

¿Que son las compuertas lógicas?

Una compuerta lógica es un dispositivo digital que realiza operaciones binarias con uno o dos estados lógicos (0,1) en la entrada y obteniendo resultados booleanos (0,1) en la salida del sistema. Se pueden combinar entre si para obtener nuevas funciones.

Las compuertas básicas son tres AND, OR y NOT, y sus compuertas complementarias son NAND, NOR. También existe una compuerta suplementaria XOR y su compuerta complementaria XNOR.

La familia **TTL (Lógica Transistor-Transistor)** mas utilizada para estas compuertas son la serie 74LSXX (LS:Low-powe-Schottky) algunas de sus características son:

- **Alimentación:** $5V \pm 0.25$, es conveniente usar una fuente regulada y condensador cerámico de $0.1 \mu F$ conectado a través del suministro cerca del CI, para eliminar picos generados por los interruptores.
- **Entradas:** Cualquier entrada que no se utilice es recomendable conectarla a $+V_{cc}$, para eliminar el ruido, ya que estos CI tienen entradas flotantes.
- **Salidas:** Pueden suministrar 16mA, pero pueden alimentar sobre los 2mA.
- **Tiempo de propagación** de la puerta: cerca de 10ns para que una señal pase a través de una puerta.
- **Frecuencia:** En condiciones ideales es de 35MHz.
- **Consumo de Potencia:** pocos mW.

Cada compuerta está asociada a un símbolo, una tabla de verdad y una operación booleana; que expresa el estado de su salida para cada combinación posible de las entradas.

Compuertas:

AND

AND

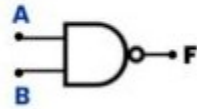


A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

- CHIP 74LS08
- Operación $F = A \cdot B$

NAND

NAND



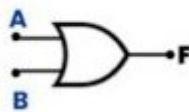
$$F = \overline{AB}$$

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

- CHIP 74LS00

OR

OR

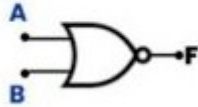


A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

- CHIP 74LS32
- $F = A + B$

NOR

NOR



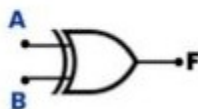
$$F = \overline{A + B}$$

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

- CHIP 74LS02

XOR

XOR



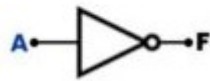
$$F = A \oplus B$$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

- CHIP 74LS86

NOT

NOT

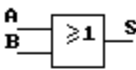
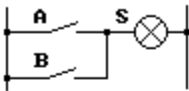
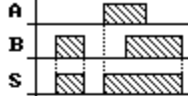
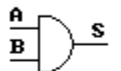
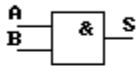
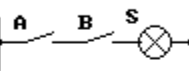
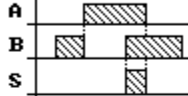
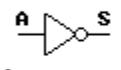
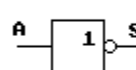
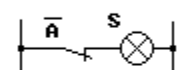
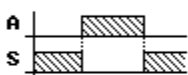
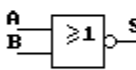
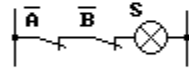
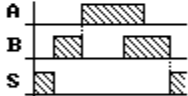
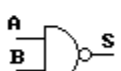
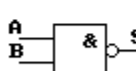
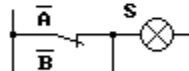
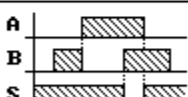
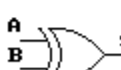
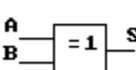
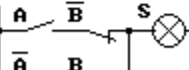
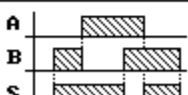
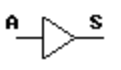
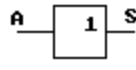
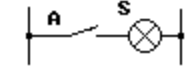
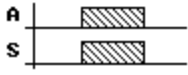


