FUNDAMENTOS DE LOS MICROPROCESADORES

Segunda edición

ROGER L. TOKHEIM
Henry Sibley High School
Mendota Heights, Minnesota

Traducción
JUAN MANUEL SANCHEZ
Departamento de Informática y Automática
Facultad de Ciencias Físicas
Universidad Complutense de Madrid

Revisión técnica
ANTONIO VAQUERO
Catedrático de Informática
Facultad de Ciencias Físicas
Universidad Complutense de Madrid

McGRAW-HILL

MÉXICO • BUENOS AIRES • CARACAS • GUATEMALA
LISBOA • MADRID • NUEVA YORK • PANAMÁ • SAN JUAN
SANTAFÉ DE BOGOTÁ • SANTIAGO • SÃO PAULO
AUCKLAND • HAMBURGO • LONDRES • MILÁN • MONTREAL
NUEVA DELHI • PARÍS • SAN FRANCISCO • SINGAPUR
ST. LOUIS • SIDNEY • TOKIO • TORONTO

Capítulo 2

NUMEROS, CODIGOS DE COMPUTADORA Y ARITMETICA

2.1. NUMEROS BINARIOS

Las computadoras digitales utilizan *números binarios*. El sistema de numeración binario, o de *base 2*, utiliza solamente los dígitos 0 y 1; los dígitos binarios se llaman *bits*. En los circuitos electrónicos de las computadoras el bit 0 habitualmente se representa por una tensión BAJA, mientras que el bit 1 corresponde a una tensión ALTA.

Las personas están acostumbradas a comprender el *sistema de numeración decimal*, o de *base 10*, que tiene 10 dígitos (0-9). Este sistema también tiene la característica de *valor por posición;* por ejemplo, la Figura 2.1*a* muestra que el número decimal 1327 es igual a 1000 más tres 100 más dos 10 más siete 1 (1000 + 300 + 20 + 7 = 1327).

El sistema de numeración binario también tiene la característica de valor por posición. El valor decimal de las cuatro primeras posiciones binarias se muestra en la Figura 2.1*b*. El número binario 1001 (se pronuncia uno, cero, cero, uno) se convierte a su equivalente decimal de 9. El bit del 1 del número binario de la Figura 2.1*b* se denomina *bit menos significativo* (LSB), mientras que el bit del 8 se denomina *bit más significativo* (MSB).

Los equivalentes binarios de los números decimales entre 0 y 15 se muestran en la Figura 2.1*c*. Las personas que trabajan con computadoras memorizan como mínimo estos números binarios.

Convertir el número binario 10110110 (se pronuncia uno, cero, uno, uno, cero, uno, uno, cero) a su equivalente decimal. El procedimiento se muestra en la Figura 2.2*a*. Por cada bit 1 del número binario se escribe debajo el valor de la posición decimal y después se suman los decimales (128 + 32 + 16 + 4 + 2 = 182), dando 182. Los pequeños *subíndices* de la Figura 2.2*b* se utilizan para anotar la base (a veces denominada raíz) del número. El número 10110110_2 es por tanto un número binario, o en base 2 y el número 182_{10} es un número decimal o en base 10.

Convertir el número decimal 155 a binario. La Figura 2.3 muestra un procedimiento para hacer esta conversión. El número decimal 155 se divide primero por 2, dando un cociente de 77 y un resto de 1; el resto se convierte en el bit menos significativo (LSB) del número binario y se transfiere a esta posición en la Figura 2.3. El cociente (77) se transfiere como muestra la flecha y se convierte en el siguiente dividendo. Los cocientes se dividen repetidamente por 2 hasta que el cociente se hace 0 con un resto de 1. La Figura 2.3 muestra este procedimiento. La línea inferior muestra el resultado de la conversión: $155_{10} = 10011011_2$.

Potencias de 10	10^3	10^2	10^1	10^0
Valor de posición	1000	100	10	1
Decimal	1	3	2	7
Decimal	1000 +	300 +	20 +	7 = 1327

(a) Valores de la posición en un número decimal

Potencias de 2	2^3	2^2	2^1	2^0
Valor de posición	8	4	2	1
	MSB			LSB
Binario	1	0	0	1
Decimal	8 +	0 +	0 +	1 = 9

(b) Valores de la posición en un número binario

Decimal	Binario				Decimal	Binario			
10 1	8	4	2	1	10 1	8	4	2	1
0				0	8	1	0	0	0
1				1	9	1	0	0	1
2			1	0	1 0	1	0	1	0
3			1	1	1 1	1	0	1	1
4		1	0	0	1 2	1	1	0	0
5		1	0	1	1 3	1	1	0	1
6		1	1	0	1 4	1	1	1	0
7		1	1	1	1 5	1	1	1	1

(c) Equivalentes decimales y binarios

Figura 2.1.

Potencias de 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valor de posición	128	64	32	16	8	4	2	1
Binario	1	0	1	1	0	1	1	0
Decimal	128	+	32 +	16	+	4	+ 2	= 182

(a) Conversión de binario a decimal

$$10110110_2 = 182_{10}$$

(b) Los índices designan la base del número

Figura 2.2.

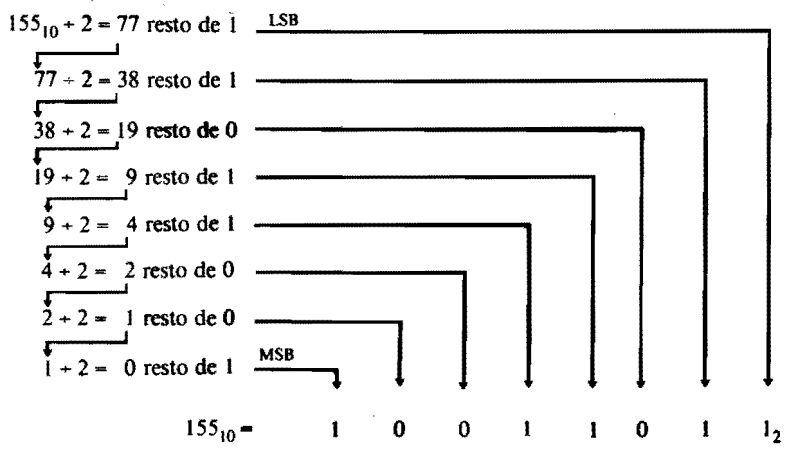


Figura 2.3.

PROBLEMAS RESUELTOS

2.1. La mayoría de las personas comprenden el sistema de numeración decimal, mientras que las computadoras digitales utilizan el sistema de numeración de base 2, o _____ .

Solución:

Las computadoras digitales utilizan el sistema de numeración de base 2, o binario, y contiene solamente los dígitos 0 y 1.

2.2. El término *bit* significa _____ _____ cuando se trata de números binarios.

Solución:

Bit significa dígito binario.

2.3. El número 100_{10} es un número en base _____ .

Solución:

El número 100_{10} es un número en base 10, como indica el subíndice 10 al final del número.

2.4. Escribir en base 2 el número uno, uno, cero, cero utilizando un subíndice.

Solución:

1100_2.

2.5. Las letras MSB significan _____ _____ _____ cuando se trata con números binarios.

Solución:

Las letras MSB significan bit más significativo. Si el número es 1000_2, el 1 es el MSB.

2.6. De memoria, convertir los siguientes números binarios en sus equivalentes decimales:
(*a*) 0001 (*b*) 0101 (*c*) 1000 (*d*) 1011 (*e*) 1111 (*f*) 0111

Solución:

Acudir a la Figura 2.1*c*. Estos números binarios han de ser memorizados. Los equivalentes decimales para los números binarios son los siguientes:

(*a*) $0001_2 = 1_{10}$ (*c*) $1000_2 = 8_{10}$ (*e*) $1111_2 = 15_{10}$
(*b*) $0101_2 = 5_{10}$ (*d*) $1011_2 = 11_{10}$ (*f*) $0111_2 = 7_{10}$

2.7. Convertir los siguientes números binarios en sus equivalentes decimales:

(*a*) 10000000 (*c*) 00110011 (*e*) 00011111
(*b*) 00010000 (*d*) 01100100 (*f*) 11111111

Solución:

Siguiendo el procedimiento mostrado en la Figura 2.1*b*, los equivalentes decimales de los números binarios son los que se indican:

(*a*) $10000000_2 = 128_{10}$ (*c*) $00110011_2 = 51_{10}$ (*e*) $00011111_2 = 31_{10}$
(*b*) $00010000_2 = 16_{10}$ (*d*) $01100100_2 = 100_{10}$ (*f*) $11111111_2 = 255_{10}$

