

95.57/TB023 Organización del Computador

U1 – Sistemas de Numeración

Agenda

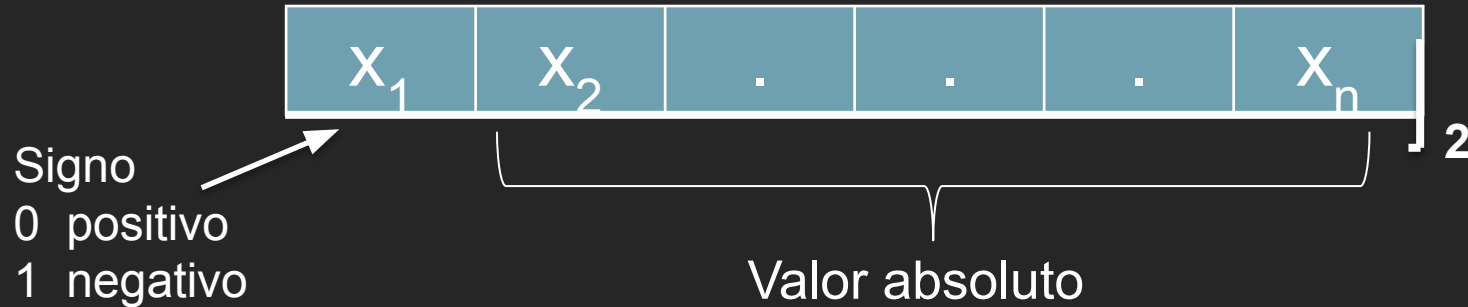
- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

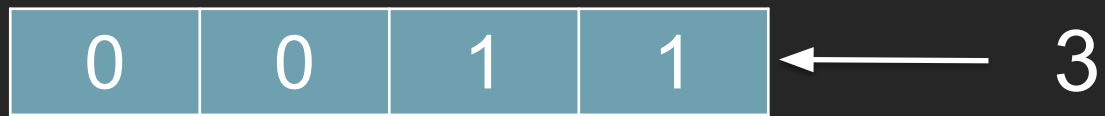
Bit de signo y valor absoluto

Base = 2 Precisión = n



Ejemplo:

Base = 2 Precisión = 4



Bit de signo y valor absoluto

Paso por paso

Base = 2 precisión = 4

Representar -6

- 1) Ver signo para determinar el primer bit: **1** (por ser negativo)
- 2) Pasar valor absoluto a base 2: $|-6_{10}| = 6_{10} = \mathbf{110}_2$
- 3) Concatenar los bits: **1110**

Indicar número almacenado en **1101**

- 1) Ver primer bit para determinar el signo: negativo (por ser **1**)
- 2) Pasar a base 10 los bits descartando el primero: $\mathbf{101}_2 = 5_{10}$
- 3) Indicar el número según signo y valor obtenidos: -5

Antes de seguir veamos B^n

$B^n = ?$

base
cantidad de
símbolos

precisión
cantidad de
dígitos



$B * B * \dots * B$
 $= B^n = \# \text{ total de números representables}$

Ejemplos:

$$B = 10 \\ n = 2 \quad B^n = 10^2 = 100$$


[00, 01, ..., 99]

