



**INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA**

**SUBDIRECCIÓN ACADÉMICA  
DIVISIÓN DE INGENIERÍA EN MECATRONICA**

**PROFESOR (A):** DR. JOSE ANGEL NIEVES VAZQUEZ  
**Reporte No.** 1 **Periodo** FEB - JUL 2023

**Nombre del Proyecto** DOCENCIA (PREPARACION DE CLASES, CORRECCION DE EXAMENES, REDACCION.

Objetivo
Realizar actividades que complementen la labor docente que garanticen la calidad en el proceso de enseñanza-aprendizaje.

Meta
3 Instrumentaciones 4 Reportes parciales del SGI 1 Reporte Final del SGI 3 Reportes de Proyectos Individuales

Actividades			
Actividad	Fecha programada de Realización	Evidencia	% avance
Investigación Documental del contenido de las asignaturas	20/02/2023-23/06/2023	Links o archivos electronicos	33%
Preparación de clases de materias de acuerdo al horario de clases asignado en este semestre.	20/02/2023-23/06/2024	archivos electrónicos	33%
Preparación de material didáctico de acuerdo al temario de las asignaturas	20/02/2023-23/06/2025	archivos electrónicos	33%
Elaboración y aplicación de instrumentos de evaluación según instrumentación didáctica	20/02/2023-23/06/2026	Lista de cotejo o guia de observación o rubrica o examen	33%
Proceso de evaluación de los trabajos de los alumnos.	20/02/2023-23/06/2027	Lista de calificación en plataforma SGI	33%
Elaboración de reportes administrativos de las actividades	20/02/2023-23/06/2028	reporte de proyectos individuales en plataforma SGI	33%

Observaciones

--

DR. JOSE ANGEL NIEVES  
VAZQUEZ

ING. YOSAFAT MORTERA  
ELIAS  
Jefe de División de  
Ingeniería Mecatrónica

MCJyS. OFELIA  
ENRIQUEZ ORDAZ  
Subdirector Académico

NOTA: Llenar este formato por cada proyecto asignado y entregar en la semana número 7 el 1er reporte; en la semana 11 2° reporte; y en la semana 18 el reporte final.

# LISTA DE COTEJO INVESTIGACION

## ELECTRÓNICA DIGITAL MTF 1013.

Nombre del estudiante: Martínez Morgado Ana Victoria

Tema: Fundamentos de Sistemas Digitales

Portada	2 %	2 %
Introducción	5 %	5 %
Desarrollo	10 %	10 %
Conclusiones	5 %	5 %
Referencias	3 %	3 %
Entrega en tiempo y forma	5 %	5 %
Total	30 %	30 %

# LISTA DE COTEJO DE PRÁCTICAS

## ELECTRÓNICA DIGITAL MTF 1013

### PRÁCTICA NÚMERO 1.

Nombre del estudiante: Martínez Morgado Ana Victoria

Tema: CONVERSIONES Y OPERACIONES ENTRE SISTEMAS.

Portada	2 %	2 %
Introducción	5 %	5 %
Desarrollo	10 %	10 %
Conclusiones	5 %	5 %
Referencias	3 %	3 %
Entrega en tiempo y forma	5 %	5 %
Total	30 %	30 %



# ELECTRÓNICA DIGITAL

Unidad 1

Fundamentos de sistemas digitales

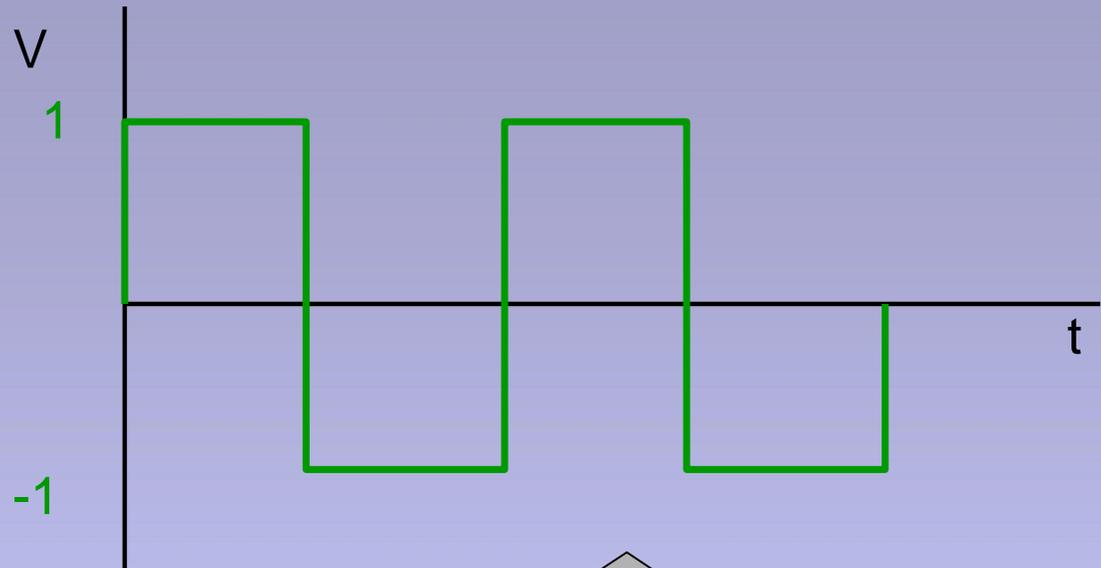
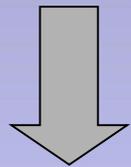
# Índice

1. Introducción a los sistemas digitales.
2. Señales análogas y digitales.
3. Relación entre los sistemas análogos y los sistemas digitales.
4. Sistemas numéricos
5. Binario
6. Octal
7. Hexadecimal
8. Conversión entre sistemas numéricos.
9. Operaciones básicas con diferentes sistemas numéricos.
10. Códigos binarios y alfanuméricos.
11. Gray
12. BCD
13. ASCII
14. UNICODE

# Señal Analógica y Señal Digital

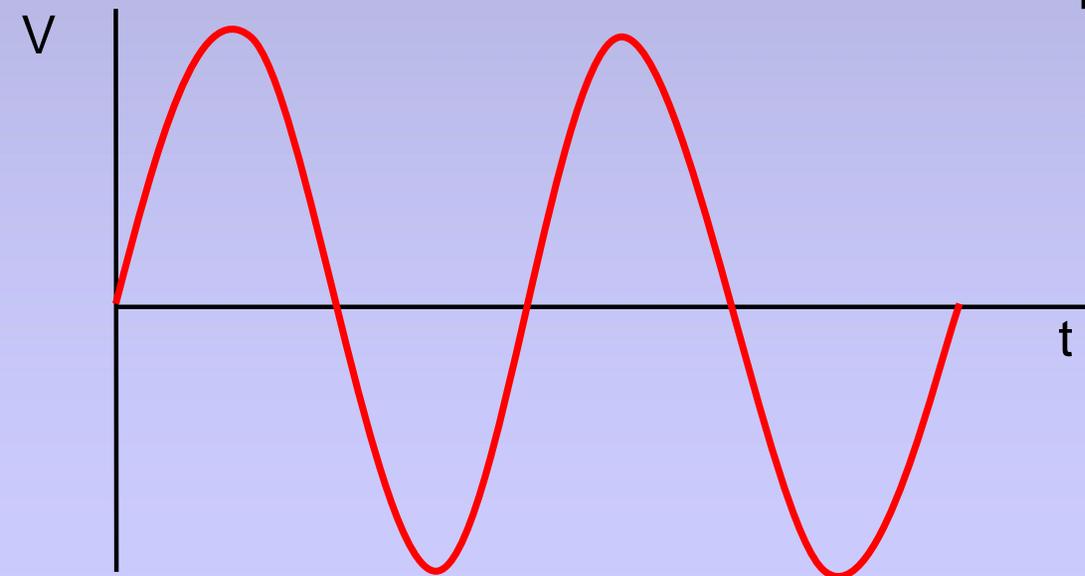
## Señal analógica

Es una señal continua.  
El nº de valores que puede tomar es infinito



## Señal digital

Es una señal discreta.  
Solo puede tomar determinados valores

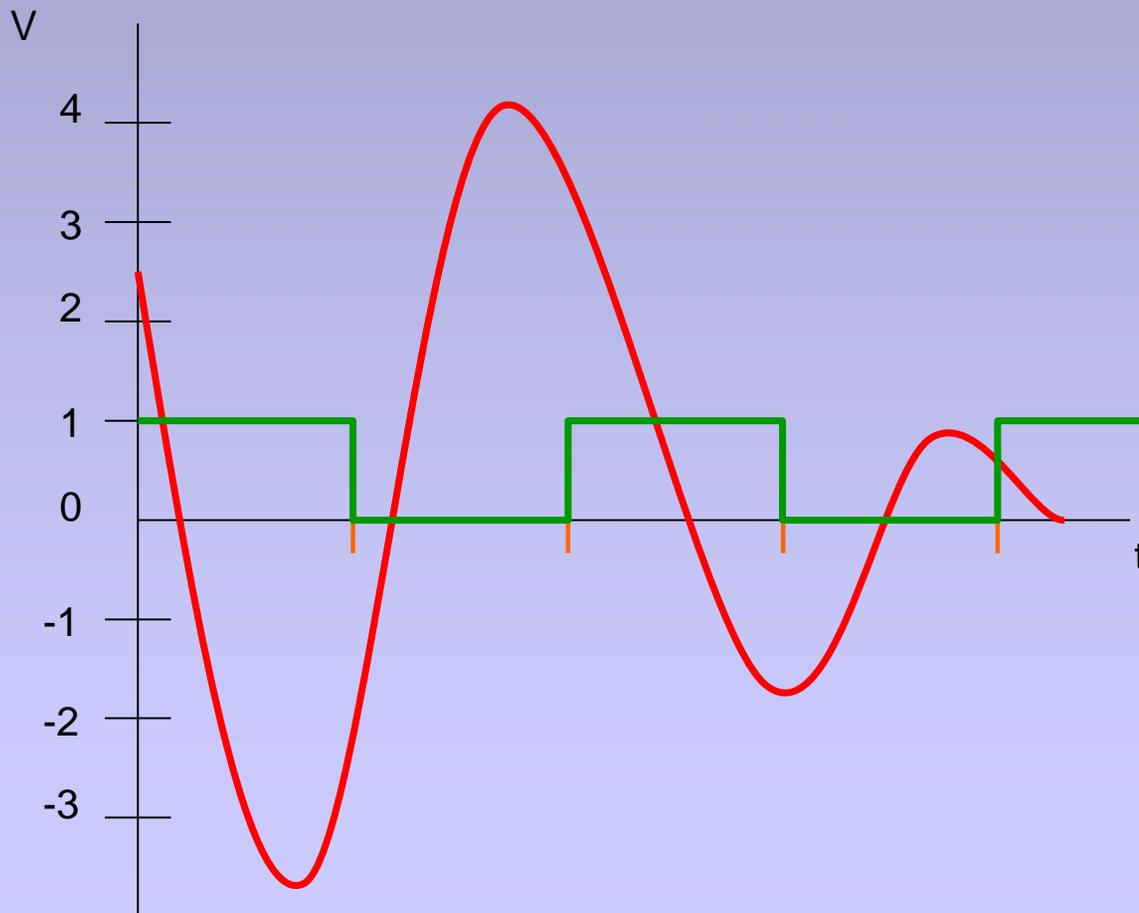


Trabaja con señales que solamente adopta dos estados eléctricos:

1 (circuito cerrado)

0 (circuito abierto)

# Electrónica Digital



Valor Analógico

$(- , 0]$

$(0, + )$

Valor Digital

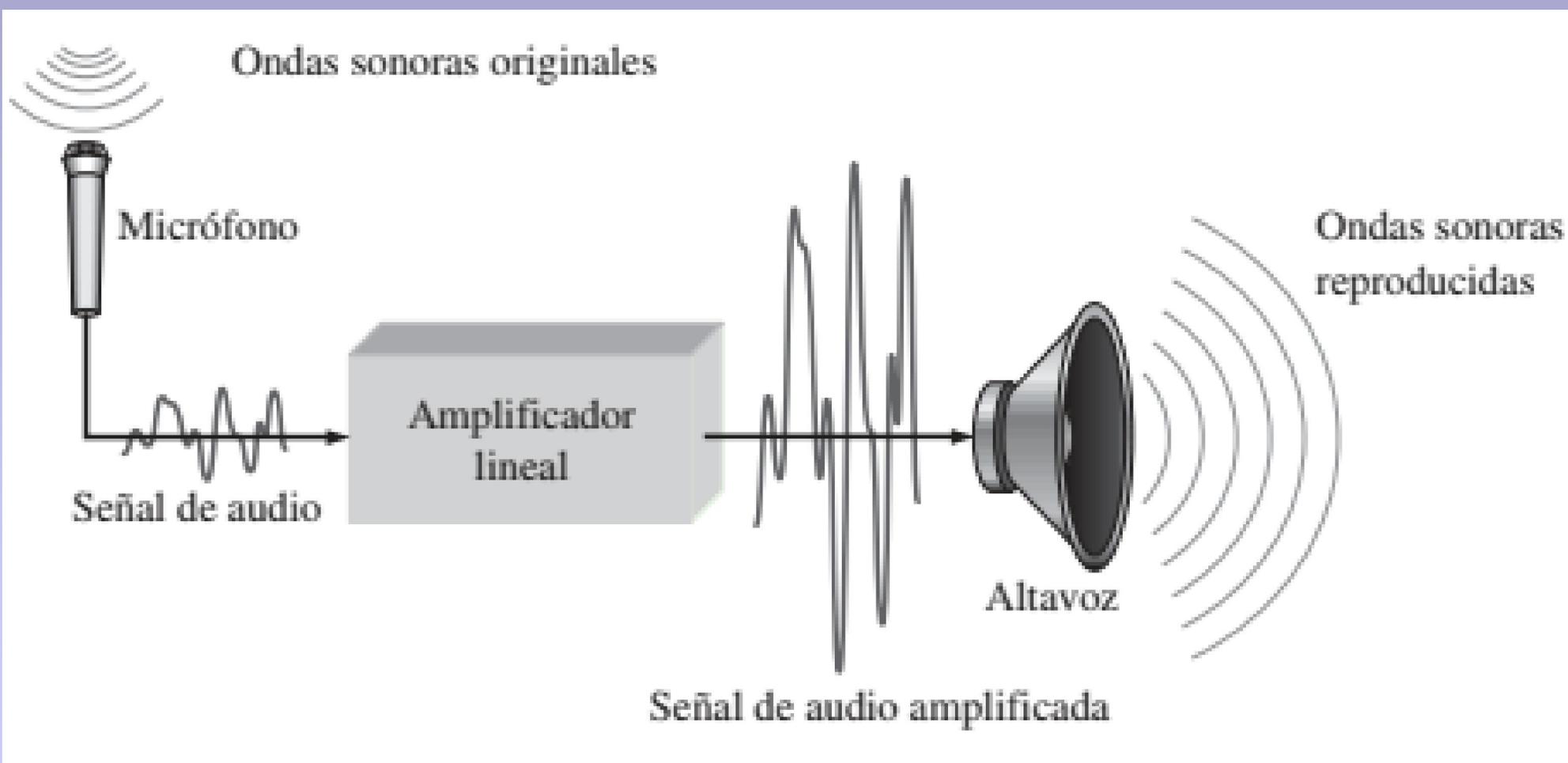
0

1

## Ventajas:

- Fáciles de reconfigurar
- Interferencias prácticamente nulas
- Coste menor
- Se puede manejar señales de distintas funciones

# Sistema electrónico analógico



# Relación entre los sistemas análogos y los sistemas digitales.

Reproductor de CD



10110011101  
Datos digitales

Convertidor digital-analógico

Reproducción analógica de la señal de audio

Amplificador lineal



Altavoz

Ondas sonoras

# Sistemas numéricos (Decimal)

- Todos estamos familiarizados con el sistema de numeración decimal porque utilizamos los números decimales todos los días.
- El sistema de numeración decimal utiliza diez dígitos de 0 a 9.
- El sistema de numeración decimal es un sistema en base 10.  
El valor de un dígito se determina por su posición dentro del número

$10^2$   $10^1$   $10^0$   $10^{-1}$   $10^{-2}$   $10^{-3}$  . . .