2.8. Convertir los siguientes números decimales en sus equivalentes binarios:

(*a*) 39 (*b*) 48

Solución:

Seguir el procedimiento mostrado en la Figura 2.3. Los equivalentes binarios de los números decimales son los siguientes:

(*a*)
$39_{10} \div 2 = 19$ resto de 1 (LSB)
$19 \div 2 = 9$ resto de 1
$9 \div 2 = 4$ resto de 1
$4 \div 2 = 2$ resto de 0
$2 \div 2 = 1$ resto de 0
$1 \div 2 = 0$ resto de 1 (MSB)
$39_{10} = 100111_2$

(*b*)
$48_{10} \div 2 = 24$ resto de 0 (LSB)
$24 \div 2 = 12$ resto de 0
$12 \div 2 = 6$ resto de 0
$6 \div 2 = 3$ resto de 0
$3 \div 2 = 1$ resto de 1
$1 \div 2 = 0$ resto de 1 (MSB)
$48_{10} = 110000_2$

2.2. NUMEROS HEXADECIMALES

Una posición de la memoria de una microcomputadora puede contener el número binario 10011110. Esta larga cadena de ceros y unos es difícil de recordar y teclear. El número 10011110_2 puede convertirse en un número decimal. Una vez convertido éste es el número 158_{10}. Este proceso de conversión es demasiado largo. La mayoría de los sistemas de microcomputadoras utilizan la *notación hexadecimal* para simplificar la tarea de recordar y teclear números binarios como por ejemplo 10011110.

El sistema de numeración hexadecimal, o de base 16, utiliza los 16 símbolos del 0 al 9, A, B, C, D, E y F. Los equivalentes binarios, hexadecimales y decimales se muestran en la Figura 2.4.

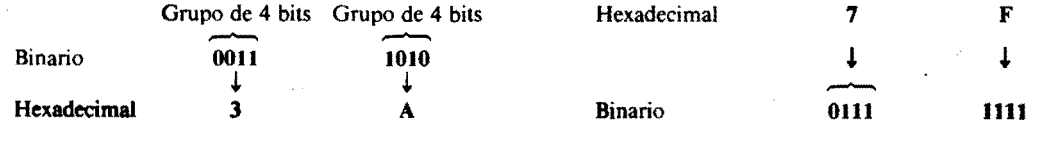
Decimal	Hexadecimal	Binario			
		8	4	2	1
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
9	9	1	0	0	1
10	A	1	0	1	0
11	B	1	0	1	1
12	C	1	1	0	0
13	D	1	1	0	1
14	E	1	1	1	0
15	F	1	1	1	1

Figura 2.4. Contar en decimal, hexadecimal y binario.

Observar en la Figura 2.4 que cada símbolo hexadecimal representa una única combinación de 4 bits. El número binario 10011110 puede entonces ser representado como 9E en hexadecimal. Esto es, la parte 1001 del número binario es igual a 9, de acuerdo con la Figura 2.4, y la parte 1110 del número binario es igual a E en hexadecimal. Por tanto 10011110_2 es igual a $9E_{16}$. Recordar que el subíndice indica la base del número.

Convertir el número binario 111010 en hexadecimal (hex). Comenzar por el LSB y dividir el número binario en grupos de 4 bits cada uno, como indica la Figura 2.5*a*. Entonces sustituir cada grupo de 4 bits por su dígito hex equivalente. El 1010_2 es igual a A en hex (ver Fig. 2.4). El 0011_2 es igual a 3 en hex. Por tanto 111010_2 es igual a $3A_{16}$.

Convertir el número hexadecimal 7F a su equivalente binario. La Figura 2.5*b* muestra que cada dígito hex es sustituido por su equivalente binario de 4 bits. En este ejemplo, el binario 0111 es sustituido por el hex 7 y 1111_2 sustituye a F_{16}. Por tanto $7F_{16}$ es igual a 1111111_2.



(a) Conversión de binario a hexadecimal *(b)* Conversión de hexadecimal a binario

Figura 2.5.

La notación hexadecimal es muy utilizada para representar números binarios. Las personas que utilizan la notación hexadecimal deben memorizar la tabla mostrada en la Figura 2.4.

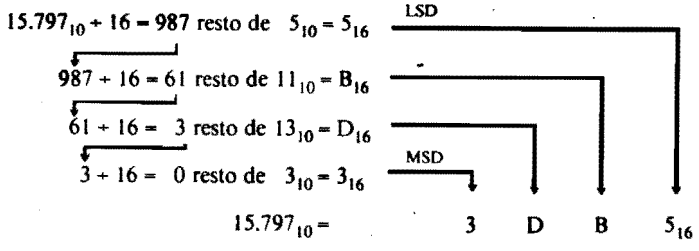
Convertir el número hexadecimal 2C6E a su equivalente decimal. El procedimiento se muestra en la Figura 2.6*a*. Los valores de posición para los cuatro primeros dígitos decimales son 4096, 256, 16 y 1. El número hexadecimal contiene catorce (E_{16}) 1, seis 16, doce (C_{16}) 256 y dos 4096. Cada valor de posición se multiplica y los productos se suman para obtener 11.374_{10}.

Convertir el número decimal 15.797 a su equivalente hexadecimal. El procedimiento se muestra en la Figura 2.6*b*. La primera línea indica 15.797_{10} dividido por 16, dando un cociente de 987_{10} con un resto de 5_{10}. El resto se convierte entonces a su equivalente hexadecimal. Por tanto 5_{10} es igual a 5_{16}. El resto hexadecimal (5_{16}) se convierte en el dígito menos significativo (LSD) del número hexadecimal. El primer cociente (987) es el dividendo en la segunda línea y se divide por 16. El segundo cociente es 61 con un resto de 11_{10}, o B hexadecimal. La línea 3 muestra 61 dividido por 16, dando un cociente de 3 con un resto de 13_{10}, o D_{16}. La cuarta línea de la Figura 2.6*b* muestra el dividendo (3) dividido por 16, dando un cociente de 0 con un resto de 3_{10}, ó 3_{16}. Cuando el cociente se hace 0, como en la línea 4, se termina el cálculo. El 3_{16} es el dígito más significativo (MSD). El procedimiento mostrado en la Figura 2.6*b* convierte el número decimal 15.797 en su equivalente hex de $3DB5_{16}$.

Potencias de 16	16^3	16^2	16^1	16^0
Valor de posición	4096	256	16	1

Hexadecimal	**2**		**C**		**6**		**E**		
	↓		↓		↓		↓		
	4096		256		16		1		
	×2		×12		×6		×14		
Decimal	8192	+	3072	+	96	+	14	=	11.374_{10}

(a) Conversión de hexadecimal a decimal



(b) Conversión de decimal a hexadecimal

Figura 2.6.

PROBLEMAS RESUELTOS

2.9. La notación hexadecimal es muy utilizada, para trabajar con las microcomputadoras, como método «abreviado» de representar números _____ (binarios, decimales).

Solución:

La notación hexadecimal es muy utilizada para representar números binarios.

2.10. El sistema de numeración hexadecimal a veces se denomina sistema de base _____.

Solución:

El sistema de numeración hexadecimal a veces se denomina sistema de base 16 debido al uso de 16 símbolos únicos.

2.11. Convertir los siguientes números hexadecimales a sus equivalentes binarios:

(*a*) C (*c*) F (*e*) 1A
(*b*) 6 (*d*) E2 (*f*) 3D

Solución:

Utilizando la tabla de la Figura 2.4, seguir el procedimiento de la Figura 2.5*b*. Los equivalentes binarios para los números hexadecimales son los siguientes:

(*a*) $C_{16} = 1100_2$ (*c*) $F_{16} = 1111_2$ (*e*) $1A_{16} = 00011010_2$
(*b*) $6_{16} = 0110_2$ (*d*) $E2_{16} = 11100010_2$ (*f*) $3D_{16} = 00111101_2$

2.12. Convertir los siguientes números binarios en sus equivalentes hexadecimales:

(*a*) 1001 (*c*) 1101 (*e*) 10000000
(*b*) 1100 (*d*) 1111 (*f*) 01111110

Solución:

Utilizando la tabla de la Figura 2.4, seguir el procedimiento de la Figura 2.5*a*. Los equivalentes hexadecimales para los números binarios son los siguientes:

(*a*) $1001_2 = 9_{16}$ (*c*) $1101_2 = D_{16}$ (*e*) $10000000_2 = 80_{16}$
(*b*) $1100_2 = C_{16}$ (*d*) $1111_2 = F_{16}$ (*f*) $01111110_2 = 7E_{16}$

2.13. Convertir los siguientes números hexadecimales en sus equivalentes decimales:

(*a*) 7E (*b*) DB (*c*) 12A3 (*d*) 34CF

Solución:

Utilizando la tabla de la Figura 2.4, seguir el procedimiento de la Figura 2.6*a*. Los equivalentes decimales a los números hexadecimales son los siguientes:

(*a*) $7E_{16} = (16 \times 7) + (1 \times 14) = 126_{10}$
(*b*) $DB_{16} = (16 \times 13) + (1 \times 11) = 219_{10}$
(*c*) $12A3_{16} = (4096 \times 1) + (256 \times 2) + (16 \times 10) + (1 \times 3) = 4771_{10}$
(*d*) $34CF_{16} = (4096 \times 3) + (256 \times 4) + (16 \times 12) + (1 \times 15) = 13.519_{10}$

2.14. $48.373_{10} =$ _____ $_{16}$.

