# TECNICAS Y ESTRUCTURAS DIGITALES

PROFESORES MG. SERGIO L. MARTINEZ - ING. VICTOR D. SANCHEZ R.

2022

## Unidad 1.1 – SISTEMAS DE REPRESENTACIÓN DE LA INFORMACIÓN

# SISTEMAS DE NUMERACIÓN

Contenido

- Sistemas de numeración.
- Sistemas posicionales.
- Representación numérica.
- Sistemas sin signo.
- Sistemas con signo.
- Sistemas con notación complemento.
- Complementación binaria.
- Aproximaciones.
- Cambio de base.
- Sistemas de punto flotante.

## Referencias (Aula Virtual)


> Wakerly J. (2001). DISEÑO DIGITAL. Capítulo 2: Sistemas y Códigos Numéricos.

> Flórez Fernández H. (2010). DISEÑO LÓGICO. Capítulo 1: Sistemas Numéricos.

> Tokheim R. (1992). FUNDAMENTO DE LOS MICROPROCESADORES. Capítulo 2: Números, Códigos de Computadora y Aritmética.

> Brown S., Vranesic Z. (2006). FUNDAMENTOS DE LÓGICA DIGITAL CON DISEÑO VHDL. Capítulo 5: Representación de números y circuitos aritméticos.

> Velasco J. (2011). REPRESENTACIÓN DE LA INFORMACIÓN. Capítulo 1: Los números y los sistemas de representación. Capítulo 2: Representación de los números en un computador. Capítulo 3: Otros tipos de representaciones.

# SISTEMAS DE NUMERACIÓN

## Definición

**Los sistemas de numeración o sistemas numéricos pueden definirse en forma general como *estructuras formada por símbolos y leyes que permiten numerar, ordenar, contar y/o efectuar operaciones aritméticas***

## Clasificación

S/estructura
- No posicionales → leyes de formación varias
- Posicionales → leyes de formación relacionadas con la posición de los símbolos

(Wakerly, pg. 26)

# Propiedades

- Se constituyen con un número finito de símbolos con los que puede representarse cualquier valor.
- La base o raíz del sistema la determina la cantidad de símbolos definidos.
- Cada símbolo aislado representa un número específico de unidades.
- Se define un símbolo especial (cero) para indicar la ausencia de elementos a contar en el sistema.
- Un mismo símbolo, dentro de la agrupación numérica, tiene diferente significación o peso según el lugar que ocupa.
- Las posiciones hacia la izquierda corresponden a las de mayor peso, a partir del peso 1 (para el caso de los sistemas numéricos).

# Teorema Fundamental de la Numeración (TFN)

**Establece que "todo número expresado en un sistema posicional de cualquier base se puede representar por la suma polinómica de cada símbolo/dígito, multiplicado por la base del sistema elevada a la posición que ocupa":**

$$(\underbrace{D_{k-1}}_{\text{signo}}\overbrace{D_{k-2}\ \ldots\ D_1\ D_0}^{k}\ ,\overbrace{D_{-1}\ \ldots\ D_{-f}}^{f})_b = \sum_{i=-f}^{k-1} D_i.b^i$$

$$= (D_{k-1}.b^{k-1} + D_{k-2}.b^{k-2} + \ldots + D_1.b^1 + D_0.b^0 + D_{-1}.b^{-1} + \ldots + D_{-f}.b^{-f})_b$$

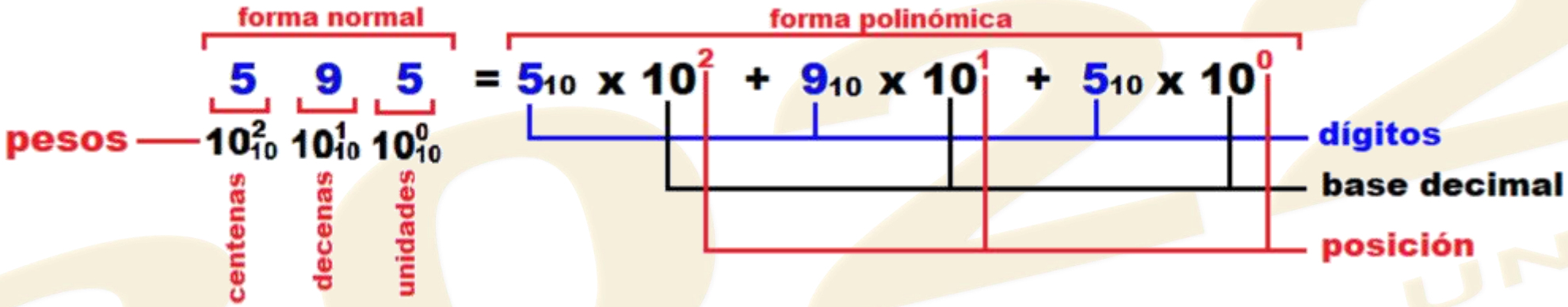
$$0 \leq D_i \leq b - 1 \quad \text{cuando } i = [\,k - 1, -f]$$

(Velasco, pg. 12)

# SISTEMAS POSICIONALES

## Representación numérica (basada en el TFN)

### Ejemplo en base 10 (decimal)




### ecuaciones de un número genérico

**Forma estándar (ordenados por posición)**

$$(D_{k-1}\ D_{k-2}\ \ldots\ D_1\ D_0)_{base}$$

**Forma polinómica (ordenados por potencias)**

$$(D_{k-1}.b^{k-1} + D_{k-2}.b^{k-2} + \ldots + D_1.b^1 + D_0.b^0)_{base}$$

**La formación, representación e interpretación de un número, en cualquier base, es similar al ejemplo anterior, considerando la cantidad de símbolos del sistema (Teorema Fundamental de la Numeración).**

# SISTEMAS POSICIONALES

## Notación

$(b,k,f)_{xx}$

- b = base del sistema.
- k = cantidad de posiciones enteras.
- f = cantidad de posiciones fraccionarias.
  - Cuando el sistema es entero, f = 0 y no se indica.
- xx = tipo de sistema
  - SS = sin signo
  - CS = con signo
  - NC = notación complemento
  - PF = punto flotante

**Tipos de sistemas posicionales**

- Enteros
  - Sin signo
  - Con signo
  - Notación complemento
- Fraccionarios
  - Con signo
  - Notacion complemento
- De punto flotante
  - Con signo
  - Notación complemento
  - Mixtos

