

# Representación y operaciones

Conceptos sobre formatos de almacenamiento y manejo de la información en computadoras.

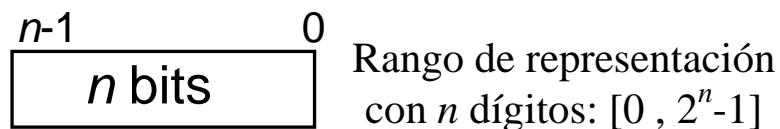
- Formatos numéricos de representación.
- Convenios de almacenamiento.
- Operaciones aritméticas y lógicas de procesadores reales.
- Conceptos avanzados de interrupciones.
- Lenguajes ensambladores y proceso de carga de programas.

## Tipos de datos que un computador puede procesar

- Bits ..... 0 ó 1
- Números {
  - Enteros sin signo (no negativos) ..... 27432
  - Enteros con signo ..... 31421  
-12311
  - Fraccionarios en coma fija ..... 0,9721221
  - Fraccionarios en coma flotante .....  $2,376 \times 10^{12}$
- Texto ..... "Esto es una cadena de texto"
- Instrucciones ..... **LD .A, [ /37 ][ .X]**

# Números enteros

## Enteros sin signo



Ejemplos:

$$n=8: B'00100100 = 36_{10)}$$

3. Complemento a dos:

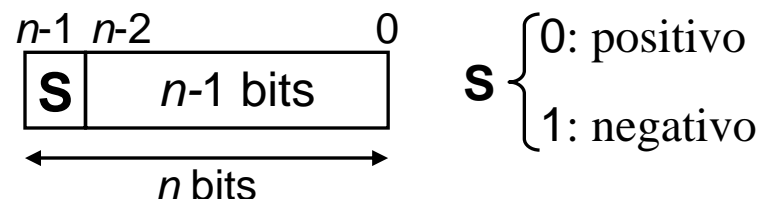
$$\begin{array}{r} B'00011001 = 25_{10)} \\ + \\ B'11100111 = -25_{10)} \\ \hline \end{array}$$

¡Fácil de sumar!  $B'1\boxed{00000000}$

Se invierten bits y se suma 1.  $\text{Repr}(-N) = 2^n - N$

Rango de representación:  $[-2^{n-1}, 2^{n-1}-1]$

## Enteros con signo



1. Formato signo  $B'00011001 = 25_{10)}$   
y magnitud:  $B'10011001 = -25_{10)}$

Rango de representación:  $[-2^{n-1}+1, 2^{n-1}-1]$

2. Complemento a uno:

$$\begin{array}{r} B'00011001 = 25_{10)} \\ B'11100110 = -25_{10)} \end{array}$$

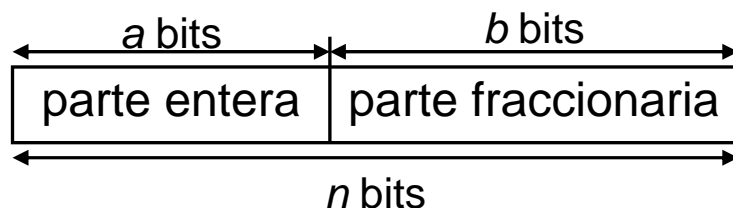
Se invierten los bits.  $\text{Repr}(-N) = 2^n - N - 1$

Rango de representación:  $[-2^{n-1}+1, 2^{n-1}-1]$

# Números fraccionarios

(representación aproximada de números reales)

## Coma fija

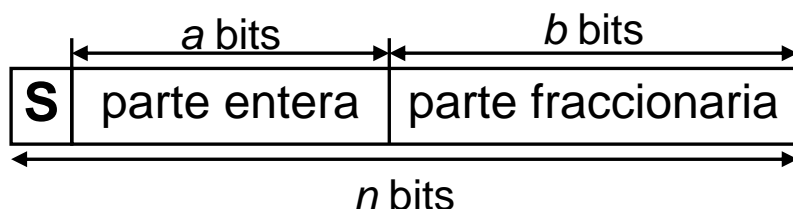


$$101,10011_2 = 5,59375_{10}$$

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5}$$

$$101,10011_2 = \frac{10110011_2}{100000_2} = \frac{179_{10}}{32_{10}} = 5,59375_{10}$$