Solución:

Utilizando la tabla de la Figura 2.4, seguir el procedimiento de la Figura 2.6*b*.

$48.373_{10} \div 16 = 3023$ resto de $5_{10} = 5_{16}$ (LSD)
$3023 \div 16 = 188$ resto de $15_{10} = F_{16}$
$188 \div 16 = 11$ resto de $12_{10} = C_{16}$
$11 \div 16 = 0$ resto de $11_{10} = B_{16}$ (MSD)

$$48.373_{10} = BCF5_{16}$$

2.3. NUMEROS BCD

Los números binarios puros se representan en notación hexadecimal para hacer más fácil la conversión. Sin embargo, la conversión binario a decimal es bastante difícil. En calculadoras, juegos e instrumentos digitales, donde son frecuentes las entradas y salidas del usuario en decimal, se utiliza un código especial para representar los números decimales. Este código se denomina *BCD (decimal-codificado-binario)*. Las equivalencias entre decimal y BCD se muestran en la tabla de la Figura 2.7*a*. Técnicamente, esta tabla detalla el *código BCD 8421*. La parte del nombre 8421 da el valor de la posición a los 4 bits del código BCD. También se utilizan otros códigos BCD, como por ejemplo el código BCD 5421 y el código de exceso 3.

Decimal	BCD			
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

(a) Código BCD 8421

Decimal	3	6	9	1
	↓	↓	↓	↓
BCD	0011	0110	1001	0001

(b) Conversión de decimal a BCD

BCD	1000	0000	0111	0010
	↓	↓	↓	↓
Decimal	8	0	7	2

(c) Conversión de BCD a decimal

Figura 2.7.

Convertir el número decimal 3691 a su equivalente BCD 8421. El procedimiento se muestra en la Figura 2.7*b*. Cada dígito decimal se traduce directamente a su equivalente BCD de 4 bits. Este ejemplo muestra que 3691_{10} es igual a $0011\ 0110\ 1001\ 0001_{BCD}$.

Convertir el número BCD 1000 0000 0111 0010 a su equivalente decimal. El procedimiento se detalla en la Figura 2.7*c*. Cada grupo de 4 bits se traduce directamente a su equivalente decimal. Este ejemplo muestra que $1000\ 0000\ 0111\ 0010_{BCD}$ es igual a 8072_{10}.

Los microprocesadores suman los números binarios puros. Sin embargo, muchos microprocesadores tienen instrucciones especiales para cambiar el resultado de las sumas a notación BCD. El número BCD se interpreta fácilmente entonces como número decimal utilizando los sencillos procedimientos mostrados en las Figuras 2.7*b* y *c*.

PROBLEMAS RESUELTOS

2.15. Las letras BCD significan _____ _____ _____ .

Solución:

Las letras BCD significan decimal codificado binario.

2.16. La notación BCD más común es el código _____ (5421, 8421).

Solución:

La notación BCD más común es el código BCD 8421.

2.17. Convertir los siguientes números decimales en sus equivalentes 8421 BCD:

(*a*) 39 (*c*) 40 (*e*) 82
(*b*) 65 (*d*) 17 (*f*) 99

Solución:

Seguir el procedimiento de la Figura 2.7*b*. Los equivalentes BCD para los números decimales son los siguientes:

(*a*) $39_{10} = 0011\ 1001_{BCD}$ (*c*) $40_{10} = 0100\ 0000_{BCD}$ (*e*) $82_{10} = 1000\ 0010_{BCD}$
(*b*) $65_{10} = 0110\ 0101_{BCD}$ (*d*) $17_{10} = 0001\ 0111_{BCD}$ (*f*) $99_{10} = 1001\ 1001_{BCD}$

2.18. Convertir los siguientes números BCD 8421 en sus equivalentes decimales:

(*a*) 1000 0000 (*c*) 1001 0010 (*e*) 0100 0011
(*b*) 0000 0001 (*d*) 0111 0110 (*f*) 0101 0101

Solución:

Seguir el procedimiento mostrado en la Figura 2.7*c*. Los equivalentes decimales para los números BCD son los siguientes:

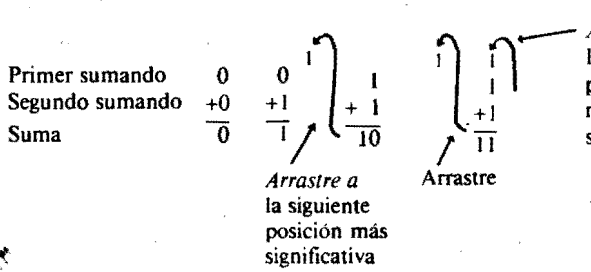
(*a*) $1000\ 0000_{BCD} = 80_{10}$ (*c*) $1001\ 0010_{BCD} = 92_{10}$ (*e*) $0100\ 0011_{BCD} = 43_{10}$
(*b*) $0000\ 0001_{BCD} = 1_{10}$ (*d*) $0111\ 0110_{BCD} = 76_{10}$ (*f*) $0101\ 0101_{BCD} = 55_{10}$

2.4. ARITMETICA BINARIA

Sumar, restar o multiplicar números binarios se realiza de forma similar a la aritmética decimal. La mayoría de los microprocesadores tienen instrucciones para sumar y restar números binarios. Los microprocesadores más avanzados tienen incluso instrucciones para multiplicar y dividir, por ejemplo, los 8086, 8088, 80286, 80386 y 68000.

Las sencillas reglas para la suma binaria se muestran en la Figura 2.8*a*. Las dos primeras reglas de la parte izquierda son obvias. La tercera regla 1 + 1 = 10 muestra que el bit más significativo es *arrastrado* a la siguiente posición de orden superior. La cuarta regla muestra que en binario 1 + 1 + 1 = 11. Aquí los sumandos y el arrastre son todos unos. El resultado es una suma de 1 con un arrastre de 1.

Sumar los números binarios 00111011 y 00101010. Este problema se ilustra en la Figura 2.8*b*. Comprobar este procedimiento. Los equivalentes decimales de los números binarios se muestran, por conveniencia, a la derecha. La suma de 00111011 y 00101010 se muestra en la Figura 2.8*b* y es 01100101_2.



(a) Reglas para la suma binaria

Arrastres

	1 1 1 1	1 1
Primer sumando	00 1 1 10 11	59
Segundo sumando	+ 00 1 0 10 10	+ 42
Suma	$01\ 1\ 00\ 01_2$	101_{10}

(b) Problema de suma binaria

Figura 2.8.

Las reglas para la resta binaria se muestran en la Figura 2.9*a*. Las tres primeras reglas son iguales que en la resta decimal. La última regla requiere un *préstamo* de la siguiente posición más significativa (la posición del 2). Con el préstamo, el minuendo se convierte en el binario 10, como el sustraendo es 1 la diferencia es 1.

				0 10
Minuendo	0	1	1	0
Sustraendo	−0	−1	−0	−1
Diferencia	0	0	1	1

(a) Reglas para la sustracción binaria

1 10	7 1
0 10 0 10	8 5
0 1 0 1 0 1 0 1	− 5 7
−0 0 1 1 1 0 0 1	28_{10}
$0\ 0\ 0\ 1\ 1\ 1\ 0\ 0_2$	

(b) Problema de sustracción binaria

Figura 2.9.

