

# 66.70 Estructura del Computador

## **Punto Flotante**

# *Punto flotante*

En muchos cálculos el intervalo de números que se usan es muy grande:

- la masa del electrón,  $9 \times 10^{-28}$  gramos
- la masa del Sol,  $2 \times 10^{33}$  gramos
- Los lenguajes de programación deben permitirme escribir variables para manejar valores extremadamente bajos y también extremadamente altos, ¿la representación binaria de estas variables podría ser en punto fijo?

# Representación en punto fijo

$M_e = 0000000000000000000000000000000000.000000000000000000000000000009$

$M_s = 2000000000000000000000000000000000.000000000000000000000000000000$

$M_e = \text{Masa del electrón} = 9 \times 10^{-28} \text{ gramos}$

$M_s = \text{Masa del sol} = 2 \times 10^{33} \text{ gramos}$

En punto fijo:  
¿Cuántos dígitos son necesarios para poder representar tanto  $M_e$  como  $M_s$ ?

- Cuántos dígitos decimales?
- Cuántos dígitos binarios?

# *Punto flotante*

$$\text{número representado} = M \times \text{base}^{\text{exp}}$$

De un total de ***N*** bits:

- > ***1*** bit para el signo de la mantisa
- > ***x*** bits para mantisa
- > ***y*** bits para el exponente (magnitud y signo)

-> Analizar diferentes valores de *x* e *y* para igual *N*  
¿Conclusiones?

# *¿Porqué un estándar?*

- Casi todos los lenguajes de programación ofrecen tipos de variable punto flotante
- Todo sistema operativo debe responder a excepciones punto flotantes (overflow)
- Desde PCs a supercomputadoras tienen coprocesadores para operaciones en PF

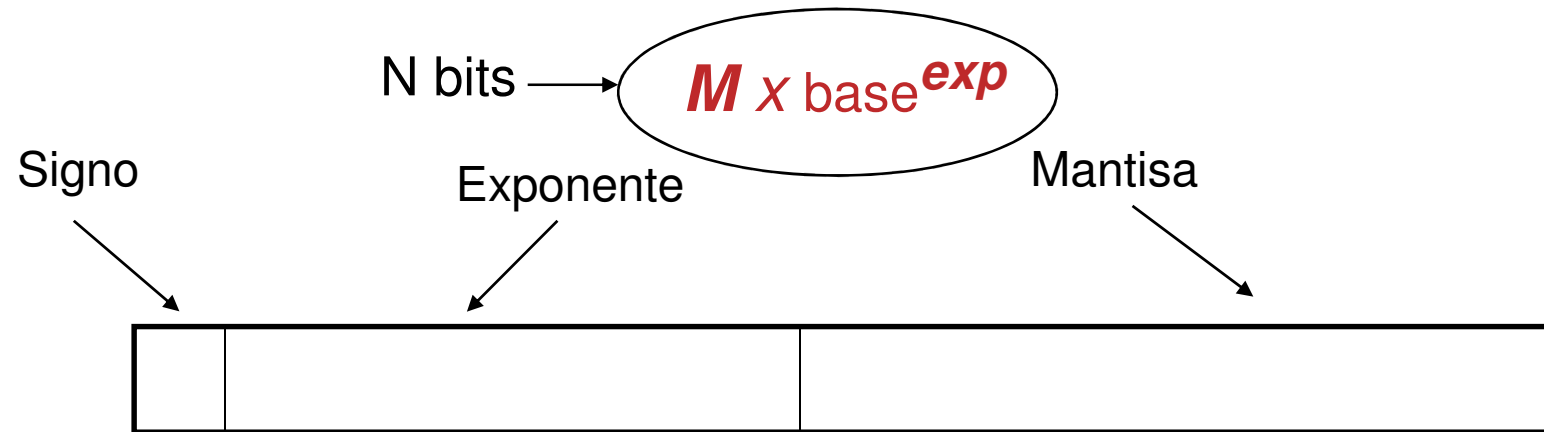
# *Punto flotante*

- Casi todos los lenguajes de programación ofrecen datos en punto flotante
- Desde PCs a supercomputadoras tienen coprocesadores para operaciones en PF

