Álgebra de Boole y Compuertas

David Alejandro González Márquez

Clase disponible en: `https://github.com/fokerman/computingSystemsCourse`

Álgebra de Boole

Las computadoras operan de forma digital, con circuitos que pueden tomar **dos estados**: *verdadero* (1) y *falso* (0).

Para describir el funcionamiento de estos circuitos, vamos a necesitar un nuevo tipo de álgebra.

Álgebra de Boole

Las computadoras operan de forma digital, con circuitos que pueden tomar **dos estados**: *verdadero* (1) y *falso* (0).

Para describir el funcionamiento de estos circuitos, vamos a necesitar un nuevo tipo de álgebra.

Álgebra de Boole - George Boole (1815-1864)

Sistema algebraico para el estudio sistemático de la lógica proposicional, donde las variables y funciones pueden adoptar dos estados. Resulta una formalización apropiada para representar información digital y permite expresar las operaciones que realizan los circuitos digitales.

Álgebra de Boole

Función Booleana

Toma una o más variables de entrada y produce un resultado que depende solo de los valores de entrada. Las variables pueden tener solo dos estados `0` (falso) o 1 (verdadero).

Álgebra de Boole

Función Booleana

Toma una o más variables de entrada y produce un resultado que depende solo de los valores de entrada. Las variables pueden tener solo dos estados `0` (falso) o 1 (verdadero).

Con n entradas, las funciones booleanas tienen 2^n combinaciones posibles para sus entradas.

Ejemplo

Para $f(A, B)$ las posibles entradas son: $f(0, 0)$, $f(0, 1)$, $f(1, 0)$ y $f(1, 1)$.

Tabla de verdad

Describe todas las posibles combinaciones de entradas y resultados para una función booleana.

Álgebra de Boole - Operadores

Tenemos tres operaciones básicas dentro del Álgebra de Boole.

NOT → $\overline{A}$

Invierte el valor de entrada de 1 a 0 y viceversa.

A	NOT
0	1
1	0

AND → $A \cdot B$

Verdadero (1) si y solo si ambas entradas son verdaderas.

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

OR → $A + B$

Verdadero (1) si al menos una entrada es verdadera.

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

Álgebra de Boole - Ejemplo

NOT	
A	$\overline{A}$
0	1
1	0

AND		
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR		
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Ejemplo:

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

Tabla de verdad

A	B	$(A \cdot B) + (\overline{A} \cdot \overline{B})$	f
0	0		
0	1		
1	0		
1	1		

Álgebra de Boole - Ejemplo

NOT	
A	$\overline{A}$
0	1
1	0

AND		
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR		
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Ejemplo:

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

Tabla de verdad

A	B	$(A \cdot B) + (\overline{A} \cdot \overline{B})$	f
0	0	$(0 \cdot 0) + (\overline{0} \cdot \overline{0}) \rightarrow$	
0	1		
1	0		
1	1		

Álgebra de Boole - Ejemplo

NOT	
A	$\overline{A}$
0	1
1	0

AND		
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR		
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Ejemplo:

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

Tabla de verdad

A	B	$(A \cdot B) + (\overline{A} \cdot \overline{B})$	f
0	0	$(0 \cdot 0) + (\overline{0} \cdot \overline{0}) \rightarrow (0 \cdot 0) + (1 \cdot 1) \rightarrow$	
0	1		
1	0		
1	1		

Álgebra de Boole - Ejemplo

NOT	
A	$\overline{A}$
0	1
1	0

AND		
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR		
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Ejemplo:

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

Tabla de verdad

A	B	$(A \cdot B) + (\overline{A} \cdot \overline{B})$	f
0	0	$(0 \cdot 0) + (\overline{0} \cdot \overline{0}) \rightarrow (0 \cdot 0) + (1 \cdot 1) \rightarrow 0 + 1$	1
0	1		
1	0		
1	1		

Álgebra de Boole - Ejemplo

NOT	
A	$\overline{A}$
0	1
1	0

AND		
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR		
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Ejemplo:

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

Tabla de verdad

A	B	$(A \cdot B) + (\overline{A} \cdot \overline{B})$	f
0	0	$(0 \cdot 0) + (\overline{0} \cdot \overline{0}) \rightarrow (0 \cdot 0) + (1 \cdot 1) \rightarrow 0 + 1$	1
0	1	$(0 \cdot 1) + (\overline{0} \cdot \overline{1}) \rightarrow (0 \cdot 1) + (1 \cdot 0) \rightarrow 0 + 0$	0
1	0	$(1 \cdot 0) + (\overline{1} \cdot \overline{0}) \rightarrow (1 \cdot 0) + (0 \cdot 1) \rightarrow 0 + 0$	0
1	1	$(1 \cdot 1) + (\overline{1} \cdot \overline{1}) \rightarrow (1 \cdot 1) + (0 \cdot 0) \rightarrow 1 + 0$	1

Álgebra de Boole

Propiedades para las operaciones $(\cdot)$ y $(+)$

Identidad	$1 \cdot A = A$	$0 + A = A$
Nulo	$0 \cdot A = 0$	$1 + A = 1$
Idempotencia	$A \cdot A = A$	$A + A = A$
Inverso	$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$
Conmutatividad	$A \cdot B = B \cdot A$	$A + B = B + A$
Asociatividad	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A + B) + C = A + (B + C)$
Distributividad	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C$
Absorción	$A \cdot (A + B) = A$	$A + A \cdot B = A$
De Morgan	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$

Álgebra de Boole

Ejemplo: reducir la siguiente expresión usando propiedades.

$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)}$$

Álgebra de Boole

Ejemplo: reducir la siguiente expresión usando propiedades.

$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)}$$

Solución:

$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)} \quad \longleftarrow \text{De Morgan}$$

Álgebra de Boole

Ejemplo: reducir la siguiente expresión usando propiedades.

$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)}$$

Solución:

$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)} \quad \longleftarrow \text{De Morgan}$$
$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{Y} \cdot \overline{Z} \quad \longleftarrow \text{distributiva}$$

Álgebra de Boole

Ejemplo: reducir la siguiente expresión usando propiedades.

$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)}$$

Solución:

$$\begin{aligned}
&\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)} && \longleftarrow \text{De Morgan} \\
&\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{Y} \cdot \overline{Z} && \longleftarrow \text{distributiva} \\
&\overline{(\overline{X} \cdot Y)} \cdot Z + (X + \overline{Y}) \cdot \overline{Z} && \longleftarrow \text{De Morgan}
\end{aligned}$$

Álgebra de Boole

Ejemplo: reducir la siguiente expresión usando propiedades.

$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)}$$

Solución:

$$\begin{array}{ll}
\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)} & \longleftarrow \text{De Morgan} \\
\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{Y} \cdot \overline{Z} & \longleftarrow \text{distributiva} \\
\overline{(\overline{X} \cdot Y)} \cdot Z + (X + \overline{Y}) \cdot \overline{Z} & \longleftarrow \text{De Morgan} \\
(X + \overline{Y}) \cdot Z + (X + \overline{Y}) \cdot \overline{Z} & \longleftarrow \text{distributiva}
\end{array}$$

Álgebra de Boole

Ejemplo: reducir la siguiente expresión usando propiedades.

$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)}$$

Solución:

$$\begin{aligned}
&\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)} && \longleftarrow \text{De Morgan} \\
&\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{Y} \cdot \overline{Z} && \longleftarrow \text{distributiva} \\
&\overline{(\overline{X} \cdot Y)} \cdot Z + (X + \overline{Y}) \cdot \overline{Z} && \longleftarrow \text{De Morgan} \\
&(X + \overline{Y}) \cdot Z + (X + \overline{Y}) \cdot \overline{Z} && \longleftarrow \text{distributiva} \\
&(X + \overline{Y}) \cdot (Z + \overline{Z}) && \longleftarrow \text{inverso}
\end{aligned}$$

Álgebra de Boole

Ejemplo: reducir la siguiente expresión usando propiedades.

$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)}$$

Solución:

$$\begin{aligned}
&\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)} && \longleftarrow \text{De Morgan} \\
&\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{Y} \cdot \overline{Z} && \longleftarrow \text{distributiva} \\
&\overline{(\overline{X} \cdot Y)} \cdot Z + (X + \overline{Y}) \cdot \overline{Z} && \longleftarrow \text{De Morgan} \\
&(X + \overline{Y}) \cdot Z + (X + \overline{Y}) \cdot \overline{Z} && \longleftarrow \text{distributiva} \\
&(X + \overline{Y}) \cdot (Z + \overline{Z}) && \longleftarrow \text{inverso} \\
&(X + \overline{Y}) \cdot 1 && \longleftarrow \text{identidad}
\end{aligned}$$

