Sistemas Digitales

Diego Enrique Fontán, CosasDePuma

2017-18

Índice

1. Introducción a los Sistemas Digitales

1.1. Tipos de señales

Tipos de señales que nos podemos encontrar en relación a los valores que pueden tomar:

- **Señales analógicas:** Pueden tomar infinitos valores distintos a lo largo del tiempo.

- **Señales digitales:** Sólo pueden tomar un número finito de valores distintos a lo largo del tiempo.

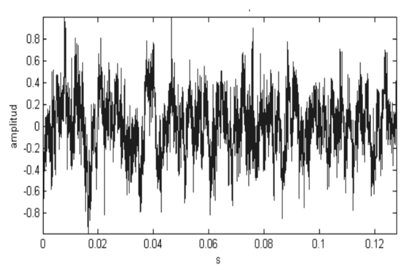

Figura 1: Ejemplo de señal analógica

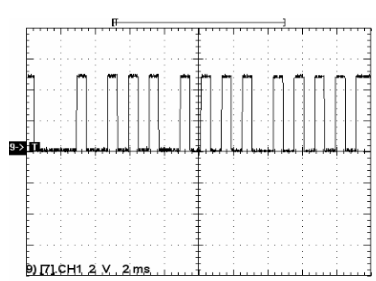

Figura 2: Ejemplo de señal digital

1.2. Señales binarias

Las señales binarias son un caso particular de las señales digitales.

Se caracterizan porque sólo pueden tomar dos valores distintos a lo largo del tiempo.

Los sistemas electrónicos digitales que se utilizan hoy en día operan con señales binarias.

Por lo tanto, según el valor que tome la señal en cada momento, se pueden determinar dos estados diferentes que posteriormente se emparejarán a números binarios.

Figura 3: Ejemplo de señal binaria

El que se denomine a los sistemas que utilizan estas señales como *sistemas digitales* en vez de *sitemas binarios* se debe a que, en general, procesan valores digitales, los cuales se codifican mediante combinaciones de valores binarios para que puedan ser tratados.

1.3. Ventajas de los sistemas digitales

Las ventajas que tienen los sistemas digitales respecto a los analógicos son, entre otras:

- Son más fáciles de diseñar.
- Son menos sensibles a agentes externos (como a las interferencias).
- Permiten almacenar y operar con grandes cantidades de información de forma rápida y segura.

2. Sistemas de numeración y códigos binarios

Un sistema de numeración no es más que un método que se usa para representar cantidades de forma simbólica, siguiendo ciertas normas y convenios.

Un número es una representación mediante símbolos de una cantidad dada.

Una misma cantidad puede ser representada mediante un número distinto en función del sistema de numeración que se considere.

Por ejemplo:

$$12_{10} = C_{16} = 14_8 = 1100_2$$

Los sistemas de numeración sólamente expresan el módulo, magnitud o valor absoluto de una cantidad, nunca si es positiva o negativa.

Hoy en día utilizamos sistemas de numeración posicionales, caracterizados por:

- Las cantidades se representan mediante una sucesión ordenada de números a ambos lados de un símbolo de referencia (punto, coma...)

Cantidad

3735928559	,	129152485
entera	símbolo	fraccionaria

- Cada dígito tiene asociado un **peso** cuyo valor depende de la **base** y de la **posición** que ocupe el dígito.

2.1. Sistemas de numeración

Un sistema de numeración de base **b** utiliza hasta **b** símbolo para representar diferentes cantidades.

La base de un sistema de numeracíon también se denomina *raíz* o *módulo*.

Sistema de numeración	Símbolos utilizados
Binario (base = 2)	0, 1
Ternario (base = 3)	0, 1, 2
Octal (base = 8)	0, 1, 3, 4, 5, 6, 7
Decimal (base = 10)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal (base = 16)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Cuadro 1: Símbolos usados según el sistema de numeración utilizado

Como hemos visto, una misma cantidad se puede representar de diferentes maneras, según la base que utilicemos. La tabla de equivalencias entre los diferentes sistemas de numeración más utilizados sería la siguiente:

