



# Universidad del Caribe

2000

CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

**Organización y diseño de computadoras**

**Compuertas lógicas básicas**

**Adolfo Tun Dzul - 170300124**

**PROFESOR:  
Ismael Jiménez Sánchez**

## Compuertas lógicas básicas

Las Compuertas Lógicas son circuitos electrónicos conformados internamente por transistores que se encuentran con arreglos especiales con los que otorgan señales de voltaje como resultado o una salida de forma booleana, están obtenidos por operaciones lógicas binarias (suma, multiplicación).

Las compuertas lógicas trabajan en dos estados "1" y "0", los cuales pueden asignarse a la lógica positiva o lógica negativa.

- La lógica positiva es aquella que con una señal en alto se acciona, representando un 1 binario y con una señal en bajo se desactiva. representado un 0 binario.
- La lógica negativa proporciona los resultados inversamente, una señal en alto se representa con un 0 binario y una señal en bajo se representa con un 1 binario.

A continuación se presentaran las compuertas lógicas básicas.

### Compuerta AND

Esta compuerta es representada por una multiplicación en el Algebra de Boole. Indica que es necesario que en todas sus entradas se tenga un estado binario 1 para que la salida otorgue un 1 binario. En caso contrario de que falte alguna de sus entradas con este estado la salida permanecerá en 0.

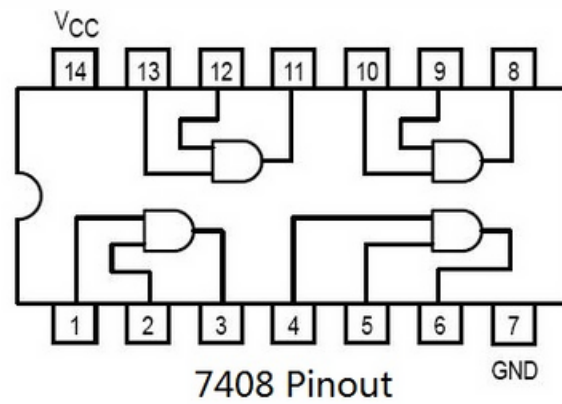
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

$$Q = A \cdot B$$

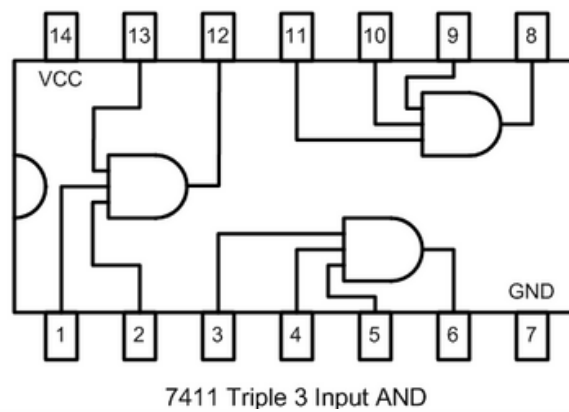


## Pinout AND

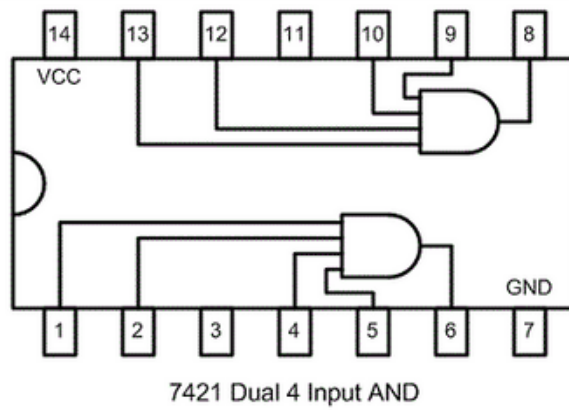
### 7408 de 2 entradas



### 7411 de 3 entradas



### 7421 de 4 entradas



## Compuerta OR

Esta compuerta permite que con cualquiera de sus entradas que este en estado binario 1, su salida pasara a un estado 1 también. Para lograr un estado 0 a la salida, todas sus entradas deben estar en el mismo valor de 0.