Coma decimal

# Ejemplos

Expresar el número decimal 47 como una suma de valores de cada dígito.

**Solución**

Como indican sus respectivas posiciones, el dígito 4 tiene un peso de 10, que es  $10^1$ . El dígito 7 tiene un peso de 1, que es  $10^0$ .

$$\begin{aligned}47 &= (4 \times 10^1) + (7 \times 10^0) \\ &= (4 \times 10) + (7 \times 1) = \mathbf{40} + 7\end{aligned}$$

**Problema relacionado\*** Determinar el valor de cada dígito en el número 939.

Expresar el número decimal 568,23 como suma de los valores de cada dígito.

**Solución**

El dígito 5 de la parte entera tiene un peso de 100, que es  $10^2$ , el dígito 6 tiene un peso de 10, que es  $10^1$ , el dígito 8 tiene un peso de 1, que es  $10^0$ ; el dígito 2 de la parte fraccionaria tiene un peso de 0,1, es decir,  $10^{-1}$ , y el dígito 3 de la parte fraccionaria tiene un peso de 0,01, que es  $10^{-2}$ .

$$\begin{aligned}568,23 &= (5 \times 10^2) + (6 \times 10^1) + (8 \times 10^0) + (2 \times 10^{-1}) + (3 \times 10^{-2}) \\ &= (5 \times 100) + (6 \times 10) + (8 \times 1) + (2 \times 0,1) + (3 \times 0,01) \\ &= \mathbf{500} + \mathbf{60} + \mathbf{8} + \mathbf{0,2} + \mathbf{0,03}\end{aligned}$$

**Problema relacionado** Determinar el valor de cada dígito del número 67,924.

# Sistemas numéricos (Binario)

- El sistema de numeración binario utiliza dos dígitos (bits) 1 y 0.
- El sistema de numeración binario es un sistema en base 2.
- El valor de un bit se determina por su posición dentro del número
- A continuación se presenta la estructura de pesos de un número binario

$$2^{n-1} \dots 2^3 2^2 2^1 2^0 , 2^{-1} 2^{-2} \dots 2^{-n}$$

 Coma binaria

- A continuación se muestra los pesos binarios

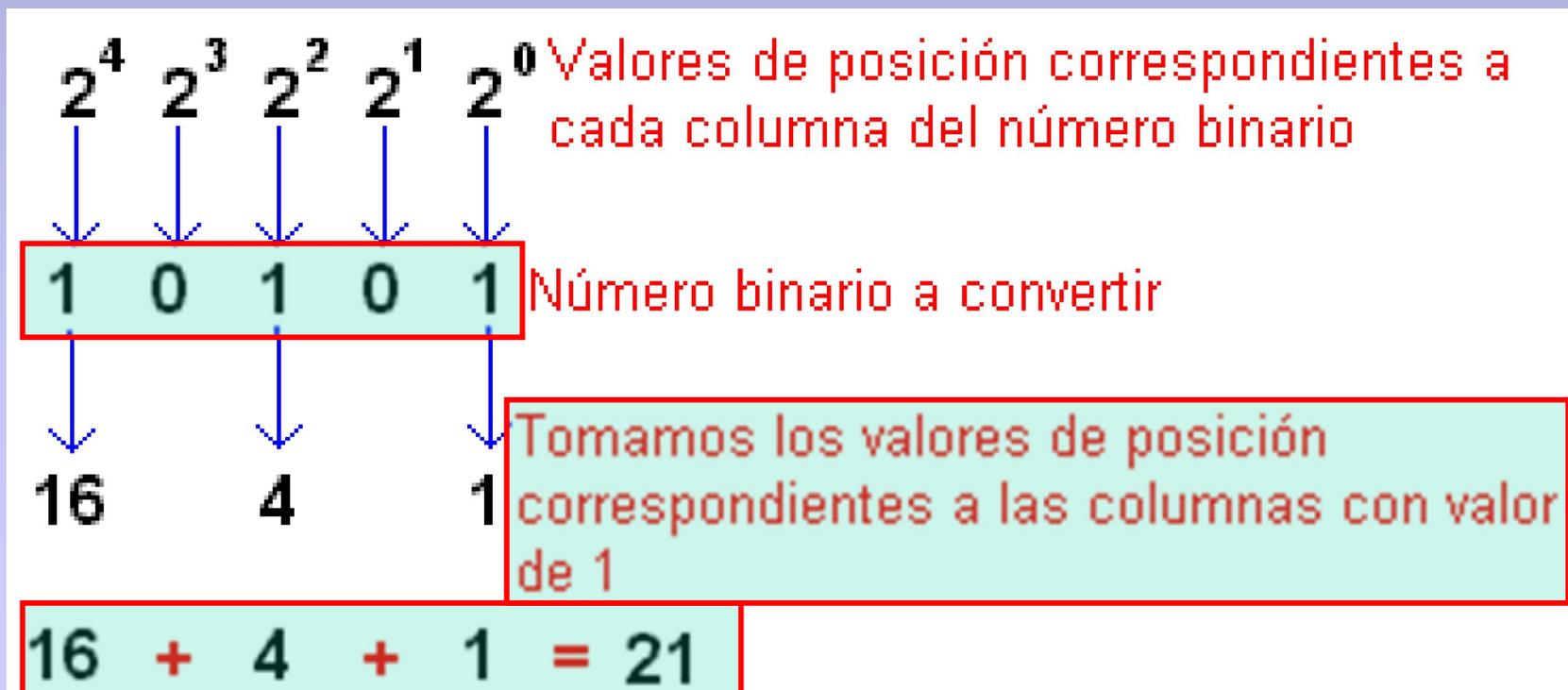
Potencias positivas de dos (números enteros)									Potencias negativas de dos (números fraccionarios)					
$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$
256	128	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16	1/32	1/64
									0,5	0,25	0,125	0,0625	0,03125	0,015625

# Conversión de un número Binario a Decimal

- Para convertir un número binario a decimal es necesario tener en cuenta los pasos que muestran en el siguiente ejemplo:

Transformar el número **10101** a número decimal

- Tomamos los valores de posición correspondiente a las columnas donde aparezcan únicamente unos (1)
- Sumamos los valores de posición para identificar el número decimal equivalente



# Ejemplos

Convertir el número entero binario 1101101 a decimal.

**Solución** Se determina el peso de cada bit que está a 1, y luego se obtiene la suma de los pesos para obtener el número decimal.

$$\begin{aligned} \text{Peso: } & 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \\ \text{Número binario: } & 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \\ 1101101 &= 2^6 + 2^5 + 2^3 + 2^2 + 2^0 \\ &= 64 + 32 + 8 + 4 + 1 = \mathbf{109} \end{aligned}$$

**Problema relacionado** Convertir el número binario 10010001 a decimal.

Convertir el número binario fraccionario 0,1011 a decimal.

**Solución** Se determina el peso de cada bit que está a 1, y luego se suman los pesos para obtener la fracción decimal.

$$\begin{aligned} \text{Peso: } & 2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4} \\ \text{Número binario: } & 0, \ 1 \ 0 \ 1 \ 1 \\ 0,1011 &= 2^{-1} + 2^{-3} + 2^{-4} \\ &= 0,5 + 0,125 + 0,0625 = \mathbf{0,6875} \end{aligned}$$

**Problema relacionado** Convertir el número binario 10,111 a decimal.

# Ejercicios

## Conversión Binario a Decimal

100
111
1010
11101
01101
010001
110011
011
11100101
1000
11011100



# Sistemas numéricos (Octal)

- El sistema de numeración octal utiliza ocho dígitos del 0 al 7.
- El sistema de numeración octal es un sistema en base 8.
- Contar en octal es parecido a contar en decimal, excepto que los dígitos 8 y 9 no se usan.
- Para contar por encima de 7, añadimos otra columna y continuamos así 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27, 30, 31, ...
- Para distinguir los números octales de los números decimales y hexadecimales, utilizaremos el subíndice 8 para indicar un número octal.

# Ejemplo

- La evaluación de un número octal en términos de su equivalente decimal se consigue multiplicando cada dígito por su peso y sumando los productos, como se muestra a continuación para  $2374_8$

Peso:  $8^3$   $8^2$   $8^1$   $8^0$

Número octal: 2 3 7 4

$$\begin{aligned} 2374_8 &= (2 \times 8^3) + (3 \times 8^2) + (7 \times 8^1) + (4 \times 8^0) \\ &= (2 \times 512) + (3 \times 64) + (7 \times 8) + (4 \times 1) \\ &= 1024 + 192 + 56 + 4 = 1276_{10} \end{aligned}$$

# Ejercicios

## Conversión Octal a Decimal

20
51
63
64
102
210
1024
41
33
16
15



# Sistemas numéricos (Hexadecimal)

- El sistema de numeración hexadecimal consta de dieciséis caracteres.
- Diez dígitos numéricos y seis caracteres alfabéticos, del 0 al 9 y de las letras A hasta F.
- Cada dígito hexadecimal se representa mediante un número binario de 4 bits.
- ¿Cómo se continúa contando en hexadecimal cuando se ha llegado a la letra F? Simplemente se inicia otra columna y se continúa contando así: 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, . . .

Con dos dígitos hexadecimales, se puede contar hasta  $FF_{16}$ , que corresponde al decimal 255.

# Ejemplo

Convertir a hexadecimal los siguientes números binarios:

(a) 1100101001010111

(b) 111111000101101001

*Solución*

(a)  $\begin{array}{cccc} \underline{1100} & \underline{1010} & \underline{0101} & \underline{0111} \\ \downarrow & \downarrow & \downarrow & \downarrow \\ C & A & 5 & 7 = \mathbf{CA57}_{16} \end{array}$

(b)  $\begin{array}{ccccc} \underline{0011} & \underline{1111} & \underline{1000} & \underline{1011} & \underline{01001} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & F & 1 & 6 & 9 = \mathbf{3F169}_{16} \end{array}$

En el apartado (b) se han añadido dos ceros para completar el grupo de 4 bits de la izquierda.

*Problema relacionado* Convertir el número binario 1001111011110011100 a hexadecimal.

# Conversión entre sistemas numéricos

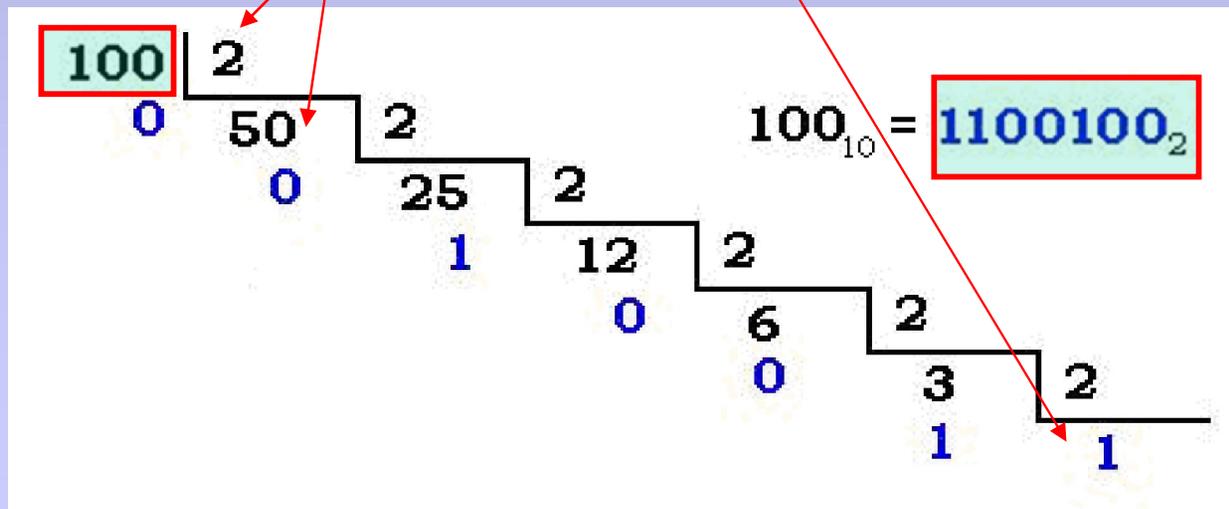
Decimal	Binario	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

# Conversión de un número Decimal a Binario

- Para esta transformación es necesario tener en cuenta los pasos que muestran en el siguiente ejemplo:

Transformar el número 100 a número binario

- Dividir el número 100 entre 2
- Dividir el cociente obtenido por 2 y repetir el mismo procedimiento hasta que el cociente sea 1.
- El número binario se forma tomando como primer dígito el último cociente, seguidos por los residuos obtenidos en cada división, seleccionándolos de derecha a izquierda, como se muestra en el siguiente esquema.