100

$$B = 2 \\ n = 4 \quad B^n = 2^4 = 16 \rightarrow$$

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

16

Bit de signo y valor absoluto

Rango de Representación

Minimo: $-(2^{n-1}-1)$

Maximo: $2^{n-1}-1$

Ventajas

- Rango simétrico

Desventajas

- Doble representación del 0
- No permite operar aritméticamente

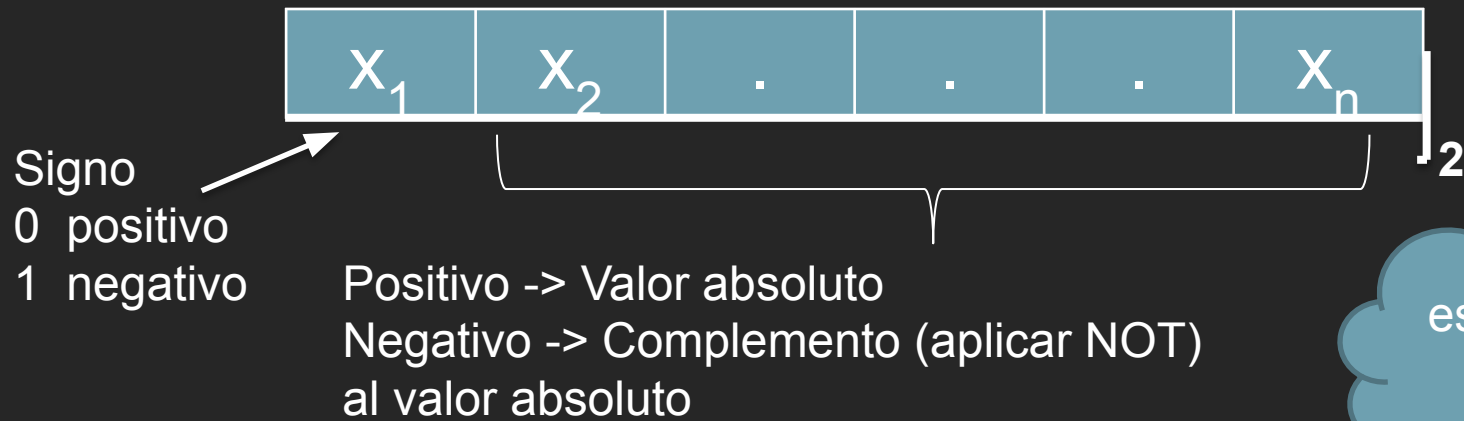
0001		1
+1001	+	-1
----		--
1010	≠	0

Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

Complemento a 1

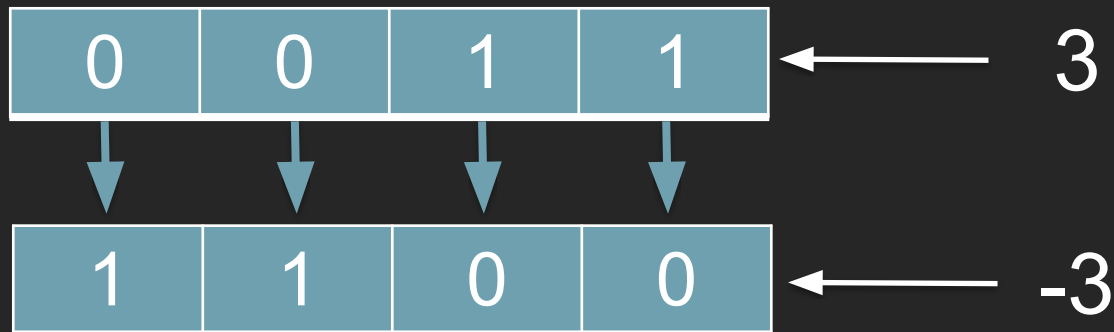
Base = 2 Precisión = n



NOT
es cambiar
1 por 0
0 por 1

Ejemplo:

Base = 2 Precisión = 4



Complemento a 1

Paso por paso

Base = 2 precisión = 4

Representar -6

1. Pasar valor absoluto a base 2: $|-6_{10}| = 6_{10} = 110_2$
2. Completar con 0 a izquierda hasta completar n: 0110
3. Si es negativo, complementar (hacer NOT): 1001

Indicar número almacenado en 1101

1. Si primer bit es 1 (es negativo), complementar (hacer NOT): 0010
2. Pasar a base 10: $0010_2 = 2_{10}$
3. Indicar el número según signo y valor obtenidos: -2

Complemento a 1

Rango de Representación

Minimo: $-(2^{n-1}-1)$

Maximo: $2^{n-1}-1$

Desventajas

- Doble representación del 0

Ventajas

- Rango simétrico
- Permite operar aritméticamente sumando el "end-around carry"

