

1. SISTEMAS Y CÓDIGOS DE NUMERACIÓN

1. SISTEMAS Y CÓDIGOS DE NUMERACIÓN

I TEORÍA

1 INTRODUCCIÓN

Los sistemas digitales funcionan bajo el control de variables que sólo pueden tener dos valores (variables binarias). Por lo general durante el control de dichos sistemas deben realizarse una serie de operaciones, por lo que será necesario utilizar sistemas y códigos de numeración binarios. Dado que nuestro sistema ordinario es el decimal, deberemos estudiar distintos sistemas y códigos de numeración que faciliten la codificación en binario.

2 SISTEMAS DE NUMERACIÓN

2.1 GENERALIDADES

Cualquier número N en un sistema de base b puede expresarse mediante un polinomio de potencias de la base, multiplicadas por el dígito perteneciente al sistema. Esto es:

$$N = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_i b^i + \dots + a_1 b^1 + a_0 b^0 + a_n b^n + a_{n-1} b^{n-1} + \dots + a_j b^j$$

donde b es la base del sistema y a_i el dígito de peso i perteneciente al sistema. Así, el número 1492,36 en decimal, puede expresarse como:

$$1492,36 = 1 \cdot 10^3 + 4 \cdot 10^2 + 9 \cdot 10^1 + 2 \cdot 10^0 + 3 \cdot 10^{-1} + 6 \cdot 10^{-2}$$

De igual modo el número 100110,101 en binario se expresará por el binomio:

$$100110,101 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

2.2 SISTEMAS BINARIO, OCTAL Y HEXADECIMAL

El sistema binario es el más empleado en los sistemas digitales puesto que sólo tiene dos dígitos para expresar cualquier valor. El interés de los sistemas octal (base 8) y hexadecimal (base 16) proviene de su fácil conversión de números en estas bases a binario y viceversa, como veremos más adelante (más información en [WB_Enlace_1_1] y en [WB_Link_1_1]). En la tabla siguiente se muestra la equivalencia entre los dieciséis primeros números de dichos sistemas y el decimal.

Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bin	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111
Oct	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Cuando se trabaja en varios sistemas se suele poner un subíndice que identifique la base,

1. SISTEMAS Y CÓDIGOS DE NUMERACIÓN

$$158,6_{10} \quad 1011,6_8 \quad 1B5,8C_{16} \quad 1011,101_2$$

Es muy común que los números hexadecimales también se indiquen poniendo una hache mayúscula o minúscula al final del número o el símbolo 0x delante del mismo (12AH, 12Ah o 0x12A serán distintas formas de indicar $12A_{16}$).

La conversión de un número en cualquiera de estas tres bases en decimal se realiza fácilmente representando el número en forma de polinomio y operando en decimal:

$$1011,6_8 = 1 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 + 6 \cdot 8^{-1} = 512 + 8 + 0,75 = 520,75_{10}$$

$$1B5,8C_{16} = 1 \cdot 16^2 + 11 \cdot 16^1 + 5 \cdot 16^0 + 8 \cdot 16^{-1} + 12 \cdot 16^{-2} = 256 + 176 + 5 + 0,5 + 0,046875 = 437,546875_{10}$$

$$1011,101_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 8 + 2 + 1 + 0,5 + 0,125 = 11,625_{10}$$

El proceso inverso debe realizarse en dos fases: una para la parte entera y otra para la fraccionaria. Matemáticamente se demuestra [Malvino 1.3] que:

1º) Si se divide un número entero expresado en la base b1 por la base b2, y el cociente se vuelve a dividir por b2 y así sucesivamente, el último cociente y los restos obtenidos equivalen al número en base b2.

2º) Si se multiplica un número fraccionario N en base b1 por la base b2, la parte entera representa la cifra más significativa en base b2. Si la parte fraccionaria del resultado se vuelve a multiplicar por b2 se obtiene la segunda cifra significativa en base b2. Continuando el proceso se obtienen todas las cifras del fraccionario en base b2 de N.

Vamos a pasar el número decimal 149,175 a binario, octal y hexadecimal, aplicando el método anterior.

Binario: $149,175_{10} = 1010001,0010110_2$

	Cociente	Resto		
149 : 2	74	1		
74 : 2	37	0		
37 : 2	18	1		
18 : 2	9	0		
9 : 2	4	1		
4 : 2	2	0		
2 : 2	1	0		
				1 0 0 1 0 1 0 1

			0,175 · 2 = 0,350	
			0,350 · 2 = 0,700	
			0,700 · 2 = 1,400	
			0,400 · 2 = 0,800	Inicio del período
			0,800 · 2 = 1,600	
			0,600 · 2 = 1,200	
			0,200 · 2 = 0,400	Final del período
			0,400 · 2 = 0,800	

