FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

86.01 Técnica Digital

Álgebra de Boole

Ing. Jorge H. Fuchs

Introducción

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Objetivos de la clase:

Analizar el concepto de sistema digital y su forma de estudio.

Estudiar la base matemática que permita la descripción, el análisis y la síntesis de un sistema digital.

Desarrollar los conceptos de funciones lógicas como herramientas de descripción del comportamiento de los circuitos elementales (compuertas) utilizados como bloques unitarios constructivos de otros más complejos.

Definiciones

Un álgebra puede definirse con un conjunto de elementos, un conjunto de operadores y un número de axiomas no probados o **postulados**.

En 1854 George **Boole** presentó un tratamiento sistemático de la lógica y desarrolló para este propósito un sistema algebraico que ahora se conoce como álgebra de Boole o álgebra booleana.

En 1938 Claude E. **Shannon** introdujo un álgebra booleana de dos valores denominada álgebra de interruptores, en la cual demostró que las propiedades de los circuitos eléctricos biestables con interruptores pueden representarse con este álgebra.

Para la definición formal del álgebra booleana, se emplean los postulados formulados por Edward V. **Huntington** en 1904. Estos postulados o axiomas no son únicos para definir el álgebra booleana, también se han usado otros conjuntos de postulados distintos a los que plantearemos aquí.

Definiciones

El álgebra booleana es una estructura algebraica definida en un conjunto de elementos M junto con dos operadores binarios + y • siempre y cuando se cumpla con los postulados de Huntington. Al operador binario + se lo denomina unión, disyunción o suma lógica (**OR**), y al operador binario • se lo denomina intersección, conjunción o producto lógico (**AND**). Además, tenemos una tercera operación que es llamada complemento, negación o inversión (**NOT**). Las operaciones OR, AND y NOT son las fundamentales de este álgebra.

El álgebra booleana rige la lógica binaria, la teoría de conjuntos, la lógica proposicional, la lógica de contactos, interruptores o pulsadores, etc.

Principio de Dualidad

Establece que a toda relación o ley lógica le corresponderá su ley dual, formada mediante el intercambio de los operadores de suma con los de producto, y de los 1s con los 0s, y viceversa.

Postulados de Huntington

P1) Conmutatividad

P1a) $A + B = B + A$

P1b) $A \bullet B = B \bullet A$

P2) Distributividad

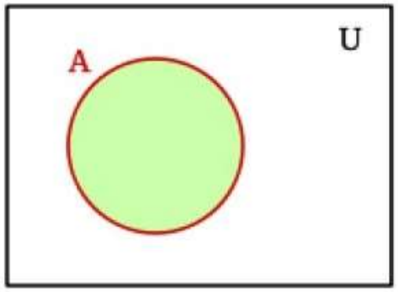
P2a) $A \bullet (B + C) = (A \bullet B) + (A \bullet C)$

P2b) $A + (B \bullet C) = (A + B) \bullet (A + C)$

P3) Identidad o Invariancia

P3a) $A + 0 = A$

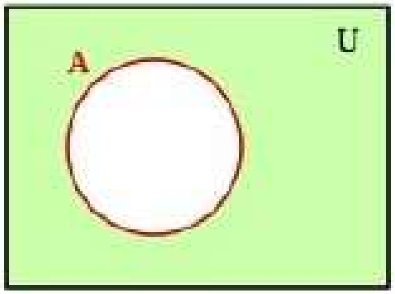
P3b) $A \bullet 1 = A$



P4) Complemento

P4a) $A + \bar{A} = 1$

P4b) $A \bullet \bar{A} = 0$

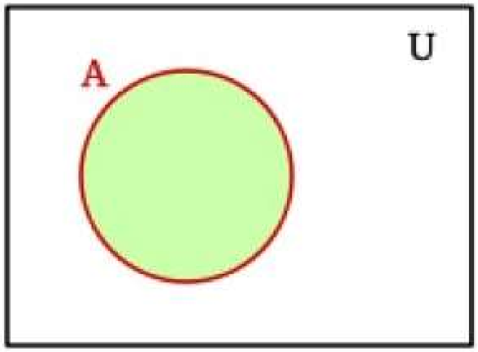


Teoremas o Propiedades

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Los **teoremas** se demuestran mediante los **postulados de Huntington**

T1) <u>Idempotencia</u>



T1a) $\mathbf{A + A = A}$

Demostración:

$$
\begin{aligned}
\mathbf{A + A} &= \mathbf{(A + A) \bullet 1} && \text{por P3b)} \\
&= \mathbf{(A + A) \bullet (A + \bar{A})} && \text{P4a)} \\
&= \mathbf{A + (A \bullet \bar{A})} && \text{P2b)} \\
&= \mathbf{A + 0} && \text{P4b)} \\
&= \mathbf{A} && \text{P3a)} \qquad \text{l.q.q.d.}
\end{aligned}
$$

