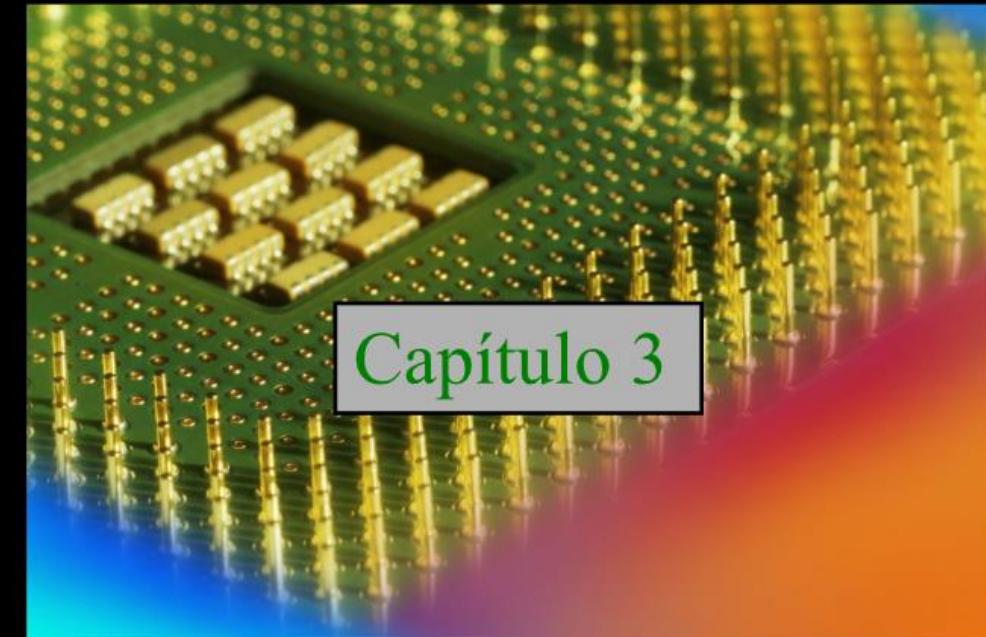


# Fundamentos de sistemas digitales

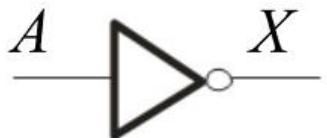
Décima Edición

Floyd



# Resumen

el inversor



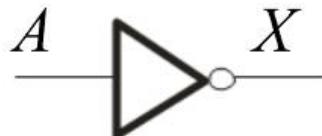
El inversor realiza la operación booleana **NOT**. Cuando la entrada es baja, la salida es alta; cuando la entrada es alta, la salida es baja.

Input	Output
<i>A</i>	<i>X</i>
BAJA (0)	ALTA (1)
ALTA (1)	BAJA (0)

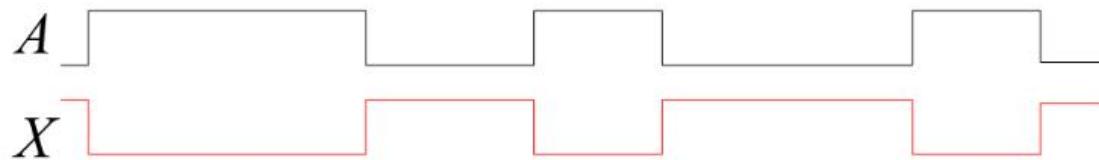
La operación **NOT** (complemento) se muestra con una barra superior. Por lo tanto, la expresión booleana para un inversor es  $X = \overline{A}$ .

