

SISTEMAS NUMÉRICOS Y CÓDIGOS

Las magnitudes físicas, como el voltaje o la corriente eléctricas pueden transportar información analógica, como la temperatura o la velocidad, ya que pueden variar de manera continua. Al contrario, la información digital o lógica es fundamentalmente discontinua y debe ser soportada por un sistema de n estados estables o discretos, correspondiendo cada estado a un valor de información llamado dígito.

Los sistemas digitales son sistemas de dos estados, de ahí el nombre de binario, y la información digital es por tanto información binaria. Las informaciones complejas se llevan a un conjunto de informaciones binarias más elementales por el proceso denominado codificación.

1.1 SISTEMAS NUMÉRICOS

El sistema numérico decimal es el adoptado mundialmente, pero no es el único que ha sido utilizado; recordemos así el sistema de números romanos. A continuación estudiaremos las cualidades de otros sistemas numéricos.

1.1.1 SISTEMAS NUMÉRICOS POSICIONALES

El concepto de número está principalmente relacionado al de cantidad de elementos que contiene un conjunto, esto se da sobre todo en los números naturales sobre los cuales trabajaremos. El objetivo de los sistemas de numeración es la representación adecuada y fácil de utilizar de una colección de elementos.

Podemos definir un sistema de numeración como las reglas que permiten, con una cantidad finita de símbolos, representar un número natural cualquiera.

La propiedad más importante de los sistemas de numeración posicional es que un mismo dígito o cifra tiene un valor distinto según sea su posición en el número. Por ejemplo, el sistema numérico utilizado cotidianamente, el sistema decimal, está formado por diez símbolos distintos con los cuales podemos representar cualquier cantidad.

En este sistema se puede representar cualquier cantidad, teniendo en cuenta la posición de los dígitos o símbolos. Por ejemplo el número 245 representa 2 centenas (2×10^2); 4 decenas (4×10^1) y 5 unidades (5×10^0) en total.

Pendiente Revisión.

Se puede inferir que el valor real de cada dígito o valor relativo, equivale a su valor absoluto multiplicado por una potencia de diez denominada peso, el cual está asociado con su posición en el número, y que la cantidad representada por las cifras es la suma de los valores relativos.

Como podemos observar, la base de las potencias del sistema numérico es el número 10, por lo que también se le llama sistema de base diez.

Siguiendo este orden de ideas tenemos:

$$(245,1) = (2 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 1 \times 10^{-1}).$$

1.1.2 NOMENCLATURA

La base o raíz de un sistema numérico es la cantidad de diferentes dígitos que pueden estar en cualquier posición de un número N.

En general un número N perteneciente a un sistema numérico posicional de base B se denotará de la siguiente forma:

$$(N)_B = (...d_2 d_1 d_0, d_{-1} d_{-2} d_{-3}...)_B$$

Donde d_i corresponde a cualesquiera de los dígitos pertenecientes al sistema de base B. El sistema numérico poseerá B dígitos consecutivos desde el cero (0) hasta el dígito B -1 (ambos inclusive).

El número $(N)_B$ representa una cantidad con parte entera y parte fraccional, cuyo equivalente en el sistema decimal está dado por la expresión en forma de polinomio:

$$(N)_B = (...d_2 B^2 + d_1 B^1 + d_0 B^0 + d_{-1} B^{-1} + d_{-2} B^{-2} ...)_{10}.$$

Es decir, cada dígito d se multiplica por la base del sistema elevada a la potencia correspondiente a su posición (B^i).

1.1.3 PESOS DE LAS POSICIONES

En la representación en polinomio, observamos que a cada uno de los dígitos le corresponde, de acuerdo a su posición, un valor o peso por el cual se multiplica el dígito para dar el valor relativo en el sistema decimal.

En la tabla 1-1 se muestran los pesos de algunos sistemas numéricos para números de tres dígitos. El dígito más a la derecha (d_{-3}) es el dígito menos significativo "LSD" (menor peso), mientras que el dígito más a la izquierda (d_2) es el dígito más significativo "MSD" (mayor peso).

Pendiente Revisión.

DIGITOS:	PESOS					
	d_2	d_1	d_0	d_{-1}	d_{-2}	d_{-3}
BASE 10	100	10	1	0,1	0,01	0,001
BASE 3	9	3	1	0,333	0,111	0,037
BASE 2	4	2	1	0,5	0,25	0,125

Tabla 1-1 Pesos de sistemas numéricos

En general un número entero de n dígitos en base B , puede cubrir o representar hasta una cantidad máxima de $(B^n - 1)_{10}$ unidades.

EJEMPLO 1-1

Calcule el máximo valor en el sistema decimal que se puede representar con dos (2) dígitos enteros en el sistema numérico de base 8.

Solución:

Tal como se expuso, los dígitos de este sistema van desde el cero (0) hasta el siete 7. El mayor número de dos dígitos en este sistema es $(77)_8$. Utilizando la representación en polinomio encontramos el valor en el sistema decimal:

$$(77)_8 = (7 \times 8 + 7 \times 1)_{10} = (56 + 7) = (63)_{10} \text{ o también:}$$

$$(77)_8 = (100)_8 - (1)_8 = (64 - 1)_{10} = (63)_{10}$$

Observe que este valor es una unidad menor que el valor del peso correspondiente a la tercera posición.

1.1.4 SISTEMA NUMÉRICO DECIMAL

Se ha utilizado el sistema numérico decimal para introducir los conceptos generales sobre los sistemas numéricos posicionales. Podemos ahora resumir las características del sistema decimal así:

Base o Raíz:

10 (Diez).

Dígitos :

0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Pendiente Revisión.

Conteo o representación de cantidades enteras con dos dígitos:

00, 01, 02,....., 09, 10, 12,...., 19, 20, 21,, 99.