Binario	Octal	Decimal	Hexadec.	Binario	Octal	Decimal	Hexadec.
0000	00	00	0	1000	10	08	8
0001	01	01	1	1001	11	09	9
0010	02	02	2	1010	12	10	A
0011	03	03	3	1011	13	11	B
0100	04	04	4	1100	14	12	C
0101	05	05	5	1101	15	13	D
0110	06	06	6	1110	16	14	E
0111	07	07	7	1111	17	15	F

2.2. Peso numérico según la base y posición

Las posiciones y los pesos asociados a cada uno de los dígitos de una cantidad guardan la siguiente relación:

$$\overbrace{\underbrace{a_{n-1}^{b^{n-1}}\ a_{n-2}^{b^{n-2}}\ ...\ a_1^{b}\ a_0^{1}}_{parte\ entera}\ ,\ \underbrace{a_{-1}^{b^{-1}}\ a_{-2}^{b^{-2}}\ ...\ a_{-(m-1)}^{b^{-(m-1)}}\ a_{-m}^{b^{-m}}}_{parte\ fraccionaria}}^{pesos}$$

Veamos los pesos asociados a la base decimal:

$$\overset{10000}{10^4}\ \overset{1000}{10^3}\ \overset{100}{10^2}\ \overset{10}{10^1}\ \overset{1}{10^0}\ \overset{0{,}1}{10^{-1}}\ \overset{0{,}01}{10^{-2}}\ \overset{0{,}001}{10^{-3}}\ \overset{0{,}0001}{10^{-4}}$$

También es importante poder identificar rápidamente los pesos asociados a la base binaria:

$$\overset{256}{2^8}\ \overset{128}{2^7}\ \overset{64}{2^6}\ \overset{32}{2^5}\ \overset{16}{2^4}\ \overset{8}{2^3}\ \overset{4}{2^2}\ \overset{2}{2^1}\ \overset{1}{2^0}\ \overset{0{,}5}{2^{-1}}\ \overset{0{,}25}{2^{-2}}\ \overset{0{,}125}{2^{-3}}\ \overset{0{,}0625}{2^{-4}}$$

2.3. Conversión entre bases

2.3.1. A base decimal

Para pasar una cantidad <u>en una base cualquiera a base decimal</u>, se utiliza la fórmula determinada como **polinomio característico**:

$$N_b = \sum_{i=-m}^{i=n-1} a_i b^i = \underbrace{a_{n-1}b^{n-1} + ... + a_1 b^1 + a_0 b^0}_{parte\ entera} + \underbrace{a_{-1}b^{-1} + a_{-2}b^{-2} + ... + a_{-m}b^{-m}}_{parte\ fraccionaria}$$

2.3.2. De decimal a binario

- **Método Teórico:** La representación de la parte entera de un número decimal se calcula como sucesivas divisiones entre 2 hasta obtener un coeficiente menor que dos. La solución está formada por los restos de las divisiones en orden inverso al que se han ido generando.

Dividendo	Divisor	Cociente	**Resto**
12	2	6	**0**
6	2	3	**0**
3	2	1	**1**
1	2	0	**1**

12 de decimal a binario

Como los restos se leen en el orden inverso al obtenido, 12_{10} en decimal es 1100_2 en binario.

Para determinar la representación de una parte fraccionaria, se realizan multiplicaciones por 2 hasta obtener el número de dígitos deseados o hasta que la parte fraccionaria de la conversión sea cero (exacta).

La representación en binario de la parte fraccionaria está formada por los valores enteros de los resultados leídos en el orden en el que se obtienen.

$$
\begin{array}{lcl}
0{,}2017 & \text{x } 2 = & \mathbf{0}{,}4034 \\
0{,}4034 & \text{x } 2 = & \mathbf{0}{,}8068 \\
0{,}8068 & \text{x } 2 = & \mathbf{1}{,}6136 \\
0{,}6136 & \text{x } 2 = & \mathbf{1}{,}2257 \\
0{,}2257 & \text{x } 2 = & \mathbf{0}{,}4544 \\
0{,}4544 & \text{x } 2 = & \mathbf{0}{,}9088 \\
0{,}9088 & \text{x } 2 = & \mathbf{1}{,}8176 \\
 & \vdots &
\end{array}
$$

0,2017 de decimal a binario

Como la parte entera se lee en el orden obtenido, 0,2017$_{10}$ en decimal es 0011001$_2$ en binario con 7 bits de precisión.