# Resumen

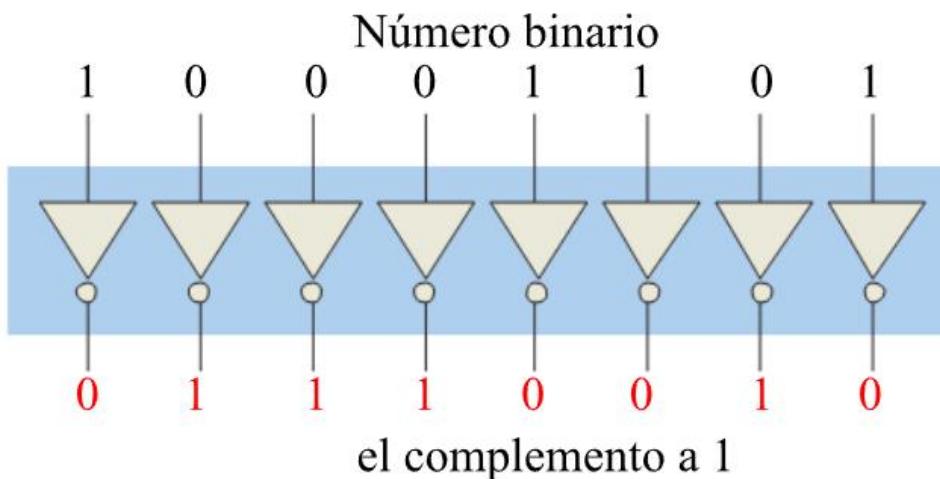
## El inversor



Ejemplo formas de onda:



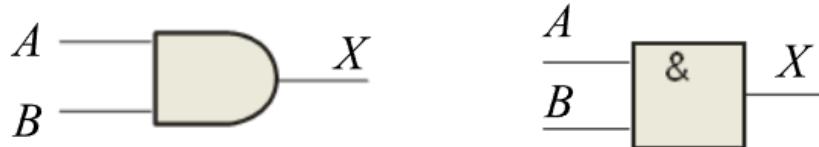
Un grupo de inversores se pueden utilizar para formar el complemento a 1 de un número binario:





# Resumen

## La puerta AND



**La puerta AND** produce una salida ALTA cuando todas las entradas son altas; de lo contrario, la salida es baja. Para una puerta de 2 entradas, la tabla de verdad es:

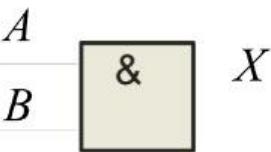
Inputs		Output
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

La operación **AND** generalmente se muestra con un punto entre las variables pero puede ser implícita (sin punto). Por lo tanto, la operación Y se escribe como  $X = A \cdot B$  o  $X = AB$ .

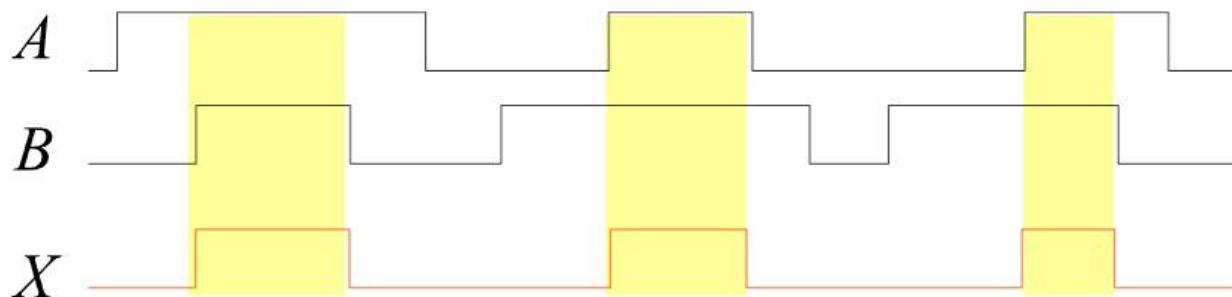


## Resumen

### La puerta AND



Ejemplo formas de onda:

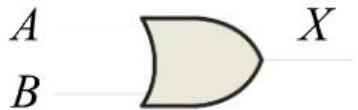


La operación AND se utiliza en la programación de ordenadores como una máscara selectiva. Si desea retener ciertos bits de un número binario pero restablecer los otros bits a 0, puede establecer una máscara con 1 en la posición de los bits retenidos.

**Ejemplo** Si el número binario 10100011 se opera AND con la máscara 00001111, ¿cuál es el resultado? **00000011**



## La puerta OR



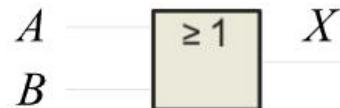
La **puerta OR** produce una salida ALTA si cualquier entrada es alta; si todas las entradas son bajas, la salida es baja. Para una puerta de 2 entradas, la tabla de verdad es

