

66.70 Estructura del Computador

Sistemas numéricos

Sistemas para la representación de números

- ✓ ¿Qué número representa “**112**” ?
- ✓ Números en la tecnología digital

Sistemas para la representación de números

Para representar números necesitamos símbolos y una forma de organizarlos

- Desarrollo histórico: marcas en bastones, nudos en una cuerda,...
- Problema con los valores altos

Sistemas para la representación de números

Solución al problema de valores altos:

- 1) *Con números pequeños se van agregando unidades por medio de marcas*
 - 2) *Cuando se alcanza un determinado número se hace una marca distinta*
- Históricamente el “punto de corte” usado por diversas culturas fue el 10 (cantidad de dedos en las manos)

Símbolos en el sistema egipcio



vara
1



talón
10



cuerda
100



flor
1 000



dedo
10 000



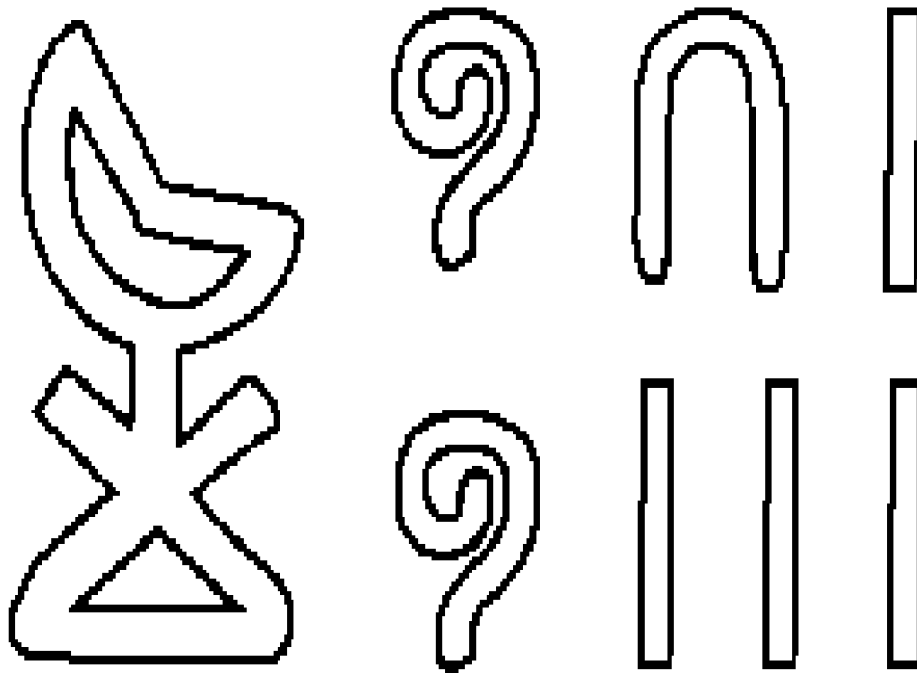
pez
100 000



hombre asustado
1 000 000

Sistemas de Numeración **Aditivos**

Sistema egipcio: 1214



			
vara 1	talón 10	cuerda 100	flor 1 000
			
dado 10 000	pez 100 000	hombre asustado 1 000 000	

Sistemas de Numeración Aditivos

Sistema griego

I	𐌲	Δ	𐌺	H	𐌸	X	𐌶	M
1	5	10	50	100	500	1000	5000	10000

X	X	X	𐌸	H	H	Δ	Δ	Δ	𐌲	I
3000	+	500	+	200	+	30	+	5	+	2
										= 3737

Sistemas de Numeración Aditivos

Han utilizado sistemas aditivos las civilizaciones:

- Egiptia
- sumeria (de **base 60, sexagesimal**)
- hitita, cretense,
- azteca (de base 20)
- romana <
- las alfabéticas de los griegos, armenios, judíos y árabes

Sistemas de Numeración Aditivos

Los números parecen palabras compuestas por letras

=> las palabras tienen un valor numérico (hay que sumar las cifras que corresponden a las letras que las componen)

Aparece una nueva suerte de disciplina mágica que estudiaba la relación entre los números y las palabras.

En algunas sociedades como la judía y la árabe, que utilizaban un sistema similar, el estudio de esta relación ha tenido una gran importancia y ha constituido una disciplina aparte: la kábala, que persigue fines místicos y adivinatorios.