2022 Facultad de Ingeniería - UNJu

# SISTEMAS POSICIONALES

## Ejemplos de progresión numérica en sistemas posicionales

| Binario | Ternario | Cuaternario | Quinario | Semario | Octal | Decimal | Duo-decimal | Hexa-decimal | Vigesimal |
|---|---|---|---|---|---|---|---|---|---|
| b=2 | b=3 | b=4 | b=5 | b=6 | b=8 | b=10 | b=12 | b=16 | b=20 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 11 | 10 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 100 | 11 | 10 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 101 | 12 | 11 | 10 | 5 | 5 | 5 | 5 | 5 | 5 |
| 110 | 20 | 12 | 11 | 10 | 6 | 6 | 6 | 6 | 6 |
| 111 | 21 | 13 | 12 | 11 | 7 | 7 | 7 | 7 | 7 |
| 1000 | 22 | 20 | 13 | 12 | 10 | 8 | 8 | 8 | 8 |
| 1001 | 100 | 21 | 14 | 13 | 11 | 9 | 9 | 9 | 9 |
| 1010 | 101 | 22 | 20 | 14 | 12 | 10 | A | A | A |
| 1011 | 102 | 23 | 21 | 15 | 13 | 11 | B | B | B |
| 1100 | 110 | 30 | 22 | 20 | 14 | 12 | 10 | C | C |
| 1101 | 111 | 31 | 23 | 21 | 15 | 13 | 11 | D | D |
| 1110 | 112 | 32 | 24 | 22 | 16 | 14 | 12 | E | E |
| 1111 | 120 | 33 | 30 | 23 | 17 | 15 | 13 | F | F |
| 10000 | 121 | 100 | 31 | 24 | 20 | 16 | 14 | 10 | G |
| 10001 | 200 | 101 | 32 | 25 | 21 | 17 | 15 | 11 | H |
| 10010 | 201 | 102 | 33 | 30 | 22 | 18 | 16 | 12 | I |
| 10011 | 202 | 103 | 34 | 31 | 23 | 19 | 17 | 13 | J |
| 10100 | 210 | 110 | 40 | 32 | 24 | 20 | 18 | 14 | 10 |

# SISTEMAS POSICIONALES

## Características de las progresiones numéricas

| Binario | Ternario | Cuaternario | Quinario | Semario | Octal | Decimal | Duo-decimal | Hexa-decimal | Vigesimal |
|---|---|---|---|---|---|---|---|---|---|
| b=2 | b=3 | b=4 | b=5 | b=6 | b=8 | b=10 | b=12 | b=16 | b=20 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 11 | 10 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 100 | 11 | 10 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 101 | 12 | 11 | 10 | 5 | 5 | 5 | 5 | 5 | 5 |
| 110 | 20 | 12 | 11 | 10 | 6 | 6 | 6 | 6 | 6 |
| 111 | 21 | 13 | 12 | 11 | 7 | 7 | 7 | 7 | 7 |
| 1000 | 22 | 20 | 13 | 12 | 10 | 8 | 8 | 8 | 8 |
| 1001 | 100 | 21 | 14 | 13 | 11 | 9 | 9 | 9 | 9 |
| 1010 | 101 | 22 | 20 | 14 | 12 | 10 | A | A | A |
| 1011 | 102 | 23 | 21 | 15 | 13 | 11 | B | B | B |
| 1100 | 110 | 30 | 22 | 20 | 14 | 12 | 10 | C | C |
| 1101 | 111 | 31 | 23 | 21 | 15 | 13 | 11 | D | D |
| 1110 | 112 | 32 | 24 | 22 | 16 | 14 | 12 | E | E |
| 1111 | 120 | 33 | 30 | 23 | 17 | 15 | 13 | F | F |
| 10000 | 121 | 100 | 31 | 24 | 20 | 16 | 14 | 10 | G |
| 10001 | 200 | 101 | 32 | 25 | 21 | 17 | 15 | 11 | H |
| 10010 | 201 | 102 | 33 | 30 | 22 | 18 | 16 | 12 | I |
| 10011 | 202 | 103 | 34 | 31 | 23 | 19 | 17 | 13 | J |
| 10100 | 210 | 110 | 40 | 32 | 24 | 20 | 18 | 14 | 10 |

- **La base de cualquier sistema posicional, expresada en el mismo sistema vale siempre 10.**
- **Cuanto mayor es la base, una misma cantidad se representa con una imagen numérica menor (y en general menos posiciones).**
- **Cuanto menor es la base, una misma cantidad se representa en general en una mayor cantidad de posiciones numéricas.**
- **Cuando la base es mayor que la del sistema decimal, para representar los símbolos adicionales mayores que nueve, se utilizan letras mayúsculas en orden alfabético creciente.**

# SISTEMAS SIN SIGNO $(b,k)_{ss}$

(Brown, pg. 246)
(Velasco, pg. 44)

- El signo no tiene significación ni representación en estas estructuras.
- Se utilizan en casos especiales como ser la representación de códigos o valores particulares (DNI, fechas, ordenaciones, etc.).
- No son apropiados para operaciones algebraicas.
- No se utilizan para números fraccionarios.

ampliar TFN
Teorema
Fundamental de
la Numeración
Velasco pg. 12

**Notación**

$(D_{k-1}\ D_{k-2}\ \ldots\ D_1\ D_0)_b \rightarrow$ forma normal

$(D_{k-1}.b^{k-1} + D_{k-2}.b^{k-2} + \ldots + D_1.b^1 + D_0.b^0)_b \rightarrow$ forma polinóm ica

**Ejemplos**

$(10,8)_{ss} \rightarrow$ Sistema decimal de 8 posiciones
DNI $\rightarrow 20456123_{10}$

$(2,6)_{ss} \rightarrow$ Sistema binario de 6 posiciones
palabra en código $\rightarrow 111000_2$

# SISTEMAS CON SIGNO

(Velasco, pg. 44)
(Wakerly, pg. 34)
(Brown, pg. 256)

cantidad posiciones enteras

cantidad posiciones fraccionarias

base

$$\text{enteros} \leftarrow (b,k)_{cs} \qquad (b,k,f)_{cs} \rightarrow \text{fraccionarios}$$

base

tipo de sistema

$(2\ ,\ 8)_{cs}$ Ej: $(1\ 0010111)_2$

$(8\ ,\ 4\ ,\ 2)_{cs}$ Ej: $(0\ 357{,}17)_8$

- El signo se representa con un número (0=positivo; 1=negativo) en la posición más significativa.
- Se pueden utilizar para operaciones algebraicas.
- La representación puede ser para números enteros y fraccionarios.

**Notación**

$$(\overbrace{D_{k-1}\ D_{k-2}\ \ldots\ D_1\ D_0}^{k}\ ,\ \overbrace{D_{-1}\ \ldots\ D_{-f}}^{f})_b \rightarrow \text{forma normal}$$

$$((-1)^{D_{k-1}}.(D_{k-2}.b^{k-2} + \ldots + D_1.b^1 + D_0.b^0 + D_{-1}.b^{-1} + \ldots + D_{-f}.b^{-f}))_b$$

$$(-1)^{D_{k-1}}.\sum_{i=-f}^{k-2} D_i.b^i)_b$$

} formas polinómicas equivalentes

$$0 \le D_i \le b-1 \quad \text{cuando } i = [-f\ ,\ k-2] \qquad \text{y} \qquad D_{k-1} = \begin{cases} 0 & \text{si } N \ge 0 \\ 1 & \text{si } N < 0 \end{cases}$$

# SISTEMAS CON SIGNO

## Ejemplos de representación

| Nº ORIGEN | SISTEMA | REPRESENTACIÓN |
|---|---|---|
| $N = +5,7_{10}$ | $(10,4,2)_{CS}$ | $N = 0005,70_{10}$ |
| $N = -5,7_{10}$ | $(10,4,2)_{CS}$ | $N = 1005,70_{10}$ |
| $N = -101,10_2$ | $(2,3,3)_{CS}$ | desborde |
| $N = -101,10_2$ | $(2,4,3)_{CS}$ | $N = 1101,100_2$ |