Teoremas o Propiedades

T1b) $\mathbf{A \bullet A = A}$

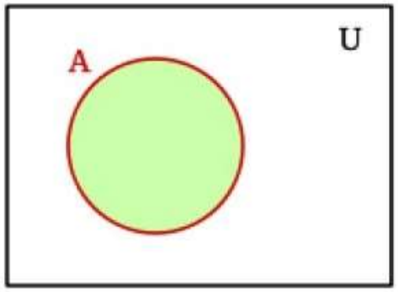
Demostración:

$$
\begin{aligned}
\mathbf{A \bullet A} \quad & \mathbf{= (A \bullet A) + 0} && \text{por P3a)} \\
& \mathbf{= (A \bullet A) + (A \bullet \bar{A})} && \text{P4b)} \\
& \mathbf{= A \bullet (A + \bar{A})} && \text{P2a)} \\
& \mathbf{= A \bullet 1} && \text{P4a)} \\
& \mathbf{= A} && \text{P3b)} \qquad \text{l.q.q.d.}
\end{aligned}
$$

T2) <u>Existencia de elementos nulos</u>

T2a) $\mathbf{A + 1 = 1}$

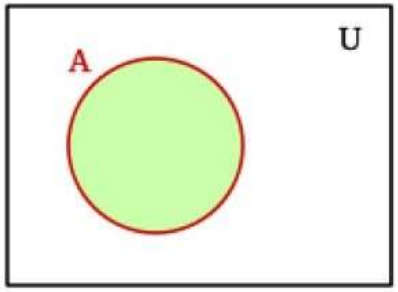
T2b) $\mathbf{A \bullet 0 = 0}$



Teoremas o Propiedades

T3) <u>Involución</u> (único)

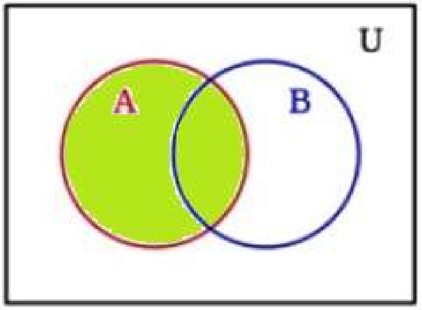
$$\overline{\overline{\mathbf{A}}} = \mathbf{A}$$



T4) <u>Absorción</u>

T4a) **A + A • B = A**

T4b) **A • (A + B) = A**



T5) <u>Asociatividad</u>

T5a) **A + (B + C) = (A + B) + C**

T5b) **A • (B • C) = (A • B) • C**

Teoremas o Propiedades

T6) <u>Leyes de De Morgan</u>

T6a) $$\overline{A + B} = \overline{A} \bullet \overline{B}$$

T6b) $$\overline{A \bullet B} = \overline{A} + \overline{B}$$

Corolarios:

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

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

Las leyes de De Morgan y sus corolarios son muy útiles y se aplican permanentemente ya que permiten transformar sumas en productos y viceversa.

Función lógica y Tabla de verdad

Una función lógica booleana es una expresión formada por variables lógicas binarias y operaciones lógicas (OR, AND, NOT).

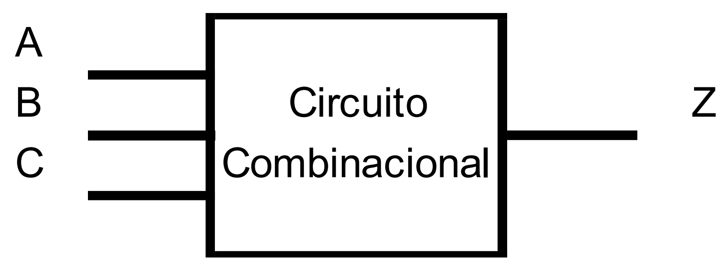
Para conocer el valor de Z para diferentes valores de las variables A, B y C se genera la **tabla de verdad** de la función. Para **n** variables tiene **2^n** filas.

Un ejemplo de función booleana Z = f(A,B,C) sería: Z = A + B • C.

