# Tema 6. Electrónica Digital

- Fundamentos
  - Introducción a los Sistemas Digitales: Origen, símbolos, niveles, realización, tecnologías
  - Etapas de desarrollo de los circuitos integrados digitales
  - Analógico - Digital. ¿Cuál es la diferencia?
- Sistemas de numeración:
  - Sistema Decimal
  - Sistema Binario
  - Sistema Octal
  - Sistema Hexadecimal
- - Circuitos Lógicos
  - Tabla de verdad
  - Niveles Lógicos
  - Circuitos combinacionales
  - Circuitos secuenciales
  - Algebra Booleana
  - Mapas de Karnaugh

Docente: Francisco Javier Llopis Cánovas
Docente: Beatriz Rodríguez Mendoza
Docente: Silvestre Rodríguez Pérez
Docente: Julio Francisco Rufo Torres

- Códigos binarios
  - Código BCD
- Componentes básicos
  - Compuerta AND o "Y"
  - Compuerta NAND o "No Y"
  - Compuerta OR o "O"
  - Compuerta NOR o "No O‟
  - Compuerta NOT o "No" o inversora
  - Compuerta XOR o compuerta OR exclusiva
  - Familia de circuitos TTL
- Tecnología TTL
  - Compuertas lógicas
  - Niveles lógicos en compuertas TTL
- Circuitos secuenciales
  - Biestable RS
  - Biestable tipo D
  - Biestable JK
- Componentes avanzados
  - Convertidor Analógico - Digital (ADC)
  - Convertidor Digital - Analógico (DAC)
  - Decodificadores
  - Multiplexadores - Multiplexores (MUX) introducción

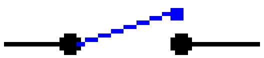
# Origen, símbolos, niveles, realización, tecnologías

- La palabra "digital" tiene origen latino:

  digitus = dedos

  (contar con los dedos)
- En la técnica digital solamente existen dos posibles valores de la señal y si bien son solo dos, hay varias maneras de representarlos.
- En la siguiente tabla se muestran los diferentes tipos de interpretaciones.

# Realización

| Valor lógico | Si / "1" | No / "0" |
|---|---|---|
| Símbolo | 1 | 0 |
| Realización |  |  |
| | Hay corriente | No hay corriente |
| | Nivel de tensión alta (High) | Nivel de tensión baja (Low) |

# Características

| Técnica digital | Técnica Analógica |
| --- | --- |
| - Sólo tensión "High" y "Low" son posibles<br>- Gran escala de integración<br>- Alta seguridad<br>- Ausencia de interferencias | - Cualquier valor de tensión es posible<br>- Problemas de ajuste y distorsión<br>- Influencia de señales por interferencia |

# Tecnologías fundamentales

- Los circuitos digitales son implementados por 3 tipos fundamentales de circuitos lógicos: AND, OR y NOT y las tecnologías utilizadas son:
  - - TTL: Lógica - transistor – transistor
  - - CMOS: Metal Oxido Semiconductor
  - - ECL: Lógica Emisores acoplados

# Etapas de desarrollo de los circuitos integrados digitales

| Nivel de integración | Número de funciones por chip | Ejemplos de aplicaciones |
|---|---|---|
| **1965: SSI (Small Scale Integration)** | **> 100** | **Circuitos básicos. Compuertas AND, compuerta OR, compuerta NAND, compuerta NOT, Compuerta NOR, etc.** |
| **1968: MSI (Medium Scale Integration)** | **de 100 a 1000** | **Registros, contadores** |
| **1972: LSI (Large Scale Integration)** | **de 1000 a 10000** | **Microprocesadores, memorias** |
| **1976: VLSI (Very Large Scale Integration)** | **de 10000 a 100000** | **Microprocesadores completos** |
| **1980 VVLSI (Very Very Large Scale Integration)** | **> 100000** | **Microprocesadores múltiples, incluyendo memoria, puertos de entrada y salida** |

# Aplicaciones

- La tecnología digital se puede manifestar en los siguientes campos
  - \- Mecánico
  - \- Electromecánico
  - \- Neumático
  - \- Hidráulico
  - \- Electrónico
- Los circuitos digitales representan el "hardware" de las computadoras, pero las funciones lógicas también son posibles de realizar por la programación de las computadoras mediante el "Software"

## Analógico y Digital


- Algunas mediciones pueden representarse en forma "analógica" otras en forma "digital".
- El término: Digital, Se refiere a cantidades discretas como la cantidad de personas en un una sala, cantidad de libros en una biblioteca, cantidad de autos en una zona de estacionamiento, cantidad de productos en un supermercado, etc..