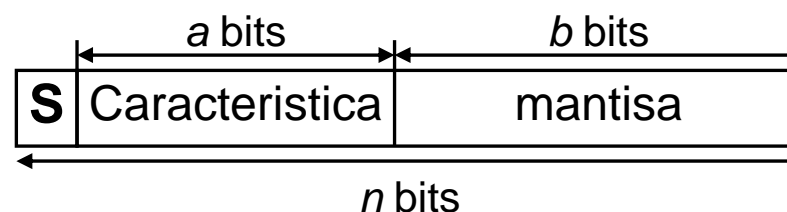
## Coma fija con signo (complemento a dos)



## Coma flotante

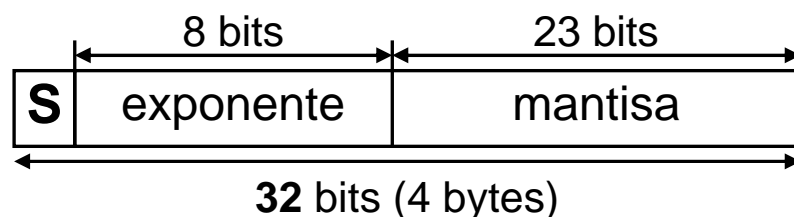
$$3,7124 \cdot 10^7 \xrightarrow{\log_a N = \frac{\log_b N}{\log_b a}} 2^{25,1458} = 1,1063 \cdot 2^{(25)}$$

Mantisa      Exponente

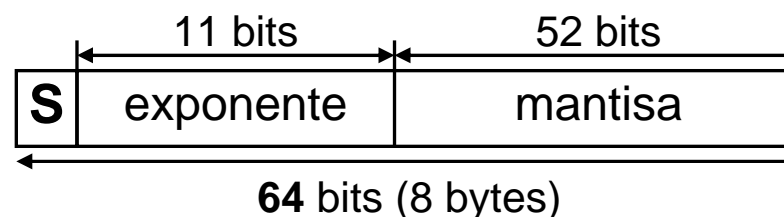


- Mantisa (1,M):  $1,1063_{10} \cong 1,0001110_2$   
Se guardan en la mantisa los bits detrás de la coma.
- Característica: en exceso de  $2^{a-1} - 1$   
Característica =  $2^{a-1} - 1 + \text{Exponente}$   
Si  $a$  es 6, entonces  
Característica =  $2^{6-1} - 1 + 25 = 56_{10}$

## Norma IEEE 754



Precisión simple (**single** en lenguaje C)



Precisión doble (**double** en lenguaje C)

Ejemplo: convertir el número  $-4,731 \times 10^{12}$  a la norma IEEE 754 precisión simple.

1. Calculamos equivalencia en exponente 2:

$$\frac{\log_{10} 4,731 \cdot 10^{12}}{\log_{10} 2} = \frac{12 + 0,674952}{0,301030} = 42,10528$$

$$4,731 \cdot 10^{12} = 2^{42,10528} = 2^{42} \cdot 2^{0,10528} = 1,0757 \cdot 2^{42}$$

2. Calculamos el exponente en exceso  $2^7 - 1 = 127$

$$\text{Repr(exponente)} = 127 + 42 = 169_{10} = 10101001_2$$

3. Calculamos la mantisa:

$$\frac{0,0757 \cdot 2^{23}}{2^{23}} = \frac{635017,6}{2^{23}} = \frac{635018}{2^{23}}$$

$$635018_{10} = 00010011011000010001010_2$$

4. Construimos el número resultante:

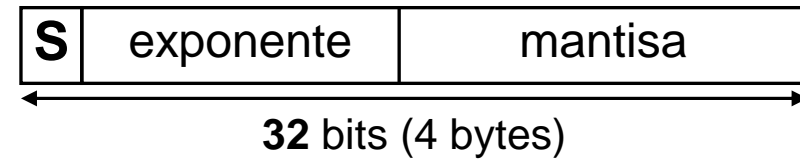
$$\begin{array}{cccccccc} 1 & 10101001 & 00010011011000010001010 \\ \hline \mathbf{D} & \mathbf{4} & \mathbf{8} & \mathbf{9} & \mathbf{B} & \mathbf{0} & \mathbf{8} & \mathbf{A} \end{array}$$

