REC

UNJBG

Tema 01:

Sistemas de numeración

Unidad 01: Álgebra booleana y análisis combinacional

Material basado en:
Fundamentos de Sistemas Digitales, Novena edición por Thomas Floyd

Números decimales

- En un **sistema numérico ponderado**, la posición de cada dígito recibe un peso determinado por la **base** del sistema.
- En el caso del **sistema decimal**, su **radix** es **10**, ya que emplea únicamente diez símbolos (del 0 al 9) para representar cualquier número.
- Los **pesos de columna** en los números decimales son potencias de diez que aumentan de derecha a izquierda, comenzando por $10^0 = 1$:

$$\ldots 10^5\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0.$$

- Para los **números decimales fraccionarios**, los pesos de columna son potencias negativas de diez, que disminuyen de izquierda a derecha.

$$10^2\ 10^1\ 10^0.\ 10^{-1}\ 10^{-2}\ 10^{-3}\ 10^{-4}\ \ldots$$

Números decimales

- Los números decimales pueden expresarse como la suma de los productos de cada dígito por el valor de la columna correspondiente a ese dígito. Así, el número 9240 puede expresarse como

$$(9 \times 10^3) + (2 \times 10^2) + (4 \times 10^1) + (0 \times 10^0)$$

or

$$9 \times 1{,}000 + 2 \times 100 + 4 \times 10 + 0 \times 1$$

- **Ejemplo**: Expresa el número 480,52 como la suma de los valores de cada dígito.
- **Solución**:

$$480.52 = (4 \times 10^2) + (8 \times 10^1) + (0 \times 10^0) + (5 \times 10^{-1}) + (2 \times 10^{-2})$$

Números binarios

- En los sistemas digitales se utiliza el sistema numérico binario. El binario es un sistema en base dos y utiliza los dígitos 0 y 1 para representar cantidades.
- Los pesos de columna de los números binarios son potencias de dos que aumentan de derecha a izquierda empezando por $2^0 = 1$:

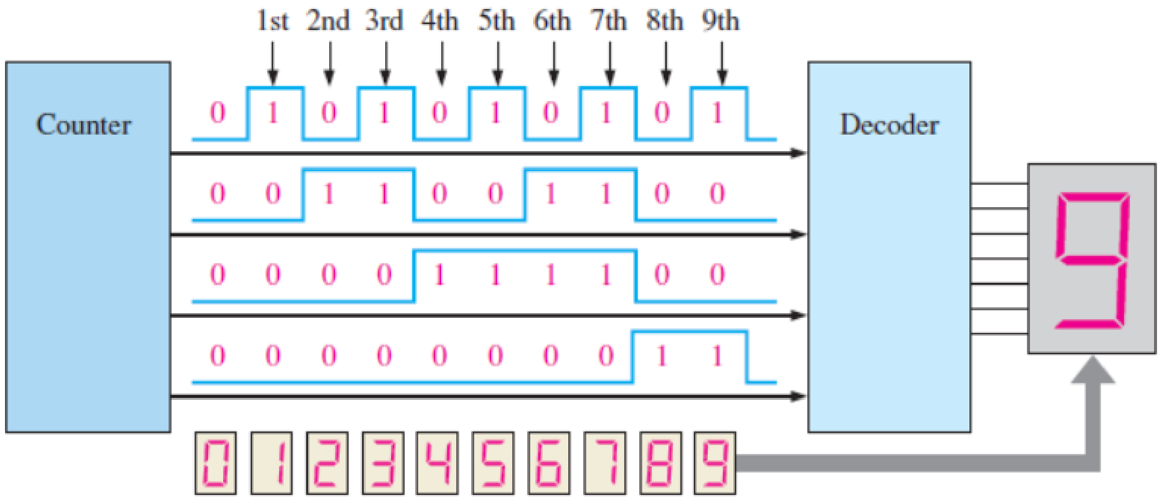
$$\ldots 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0.$$

- Para los números binarios fraccionarios, los pesos de columna son potencias negativas de dos que disminuyen de izquierda a derecha:

$$2^2\ 2^1\ 2^0.\ 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}\ \ldots$$

Números binarios

- Se muestra una secuencia de recuento binario para los números del cero al quince.
- Observa el patrón de ceros y unos en cada columna.
- Los contadores digitales suelen tener este mismo patrón de dígitos:



Número decimal	Número binario			
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
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Conversiones binarias

- El equivalente decimal de un número binario se puede determinar sumando los valores de columna de todos los bits que son 1 y descartando todos los bits que son 0.
- **Ejemplo**: Convierte el número binario `100101.01` a decimal.
- **Solución**: Empieza escribiendo los pesos de las columnas; luego suma los pesos que corresponden a cada 1 del número.

$$
\begin{array}{cccccccc}
2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0. & 2^{-1} & 2^{-2} \\
32 & 16 & 8 & 4 & 2 & 1\,. & \tfrac{1}{2} & \tfrac{1}{4} \\
1 & 0 & 0 & 1 & 0 & 1. & 0 & 1 \\
32 & & & +4 & & +1 & & +\tfrac{1}{4} = 37\tfrac{1}{4}
\end{array}
$$

Conversiones binarias

- Puedes convertir un número entero decimal a binario invirtiendo el procedimiento. Escribe el peso decimal de cada columna y coloca 1's en las columnas que sumen el número decimal.
- **Ejemplo**: Convertir el número decimal **49** a binario.
- **Solución**: Escribe los pesos de las columnas hasta que el último número sea mayor que el que quieres convertir.

$$
\begin{array}{ccccccc}
2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0. \\
64 & 32 & 16 & 8 & 4 & 2 & 1. \\
0 & 1 & 1 & 0 & 0 & 0 & 1.
\end{array}
$$

Conversiones binarias

- Puedes convertir una fracción decimal a binario multiplicando repetidamente los resultados fraccionarios de multiplicaciones sucesivas por 2. Los acarreos forman el número binario.
- **Ejemplo**: Convierte la fracción decimal `0.188` a binario multiplicando repetidamente los resultados fraccionarios por 2.
- **Solución**:

$$
\begin{array}{lll}
0.188 \times 2 = 0.376 & \text{acarreo} = 0 & \text{MSB} \\
0.376 \times 2 = 0.752 & \text{acarreo} = 0 & \\
0.752 \times 2 = 1.504 & \text{acarreo} = 1 & \\
0.504 \times 2 = 1.008 & \text{acarreo} = 1 & \\
0.008 \times 2 = 0.016 & \text{acarreo} = 0 & \downarrow
\end{array}
$$

Respuesta = .00110 (para cinco cifras significativas)

Aritmética binaria

Suma binaria

- Las cuatro reglas básicas para sumar dígitos binarios son:

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

- Cuando un acarreo de entrada = 1 debido a un resultado anterior, las reglas son

Bits de acarreo

$1 + 0 + 0 = 01$	Suma de 1 con acarreo 0
$1 + 1 + 0 = 10$	Suma de 0 con acarreo 1
$1 + 0 + 1 = 10$	Suma de 0 con acarreo 1
$1 + 1 + 1 = 11$	Suma de 1 con acarreo 1

Aritmética binaria

Suma binaria

- **Ejemplo:** Suma los números binarios 00111 y 10101 y muestra la suma decimal equivalente.
- **Solución:**

$$
\begin{array}{rcr}
{\scriptstyle 0\,1\,1\,1\ } & & \\
00111 & & 7 \\
10101 & & 21 \\
\hline
11100 & = & 28
\end{array}
$$

Aritmética binaria

Resta binaria

- Las cuatro reglas básicas para la resta de números binarios son:

$$0 - 0 = 0$$
$$1 - 1 = 0$$
$$1 - 0 = 1$$
$$10 - 1 = 1 \qquad 0 - 1 \text{ con acarreo negativo de } 1$$

- **Ejemplo**: Resta el número binario `00111` de `10101` y muestra la resta decimal equivalente.
- **Solución**:

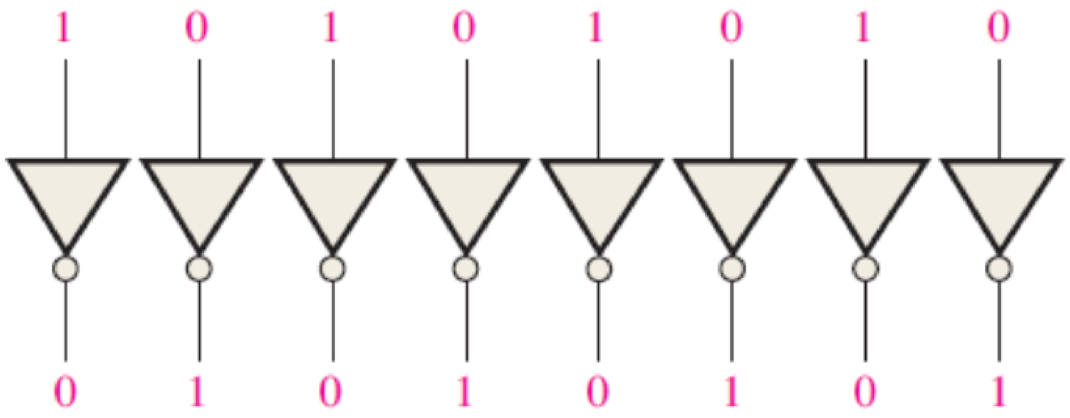
$$\begin{array}{r} {\scriptstyle 1\,1\,1} \\ \cancel{1}\cancel{0}\cancel{1}01 \\ \underline{00111} \\ 01110 \end{array} \qquad \begin{array}{r} \\ 21 \\ \underline{7} \\ = 14 \end{array}$$

Complemento a uno

- El complemento a 1 de un número binario es la inversa de los dígitos. Para formar el complemento a 1, cambia todos los 0 por 1 y todos los 1 por 0.
- Por ejemplo, el complemento a 1 de `1 0 1 0 1 0 1 0` es:

  `0 1 0 1 0 1 0 1`

- En los circuitos digitales, el complemento a 1 se forma utilizando inversores:

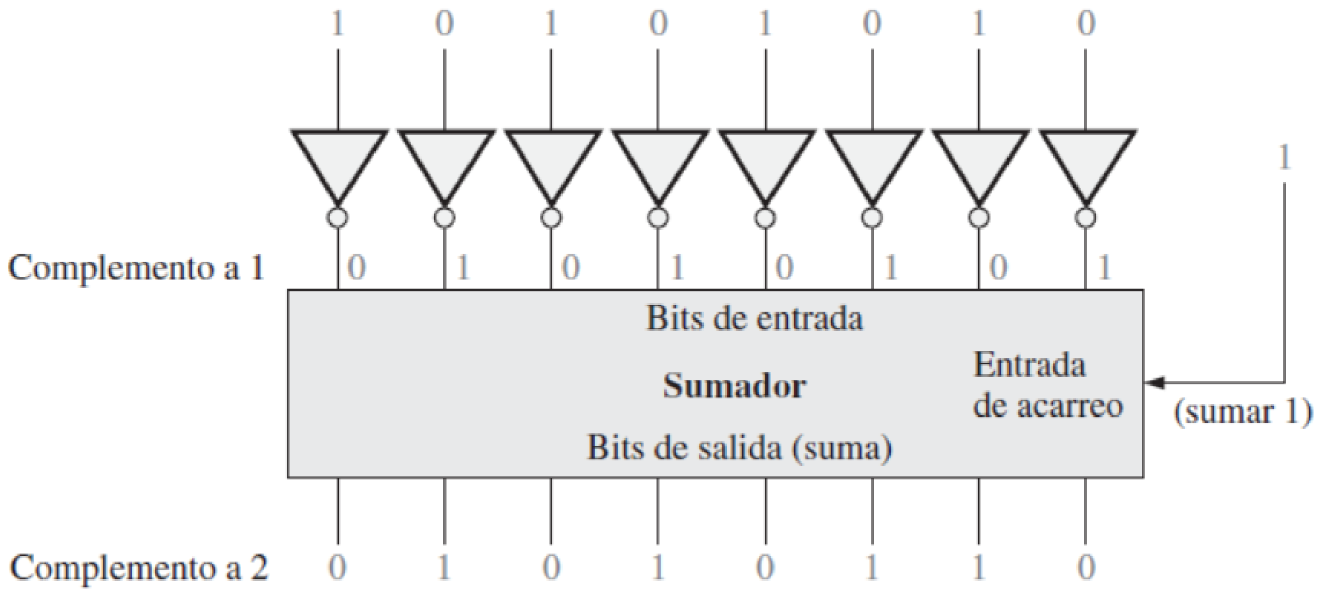


Complemento a dos

- El complemento a 2 de un número binario se obtiene sumando 1 al LSB del complemento a 1.
- Por ejemplo, el complemento a 1 de **10110010** es:

01001101	Complemento a 1
+ 1	Sumar 1
01001110	Complemento a 2

- Para formar el complemento a 2, suma 1:



Números con signo

- Existen varias formas de representar números binarios con signo.
  - Formato signo-magnitud (el método menos utilizado)
  - Formato complemento a 1
  - Formato complemento a 2 (el método más utilizado)
- En todos los casos, el MSB de un número con signo es el bit de signo, que indica si el número es positivo o negativo.
  - 0 indica número positivo
  - 1 indica número negativo
- Nota: Véase el Ej. 2-14 página 70

EJEMPLO 2.14

Expresar el número decimal –39 como un número de 8 bits en los formatos signo-magnitud, complemento a 1 y complemento a 2.

Solución En primer lugar, escribimos el número de 8 bits para +39.

00100111

En el *formato signo-magnitud,* –39 se obtiene cambiando el bit de signo a 1, y dejando los bits de magnitud como están. El número es:

10100111

En el *formato de complemento a 1,* –39 se obtiene calculando el complemento a 1 de +39 (00100111).

11011000

En el *formato de complemento a 2,* –39 se obtiene calculando el complemento a 1 de +39 (00100111), como sigue

11011000	Complemento a 1
+ 1	
11011001	Complemento a 2

Problema relacionado Expresar –19 y + 19 en los formatos signo-magnitud, complemento a 1 y complemento a 2, con ocho bits.

Números con signo

- Los computadores utilizan un complemento a 2 modificado para los números con signo. Los números positivos se almacenan en forma real (con un 0 para el bit de signo) y los negativos en forma de complemento (con un 1 para el bit de signo).
- Por ejemplo, el número positivo 58 se escribe utilizando 8 bits como `00111010` (forma real).

El valor decimal de los números negativos

- **Signo-magnitud**: suma los pesos donde hay 1's en las posiciones de bit de magnitud
- **Complemento a 1**: suma los pesos donde hay 1's y añade 1 al resultado. (dar un signo negativo al bit de signo)
- **Complemento a 2**: suma los pesos donde hay 1's dando un valor negativo al peso del bit de signo
- Ver Ej. 2-15, 2-16 y 2-17

EJEMPLO 2.15

Determinar el valor decimal del número binario con signo expresado como signo-magnitud: 10010101.

Solución Los siete bits de magnitud y sus pesos potencias de dos son los siguientes:

2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	1	0	1	0	1

Sumando los pesos de las posiciones donde hay 1s, tenemos

$$16 + 4 + 1 = 21$$

El bit de signo es 1; por tanto, el número decimal es **–21.**

Problema relacionado Determinar el valor decimal del número signo-magnitud 01110111.

EJEMPLO 2.16

Determinar los valores decimales de los números binarios con signo expresados en complemento a 1: **(a)** 00010111 **(b)** 11101000

Solución **(a)** Los bits y sus pesos según las potencias de dos para el número positivo son:

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	1	0	1	1	1

Sumando los pesos donde hay 1s,

$$16 + 4 + 2 + 1 = +\mathbf{23}$$

(b) Los bits y sus pesos según las potencia de dos para el número negativo son los siguientes. Observe que el bit de signo negativo tiene un peso de -2^7, es decir, -128.

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	1	0	0	0

Sumando los pesos de las posiciones donde hay 1s,

$$-128 + 64 + 32 + 8 = -24$$

Sumando 1 al resultado, el número decimal final es:

$$-24 + 1 = -\mathbf{23}$$

Problema relacionado Determinar el valor decimal del número en complemento a 1: 11101011.

EJEMPLO 2.17

Determinar los valores decimales de los siguientes números binarios con signo expresados en complemento a 2 :
(a) 01010110 **(b)** 10101010

Solución **(a)** Los bits y sus pesos según las potencias de dos para el número positivo son:

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	1	0	1	1	0

Sumando los pesos donde hay 1s,

$$64 + 16 + 4 + 2 = \mathbf{+86}$$

(b) Los bits y sus pesos según las potencias de dos para el número negativo son los siguientes. Observe que el bit de signo negativo tiene un peso de $-2^7 = -128$.

