1 Sistemas binarios

1-1 SISTEMAS DIGITALES

Los sistemas digitales desempeñan un papel tan destacado en la vida cotidiana que el actual periodo tecnológico se conoce como “era digital”. Los sistemas digitales se utilizan en comunicaciones, transacciones de negocios, control de tráfico, navegación espacial, tratamiento médico, monitoreo meteorológico, Internet y muchas empresas comerciales, industriales y científicas. Tenemos teléfonos digitales, televisión digital, discos versátiles digitales, cámaras digitales y, desde luego, computadoras digitales. La propiedad más notable de la computadora digital es su generalidad. Es capaz de seguir una secuencia de instrucciones, llamada programa, que opera con ciertos datos. El usuario puede especificar y modificar el programa o los datos según necesidades determinadas. Gracias a esta flexibilidad, las computadoras digitales de uso general son capaces de realizar diversas tareas de procesamiento de información que cubren una amplia gama de aplicaciones.

Una característica de los sistemas digitales es su capacidad para manipular elementos discretos de información. Todo conjunto restringido a un número finito de elementos contiene información discreta. Ejemplos de conjuntos discretos son los 10 dígitos decimales, las 26 letras del alfabeto, los 52 naipes de la baraja común y las 64 casillas de un tablero de ajedrez. Las primeras computadoras digitales se usaron para efectuar cálculos numéricos. En este caso, los elementos discretos que se usaron fueron los dígitos. El término *digital* surgió de esta aplicación. En un sistema digital, los elementos discretos de información se representan mediante cantidades físicas llamadas señales. Las más comunes son señales eléctricas, como voltajes y corrientes. En los circuitos que implementan dichas señales predominan los dispositivos electrónicos llamados transistores. En casi todos los sistemas digitales electrónicos actuales, las señales emplean sólo dos valores discretos, por lo que decimos que son *binarios*. Un dígito binario, llamado *bit*, tiene dos valores: 0 y 1. Los elementos discretos de información se representan con grupos de bits llamados *códigos binarios*. Por ejemplo, los dígitos decimales 0 a 9 se representan en un sistema digital con un código de cuatro bits. Mediante el uso de diversas técnicas, es posible hacer que los grupos de bits representen símbolos discretos, y luego usar-los para desarrollar el sistema en un formato digital. Así, un sistema digital es un sistema que manipula elementos discretos de información representados internamente en forma binaria.

Las cantidades discretas de información podrían surgir de la naturaleza de los datos procesados, o bien cuantizarse a partir de un proceso continuo. Por ejemplo, una lista de nómina es un proceso inherentemente discreto que contiene nombres de empleados, números de seguro social, salarios quincenales, impuestos sobre la renta, etcétera. El cheque de paga de un empleado se procesa empleando valores discretos de datos como las letras del alfabeto (nombres), los dígitos (salario) y símbolos especiales (como $). Por otra parte, un investigador científico podría observar un proceso continuo, pero registrar únicamente cantidades específicas en forma tabular. Así, el científico está cuantizando sus datos continuos, haciendo que cada cifra de su tabla sea una cantidad discreta. En muchos casos, un convertidor analógico a digital puede efectuar automáticamente la cuantización de un proceso.

La computadora digital de uso general es el ejemplo más conocido de un sistema digital. Las partes principales de una computadora son la unidad de memoria, la unidad central de procesamiento y las unidades de entrada y salida. La unidad de memoria guarda los programas, entradas, salidas y datos intermedios. La unidad central de procesamiento realiza operaciones aritméticas y otras operaciones de procesamiento de datos especificadas por el programa. El programa y los datos que el usuario preparó se transfieren a la memoria mediante un dispositivo de entrada, como un teclado. Un dispositivo de salida, por ejemplo, una impresora, recibe los resultados de los cálculos y presenta al usuario los resultados impresos. Las computadoras digitales pueden manejar muchos dispositivos de entrada y salida. Uno muy útil es la unidad de comunicación que permite interactuar con otros usuarios a través de Internet. Las computadoras digitales son instrumentos potentes capaces de efectuar no sólo cálculos aritméticos sino también operaciones lógicas. Además, se les puede programar para que tomen decisiones con base en condiciones internas y externas.

