

Sistema Hexadecimal, posee una raíz de 16. Se denomina *sistema de numeración en base 16*. Utiliza los símbolos 0–9, A, B, C, D, E y F como se muestra en la columna hexadecimal de la tabla de la Figura 1. La letra A se utiliza para el 10, la B para el 11, la C para el 12, la D para el 13, la E para el 14, y la F para el 15. La ventaja del sistema hexadecimal es que es útil para convertir directamente números binarios de 4 bits. Observar en la sección sombreada de la Figura 1 que cada número binario de cuatro bits, del 0000 al 1111, puede ser representado por un único dígito hexadecimal.

Figura 1. Contar en los sistemas de numeración decimal, binario y hexadecimal.

Decimal	Binario	Hexadecimal	Decimal	Binario	Hexadecimal
0	0000	0	16	10000	10
1	0001	1	17	10001	11
2	0010	2	18	10010	12
3	0011	3	19	10011	13
4	0100	4	20	10100	14
5	0101	5	21	10101	15
6	0110	6	22	10110	16
7	0111	7	23	10111	17
8	1000	8	24	11000	18
9	1001	9	25	11001	19
10	1010	A	26	11010	1 ^a
11	1011	B	27	11011	1B
12	1100	C	28	11100	1C
13	1101	D	29	11101	1D
14	1110	E	30	11110	1E
15	1111	F	31	11111	1F

Observar la línea que corresponde al 16 en la columna decimal de la Figura 1. El hexadecimal equivalente es 10. Esto muestra que el sistema de numeración hexadecimal utiliza la idea de valor por posición. El 1 (en 1016) significa 16 unidades, mientras que el 0 significa cero unidades.

a) Conversión hexadecimal– decimal: convertir el número hexadecimal 2B6 en decimal. La Figura 1.1 muestra el proceso familiar. El 2 está en la posición del 256, por tanto $256 \times 2 = 512$, que se escribe en la línea decimal. El dígito hexadecimal B aparece en la columna del 16. Observar en la Figura 1.1 que el hexadecimal B corresponde al decimal 11. Esto significa que hay once 16 (16 $\times$ 11), dando 176. El 176 se suma al total decimal en la parte inferior de la Figura 1.1. La columna del 1 muestra seis 1. El 6 se suma a la línea decimal. Los valores decimales son sumados ($512 + 176 + 6 = 694$), dando 694. La Figura 1.1 muestra que 2B6 es igual a 694.

- Conversión hexadecimal – decimal. Figura 1.1.

Potencias de 16	16 ²	16 ¹	16 ⁰
Valor de la posición	256	16	1

Número Hexadecimal 2 B 6

256

2

512
16

11

176
1

6

6

Decimal $512 + 176 + 6 = 69410$

b) Conversión hexadecimal fraccionario– decimal: convertir el número hexadecimal A3F.C en su decimal equivalente. La Figura 1.2 detalla este problema. Primero considerar la columna del 256. El dígito hexadecimal A significa que 256 debe ser multiplicado por 10, dando un producto de 2560. El número hexadecimal muestra que contiene tres 16, y por tanto 16

$3 \times 48 = 144$ que se suma a la línea decimal. La columna del 1 contiene el dígito hexadecimal F, que significa 15 = 15. El 15 se suma a la línea decimal. La columna del 0.0625 contiene el dígito hexadecimal C, que significa 12

$0.0625 \times 12 = 0.75$. El 0.75 se suma a la línea decimal. Al sumar los contenidos de la línea decimal ($2560 + 144 + 15 + 0.75 = 2623.75$) se obtiene el número decimal 2623.75. La Figura 1.2 convierte A3F.C16 en 2623.7510.

- Conversión hexadecimal fraccionario – decimal. Figura 1.2.

Potencias de 16	16 ²	16 ¹	16 ⁰	1/16 ¹
Valor de la posición	256	16	1	.0625

Número hexadecimal A 3 F . C

256

10

2560
16

3

48
1

15

15
,0625

12

0,75

Decimal $2560 + 48 + 15 + 0.75 = 2623.7510$

c) Conversión decimal– hexadecimal: convertir el número decimal 45 en su equivalente hexadecimal. La Figura 1.3 detalla el proceso familiar de división repetida por 16. El número decimal 45 se divide primero por 16, dando un 2 de cociente con resto 13. El resto 13 (D en hexadecimal) se convierte en LSD del número hexadecimal. El cociente (2) se transfiere a la posición del dividendo y se divide por 16. Así se obtiene un cociente de 0 con un resto de 2. El 2 se convierte en el siguiente dígito del número hexadecimal. El proceso finaliza cuando la parte entera del cociente es 0. El proceso de la Figura 1.3 convierte el número decimal 45 en el número hexadecimal 2D.

