

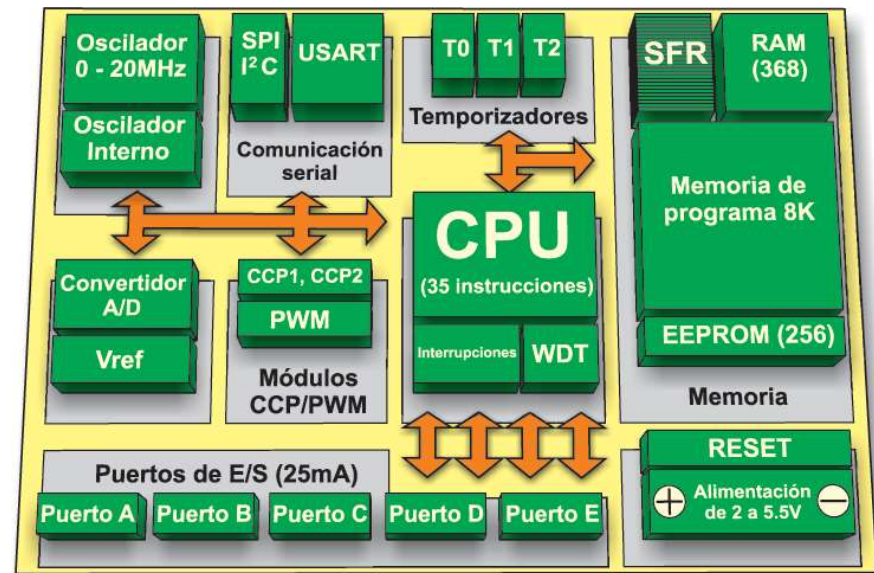
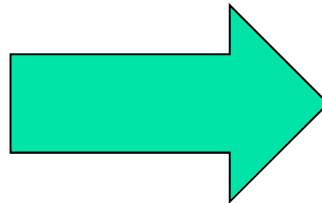
Sistemas Digitales I

Sistemas Digitales I

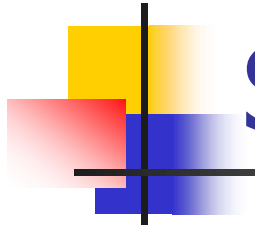
De las variables "binarias" a los microprocesadores/microcontroladores



Geroge Boole
1815–1864



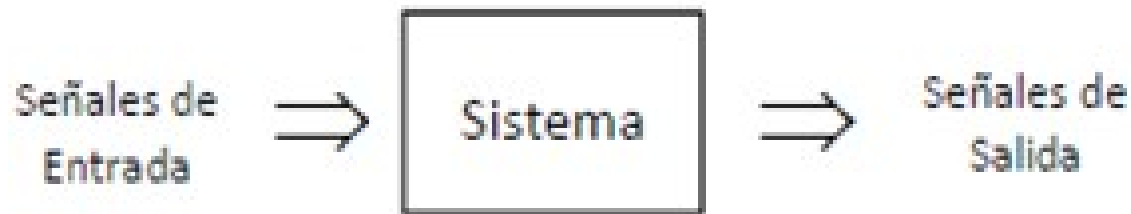
ACTUALIDAD



Sistemas Digitales I

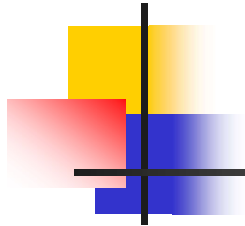
SISTEMA

conjunto de elementos que interactúan entre sí con un fin común predeterminado



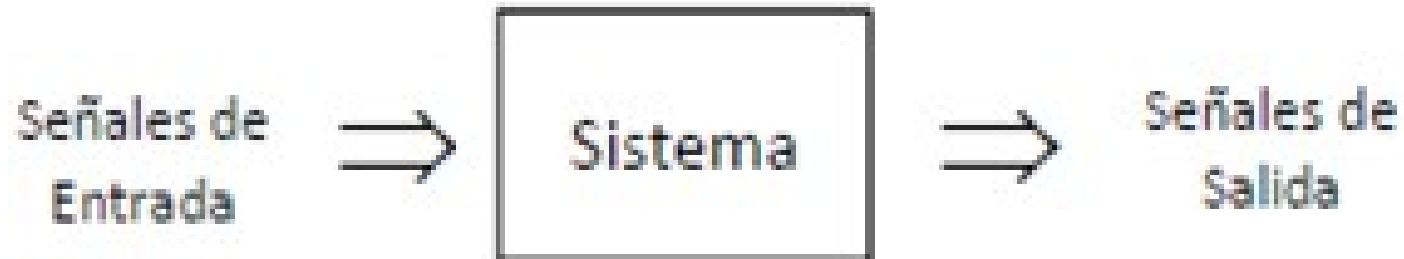
COMPORTAMIENTO:

tiene que ver con la manera en que reacciona el mismo a la información recibida



Sistemas Digitales I

Sistema Lógico



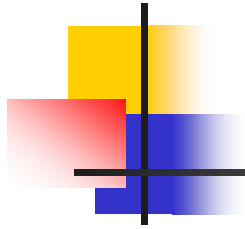
es un sistema cuyo ***comportamiento es predecible***

Sistemas Digitales I

- señales

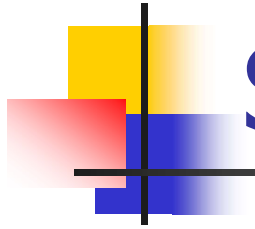


- Las señales son variables físicas, en su mayoría de carácter continuo (analógico).
- Un sistema *digital* trata con variables digitales, es decir, que sólo pueden tomar valores discretos.