Representación en polinomio:

$$(N)_{10} = (...d_2 10^2 + d_1 10^1 + d_0 10^0 + d_{-1} 10^{-1} + d_{-2} 10^{-2} ...)_{10}.$$

Peso de cada posición:

$$(N)_{10} = (...10^2 10^1 10^0) \text{ Parte entera}$$

$$(N)_{10} = (10^{-1} 10^{-2} 10^{-3} ...) \text{ Parte fraccional}$$

$$(N)_{10} = (...10^2 10^1 10^0, 10^{-1} 10^{-2} ...) \text{ Partes entera y fraccional}$$

1.1.5 SISTEMA NUMÉRICO OCTAL

Como se verá posteriormente, el sistema numérico octal proporciona un método conveniente para la representación de los códigos y números binarios utilizados en los sistemas digitales. Podemos concretar las características del sistema octal como:

Base o Raíz:

8 (Ocho).

Dígitos :

0, 1, 2, 3, 4, 5, 6, 7.

Conteo o representación con dos dígitos:

00, 01, 02,....., 07, 10, 11, 12,...., 17, 20, 21,,27, 30, ..., 77.

Representación en Polinomio:

$$(N)_8 = (...d_2 8^2 + d_1 8^1 + d_0 8^0 + d_{-1} 8^{-1} + d_{-2} 8^{-2} ...)_{10}.$$

Peso de cada posición:

$$(...8^2 8^1 8^0) \text{ Parte entera.}$$

$$(8^{-1} 8^{-2} 8^{-2}) \text{ Parte fraccional.}$$

$$(N)_8 = (...64 8^1, 1/8 1/64...) \text{ Partes entera y fraccional.}$$

EJEMPLO 1-2

Determine el valor decimal del número octal 102.

Solución:

Utilizando la representación en polinomio:

$$(N)_8 = (...1 \times 8^2 + 0 \times 8^1 + 2 \times 8^0 + 0 \times 8^{-1} + 0 \times 8^{-2} ...)_{10}$$

$$(N)_8 = (66)_{10}$$

Pendiente Revisión.

Utilizando los pesos:

PESOS	64	8	1
DIGITOS	1	0	2
VALORES RELATIVOS	64	0	2
CANTIDAD	66		

EJEMPLO 1-3

Determine el valor decimal del número octal 213,71.

Solución:

Utilizando la representación en polinomio:

$$(N)_8 = (...2 \times 8^2 + 1 \times 8^1 + 3 \times 8^0 + 7 \times 8^{-1} + 1 \times 8^{-2} ...)_{10}$$

$$(N)_{8.} = (139 + 57/64)_{10} = (139,890625)_{10}$$

Utilizando los pesos:

PESOS	64	8	1	1/8	1/64
DIGITOS	2	1	3	7	1
VALOR DECIMAL	128	8	3	7/8	1/64
CANTIDAD	139			+ 57/64	

EJEMPLO 1-4

Determine en el sistema decimal el mínimo valor o diferencia entre dos números octales consecutivos consistentes de dos dígitos enteros y dos para la parte fraccional.

Solución:

Si el número menor es él $(01, 00)_8$

El siguiente número difiere en el dígito de menor peso y por tanto es el $(01,01)_8$.

La diferencia decimal es el valor del peso menor:

$$(01, 01)_8 - (01, 00)_8 = (00, 01)_8 = (1/16)_{10}$$

Una conclusión indirecta de este análisis es que para un número octal con dos dígitos en la parte fraccional, las cantidades no difieren más de 1/16 en valor decimal, lo cual es equivalente a decir que la precisión decimal de la representación octal es 1/16. Cualquier cantidad decimal representada con dos dígitos octales en la parte fraccional, diferirá del valor decimal real en menos de 1/16.

1.1.6 SISTEMA NUMÉRICO BINARIO

Introducido por Leibniz en el siglo XVII, este sistema de base 2 es el más sencillo de todos por poseer solo dos dígitos. Los dos dígitos, llamados “bits”, son el 1 y el 0, por lo cual el equivalente decimal se obtendrá al sumar los pesos correspondientes a los bits con valor uno .

El sistema binario se utiliza para representar señales digitales. Un grupo de 8 bits recibe el nombre de **byte**, mientras que un grupo de 4 bits se denomina **nibble**.

A continuación se presenta el resumen de las características del sistema binario.

Base o Raíz: 2 (dos).

Dígitos : 0, 1.

Conteo o representación de cantidades enteras con tres dígitos:

000, 001, 010, 011, 100, 101, 110, 111.

Representación En polinomio(valor equivalente en el sistema decimal):

$$(N)_2 = (...d_2 2^2 + d_1 2^1 + d_0 2^0 + d_{-1} 2^{-1} + d_{-2} 2^{-2} ...)_{10}.$$

Peso de cada posición:

$$(...2^2 2^1 2^0). \text{ Parte entera.}$$

$$(0, 2^{-1} 2^{-2} 2^{-3}...). \text{ Parte fraccional.}$$

$$(N)_2 = (...16 8 4 2 1, 1/2 1/4 1/8...). \text{ Partes entera y fraccional.}$$

EJEMPLO 1-5

Determine el valor decimal del número $(1010,11)_2$

Solución:

Utilizando la representación en polinomio:

$$(N)_2 = (1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2})_{10}.$$

$$(N)_2 = (10 + 3/4)_{10} = (10,75)_{10}.$$

Utilizando los pesos:

PESOS	8	4	2	1	1/2	1/4
DIGITOS	1	0	1	0	1	1
VALOR DECIMAL	8	0	2	0	1/2	1/4
CANTIDAD	10	+ 3/4				

1.1.7 SISTEMA NUMÉRICO HEXADECIMAL

Este sistema de numeración posee 16 dígitos y se utiliza ampliamente para representar de forma simplificada números binarios de gran cantidad de bits puesto que la conversión entre el sistema binario y el hexadecimal es muy sencilla.

El sistema hexadecimal está asociado con el manejo de datos en computadoras en términos de bytes los cuales pueden ser fácilmente representados en términos hexadecimales.

Puesto que solo se conocen diez dígitos, provenientes del sistema decimal, se utilizan las letras A, B, C, D, E y F para completar los dieciséis dígitos hexadecimales que cubrirán las cantidades diez, once, doce, trece, catorce y quince respectivamente.

Por tanto, las características del sistema hexadecimal son las siguientes:

Base o Raíz: 16 (dieciséis).