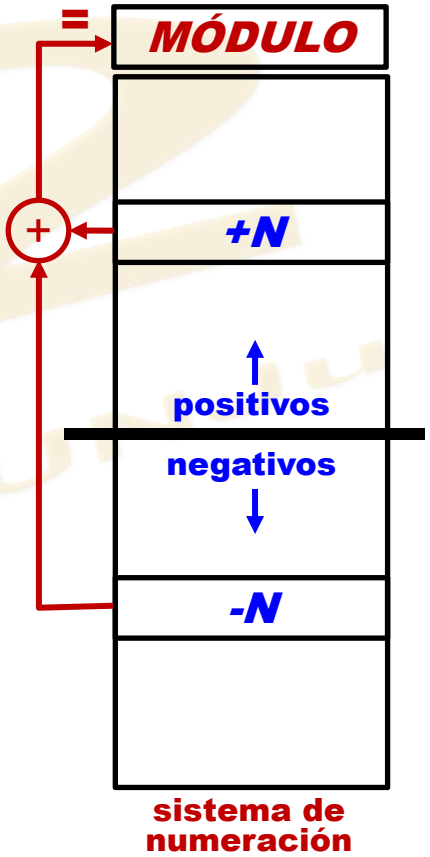

- La representación en sistemas con signo es simple y directa.
- A partir de un número con signo explícito, los dígitos/símbolos se ubican en las posiciones correspondientes.
- Si el número de origen es positivo, se coloca un 0 en la posición k-1. Si es negativo se coloca un 1 en esa misma posición.
- Si hay posiciones excedentes en el sistema, se rellenan con 0.
- Si no alcanzan las posiciones en la parte fraccionaria se puede aproximar. Si no alcanzan las posiciones en la parte entera no se puede representar (desborde/overflow).

# SISTEMAS CON NOTACIÓN COMPLEMENTO

(Tokheim, pg. 24)
(Velasco, pg. 36)

$(b,k)_{NC} \rightarrow$ enteros $\qquad (b,k,f)_{NC} \rightarrow$ fracc.

- Es otro formato para representar números con signo.
- La representación puede ser para números enteros y fraccionarios.
- Es el formato que utilizan las computadoras.
- El signo se representa con un número (0=positivo; 1=negativo) en la posición más significativa (k-1).
- Se pueden utilizar para operaciones algebraicas.
- La suma de cualquier +N y -N siempre produce el mismo valor (módulo).
- A diferencia de los números con signo, los números negativos cambian de apariencia respecto de sus equivalentes positivos.




**Notación**

$$(\overbrace{\underbrace{D_{k-1}}_{\text{signo}} D_{k-2} \dots D_1 D_0}^{k} , \overbrace{D_{-1} \dots D_{-f}}^{f})_b \quad \leftarrow \text{forma normal}$$

$$(-\underbrace{D_{k-1}}_{\text{signo}}.b^{k-1} + D_{k-2}.b^{k-2} + \dots + D_1.b^1 + D_0.b^0 + D_{-1}.b^{-1} + \dots + D_{-f}.b^{-f})_b$$

$$(-\underbrace{D_{k-1}}_{\text{signo}}.b^{k-1} + \sum_{i=-f}^{k-2} D_i.b^i)_b \quad \leftarrow \text{formas polinómicas}$$

$$0 \le D_i \le b-1 \quad \text{cuando } i = [\,k-2\,,-f\,] \qquad \text{y} \qquad D_{k-1} = \begin{cases} 0 & \text{si } N \ge 0 \\ 1 & \text{si } N < 0 \end{cases} \leftarrow \text{signo}$$

# SISTEMAS CON NOTACIÓN COMPLEMENTO

## Ejemplos de representación

signo

| Nº ORIGEN | SISTEMA | REPRESENTACIÓN |
|---|---|---|
| $N = +5,7_{10}$ | $(10,4,2)_{NC}$ | $N = 0005,70_{10}$ |
| $N = -5,7_{10}$ | $(10,4,2)_{NC}$ | $N = 1994,30_{10}$ |
| $N = -101,10_2$ | $(2,4,3)_{NC}$ | $N = 1010,100_2$ |

relleno de posiciones es diferente para +N (ceros) que para -N (b-1) en parte entera y 0 en parte fraccionaria

- La representación en sistemas con notación complemento es bastante particular. El bit de signo va en la posición k-1.
- Si el número de origen es positivo, se representa igual que en los sistemas con signo.
- Si el número es negativo, requiere del proceso de complementación como paso previo a su representación. En este último caso, la imagen de los números negativos cambia.
- Las posiciones excedentes se rellenan con 0 para los +N y con (b-1) para la parte entera de los -N y con 0 para la parte fraccionaria de los -N.
- Si un número en NC debe cambiar de signo, requiere del proceso de complementación, ya sea positivo o negativo.

# Complementación

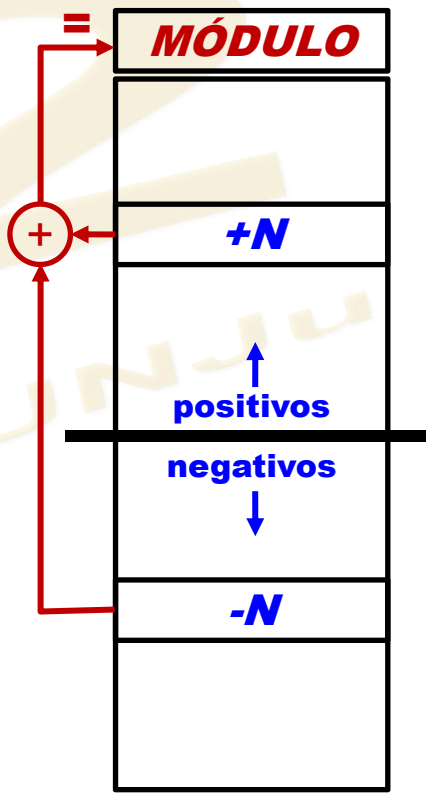
(Florez, pg. 32)
(Velasco, pg. 36)

- Es un procedimiento que se aplica en sistemas $(b,k)_{NC}$ o $(b,k,f)_{NC}$ cuando se desea ***cambiar el signo de un número***.
- La suma de un número y su correspondiente complemento produce siempre el mismo resultado, un valor que se denomina ***módulo***.
- El módulo está representado por la unidad seguida de tantos ceros como posiciones provea

  Módulo = $b^k = 10\ldots0_b$ → para sistemas enteros (k)

  Módulo = $b^k = 10\ldots0,0\ldots0_b$ → para sistemas fraccionarios (k+f)

- El proceso anterior genera el complemento llamado ***a la base***.
- Otro formato de complemento se denomina *a la* ***base menos uno*** y se obtiene restando 1 al complemento a la base (solo para sistemas enteros).




sistema de numeración con notación complemento

# Complementación en $(b,k,f)_{NC}$

(Wakerly, pg. 35)

$(b,k)_{NC} \rightarrow$ Sistemas enteros con notación complemento

$$\text{Módulo} = b^k = N + \bar{N} \Rightarrow \bar{N} = b^k - N = (-1+1) - N$$

$$\bar{N} = [1\,(b-1)\ldots(b-1)]_b - N + [0\,0\ldots1]_b \leftarrow \text{Complemento a la base}$$

$$\bar{N}_{b-1} = \bar{N} - 1 \Rightarrow \bar{N}_{b-1} = [1\,(b-1)\ldots(b-1)]_b - N \leftarrow \text{Complemento a la base menos 1}$$