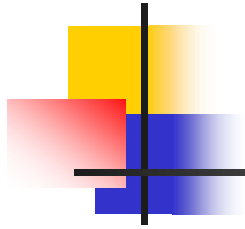
Sistemas Digitales

- **Ventajas de las técnicas digitales**
 - ***más fáciles de diseñar*** (sólo el rango de tensión -ALTO o BAJO)
 - ***Facilidad para almacenar la información*** (ej. Flip Flop)
 - ***Mayor exactitud y precisión*** (con agregar más dígitos se aumenta la precisión)
 - ***Menos afectados por el ruido*** (no dependen de valores exactos de tensión –rango-)



Sistemas Digitales I

- **Limitaciones de las técnicas digitales El mundo real es fundamentalmente analógico**
 - *El mundo real es fundamentalmente analógico*
 - *Facilidad para almacenar la información* (ej. Flip Flop)
 - *Mayor exactitud y precisión* (con agregar más dígitos se aumenta la precisión)
 - *Circuitos menos afectados por el ruido* (no dependen de valores exactos -rango de tensión-)

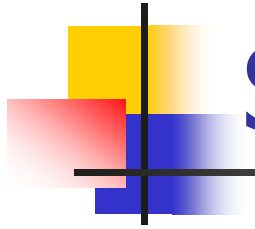


Sistemas Digitales

- **Limitaciones de las técnicas digitales**

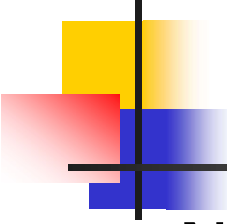
El mundo real es fundamentalmente analógico, si las entradas y las salidas son analógicas será necesario:

- Convertir las entradas analógicas del “mundo real” a la forma digital (CAD)
- Procesar digitalmente
- Convertir de nuevo las salidas digitales a la forma analógica del “mundo real” (CDA)



Sistemas Digitales I (repaso)

- Analógico = continuo
- Digital = discreto
- Variables DIGITALES
 - las cantidades no se denotan por valores proporcionales, sino por símbolos denominados dígitos.



Sistemas Digitales I (repaso)

- Numeración decimal

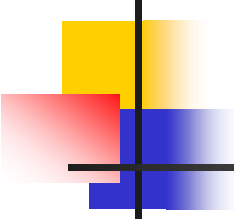
1's column
10's column
100's column
1000's column

$$5374_{10} =$$

- Numeración binaria

1's column
2's column
4's column
8's column

$$1101_2 =$$



Sistemas Digitales I (repaso)

- Numeración decimal

1's column
10's column
100's column
1000's column

$$5374_{10} = 5 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$$

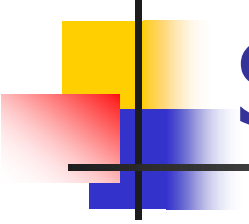
five three seven four
thousands hundreds tens ones

- Numeración binaria

1's column
2's column
4's column
8's column

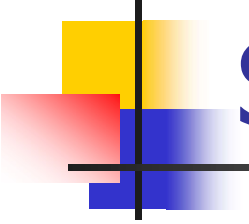
$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

one one no one
eight four two one



Sistemas Digitales I (repaso)

Hex Digit	Decimal Equivalent	Binary Equivalent
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
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111



Sistemas Digitales I (repaso)

- Bits

10010110

most significant bit least significant bit

- Bytes & Nibbles

byte

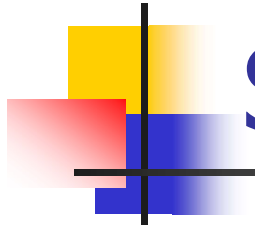
10010110

nibble

- Bytes

CEBF9AD7

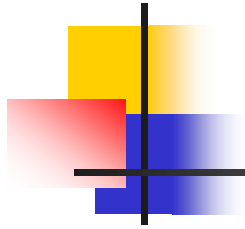
most significant byte least significant byte



Sistemas Digitales I (repaso)

- $2^{10} = 1 \text{ kilo} \approx 1000 \text{ (1024)}$
- $2^{20} = 1 \text{ mega} \approx 1 \text{ million (1,048,576)}$
- $2^{30} = 1 \text{ giga} \approx 1000 '' \text{ (1,073,741,824)}$

- $2^{32} = 2^2 \times 2^{30} = 4 \text{ giga}$

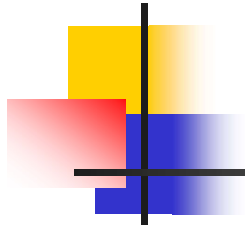


Sistemas Digitales

Representaciones digitales (discretas)

la más simple de simular físicamente es la BINARIA:

- solo son necesarios dos estados
- 1 y 0, alto y bajo, abierto y cerrado, etc.
- que se pueden hacer corresponder fácilmente con el corte y la saturación de los transistores.



Sistemas Digitales

- Variables binarias
 - Dos estados: 1/0, true/false, open/close
- Tipos de Sistemas Digitales
 - Combinacionales
 - Secuenciales
 - Asincrónicos
 - Sincrónicos

Sistemas Digitales Combinacionales

- Las salidas actuales sólo dependen del estado actual de las entradas

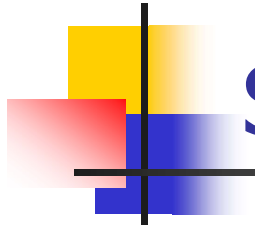


- Las combinaciones posibles de las entradas y en cada caso su resultado como salida pueden explicarse mediante una "**tabla de comportamiento**".
- Esta dependencia o combinación se puede representar mediante ecuaciones algebraicas

Sistemas Digitales Combinacionales

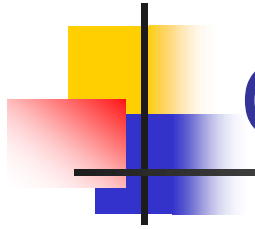
■ VER EJEMPLO SISTEMAS LÓGICOS





Sistemas Digitales Combinacionales

- Algebra de Boole: como cualquier algebra pero las variables toman solo dos valores (0 y 1)
- POSTULADOS del Álgebra de Boole:
 - AND, OR, NOT
 - Una proposición lógica p es "verdadera" (V) o "falsa" (F).