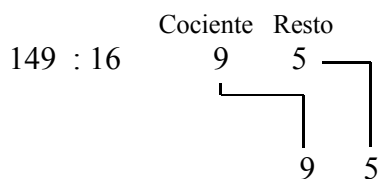
Octal: $149,175_{10} = 225,13146_8$

	Cociente	Resto			
149 : 8	18	5			
18 : 8	2	2			
				2 2 5	

			0,175 · 8 = 1,400	
			0,400 · 8 = 3,200	Inicio del período
			0,200 · 8 = 1,600	
			0,600 · 8 = 4,800	
			0,800 · 8 = 6,400	Final del período
			0,400 · 8 = 3,200	

1. SISTEMAS Y CÓDIGOS DE NUMERACIÓN

Hexadecimal: $149,175_{10} = 95,2C_{16}$



$$0,175 \cdot 16 = 2,800$$

$$0,800 \cdot 16 = 12,800 \quad \text{Período}$$

Más arriba comentamos que el interés de los sistemas octal y hexadecimal proviene de su fácil conversión de números en estas bases a binario y viceversa. Observe los resultados obtenidos:

$$1010001,00101110_2 = 225,13146_8 = 95,2C_{16}$$

Reagrupe los dígitos binarios (bits) en paquetes de 3 y 4 bits y encuentre el modo de una fácil conversión a los sistemas octal y hexadecimal respectivamente.

3 CÓDIGOS BINARIOS

3.1 DEFINICIÓN

Por código entendemos la representación unívoca de las cantidades por una combinación de símbolos. Cuando los símbolos son bits, hablamos de un código binario.

3.2 TIPOS

Podemos encontrarnos con códigos cíclicos, ponderados o no, detectores y correctores de error. En la siguiente tabla se describen cada uno de estos códigos.

TIPO	DESCRIPCIÓN	EJEMPLOS
Cíclico	Las combinaciones adyacentes sólo se diferencian en un bit	Gray, Jonson
Ponderado	La posición de cada bit tiene un peso fijo.	BCD natural, BCD Aiken, Biquinario
No ponderado	La posición de cada bit no tiene un peso fijo.	BCD exceso 3, Gray, Jonson
Detector de error	Es capaz de detectar errores en la transmisión.	Códigos con paridad par o impar
Corrector de error	Es capaz de detectar y corregir errores en la transmisión.	Hamming
Alfanumérico	Además de información numérica, codifica letras y símbolos.	ASCII

Los códigos BCD (Decimal Codificado en Binario) tienen la particularidad de convertirse fácilmente en decimal. En la página siguiente se pueden observar la mayoría de los códigos mencionados (más información en [CD_Link_1_1] y en [WB_Enlace_1_2]).

1. SISTEMAS Y CÓDIGOS DE NUMERACIÓN

CÓDIGOS BINARIOS								
DE C	BCD				DETECTORES DE ERRORES			
	NATURAL	AIKEN 2 4 2 1	5 4 2 1	EXCESO 3	BCD NAT. CON BIT DE PARIDAD PAR	BCD NAT. CON BIT DE PARIDAD NON	2 ENTRE 5	BIQUINARIO 5 0 4 3 2 1 0
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0 0	1 0 0 0 0	0 1 1 0 0	0 1 0 0 0 0 1
1	0 0 0 1	0 0 0 1	0 0 0 1	0 1 0 0	1 0 0 0 1	0 0 0 0 1	1 1 0 0 0	0 1 0 0 0 1 0
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 1	1 0 0 1 0	0 0 0 1 0	1 0 1 0 0	0 1 0 0 1 0 0
3	0 0 1 1	0 0 1 1	0 0 1 1	0 1 1 0	0 0 0 1 1	1 0 0 1 1	1 0 0 1 0	0 1 0 1 0 0 0
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1	1 0 1 0 0	0 0 1 0 0	0 1 0 1 0	0 1 1 0 0 0 0
5	0 1 0 1	1 0 1 1	1 0 0 0	1 0 0 0	0 0 1 0 1	1 0 1 0 1	0 0 1 1 0	1 0 0 0 0 0 1
6	0 1 1 0	1 1 0 0	1 0 0 1	1 0 0 1	0 0 1 1 0	1 0 1 1 0	1 0 0 0 1	1 0 0 0 0 1 0
7	0 1 1 1	1 1 0 1	1 0 1 0	1 0 1 0	1 0 1 1 1	0 0 1 1 1	0 1 0 0 1	1 0 0 0 1 0 0
8	1 0 0 0	1 1 1 0	1 0 1 1	1 0 1 1	1 1 0 0 0	0 1 0 0 0	0 0 1 0 1	1 0 0 1 0 0 0
9	1 0 0 1	1 1 1 1	1 1 0 0	1 1 0 0	0 1 0 0 1	1 1 0 0 1	0 0 0 1 1	1 0 1 0 0 0 0
OTROS CÓDIGOS BINARIOS								
DE C	NATURAL	CONTINUOS				CORRECTOR DE ERRORES		
		GRAY 2 BITS	GRAY 3 BITS	GRAY 4 BITS	JOHNSON 5 BITS	HAMMING b7 b6 b5 b4 b3 b2 b1		
0	0 0 0 0	00	000	0000	00000	0 0 0 0 0 0 0		
1	0 0 0 1	01	001	0001	00001	0 0 0 0 1 1 1		
2	0 0 1 0	11	011	0011	00011	0 0 1 1 0 0 1		
3	0 0 1 1	10	010	0010	00111	0 0 1 1 1 1 0		
4	0 1 0 0	-	110	0110	01111	0 1 0 1 0 1 0		
5	0 1 0 1	-	111	0111	11111	0 1 0 1 1 0 1		
6	0 1 1 0	-	101	0101	11110	0 1 1 0 0 1 1		
7	0 1 1 1	-	100	0100	11100	0 1 1 0 1 0 0		
8	1 0 0 0	-	-	1100	11000	1 0 0 1 0 1 1		
9	1 0 0 1	-	-	1101	10000	1 0 0 1 1 0 0		
10	1 0 1 0	-	-	1111	-	-		
11	1 0 1 1	-	-	1110	-	-		
12	1 1 0 0	-	-	1010	-	-		
13	1 1 0 1	-	-	1011	-	-		
14	1 1 1 0	-	-	1001	-	-		
15	1 1 1 1	-	-	1000	-	-		