Símbolos del sistema árabe

٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩

0 1 2 3 4 5 6 7 8 9

El sistema árabe (decimal)

- ✓ No es aditivo sino **posicional** con un **número limitado de símbolos**
- Desarrollado en India antes del siglo VII e introducido en Europa por los árabes.
- Babilonios, chinos y mayas en distintas épocas llegaron al mismo principio
- Entre el sistema actual y el de los Indios sólo hay diferencias en la forma que escribimos los 9 dígitos y el cero

Distintas versiones de los símbolos del sistema numérico decimal

Europeo	0	1	2	3	4	5	6	7	8	9
Arábico-Índico	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
Arábico-Índico Oriental (Persa y Urdu)	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
Devanagari (Hindi)	०	१	२	३	४	५	६	७	८	९
Tamil		௦	௧	௨	௩	௪	௫	௬	௭	௮

El sistema decimal no es el único sistema numérico posicional

Un sistema numérico posicional queda definido por:

- ✓ Símbolos disponibles
- ✓ Peso de cada posición

Analizar los siguientes ejemplos:

- Base 10
- Base cualquiera (base 3, octal, hexadecimal. etc.)

Representar $42|_{10}$ en diferentes bases

Conversión entre diferentes bases

Casos:

- Conversión de cualquier base a base 10
- Conversión de base 10 a otra base
- Conversión entre dos bases diferentes de 10
- Bases potencias de otras bases

Conversión entre bases - Métodos

- A base 10 -> sumatoria
- De base 10 a otra base -> Divisiones sucesivas
- Estimación en base a los pesos a incluir en la sumatoria
- Base que es potencia de otra base:
 - Agrupar y convertir cada grupo en un dígito
 - Desagrupar dígito a dígito

Conversión entre bases

Binary (base 2)	Octal (base 8)	Decimal (base 10)	Hexadecimal (base 16)
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F



Menor la base -> mayor cantidad de dígitos

- Representar 2532 en binario

Elección de un sistema para representar números enteros

(a) Símbolos

(b) Pesos

(c) Cantidad de dígitos

■ Rango representable

- Cuántos valores distintos?
- Cuál es el valor máximo?
- Cuál es el valor mínimo?

Ej.; Procesador de 16 bits

Sistema de punto fijo

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

Alternativa: PUNTO FLOTANTE

Ej.; Procesador de 16 bits

Sistemas para representar números enteros con signo

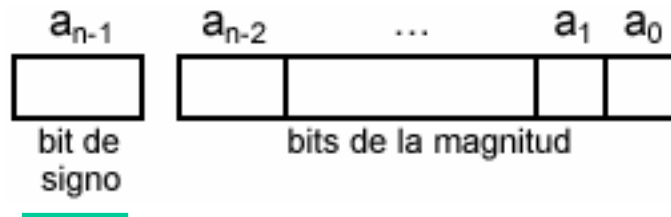
- *Magnitud y signo*
- *Complemento a la base menos 1*
- *Complemento a la base*

Sistemas para representar signo

<i>Decimal</i>	<i>Two's Complement</i>	<i>Ones' Complement</i>	<i>Signed Magnitude</i>
-8	1000	—	—
-7	1001	1000	1111
-6	1010	1001	1110
-5	1011	1010	1101
-4	1100	1011	1100
-3	1101	1100	1011
-2	1110	1101	1010
-1	1111	1110	1001
0	0000	1111 or 0000	1000 or 0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111

Representación en magnitud y signo

- Forma de representación



+ N \rightarrow Idem binario puro

- N \rightarrow Primer bit a izq es 1

- Rango representable

$$(-2^{n-1} + 1)_{10} \leq x \leq (2^{n-1} - 1)_{10}$$

Con 4 bits

1111	-7
1110	-6
1101	-5
1100	-4
1011	-3
1010	-2
1001	-1
1000 or 0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Representación en complemento a la base menos 1

En binario (base 2) => “COMPLEMENTO A 1”

- Forma de representación

+ N → Idem binario puro

- N → $C_{b-1}(N) = b^n - 1 - N$
b: base n: cant. de dígitos

En binario:

- N → $C_1(N) = 2^n - 1 - N$

Se puede obtener haciendo la resta o invirtiendo bit a bit

- Rango representable

$$(-2^{n-1} + 1)_{10} \leq x \leq (2^{n-1} - 1)_{10}$$

¿Cuál es el rango representable con 8 bits, 16 bits y 32 bits ?