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	0	1	0

Sumando los pesos donde hay 1s,

$$-128 + 32 + 8 + 2 = \mathbf{+86}$$

Problema relacionado Determinar el valor decimal del número expresado en complemento a 2: 11010111.

Rango de números enteros con signo

Para hallar el número de combinaciones diferentes de n bits:

- Combinaciones totales = 2^n
- Para complemento a 2 Rango = $-(2^{n-1})$ a $+(2^{n-1}-1)$
- ej. $n=4$, Rango = $-(2^3) = -8$ a $2^3-1 = +7$

Operaciones aritméticas con números con signo

- El uso de la notación de números con signo con números negativos en forma de complemento a 2 simplifica la suma y la resta de números con signo.
- **Reglas para la suma**: Suma los dos números con signo. Descarte cualquier acarreo final. El resultado está en forma con signo.
- **Ejemplo**:

$$\begin{array}{r} 00011110 = +30 \\ 00001111 = +15 \\ \hline 00101101 = +45 \end{array} \qquad \begin{array}{r} 00001110 = +14 \\ 11101111 = -17 \\ \hline 11111101 = -3 \end{array} \qquad \begin{array}{r} 11111111 = -1 \\ 11111000 = -8 \\ \hline \cancel{1}\,11110111 = -9 \end{array}$$

Descartar acarreo

Operaciones aritméticas con números con signo

- Tenga en cuenta que si se supera el número de bits necesarios para la respuesta, se producirá un desbordamiento. Esto ocurre sólo si ambos números tienen el mismo signo. El desbordamiento se indicará mediante un bit de signo del resultado distinto del bit de signo de los dos números sumados. Dos ejemplos son:

$$\begin{array}{r}
01000000 = +128 \\
01000001 = +129 \\
\hline
10000001 = \cancel{-126}
\end{array}
\qquad
\begin{array}{r}
10000001 = -127 \\
10000001 = -127 \\
\hline
\cancel{1}00000010 = \cancel{+2}
\end{array}$$

Descartar acarreo →

Error! La respuesta es incorrecta y el bit de signo ha cambiado.

Operaciones aritméticas con números con signo

- **Reglas para la resta**: 2 complementa el sustraendo y suma los números. Descarta cualquier acarreo final. El resultado está en forma de signo.

- Repite los ejemplos hechos anteriormente, pero restando:

$$\begin{array}{rl} 00011110 & (+30) \\ -\ \underline{00001111} & -(+15) \end{array} \qquad \begin{array}{rl} 00001110 & (+14) \\ -\ \underline{11101111} & -(-17) \end{array} \qquad \begin{array}{rl} 11111111 & (-1) \\ -\ \underline{11111000} & -(-8) \end{array}$$

- Suma y resta del complemento a 2:

$$\begin{array}{rl} 00011110 & = +30 \\ \underline{11110001} & = -15 \\ \cancel{1}00001111 & = +15 \end{array} \qquad \begin{array}{rl} 00001110 & = +14 \\ \underline{00010001} & = +17 \\ 00011111 & = +31 \end{array} \qquad \begin{array}{rl} 11111111 & = -1 \\ \underline{00001000} & = +8 \\ \cancel{1}00000111 & = +7 \end{array}$$

Descartar acarreo

Descartar acarreo

- Véase también Ex. 2-20, p77

Números hexadecimales

- El hexadecimal utiliza dieciséis caracteres para representar números: los números del 0 al 9 y los caracteres alfabéticos de la A a la F.
- Contar en hexadecimal:
- ..., E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, ...

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

Conversión de binario a hexadecimal

- Un número binario grande puede convertirse fácilmente a hexadecimal agrupando los bits de 4 en 4 y escribiendo el carácter hexadecimal equivalente.
- **Ejemplo**: Exprese $1001\ 0110\ 0000\ 1110_2$ en hexadecimal:
- **Solución**: Agrupa el número binario de 4 en 4 bits empezando por la derecha. De este modo obtenemos **960E**

Conversión de hexadecimal a decimal

- El hexadecimal es un sistema numérico ponderado. Los pesos de las columnas son potencias de 16, que aumentan de derecha a izquierda.