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

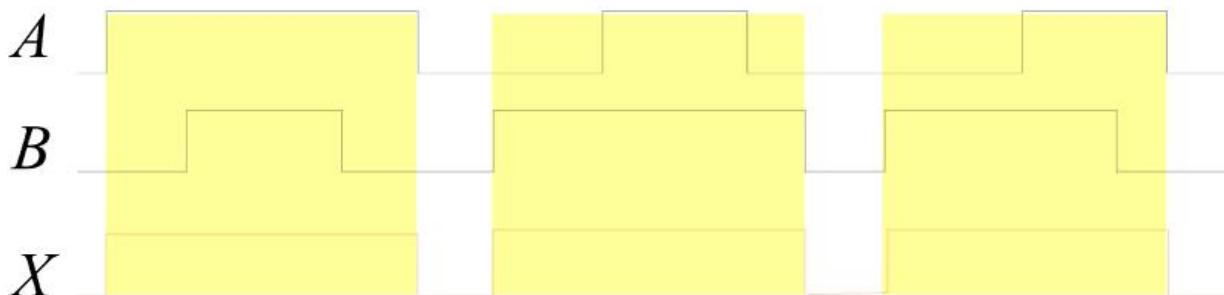
La operación **OR** se muestra con un signo más (+) entre las variables. Así, la operación OR se escribe como  $X = A + B$ .



## La puerta OR



Ejemplo formas de onda:



## Ejemplo

Las letras ASCII tienen un 1 en la posición de bit 5 para letras minúsculas y un 0 en esta posición para mayúsculas. (Las posiciones de los bits están numeradas de derecha a izquierda comenzando con 0.) ¿Cuál será el resultado si operas un OR de una letra ASCII con la máscara de 8 bits 00100000?

## Solución

La letra resultante será minúscula.

# Resumen

## La puerta NAND

A



X

A



X

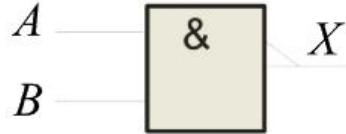
La **NAND** produce una salida baja cuando todas las entradas son altas; de lo contrario, la salida es ALTA. Para una puerta de 2 entradas, la tabla de verdad es

Inputs		Output
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

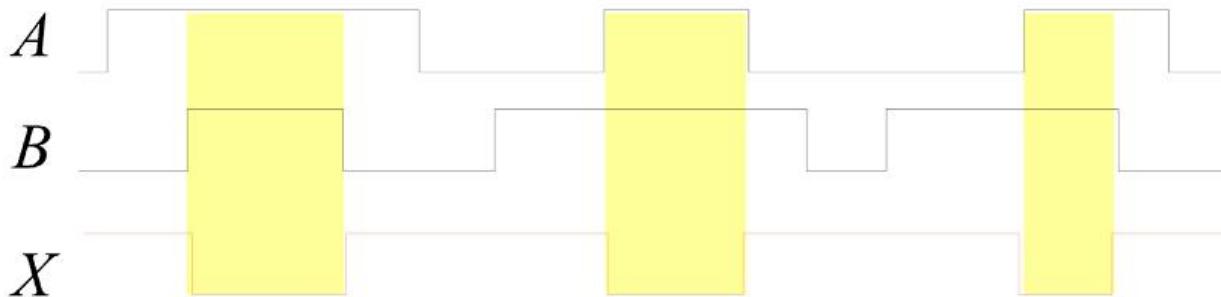
La operación **NAND** se muestra con un punto entre las variables y una barra superior encima. Por lo tanto, la operación NAND se escribe como  $X = \overline{A \cdot B}$  (Alternativamente,  $X = \overline{AB}$ .)

# Resumen

## La puerta NAND



Ejemplo formas de onda:



La puerta NAND es particularmente útil porque es una puerta “universal” - todas las demás puertas básicas se pueden construir de puertas NAND.

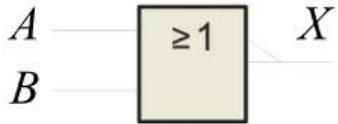
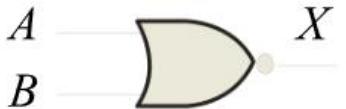
## Pregunta

¿Cómo conectar una puerta NAND de 2 entradas para formar un inversor básica?