Dígitos : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Conteo o representación de cantidades con dos dígitos:

00, 01, 02, ..., 09, 0A, 0B, ..., 0F, 10, 11, 12, ..., 1F, 20, 21, ..., FF.

Representación en polinomio (valor equivalente en el sistema decimal):

$$(N)_{16} = (N)_{\text{HEX}} = (...d_2 16^2 + d_1 16^1 + d_0 16^0 + d_{-1} 16^{-1} + d_{-2} 16^{-2} ...)_{10}$$

Peso de cada posición:

$$(N)_{16} = (...256 16^1, 1/16, 1/256, 1/4096...). \text{ Partes entera y fraccional.}$$

EJEMPLO 1-6

Determine el valor decimal del número $(\text{CAFE}, 1\text{F})_{16}$

Solución:

Utilizando la representación en polinomio:

$$(N)_{16} = (C 16^3 + A 16^2 + F 16^1 + E 16^0 + 1 16^{-1} + F 16^{-2})_{10}$$

$$(N)_{16} = (12 16^3 + 10 16^2 + 15 16^1 + 14 16^0 + 1 16^{-1} + 15 16^{-2})_{10}$$

$$(N)_{16} = (51966,12109375)_{10}$$

Utilizando los pesos:

PESOS	4096	256	16	1	1/16	1/256
DIGITOS	12	10	15	14	1	15
VALOR DECIMAL	49152	2560	240	14	1/16	15/256
CANTIDAD	51966		+ 31/256			

1.1.8 CONVERSIÓN DE BASE 10 EN BASE B

Al exponer los diferentes sistemas hemos indicado como calcular el valor decimal equivalente de un número cualquiera en base B, enseguida se exponen varios métodos para convertir un número decimal en un número de base B. Dado un número en base 10, se desea obtener su equivalente en cualquier base:

$$(N)_{10} = (... , d_3 \ d_2 \ d_1 \ d_0 \ d_{-1} \ d_{-2} \ d_{-3} \ ...)_B$$

Es decir, se requiere encontrar los dígitos d_i tal que:

$$(N)_{10} = (... d_3 B^3 + d_2 B^2 + d_1 B^1 + d_0 B^0 + d_{-1} B^{-1} + d_{-2} B^{-2} + d_{-3} B^{-3} ...)_{10}$$

Dividiremos el problema en dos partes:

- Deducción de los dígitos de la parte entera (... $d_3 \ d_2 \ d_1 \ d_0$)_B
- Deducción de los dígitos de la parte fraccional ($d_{-1} \ d_{-2} \ d_{-3}...$)_B

Parte entera

□ **Método de aproximaciones sucesivas**

Consiste en determinar el mayor valor (dígito x peso) = ($d_i \times B^i$) que no exceda el valor del número decimal N para obtener un primer valor d_i . Si el valor ($d_i \times B^i$) no corresponde al valor exacto del número decimal, se procede a determinar el siguiente mayor valor ($d_i \times B^i$) y por tanto el siguiente d_i que cubrirá el resto del número decimal. Continúa así el proceso hasta tener el valor exacto equivalente al número decimal.

EJEMPLO 1-7

Determine el número en base 3 equivalente al decimal 15.

Solución:

Los pesos de la parte entera del sistema base 3 son:

PESOS: ... 81, 27, 9, 3, 1.

DIGITOS (... ? ? ? ? ?)₃

Se observa que el mayor peso a incluir en el número base 3, será el 9, y para no sobrepasar la cantidad de 15 el mayor dígito debe ser el 1 ya que $1 \times 9 = 9 \leq 15$.

PESOS: ... 81, 27, 9, 3, 1.

DIGITOS (... 0 0 1 ? ?)₃ = 9

Al escoger el dígito 1 en el peso 9, queda por cubrir $15 - 9 = 6$ unidades, esto se realiza repitiendo el procedimiento y seleccionando el peso 3 y el dígito 2 ($2 \times 3 = 6$).

Pendiente Revisión.

PESOS: ... 81, 27, 9, 3, 1.

DIGITOS (... 0 0 1 2 ?)₃ = 15.

Por estar cubierta la cantidad decimal el resto de los dígitos serán cero.

PESOS: ... 81, 27, 9, 3, 1.

DIGITOS (... 0 0 1 2 0)₃ = 15

Por tanto:

$$(15)_{10} = (00120)_3 = (120)_3$$

□ Método de divisiones sucesivas por la base

Dada la parte entera de un número en base 10 queremos obtener su equivalente en cualquier base.

$$(N)_{10} = (... d_3 d_2 d_1 d_0 d_{-1} d_{-2} d_{-3} ...)_{10}$$

$$\text{Parte entera de } (N)_{10} = (... d_3 d_2 d_1 d_0)_B$$

Es decir, para convertir la parte entera de un número decimal se requiere encontrar los dígitos d_i tal que:

$$\text{Parte entera de } (N)_{10} = (... d_3 B^3 + d_2 B^2 + d_1 B^1 + d_0 B^0)_{10}.$$

Si dividimos el número decimal por la base B, obtendremos:

$$(N/B)_{10} = (... d_3 B^{3-1} + d_2 B^{2-1} + d_1 B^{1-1} + d_0 B^{0-1})_{10}$$

$$(N/B)_{10} = (... d_3 B^2 + d_2 B^1 + d_1 B^0 + d_0/B)_{10}$$

Como $B > d_0$, el resultado posee una parte entera y un residuo de valor d_0 .

$$(N/B)_{10} = (... d_3 B^2 + d_2 B^1 + d_1 B^0)_{10} + (d_0/B)_{10}$$

El residuo de la división por la base del número decimal es el dígito menos significativo del número en base B (d_0).

Aplicando el mismo procedimiento al resultado: $(... d_3 B^2 + d_2 B^1 + d_1 B^0)$ se obtendrá el siguiente dígito d_1 del número en base B.

Esto muestra que dividiendo sucesivamente por la base, encontraremos cada uno de los dígitos d_i como el residuo de cada una de las divisiones de la parte entera del número decimal. Es de observar que la división se hace en el sistema decimal.

EJEMPLO 1-8

Determine el número binario equivalente al decimal 125.