## **Estandarización** del formato PF: **IEEE 754**

- En 1982 la IEEE definió el estándar IEEE-754
- Lo implantó por primera vez en los Intel 8087
- En 1985 este formato fue aceptado como el estándar universal
- En 2008 se incluyeron modificaciones a la norma original

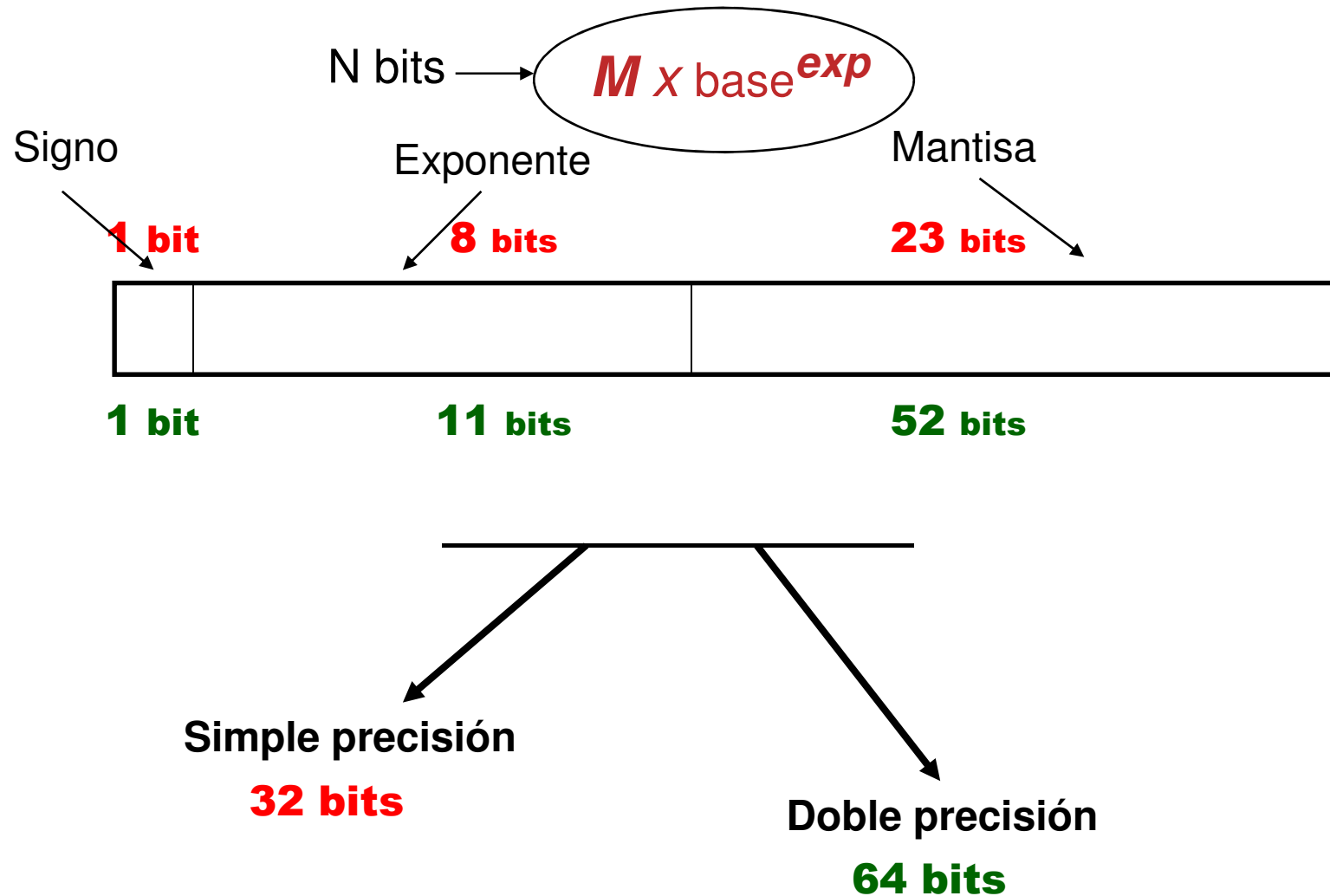
# Norma IEEE 754



Simple precisión

Doble precisión

# Norma IEEE 754





# Definiendo la Norma IEEE 754

## Cuestiones a establecer:

- Qué base utilizar?
- Números ‘normalizados’
- \_Formato para guardar el exponente? (*entero con signo*)
- Valores “especiales”

# Definiendo la Norma IEEE 754

## **¿Qué base utilizar?**

- ✓ Cuál elegir? 2, 10 , 16 ...
- ✓ Qué efecto tiene sobre la representación?
- ✓ Conveniencia al realizar operaciones aritméticas

# Definiendo la Norma IEEE 754

## Valores normalizados

- Bit implícito vale 1

### Ventajas:

- ✓ *La representación binaria es única para un número dado*
- ✓ *Todos los bits de la mantisa son significativos*
- ✓ *Es más fácil comparar dos números:*
  - 1º) Comparo exponentes 2º) Comparo mantisas*

# Definiendo la Norma IEEE 754

## Representación del exponente

- El exp. es un número entero con signo
- Sistema para su representación
  - Magnitud y Signo?
  - Complemento a 1 ?
  - Complemento a 2 ?
  - “Exceso-N” ?

# Representación “exceso 7”

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