Álgebra de Boole

Ejemplo: reducir la siguiente expresión usando propiedades.

$$\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)} \qquad \rightarrow \qquad X + \overline{Y}$$

Solución:

$$\begin{aligned}
&\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{(Y + Z)} && \longleftarrow \text{De Morgan} \\
&\overline{(\overline{X} \cdot Y)} \cdot Z + X \cdot \overline{Z} + \overline{Y} \cdot \overline{Z} && \longleftarrow \text{distributiva} \\
&\overline{(\overline{X} \cdot Y)} \cdot Z + (X + \overline{Y}) \cdot \overline{Z} && \longleftarrow \text{De Morgan} \\
&(X + \overline{Y}) \cdot Z + (X + \overline{Y}) \cdot \overline{Z} && \longleftarrow \text{distributiva} \\
&(X + \overline{Y}) \cdot (Z + \overline{Z}) && \longleftarrow \text{inverso} \\
&(X + \overline{Y}) \cdot 1 && \longleftarrow \text{identidad} \\
&X + \overline{Y}
\end{aligned}$$

Lógica Digital

Las computadoras están construidas mediante **circuitos electrónicos**.

Los circuitos operan de forma digital, tal que cada **cable** puede tomar dos estados posibles:

- Nivel lógico `0` — apagado — ausencia de tensión
- Nivel lógico 1 — encendido — presencia de tensión

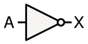
Los ladrillos que nos permiten construir circuitos digitales se denominan **compuertas**.

Las compuertas se construyen con componentes electrónicos como **transistores**,
la base de toda la computación moderna basada en semiconductores.

Compuertas

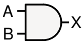
Las compuertas nos permiten hacer operaciones con las señales eléctricas (0 y 1). Toman una o dos entradas, y generan una salida.

Estos son ejemplos de las **compuertas básicas**, junto a una tabla que describe su comportamiento.



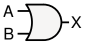
$\overline{A}$

A	NOT
0	1
1	0



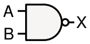
$A \cdot B$

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1



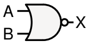
$A + B$

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1



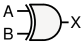
$\overline{A \cdot B}$

A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0



$\overline{A + B}$

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0



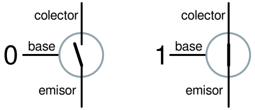
$A \oplus B$

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

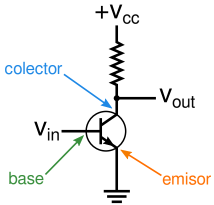
Compuertas - Funcionamiento interno

El funcionamiento interno de las compuertas supera los alcances de esta materia.

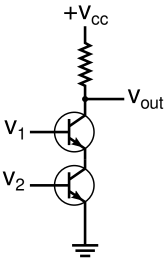
A modo ilustrativo los **transistores** funcionan como llave.
La presencia de tensión en la base, “conecta” el emisor con el colector.



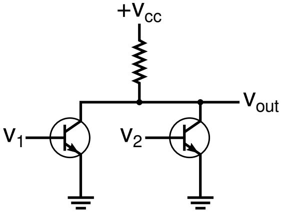
NOT



NAND



NOR



Construir funciones booleanas con su tabla de verdad

Dos formas canónicas de expresiones booleanas:

Construir funciones booleanas con su tabla de verdad

Dos formas canónicas de expresiones booleanas:

- Suma de Productos
  Expresión que hace la suma de todas las combinaciones que resulten en 1.

Construir funciones booleanas con su tabla de verdad

Dos formas canónicas de expresiones booleanas:

- Suma de Productos
  Expresión que hace la suma de todas las combinaciones que resulten en 1.
- Producto de Sumas
  Expresión que hace el producto de todas las combinaciones que resulten en 0.

Construir funciones booleanas con su tabla de verdad

Dos formas canónicas de expresiones booleanas:

- Suma de Productos
  Expresión que hace la suma de todas las combinaciones que resulten en 1.
- Producto de Sumas
  Expresión que hace el producto de todas las combinaciones que resulten en 0.

Ejemplo:

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

Suma de Productos

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

Construir funciones booleanas con su tabla de verdad

Dos formas canónicas de expresiones booleanas:

- Suma de Productos
  Expresión que hace la suma de todas las combinaciones que resulten en 1.
- Producto de Sumas
  Expresión que hace el producto de todas las combinaciones que resulten en 0.

