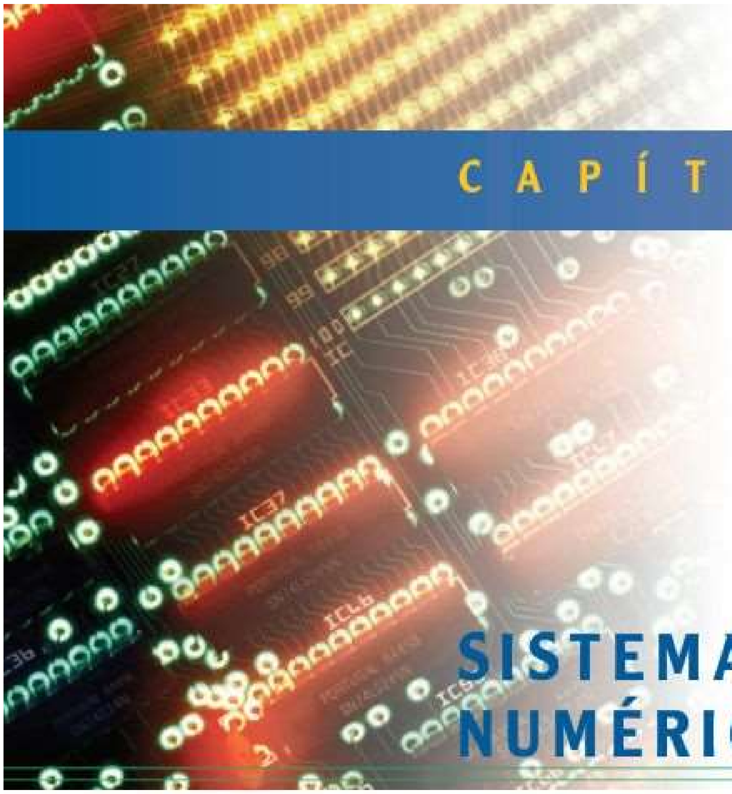


CAPÍTULO 2

SISTEMAS Y CÓDIGOS NUMÉRICOS

CONTENIDO

OBJETIVOS

Al terminar este capítulo, usted podrá:

- Convertir un número de un sistema numérico (decimal, binario, hexadecimal) a su equivalente en uno de los otros sistemas numéricos.
- Citar las ventajas del sistema numérico hexadecimal.
- Contar en hexadecimal.
- Representar números decimales mediante el código BCD; y citar las ventajas y desventajas en cuanto al uso del código BCD.
- Comprender la diferencia entre BCD y binario directo.
- Entender el propósito de los códigos alfanuméricos tales como el código ASCII.
- Explicar el método de paridad para la detección de errores.
- Determinar el bit de paridad que se adjuntará a una cadena de datos digitales.

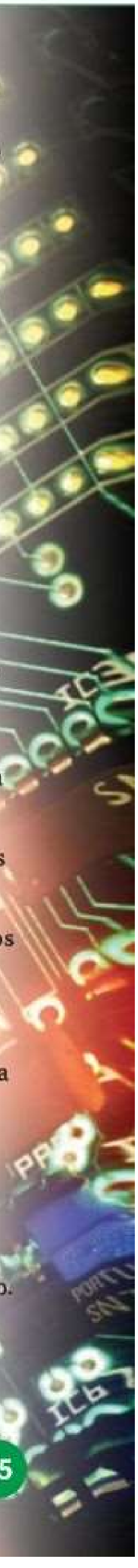
INTRODUCCIÓN

El sistema numérico binario es el más importante en los sistemas digitales, aunque también hay otros que son importantes. El sistema decimal es importante ya que se utiliza de manera universal para representar cantidades fuera de un sistema digital. Esto significa que habrá situaciones en las que los valores decimales deban convertirse en valores binarios antes de introducirse en el sistema digital. Por ejemplo, cuando oprime un número decimal en su calculadora de bolsillo (o computadora), los circuitos en el interior de la máquina convierten el número decimal en un valor binario.

De igual forma, habrá situaciones en las que los valores binarios en las salidas de un sistema digital deberán convertirse en valores decimales para presentarlos al mundo exterior. Por ejemplo, su calculadora (o computadora) utiliza números binarios para calcular la solución a un problema y después convierte los resultados en dígitos decimales antes de mostrarlos en pantalla.

