



**UAEM** | Universidad Autónoma  
del Estado de México

# Lógica Secuencial y Combinatoria



CENTRO UNIVERSITARIO  
UAEM ZUMPANGO

Dr. Arturo Redondo Galván





## UNIDAD II

Desarrollar cálculos distintos sistemas de numeración y llevar a cabo operaciones aritméticas en el álgebra Booleana y optimizar funciones mediante métodos de minimización de éstas.



# SISTEMAS NUMÉRICOS



## OBJETIVOS:

- Conocer los diferentes tipos de sistemas numéricos.
- Realizar conversiones entre diferentes bases.
- Desarrollar operaciones en binario.
- Analizar los códigos binarios.



## Sistemas numéricos: decimal (1/5)

- El sistema decimal, también conocido como **sistema de base 10**, se componen de 10 símbolos (0,1,2,3,4,5,6,7,8,9) con los que se puede expresar cualquier cantidad.
- El valor de cada digito depende de su posición.

2643.43



2 millares  
6 centenas  
4 decenas  
3 unidades  
4 décimos  
3 centésimas

$$2 \times 10^3 + 6 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 3 \times 10^{-2}$$



## Sistemas numéricos: decimal (2/5)

- Generalizando, un número expresado en un sistema base  $r$  tiene coeficientes multiplicados por potencias de  $r$ :

$$a_n \cdot r^n + a_{n-1} \cdot r^{n-1} + \dots + a_2 \cdot r^2 + a_1 \cdot r^1 + a_0 \cdot r^0 + a_{-1} \cdot r^{-1} + a_{-2} \cdot r^{-2} + \dots + a_{-m} \cdot r^{-m}$$

- El valor del subíndice indica la posición y, por tanto, la potencia de la base.



## Sistemas numéricos: binario (3/5)

- Sus coeficientes sólo pueden tener dos valores: **0** o **1**, (base dos).
- Cada coeficiente  $a_j$  se multiplica por  $2^j$ .

$$11011.01_2 \quad \longrightarrow \quad 27.25_{10}$$

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 27.25_{10}$$



## Sistemas numéricos: octal (4/5)

- El sistema numérico octal es un sistema base 8 que tiene ocho dígitos; 0, 1, 2, 3, 4, 5, 6 y 7.
- Cada coeficiente  $a_j$  se multiplica por  $8^j$ .

$$127.4_8 \quad \longrightarrow \quad 87.5_{10}$$

$$1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} = 87.5_{10}$$



## Sistemas numéricos: hexadecimal (5/5)

- En el sistema hexadecimal (base 16), se utilizan los números del 0 al 9 y las letras A, B, C, D E y F, para los dígitos 10, 11, 12, 13, 14 y 15 respectivamente.

$$B65F_{16} \quad \longrightarrow \quad 46687_{10}$$

$$11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = 46687_{10}$$



## Conversiones de base numérica (1/6)

- La conversión de un número base  $r$  a decimal se efectúa expandiendo el número a una serie de potencias y sumando todos los términos.
- Si el número tiene punto, para realizar la conversión se separa la parte entera de la parte fraccionaria y se procede a la operación.
- La conversión de un entero decimal en un número base  $r$  se efectúa dividiendo el número y todos sus cocientes sucesivos entre  $r$  y acumulando los residuos.



## Conversiones de base numérica (2/6)

### Conversión de decimal a binario

1. Dividir el número decimal entre 2.
2. Divida cada uno de los cocientes entre 2 hasta que el cociente sea 0.
3. Tome cada uno de los residuos iniciando con  $a_0$  hasta  $a_n$ , donde  $a_n$  es el último residuo.



## Conversiones de base numérica (3/6)

### Conversión de decimal a binario

Ejemplo: convierta el número 41 a binario.

Entero	Residuo
41	
20	1
10	0
5	0
2	1
1	0
0	1

respuesta= 1010001



## Conversiones de base numérica (4/6)

### Conversión de decimal a binario

Ejemplo: convierta el número 0.6875 a binario.