11		
0101		5
+1110	+	-1
----		--
0011	≠	4
+ 1		

0100	=	4

Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

Complemento a la Base

Base = B Precisión = n

$$\begin{aligned}\text{Rep}(x_b) &= x_b && \text{si } x \geq 0 \\ \text{Rep}(x_b) &= \text{Cb}(|x_b|) && \text{si } x < 0\end{aligned}$$

Ejemplos:

$$B = 10 \quad n = 2 \Rightarrow B^n = 10^2 = 100$$

$$\text{Rep}(3) = 03$$

$$\text{Rep}(-3) = \text{Cb}(3) = 100 - 3 = 97$$

$$\text{Rep}(-1) = \text{Cb}(1) = 100 - 1 = 99$$

$$\text{Rep}(-50) = \text{Cb}(50) = 100 - 50 = 50$$

$$\text{Rep}(50) \Rightarrow \text{NO SE PUEDE}$$

\Rightarrow 49 es el mayor positivo representable

¿ Pero que es Cb?

$$\begin{aligned}\text{Cb}(r) + r &= B^n \\ \Rightarrow \text{Cb}(r) &= B^n - r\end{aligned}$$

Notar que:

Si

k es complemento de r

$$\Rightarrow k + r = B^n$$

\Rightarrow r es complemento de k

Complemento a la Base

Rango de Representación

Mínimo: $-(B^n/2)$

Máximo: $(B^n/2) - 1$

Desventajas

- Rango asimétrico (un negativo más)

Ventajas

- Única representación del 0
- Permite operar aritméticamente

Complemento a la Base

Permite operar aritméticamente:

Sumas (Con B = 10 y n = 2)

$$5 + 2 \quad (\text{es } 7)$$

$$05 + 02$$

$$\begin{array}{r} 00 \\ 05 \\ +02 \\ \hline 07 \end{array} \quad \text{(A)}$$

$$5 + (-2) \quad (\text{es } 3)$$

$$05 + 98$$

$$\begin{array}{r} 11 \\ 05 \\ + 98 \\ \hline 03 \end{array} \quad \text{(B)}$$

$$-5 + 2 \quad (\text{es } -3)$$

$$95 + 02$$

$$\begin{array}{r} 00 \\ 95 \\ + 02 \\ \hline 97 \end{array} \quad \rightarrow -3 \quad \text{(C)}$$

$$-5 + (-2) \quad (\text{es } -7)$$

$$95 + 98$$

$$\begin{array}{r} 11 \\ 95 \\ + 98 \\ \hline 93 \end{array} \quad \rightarrow -7 \quad \text{(D)}$$

Restas: al plantear A-B se transforma en A+Bcomp

$$5 - 2 \quad (\text{es } 3)$$

$$\begin{array}{r} 05 - 02 \\ 05 + \text{Cb}(2) \\ = 05 + 98 \\ = 03 \end{array} \quad \text{(B)}$$

$$5 - (-2) \quad (\text{es } 7)$$

$$\begin{array}{r} 05 - 98 \\ 05 + \text{Cb}(98) \\ = 05 + 02 \\ = 07 \end{array} \quad \text{(A)}$$

$$-5 - 2 \quad (\text{es } -7)$$

$$\begin{array}{r} 95 - 02 \\ 95 + \text{Cb}(2) \\ = 95 + 98 \\ = 93 \end{array} \quad \text{(D)}$$

$$-5 - (-2) \quad (\text{es } -3)$$

$$\begin{array}{r} 95 - 98 \\ 95 + \text{Cb}(98) \\ = 95 + 02 \\ = 97 \end{array} \quad \text{(C)}$$

Complemento a la Base

Permite operar aritméticamente: CONCLUSIÓN

$$A - B$$

Se trabaja como

$$A + \text{Comp}(B)$$

***** NO IMPORTA EL SIGNO DE B ******

Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

Complemento a 2

Base = 2 Precisión = n

$$\begin{aligned}\text{Rep}(x_{10}) &= x_2 & \text{si } x \geq 0 \\ \text{Rep}(x_{10}) &= \text{NOT}(|x_2|) + 1 & \text{si } x < 0\end{aligned}$$