- Conversión decimal – hexadecimal. Figura 1.3.

$$45 \div 16 = 2$$

resto de 13

$$2 \div 16 = 0$$

resto de 2

$$45_{10} = 2D_{16}$$

d) Conversión de decimal fraccionario– hexadecimal: convertir el número decimal 250.25 a hexadecimal. La conversión debe hacerse utilizando dos procesos como muestra la Figura 1.4. La parte entera del número decimal (250) se convierte en hexadecimal utilizando el proceso de división repetida por 16. Los restos de 10 (A en hexadecimal) y 15 (F en hexadecimal) forman el número hexadecimal entero FA. La parte fraccionaria del 250.25 se multiplica por 16 (0.25

16). El resultado es 4.00. El entero 4 es transferido a la posición mostrada en la Figura 1.4. La conversión completa muestra que el número decimal 250.25 es igual al hexadecimal FA.4.

- d) Conversión decimal fraccionario– hexadecimal. Figura 1.4.

$$250 \div 16 = 15$$

resto de 10

$$15 \div 16 = 0$$

resto de 15

$$250.25_{10} = FA.4_{16}$$

$$0.25 \div 16 = 4.00$$

$$0.00 \div 16 = 0.00$$

e) Conversión hexadecimal– binario: la principal ventaja del sistema hexadecimal es su fácil conversión al binario. La Figura 1.5 muestra el número hexadecimal 3B9 convertido a binario. Observar que cada dígito hexadecimal forma un grupo de cuatro dígitos binarios o bits. Los grupos de bits son entonces combinados para formar el número binario. En este caso 3B9₁₆ es igual a 11101110012.

- e) Conversión hexadecimal– binario. Figura 1.5.

$$3B9_{16}$$

$$3B9_{16} = 1110111001_2$$

0011 1011 1001

f) Conversión hexadecimal fraccionario– binario: esta conversión se detalla en la Figura 1.6. De nuevo, cada dígito hexadecimal forma un grupo de cuatro bits en el número binario. El punto hexadecimal es bajado para formar el punto binario. El número hexadecimal 47.FE se convierte en el número binario 1000111.1111111. Es claro que los números hexadecimales, debido a su compactación, son mucho más fáciles de escribir que largas cadenas de 1 y 0 en binario. El sistema hexadecimal puede considerarse como un método abreviado de escribir números binarios.

f) Conversión hexadecimal fraccionario– binario. Figura 1.6.

4 7 . F E

47.FE₁₆= 1000111.1111111₂

0100 0111 . 1111 1110

g) Conversión binario– hexadecimal: la Figura 1.7 muestra la conversión del número binario 101010000101 a hexadecimal. Primero dividir el número binario en grupos de cuatro bits *comenzando en el punto binario*. Cada grupo de cuatro bits se traduce a su dígito hexadecimal equivalente. La Figura 1.7 muestra que el número binario 101010000101 es igual al hexadecimal A85.

g) Conversión de binario– hexadecimal. Figura 1.7.

1010 1000 0101.

101010000101₂= A85₁₆

A 8 5

h) Conversión binario fraccionario– hexadecimal: otra conversión de binario a hexadecimal se ilustra en la Figura 1.8. Aquí el número binario 10010.011011 se traduce a hexadecimal. Primero el número binario se divide en grupos de cuatro bits comenzando en el punto binario. Tres 0 se añaden al grupo de más a la izquierda, formando 0001. Dos 0 se añaden al grupo de más a la derecha, formando 1100. Cada grupo tiene ahora 4 bits y se traduce a un dígito hexadecimal como muestra la figura 1.8. El número binario 10010.01101 es igual a 12.6 C₁₆.

h) Conversión binario fraccionario– hexadecimal. Figura 1.8.

0001 0010 . 0110 1100

10010.011011₂= 12.6C₁₆

1 2 . 6 C

En cambio, el **Sistema Octal** hace uso de grupo de 3 bits, de acuerdo con la siguiente tabla, esta es meramente ilustrativa, no está en toda su extensión:

Tabla del Sistema Octal. Figura 1.9

Binario	Dígito Octal
0	0

001	1
010	2
011	3
100	4
101	5
110	6
111	7

Cada símbolo en octal representa la equivalencia numérica de un grupo de 3 dígitos binarios, y los ocho símbolos forman un sistema numérico en base 8. En este caso, es necesario un teclado con ocho teclas para guardar los datos, pero solo se necesita presionar las teclas un tercio de las veces que se hiciera en un binario.