# Analógico


www.unicrom.com

- El término: Analógico, Se refiere a las magnitudes o valores que varían con el tiempo en forma continua como la distancia y la temperatura, la velocidad, que podrían variar muy lento o muy rápido.
- En la vida cotidiana el tiempo se representa en forma analógica por relojes (de agujas), y en forma discreta (digital) por displays digitales.

## tecnología analógica

- En la tecnología analógica es muy difícil almacenar, manipular, comparar, calcular y recuperar información con exactitud cuando esta ha sido guardada.
- En cambio en la tecnología digital (computadoras, por ejemplo), se pueden hacer tareas muy rápidamente, muy exactas, muy precisas y sin detenerse.

# Analógico o digital

- La electrónica moderna usa electrónica digital para realizar muchas funciones que antes desempeñaba la electrónica analógica.
- Un ejemplo muy evidente es el hecho de que la música actualmente se graba en discos compactos (CD's), que previamente ha sido convertida a formato digital del original que es el formato analógico.

## Analógico o digital

- El equipo creado está lleno de circuitos lógicos digitales.
- A diferencia, los discos de acetato (los discos de 45 r.p.m. y L.P. de color negro) utilizaban una aguja que recorría los surcos en el disco para poder reproducir la música grabada.
- Nadie duda de la calidad de los discos compactos de hoy, pues tienen un sonido excelente.

## Sistemas de Numeración

- Los Sistemas de numeración son aquellos que permiten representar una cantidad de unidades de cualquier tipo.
- Un sistema muy interesante y que todavía se utiliza es el Sistema de los números romanos.

# Sistema de Numeración Decimal

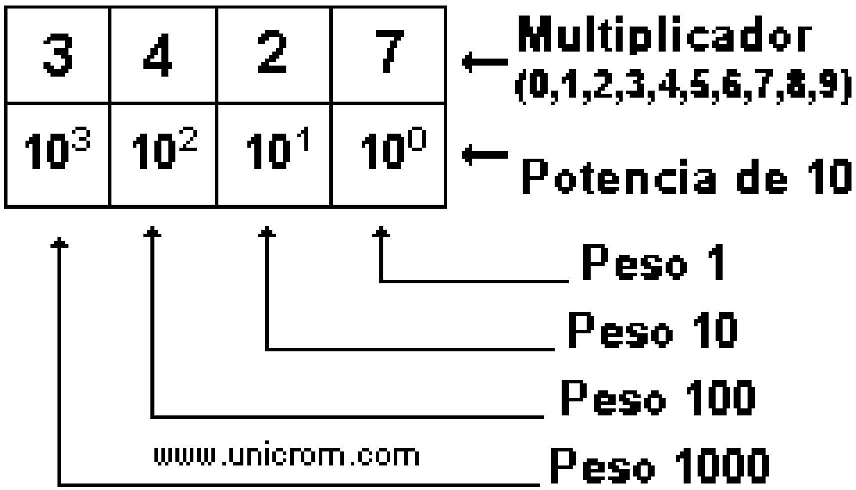
- El **Sistema Decimal** es el sistema es que todos utilizamos.
- El **Sistema Decimal** utiliza 10 cifras (del 0 al 9). Al combinar estas cifras se consigue expresar número más grandes.
- Ejemplo: 2005 o 235689, etc.
- La razón de utilizar el **Sistema Decimal** es que los seres humanos tenemos en las manos diez (10) dedos.

# Numeración Decimal

- Tal vez si tuviésemos una cantidad diferente de dedos hubiésemos utilizado un sistemas diferente.
- Esto podría ser cierto o no.

# ¿Cómo trabaja o funciona el sistema decimal?

- Un número en el **Sistema Decimal** se divide en cifras con diferente peso.
- Las unidades tienen peso 1, las decenas peso 10, las centenas peso 100, los miles peso 1000, etc.



