Tema 1: Introducción a los Sistemas Digitales

Universidad de Huelva

Escuela Técnica Superior de Ingeniería

Departamento de
Ingeniería Electrónica, Sistemas Informáticos y Automática

Concepto y definición de sistema

El concepto de SISTEMA se extiende a una amplia variedad de campos:

- **Comunicaciones**
- **Aeronáutica y Astronáutica**
- **Diseño de Circuitos**
- **Acústica**
- **Sismología**
- **Ingeniería Biomédica**
- **Sistemas de Distribución y de generación de Energía**
- **Control de Procesos Químicos**
- **Procesado de Señales**

Todos los sistemas responden a una excitación en particular con una señal de respuesta y un comportamiento definido. **Un sistema puede verse como cualquier proceso que realice transformación de señales**:

Entrada y salida de un sistema

- La **entrada** es el estímulo, la excitación que recibe un sistema desde una fuente de energía externa, usualmente para producir una respuesta específica.
- La **salida** es la respuesta real que se obtiene de un sistema. Puede ser o no igual a la respuesta implícita especificada por la entrada.

Sistemas continuos y discretos en el tiempo

- Las señales de entrada y salida de un sistema, así como las que intervienen en la dinámica del sistema, son siempre función de una variable independiente, normalmente el **tiempo *t***.

- Una señal dependiente de valores continuos de la variable independiente *t* se denomina **señal continua** en el tiempo. (**SEÑAL ANALÓGICA).**

- Una señal definida solamente en instantes discretos del valor de la variable independiente *t* de la que depende, se denomina **señal discreta** en el tiempo. (**SEÑAL DIGITAL).**

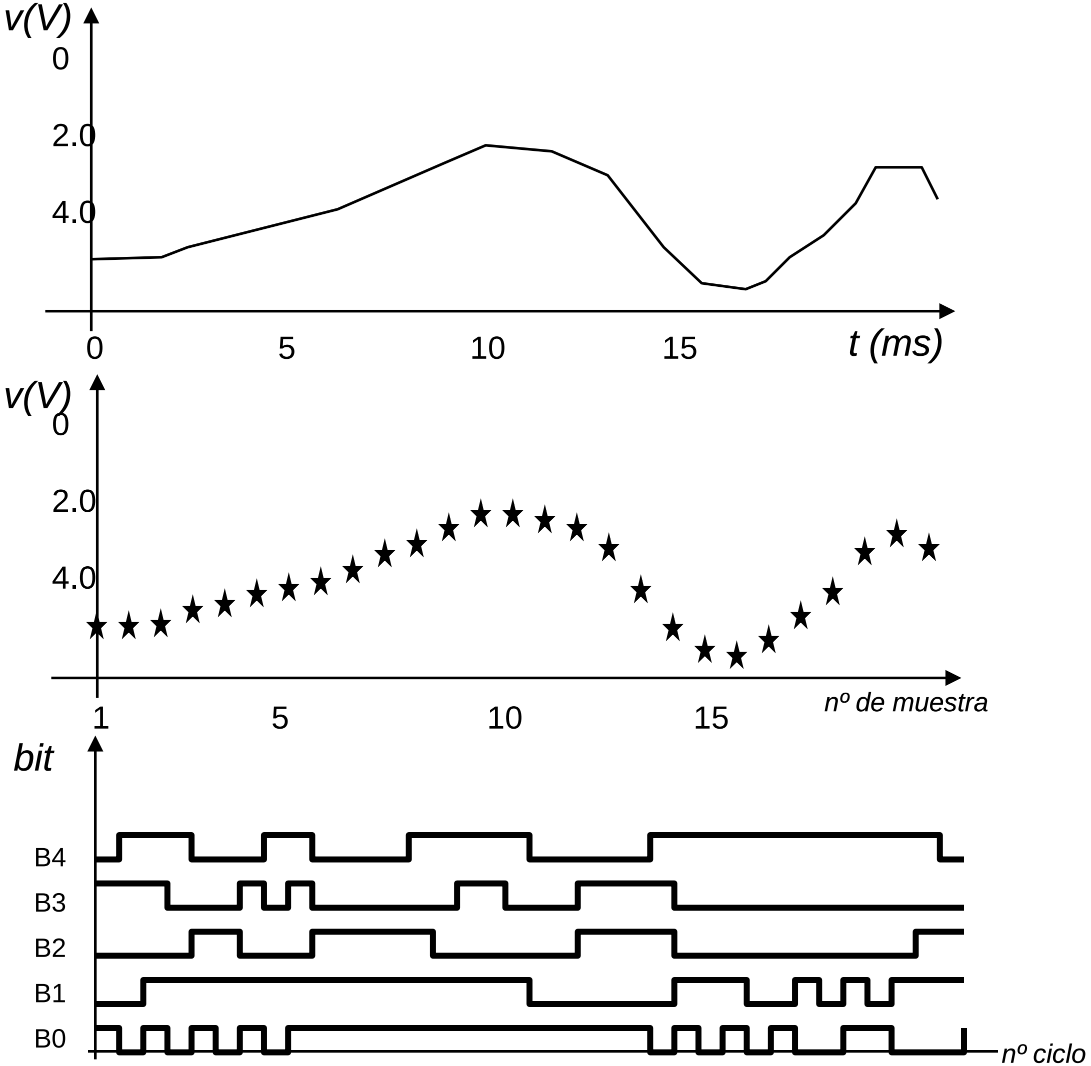
Ejemplos de Señales:

- El voltaje continuo que varia senoidalmente *v*(*t*) o la corriente alterna *i*(*t*), que proporciona la red eléctrica es una señal continua en el tiempo, pues está definida para cada instante de tiempo *t*. Aunque solo se habla de su valor eficaz (220 V) y su frecuencia (50Hz).
- El intermitente de un automóvil proporciona una señal discreta, al igual que una baliza de señalización marina. O está encendido o está apagado.
- La temperatura ambiente de una habitación es una señal claramente continua en el tiempo (podemos saber en cualquier momento cual es la temperatura); sin embargo, si tomamos la temperatura cada hora, estamos obteniendo una señal discreta. Este proceso se llama muestreo y permite establecer una relación entre el mundo ANALÓGICO y el mundo DIGITAL .

Por otro lado, en función del tipo de señales que "procesen" los sistemas, podemos considerar **sistemas continuos "Analógicos"**, si lo que procesan son señales continuas, o **sistemas discretos "Digitales"**, si lo hacen con señales discretas.

Los computadores no son más que sistemas DIGITALES, procesan señales DIGITALES: UNOS Y CEROS.

Señales continuas y discretas en el tiempo



Tipos de Sistemas Digitales

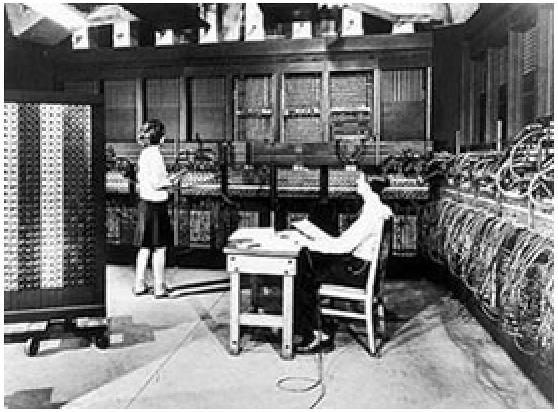
Están divididos básicamente en dos tipos:

- ***CIRCUITOS DIGITALES COMBINACIONALES***: Las salidas en un instante determinado sólo son función de las entradas en dicho instante.

- ***CIRCUITOS DIGITALES SECUENCIALES***: Las salidas en un instante determinado no son sólo función de las entradas en dicho instante, si no que además, dependen de entradas previas. Son sistemas con memoria.

