TEMA 5.1

FUNDAMENTOS DE ELECTRÓNICA DIGITAL

CUD

TEMA 5
SISTEMAS DIGITALES

FUNDAMENTOS DE ELECTRÓNICA

Centro Universitario de la Defensa Zaragoza

19 de febrero de 2015

TEMA 5.1 - FUNDAMENTOS

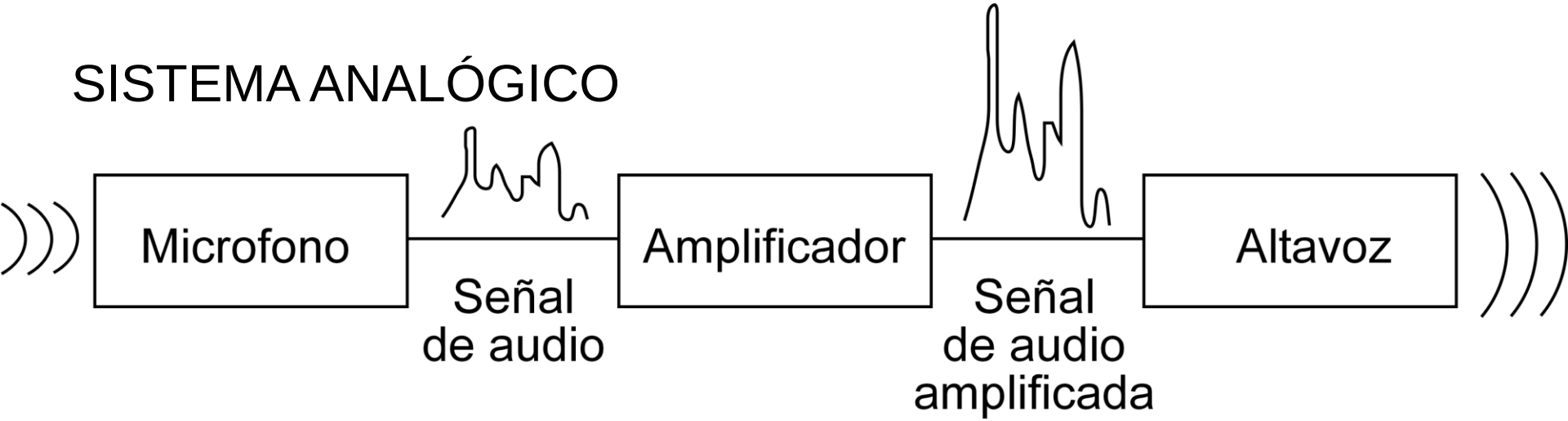
- Introducción a la Electrónica Digital
- Códigos binarios
- Aritmética binaria
- Algebra de Boole

TEMA 5.1 - FUNDAMENTOS

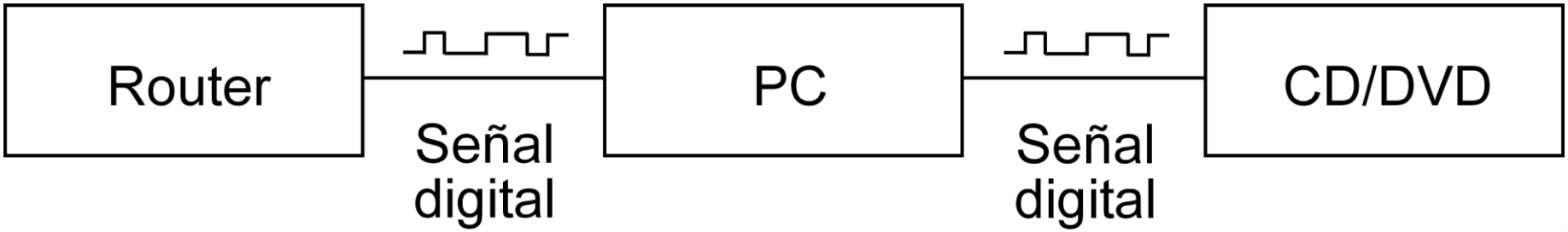
- Introducción a la Electrónica Digital
- Códigos binarios
- Aritmética binaria
- Algebra de Boole