Como verá más adelante, no es fácil ver un número binario y convertirlo en su valor decimal equivalente. Resulta muy tedioso tener que introducir una larga secuencia de unos (1s) y ceros (0s) en un teclado numérico, o escribir números binarios extensos en papel. En especial es difícil tratar de comunicar una cantidad binaria cuando se habla con otra persona. El sistema numérico hexadecimal (base 16) se ha convertido la forma estándar de comunicar valores numéricos en los sistemas digitales. La gran ventaja es que los números hexadecimales pueden convertirse con facilidad a binario y viceversa.

Se han ideado otros métodos para representar cantidades decimales con dígitos codificados en binario, los cuales no son en sí verdaderos sistemas numéricos, pero ofrecen la facilidad de conversión entre el código binario y el sistema numérico decimal. A esto se le conoce como decimal codificado en binario. Es muy importante que usted pueda interpretar los valores en cualquier sistema y realizar conversiones entre cualquiera de estas representaciones numéricas, ya



que las cantidades y los patrones de bits podrían estar representadas mediante cualquiera de estos métodos en cualquier sistema dado y en todo el material escrito de soporte para esos sistemas. También cubriremos otros códigos que utilizan 1s y 0s para representar cosas tales como caracteres alfanuméricos, ya que son muy comunes en los sistemas digitales.

2-1 CONVERSIONES DE BINARIO A DECIMAL

Como se explicó en el capítulo 1, el sistema numérico binario es un sistema posicional, en el cual cada dígito binario (bit) lleva un cierto peso basado en su posición relativa al LSB. Cualquier número binario puede convertirse en su equivalente decimal con sólo sumar todos los pesos de las diversas posiciones en el número binario que contengan 1. Para ilustrar esto, vamos a convertir el número 11011_2 a su equivalente decimal.

$$\begin{array}{l} 1 \quad\; 1 \quad\; 0 \quad\; 1 \quad\; 1_2 \\ 2^4 + 2^3 + 0 + 2^1 + 2^0 = 16 + 8 + 2 + 1 \\ \qquad\qquad\qquad\qquad\;\; = 27_{10} \end{array}$$

Hagamos ahora otro ejemplo con un mayor número de bits:

$$\begin{array}{l} 1 \quad\; 0 \quad\; 1 \quad\; 1 \quad\; 0 \quad\; 1 \quad\; 0 \quad\; 1_2 = \\ 2^7 + 0 + 2^5 + 2^4 + 0 + 2^2 + 0 + 2^0 = 181_{10} \end{array}$$

Observe que el procedimiento es encontrar los pesos (es decir, potencias de 2) para cada posición de bit que contenga un 1, y después hay que sumarlos. Observe también que el MSB tiene un peso de 2^7, aun y cuando es el octavo bit; esto se debe a que el LSB es el primer bit y tiene un peso de 2^0.

PREGUNTAS DE REPASO

1. Convierta el número 100011011011_2 en su equivalente decimal.
2. ¿Cuál es el peso del MSB de un número de 16 bits?

2-2 CONVERSIONES DE DECIMAL A BINARIO

Existen dos métodos para convertir un número *entero* decimal en su representación equivalente en el sistema binario. El primer método es el proceso inverso al que se describió en la sección 2-1. El número decimal tan sólo se expresa como una suma de potencias de 2, y después se escriben 1s y 0s en las posiciones de bit apropiadas. Para ilustrar lo anterior veamos lo siguiente:

$$\begin{array}{l} 45_{10} = 32 + 8 + 4 + 1 = 2^5 + 0 + 2^3 + 2^2 + 0 + 2^0 \\ \qquad\qquad\qquad\qquad\quad\;\; = 1 \quad\; 0 \quad\; 1 \quad\; 1 \quad\; 0 \quad\; 1_2 \end{array}$$

Observe que se coloca un 0 en las posiciones 2^1 y 2^4, ya que todas las posiciones deben tomarse en cuenta. Veamos otro ejemplo:

$$\begin{array}{l} 76_{10} = 64 + 8 + 4 = 2^6 + 0 + 0 + 2^3 + 2^2 + 0 + 0 \\ \qquad\qquad\qquad\quad\;\; = 1 \quad\; 0 \quad\; 0 \quad\; 1 \quad\; 1 \quad\; 0 \quad\; 0_2. \end{array}$$

División repetida

El segundo método para convertir enteros decimales a binario es el que utiliza la división entre 2. Para la conversión, que se muestra a continuación para el número 25_{10}, se requiere dividir en forma repetida el número decimal entre 2 y anotar el residuo después de cada división hasta que se obtenga un cociente de 0. El resultado binario se obtiene al escribir el primer residuo como el LSB y el último como el MSB. Este proceso, que se ilustra en el diagrama de flujo de la figura 2-1, puede usarse también para convertir de decimal a cualquier otro sistema numérico, como veremos más adelante.