$(b,k,f)_{NC} \rightarrow$ Sistemas fraccionarios con notac. complemento

$$\text{Módulo} = b^k = N + \bar{N} \Rightarrow \bar{N} = b^k - N = (-b^{-f} + b^{-f}) - N$$

$$\bar{N} = [1\,(b-1)\ldots(b-1),(b-1)\ldots(b-1)]_b - N + [0\,0\ldots0,0\ldots1]_b$$

Complemento a la base

# Complementación binaria

(Velasco, pg. 38)

$(2,k)_{NC}$ → Sistemas enteros con notación complemento

$(2,k,f)_{NC}$ → Sistemas fraccionarios con notación complemento

**Método 1: (enteros)** $\bar{N}_b = -1_b - N_b + 1_b$

**(fraccionarios)** $\bar{N}_b = -b^{-f}{}_b - N_b + b^{-f}{}_b$

} Válido para cualquier base

**Método 2a:** (enteros) Intercambiar ceros por unos y viceversa (complemento a la base menos uno) y sumar 1 al resultado (complemento a la base).

**Método 2b:** (fraccionarios) Intercambiar ceros por unos y viceversa y sumar $b^{-f}$ al resultado (complemento a la base).

**Método 3:** Desde la derecha se mantienen los bits hasta el primer 1 inclusive, a partir de allí se intercambian 0 por 1 y viceversa (complemento a la base).

# SISTEMAS CON NOTACIÓN COMPLEMENTO

## Complementación ejemplos

$(-b^{-f}) \rightarrow 1\ 9\ 9\ 9\ ,\ 9\ 9\ _{10}$

$-\ (N) \rightarrow 1\ 5\ 8\ 3\ ,\ 1\ 6\ _{10}$

$0\ 4\ 1\ 6\ ,\ 8\ 3\ _{10}$

$+\ (b^{-f}) \rightarrow 0\ 0\ 0\ 0\ ,\ 0\ 1\ _{10}$

$(\bar{N})\quad 0\ 4\ 1\ 6\ ,\ 8\ 4\ _{10}$

**complementación en sistema decimal**

$(-b^{-f}) \rightarrow 1\ F\ F\ F\ ,\ F\ F\ _{16}$

$-\ (N) \rightarrow 0\ A\ 3\ B\ ,\ 1\ 0\ _{16}$

$1\ 5\ C\ 4\ ,\ E\ F\ _{16}$

$+\ (b^{-f}) \rightarrow 0\ 0\ 0\ 0\ ,\ 0\ 1\ _{16}$

$(\bar{N})\quad 1\ 5\ C\ 4\ ,\ F\ 0\ _{16}$

**complementación en sistema hexadecimal**

$(-b^{-f}) \rightarrow 1\ 1\ 1\ 1\ ,\ 1\ 1\ _{2}$

$-\ (N) \rightarrow 1\ 0\ 0\ 1\ ,\ 1\ 0\ _{2}$

$0\ 1\ 1\ 0\ ,\ 0\ 1\ _{2}$

$+\ (b^{-f}) \rightarrow 0\ 0\ 0\ 0\ ,\ 0\ 1\ _{2}$

$(\bar{N})\quad 0\ 1\ 1\ 0\ ,\ 1\ 0\ _{2}$

**complementación en sistema binario**

**Mismo ejemplo con dos métodos "diferentes"**

**Desde la derecha se mantienen los bits hasta el primer 1 inclusive, a partir de allí se intercambian 0 por 1 y viceversa (complemento a la base).**

$(N) \rightarrow 1\ 0\ 0\ 1\ ,\ 1\ 0\ _{2}$

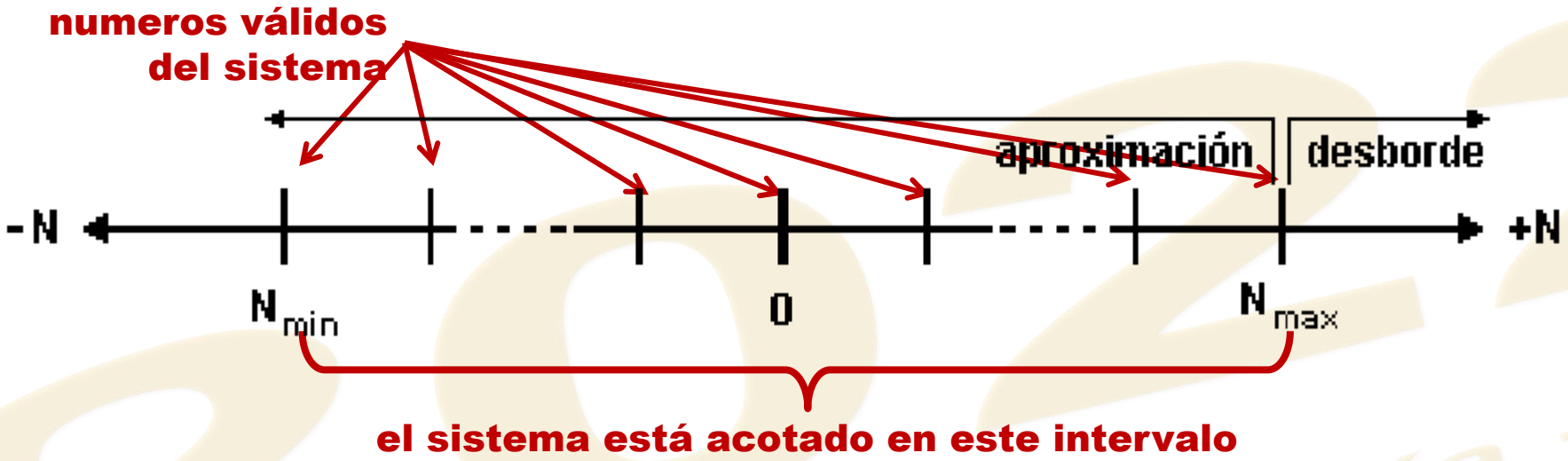
$\downarrow\ \downarrow\ \downarrow\ \downarrow\ \downarrow\ \downarrow$

$(\bar{N})\quad 0\ 1\ 1\ 0\ ,\ 1\ 0\ _{2}$

**complementación en sistema binario**

# Aproximaciones numéricas $(b,k,f)_{xx}$

(Velasco, pg. 27)




Sistemas acotados y discretos

Si se intenta representar un número mayor que $N_{máx}$ (fuera del rango) se produce una condición de desborde y el número no puede mostrarse en el sistema.