# Resumen

## La puerta NOR



La **puerta NOR** produce una salida baja si cualquier entrada es alta; si todas las entradas son altas, la salida es baja. Para una puerta de 2 entradas, la tabla de verdad es

Inputs		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

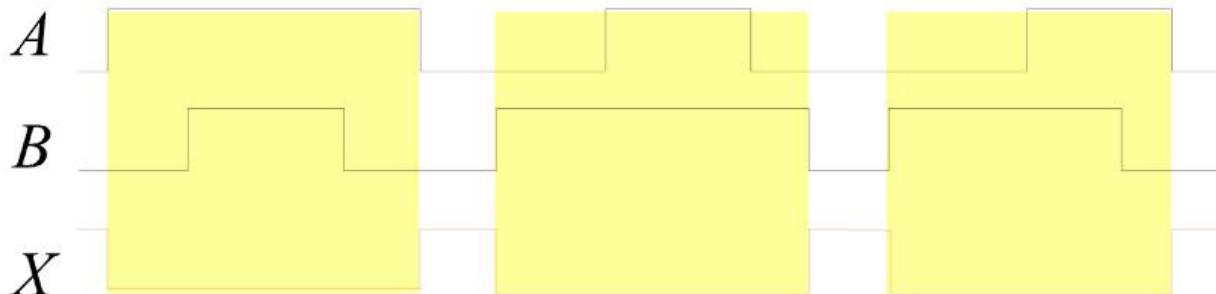
La operación **NOR** se muestra con un signo más (+) entre las variables y una barra superior encima. Por lo tanto, la operación NOR se escribe como  $X = \overline{A + B}$ .

# Resumen

## La puerta NOR



Ejemplo formas de onda:



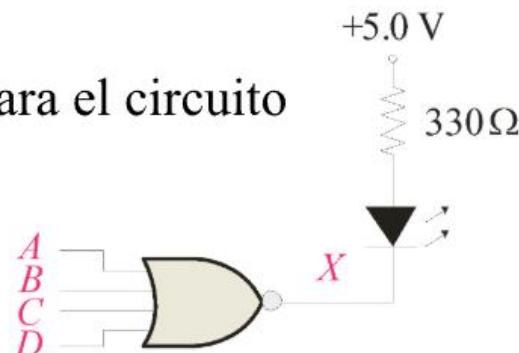
La operación NOR producirá BAJO si cualquier entrada es ALTA.

## Ejemplo

Cuando está el LED está encendido para el circuito mostrado?

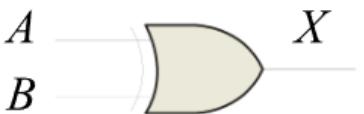
## Solución

El LED se enciende cuando cualquiera de las cuatro entradas son altas.





## La puerta XOR



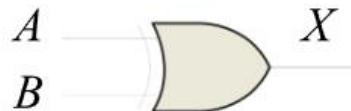
La **puerta XOR** produce una salida ALTA sólo cuando ambas entradas se encuentran en niveles lógicos opuestos. La tabla de verdad es

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

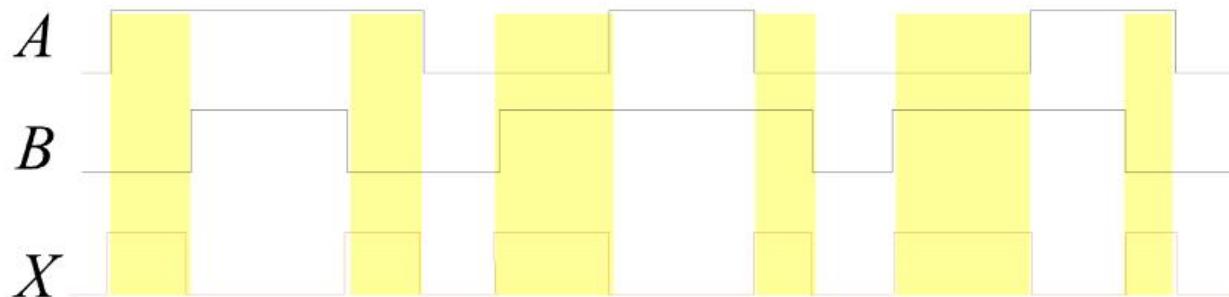
La operación **XOR** se escribe como  $X = \bar{A}B + A\bar{B}$ . Alternativamente, se puede escribir con un círculo el signo más entre las variables  $X = A \oplus B$ .

# Resumen

## La puerta XOR



Ejemplo formas de onda:



Observe que la puerta XOR se producen un alto sólo cuando exactamente una entrada es ALTA.

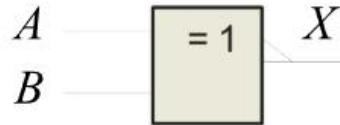
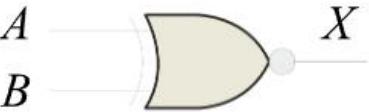
## Pregunta

Si las formas de onda A y B están invertidas para las formas de onda anteriores, ¿cómo se ve afectada la salida?

