Tema 2: Álgebra de Boole, Lógica Binaria y Puertas Lógicas

Universidad de Huelva

Escuela Técnica Superior de Ingeniería

Departamento de
Ingeniería Electrónica, Sistemas Informáticos y Automática

Álgebra de Boole

Un **Álgebra de Boole** es un sistema de elementos **b**={0,1} y los operadores binarios (·) y (+) y (’) definidos de la siguiente forma:

OPERADOR + → OPERADOR OR
OPERADOR · → OPERADOR AND
OPERADOR ‘ (¯) → OPERADOR NOT

que cumplen las siguientes propiedades (Postulados del Álgebra de Boole) :

- **PROPIEDAD CONMUTATIVA**:

$$A + B = B + A$$
$$A \cdot B = B \cdot A$$

- **PROPIEDAD DISTRIBUTIVA**:

$$A \cdot (B+C) = A \cdot B + A \cdot C$$
$$A + B \cdot C = (A+B) \cdot (A+C)$$

- **ELEMENTOS NEUTROS DIFERENTES**

$$A + 0 = A$$
$$A \cdot 1 = A$$

- **SIEMPRE EXISTE EL COMPLEMENTO DE A, DENOMINADO A’**

$$A + A' = 1$$
$$A \cdot A' = 0$$

A	B	A+B	A·B
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

A	A’
0	1
1	0

Álgebra de Boole (II)

- **PRINCIPIO DE DUALIDAD**: cualquier teorema o identidad algebraica deducible de los postulados anteriores puede transformarse en un segundo teorema o identidad válida sin mas que intercambiar (+) por (·), (·) por (+), ‘1’ por ‘0’ y ‘0’ por ‘1’.

- **CONSTANTE**: cualquier elemento del conjunto **B**, es decir, ‘0’ o ‘1’.

- **VARIABLE**: símbolo que representa un elemento arbitrario del álgebra, ya sea constante o fórmula completa.

Álgebra de Boole. Algunos teoremas importantes (I)

- **TEOREMA 1**: el elemento complemento A’ es único.

- **TEOREMA 2 (ELEMENTOS NULOS)**: para cada elemento de B se verifica:

  $$A+1 = 1$$
  $$A \cdot 0 = 0$$

- **TEOREMA 3**: cada elemento identidad es el complemento del otro.

  $$0'=1$$
  $$1'=0$$

- **TEOREMA 4 (IDEMPOTENCIA)**: para cada elemento de B, se verifica:

  $$A+A=A$$
  $$A \cdot A=A$$

- **TEOREMA 5 (INVOLUCIÓN)**: para cada elemento de B, se verifica:

  $$(A')' = A$$

- **TEOREMA 6 (ABSORCIÓN)**: para cada par de elementos de B, se verifica:

  $$A+A \cdot B=A$$
  $$A \cdot (A+B)=A$$

Tema 2: Álgebra de Boole, Lógica Binaria y Puertas Lógicas

Álgebra de Boole. Algunos teoremas importantes (II)

- **TEOREMA 7**: para cada par de elementos de B, se verifica:
  A + A'·B = A + B
  A · (A' + B) = A · B

- **TEOREMA 8 (ASOCIATIVIDAD)**: cada uno de los operadores binarios (+) y (·) cumple la propiedad asociativa:
  A+B+C = A+(B+C) = (A+B)+C = (A+C)+B
  A·B·C = (A·B)·C = A·(B·C) = (A·C)·B

- **LEYES DE DE MORGAN**: para cada par de elementos de B, se verifica:
  (A+B)' = A'·B'
  (A·B)' = A' + B'

Funciones lógicas y conceptos relacionados

- Se denomina **función lógica** o ***booleana*** a aquella función matemática cuyas variables son binarias y están unidas mediante los operadores del álgebra de Boole suma lógica (+), producto lógico (·) o negación(').

- Una **tabla de verdad** es aquella que representa el valor de la función para cada combinación de entrada. Si la función está definida para todas las combinaciones se llama **completa**, si no, se denomina **incompleta**.

- Una **Fórmula de conmutación** es la expresión de una función lógica.

- Un **LITERAL** es una variable (A) o complemento de una variable (A').

- Un **TÉRMINO PRODUCTO** es una operación AND de un número de literales.