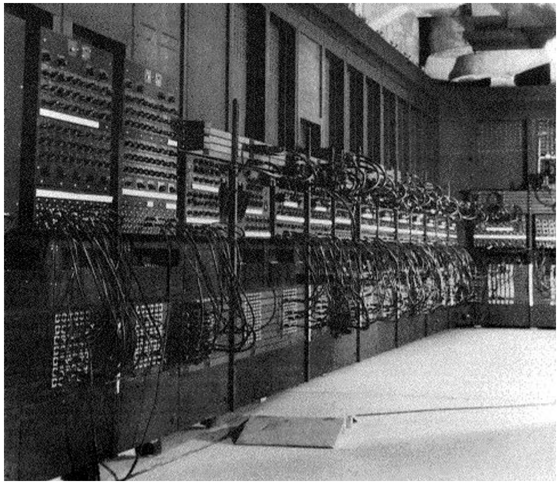
EL PRIMER COMPUTADOR ELECTRÓNICO

John P. Eckert y John W. Mauchly construyeron en **1946, en la Universidad de Pennsylvania**, el **ENIAC** (Electronic Numerical Integrator and Computer) , primer computador electrónico, compuesto de 17.468 válvulas o tubos de vidrio al vacío (más resistencias, condensadores, etc.), esto producía un problema ya que la vida media de un tubo era de unas 3000 horas por lo que aproximadamente cada 10 minutos se estropeaba un tubo y no era nada sencillo buscar un tubo entre 18000, consumiéndose gran cantidad de tiempo en ello. El ENIAC **pesaba 32 toneladas** y **ocupaba 2,40 metros de ancho y 30 metros de largo**.

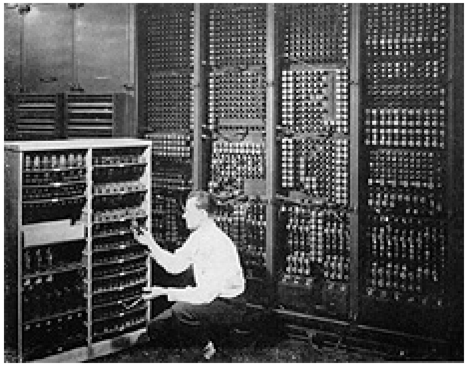


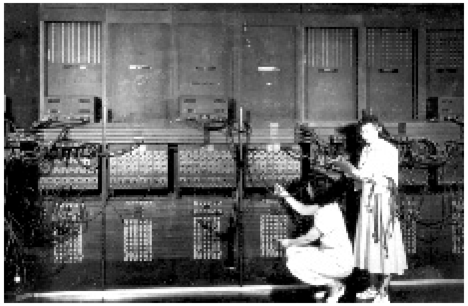
El calor de las válvulas elevaba la temperatura del local hasta los 50º. **Para efectuar diferentes operaciones, debían cambiarse las conexiones (cables) como en las viejas centrales telefónicas**, lo cual era un trabajo que podía tomar varios días. Era capaz de calcular con gran velocidad la trayectorias de proyectiles, principal objetivo inicial de su construcción.

El ENIAC fue un ordenador electrónico digital con fines generales a gran escala. Tenía dos innovaciones técnicas, la primera es que combinaba diversos componentes técnicos (40000 componentes entre tubos, condensadores, resistencias, interruptores, etc.) e ideas de diseño en un único sistema que era capaz de realizar **5000 sumas y 300 multiplicaciones por segundo**. La segunda era la fiabilidad de la máquina. Para resolver el problema de los tubos de vacío, se aplicaron unos estrictos controles de calidad de los componentes utilizados. Salió a la luz pública el 14 de febrero de 1946, apareciendo en la prensa con calificativos como "cerebro electrónico", "Einstein mecánico" o "Frankenstein matemático", como por ejemplo en el diario *Newsweek*.



Su programación era terriblemente tediosa, requería la operación manual de unos 6.000 interruptores, y su programa o software, cuando requería modificaciones, tardaba semanas de instalación manual. Sus tubos debían cambiarse continuamente. Su tiempo medio de rotura era de 1 hora.