No habría ningún cambio en la salida.

# Resumen

## La puerta XNOR



La **puerta XNOR** produce una salida ALTA sólo cuando ambas entradas están en el mismo nivel lógico. La tabla de verdad es

Inputs		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

La operación **XNOR** mostrada como  $X = \overline{A}\overline{B} + AB$ . Alternativamente, la operación XNOR puede demostrarse con un punto de un círculo entre las variables. Por lo tanto, se puede demostrar como  $X = A \odot B$ .

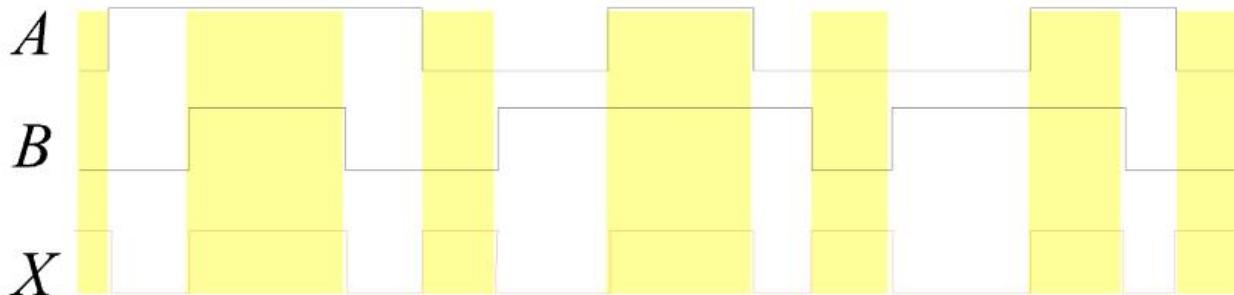


# Resumen

## La puerta XNOR



Ejemplo formas de onda:



Observe que la puerta XNOR se producen un alto cuando ambas entradas son iguales. Esto hace que sea útil para funciones de comparación.

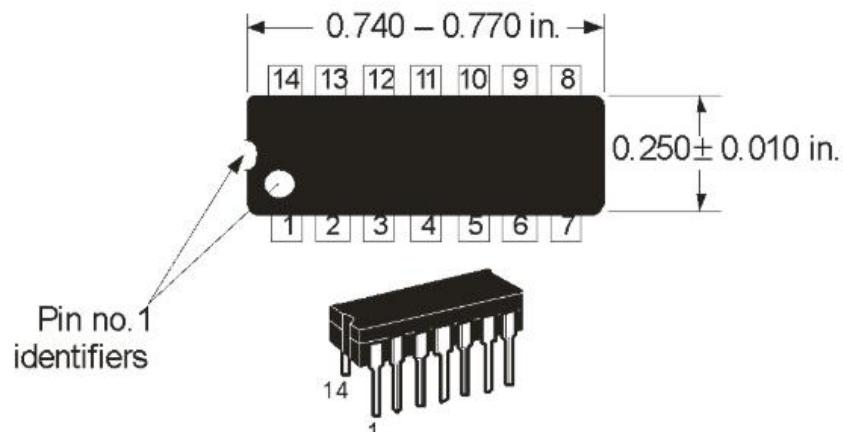
**Pregunta** Si la forma de onda A está invertida pero B sigue siendo la misma, ¿cómo se ve afectada la salida?

La salida se invierte.

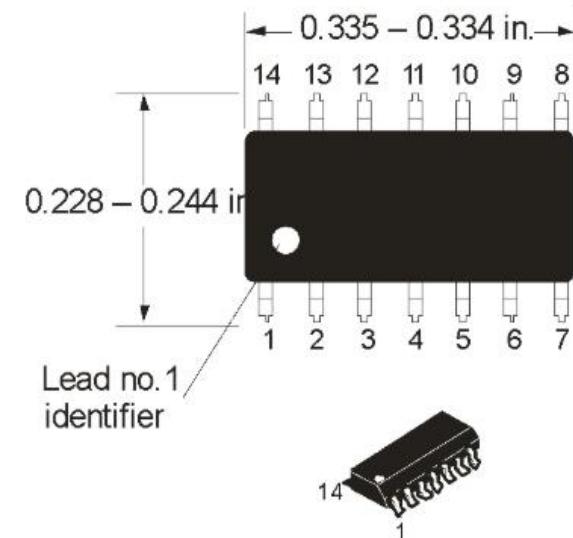
# Resumen

## Lógica de función fija

Dos de las principales familias de lógica de función fijas son TTL y CMOS. Una tercera es la tecnología BiCMOS, que combina los dos primeros. Se muestran los encapsulados para la lógica de función fija.