Con 4 bits

1000	-7
1001	-6
1010	-5
1011	-4
1100	-3
1101	-2
1110	-1
1111 or 0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Representación en complemento a la base

En binario (base 2) => “COMPLEMENTO A 2”

- Forma de representación

+ N → Idem binario puro

- N → $C_b(N) = b^n - N$

b: base n: cant. de dígitos

En binario:

- N → $C2(N) = 2^n - N$

Se puede obtener haciendo la resta

o invirtiendo bit a bit y sumando 1

← Basado en
complemento a 1

- Rango representable

$$(-2^{n-1})_{10} \leq x \leq (2^{n-1} - 1)_{10}$$

¿Cuál es el rango representable con 8 bits, 16 bits y 32 bits ?

Con 4 bits

1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Sistemas para representar signo

<i>Decimal</i>	<i>Two's Complement</i>	<i>Ones' Complement</i>	<i>Signed Magnitude</i>
-8	1000	—	—
-7	1001	1000	1111
-6	1010	1001	1110
-5	1011	1010	1101
-4	1100	1011	1100
-3	1101	1100	1011
-2	1110	1101	1010
-1	1111	1110	1001
0	0000	1111 or 0000	1000 or 0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111

Convertir a base 10 números en complemento a 1 o números en complemento a 2

- Si el bit más significativo es 0

- ✓ El número es positivo
- ✓ Se convierte como si estuviera en binario puro (sumatoria de pesos)

- Si el bit más significativo es 1

- ✓ El número es negativo
- ✓ Fue obtenido complementando su módulo

Invierto ese proceso
calculando su complemento

¿ Justificación ?

Tengo su módulo
Convierto por sumatoria de pesos

¿ Cómo puedo distinguir si se trata de un entero con signo o sin signo ?
¿ Cómo puedo distinguir si se trata de complemento a 1 o complemento a 2 ?

Suma de números binarios

Sistema numérico:

- 8 bits
- Enteros sin signo

$$\begin{array}{r} 01110101 \\ + 11010110 \\ \hline ???????? \end{array}$$

Sistema numérico:

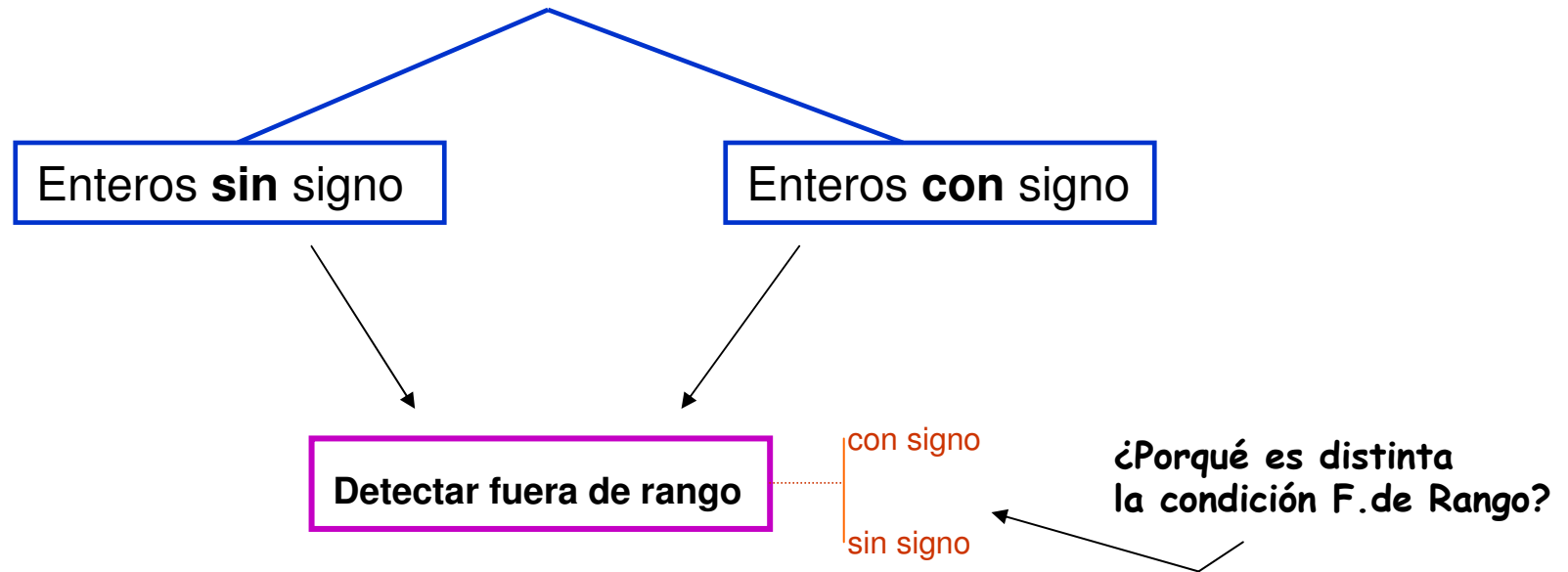
- 8 bits
- Enteros con signo
- Repres. en compl. a 2

$$\begin{array}{r} 01010110 \\ + 11010010 \\ \hline ???????? \end{array}$$

Resultado:

- Suma
- Se fue de rango?