# sistema decimal

- Cada peso tiene asociado una potencia de 10. En el caso de las unidades la potencia de diez es $10^0$, en el caso de los miles o millares la potencia de diez es $10^3$.
- Entonces para formar el número 3427:

| | | | |
|---|---|---|---|
| $3 \times 10^3$ | $= 3 \times 1000 = 3000$ | | 3000 |
| $4 \times 10^2$ | $= 4 \times 100 = 400$ | + | 400 |
| $2 \times 10^1$ | $= 2 \times 10 = 20$ | + | 20 |
| $7 \times 10^0$ | $= 7 \times 1 = 7$ | + | 7 |
| | | = | 3427 |

www.uniorom.com

# Sistema de Numeración Binario

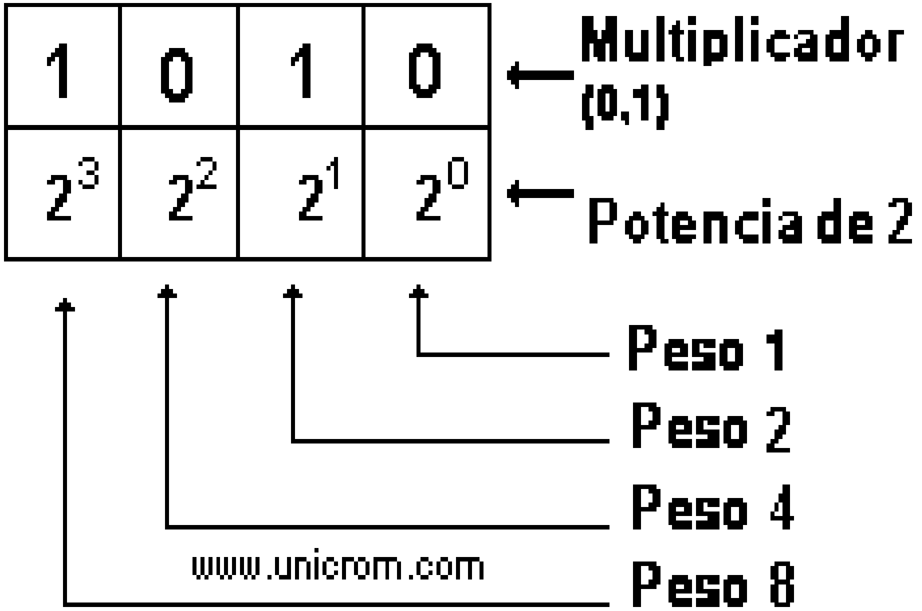
- El Sistema Binario, sólo necesita dos (2) cifras: el "0" y el "1".
- El Sistema Binario es de especial importancia en la electrónica digital, donde sólo son posibles dos valores: el "1" o valor de voltaje "alto" y el "0" o nivel de voltaje "bajo".

# Sistema Binario

- Los valores de "1" y "0" se asocian con:
- - "nivel alto" y "nivel bajo",
  - "cerrado" y "abierto",
  - "encendido" y "apagado",
  - "conectado" y "desconectado",
  - "high" y "low",
  - "on" y "off",
  etc..

# Sistema Binario

- Un número en el Sistema de Numeración Binario se divide en cifras con diferente peso: 1, 2, 4, 8, 16, 32, 64, 128,.... etc.



# Sistema Binario

- Cada peso tiene asociado una potencia de 2. En el primer número (de derecha a izquierda) la potencia de dos es $2^0$, en el segundo número la potencia de dos es $2^1$ y así hasta el último número del lado izquierdo.
- Entonces para formar el número $1010_2$: (el número 10 en binario)

| | | | |
|---|---|---|---|
| $1 \times 2^3$ | $= 1 \times 8 = 8$ | | 8 |
| $0 \times 2^2$ | $= 0 \times 4 = 0$ | + | 0 |
| $1 \times 2^1$ | $= 1 \times 2 = 2$ | + | 2 |
| $0 \times 2^0$ | $= 0 \times 1 = 0$ | + | 0 |
| equivalente decimal ---> | | = | 10 |

# Sistema de Numeración Octal

- En el Sistema de Numeración Octal (base 8), sólo se utilizan 8 cifras (0, 1, 2, 3, 4, 5, 6, 7)
- Este Sistema de numeración una vez que se llega a la cuenta 7 se pasa a 10, etc.
- La cuienta hecha en octal: 0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, .....
- Se puede observar que en este sistema numérico no existen los números: 8 y 9

# Sistema de Numeración Hexadecimal

- El sistema hexadecimal, necesita 16 cifras y/o letras para poder expresar una cantidad.
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Si se cuentan las letras y números anteriores se tienen 16.
- En la siguiente tabla se ve una comparación de los números superiores a 9 en el Sistema de Numeración Hexadecimal y el Sistema de Numeracion Decimal.