	Entero	Fracción
$0.6875 \times 2$	1	0.3750
$0.3750 \times 2$	0	0.7500
0.7500	1	0.5000
0.5000	1	0.0000

↓  
**respuesta= 0.1011**



## Conversiones de base numérica (5/6)

### Conversión de hexadecimal a binario

La conversión de hexadecimal a binario se efectúa tomando cada dígito hexadecimal y representándolo en binario con cuatro bits.

Ejemplo: convierta el número  $2C6B.F2_{16}$  a binario.

10	1100	0110	1011	.	1111	0010
2	C	6	B		F	2



## Conversiones de base numérica (6/6)

### Conversión de octal a binario

La conversión de octal a binario se efectúa tomando cada dígito octal y representándolo en binario con tres bits.

Ejemplo: convierta el número  $26153.7406_8$  a binario.

10	110	001	101	011	.	111	100	000	110
2	6	1	5	3		7	4	0	6



## Complementos (1/3)

- Los complementos se usan para simplificar la operación de resta y para efectuar manipulaciones lógicas en las computadoras digitales.
- Hay dos tipos de complementos:
  - Complemento a la base ( $r$ )
  - Complemento a la base disminuida ( $r-1$ )



## Complementos (2/3)

### Complemento a la base disminuida:

Dado un número  $N$  en base  $r$  que tiene  $n$  dígitos, el complemento a  $(r-1)$  de  $N$  se define como  $(r^n - 1) - N$ .

Complemento a 9:

546700



$999999 - 546700 = 453299$

Complemento a 1:

1011000



$1111111 - 1011000 = 0100111$



## Complementos (3/3)

### Complemento a la base:

Dado un número  $N$  en base  $r$  que tiene  $n$  dígitos, el complemento a  $(r)$  de  $N$  se define como  $r^n - N$ .

Complemento a 10:

246700



$1000000 - 246700 = 753300$

Complemento a 1:

0110111



$10000000 - 0110111 = 1001001$



## Resta con complementos (1/7)

Al realizar una resta comúnmente se utiliza el método de préstamo de la posición más alta cuando el minuendo es menor que el sustraendo. Sin embargo en un sistema digital resulta más eficiente el método de sustracción por complementos.



## Resta con complementos (2/7)

### Resta con complemento a $r$

El procedimiento de sustracción utilizando el complemento de  $r$  para dos números positivos ( $M - N$ ) es:

1. Sumar  $M$  al complemento  $r$  de  $N$ .
2. Verificar resultado. ¿Existe acarreo final?
  - **Si:** El acarreo se descarga y el resultado es el que se obtuvo en el paso 1.
  - **No:** se toma el complemento  $r$  del resultado del paso 1 y se le antepone un signo negativo.



## Resta con complementos (3/7)

### Resta con complemento a $r$

Ejemplo: utilice el complemento a 10 para realizar  $M - N$  con los números decimales:  $M= 03250$  y  $N= 72532$

$$\begin{array}{r} M= 03250 \\ N= 72532 \quad \text{complemento a 10: } 27468 \\ \hline \end{array} \quad \begin{array}{r} 03250 \\ + 27468 \\ \hline 30718 \end{array}$$

No hay acarreo por lo tanto se obtiene el complemento a 10 y se agrega un signo negativo, el resultado es **-69282**



## Resta con complementos (4/7)

### Resta con complemento a $r$

Ejemplo: utilice el complemento a 2 para realizar  $M - N$  con los números binarios:  $M = 1010100$  y  $N = 1000100$

$$\begin{array}{r} M = 1010100 \\ N = 1000100 \quad \text{complemento a 2: } 0111100 \\ \hline \phantom{M = } \phantom{N = } \phantom{\text{complemento a 2: }} 1010100 \\ + \phantom{M = } \phantom{N = } \phantom{\text{complemento a 2: }} 0111100 \\ \hline 10001000 \end{array}$$

El acarreo se descarta y el resultado es **10000**.



## Resta con complementos (5/7)

### Resta con complemento a $r - 1$

El procedimiento de sustracción utilizando el complemento de  $r - 1$  para dos números positivos ( $M - N$ ) es:

1. Sumar  $M$  al complemento  $r - 1$  de  $N$ .
2. Verificar resultado. ¿Existe acarreo final?
  - **Si:** sumarlo al dígito menos significativo del resultado del paso 1.
  - **No:** se toma el complemento  $r - 1$  del resultado del paso 1 y se le antepone un signo negativo.