# Representación “exceso 7”

<i>Decimal</i>	<i>Two's Complement</i>	<i>Ones' Complement</i>	<i>Signed Magnitude</i>	<u><i>Exceso 7</i></u>	
-8	1000	—	—	<del>0 0 0 0</del>	Valor reservado en IEEE 754
-7	1001	1000	1111	0 0 0 0	
-6	1010	1001	1110	0 0 0 1	- 6
-5	1011	1010	1101	0 0 1 0	
-4	1100	1011	1100	0 0 1 1	- 4
-3	1101	1100	1011	0 1 0 0	
-2	1110	1101	1010	0 1 0 1	
-1	1111	1110	1001	0 1 1 0	
0	0000	1111 or 0000	1000 or 0000	0 1 1 1	→ 0
1	0001	0001	0001	1 0 0 0	
2	0010	0010	0010	1 0 0 1	
3	0011	0011	0011	1 0 1 0	
4	0100	0100	0100	1 0 1 1	
5	0101	0101	0101	1 1 0 0	+5
6	0110	0110	0110	1 1 0 1	
7	0111	0111	0111	1 1 1 0	+7
				<del>1 1 1 1</del>	Valor reservado en IEEE 754

# Representación “exceso 7”

<i>Decimal</i>	<i>Two's Complement</i>	<i>Ones' Complement</i>	<i>Signed Magnitude</i>	<u><i>Exceso 7</i></u>	
-8	1000	—	—	<del>0 0 0 0</del>	Valor reservado en IEEE 754
-7	1001	1000	1111	0 0 0 0	
-6	1010	1001	1110	0 0 0 1	- 6
-5	1011	1010	1101	0 0 1 0	
-4	1100	1011	1100	0 0 1 1	- 4
-3	1101	1100	1011	0 1 0 0	
-2	1110	1101	1010	0 1 0 1	
-1	1111	1110	1001	0 1 1 1	
0	0000	1111 or 0000	1000 or 0000	0 1 1 1 →	0
1	0001	0001	0001	1 0 0 0	
2	0010	0010	0010	1 0 0 1	
3	0011	0011	0011	1 0 1 0	
4	0100	0100	0100	1 0 1 1	
5	0101	0101	0101	1 1 0 0	+5
6	0110	0110	0110	1 1 0 1	
7	0111	0111	0111	1 1 1 0	+7
				<del>1 1 1 1</del>	Valor reservado en IEEE 754

¿Ventajas?

