# TECNOLOGÍA ELECTRÓNICA DE COMPUTADORES

2º Curso – GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

BLOQUE III: CIRCUITOS ELECTRÓNICOS DIGITALES

Tema 5. Circuitos integrados digitales: familias lógicas.

Lección 6. Introducción a los sistemas digitales.



## Lección 6. Introducción a los sistemas digitales.

- 6.1 Sistemas analógicos y digitales
- 6.2 Representación de valores digitales: el sistema base dos
- 6.3 Códigos binarios. Binario natural, BCD y Gray



## Bibliografía de la lección

#### **Lectura clave:**

Thomas L.Floyd. Fundamentos de sistemas digitales.

Ed. Prentice Hall - Pearson Education.

Introducción: Tema 1. Conceptos digitales, apartados 1.1. y 1.2

Tema 2. Sistemas de numeración, operaciones y códigos

#### Otros:

Enlaces a características de circuitos integrados digitales de vendedores o fabricantes

#### **Ejemplos:**

- Serie 74xxx http://www.futurlec.com/IC74Series.shtml
- Serie 40xxx http://www.futurlec.com/IC4000Series.shtml
- NXP Serie 74HC/T: <a href="http://ics.nxp.com/products/hc/all/">http://ics.nxp.com/products/hc/all/</a>

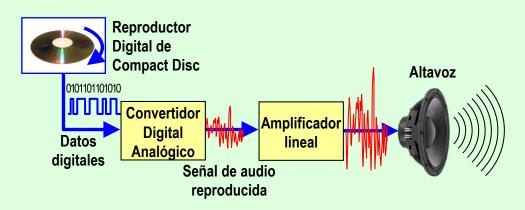


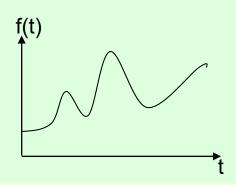
## 6.1 Sistemas Digitales.

#### Sistema Analógico

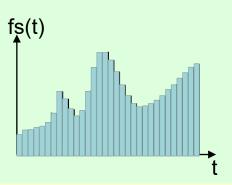


#### **Sistema Digital**









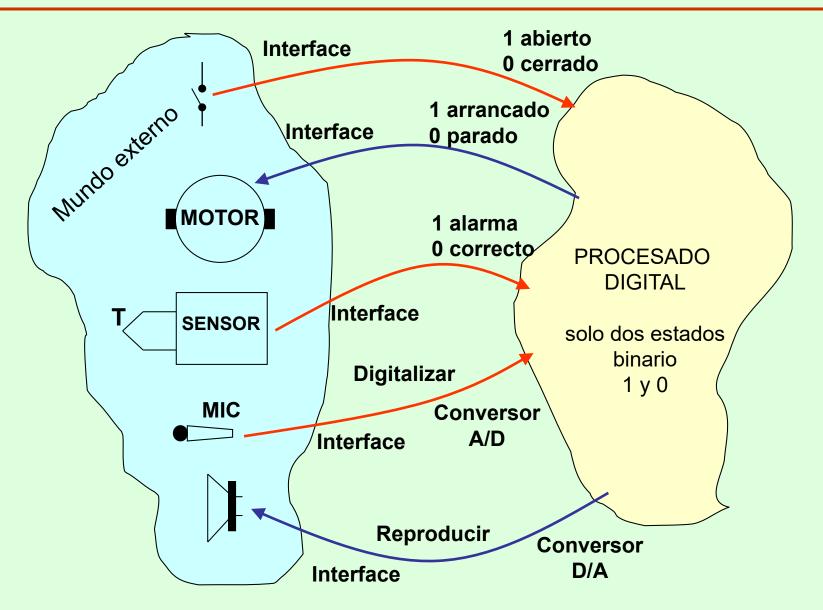
#### SEÑAL ANALÓGICA:

- Cualquier valor posible dentro de un rango
- Los valores son números reales

#### SEÑAL DIGITALIZADA:

- Conjunto de valores posibles discreto
- Se representa con códigos binarios



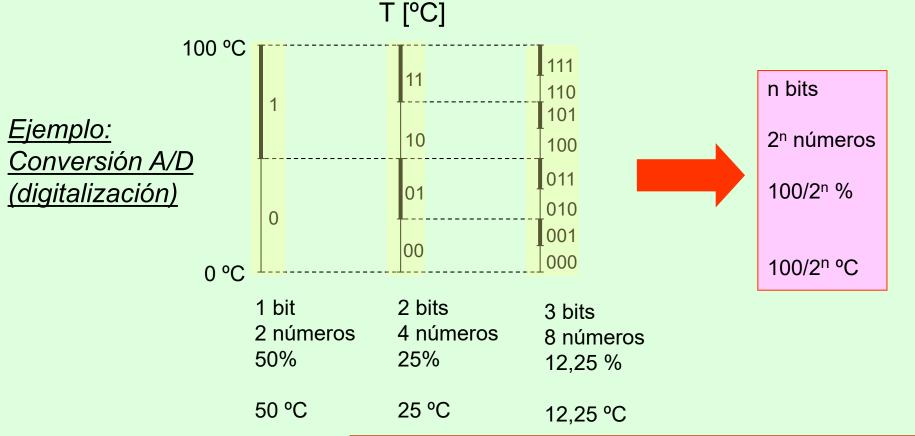




#### Conversiones A/D y D/A

SEÑAL ANALÓGICA: Cualquier valor dentro de un rango SEÑAL DIGITALIZADA: Se representa con códigos binarios

La correspondencia depende de los criterios usados al convertir: número de bits, etc





En Electrónica Digital TODO se codifica en binario (tensiones, temperaturas, caracteres, posiciones de máquinas, etc)

Importante conocer 2 cosas:

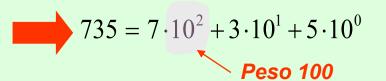
- CODIGO BINARIO (y todos su códigos derivados)
- HERRAMIENTA MATEMÁTICA: Álgebra de Boole (George Boole 1854)

Aplicada por Shannon a la Electrónica

Digital en 1938 (Tesis Doctoral MIT)

#### **DECIMAL VERSUS BINARIO**

Numero decimal (Base 10)



Dígitos:

0123456789

Numero binario (Base 2)

$$101 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Peso 4

Dígitos: 0 1

**NOTA**: Se utilizan también otras bases (p.e. Hexadecimal para simplificar las notaciones)



#### Interpretación digital de las señales eléctricas/electrónicas

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

Los valores 0 y 1 se corresponderán con los valores de magnitudes eléctricas, o relacionadas con ellas.

De forma simplificada, pueden ser:

#### Tensión:

1 Hay tensión 0 No hay tensión

#### Corriente:

1 Hay corriente 0 No hay corriente

#### Estado de Interruptores (Transistores)

1 Interruptor cerrado (Transistor en condución, saturado)

0 Interruptor Abierto (Transistor cortado)

Los componentes electrónicos fundamentales en Electrónica Digital son los transistores.

Normalmente los "unos" y los "ceros" se interpretan en tensión:

Ejemplo: "1" = +5 V "0" = 0 V



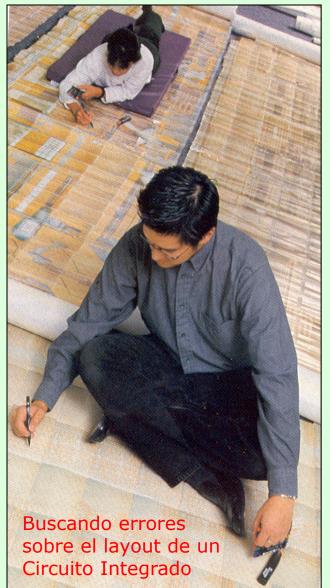
#### Formas de onda digital: Ejemplo de tensión en un punto del circuito