$$F = \overline{A}$$

A	F
1	0
0	1

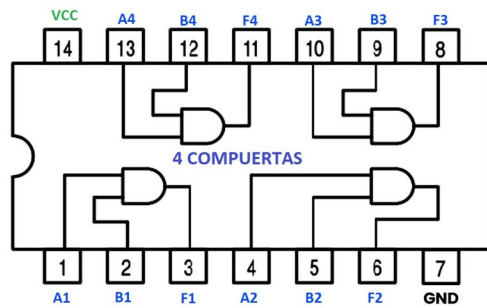
- CHIP 74LS86

Resumen

Función	Ecuación lógica	Símbolos			Tabla de verdad	Cronograma															
		Norma MIL	Norma IEC	Circuito físico con contactos																	
OR	$S = A + B$				<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1	
A	B	S																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	1																			
AND	$S = A \cdot B$				<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1	
A	B	S																			
0	0	0																			
0	1	0																			
1	0	0																			
1	1	1																			
NOT	$S = \bar{A}$	 inversor			<table><tr><th>A</th><th>S</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	S	0	1	1	0										
A	S																				
0	1																				
1	0																				
NOR (OR+NOT)	$S = \overline{A + B}$ $S = \bar{A} \cdot \bar{B}$				<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	0	
A	B	S																			
0	0	1																			
0	1	0																			
1	0	0																			
1	1	0																			
NAND (AND+NOT)	$S = \overline{A \cdot B}$ $S = \bar{A} + \bar{B}$				<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	1	0	1	1	1	0	1	1	1	0	
A	B	S																			
0	0	1																			
0	1	1																			
1	0	1																			
1	1	0																			
EXOR	$S = A \oplus B = A\bar{B} + \bar{A}B$				<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	0	
A	B	S																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	0																			
EXCI TADOR	$S = A$				<table><tr><th>A</th><th>S</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	S	0	0	1	1										
A	S																				
0	0																				
1	1																				

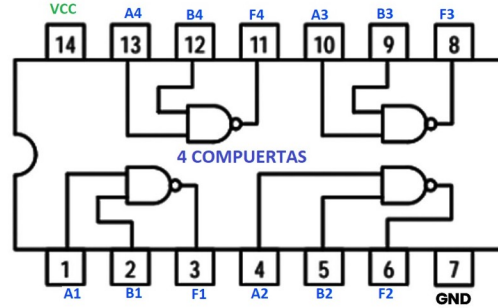
74LS08

AND



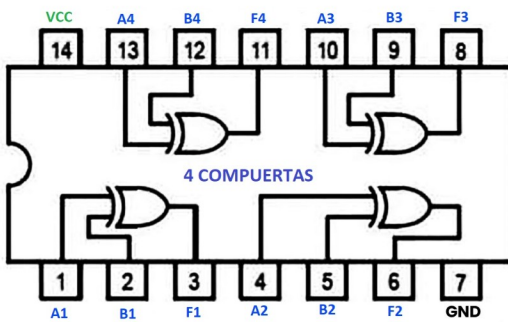
74LS00

NAND



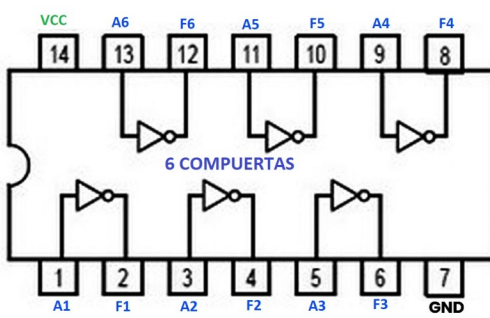
74LS86

XOR



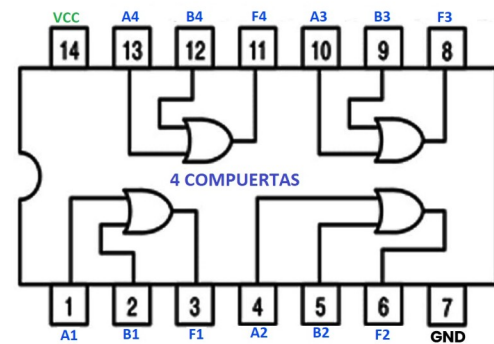
74LS04

NOT



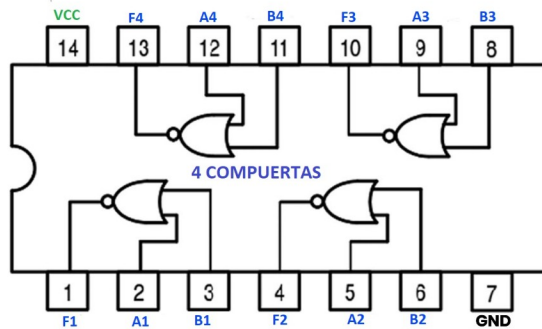
74LS32

OR



74LS02

NOR



<https://blog.uelectronics.com/electronica/circuitos-integrados-compuertas-logicas-and-or-nand-xor-y-not/>