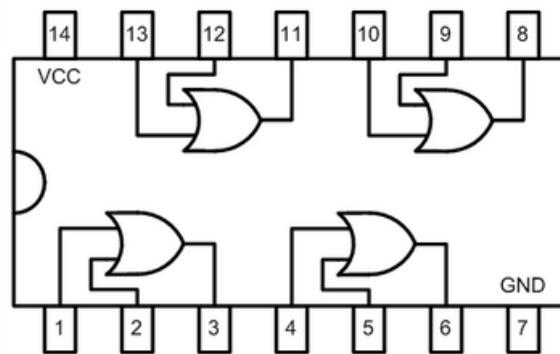
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

$$Q = A + B$$



## Pinout OR

### 7432 de 2 entradas



7432 Quad 2 Input OR

## Compuerta NOT

En este caso esta compuerta solo tiene una entrada y una salida y esta actúa como un inversor. Para esta situación en la entrada se colocara un 1 y en la salida otorgara un 0 y en el caso contrario esta recibirá un 0 y mostrara un 1.

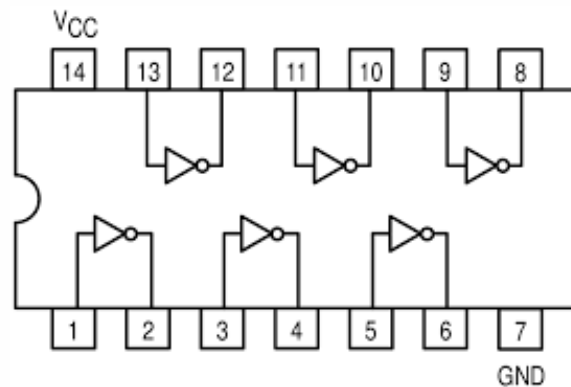
Q	Q'
0	1
1	0

$$Q = \bar{Q}$$



### Pinout NOT

7404 (x6)



### Compuerta NAND

También denominada como AND negada, esta compuerta trabaja al contrario de una AND ya que al no tener entradas en 1 o solamente alguna de ellas, esta concede un 1 en su salida, pero si esta tiene todas sus entradas en 1 la salida se presenta con un 0.

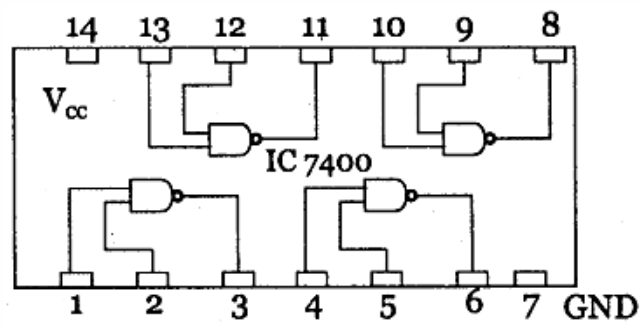
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

