

Compuertas Universales (NOR)

Compuerta NOR = compuerta universal

La compuerta **NOR** puede construir cualquier otra compuerta lógica.

Es suficiente por sí sola para implementar **toda lógica booleana**.

Su función básica es:

$$\text{NOR}(A, B) = \neg (A + B)$$



Cómo formar otras compuertas usando solo NOR

NOT con NOR

Usas la misma entrada dos veces.

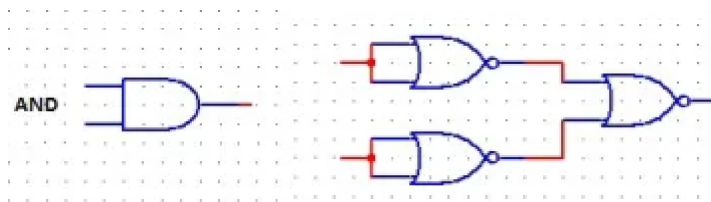
$$\neg A = \text{NOR}(A, A)$$



AND con NOR

Aplicamos De Morgan:

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



Implementación:

Con NOR:

$$A1 = \text{NOR}(A, A) \rightarrow \neg A$$

$$B1 = \text{NOR}(B, B) \rightarrow \neg B$$

$$\text{AND} = \text{NOR}(A1, B1)$$

OR con NOR

OR es la negación del NOR:

$$A + B = \neg (\text{NOR}(A, B))$$

Con NOR solamente:

$$X = \text{NOR}(A, B)$$

$$\text{OR} = \text{NOR}(X, X)$$

$$A + B = \neg (\text{NOR}(A, B))$$



Una sola compuerta puede implementar **todo**:

expresiones booleanas,
simplificaciones,
mapas de Karnaugh,
circuitos digitales completos.

Por eso se llaman **universales**.

Funcionamiento básico de la compuerta NOR

La compuerta NOR entrega:

1 solo cuando **ambas entradas son 0**.

En todos los demás casos, entrega **0**.



¿Por qué la NOR es universal?

Porque cumple dos requisitos fundamentales:

1. Puede generar la negación (NOT).

$$\neg A = \text{NOR}(A, A)$$

2. Puede generar las operaciones básicas AND y OR.

Y, una vez que tienes AND, OR y NOT, cualquier circuito digital puede construirse.

Toda la lógica booleana se basa en combinaciones de esas operaciones.

Ventajas de la compuerta NOR

Facilita el diseño físico de circuitos digitales.

Es ideal para tecnologías de puertas basadas en transistores MOS.

Reduce el número de tipos de compuertas necesarias en un circuito.

Permite armar redes lógicas completas usando un solo tipo de elemento.

En memoria RAM y ciertos diseños de microprocesadores, la compuerta NOR es muy utilizada debido a su eficiencia en MOSFET.

Relación con la implementación de hardware y mapas de Karnaugh

Los mapas de Karnaugh dan expresiones minimizadas en términos de AND, OR y NOT.

En hardware real, a menudo se convierten a NOR para uniformar la tecnología.

Ejemplo:

Si el mapa te da:

$$F = A + B \cdot C$$

Se convierte a NOR:

NOT B \rightarrow NOR(B, B)

NOT C \rightarrow NOR(C, C)

AND(B, C) \rightarrow NOR(NOT B, NOT C)

OR \rightarrow NOR(NOR(A,A), NOR(AND,AND))

Resultado: circuito completamente implementado con NOR.