ANALÓGICA vs DIGITAL

SISTEMA ANALÓGICO

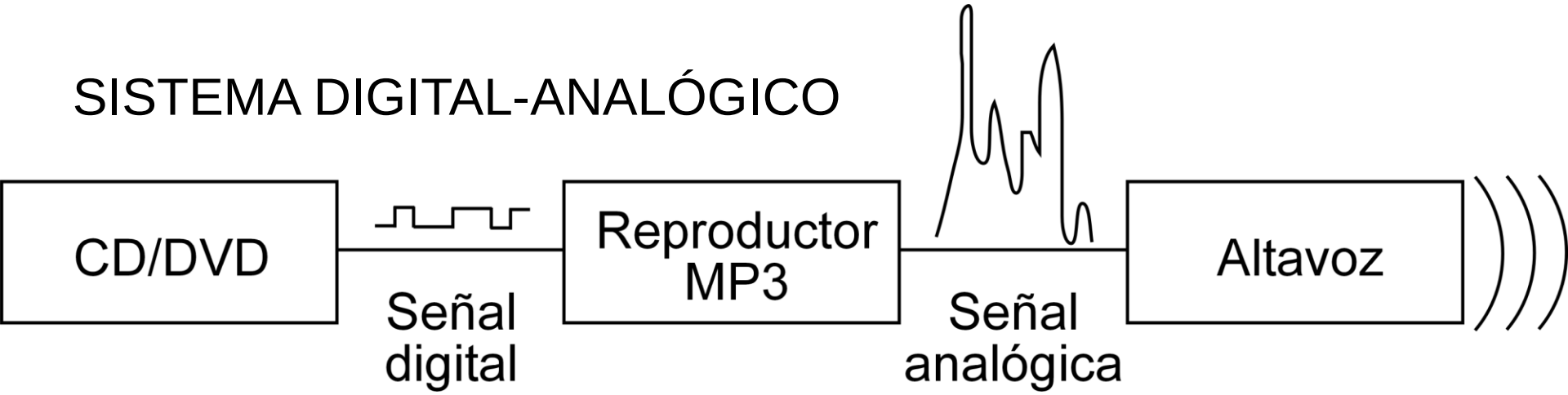


SISTEMA DIGITAL

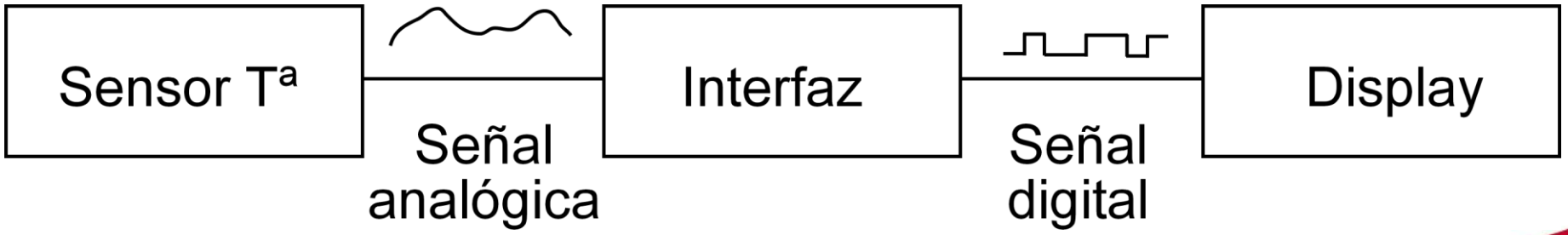


ANALÓGICA vs DIGITAL

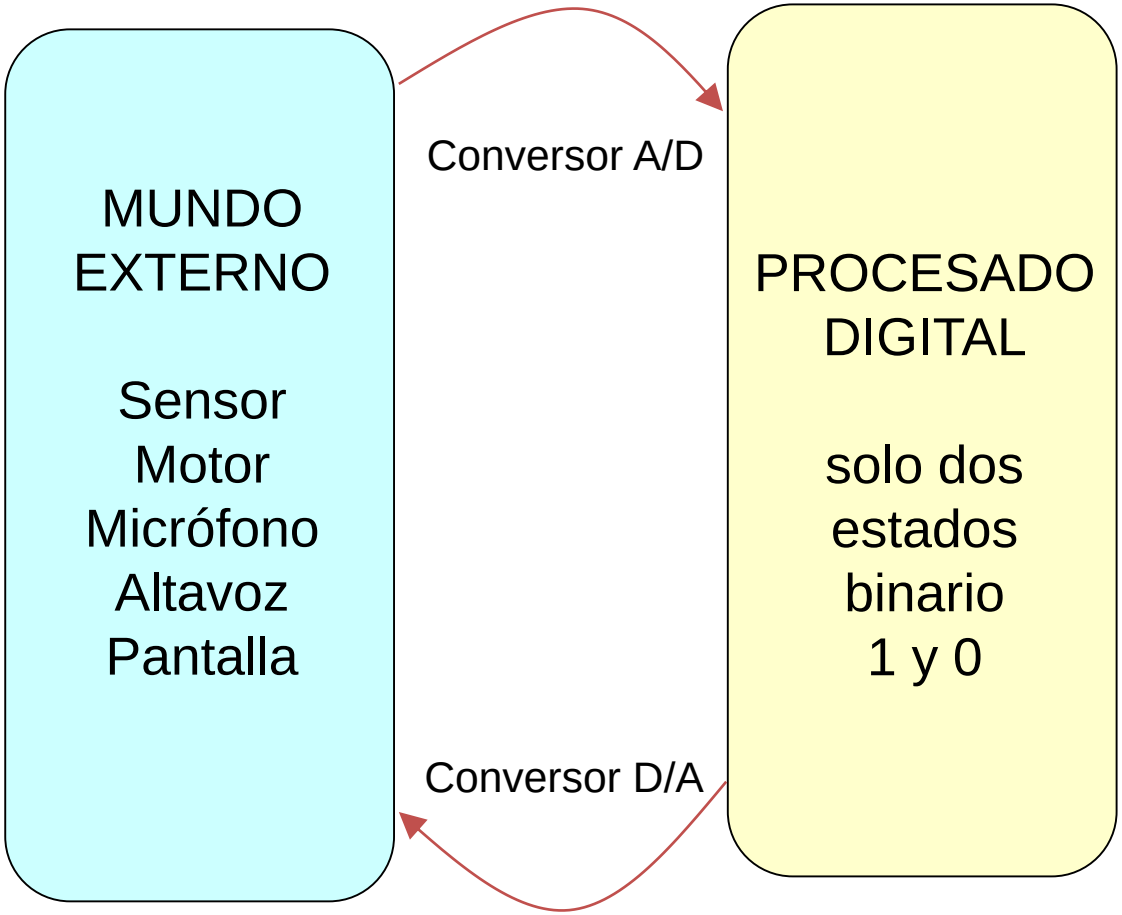
SISTEMA DIGITAL-ANALÓGICO



SISTEMA ANALÓGICO-DIGITAL



DIGITALIZACIÓN



La ventaja fundamental es la inmunidad al ruido

RESOLUCIÓN

T [0 – 100 ºC]

Analógica	1 bit	2 bits	3 bits
0 – 5V	1 (5V)	11	111
			110
		10	101
			100
	0 (0V)	01	011
			010
		00	001
			000

n bits

2^n números

Intervalo $100/2^n$

$T \propto V$

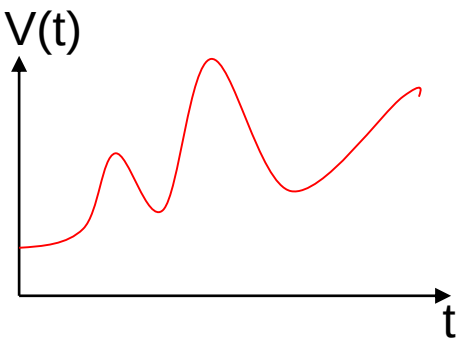
1 bit
2 números