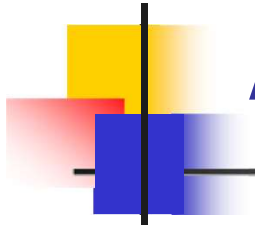


Operaciones lógicas

Operaciones lógicas con las proposiciones lógicas p_i :

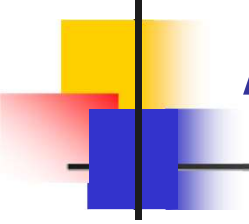
- Convención: $V = 1$, $F = 0$ (lógica POSITIVA)

$p1 \text{ AND } p2 = p3$	donde:	$p3=V \Leftrightarrow p1=V \text{ "y" } p2=V$	$p3=F \Leftrightarrow p1=F \text{ "o" } p2=F$
$p1 \text{ OR } p2 = p3$	donde:	$p3=V \Leftrightarrow p1=V \text{ "o" } p2=V$	$p3=F \Leftrightarrow p1=F \text{ "y" } p2=F$
$\text{NOT } p1=p2$	donde:	$P2=V \Leftrightarrow p1=F$	$P2=F \Leftrightarrow p1=V$



ALGEBRA DE BOOLE

- Geroge Boole (1815-1864)
- Formula un álgebra para representar la lógica aristotélica (la silogística)
 - ▶ Lógica proposicional
- Define un conjunto de elementos, un conjunto de operadores y varios axiomas o postulados .



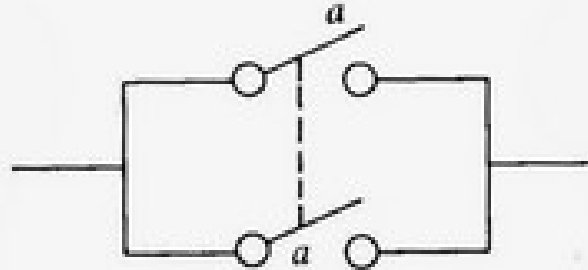
ALGEBRA DE BOOLE

- El trabajo de Boole llegó a ser un paso fundamental en la revolución de los computadores, cuando Claude Shannon en 1938, demostró cómo las operaciones booleanas elementales se podían representar mediante circuitos conmutadores eléctricos. Y también cómo la combinación de éstos podía representar operaciones aritméticas y lógicas complejas.
- Shannon demostró asimismo que el álgebra de Boole se podía utilizar para simplificar circuitos de conmutación.

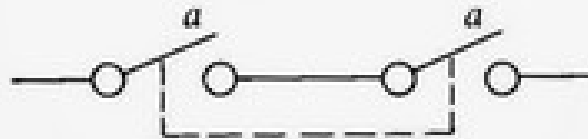
Operaciones lógicas

- Álgebra de Boole (analogía/circuito):

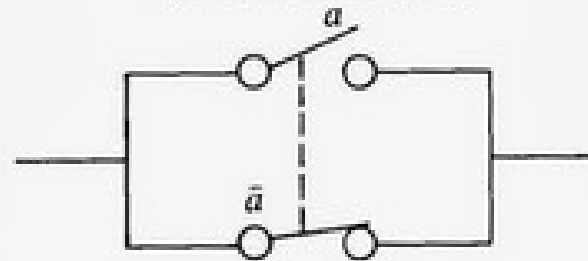
$$a + a = a$$



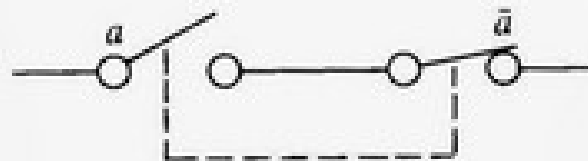
$$a \cdot a = a$$



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



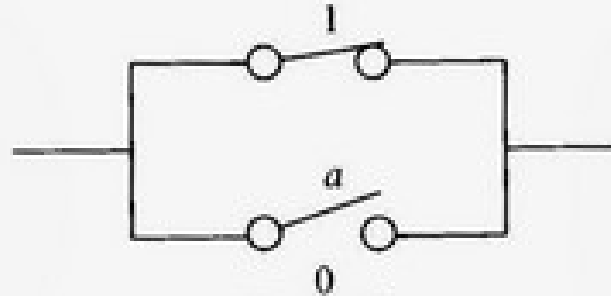
$$a \cdot \bar{a} = 0$$



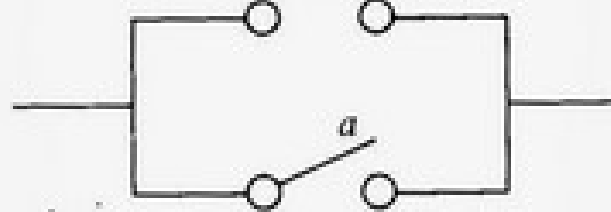
Operaciones lógicas

- Álgebra de Boole (analogía/circuito):

$$a + 1 = 1$$



$$a + 0 = a$$

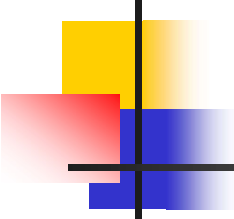


$$a \cdot 1 = a$$



$$a \cdot 0 = 0$$





Sistemas Digitales Combinacionales

- Leyes y reglas del álgebra de Boole :