- Una **fórmula normal disyuntiva** es una suma de términos productos.

- Un **TÉRMINO SUMA** es una operación OR de un número de literales.

- Una **fórmula normal conjuntiva** es un producto de términos sumas.

	X_3	X_2	X_1	X_0	$F(X_3, X_2, X_1, X_0)$
(0)	0	0	0	0	F(0,0,0,0)
(1)	0	0	0	1	F(0,0,0,1)
(2)	0	0	1	0	F(0,0,1,0)
(3)	0	0	1	1	F(0,0,1,1)
(4)	0	1	0	0	F(0,1,0,0)
(5)	0	1	0	1	F(0,1,0,1)
(6)	0	1	1	0	F(0,1,1,0)
(7)	0	1	1	1	F(0,1,1,1)
(8)	1	0	0	0	F(1,0,0,0)
(9)	1	0	0	1	F(1,0,0,1)
(10)	1	0	1	0	F(1,0,1,0)
(11)	1	0	1	1	F(1,0,1,1)
(12)	1	1	0	0	F(1,1,0,0)
(13)	1	1	0	1	F(1,1,0,1)
(14)	1	1	1	0	F(1,1,1,0)
(15)	1	1	1	1	F(1,1,1,1)

Representación de funciones lógicas: Fórmula Canónica Disyuntiva (SOP)

- **MINTÉRMINO (m_i)**: término producto en el que aparecen todas las variables, ya sean complementadas o sin complementar.
- **FÓRMULA CANÓNICA DISYUNTIVA O DE MINTÉRMINOS**: suma de mintérminos. (Suma de Productos)

- Un **mintérmino** es un término producto representado en una línea de la tabla de verdad y que hace que la función valga '1'.
- La fórmula compuesta por todos los mintérminos será idénticamente 1.
- Cada fórmula de conmutación puede expresarse como suma de mintérminos, y esa fórmula es única.
- NOTACIÓN: Un **mintérmino** se designa por "m_i" siendo i el número decimal correspondiente de la tabla de verdad.
  Para el producto, el '0' se asocia a la variable complementada y el '1' a la variable sin complementar.

C	B	A	F(C,B,A)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$F(C,B,A) = m_0 + m_2 + m_3 + m_7 = \Sigma_3(0,2,3,7)$$

$$F(C,B,A) = C'\cdot B'\cdot A' + C'\cdot B\cdot A' + C'\cdot B\cdot A + C\cdot B\cdot A$$

O bien:

$$F(C,B,A) = \overline{C}\cdot\overline{B}\cdot\overline{A} + \overline{C}\cdot B\cdot\overline{A} + \overline{C}\cdot B\cdot A + C\cdot B\cdot A$$

Representación de funciones lógicas: Fórmula Canónica Conjuntiva (POS)

- **MAXTÉRMINO (Mi)**: término suma en el que aparecen todas las variables, ya sean complementadas o sin complementar.
- **Fórmula Canónica Conjuntiva o de Maxtérminos**: producto de maxtérminos. (Producto de sumas)
- Un **maxtérmino** es un término suma representado en una línea de la tabla de verdad y que vale ‘0’.
- La fórmula compuesta por todos los maxtérminos será idénticamente 0.
- Cada fórmula puede expresarse como producto de maxtérminos, y es única.
- NOTACIÓN: Un **maxtérmino** se designa por “M_i” siendo i el número decimal correspondiente de la tabla de verdad.
  En la suma, el ‘1’ se asocia a la variable complementada y el ‘0’ a la variable sin complementar.

C	B	A	F(C,B,A)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$F(C,B,A) = M_1 \cdot M_4 \cdot M_5 \cdot M_6 = \prod{}_3(1,4,5,6)$$

$$F(C,B,A) = (C+B+A') \cdot (C'+B+A) \cdot (C'+B+A') \cdot (C'+B'+A)$$

O bien:

$$F(C,B\cdot,A) = (C+B+\overline{A}) \cdot (\overline{C}+B+A)(\overline{C}+B+\overline{A})(\overline{C}+\overline{B}+A)$$

Conversión y manipulación de fórmulas

- El **complemento de una fórmula de mintérminos** está formado por la suma de los mintérminos que no aparecen.
- El **complemento de una fórmula de maxtérminos** está formado por el producto de los maxtérminos que no aparecen.