50 ºC

2 bits
4 números
25 ºC

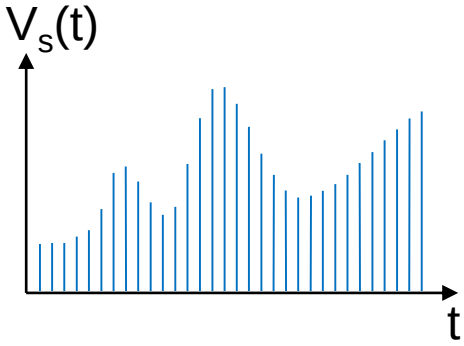
3 bits
8 números
12,25 ºC

CONVERSIÓN A/D Y D/A

SEÑAL ANALÓGICA



SEÑAL DIGITALIZADA

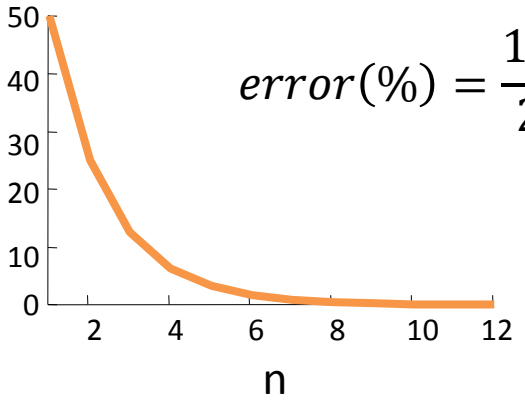


PRIMERA PREGUNTA CLAVE:

¿Cuantos bits necesito para digitalizar la señal?

El número de bits (n) utilizados nos define el error:

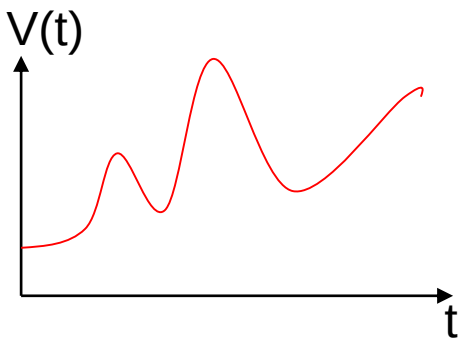
error (%)



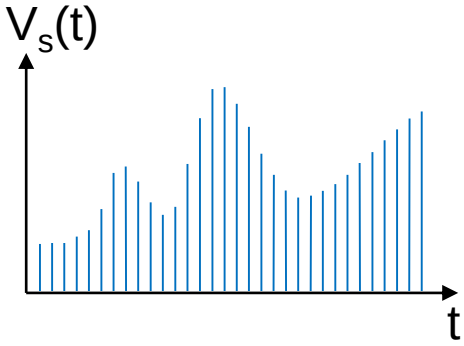
$$error(\%) = \frac{100}{2^n}$$

CONVERSIÓN A/D Y D/A

SEÑAL ANALÓGICA



SEÑAL DIGITALIZADA



SEGUNDA PREGUNTA CLAVE:

¿Cada cuanto tiempo muestreo?