Ejemplos:

B = 2 n = 4

$$\text{Rep}(3) = 0011$$

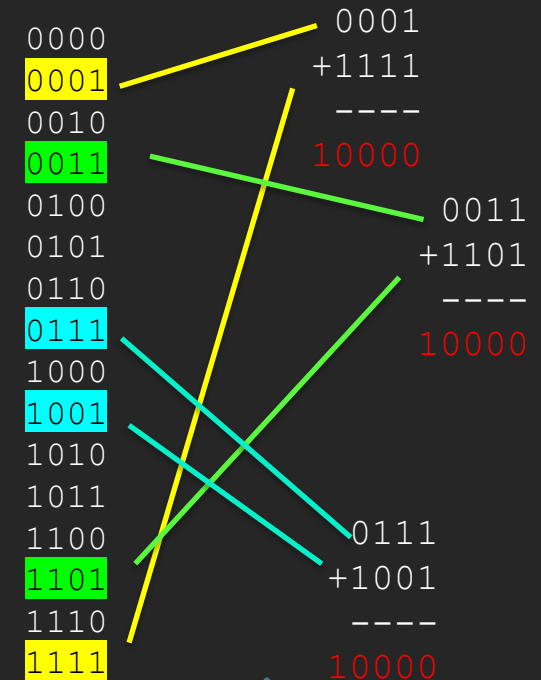
$$\begin{aligned}\text{Rep}(-3) &= \text{NOT}(0011) + 1 \\ &= 1100 + 1 \\ &= 1101\end{aligned}$$

$$\begin{aligned}\text{Rep}(-1) &= \text{NOT}(0001) + 1 \\ &= 1110 + 1 \\ &= 1111\end{aligned}$$

$$\begin{aligned}\text{Rep}(-8) &= \text{NOT}(1000) + 1 \\ &= 0111 + 1 \\ &= 1000\end{aligned}$$

Rep(8) => NO SE
PUEDE
=> $7 = 0111_2$ es el mayor
positivo representable

Veamos...



Complemento a 2

Paso por paso

Base = 2 precisión = 4

Representar -6

1. Pasar valor absoluto a base 2: $|-6_{10}| = 6_{10} = 110_2$
2. Completar con 0 a izquierda hasta completar n: 0110
3. Si es negativo, complementar (hacer NOT + 1): $1001 + 1 = 1010$

Indicar número almacenado en 1101

1. Si primer bit es 1 (es negativo), complementar (hacer NOT+1): $0010 + 1 = 0011$
2. Pasar a base 10 los bits: $0011_2 = 3_{10}$
3. Indicar el número según signo y valor obtenidos: -3

Complemento a 2

Rango de Representación

Mínimo: $-(2^{n-1})$

Máximo: $2^{n-1} - 1$

Desventajas

- Rango asimétrico (un negativo más)

Ventajas

- Única representación del 0
- Permite operar aritméticamente.
Aplica la misma mecánica de resolver $X-Y$ como $X+Cb(Y)$

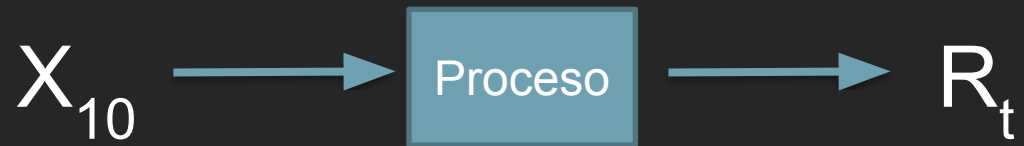
Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso (*)
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754 (*)