**Nota:** Utilizando más decimales en los cálculos logarítmicos intermedios se logra una mayor precisión en la representación numérica

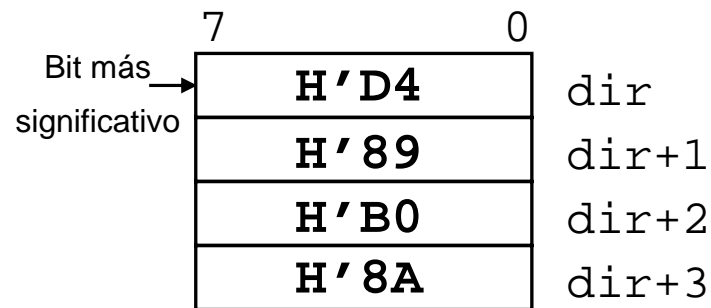
## Almacenamiento en memoria

¿ Cómo almacenar números de 32 bits en un computador con tamaño de palabra de 8 bits ?

Utilizando 4 palabras contiguas de memoria.

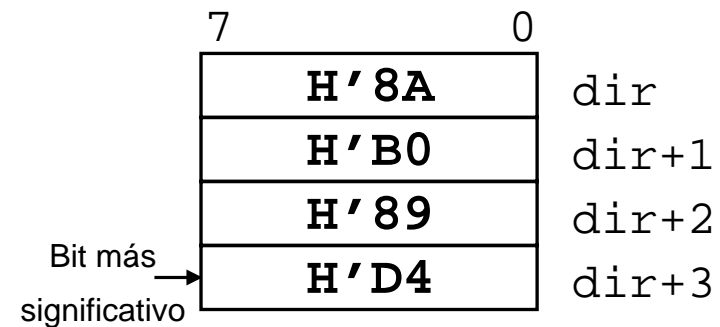


**H'D489B08A**



Convenio “extremista mayor”  
(*big endian*)

Motorola (MAC)



Convenio “extremista menor”  
(*little endian*)

Power PC

Intel (IBM PC)

## Operaciones aritméticas y lógicas. Indicadores.

- **Operaciones aritméticas** Suma, resta, multiplicación, división, etc ...

- **Operaciones lógicas**

Se realizan bit a bit  
entre los operandos

op1	op2	AND	OR	XOR	NAND	NOR
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	1	0
1	1	1	1	0	0	0

- **Indicadores de la unidad aritmética lógica (ALU):**

**Z** → Resultado cero (*zero*)      **V** → Desbordamiento (*overflow*)

**N** → Resultado negativo      **C** → Acarreo ; “llevada” (*carry*)

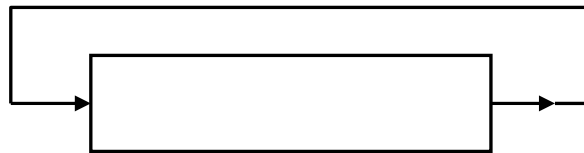
## Operaciones de desplazamiento



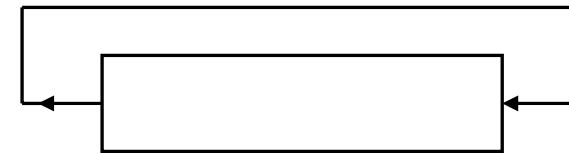
Desplazamiento lógico a la derecha (SHR)