Ejemplo 1. Expresa el número 247 en octal como un binario y uno decimal.

El número en octal es posicional, con el dígito de menor orden (el más a la derecha) ponderado o con el peso de $8^0 = 1$ y el mayor orden $8^2 = 64$. De este modo $247 = 2 \times 64 + 4 \times 8 + 7 \times 1 = 167_{10}$.

Refiriéndose a la tabla anterior, la cual te indica que la conversión a binario se puede llevar a cabo fácilmente agrupando:

247

010 100 111

Esta conversión se puede verificar con facilidad determinando el equivalente decimal del número binario resultante, 10100111. Se observa que los ceros se pueden eliminar conservando la posición de los unos. El bit más significativo está en la octava posición y por lo tanto tiene un peso de $2^7 = 128$. De esta manera, $10100111 = 128 + 32 + 4 + 2 + 1 = 167$.

Código decimal codificado binario (BCD), éste realiza la conversión a decimal de forma mucho más fácil. La Figura 2 muestra el código BCD de cuatro bits para los dígitos decimales del 0 al 9. Observar que el código BCD es un código con peso. El bit más significativo tiene un peso de 8, y el menos significativo de 1. Este código se conoce más precisamente como *código BCD 8421*. La parte 8421 del nombre da los pesos de cada posición en el código de 4 bits. Hay otros códigos BCD que asignan otros pesos a las cuatro posiciones. Como el código BCD 8421 es más popular, es costumbre referenciarlo como código BCD.

Figura 2. El Código BCD 8421.

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) Conversión decimal– BCD: convertir el número decimal 150 en BCD. La Figura 2.1 muestra la técnica. Cada dígito decimal se convierte en su equivalente BCD de 4 bits. El número decimal 150 es entonces igual al número BCD 000101010000.

- Conversión decimal– BCD. Figura 2.1.

Decimal 1 5 0

BCD 0001 0101 0000

b) Conversión decimal fraccionario– BCD: en la Figura 2.2 se ilustra la conversión, en donde cada dígito decimal se convierte a su equivalente BCD. El punto decimal se convierte en punto binario. La Figura 2.2 muestra que el decimal 32.84 es igual al número BCD 00110010.10000100.

(b) Conversión decimal fraccionario– BCD. Figura 2.2.

Decimal 3 2 . 8 4

BCD 0011 0010 . 1000 0100

c) Conversión BCD– decimal: la Figura 2.3 muestra la técnica. El número BCD 10010110 se divide primero en grupos de 4 bits comenzando en el punto binario. Cada grupo de 4 bits se convierte entonces en su dígito decimal equivalente, que se anota debajo. El número BCD 10010110 es igual, entonces, al decimal 96.

(c) Conversión BCD– decimal. Figura 2.3.

BCD 1001 0110.

Decimal 9 6 .

d) Conversión BCD fraccionario– decimal: convertir el número fraccionario BCD 01110001.00001000 en su equivalente decimal. La Figura 2.4 muestra el procedimiento. El número BCD se divide primero en grupos de 4 bits, comenzando en el punto binario. Cada grupo de 4 bits se convierte entonces en su equivalente decimal. El punto binario se convierte en el punto decimal del número decimal. La Figura 2.4 muestra el número BCD 01110001.00001000 convertido en su equivalente decimal 71.08.

(d) Conversión BCD fraccionario– decimal. Figura 2.4.

BCD 0111 0001 . 0000 1000

Decimal 7 1 . 0 8

Código Gray, es un código binario sin peso. El código de Gray no es un código tipo BCD. La Figura 3 compara números en código Gray con sus equivalentes binarios y decimales. Observar cuidadosamente el código de Gray: cada incremento en la cuenta va acompañado *solamente por el cambio de estado de 1 bit*. Observar el cambio de la línea del decimal 7 a la del decimal 8. En binario los 4 bits cambian de estado (del 0111 al 1000). En la misma línea, el código de Gray ha cambiado de estado solamente el bit de la izquierda (0100 a 1100). Este cambio de un solo bit, en el grupo del código, a causa del incremento es importante en algunas aplicaciones de electrónica digital.

Figura 3. El código de Gray.