# Ejercicios

## Conversión Decimal a Binario

20
51
63
64
102
210
1024
41
33
16
15



# Conversión de un número Octal a Decimal

Peso:  $8^3$   $8^2$   $8^1$   $8^0$

Número octal: 2 3 7 4

$$\begin{aligned} 2374_8 &= (2 \times 8^3) + (3 \times 8^2) + (7 \times 8^1) + (4 \times 8^0) \\ &= (2 \times 512) + (3 \times 64) + (7 \times 8) + (4 \times 1) \\ &= 1024 + 192 + 56 + 4 = 1276_{10} \end{aligned}$$

# Conversión de un número Octal a Binario

El sistema octal es una forma conveniente de representar los números binarios, aunque no es tan comúnmente utilizado como el hexadecimal. Cada dígito octal se representa mediante **tres bits**, como se muestra a continuación.

<b>Dígito octal</b>	0	1	2	3	4	5	6	7
<b>Binario</b>	000	001	010	011	100	101	110	111

# Ejemplo

Convertir a binario cada uno de los siguientes números octales:

(a)  $13_8$     (b)  $25_8$     (c)  $140_8$     (d)  $7526_8$

*Solución*

(a)  $\begin{array}{cc} 1 & 3 \\ \downarrow & \downarrow \\ \overline{011} & \overline{011} \end{array}$

(b)  $\begin{array}{cc} 2 & 5 \\ \downarrow & \downarrow \\ \overline{010} & \overline{101} \end{array}$

(c)  $\begin{array}{ccc} 1 & 4 & 0 \\ \downarrow & \downarrow & \downarrow \\ \overline{001} & \overline{100} & \overline{000} \end{array}$

(d)  $\begin{array}{cccc} 7 & 5 & 2 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \overline{111} & \overline{101} & \overline{010} & \overline{110} \end{array}$

# Ejercicios

## Conversión Octal a Binario

20
51
63
64
102
210
1024
41
33
16
15



# Conversión Binario a octal

Convertir a octal cada uno de los siguientes números binarios:

(a) 110101    (b) 101111001    (c) 100110011010    (d) 11010000100

*Solución*

$$\begin{array}{c} \text{(a)} \quad \overbrace{110} \quad \overbrace{101} \\ \downarrow \quad \downarrow \\ 6 \quad 5 = 65_s \end{array}$$

$$\begin{array}{c} \text{(b)} \quad \overbrace{101111} \quad \overbrace{001} \\ \downarrow \quad \downarrow \quad \downarrow \\ 5 \quad 7 \quad 1 = 571_s \end{array}$$

$$\begin{array}{c} \text{(c)} \quad \overbrace{100110} \quad \overbrace{011010} \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 4 \quad 6 \quad 3 \quad 2 = 4632_s \end{array}$$

$$\begin{array}{c} \text{(d)} \quad \overbrace{011010} \quad \overbrace{000100} \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 3 \quad 2 \quad 0 \quad 4 = 3204_s \end{array}$$

*Problema relacionado*    Convertir a octal el número binario 1010101000111110010.

# Ejercicios

## Conversión Binario a octal

100
111
1010
11101
01101
010001
110011
011
11100101
1000
11011100



# Conversión de un número Hexadecimal a Binario

La conversión entre hexadecimal y binario es **directa y muy fácil**. El sistema hexadecimal se usa ampliamente para representar los números binarios en programación, salidas de impresora y displays.

Determinar los números binarios correspondientes a los siguientes números hexadecimales:

(a)  $10A4_{16}$     (b)  $CF8E_{16}$     (c)  $9742_{16}$

*Solución*

(a)  $\begin{array}{cccc} 1 & 0 & A & 4 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \underline{1} & \underline{0000} & \underline{1010} & \underline{0100} \end{array}$

(b)  $\begin{array}{cccc} C & F & 8 & E \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \underline{1100} & \underline{1111} & \underline{1000} & \underline{1110} \end{array}$

(c)  $\begin{array}{cccc} 9 & 7 & 4 & 2 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \underline{1001} & \underline{0111} & \underline{0100} & \underline{0010} \end{array}$

En el apartado (a), el MSB se entiende que tiene tres ceros delante del 1 para formar un grupo de 4 bits.

*Problema relacionado*    Convertir el número hexadecimal  $6BD3$  a binario.

# Conversión de un número Binario a Hexadecimal

La conversión entre hexadecimal y binario es directa y muy fácil  
El sistema hexadecimal se usa ampliamente para representar los números binarios en programación, salidas de impresora y displays.

Convertir a hexadecimal los siguientes números binarios:

(a) 1100101001010111

(b) 111111000101101001

*Solución*

(a)  $\begin{array}{cccc} \underline{1100} & \underline{1010} & \underline{0101} & \underline{0111} \\ \downarrow & \downarrow & \downarrow & \downarrow \\ C & A & 5 & 7 \end{array} = \mathbf{CA57}_{16}$

(b)  $\begin{array}{ccccc} \underline{0011} & \underline{1111} & \underline{1000} & \underline{1011} & \underline{01001} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & F & 1 & 6 & 9 \end{array} = \mathbf{3F169}_{16}$

En el apartado (b) se han añadido dos ceros para completar el grupo de 4 bits de la izquierda.

*Problema relacionado*

Convertir el número binario 1001111011110011100 a hexadecimal.

# Conversión de un número Hexadecimal a Decimal

Primero, convertir el número hexadecimal a binario, y después, el binario a decimal.

Convertir los siguientes números hexadecimales a decimal:

(a)  $1C_{16}$       (b)  $A85_{16}$

**Solución**      Recuerde que primero se hace la conversión del número hexadecimal a binario y luego a decimal.

$$(a) \quad \begin{array}{cc} 1 & C \\ \downarrow & \downarrow \\ \overbrace{0001} & \overbrace{1100} \\ 00011100 & = 2^4 + 2^3 + 2^2 = 16 + 8 + 4 = \mathbf{28}_{10} \end{array}$$

$$(b) \quad \begin{array}{ccc} A & 8 & 5 \\ \downarrow & \downarrow & \downarrow \\ \overbrace{1010} & \overbrace{1000} & \overbrace{0101} \\ 101010000101 & = 2^{11} + 2^9 + 2^7 + 2^2 + 2^0 = 2048 + 512 + 128 + 4 + 1 = \mathbf{2693}_{10} \end{array}$$

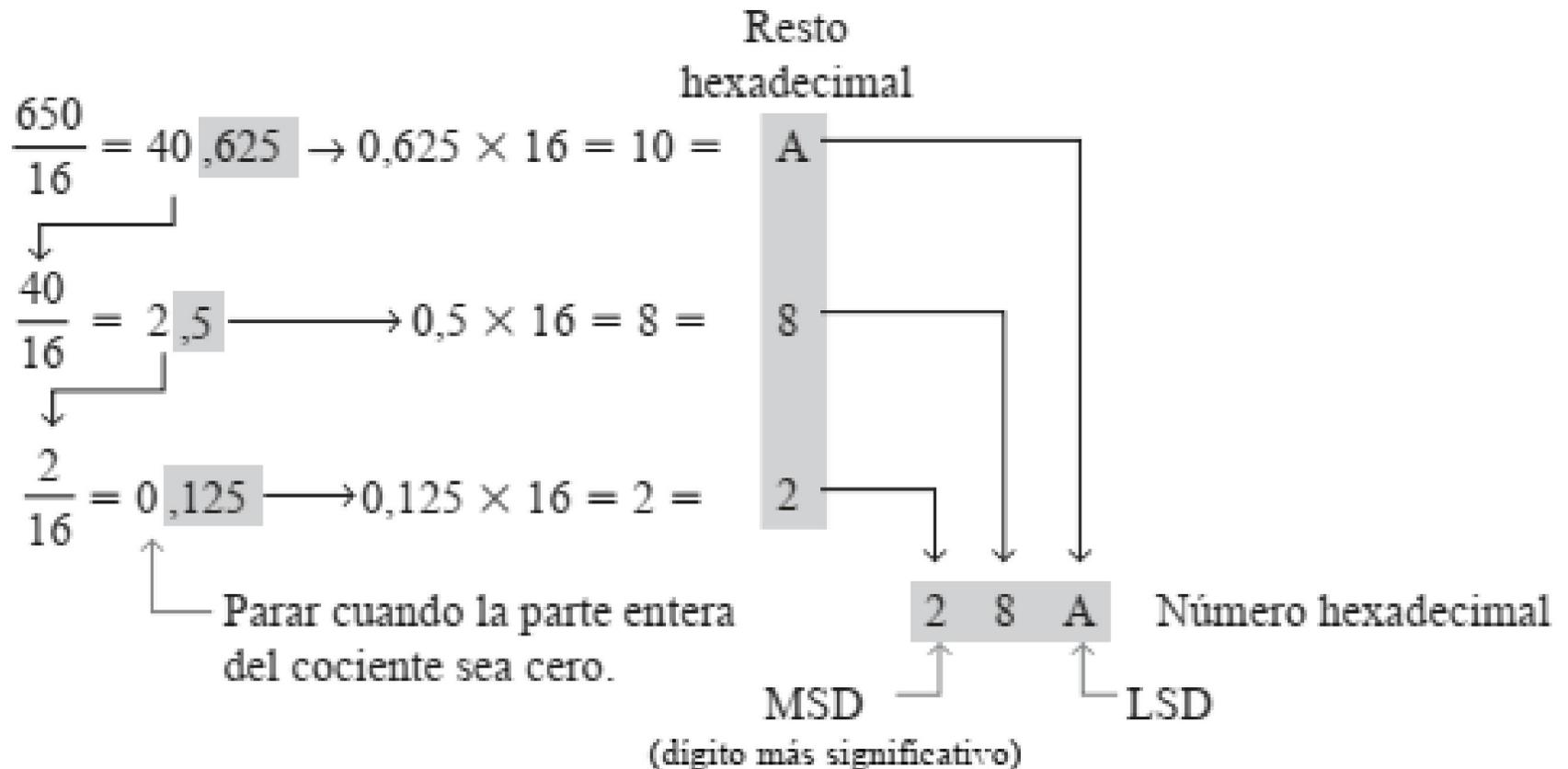
**Problema relacionado**      Convertir a decimal el número hexadecimal 6BD.

# Conversión de un número Decimal a Hexadecimal

Primero, convertir el número hexadecimal a binario, y después, el binario a decimal.

Convertir el número decimal 650 en hexadecimal mediante el método de división sucesiva por 16.

*Solución*

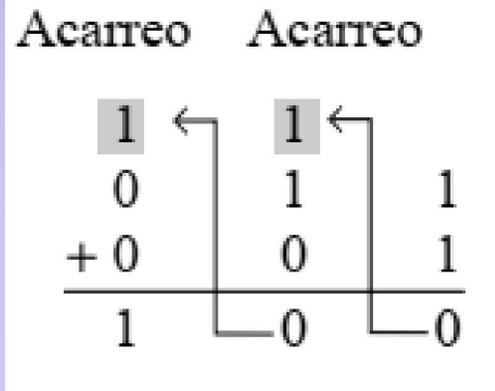


*Problema relacionado* Convertir a hexadecimal el número decimal 2591.

# Operaciones Básicas con diferentes sistemas numéricos

Las cuatro reglas básicas para sumar dígitos binarios son:

$0 + 0 = 0$	Suma 0 con acarreo 0
$0 + 1 = 1$	Suma 1 con acarreo 0
$1 + 0 = 1$	Suma 1 con acarreo 0
$1 + 1 = 10$	Suma 0 con acarreo 1



# Suma binaria

Sumar los siguientes números binarios:

**(a)**  $11 + 11$       **(b)**  $100 + 10$       **(c)**  $111 + 11$       **(d)**  $110 + 100$

**Solución**

La suma decimal equivalente también se muestra como referencia.

<b>(a)</b>	$11$	$3$	<b>(b)</b>	$100$	$4$	<b>(c)</b>	$111$	$7$	<b>(d)</b>	$110$	$6$
	$\underline{+11}$	$\underline{+3}$		$\underline{+10}$	$\underline{+2}$		$\underline{+11}$	$\underline{+3}$		$\underline{+100}$	$\underline{+4}$
	<b>110</b>	<b>6</b>		<b>110</b>	<b>6</b>		<b>1010</b>	<b>10</b>		<b>1010</b>	<b>10</b>

**Problema relacionado** Sumar  $1111$  y  $1100$ .

# Resta binaria

- Las cuatro reglas básicas son:

$$\begin{array}{l} 0 - 0 = 0 \\ 1 - 1 = 0 \\ 1 - 0 = 1 \\ 10 - 1 = 1 \end{array}$$

Realizar las siguientes restas binarias:

(a)  $11 - 01$     (b)  $11 - 10$

*Solución*

$$\begin{array}{r} \text{(a)} \quad 11 \quad 3 \\ \quad \underline{-01} \quad \underline{-1} \\ \quad \mathbf{10} \quad \mathbf{2} \end{array} \qquad \begin{array}{r} \text{(b)} \quad 11 \quad 3 \\ \quad \underline{-10} \quad \underline{-2} \\ \quad \mathbf{01} \quad \mathbf{1} \end{array}$$

En este ejemplo no se han generado acarreo negativos. El número binario 01 es el mismo que el 1.

*Problema relacionado*    Restar 100 de 111.

# Resta binaria

Restar 011 de 101.

*Solución*

$$\begin{array}{r} 101 \quad 5 \\ -011 \quad -3 \\ \hline 010 \quad 2 \end{array}$$

Examinemos detalladamente cómo se ha obtenido la resta de los dos números binarios, ya que es necesario un acarreo negativo. Empezamos por la columna de la derecha.

Columna izquierda:

Cuando se acarrea un 1, queda 0, luego  $0 - 0 = 0$ .

Columna central:

Acarreo negativo de 1 de la columna siguiente que da lugar a 10 en esta columna, luego  $10 - 1 = 1$ .

Columna derecha:

$$1 - 1 = 0$$

The diagram shows the binary subtraction  $101 - 011$  with annotations. A vertical line is drawn under the 1 in the rightmost column. An arrow points from the text 'Columna izquierda: Cuando se acarrea un 1, queda 0, luego  $0 - 0 = 0$ .' to the 0 in the middle column of the result. Another arrow points from the text 'Columna central: Acarreo negativo de 1 de la columna siguiente que da lugar a 10 en esta columna, luego  $10 - 1 = 1$ .' to the 1 in the middle column of the result. A third arrow points from the text 'Columna derecha:  $1 - 1 = 0$ ' to the 0 in the rightmost column of the result. The result is written as 010.

$$\begin{array}{r} 0101 \\ -011 \\ \hline 010 \end{array}$$

*Problema relacionado* Restar 101 de 110.