  $\mathbf{m_i' = M_i}$

  $\mathbf{M_i' = m_i}$

- La transformación de una fórmula de mintérminos (disyuntiva) en otra de maxtérminos (conjuntiva) puede obtenerse desde la tabla de verdad o desde las expresiones canónicas numéricas.

Funciones incompletas

Son funciones para las que no están definidas todos los términos posibles de sus variables de entrada. En la tabla de verdad aparecerá una **X** o una letra **d** (del inglés ***don't care***) refiriéndose a términos sin especificar.

Complemento de una función incompleta: otra función incompleta con los mismos términos "no importa" y el complemento de la función completa.

Ejemplo (según la tabla de verdad que se muestra):

$$\mathbf{F(C,B,A) = \Sigma_3(0,2,7) + \Sigma_\Phi(3,5)}$$

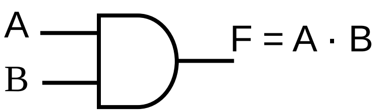
$$\mathbf{F(C,B,A) = \Pi_3(1,4,6) \cdot \Sigma_\Phi(3,5)}$$

C	B	A	F(C,B,A)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	0
1	0	1	X
1	1	0	0
1	1	1	1

Funciones básicas (I)

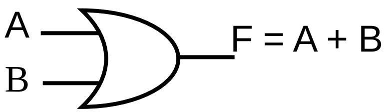
FUNCIÓN AND, PUERTA AND

A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1



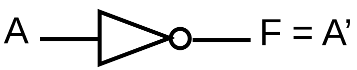
FUNCIÓN OR, PUERTA OR

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1



FUNCIÓN NOT, INVERSOR

A	A’
0	1
1	0



Usando estos tres tipos de puertas puede realizarse cualquier función de conmutación (forman un conjunto completo)

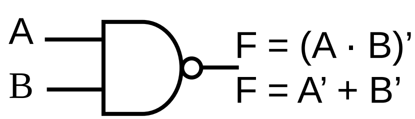
Funciones básicas (II)

Un **CONJUNTO DE PUERTAS COMPLETO** es aquel con el que se puede implementar cualquier función lógica. Los conjuntos posibles de este tipo son:

- **Puerta AND, puerta OR e INVERSOR**
- **Puerta AND e INVERSOR**
- **Puerta OR e INVERSOR**
- **Puerta NAND**
- **Puerta NOR**

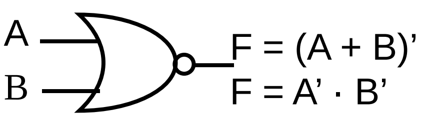
FUNCIÓN NAND, PUERTA NAND

A	B	(A·B)’
0	0	1
0	1	1
1	0	1
1	1	0



FUNCIÓN NOR, PUERTA NOR

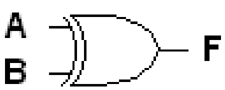
A	B	(A+B)’
0	0	1
0	1	0
1	0	0
1	1	0



Funciones básicas (III)

FUNCIÓN XOR, PUERTA XOR o PUERTA “OR EXCLUSIVA”

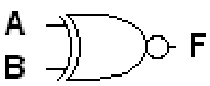
A	B	(A⊕B)
0	0	0
0	1	1
1	0	1
1	1	0



$F = (A \oplus B)$
$F = A'B + AB'$

FUNCIÓN XNOR, PUERTA XNOR o PUERTA “NOR EXCLUSIVA”

A	B	(A⊕B)’
0	0	1
0	1	0
1	0	0
1	1	1



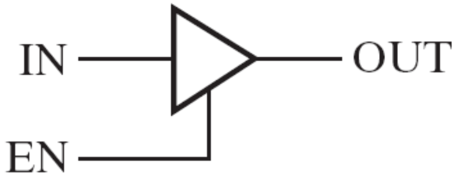
$F = (A \oplus B)'$
$F = AB + A'B'$

Funciones básicas (IV)

BUFFERS TRIESTADO

Los buffers triestado proporcionan un tercer estado en la salida denominado estado de **ALTA IMPEDANCIA**. Este valor se comporta como un circuito abierto, como si desconectáramos la salida.