Ejemplo:

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

Suma de Productos

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

Producto de Sumas

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

Construir circuitos a partir de una función booleana

Ejemplo:

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

Suma de Productos

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

Producto de Sumas

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

Construir circuitos a partir de una función booleana

Ejemplo:

Suma de Productos

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

Suma de Productos

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

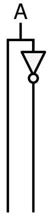
Producto de Sumas

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

Construir circuitos a partir de una función booleana

Ejemplo:

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



Suma de Productos

Suma de Productos

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

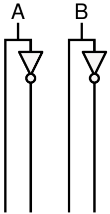
Producto de Sumas

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

Construir circuitos a partir de una función booleana

Ejemplo:

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



Suma de Productos

Suma de Productos

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

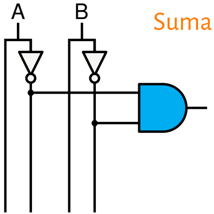
Producto de Sumas

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

Construir circuitos a partir de una función booleana

Ejemplo:

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



Suma de Productos

Suma de Productos

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

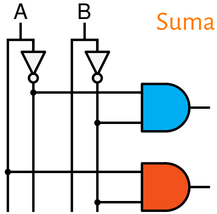
Producto de Sumas

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

Construir circuitos a partir de una función booleana

Ejemplo:

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



Suma de Productos

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

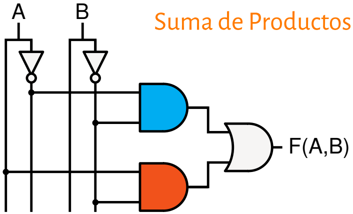
Producto de Sumas

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

Construir circuitos a partir de una función booleana

Ejemplo:

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



Suma de Productos

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

Producto de Sumas

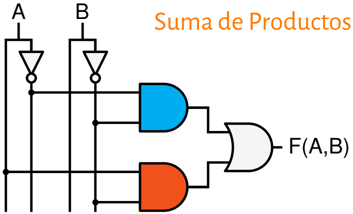
$$F(A,B) = (A + \overline{B}) \cdot (\overline{A} + \overline{B})$$

Construir circuitos a partir de una función booleana

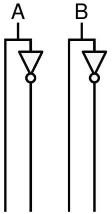
Ejemplo:

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

Suma de Productos



Producto de Sumas



Suma de Productos

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

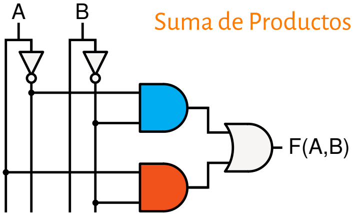
Producto de Sumas

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

Construir circuitos a partir de una función booleana

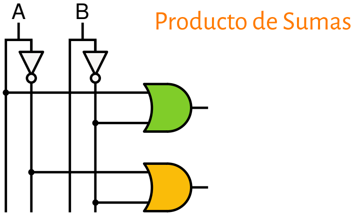
Ejemplo:

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



Suma de Productos

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



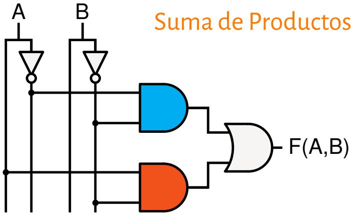
Producto de Sumas

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

Construir circuitos a partir de una función booleana

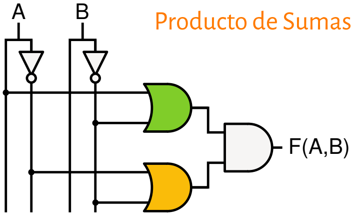
Ejemplo:

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



Suma de Productos

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



Producto de Sumas

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

Construir circuitos a partir de una función booleana

Ejemplo:

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

Suma de Productos

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

Producto de Sumas

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

Construir circuitos a partir de una función booleana

Ejemplo:

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

Suma de Productos

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

Producto de Sumas

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

Suma de Productos

$$F(A,B) = (\overline{A} \cdot \overline{B}) + (A \cdot \overline{B})$$
$$F(A,B) = (\overline{A} + A) \cdot \overline{B}$$
$$F(A,B) = 1 \cdot \overline{B}$$
$$F(A,B) = \overline{B}$$

Construir circuitos a partir de una función booleana

Ejemplo:

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

Suma de Productos

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

Producto de Sumas

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

Suma de Productos

$$F(A,B) = (\overline{A} \cdot \overline{B}) + (A \cdot \overline{B})$$
$$F(A,B) = (\overline{A} + A) \cdot \overline{B}$$
$$F(A,B) = 1 \cdot \overline{B}$$
$$F(A,B) = \overline{B}$$

Producto de Sumas

$$F(A,B) = (A + \overline{B}) \cdot (\overline{A} + \overline{B})$$
$$F(A,B) = (A \cdot \overline{A}) + \overline{B}$$
$$F(A,B) = 0 + \overline{B}$$
$$F(A,B) = \overline{B}$$

Construir circuitos a partir de una función booleana

Ejemplo:

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

Suma de Productos

$$F(A,B) = (\overline{A} \cdot \overline{B}) + (A \cdot \overline{B})$$
$$F(A,B) = (\overline{A} + A) \cdot \overline{B}$$
$$F(A,B) = 1 \cdot \overline{B}$$
$$F(A,B) = \overline{B}$$

Producto de Sumas

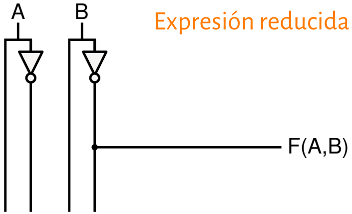
$$F(A,B) = (A + \overline{B}) \cdot (\overline{A} + \overline{B})$$
$$F(A,B) = (A \cdot \overline{A}) + \overline{B}$$
$$F(A,B) = 0 + \overline{B}$$
$$F(A,B) = \overline{B}$$

Suma de Productos

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

Producto de Sumas

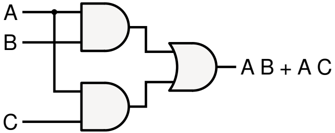
$$F(A,B) = (A + \overline{B}) \cdot (\overline{A} + \overline{B})$$



Expresión reducida

Equivalencia de circuitos

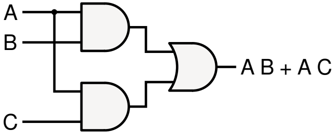
Existen múltiples formas de construir circuitos con compuertas, por ejemplo:

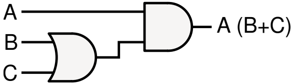


A	B	C	A·B	A·C	(A·B)+(A·C)
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Equivalencia de circuitos

Existen múltiples formas de construir circuitos con compuertas, por ejemplo:



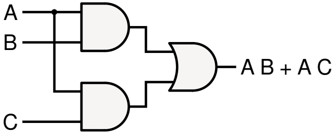


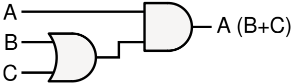
A	B	C	A·B	A·C	(A·B)+(A·C)
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

A	B	C	A	B+C	A·(B+C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Equivalencia de circuitos

Existen múltiples formas de construir circuitos con compuertas, por ejemplo:





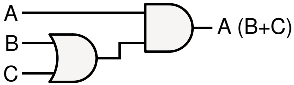
A	B	C	A·B	A·C	(A·B)+(A·C)
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

A	B	C	A	B+C	A·(B+C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Idealmente, se busca diseñar circuitos con la menor cantidad de componentes.

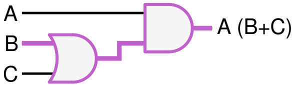
Tiempos de respuesta

Las compuertas son circuitos electrónicos y como tales tienen un tiempo de respuesta.



Tiempos de respuesta

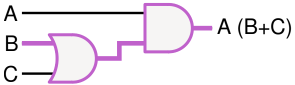
Las compuertas son circuitos electrónicos y como tales tienen un tiempo de respuesta.

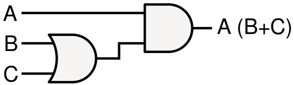


Modificar B requiere esperar que **dos** compuertas generen su salida.

Tiempos de respuesta

Las compuertas son circuitos electrónicos y como tales tienen un tiempo de respuesta.

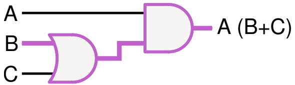


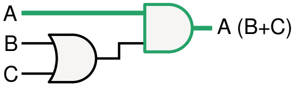


Modificar B requiere esperar que **dos** compuertas generen su salida.

Tiempos de respuesta

Las compuertas son circuitos electrónicos y como tales tienen un tiempo de respuesta.