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

a) Conversión código de Gray– binario: convertir el número en código de Gray a su equivalente binario. La Figura 3.1 detalla el procedimiento. Primero el bit de la izquierda (1) es transferido a la línea binaria para formar el bit del 8. El bit del 8 del número binario es transferido (véase flecha) al siguiente bit del número en código de Gray, y se suman los dos. La suma es 1 ($1 + 0 = 1$), que se escribe en la posición del bit del 4 del número binario. El bit del 4 (1) se suma entonces al siguiente bit del número en código de Gray. La suma es 1 ($1 + 0 = 1$). Este 1 se escribe en la posición del 2 del número binario. El bit del 2 binario (1) se suma al bit de la derecha del número en código de Gray. La suma es 0 ($1 + 1 = 10$) ya que se desprecia el arrastre. Este 0 se escribe en la posición del 1 del número binario. La Figura 3.1 muestra el número en código de Gray 1001 convertido en su número binario equivalente 1110. Esta conversión puede verificarse observando la línea del decimal 14 en la Figura 3.

- Conversión código Gray– binario. Figura 3.1.

1 1 1

Código Gray 1 0 0 1

suma suma suma

Binario 1 1 1 0

Ventajas del código Gray

- 1) Es muy conveniente para fines de medición.
- 2) Los dos dígitos más a la izquierda son los mismos tanto en el sistema Binario como en el de Gray.
- 3) Léase el número Binario de izquierda a derecha. Un cambio (de 0 a 1 ó de 1 a 0) genera un 1 en el número codificado en Gray; de otro modo, se genera un 0.
- 4) Léase el número Gray de izquierda a derecha. Un 1 significa que el próximo dígito binario debe cambiar; un 0 significa que el próximo dígito binario es idéntico al de su izquierda.
- 5) En comparación con los otros sistemas, la codificación de Gray no genera grandes errores.

Código ASCII, los bits también pueden ser codificados para que representen letras del alfabeto, números y símbolos de puntuación. Un código de 7 bits es el *American Standard Code for Information Interchange* (Código Americano para Intercambio de Información) (ASCII), mostrado en la Figura 4. Observar que la letra

A se representa por 100 0001, mientras que la B en código ASCII es 100 0010. El código ASCII es muy utilizado en los pequeños sistemas computadores para traducir los caracteres del teclado al lenguaje de la computadora. La tabla de la Figura 4 no es una lista completa de todas las combinaciones del código ASCII.

Los códigos que pueden representar letras y números se denominan *códigos alfanuméricos*. El código alfanumérico ASCII es un código moderno para introducir y sacar información de las microcomputadoras. ASCII es utilizado para realizar interfaces en teclados de computadoras, impresoras y pantallas de vídeo. ASCII ha llegado a ser el código estándar de entrada/salida para las microcomputadoras.

Figura 4. Conjunto representativo de código ASCII.

CARÁCTER	ASCCII		HEXADECIMAL		CARÁCTER	ASCCII		HEXADECIMAL
A	100	0001	41		T	101	0100	54
B	100	0010	42		U	101	0101	55
C	100	0011	43		V	101	0110	56
D	100	0100	44		W	101	0111	57
E	100	0101	45		X	101	1000	58
F	100	0110	46		Y	101	1001	59
G	100	0111	47		Z	101	1010	5A
H	100	1000	48		ESPACIO	010	0000	20
I	100	1001	49		(	010	1000	28
J	100	1010	4 ^a		)	010	1001	29
K	100	1011	4B		+	010	1011	2B
L	100	1100	4C		0	011	0000	30
M	100	1101	4D		1	011	0001	31
N	100	1110	4E		2	011	0010	32
O	100	1111	4F		3	011	0011	33
P	101	0000	50		4	011	0100	34
Q	101	0001	51		5	011	0101	35
R	101	0010	52		6	011	0110	36
S	101	0011	53		7	011	0111	37
					8	011	1000	38
					9	011	1001	39

Conjunto de caracteres ASCII, en informática, un código estándar de 7 bits para la representación de caracteres: letras, dígitos, signos de puntuación e instrucciones de control; con valores binarios. El intervalo de los valores del código es de 0 a 127. Aunque ASCII carece tanto de los acentos como de los caracteres especiales utilizados en diversos idiomas europeos, y no es capaz de representar caracteres en los alfabetos no latinos, reviste importancia por ser el sistema de codificación de caracteres más internacional. Muchos conjuntos de caracteres latinos no ingleses son extensiones o modificaciones del sistema de codificación ASCII. La mayoría de los sistemas de ordenadores o computadoras personales utilizan un código ASCII extendido o modificado de 8 bits, con 128 caracteres adicionales para símbolos especiales, letras y signos de puntuación de diversos idiomas, y símbolos gráficos.