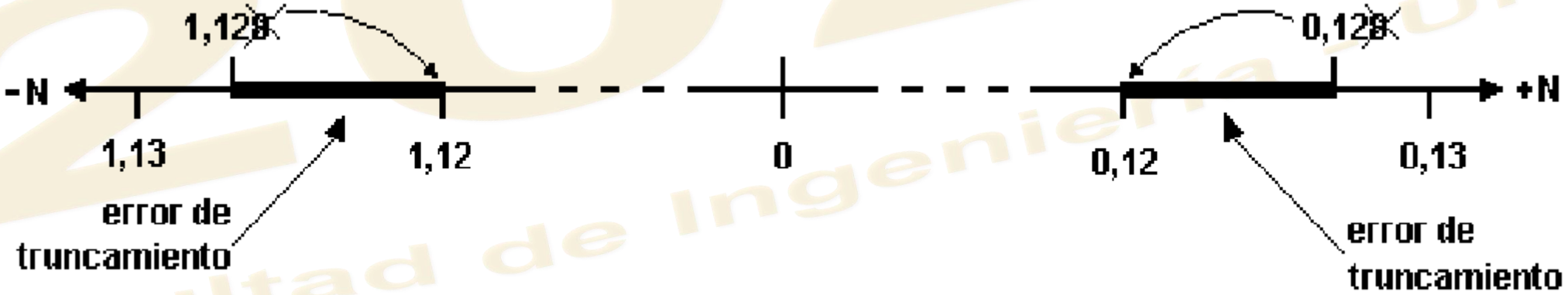
Si se intenta representar un número menor que $N_{min}$ o bien, cuando el número, perteneciente al rango del sistema, tiene una distancia al número más próximo, menor que $b^{-f}$, se puede obtenerse una representación aproximada.

# SISTEMAS POSICIONALES - APROXIMACIONES

## Aproximación por truncamiento $(b,k,f)_{xx}$

**El criterio para esta aproximación consiste en eliminar y despreciar el valor de las cifras fraccionarias excedentes hasta llegar a la cantidad que establece el sistema dado.**

**El proceso siempre se aplica sobre números positivos en los sistemas con notación complemento.**




**Ejemplo de aproximación por truncamiento en un sistema $(10,1,2)_{cs}$**

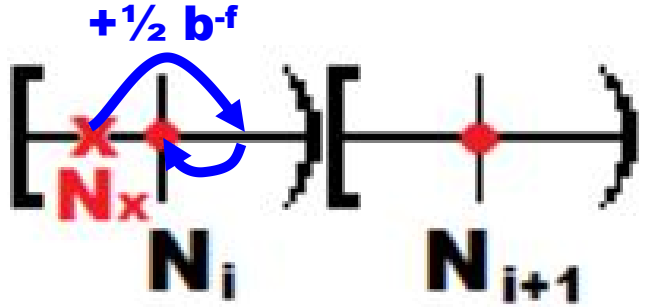
**Error de truncamiento → $e_T < b^{-f}$**

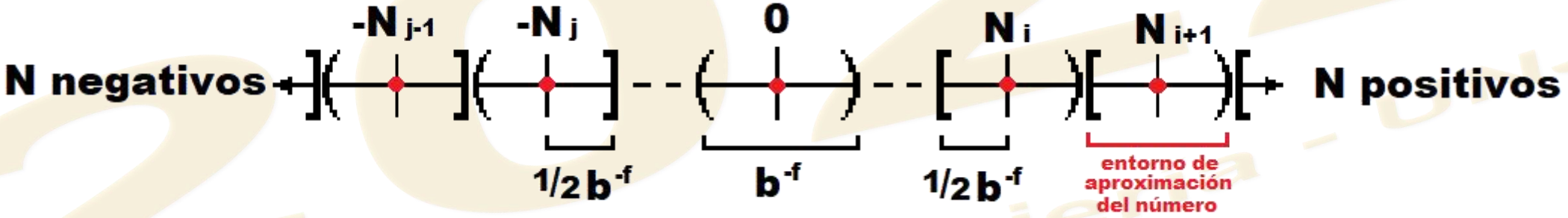
# SISTEMAS POSICIONALES - APROXIMACIONES

## Aproximación por redondeo $(b,k,f)_{xx}$

**Para sistemas $(b,k,f)_{CS}$**

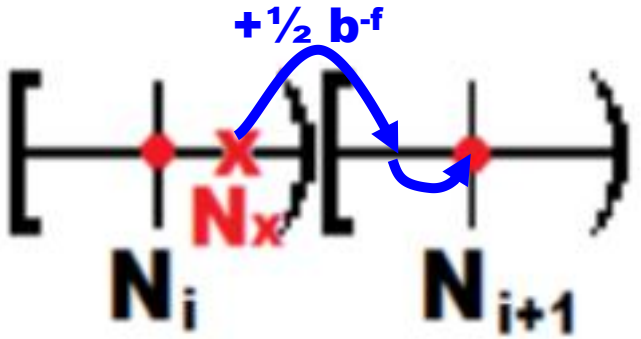
- Se trunca el número a f+1 posiciones fraccionarias.
- Se suma ½ $b^{-f}$.
- Se trunca el resultado a f posiciones.




**Para sistemas $(b,k,f)_{NC}$**

- Se trunca el **valor positivo** del número a f+1 posiciones.
- Se suma ½ $b^{-f}$ (mitad del intervalo entre 2 números consecutivos.
- Se trunca el resultado a f posiciones.




**Error de redondeo → $e_R < ½.b^{-f}$**

# Aritmética en base origen - Enteros

(Velasco, pg. 13; Wakerly, pg. 29)
(Florez, pg. 25)

Dado un número *N* en base *b* (origen), para obtener su representación en base *p* (destino), se expresa la base destino en el sistema origen ($p_b$) y se efectúa el cociente entero entre *N* y $p_b$,

$$\begin{array}{r|l} N & p_b \\ \hline -R_1- & C_1 \end{array}$$

cociente que puede expresarse como

$$N = C_1 \cdot p_b + R_1 \qquad \text{con} \quad 0 \leq R_1 \leq p_b - 1$$

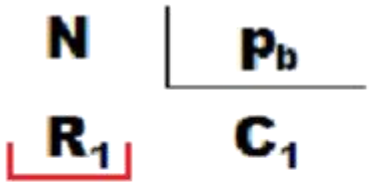
luego, el resto $R_1$ es una cifra que pertenece al número *N* en base *p*. Si resulta

$C_1 \geq p_b$ es necesario aplicar nuevamente el procedimiento

$$\begin{array}{r|l} C_1 & p_b \\ \hline -R_2- & C_2 \end{array} \Rightarrow C_1 = C_2 \cdot p_b + R_2 \text{ con } \quad 0 \leq R_2 \leq p_b -$$

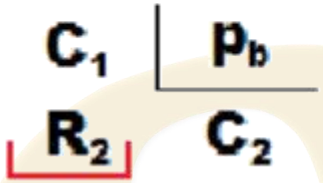
# Aritmética en base origen - Enteros

1° división ⟶




$$\Rightarrow N = C_1 \cdot p_b + R_1 \quad \text{con} \quad 0 \leq R_1 < p_b$$

2° división ⟶




$$\Rightarrow C_1 = C_2 \cdot p_b + R_2 \quad \text{con} \quad 0 \leq R_2 < p_b$$

**Si $C_2 \geq p_b$ se continúa aplicando el algoritmo hasta lograr que el cociente resulte finalmente menor que $p_b$.**

**Considerando que $C_2 < p_b$, se reemplazan los resultados parciales obtenidos en la expresión de N**

$$N = (C_2 \cdot p_b + R_2) \cdot p_b + R_1 = C_2 \cdot p_b^2 + R_2 \cdot p_b + R_1$$