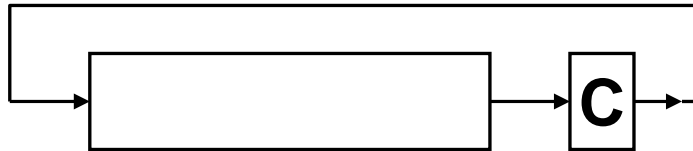
Desplazamiento lógico a la izquierda (SHL)



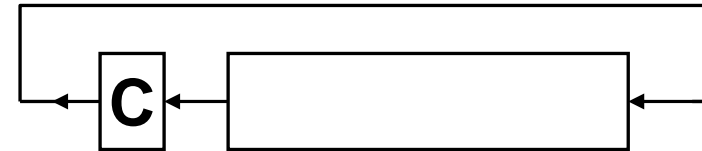
Desplazamiento circular a la derecha (ROR)



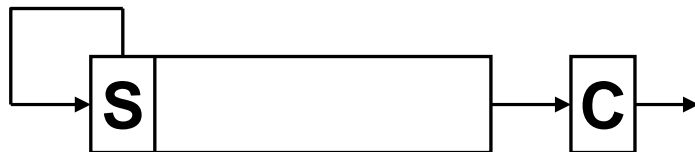
Desplazamiento circular a la izquierda (ROL)



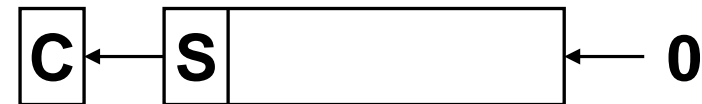
Despl. circular con acarreo a la derecha (RORC)



Despl. circular con acarreo a la izquierda (ROLC)



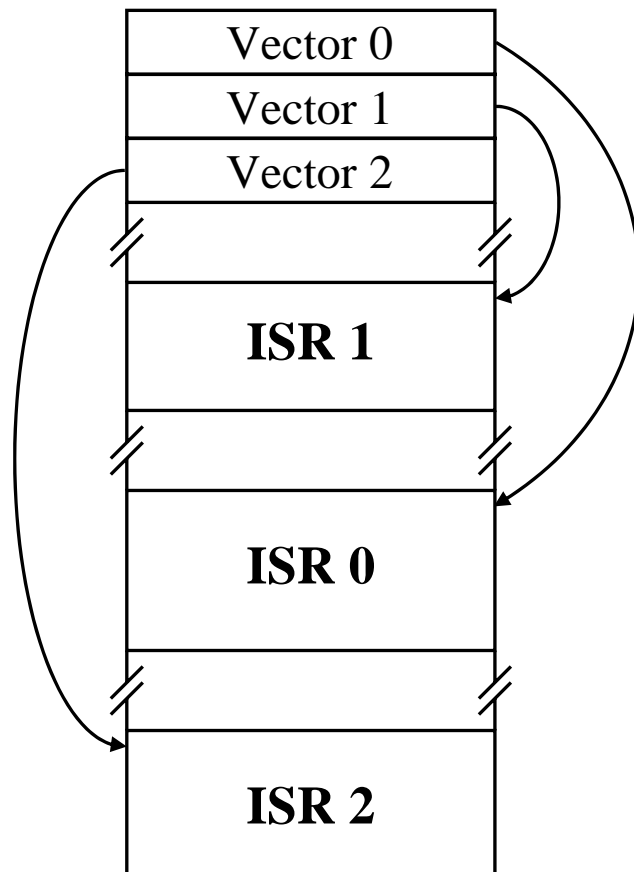
Desplazamiento aritmético a la derecha (ASR)



Desplazamiento aritmético a la izquierda (ASL)



## Vectores de interrupción



### Posibles fuentes de interrupción:

#### Internas al procesador (síncronas):

- Desbordamiento en operaciones aritméticas.
- Código de operación ilegal.
- Interrupción forzada por el programa.
- Fallo en la lectura de la memoria (memory fault)
- Interrupción “perro guardian” (watchdog)

...

#### Externas al procesador (asíncronas):

- Periféricos (teclado, pantalla, comunicaciones ...)
- Reloj de tiempo real.
- Fallo de alimentación del procesador.

...