Solución:

Dividiendo sucesivamente por la base:

$$125 \div 2 = 62 \quad + \text{ residuo } 1. \text{ Dígito } d_0 = 1.$$

Pendiente Revisión.

$$62 \div 2 = 31 \quad + \text{ residuo } 0. \text{ Dígito } d_1 = 0$$

$$31 \div 2 = 15 \quad + \text{ residuo } 1. \text{ Dígito } d_2 = 1$$

$$15 \div 2 = 7 \quad + \text{ residuo } 1. \text{ Dígito } d_3 = 1$$

$$7 \div 2 = 3 \quad + \text{ residuo } 1. \text{ Dígito } d_4 = 1$$

$$3 \div 2 = 1 \quad + \text{ residuo } 1. \text{ Dígito } d_5 = 1$$

$$1 \div 2 = 0 \quad + \text{ residuo } 1. \text{ Dígito } d_6 = 1$$

Cociente = 0, fin de la conversión.

Por tanto:

$$(125)_{10} = (1111101)_2$$

EJEMPLO 1-9

Determine el número octal equivalente al decimal 125.

Solución:

$$125 \div 8 = 15 \quad + \text{ residuo } 5. \text{ Dígito } d_0 = 5$$

$$15 \div 8 = 1 \quad + \text{ residuo } 7. \text{ Dígito } d_1 = 7$$

$$1 \div 8 = 0 \quad + \text{ residuo } 1. \text{ Dígito } d_2 = 1$$

Cociente = 0, fin de la conversión.

$$(125)_{10} = (175)_8$$

Parte fracción

Dada la parte fraccional de un número en base 10 queremos obtener su equivalente en cualquier base.

$$\text{Parte fraccional de } (N)_{10} = (0, d_0 \ d_1 \ d_2 \ d_3 \ \dots)_B$$

Para convertir la parte fraccional de un número decimal se requiere encontrar los dígitos d_i tal que:

$$\text{Parte fraccional de } (N)_{10} = (d_1 B^{-1} + d_2 B^{-2} + d_3 B^{-3} \dots)_{10}$$

Si multiplicamos el número decimal por la base B, obtendremos :

$$(N \times B)_{10} = (d_1 B^{-1+1} + d_2 B^{-2+1} + d_3 B^{-3+1} \dots)_{10}$$

$$(N \times B)_{10} = (d_1 + d_2 B^{-1} + d_3 B^{-2} \dots)_{10}$$

El resultado es una parte entera de valor igual a d_1 más una parte fraccional.

$$(N \times B)_{10} = d_1 + (d_2 B^{-1} + d_3 B^{-2} \dots)_{10}$$

Repitiendo el procedimiento para la parte fraccional $(d_2 B^{-1} + d_3 B^{-2} \dots)_{10}$ se obtendrá el dígito d_2 del número en base B.

Pendiente Revisión.

Por tanto multiplicando sucesivamente la parte fraccional por la base, encontraremos cada uno de los dígitos d_i como la parte entera del resultado de la operación.

EJEMPLO 1-10

Determine el número binario equivalente al decimal (0,715).

Solución:

$$0,715 \times 2 = 1,430 \quad \text{Parte entera: } 1, d_1 = 1$$

$$0,430 \times 2 = 0,860 \quad \text{Parte entera: } 0, d_2 = 0$$

$$0,860 \times 2 = 1,720 \quad \text{Parte entera: } 1, d_3 = 1$$

$$0,720 \times 2 = 1,440 \quad \text{Parte entera: } 1, d_4 = 1$$

$$0,440 \times 2 = 0,880 \quad \text{Parte entera: } 1, d_5 = 0$$

Para cinco multiplicaciones tenemos como resultado: $(0,10110)_2$

Lo que equivale a: $(0,6875)_{10}$, es decir, $(0,038)\%$ de diferencia con el número a convertir.

Continuando el procedimiento para mejorar la aproximación tenemos:

$$0,880 \times 2 = 1,760 \quad \text{Parte entera: } 1, d_6 = 1$$

$$0,760 \times 2 = 1,520 \quad \text{Parte entera: } 1, d_7 = 1$$

$$0,520 \times 2 = 1,040 \quad \text{Parte entera: } 1, d_8 = 1$$

$$0,040 \times 2 = 0,800 \quad \text{Parte entera: } 0, d_9 = 0$$

La conversión da como resultado:

$(0,101101110)_2$ lo que equivale a: $(0,71484375)_{10}$ es decir, $(0,0002)\%$ de error.

1.1.9 CONVERSIÓN EN SISTEMAS DE BASE 2^N

Conversión entre sistemas binario y octal

□ Rango de dígitos octales consecutivos

Sea un número N en el sistema base $8 = 2^3$ (octal).

$$(N)_8 = (... d_3 d_2 d_1 d_0 d_{-1} d_{-2} d_{-3} ...)_8$$

$$(N)_{10} = (... d_3 \times 8^3 + d_2 \times 8^2 + d_1 \times 8^1 + d_0 \times 8^0 + d_{-1} \times 8^{-1} + d_{-2} \times 8^{-2} + d_{-3} \times 8^{-3} ...)_{10}.$$

Cada dígito d_i dígito permite representar un conjunto de cantidades decimales específicas ($d_i \times 8^i$) o valores relativos.

Pendiente Revisión.

Observemos las cantidades que se pueden representar con dígitos octales consecutivos:

El dígito d_0 permite representar desde cero unidades hasta un máximo de 7×1 unidades ($d_0 \times 8^0$) en múltiplos o pasos de una unidad.

El dígito d_1 permite representar desde cero unidades hasta un máximo de 56 unidades ($d_1 \times 8^1$) en múltiplos de 8. Es decir, d_1 puede representar las cantidades 0, 8, 16, 24, ..., 56.

El dígito d_1 permite representar desde cero unidades hasta un máximo de $7 \times 1/8 = 0,875$ unidades (7×8^{-1}) en múltiplos o pasos de $1/8$, es decir, puede representar las cantidades 0, $1/8$, $2/8$, ..., $7/8$.