# Sistema Hexadecimal

| | |
|---|---|
| $A_{16} = 10_{10}$ | $D_{16} = 13_{10}$ |
| $B_{16} = 11_{10}$ | $E_{16} = 14_{10}$ |
| $C_{16} = 12_{10}$ | $F_{16} = 15_{10}$ |

## Suma o adición binaria

- Para que las computadoras funcionen adecuadamente deben ser capaces de realizar operaciones aritméticas.
- Una de ellas es la suma o adición binaria.
- Una vez que esta operación se entienda, será fácil entender también la resta, multiplicación y división con números binarios.
- En la tabla mas abajo se pueden ver las sumas más sencillas que se pueden hacer con dos número binarios de una cifra

# Suma binaria

- Los tres primeros renglones de la tabla muestran una suma fácilmente entendible, pero cuando se hace la última suma, se ve que el resultado tiene dos cifras.
- 0 + 0 = 0
  0 + 1 = 1
  1 + 0 = 1
  1 + 1 = 10

## Suma binaria

- Esto es así debido a que utilizamos el sistema de numeración binario que tiene solo dos número, el "0" y el "1".
- En el caso de la última suma, esta debe dar como resultado "2" (en decimal) pero esta suma es en binario y el 2 no existe.

## Suma binaria

- Al igual que en el sistema decimal, cuando los números ya no alcanzan (solo hay hasta el 9), se utilizan combinaciones de estos para ampliar el alcance de la cuenta.
- En este caso se combina y el resultado es "10" que es 2 en sistema de numeración binario.
- Como se puede ver aparece un "acarreo" (el "1" a la izquierda del 0)

# ¿Qué es un circuito lógico?

- Circuito lógico es aquel que maneja la información en forma de "1" y "0", dos niveles lógicos de voltaje fijos.
- "1" nivel alto o "high" y "0" nivel bajo o "low".
- Los circuitos lógicos están compuestos por elementos digitales como
  - la compuerta AND (Y),
  - compuerta OR (O),
  - compuerta NOT (NO)......

    y combinaciones poco o muy complejas de los circuitos antes mencionados.

# circuito lógico

- Estas combinaciones dan lugar a otros tipos de elementos digitales como las compuertas, entre otros.
  - \- compuerta nand (No Y)
    \- compuerta nor (No O)
    \- compuerta or exclusiva (O exclusiva)
    \- multiplexores o multiplexadores
    \- demultiplexores o demultiplexadores
    \- decodificadores- codificadores
    \- memorias
    \- flip-flops
    \- microprocesadores
    \- microcontroladores
    \- etc.

# circuitos lógicos

- La electrónica moderna usa electrónica digital para realizar muchas funciones.
- Aunque los circuitos electrónicos podrían parecer muy complejos, en realidad se construyen de un número muy grande de circuitos muy simples.

# circuito lógico digital

- En un **circuito lógico digital** se transmite información binaria (ceros y unos) entre estos circuitos y se consigue un circuito complejo con la combinación de bloques de circuitos simples.
- La información binaria se representa en la forma de:

  - "0" ó "1",
  - "abierto" ó "cerrado" (interruptor),
  - "On" y "Off",
  - "falso" o "verdadero", etc.


# circuitos lógicos

- Los **circuitos lógicos** se pueden representar de muchas maneras. En el circuito la lámpara puede estar encendida o apagada ("on" o "off"), dependiendo de la posición del interruptor. (apagado o encendido)
- Los posibles estados del interruptor o interruptores que afectan un circuito se pueden representar en una **tabla de verdad**.

## Tabla de verdad

- La **tabla de verdad** es un instrumento utilizado para la simplificación de **circuitos digitales**.
- Las tablas de verdad pueden tener muchas columnas, pero todas las tablas funcionan de igual forma.
- Hay siempre una columna de salida que representa el resultado de todas las posibles combinaciones de las entradas.