COMPUTADORES ACTUALES (sólo 60 años después)

En Informática, FLOPS es el acrónimo de *Floating point Operations Per Second* (operaciones en coma flotante por segundo). Se usa como una medida del rendimiento de una computadora, especialmente en cálculos científicos que requieren un gran uso de operaciones de punto flotante. FLOPS, al ser un acrónimo, no debe nombrarse en singular como FLOP, ya que la S final alude a second (o segundo) y no al plural.

Una ordenador personal moderno de escritorio dispone de procesadores con uno o varios núcleos que operan a más de 2 **GHz**, y es capaz de alcanzar una velocidad de unos cuantos GFLOPS (Gigaflops).

Las computadoras exhiben un amplio rango de rendimientos en punto flotante, por lo que a menudo se usan unidades mayores que el FLOPS. Los prefijos estándar pueden ser usados para este propósito, dando como resultado **megaFLOPS** (MFLOPS, 10^6 FLOPS), **gigaFLOPS** (GFLOPS, 10^9 FLOPS), **teraFLOPS** (TFLOPS, 10^{12} FLOPS), y **petaFLOPS** (PFLOPS, 10^{15} FLOPS).

La primera supercomputadora Cray-1 fue puesta en marcha en el Laboratorio Nacional de Los Álamos en 1976. La Cray-1 era capaz de operar a 80 MFLOPS. En menos de treinta años desde entonces la velocidad computacional de las supercomputadoras es más de un millón de veces mayor.

La computadora más rápida del mundo hasta la fecha (12 Junio 2008) es la supercomputadora Roadrunner, capaz de un pico máximo de 1026 TeraFLOPS (más de 1 PetaFLOPS) con 12.960 procesadores Cell modificados (Play Station 3) y 6480 procesadores Opteron de AMD, que deben correr en paralelo para que la máquina funcione correctamente. Esta instalado en el Departamento Nacional de Seguridad Nuclear de Estados Unidos (NNSA por sus siglas en Inglés). La misma NNSA es dueña de «Blue Gene/L» y «ASC Purple» que actualmente son el segundo y cuarto supercomputadores más rápidos.

Representación de los números

Un **sistema de numeración** es un conjunto de símbolos empleados para representar información numérica.

El **alfabeto** o conjunto de símbolos de los que disponemos depende de la **base** de dicho sistema de numeración.

- **SISTEMA BINARIO** (0,1)
- **SISTEMA DECIMAL** (0,1,2,3,4,5,6,7,8,9)
- **SISTEMA OCTAL** (0,1,2,3,4,5,6,7)
- **SISTEMA HEXADECIMAL** (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)

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

Cualquier número tiene su representación en una base b de la forma $(X_n X_{n-1} \ldots X_1 X_0)_b$, cuyo valor numérico es:

$$(N)_b = X_n * b^n + X_{n-1} * b^{n-1} + \ldots + X_1 * b^1 + X_0 * b^0$$

Conversión entre las distintas bases (I)

CONVERSIÓN DE BINARIO A OCTAL (HEXADECIMAL) Y VICEVERSA:

- Cada dígito en octal equivale a tres dígitos en binario: $8 = 2^3$
- Cada dígito hexadecimal equivale a cuatro dígitos en binario: $16 = 2^4$

Base	*Resultado*					
Binaria	**0**	**0**	**1**	**0**	**1**	**1**
Octal	**1**			**3**		
Hexadecimal			**B**			

CONVERSION ENTRE TODO TIPO DE BASES

Método polinómico

- Expresa el número de la base fuente como un polinomio
- Se evalúa dicho polinomio según la aritmética de la base destino:

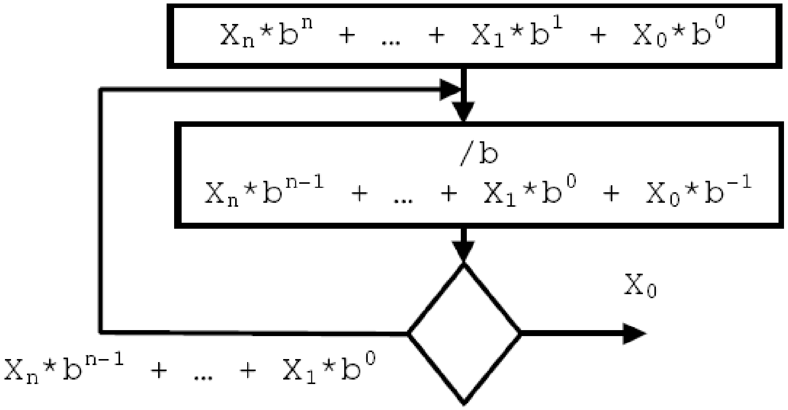
$$(N)_{b1} = X_n{*}b_2^{\,n} + \ldots + X_0{*}b_2^{\,0} + \ldots + X_{-q}{*}b_2^{\,-q} = \Sigma\, X_i{*}b_2^{\,i}$$

- La parte decimal se trata igual que la entera salvo que los exponentes de las bases son negativos
- Útil cuando la base destino sea la decimal

Conversión entre las distintas bases (II)

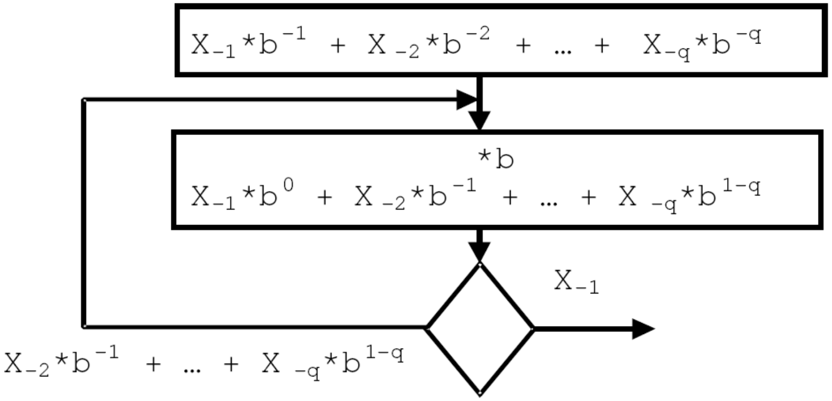
MÉTODO ITERATIVO

Conversión de la parte entera



- La parte entera se divide por b
- El resto es el dígito menos significativo
- Se repite con el cociente y el resto obtenido es el 2º dígito menos significativo
- Continuar hasta que el cociente obtenido sea menor que la base b

Conversión de la parte decimal



- La parte decimal se multiplica por b
- La parte entera de la multiplicación es el dígito más significativo
- Se extrae esta parte entera y se vuelve a realizar la misma operación
- Continuar con el proceso hasta obtener los decimales requeridos

Tema 1: Introducción a los Sistemas Digitales

Codificación

Es la relación entre el alfabeto fuente y el alfabeto destino.

En el **Sistema binario** de numeración (0,1):

- Cada dígito se denomina ***bit*. El bit es la cantidad más pequeña posible de información**
- **MSB**: *Most Significant Bit*, bit más significativo
- **LSB**: *Least Significant Bit*, bit menos significativo

DISTANCIA DE HAMMING es el número de dígitos mínimo en que difieren dos palabras de un código.

CÓDIGOS CON PESO (PONDERADOS) son códigos de la forma:

$$X_n{*}W_n + X_{n-1}{*}W_{n-1} + \ldots + X_1{*}W_1$$

Representado por $X_n\, X_{n-1} \ldots X_1$ y el Vector peso $W_n\, W_{n-1} \ldots W_1$

CÓDIGOS AUTOCOMPLEMENTARIOS son código cuyas palabras correspondientes a un valor cualquiera D y a 9-D tienen los unos por ceros y viceversa.

CÓDIGOS PROGRESIVOS códigos en los que las combinaciones correspondientes a números decimales consecutivos son adyacentes. Dos combinaciones son adyacentes cuando difieren en un solo bit (por ejemplo 0011 y 0001). Si la última combinación es adyacente a la primera, se llaman CÍCLICOS.