□ Rango de dígitos binarios consecutivos

Realicemos un análisis similar para el sistema binario. Sea un número N en el sistema base 2 .

$$(N)_2 = (... d_3 d_2 d_1 d_0, d_{-1} d_{-2} d_{-3} ...)_2$$

$$(N)_{10} = (... d_3 2^3 + d_2 2^2 + d_1 2^1 + d_0 2^0 + d_{-1} 2^{-1} + d_{-2} 2^{-2} + d_{-3} 2^{-3} ...)_{10}$$

Sabemos que con un grupo de tres (3) bits se pueden representar los números 0 al 7, los cuales corresponden exactamente a los dígitos del sistema octal. Analicemos que cantidades se pueden representar con grupos consecutivos de tres (3) bits.

Así tenemos que:

Los tres (3) dígitos enteros menos significativos $d_2 d_1 d_0$ permiten representar desde cero unidades hasta un máximo de 7 ($2^2 + 2^1 + 2^0$) unidades en pasos de 1 unidad (2^0).

Los siguientes tres (3) dígitos enteros $d_5 d_4 d_3$ permiten representar desde cero unidades hasta un máximo de 56 ($2^5 + 2^4 + 2^3$) unidades en pasos de 8 (2^3).

Los tres dígitos de la parte fraccional $d_{-1} d_{-2} d_{-3}$ permiten representar desde cero unidades hasta un máximo de ($2^{-1} + 2^{-2} + 2^{-3}$) = 0,875 unidades en múltiplos de $1/8$ (2^{-3}), es decir, puede representar las cantidades: 0, $1/8$, $2/8$, ..., $7/8$.

Comparando los resultados anteriores para los sistemas octal y binario, se concluye que conjuntos de grupos de tres bits de un número binario equivalen o cubren las mismas cantidades que un dígito octal.

Por tanto, para convertir un número binario en octal, se puede agrupar los bits del número binario en conjuntos o grupos de tres bits (octal $= 2^3$) y convertir cada gru-

Pendiente Revisión.

po directamente en su dígito octal equivalente. Evidentemente el proceso inverso (conversión octal a binaria) también es correcto.

EJEMPLO 1-11

Realice la conversión al sistema octal de $(10101,00101)_2$

Solución:

Dividiendo el número en grupos de tres bits tenemos:

$(10\ 101, 001\ 01)_2$.

Lo cual puede ser expresado como:

$(010\ 101, 001\ 010)_2$

Convirtiendo cada grupo de tres bits en su equivalente decimal y representando este en base octal obtenemos el número:

$(2\ 5, 1\ 2)_8$

EJEMPLO 1-12

Realice la conversión al sistema binario de $(357,24)_8$

Solución:

Convirtiendo cada dígito octal en su equivalente binario de tres bits obtenemos:

$$\begin{array}{cccccc} 3 & 5 & 7 & 2 & 4 \\ (011\ 101\ 111, 010\ 100)_2 \end{array}$$

Conversión entre sistemas binario y hexadecimal

Utilizando cuatro (4) bits se pueden representar los 15 dígitos (0 al F) correspondientes a los dígitos del sistema hexadecimal.

Un análisis similar al realizado anteriormente para los sistemas binario y octal nos permitiría concluir que, para convertir un número binario a hexadecimal, se debe agrupar el número binario en conjuntos de cuatro bits (hexadecimal = 2^4) y convertir cada conjunto o grupo en su dígito hexadecimal equivalente.

La tabla 1-2 muestra ejemplos del método de conversión entre sistemas 2^N , se ha utilizado el símbolo “/” para separar los bits en grupos adecuados para la conversión.

Pendiente Revisión.

BINARIO (5 bits)	OCTAL	BINARIO (7 bits)	HEX.
10/001	21	101/0111	57
11/111	37	111/1111	7F
1,101/1	1,54	1,0111/01	1,74
0,000/1	0,04	0,1110/11	0,EC
10/011,1	23,4	10/0101,1	25,8

Tabla 1-2 Conversión entre sistemas 2^n

1.2 CÓDIGOS BINARIOS

Un grupo de tres bits o dígitos binarios puede asumir hasta ocho valores, estados o configuraciones posibles. Estas combinaciones van desde la 000 hasta la 111. Un conjunto de n dígitos binarios podrá adoptar 2^n estados o combinaciones, cada una de las cuales puede ser usada para representar, no solo cantidades, sino informaciones distintas. Los sistemas de numeración estudiados son específicamente códigos de representación de cantidades.

Codificación

La codificación consiste en establecer una correspondencia, llamada código, entre cada una de las informaciones por representar y los estados binarios. Generalmente, esta relación es tal que para cada una de las informaciones corresponde únicamente un estado o combinación, dos informaciones cualesquiera tendrán diferentes estados de representación.

El conjunto de bits o código binario que representa una información (cantidad, letra, símbolo) es denominado “palabra código” o simplemente “palabra”. El proceso inverso a la codificación, es decir la transformación de un código en la información que representa se denomina descodificación o decodificación.

Con n bits se pueden codificar 2^n informaciones diferentes, sin embargo cada combinación puede ser asignada a cualquiera de las informaciones, por lo que el número de asignaciones posible es la permutación: $2^n !$, este número constituye todos los posibles códigos binarios.

Entre esta amplia gama de posibles códigos se han escogido los sistemas o codificaciones más apropiados o cómodos para una aplicación específica.

Pendiente Revisión.

A continuación estudiaremos los códigos que por poseer alguna característica importante son usados en sistemas digitales.

1.2.1 CÓDIGOS BINARIOS PONDERADOS

Los códigos pesados son aquellos que tienen un peso o valor relativo para cada posición o dígito binario, cada cifra decimal es representada por un código o palabra binaria cuya suma de pesos para los bits con valor 1 es igual a la cifra decimal.

Aunque el sistema numérico binario es ampliamente utilizado en los procesos digitales, algunas veces es conveniente procesar la información en el sistema decimal, como es el caso del intercambio de datos hombre máquina. Para la representación de las cifras decimales (0 - 9) son necesarios cuatro bits los cuales permiten 16 combinaciones, por lo que existe una gran cantidad de formas diferentes de asignar o repartir los 16 posibles códigos entre las diez cifras decimales.