Pesos de columna $\begin{cases} 16^3 & 16^2 & 16^1 & 16^0. \\ 4096 & 256 & 16 & 1\ . \end{cases}$

- **Ejemplo**: Expresar $1A2F_{16}$ en decimal.
- **Solución**: Empieza escribiendo los pesos de columna:

$$\begin{array}{cccc} 4096 & 256 & 16 & 1 \\ 1 & A & 2 & F_{16} \end{array}$$

$$1(4096) + 10(256) + 2(16) + 15(1) = 6703_{10}$$

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

Números octales

- El sistema octal utiliza ocho caracteres, del 0 al 7, para representar números. En octal no hay caracteres 8 ni 9.
- Un número binario puede convertirse fácilmente a octal agrupando los bits de 3 en 3 y escribiendo el carácter octal equivalente para cada grupo.
- **Ejemplo**: Expresar $1\ 001\ 011\ 000\ 001\ 110_2$ en octal:.
- **Solución**: Agrupar el número binario de 3 en 3 bits empezando por la derecha. Así pues, 113016_8

Decimal	Octal	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	10	1000
9	11	1001
10	12	1010
11	13	1011
12	14	1100
13	15	1101
14	16	1110
15	17	1111

Conversión Octal a Decimal
Conversion Decimal a Octal

- El octal también es un sistema numérico ponderado. Los pesos de las columnas son potencias de 8, que aumentan de derecha a izquierda.

$$\text{Pesos de columna}\begin{cases} 8^3 & 8^2 & 8^1 & 8^0 \,. \\ 512 & 64 & 8 & 1 \,. \end{cases}$$

- **Ejemplo**: Expresar 3702_8 en decimal:
- **Solución**: Empieza por escribir los pesos de las columnas:

$$\begin{array}{cccc} 512 & 64 & 8 & 1 \\ 3 & 7 & 0 & 2_8 \end{array}$$

$$3(512) + 7(64) + 0(8) + 2(1) = 1986_{10}$$

Decimal	Octal	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	10	1000
9	11	1001
10	12	1010
11	13	1011
12	14	1100
13	15	1101
14	16	1110
15	17	1111

BCD

- El decimal codificado en binario (BCD) es un código con pesos que se utiliza habitualmente en sistemas digitales cuando es necesario mostrar números decimales, como en las pantallas de los relojes.
- La tabla ilustra la diferencia entre binario directo y BCD. El BCD representa cada dígito decimal con un código de 4 bits. Observe que los códigos `1010` a `1111` no se utilizan en BCD.

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

BCD

- Puede pensar en BCD en términos de pesos de columna en grupos de cuatro bits. Para un número BCD de 8 bits, los pesos de columna son: `80 40 20 10 8 4 2 1`.
- **Pregunta**: ¿Cuáles son los pesos de columna para el número BCD

  `1000 0011 0101 1001`?
- **Respuesta**:

  $$8000\ 4000\ 2000\ 1000\quad 800\ 400\ 200\ 100\quad 80\ \ 40\ \ 20\ \ 10\quad 8\ \ 4\ \ 2\ \ 1$$

  Ten en cuenta que podrías sumar los pesos de columna donde hay un 1 para obtener el número decimal. En este caso:

  $$8000 + 200 + 100 + 40 + 10 + 8 + 1 = 8359_{10}$$

Suma en BCD

- **Pregunta**: Suma los siguientes números BCD

  a) `0011 + 0100`

  b) `010001010000 + 010000010111`

  c) `1001+0100`

  d) Ver más ejemplos en el libro

- **Respuesta**:

  a) 0011+0100=0111

  b) 010001010000+010000010111=100 0110 0111

  c) 1001+0100=0001 0011 (Suma 6 al número BCD inválido que es >9)

EJEMPLO 2.35

Sumar los siguientes números BCD:

(a) 0011 + 0100 **(b)** 00100011 + 00010101

(c) 10000110 + 00010011 **(d)** 010001010000 + 010000010111

Solución Se muestra la suma decimal con propósitos de comparación.