Restar el número binario 00111001 de 01010101. Este problema ejemplo se detalla en la Figura 2.9*b*. Las columnas del 4, 2 y 1 del problema de resta binaria son bastante sencillas de seguir utilizando las tres primeras reglas de la Figura 2.9*a*. La columna del 8 muestra un 1 restándose de un 0. Se toma prestado un 1 de la columna del 16 y se resta 1 de 10_2, obteniéndose una diferencia de 1 de acuerdo con la cuarta regla de la Figura 2.9*a*. Después del préstamo, la columna del 16 muestra un 1 restado del nuevo minuendo 0. Según la regla 4, hay que tomar prestado un 1 de la siguiente posición más significativa (la posición del 32). El minuendo de la posición 32 es 0, por lo que entonces hay que tomar prestado el 1 de la posición 64. La posición del 32 toma un préstamo de la posición del 64. Finalmente, la posición del 16 puede tomar prestado de la posición del 32. El minuendo de la posición del 16 es entonces 10_2, y el sustraendo es 1 dando una diferencia de 1. La posición del 32 ahora muestra 1 − 1 dando una diferencia de 0. La posición del 64 muestra 0 − 0 dando una diferencia de 0. La posición del 128 muestra 0 − 0 dando también una diferencia de 0. En resumen, el problema ejemplo de la Figura 2.9*b* muestra que el binario 00111001 restado de 01010101_2 da una diferencia de 00011100_2. El problema también se muestra en forma decimal a la derecha.

Las reglas para la multiplicación binaria se muestran en la Figura 2.10*a*. Las dos primeras reglas no necesitan explicación. El multiplicador es 1 en las dos últimas reglas. Cuando el multiplicador es 1 en la multiplicación binaria, el *multiplicando se copia* como producto. Cuando el multiplicador es 0, el producto siempre es 0.

Multiplicando	0	1	0	1
Multiplicador	×0	×0	×1	×1
Producto	0	0	0	1

(a) Reglas para la multiplicación binaria

Multiplicando	1101	13
Multiplicador	× 101	× 5
Primer producto parcial	1101	65_{10}
Segundo producto parcial	0000	
Tercer producto parcial	1101	
Producto final	1000001_2	

(b) Problema de multiplicación binaria

Figura 2.10.

Multiplicar los números binarios 1101 y 101. Este problema ejemplo se muestra en la Figura 2.10*b*. Como en la multiplicación decimal, el multiplicando se multiplica primero por el dígito menos significativo (en este caso el bit del 1). El bit del 1 del multiplicador es 1, por tanto el multiplicando se copia como primer producto parcial. El bit del 2 del multiplicador es un 0, por tanto el segundo producto parcial es 0000. Observar que éste se *desplaza* una posición a la izquierda. El bit del 4 del multiplicador es 1, por tanto el multiplicando se copia como tercer producto parcial. Observar que 1101 se copia después del segundo desplazamiento a la izquierda. Los productos parciales primero, segundo y tercero se suman, dando el producto final de 1000001_2. En resumen, la Figura 2.10*b* muestra que $1101_2 \times 101_2 = 1000001_2$ o que $13_{10} \times 5_{10} = 65_{10}$.

PROBLEMAS RESUELTOS

2.19. Resolver los siguientes problemas de suma binaria:

(*a*) 1010 + 0101 (*b*) 1101 + 0101 (*c*) 01011011 + 00001111 (*d*) 00111111 + 00011111

Solución:

Acudir a la Figura 2.8. Las sumas binarias de los problemas son las siguientes:

(*a*) 1111 (*b*) 10010 (*c*) 01101010 (*d*) 01011110

2.20. Resolver los siguientes problemas de resta binaria:

(*a*) 1110 − 1000 (*b*) 1010 − 0101 (*c*) 01100110 − 00011010 (*d*) 01111000 − 00111111

Solución:

Acudir a la Figura 2.9. Las diferencias binarias de los problemas son las siguientes:

(*a*) 0110 (*b*) 0101 (*c*) 01001100 (*d*) 00111001

2.21. En un problema de multiplicación el número de arriba se denomina _____ mientras que el de abajo se denomina multiplicador y el resultado se denomina _____.

Solución:

En un problema de multiplicación el número de arriba se llama multiplicando, mientras que el de abajo se llama multiplicador y el resultado se llama producto.

2.22. Resolver los siguientes problemas de multiplicación binaria:

(*a*) 1001 ×11 (*b*) 1101 ×1001 (*c*) 1111 ×101 (*d*) 1110 ×1110

Solución:

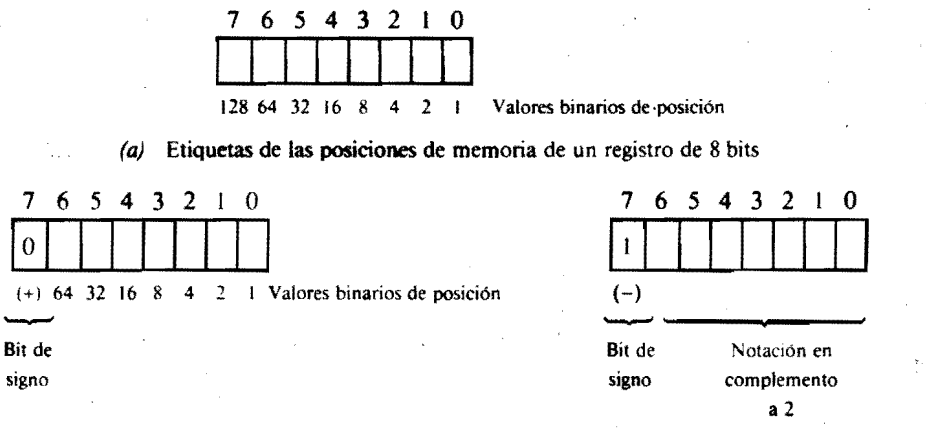
Acudir a la Figura 2.10. Los productos de los problemas son los siguientes:

(*a*) 11011 (*b*) 1110101 (*c*) 1001011 (*d*) 11000100

2.5. NOTACION EN COMPLEMENTO A 2

Generalmente, en las computadoras se utilizan los números binarios. Sin embargo, a veces se utiliza un código especial denominado *notación en complemento a 2* cuando se necesitan *números con signo.* Este sistema simplifica la circuitería de la computadora.

Un registro o posición de almacenamiento en un microprocesador puede ser como el de la Figura 2.11*a*. Este registro tiene espacio para datos de 8 bits. Las posiciones de los bits se numeran del 7 al 0. Los valores de las posiciones binarias se muestran en la parte inferior del registro. El bit 7 será el de la posición del 128, el bit 6 el de la posición del 64, etc.



(*a*) Etiquetas de las posiciones de memoria de un registro de 8 bits

(*b*) Los números positivos se identifican por un 0 en la posición del bit del signo del registro

(*c*) Los números negativos se identifican por un 1 en la posición del bit de signo del registro

Figura 2.11.

La organización más frecuente de un registro de 8 bits utilizado para almacenar *números con signo* se muestra en las Figuras 2.11*b* y *c*. El bit 7 en ambos registros es el *bit de signo.* Este bit dice si el número es (+) positivo o (−) negativo. Un 0 en la posición del bit de signo significa que el número es positivo, mientras que un 1 indica que el número es negativo.

Si el número con signo es positivo como en la Figura 2.11*b*, las restantes posiciones de memoria (6-0) contienen un número binario de 7 bits. Por ejemplo, si el contenido del registro de la Figura 2.11*b* fuese 01000001, significaría el decimal +65 (bit de signo positivo + 64 + 1). Si el contenido del registro de la Figura 2.11*b* fuese 01111111, sería $+127_{10}$ (bit de signo positi-

Decimal	Representación de números con signo	
+127	0111 1111	Números positivos representados igual que en binario puro
⋮	⋮	
+8	0000 1000	
+7	0000 0111	
+6	0000 0110	
+5	0000 0101	
+4	0000 0100	
+3	0000 0011	
+2	0000 0010	
+1	0000 0001	
+0	0000 0000	
−1	1111 1111	Números negativos representados en forma de complemento a 2
−2	1111 1110	
−3	1111 1101	
−4	1111 1100	
−5	1111 1011	
−6	1111 1010	
−7	1111 1001	
−8	1111 1000	
⋮	⋮	
−128	1000 0000	

Figura 2.12. Números decimales con signo y sus equivalentes en la notación de complemento a 2.

vo + 64 + 32 + 16 + 8 + 4 + 2 + 1). Este es el mayor número positivo que puede ser representado en este registro de 8 bits.

Si el número con signo es *negativo* como en la Figura 2.11*c*, el registro contendrá la *forma en complemento a 2* de ese número. La tabla de la Figura 2.12 muestra la notación en complemento a 2 para números positivos y negativos. Observar que los números positivos tienen un 0 en el MSB, mientras que el resto de los números corresponden a un número binario. Los números negativos tienen un 1 en el MSB. Observar la línea +0 de la tabla de la Figura 2.12. La notación en complemento a 2 para +0 es 00000000. En la línea siguiente, observar que 11111111 es la notación en complemento a 2 de −1. Imaginar que el sistema es un odómetro que cuenta hacia atrás a medida que se avanza de 00000000 a 11111111.