Así tenemos que para el manejo de los números decimales los códigos más corrientes son el código BCD, el código 2421, el código con exceso de 3 y el código 2 de 5. Es de notar que las combinaciones no utilizadas en cualquier de los códigos se conocen como códigos no permitidos o prohibidos puesto que no pertenecen al código.

En la tabla 1-3 se muestran tres códigos binarios pesados de cuatro bits.

DECIMAL	8 4 2 1	2 4 2 1	6 4 2 -3
0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 1 0 1
2	0 0 1 0	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1	1 0 0 1
4	0 1 0 0	0 1 0 0	0 1 0 0
5	0 1 0 1	1 0 1 1	1 0 1 1
6	0 1 1 0	1 1 0 0	0 1 1 0
7	0 1 1 1	1 1 0 1	1 1 0 1
8	1 0 0 0	1 1 1 0	1 0 1 0
9	1 0 0 1	1 1 1 1	1 1 1 1

Tabla 1-3. Códigos Binarios con pesos

El Código 2421

Se dice que un código es autocomplementado si el intercambio de todos los 1 por 0 en una palabra cualquiera del código que representa el dígito $(N)_{10}$, produce la

Pendiente Revisión.

palabra del código que representa al dígito decimal complemento a 9 de N, es decir, el número decimal $(9-N)_{10}$.

Aunque el código 2421 también es pesado (ver tabla 1-3), se observa que la representación de algunas cifras pudo haber sido diferente, tal es el caso del decimal 7 que se puede representar como 0111 y 1101. Sin embargo, adoptando las representaciones mostradas en la tabla 1-3, el código posee la característica de ser autocomplementado.

El Código 642-3

Los pesos asignados a los dígitos binarios pueden ser negativos como en el código 642-3. Se puede demostrar que una condición necesaria para que un código pesado sea autocomplementado es que la suma de sus pesos sea igual a 9. Existen 13 posibles códigos, con pesos positivos y negativos, autocomplementados.

Código binario natural

En este código cada cifra decimal se representa por una palabra que es el equivalente en el sistema numérico binario al dígito decimal. Por tanto la codificación de cualquier dígito decimal puede ser realizada como una conversión decimal a binario. En la tabla 1-3 se observa el código binario natural de 4 bits denominado también código 8421. El código binario natural de n bits permite la representación de las cantidades decimales desde 0 hasta $2^n - 1$ unidades.

1.2.2 CÓDIGOS SIN PESOS

Existe una diversidad de códigos binarios que no poseen pesos, es decir, no existe una relación entre la posición de un bit y un peso binario. Algunos de estos códigos se muestran en la tabla 1-4.

Código BCD natural

En este código, denominado también código 8421 o simplemente BCD, cada cifra decimal se representa por una palabra que es el equivalente en el sistema numérico binario al dígito decimal (cada **D**ecimal **C**odificado en **B**inario). Por ejemplo, para representar un número de dos cifras como el 81 se codifica cada dígito decimal obteniéndose el 1000 0001, es decir:

Pendiente Revisión.

$$(81)_{10} = (1000\ 0001)_{BCD}$$

Note que para cada dígito la representación puede ser interpretada como una conversión decimal a binario lo cual es muy conveniente.

DEC	2 DE 5	JOHNSON	1 DE 10	EXCESO3
0	11000	00000	1000000000	0011
1	00011	00001	0100000000	0100
2	00101	00011	0010000000	0101
3	00110	00111	0001000000	0110
4	01001	01111	0000100000	0111
5	01010	11111	0000010000	1000
6	01100	11110	0000001000	1001
7	10001	11100	0000000100	1010
8	10010	11000	0000000010	1011
9	10100	10000	0000000001	1100

Tabla 1-4. Códigos Binarios sin pesos

Código BCD exceso 3 (X3)

Este es un código no pesado de 4 bits en el que los dígitos decimales se representan por su equivalente binario natural o directo aumentado en 3. El Exceso 3 es un código autocomentado y posee cualidades que lo hicieron muy práctico en las primeras computadoras decimales. Por ejemplo, para representar un número de dos cifras como el 41 se codifica cada dígito decimal obteniéndose: $(41)_{10} = (01110100)_{X3}$.

Códigos continuos

Existen una serie de códigos cuyas características los hacen ideales para la detección y corrección de errores de codificación. Entre estos tenemos el código Gray, los códigos con bit de paridad y los códigos de Hamming. A continuación se definen algunos términos utilizados en la teoría de códigos para detección y corrección de errores.

□ **Distancias en un Código y Adyacencia**

El número de bits en que difieren dos representaciones o palabras cualesquiera de un código se denomina distancia.

Dos palabras o combinaciones son adyacentes si difieren en sólo en un bit (poseen distancia unitaria).

□ **Distancia Mínima de un Código**

La menor distancia que posee un código se denomina distancia mínima del código. El código 8421 (tabla 1-3) posee una distancia mínima unitaria.

□ **Código continuos y cíclicos**

Un código se dice que es continuo si las palabras o combinaciones correspondientes a números decimales consecutivos son adyacentes.

Un código continuo en el que la última palabra es adyacente a la primera se denomina cíclico. Es decir, un código cíclico posee una distancia unitaria entre cualesquiera dos palabras consecutivas.

De las anteriores definiciones se deduce que el código BCD no es un código continuo y posee una distancia mínima unitaria, mientras que el código Johnson es un código continuo cíclico con distancia mínima unitaria.

Código Gray

El código Gray es un código continuo y cíclico. La tabla 1-5 muestra el código gray de tres (3) bits. El código Gray tiene, por su naturaleza, distancias unitarias entre las palabras consecutivas. Las palabras binarias o representaciones de dos números sucesivos no se distinguen más que por la modificación de un solo bit. Esta característica lo hace conveniente en algunos casos en los que se necesita convertir ciertas cantidades físicas directamente a códigos binarios evitando errores en la conversión.

Pendiente Revisión.