# Definiendo la Norma IEEE 754

- IEEE 754 expresa el componente en exceso-N
- Cuál debería ser el valor de N ?



# Rango representable *en simple precisión*

## Rango del exponente

**8 bits , exceso 127**

*No admite  $Exp=0000..0000$  ni  $Exp=1111..1111$*

*Máximo exponente representable (valor positivo): 1111 1110 -> 127*

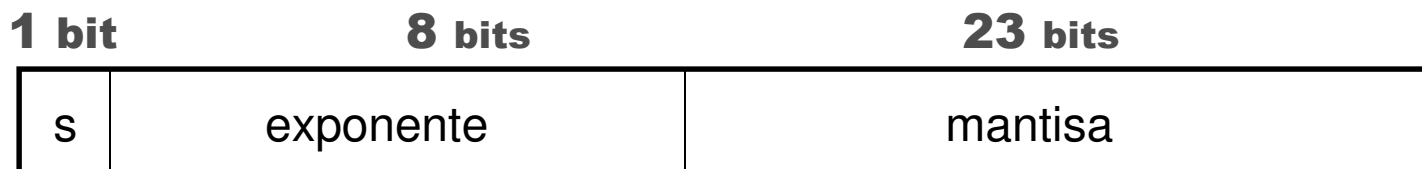
*Mínimo exponente representable (valor negativo): 0000 0001 -> -126*

## Rango de la mantisa

**23 bits**

*normalizar => bit implícito => 24 bits =>  $Mantisa = 1.0 + Mantisa\ guardada$*

*=>  $1 \leq Mantisa < 2$*



# Rango representable *en doble precisión*

## Rango del exponente

**11 bits , exceso 1023**

No admite  $Exp=0000..0000$  ni  $Exp=1111..1111$

Máximo exponente representable (valor positivo): 1111 1110  $\rightarrow$  1023

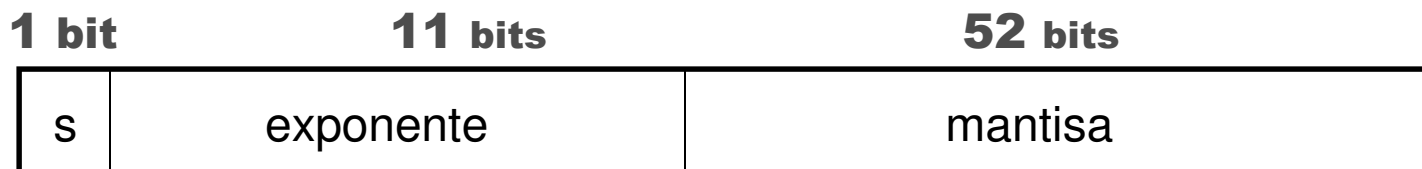
Mínimo exponente representable (valor negativo): 0000 0001  $\rightarrow$  -1022

## Rango de la mantisa

**52 bits**

normalizar  $\Rightarrow$  bit implícito  $\Rightarrow$  53 bits  $\Rightarrow$   $Mantisa = 1.0 + Mantisa\ guardada$