$$Q = \overline{A * B}$$

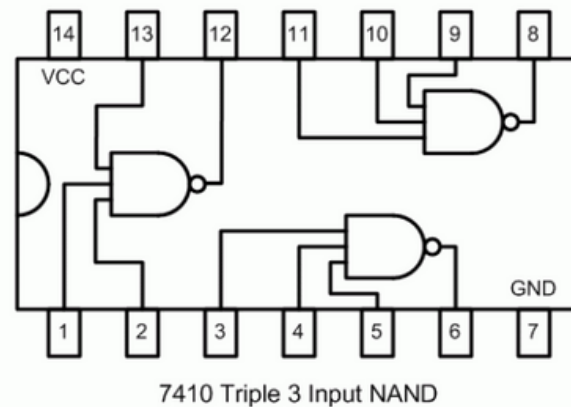


## Pinout NAND

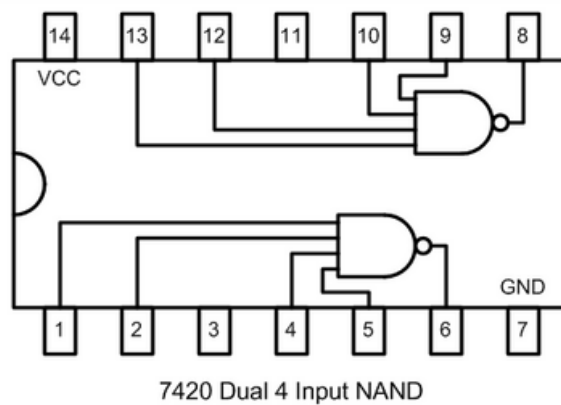
### 7400 de 2 entradas



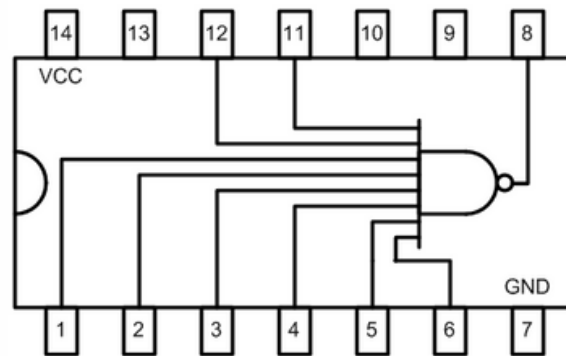
### 7410 de 3 entradas



### 74120 de 4 entradas

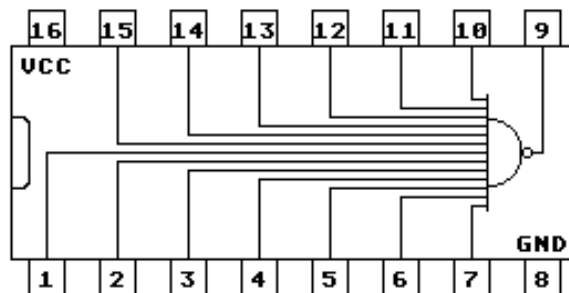


### 7430 de 4 entradas



7430 8 Input NAND

### 74133 de 13 entradas



## Compuerta NOR

La compuerta OR también tiene su versión inversa. Esta compuerta cuando tiene sus entradas en estado 0 su salida estará en 1, pero si alguna de sus entradas pasa a un estado 1 sin importar en qué posición, su salida será un estado 0.

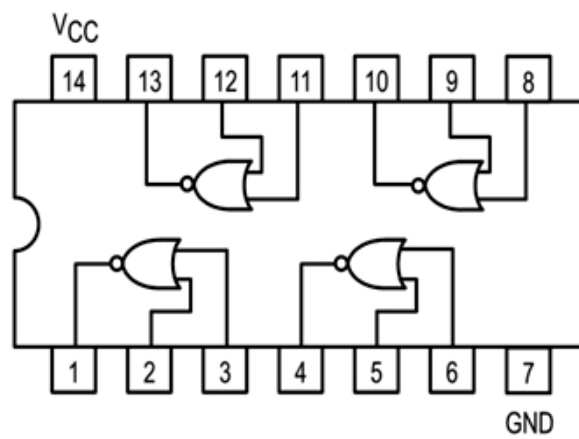
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

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

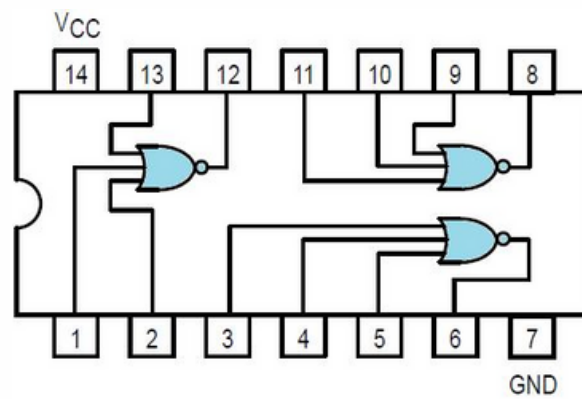


## Pinout NOR

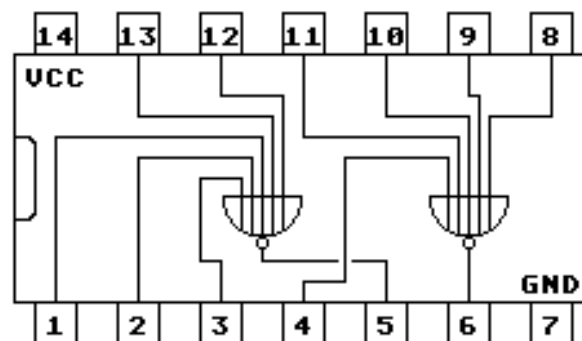
### 7402 de 2 entradas



### 7427 de 3 entradas



### 74260 de 5 entradas





## Compuerta XOR

También llamada OR exclusiva, esta actúa como una suma binaria de un dígito cada uno y el resultado de la suma sería la salida. Otra manera de verlo es que con valores de entrada igual el estado de salida es 0 y con valores de entrada diferente, la salida será 1.