Obtener la tabla de verdad que corresponde a la siguiente función lógica:

$$F = (A + B) * C$$

Orden de operaciones:

1. Negación individual
2. Paréntesis
3. Multiplicación
4. Suma

ENTRADAS			Cálculos	
A	B	C	(A + B)	(A + B) * C
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

Arduino

Probando el led inteno

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(13,OUTPUT);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite(13,LOW);  
}
```

Practica Arduino:

Simularlo en <https://www.tinkercad.com> y realizar el laboratorio.

Compuerta AND:

Elementos :

- 3 Resistencias 220 Ω
- 2 Interruptores o Pulsadores
- 1 Led
- 1 Arduino UNO

Código Arduino

// Declarar los pines

const int boton1 = 2; // Pin para el primer botón

const int boton2 = 3; // Pin para el segundo botón

const int led = 7; // Pin para el LED de salida

void setup() {

// Configurar los pines

pinMode(boton1, INPUT); // El pin del botón 1 será una entrada

pinMode(boton2, INPUT); // El pin del botón 2 será una entrada

pinMode(led, OUTPUT); // El pin del LED será una salida

}

void loop() {

// Leer el estado de los botones

int estadoBoton1 = digitalRead(boton1);

int estadoBoton2 = digitalRead(boton2);

// Comprobar la lógica AND: si ambos botones están presionados (HIGH)

if (estadoBoton1 == HIGH && estadoBoton2 == HIGH) {

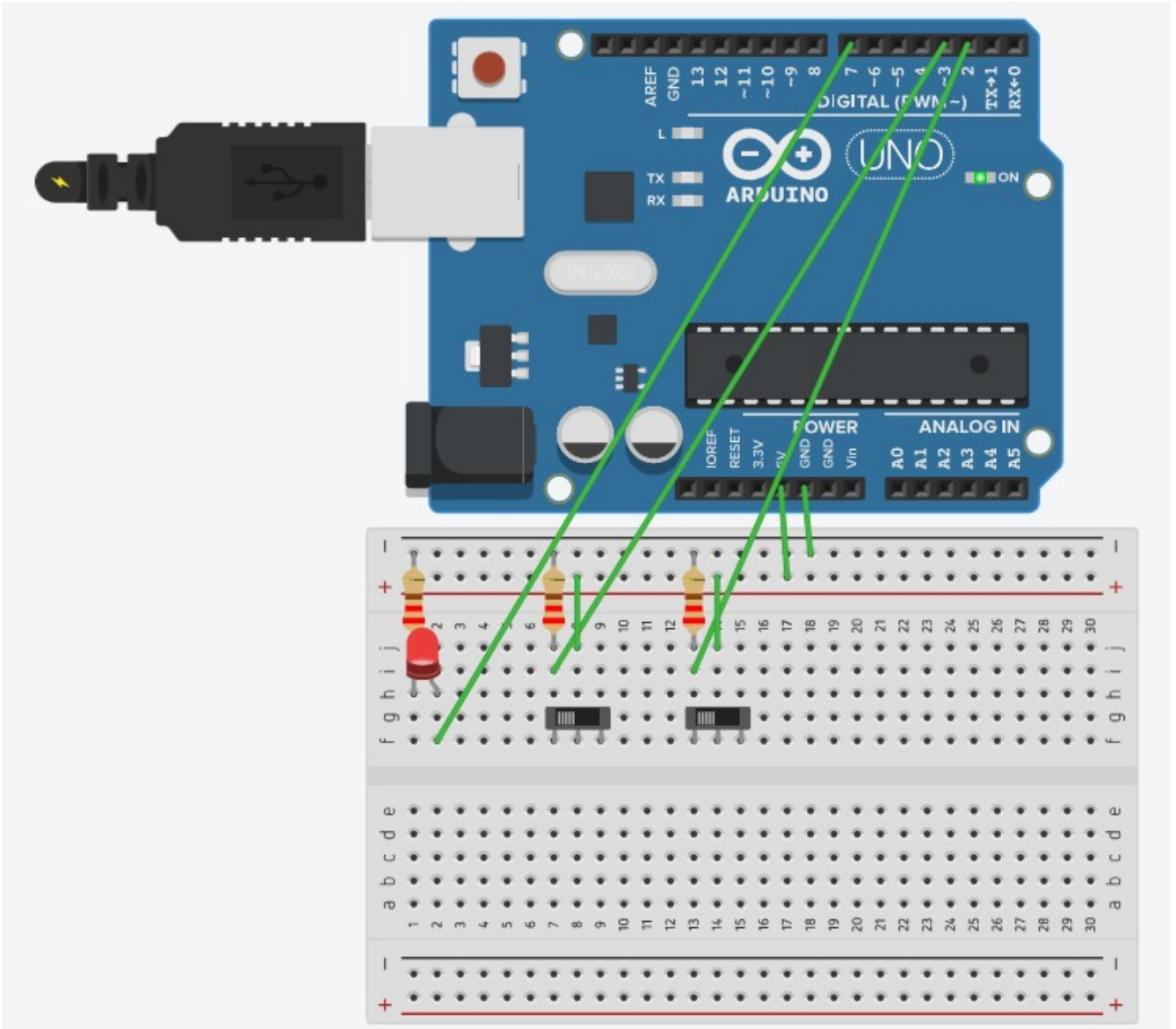
digitalWrite(led, HIGH); // Enciende el LED

} else {

digitalWrite(led, LOW); // Apaga el LED

}

}



Compuerta NAND

Elementos :

- 3 Resistencias 220 Ω
- 2 Interruptores o Pulsadores
- 1 Led
- 1 Arduino UNO

Código Arduino