# Tabla de verdad

- El número total de columnas en una **tabla de verdad** es la suma de las entradas que hay + 1 (la columna de la salida).

| Tabla de verdad | |
|---|---|
| **Columna(s) de entrada** | **Columna de salida** |
| Entrada (interruptor) | Salida (lámpara) |
| Abierto | Apagado |
| Cerrado | Encendido |



- El número de filas de la tabla de verdad es la cantidad de combinaciones que se pueden lograr con las entradas

# Tabla de verdad

| Tabla de verdad | | | |
|---|---|---|---|
| Switch 1 | Switch 2 | Switch 3 | Salida |
| 0 | 0 | 0 | ? |
| 0 | 0 | 1 | ? |
| 0 | 1 | 0 | ? |
| 0 | 1 | 1 | ? |
| 1 | 0 | 0 | ? |
| 1 | 0 | 1 | ? |
| 1 | 1 | 0 | ? |
| 1 | 1 | 1 | ? |

## circuitos lógicos

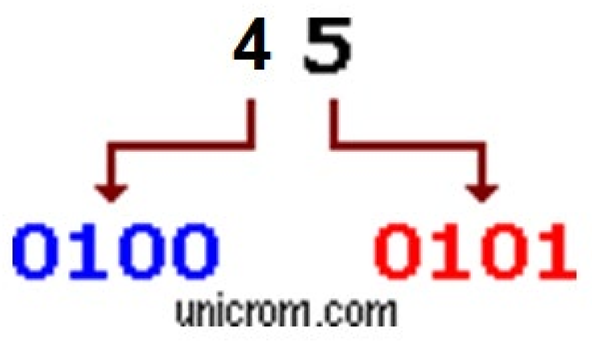
- Los **circuitos lógicos** son básicamente un arreglo de interruptores, conocidos como "compuertas lógicas" (compuertas AND, NAND, OR, NOR, NOT, etc.).
- Cada compuerta lógica tiene su **tabla de verdad**.
- Si pudiéramos ver con más detalle la construcción de las "compuertas lógicas", veríamos que son circuitos constituidos por transistores, resistencias, diodos, etc., conectados de manera que se obtienen salidas específicas para entradas específicas

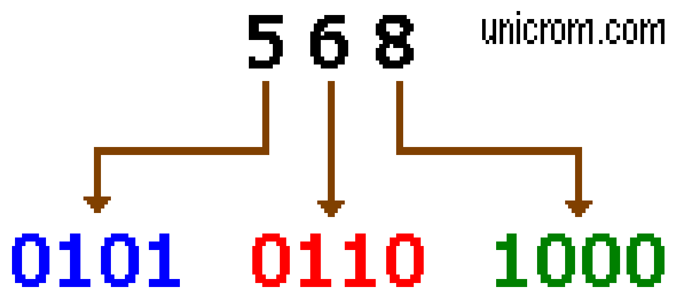
## circuitos integrados

- La utilización extendida de las **compuertas lógicas**, simplifica el diseño y análisis de circuitos complejos.
- La tecnología moderna actual permite la construcción de **circuitos integrados** (ICs) que se componen de miles (o millones) de **compuertas lógicas**.

## código BCD

- El **código BCD** utiliza 4 **dígitos binarios** para representar un dígito decimal (0 al 9). Cuando se hace conversión de binario a decimal típica no hay una directa relación entre el dígito decimal y el dígito binario.



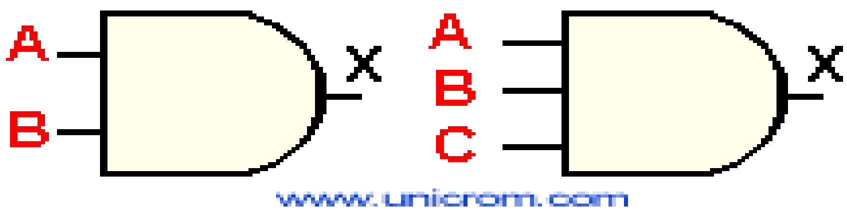
# código BCD

| DECIMAL | BCD |
|---|---|
| | 8 4 2 1 |
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

# Compuertas Digitales

## La compuerta lógica AND o compuerta Y

- La compuerta AND o Y lógica es una de las compuertas más simples dentro de la Electrónica Digital.




- La compuerta Y lógica más conocida tiene dos entradas A y B, aunque puede tener muchas más (A,B,C, etc.) y sólo tiene una salida X.

# AND

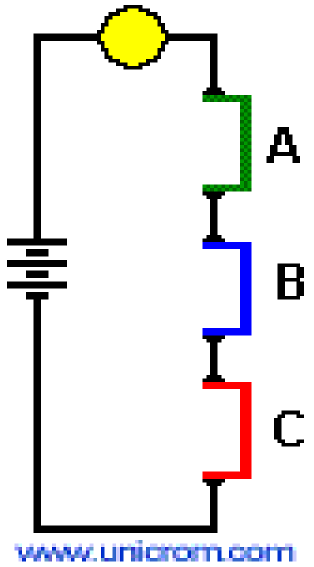
- La compuerta AND de 2 entradas tiene la siguiente tabla de verdad.