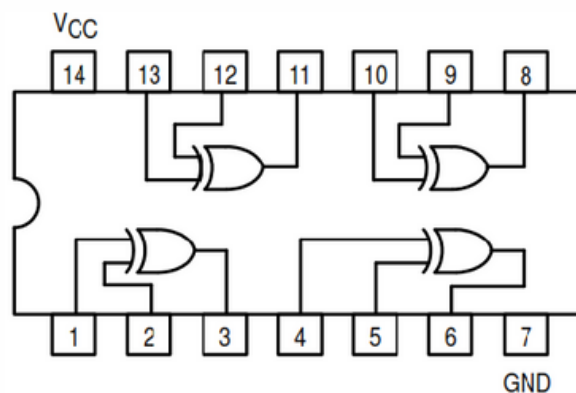
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

$$Q = A * B + \overline{A} * \overline{B}$$

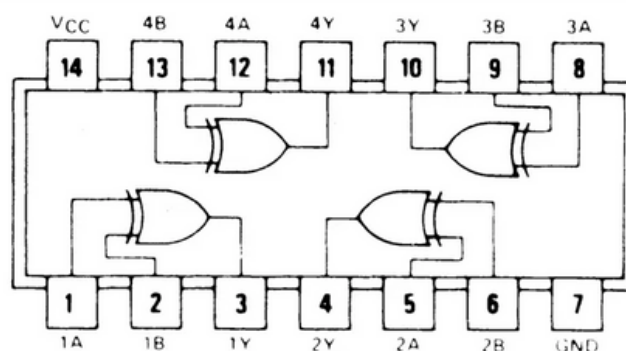


## Pinout XOR

### 7486 de 2 entradas



### 74386 de 2 entradas

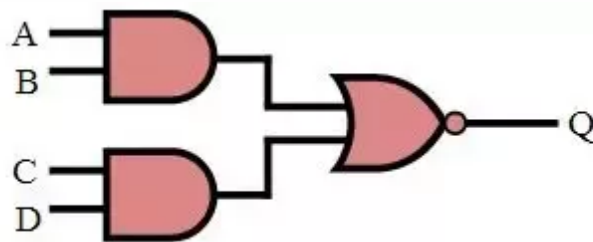


## Compuerta AOI

La lógica AND-OR-Invert (AOI) y las puertas AOI son funciones lógicas compuestas (o complejas) de dos niveles construidas a partir de la combinación de una o más puertas AND seguidas de una puerta NOR.

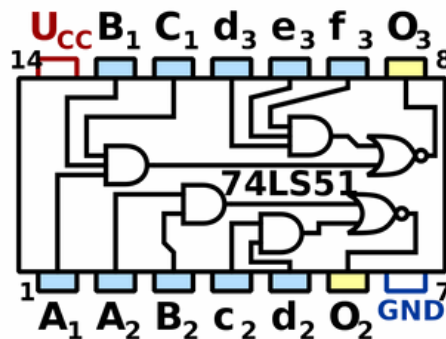
INPUT				OUTPUT
A	B	C	D	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

$$Q = \overline{(A \wedge B) \vee (C \wedge D)}$$

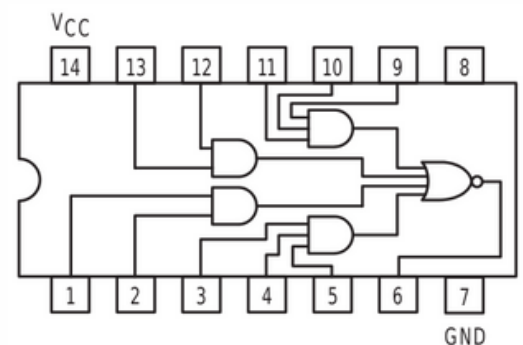


### Pinout AOI

#### 7451 de 2-3 entradas



#### 7453 de 3-2-2-3 entradas



#### 7455 2-ancho, 4-entradas