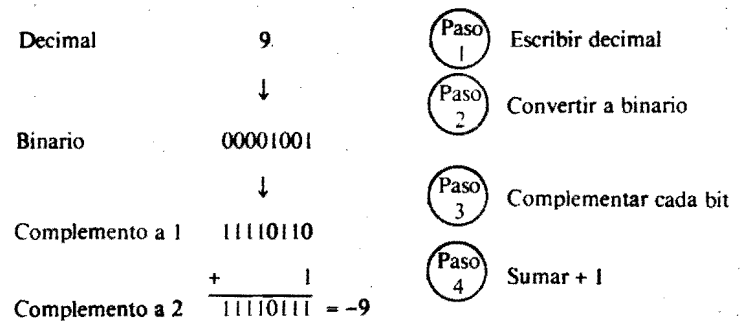
¿Cuál sería la notación en complemento a 2 de −9? Los pasos para hacer esta conversión se esbozan en la Figura 2.13*a* y son los siguientes:

Paso 1. Listar el número decimal sin signo. Escribir 9 en este ejemplo.

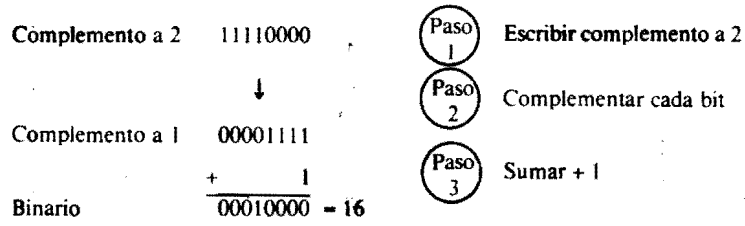
Paso 2. Convertir el número decimal a binario. Escribir el número binario 00001001 en este ejemplo.

Paso 3. Complementar cada bit formando el *complemento a 1*. En este ejemplo escribir 11110110 como complemento a 1.

Paso 4. Sumar 1 al número en complemento a 1. En este ejemplo sumar 1 a 11110110.



(a) Formación del complemento a 2 de un número negativo



(b) Cálculo del decimal equivalente para un número en complemento a 2

Figura 2.13.

El resultado es la notación en complemento a 2 para el número decimal negativo. En este ejemplo, en la Figura 2.13*a*, −9 es igual a 11110111 en forma de complemento a 2. Observar que el bit de signo (11110111) es 1, lo que significa que se trata de un número negativo.

¿Cuál es el equivalente decimal del número en complemento a 2 11110000? El procedimiento para hacer esta conversión se detalla en la Figura 2.13*b*. El procedimiento de complementar y sumar 1 es el mismo que se utiliza para convertir de binario a complemento a 2. El procedimiento de la Figura 2.13*b* muestra como se cambia cada bit del número en complemento a 2, formando el complemento a 1. Entonces se suma un 1 al número en complemento a 1, formando el número binario 00010000, que es igual a 16. Esto significa que la notación en complemento a 2 de 11110000 es igual a -16_{10}. El 16 debe ser negativo porque el bit de signo (MSB) del número en complemento a 2 es un 1.

PROBLEMAS RESUELTOS

2.23. Cuando los números con signo se almacenan en un registro de 8 bits de un microprocesador, el MSB (bit 7) se denomina bit de _____ .

Solución:

Cuando los números con signo se almacenan en un registro de 8 bits, el MSB (bit 7) se denomina bit de signo.

2.24. Determinar si los siguientes números en notación de complemento a 2 son positivos o negativos.

(*a*) 01110000 (*b*) 11001111 (*c*) 10001111 (*d*) 01010101

Solución:

(*a*) El complemento a 2 de 01110000 es un número positivo porque el bit de signo vale 0.
(*b*) El complemento a 2 de 11001111 es un número negativo porque el bit de signo vale 1.
(*c*) El complemento a 2 de 10001111 es un número negativo porque el bit de signo vale 1.
(*d*) El complemento a 2 de 01010101 es un número positivo porque el bit de signo es igual a 0.

2.25. Utilizando la tabla de la Figura 2.12, dar la notación en complemento a 2 de los siguientes números decimales con signo:

(*a*) + 1 (*b*) + 5 (*c*) + 127 (*d*) − 1 (*e*) − 2 (*f*) − 128

Solución:

Acudir a la Figura 2.12.

(*a*) + 1 = 00000001 (notación en complemento a 2)
(*b*) + 5 = 00000101 (notación en complemento a 2)
(*c*) + 127 = 01111111 (notación en complemento a 2)
(*d*) − 1 = 11111111 (notación en complemento a 2)
(*e*) − 2 = 11111110 (notación en complemento a 2)
(*f*) − 128 = 10000000 (notación en complemento a 2)

2.26. Utilizando el procedimiento de la Figura 2.13*a*, convertir los siguientes números decimales con signo en su forma de complemento a 1:

(*a*) − 10 (*b*) − 21 (*c*) − 34 (*d*) − 96

Solución:

Acudir a la Figura 2.13*a*. Las notaciones en complemento a 2 de los números decimales con signo son como sigue:

(*a*) Primer paso (convertir el decimal a binario): $10_{10} = 00001010_2$
Segundo paso (complementar): $00001010_2 \rightarrow 11110101$
Tercer paso (sumar 1): 11110101 + 1 = 11110110 (complemento a 2) = -10_{10}

(*b*) Primer paso (convertir el decimal a binario): $21_{10} = 00010101_2$
Segundo paso (complementar): $00010101_2 \rightarrow 11101010$
Tercer paso (sumar 1): 11101010 + 1 = 11101011 (complemento a 2) = -21_{10}

(*c*) Primer paso (convertir el decimal a binario): $34_{10} = 00100010_2$
Segundo paso (complementar): $00100010_2 \rightarrow 11011101$
Tercer paso (sumar 1): 11011101 + 1 = 11011110 (complemento a 2) = -34_{10}

(*d*) Primer paso (convertir el decimal a binario): $96_{10} = 01100000_2$
Segundo paso (complementar): $01100000_2 \rightarrow 10011111$
Tercer paso (sumar 1): 10011111 + 1 = 10100000 (complemento a 2) = -96_{10}

2.27. El complemento a 2 y los patrones de bits _____ (BCD, binario) son los mismos para los números decimales *positivos*.

Solución:

El complemento a 2 y los patrones de bits binarios son los mismos para los números decimales positivos.

2.28. Utilizando el procedimiento de la Figura 2.13*b*, traducir las siguientes notaciones en complemento a 2 a sus decimales equivalentes con signo:
(*a*) 11111011 (*b*) 00001111 (*c*) 10001111 (*d*) 01110111

Solución:

Los decimales equivalentes con signo de las notaciones en complemento a 2 son los siguientes:

(*a*) 11111011 (complemento a 2) = -5_{10}

$$11111011 \xrightarrow{\text{complementa}} \begin{array}{r} 00000100 \\ +\quad\ 1 \\ \hline 00000101 = 5 \end{array}$$

(*b*) 00001111 (complemento a 2) = + 15

(*c*) 10001111 (complemento a 2) = − 113

$$10001111 \xrightarrow{\text{complementa}} \begin{array}{r} 01110000 \\ +\quad\ 1 \\ \hline 01110001 = 113 \end{array}$$

(*d*) 01110111 (complemento a 2) = + 119

2.6. ARITMETICA EN COMPLEMENTO A 2

Un microprocesador puede utilizar números en complemento a 2 porque puede *complementar, incrementar* (sumar +1 a un número) y *sumar* números binarios. Los microprocesadores no tienen circuitería para restar, en su lugar utilizan un sumador y números en complemento a 2 para realizar la sustracción.

Sumar los números decimales con signo +5 y +3. El proceso se muestra en la Figura 2.14*a* en decimal y en la forma en complemento a 2. A partir de la tabla de la Figura 2.12 se encuentra que +5 es igual a 00000101 en su notación en complemento a 2, mientras que +3 es igual a 00000011. Los números en complemento a 2 00000101 y 00000011 se suman entonces igual que los números binarios regulares, obteniéndose la suma en complemento a 2 de 00001000. La suma 00001000 es igual a $+8_{10}$.