```
// Declarar los pines
const int boton1 = 2; // Pin para el primer botón
const int boton2 = 3; // Pin para el segundo botón
const int led = 7; // Pin para el LED de salida

void setup() {
  // Configurar los pines
  pinMode(boton1, INPUT); // El pin del botón 1 será una entrada
  pinMode(boton2, INPUT); // El pin del botón 2 será una entrada
  pinMode(led, OUTPUT); // El pin del LED será una salida
}

void loop() {
  // Leer el estado de los botones
  int estadoBoton1 = digitalRead(boton1);
  int estadoBoton2 = digitalRead(boton2);

  // Lógica NAND: el LED se apaga SOLO si ambos botones están presionados (HIGH)
  if (estadoBoton1 == HIGH && estadoBoton2 == HIGH) {
    digitalWrite(led, LOW); // Apaga el LED
  } else {
    digitalWrite(led, HIGH); // Enciende el LED en cualquier otro caso
  }
}
```

Compuerta OR

Elementos :

- 3 Resistencias 220 Ω
- 2 Interruptores o Pulsadores
- 1 Led
- 1 Arduino UNO

Código Arduino

```
// Declarar los pines
const int boton1 = 2; // Pin para el primer botón
const int boton2 = 3; // Pin para el segundo botón
const int led = 7; // Pin para el LED de salida

void setup() {
  // Configurar los pines
  pinMode(boton1, INPUT); // El pin del botón 1 será una entrada
  pinMode(boton2, INPUT); // El pin del botón 2 será una entrada
  pinMode(led, OUTPUT); // El pin del LED será una salida
}

void loop() {
  // Leer el estado de los botones
  int estadoBoton1 = digitalRead(boton1);
  int estadoBoton2 = digitalRead(boton2);