DECIMAL	GRAY
0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100

Tabla 1-5. Código Gray de 3 bits

Un ejemplo de utilización del código es la codificación de la posición angular de un eje, digamos de una antena de radar; de forma parecida a la mostrada en la figura 1-1.

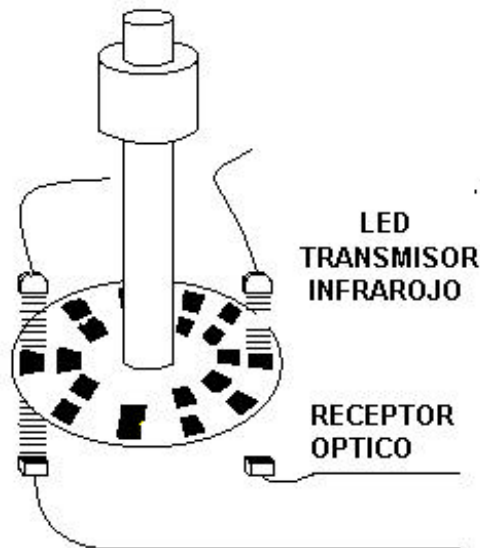


Figura 1-1 Codificación de la posición de un eje

Para realizar esto se fija al eje un disco dividido, digamos, en 8 zonas o sectores. Cada zona del disco está dividida en 3 subsectores y en algunos de estos subsectores se hacen agujeros. Frente a estos subsectores ponemos lámparas o leds y en el otro lado fotoreceptores. Cuando una de estas zonas o sectores este frente a las lámparas, los fotoreceptores producirán el código binario correspondiente.

Pendiente Revisión.

ZONA	DESDE (Grados)	HASTA (Grados)	BINARIO NATURAL	REFLEJADO	GRAY
1	00	45	000	1 11	000
2	45	90	001	1 10	001
3	90	135	010	1 00	011
4	135	180	011	<u>1 01</u>	<u>010</u>
5	180	225	100	0 01	110
6	225	270	101	0 00	111
7	270	315	110	0 10	101
8	315	360	111	0 11	100

Tabla 1-6. Codificación de posición

Consideremos la utilización del código binario natural para representar las 8 zonas en un disco como se muestra en la tabla 1-6. Cuando se pasa de la ZONA 4 a la ZONA 5 es decir, del código 011 al 100, estaremos frente a un efecto de ambigüedad no deseado en el cual el bit más significativo cambia un poco antes que los otros bits, entonces, el sistema puede interpretar una lectura momentánea de 111 como una posición valedera. Este problema es resuelto utilizando un código especial donde la transición de un número a otro inmediato se produce con el cambio de un solo bit.

El código Gray es uno de los códigos sin peso que cumple con la condición anteriormente indicada y posee la ventaja de fácil conversión con el código binario.

1.2.3 CÓDIGOS REFLEJADOS

En la tabla 1-6 también se muestra otro código en el cual solo un cambia bit entre números consecutivos y entre los correspondientes a la 1era y última combinación; también se observa el eje de reflexión o simetría del que proviene el nombre dado a este tipo de códigos.

El método de generación de este tipo de código, consiste en tomar un grupo de dos o más palabras de n bits y copiarlas girándolas o reflejándolas en relación con un eje como se indica en la tabla 1-6. Se generan así 2n palabras a las que luego se adiciona un bit que será el de mayor peso (MSB), el bit adicionado se le asigna el valor de uno (ó cero) para el grupo original y cero (ó uno) para el grupo reflejado obteniéndose un código continuo y cíclico de 2n combinaciones o palabras.

1.2.4 CÓDIGOS DETECTORES DE ERROR

La transmisión de información digital es más inmune al ruido que la analógica, sin embargo es posible que se produzcan errores. Cuando en un código binario de n bits están presentes todas las posibles 2^n combinaciones, es imposible detectar un error, puesto que un dato con error corresponde a otro perteneciente al código.

La detección de error en un código binario se puede realizar si el error en un dato lo transforma en una combinación no perteneciente al código. La condición necesaria para la detección de error es la distancia mínima del código. La distancia mínima es el mínimo número de posiciones de bits en que difieren dos combinaciones, palabras o cadenas de bits cualesquiera del código.

Si la distancia mínima de un código es dos, un error transformará el dato o palabra en otra diferente a cualquiera del código, siendo posible su detección.

Existen diversos tipos de códigos detectores de errores, tales como el código 2 de 5 y los códigos de paridad.

Códigos de paridad

Estos códigos se obtienen al adicionar a cada palabra o combinación del código original un bit llamado bit de paridad.

Si el código que se desea obtener es de paridad par, el bit de paridad deberá ser tal que el número de unos de la nueva combinación sea par, en el caso contrario se generará un código de paridad impar.

El efecto del bit de paridad es aumentar la distancia del código permitiendo la detección de un error. La detección se realiza fácilmente al comprobar si el número de unos del dato y su bit de paridad (nuevo código) recibido corresponde a la paridad original enviada.

En la tabla 1-7 se muestran en una columna los datos en código original y en otra los correspondientes datos con su bit de paridad.

El bit de paridad permite la detección de un error en una palabra pero no determina en que bit ocurrió dicho error.

Pendiente Revisión.

	DATO ORIGINAL	DATO + BIT PARIDAD PAR	DATO + BIT PARIDAD IMPAR
a)	101011011	101011011 <u>0</u>	101011011 <u>1</u>
b)	100100010	100100010 <u>1</u>	100100010 <u>0</u>

Tabla 1-7. Codificación con bit de paridad

1.2.5 CÓDIGOS CORRECTORES DE ERROR

Los códigos correctores de error proporcionan la información sobre cual es el dígito o bit con error permitiendo así, por simple inversión de este, su corrección. Estos códigos se utilizan solamente cuando la transmisión de datos no es posible realizarla nuevamente en caso de detección de error.

Para poder corregir N errores la distancia mínima del código debe ser:

$$D = 2N + 1$$

Códigos Hamming

En 1950, R. Hamming estableció un método para codificar datos de tal manera que fuese posible la corrección de un error. Los códigos de Hamming permiten la corrección de un error al convertir un código original de distancia unidad en un código de distancia tres, siendo la posición del error la indicada por un conjunto de p bits. Así, a un código original de n bits (datos) se le adicionan p bits formando un nuevo código (Hamming) de n+ p bits. En este nuevo código los p bits son bits de paridad y cada uno se genera para detectar error en algunas de las n+p posiciones o bits del nuevo código.

