

TEÓRICO



Key Color

TÍTULO (0,6 mm)

SUBTÍTULO (0,4 mm)

SUBTÍTULO (0,4 mm)

Def : 

Prop : 

Ejemplo:

Dem :

Ejercicio

2 parciales \rightarrow 1° mitad cursado (aprox Semana 8)
 \rightarrow 2° ante última semana del cuatrimestre

listafiuba6670@gmail.com

SISTEMAS DE REPRESENTACIÓN NUMÉRICA

Sist. Numérico $\left\{ \begin{array}{l} \text{conj. símbolos (desarrollo histórico: monedas, nudos en una cuerda, simbología...)} \\ \text{reglas para su organización (sistemas aditivos y sist. posicionales)} \end{array} \right.$

egipcio
(va sumando)

árabe
↳ # limitada de símbolos
↳ decimal (base 10)

SISTEMA NUMÉRICO POSICIONAL

quede definidos por :

- símbolos
- # símbolos
- peso de cada posición
- # posiciones

Ejercicio: Representar 421_{10} en 4 bases:

a) Base 3

b) Octal

c) Base 2 $\rightarrow \begin{array}{cccccccc} 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array} \Bigg|_2 = 421_{10}$

d) Hexadecimal (0...9, A...F) $\rightarrow \begin{array}{ccc} 16^2 & 16 & 1 \\ 0 & 2 & 4 \end{array}$

$$N_{10} = \sum_{i=0}^n d_i b^i$$

SISTEMA BINARIO

8 bits $2^8 = 256$ 00000000

$2^8 - 1 = 255$ 11111111

16 bits $2^{16} = 65536$

$2^{16} - 1 = 65535 \rightarrow \text{max int}$

32 bits $2^{32} = 4GB = 4.294.967.296$

↓
 Los microprocesadores de 32b no usan memorias de más de 4GB porque no lee algo mayor

64 bits $2^{64} = 18.446.744.073.709.551.616$

REPRESENTACIÓN DE NÚMEROS CON PARTE FRACCIONARIA

1) Sistemas de Punto fijo

$$\sum_{i=0}^N d_i b^i + \sum_{i=1}^n d_i b^{-i}$$

- Puede aplicarse a una base cualquiera
- Cómo convertir un número en base 10 a otra base
- Cómo convertir un número a base 10
- Precisión de la conversión

2) Sistemas de Punto Flotante

$$0,32 \cdot 2 = 0,64$$

$$0,64 \cdot 2 = 1,28$$

$$0,28 \cdot 2 = 0,56$$

$$0,56 \cdot 2 = 1,12$$

así hasta el 0000... que significa que no hay error

$0,75_{10} \rightarrow$ base 2 es exacto.

Criterio de paro

$$1010,1010 \mid \dots$$

error que cometo = 2^{-4}

REPRESENTACIÓN DE ENTEROS CON SIGNO

- Magnitud y signo
 - Complemento a la base menos 1
 - Complemento a la base
- } Métodos

-25 \rightarrow Complemento : represento lo que le falta para llegar al 99
 \hookrightarrow M y S

00	}	COMPLEMENTO A 9
01		
02		
...		
74 \rightarrow -25		
75 \rightarrow -24		

lo llevamos al binario: Complemento a 1 \rightarrow se obtiene negando bit a bit

$$\begin{array}{r} 0011 +3 \\ 1100 -3 \end{array} \quad (\text{negación})$$

Complemento a dos (el de los microprocesadores)

$$\begin{array}{r} -3 \rightarrow \text{C2} : \quad 10000 \quad (16) \\ - \quad 0011 \\ \hline + \quad 1100 \\ \hline \quad \quad 1 \\ \hline 1101 \rightarrow -3 \text{ C2} \end{array}$$

$$\begin{array}{r} -1 \rightarrow \text{C2} : \quad 0001 \rightarrow \text{represento} \\ 1110 \rightarrow \text{niego} \\ \hline \quad 1 \rightarrow \text{sumo 1} \\ \hline 1111 \rightarrow -1 \text{ C2} \end{array}$$

CONVERTIR A BASE 10 NÚMEROS EN C21 y C22

$$\begin{array}{r} \text{C2:} \quad 10010011 \\ + \quad 01101100 \rightarrow \text{niego} \\ \hline \quad \quad 1 \rightarrow \text{sumo} \\ \hline 01101101 \rightarrow \text{calculo pesos} \\ 64 \quad 32 \quad 8 \quad 4 \quad 1 \rightarrow 109 \Rightarrow 10010011_2 = -109_{10} \end{array}$$

No hay forma de darse cuenta si una sec. de bits viene en negativo o positivo.