$$\frac{25}{2} = 12 + \text{residuo de } 1 \quad \text{LSB}$$
$$\frac{12}{2} = 6 + \text{residuo de } 0$$
$$\frac{6}{2} = 3 + \text{residuo de } 0$$
$$\frac{3}{2} = 1 + \text{residuo de } 1$$
$$\frac{1}{2} = 0 + \text{residuo de } 1 \quad \text{MSB}$$
$$25_{10} = 1\ 1\ 0\ 0\ 1_2$$

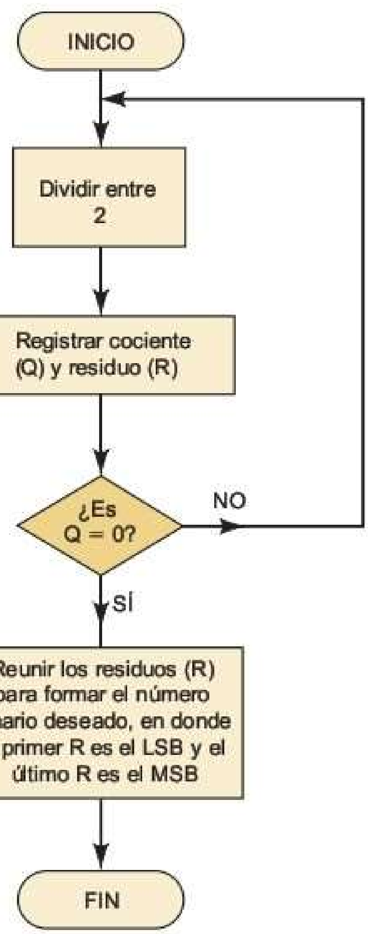


FIGURA 2-1 Diagrama de flujo para el método de división repetida de la conversión de enteros decimales a binarios. Puede usarse el mismo proceso para convertir un entero decimal a cualquier otro sistema numérico.

SUGERENCIA PARA EL USO DE CALCULADORA:

Si utiliza una calculadora para realizar las divisiones entre 2, podrá saber si el residuo es 0 o 1 y si el resultado tiene o no una parte fraccionaria. Por ejemplo, 25/2 produciría 12.5. Como hay una parte fraccionaria (.5), el residuo es un 1. Si no hubiera parte fraccionaria, como en 12/2 = 6, entonces el residuo sería 0. El siguiente ejemplo demuestra esto.

EJEMPLO 2-1

Convierta el número 37_{10} en binario. Trate de resolverlo por su cuenta antes de ver la solución.

Solución

$$\frac{37}{2} = \boxed{18}.5 \longrightarrow \text{residuo de 1 (LSB)}$$

$$\frac{\boxed{18}}{2} = 9.0 \longrightarrow \quad 0$$

$$\frac{9}{2} = 4.5 \longrightarrow \quad 1$$

$$\frac{4}{2} = 2.0 \longrightarrow \quad 0$$

$$\frac{2}{2} = 1.0 \longrightarrow \quad 0$$

$$\frac{1}{2} = 0.5 \longrightarrow \quad 1 \text{ (MSB)}$$

Por lo tanto, $37_{10} = \mathbf{100101_2}$.

Alcance de conteo

Recuerde que si utilizamos N bits, podemos contar hasta 2^N números decimales distintos que van desde 0 hasta $2^N - 1$. Por ejemplo, para $N = 4$ podemos contar desde 0000_2 hasta 1111_2, 0_{10} hasta 15_{10}, para un total de 16 números distintos. Aquí, el valor decimal más grande es $2^4 - 1 = 15$, y hay 2^4 números distintos.

Entonces, en general, podemos decir que:

Si utilizamos N bits, podemos representar números decimales que van desde 0 hasta $2^N - 1$, un total de 2^N números distintos.

EJEMPLO 2-2

(a) ¿Cuál es el intervalo de valores decimales que pueden representarse en ocho bits?

(b) ¿Cuántos bits se necesitan para representar valores decimales que van desde 0 hasta 12,500?

Solución

(a) Aquí tenemos que $N = 8$. Por ende, podemos representar números decimales desde 0 hasta $2^8 - 1 = \mathbf{255}$. Para verificar lo anterior podemos comprobar si 11111111_2 se convierte en 255_{10}.

(b) Con 13 bits podemos contar desde el 0 decimal hasta $2^{13} - 1 = 8191$. Con 14 bits podemos contar desde 0 hasta $2^{14} - 1 = 16{,}383$. Es evidente que no son suficientes 13 bits, pero 14 bits nos llevarán más allá de 12,500. Por ende, el número de bits requerido es **14**.