$$\textbf{(a)}\quad \begin{array}{r} 0011 \\ +\,\underline{0100} \\ \mathbf{0111} \end{array} \qquad \begin{array}{r} 3 \\ +\,\underline{4} \\ 7 \end{array} \qquad\qquad \textbf{(b)}\quad \begin{array}{r} 0010\ \ 0011 \\ +\,\underline{0001\ \ 0101} \\ \mathbf{0011\ \ 1000} \end{array} \qquad \begin{array}{r} 23 \\ \underline{+15} \\ 38 \end{array}$$

$$\textbf{(c)}\quad \begin{array}{r} 1000\ \ 0110 \\ +\,\underline{0001\ \ 0011} \\ \mathbf{1001\ \ 1001} \end{array} \qquad \begin{array}{r} 86 \\ +\,\underline{13} \\ 99 \end{array} \qquad\qquad \textbf{(d)}\quad \begin{array}{r} 0100\ \ 0101\ \ 0000 \\ +\,\underline{0100\ \ 0001\ \ 0111} \\ \mathbf{1000\ \ 0110\ \ 0111} \end{array} \qquad \begin{array}{r} 450 \\ +\,\underline{417} \\ 867 \end{array}$$

Observe que en ningún caso la suma de las columnas de 4 bits excede 9, por lo que los resultados son números BCD válidos.

Problema relacionado Sumar los números BCD: 1001000001000011 + 0000100100100101.

EJEMPLO 2.36

Sumar los siguientes números BCD

(a) 1001 + 0100 **(b)** 1001 + 1001

(c) 00010110 + 00010101 **(d)** 01100111 + 01010011

Solución Se muestra la suma decimal con propósitos de comparación.

(a)

$$\begin{array}{rlr} 1001 & & 9 \\ \underline{+0100} & & \underline{+4} \\ 1101 & \text{Número BCD no válido (>9)} & 13 \\ \underline{+0110} & \text{Sumar 6} & \\ \underbrace{\underline{\mathbf{0001}}}_{\substack{\downarrow \\ 1}}\ \ \underbrace{\underline{\mathbf{0011}}}_{\substack{\downarrow \\ 3}} & \text{Número BCD válido} & \end{array}$$

(b)

$$\begin{array}{rlr} 1001 & & 9 \\ \underline{+1001} & & \underline{+\ 9} \\ 1\ \ \ 0010 & \text{No válido debido al acarreo} & 18 \\ \underline{+\ 0110} & \text{Sumar 6} & \\ \underbrace{\underline{\mathbf{0001}}}_{\substack{\downarrow \\ 1}}\ \ \ \underbrace{\underline{\mathbf{1000}}}_{\substack{\downarrow \\ 8}} & \text{Número BCD válido} & \end{array}$$

(c)	0001	0110			16
	+0001	0101			+ 15
	0010	1011		El grupo de la derecha no es válido (>9), el grupo de la izquierda es válido.	31
		+0110		Sumar 6 al código no válido. Sumar el acarreo, 0001, al siguiente grupo.	
	$\underbrace{\mathbf{0011}}_{\downarrow \atop 3}$	$\underbrace{\mathbf{0001}}_{\downarrow \atop 1}$		Número BCD válido	

(d)		0110	0111		67
		+ 0101	0011		+53
		1011	1010	Ambos grupos no son válidos (>9)	120
		+ 0110	+0110	Sumar 6 a ambos grupos	
	$\underbrace{\mathbf{0001}}_{\downarrow \atop 1}$	$\underbrace{\mathbf{0010}}_{\downarrow \atop 2}$	$\underbrace{\mathbf{0000}}_{\downarrow \atop 0}$	Número BCD válido	

Problema relacionado Sumar los números BCD: 01001000 + 00110100.