## Resta con complementos (6/7)

### Resta con complemento a $r - 1$

Ejemplo: utilice el complemento a 9 para realizar  $M - N$  con los números decimales:  $M = 72532$  y  $N = 03250$

$M = 72532$

$N = 03250$

complemento a 9: 96749

$$\begin{array}{r} 72532 \\ + 96749 \\ \hline 169281 \\ \phantom{169281} + 1 \\ \hline \end{array}$$

El acarreo se adiciona al bits menos significativo del resultado, por lo tanto, el resultado es **69282**.



## Resta con complementos (7/7)

### Resta con complemento a $r - 1$

Ejemplo: utilice el complemento a 1 para realizar  $M - N$  con los números binarios:  $M = 1000100$  y  $N = 1010100$

$$\begin{array}{r} M = 1000100 \\ N = 1010100 \quad \text{complemento a 1: } 0101011 \\ \hline \phantom{M = } \phantom{N = } \phantom{\text{complemento a 1: }} 1010100 \\ \phantom{M = } \phantom{N = } \phantom{\text{complemento a 1: }} + 0101011 \\ \hline \phantom{M = } \phantom{N = } \phantom{\text{complemento a 1: }} 1101111 \end{array}$$

No hay acarreo por lo tanto se obtiene el complemento a 1 y se agrega un signo negativo, el resultado es **-10000**.



## Códigos binarios (1/5)

Los sistemas digitales emplean señales que tienen dos valores distintos, y elementos de circuito que tienen dos estados estables.

Cualquier elemento discreto de información distinto dentro de un grupo de cantidades se puede representar con un código binario.

Un código binario de  $n$  bits es un grupo de  $n$  bits que puede tener hasta  $2^n$  combinaciones distintas de unos y ceros; cada combinación representa un elemento del conjunto que está codificado.



## Códigos binarios (2/5)

### Código BCD

El código **decimal codificado en binario** (BCD) representa los 10 dígitos decimales con cuatro bits. Las combinaciones de bits del 10 al 16 no se utilizan.

Decimal	BCB
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001



## Códigos binarios (3/5)

### Otros códigos

Decimal	BCD	2421	Exceso-3	8 4-2-1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	111	1100	1111



## Códigos binarios (4/5)

### Código de Gray

El **código de Gray** se utiliza para representar datos digitales obtenidos por conversión de datos analógicos. Su ventaja es que la diferencia entre dos números consecutivos en este código es de un solo bit.

Decimal	Código Gray
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101



## Códigos binarios (5/5)

### Código de ASCII

Es el código estándar para los caracteres alfanuméricos. Utiliza siete bits para codificar 128 caracteres. Como las computadoras trabajan con ocho bits, el bit restante se utiliza para otras aplicaciones, por ejemplo como bit de paridad para detectar errores.

Carácter	ASCII
A	1000001
B	1000010
C	1000011
D	1000100
E	1000101
F	1000110
G	1000111
H	1001000
I	1001001
J	1001010



## REFERENCIAS (1/1)

1. Mandado, Enrique *“Sistemas electrónicos digitales” 7ª Edición Ed. Marcombo (1992) Barcelona ISBN 8426711707.*
2. Morris, M. Mano *“Diseño digital” Ed. Prentice Hall (2003) México ISBN 9702604389.*
3. Tocci, Ronald J; Widmer, Noel S. *“Sistemas digitales, principios y aplicaciones” Ed. Pearson Educación de México (2003) México 8ª Edición ISBN 9702602971.*
4. Hill, Frederick J.; Peterson, Gerald R. *“Introduction to switching theory and logical design” Ed. John Wiley & sons (1981) New York 3ª Edición ISBN 0471042730.*
5. Téllez, G.R. *“Notas para el curso de circuitos lógicos” Vol. 1 Impresos Instituto Politécnico Nacional (1990) México.*
6. García Guerra, Hugo *“Diseño computacional de circuitos lógicos” Instituto tecnológico de la ciudad de Madero (1995).*
7. Motorota; Texas Instruments; Nacional Semiconductor *“Manual de circuitos integrados digitales. Familia TTL”.*