PREGUNTAS DE REPASO

1. Convierta 83_{10} en binario usando ambos métodos.
2. Convierta 729_{10} en binario usando ambos métodos. Revise su respuesta y convierta el resultado de vuelta en decimal.
3. ¿Cuántos bits se requieren para contar hasta 1 millón decimal?

2-3 SISTEMA NUMÉRICO HEXADECIMAL

El **sistema numérico hexadecimal** utiliza la base 16. En consecuencia, tiene 16 símbolos posibles para los dígitos. Utiliza los dígitos del 0 al 9 más las letras A, B, C, D, E y F como símbolos para los 16 dígitos. Las posiciones de los dígitos se ponderan como potencias de 16, como se muestra a continuación, en lugar de ponderarse como potencias de 10, como en el sistema decimal.

16^4	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}	16^{-3}	16^{-4}

Punto hexadecimal

La tabla 2-1 muestra las relaciones entre hexadecimal, decimal y binario. Observe que cada dígito hexadecimal representa un grupo de cuatro dígitos binarios. Es importante recordar que los dígitos hex (abreviación de "hexadecimal") de la A a la F son equivalentes a los valores decimales del 10 al 15.

TABLA 2-1

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

Conversión de hexadecimal a decimal

Es posible convertir un número hexadecimal a su equivalente decimal gracias a que la posición de cada dígito hexadecimal tiene un peso equivalente a una potencia de 16. El LSD tiene un peso de $16^0 = 1$; la siguiente posición de dígito tiene un peso de $16^1 = 16$; la siguiente tiene un peso de $16^2 = 256$; y así sucesivamente. En el siguiente ejemplo demostraremos el proceso de conversión.

SUGERENCIA PARA EL USO DE CALCULADORA:

Puede usar la función y^x de la calculadora para evaluar las potencias de 16.

$$\begin{aligned} 356_{16} &= 3 \times 16^2 + 5 \times 16^1 + 6 \times 16^0 \\ &= 768 + 80 + 6 \\ &= 854_{10} \end{aligned}$$

$$\begin{aligned} 2AF_{16} &= 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 \\ &= 512 + 160 + 15 \\ &= 687_{10} \end{aligned}$$

Observe que en el segundo ejemplo, el valor 10 se sustituyó por A y el valor 15 por F en la conversión a decimal.

Para practicar, verifique que $1BC2_{16}$ sea igual a 7106_{10}.

Conversión de decimal a hexadecimal

Recuerde que para realizar la conversión de decimal a binario utilizamos la división repetida entre 2. De igual forma, la conversión de decimal a hexadecimal puede realizarse mediante el uso de la división repetida entre 16 (figura 2-1). El siguiente ejemplo contiene dos casos de esta conversión.

EJEMPLO 2-3

(a) Convierta 423_{10} a hexadecimal.

Solución

$$\frac{423}{16} = \boxed{26} + \text{residuo de } 7$$

$$\frac{\boxed{26}}{16} = 1 + \text{residuo de } 10$$

$$\frac{1}{16} = 0 + \text{residuo de } 1$$

$$423_{10} = \boxed{1A7_{16}}$$

(b) Convierta 214_{10} a hexadecimal.

Solución

$$\frac{214}{16} = 13 + \text{residuo de } 6$$

$$\frac{13}{16} = 0 + \text{residuo de } 13$$

$$214_{10} = \boxed{D6_{16}}$$

Observe de nuevo que los residuos de los procesos de división forman los dígitos del número hexadecimal. Observe también que cualquier residuo mayor de 9 se representa por las letras de la A a la F.

SUGERENCIA PARA EL USO DE CALCULADORA:

Si utiliza una calculadora para realizar las divisiones en el proceso de conversión, los resultados incluirán una fracción decimal en vez de un residuo. Para obtener el residuo multiplique la fracción por 16. En el ejemplo 2-3(b) la calculadora habría producido lo siguiente:

$$\frac{214}{16} = 13.375$$

El residuo se convierte en $(0.375) \times 16 = 6$.