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Ejemplo:

Z = Lámpara encendida
A = Llave encendida
B = Puerta cerrada
C = Ventana cerrada

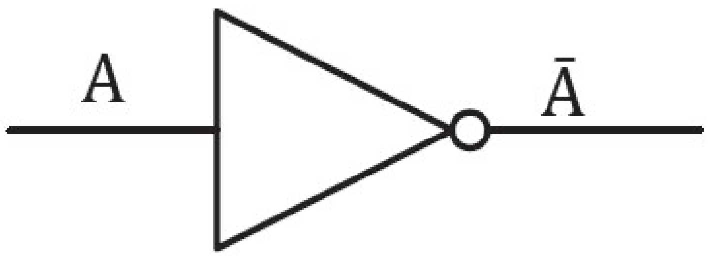


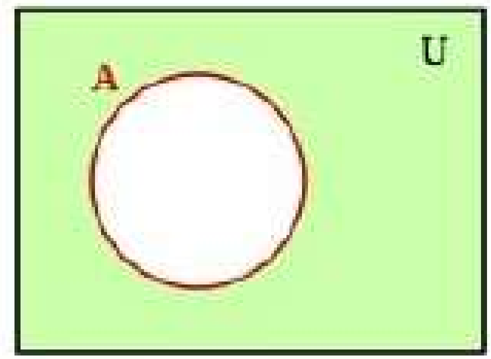
Operaciones y Compuertas lógicas

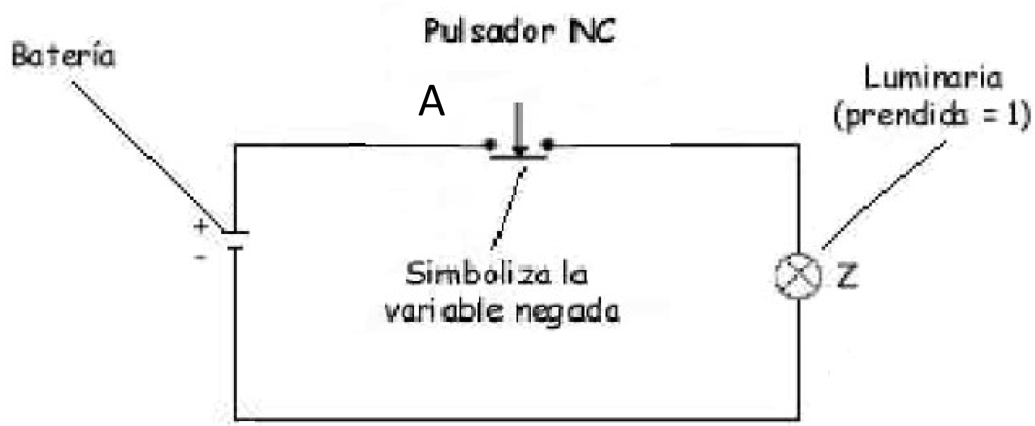
NOT: complemento, negación o inversión.