Hay razones de peso para incluir circuitos digitales en productos comerciales. Al igual que las computadoras digitales, casi todos los dispositivos digitales son programables. Si modificamos el programa de un dispositivo programable, podremos usar el mismo hardware para muchas aplicaciones distintas. El costo de los dispositivos digitales ha bajado drásticamente gracias a los adelantos en la tecnología de los circuitos integrados digitales. A medida que aumenta el número de transistores que es posible incluir en un trozo de silicio, a fin de producir funciones complejas, el costo por unidad baja y el precio de los dispositivos digitales se reduce. Los equipos construidos con circuitos integrados digitales pueden efectuar cientos de millones de operaciones por segundo. Es posible extremar la fiabilidad con que operan los sistemas digitales empleando códigos de corrección de errores. Un ejemplo de esto es el disco digital versátil (DVD, *digital versatile disk*), en el que se graba información digital que contiene vídeo, audio y otros tipos de datos, sin perder un solo elemento. La información digital de un DVD se graba de forma tal que, al examinarse el código de cada muestra digital antes de reproducir su información, es posible identificar y corregir automáticamente cualquier error.

Un sistema digital es una interconexión de módulos digitales. Si queremos entender cómo funciona cada módulo digital, necesitaremos conocimientos básicos de circuitos digitales y de su función lógica. Los primeros siete capítulos del libro presentan las herramientas básicas del diseño digital, como las estructuras de compuertas lógicas, los circuitos combinacionales y secuenciales, y los dispositivos lógicos programables. El capítulo 8 presenta el diseño digital en el nivel de transferencia de registros (RTL, *register transfer level*). Los capítulos 9 y 10 se ocupan de los circuitos secuenciales asincrónicos y de las diversas familias de lógica digital integrada. Los capítulos 11 y 12 presentan los circuitos integrados comerciales y muestran cómo se pueden conectar en el laboratorio para realizar experimentos con circuitos digitales.

Una tendencia importante en el campo del diseño digital es el uso del lenguaje de descripción de hardware (HDL, *hardware description language*). HDL se parece a los lenguajes de programación y permite describir circuitos digitales en forma textual. Sirve para simular sistemas digitales y verificar su funcionamiento antes de crearlos en hardware. También se utiliza junto con herramientas de síntesis lógica para automatizar el diseño. Presentaremos en todo el libro descripciones de circuitos digitales en HDL.

Como ya se explicó antes, los sistemas digitales manipulan cantidades discretas de información que se representan en forma binaria. Los operandos de los cálculos podrían expresarse en el sistema numérico binario. Otros elementos discretos, como los dígitos decimales, se representan con códigos binarios. Los datos se procesan empleando señales binarias manipuladas por elementos lógicos binarios. Las cantidades se guardan en elementos de almacenamiento binarios. El objetivo de este capítulo es presentar los diversos conceptos binarios como marco de referencia para los temas que se estudiarán en los capítulos siguientes.

1-2 NÚMEROS BINARIOS

Un número decimal, como 7,392, representa una cantidad igual a 7 millares más 3 centenas, más 9 decenas, más 2 unidades. Los millares, centenas, etcétera, son potencias de 10 que están implícitas en la posición de los coeficientes. Si queremos ser más exactos, deberíamos escribir 7,392 así:

$$7 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$$

No obstante, por convención, se escriben únicamente los coeficientes y se deducen las potencias necesarias de 10 de la posición que dichos coeficientes ocupan. En general, un número con punto decimal se representa con una serie de coeficientes, así:

$$a_5 a_4 a_3 a_2 a_1 a_0 \cdot a_{-1} a_{-2} a_{-3}$$

Los coeficientes a_j son cualesquiera de los 10 dígitos (0, 1, 2, . . . , 9); el valor del subíndice j indica el valor de posición y, por tanto, la potencia de 10 por la que se deberá multiplicar ese coeficiente. Esto puede expresarse así:

$$10^5 a_5 + 10^4 a_4 + 10^3 a_3 + 10^2 a_2 + 10^1 a_1 + 10^0 a_0 + 10^{-1} a_{-1} + 10^{-2} a_{-2} + 10^{-3} a_{-3}$$

Decimos que el sistema numérico decimal es *base* 10 porque usa 10 dígitos y los coeficientes se multiplican por potencias de 10. El sistema *binario* es un sistema numérico diferente. Sus coeficientes sólo pueden tener dos valores: 0 o 1. Cada coeficiente a_j se multiplica por 2^j. Por ejemplo, el equivalente decimal del número binario 11010.11 es 26.75, como puede verse si multiplicamos los coeficientes por potencias de 2:

