+9)
```

OPERACIONES ARITMÉTICAS DE NUMEROS CON SIGNO: COMPLEMENTO A 1

❖ Suma y resta en complemento a 1

$$\begin{array}{lr} +5 & \boxed{0}101 \\ +2 & \underline{\boxed{0}010} \\ +7 & \boxed{0}111 \end{array}$$

$$\begin{array}{lr} +\ 8 & \boxed{0}1000 \\ -12 & \underline{\boxed{1}0011} \\ -\ 4 & \boxed{1}1011 \end{array}$$

$$\begin{array}{lr} -2 & \boxed{1}101 \\ -4 & \underline{\boxed{1}011} \\ & \mathbf{1}1000 \\ & \underline{1} \\ -6 & \boxed{1}001 \end{array}$$

Giro con desplazamiento circular del bit de acarreo

OPERACIONES ARITMÉTICAS DE NUMEROS CON SIGNO: COMPLEMENTO A 1

❖ Suma y resta en complemento a 1

$+6$ $\boxed{0}110$

$+4$ $\boxed{0}100$

-5 $\boxed{1}010$

DESBORDAMIENTO

-6 $\boxed{1}001$

-7 $\boxed{1}000$

10001

1

$+2$ 0010

$-13 = \boxed{1}1\,1\,0\,1$

Giro con desplazamiento circular del bit de acarreo

$\boxed{0}\,\boxed{0}\,1\,1\,0$ +

$\boxed{0}\,\boxed{0}\,1\,0\,0$

$0\,0\,1\,0\,1\,0$ +

0

$\boxed{0}\,1\,0\,1\,0$

Giro con desplazamiento circular del bit de acarreo

REPRESENTACION DE NUMEROS CON SIGNO
COMPLEMENTO A 1

❖ Representación con complemento a 1

Características

- Su representación decimal es medianamente fácil de interpretar.
- Negar un número o cambiar su signo supone cambiar el bit de signo y también los dígitos del número que representan su magnitud para obtener su complemento a 1.
- Con n bits el rango es: $-(2^{n-1}-1)$ a $+(2^{n-1}-1)$.
- Existe el +0 y el -0.
- Es cómodo para operar.

REPRESENTACION DE NUMEROS CON SIGNO
COMPLEMENTO A 2

- Los números positivos se representan igual que en signo-magnitud. Los negativos como $\boxed{2^n \text{ - el número.}}$

- Rango : $[-(2^{n-1}) .. (2^{n-1}-1)] \rightarrow$ total $= 2^n$

- Ejemplos: n = 4 bits

  $+7 \Rightarrow 0111$

  $-7 \Rightarrow (2^4 - 7)_{10} = (16 - 7)_{10} = 9_{10}$

  $9_{10} = 1001_{C2}$

Número		c a 2		
0_{10}	$\Rightarrow$	0 000	$\Rightarrow$	0
$+1_{10}$	$\Rightarrow$	0 001	$\Rightarrow$	1
-1_{10}	$\Rightarrow$	1 111	$\Rightarrow$	$15 = 2^4 - 1$
$+2_{10}$	$\Rightarrow$	0 010	$\Rightarrow$	2
-2_{10}	$\Rightarrow$	1 110	$\Rightarrow$	$14 = 2^4 - 2$
$+3_{10}$	$\Rightarrow$	0 011	$\Rightarrow$	3
-3_{10}	$\Rightarrow$	1 101	$\Rightarrow$	$13 = 2^4 - 3$
$+4_{10}$	$\Rightarrow$	0 100	$\Rightarrow$	4
-4_{10}	$\Rightarrow$	1 100	$\Rightarrow$	$12 = 2^4 - 4$
$+5_{10}$	$\Rightarrow$	0 101	$\Rightarrow$	5
-5_{10}	$\Rightarrow$	1 011	$\Rightarrow$	$11 = 2^4 - 5$
$+6_{10}$	$\Rightarrow$	0 110	$\Rightarrow$	6
-6_{10}	$\Rightarrow$	1 010	$\Rightarrow$	$10 = 2^4 - 6$
$+7_{10}$	$\Rightarrow$	0 111	$\Rightarrow$	7
-7_{10}	$\Rightarrow$	1 001	$\Rightarrow$	$9 = 2^4 - 7$
-8_{10}	$\Rightarrow$	1 000	$\Rightarrow$	$8 = 2^4 - 8$

- Operaciones aritméticas:

Parecido al complemento a 1 excepto que no se necesita realizar el giro con desplazamiento circular del bit de acarreo.

REPRESENTACION DE NUMEROS CON SIGNO
COMPLEMENTO A 2

❖ Representación con complemento a 2 mediante complemento a 1

Por ejemplo:

(+17)

0	10001_{C2}
signo	magnitud

(-17)

1	01111_{C2}
signo	magnitud

Aquí se complementa a 1 la expresión S-M, e inclusive su bit de signo y luego a ese resultado se le suma 1

$\boxed{1}$ 0 1 1 1 0 + (C1 de S-M)
1

$\boxed{1}$ 0 1 1 1 1 (C2)

OPERACIONES ARITMÉTICAS DE NUMEROS CON SIGNO: COMPLEMENTO A 2

- Otras formas de calcular el complemento a 2:

  C2(número) = C1(número) + 1

  Ejemplo: C2(-90)=C1(-90)+1=10100101=10100110

De Izquierda a derecha copiar todos los bits hasta encontrar el primer '1'. Después de ese bit (1) complementar el resto de bits.