EN	IN	OUT
0	X	**Alta Impedancia**
1	0	0
1	1	1



Lógica positiva y negativa

Existen dos maneras diferentes de asignar niveles de señal a valores lógicos:

- Si se elige el nivel alto H (HIGH) para representar el '1' lógico, hablamos de sistema en **lógica positiva**.
- Si se elige el nivel bajo L (LOW) para representar el '1' lógico, hablamos de sistema en **lógica negativa**.

Las puertas lógicas y los sistemas digitales en la práctica (I)

- Las puertas lógicas (y los sistemas digitales en general) se construyen a partir de **circuitos integrados** (IC). Un IC es un cristal semiconductor de silicio (a veces llamado 'chip'), que contiene componentes electrónicos como puertas lógicas o elementos de almacenamiento.
- Para formar un IC, el chip se monta sobre un soporte cerámico o plástico, y las conexiones se sueldan desde el chip hasta pines externos. El número de estos pines pueden oscilar desde 14 en los chips digitales más pequeños hasta varios cientos en los encapsulados mayores.
- Según el número de puertas contenidas en un chip (nivel de integración), podemos clasificar los IC en:
  - **Integración a pequeña escala** (**SSI**, *Small Scale Integration*). Contienen algunas puertas dentro del chip.
  - **Integración a media escala** (**MSI**, *Medium Scale Integration*). De 10 a 100 puertas por chip.
  - **Alta escala de integración** (**LSI**, *Large Scale Integration*). Entre 100 y algunos miles de puertas por encapsulado.
  - **Muy alta escala de integración** (**VLSI**, *Very Large Scale Integration*). Pueden contener desde varios miles hasta decenas de millones de puertas por chip.

Las puertas lógicas y los sistemas digitales en la práctica (II)

TECNOLOGÍAS DE CIRCUITOS

Los IC digitales también pueden clasificarse según la tecnología empleada para su implementación. Existen varias tecnologías: TTL, CMOS, basados en Arseniuro de Galio, etc.

Una de las más extendidas es la tecnología CMOS, por su elevada densidad circuital. Dentro de estas tecnologías existen diferentes versiones, que pueden proporcionar algunas mayor velocidad, otras menor consumo, etc.

PARÁMETROS TECNOLÓGICOS

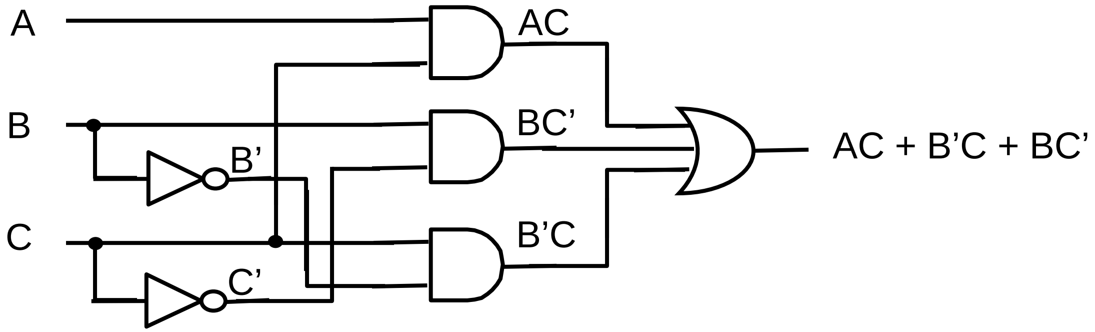
Los principales parámetros que caracterizan cualquier tecnología son:

- **Fan-in**: Especifica el número de entradas disponibles en una puerta.
- **Fan-out**: Especifica el número de cargas estándar atacadas por la salida de una puerta. El fan-out máximo para una salida determina el número de entradas que la salida de la puerta puede atacar sin afectar a las prestaciones de dicha puerta.
- **Margen de ruido**: Máximo nivel de voltaje de ruido externo que, superpuesto a la señal de entrada, no provoca cambios indeseados en la salida de la puerta.
- **Tiempo de propagación**. Es el tiempo necesario para que un cambio de señal se propague desde la entrada a la salida de la puerta.
- **Potencia disipada**: Potencia extraída de la fuente de alimentación y consumida por la puerta. Esta potencia se disipa en forma de calor (hay que tenerla presente en función de la temperatura de trabajo y de los requisitos de refrigeración del chip).