$$Z = \bar{A}$$

A	$\bar{A}$
0	1
1	0





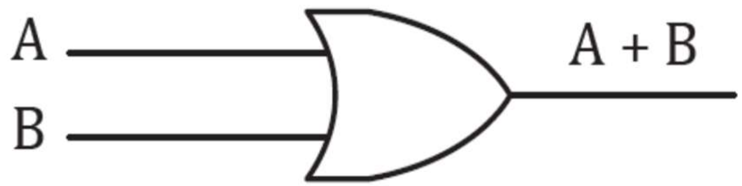


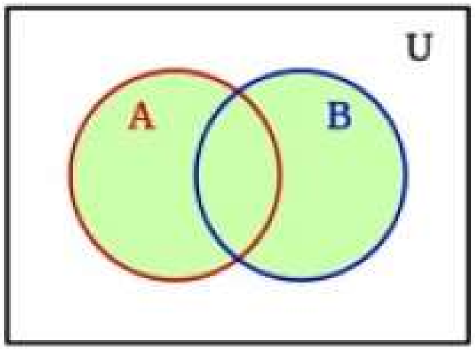
Operaciones y Compuertas lógicas

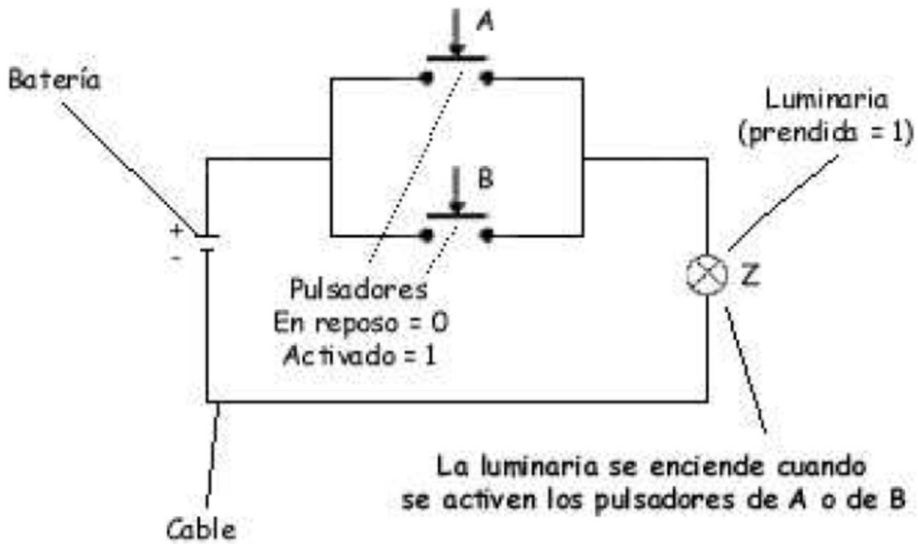
OR: operación "O", suma lógica, disyunción o unión

$$Z = A + B$$

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







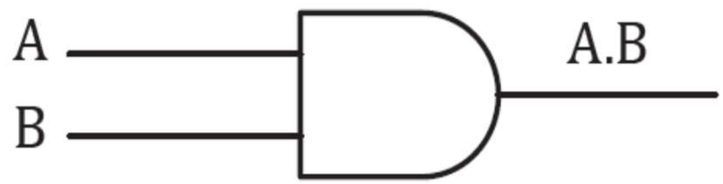
Operaciones y Compuertas lógicas

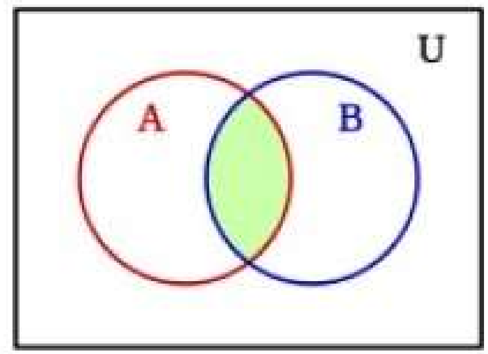
FACULTAD DE INGENIERIA
Universidad de Buenos Aires

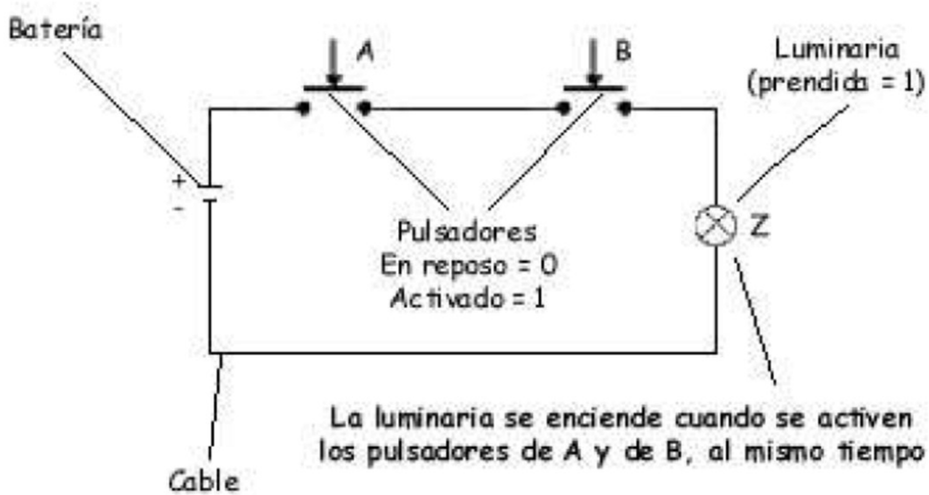
AND: operación “Y”, producto lógico, conjunción o intersección