Nro. Orden	Nombre	a) Forma OR	b) Forma AND
1	Ley de IDEMPOTENCIA	$A+A=A$	$A.A=A$
2	Existencia de Elementos Neutros (Ley de Identidad)	$A+0=A$	$A.1=A$
3	Ley de ANULACION (Prop. de los elementos neutros)	$A+1=1$	$A.0=0$
4	Ley CONMUTATIVA	$A+B=B+A$	$A.B=B.A$
5	Ley ASOCIATIVA	$(A+B)+C=A+(B+C)$	$(A.B).C=A.(B.C)$
6	Ley del INVERSO	$A+A'=1$	$A.A'=0$
7	Ley DISTRIBUTIVA	$A.(B+C)=A.B+AC$	$A+B.C=(A+B).(A+C)$
8	Ley de ABSORCION	$A+A.B=A$	$A.(A+B)=A$
9	Ley de ABSORCION del COMPLEMENTO	$A+(A'.B)=A+B$	$A.(A'+B)=A.B$
10	de CONSENSO	$AB+BC+A'C = AB+A'C$	$(A+B).(B+C).(A'+C) = (A+B).(A'+C)$
11	Ley de MORGAN	$(A+B)'=A'.B'$	$(A.B)'=A'+B'$
12	DOBLE NEGACION (Ley de Involución)	$(A')'=A$	

Sistemas Digitales Combinacionales

- álgebra de Boole :

10	$(B \bullet C) + (\overline{B} \bullet D) + (C \bullet D) =$ $(B \bullet C) + (\overline{B} \bullet D)$	consenso
----	---	----------

Prove using other theorems and axioms:

$$B \bullet C + \overline{B} \bullet D + C \bullet D$$

$$= BC + \overline{B}D + (CDB + CDB\overline{B}) \quad \{CD + (B \cdot \overline{B})\}$$

$$= BC + \overline{B}D + BCD + \overline{B}CD$$

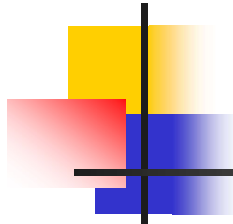
$$= BC + BCD + \overline{B}D + \overline{B}CD \quad \text{CONMUTATIVA}$$

$$= (BC + BCD) + (\overline{B}D + \overline{B}CD) \quad \text{ASOCIATIVA}$$

$$= BC + \overline{B}D \quad \text{ABSORCIÓN}$$

- generalización

$\overline{B_0 \bullet B_1 \bullet B_2 \dots} =$ $\overline{B_0 + B_1 + B_2 \dots}$	$\overline{B_0 + B_1 + B_2 \dots} =$ $\overline{B_0} \bullet \overline{B_1} \bullet \overline{B_2} \dots$	DeMorgan
---	---	----------



Álgebra booleana

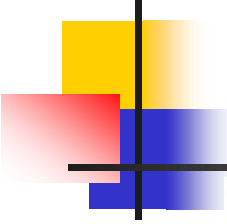
Término *canónico* de una función lógica:

- todo *producto o suma* en que aparecen **todas las variables** en su forma directa o inversa

Todas las funciones lógicas son expresables en forma canónica

- como una *disyunción lógica* (OR) "**suma de productos**"
- como una *conjunción lógica* (AND) "**producto de sumas**"

Una función formada exclusivamente por términos canónicos recibe el nombre de "función canónica".



Álgebra booleana

Suma de productos

- También conocida como expansión de **minterminos**

			$F = 001 \quad 011 \quad 101 \quad 110 \quad 111$	
			$F = A'B'C + A'BC + AB'C + ABC' + ABC$	
A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Diagram illustrating the expansion of the Boolean function F into its minterms. The truth table shows the values of A, B, and C, and the corresponding values of F and F'. The function F is expanded as the sum of minterms where F is 1: $F = A'B'C + A'BC + AB'C + ABC' + ABC$. Arrows connect the minterms in the expansion to the rows in the truth table where F is 1.

suma de productos

- Términos son productos (o minterms)
 - productos AND de literales - para las combinaciones de input para las que el output es verdad
 - en cada producto cada variable aparece exactamente una vez (puede estar invertida)

A	B	C	minterms
0	0	0	$A'B'C'$ m0
0	0	1	$A'B'C$ m1
0	1	0	$A'BC'$ m2
0	1	1	$A'BC$ m3
1	0	0	$AB'C'$ m4
1	0	1	$AB'C$ m5
1	1	0	ABC' m6
1	1	1	ABC m7

forma corta de escribir minterms
(ejemplo de 3 terminos o $2^3 = 8$ minterms)

F en forma canónica:

$$\begin{aligned}F(A, B, C) &= \Sigma m(1,3,5,6,7) \\&= m1 + m3 + m5 + m6 + m7 \\&= A'B'C + A'BC + AB'C + ABC' + ABC\end{aligned}$$

forma canónica $\neq$ forma minima

$$\begin{aligned}F(A, B, C) &= A'B'C + A'BC + AB'C + ABC + ABC' \\&= (A'B' + A'B + AB' + AB)C + ABC' \\&= ((A' + A)(B' + B))C + ABC' \\&= C + ABC' \\&= ABC' + C \\&= AB + C\end{aligned}$$

Álgebra booleana

Producto de sumas

- También conocida como expansión de maxterminos

A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

$F = \begin{matrix} 000 & 010 & 100 \\ (A + B + C) & (A + B' + C) & (A' + B + C) \end{matrix}$

Producto de sumas

- Términos son sumas (o maxterminos)
 - suma OR de literales - para las combinacion de input para los que el output es falso
 - en cada producto cada variable aparece exactamente una ves (puede estar invertida)

A	B	C	maxterms
0	0	0	A+B+C M0
0	0	1	A+B+C' M1
0	1	0	A+B'+C M2
0	1	1	A+B'+C' M3
1	0	0	A'+B+C M4
1	0	1	A'+B+C' M5
1	1	0	A'+B'+C M6
1	1	1	A'+B'+C' M7

