



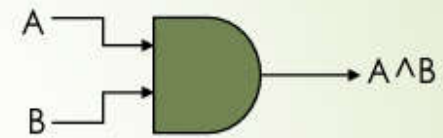
Compuertas lógicas

Compuerta lógica

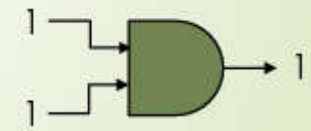
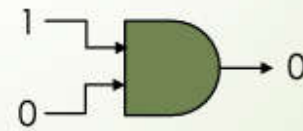
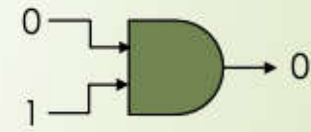
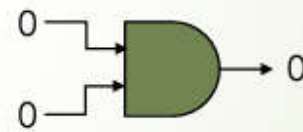
Es un dispositivo que implementa una función lógica simple.
Traduce un conjunto de entradas en **una sola salida**.

Compuertas lógicas: AND

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

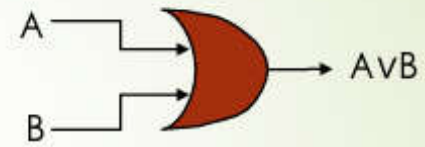


Casos:

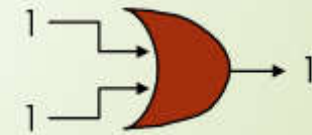
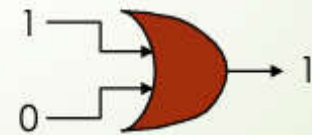
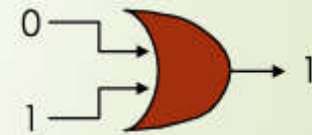
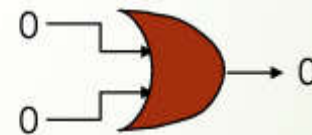


Compuertas lógicas: OR

A	B	A ∨ B
0	0	0
0	1	1
1	0	1
1	1	1



Casos:




Compuertas lógicas: NOT

A	$\neg A$
0	1
1	0




Casos:

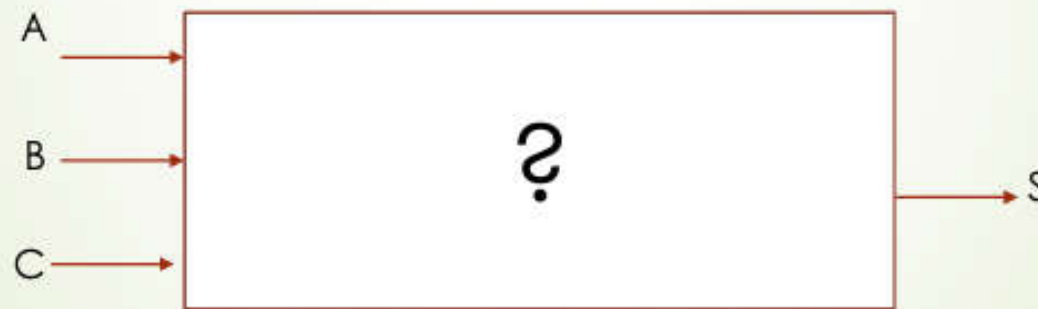




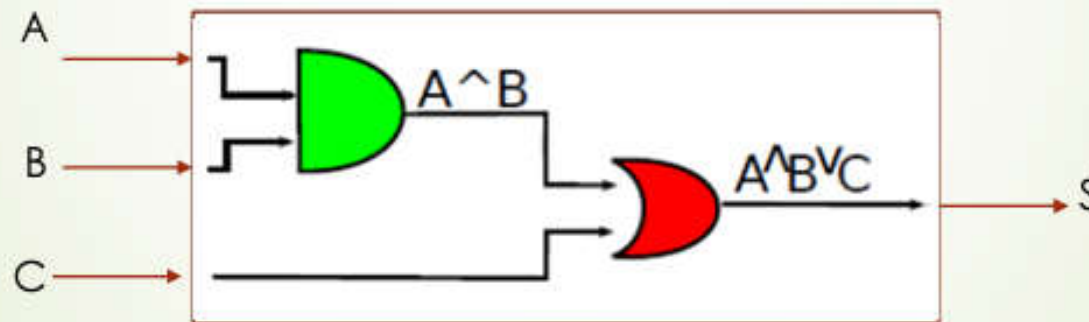
¿Qué son los circuitos?


- Puertas conectadas entre sí.
 - Traduce un conjunto de entradas en un conjunto de salidas, de acuerdo a una o más funciones lógicas.
 - Cada salida es función de las entradas.
 - Las salidas se actualizan de inmediato al cambiar las entradas.
- 

Ejemplo: ¿cómo es el circuito?
 $S = (A \wedge B) \vee C$





Ejemplo: ¿cómo es el circuito?
 $S = (A \wedge B) \vee C$





Los circuitos pueden obtenerse a partir de...

- Una fórmula
 - Una tabla de verdad
 - Un enunciado (en lenguaje natural)
- 



Los circuitos pueden obtenerse a partir de...

- Una fórmula (como en el ejemplo anterior)
- Una tabla de verdad ➡ Fórmula
- Un enunciado (en lenguaje natural) ➡ Tabla ➡ Fórmula

Diseñar un circuito

- 3 entradas
- Que calcule la función **mayoría**:
 - Si dos o más entradas valen 1, la salida es 1.
 - De lo contrario, la salida es 0.



Lenguaje Natural → Tabla de verdad

- 3 entradas, 1 salida
- Que calcule la función **mayoría**:
 - Si dos o más entradas valen 1, la salida es 1.
 - De lo contrario, la salida es 0.

E0	E1	E2	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Tabla de verdad → Fórmula Booleana (SoP)

E0	E1	E2	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- Construir la tabla de verdad
- Plantear la fórmula correspondiente a cada caso donde la salida vale "1"
- Unir los casos con disyunción

Tabla de verdad → Fórmula Booleana (SoP)

E0	E1	E2	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1 ($\bar{E}_0 \cdot E_1 \cdot E_2$)
1	0	0	0
1	0	1	1 ($E_0 \cdot \bar{E}_1 \cdot E_2$)
1	1	0	1 ($E_0 \cdot E_1 \cdot \bar{E}_2$)
1	1	1	1 ($E_0 \cdot E_1 \cdot E_2$)

- Construir la tabla de verdad
- Plantear la fórmula correspondiente a cada caso donde la salida vale "1"
- Unir los casos con disyunción

$$S = (\bar{E}_0 \cdot E_1 \cdot E_2) + (E_0 \cdot \bar{E}_1 \cdot E_2) + (E_0 \cdot E_1 \cdot \bar{E}_2) + (E_0 \cdot E_1 \cdot E_2)$$



Se obtiene la “Suma de productos” (SoP)

$$S = (\overline{E}_0 \cdot E_1 \cdot E_2) + (E_0 \cdot \overline{E}_1 \cdot E_2) + (E_0 \cdot E_1 \cdot \overline{E}_2) + (E_0 \cdot E_1 \cdot E_2)$$

SoP (Suma de Productos)

Fórmula booleana compuesta por disyunciones (\vee / +) entre “*minitérminos*”, que son términos que contienen todas las variables booleanas de la tabla (negada o no) unidas por conjunciones (\wedge / \cdot)