Primer sumando	(+5)	00000101
Segundo sumando	+ (+3)	+00000011
Suma	(+8)	00001000

(a) Problema de suma en complemento a 2

Primer sumando	(+3)	00000011
Segundo sumando	+ (−8)	+11111000
Suma	(−5)	11111011

(c) Problema de suma en complemento a 2

Primer sumando	(+7)	00000111
Segundo sumando	+ (−3)	+ 11111101
Suma	(+4)	(1) 00000100

Descarta overflow

(b) Problema de suma en complemento a 2

Primer sumando	(−2)	11111110
Segundo sumando	+ (−5)	+ 11111011
Suma	(−7)	(1) 11111001

Descarta overflow

(d) Problema de suma en complemento a 2

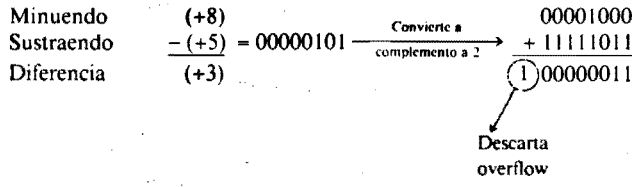
Figura 2.14.

Sumar los números decimales con signo +7 y −3. El procedimiento utilizando la suma en decimal y en complemento a 2 se muestra en la Figura 2.14*b*. A partir de la tabla de la Figura 2.12 se encuentra que +7 = 00000111 y −3 = 11111101 en la notación en complemento a 2. Los números en complemento a 2 se suman entonces (00000111 y 11111101) como si fuesen números binarios regulares, obteniéndose la suma de 100000100. El MSB es un arrastre de «overflow» del registro de 8 bits y se descarta. Por tanto, la suma en complemento a 2 es igual a 00000100, ó $+4_{10}$.

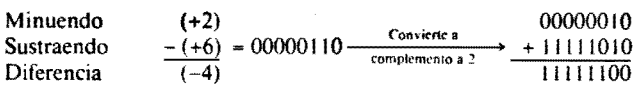
Sumar los números decimales con signo +3 y −8. El procedimiento se detalla en la Figura 2.14*c*. A partir de la tabla de la Figura 2.12 se encuentra que +3 = 00000011 y −8 = 11111000 en notación en complemento a 2. Entonces se suman estos números en complemento a 2 como si fuesen números binarios regulares, obteniéndose una suma de 11111011. De nuevo de la Figura 2.12 se determina que la suma en complemento a 2 de 11111011 es igual -5_{10}.

Sumar los decimales −2 y −5. El procedimiento se detalla en la Figura 2.14*d*. A partir de la tabla de la Figura 2.12 se encuentra que la notación en complemento a 2 de −2 = 11111110 y la de −5 = 11111011. Estos números se suman entonces como si fuesen números binarios regulares. La suma es 111111001. El MSB es un arrastre de «overflow» del registro de 8 bits y se descarta. La suma de los números en complemento a 2 (11111110 y 11111011) es entonces 11111001 (en complemento a 2). Utilizando la Figura 2.12 se determina que la suma en complemento a 2 de 11111001 es igual a -7_{10}.

Restar los números decimales con signo +5 de +8. El procedimiento se esboza en la Figura 2.15*a*. El minuendo (+8) es igual a 00001000. El sustraendo es +5, ó 00000101. El 00000101 debe convertirse a su forma en complemento a 2 (complementar y sumar 1), dando 11111011. El «minuendo» de 00001000 se suma al complemento a 2 del sustraendo 11111011 como si fuesen números binarios. La suma es igual a 100000011. El MSB es un arrastre de «overflow» del registro y se descarta, obteniéndose la suma de 00000011. A partir de la tabla de la Figura 2.12 se determina que la suma en complemento a 2 de 00000011 es igual a $+3_{10}$. Observar que al restar, el sustraendo se convierte a su forma en complemento a 2 y después se suma al minuendo. *Utilizando la representación en complemento a 2 y un sumador, el microprocesador puede realizar la sustración.*



(a) Problema de resta en complemento a 2 utilizando suma



(b) Problema de resta en complemento a 2 utilizando suma

Figura 2.15.

Restar el número decimal +6 de +2. El procedimiento se muestra en la Figura 2.15*b*. El minuendo +2 es igual 00000010. El sustraendo +6 es igual 00000110. Este sustraendo se convierte a su forma en complemento a 2 (complementar y sumar 1) dando 11111010. Los números

(00000010 y 11111010) se suman como si fuesen números binarios dando una suma de 11111100. De la tabla de la Figura 2.12 se determina que la suma en complemento a 2 de 11111100 es igual a -4_{10}.

PROBLEMAS RESUELTOS

2.29. Sumar los siguientes números decimales con signo utilizando números en complemento a 2:

(*a*) $(+7) + (+1)$ (*b*) $(+31) + (+26)$

Solución:

Seguir el procedimiento de la Figura 2.14*a*. Las sumas son las siguientes:

(*a*) $(+7)$ 00000111
$+(+1)$ +00000001
$(+8) =$ 00001000 (complemento a 2)

(*b*) $(+31)$ 00011111
$+(+26)$ +00011010
$(+57) =$ 00111001 (complemento a 2)

2.30. Sumar los siguientes números decimales con signo utilizando números en complemento a 2:

(*a*) $(+8) + (-5)$ (*b*) $(+89) + (-46)$

Solución:

Seguir el procedimiento de la Figura 2.14*b*. Las sumas son las siguientes:

(*a*) $(+8)$ 00001000
$+(-5)$ +11111011
$(+3) =$ (1)00000011 (complemento a 2)
descarta overflow

(*b*) $(+89)$ 01011001
$+(-46)$ +11010010
$(+43) =$ (1)00101011 (complemento a 2)
descarta overflow

2.31. Sumar los siguientes números decimales con signo utilizando números en complemento a 2:

(*a*) $(+1) + (-6)$ (*b*) $(+20) + (-60)$

Solución:

Seguir el procedimiento de la Figura 2.14*c*. Las sumas son las siguientes:

(*a*) $(+1)$ 00000001
$+(-6)$ +11111010
$(-5) =$ 11111011 (complemento a 2)

(*b*) $(+20)$ 00010100
$+(-60)$ +11000100
$(-40) =$ 11011000 (complemento a 2)

2.32. Sumar los siguientes números decimales con signo utilizando números en complemento a 2:

(*a*) $(-3) + (-4)$ (*b*) $(-13) + (-41)$

Solución:

Seguir el procedimiento de la Figura 2.14*d*. Las sumas son las siguientes:

(*a*) (-3) 11111101
$+(-4)$ $+11111100$
$(-7) = (1)11111001$ (complemento a 2)

descarta overflow

(*b*) (-13) 11110011
$+(-41)$ $+11010111$
$(-54) = (1)11001010$ (complemento a 2)

descarta overflow

2.33. Restar los siguientes números decimales con signo utilizando números en complemento a 2:

(*a*) $(+7)$ $-(+2)$ (*b*) $(+113)$ $-(+50)$

Solución:

Seguir el procedimiento de la Figura 2.15*a*. Los resultados de los problemas de resta son los siguientes:

(*a*)
$(+7)$
$-(+2) = 00000010$ —complementa a 2 y suma→ $+11111110$
$(+5) =$ $(1)00000101$ (complemento a 2)

descarta overflow

(*b*)
$(+113)$
$-(+50)$ 00110010 —complementa a 2 y suma→ $+11001110$
$(+63)$ $(1)00111111$ (complemento a 2)

descarta overflow

2.34. Restar los siguientes números decimales con signo utilizando números en complemento a 2:

(*a*) $(+3)$ $-(+8)$ (*b*) $(+12)$ $-(+63)$

Solución:

Seguir el procedimiento de la Figura 2.15*b*. Los resultados de los problemas de resta son los siguientes:

(*a*)
$(+3)$
$-(+8) = 00001000$ —complementa a 2 y suma→ $+11111000$
$(-5) =$ $+11111011$ (complemento a 2)

(*b*)
$(+12)$
$-(+63) = 00111111$ —complementa a 2 y suma→ $+11000001$
$(-51) =$ 11001101 (complemento a 2)

2.7. AGRUPACIONES DE BITS

Un simple dígito binario se denomina *bit*. Cuatro bits agrupados se denominan *nibble*. Ocho bits agrupados se denominan *byte*.

Una característica muy importante de cualquier microprocesador es el tamaño del acumulador. Los microprocesadores sencillos, comúnmente utilizan acumuladores de 8 bits. El *tamaño de la palabra* del microprocesador entonces es de 8 bits. En este caso 1 byte forma una palabra. Los microprocesadores tienen longitudes de palabra de 4, 8, 16 o incluso 32 bits. Un microprocesador de 16 bits tiene por tanto una longitud de palabra de 2 bytes, o de 16 dígitos binarios. Una *palabra* es un grupo de bits que es procesada como un simple número o instrucción por el microprocesador. Un microprocesador de 8 bits transfiere y almacena todos los datos en grupos de 8 bits, vía ocho conductores paralelos denominados bus de datos.