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 26.75$$

En general, un número expresado en un sistema base r consiste en coeficientes que se multiplican por potencias de r:

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

El valor de los coeficientes a_j varía entre 0 y $r - 1$. Para distinguir entre números con diferente base, encerramos los coeficientes en paréntesis y añadimos un subíndice que indica la base empleada (aunque a veces se hace una excepción en el caso de los números decimales, si por el contexto es obvio que la base es 10). Un ejemplo de número base 5 es

$$(4021.2)_5 = 4 \times 5^3 + 0 \times 5^2 + 2 \times 5^1 + 1 \times 5^0 + 2 \times 5^{-1} = (511.4)_{10}$$

Los valores de los coeficientes en base 5 sólo pueden ser 0, 1, 2, 3 y 4. El sistema numérico octal es un sistema base 8 que tiene ocho dígitos: 0, 1, 2, 3, 4, 5, 6 y 7. Un ejemplo de número octal es 127.4. Para determinar su valor decimal equivalente, expandimos el número como una serie de potencias con base 8:

$$(127.4)_8 = 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} = (87.5)_{10}$$

Advierta que los dígitos 8 y 9 no pueden aparecer en un número octal.

Se acostumbra tomar del sistema decimal los r dígitos requeridos si la base del número es menor que 10, y utilizar las letras del alfabeto para complementar los 10 dígitos decimales si la base del número es mayor que 10. Por ejemplo, en el sistema numérico *hexadecimal* (base 16), los primeros 10 dígitos se toman del sistema decimal, y se usan las letras A, B, C, D, E y F para los dígitos 10, 11, 12, 13, 14 y 15, respectivamente. He aquí un ejemplo de número hexadecimal:

$$(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = (46{,}687)_{10}$$

Como ya se señaló, los dígitos de los números binarios se llaman *bits*. Si un bit es igual a 0, no contribuye a la suma durante la conversión. Por tanto, la conversión de binario a decimal puede efectuarse sumando los números con potencias de 2 correspondientes a los bits que son 1. Por ejemplo,

$$(110101)_2 = 32 + 16 + 4 + 1 = (53)_{10}$$

Este número binario tiene cuatro unos. El número decimal equivalente es la suma de las cuatro potencias de 2 correspondientes. En la tabla 1-1 se presentan los primeros 24 números que se obtienen al elevar 2 a la n potencia. En computación, llamamos K (kilo) a 2^{10}, M (mega) a 2^{20}, G (giga) a 2^{30} y T (tera) a 2^{40}. Así, $4K = 2^{12} = 4096$ y $16M = 2^{24} = 16{,}777{,}216$. La capacidad de las computadoras por lo regular se da en bytes. Un *byte* es igual a ocho bits y puede representar un carácter del teclado. Un disco duro para computadora con capacidad de 4 gigabytes puede almacenar $4G = 2^{32}$ bytes (aproximadamente 4,000 millones de bytes).

Las operaciones aritméticas con números base r siguen las mismas reglas que los números decimales. Cuando se utiliza una base distinta de la conocida base 10, hay que tener cuidado

Tabla 1-1
Potencias de dos

n	2n	n	2n	n	2n
0	1	8	256	16	65,536
1	2	9	512	17	131,072
2	4	10	1,024	18	262,144
3	8	11	2,048	19	524,288
4	16	12	4,096	20	1,048,576
5	32	13	8,192	21	2,097,152
6	64	14	16,384	22	4,194,304
7	128	15	32,768	23	8,388,608

de usar únicamente los r dígitos permitidos. He aquí ejemplos de suma, resta y multiplicación de dos números binarios:

$$
\begin{array}{lr@{\quad}lr@{\quad}lr}
\text{sumando:} & 101101 & \text{minuendo:} & 101101 & \text{multiplicando:} & 1011 \\
\text{sumando:} & \underline{+100111} & \text{sustraendo:} & \underline{-100111} & \text{multiplicador:} & \underline{\times\ 101} \\
\text{suma:} & 1010100 & \text{diferencia:} & 000110 & & 1011 \\
 & & & & & 0000 \\
 & & & & & \underline{1011} \\
 & & & & \text{producto:} & 110111
\end{array}
$$

La suma de dos números binarios se calcula empleando las mismas reglas de la suma decimal, excepto que los dígitos de la suma en toda posición significativa sólo pueden ser 0 o 1. Cualquier acarreo que se genere en una posición significativa dada se sumará al par de dígitos que está en la siguiente posición significativa más alta. La resta es un poco más complicada. Las reglas siguen siendo las de la resta decimal, sólo que el préstamo en una posición significativa dada suma 2 al dígito del minuendo. (En el sistema decimal, un préstamo suma 10 al dígito del minuendo.) La multiplicación es muy sencilla. Los dígitos del multiplicador siempre son 1 o 0; por tanto, los productos parciales o bien son iguales al multiplicando, o son 0.