Conversión de hexadecimal a binario

El sistema numérico hexadecimal se utiliza principalmente como método "abreviado" para representar números binarios. En realidad es muy sencillo convertir un número hexadecimal en binario. *Cada* dígito hexadecimal se convierte en su equivalente binario de cuatro bits (tabla 2-1). Esto se ilustra a continuación para el número $9F2_{16}$.

$$\begin{aligned} 9F2_{16} &= \quad 9 \qquad\quad F \qquad\quad 2 \\ &\quad\;\; \downarrow \qquad\quad \downarrow \qquad\quad \downarrow \\ &= 1\;0\;0\;1 \quad 1\;1\;1\;1 \quad 0\;0\;1\;0 \\ &= 100111110010_2 \end{aligned}$$

Para practicar, verifique que $BA6_{16} = 101110100110_2$.

Conversión de binario a hexadecimal

Esta conversión es sólo el inverso del proceso antes mencionado. El número binario se separa en grupos de *cuatro* bits, y cada grupo se convierte en su dígito hexadecimal equivalente. Se agregan ceros (los que se muestran ensombrecidos) según sea necesario para completar un grupo de cuatro bits en el MSD.

$$\begin{aligned} 1\,1\,1\,0\,1\,0\,0\,1\,1\,0_2 &= \underbrace{0\;0\;1\;1}_{3}\;\underbrace{1\;0\;1\;0}_{A}\;\underbrace{0\;1\;1\;0}_{6} \\ &= 3A6_{16} \end{aligned}$$

Para hacer las conversiones entre hexadecimal y binario es necesario conocer los números binarios de cuatro bits (del 0000 hasta el 1111) y sus dígitos hexadecimales correspondientes. Una vez que memorize, la conversión podrá realizarse con rapidez sin necesidad de hacer cálculos. Esto explica por qué el sistema hexadecimal es tan útil para representar números binarios extensos.

Para practicar, verifique que $101011111_2 = 15F_{16}$.

Conteo en hexadecimal

Al contar en hexadecimal, puede incrementarse (en 1) la posición de cada dígito, desde el 0 hasta la F. Una vez que la posición de un dígito llega al valor F, se restablece a 0 y se incrementa la posición del siguiente dígito. En las siguientes secuencias de conteo hexadecimal se ilustra esto:

(a) 38, 39, 3A, 3B, 3C, 3D, 3F, 40, 41, 42
(b) 6F8, 6F9, 6FA, 6FB, 6FC, 6FD, 6FE, 6FF, 700

Observe que cuando hay un 9 en la posición de un dígito, se convierte en A cuando se incrementa.

Con N posiciones de dígitos hexadecimales podemos contar desde el 0 decimal hasta $16^N - 1$, para un total de 16^N valores distintos. Por ejemplo, con tres dígitos hexadecimales podemos contar desde 000_{16} hasta FFF_{16}, 0_{10} hasta 4095_{10}, para un total de $4096 = 16^3$ valores distintos.

Utilidad de los números hexadecimales

Los números hexadecimales se utilizan a menudo en un sistema digital como una manera "abreviada" de representar cadenas de bits. Al trabajar con las computadoras, es muy común usar cadenas de hasta 64 bits. Estas cadenas binarias no siempre representan un valor numérico, sino que, como veremos más adelante, pueden indicar algún tipo de código que conlleve información no numérica. Al trabajar con un número extenso de bits es más conveniente y menos errático escribir los números binarios en hexadecimal, ya que, como hemos visto, es bastante sencillo realizar conversiones entre hexadecimal y binario, o viceversa. Para ilustrar la ventaja de la representación hexadecimal de una cadena binaria, suponga que tiene en su poder una impresión del contenido de 50 ubicaciones de memoria, cada una de las cuales es un número de 16 bits y usted tiene que revisarlas comparándolas con una lista. ¿Qué preferiría revisar, 50 números como éste: 0110111001100111, o 50 números como éste: 6E67? ¿Y con cuál sería más probable equivocarse? No obstante, es importante tener en cuenta que todos los circuitos digitales trabajan en binario. Los números hexadecimales sólo se utilizan como una conveniencia para los humanos involucrados. Sería conveniente que memorizara el patrón binario de 4 bits para cada dígito hexadecimal. Sólo entonces se dará cuenta de la utilidad de esta herramienta en los sistemas digitales.

EJEMPLO 2-4

Convierta el número 378 decimal en un número binario de 16 bits, primero convirtiendo el número en hexadecimal.

Solución

$$\frac{378}{16} = 23 + \text{residuo de } 10_{10} = A_{16}$$

$$\frac{23}{16} = 1 + \text{residuo de } 7$$

$$\frac{1}{16} = 0 + \text{residuo de } 1$$

Por lo tanto, $378_{10} = 17A_{16}$. Este valor hexadecimal puede convertirse con facilidad en el número binario 000101111010. Por último, podemos expresar el número 378_{10} como un número de 16 bits si le agregamos cuatro 0s a la izquierda:

$$378_{10} = 0000 \quad 0001 \quad 0111 \quad 1010_2$$

EJEMPLO 2-5

Convierta el número $B2F_{16}$ en decimal.