El contenido de la memoria de una microcomputadora de 8 bits puede ser el reflejado en la Figura 2.16*a*. Observar que cada posición de dirección (contenido etiquetado) contiene información en grupos de 8 bits. Cada byte se denomina *palabra de memoria*, ya que el microprocesador es una unidad de 8 bits. Cada palabra de memoria tiene un significado específico cuando es buscada y decodificada por el microprocesador. El contenido binario de la memoria de la Figura 2.16*a* puede representar:

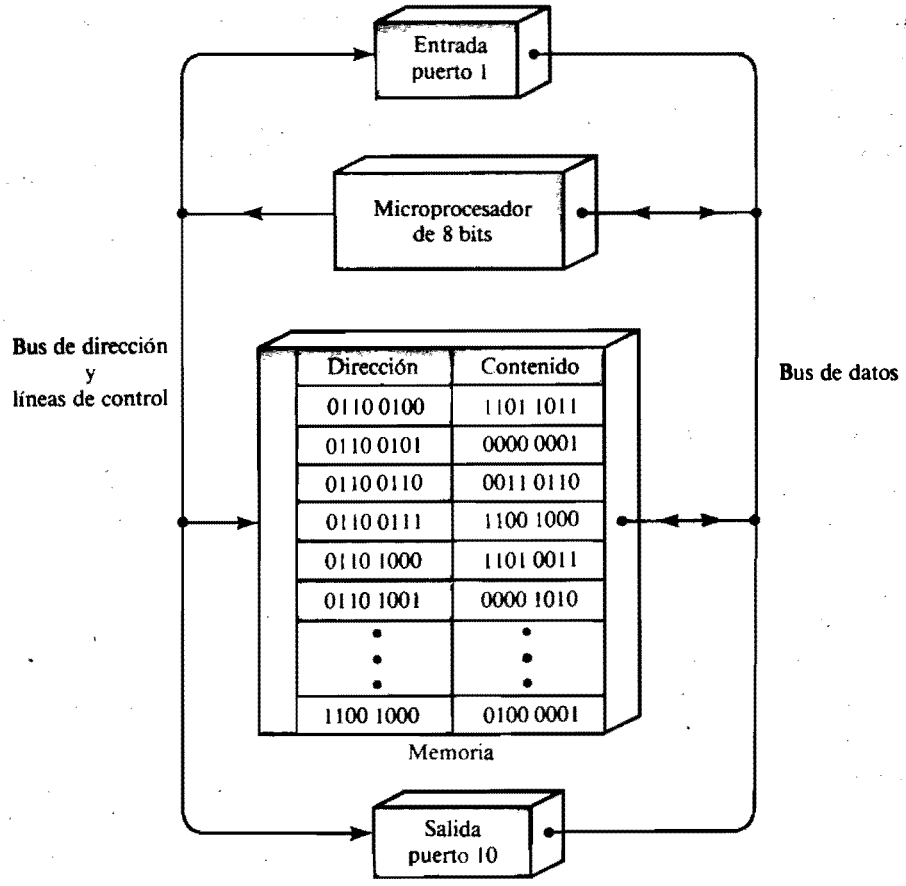
1. Un número binario.
2. Un número binario con signo.
3. Un número BCD.
4. Un carácter (una letra del alfabeto).
5. Una instrucción.
6. Una dirección de memoria.
7. Una dirección de un puerto de entrada o salida.

Considerar la posición superior de la memoria (01100100_2) de la Figura 2.16*a*. El contenido de esta posición de memoria es 11011011. Esta palabra de memoria puede interpretarse como sigue:

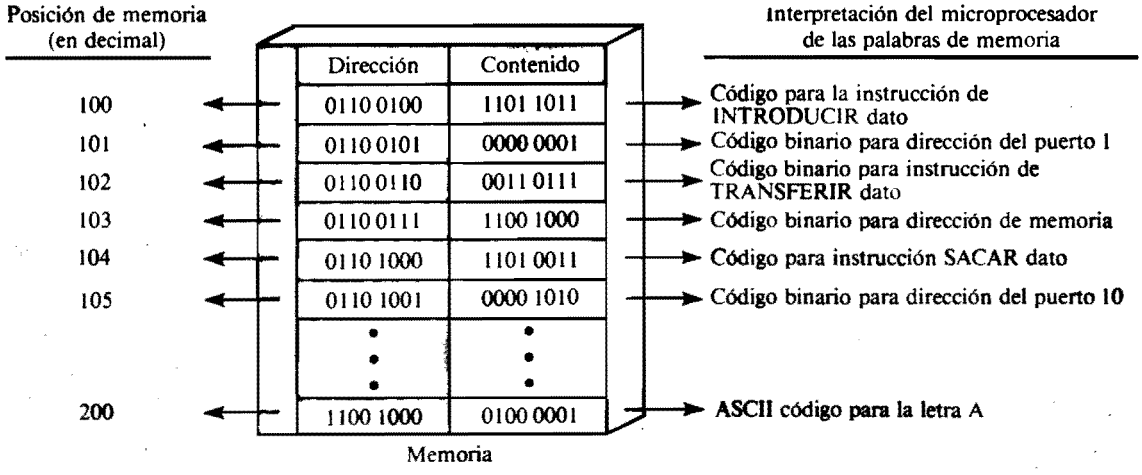
1. Como un número binario —$11011011_2 = 219_{10}$.
2. Como un número binario con signo —11011011 (complemento a 2) = -37_{10}.
3. Como un número BCD —esto no puede ser un número decimal codificado binario porque ni 1101 ni 1011 son códigos BCD.
4. Como un carácter —esto no puede ser ningún carácter ASCII (ASCII es un código alfanumérico popular especial).
5. Como una instrucción —11011011 = la instrucción INPUT (ENTRAR) del popular microprocesador Intel 8080/8085.
6. Como una dirección de memoria —$11011011_2 = DB_{16}$ = posición de memoria 219_{10}.
7. Como una dirección para un puerto de entrada o salida —11011011_2 = puerto 219_{10}.

La palabra superior de la memoria de la Figura 2.16*a* puede ser el número binario 219_{10}, el número binario con signo -37_{10}, la instrucción INPUT (ENTRAR) del microprocesador 8085, la dirección de la posición de memoria DB_{16} o la dirección del puerto de entrada o salida 219_{10}. En este ejemplo la palabra superior de la memoria (11011011) puede significar una de cinco posiciones diferentes, instrucciones o cantidades.

Si el operador de la microcomputadora 8085 arrancase el contador de programa en la dirección 100_{10} (01100100_2), entonces el microprocesador buscaría y decodificaría la palabra de memoria 11011011 como una instrucción de INTRODUCIR dato (INPUT). El microprocesador continuaría posteriormente en la siguiente dirección: 101_{10} (01100101_2). El contenido de la me-



(a) Contenido típico de una memoria binaria de una microcomputadora



(b) Interpretación por el microprocesador del contenido de la memoria

Figura 2.16.

moria en la Figura 2.16*a* es el mismo que el de la **Figura 1.4.** Recordar que el programa de la Figura 1.4 ejecuta las siguientes instrucciones:

1. INTRODUCIR (INPUT) el dato del puerto 1.
2. ALMACENAR (STORE) el dato en la posición de memoria 200.
3. SACAR (OUTPUT) el dato al puerto 10.

La interpretación que el microprocesador hace del contenido de la memoria se detalla en la Figura 2.16*b*. Las instrucciones del programa se listan en las seis posiciones superiores de memoria ($100\text{-}105_{10}$). La posición inferior de memoria (200_{10}) es la posición donde se almacena el dato. En este caso, el código ASCII de la letra A se almacena en esta posición de memoria.

En resumen, es importante observar que en las microcomputadoras, los bits se agrupan en palabras. Estas palabras de la memoria de programa son interpretadas por el microprocesador secuencialmente, *una cada vez*. Es importante para el programador saber cómo el microprocesador realiza las secuencias e interpreta los datos. Cada microprocesador tiene su propio código, único, de instrucciones; sin embargo, todos los microprocesadores realizan el secuenciamiento de las posiciones de memoria de forma similar.

PROBLEMAS RESUELTOS

2.35. Cuatro bits agrupados se denominan nibble, mientras que un grupo de 8 bits se denomina _____ .

Solución:

Una agrupación de 8 bits se denomina byte. Bajo circunstancias especiales esta agrupación de 8 bits también puede denominarse palabra.

2.36. El tamaño de _____ es una característica importante de un microprocesador y está relacionada con el número de bits transferidos y procesados como un grupo.

Solución:

El tamaño de palabra es una de las características más importantes de un microprocesador.

2.37. Acudir a la Figura 2.16*a*. El byte de datos almacenados en cualquier posición de memoria se denomina palabra de _____ .