Dado que con p bits se tienen 2^p combinaciones se debe escoger un valor de p que satisfaga la relación:

$$2^p \geq n+p+1$$

□ **Codificación de Datos**

Como ejemplo, generemos un código de Hamming para un código original (datos) de 4 bits. Puesto que n = 4, se deben adicionar p = 3 bits para cubrir todas las n+p posiciones de posible error y la condición de cero error.

Sean P3 P2 P1 los bits de paridad. Para adicionar estos bits tendremos en cuenta la formación del nuevo código según el esquema de la tabla 1-8:

Pendiente Revisión.

CÓDIGO ORIGINAL

$D_4 \quad D_3 \quad D_2 \quad D_1$

CÓDIGO DE HAMMING

B_7	B_6	B_5	B_4	B_3	B_2	B_1
D_4	D_3	D_2	P_3	D_1	P_2	P_1

Tabla 1-8. Codificación de datos

Los bits de paridad se localizan en las posiciones $B_i = 2^n$ y los datos en el resto de posiciones del código. Los bits de paridad serán del valor adecuado para generar paridad (par o impar) en grupos de bits que permitan detectar un error en estos grupos.

□ **Grupos de datos para paridad**

Los bits p toman el valor correspondiente por paridad de acuerdo al valor de grupos de datos o bits ubicados en las posiciones siguientes:

P_1 paridad para posiciones $B_3B_5B_7$

P_2 paridad para posiciones $B_3B_6B_7$

P_3 paridad para posiciones $B_5B_6B_7$

Nótese que para generar cada bit de paridad se verifican las posiciones B_i en las cuales el bit de paridad P_i toma el valor de 1 en el número equivalente binario $(P_3P_2P_1)_2$.

EJEMPLO 1-13

Codifique en Hamming los datos $(D_4D_3D_2D_1) = (1110)$.

Solución:

Siguiendo el esquema antes indicado se tiene:

B_7	B_6	B_5	B_4	B_3	B_2	B_1
D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	1	1		0		

Pendiente Revisión.

Se obtienen los valores apropiados para los bits p utilizando paridad par :

B_7	B_6	B_5	B_4	B_3	B_2	B_1
D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	1	1	1	0	0	0

El dato será transmitido en código Hamming como (1111000).

❑ **Determinación del Error**

El código Hamming construido con las reglas anteriores podrá ser utilizado para localizar un (1) error en cualquiera de las posiciones o n+p bits.

Sea (C3 C2 C1) un conjunto de bits que indicará la existencia y ubicación de un error en el código Hamming para cuatro bits de datos. Los bits (C3C2C1) formarán un número en el sistema binario natural cuyo equivalente decimal dará la posición del bit erróneo. En el caso que no exista error el número será cero. Cada uno de estos bits detectará la presencia de un error por verificación de paridad en los grupos de bits de datos y en su bit de paridad. En caso de error cada bit asume el valor de uno y en conjunto (C3 C2 C1) formarán el número que indica la posición del error.

Las posiciones o bits que serán verificadas por los bits (C3 C2 C1) serán entonces, para el caso de cuatro bits de datos, las siguientes:

C_1 paridad en las posiciones $B_1B_3B_5B_7$; es decir, los bits $D_4D_2D_1P_1$

C_2 paridad en las posiciones $B_2B_3B_6B_7$; es decir, los bits $D_4D_3D_1P_2$

C_3 paridad en las posiciones $B_4B_5B_6B_7$; es decir, los bits $D_4D_3D_2P_3$

EJEMPLO 1-14

Supongamos que el dato antes codificado en Hamming con paridad par, se recibe con un error: (1011000).

Determinar si ocurrió un error y en que posición o bit.

Solución:

La revisión de la paridad par, o generación de los C_i bits produce:

Pendiente Revisión.

C_1 (paridad par para las posiciones $B_1B_3B_5B_7$) = 0 (bits sin error)

C_2 (paridad par para las posiciones $B_2B_3B_6B_7$) = 1 (uno de los bits con error).

C_3 (paridad par para las posiciones $B_4B_5B_6B_7$) = 1 (uno de los bits con error)

Por tanto el error se localiza en la posición número seis, $(C_3C_2C_1) = (110)$, o en el bit D_3 .

1.2.6 CÓDIGOS ALFANUMÉRICOS

Estos códigos se utilizan cuando se necesita representar caracteres del alfabeto, signos de puntuación, operadores, caracteres especiales y caracteres para el control o dialogo entre máquinas y periféricos. Existen diferentes códigos alfanuméricos, el más conocido es el ASCII, aunque la mayoría presentan las siguientes características:

- La representación de cifras decimales se realiza al utilizar un código ya establecido (binario natural, exceso 3, etc.)
- La representación incluye un sistema para detección de errores.
- La forma del código permite añadir nuevos caracteres y mejorarlo.

En la tabla 1-9 se muestra un código de 8 bits utilizado para presentar los caracteres alfanuméricos en un visualizador o pantalla LCD, es de observar que la ilustración pertenece a un equipo o fabricante específico.

Pendiente Revisión.

Higher Lower 4bit 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000		0	1	2	3	4	5	6	7	8	9	A	B
xxxx0001		C	D	E	F	G	H	I	J	K	L	M	N
xxxx0010		O	P	Q	R	S	T	U	V	W	X	Y	Z
xxxx0011		[\	^	_	`	a	b	c	d	e	f	g
xxxx0100		h	i	j	k	l	m	n	o	p	q	r	s
xxxx0101		t	u	v	w	x	y	z	{		}	~	
xxxx0110													
xxxx0111													
xxxx1000													
xxxx1001													
xxxx1010													
xxxx1011													
xxxx1100													
xxxx1101													
xxxx1110													
xxxx1111													

Tabla 1-9 Código alfanumérico Fuente: <http://www.hitachi.com/lcddisplay/hd44780>