Me lo tienen que decir. Ídem si un número es C21 ó C22.

SUMA DE NÚMEROS BINARIOS

$\begin{array}{r} 1 \rightarrow \text{CARRY} \\ \begin{array}{r} 01110101 \\ 11010110 \\ \hline 01001011 \end{array} \end{array}$ <p>Resultado incorrecto</p>	}	<p>la suma no entra en 8 bits</p>	$\begin{array}{r} 01010110 \\ 11010010 \\ \hline 00101000 \end{array}$ <p>no hay overflow</p>	<p>(C22)</p> <table border="0"> <tr><td>decimal \rightarrow</td><td>86</td></tr> <tr><td>\rightarrow</td><td>-46</td></tr> <tr><td></td><td><u>40</u></td></tr> </table> <p>no hay error, y sin embargo me dio un carry</p>	decimal \rightarrow	86	\rightarrow	-46		<u>40</u>
decimal \rightarrow	86									
\rightarrow	-46									
	<u>40</u>									

El carry es una condición de error para números sin signo

El overflow es una condición de error para números con signo.

El overflow se da cuando entra un número y sale otro \neq .

C CARRY
V OVERFLOW
Z ZERO
N NEGATIVE

FLAGS / INDICADORES \rightarrow van con cero o uno

también es el flag S de "signo"

29/03/2022

"Hexadecimal" = "43h" = "0x43" \rightarrow en un esquema hexadecimal, un complemento a la

$$43h = 10 \cdot 16^1 + 3 \cdot 16^0 = 10100011_2$$

es un sist
número
sin signo

base sería que llegue al 100

"sin signo"

$$ss \ 10100011 = 143$$

$$C2 \ 10100011 =$$

$$01011100 \rightarrow \text{niego}$$

$$\underline{\quad\quad\quad 1 \rightarrow \text{suma 1}}$$

$$01011101 = -93 = -1011101$$

$$43h + DAh \Rightarrow \begin{array}{r} 110100011 \\ + 11011010 \\ \hline 01111101 \end{array}$$

carry

$$C=1$$

V, Z, N \rightarrow los 4 flags que resultan de la op.

el bit
de signo

se fue de rango? \rightarrow ss?

si \Rightarrow si \rightarrow quedó en carry

no \Rightarrow si \rightarrow entran 2 positivos y sale un negativo

$$V=1$$

El resultado \neq cero \Rightarrow $Z=0$

El resultado \in positivo \Rightarrow $N=0$

PUNTO FIJO

$$13,75 \text{ en binario} \rightarrow 1101,1100$$

$$\left. \begin{array}{l} 13,32 \rightarrow 0,32 \cdot 2 = 0,64 \\ 0,64 \cdot 2 = 1,28 \\ 0,28 \cdot 2 = 0,56 \\ 0,56 \cdot 2 = 1,12 \end{array} \right\} \Rightarrow 13,32 = 1101,0101$$

tengo el proceso en este punto porque tengo un sistema de 8 bits de los cuales tengo 4 significativos enteros y 4 significativos decimales

hay un error de 2^{-4} (cota)

$$1101,0101 \mid \dots \left\{ \begin{array}{l} 1111 \dots \text{hay un error de } 2^{-4} \\ 0000 \dots \text{hay un error de } 0 \end{array} \right.$$

$$\begin{array}{r} 1101,1100 \\ + 0110,1001 \\ \hline 0001,1111 \end{array}$$

MULTIPLICACIÓN Y DIVISIÓN DE NÚMEROS ENTEROS

$\begin{array}{r} 101101 \rightarrow 45 \\ 010110 \rightarrow 22 \end{array}$ \in número redondeado de 45

- desplazo a derecha una posición y eso es equivalente a dividir por 2
 - si desplazo a izquierda es como multiplicar por 2
 - si quiero multiplicar / dividir por 8 desplazo a izquierda / derecha 3 posiciones
- método para operar con potencias de 2