  // Lógica OR: el LED se enciende si el botón 1 O el botón 2 están presionados
  if (estadoBoton1 == HIGH || estadoBoton2 == HIGH) {
    digitalWrite(led, HIGH); // Enciende el LED
  } else {
    digitalWrite(led, LOW); // Apaga el LED si ambos botones están sin presionar
  }
}
```

Compuerta NOR

Elementos :

- 3 Resistencias 220 Ω
- 2 Interruptores o Pulsadores
- 1 Led
- 1 Arduino UNO

Código Arduino

```
// Declarar los pines
const int boton1 = 2; // Pin para el primer botón
const int boton2 = 3; // Pin para el segundo botón
const int led = 7;  // Pin para el LED de salida

void setup() {
  // Configurar los pines
  pinMode(boton1, INPUT); // El pin del botón 1 será una entrada
  pinMode(boton2, INPUT); // El pin del botón 2 será una entrada
  pinMode(led, OUTPUT);  // El pin del LED será una salida
}

void loop() {
  // Leer el estado de los botones
  int estadoBoton1 = digitalRead(boton1);
  int estadoBoton2 = digitalRead(boton2);

  // Lógica NOR: el LED se enciende SOLO si ambos botones NO están presionados
  if (estadoBoton1 == LOW && estadoBoton2 == LOW) {
    digitalWrite(led, HIGH); // Enciende el LED
  } else {
    digitalWrite(led, LOW);  // Apaga el LED en cualquier otro caso
  }
}
```

Compuerta XOR

Elementos :

- 3 Resistencias 220 Ω
- 2 Interruptores o Pulsadores
- 1 Led
- 1 Arduino UNO

Código Arduino

```
// Declarar los pines
const int boton1 = 2; // Pin para el primer botón
const int boton2 = 3; // Pin para el segundo botón
const int led = 7; // Pin para el LED de salida

void setup() {
  // Configurar los pines
  pinMode(boton1, INPUT); // El pin del botón 1 será una entrada
  pinMode(boton2, INPUT); // El pin del botón 2 será una entrada
  pinMode(led, OUTPUT); // El pin del LED será una salida
}

void loop() {
  // Leer el estado de los botones
  int estadoBoton1 = digitalRead(boton1);
  int estadoBoton2 = digitalRead(boton2);

  // Lógica XOR: el LED se enciende si solo UNO de los botones está presionado
  if ((estadoBoton1 == HIGH && estadoBoton2 == LOW) || (estadoBoton1 == LOW &&
estadoBoton2 == HIGH)) {
    digitalWrite(led, HIGH); // Enciende el LED
  } else {
    digitalWrite(led, LOW); // Apaga el LED en cualquier otro caso
  }
}
```

Compuerta NOT

Elementos :

- 2 Resistencias 220 Ω
- 1 Interruptores o Pulsadores
- 1 Led
- 1 Arduino UNO

Código Arduino

```
// Declarar los pines
const int boton1 = 2; // Pin para el único botón de entrada
const int led = 7; // Pin para el LED de salida

void setup() {
  // Configurar los pines
  pinMode(boton1, INPUT); // El pin del botón será una entrada
  pinMode(led, OUTPUT); // El pin del LED será una salida
}

void loop() {
  // Leer el estado del botón
  int estadoBoton1 = digitalRead(boton1);

  // Lógica NOT: el LED se enciende si el botón NO está presionado
  if (estadoBoton1 == LOW) { // Se usa LOW porque el botón no está presionado
    digitalWrite(led, HIGH); // Enciende el LED
  } else {
    digitalWrite(led, LOW); // Apaga el LED si el botón está presionado
  }
}
```

Otra opción:

```
// Declarar los pines
const int boton1 = 2; // Pin para el único botón de entrada
const int led = 7; // Pin para el LED de salida

void setup() {
  // Configurar los pines
  pinMode(boton1, INPUT); // El pin del botón será una entrada
  pinMode(led, OUTPUT); // El pin del LED será una salida
}

void loop() {
  // Leer el estado del botón
  int estadoBoton1 = digitalRead(boton1);

  // Lógica NOT: el LED se enciende si el estado del botón NO es HIGH
  if (!digitalRead(boton1)) { // Si la lectura del pin es LOW
```

```
    digitalWrite(led, HIGH); // Enciende el LED
} else {
    digitalWrite(led, LOW); // Apaga el LED
}
}
```