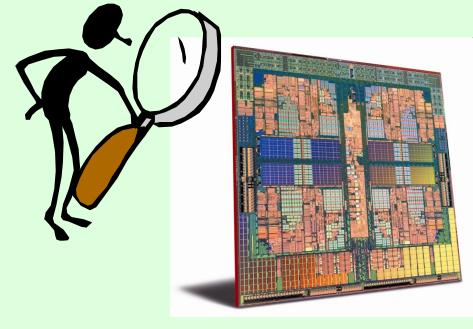
Los elementos de la Electrónica Digital son muy simples, pero agregados dan lugar a circuitos muy complejos:

- Utilizando transistores se realizan los bloques SSI básicos (puertas lógicas).
- Utilizando puertas se hacen bloques más complejos (Codificadores, ALU, Biestables), llamados MSI (Medium Scale Integration).
- Utilizando Bloques MSI se hacen bloques de mayor complejidad, llamados LSI (LArge Scale Integration) como PLDs, MPUs, etc
- La complejidad puede aumentarse dando lugar a los bloques VLSI y VHLSI, para constuir DSPs, o MCUs de mayor potencia.





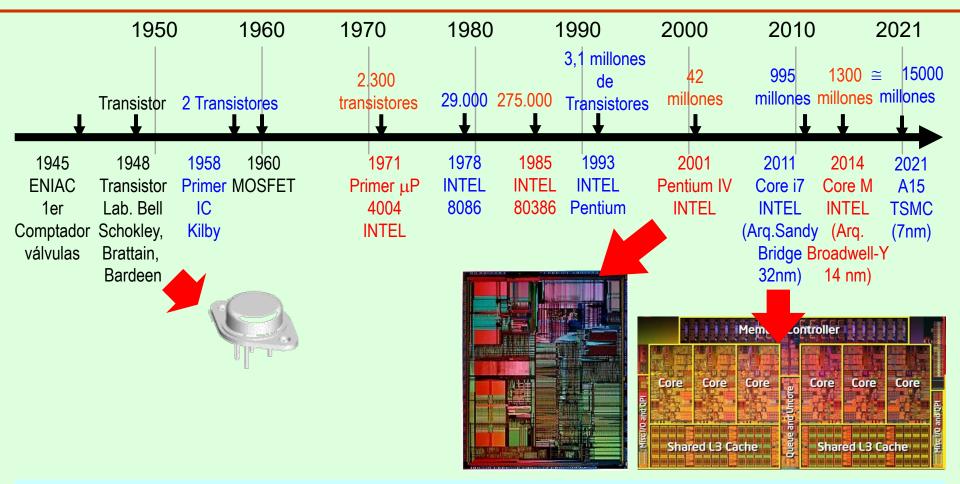
LOS CIRCUITOS INTEGRADOS PUEDEN LLEGAR A TENER MUCHÍSIMOS COMPONENTES REALIZADOS SOBRE EL MISMO CIRCUITO INTEGRADO





#### Tecnología Electrónica

#### Área de Tecnología Electrónica



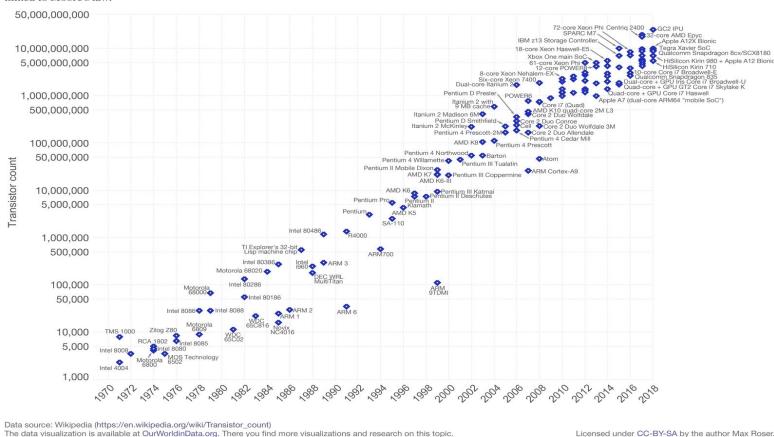
- Evolución del tamaño de los circuitos integrados espectacular
- Hoy (2021): 55.000 millones de transistores en un microprocesador
- Tamaños ya muy cerca del límite estimado por aparecer efectos cuánticos (actualmente en torno a los 5 nm), con anchos de canal de unos pocos átomos



#### Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

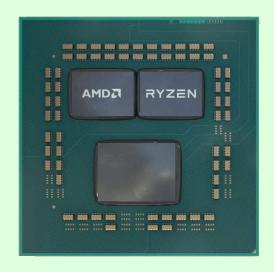


Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



- La Ley de Moore indica que el número de transistores se duplica cada dos años (aproximadamente)
- Actualmente se ha reducido el ritmo y se duplica cada dos años y medio















Procesadores móviles 5nm (TSMC)

#### Hoy (2021):

- Tecnología de 5 nm en producción para memorias (TSMC/Samsung). Para 2021: 4nm
- Microprocesadores: AMD (TSMC) Ryzen y Epyc (7nm) con arquitectura Zen 3. Para 2022: 5nm (Zen 4) para EPYC y Ryzen 7000. En desarrollo: Zen 5 (3 nm) para Ryzen 8000.
- Procesadores móviles (2020): Apple A15 Bionic (5 nm) para iPhone13, Qualcomm Snapdragon 888, Huawei Kirin 9000 (fabricado por TSMC, 5nm) y Samsung Exynos 1080 (Samsung, 5nm). Año 2022: 4 nm/ 3 nm (Samsung y TSMC)



# 6.2. Representación de valores digitales: el sistema base dos

#### Introducción:

El sistema decimal es un sistema posicional, en el que cada dígito de un número tiene un peso asignado en potencias de 10.

Ejemplo: 
$$7134 = 7*10^3 + 1*10^2 + 3*10^1 + 4*10^0$$

NO ES EL FORMATO ÓPTIMO PARA MANEJAR, ALMACENAR, O MODIFICAR DE CUALQUIER FORMA LA INFORMACIÓN, USANDO CIRCUITOS ELECTRÓNICOS



Los transistores (BIPOLARES y MOS) tienen dos zonas de trabajo claramente diferenciadas e inequívocas:

SATURACIÓN / ZONA RESISTIVA: INTERRUPTOR CERRADO

Y CORTE (INTERRUPTOR ABIERTO)

INTERESA UN CÓDIGO QUE PUEDA IMPLEMENTARSE CON DOS ESTADOS SOLAMENTE



## **Código binario:**

Es un código posicional en el que la base es el número 2; se consigue reducir el número de estados a diferenciar a dos: 0 y 1

Ejemplo: 
$$1(0)10 = 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0 = 8 + 0 + 2 + 0 = 10$$

Cada dígito (BIT) sólo puede tomar los valores 0 ó 1

SE PUEDE IMPLEMENTAR CON TRANSISTORES TRABAJANDO EN:

CORTE (INTERRUPTOR ABIERTO)

ó

- SATURACIÓN / RESISTENCIA CASI NULA (INTERRUPTOR CERRADO)



#### Códigos octal y hexadecimal:

Códigos en base 8 y base 16, además de poder verse como una simplificación de escritura del código binario

En octal (base 8), d<sub>i</sub> ∈ [0,7], cada dígito toma 8 valores diferentes: entre 0 y 7

Ejemplo:  $5470_8 = 5*8^3 + 4*8^2 + 7*8^1 + 0*8^0 = 2872_{10}$ 

En hexadecimal (base 16), cada dígito puede tomar 16 valores diferentes: toma valores entre 0 y 15; se hace preciso distinguir de alguna forma los dígitos que tienen dos cifras decimales

Ejemplo:

¿ Es "uno" y "cinco" o es "quince"?