Solución

$$\begin{aligned} B2F_{16} &= B \times 16^2 + 2 \times 16^1 + F \times 16^0 \\ &= 11 \times 256 + 2 \times 16 + 15 \\ &= 2863_{10} \end{aligned}$$

Resumen de las conversiones

En estos momentos es probable que su cabeza esté dando vueltas a medida que trata de mantener el sentido con todas estas distintas conversiones de un sistema numérico a otro. Tal vez se haya dado cuenta que muchas de estas conversiones pueden realizarse en forma *automática* en su calculadora con sólo oprimir una tecla, pero es importante que las domine para que pueda comprender el proceso. Además, ¿qué pasaría si su calculadora se quedara sin energía en un momento crucial y no tuviera un reemplazo a la mano? El siguiente resumen le ayudará, pero nada se compara con la práctica continua.

1. Al convertir de binario (o hexadecimal) a decimal, utilice el método de tomar la suma ponderada de la posición de cada bit.
2. Al convertir de decimal a binario (o hexadecimal), utilice el método de la división repetida entre 2 (o 16) y recolectar los residuos (figura 2-1).
3. Al convertir de binario a hexadecimal, divida el número en grupos de cuatro bits y convierta cada grupo en el dígito hexadecimal correcto.
4. Al convertir de hexadecimal a binario, convierta cada dígito en su equivalente de cuatro bits.

PREGUNTAS DE REPASO

1. Convierta el número $24CE_{16}$ en decimal.
2. Convierta el número 3117_{10} en hexadecimal y después de hexadecimal a binario.
3. Convierta el número 1001011110110101_2 en hexadecimal.
4. Escriba los siguientes cuatro números en esta secuencia de conteo hexadecimal: E9A, E9B, E9C, E9D, _____, _____, _____, _____.
5. Convierta el número 3527_{16} en binario.
6. ¿Qué rango de valores decimales puede representarse mediante un número hexadecimal de cuatro dígitos?

2-4 CÓDIGO BCD

Cuando se representan números, letras o palabras mediante un grupo especial de símbolos, decimos que están siendo codificados, y al grupo de símbolos se le llama *código*. Tal vez uno de los códigos más conocidos sea el Morse, en el cual una seria de puntos y rayas representan las letras del alfabeto.

Hemos visto que cualquier número decimal puede representarse mediante un número binario equivalente. El grupo de 1s y 0s en el número binario puede considerarse como un código que representa el número decimal. Cuando un número decimal se representa por su número binario equivalente, le llamamos **código binario directo**.

Todos los sistemas digitales utilizan cierta forma de números binarios para su operación interna, pero el mundo externo es decimal por naturaleza. Esto significa que con frecuencia se realizan conversiones entre los sistemas decimal y binario. Hemos visto que las conversiones entre decimal y binario pueden volverse extensas y complicadas cuando se manejan números grandes. Por esta razón, en ciertas situaciones se utiliza un medio para codificar números decimales que combina algunas características tanto del sistema decimal como del sistema binario.

Código decimal codificado en binario

Si *cada* dígito de un número decimal se representa mediante su equivalente binario, el resultado es un código que se conoce como **decimal codificado en binario** (que en lo sucesivo abreviaremos como BCD). Como un dígito decimal puede llegar hasta el 9, se requieren cuatro bits para codificar cada dígito (el código binario para el 9 es 1001).

Para ilustrar el código BCD, considere como ejemplo el número decimal 874. Cada *dígito* se cambia a su equivalente binario de la siguiente manera:

$$\begin{array}{cccl} 8 & 7 & 4 & \text{(decimal)} \\ \downarrow & \downarrow & \downarrow & \\ 1000 & 0111 & 0100 & \text{(BCD)} \end{array}$$

Como segundo ejemplo, vamos a cambiar el número 943 a su representación en código BCD:

$$\begin{array}{cccl} 9 & 4 & 3 & \text{(decimal)} \\ \downarrow & \downarrow & \downarrow & \\ 1001 & 0100 & 0011 & \text{(BCD)} \end{array}$$

Una vez más, cada dígito decimal se cambia a su equivalente binario directo. Observe que *siempre* se utilizan cuatro bits para cada dígito.