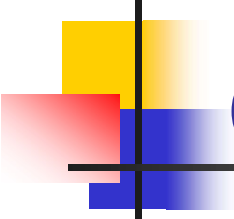
F en forma canónica:

$$\begin{aligned}
 F(A, B, C) &= \Pi M(0,2,4) \\
 &= M0 \bullet M2 \bullet M4 \\
 &= (A + B + C) (A + B' + C) (A' + B + C)
 \end{aligned}$$

forma canónica $\neq$ forma minima

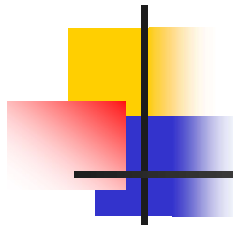
$$\begin{aligned}
 F(A, B, C) &= (A + B + C) (A + B' + C) (A' + B + C) \\
 &= (A + B + C) (A + B' + C) \\
 &\quad (A + B + C) (A' + B + C) \\
 &= (A + C) (B + C)
 \end{aligned}$$

forma corta de escribir minterminos
(ejemplo de 3 términos o $2^3 = 8$ minterminos)



Conversión entre formas canónicas

- Conversión de **minterminos a maxterminos**
 - usar maxterminos cuyos índices no aparecen en expansión de minterminos
 - e.g., $F(A,B,C) = \Sigma m(1,3,5,6,7) = \Pi M(0,2,4)$
- Conversión de **maxterminos a minterminos**
 - usar minterminos cuyos índices no aparecen en expansión de maxterminos
 - e.g., $F(A,B,C) = \Pi M(0,2,4) = \Sigma m(1,3,5,6,7)$



Álgebra booleana

la función lógica $f_{(A, B, C)}$ puede expresarse:

- Como suma de productos: 1º forma canónica (algebraica o numérica)
- Como producto de sumas: 2º forma canónica (algebraica o numérica)

Nº	A	B	C	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

$$f = A' \cdot B' \cdot C + A' \cdot B \cdot C' + A \cdot B' \cdot C' + A \cdot B' \cdot C + A \cdot B \cdot C \quad (1^\circ \text{ F.C.A.})$$

$$f = \sum_3 m_i (1, 2, 4, 5, 7) \quad (1^\circ \text{ F.C.N.})$$

$$f = (A+B+C) \cdot (A+B'+C') \cdot (A'+B'+C) \quad (2^\circ \text{ F.C.A.})$$

$$f = \prod_3 M_i (0, 3, 6) \quad (2^\circ \text{ F.C.N.})$$

Funciones



Desarrollar la lógica para que se encienda una luz cada vez que el carro llega a un extremo. Bastará con colocar un sensor (A y B) en cada extremo para detectarlo. podría pensarse en una tabla que describa el comportamiento:

A	B	f
0	0	0
0	1	1
1	0	1

la función canónica será: $f_{(A, B)} = A'B + AB'$

- a) ¿parece una función más compleja que la solución intuitiva?
- b) ¿si hay una solución más simple, representa realmente la tabla?

Funciones

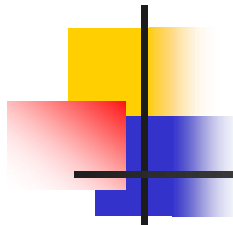


Jamás podrá producirse el caso en que A y B sean accionados simultáneamente por el vagón.

Estas situaciones se definen como redundancias (entradas que jamás se producirán en el comportamiento que debemos modelar).

Debe explicitarse el resultado para todas combinaciones de las entradas, es lo que se denomina una tabla de verdad válida (**T.V.V.**)

A	B	f
0	0	0
0	1	1
1	0	1
1	1	—



Funciones

A la hora de escribir la ecuación podremos darle el valor más conveniente a esa combinación de entradas. Ecuación algebraica (suma de productos):

<i>A</i>	<i>B</i>	<i>f</i>
0	0	0
0	1	1
1	0	1
1	1	—

$$\mathbf{f_{(A, B,)} = A'B + AB'}$$

(donde $f=0$ para $A=B=1$)

o también

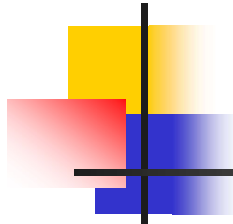
$$\mathbf{f_{(A, B,)} = A'B + AB' + AB}$$

(donde $f=1$ para $A=B=1$)

En este último caso $\mathbf{A'B + AB' + AB = A'B + A.(B' + B) = A'B + A = B + A = A + B}$

simplificada: $\mathbf{f_{(A, B,)} = A + B}$

¿ qué tabla representa la ecuación hallada ?



Funciones

A la hora de escribir la ecuación podremos darle el valor más conveniente a esa combinación de entradas

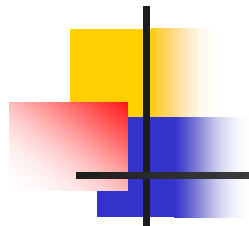
<i>A</i>	<i>B</i>	<i>f</i>
0	0	0
0	1	1
1	0	1
1	1	—

primera forma canónica numérica

$$\mathbf{f}_{(A, B,)} = \Sigma \mathbf{m_i} (\mathbf{1,2}) + \Sigma \emptyset \mathbf{i} (\mathbf{3})$$

segunda forma canónica numérica

$$\mathbf{f}_{(A, B,)} = \Pi \mathbf{M_i} (\mathbf{0}) . \Pi \emptyset \mathbf{i} (\mathbf{3})$$