**obteniéndose la representación polinómica del número N en el sistema de base p.**

**Expresando N en forma regular, resulta**

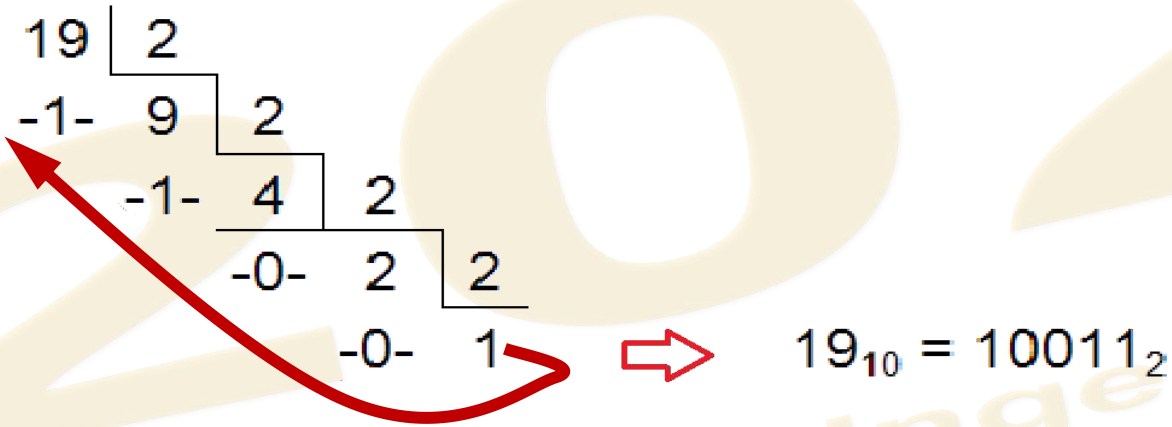
$$N = (C_2\ R_2\ R_1)_p$$

# Aritmética en base origen - Enteros

**Ejemplo:** Representar el número decimal $N = +19_{10}$ en el sistema binario con aritmética en base origen.

$N = +19_{10}$ $b = 10_{10}$ (base origen)

$p = 10_2$ (base destino) $p_b = 2_{10}$ (base destino expresada en sistema origen)




$19_{10} = 10011_2$

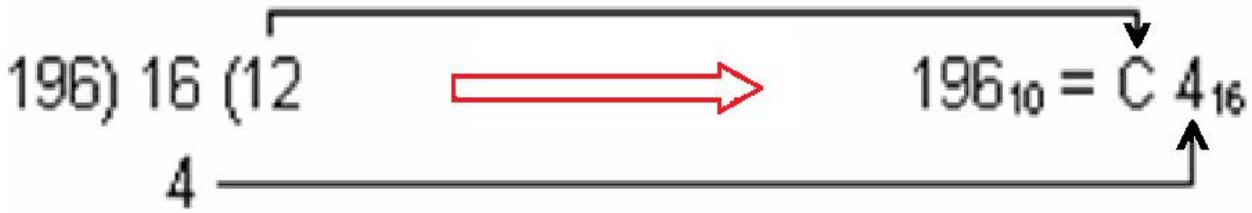
nótese que el número en la nueva base se forma con el último cociente como cifra más significativa, hasta llegar al primer resto como cifra menos significativa.

**Ejemplo:** Representar el número decimal $N = +196_{10}$ en el sistema hexadecimal con aritmética en base origen.

$N = +196_{10}$ $b = 10_{10}$ (base origen)

$p = 10_{16}$ (base destino)

$p_b = 16_{10}$ (base destino expresada en sistema origen)



# Aritmética en base origen - Fraccionarios *(continuación)*

Considerando a $N_f$ como la parte fraccionaria de un número general N en base b

$$(N = N_e , N_f)_b = N_e + N_f$$

se multiplica la parte $N_f$ por la base destino expresada en sistema origen $p_b$

$$N_f \cdot p_b = N_{e1} + N_{f1} \qquad (1)$$

lo que produce nuevamente un valor formado por una parte entera ($N_{e1}$) y una parte fraccionaria ($N_{f1}$).

Dividiendo ambos miembros de la ec. (1) por $p_b$ , se tiene

$$N_f \cdot \cancel{p_b} \cdot \cancel{p_b^{-1}} = N_{e1} \cdot p_b^{-1} + N_{f1} \cdot p_b^{-1} \qquad (2)$$

donde $N_{e1}$ resulta el primer símbolo o primera cifra fraccionaria del número general N, expresada en el sistema destino.

Para obtener una segunda cifra fraccionaria ($N_{e2}$) en el sistema destino, se repite el proceso anterior

$$N_{f1} \cdot p_b = N_{e2} + N_{f2}$$

$$N_{f1} = N_{e2} \cdot p_b^{-1} + N_{f2} \cdot p_b^{-1} \qquad (3)$$

Finalmente el proceso se generaliza hasta obtener la cantidad necesaria de cifras fraccionarias o se llega a una conversión exacta.

# Aritmética en base origen - Fraccionarios

**Reemplazando ec. (3) en (2), se tiene**

$$N_f = N_{e1} \cdot p_b^{-1} + N_{f1} \cdot p_b^{-1} \qquad (2)$$

$$N_{f1} = N_{e2} \cdot p_b^{-1} + N_{f2} \cdot p_b^{-1} \qquad (3)$$

$$N_f = N_{e1} \cdot p_b^{-1} + (N_{e2} \cdot p_b^{-1} + N_{f2} \cdot p_b^{-1}) \cdot p_b^{-1}$$

$$N_f = N_{e1} \cdot p_b^{-1} + N_{e2} \cdot p_b^{-2} + N_{f2} \cdot p_b^{-2} \qquad (4)$$

**nótese cómo se va formando la representación polinómica de la parte fraccionaria ($N_f$) del número general ($N$).**

**Si el proceso se generaliza, se obtiene**

$$N_f = N_{e1} \cdot p_b^{-1} + N_{e2} \cdot p_b^{-2} + \ldots + N_{ef} \cdot p_b^{-f} + \underbrace{N_{ff} \cdot p_b^{-f}}_{\text{error}} \qquad (5)$$

**La conversión final resulta**

$$N_f \approx (0, N_{e1}\, N_{e2} \ldots N_{ef})_p$$

**con un error dado por el término final de (5), que se comete al considerar f cifras fraccionarias; y puede ser cero si la conversión es exacta.**

# Aritmética en base origen - Fraccionarios

**Ejemplo:** Representar el número fraccionario $Nf = +0,59_{10}$ en el sistema binario con error menor al 5%, aplicando aritmética en base origen.