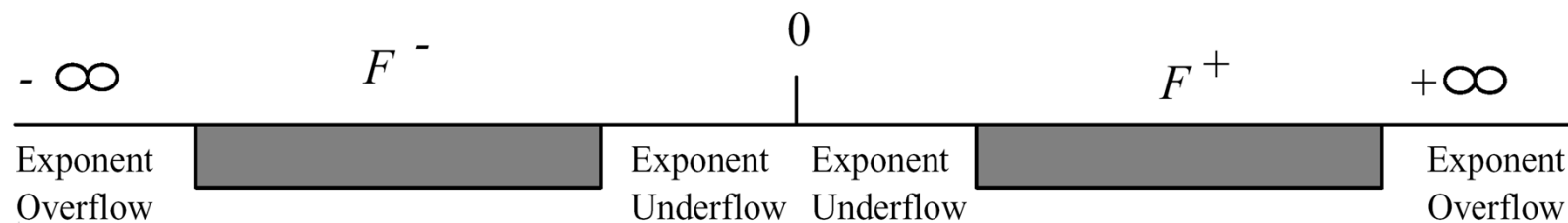
$\Rightarrow 1 \leq Mantisa < 2$



# Rango representable

## Overflow y Underflow

$$M_{min} \cdot \text{base}^{exp_{min}} \leq Núm. \leq M_{max} \cdot \text{base}^{exp_{max}}$$



# *Resolución*

*Números reales, su representación en punto fijo y en punto flotante*

**Dada una cadena de 32/64 bits**

- *Cuántos números diferentes puedo representar?*
- *En qué rango de valores?*
- *Cuál es la distancia entre dos valores sucesivos?*
- *Es uniforme esa distancia?*

# Valores de referencia en IEEE-754

	Simple precisión	Doble precisión
Bits del signo	1	1
Bits del exponente	8	11
Bits de la mantisa	23	52
Total de bits	32	64
Sistema de exponente	Exceso en 127	Exceso en 1023
Intervalo del exponente	-126 a +127	-1022 a +1023
Número normalizado más pequeño	$2^{-126}$	$2^{-1022}$
Número normalizado más grande	aprox. $2^{128}$	aprox. $2^{1024}$
Intervalo decimal	aprox. $10^{-38}$ a $10^{38}$	aprox. $10^{-308}$ a $10^{308}$

Tipo	Tam. Bits	Rango	
		Min	Max
Bool	8	0	1
Char	8	-128	127
Signed char	8	-128	127
unsigned char	8	0	255
short int	16	-32,768	32,767
unsigned short int	16	0	65,535
Int	32	-2,147,483,648	2,147,483,647
unsigned int	32	0	4,294,967,295
long int	32	-2,147,483,648	2,147,483,647
unsigned long int	32	0	4,294,967,295
long long int	64	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
unsigned long long int	64	0	18,446,744,073,709,551,615
Float	32	1.17549e-38	3.40282e+38
Double	64	2.22507e-308	1.79769e+308

# La Explosión del Ariane 5

El 4 de junio de 1996, el cohete Ariane 5 fue lanzado en Kourou, Guayana francesa por la Agencia Espacial Europea pero...

...explotó 40 seg después su lanzamiento en



*El cohete estaba en su primer viaje, después de una década de desarrollo que costó u\$s 7 billones. El cohete destruido y su carga estaban estimadas en u\$s 500 millones*

# Norma IEEE 754

## *Valores especiales*

### Cero

- Todos los bits en cero. Signo.

### Infinito

- Exp=todos 1's , Mantisa = todos 0's . Signo.

### NaN (*"Not a number"*)

- E=todos 1's , Mantisa  $\neq 0$ , Signo = *no importa*



# Sumar dos números en punto flotante

- 1) Calcular la diferencia entre los exponentes  $d = |Exp1 - Exp2|$   
*=> determino cuál es el número mayor y cuál el menor*
- 2) Correr  $d$  posiciones a la derecha la coma del número menor
- 3) Encolumnar y sumar las mantisas
- 4) El exponente del resultado es el exponente del número mayor
- 5) Normalizar la mantisa del resultado ajustando el exponente si fuese necesario

# Punto fijo VS. Punto flotante

- ❖ Precisión
- ❖ Rango dinámico
- ❖ Velocidad
- ❖ Requerimientos de hardware