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

- Esta situación se representa en álgebra booleana como: X = A*B o X = AB.

# AND

- Una compuerta AND de 3 entradas se puede implementar con interruptores.
- La tabla de verdad se muestra donde:
  A = Abierto y C = Cerrado.

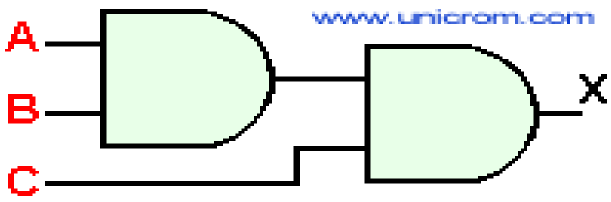



| A | B | C | Lámpara |
|---|---|---|---|
| A | A | A | Apagada |
| A | A | C | Apagada |
| A | C | A | Apagada |
| A | C | C | Apagada |
| C | A | A | Apagada |
| C | A | C | Apagada |
| C | C | A | Apagada |
| C | C | C | **Encendida** |

www.unicrom.com

# AND

- Una compuerta AND puede tener muchas entradas.
- Una compuerta AND de múltiples entradas puede ser creada conectando compuertas simples en serie.

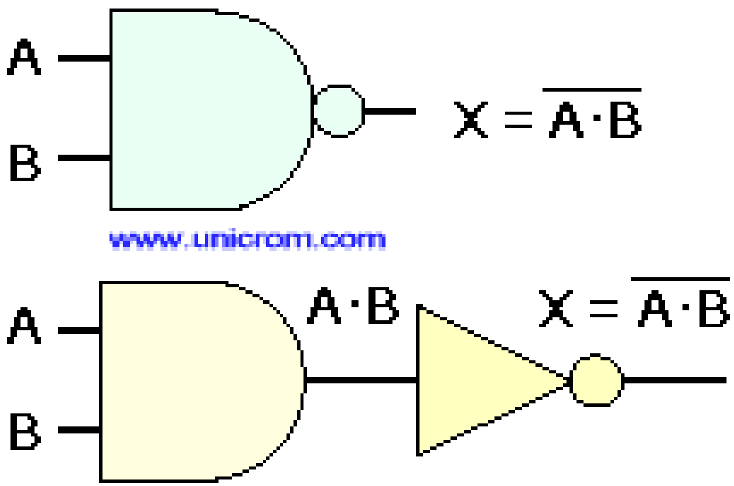



| Tabla de verdad | | | |
|---|---|---|---|
| A | B | C | X |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

www.unicrom.com

## Compuerta lógica NAND o compuerta No Y

- Una compuerta NAND (NO Y) de dos entradas, se puede implementar con la concatenación de una compuerta AND o "Y" de dos entradas y una compuerta NOT o "No" o inversora.



# NAND

- Al igual que en el caso de la compuerta AND, ésta se puede encontrar en versiones de 2, 3 o más entradas.
- Tablas de verdad de la compuerta NAND:

| A | B | X |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Compuerta NAND de 2 entradas

www.unicrom.com

| A | B | C | X |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Compuerta NAND de 3 entradas

# Compuerta lógica OR o compuerta O

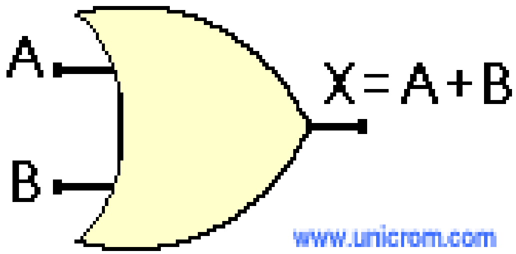
- La compuerta O lógica o compuerta OR es una de las compuertas mas simples dentro de la Electrónica Digital.
- La salida X de la compuerta OR será "1" cuando la entrada "A" o la entrada "B" estén en "1".
- En una compuerta OR,  la salida será "1", cuando en cualquiera de sus entradas haya un "1".