Modificar B requiere esperar que **dos** compuertas generen su salida.
Modificar A requiere esperar que solo **una** compuerta genere su salida.

Tiempos de respuesta

Las compuertas son circuitos electrónicos y como tales tienen un tiempo de respuesta.

Modificar B requiere esperar que **dos** compuertas generen su salida.
Modificar A requiere esperar que solo **una** compuerta genere su salida.

Reducir la cantidad de compuertas por las que tiene que pasar una señal,
acelera los tiempos de respuesta.

Tiempos de respuesta

Las compuertas son circuitos electrónicos y como tales tienen un tiempo de respuesta.

Modificar B requiere esperar que **dos** compuertas generen su salida.
Modificar A requiere esperar que solo **una** compuerta genere su salida.

Reducir la cantidad de compuertas por las que tiene que pasar una señal,
acelera los tiempos de respuesta.

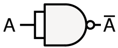
Los diseñadores de circuitos tratan de reducir el número de compuertas utilizando **circuitos equivalentes**.

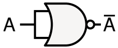
Compuertas universales

Existen dos tipos de compuertas denominadas **completas** o **universales**.
Con estas compuertas es posible construir cualquier circuito.

Veamos sus equivalencias:

NOT



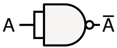


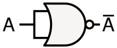
Compuertas universales

Existen dos tipos de compuertas denominadas **completas** o **universales**.
Con estas compuertas es posible construir cualquier circuito.

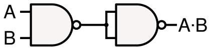
Veamos sus equivalencias:

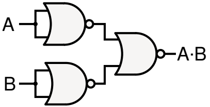
NOT





AND



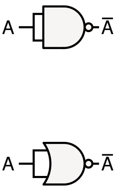


Compuertas universales

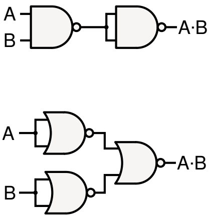
Existen dos tipos de compuertas denominadas **completas** o **universales**.
Con estas compuertas es posible construir cualquier circuito.

Veamos sus equivalencias:

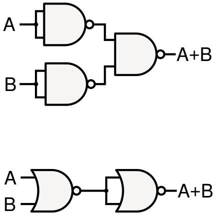
NOT



AND

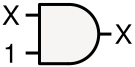


OR

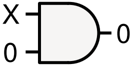


Uso de compuertas para seleccionar y habilitar

Habilitar

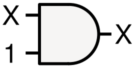


AND de X y 1 deja pasar X



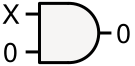
AND de X y 0 no deja pasar X

Uso de compuertas para seleccionar y habilitar

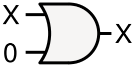


AND de X y 1 deja pasar X

Habilitar

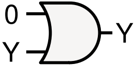


AND de X y 0 no deja pasar X



OR de X y 0 deja pasar X

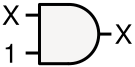
Seleccionar



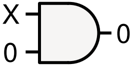
OR de Y y 0 deja pasar Y

Uso de compuertas para seleccionar y habilitar

Habilitar

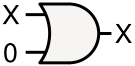


AND de X y 1 deja pasar X

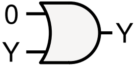


AND de X y 0 no deja pasar X

Seleccionar

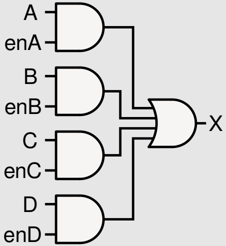


OR de X y 0 deja pasar X



OR de Y y 0 deja pasar Y

Ejemplo



Las señales enA, enB, enC y enD son de habilitación, para los datos de las señales A, B, C y D.

Bibliografía

- Tanenbaum, “Organización de Computadoras. Un Enfoque Estructurado”, 4ta Edición, 2000.
    - **Capítulo 3 - El nivel de lógica digital** - Páginas 117-127

- Null, “Essentials of Computer Organization and Architecture”, 5th Edition, 2018.
    - **Chapter 3 - Boolean Algebra and Digital Logic:**
        - 3.2 - Boolean Algebra
        - 3.3 - Logic Gates

Ejercicios

Con lo visto, ya pueden resolver los primeros tres ejercicios de la Guía de Lógica Digital.

¡Gracias!

Recuerden leer los comentarios adjuntos
en cada clase por aclaraciones.