$$Z = A \bullet B$$

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1





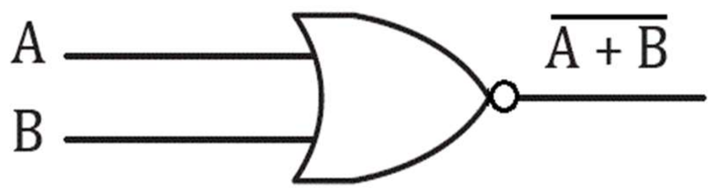


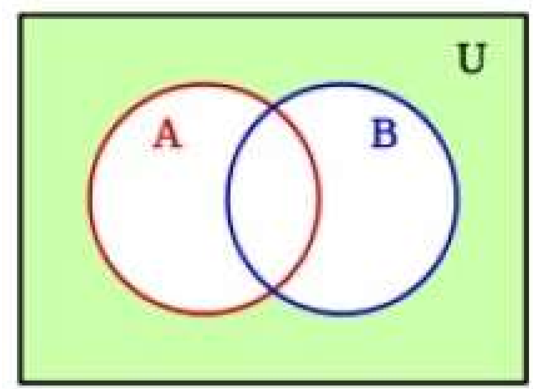
Operaciones y Compuertas lógicas

NOR: operación "No O", suma lógica negada o disyunción opuesta

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

A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0



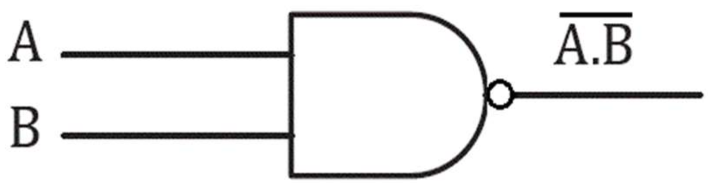


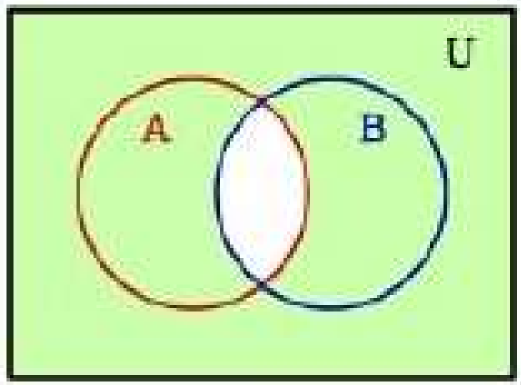
Operaciones y Compuertas lógicas

NAND: operación "No Y", producto lógico negado o conjunción opuesta

$$Z = \overline{A \bullet B}$$

A	B	$\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0





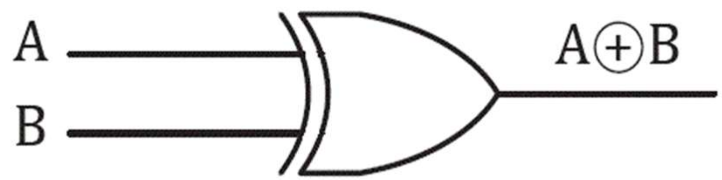
Operaciones y Compuertas lógicas

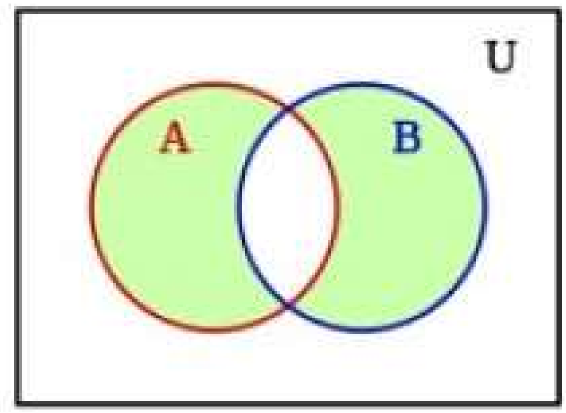
FACULTAD DE INGENIERIA
Universidad de Buenos Aires

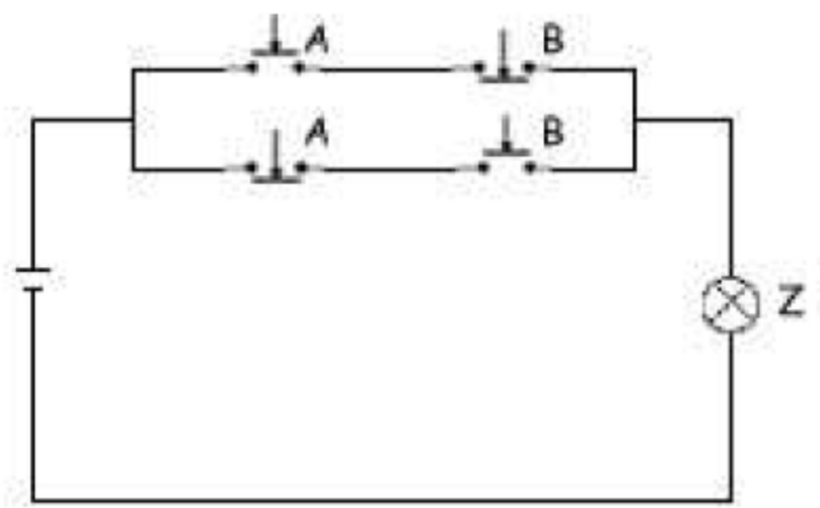
X-OR: operación "O Exclusiva", suma lógica exclusiva o disyunción exclusiva