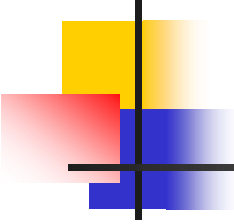
Formas canónicas

- Los índices que aparecen en una forma son los que faltan en la otra

$$\begin{aligned}F_2(x, y, z) &= \Sigma(0, 1, 4, 7) \\ &= \Pi(2, 3, 5, 6)\end{aligned}$$

x	y	z	F_2
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$\begin{aligned}F_2(x, y, z) &= x'y'z' + x'y'z + xy'z' + xyz \\ &= (x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)\end{aligned}$$



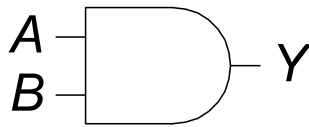
Sistemas Digitales Combinacionales

- Implementación electrónica
 - Mediante compuertas
 - La combinación última de compuertas permite obtener la funcionalidad deseada aunque esta sea muy compleja
 - Los axiomas y teoremas del álgebra de Boole permiten simplificar las funciones
 - Los mapas de Karnaugh permiten simplificar gráficamente las funciones

COMPUERTAS LÓGICAS

■ Implementación electrónica

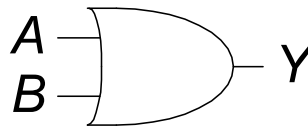
AND



$$Y = AB$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

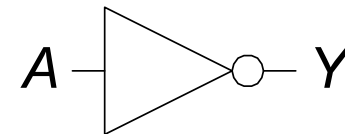
OR



$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT



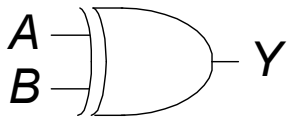
$$Y = \overline{A}$$

A	Y
0	1
1	0

COMPUERTAS LÓGICAS

■ Implementación electrónica

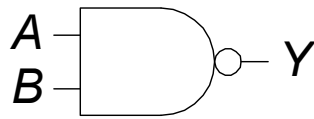
XOR



$$Y = A \oplus B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

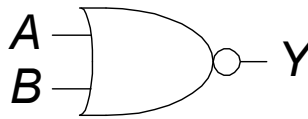
NAND



$$Y = \overline{AB}$$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

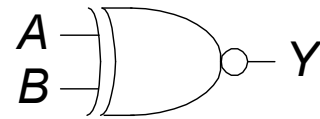
NOR



$$Y = \overline{A + B}$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

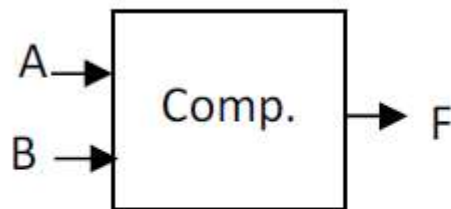
XNOR



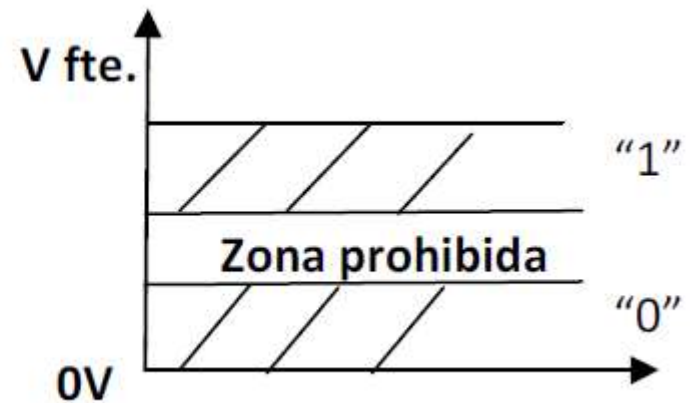
$$Y = \overline{A \oplus B}$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

COMPUERTAS LÓGICAS



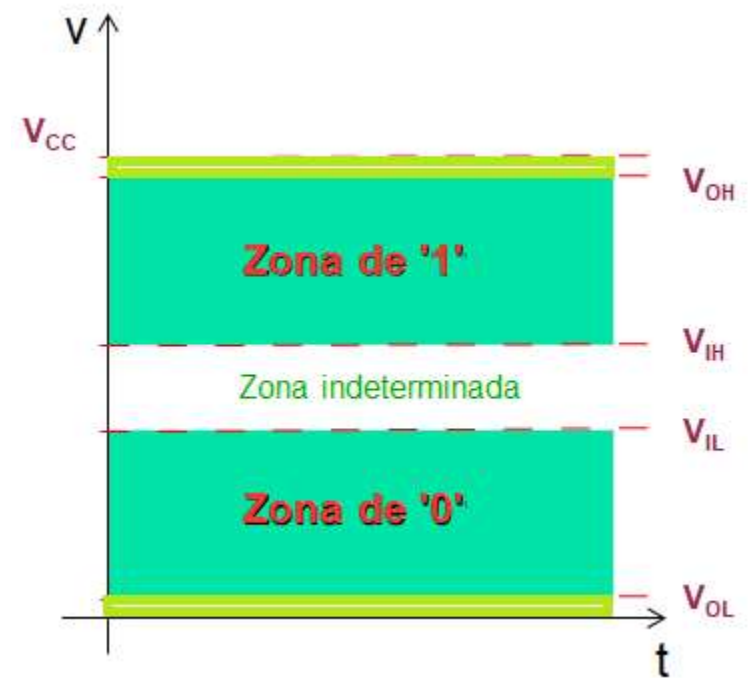
A	B	F
L	L	
L	H	
H	L	
H	H	

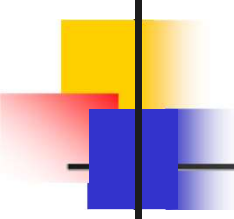


- Convención (Lógica POSITIVA):
 $V = "1"$, $F = "0"$

COMPUERTAS LÓGICAS (características)

- V_{OH} : Mínima tensión de salida de la compuerta con el nivel lógico ALTO ó '1'.
- V_{OL} : Máxima tensión de salida de la compuerta con el nivel lógico BAJO ó '0'.
- V_{IH} : Mínima tensión de entrada de la compuerta con el nivel lógico ALTO ó '1'.
- V_{IL} : Máxima tensión de entrada de la compuerta con el nivel lógico BAJO ó '0'.