Decimal	Binario	Johnson	GRAY
0	0000	00000	0000
1	0001	00001	0001
2	0010	00011	0011
3	0011	00111	0010
4	0100	01111	0110
5	0101	11111	0111
6	0110	11110	0101
7	0111	11100	0100
8	1000	11000	1100
9	1001	10000	1101
10	1010		1111
11	1011		1110
12	1100		1010
13	1101		1011
14	1110		1001
15	1111		1000

Códigos BCD

Son códigos que representan los dígitos de 0 al 9 (Decimal Codificado en Binario)

DÍGITO DECIMAL	Natural				Exceso 3				Aiken				5421			
	8	4	2	1					2	4	2	1	5	4	2	1
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	1
2	0	0	1	0	0	1	0	1	0	0	1	0	0	0	1	0
3	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	1	1	0	1	0	0	0	1	0	0
5	0	1	0	1	1	0	0	0	1	0	1	1	1	0	0	0
6	0	1	1	0	1	0	0	1	1	1	0	0	1	0	0	1
7	0	1	1	1	1	0	1	0	1	1	0	1	1	0	1	0
8	1	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1
9	1	0	0	1	1	1	0	0	1	1	1	1	1	1	0	0

Códigos Alfanuméricos

Como el **ASCII** (*American Standard Code for Information Interchange*)

Carácter	ASCII	Carácter	ASCII
A	0100 0001	W	0101 0111
B	0100 0010	X	0101 1000
C	0100 0011	Y	0101 1001
D	0100 0100	Z	0101 1010
E	0100 0101	0	0011 0000
F	0100 0110	1	0011 0001
G	0100 0111	2	0011 0010
H	0100 1000	3	0011 0011
I	0100 1001	4	0011 0100
J	0100 1010	5	0011 0101
K	0100 1011	6	0011 0110
L	0100 1100	7	0011 0111
M	0100 1101	8	0011 1000
N	0100 1110	9	0011 1001
O	0100 1111	+	0010 1011
P	0101 0000	-	0010 1101
Q	0101 0001	*	0010 1010
R	0101 0010	:	0011 1010
S	0101 0011	=	0011 1101
T	0101 0100	<	0011 1100
U	0101 0101	;	0011 1011
V	0101 0110		

Códigos detectores de errores

- Son códigos que permiten detectar que existe un error en la combinación binaria que se recibe
- Se pueden obtener mediante la adición de un bit de paridad a un código no detector
- Cualquier código de detección de fallo en un solo bit debe tener al menos una distancia de dos entre cualesquiera de las combinaciones del código.
- Las palabras con una distancia mínima de dos se pueden usar como códigos de detección de error en un bit.

	2 entre 5					Biquinario							Binario con paridad				
						5	0	4	3	2	1	0	b_3	b_2	b_1	b_0	p
0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1
2	1	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	1
3	1	0	0	1	0	0	1	0	1	0	0	0	0	0	1	1	0
4	0	1	0	1	0	0	1	1	0	0	0	0	0	1	0	0	1
5	0	0	1	1	0	1	0	0	0	0	0	1	0	1	0	1	0
6	1	0	0	0	1	1	0	0	0	0	1	0	0	1	1	0	0
7	0	1	0	0	1	1	0	0	0	1	0	0	0	1	1	1	1
8	0	0	1	0	1	1	0	0	1	0	0	0	1	0	0	0	1
9	0	0	0	1	1	1	0	1	0	0	0	0	1	0	0	1	0

Códigos correctores de error (I)

- Los más conocidos son los códigos de Hamming
- A un código de distancia unidad de **n** bits, se añaden **p** bits de paridad (o comprobación), formando un nuevo código de (**n+p**) bits
- El número **p** de bits añadidos debe ser suficiente para detectar error y ausencia de error en las (**n+p**) posiciones de cada palabra del código
- Dado que con **p** bits se obtienen 2^p posiciones, se ha de cumplir la relación $2^p \geq n + p + 1$
- La condición necesaria y suficiente para que cualquier conjunto de palabras binarias sea un código corrector de un error en un solo bit, es que la distancia mínima entre ellas sea de **tres**.