Solución:

El byte de datos almacenados en cualquier posición de memoria se denomina palabra de memoria.

2.38. Citar siete interpretaciones posibles de una palabra de memoria de 8 bits.

Solución:

El contenido de una posición de memoria puede interpretarse como un número binario, un número binario con signo (notación en complemento a 2), un número BCD, un carácter, una instrucción, una dirección de memoria y una dirección de un puerto de entrada o salida.

2.39. Acudir a la Figura 2.16*b*. ¿Cómo interpretaría el microprocesador la palabra de memoria (00000001) de la dirección 101_{10}?

Solución:

Acudir a la Figura 2.16*b*. El microprocesador busca la palabra de memoria (00000001) esperando que le diga por *qué puerto* va a entrar el dato. La palabra de memoria indica al microprocesador que va a entrar un dato por el puerto 1.

2.40. Acudir a la Figura 2.16*b*. ¿Cómo interpretaría el microprocesador, la palabra de memoria (00110111) de la dirección 102_{10}?

Solución:

Acudir a la Figura 2.16*b*. El microprocesador busca la palabra de memoria (00110111) esperando que sea una nueva instrucción. La palabra 00110111 es decodificada por el microprocesador y significa TRANSFERIR (MOVE) dato desde el acumulador hasta la posición de memoria cuya dirección se encuentra en la siguiente posición de memoria.

2.41. Los códigos utilizados para las instrucciones son los mismos para todos los microprocesadores, y por tanto un programa escrito para un microprocesador Intel funcionará en una unidad de Motorola (verdadero o falso).

Solución:

Falso. Cada microprocesador tiene su propio código único de instrucciones.

2.8. CODIGOS ALFANUMERICOS

Los códigos que contienen caracteres alfabéticos y numéricos son necesarios cuando los microprocesadores se comunican con dispositivos como teletipos o terminales CRT (tubo de rayos catódicos). Estos códigos se denominan *códigos alfanuméricos.*

El código alfanumérico más utilizado en los sistemas microcomputadores es el *American Standard Code for Information Interchange* o ASCII (pronunciado «ask-i») *(Código Estándar Americano para Intercambio de Información).* Un listado parcial del código de 7 bits se muestra en la Figura 2.17. Este listado contiene códigos de 7 bits para números, letras mayúsculas y caracteres de puntuación. El código completo tiene también códigos para las letras minúsculas y los caracteres de control.

PROBLEMAS RESUELTOS

2.42. Los códigos binarios que representan números y letras se denominan códigos _____.

Solución:

Los códigos binarios utilizados para representar números y letras son los códigos alfanuméricos.

2.43. La representación ASCII del número 0 es 011 0000. El número 9 se representa en ASCII por _____ (7 bits).

Solución:

Acudir a la Figura 2.17. El número 9 se representa en ASCII por 011 1001.

Carácter	ASCII	Carácter	ASCII
Espacio	010 0000	A	100 0001
!	010 0001	B	100 0010
"	010 0010	C	100 0011
#	010 0011	D	100 0100
$	010 0100	E	100 0101
%	010 0101	F	100 0110
&	010 0110	G	100 0111
'	010 0111	H	100 1000
(	010 1000	I	100 1001
)	010 1001	J	100 1010
*	010 1010	K	100 1011
+	010 1011	L	100 1100
,	010 1100	M	100 1101
-	010 1101	N	100 1110
.	010 1110	O	100 1111
/	010 1111	P	101 0000
0	011 0000	Q	101 0001
1	011 0001	R	101 0010
2	011 0010	S	101 0011
3	011 0011	T	101 0100
4	011 0100	U	101 0101
5	011 0101	V	101 0110
6	011 0110	W	101 0111
7	011 0111	X	101 1000
8	011 1000	Y	101 1001
9	011 1001	Z	101 1010

Figura 2.17. Lista parcial del conjunto de caracteres ASCII.

2.44. Al eliminar los tres bits más significativos de la representación ASCII del 0 al 9 se obtendrá el _____ equivalente de ese número.

Solución:

Acudir a la Figura 2.17. Al eliminar los tres bits más significativos en la representación ASCII del 0 al 9 se obtendrá el equivalente binario o BCD de ese número.

PROBLEMAS SUPLEMENTARIOS

2.45. Mentalmente, convertir los siguientes números binarios a sus equivalentes decimales:
(*a*) 0000 (*b*) 0010 (*c*) 0011 (*d*) 0111 (*e*) 1001 (*f*) 1100
Res. (*a*) 0 (*b*) 2 (*c*) 3 (*d*) 7 (*e*) 9 (*f*) 12

2.47. $01101001_2 =$ _____ $_{10}$.
Res. 105_{10}.

2.47. $60_{10} =$ _____ $_2$.
Res. 111100_2.

2.48. El número binario 10011100 se representa como 9C en notación _____.
Res. hexadecimal.

2.49. $8D_{16} =$ _____ $_2$.
Res. 10001101_2.

2.50. $01011111_2 =$ _____ $_{16}$.
Res. $5F_{16}$.

2.51. $3C_{16} =$ _____ $_{10}$.
Res. 60_{10}.

2.52. $90_{10} =$ _____ $_{16}$.
Res. $5A_{16}$.

2.53. $92_{10} =$ _____ $_{BCD}$.
Res. $1001\ 0010_{BCD}$.

2.54. $1000\ 0110_{BCD} =$ _____ $_{10}$.
Res. 86_{10}.

2.55. Resolver los siguientes problemas de suma binaria:

(*a*)
$$\begin{array}{r} 11000011 \\ +\ 00111100 \\ \hline \end{array}$$

(*b*)
$$\begin{array}{r} 01101110 \\ +\ 00111101 \\ \hline \end{array}$$

Res. (*a*) 11111111_2 (*b*) 10101011_2

2.56. $11011000_2 - 00110011_2 =$ _____ $_2$.
Res. 10100101_2.

2.57. $1001_2 \times 1101_2 =$ _____ $_2$.
Res. 1110101_2.

2.58. Cuando se almacenan números positivos y negativos en un registro del microprocesador, si el bit de signo (MSB) es 1, el número es _____ (negativo, positivo).
Res. negativo.

2.59. 01111110 en notación en complemento a 2 representa un número _____ (negativo, positivo).

Res. positivo.

2.60. Traducir los siguientes números decimales con signo a su forma en complemento a 2 de 8 bits:
(*a*) + 12 (*b*) − 12.

Res. (*a*) 00001100 (complemento a 2) (*b*) 11110100 (complemento a 2).

2.61. Traducir los siguientes números en complemento a 2 a sus decimales con signo equivalentes:
(*a*) 01110100 (*b*) 11011101.

Res. (*a*) $+116_{10}$ (*b*) -35_{10}.

2.62. Sumar los siguientes números decimales con signo utilizando números de 8 bits en complemento a 2:

(*a*) (+ 13) (*b*) (+ 17) (*c*) (− 6)
+ (+ 8) + (− 8) + (− 14)

Res. (*a*) 00010101 (complemento a 2) (*b*) 00001001 (complemento a 2)
(*c*) 11101100 (complemento a 2)

2.63. Restar los siguientes decimales con signo utilizando números en complemento a 2 de 8 bits:

(*a*) (+ 13) (*b*) (+ 19)
− (+ 5) − (+ 29)

Res. (*a*) 00001000 (complemento a 2) (*b*) 11110110 (complemento a 2)

2.64. Un byte es un grupo que contiene _____ bits.

Res. 8.

2.65. Un nibble es un grupo que contiene _____ bits.

Res. 4.

2.66. El tamaño de palabra más popular para los sencillos microprocesadores es de _____ (8,48) bits.

Res. 8.

2.67. Acudir a la Figura 2.16*a*. Se trata de un diagrama de bloques básicos de _____ (una microcomputadora, un microprocesador).

Res. una microcomputadora.

2.68. Acudir a la Figura 2.16*b*. ¿Cómo interpreta el microprocesador la palabra de memoria (11010011) de la dirección 104_{10}?

Res. Acudir a la Figura 2.16*b*. El microprocesador busca la palabra de memoria (11010011) esperando que sea una nueva instrucción. La palabra (11010011) es decodificada por el microprocesador y significa SACAR (OUTPUT) dato del acumulador al puerto de salida cuyo número está en la siguiente posición de memoria.

2.69. Las letras ASCII significan _____ .

Res. American Standard Code for Information Interchange (Código Americano Estándar para Intercambio de Información).

2.70. Un código _____ se utiliza, probablemente, para traducir información desde un dispositivo de entrada (teclado) a un sistema microcomputador.

Res. alfanumérico o ASCII.