El ancho de banda (B) de la señal f(t) nos define la frecuencia de muestreo.

F muestreo $\geq$ 2 B

Teorema de NYQUIST

EJEMPLO: Música en fichero .WAV

La música se muestrea 44.100 veces en un segundo (44.1 KHz) y se emplean 16 bits.

Se toman muestras separadas en el canal izquierdo y en el derecho (estéreo).

TEMA 5.1 - FUNDAMENTOS

- Introducción a la Electrónica Digital
- Códigos binarios
- Aritmética binaria
- Algebra de Boole

DECIMAL vs BINARIO

Numero decimal
(Base 10)

$$735 = 7 \cdot 10^2 + 3 \cdot 10^1 + 5 \cdot 10^0$$

Dígitos:
0 1 2 3 4 5 6 7 8 9

Peso 100

Conversión binario a decimal

Numero binario
(Base 2)

$$101 \Rightarrow 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$$

Dígitos:
0 1

Peso 4

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

CONVERSIÓN DE DECIMAL A BINARIO

- "División Repetida", esta manera de conversión se basa en repetir la división del número decimal entre dos, hasta llegar al cero. Si el residuo de la división no es un número entero, se marca un 1 y se toma el número entero para volver a dividir entre dos, cuando el residuo es un número entero, se marca un cero y se toma el número para volver a dividir entre dos. El residuo de la primero división es el (LSB, primer Bit), el residuo de la última división es el (MSB, último Bit). Esto se ilustra así:

150 / 2 = 75	(número entero)	LSB,	**Primer**	Bit = 0
75 / 2 = 37.5	(número con decimal)		**Segundo**	Bit = 1
37 / 2 = 18.5	(número con decimal)		**Tercer**	Bit = 1
18 / 2 = 9	(número entero)		**Cuarto**	Bit = 0
9 / 2 = 4.5	(número con decimal)		**Quinto**	Bit = 1
4 / 2 = 2	(número entero)		**Sexto**	Bit = 0
2 / 2 = 1	(número entero)		**Séptimo**	Bit = 0
1 / 2 = 0.5	(número con decimal)	MSB,	**Octavo**	Bit = 1

Binario: 10010110 ← LSB (menos significativo)

MSB (más significativo)

MAS NOMENCLATURA

- BIT = 1
- NIBBLE = 4 bits = 1101
- BYTE = 8 BITs = 11011110
- WORD (Palabra) = 16 bits = 1001 1001 1110 0011 = 99E3
- 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)
- El hexadecimal es muy útil para trabajar con tiras de bits largas.

Decimal	Hex.	Binario
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

CÓDIGO BCD

- Ahora ya sabemos que los números del sistema decimal tienen equivalentes en el sistema binario. La agrupación ordenada de los 0 y 1 de un número binario representa algún número decimal.
- Los sistemas digitales utilizan por fuerza los números en sistema binario, pero para nosotros en el mundo real siempre tienen que ser convertidos al sistema decimal, como hemos visto, las conversiones entre uno y otro sistema de números pueden llevarnos demasiado tiempo y ser muy complicadas, por ejemplo, si usamos números muy grandes. Para este tipo de conversiones y usos, se utiliza un método sencillo que combina las características de los sistemas decimal y binario, este método lleva el nombre de **Codificación Binaria Directa**.

CÓDIGO BCD

- Cuando tomamos cada uno de los dígitos del sistema decimal, y lo representamos con su equivalente del sistema binario, estamos generando un “nuevo” código, el cuál lleva el nombre de **Código Decimal Codificado en Binario (BCD)**.
- Partiendo de este nuevo código, el mayor número que podemos representar es el 9 (1001), por lo tanto forzosamente necesitamos de un número binario de 4 bits para hacerlo. Pero veamos gráficamente que es y como funciona el BCD.
- En esta ocasión usaremos los números decimales 586 y 397, el proceso de convertir cada dígito por un equivalente binario sería el siguiente:

Número decimal	5	8	6
BCD	0101	1000	0110
Número decimal	**1**	**0**	**4**
BCD	0001	0000	0100

CÓDIGO BCD

- Cada uno de los dígitos del número decimal es convertido en su equivalente binario, siempre utilizando 4 bits para este proceso. En resumen, el código BCD representa por separado cada uno de los numerales decimales, empleando para ello números binarios de 4 bits.
- Como es lógico, si sólo se puede representar un solo número decimal por cada código BCD, los números del 10 al 15 (que es el número decimal más alto para un código binario de 4 bits, 1111), están fuera del código, de hecho, si tenemos algún circuito digital que trabaja sobre código BCD y nos diera una salida como las siguientes, algo no está funcionando bien:

Decimal 10 = Binario 1010
Decimal 11 = Binario 1011
Decimal 12 = Binario 1100
Decimal 13 = Binario 1101
Decimal 14 = Binario 1110
Decimal 15 = Binario 1111

SISTEMA BINARIO vs CÓDIGO BCD

- Como el nombre lo indica, el Código BCD no puede ser catalogado como un sistema (como el binario, octal y hex). Sólo es una forma de codificar el sistema binario.