Formato y Configuración

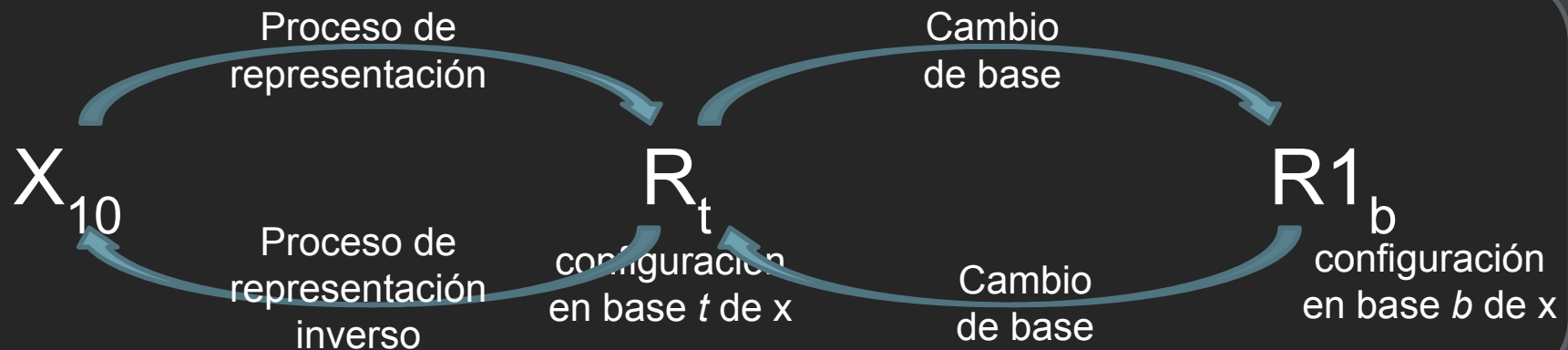
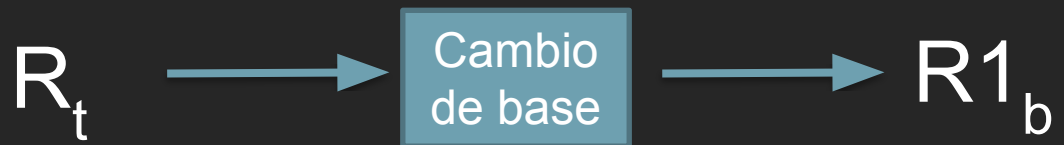
Formato:

Representación
computacional de
un número



Configuración:

Expresión en una
determinada base de un
número en un formato

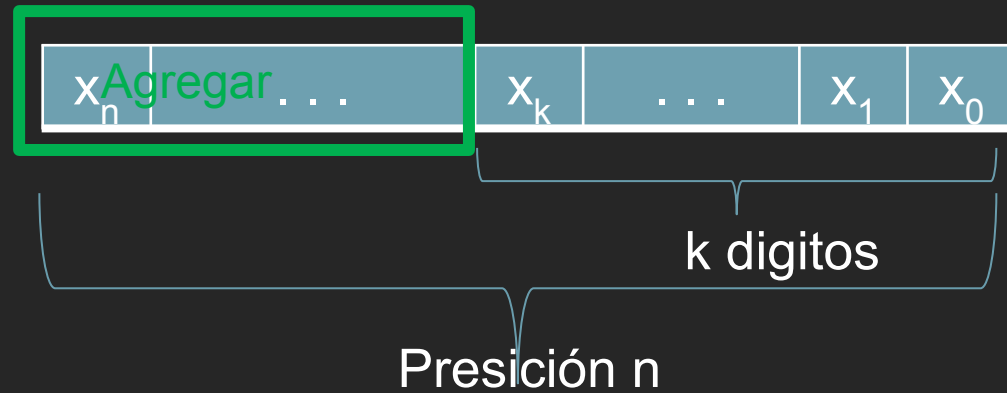


Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

Expansión y Truncamiento

Expansión:



Truncamiento:



Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

Binario de punto fijo sin signo

Base = 2 Precisión = n Enteros positivos

Como almacenar un número

- 1) Pasar el nro a base 2
- 2) Completar con 0 a izquierda hasta alcanzar n dígitos

Como recuperar un número almacenado

Pasos anteriores en orden inverso

Rango de representación

Mínimo: 0

Máximo: $2^n - 1$

Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

Binario de punto fijo con signo

Base = 2 Precisión = n Enteros positivos y negativos

Es la implementación del método complemento a 2

Como almacenar un número

- 1) Pasar el nro a base 2
- 2) Completar con 0 a izquierda hasta alcanzar n digitos
- 3) Si el nro es negativo, complementar usando método de "complemento a 2" (Not +1)

Como recuperar un número almacenado

- 1) Si el primer bit es 1 (es negativo), complementar.
- 2) Quitar 0 a izquierda.
- 3) Pasar a base 10 y colocar el signo que corresponda.