VALOR DECIMAL	M_7	M_6	M_5	C_4	M_3	C_2	C_1
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	0	1	1	0	0	1
3	0	0	1	1	1	1	0
4	0	1	0	1	0	1	0
5	0	1	0	1	1	0	1
6	0	1	1	0	0	1	1
7	0	1	1	0	1	0	0
8	1	0	0	1	0	1	1
9	1	0	0	1	1	0	0

Códigos correctores de error (II)

- Vamos a generar un código Hamming que detecte errores de un bit en los dígitos del 0 al 9. Según la relación **$2^p \geq n + p + 1$,** necesitamos un código base de 4 bits y añadimos 3 bits de comprobación, generando por tanto un código de 7 bits. Partiremos del código BCD natural y le añadiremos tres bits.

BIT ERRONEO	D_3	D_2	D_1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

- El bit D_1 ha de tomar el valor 1 si se produce un error en los bits b1, b3, b5 y b7 de la combinación del código. Si el número de unos existentes en esas cuatro posiciones es siempre par, un error en uno cualquiera de esos cuatro bits lo convierte en impar.
- Por tanto D_1 ha de valer uno si el número de unos en las posiciones b1, b3, b5 y b7, es impar, y cero en caso contrario.
- Utilizamos la función XOR en D1, D2, D3 que algebraicamente queda así:

  - **$D1 = B1 \oplus B3 \oplus B5 \oplus B7$**
  - **$D2 = B2 \oplus B3 \oplus B6 \oplus B7$**
  - **$D3 = B4 \oplus B5 \oplus B6 \oplus B7$**

Códigos correctores de error (III)

De las ecuaciones anteriores podemos extraer las siguientes. El bit B1 debe valer uno si el número de unos B3, B5 y B7 es impar y cero en caso contrario. Para los bits B1, B2 y B4 obtendríamos:

$$B1 = B3 \oplus B5 \oplus B7$$

$$B2 = B3 \oplus B6 \oplus B7$$

$$B4 = B5 \oplus B6 \oplus B7$$

Con estas ecuaciones obtenemos los bits B1, B2 y B4. El resto de bits está ocupado por el código original del que partimos (en este caso, el BCD natural)

VALOR DECIMAL	B_7	B_6	B_5	B_4	B_3	B_2	B_1
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	0	1	1	0	0	1
3	0	0	1	1	1	1	0
4	0	1	0	1	0	1	0
5	0	1	0	1	1	0	1
6	0	1	1	0	0	1	1
7	0	1	1	0	1	0	0
8	1	0	0	1	0	1	1
9	1	0	0	1	1	0	0

Códigos correctores de error (IV)

Un error en un bit lo detectamos mediante las ecuaciones de D1, D2 y D3. Supongamos que recibimos la combinación errónea **0011101**. Aplicando las ecuaciones de D1, D2 y D3 obtenemos:

D1 = B1 ⊕ B3 ⊕ B5 ⊕ B7 = 1 ⊕ 1 ⊕1 ⊕ 0 = 1
D2 = B2 ⊕ B3 ⊕ B6 ⊕ B7 = 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1
D3 = B4 ⊕ B5 ⊕ B6 ⊕ B7 = 0 ⊕ 0 ⊕ 1 ⊕ 1 = 0

Lo cual equivale a la combinación 011, es decir, el número 3. Este es el dígito que es erróneo. Basta cambiar su valor para corregir la combinación recibida. Por tanto, la combinación correcta será **0011001**, que es la que se muestra en la tabla.

VALOR DECIMAL	B_7	B_6	B_5	B_4	B_3	B_2	B_1
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	0	1	1	0	0	1
3	0	0	1	1	1	1	0
4	0	1	0	1	0	1	0
5	0	1	0	1	1	0	1
6	0	1	1	0	0	1	1
7	0	1	1	0	1	0	0
8	1	0	0	1	0	1	1
9	1	0	0	1	1	0	0