Si lo que queremos es representar un número con parte entera y parte fraccionaria, seguiremos los pasos descritos anteriormente y los uniremos en un mismo número.

De esta forma:

$$12,2017_{10} = 1100,0011001_2$$

- **Método práctico:** Sabiendo los pesos de la base binaria, pondremos (de derecha a izquierda) un 1 en la posición que nos de una cantidad menor o igual al número que estamos buscando y un 0 en el que nos de un valor mayor al deseado.

90,25 de decimal a binario

64	32	16	8	4	2	1	0.5	0.25	0.125	0.0625	Decimal acumulado
0	0	0	0	0	0	0	0	0	0	0	00,00
1	0	0	0	0	0	0	0	0	0	0	64,00
~~1~~	~~**1**~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~96,00~~
1	0	**1**	0	0	0	0	0	0	0	0	80,00
1	0	1	**1**	0	0	0	0	0	0	0	88,00
~~1~~	~~0~~	~~1~~	~~1~~	~~**1**~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~92,00~~
1	0	1	1	0	**1**	0	0	0	0	0	90,00
~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~1~~	~~**1**~~	~~0~~	~~0~~	~~0~~	~~0~~	~~91,00~~
~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~1~~	~~0~~	~~**1**~~	~~0~~	~~0~~	~~0~~	~~90,50~~
1	0	1	1	0	1	0	0	**1**	0	0	**90,25**

De esta forma obtenemos que:

$$90{,}25_{10} = 1011010,01_2$$

2.3.3. De decimal a cualquier base

Para ello tendremos que seguir los pasos descritos en el apartado anterior pero cambiando el operando de las multiplicaciones y divisiones. En vez de utilizar un 2, utilizaremos el módulo de nuestra base. Por ejemplo:

Dividendo	Divisor	Cociente	**Resto**
2894	16	180	**E** (14)
180	16	11	**4**
11	16	0	**B** (11)

2894_{10} es lo mismo que $B4E_{16}$

Lo mismo ocurre si nos sabemos los pesos, aunque a causa de tener más símbolos que 0 y 1, este método se vuelve inviable:

545 de decimal a octal

512	64	8	1	Decimal acumulado
0	0	0	0	000
1	0	0	0	512
~~1~~	~~**1**~~	~~0~~	~~0~~	~~576~~
~~1~~	~~0~~	~~**7**~~	~~0~~	~~568~~
~~1~~	~~0~~	~~**6**~~	~~0~~	~~560~~
~~1~~	~~0~~	~~**5**~~	~~0~~	~~552~~
1	0	**4**	0	544
1	0	4	**1**	**545**

545_{10} es equivalente a 1041_8

2.4. De binario a octal o hexadecimal

Dado que tanto binario, cuaternario, octal, hexadecimal... son bases que tienen como módulo una potencia de 2, se da entre ellas una propiedad que permite realizar cambios de una manera mucho más cómoda y fluída.

Por ejemplo, si queremos pasar de binario a octal, deberemos separar desde la coma grupos de 3 dígitos binarios y rellenar aquellos que sobren con ceros.

Sea el número $10111110{,}1_2$:

$$\underbrace{\mathbf{0}10}\underbrace{111}\underbrace{110}\;,\;\underbrace{1\mathbf{00}}$$

Posteriormente sustituímos cada grupo de tres dígitos por su equivalente en octal:

$$\underbrace{010}_{2}\underbrace{111}_{7}\underbrace{110}_{6}\;,\;\underbrace{100}_{4}$$

De esta forma, el número $10111110{,}1_2$ es equivalente a $276{,}4_8$.

El que separemos los dígitos en binario de tres en tres se debe a que $8 = 2^3$. En el caso de que queramos pasarlo a hexadecimal ($16 = 2^4$) tendremos que separarlos en grupos de 4 y repetir el proceso:

$$\underbrace{1011}_{B}\;\underbrace{1110}_{E}\;,\;\underbrace{1000}_{8}$$

De esta forma, el número $10111110{,}1_2$ es equivalente a $BE,8_{16}$.

3. Aritmética binaria