Binario de punto fijo con signo

Validación Overflow en operaciones aritméticas

B=2 n=4

Resolver 7 + 1

$$\begin{array}{lcl} 7_{10} & = & 0111_2 \\ 1_{10} & = & 0001_2 \end{array}$$

```

0111  Ultimos 2 acarreo distintos
0111 => OVERFLOW
+ 0001
----
1000

```

Resolver 7 - 1

$$\begin{array}{lcl} 7_{|10} & = & 0111_{|2} \\ 1_{|10} & = & 0001_{|2} \end{array}$$

Hallo $C(1)$ para hacer $7+C(1)$

$$\begin{array}{r} \text{NOT}(0001) = 1110 \\ + \quad 1 \\ \hline 1111 \end{array}$$

Ahora sumo

```

1111    Ultimos 2 acarreos iguales
 0111    => VALIDO
+ 1111
  ---
 0110

```


Agenda

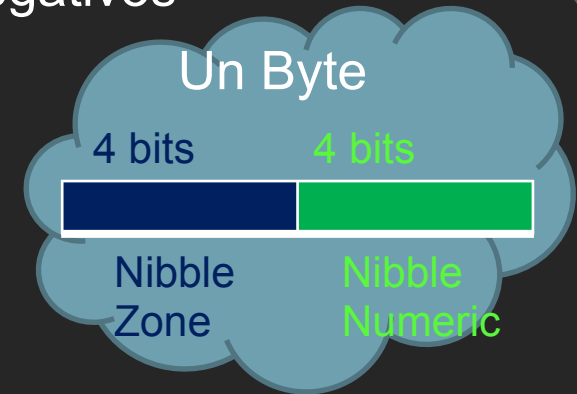
- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

BCD Empaquetado

Base = 16 Precisión = n Enteros positivos y negativos

Como almacenar un número

- 1) Pasar el nro a base 10
- 2) Colocar c/digito en los nibbles dejando libre el último (el de la derecha)
- 3) Colocar en el último nibble el signo siendo
C, A, F o E para positivos
B o D para negativos



Ej. n=3 +123₁₀ --> 00123^A₁₆ -456₁₀ --> 00456^B₁₆

Como recuperar un número almacenado

- 1) Tomar cada digito de los nibbles (excepto el último) para armar la cadena en base 10
- 2) Colocar el signo según el dígito del último nibble

Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

Zoneado

Base = 16 Precisión = n Enteros positivos y negativos

Como almacenar un número

- 1) Pasar el nro a base 10
- 2) Colocar c/digito en los nibbles numeric
- 3) Colocar una F en cada nibble zone excepto en el último
- 4) Colocar en el último nibble zone el signo siendo
C, A, F o E para positivos
B o D para negativos

Ej. n=4 +123₁₀ --> F0F1F2A3₁₆ -456₁₀ --> F0F4F5B6₁₆

Como recuperar un número almacenado

- 1) Tomar cada digito de los nibbles numeric para armar la cadena en base 10
- 2) Colocar el signo según el dígito del último nibble zone

Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

ASCII / EBCDIC / UNICODE

- Representación en forma digital/numéricas de los caracteres
 - ASCII (American Standard Code for Information Interchange)
 - 7 bits □ ASCII Básico
 - 8 bits □ ASCII Extendido
 - EBCDIC (Extended Binary Coded Decimal Interchange Code)
 - 8 bits
 - UNICODE Consortium
 - Codificación Universal/Estandar de los Caracteres
 - Consorcio Unicode □ Unicode 14.0
 - UTF-8 / UTF-16 / UTF-32T
 - <https://home.unicode.org/>