# Multiplicación binaria

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

# Multiplicación binaria

Realizar las siguientes multiplicaciones binarias:

(a)  $11 \times 11$     (b)  $101 \times 111$

*Solución*

$$\begin{array}{r} \text{(a)} \quad \quad \quad 11 \quad \quad 3 \\ \quad \quad \quad \times 11 \quad \quad \times 3 \\ \hline \text{Productos} \left\{ \begin{array}{l} 11 \\ + 11 \end{array} \right. \\ \hline \text{parciales} \quad \quad \quad \mathbf{1001} \end{array}$$

$$\begin{array}{r} \text{(b)} \quad \quad \quad 111 \quad \quad 7 \\ \quad \quad \quad \times 101 \quad \quad \times 5 \\ \hline \text{Productos} \left\{ \begin{array}{l} 111 \\ 000 \end{array} \right. \\ \hline \text{parciales} \quad \quad \quad + 111 \\ \hline \quad \quad \quad \quad \quad \quad \mathbf{100011} \end{array}$$

*Problema relacionado*    Multiplicar  $1101 \times 1010$ .

# División binaria

- La división binaria sigue el mismo procedimiento que la división decimal, como ilustra el Ejemplo.

Realizar las siguientes divisiones binarias:

(a)  $110 \div 11$     (b)  $110 \div 10$

*Solución*

(a)	$\begin{array}{r} \mathbf{10} \\ \underline{11)110} \\ 11 \\ \underline{\phantom{11}00} \\ 000 \end{array}$	$\begin{array}{r} \mathbf{2} \\ \underline{3)6} \\ 6 \\ \underline{\phantom{6}0} \\ 0 \end{array}$	(b)	$\begin{array}{r} \mathbf{11} \\ \underline{10)110} \\ 10 \\ \underline{\phantom{10}10} \\ 10 \\ \underline{\phantom{10}00} \\ 00 \end{array}$	$\begin{array}{r} \mathbf{3} \\ \underline{2)6} \\ 6 \\ \underline{\phantom{6}0} \\ 0 \end{array}$
-----	---	--	-----	--	--

*Problema relacionado*    Dividir 1100 entre 100.

# Suma hexadecimal

- En cualquier columna dada de una suma, pensar en los dos dígitos hexadecimales en términos de sus valores decimales. Por ejemplo,  $5_{16} = 5_{10}$  y  $C_{16} = 12_{10}$ .
- Si la suma de los dos dígitos es  $15_{10}$  o menor, reducir al dígito hexadecimal correspondiente.
- Si la suma de los dos dígitos es mayor que  $15_{10}$ , hay que reducir la suma que excede de  $16_{10}$  y pasar el acarreo de 1 a la siguiente columna.

# Suma hexadecimal

Sumar los siguientes números hexadecimales:

(a)  $23_{16} + 16_{16}$     (b)  $58_{16} + 22_{16}$     (c)  $2B_{16} + 84_{16}$     (d)  $DF_{16} + AC_{16}$

*Solución*

(a) 
$$\begin{array}{r} 23_{16} \\ +16_{16} \\ \hline 39_{16} \end{array}$$
 columna derecha:  $3_{16} + 6_{16} = 3_{10} + 6_{10} = 9_{10} = 9_{16}$   
columna izquierda:  $2_{16} + 1_{16} = 2_{10} + 1_{10} = 3_{10} = 3_{16}$

(b) 
$$\begin{array}{r} 58_{16} \\ +22_{16} \\ \hline 7A_{16} \end{array}$$
 columna derecha:  $8_{16} + 2_{16} = 8_{10} + 2_{10} = 10_{10} = A_{16}$   
columna izquierda:  $5_{16} + 2_{16} = 5_{10} + 2^{10} = 7_{10} = 7_{16}$

(c) 
$$\begin{array}{r} 2B_{16} \\ +84_{16} \\ \hline AF_{16} \end{array}$$
 columna derecha:  $B_{16} + 4_{16} = 11_{10} + 4_{10} = 15_{10} = F_{16}$   
columna izquierda:  $2_{16} + 8_{16} = 2_{10} + 8_{10} = 10_{10} = A_{16}$

(d) 
$$\begin{array}{r} DF_{16} \\ +AC_{16} \\ \hline 18B_{16} \end{array}$$
 columna derecha:  $F_{16} + C_{16} = 15_{10} + 12_{10} = 27_{10}$   
 $27_{10} - 16_{10} = 11_{10} = B_{16}$  con un acarreo de 1  
columna izquierda:  $D_{16} + A_{16} + 1_{16} = 13_{10} + 10_{10} + 1_{10} = 24_{10}$   
 $24_{10} - 16_{10} = 8_{10} = 8_{16}$  con un acarreo de 1

*Problema relacionado* Sumar  $4C_{16}$  y  $3A_{16}$

# Resta hexadecimal

- $-1 + 16$
- $16 + 5 = 21 - A = 21 - 10 = 11$
- 4 5 F
- B = 11
- $\underline{-2 \quad A \quad 1}$
- 1 B E

Restar los siguientes números hexadecimales:

(a)  $84_{16} - 2A_{16}$       (b)  $C3_{16} - 0B_{16}$

*Problema relacionado* Restar  $173_{16}$  de  $BCD_{16}$ .

# Multiplicación hexadecimal

MULTIPL. HEXA.

$$\begin{array}{r} 4B2 \\ \times 23 \\ \hline 12336 \\ 8224 \\ \hline 8^2 34^2 376 \\ -32 \quad -32 \\ \hline A456x \end{array}$$

$$\begin{array}{r} 4B2 \\ \times 20 \\ \hline 000 \\ 8224 \\ \hline 8^2 2240 \\ -16 \\ \hline 9640 \end{array}$$

$$\begin{array}{r} 4C2 \\ \times 21 \\ \hline 4122 \\ 8244 \\ \hline 8^2 28^1 162 \\ -16 \quad -16 \\ \hline 91302 \\ 9D02x \end{array}$$

$2 \times 16 = 32$   
 $3 \times 16 = 48$

# División hexadecimal

DIVISION HEXA.

$$\begin{array}{r} 51 \\ A \overline{) 32B} \\ \underline{-32} \\ 0B \\ \underline{-A} \\ 1 \end{array}$$
$$\begin{array}{r} A \\ \times 5 \\ \hline 50 \\ 48 \\ \hline 32 \end{array}$$

$2 \times 16 = 32$

$$\begin{array}{r} 14 \\ 19 \overline{) 1F4} \\ \underline{-19} \\ 0B4 \\ \underline{-64} \\ 00 \end{array}$$
$$\begin{array}{r} 1F \\ -19 \\ \hline 0B \end{array}$$
$$\begin{array}{r} 4 \\ 4 \overline{) 36} \\ \underline{-32} \\ 64 \end{array}$$

# Suma octal

SUMA OCTAL.

ADD  $(3527)_8$  AND  $(4167)_8$

$$\begin{array}{r} \phantom{0} \phantom{0} \\ 3527 \\ + 4167 \\ \hline (7716)_8 \end{array}$$
$$\begin{array}{r} \phantom{0} \phantom{0} \\ 3527 \\ + 4167 \\ \hline 7716 \end{array}$$
$$\begin{array}{r} \phantom{0} \phantom{0} \\ 14 \\ - 8 \\ \hline 6 \\ 9 \\ - 3 \\ \hline 6 \end{array}$$

# Resta octal

RESTA OCTAL

SUBTRACT  $(415)_8$  FROM  $(614)_8$

$\begin{array}{r} 614 \\ -415 \\ \hline 177 \end{array}$	$\begin{array}{r} 12 \\ -5 \\ \hline 7 \end{array}$	$\begin{array}{r} +8 \\ -1 \\ +1 \\ -1 \\ \hline 7 \end{array}$	$\begin{array}{r} -1 \\ +6 \\ -4 \\ \hline 1 \end{array}$
--	---	---	---

$\begin{array}{r} +8 \\ -1 \quad -1 +8 \\ 4327 \\ -1541 \\ \hline 2566 \end{array}$	$\begin{array}{r} +8 \\ -1 \\ +3 \\ -5 \\ \hline 5 \end{array}$	$\begin{array}{r} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}$
---	---	--

# Multiplicación octal

**MULTIPLICACION OCTAL**

$$\begin{array}{r} 25 \\ \times 16 \\ \hline 12 \ 30 \\ 5 \end{array}$$

$$\begin{array}{r} 25 \\ \times 16 \\ \hline 12 \ 30 \\ + 2 \ 5 \\ \hline 2^2 \ 17^3 \ 30 \\ - 16 - 24 \\ \hline 4 \ 4 \ 6 \end{array}$$

$8 \times 3 = 24$   
 $8 \times 2 = 16$   
 $17 + 3 = 20$   
 $2 + 2 = 4$

**Ej 22**

$$\begin{array}{r} 2364 \\ \times 24 \\ \hline 8 \ 12 \ 24 \ 16 \\ 4612 \ 8 \\ \hline 4^2 \ 14^3 \ 24^4 \ 32^2 \ 16^5 \\ - 16 - 24 - 32 - 16 \\ \hline 6142016 \end{array}$$

$8 \times 2 = 16$   
 $8 \times 4 = 32$   
 $8 \times 3 = 24$

$$\begin{array}{r} 34 \ 28 \\ - 32 \ -24 \\ \hline 2 \ 4 \\ 17 \\ - 16 \\ \hline 1 \end{array}$$

**Ej 22**

$$\begin{array}{r} 14363 \\ \times 56 \\ \hline 6 \ 24 \ 18 \ 36 \ 18 \\ 520 \ 15 \ 30 \ 15 \\ \hline 1^5 \ 3^2 \ 6^3 \ 39^6 \ 48^6 \ 51^2 \ 18 \\ - 8 - 24 - 40 - 48 - 48 - 16 \\ \hline 1075652 \end{array}$$

$6 \times 8 = 48$   
 $5 \times 8 = 40$   
 $3 \times 8 = 24$   
 $1 \times 8 = 8$

$$\begin{array}{r} 237_8 \\ \times 23_8 \\ \hline \text{sol. } 5715 \end{array}$$

$$\begin{array}{r} 354 \\ \times 23 \\ \hline \text{sol. } 10604 \end{array}$$

# División octal

División Octal

$$\begin{array}{r} 215 \\ 3 \overline{) 651} \\ \underline{-6} \phantom{1} \\ 05 \phantom{1} \\ \underline{-3} \phantom{1} \\ 21 \phantom{1} \\ \underline{-17} \\ 02 \text{ Res.} \end{array}$$
$$\begin{array}{r} 7 \\ \times 3 \\ \hline 21 \\ \underline{-16} \\ 25 \end{array}$$
$$\begin{array}{r} 6 \\ \times 3 \\ \hline 18 \\ \underline{-16} \\ 22 \end{array}$$
$$\begin{array}{r} 5 \\ \times 3 \\ \hline 15 \\ \underline{-8} \\ 17 \end{array}$$
$$\begin{array}{r} -1+8 \\ 21 \\ \underline{-17} \\ 02 \end{array}$$

# Códigos binarios y alfanuméricos

- Gray
- BCD (numérico)
- ASCII
- UNICODE

# Gray (alfanumérico)

- Código alfanuméricos se utilizan para representar números, letras, símbolos e instrucciones.
- Ejemplos son el código Gray y el código ASCII.
- El código Gray es un código sin pesos y no aritmético; es decir, no existen pesos específicos asignados a las posiciones de los bits.
- La característica más importante del código Gray es que sólo varía un bit de un código al siguiente.

Decimal	Binario	Código Gray	Decimal	Binario	Código Gray
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

# BCD (código numérico)

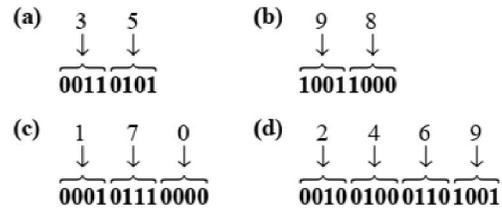
- El código decimal binario (BCD, Binary Coded Decimal) es una forma de expresar cada uno de los dígitos decimales con un código binario.
- Puesto que en el sistema BCD sólo existen diez grupos de código, es muy fácil convertir entre decimal y BCD.
- Como nosotros leemos y escribimos en decimal, el código BCD proporciona una excelente interfaz para los sistemas binarios.
- Ejemplos de estas interfaces son las entradas por teclado y las salidas digitales.

# BCD

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

Convertir a BCD los siguientes números decimales: (a) 35 (b) 98 (c) 170 (d) 2469

**Solución**



**Problema relacionado** Convertir a BCD el número decimal 9673.

Convertir a decimal cada uno los siguientes códigos BCD:

(a) 10000110    (b) 001101010001    (c) 1001010001110000

**Solución**

<p>(a) 1000 0110</p> <p style="text-align: center;">↓    ↓</p> <p style="text-align: center;">8    6</p>	<p>(b) 0011 0101 0001</p> <p style="text-align: center;">↓    ↓    ↓</p> <p style="text-align: center;">3    5    1</p>	<p>(c) 1001 0100 0111 0000</p> <p style="text-align: center;">↓    ↓    ↓    ↓</p> <p style="text-align: center;">9    4    7    0</p>
--	---	--

**Problema relacionado** Convertir a decimal el código BCD 10000010001001110110.

# ASCII (American Standard Code for Information Interchange )

- El código estándar americano para el intercambio de información) es un código alfanumérico universalmente aceptado, que se usa en la mayoría de las computadoras y otros equipos electrónicos.
- La mayor parte de los teclados de computadora se estandarizan de acuerdo con el código ASCII, y cuando se pulsa una letra, un número o un comando de control, es el código ASCII el que se introduce en la computadora.
- El código ASCII dispone de 128 caracteres que se representan mediante un código binario de 7 bits.