# OR

- La compuerta OR se representa con la siguiente función booleana:

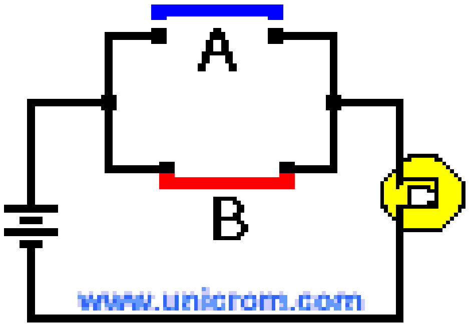
  X = A+B   ó   X = B+A

- La representación de la compuerta "OR" de 2 entradas y su tabla de verdad es:



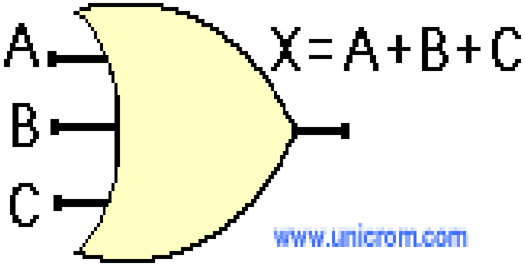

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

www.unicrom.com



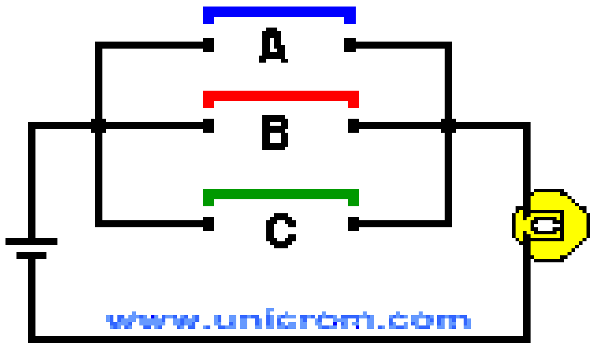
# Compuerta OR de tres entradas

- En las siguientes figuras se muestran la representación de la compuerta "OR" de tres entradas con su tabla de verdad  y la implementación con interruptores



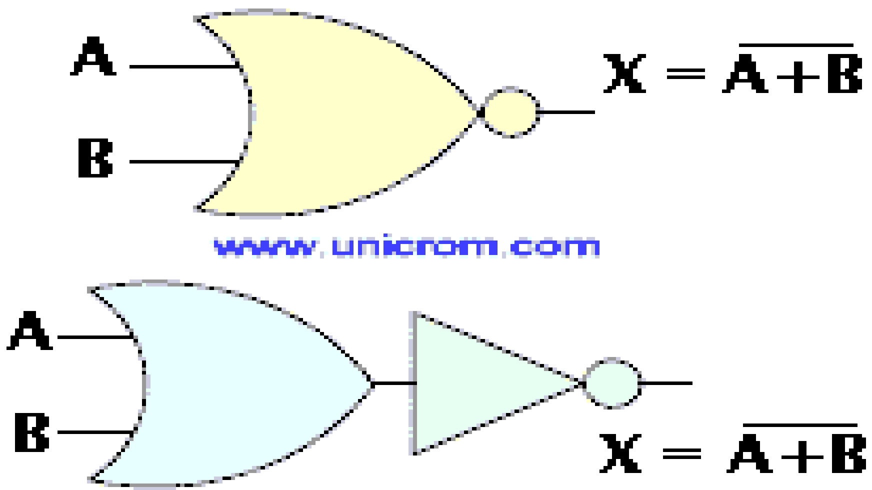

| A | B | C | X |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

www.unicrom.com



## Compuerta lógica NOR o compuerta NO O

- Una compuerta lógica NOR (No O) se puede implementar con la concatenación de una compuerta OR con una compuerta NOT.



# NOR

- Al igual que en el caso de la compuerta lógica OR, ésta se puede encontrar en versiones de 2, 3 o más entradas.

Tabla de verdad de una compuerta NOR de 2 entradas

| A | B | $X=\overline{A+B}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

www.unicrom.com

Tabla de verdad de una compuerta NOR de 3 entradas

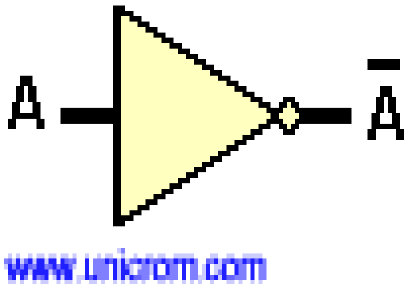
| A | B | C | $X=\overline{A+B+C}$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

# Compuerta lógica NOT o Compuerta inversora

- Dentro de la electrónica digital, no se podrían lograr muchas cosas si no existiera la compuerta NOT o compuerta No, también llamada compuerta inversora.
- La compuerta NOT como la compuerta AND y la compuerta OR es muy importante.
- La compuerta NOT entrega en su salida el inverso (opuesto) de la entrada.