Solución: los dígitos a partir del 10 (inclusive) se denominan con letras: A, B, C, D, E y F

Ejemplo:  $5470_{16} = 5*16^3 + 4*16^2 + 7*16^1 + 0*16^0 = 21616_{10}$ 



## Tecnología Electrónica

Los códigos octal y hexadecimal pueden verse como una simplificación de escritura del binario:

Decimal <sub>10</sub>	Binario <sub>2</sub>	Octal <sub>8</sub>	Hexadecimal <sub>16</sub>
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100 <sub>De</sub>	3 en 3 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	Α
11	1011	13	В
12	1100	14	С
13	1101	15	D
14	1110	16	E
15	1111	17	F



#### Tecnología Electrónica

Los códigos octal y hexadecimal pueden verse como una simplificación de escritura del binario:

Decimal <sub>10</sub>	Binario <sub>2</sub>	Octal <sub>8</sub>	Hexadecimal <sub>16</sub>
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04 De 4	en 4 4
5	0101	05	<b>—</b> 5
6	0110	06 Abre	U
7	0111	Λ7	ás a 7
8	1000		itura 8
9	1001	11	9
10	1010	12	Α
11	1011	13	В
12	1100	14	С
13	1101	15	D
14	1110	16	E
15	1111	17	F



## Operando con códigos binario, octal y hexadecimal.

Interesa conocer:

- Como convertir desde una base cualquiera a decimal
- Como convertir desde decimal a una base cualquiera

#### Conversión de un número desde una base cualquiera (b) a decimal:

Sea

$$Q = (d_n^*b^n) + (d_{n-1}^*b^{n-1}) + \dots + (d_1^*b^1) + (d_0^*b^0) + (d_{-1}^*b^{-1}) + \dots + (d_{-m}^*b^{-m})$$

b (Base), d<sub>i</sub> € [0, b-1]

A derecha e izquierda de la coma decimal

En binario (base 2):  $d_i \in [0,1]$ 

$$Q = 1*2^3 + 1*2^1 + 1*2^0 + 1*2^{-2} = 11,25$$

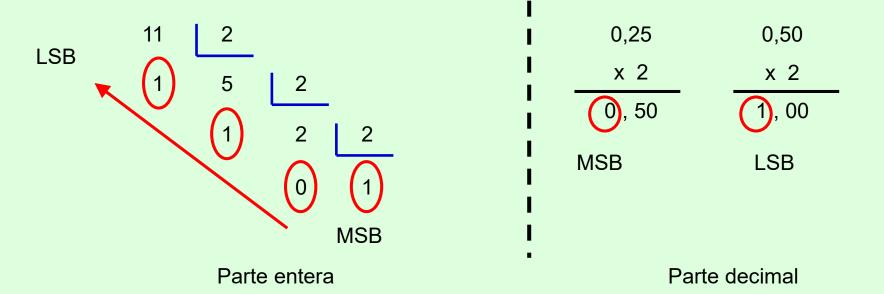


## Conversión desde decimal a una base (b) cualquiera:

Para obtener los distintos d<sub>i</sub> que forman la parte entera del número, se realizan sucesivas divisiones por la base.

Para obtener los distintos d<sub>i</sub> que forman la parte decimal del número, se realizan sucesivas multiplicaciones por la base.

Ejemplo:  $11,25_{10} = 1011,01_2$ 





#### Resumen: ¿Cómo convertir de una base a otra?

- Lo más sencillo es trabajar con decimal (al que estamos habituados) y binario.
- Las otras bases permiten "compactar" la notación respecto al binario

#### Convertir BINARIO → DECIMAL

BINARIO: digitos binarios

► 2<sup>6</sup> 2<sup>5</sup> 2<sup>4</sup> 2<sup>3</sup> 2<sup>2</sup> 2<sup>1</sup> 2<sup>0</sup>

Valor posicional:

Valor en decimal: sumamos  $\longrightarrow$  2<sup>6</sup>+ 0 + 0 + 2<sup>3</sup>+ 2<sup>2</sup> +0+ 2<sup>0</sup> = 64+8+4+1=77

SUMA [(digito) · (valor posicional)]

#### Convertir DECIMAL → BINARIO

Opción 1 (valores no muy grandes): realizar la operación anterior "al revés" buscando descomponer el número en suma de potencias de 2, que son valores conocidos:

1, 2, 4, 8, 16, 32, 64,128, 256, etc Ejemplo:  $20 = 16 + 4 = 2^4 + 2^2 = 10100$ 

Opción 2: método de la división sucesiva por 2

- Se divide hasta que el cociente sea 0.
- Los restos porporcionan los digitos, de menor a mayor peso

Ejemplo: 
$$20 \ \underline{2}$$
  $0 \ 10 \ \underline{2}$   $0 \ 5 \ \underline{2}$   $0 \ 1 \ \underline{2}$ 



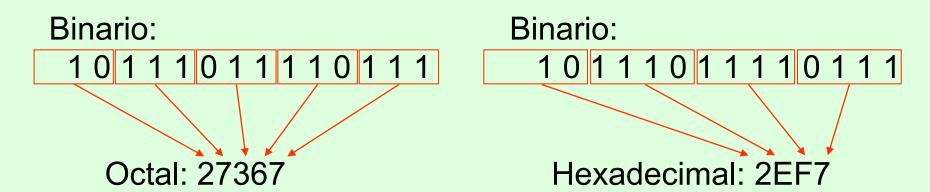
## Resumen de nomenclatura y códigos interesantes para simplificar las notaciones

MSB (bit más significativo)

LSB (bit menos significativo)

Binario: 1011101111

#### LOS CÓDIGOS OCTAL Y HEXADECIMAL PERMITEN ABREVIAR LAS NOTACIONES



En especial, el código Hexadecimal está muy extendido en el mundo de los MPU y MCU



## Resumen de nomenclatura y códigos interesantes para simplificar las notaciones

#### Más nomenclatura

BIT = 1

BYTE = 8 BITs = 11011110

NIBBLE = 4 bits = 1101

WORD (Palabra) = 16 bits = 1001 1001 1110 0011 = 99E3 (El hexadecimal es muy útil).

LONG WORD (Palabra larga) = 32 bits , 64 bits y 128 bits (Se suele emplear también palabra de 32 bits y palabra de 64 bits) (en ingles 32-bit-word 64-bit-word)

Obviamente el hexadecimal es también muy útil para trabajar con tiras de bits tan largas.



## Diferentes códigos basados en el binario:

- Código Decimal codificado en Binario (BCD). Representa cada dígito decimal (0 a 9) mediante un código binario de 4 bits.
- Códigos autocomplementados (BCD exceso-3: El complemento a 1 de un número en exceso-3 coincide con el código de exceso-3 del complemento a 9 del número)
- Códigos progresivos
- Códigos de carácter
- Códigos detectores de error:
  - Paridad par
  - Paridad impar
- Codigos correctores de error