# Tabla con el código ASCII

Caracteres de control				Símbolos gráficos											
Nombre	Dec	Binario	Hex	Símbolo	Dec	Binario	Hex	Nombre	Dec	Binario	Hex	Símbolo	Dec	Binario	Hex
NUL	0	000000	00	space	32	010000	20	@	64	100000	40	`	96	110000	60
SOH	1	000001	01	!	33	010001	21	A	65	100001	41	a	97	110001	61
STX	2	000010	02	"	34	010010	22	B	66	100010	42	b	98	110010	62
ETX	3	000011	03	#	35	010011	23	C	67	100011	43	c	99	110011	63
EOT	4	000100	04	\$	36	010100	24	D	68	100100	44	d	100	110100	64
ENQ	5	000101	05	%	37	010101	25	E	69	100101	45	e	101	110101	65
ACK	6	000110	06	&	38	010110	26	F	70	100110	46	f	102	110110	66
BEL	7	000111	07	'	39	010111	27	G	71	100111	47	g	103	110111	67
BS	8	001000	08	(	40	011000	28	H	72	101000	48	h	104	111000	68
HT	9	001001	09	)	41	011001	29	I	73	101001	49	i	105	111001	69
LF	10	001010	0A	*	42	011010	2A	J	74	101010	4A	j	106	111010	6A
VT	11	001011	0B	+	43	011011	2B	K	75	101011	4B	k	107	111011	6B
FF	12	001100	0C	,	44	011100	2C	L	76	101100	4C	l	108	111100	6C
CR	13	001101	0D	-	45	011101	2D	M	77	101101	4D	m	109	111101	6D
SO	14	001110	0E	.	46	011110	2E	N	78	101110	4E	n	110	111110	6E
SI	15	001111	0F	/	47	011111	2F	O	79	101111	4F	o	111	111111	6F
DLE	16	010000	10	0	48	011000	30	P	80	101000	50	p	112	111000	70
DC1	17	010001	11	1	49	011001	31	Q	81	101001	51	q	113	111001	71
DC2	18	010010	12	2	50	011010	32	R	82	101010	52	r	114	111010	72
DC3	19	010011	13	3	51	011011	33	S	83	101011	53	s	115	111011	73
DC4	20	010100	14	4	52	011010	34	T	84	101010	54	t	116	111010	74
NAK	21	010101	15	5	53	011011	35	U	85	101011	55	u	117	111011	75
SYN	22	010110	16	6	54	011011	36	V	86	101011	56	v	118	111011	76
ETB	23	010111	17	7	55	011011	37	W	87	101011	57	w	119	111011	77
CAN	24	011000	18	8	56	011100	38	X	88	101100	58	x	120	111100	78
EM	25	011001	19	9	57	011101	39	Y	89	101101	59	y	121	111101	79
SUB	26	011010	1A	:	58	011101	3A	Z	90	101101	5A	z	122	111101	7A
ESC	27	011011	1B	;	59	011101	3B	[	91	101101	5B	{	123	111101	7B
FS	28	011100	1C	<	60	011100	3C	\	92	101100	5C		124	111100	7C
GS	29	011101	1D	=	61	011101	3D	]	93	101101	5D	}	125	111101	7D
RS	30	011110	1E	>	62	011110	3E	^	94	101110	5E	~	126	111110	7E
US	31	011111	1F	?	63	011111	3F	_	95	101111	5F	Del	127	111111	7F

# Ejemplo

Determinar los códigos binarios ASCII que se han introducido a través del teclado de la computadora cuando se ha escrito la instrucción BASIC siguiente. Expresar también cada código en hexadecimal.

```
20 PRINT "A=";X
```

**Solución**

En la Tabla 2.7 puede encontrar el código ASCII correspondiente a cada carácter.

Símbolo	Binario	Hexadecimal
2	0110010	32 <sub>16</sub>
0	0110000	30 <sub>16</sub>
Space	0100000	20 <sub>16</sub>
P	1010000	50 <sub>16</sub>
R	1010010	52 <sub>16</sub>
I	1001001	49 <sub>16</sub>
N	1001110	4E <sub>16</sub>
T	1010100	54 <sub>16</sub>
Space	0100000	20 <sub>16</sub>
"	0100010	22 <sub>16</sub>
A	1000001	41 <sub>16</sub>
=	0111101	3D <sub>16</sub>
"	0100010	22 <sub>16</sub>
;	0111011	3B <sub>16</sub>
X	1011000	58 <sub>16</sub>

**Problema relacionado**

Determinar la secuencia de los códigos ASCII requerida para la siguiente instrucción de programa y expresarla en hexadecimal:

```
80 INPUT Y
```

# Caracteres del código ASCII extendido

Nombre	Decimal	Hex	Tecla	Descripción
NUL	0	00	CTRL @	Carácter nulo
SOH	1	01	CTRL A	Inicio de cabecera
STX	2	02	CTRL B	Inicio de texto
ETX	3	03	CTRL C	Fin de texto
EOT	4	04	CTRL D	Fin de transmisión
ENQ	5	05	CTRL E	Petición
ACK	6	06	CTRL F	Reconocimiento
BEL	7	07	CTRL G	Timbre
BS	8	08	CTRL H	Barra espaciadora
HT	9	09	CTRL I	Tabulador horizontal
LF	10	0A	CTRL J	Avance de línea
VT	11	0B	CTRL K	Tabulador vertical
FF	12	0C	CTRL L	Salto de página
CR	13	0D	CTRL M	Retorno de carro
SO	14	0E	CTRL N	Desplazamiento de salida
SI	15	0F	CTRL O	Desplazamiento de entrada
DLE	16	10	CTRL P	Escape de enlace de datos
DC1	17	11	CTRL Q	Dispositivo de control 1
DC2	18	12	CTRL R	Dispositivo de control 2
DC3	19	13	CTRL S	Dispositivo de control 3
DC4	20	14	CTRL T	Dispositivo de control 4
NAK	21	15	CTRL U	Confirmación negativa
SYN	22	16	CTRL V	Sincronismo
ETB	23	17	CTRL W	Fin del bloque de transmisión
CAN	24	18	CTRL X	Cancelación
EM	25	19	CTRL Y	Fin del dispositivo
SUB	26	1A	CTRL Z	Sustitución
ESC	27	1B	CTRL [	Escape
FS	28	1C	CTRL /	Separador de archivo
GS	29	1D	CTRL ]	Separador de grupo
RS	30	1E	CTRL ^	Separador de registro
US	31	1F	CTRL _	Separador de unidad

# Caracteres ASCII extendidos

Símbolo	Dec	Hex									
Ç	128	80	á	160	A0	Ł	192	C0	α	224	E0
ü	129	81	í	161	A1	±	193	C1	β	225	E1
é	130	82	ó	162	A2	‡	194	C2	Γ	226	E2
â	131	83	ú	163	A3	‡	195	C3	π	227	E3
ã	132	84	û	164	A4	—	196	C4	Σ	228	E4
ä	133	85	Ñ	165	A5	+	197	C5	σ	229	E5
å	134	86	ä	166	A6	‡	198	C6	μ	230	E6
ç	135	87	Ω	167	A7	‡	199	C7	τ	231	E7
è	136	88	ℓ	168	A8	ℓ	200	C8	Φ	232	E8
ë	137	89	ƒ	169	A9	π	201	C9	Θ	233	E9
è	138	8A	ƒ	170	AA	≡	202	CA	Ω	234	EA
ÿ	139	8B	½	171	AB	≡	203	CB	δ	235	EB
ı	140	8C	¼	172	AC	≡	204	CC	ø	236	EC
ı	141	8D	ı	173	AD	≡	205	CD	φ	237	ED
Ä	142	8E	«	174	AE	≡	206	CE	ε	238	EE
Å	143	8F	»	175	AF	≡	207	CF	∩	239	EF
É	144	90	◻	176	B0	≡	208	D0	≡	240	F0
æ	145	91	◻	177	B1	≡	209	D1	≡	241	F1
Æ	146	92	◻	178	B2	‡	210	D2	≡	242	F2
ö	147	93		179	B3	ƒ	211	D3	≡	243	F3
õ	148	94	‡	180	B4	π	212	D4	ı	244	F4
ò	149	95	≡	181	B5	π	213	D5	ı	245	F5
ú	150	96	≡	182	B6	ƒ	214	D6	÷	246	F6
ù	151	97	‡	183	B7	≡	215	D7	≈	247	F7
ÿ	152	98	≡	184	B8	≡	216	D8	°	248	F8
Ö	153	99	≡	185	B9	Ł	217	D9	•	249	F9
Û	154	9A	≡	186	BA	ƒ	218	DA	·	250	FA
é	155	9B	≡	187	BB	■	219	DB	√	251	FB
£	156	9C	≡	188	BC	■	220	DC	η	252	FC
¥	157	9D	≡	189	BD	■	221	DD	²	253	FD
Pr	158	9E	≡	190	BE	■	222	DE	■	254	FE
f	159	9F	ƒ	191	BF	■	223	DF	□	255	FF

# Unicode

- **Unicode Transform Protocol (UTF).**
- The UTF encodings are defined by the Unicode standard, and are able to encode every single Unicode **code point** we need.
- But there are different types of UTF standards. They differ depending on the amount of bytes used to encode one **code point**. It also depends on whether you're using **UTF-8** (one byte per code point), **UTF-16** (two bytes per code point) or **UTF-32** (four bytes per code point).

Convertir los siguientes números hexadecimales a decimal:

(a)  $E5_{16}$       (b)  $B2F8_{16}$

*Solución*



$$\begin{aligned} \text{(a) } E5_{16} &= (E \times 16) + (5 \times 1) = (14 \times 16) + (5 \times 1) = 224 + 5 \\ &= \mathbf{229}_{10} \end{aligned}$$

$$\begin{aligned} \text{(b) } B2F8_{16} &= (B \times 4096) + (2 \times 256) + (F \times 16) + (8 \times 1) \\ &= (11 \times 4096) + (2 \times 256) + (15 \times 16) + (8 \times 1) \\ &= 45.056 + 512 + 240 + 8 \\ &= \mathbf{45.816}_{10} \end{aligned}$$

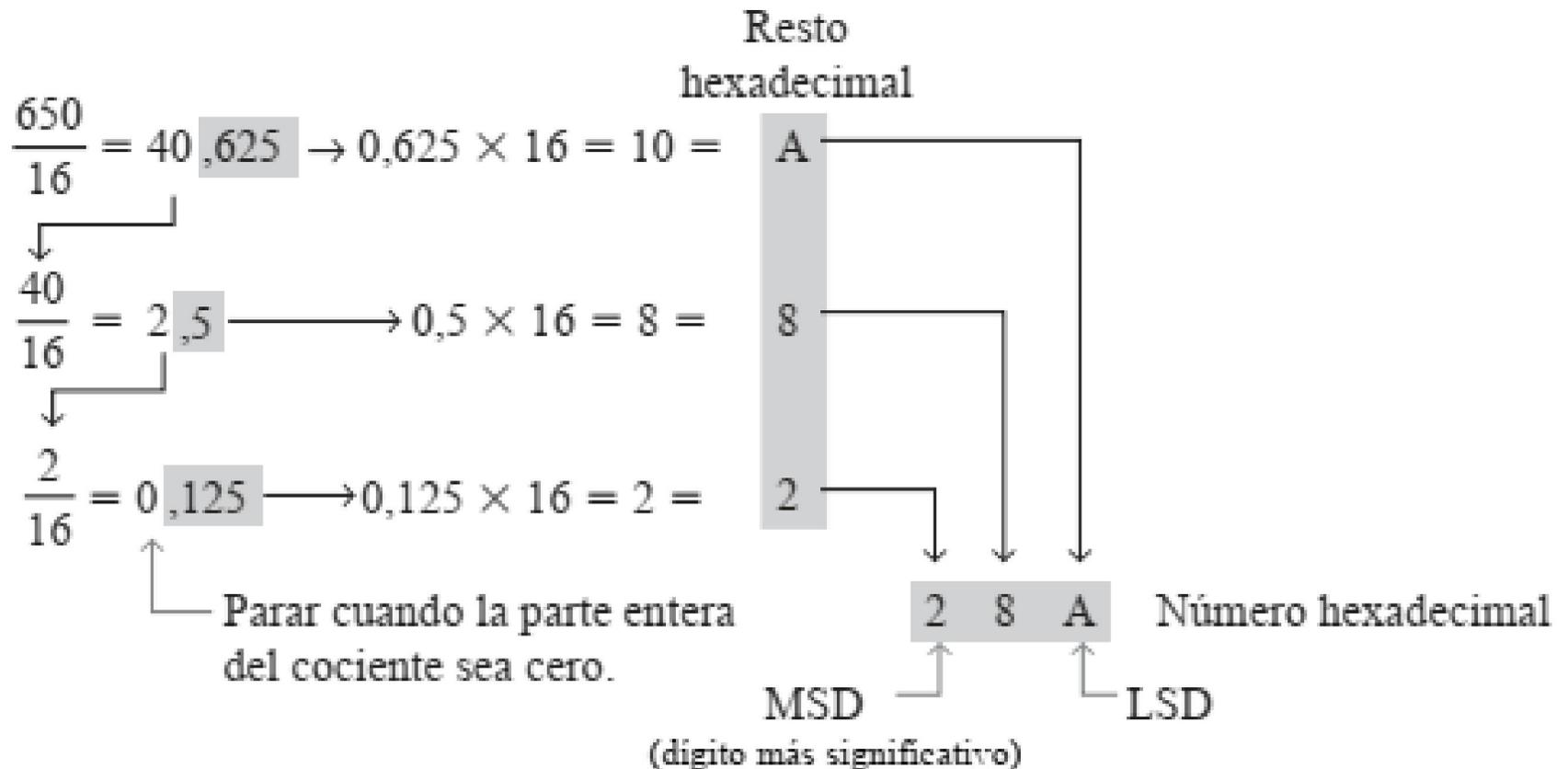
*Problema relacionado* Convertir  $60A_{16}$  a decimal.

# Conversión de un número Decimal a Hexadecimal

Primero, convertir el número hexadecimal a binario, y después, el binario a decimal.

Convertir el número decimal 650 en hexadecimal mediante el método del división sucesiva por 16.

*Solución*



*Problema relacionado* Convertir a hexadecimal el número decimal 2591.

# Ejercicios

## Conversión Decimal a Hexadecimal

20
51
63
64
102
210
1024
41
33
16
15



Thomas L. Floyd

9ª edición

# Fundamentos de sistemas digitales

[www.librosite.net/floyd](http://www.librosite.net/floyd)

PEARSON  
Prentice  
Hall



# **FUNDAMENTOS DE SISTEMAS DIGITALES**



# FUNDAMENTOS DE SISTEMAS DIGITALES

Novena Edición

THOMAS L. FLOYD

**Traducción**

Vuelapluma

**Revisión Técnica**

Eduardo Barrera López de Turiso

*Departamento de Sistemas Electrónicos y de Control*

*Universidad Politécnica de Madrid*



Madrid ● México ● Santa Fe de Bogotá ● Buenos Aires ● Caracas ● Lima  
Montevideo ● San Juan ● San José ● Santiago ● São Paulo ● White Plains ●

**Datos de catalogación bibliográfica**

**FUNDAMENTOS DE SISTEMAS DIGITALES**

**Thomas L. Floyd**

PEARSON EDUCACIÓN S.A., Madrid, 2006

ISBN 10: 84-8322-085-7

ISBN 13: 978-84-832-2720-6

Materia: Informática, 0004.4

Formato: 195 x 250 mm.

Páginas: 1024

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. Código Penal*).

**DERECHOS RESERVADOS**

© 2006 por PEARSON EDUCACIÓN S.A.