ASCII / EBCDIC / UNICODE

Hex	Dec	EBCDIC	ASCII
23	35		#
30	48		0
31	49		1
42	66		B
62	98		b
7B	123	#	{
82	130	b	
C0	192	{	
C2	194	B	
F0	240	0	
F1	241	1	

[Documento tablas ASCII & EBCDIC](#)

Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

Exceso a la base

Base = B Precisión = n

$\text{Rep}(x_b) = x_b + \text{exceso}$ (para todo x)

Con B = 2

$$\begin{aligned}\text{Exceso} &= 2^n / 2 \\ &= 2 * 2 \dots * 2 / 2 \\ &= 2^{n-1}\end{aligned}$$

Ejemplos:

B = 10 n = 2

$$\Rightarrow B^n / 2 = 10^2 / 2 = 50$$

$$\text{Rep}(3) = 3 + 50 = 53$$

$$\text{Rep}(-3) = -3 + 50 = 47$$

$$\text{Rep}(0) = 0 + 50 = 50$$

$$\text{Rep}(-50) = -50 + 50 = 0$$

$\text{Rep}(-51)$ NO SE PUEDE (da -1)

$$\text{Rep}(49) = 49 + 50 = 99$$

$\text{Rep}(50)$ NO SE PUEDE (da 100)

¿ Pero que es el exceso?

Es $B^n / 2$

Entendamos por qué:

Con B=10 y n = 1

$B^n = 10$ valores posibles

¿Cuál sería el rango de números a representar "más justo"?

-5 -4 -3 -2 -1 0 1 2 3 4

10 en total

Cuánto hay q sumar como mínimo a cada negativo para que "desaparezca" el signo?

5

Que termina siendo $B^n / 2$

Números a representar

-5 -4 -3 -2 -1 0 1 2 3 4

Como se representan

0 1 2 3 4 5 6 7 8 9

Agenda

- **Métodos de representación de números negativos**
 - Bit de signo y valor absoluto
 - Complemento a 1
 - Complemento a la base
 - Complemento a 2
 - Exceso
- **Conceptos de Formato y Configuración**
- **Conceptos de Expansión y Truncamiento**
- **Formatos de representación de números enteros**
 - Binario de punto fijo sin signo
 - Binario de punto fijo con signo
 - BCD Empaquetado
 - Zoneado
- **Formatos de representación caracteres**
 - ASCII
 - EBCDIC
 - UNICODE
- **Formatos de representación números decimales**
 - Binario punto Flotante IEEE 754

Binario punto Flotante IEEE 754

- Notación Científica

$$S M \times B^E$$

- $-/+ 765,987 \times 10^{-3}_{10}$
- $-/+765987 \times 10^{-6}_{10}$
- $-/+ 7,65987 \times 10^{-1}_{10}$
- $-/+ 0,0765987 \times 10^1_{10}$

- Mantisa Normalizada

$$0 < M < B$$

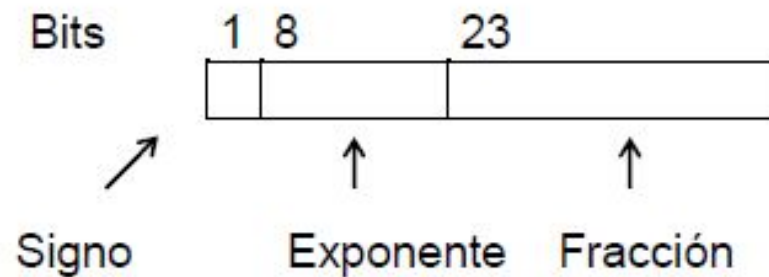
¿Cuál es el dígito de la cifra significativa en Binario?