Utilizados en transmisiones de datos digitales



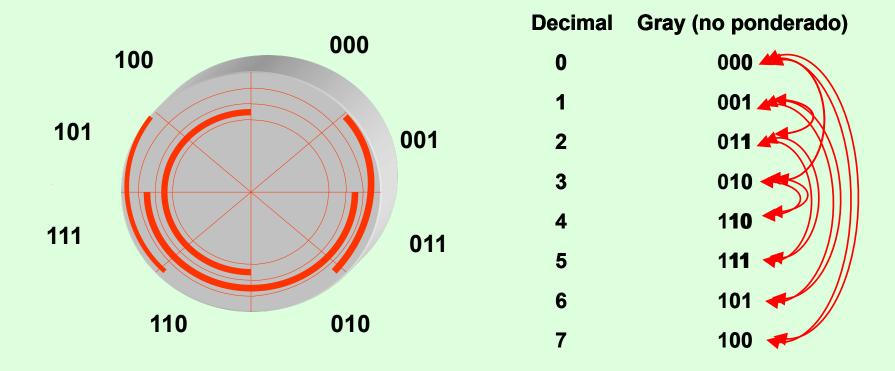
## **Códigos BCD: variantes**

	BCD (8421)		BCD exceso-3		<u>2</u>	<u>421 (Aitken)</u>
0	0000		0	0011	0	0000
1	0001		1	0100	1	0001
2	0010		2	0101	2	0010
3	0011		3	0110	3	0011
4	0100		4	0111	4	0100
5	0101		5	1000	5	1011
6	0110		6	1001	6	1100
7	0111		7	1010	7	1101
8	1000		8	1011	8	1110
9	1001		9	1100	9	1111
	Ponderado (con pesos)	· · · · · · · · · · · · · · · · · · ·		•	Ponderado, autocomplementado	



## Códigos progresivos:

- Sólo cambia un bit de una combinación a otra.
- Útiles para codificar posiciones (la posibilidad de error crece con el número de bits que cambian entre posiciones adyacentes).





## Códigos de carácter:

Representan números (los diez dígitos decimales), letras (26 letras del alfabeto) y otros símbolos/instrucciones.

Ejemplo: Código ASCII de 128 caracteres (American Standard Code Information Interchange)

J=01001010

@=01000000

retorno de carro (CR)= 0001101

Código ASCII extendido con 128 caracteres adicionales. Adoptado inicialmente por IBM en sus PCs, es un estándar "no oficial".



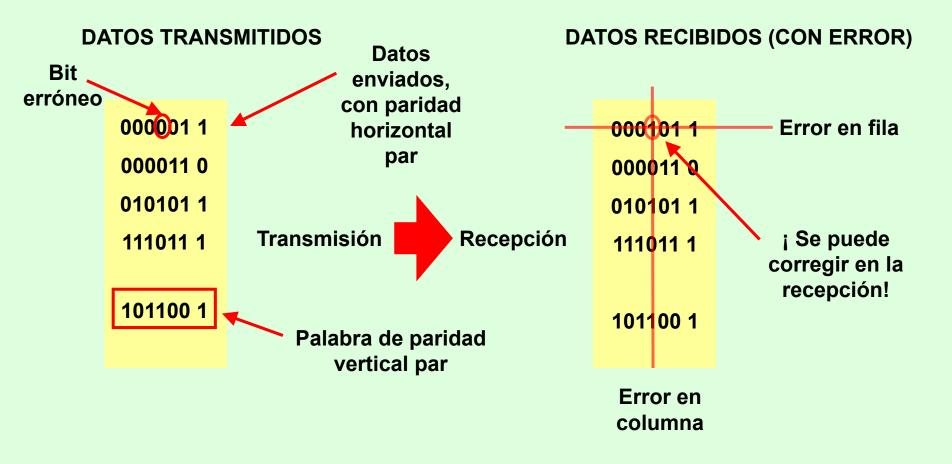
## Códigos detectores de error – bits de paridad

BCD		BCD I	paridad par	BCD par	BCD paridad impar	
0	0000	0000	0	0000	1	
1	0001	0001	1	0001	0	
2	0010	0010	1	0010	0	
3	0011	0011	0	0011	1	
4	0100	0100	1	0100	0	
5	0101	0101	0	0101	1	
6	0110	0110	0	0110	1	
7	0111	0111	1	0111	0	
8	1000	1000	1	1000	0	
9	1001	1001	0	1001	1	



#### Códigos correctores de error

La paridad simple detecta pero no corrige; se hace preciso acudir a la Paridad entrelazada.





## Códigos correctores de error

Existen más códigos detectores y correctores de error:

- Códigos de Hamming
- Códigos de CRC (Redundancia cíclica)
- Códigos m de n
- Y muchos más