$$Z = A \oplus B$$

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0





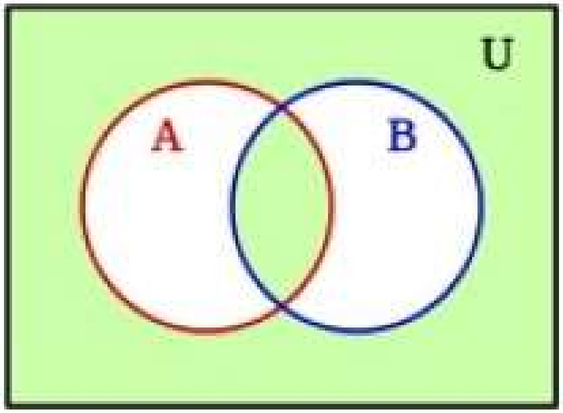


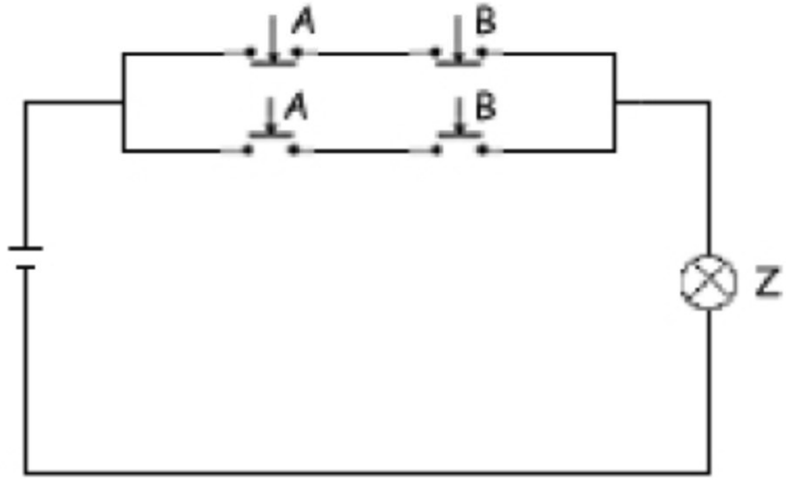
Operaciones y Compuertas lógicas

X-NOR: operación "No O Exclusiva" o suma lógica exclusiva negada

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

A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

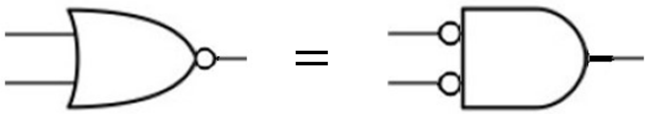




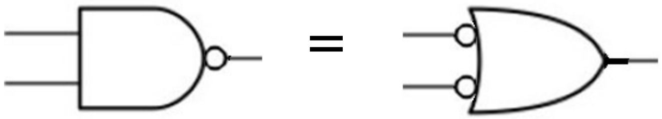
Equivalencia de compuertas

<u>**Leyes de De Morgan**</u>

$$\overline{A + B} = \bar{A} \bullet \bar{B}$$

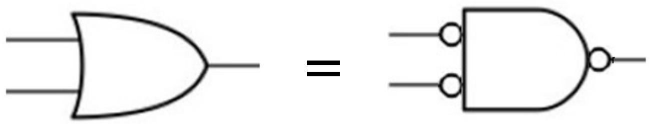


$$\overline{A \bullet B} = \bar{A} + \bar{B}$$

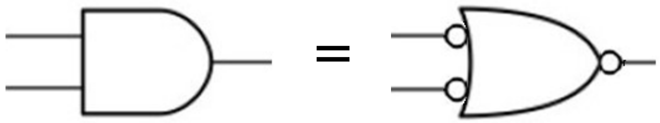


Corolarios:

$$A + B = \overline{\bar{A} \bullet \bar{B}}$$



$$A \bullet B = \overline{\bar{A} + \bar{B}}$$



Las leyes de De Morgan permiten transformar sumas en productos y viceversa.

Operaciones con registros

Las operaciones lógicas se pueden aplicar no solo a variables lógicas (1 bit) sino también a registros (conjunto de bits).

