Sistemas Numéricos

Semana X

Guía Práctica - Sistemas Numéricos

Ing. José Óscar Sánchez Rendón

Ingeniería de Sistemas
Universidad Tecnológica de Pereira
- Pereira OCT 2025 -

Índice

1 Introducción a las Bases Numéricas

1.1 ¿Qué es una base numérica?

Una **base numérica** es la cantidad de dígitos diferentes disponibles en un sistema de numeración para representar cantidades.

1.2 Sistemas numéricos principales

- **Base 10 (Decimal)**: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- **Base 2 (Binario)**: 0, 1
- **Base 8 (Octal)**: 0, 1, 2, 3, 4, 5, 6, 7
- **Base 16 (Hexadecimal)**: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

1.3 Valor posicional

En cualquier base, el valor de cada dígito depende de su posición:

$$\text{Valor} = d_n \times b^n + d_{n-1} \times b^{n-1} + \cdots + d_1 \times b^1 + d_0 \times b^0$$

Donde:

- d_i = dígito en la posición i
- b = base del sistema
- n = posición más significativa

2 Sistema Binario

2.1 Conceptos básicos

El sistema binario utiliza solo dos dígitos: 0 y 1. Cada dígito se llama **bit**.

2.1.1. Ejemplo de conversión

$$1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11_{10}$$

2.2 Conversión entre bases

2.2.1. Decimal a Binario (Método de divisiones sucesivas)

Ejemplo: Convertir 25_{10} a binario

Operación	Cociente	Residuo
$25 \div 2$	12	$1 \leftarrow$ LSB
$12 \div 2$	6	0
$6 \div 2$	3	0
$3 \div 2$	1	1
$1 \div 2$	0	$1 \leftarrow$ MSB

Resultado: $25_{10} = 11001_2$

2.2.2. Binario a Decimal

Ejemplo: Convertir 110101_2 a decimal

$$1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$= 32 + 16 + 0 + 4 + 0 + 1 = 53_{10}$$

3 Operaciones Aritméticas en Binario

3.1 Suma binaria

3.1.1. Reglas básicas

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 0$, acarreo 1
- $1 + 1 + 1 = 1$, acarreo 1

3.1.2. Ejemplo:

Sumar $1101_2 + 1011_2$

Acarreo:	1	1	1	1	
		1	1	0	1 (13)
+		1	0	1	1 (11)
	1	1	0	0	0 (24)

Verificación: $13 + 11 = 24$

3.2 Resta binaria

3.2.1. Reglas básicas

- 0 - 0 = 0
- 1 - 0 = 1
- 1 - 1 = 0
- 0 - 1 = 1, préstamo 1

3.2.2. Ejemplo:

Restar $1101_2 - 1011_2$

$$
\begin{array}{lccccl}
\text{Préstamo:} & 0 & 1 & 0 & 0 & \\
 & & 1 & 1 & 0 & 1\ (13) \\
- & & 1 & 0 & 1 & 1\ (11) \\
\hline
 & & 0 & 0 & 1 & 0\ (2)
\end{array}
$$

Verificación: 13 - 11 = 2

3.3 Multiplicación binaria

3.3.1. Ejemplo:

Multiplicar $1101_2 \times 101_2$ (13×5)

$$
\begin{array}{ccccccc}
 & & & 1 & 1 & 0 & 1\ (13) \\
\times & & & & 1 & 0 & 1\ (5) \\
\hline
 & & & 1 & 1 & 0 & 1 \\
 & & 0 & 0 & 0 & 0 & \\
1 & 1 & 0 & 1 & & & \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 1\ (65)
\end{array}
$$

Verificación: $13 \times 5 = 65$

3.4 División binaria

3.4.1. Ejemplo:

Dividir $11011_2 \div 101_2$ $(27 \div 5)$

$$
\begin{array}{rcccccl}
 & & & 1 & 0 & 1 & (5,\ \text{cociente}) \\
\cline{2-7}
101 &) & 1 & 1 & 0 & 1 & 1\ (27) \\
 & - & 1 & 0 & 1 & & \\
\cline{3-5}
 & & & 1 & 1 & 1 & \\
 & & - & 1 & 0 & 1 & \\
\cline{4-6}
 & & & & 1 & 0 & (2,\ \text{residuo})
\end{array}
$$

Resultado: Cociente = 101_2 (5), Residuo = 10_2 (2)

4 Operaciones en Otras Bases

4.1 Suma en base 8 (Octal)

4.1.1. Ejemplo:

Sumar $457_8 + 263_8$

$$\begin{array}{rccc} \text{Acarreo:} & 1 & 1 & \\ & 4 & 5 & 7 \\ + & 2 & 6 & 3 \\ \hline & 7 & 4 & 2 \end{array}$$