ASCII incluye 256 códigos divididos en dos conjuntos, estándar y extendido, de 128 cada uno. Estos conjuntos representan todas las combinaciones posibles de 7 u 8 bits, siendo esta última el número de bits en un byte. El conjunto ASCII básico, o estándar, utiliza 7 bits para cada código, lo que da como resultado 128 códigos de caracteres desde 0 hasta 127 (00H hasta 7FH hexadecimal). El conjunto ASCII extendido utiliza 8

bits para cada código, dando como resultado 128 códigos adicionales, numerados desde el 128 hasta el 255 (80H hasta FFH extendido).

Además, se tiene que el conjunto de caracteres ASCII estándar es universal en el *hardware* y el *software* de los microordenadores, mientras que los caracteres ASCII extendido pueden interpretarse correctamente sólo si un programa, computadora o impresora han sido diseñados para ello.

Método de Paridad, en el procesamiento electrónico de datos es imposible evitar la presencia de ruido aleatorio, con la consecuente probabilidad de la introducción de errores. Una de las formas más comunes de enfrentar esta situación es agregar dígitos redundantes llamados *bits de paridad*. Cada bit de paridad está asociado con un grupo de bits de datos y se selecciona para hacer que el número total de 1s en el grupo ya sea par (*paridad par*) o impar (*paridad impar*). En cualquier momento posterior, el número 1s en el grupo completo se puede contar y compararse con la paridad originalmente creada. Esta técnica conocida como verificación de paridad simple, siempre detectará un número impar de errores entres los miembros del grupo, y es muy efectiva en un ambiente donde prevalecen los errores aislados.

Como un ejemplo, considérense caracteres DCB de 4 bits codificado con paridad impar. El dígito de paridad es el bit que está en la extrema derecha.

Figura 5. Método de Paridad.

Decimal	DCB	Bit de Paridad Agregado
0	0000	00001
1	0001	00010
2	0010	00100
3	0011	00111
4	0100	01000
5	0101	01011
6	0110	01101
7	0111	01110
8	1000	10000
9	1001	10001

Si se procesara un 5 (0101 en DCB), se almacenará el grupo 01011 con verificación de paridad en DCB. Si más adelante se obtiene el grupo como 11011 (error en el primer bit), la paridad será entonces par y se detecta el error.

Corrección de Errores, la paridad también se puede utilizar en combinaciones especiales que permiten tanto la corrección como la detención de ciertas clases de errores.

Un ejemplo del llamado código Hamming se muestra en la figura a continuación.

		D1	
		D2	
D1		D3	
D2		D4	

D3	LÓGICA DIGITAL		
D4			
		P1	
		P2	
		P3	

En este caso se usan 4 bits de datos (D1, D2, D3, D4) para crear tres bits de paridad adicionales (P1, P2, P3) los cuales se agregan a los cuatros originales para formar siete dígitos binarios que se almacenan o lo transmiten como grupo.

El procesamiento de código Hamming se realiza mediante un circuito lógico el cual lleva a cabo el procedimiento definido por la siguiente tabla.

	D4	D3	D2	P3	D1	P2	P1
grupo de paridad # 1	X		X		X		X
grupo de paridad # 2	X	X			X	X	
grupo de paridad # 3	X	X	X	X			

Se forman tres grupos de paridad a partir de diferentes combinaciones de tres dígitos de datos y uno sólo de paridad, el cual se genera para cada uno de esos grupos. La pertenencia a un grupo de paridad se indica con una X en la columna del dígito apropiado. El conocimiento del patrón de falla de paridad permite la identificación del dígito con error.

Ejemplo 1: considérese el caso en el que los dígitos de datos D4, D3, D2, D1, son 1011. Refiriéndose a la tabla anterior, se ve que para el grupo de paridad 2, la paridad impar requiere que P2 sea 1 ya que D1 y D4 son miembros del grupo, D3 es 0, y D2 (que no es miembro) no se toma en cuenta. Las reglas de generación de paridad impar combinada con algoritmo definido en la tabla presentada anteriormente, producirá los bits de paridad P3, P2, P1= 110. El código de Hamming se transmite con la secuencia D4, D3, D2, P3, D1, P2, P1, de tal manera que en este ejemplo la salida del circuito lógico se transmite como 1011110.

Ahora supóngase que el segundo dígito más a la izquierda tiene un error, esto es, la secuencia recibida es 1111110. Cuando se obtiene, los grupos de paridad se verificaran individualmente y al referirse a la tabla, se muestra que los grupos de paridad 2 y 3 fallarán (muestran paridad impar). La única situación en los grupos 2 y 3 fallan y el grupo 1 no es cuando D3 tiene un error. Su valor erróneo es 1 y, por lo tanto, se debe corregir a 0. Si un bit de paridad tiene un error, solamente fallará el grupo de paridad al que pertenece, permitiendo su identificación y corrección.