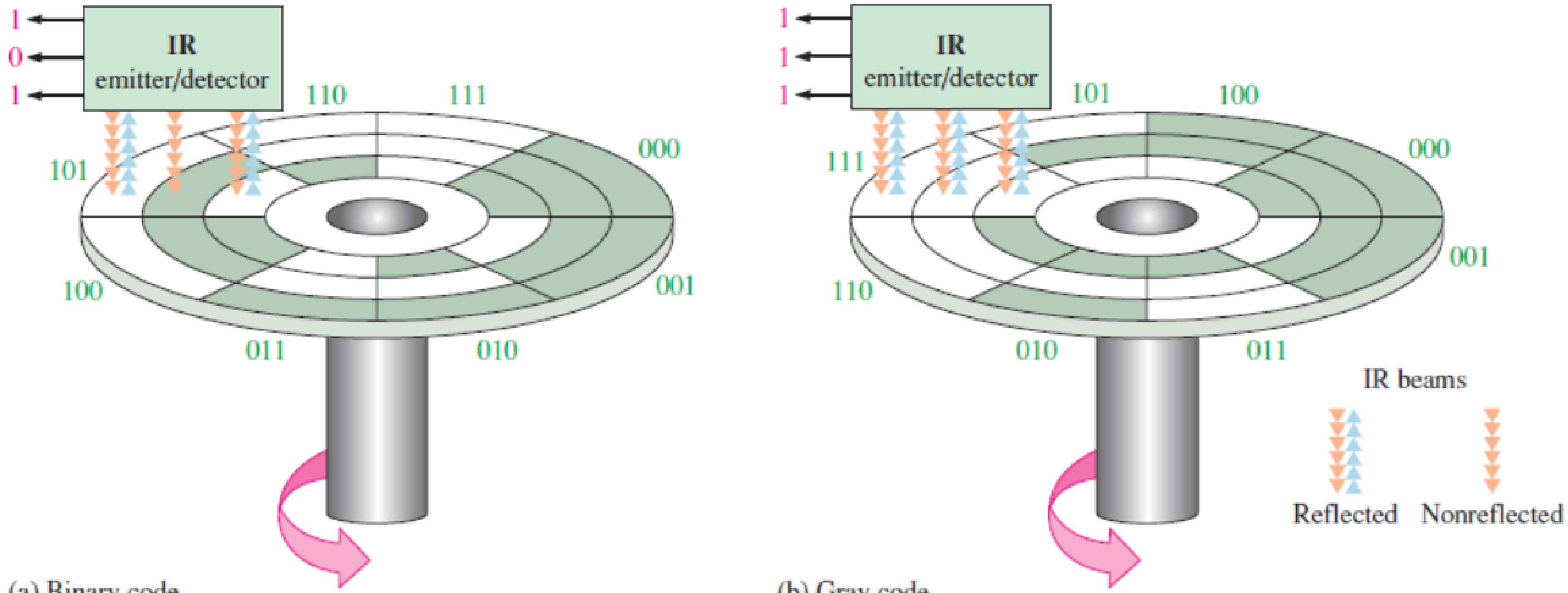
Código Gray

- El código gris es un código no ponderado que tiene un único bit de cambio entre una palabra de código y la siguiente de una secuencia. El código gris se utiliza para evitar problemas en sistemas en los que puede producirse un error si cambia más de un bit a la vez.
- Leer el libro para obtener más información sobre la conversión de binario a gris y viceversa.

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

Código Gray

- Un codificador de eje es una aplicación típica. Se utilizan tres emisores/detectores IR para codificar la posición del eje. El codificador de la izquierda utiliza código binario y pueden cambiar tres bits a la vez, lo que puede dar lugar a errores. El codificador de la derecha utiliza código gris y sólo cambia 1 bit, eliminando posibles errores.



(a) Binary code (b) Gray code

ASCII

- ASCII es un código para caracteres alfanuméricos y caracteres de control. En su forma original, ASCII codificaba 128 caracteres y símbolos utilizando 7 bits. Los primeros 32 caracteres son caracteres de control, que se basan en requisitos de teletipo obsoletos, por lo que estos caracteres suelen asignarse a otras funciones en el uso moderno.
- En 1981, IBM introdujo el ASCII extendido, que es un código de 8 bits y aumentó el conjunto de caracteres a 256. Se han introducido otros conjuntos extendidos (como Unicode) para manejar caracteres en idiomas distintos del inglés.

Método de paridad

- El método de paridad es un método de detección de errores para errores de transmisión simples que implican un bit (o un número impar de bits). Un bit de paridad es un bit "extra" que se añade a un grupo de bits para forzar que el número de 1's sea par (paridad par) o impar (paridad impar). Puede añadirse al código al principio o al final, dependiendo del diseño del sistema.
- **Ejemplo**: El carácter ASCII para "a" es 1100001 y para "A" es 1000001. ¿Cuál es el bit correcto que hay que añadir para que ambos tengan paridad impar?
- **Solución**: La "a" ASCII tiene un número impar de bits que son iguales a 1; por tanto, el bit de paridad es 0. La "A" ASCII tiene un número par de bits que son iguales a 1; por tanto, el bit de paridad es 1.