Encapsulado DIP



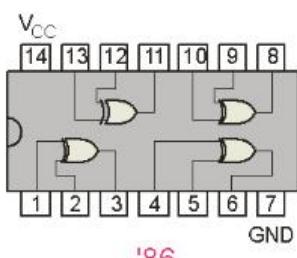
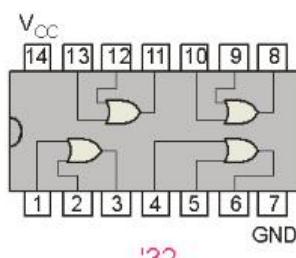
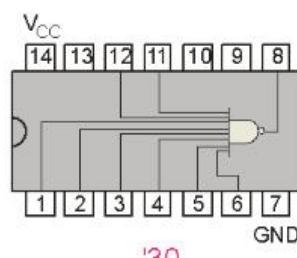
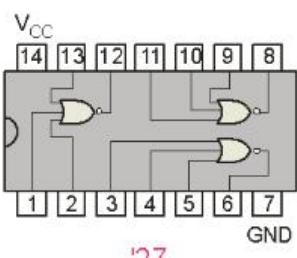
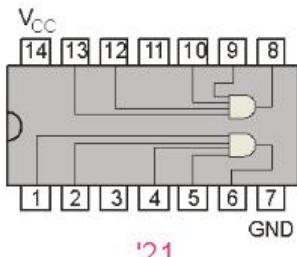
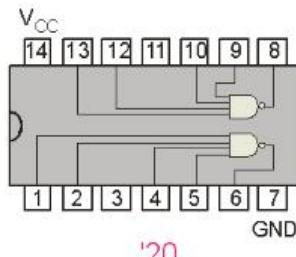
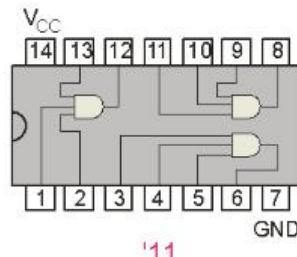
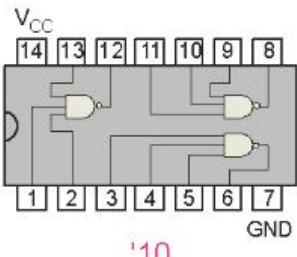
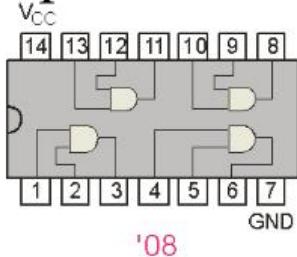
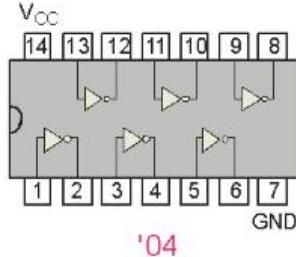
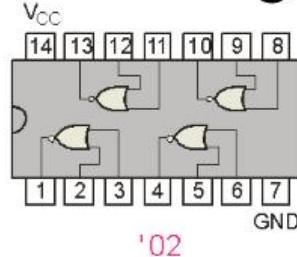
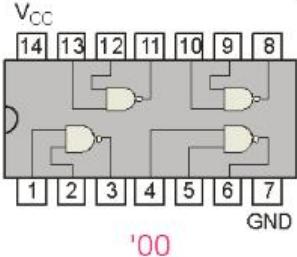
Encapsulado SOIC