1-3 CONVERSIONES DE BASE NUMÉRICA

La conversión de un número base r a decimal se efectúa expandiendo el número a una serie de potencias y sumando todos los términos, como ya se explicó. A continuación presentaremos un procedimiento general para la operación inversa de convertir un número decimal en un número base r. Si el número lleva punto, será necesario separar la parte entera de la parte fraccionaria, pues cada parte se convierte de manera distinta. La conversión de un entero decimal en un número base r se efectúa dividiendo el número y todos sus cocientes sucesivos entre r y acumulando los residuos. La mejor forma de explicar el procedimiento es con un ejemplo.

EJEMPLO 1-1

Convertir 41 decimal a binario. Primero, se divide 41 entre 2 para dar un cociente entero de 20 y un residuo de $\frac{1}{2}$. Se vuelve a dividir el cociente entre 2 para dar un nuevo cociente y un nuevo residuo. El proceso se continúa hasta que el cociente entero es 0. Los *coeficientes* del número binario deseado se obtienen a partir de los *residuos*, como sigue:

	Cociente entero		**Residuo**	**Coeficiente**
$41/2 =$	20	+	$\frac{1}{2}$	$a_0 = 1$
$20/2 =$	10	+	0	$a_1 = 0$
$10/2 =$	5	+	0	$a_2 = 0$
$5/2 =$	2	+	$\frac{1}{2}$	$a_3 = 1$
$2/2 =$	1	+	0	$a_4 = 0$
$1/2 =$	0	+	$\frac{1}{2}$	$a_5 = 1$

Por tanto, la respuesta es $(41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$

El proceso aritmético se puede plantear de forma más conveniente como sigue:

Entero	**Residuo**	
41		
20	1	
10	0	
5	0	
2	1	
1	0	
0	1	101001 = respuesta

La conversión de enteros decimales a cualquier sistema base *r* es similar a este ejemplo, sólo que se divide entre *r* en vez de entre 2.

EJEMPLO 1-2

Convertir 153 decimal a octal. La base *r* en este caso es 8. Primero dividimos 153 entre 8 para obtener un cociente entero de 19 y un residuo de 1. Luego dividimos 19 entre 8 para obtener un cociente entero de 2 y un residuo de 3. Por último, dividimos 2 entre 8 para obtener un cociente de 0 y un residuo de 2. Este proceso se puede plantear así:

153		
19	1	
2	3	
0	2	$= (231)_8$

La conversión de una *fracción* decimal a binario se efectúa con un método similar al que se utiliza con enteros, pero se multiplica en lugar de dividir y se acumulan enteros en vez de residuos. En este caso, también, la mejor explicación es un ejemplo.

EJEMPLO 1-3

Convertir $(0.6875)_{10}$ a binario. Primero, multiplicamos 0.6875 por 2 para obtener un entero y una fracción. La nueva fracción se multiplica por 2 para dar un nuevo entero y una nueva fracción. El proceso se continúa hasta que la fracción es 0 o hasta que se tienen suficientes dígitos para la precisión deseada. Los coeficientes del número binario se obtienen de los enteros, así:

	Entero		**Fracción**	**Coeficiente**
$0.6875 \times 2 =$	1	+	0.3750	$a_{-1} = 1$
$0.3750 \times 2 =$	0	+	0.7500	$a_{-2} = 0$
$0.7500 \times 2 =$	1	+	0.5000	$a_{-3} = 1$
$0.5000 \times 2 =$	1	+	0.0000	$a_{-4} = 1$

Por tanto, la respuesta es $(0.6875)_{10} = (0.a_{-1}a_{-2}a_{-3}a_{-4})_2 = (0.1011)_2$

Para convertir una fracción decimal a un número expresado en base r, seguimos un procedimiento similar, multiplicando por r en vez de por 2. Los coeficientes obtenidos a partir de los enteros tendrán valores entre 0 y $r - 1$, en vez de ser sólo 0 y 1.