Por ejemplo, tenemos 2 registros de 8 bits, A y B:

A = 10100110 B = 00111010

El resultado de la operación A **AND** B resulta:

$$Z = A \bullet B = (10100110) \bullet (00111010) = 00100010$$

El resultado de la operación A **OR** B resulta:

$$Z = A + B = (10100110) + (00111010) = 10111110$$

Operaciones **NOT**:

$$Z = NOT(A) = 01011001$$

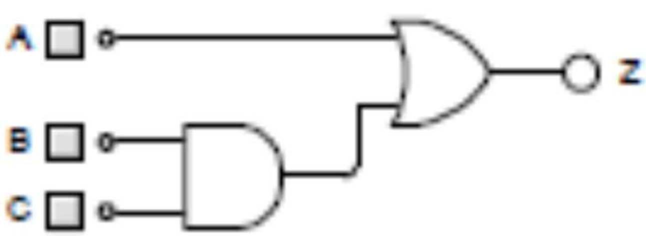
$$Z = NOT(A + B) = 01000001$$

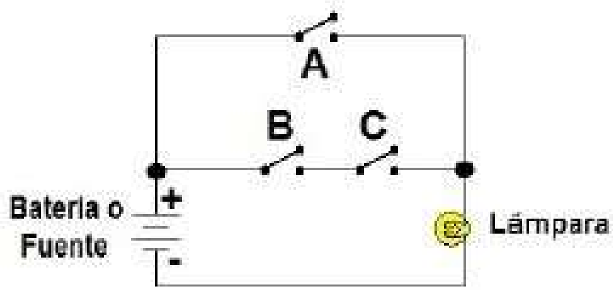
Proposiciones y Funciones lógicas

Supongamos la siguiente **proposición**: la lámpara (Z) deberá encenderse cuando el interruptor (A) se encuentre encendido **o** cuando la puerta (sensor B) **y** la ventana (sensor C) estén ambas cerradas. En este caso **simple** podemos obtener la **función**: $Z = A + B \bullet C$

Z = Lámpara encendida
A = Interruptor encendido
B = Puerta cerrada
C = Ventana cerrada

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



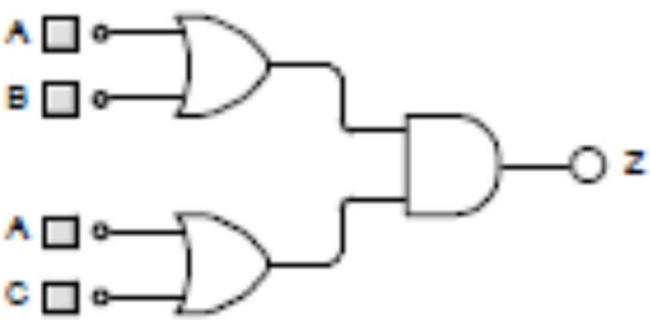


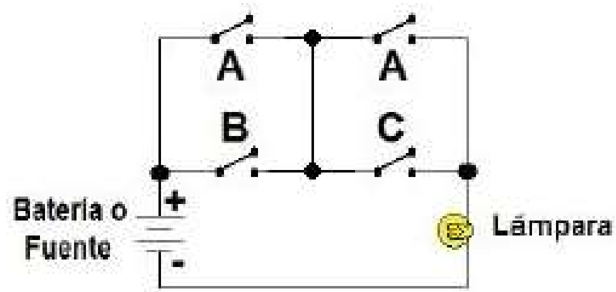
Proposiciones y Funciones lógicas

De la función anterior: $Z = A + B \bullet C$

Operando obtenemos: $Z = (A + B) \bullet (A + C)$

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1





Son **funciones equivalentes**, responden a la misma tabla de verdad, pero tienen distinto **costo lógico**.

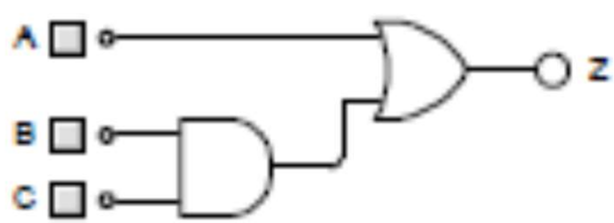
Función mínima - Costo lógico

Uno de los objetivos del estudio del Álgebra de Boole es encontrar la expresión mínima de una función o **función mínima**.