# Resumen

## Lógica de función fija

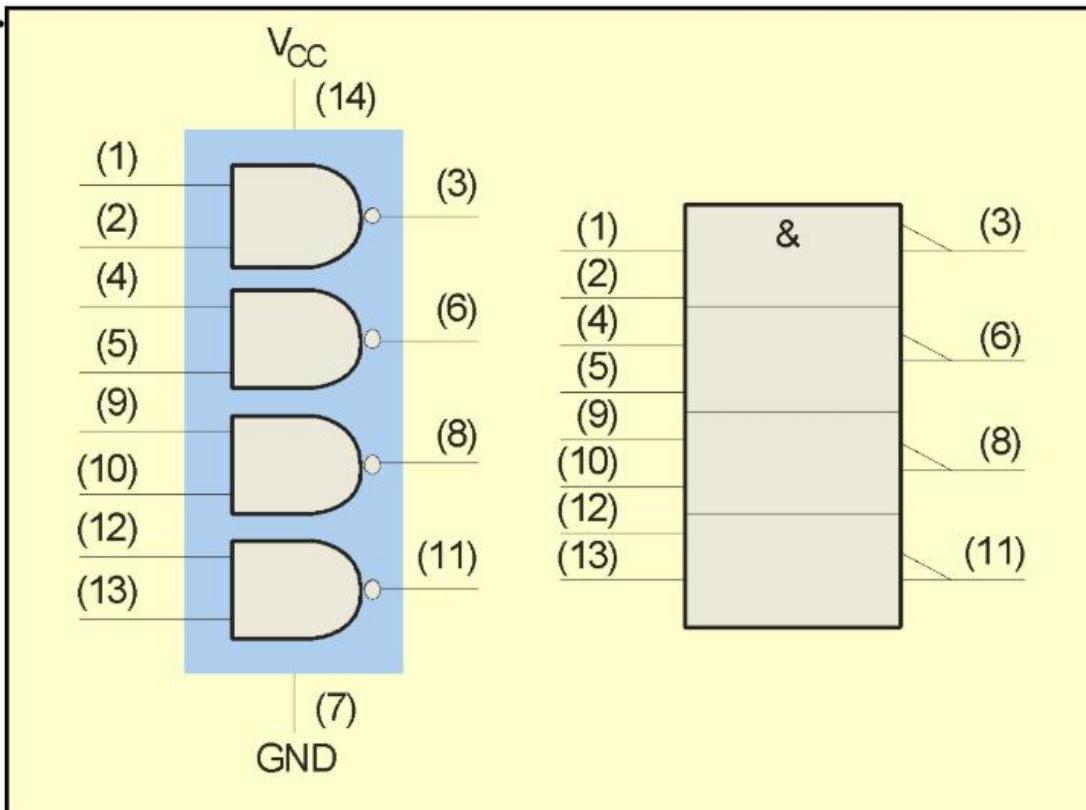
Se muestran algunas configuraciones de compuerta comunes.



# Resumen

## Lógica de función fija

Símbolos lógicos muestran las puertas y los números PIN asociado.



# Resumen

## Lógica de función fija

Las hojas de datos incluyen límites y condiciones establecidas por el fabricante, así como características de CC y CA. Por ejemplo, algunos valores máximos para un 74HC00A son:

MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
$V_{CC}$	DC Supply Voltage (Referenced to GND)	-0.5 to + 7.0 V	V
$V_{in}$	DC Input Voltage (Referenced to GND)	-0.5 to $V_{CC}$ +0.5 V	V
$V_{out}$	DC Output Voltage (Referenced to GND)	-0.5 to $V_{CC}$ +0.5 V	V
$I_{in}$	DC Input Current, per pin	$\pm 20$	mA
$I_{out}$	DC Output Current, per pin	$\pm 25$	mA
$I_{CC}$	DC Supply Current, $V_{CC}$ and GND pins	$\pm 50$	mA
$P_D$	Power Dissipation in Still Air, Plastic or Ceramic DIP SOIC Package TSSOP Package	750 500 450	mW
$T_{stg}$	Storage Temperature	-65 to + 150	°C
$T_L$	Lead Temperature, 1 mm from Case for 10 Seconds Plastic DIP, SOIC, or TSSOP Package Ceramic DIP	260 300	°C



# Términos clave

**Inversor** Un circuito lógico que invierte o complementa sus entradas.

**Tabla de verdad** Una tabla que muestra las entradas y la salida (s) de un circuito lógico correspondiente.

**Diagrama de tiempo** Un diagrama de formas de onda que muestran la relación de tiempo adecuada de todas las formas de onda.

**Álgebra de Boole** Las matemáticas de circuitos lógicos.

**Puerta AND** Una puerta lógica que produce una salida ALTA sólo cuando todas sus entradas son altas.



# Términos clave

**Puerta OR** Una puerta lógica que produce una salida ALTA cuando una o más entradas son altas.

**Puerta NAND** Una puerta lógica que produce una salida de baja sólo cuando todas sus entradas son altas.

**Puerta NOR** Una puerta lógica que produce una salida baja cuando una o más entradas son altas.

**La puerta OR exclusiva** Una puerta lógica que produce una salida ALTA sólo cuando sus dos entradas están en niveles opuestos.