EJEMPLO 1-4

Convertir $(0.513)_{10}$ a octal.

$$\begin{aligned}
0.513 \times 8 &= 4.104 \\
0.104 \times 8 &= 0.832 \\
0.832 \times 8 &= 6.656 \\
0.656 \times 8 &= 5.248 \\
0.248 \times 8 &= 1.984 \\
0.984 \times 8 &= 7.872
\end{aligned}$$

La respuesta, con siete cifras significativas, se obtiene de la parte entera de los productos

$$(0.513)_{10} = (0.406517\ldots)_8$$

La conversión de números decimales que tienen tanto parte entera como parte fraccionaria se efectúa convirtiendo por separado las dos partes y combinando después las dos respuestas. Si usamos los resultados de los ejemplos 1-1 y 1-3, obtendremos

$$(41.6875)_{10} = (101001.1011)_2$$

De los ejemplos 1-2 y 1-4 tenemos

$$(153.513)_{10} = (231.406517)_8$$

1-4 NÚMEROS OCTALES Y HEXADECIMALES

Las conversiones entre binario, octal y hexadecimal desempeñan un papel importante en las computadoras digitales. Puesto que $2^3 = 8$ y $2^4 = 16$, cada dígito octal corresponde a tres dígitos binarios y cada dígito hexadecimal corresponde a cuatro dígitos binarios. En la tabla 1-2 se presentan los primeros 16 números de los sistemas numéricos decimal, binario, octal y hexadecimal.

La conversión de binario a octal se efectúa fácilmente acomodando los dígitos del número binario en grupos de tres, partiendo del punto binario tanto a la izquierda como a la derecha. Luego, se asigna el dígito octal correspondiente a cada grupo. Este ejemplo ilustra el procedimiento:

$$\begin{array}{cccccccccc}
(\,10 & 110 & 001 & 101 & 011 & \cdot & 111 & 100 & 000 & 110\,)_2 = (26153.7460)_8 \\
2 & 6 & 1 & 5 & 3 & & 7 & 4 & 0 & 6
\end{array}$$

Tabla 1-2
Números con diferente base

Decimal (base 10)	Binario (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

La conversión de binario a hexadecimal es similar, sólo que el número binario se divide en grupos de cuatro dígitos:

$$\begin{array}{cccccccl} (10 & 1100 & 0110 & 1011 & \cdot & 1111 & 0010)_2 & = (2C6B.F2)_{16} \\ 2 & C & 6 & B & & F & 2 & \end{array}$$

Es fácil recordar el dígito hexadecimal (u octal) que corresponde a cada grupo de dígitos binarios si se examinan los valores de la tabla 1-2.

La conversión de octal o hexadecimal a binario se hace invirtiendo el procedimiento anterior. Cada dígito octal se convierte a su equivalente binario de tres dígitos. Asimismo, cada dígito hexadecimal se convierte en su equivalente binario de cuatro dígitos. Los ejemplos siguientes ilustran el procededimiento:

$$\begin{array}{lcccccccl} (673.124)_8 = (& 110 & 111 & 011 & \cdot & 001 & 010 & 100 &)_2 \\ & 6 & 7 & 3 & & 1 & 2 & 4 & \end{array}$$

y

$$\begin{array}{lcccccl} (306.D)_{16} = (& 0011 & 0000 & 0110 & \cdot & 1101 &)_2 \\ & 3 & 0 & 6 & & D & \end{array}$$

Es difícil trabajar con números binarios porque requieren tres o cuatro veces más dígitos que sus equivalentes decimales. Por ejemplo, el número binario 111111111111 equivale al 4095 decimal. No obstante, las computadoras digitales emplean números binarios y hay ocasiones en que el operador o usuario humano necesita comunicarse directamente con la máquina empleando números binarios. Un esquema que retiene el sistema binario en la computadora, pero reduce el número de dígitos que la persona debe considerar, aprovecha la relación entre el sistema numérico binario y los sistemas octal y hexadecimal. Con ese método, la persona piensa en términos de números octales o hexadecimales y efectúa la conversión requerida por inspección cuando se hace necesaria la comunicación directa con la máquina. Así, el número binario 111111111111 tiene 12 dígitos y se expresa en octal como 7777 (cuatro dígitos) y en hexadecimal como FFF (tres dígitos). Cuando dos personas hablan entre sí (acerca de números binarios en la computadora), la representación octal o hexadecimal es más deseable porque se puede expresar de manera más compacta con una tercera o cuarta parte de los dígitos requeridos para el número binario equivalente. Por ello, casi todos los manuales de computadora utilizan números octales o hexadecimales para especificar cantidades binarias. La selección de cualquiera de estos dos sistemas es arbitraria, aunque se suele preferir el hexadecimal porque puede representar un byte con dos dígitos.