Así, el código BCD representa cada dígito del número decimal mediante un número binario de cuatro bits. Es evidente que sólo se utilizan los números binarios de cuatro bits del 0000 al 1001. El código BCD no utiliza los números 1010, 1011, 1100, 1101, 1110 y 1111. En otras palabras, sólo se utilizan 10 de los 16 posibles grupos de código binario de cuatro bits. Si llega a aparecer uno de los números "prohibidos" de cuatro bits en una máquina que utilice el código BCD, por lo general, es una indicación de que se produjo un error.

EJEMPLO 2-6

Convierta el número 0110100000111001 (BCD) en su equivalente decimal.

Solución

Divida el número BCD en grupos de cuatro bits y convierta cada grupo en decimal.

$$\underbrace{0110}_{6}\,\underbrace{1000}_{8}\,\underbrace{0011}_{3}\,\underbrace{1001}_{9}$$

EJEMPLO 2-7

Convierta el número BCD 011111000001 en su equivalente decimal.

Solución

$$\underbrace{0111}_{7}\,\underbrace{1100}_{\downarrow}\,\underbrace{0001}_{1}$$

El grupo con el código prohibido
indica un error en el número BCD

Comparación entre BCD y binario

Es importante entender que BCD no es otro sistema numérico como el binario, el decimal o el hexadecimal. De hecho, se utiliza el sistema decimal pero cada dígito está codificado en su equivalente binario. También es importante comprender que un número BCD *no* es lo mismo que un número binario directo. Un número binario directo toma el número decimal *completo* y lo representa en binario; el código BCD convierte *cada dígito* decimal en binario de manera individual. Para ilustrar esto, tome el número 137 y compare sus códigos binario directo y BCD:

$$137_{10} = 10001001_2 \qquad \text{(binario)}$$

$$137_{10} = 0001\ \ 0011\ \ 0111 \qquad \text{(BCD)}$$

Para representar el 137, el código BCD requiere 12 bits, mientras que el código binario directo sólo requiere de ocho bits. El código BCD requiere más bits que el binario directo para representar números decimales de más de un dígito, ya que no utiliza todos los grupos de cuatro bits posibles, como se indicó antes, y es, por lo tanto, algo ineficiente.

La principal ventaja del código BCD es la relativa facilidad de convertir BCD en decimal y viceversa. Sólo necesitan recordarse los grupos de código de cuatro bits para los dígitos decimales del 0 al 9. Esta facilidad de conversión es muy importante desde el punto de vista del hardware, ya que en un sistema digital son los circuitos lógicos los que realizan las conversiones hacia y desde decimal.

PREGUNTAS DE REPASO

1. Represente el valor decimal 178 mediante su equivalente binario directo. Luego codifique el mismo número decimal en BCD.
2. ¿Cuántos bits se requieren para representar un número decimal de ocho dígitos en BCD?
3. ¿Cuál es la ventaja de codificar un número decimal en BCD, en lugar de hacerlo en binario directo? ¿Cuál es la desventaja?

2-5 CÓDIGO GRAY

Los sistemas digitales operan a velocidades muy elevadas y responden a los cambios que se producen en las entradas digitales. Al igual que en la vida real, cuando varias condiciones de entrada están cambiando al mismo tiempo la situación puede malinterpretarse, con lo cual se puede llegar a producir una reacción errónea. Cuando se ven los bits en una secuencia de conteo binario, a menudo hay varios bits que deben cambiar estados al mismo tiempo. Por ejemplo, considere cuando el número binario de tres bits correspondiente al 3 decimal cambia a 4: los tres bits deben cambiar de estado.

Para reducir la probabilidad de que un circuito digital malinterprete una entrada cambiante, se desarrolló el **código Gray** como una manera de representar una secuencia de números. El aspecto único del código Gray es que, entre dos números sucesivos en la secuencia sólo un bit cambia. La tabla 2-2 muestra la traducción entre el valor del código binario de tres bits y el código Gray. Para convertir de binario a Gray sólo hay que empezar en el bit más significativo y usarlo como el MSB de Gray, como muestra la figura 2-2(a). Después se compara el MSB binario con el siguiente bit binario (B1). Si son iguales, entonces G1 = 0; si son distintos, entonces G1 = 1. Para encontrar G0 se compara B1 con B0.

TABLA 2-2 Equivalencia entre el código binario de tres bits y el código Gray.