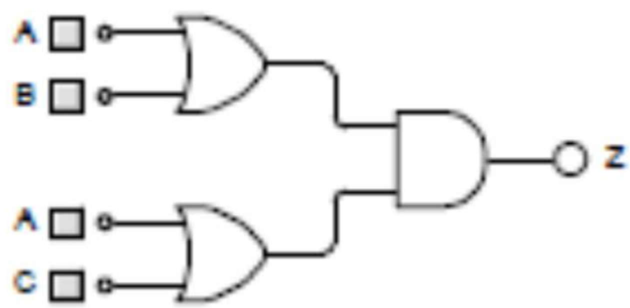
Para una determinada TV, la función mínima es la que tiene el menor **costo lógico**. Para comparar funciones equivalentes verificamos los siguientes criterios:

1) Menor cantidad de niveles (menos retardo)
2) Menor cantidad de compuertas
3) Menor cantidad de entradas

Z = A + B • C



Z = (A + B) • (A + C)



Formas normales o canónicas

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Minitérmino (m): Es un producto lógico (AND) en el cual figuran **todas** las variables lógicas de la función. Estas variables pueden estar o no negadas.

Maxitérmino (M): Es una suma lógica (OR) en la cual figuran **todas** las variables lógicas de la función. Estas variables pueden estar o no negadas.

Forma Normal Disyuntiva:

Está dada por la sumatoria de los **minitérminos** para los cuales la función vale 1. Es decir, es una suma de productos (**SPm**).

Forma Normal Conjuntiva

Está dada por el producto de los **maxitérminos** para los cuales la función vale 0. Es decir es un producto de sumas (**PSM**).

Toda función lógica se puede representar por sus **FN**. (¿pero es la **mínima**?)

Formas normales o canónicas

Hallaremos las formas normales de una función a partir de su **TV**:

FND (SPm): En las filas donde la función vale 1 se forma el producto de todas las variables, remplazando los 0's por su respectiva variable negada, y los 1's por su correspondiente variable sin negar. Luego se realiza la suma lógica de los minitérminos así determinados. El número de minitérminos será igual al número de 1's en la TV de la función.

A	B	C	Z	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{A}BC$
1	0	0	1	$A\overline{B}\overline{C}$
1	0	1	1	$A\overline{B}C$
1	1	0	1	$AB\overline{C}$
1	1	1	1	ABC

$$Z = f(A, B, C) = \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C} + ABC$$

Formas normales o canónicas

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

De la misma forma y teniendo en cuenta el principio de Dualidad:

FNC (PSM): En las filas donde la función vale 0 se forma la suma de todas las variables, reemplazando los 1's por su respectiva variable negada, y los 0's por su correspondiente variable sin negar. Luego se realiza el producto lógico de los maxiterminos así constituidos. El número de maxitérminos será igual al número de 0's en la TV de la función.

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$$\boldsymbol{A+B+C}$$
$$\boldsymbol{A+B+\overline{C}}$$
$$\boldsymbol{A+\overline{B}+C}$$

$$\boldsymbol{Z = f(A,B,C) = (A+B+C)\ (A+B+\overline{C})\ (A+\overline{B}+C)}$$

Minimización mediante Álgebra de Boole

Podemos **minimizar** cualquier función aplicando los postulados y propiedades del **Álgebra de Boole**.

Como ejemplo, a partir de una FND:

$$\boldsymbol{Z = ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\,\overline{C} + \overline{A}BC}$$

$$\boldsymbol{Z = AB\,(C + \overline{C}) + A\overline{B}\,(C + \overline{C}) + \overline{A}BC + {\color{red}ABC}}$$

$$\boldsymbol{Z = AB \; + A\overline{B} \; + (\overline{A} + A)\,BC}$$

$$\boldsymbol{Z = A\,(B \; + \overline{B}\,) + 1\,BC}$$

$$\boldsymbol{Z = A + BC}$$

Obtención de FNs mediante Álg. Boole

A partir de cualquier función podemos obtener ambas **FND** y **FNC** aplicando postulados y propiedades del **Álgebra de Boole**.

Como ejemplo, la función ya analizada:

$$Z = A + BC$$

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$$Z = A\,.\,1\,.\,1 + 1\,.\,BC$$

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

$$Z = ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\,\overline{C} + {\color{red}ABC} + \overline{A}BC$$

$$Z = ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\,\overline{C} + \overline{A}BC$$

Conclusiones

1) Estamos en condiciones de realizar **análisis** y **síntesis** de circuitos lógicos.
2) Una función de 3 niveles, puede ser mínima?
3) Simplificar, minimizar, sintetizar, aquí son sinónimos, pero para qué?

Menor cantidad de niveles — Mayor velocidad

Menor cantidad de compuertas — Menor consumo de energía

Menor tamaño

Menor costo económico

Métodos algorítmicos de minimización:

Mapa de Veitch – Karnaugh
Método de Quine – McCluskey
Método de Consenso, etc.

FACULTAD
DE INGENIERIA
Universidad de Buenos Aires