1-5 COMPLEMENTOS

En las computadoras digitales se usan complementos para simplificar la operación de resta y para efectuar manipulaciones lógicas. Hay dos tipos de complementos para cada sistema base r: el complemento a la base y el complemento a la base disminuida. El primero se denomina complemento a r, mientras que el segundo es el complemento a $(r - 1)$. Si sustituimos el valor de la base r en estos nombres, los dos tipos son el complemento a dos y el complemento a uno, en el caso de los números binarios, y el complemento a diez y el complemento a nueve en el caso de los números decimales.

Complemento a la base disminuida

Dado un número N en base r que tiene n dígitos, el complemento a $(r - 1)$ de N se define como $(r^n - 1) - N$. En el caso de números decimales, $r = 10$ y $r - 1 = 9$, así que el complemento a nueve de N es $(10^n - 1) - N$. En este caso, 10^n representa un número que consiste en un uno seguido de n ceros. $10^n - 1$ es un número representado por n nueves. Por ejemplo, si $n = 4$, tenemos $10^4 = 10{,}000$ y $10^4 - 1 = 9999$. De esto se sigue que el complemento a nueve de un número decimal se obtiene restando cada dígito a nueve. He aquí algunos ejemplos numéricos:

El complemento a nueve de 546700 es $999999 - 546700 = 453299$.

El complemento a nueve de 012398 es $999999 - 012398 = 987601$.

En el caso de los números binarios, $r = 2$ y $r - 1 = 1$, así que el complemento a uno de N es $(2^n - 1) - N$. Aquí también, 2^n se representa con un número binario que consiste en un uno seguido de n ceros. $2^n - 1$ es un número binario representado por n unos. Por ejemplo, si $n = 4$, tenemos $2^4 = (10000)_2$ y $2^4 - 1 = (1111)_2$. Así, el complemento a uno de un número binario se obtiene restando cada dígito a uno. Sin embargo, al restar dígitos binarios a 1 podemos tener $1 - 0 = 1$ o bien $1 - 1 = 0$, lo que hace que el bit cambie de 0 a 1 o de 1 a 0. Por tanto, el complemento a uno de un número binario se forma cambiando los unos a ceros y los ceros a unos. He aquí algunos ejemplos numéricos:

El complemento a uno de 1011000 es 0100111.

El complemento a uno de 0101101 es 1010010.

El complemento a $(r - 1)$ de los números octales y hexadecimales se obtiene restando cada dígito a 7 y F (15 decimal), respectivamente.

Complemento a la base

El complemento a r de un número N de n dígitos en base r se define como $r^n - N$, para $N \neq 0$, y 0 para $N = 0$. Si comparamos con el complemento a $(r - 1)$, veremos que el complemento a r se obtiene sumando 1 al complemento a $(r - 1)$, ya que $r^n - N = [(r^n - 1) - N] + 1$. Así pues, el complemento a 10 del número decimal 2389 es $7610 + 1 = 7611$, y se obtiene sumando 1 al valor del complemento a nueve. El complemento a dos del número binario 101100 es $010011 + 1 = 010100$, y se obtiene sumando 1 al valor del complemento a uno.

Puesto que 10^n es un número que se representa con un uno seguido de n ceros, $10^n - N$, que es el complemento a 10 de N, también puede formarse dejando como están todos los ceros menos significativos, restando a 10 el primer dígito menos significativo distinto de cero, y restando a 9 los demás dígitos a la izquierda.

El complemento a 10 de 012398 es 987602.

El complemento a 10 de 246700 es 753300.

El complemento a 10 del primer número se obtiene restando 8 a 10 en la posición menos significativa y restando a 9 todos los demás dígitos. El complemento a 10 del segundo número se obtiene dejando como están los dos ceros de la derecha, restando 7 a 10 y restando a 9 los otros tres dígitos.

De forma similar, el complemento a dos se forma dejando como están todos los ceros menos significativos y el primer uno, y sustituyendo los unos por ceros y los ceros por unos en las demás posiciones a la izquierda.

El complemento a dos de 1101100 es 0010100.

El complemento a dos de 0110111 es 1001001.