Verificación:

$$\begin{aligned} 457_8 &= 4 \times 64 + 5 \times 8 + 7 = 303_{10} \\ 263_8 &= 2 \times 64 + 6 \times 8 + 3 = 179_{10} \\ \text{Suma: } 303 + 179 &= 482 \\ 742_8 &= 7 \times 64 + 4 \times 8 + 2 = 482_{10} \checkmark \end{aligned}$$

4.2 Suma en base 16 (Hexadecimal)

4.2.1. Ejemplo:

Sumar $A3F_{16} + 2C8_{16}$

$$\begin{array}{rccc} \text{Acarreo:} & 1 & & \\ & \text{A} & 3 & \text{F} \\ + & 2 & \text{C} & 8 \\ \hline & \text{D} & 0 & 7 \end{array}$$

Verificación:

$$\begin{aligned} A3F_{16} &= 10 \times 256 + 3 \times 16 + 15 = 2623_{10} \\ 2C8_{16} &= 2 \times 256 + 12 \times 16 + 8 = 712_{10} \\ \text{Suma: } 2623 + 712 &= 3335 \\ D07_{16} &= 13 \times 256 + 0 \times 16 + 7 = 3335_{10} \checkmark \end{aligned}$$

5 Representación de Signo en Binario

5.1 Magnitud y Signo

- **Bit más significativo** indica el signo
- $0 =$ positivo, $1 =$ negativo
- Resto de bits representan la magnitud

5.1.1. Ejemplos (8 bits):

- 00000101_2 = +5
- 10000101_2 = -5

Problema: Tiene dos representaciones para el cero (+0 y -0)

5.2 Complemento a 1

- Los números positivos se mantienen igual
- Los negativos se obtienen invirtiendo todos los bits del positivo

5.2.1. Ejemplo (8 bits):

- +5 = 00000101_2
- -5 = 11111010_2

Problema: Sigue teniendo dos representaciones para el cero

5.3 Complemento a 2 (Más utilizado)

5.3.1. Para obtener un número negativo:

1. Invertir todos los bits (complemento a 1)
2. Sumar 1 al resultado

5.3.2. Ejemplo:

Representar -5 en complemento a 2 (8 bits)

$$\begin{aligned} \text{Paso 1: } +5 &= 00000101 \\ \text{Paso 2: Complemento a 1} &= 11111010 \\ \text{Paso 3: Sumar 1} &= 11111011 \\ \text{Resultado: } -5 &= 11111011_2 \end{aligned}$$

Ventaja: Solo una representación para el cero

5.4 Rango de representación

Para n bits en complemento a 2:

- Valor máximo: $2^{n-1} - 1$
- Valor mínimo: -2^{n-1}

5.4.1. Ejemplo (8 bits):

- Rango: -128 a $+127$

6 Ejercicios Prácticos

6.1 Ejercicio 1: Convertir y operar

Convertir a binario y realizar la operación: $38 + 19$

6.1.1. Solución:

$$38_{10} = 100110_2$$
$$19_{10} = 010011_2$$

$$\begin{array}{ccccccc}
\text{Acarreo:} & 1 & 1 & 1 & 1 & & \\
 & & 1 & 0 & 0 & 1 & 1 \\
0 & & & & & & \\
+ & & 0 & 1 & 0 & 0 & 1 \\
1 & & & & & & \\
\hline
 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}$$

$111001_2 = 57_{10}$

6.2 Ejercicio 2: Resta usando complemento a 2

Realizar $15 - 7$ usando complemento a 2 (6 bits)

6.2.1. Solución:

$$\begin{aligned}
+15 &= 001111_2 \\
+7 &= 000111_2 \\
-7: \text{ complemento a } 1 &= 111000_2 \\
+1 &= 111001_2
\end{aligned}$$

Acarreo:	1	1	1	1		
(+15)	0	0	1	1	1	1
+ (-7)	1	1	1	0	0	1
1 (Ignorar acarreo)	0	0	1	0	0	0

Resultado: $001000_2 = 8_{10}$

7 Resumen

- El sistema binario es fundamental en computación
- Las operaciones aritméticas siguen reglas similares al decimal
- El complemento a 2 es el método más eficiente para representar números con signo
- La comprensión de diferentes bases es esencial para programación y arquitectura de computadoras

Práctica recomendada: Realizar conversiones y operaciones regularmente para desarrollar fluidez en el manejo de diferentes bases numéricas.