**Puerta NOR exclusiva** Una puerta lógica que produce una salida de baja sólo cuando sus dos entradas están en niveles opuestos.

# Examen

1. La tabla de verdad para una puerta AND de 2 entradas es

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	0
0	1	1
1	0	1
1	1	0

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	0
0	1	0
1	0	0
1	1	1

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	1
0	1	0
1	0	0
1	1	0

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	0
0	1	1
1	0	1
1	1	1

a.

b.

c.

d.

# Examen

2. La tabla de verdad para una puerta NOR de 2 entradas es

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	0
0	1	1
1	0	1
1	1	0

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	0
0	1	0
1	0	0
1	1	1

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	1
0	1	0
1	0	0
1	1	0

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	0
0	1	1
1	0	1
1	1	1

# Examen

3. La tabla de verdad para una puerta XOR de 2-entradas es

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	0
0	1	1
1	0	1
1	1	0

a.

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	1
0	1	0
1	0	0
1	1	0

b.

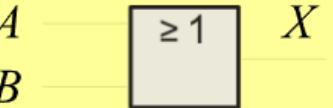
Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	0
0	1	0
1	0	0
1	1	1

c.

Inputs		Output
<i>A</i>	<i>B</i>	<i>X</i>
0	0	0
0	1	1
1	0	1
1	1	1

d.

# Examen

4. El símbolo  es para una
- a. OR gate
  - b. AND gate
  - c. NOR gate
  - d. XOR gate

# Examen

5. El símbolo  es para una
- a. OR gate
  - b. AND gate
  - c. NOR gate
  - d. XOR gate

# Examen

6. Una puerta lógica que produce una salida ALTA sólo cuando todas sus entradas son altas es una
- a. OR gate
  - b. AND gate
  - c. NOR gate
  - d. NAND gate

# Examen

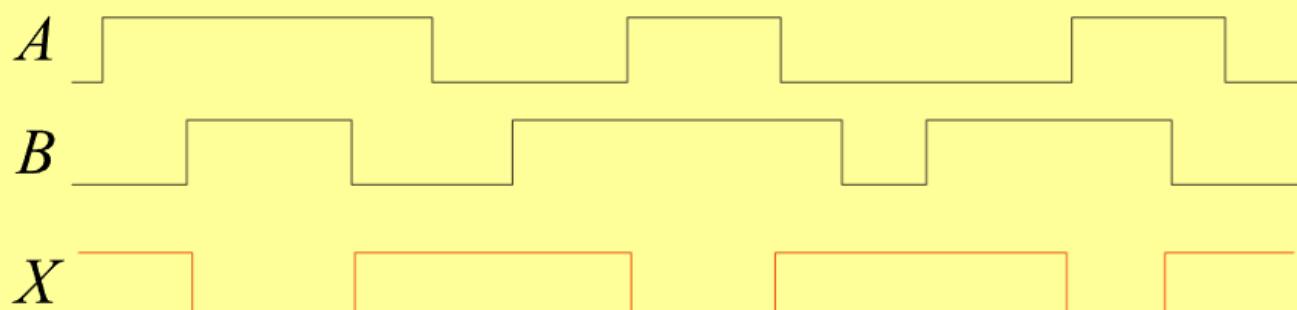
7. La expresión  $X = A \oplus B$  significa

- a.  $A$  OR  $B$
- b.  $A$  AND  $B$
- c.  $A$  XOR  $B$
- d.  $A$  XNOR  $B$

# Examen

8. Una puerta de 2-entrada produce el resultado que se muestra. ( $X$  representa la salida.) Este es un (n)

- a. OR gate
- b. AND gate
- c. NOR gate
- d. NAND gate



# Examen

9. Una puerta de 2-entrada produce una salida ALTA sólo cuando las entradas son iguales. Este tipo de puerta es un
- a. OR gate
  - b. AND gate
  - c. NOR gate
  - d. XNOR gate

# Examen

10. La lógica necesaria para un PLD se puede especificar en un lenguaje de descripción de hardware por
- a. la introducción de texto
  - b. la introducción del esquema
  - c. diagramas de estado
  - d. Todas las anteriores

# Examen

## Answers:

- |      |       |
|------|-------|
| 1. c | 6. b  |
| 2. b | 7. c  |
| 3. a | 8. d  |
| 4. a | 9. d  |
| 5. d | 10. d |