Ribera del Loira, 28

28042 Madrid

**FUNDAMENTOS DE SISTEMAS DIGITALES**

Thomas L. Floyd

**ISBN 10: 84-8322-085-7**

**ISBN 13: 978-84-8322-085-6**

Deposito Legal:

PRENTICE HALL es un sello editorial autorizado de PEARSON EDUCACIÓN S.A.

Authorized translation from the English language edition, entitled DIGITAL FUNDAMENTALS, 9<sup>TH</sup> Edition by FLOYD, THOMAS L., published by Pearson Education Inc, publishing as Prentice Hall, Copyright © 2006

**EQUIPO EDITORIAL**

Editor: Miguel Martín-Romo

Técnico editorial: Marta Caicoya

**EQUIPO DE PRODUCCIÓN:**

Director: José A. Clares

Técnico: María Alvear

**Diseño de Cubierta: Equipo de diseño de Pearson Educación S.A.**

**Impreso por:**

IMPRESO EN ESPAÑA - PRINTED IN SPAIN

Este libro ha sido impreso con papel y tintas ecológicos

# CONTENIDO

---

■ *Los temas marcados con este símbolo pueden considerarse opcionales.*

## 1 CONCEPTOS DIGITALES 2

- 1.1 Magnitudes analógicas y digitales 4
- 1.2 Dígitos binarios, niveles lógicos y formas de onda digitales 6
- 1.3 Operaciones lógicas básicas 14
- 1.4 Introducción a las funciones lógicas básicas 16
- 1.5 Circuitos integrados de función fija 22
- 1.6 Introducción a la lógica programable 25
- 1.7 Instrumentos de medida y prueba 31  
Aplicación a los sistemas digitales 43

## 2 SISTEMAS DE NUMERACIÓN, OPERACIONES Y CÓDIGOS 52

- 2.1 Números decimales 54
- 2.2 Números binarios 56
- 2.3 Conversión decimal a binario 60
- 2.4 Aritmética binaria 63
- 2.5 Complemento a 1 y complemento a 2 de los números binarios 67
- 2.6 Números con signo 69
- 2.7 Operaciones aritméticas de números con signo 75
- 2.8 Números hexadecimales 82
- 2.9 Números octales 90
- 2.10 Código decimal binario (BCD) 93
- 2.11 Códigos digitales 96
- 2.12 Detección de errores y códigos de corrección 104

## 3 PUERTAS LÓGICAS 122

- 3.1 El inversor 124

- 3.2 La puerta AND 127
- 3.3 La puerta OR 134
- 3.4 La puerta NAND 139
- 3.5 La puerta NOR 145
- 3.6 Puertas OR-exclusiva y NOR-exclusiva 151
- 3.7 Lógica programable 155
- 3.8 Lógica de función fija 164
- 3.9 Localización de averías 174

## 4 ÁLGEBRA DE BOOLE Y SIMPLIFICACIÓN LÓGICA 198

- 4.1 Operaciones y expresiones booleanas 200
- 4.2 Leyes y reglas del Álgebra de Boole 202
- 4.3 Teoremas de DeMorgan 207
- 4.4 Análisis booleano de los circuitos lógicos 211
- 4.5 Simplificación mediante el Álgebra de Boole 213
- 4.6 Formas estándar de las expresiones booleanas 217
- 4.7 Expresiones booleanas y tablas de verdad 225
- 4.8 Mapas de Karnaugh 228
- 4.9 Minimización de una suma de productos mediante el mapa de Karnaugh 231
- 4.10 Minimización de un producto de sumas mediante el mapa de Karnaugh 242 ■
- 4.11 Mapa de Karnaugh de cinco variables 247 ■
- 4.12 VHDL(opcional) 249  
Aplicación a los sistemas digitales 252

## VI ■ CONTENIDO

### 5 ANÁLISIS DE LA LÓGICA COMBINACIONAL 270

- 5.1 Circuitos lógicos combinacionales básicos 272
- 5.2 Implementación de la lógica combinacional 277
- 5.3 La propiedad universal de las puertas NAND y NOR 284
- 5.4 Lógica combinacional con puertas NAND y NOR 286
- 5.5 Funcionamiento de los circuitos lógicos con trenes de impulsos 292
- 5.6 Lógica combinacional con VHDL 295
- 5.7 Localización de averías 302  
Aplicación a los sistemas digitales 308

### 6 FUNCIONES DE LA LÓGICA COMBINACIONAL 326

- 6.1 Sumadores básicos 328
- 6.2 Sumadores binarios en paralelo 332
- 6.3 Sumadores con acarreo serie y acarreo anticipado 340 ■
- 6.4 Comparadores 344
- 6.5 Decodificadores 348
- 6.6 Codificadores 359
- 6.7 Convertidores de código 364
- 6.8 Multiplexores (selectores de datos) 367
- 6.9 Demultiplexores 377
- 6.10 Generadores / comprobadores de paridad 379
- 6.11 Localización de averías 383  
Aplicación a los sistemas digitales 386

### 7 LATCHES, FLIP-FLOPS Y TEMPORIZADORES 410

- 7.1 Latches 412
- 7.2 Flip-flops disparados por flanco 419
- 7.3 Características de funcionamiento de los flip-flops 433
- 7.4 Aplicaciones de los flip-flops 436
- 7.5 Monoestables 441

- 7.6 El temporizador 555 448
- 7.7 Localización de averías 454  
Aplicación a los sistemas digitales 457

### 8 CONTADORES 474

- 8.1 Funcionamiento del contador asíncrono 476
- 8.2 Funcionamiento del contador síncrono 485
- 8.3 Contador síncrono ascendente/descendente 494
- 8.4 Diseño de contadores síncronos 499 ■
- 8.5 Contadores en cascada 509
- 8.6 Decodificación de contadores 514
- 8.7 Aplicaciones de los contadores 518
- 8.8 Símbolos lógicos con notación de dependencia 523 ■
- 8.9 Localización de averías 525  
Aplicación a los sistemas digitales 530

### 9 REGISTROS DE DESPLAZAMIENTO 550

- 9.1 Funciones básicas de los registros de desplazamiento 552
- 9.2 Registros de desplazamiento con entrada y salida serie 553
- 9.3 Registros de desplazamiento con entrada serie y salida paralelo 558
- 9.4 Registros de desplazamiento con entrada paralelo y salida serie 560
- 9.5 Registros de desplazamiento con entrada y salida paralelo 564
- 9.6 Registros de desplazamiento bidireccionales 566
- 9.7 Contadores basados en registros de desplazamiento 569
- 9.8 Aplicaciones de los registros de desplazamiento 573
- 9.9 Símbolos lógicos con notación de dependencia 581 ■
- 9.10 Localización de averías 583  
Aplicación a los sistemas digitales 586

## 10 MEMORIAS Y ALMACENAMIENTO 600

- 10.1 Principios de las memorias semiconductoras 602
- 10.2 Memorias de acceso aleatorio (RAM) 607
- 10.3 Memorias de sólo lectura (ROM) 622
- 10.4 Memorias ROM programables (PROM y EPROM) 629
- 10.5 Memorias flash 632
- 10.6 Expansión de memorias 637
- 10.7 Tipos especiales de memorias 644
- 10.8 Memorias ópticas y magnéticas 650
- 10.9 Localización de averías 657  
Aplicación a los sistemas digitales 661

## 11 SOFTWARE Y LÓGICA PROGRAMABLE 680

- 11.1 Lógica programable: SPLD y CPLD 682
- 11.2 Dispositivos CPLD de Altera 690
- 11.3 Dispositivos CPLD de Xilinx 697
- 11.4 Macroceldas 701
- 11.5 Lógica programable: dispositivos FPGA 706
- 11.6 Dispositivos FPGA de Altera 712
- 11.7 Dispositivos FPGA de Xilinx 716
- 11.8 Software de lógica programable 723
- 11.9 Lógica de exploración de contorno 736
- 11.10 Localización de averías 744  
Aplicación a los sistemas digitales 751

## 12 INTRODUCCIÓN A LAS COMPUTADORAS 778

- 12.1 Una computadora básica 780
- 12.2 Microprocesadores 784
- 12.3 Una familia específica de microprocesadores 787
- 12.4 Programación de computadoras 795
- 12.5 Interrupciones 806
- 12.6 Acceso directo a memoria (DMA) 809
- 12.7 Interfaces internas 810

- 12.8 Buses estándar 815

## 13 INTRODUCCIÓN AL PROCESAMIENTO DIGITAL DE LA SEÑAL 834

- 13.1 Fundamentos del procesamiento digital de la señal 836
- 13.2 Conversión de señales analógicas a formato digital 837
- 13.3 Métodos de conversión analógica-digital 844
- 13.4 Procesador digital de la señal (DSP) 856
- 13.5 Métodos de conversión digital-analógica 864

## 14 TECNOLOGÍAS DE CIRCUITOS INTEGRADOS 882

- 14.1 Parámetros y características de operación básicas 884
- 14.2 Circuitos CMOS 893
- 14.3 Circuitos TTL 899
- 14.4 Consideraciones prácticas sobre el uso de TTL 905
- 14.5 Comparación de las prestaciones de CMOS y TTL 914
- 14.6 Circuitos ECL 915
- 14.7 PMOS, NMOS y E2CMOS 917

## APÉNDICES

- A Conversiones 931
- B Interfaz de las luces de los semáforos 933  
Respuestas a los problemas impares 935  
Glosario 977  
Índice 995



# PREFACIO

---

Bienvenido a *Fundamentos de sistemas digitales. Novena edición*. Unos conocimientos sólidos sobre los fundamentos básicos de la tecnología digital son imprescindibles para cualquiera que desee desarrollar una carrera en esta excitante industria. Este texto se ha organizado cuidadosamente para incluir información actualizada de temas que pueden cubrirse por completo, utilizarse en formato condensado, u omitirse dependiendo del enfoque del curso.

Los temas tratados en el texto se cubren con el mismo formato claro, directo y cuidadosamente ilustrado que se ha empleado en las ediciones anteriores. Muchos temas se han reforzado o mejorado y pueden encontrarse numerosas mejoras a lo largo del libro.

Probablemente, encontrará más temas de los que se pueden cubrir en un curso cuatrimestral. Este amplio rango de temas proporciona la flexibilidad para diseñar una amplia variedad de cursos. Por ejemplo, algunos de los temas orientados al diseño o a las aplicaciones de los sistemas pueden no ser apropiados en algunos cursos. Otros cursos pueden no cubrir la lógica programable, mientras que otros pueden no disponer del tiempo necesario para tratar temas como las computadoras, microprocesadores o el procesamiento digital de la señal. También, en algunos cursos puede no ser necesario entrar en los detalles de la circuitería interna de los chips. Estos y otros temas se pueden omitir o verse por encima sin que los temas fundamentales se vean afectados. Disponer de conocimientos sobre los circuitos de transistores no es un prerrequisito para este libro de texto, aunque la tecnología de circuitos integrados se cubre en un "capítulo flotante", que es opcional.

El texto tiene una organización modular que permite incluir o excluir varios temas sin que tenga repercusión sobre el resto de los temas incluidos en un determinado curso. Dado que la lógica programable está adquiriendo cada vez más importancia, se ha dedicado un capítulo completo al tema (Capítulo 11), incluyendo el estudio de los dispositivos PAL, GAL, CPLD y FPGA, así como de dispositivos específicos de Altera y Xilinx. También se ha incluido una introducción de carácter general al software de los dispositivos lógicos programables.

## Nuevo en esta edición

- Código de detección y corrección de errores Hamming
- Sumadores con acarreo anticipado
- Una breve introducción a VHDL
- Información ampliada y mejorada sobre instrumentos de prueba
- Información ampliada y reorganizada sobre los dispositivos lógicos programables y su software.
- Información mejorada sobre la localización de averías
- Nuevo enfoque en las secciones Aplicación a los sistemas digitales

## Características

- Notas intercaladas en el texto que proporcionan información en un formato resumido.
- Las palabras clave se enumeran al principio de cada capítulo. Dentro del capítulo, estas palabras clave se resaltan en negrita y cursiva. Cada palabra clave se define al final del capítulo, así como al final del libro en un extenso glosario.

## X ■ PREFACIO

- El Capítulo 14 se ha diseñado como un capítulo “flotante” para proporcionar información opcional sobre la tecnología de circuitos integrados (circuitería interna del chip), que se puede estudiar en cualquier momento a lo largo del curso.
- Al principio de cada capítulo se enumeran los objetivos y se hace una breve introducción.
- Se incluye al principio de cada sección una introducción y los objetivos de la misma.
- Al final de cada sección se plantean ejercicios y cuestiones de repaso.
- Se incluye un problema relacionado en cada ejemplo resuelto.
- Se han intercalado Notas Informáticas a lo largo del texto que proporcionan información interesante sobre la tecnología informática relacionada con la cuestión que se está estudiando.
- Consejos prácticos intercalados proporcionan información útil y práctica.
- Las secciones Aplicación a los sistemas digitales se incluyen al final de muchos de los capítulos y exponen aplicaciones interesantes y prácticas de los fundamentos de los sistemas lógicos.
- Resúmenes al final de cada capítulo.
- Autotest con múltiples respuestas al final de cada capítulo.
- Conjuntos de problemas organizados por secciones al final de cada capítulo, incluyendo problemas básicos, de localización de averías, de aplicaciones de sistemas y de diseños especiales.
- Se cubre el uso y aplicación de instrumentos de prueba, como el osciloscopio, el analizador lógico, el generador de funciones y los multímetros digitales (DMM).
- El Capítulo 12 proporciona una introducción a las computadoras.
- El Capítulo 13 presenta el procesamiento digital de la señal, incluyendo la conversión analógica-digital y la conversión digital-analógica.
- Al principio del Capítulo 1 se presentan conceptos sobre la lógica programable.
- Se presentan a lo largo del texto circuitos integrados específicos que implementan una función determinada.
- El Capítulo 11 aborda los dispositivos PAL, GAL y FPGA, así como una exposición de carácter general sobre la programación de dispositivos PLD.
- En el Capítulo 11 se introduce la lógica de exploración de contorno asociada con los dispositivos programables.
- Además de la técnica de exploración de contorno, el tema de la localización de averías incluye otros métodos para probar los dispositivos programables, como las pruebas tradicionales y las camas de pinchos.
- Para aquellos que deseen incluir una introducción a la programación con ABEL, pueden encontrar información en el sitio web [www.librosite.net/floyd](http://www.librosite.net/floyd).

### Otros recursos para el estudiante

- *Experiments in Digital Fundamentals* de David M. Buchla es un manual de laboratorio. Las soluciones de este manual están disponibles en el manual del profesor *Instructor's Resource Manual*.