El complemento a dos del primer número se obtiene dejando como están los dos ceros menos significativos y el primer uno, y sustituyendo después los unos por ceros y los ceros por unos en las cuatro posiciones más significativas. El complemento a dos del segundo número se obtiene dejando como está el uno menos significativo y complementando todos los demás dígitos a la izquierda.

En las definiciones anteriores se supuso que los números no llevan punto. Si el número N original lleva punto, deberá quitarse temporalmente para formar el complemento a r o a $(r - 1)$, y volver a colocarlo después en el número complementado en la misma posición relativa. También vale la pena mencionar que el complemento del complemento restablece el valor original del número. El complemento a r de N es $r^n - N$. El complemento del complemento es $r^n - (r^n - N) = N$, o sea, el número original.

Resta con complementos

El método directo que se enseña en la escuela primaria para restar utiliza el concepto de préstamo. Pedimos prestado un uno de la siguiente posición más significativa cuando el dígito del minuendo es menor que el del sustraendo. El método funciona bien cuando se resta con lápiz y papel, pero cuando la resta se implementa en hardware digital el método es menos eficiente que si se usan complementos.

La resta de dos números de n dígitos sin signo, $M - N$, en base r se efectúa así:

1. Sume el minuendo, M, al complemento a r del sustraendo, N. Esto da $M + (r^n - N) = M - N + r^n$.
2. Si $M \geq N$, la suma producirá un acarreo final, r^n, que puede desecharse; lo que queda es el resultado $M - N$.
3. Si $M < N$, la suma no produce un acarreo final y es igual a $r^n - (N - M)$, que es el complemento a r de $(N - M)$. Para obtener la respuesta en una forma conocida, se toma el complemento a r de la suma y se le antepone un signo de menos.

Los ejemplos que siguen ilustran el procedimiento:

EJEMPLO 1-5

Utilizando complemento a 10, restar $72532 - 3250$.

$$\begin{aligned} M &= \ 72532 \\ \text{Complemento a 10 de } N &= +\ \underline{96750} \\ \text{Suma} &= 169282 \\ \text{Desechar acarreo final } 10^5 &= -\underline{100000} \\ \textit{Respuesta} &= \ 69282 \end{aligned}$$

Observe que M tiene cinco dígitos y N sólo tiene cuatro. Ambos números deben tener el mismo número de dígitos, así que escribimos N como 03250. La obtención del complemento a 10 de N produce un nueve en la posición más significativa. El acarreo final indica que $M \geq N$ y que el resultado es positivo.

EJEMPLO 1-6

Utilizando complemento a 10, restar $3250 - 72532$.

$$\begin{aligned} M &= \ 03250 \\ \text{Complemento a 10 de } N &= +\ \underline{27468} \\ \text{Suma} &= \ 30718 \end{aligned}$$

No hay acarreo final.
Por tanto, la respuesta es $-$(complemento a 10 de 30718) $= -69282$

Cabe señalar que, dado que $3250 < 72532$, el resultado es negativo. Puesto que estamos manejando números sin signo, en realidad es imposible obtener un resultado sin signo para este caso. Al restar con complementos, la respuesta negativa se reconoce por la ausencia de acarreo final y por el resultado complementado. Cuando trabajamos con lápiz y papel, podemos convertir la respuesta en un número negativo con signo y así expresarlo en una forma conocida.

La resta con complementos es similar en el caso de los números binarios, y se usa el procedimiento ya expuesto.

EJEMPLO 1-7

Dados los números binarios $X = 1010100$ y $Y = 1000011$, realizar las restas **a)** $X - Y$ y **b)** $Y - X$ empleando complementos a dos.

a)

$$
\begin{aligned}
X &= 1010100\\
\text{Complemento a dos de } Y &= +\underline{\;0111101}\\
\text{Suma} &= 10010001\\
\text{Desechar acarreo final } 2^7 &= -\underline{10000000}\\
\textit{Respuesta}\text{: } X - Y &= 0010001
\end{aligned}
$$

b)

$$
\begin{aligned}
Y &= 1000011\\
\text{Complemento a dos de } X &= +\underline{\;0101100}\\
\text{Suma} &= 1101111
\end{aligned}
$$

No hay acarreo final.
Por tanto, la respuesta es $Y - X = -$(complemento a dos de 1101111) $= -0010001$

La resta de números sin signo también se puede efectuar usando el complemento a $(r - 1)$. Recordemos que el complemento a $(r - 1)$ es uno menos que el complemento a r. Por ello, el resultado de sumar el minuendo al complemento del sustraendo produce una suma uno menos que la diferencia correcta cuando hay acarreo final. Quitar el acarreo final y sumar 1 a la suma se denomina *acarreo circular*.