- Teniendo muy presente este hecho, un número en código BCD, **NO** es lo mismo que un número binario directo. El código BCD toma cada uno de los dígitos de un número decimal y los representa. Un número del sistema binario representa el número decimal completo. Para comprender mejor el concepto, usaremos el número decimal 387.

SISTEMA BINARIO vs CÓDIGO BCD

Tabla de conversión al Sistema Binario

Número Decimal	256	128	64	32	16	8	4	2	1
387	1	1	0	0	0	0	0	1	1

$$1 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 387$$

Tabla de conversión al Código BCD

Número Decimal	8	4	2	1	8	4	2	1	8	4	2	1
387	0	0	1	1	1	0	0	0	0	1	1	1

3 8 7

PARIDAD

- Se utiliza en transmisiones digitales con el objeto de detectar errores en la misma
- Se incluye un bit adicional que establece si el número de unos es par (paridad par) o impar (paridad impar)
- Por ejemplo, si se quiere transmitir el número 72 en código BCD con paridad impar: 011100101
  - 7: 0111
  - 2: 0010
  - Bit de paridad: 1
- Otro ejemplo, el número +24 en complemento a 2 en 8 bits con paridad par: 000110000
  - +24: 00011000
  - Bit de paridad: 0

PARIDAD

- Si en las secuencias anteriores no se produce error en la transmisión, el número de unos se corresponde con la paridad:
  - 011100101 con paridad impar. OK
  - 000110000 con paridad par. OK
- Si se produce un error (un uno es considerado 0, o un cero es considerado 1):
  - 011110101 con paridad impar. ERROR
  - 000110010 con paridad par. ERROR
- Si se producen dos errores, este sistema no detecta el error, pero la probabilidad de que se produzca un error es pequeña, así que la probabilidad de que se produzcan dos errores en la misma secuencia se puede considerar nula

CÓDIGOS DE DETECCIÓN DE ERROR

- Existen más códigos detectores y correctores de error:
  - Códigos de Hamming
  - Códigos de CRC (Redundancia cíclica)
  - Códigos m de n
  - ....

TEMA 5.1 - FUNDAMENTOS

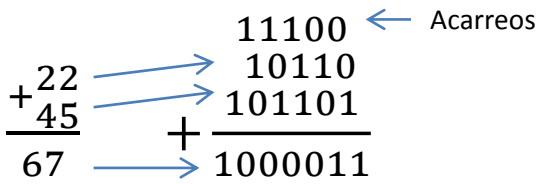
- Introducción a la Electrónica Digital
- Códigos binarios
- Aritmética binaria
- Algebra de Boole

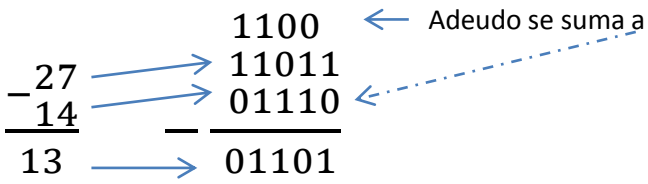
ARITMÉTICA BINARIA

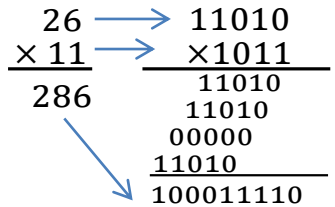
Suma	Acarreo
0+0 = 0	0
0+1 = 1	0
1+0 = 1	0
1+1 = 0	1

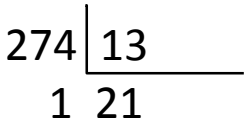
Resta	Adeudo
0-0 = 0	0
0-1 = 1	1
1-0 = 1	0
1-1 = 0	0

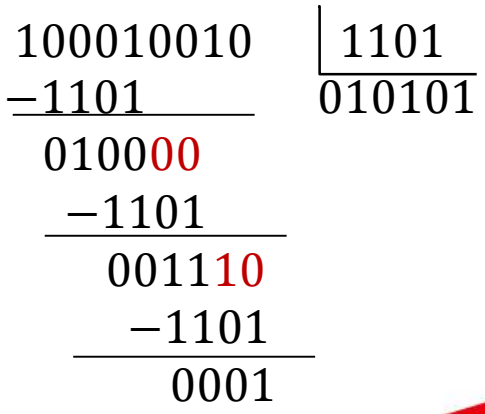
Producto
0x0 = 0
0x1 = 0
1x0 = 0
1x1 = 1