### Recursos para el profesor

- *Sitio web* [www.librosite.net/floyd](http://www.librosite.net/floyd). Este sitio web ofrece al profesor la posibilidad de publicar su plan de estudios en línea con nuestro programa Syllabus Manager™. Se trata de una excelente solución para la enseñanza a distancia, autodidacta o asistida por computadora.

- *Instructor's Resource Manual*. Este manual incluye las soluciones a los problemas planteados en los capítulos, las soluciones a las secciones de Aplicación a los sistemas digitales y los resultados de laboratorio para el manual de David M. Buchla (impreso y en línea)
- *Test Item File*. Esta edición incorpora más de 900 cuestiones.
- *TestGen*.<sup>®</sup> Es una versión electrónica de *Tests Item File*, que permite a los profesores personalizar los exámenes para cada curso.

## Características de los capítulos

**Introducción del capítulo** Las dos primeras páginas de cada capítulo tienen el formato que se indica en la Figura P.1. La página de la izquierda contiene la lista de las secciones y la lista de los objetivos del capítulo. En la página de la derecha se presenta la introducción, una lista de los dispositivos específicos que se verán en el capítulo (cada nuevo dispositivo se indica mediante el logotipo de un circuito integrado en el lugar donde se introduce), una breve descripción de la aplicación a los sistemas digitales que se verá en el capítulo y una lista de palabras clave.

**Introducción de la sección** Cada sección del capítulo comienza con una breve introducción, que proporciona una visión general y una lista de los objetivos de la misma. En la Figura P.2 se muestra un ejemplo.

**Revisión de la sección** Cada sección termina con una revisión, en la que se incluyen preguntas o ejercicios sobre los principales conceptos presentados, como se muestra en la Figura P.2. Las respuestas a estos ejercicios se encuentran al final de cada capítulo.

# 3

## PUERTAS LÓGICAS

- Elaborar los diagramas de tiempos que muestran las relaciones de tiempo de las entradas y las salidas de las diferentes puertas lógicas.
- Establecer las comparaciones básicas entre las principales tecnologías de circuitos integrados: TTL y CMOS.
- Explicar las diferencias entre las series de las familias TTL y CMOS.
- Definir, para las puertas lógicas, los siguientes parámetros: *tiempo de retardo de propagación*, *disipación de potencia*, *producto velocidad-potencia* y *fan-out*.
- Enumerar circuitos integrados de función fija que contengan varias puertas lógicas.
- Utilizar cada puerta lógica en aplicaciones sencillas.
- Localización de averías en las puertas lógicas debidas a circuitos abiertos o cortocircuitos, utilizando el pulsador y la sonda lógica o el osciloscopio.

**CONTENIDO DEL CAPÍTULO**

- 3.1 El inversor
- 3.2 La puerta AND
- 3.3 La puerta OR
- 3.4 La puerta NAND
- 3.5 La puerta NOR
- 3.6 Puertas OR-exclusiva y NOR-exclusiva
- 3.7 Lógica programable
- 3.8 Lógica de función fija
- 3.9 Localización de averías

**OBJETIVOS DEL CAPÍTULO**

- Describir el funcionamiento del inversor y de las puertas AND y OR.
- Describir el funcionamiento de las puertas NAND y NOR.
- Expresar las operaciones de las puertas NOT, AND, OR, NAND y NOR mediante el Álgebra de Boole.
- Describir el funcionamiento de las puertas OR-exclusiva y NOR-exclusiva.
- Reconocer y utilizar los símbolos distintivos y los símbolos rectangulares de las puertas lógicas según el estándar ANSI/IEEE 91-1984.

**DISPOSITIVO OBJETIVO**

- JTAG
- CMOS
- TTL
- Tiempo de retardo de propagación
- Fan-out
- Carga unidad

**INTRODUCCIÓN**

Este capítulo hace énfasis en el funcionamiento lógico, las aplicaciones y la localización de averías de las puertas lógicas. Se cubre la relación entre las formas de onda de entrada y de salida de una puerta utilizando los diagramas de tiempos.

Los símbolos lógicos que se usan para representar las puertas lógicas están de acuerdo con el estándar ANSI/IEEE 91-1984. Este estándar ha sido adoptado por la industria privada, y la industria militar lo utiliza para su documentación interna así como para sus publicaciones.

En este capítulo se aborda tanto la lógica programable como la lógica de función fija. Puesto que en todas las aplicaciones se usan los circuitos integrados (CI), generalmente, la función lógica de un dispositivo es más importante para el técnico que los detalles de operación del circuito en el nivel de componentes en el interior del CI. Por tanto, la cobertura detallada de los dispositivos en el nivel de componente puede tratarse como un tema opcional. Para aquellos que lo necesitan y tengan tiempo, en el Capítulo 14 se cubren las tecnologías de los circuitos integrados digitales, haciéndose referencia a partes del mismo a lo largo del texto. *Sugerencia:* repase la Sección 1.3 antes de comenzar con este capítulo.

**DISPOSITIVOS LÓGICOS DE FUNCIÓN FIJA**

(SERIES CMOS y TTL)

74XX00	74XX02	74XX04
74XX08	74XX10	74XX11
74XX20	74XX21	74XX27
74XX30	74XX32	74XX86
74XX266		

**PALABRAS CLAVE**

- Inversor
- Tabla de verdad
- Diagrama de tiempos
- Álgebra booleana
- Complemento
- Puerta AND
- Habilitar
- Puerta OR
- Puerta NAND
- Puerta NOR
- Puerta OR-exclusiva
- Puerta NOR-exclusiva
- Matriz AND
- Fusible
- Antifusible
- EPROM
- EEPROM
- SRAM

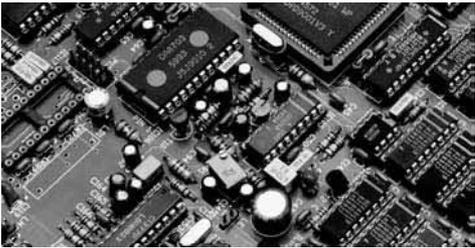


FIGURA P.1 Introducción del capítulo.

## XII ■ PREFACIO

Ejercicios de revisión al final de cada sección.

Párrafo de introducción y lista de los objetivos de la sección al comienzo de la misma.

Las Notas Informáticas se encuentran a lo largo del texto

**332 ■ FUNCIONES DE LA LÓGICA COMBINACIONAL**

**REVISIÓN DE LA SECCIÓN 6.1**

- Determinar la suma (?) y el acarreo de salida ( $C_{out}$ ) de un semi-sumador para cada uno de los siguientes grupos de bits de entrada:  
(a) 01 (b) 00 (c) 10 (d) 11
- Un sumador completo tiene  $C_{in} = 1$ . ¿Cuánto vale la suma ( $\Sigma$ ) y el acarreo de salida ( $C_{out}$ ) cuando  $A = 1$  y  $B = 1$ ?

**6.2 SUMADORES BINARIOS EN PARALELO**

Para formar un sumador binario en paralelo se conectan dos o más sumadores completos. En esta sección aprenderemos los principios básicos de este tipo de sumador, de manera que podamos entender todas las funciones necesarias de entrada y salida cuando se trabaja con este tipo de dispositivos.

Al finalizar esta sección, el lector deberá ser capaz de:

- Utilizar sumadores completos para implementar un sumador binario en paralelo.
- Explicar el proceso de adición en un sumador binario en paralelo.
- Emplear la tabla de verdad para un sumador en paralelo de 4 bits.
- Utilizar dos dispositivos 74LS283 para sumar dos números binarios de 4 bits.
- Ampliar el sumador de 4 bits para poder realizar adiciones de 8 bits o 16 bits.

Como se ha visto en la Sección 6.1, un único sumador completo es capaz de sumar dos números binarios de 1 bit y un acarreo de entrada. Para sumar números binarios de más de un bit, se tienen que utilizar sumadores completos adicionales. Cuando se suman dos números binarios, cada columna genera un bit de suma y un 1 ó 0, correspondiente al bit de acarreo, que se añade a la columna inmediata de la izquierda, como se muestra a continuación con dos números de 2 bits.

Bit de acarreo de la columna de la derecha

$$\begin{array}{r} 1 \\ 11 \\ +01 \\ \hline 100 \end{array}$$

En este caso, el bit de acarreo de la segunda columna se convierte en un bit de suma.

Para sumar dos números binarios, se necesita un sumador completo por cada bit que tengan los números que se quieren sumar. Así, para números de dos bits se necesitan dos sumadores, para números de cuatro bits hacen falta cuatro sumadores, y así sucesivamente. La salida de acarreo de cada sumador se conecta a la entrada de acarreo del sumador de orden inmediatamente superior, como se muestra en la Figura 6.7 para un sumador de 2 bits. Téngase en cuenta que se puede usar un semi-sumador para la posición menos significativa, o bien se puede poner a 0 (masa) la entrada de acarreo de un sumador completo, ya que no existe entrada de acarreo en la posición del bit menos significativo.

En la Figura 6.7 los bits menos significativo (LSB) de los dos números se representan como  $A_1$  y  $B_1$ . Los siguientes bits de orden superior se representan como  $A_2$  y  $B_2$ . Los tres bits de suma son  $\Sigma_1$ ,  $\Sigma_2$  y  $\Sigma_3$ . Observe

**NOTAS INFORMÁTICAS**

 Las computadoras realizan la operación de suma con dos números a un tiempo, denominados *operandos*. El *operando fuente* es un número que se añade a un número existente denominado *operando de destino*, que es el que se almacena en un registro de la UAL, tal como el acumulador. A continuación, la suma de los dos números se almacena de nuevo en el acumulador. La adición se realiza con números enteros o números en coma flotante utilizando, respectivamente, las instrucciones ADD o FADD.

FIGURA P.2 Introducción y revisión de una sección.

**Ejemplos resueltos y problemas relacionados** Numerosos ejemplos resueltos ayudan a ilustrar y clarificar los conceptos básicos o procedimientos específicos. Cada ejemplo concluye con un problema relacionado que le refuerza o amplía, que requieren que el estudiante resuelva siguiendo pasos similares a los seguidos en el ejemplo. En la Figura P.3 se muestra una página con un ejemplo resuelto típico y un problema relacionado.

**Sección de localización de averías** Muchos capítulos incluyen una sección dedicada a la localización de averías, que hace referencia a los temas cubiertos en el capítulo y que se centra en las técnicas de localización de averías y el uso de instrumentos de prueba. En la Figura P.4 se muestra una parte de una sección típica sobre la localización de averías.

**Aplicación a los sistemas digitales** La última sección de la mayor parte de los capítulos presenta una aplicación práctica sobre los conceptos y dispositivos cubiertos en el capítulo. Cada una de estas secciones presenta un sistema del mundo real, en el que se implementan las etapas de análisis, diseño y localización de averías utilizando los procedimientos vistos en el capítulo. Algunas de las aplicaciones a sistemas están limitadas a un único capítulo, y otras se extienden a lo largo de dos o más. Las aplicaciones a los sistemas digitales y sus capítulos asociados son las siguientes:

126 ■ PUERTAS LÓGICAS

**EJEMPLO 3.1**

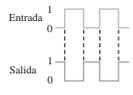
Al inversor de la Figura 3.4 se le aplica una señal. Determinar la forma de onda de salida correspondiente a la entrada y dibujar el diagrama de tiempos. De acuerdo con el emplazamiento del círculo ¿cuál es el estado activo de salida?



**FIGURA 3.4**

**Solución**

La forma de onda de salida es exactamente la opuesta a la de entrada (es la entrada invertida), como se muestra en la Figura 3.5, que es el cronograma básico. El estado activo o verdadero de salida es 0.



**FIGURA 3.5**

**Problema relacionado\*** Si el inversor tiene el indicador negativo (círculo) en la entrada en lugar de en la salida, ¿cómo afecta esto al diagrama de tiempos?

\* Las respuestas se encuentran al final del capítulo.

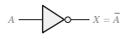
▲ El álgebra booleana utiliza variables y operadores para describir un circuito lógico.

encima de la letra. Una variable puede tomar uno de dos valores, 1 ó 0. Si una variable dada es 1, su complemento es 0, y viceversa.

El modo de operación de un inversor (circuito NOT) puede expresarse del siguiente modo: si la variable de entrada se designa por  $A$  y la variable de salida por  $X$ , entonces

$$X = \bar{A}$$

Esta expresión establece que la salida es el complemento de la entrada, de modo que si  $A = 0$ , entonces  $X = 1$ , y si  $A = 1$ , entonces  $X = 0$ . La Figura 3.6 ilustra esto. La variable complementada  $\bar{A}$  se lee "A barra" o "no A".



**FIGURA 3.6** El inversor complementa una variable de entrada.

**Aplicación**

La Figura 3.7 muestra un circuito que genera el complemento a 1 de un número binario de 8 bits. Los bits del número binario se aplican a las entradas del inversor y el complemento a 1 se obtiene en las salidas.

Cada ejemplo queda delimitado mediante un recuadro

Cada ejemplo contiene un problema relacionado con el mismo.

**FIGURA P.3** Un ejemplo y un problema relacionado.

- Sistema de control y recuento de pastillas: Capítulo 1.
- Display digital: Capítulos 4 y 11.
- Sistema de control de un tanque de almacenamiento: Capítulo 5.
- Sistema de control de semáforos: Capítulos 6, 7 y 8.
- Sistema de seguridad: Capítulos 9 y 10.

Las aplicaciones a los sistemas digitales pueden tratarse como secciones opcionales, ya que su omisión no afecta al resto del material incluido en el texto. La Figura P.5 muestra una parte de una sección de "Aplicación a los sistemas digitales".

**Fin del capítulo** Al final de cada capítulo se incluye la siguiente información:

- Resumen
- Glosario de las palabras clave
- Autotest

3.9 LOCALIZACIÓN DE AVERÍAS

La localización de averías es el proceso de reconocer, aislar y corregir un fallo en un sistema o circuito. Para poder localizar las averías de forma efectiva, debe entender cómo se supone que trabaja el circuito o sistema y debe estar en disposición de reconocer un funcionamiento incorrecto. Por ejemplo, para determinar si una puerta lógica tiene un fallo, debe saber cuál debe ser la salida para unas entradas dadas.

Al finalizar esta sección, el lector deberá ser capaz de:

- Comprobar la existencia de entradas y salidas abiertas internamente en las puertas lógicas de los CI.
- Reconocer los efectos de una entrada o una salida del CI cortocircuitada.
- Detectar en una tarjeta de circuito impreso la existencia de fallos externos.
- Localizar las averías en un sencillo contador de frecuencia utilizando un osciloscopio.

Fallos internos en las puertas lógicas de los CI

Los circuitos abiertos y los cortocircuitos son los fallos más comunes en las puertas internas del CI. Se pueden producir tanto en las entradas como en la salida de una puerta contenida en el encapsulado del CI. Antes de intentar solucionar cualquier avería, compruebe que la alimentación continua y la masa son correctas.