- $+90_{C2} = 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0$

  $\downarrow\downarrow\ \downarrow\ \downarrow\downarrow\ \downarrow\ \downarrow$

  $-90_{C2} = 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0$

Hasta el 1er. '1' todos los bits son iguales

El 1er. '1' se mantiene

El resto de valores se complementa

OPERACIONES ARITMÉTICAS DE NUMEROS CON SIGNO: COMPLEMENTO A 2

$(2^4-3)=1101_2$

$$\begin{array}{rrl} & \boxed{0}100 & (+4) \\ + & \boxed{1}101 & (-3) \\ \hline & \boxed{0}001 & (+1) \end{array}$$

CORRECTO

$(2^4-5)=1011_2$

$$\begin{array}{rrl} & \boxed{0}010 & (+2) \\ + & \boxed{1}011 & (-5) \\ \hline & \boxed{1}101 & (-3) \end{array}$$

CORRECTO

$(2^4-4)=1100_2$

$(2^4-5)=1100_2$

$$\begin{array}{rrl} & \boxed{1}100 & (-4) \\ + & \boxed{1}011 & (-5) \\ \hline & 0111 & (+7) \end{array}$$

DESBORDAMIENTO

$$\begin{array}{rrl} & \boxed{1}\boxed{1}100 & (-4) \\ + & \boxed{1}\boxed{1}011 & (-5) \\ \hline \cancel{1} & \boxed{1}0111 & (-9) \end{array}$$

CORRECTO

IMPORTANTE: EN CASO DE DESBORDAMIENTO, LA SITUACIÓN SE SOLUCIONA ADICIONANDO A LA IZQUIERDA UN 1 SI SE SUMAN NÚMEROS NEGATIVOS O UN 0 SI SE SUMAN NÚMEROS POSITIVOS.

Debido a que no se necesita realizar el giro con desplazamiento circular del bit de acarreo, este bit se omite en la expresión final.

Tener en cuenta lo siguiente:

C1 = C2 – 1
S-M = Bit Signo | (C1)'

OPERACIONES ARITMÉTICAS DE NUMEROS CON SIGNO: COMPLEMENTO A 2

❖ Suma y resta en complemento a 2

$$\begin{array}{lr} +5 & \boxed{0}101 \\ +2 & \underline{\boxed{0}010} \\ +7 & \boxed{0}111 \end{array}$$

$(2^5-12)=10100_2$

$(2^5-4)=11100_2$

$$\begin{array}{lr} +\ 8 & \boxed{0}1000 \\ -12 & \underline{\boxed{1}0100} \\ -\ 4 & \boxed{1}1100 \end{array}$$

$(2^4-2)=1110_2$

$(2^4-4)=1100_2$

$(2^4-6)=1010_2$

$$\begin{array}{lr} -2 & \boxed{1}110 \\ -4 & \underline{\boxed{1}100} \\ -6 & \cancel{1}\boxed{1}010 \end{array}$$

Debido a que no se necesita realizar el giro con desplazamiento circular del bit de acarreo, este bit se omite en la expresión final.

OPERACIONES ARITMÉTICAS DE NUMEROS CON SIGNO: COMPLEMENTO A 2

Desbordamiento

IMPORTANTE: EN CASO DE DESBORDAMIENTO, LA SITUACIÓN SE SOLUCIONA ADICIONANDO A LA IZQUIERDA UN 1 SI SE SUMAN NÚMEROS NEGATIVOS O UN 0 SI SE SUMAN NÚMEROS POSITIVOS.

+6	$\boxed{0}110$
+4	$\boxed{0}100$
-2	1010

$(2^4-6)=1010_2$

$(2^4-7)=1001_2$

−6	$\boxed{1}010$
−7	$\boxed{1}001$
+3	$\cancel{1}0011$

DESBORDAMIENTO

```
0  0  1  1  0 +
0  0  1  0  0
--------------------------
0  1  0  1  0 - (C2)
```

```
   0  0  1  1 - (C2)
            1
----------------------
   0  0  1  0  (C1)
1  1  1  0  1  (S-M)
```

Debido a que no se necesita realizar el giro con desplazamiento circular del bit de acarreo, este bit se omite en la expresión final.

Tener en cuenta lo siguiente:

C1 = C2 – 1
S-M = Bit Signo | (C1)'

REPRESENTACION DE NUMEROS CON SIGNO
COMPLEMENTO A 2

❖ Representación con complemento a 2

Características

- Su representación decimal no es fácil de interpretar.
- Negar un número o cambiar su signo supone cambiar el bit de signo y también los dígitos del número para obtener su complemento a 2.
- Con n bits el rango es: $-(2^{n-1})$ a $(2^{n-1}-1)$.
- Sólo existe el 0.
- Es cómodo para operar.