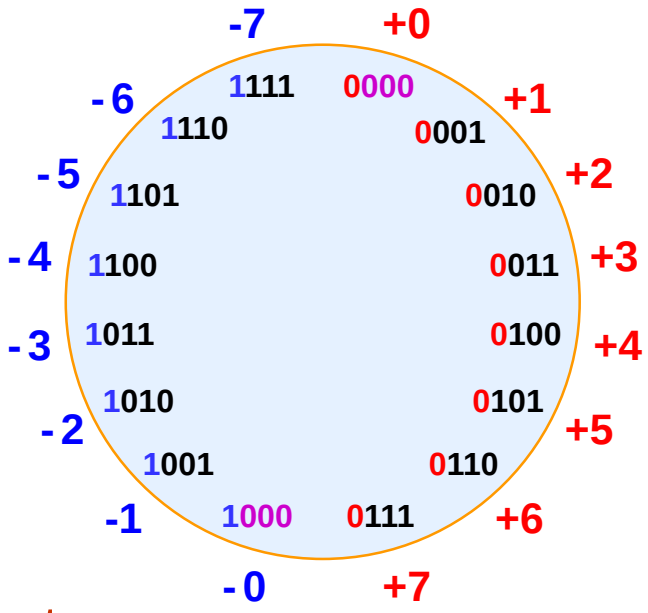






NÚMEROS NEGATIVOS. SM

- Signo y magnitud independientes
  - signo : bit más significativo (N-1)
    - + → 0
    - - → 1
  - magnitud : resto de bits (N-2 → 0)



- Rango : $-(2^{N-1}-1)$ a $2^{N-1}-1$
  - 0 → dos representaciones (0000 y 1000)

- Dificultad de realizar operaciones de resta
  - comparadores para signo del resultado
  - lógica adicional

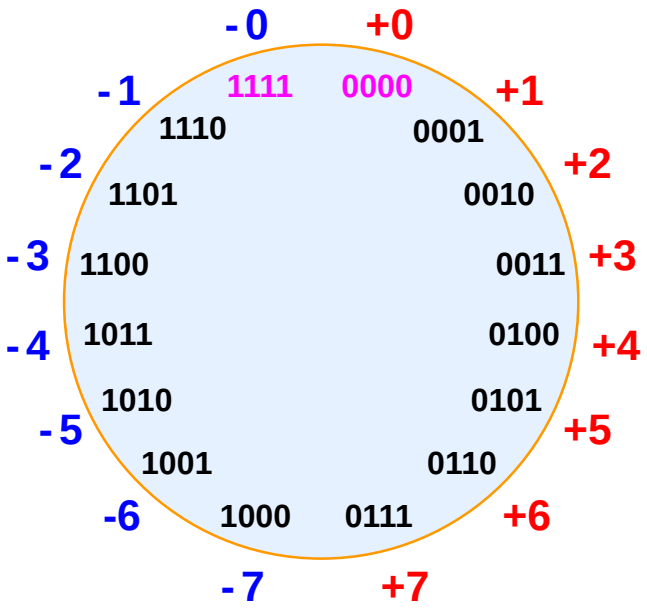
- Su utilización es limitada

NÚMEROS NEGATIVOS. Ca1

- Módulo : $2^N - 1$
- Números negativos

$$-X = C_1(X) = (2^N - 1) - X$$

- Rango : $-(2^{N-1} - 1)$ a $2^{N-1} - 1$
  - **0** → dos representaciones (0000 y 1111)
- Cálculo muy simple
  - $2^N - 1 = 111 \ldots 111 \Rightarrow 0 \leftrightarrows 1$
  - lógica elemental : inversores
- No se utiliza habitualmente

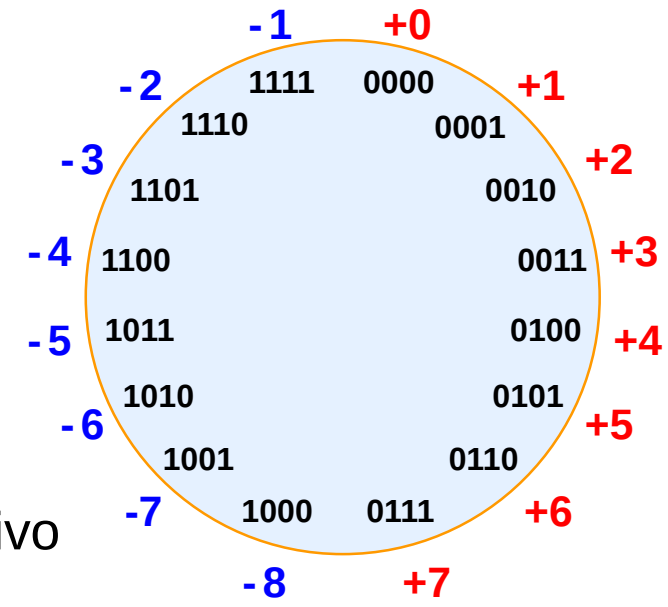


NÚMEROS NEGATIVOS. Ca2

- Módulo : 2^N
- Números negativos :

$$-X = C_2(X) = 2^N - X$$

- Rango : -2^{N-1} a $2^{N-1}-1$
  - representación única para el cero (0000)
- Signo : "coincide" con el bit más significativo
  - $0 \rightarrow +$
  - $1 \rightarrow -$
- $C_2(X) = C_1(X) + 1$
- Representación más utilizada