II IMPROVE YOUR TECHNICAL ENGLISH

1 Binary numbers

Binary numbers can get very confusing. A number can be stated in several different ways. Most numbers start out as a binary numbers or a number in base two. Then you can transform that number into base ten, or decimal numbers. Base eight numbers, or octal numbers, travel in order:0,1,2,3,4,5,6,7. Base sixteen numbers, or hexadecimal numbers use both numbers and characters:0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Bases is a hard concept to learn. You have to try hard at it and practice it sometimes.

2 Converting BCD codes to decimal

To convert BCD codes to decimal, break the binary digits into groups of four starting from the binary point and convert each group into its appropriate BCD digit.

Examples:

Convert each of the BCD numbers below to their decimal equivalent

A) 1110000111 B) 1100111.0100

$$A) (1110000111)_{BCD} = 0011\ 1000\ 0111 = (387)_{10}$$

$$B) (1100111.0100)_{BCD} = 0110\ 0111.\ 0100 = (67.4)_{10}$$

3 The Nibble

A nibble is a collection of bits on a 4-bit boundary. It wouldn't be a particularly interesting data structure except for two items: BCD (binary coded decimal) numbers and hexadecimal (base 16) numbers. It takes four bits to represent a single BCD or hexadecimal digit.

With a nibble, we can represent up to 16 distinct values. In the case of hexadecimal numbers, the values 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F are represented with four bits. BCD uses ten different digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) and requires four bits. In fact, any sixteen distinct values can be represented with a nibble, but hexadecimal and BCD digits are the primary items we can represent with a single nibble.

4 Gray Code

The Gray code belongs¹⁾ to a class of codes called minimum change codes, in which only one bit in the code changes when moving from one step to next step. The Gray code is non-weighted²⁾ code, as the position of bit does not contain any weight. In digital Gray code has got special place (see [WB_Link_1_2]).

1) pertenece

2) no ponderado

III PRÁCTICA Y EJERCICIOS

1. APARTADOS PRÁCTICOS

1. Con [CD_Enlace_1_1] puedes acceder al fichero de texto *tema01_ej3* y con [WinHex] a un editor en hexadecimal. A partir de ello realice las siguientes actividades:

- Mire las propiedades del fichero de texto, anote su tamaño en bytes y ábralo para ver su contenido.
- Ejecute la aplicación WinHex, abra el fichero *tema01_ej3* y razone el código hexadecimal que aparece.
- Razone el valor indicado en el Offset con el tamaño del fichero.
- Cree un nuevo documento con WinHex y codifique en hexadecimal y en octal los siguientes 16 caracteres: ~oO-* ECAM *-Oo~

2. Obtenga el código BCD natural utilizando el generador de palabras del simulador digital y visualice su salida mediante sondas rojas.

3. Utilice el visualizador de 7 segmentos codificado del simulador digital, conéctelo al generador de palabras con el peso adecuado y observe los dieciséis valores visualizados.

4. El código Hamming se genera a partir del BCD natural (bits b7, b6, b5 y b3) y mediante la operación suma exclusiva (bits b4, b2 y b1). En base a ello monte en el simulador digital un circuito que genere el código Hamming a partir del BCD natural generado en el apartado 2.

2. EJERCICIOS

1. Complete adecuadamente la siguiente tabla con números en binario, octal, decimal y hexadecimal según corresponda.

BINARIO	OCTAL	DECIMAL	HEXADECIMAL
101010110,1101			
	246,162		
		107,24	
			10B,A

2. Los conjuntos de bits, en especial los que son potencia de dos suelen recibir un nombre característico. Así al conjunto de 4 bits se le denomina cuarteto o nibble en inglés. Indique el nombre de los conjuntos de bits más empleados. [WB_Link_1_3].

3. Explique el concepto de paridad y obtenga a partir del código Gray de 3 bits un código de paridad par.

4. Indique las diferencias entre los códigos ANSI ASCII e IBM ASCII.