Binario punto Flotante IEEE 754

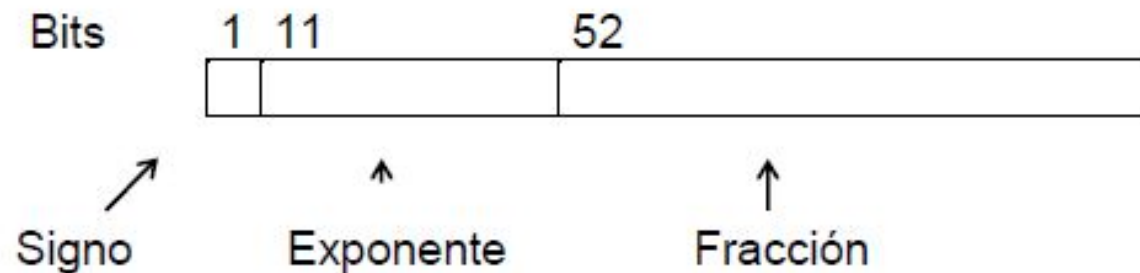
- 80s □ Institute of Electrical and Electronics
- Precisión
 - Simple (32) / Doble (64) / Extendida (128)
- Representación
 - Signo
 - Exponente
 - Mantisa
- Exponente en Exceso
 - 127 / 1023 / 16383
- Mantisa normalizada

Binario punto Flotante IEEE 754

Simple precisión



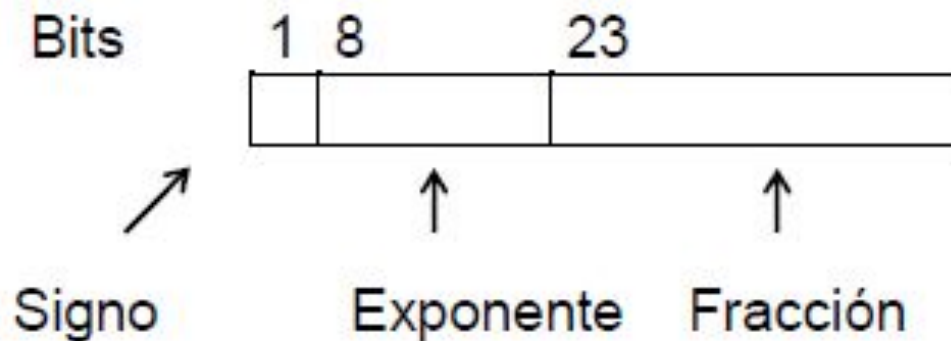
Doble precisión



Binario punto Flotante IEEE 754

- Precisión Simple

Simple precisión



- Exceso de 127
 - $E_{\text{Exceso}} = \text{Exp} + 127_{10}$
- Mantisa Normalizada
 - 1 implícito

Binario punto Flotante IEEE 754

$-123,456_{10}$

1) Paso de base 10 \square base 2

$123_{10} \square 1111011_2$

$0,456_{10} \square 01110100_2$

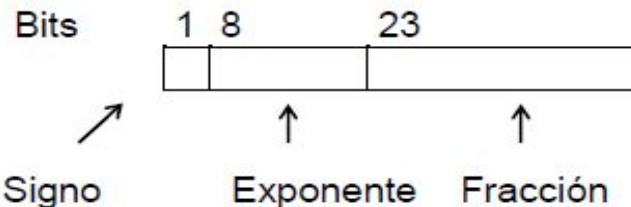
$-123,456_{10} \square -1111011,01110100_2$

2) Normalizo

$-1,11101101110100 \times 10^{110}_2$

3) Almaceno

Simple precisión



Binario punto Flotante IEEE 754

3) Almaceno

Signo = 1

EExceso

$$6 + 127 = 133_{10}$$

$$133_{10} \square 10000101_2$$

Mantisa

$$1,11101101110100_2$$

4) Nro. Final

$$1 \mid 10000101 \mid 11101101110100 \textcolor{red}{0000000000}_2$$

Binario punto Flotante IEEE 754

Normalizado	\pm $0 < \text{Exp} < \text{Max}$	Cualquier patrón de bits
Desnormalizado	\pm 0	Cualquier patrón de bits $\neq 0$
Cero	\pm 0	0
Infinito	\pm 111...1	0
NAN	\pm 111...1	Cualquier patrón de bits $\neq 0$