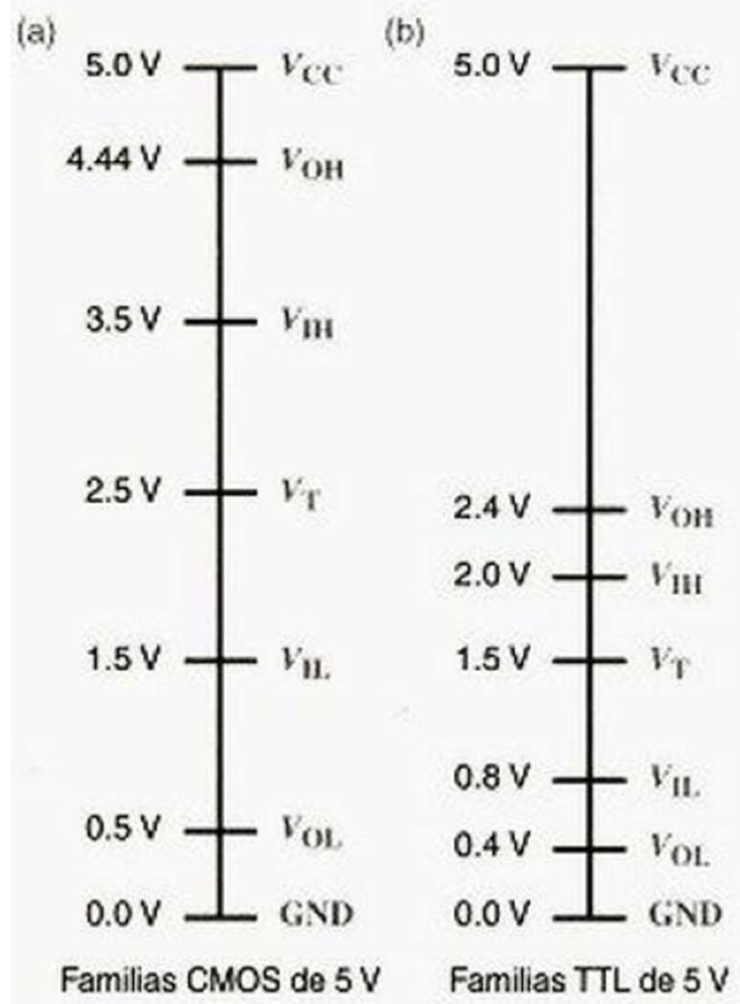


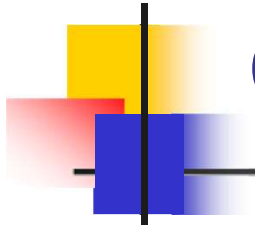
COMPUERTAS LÓGICAS (características)

*Niveles de corrientes de entrada y salida definirán:
capacidad de carga que tendrá una compuerta
y corriente necesaria para lograr la excitación de la misma.
Los valores de corriente para los niveles lógicos '0' y '1' NO SIEMPRE
SON SIMÉTRICOS y depende de la familia de circuitos integrados.*

- I_{OH} : Corriente de salida de la compuerta con el nivel lógico ALTO ó '1'.
- I_{OL} : Corriente de salida de la compuerta con el nivel lógico BAJO ó '0'.
- I_{IH} : Corriente de entrada de la compuerta con el nivel lógico ALTO ó '1'.
- I_{IL} : Corriente de entrada de la compuerta con el nivel lógico BAJO ó '0'.

Niveles de Tensión – CMOS y TTL





CARACTERÍSTICAS GENERALES

- Los transistores que conforman un circuito lógico trabajan en modo conmutación (saturación-corte) que corresponden a '0' y '1' lógicos.
- El traspaso de un estado a otro no es instantáneo.
- Los retardos de conmutación son relevantes cuando se trabaja a altas frecuencia.
- En la práctica, no pueden existir ondas perfectamente cuadradas, porque la tensión necesita siempre un tiempo (por pequeño que sea) para pasar de un valor máximo a un mínimo (o viceversa).

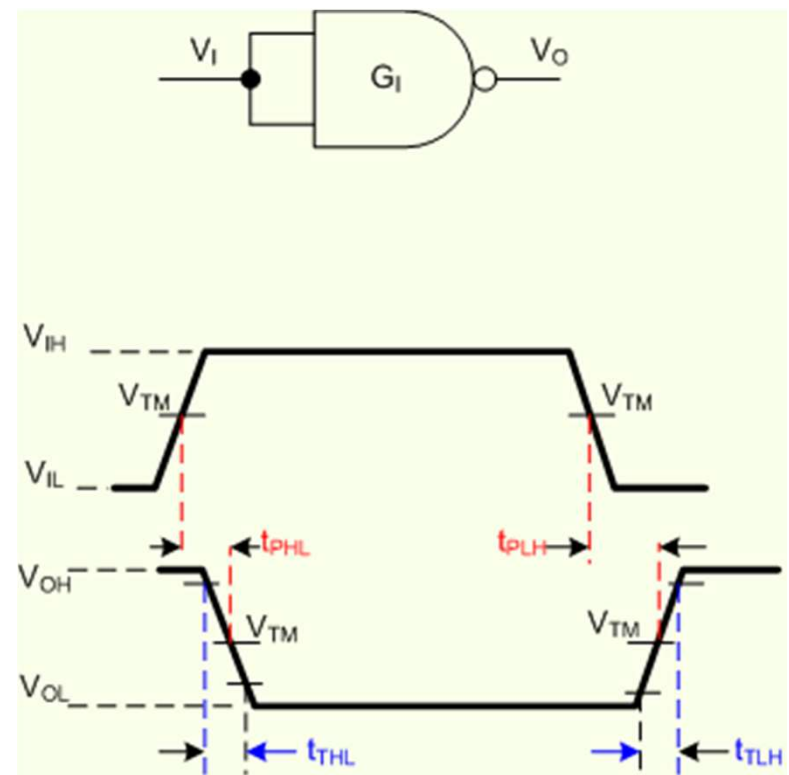
CARACTERÍSTICAS GENERALES

→ **Tiempo de subida (t_{TLH}) 'Rise Time':** Es el tiempo que le toma a la salida crecer desde el 10% al 90% del valor total de tensión.