EJEMPLO 1-8

Repetir el ejemplo 1-7 empleando complemento a uno.

a) $X - Y = 1010100 - 1000011$

$$
\begin{aligned}
X &= 1010100\\
\text{Complemento a uno de } Y &= +\underline{\;0111100}\\
\text{Suma} &= 10010000\\
\text{Acarreo circular} &= +\underline{1}\\
\textit{Respuesta}\text{: } X - Y &= 0010001
\end{aligned}
$$

b) $Y - X = 1000011 - 1010100$

$$
\begin{aligned}
Y &= 1000011\\
\text{Complemento a uno de } X &= +\underline{\;0101011}\\
\text{Suma} &= 1101110
\end{aligned}
$$

No hay acarreo final.
Por tanto, la respuesta es $Y - X = -$(complemento a uno de 1101110) $= -0010001$

Observe que el resultado negativo se obtiene tomando el complemento a uno de la suma, ya que éste es el tipo de complemento empleado. El procedimiento con acarreo circular también es válido para restar números decimales sin signo, utilizando complemento a nueve.

1-6 NÚMEROS BINARIOS CON SIGNO

Los enteros positivos (incluido el cero) se representan como números sin signo. Sin embargo, para representar enteros negativos se necesita una notación que distinga a los valores negativos. En la aritmética ordinaria, indicamos un número negativo con un signo de menos, y uno positivo, con un signo de más. Por limitaciones del hardware, las computadoras deben representar todo con dígitos binarios. Se acostumbra representar el signo con un bit colocado en la posición extrema izquierda del número. La convención es que el bit sea cero si el número es positivo, y uno si es negativo.

Es importante darse cuenta de que los números binarios, tanto con signo como sin él, se representan en las computadoras con una cadena de bits. El usuario determina si el número tiene signo o no. Si el número binario posee signo, el bit de la extrema izquierda representará el signo y el resto de los bits representarán el número. Si se supone que el número binario carece de signo, el bit de la extrema izquierda será el bit más significativo del número. Por ejemplo, la cadena de bits 01001 se considera como 9 (binario sin signo) o $+9$ (binario con signo), porque el bit de la extrema izquierda es cero. La cadena de bits 11001 representa el equivalente binario de 25 cuando se le considera un número sin signo, o -9 cuando se le considera un número con signo. Ello se debe a que el uno de la posición extrema izquierda indica que el número es negativo, y los otros cuatro bits representan 9 en binario. Normalmente, no hay problema para identificar los bits si se conoce con antelación el tipo de representación del número.

La representación de los números con signo de nuestro último ejemplo usa la convención de *magnitud con signo*. En esta notación, el número consiste en una magnitud y un símbolo ($+$ o $-$) o un bit (0 o 1) que indica el signo. Ésta es la representación de números con signo que se emplea en la aritmética ordinaria. Al implementar operaciones aritméticas en una computadora, es más conveniente usar un sistema distinto para representar números negativos, denominado sistema de *complemento con signo*. En este sistema, los números negativos se indican con su complemento. Mientras que el sistema de magnitud con signo hace negativo a un número cambiando su signo, el sistema de complemento con signo hace negativo a un número convirtiéndolo en su complemento. Puesto que los números positivos siempre inician con cero (más) en la posición de extrema izquierda, el complemento siempre iniciará con uno, lo que indica un número negativo. El sistema de complemento con signo puede utilizar el complemento a uno o a dos, aunque este último es el más común.

Por ejemplo, considere el número 9 representado en binario con ocho bits. $+9$ se representa con un bit de signo cero en la posición de extrema izquierda, seguido del equivalente binario de 9, lo que da 00001001. Cabe señalar que los ocho bits deben tener valor, por lo que se insertan ceros después del bit de signo, hasta el primer uno. Aunque sólo hay una forma de representar $+9$, hay tres formas de representar -9 con ocho bits:

representación de magnitud con signo:	10001001
representación de complemento a uno con signo:	11110110
representación de complemento a dos con signo:	11110111

En el sistema de magnitud con signo, se obtiene -9 a partir de $+9$ cambiando el bit de signo en la posición de extrema izquierda, de cero a uno. En complemento a uno con signo, se ob-