NÚMEROS NEGATIVOS. Ca2

- El complemento a dos de un número también se puede obtener de forma equivalente como:

  - Se complementan todos los bits, y al resultado de esta complementación, considerando como un número binario sin signo, sumarle 1.

  - Buscar el 1 menos significativo del número a complementar, dejar sin modificar ese 1 y todos los ceros a su derecha, y complementar todos los bits a su izquierda.

OPERACIONES CON Ca2

- $C_2(C_2(X)) = -(-X) = X$
- Signo incluido en los operandos
  - suma/resta es la misma operación
- Sumas aritméticas con N bits
  - se ignora el carry de salida (módulo 2^N)
- $X, Y > 0 \qquad Z = X - Y = X + C_2(Y) = X + (2^N - Y) = 2^N + (X - Y)$

  $X > Y \Rightarrow Z = 2^N + X - Y = X - Y$

  $X < Y \Rightarrow Z = 2^N - (Y - X) = C_2(Y - X) = -(Y - X) = X - Y$
- $X, Y > 0 \qquad Z = -X - Y = C_2(X) + C_2(Y) = (2^N - X) + (2^N - Y) =$

  $= 2^N + 2^N - (X + Y) = 2^N + C_2(X + Y) = -(X + Y)$

OPERACIONES CON Ca2

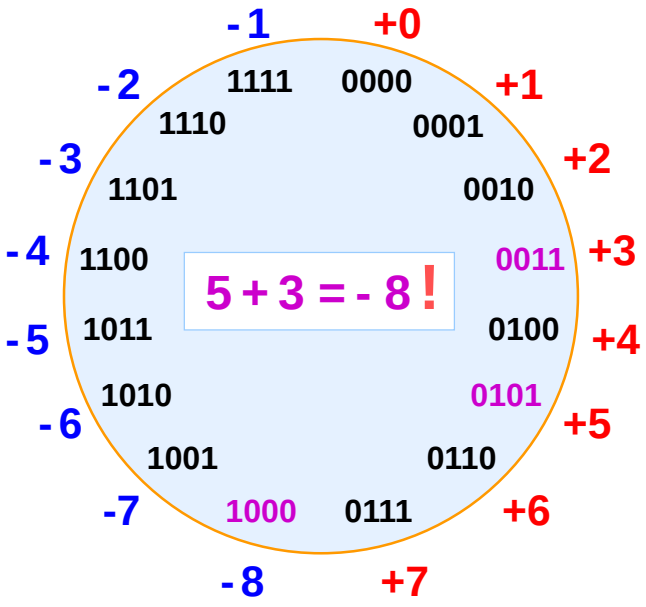
- Sumador de 4 bits
- ¿ Acarreo de salida ?
  - no influye en el resultado
- Resultado con 4 bits correcto

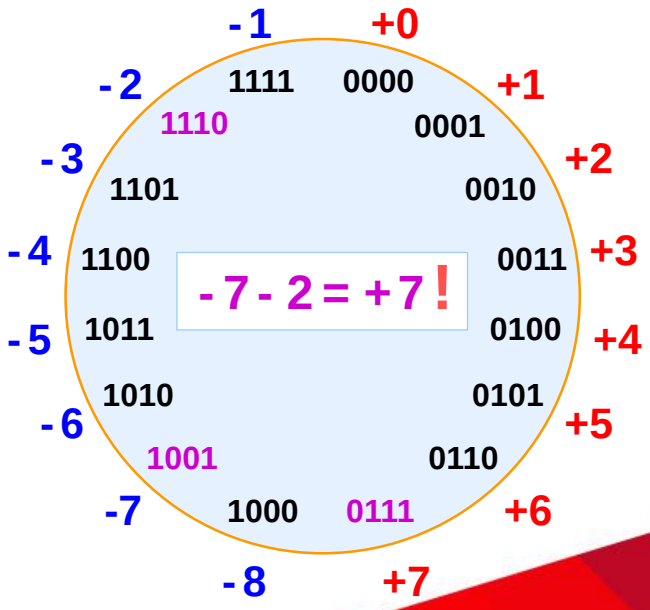
4	0100	-4	1100
3	0011	-3	1101
7	0 0111	-7	1 1001