Suma de números binarios



Indicadores (*flags*)

- C Carry
- V Overflow
- Z Cero
- N Signo
- P Paridad

*Operación **resta** en binario*

- ☐ *Forma directa*
- ☐ *Como suma del complemento*
- ☐ *“Borrow”*

álgebra de números VS. computadoras

Ley asociativa

El álgebra dice que:

$$a + (b + c) = (a + b) + c$$

Qué dice la
computadora?



$a = 90$
 $b = 105$
 $c = -50$

Procesador de 8 bits

*Suma de dos o más números definidos con
distinta cantidad de bits*



• **Enteros sin signo**

Enteros con signo

“EXTENSIÓN” DEL SIGNO

Suma de Números con parte fraccionaria

(Punto fijo)

Ejemplo:

$$\begin{array}{r} 3.25 \quad 0011,0100 \\ + \\ 10.75 \quad 1010,1100 \\ \hline \end{array}$$

Suma de **Números con parte fraccionaria**

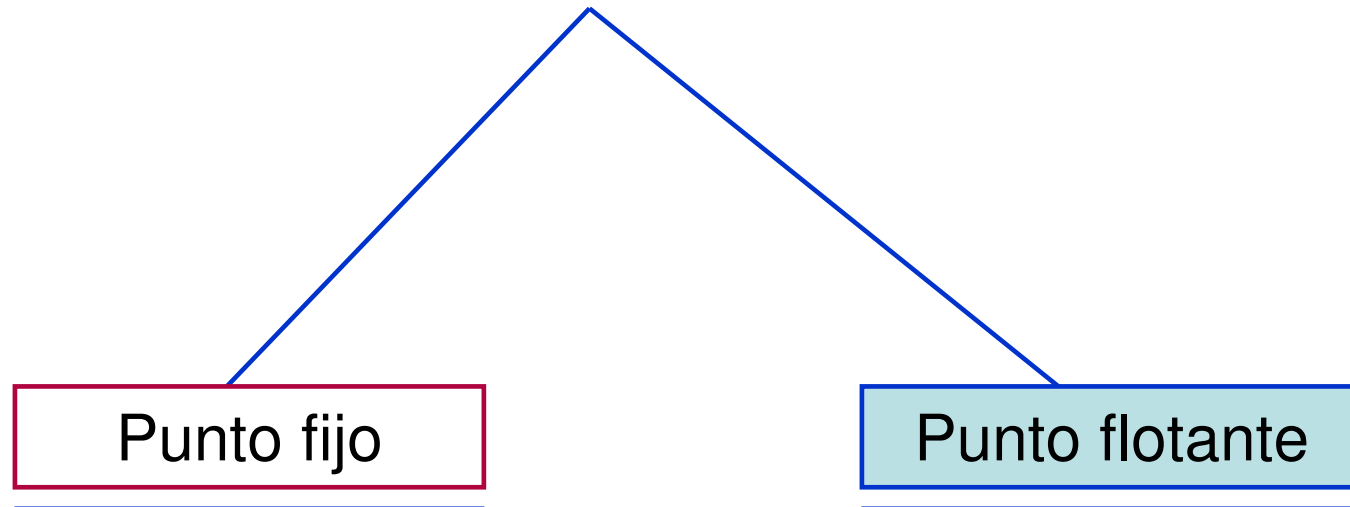
(Punto fijo)

Ejemplo:

$$\begin{array}{r} 3.25 \quad 0011,0100 \\ + \\ 10.75 \quad 1010,1100 \\ \hline \end{array}$$

- Opera como en números enteros
- Se implementa operando con números enteros y definiendo un factor de escala fijo

Representación de **Números con parte fraccionaria**



Otras operaciones básicas con números enteros

Caso general de

- Multiplicación
- División

→ *Dentro de pocas semanas...*

Multiplicación y división

... mediante desplazamientos a derecha e izquierda

- ✓ Implementación sencilla
- ✓ Alta velocidad de proceso