→ **Tiempo de bajada (t_{TLH}) 'Fall Time':** Es el tiempo que le toma a la salida decrecer desde el 90% al 10% del valor total de tensión.

→ **Tiempo de propagación HL (t_{PHL}):** Es el retraso entre el frente de entrada y el de salida cuando esta pasa de '1' a '0', medido al 50% de ambas señales.

→ **Tiempo de propagación LH (t_{PLH}):** Es el retraso entre el frente de entrada y el de salida cuando esta pasa de '0' a '1', medido al 50% de ambas señales.

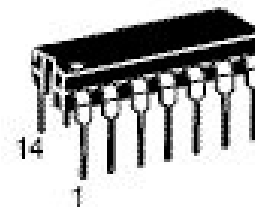
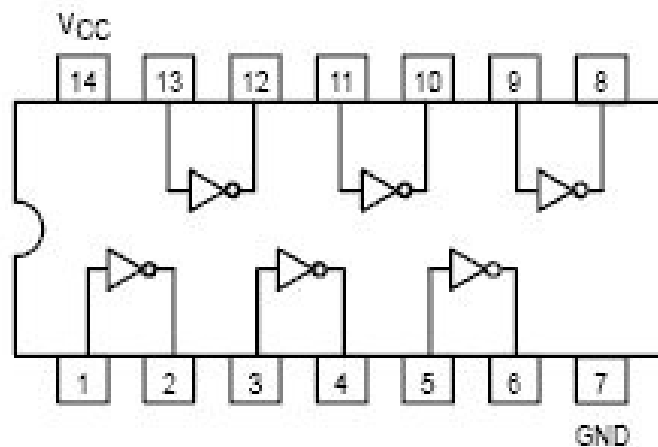


COMPUERTAS LÓGICAS

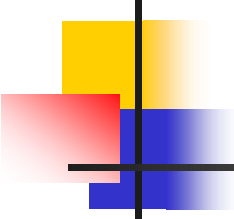
Integración a pequeña escala **SSI** (small-scale integration)

- Hasta 10 compuertas / 100 componentes por C.I.

74LS04



- NOTA: OBSERVAR HOJAS DE DATOS DE FABRICANTES



Sistemas Digitales Combinacionales

- Implementación electrónica

Dada la tabla de verdad llegar al circuito

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Suma de productos

- Términos son productos (o minterms)
 - productos AND de literales - para las combinaciones de input para las que el output es verdad
 - en cada producto cada variable aparece exactamente una vez (puede estar invertida)

A	B	C	minterms
0	0	0	$A'B'C'$ m0
0	0	1	$A'B'C$ m1
0	1	0	$A'BC'$ m2
0	1	1	$A'BC$ m3
1	0	0	$AB'C'$ m4
1	0	1	$AB'C$ m5
1	1	0	ABC' m6
1	1	1	ABC m7

forma corta de escribir minterms
(ejemplo de 3 terminos o $2^3 = 8$ minterms)

F en forma canónica:

$$\begin{aligned}
 F(A, B, C) &= \Sigma m(1,3,5,6,7) \\
 &= m1 + m3 + m5 + m6 + m7 \\
 &= A'B'C + A'BC + AB'C + ABC' + ABC
 \end{aligned}$$

forma canónica $\neq$ forma minima

$$\begin{aligned}
 F(A, B, C) &= A'B'C + A'BC + AB'C + ABC + ABC' \\
 &= (A'B' + A'B + AB' + AB)C + ABC' \\
 &= ((A' + A)(B' + B))C + ABC' \\
 &= C + ABC' \\
 &= ABC' + C \\
 &= AB + C
 \end{aligned}$$

Producto de sumas

- Términos son sumas (o maxterminos)
 - suma OR de literales - para las combinacion de input para los que el output es falso
 - en cada producto cada variable aparece exactamente una ves (puede estar invertida)

A	B	C	maxterms
0	0	0	A+B+C M0
0	0	1	A+B+C' M1
0	1	0	A+B'+C M2
0	1	1	A+B'+C' M3
1	0	0	A'+B+C M4
1	0	1	A'+B+C' M5
1	1	0	A'+B'+C M6
1	1	1	A'+B'+C' M7

F en forma canónica:

$$\begin{aligned}
 F(A, B, C) &= \Pi M(0,2,4) \\
 &= M0 \bullet M2 \bullet M4 \\
 &= (A + B + C) (A + B' + C) (A' + B + C)
 \end{aligned}$$

forma canónica $\neq$ forma minima

$$\begin{aligned}
 F(A, B, C) &= (A + B + C) (A + B' + C) (A' + B + C) \\
 &= (A + B + C) (A + B' + C) \\
 &\quad (A + B + C) (A' + B + C) \\
 &= (A + C) (B + C)
 \end{aligned}$$

forma corta de escribir minterminos
(ejemplo de 3 términos o $2^3 = 8$ minterminos)

Implementación / circuito

$$F(A,B,C) = AB + C \quad / \quad (A+C).(B+C)$$