4	0100	-4	1100
-3	1101	+3	0011
1	1 0001	-1	0 1111

ERROR DE DESBORDAMIENTO EN Ca2

- Suma números del mismo signo excede rango permitido
  - error de **overflow** (desbordamiento)
    - necesario detectar su presencia
    - Detección basada en: suma números del mismo signo → resultado con signo distinto





TEMA 5.1 - FUNDAMENTOS

- Introducción a la Electrónica Digital
- Códigos binarios
- Aritmética binaria
- Algebra de Boole

ALGEBRA DE BOOLE

- El Algebra de Boole son los fundamentos matemáticos de los circuitos digitales

- Denominada Álgebra de Boole en honor de su inventor, George Boole
  - "*An Investigation of the Laws of Thought*" (1854)

- Un álgebra se define por un conjunto de elementos con unas operaciones. En nuestro caso:
  - B = {0, 1}
  - Φ = {+, •}

POSTULADOS DEL ÁLGEBRA DE BOOLE

- Ley de composición interna

$$\forall\ a, b \in B \Rightarrow a + b \in B,\ a \bullet b \in B$$

- Elementos neutros

$$\forall\ a \in B \Rightarrow \exists \text{ elementos neutros}$$

$$a + 0 = a$$

$$a \bullet 1 = a$$

- Elemento inverso o complementario

$$\forall\ a \in B \Rightarrow \exists\ \bar{a} \in B$$

$$a + \bar{a} = 1$$

$$a \bullet \bar{a} = 0$$

POSTULADOS DEL ÁLGEBRA DE BOOLE

- Propiedad conmutativa

$$\forall\ a, b \in B \Rightarrow a + b = b + a$$

$$a \bullet b = b \bullet a$$

- Propiedad distributiva

$$\forall\ a, b, c \in B \Rightarrow a + (b \bullet c) = (a + b) \bullet (a + c)$$

$$a \bullet (b + c) = a \bullet b + a \bullet c$$

PROPIEDADES FUNDAMENTALES

- Dualidad: Toda ley válida tiene una dual, que se obtiene cambiando

$$0 \leftrightarrow 1 \text{ y } + \leftrightarrow \bullet$$

- Idempotencia

$$\forall\, a \in B \Rightarrow a + a = a$$

$$a \bullet a = a$$

  - Demostración:

$$a + a = (a + a) \bullet 1 = (a + a) \bullet (a + \bar{a}) = a + (a \bullet \bar{a}) = a + 0 = a$$

  - Por dualidad se demuestra para •

- $\forall\, a \in B \Rightarrow a + 1 = 1 \qquad a \bullet 0 = 0$

PROPIEDADES FUNDAMENTALES

- De las propiedades anteriores se pueden definir las operaciones básicas.

a	b	a+b
0	0	0
0	1	1
1	0	1
1	1	1

a	b	a•b
0	0	0
0	1	0
1	0	0
1	1	1

a	$\bar{a}$
0	1
1	0

- Tabla de verdad: proporciona el valor de una función para todas las posibles combinaciones de valores de las entradas

PROPIEDADES FUNDAMENTALES

- Involución

$$\forall\, a \in B \Rightarrow \bar{\bar{a}} = a$$

- Absorción

$$\forall\, a, b \in B \Rightarrow a + ab = a$$

$$a\,(a+b) = a$$

  - Demostración:

$$a + ab = a \bullet 1 + ab = a\,(1+b) = a \bullet 1 = a$$

- Propiedad asociativa

$$\forall\, a, b, c \in B \Rightarrow (a + b) + c = a + (b + c)$$

$$(a \bullet b) \bullet c = a \bullet (b \bullet c)$$

PROPIEDADES FUNDAMENTALES

- Leyes de De Morgan:

$$\forall\, a, b \in B \Rightarrow \overline{a+b} = \bar{a}\bar{b}$$
$$\overline{a \bullet b} = \bar{a} + \bar{b}$$

  - Demostración: Basta probar que $\bar{a}\bar{b}$ es el complemento de a + b

$$(a+b)+\bar{a}\bar{b} = (a+b+\bar{a})\left(a+b+\bar{b}\right) = 1 \bullet 1 = 1$$

$$(a+b) \bullet \bar{a}\bar{b} = \left(a\bar{a}\bar{b}\right) + \left(b\bar{a}\bar{b}\right) = 0+0 = 0$$

FUNCIONES Y EXPRESIONES BOOLEANAS

- Definiciones:
  - Una variable lógica o booleana es cualquier elemento

$$x \in B = \{0, 1\}$$

  - Función lógica o booleana es una función con n entradas booleanas y una única salida booleana

$$f : B_n \rightarrow B$$

$$(x_1, x_2, \ldots, x_n) \rightarrow y$$

FUNCIONES Y EXPRESIONES BOOLEANAS

Las funciones lógicas o booleanas se representan mediante:

Expresión

f(a, b) = a + b

literal

Tabla de verdad

a b	f(a,b)
0 0	0
0 1	1
1 0	1
1 1	1