**Efectos de una entrada que se encuentra en circuito abierto internamente.** Un circuito abierto interno es el resultado de un componente en circuito abierto o de una ruptura en la conexión entre el chip y el pin del encapsulado. Una entrada en circuito abierto impide que una señal de impulsos en esta entrada dé lugar a una salida, como se muestra en la Figura 3.67(a) para la puerta NAND de 2 entradas. Una entrada TTL, en abierto actúa como un nivel ALTO, por lo que los impulsos aplicados a la entrada que está en buen estado pasan a través de la puerta NAND hasta la salida, como se muestra en la Figura 3.67(b).

**Condiciones para probar las puertas.** Al probar una puerta NAND o una puerta AND, debe asegurarse siempre de que las entradas a las que no se aplican impulsos se encuentren a nivel ALTO, para activar la puerta. Cuando pruebe una puerta NOR o una puerta OR, debe asegurarse siempre de que las entradas a las que no se aplican impulsos se encuentran a nivel BAJO. Cuando se pruebe una puerta XOR o XNOR, el nivel de la entrada a la que no se aplican impulsos no importa, ya que los impulsos aplicados en la otra entrada forzarán a que las entradas se encuentren, alternativamente, en el mismo nivel o en niveles opuestos.

**Localización de fallos: entrada en circuito abierto.** La localización de este tipo de fallo es, en la mayoría de los casos, muy fácil utilizando un osciloscopio y un generador de funciones, como se muestra en la Figura 3.68, para el caso de una puerta NAND de 2 entradas. Al medir las señales digitales con un osciloscopio, emplee siempre el acoplamiento en continua.

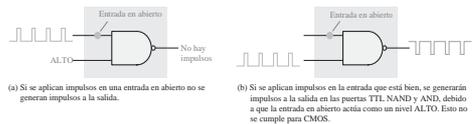


FIGURA 3.67 Efecto de una entrada en circuito abierto en una puerta NAND.

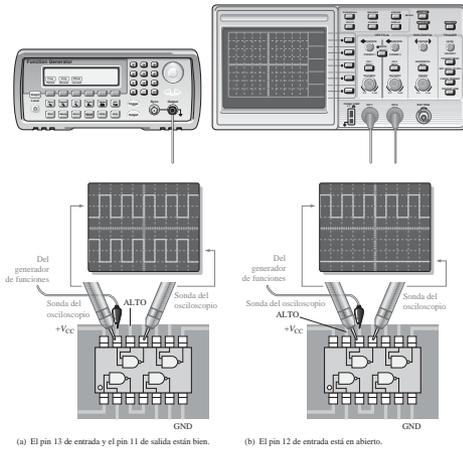


FIGURA 3.68 Localización de averías en una puerta NAND con una entrada en circuito abierto.

El primer paso en la localización de averías de un CI, cuando se sospecha que está fallando, es asegurarse de que la tensión de alimentación continua ( $V_{CC}$ ) y la masa están conectadas a los pines apropiados del CI. Después, se aplican impulsos continuos a una de las entradas de la puerta, asegurándose de que las otras entradas están a nivel ALTO (en el caso de una puerta NAND). En la Figura 3.68(a), se comienza a aplicar los impulsos en el pin 13, ya que se ha determinado que es una de las entradas de la puerta de la que se sospecha el fallo. Si en la salida correspondiente (en este caso el pin 11) se detecta un tren de impulsos, entonces el pin 13 de entrada no está en abierto. Consecuentemente, esto prueba también que la salida no está en abierto. A continuación, se aplica otro tren de impulsos a otra entrada de la puerta (pin 12), asegurándose de que la otra entrada está a nivel ALTO. En la salida (en el pin 11) no se detecta un tren de impulsos y la salida está a nivel BAJO, lo que indica que la entrada del pin 12 está en abierto, como se muestra en la Figura 3.68(b). Observe que la entrada en la que no se aplican impulsos debe estar a nivel ALTO en el caso de una puerta NAND o

FIGURA P.4 Páginas representativas de una sección típica dedicada a la localización de averías.

- Un conjunto de problemas, que incluye algunas o todas las categorías siguientes: problemas básicos, problemas sobre localización de averías, problemas sobre aplicaciones a sistemas y problemas de diseño.
- Respuestas a las revisiones de las secciones.
- Respuestas a los problemas relacionados de los ejemplos.
- Respuestas al autotest.

Al estudiante

¡La tecnología está de moda! Casi todo se está digitalizando o se digitalizará en un futuro próximo. Por ejemplo, los teléfonos móviles y otros medios de comunicación inalámbricos, la televisión, la radio, el control de procesos, la electrónica de automoción, la electrónica de consumo, las técnicas de posicionamiento global, los sistemas militares, por nombre sólo unas pocas aplicaciones, dependen enormemente de la electrónica digital.

Conocer en profundidad los fundamentos de la tecnología digital le preparará para poder acceder en un futuro a trabajos bien remunerados y de alta capacitación. Lo más importante que puede tratar de hacer el lector es comprender los fundamentos básicos. Habiéndolos dominado, tendrá en sus manos hacer lo que desee.

Además, la lógica programable está adquiriendo una importancia extraordinaria en el panorama tecnológico actual y ese es uno de los temas fundamentales cubiertos en el libro. Por supuesto, las habilidades necesarias para un diagnóstico eficiente también son altamente demandadas. El libro incluye, por ello, métodos de



**APLICACIÓN A LOS SISTEMAS DIGITALES**

En esta aplicación, vamos a comenzar a trabajar con el sistema de control de semáforos. En esta sección se establecen los requisitos del sistema, se desarrolla un diagrama de bloques, así como un diagrama de estados para ayudar a establecer la secuencia de funcionamiento. Diseñaremos la parte del sistema que involucra lógica combinatorial y se prepararán los métodos de prueba. En los Capítulos 7 y 8 se tratarán los circuitos de lógica secuencial y de temporización del sistema.

**Requerimientos generales del sistema**

Se requiere un controlador digital para controlar un semáforo en la intersección de una calle de tráfico muy denso con una calle de tráfico moderado. La calle principal va a tener una luz verde durante un mínimo de 25 seg. o mientras no haya ningún vehículo en la calle perpendicular.

Esta calle lateral tiene que tener la luz verde hasta que no circule ningún coche por ella, o durante un máximo de 25 seg. La luz ámbar de precaución tiene que durar 4 seg. en los cambios de luz verde a roja en ambos calles, principal y lateral. Estos requisitos se muestran en el diagrama de la Figura 6.65.

**Desarrollo de un diagrama de bloques del sistema**

A partir de los requisitos, se puede desarrollar un diagrama de bloques del sistema. En primer lugar, sabemos que el sistema tiene que controlar seis pares de luces diferentes. Estas son las luces roja, ámbar y verde para ambos sentidos, tanto en la calle principal como en la lateral. También sabemos que existe una entrada externa (además de la alimentación) que proviene de un sensor de vehículos situado en la calle lateral. En la Figura 6.66, puede ver un diagrama de bloques mínimo que ilustra estos requisitos.

A partir del diagrama de bloques mínimo vamos a ir entrando en los detalles. El sistema tiene cuatro estados, como se indica en la Figura 6.65, por lo que se necesita un circuito lógico para controlar la secuencia de estados (lógica secuencial). Además, se necesitan circuitos para generar los intervalos de tiempo adecuados de 25 seg. y 4 seg., que se requieren en el sistema y para generar una señal de reloj cíclica en el sistema (circuitos de temporización). Los intervalos de tiempo (largo y corto) y el sensor de vehículos son entradas de la lógica secuencial, dado que la secuenciación de estados es una función de estas variables. Se necesitan también circuitos lógicos para determinar cuál de los cuatro estados del sistema está activo en un determinado instante de tiempo, para así generar las salidas adecuadas en las luces (decodificación de estados y lógica de salida), y para iniciar los intervalos de tiempo largo y corto. Finalmente, se necesita un circuito de interfaz para convertir los niveles lógicos de la decodificación y del circuito de salida en las tensiones y corrientes requeridas para encender cada una de las luces. La Figura 6.67 representa un diagrama de bloques más detallado que muestra estos elementos esenciales.

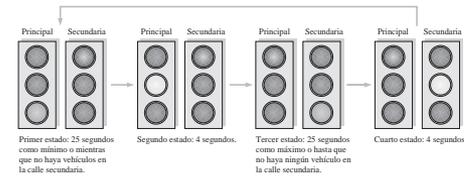


FIGURA 6.65 Requisitos para la secuencia de luces de los semáforos.

das adecuadas en las luces (decodificación de estados y lógica de salida), y para iniciar los intervalos de tiempo largo y corto. Finalmente, se necesita un circuito de interfaz para convertir los niveles lógicos de la decodificación y del circuito de salida en las tensiones y corrientes requeridas para encender cada una de las luces. La Figura 6.67 representa un diagrama de bloques más detallado que muestra estos elementos esenciales.

**El diagrama de estados**

Un diagrama de estados nos muestra gráficamente la secuencia de estados en un sistema y las condiciones de cada estado y de las transiciones entre cada uno de ellos. En realidad, la Figura 6.65 es, en cierta medida, un diagrama de estados, ya que muestra la secuencia de estados y las distintas condiciones.

**Definición de las variables.** Antes de poder desarrollar un diagrama de estados tradicional, es necesario definir las variables que determinan cómo pasa el sistema a través de los diferentes estados. A continuación se enumeran estas variables y sus símbolos:

- Presencia de vehículos en la calle lateral =  $V_L$
- El temporizador de 25 s. (largo) está activado =  $T_L$
- El temporizador de 4 s. (corto) está activado =  $T_C$

El uso de variables complementadas indica la condición contraria. Por ejemplo,  $\bar{V}_L$  indica que no hay ningún

vehículo en la calle lateral;  $\bar{T}_L$  indica que el temporizador de larga duración está desactivado y  $\bar{T}_C$  indica que el temporizador de corta duración está desactivado.

**Descripción del diagrama de estados.** En la Figura 6.68 se muestra un diagrama de estados. Cada uno de los cuatro estados se etiqueta de acuerdo a la secuencia de 2 bits en código Gray, como se indica mediante los círculos. La flecha circular en cada estado indica que el sistema permanece en dicho estado bajo la condición definida por la variable o expresión asociada. Cada una de las flechas que van de un estado al siguiente indican un cambio de estado cuando se produce la condición definida por la variable o expresión asociada.

**Primer estado** El código Gray para este estado es 00. El semáforo de la calle principal está en verde y el de la calle lateral está en rojo. El sistema permanece en este estado al menos 25 segundos cuando el temporizador largo se encuentra activado o mientras que no haya ningún vehículo en la calle lateral ( $T_L + \bar{V}_L$ ). El sistema pasa al siguiente estado cuando el temporizador de 25 segundos está desactivado o cuando aparece algún vehículo en la calle secundaria ( $\bar{T}_L V_C$ ).

**Segundo estado** El código Gray para este estado es 01. El semáforo de la calle principal está en ámbar (precaución) y el de la calle lateral está en rojo. El sistema permanece en este estado durante 4 segundos mientras el temporizador corto está activado ( $T_C$ ) y pasa al siguiente estado cuando este temporizador se desactiva ( $\bar{T}_C$ ).

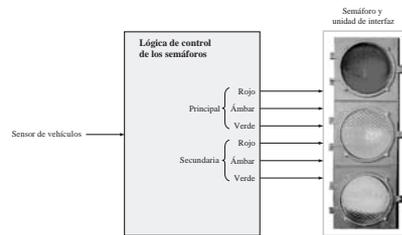


FIGURA 6.66 Diagrama de bloques mínimo del sistema.

FIGURA P.5 Páginas representativas de una sección típica dedicada a las aplicaciones de sistemas digitales.

prueba y localización de averías que van desde las pruebas tradicionales a las técnicas utilizadas en los procesos de fabricación, como la de cama de pinchos, la de sonda volante y la de exploración de contorno. Estos son algunos ejemplos de los conocimientos que podrá adquirir si se aplica con decisión al aprendizaje de los conceptos presentados.

**Agradecimientos**

Este innovador texto es el resultado de los esfuerzos y las habilidades de muchas personas. Creo que hemos conseguido lo que pretendíamos hacer, que era escribir un libro de texto de excelente calidad. En Prentice Hall, Kate Linsner y Rex Davidson han aportado su tiempo, talento y esfuerzo a lo largo de las muchas fases de este proyecto con la finalidad de obtener un libro como el que el lector tiene en sus manos. Lois Porter ha hecho un fantástico trabajo de edición del manuscrito. Ha sido capaz de desvelar los misterios de las marcas de este autor y con frecuencia sus prácticamente ilegibles anotaciones y, a partir de un desorden completo, extraer un manuscrito increíblemente organizado y estupendamente editado. También Jane Lopez ha hecho un estupendo trabajo con las imágenes. Otra persona que ha contribuido significativamente a este libro es Gary Snyder, quien ha proporcionado todos los archivos de los circuitos Multisim (en Multisim Versions 2001, 7 y 8, los cuales se incluyen en el sitio web de acompañamiento [www.librosite.net/floyd](http://www.librosite.net/floyd)). Quiero dar las gracias también a todas aquellas personas que de alguna manera, aunque sea de forma indirecta, han colaborado en este proyecto.

Para la revisión de éste y de todos los libros de texto, los autores dependemos de las inteligentes observaciones de los lectores y del equipo de revisión. Quiero dar mis más sinceras gracias a los siguientes revisores,

## XVI ■ PREFACIO

que han proporcionado multitud de sugerencias y han hecho una crítica enormemente constructiva: Bo Barry, Universidad de Carolina del Norte; Chuck McGlumphy, Belmont Technical College; y Amy Ray, Mitchell Community College.

Mi gratitud a David Buchla por sus esfuerzos a la hora de garantizar que el manual de laboratorio estuviera coordinado con el texto, así como por sus valiosas sugerencias. También quiero agradecer las sugerencias de Muhammed Arif Shabir en lo que respecta a los registros de desplazamiento.

Gracias a todos los miembros del equipo comercial de Prentice Hall, cuyo enorme trabajo ha ayudado a que mis libros estuvieran a disposición de un gran número de estudiantes en todo el mundo. Además, mi reconocimiento a todos vosotros, los profesores que habéis adoptado esta obra como libro de texto. Sin vosotros, esta obra no existiría. Espero que el lector encuentre en esta obra una valiosa herramienta de aprendizaje y un útil texto de referencia.

Tom Floyd

# **FUNDAMENTOS DE SISTEMAS DIGITALES**