$N = +0,59_{10}$ $\quad b = 10_{10}$ (base origen)

$p = 10_2$ (base destino) $\quad p_b = 2_{10}$ (base destino expresada en sistema origen)

| | | error de conversión en base origen |
|---|---|---|
| $0,59_{10} \times 2 = 1 + 0,18$ | $\rightarrow 0,59_{10} = 1 \times 2^{-1} + 0,18 \times 2^{-1}$ | $0,09_{10}$ |
| $0,18_{10} \times 2 = 0 + 0,36$ | $\rightarrow 0,59_{10} = 1 \times 2^{-1} + 0 \times 2^{-2} + 0,36 \times 2^{-2}$ | $0,09_{10}$ |
| $0,36_{10} \times 2 = 0 + 0,72$ | $\rightarrow 0,59_{10} = 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 0,72 \times 2^{-3}$ | $0,09_{10}$ |
| $0,72_{10} \times 2 = 1 + 0,44$ | $\rightarrow 0,59_{10} = 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 0,44 \times 2^{-4}$ | $0,03_{10}$ |
| $0,44_{10} \times 2 = 0 + 0,88$ | $\rightarrow 0,59_{10} = 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + 0,88 \times 2^{-5}$ | $0,03_{10}$ |
| | *cifras fraccionarias en sistema destino* | |

**finalmente** $\quad N = +0,59_{10} = +0,10010_2$

**con error →**

- $e_a = (0,88 \times 2^{-5})_{10} = 0,03_{10}$ → **error absoluto**
- $e_r\% = (e_a / N) \times 100 = (0,03_{10} / 0,59_{10}) \times 100 = 4,7\% < 5\%$ → **error relativo**

# Aritmética en base destino - Enteros y fraccionarios

(Velasco, pg. 14)
(Florez, pg. 26)

**Este procedimiento está basado en el TFN (*Teorema Fundamental de la Numeración*) y es válido para números enteros y fraccionarios.**
**Dado un número *N* en base *b* (origen), a partir de su expresión polinómica, se reemplazan cada uno de los símbolos *Di*, la base *b* y los exponentes, por sus equivalentes en base destino *p* y se efectúan las operaciones indicadas, en el sistema destino.**

$$N_b = \overbrace{D_{k-1}.b^{k-1} + \ldots + D_i.b^i + \ldots + D_0.b^0}^{\text{parte entera}} + \overbrace{D_{-1}.b^{-1} + \ldots + +D_{-f}.b^{-f}}^{\text{parte fraccionaria}}$$

$$N_p = Dp_{k-1}.b_p^{(k-1)_p} + \ldots + Dp_i.b_p^{i_p} + Dp_0.b_p^0 + Dp_{-1}.b_p^{-1} + \ldots + Dp_{-f}.b_p^{-f_p}$$

**donde**
- $N_b$ = número en base origen
- $N_p$ = número en base destino
- $D$ = símbolo en base origen
- $D_p$ = símbolo equivalente en base destino
- $b$ = base origen
- $b_p$ = base origen expresada en sistema destino
- $i$ = exponente en base origen
- $i_p$ = exponente equivalente expresado en sistema destino

# Aritmética en base destino - Enteros y fraccionarios

***Ejemplo***: Representar el número decimal N = $+19_{10}$ en el sistema binario, con aritmética en base destino.

N = 19 b = 10 (base origen) p = 2 (base destino)

*expresando N en forma polinómica, se reemplazan sus cifras, base y exponentes por sus equivalentes en el sistema binario (destino) y se resuelve la nueva polinómica*[3]

$$19_{10} = 1_{10} \times 10^{1}{}_{10} + 9_{10} =$$

$$= 1_2 \times 1010^{1}{}_2 + 1001_2 = 10011_2$$

***Ejemplo***: Representar el número N = $+196_{10}$ en el sistema hexadecimal, con aritmética en base destino.

N = $196_{10}$

b = 10 (origen)

p = 16 (destino)

$$196_{10} = 1_{10} x 10^2_{10} + 9_{10} x 10_{10} + 6_{10}$$

$$196_{10} = 1_{16} x A^2_{16} + 9_{16} x A_{16} + 6_{16}$$

$$196_{10} = 64_{16} + 5A_{16} + 6_{16} = C4_{16}$$

# CAMBIO DE BASE DIRECTO

**Relación de bases → b = base origen → p = base destino**

**base destino mayor ← $p = b^n$ ó $p^n = b$ → base origen mayor**

**Primer caso**
**$p = b^n$**

**Dado un número *N* en base *b*, se forman grupos de *n* cifras hacia la izquierda y hacia la derecha del separador fraccionario[6] agregando ceros si fuera necesario para completar los grupos extremos. Luego, cada uno de esos grupos representan una cifra en la base destino *p*, expresada aún en la base origen *b*, que deberán ser convertidas aisladamente.**

$$N_b = \underbrace{D_{k-1}\,D_{k-2}}_{n} \ldots \underbrace{D_i}_{n} \ldots \underbrace{D_1\,D_0}_{n} , \underbrace{D_{-1}\,D_{-2}}_{n} \ldots \underbrace{D_{-j}}_{n} \ldots \underbrace{D_{-f}}_{n}$$

$D_{p_i}$ $D_{p_j}$

*Ejemplo:* Convertir en forma directa el número N = $+1101110{,}1001_2$ a octal

$b = 2_{10}$ --> base origen $p = 8_{10} = 2^3_{10}$ --> base destino

$$N_2 = \overline{001}\ \overline{101}\ \overline{110}\ ,\ \overline{100}\ \overline{100}_2 =$$

$$= 1\ \ 5\ \ 6\ ,\ 4\ \ 4_8 = N_8$$

# CAMBIO DE BASE DIRECTO

**Relación de bases → b = base origen → p = base destino**

**$p^n = b$ → base origen mayor**

**Segundo caso**
**$p^n = b$**

**En este caso, cada cifra del número en base origen se reemplaza por *n* cifras en base destino. Luego, los conjuntos obtenidos se colocan en el mismo orden que el número original. El proceso es por lo tanto inverso al caso anterior, y como se aplica bajo el mismo criterio tanto para la parte entera como para la fraccionaria, es válido para todo el número.**

$$N_b = D_k \dots D_i \dots D_0 , D_{-1} \dots D_{-j} \dots D_{-f}$$

$$\overbrace{D_{pk_{n-1}} D_{pk_{n-2}} \dots D_{pk_0}} \dots \overbrace{D_{p0_{n-1}} D_{p0_{n-2}} \dots D_{p0_0}} , \overbrace{D_{p-1_{n-1}} D_{p-1_{n-2}} \dots D_{p-1_0}} \dots \overbrace{D_{p-f_{n-1}} D_{p-f_{n-2}} \dots D_{p-f_0}}$$

# CAMBIO DE BASE DIRECTO

***Ejemplo***: Hacer un cambio de base directo al número $9AB,51_{16}$ del sistema hexadecimal al sistema binario.

b = 16 = $2^4$ (base origen) p = 2 (base destino)

$N_{16}$ = 9 A B , 5 $1_{16}$ =

= 1001 1010 1011 , 0101 $0001_2 = N_2$

***Ejemplo:*** Hacer un cambio de base "directo" del número $732,25_8$ octal a hexadecimal

7 3 2 , 2 $5_8$

111 011 010 , 010 $101_2$

reagrupamiento

0001 1101 1010 , 0101 $0100_2$

1 C A , 5 $4_{16}$

**En este caso, no se pueden relacionar las bases en forma conveniente. Se realiza un cambio directo a binario, se reagrupan los bits y se vuelve a realizar un cambio directo a hexadecimal. Se conoce como cambio semi-directo.**

# SISTEMAS DE PUNTO FLOTANTE

(Velasco, pg. 60)