Implementación de funciones booleanas mediante conjuntos completos (I)

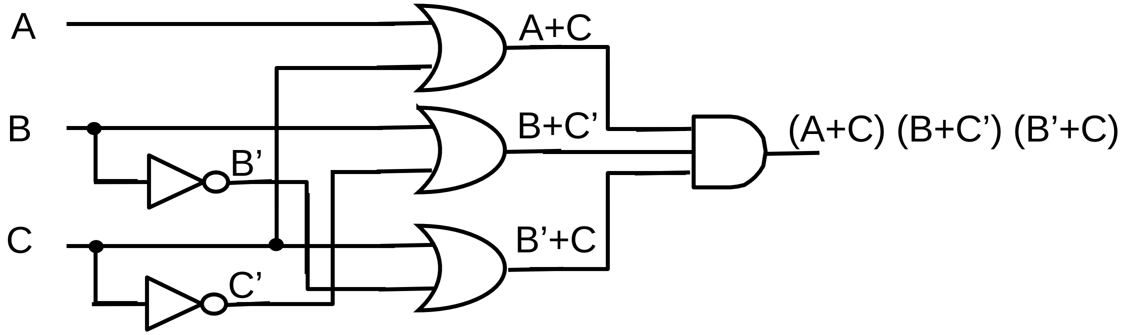
- **NOT-AND-OR** (preferentemente con SUMA de PRODUCTOS)

Ejemplo: **F(A,B,C) = AC + B'C + BC'**



- **NOT-OR-AND** (preferentemente con PRODUCTO de SUMAS)

Ejemplo: **F(A,B,C) = (A+C) (B+C') (B'+C)**



Implementación de funciones booleanas mediante conjuntos completos (II)

- **NAND-NAND** (preferentemente con SUMA de PRODUCTOS)

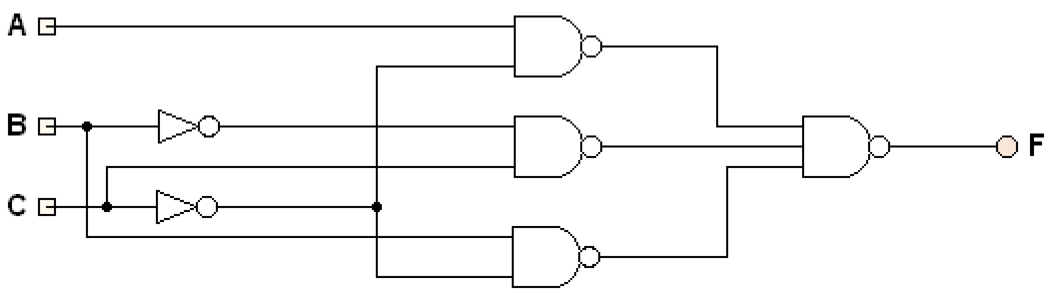
  Buscamos grupos de variables con la forma de salida de una puerta NAND.

Ejemplo:

$$F(A,B,C) = A\overline{C} + \overline{B}C + B\overline{C}$$

Negamos 2 veces: $$\overline{\overline{F(A,B,C)}} = \overline{\overline{A\overline{C} + \overline{B}C + B\overline{C}}}$$

Aplicamos DeMorgan: $$F(A,B,C) = \overline{\overline{A\overline{C}} \cdot \overline{\overline{B}C} \cdot \overline{B\overline{C}}}$$



Implementación de funciones booleanas mediante conjuntos completos (III)

- **NOR-NOR** (preferentemente con PRODUCTO de SUMAS)

  Buscamos grupos de variables con la forma de salida de una puerta NOR.

  Ejemplo:

$$F(A,B,C) = (A+C)\,(B+\overline{C})\,(\overline{B}+C)$$

Negamos 2 veces:

$$\overline{\overline{F(A,B,C)}} = \overline{\overline{(A+C)\,(B+\overline{C})\,(\overline{B}+C)}}$$

Aplicamos DeMorgan:

$$F(A,B,C) = \overline{\overline{(A+C)} + \overline{(B+\overline{C})} + \overline{(\overline{B}+C)}}$$