B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

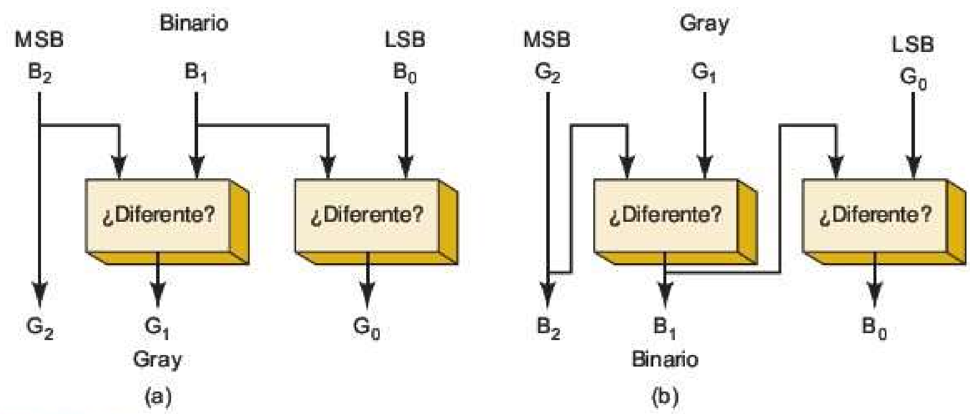
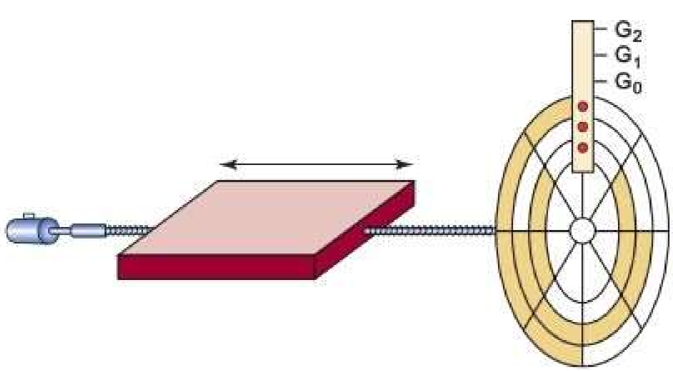


FIGURA 2-2 Conversión de (a) binario a Gray y de (b) Gray a binario.

La figura 2-2(b) muestra la conversión del código Gray a binario. Observe que el MSB en Gray siempre es el mismo que el MSB en binario. El siguiente bit binario se encuentra comparando el bit *binario* a la izquierda con el *bit correspondiente en código Gray*. Los bits similares producen un 0 y los bits distintos un 1. La aplicación más común del código Gray es en los codificadores de posición de eje, como muestra la figura 2-3. Estos dispositivos producen un valor binario que representa la posición de un eje mecánico giratorio. Un codificador de eje práctico utiliza mucho más de tres bits y divide la rotación en mucho más de ocho segmentos, por lo que puede detectar incrementos de rotación mucho más pequeños.

FIGURA 2-3 Un codificador de eje de ocho posiciones y tres bits.



PREGUNTAS DE REPASO

1. Convierta el número 0101 (binario) en su equivalente en código de Gray.
2. Convierta el número 0101 (código de Gray) en su número binario equivalente.

2-6 INTEGRACIÓN DE LOS SISTEMAS NUMÉRICOS

La tabla 2-3 muestra la representación de los números decimales del 1 al 15 en los sistemas numéricos binario y hexadecimal, y también en los códigos BCD y Gray. Examine esta tabla con cuidado y asegúrese de comprender de dónde proviene. Observe en especial cómo la representación en BCD siempre usa cuatro bits para cada dígito decimal.

TABLA 2-3

Decimal	Binario	Hexadecimal	BCD	GRAY
0	0	0	0000	0000
1	1	1	0001	0001
2	10	2	0010	0011
3	11	3	0011	0010
4	100	4	0100	0110
5	101	5	0101	0111
6	110	6	0110	0101
7	111	7	0111	0100
8	1000	8	1000	1100
9	1001	9	1001	1101
10	1010	A	0001 0000	1111
11	1011	B	0001 0001	1110
12	1100	C	0001 0010	1010
13	1101	D	0001 0011	1011
14	1110	E	0001 0100	1001
15	1111	F	0001 0101	1000

2-7 BYTE, NIBBLE Y PALABRA

Bytes

La mayoría de las microcomputadoras maneja y almacena datos binarios e información en grupos de ocho bits, por lo que una cadena de ocho bits tiene un nombre especial: **byte**. Un byte consiste de ocho bits y puede representar cualquier tipo de datos o de información. Los siguientes ejemplos ilustrarán este punto.

EJEMPLO 2-8

¿Cuántos bytes hay en una cadena de 32 bits?

Solución

32/8 = 4; por lo tanto, hay **cuatro** bytes en una cadena de 32 bits.