**Un sistema de punto flotante, generalizado para números fraccionarios, se identifica como $(b,k,f)_{FL}$ , donde un número presenta la siguiente estructura general**

$$N = M \cdot b^E$$

**donde M es la *mantisa*, perteneciente a un sistema (b,1,f -1) de punto fijo.**

$$M = (M_0 , M_{-1} M_{-2} \dots M_{-(f-1)})_b$$

**b es la base del sistema.**

**E es el exponente perteneciente a un sistema (b,k) entero.**

$$E = (E_{k-1} \dots E_0)_b$$

2024 Facultad de Ingeniería - UNJu

# SISTEMAS DE PUNTO FLOTANTE

$$N = M \cdot b^E$$

**Los sistemas de punto flotante pueden ser a su vez:**

***Con signo:*** **La mantisa y el exponente se expresan en sistemas con notación de signo.**

***Con notación complemento:*** **La mantisa y el exponente se expresan en sistemas con notación complemento.**

***Mixtos:*** **Cualquier combinación en las notaciones de signo o complemento en los sistemas de representación de la mantisa y el exponente.**

***Ejemplo:***

$N_1 = +0{,}002597_{10}$ en $(10,3,4)_{FL\text{-}CS} \Rightarrow N_1 = 0{,}260 \times 10^{102}$

$N_2 = -101{,}001_2$ en $(2,4,5)_{FL\text{-}NC} \Rightarrow N_2 = 1{,}0110 \times 10^{0011}$

$N_3 = -0{,}05BF_{16}$ en $(16,2,6)_{FL\text{-}MIXTO} \Rightarrow N_3 = 1{,}A7100 \times 10^{11}$

**(mantisa en NC, exponente CS).**

# SISTEMAS DE PUNTO FLOTANTE

## Estructura normalizada

$$N = M \cdot b^E$$

Número $+2{,}5_{10}$ en un sistema $(10,3,4)_{FL}$

$2{,}500 \times 10^{000} = 0{,}250 \times 10^{001} = 0{,}002 \times 10^{003}$

Para unificar la representación y permitir la presentación de la mayor cantidad de cifras significativas que permita el sistema empleado, se considera que el sistema está ***normalizado*** cuando

- En una mantisa con signo es $M_{-1} \neq 0$
- En una mantisa con notación complemento es

$$\begin{cases} M_{-1} \neq 0 & \text{si } M_0 = 0 \\ M_{-1} \neq (b-1) & \text{si } M_0 = 1 \end{cases}$$

**Normalizado** ⟶ $0{,}250 \times 10^{001}$

# SISTEMAS DE PUNTO FLOTANTE

## Estándar IEEE 754

(Brown, pg. 295)

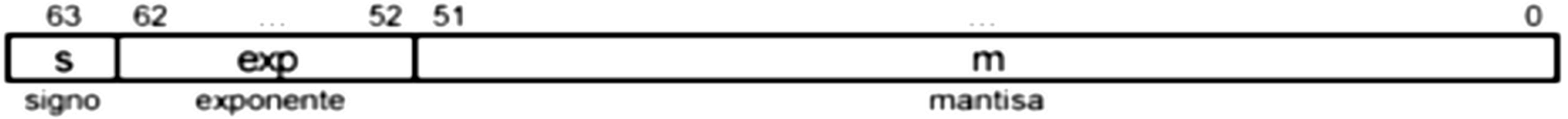
**El estándar del IEEE para aritmética en punto (coma) flotante (IEEE 754) es la norma o estándar técnico para computación en punto (coma) flotante, establecida en 1985 por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE).**

**La norma abordó muchos problemas encontrados en las diversas implementaciones de coma flotante que las hacían difíciles de usar de forma fiable y portátil.**

**Muchas unidades de coma flotante de hardware utilizan ahora el estándar IEEE 754.**

**Precisión simple:** **Se usan 32 bits (4 bytes): 1 bit para el signo (s), 23 bits para la mantisa (m) y 8 bits para el exponente (exp).**


**Precisión doble:** **Se emplean 64 bits (8 bytes): 1 bit para el signo (s), 52 bits para la mantisa (m) y 11 bits para el exponente (exp).**



# SISTEMAS DE PUNTO FLOTANTE

## Estándar IEEE 754 — Ejemplo → Precisión simple

**Representar + 101110 , 01010111010000111111000011111100 $_2$**

**1→ Signo = 0**

**2→ Normalizar:** $1{,}01110010101110100001111110000111110 0 \times 10^{-101}{}_2$

$(\times\ 2^{-5})_{10}$

**3→ Exponente en exceso 127** $\longrightarrow$ $2^{n-1} - 1 = 2^7 - 1 = 127_{10}$

$5_{10} + 127_{10} = 132_{10}$

$101_2 + 1111111_2 = 10000100_2$

**4→ Quitar el entero y aproximar la mantisa a 23 bits**

~~1,~~ 01110010101110100001111 ~~100001111100~~

**5→ Disposición final**

| 0 | 10000100 | 01110010101110100001111 |
|---|---|---|

**6→ Representación hexadecimal:** 4 2 3 9 5 D 0 F

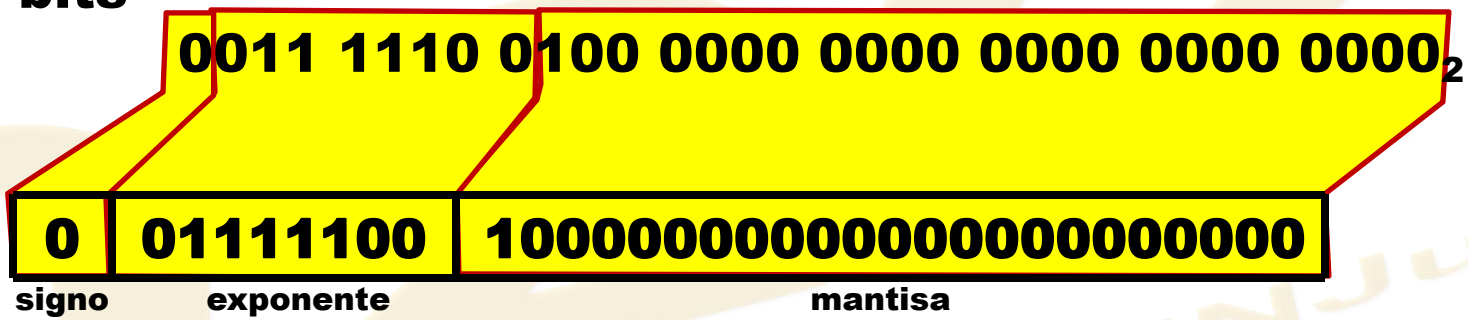
# SISTEMAS DE PUNTO FLOTANTE

## Estándar IEEE 754 Ejemplo → Precisión simple

**Decodificar a decimal ..... $3\ E\ 4\ 0\ 0\ 0\ 0\ 0_{16}$**

1→ Convertir a binario de 4 bits

2→ Identificar y distribuir




3→ Decodificar exponente en exceso $127_{10}$

$$01111100_2 - 01111111_2 = -11_2$$

$$124_{10} - 127_{10} = -3_{10}$$

4→ Reponer el bit implícito a la mantisa

$$1, \longrightarrow 10000...0 = 1{,}1_2$$

5→ Valor final

$$(1{,}1 \times 10^{-11})_2 = 0{,}0011_2$$

$$((1x2^0 + 1x2^{-1})*2^{-3})_{10} = +0{,}1875_{10}$$