# NOT

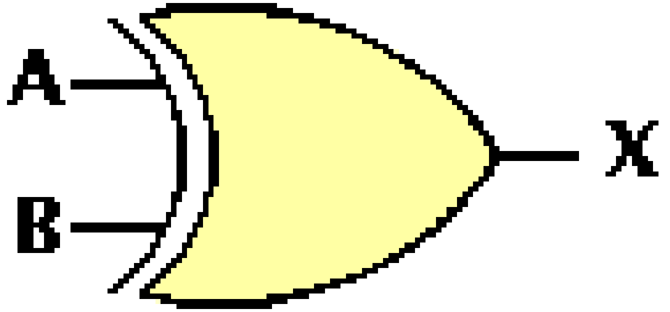
- El símbolo y la tabla de verdad son los siguientes:




| A | X |
|---|---|
| 0 | 1 |
| 1 | 0 |

## La compuerta lógica "O" exclusiva o XOR

- En la electrónica digital hay unas compuertas que no son comunes. Una de ellas es la compuerta XOR ó compuerta O exclusiva ó compuerta O excluyente.
- El siguiente diagrama muestra eñ símbolo de una compuerta XOR (O exclusiva) de 2 entradas:



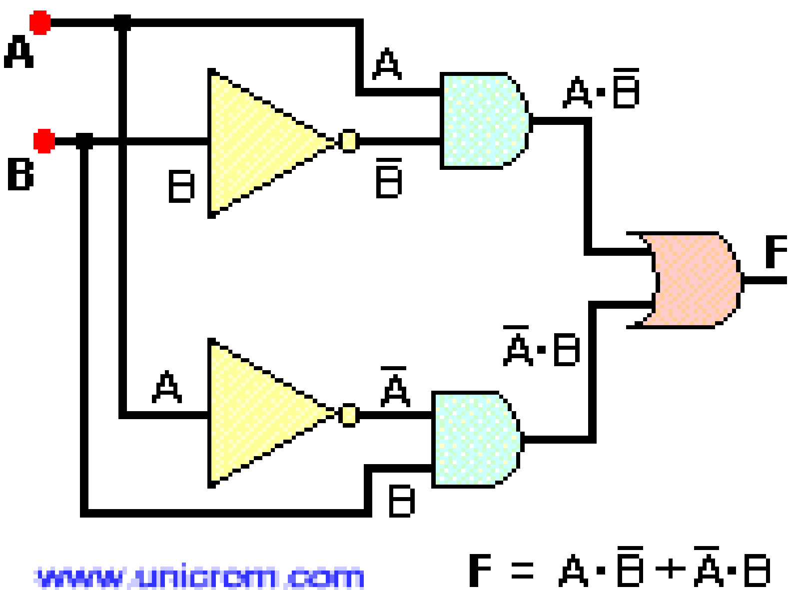
# XOR

- La figura muestra la tabla de verdad de una compuerta XOR de 2 entradas.
- Y se representa con la siguiente función booleana $X = \overline{A}.B + A.\overline{B}$

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# XOR de 3 entradas

- La siguiente figura muestra la tabla de verdad de una compuerta XOR de 3 entradas




| A | B | C | X |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

# Circuitos integrados digitales TTL (introducción)

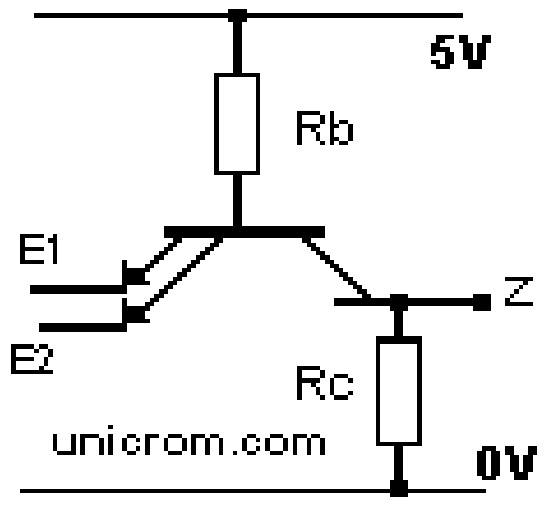
- TTL - Viene de las iniciales:

  Transistor - Transistor - Logic ó

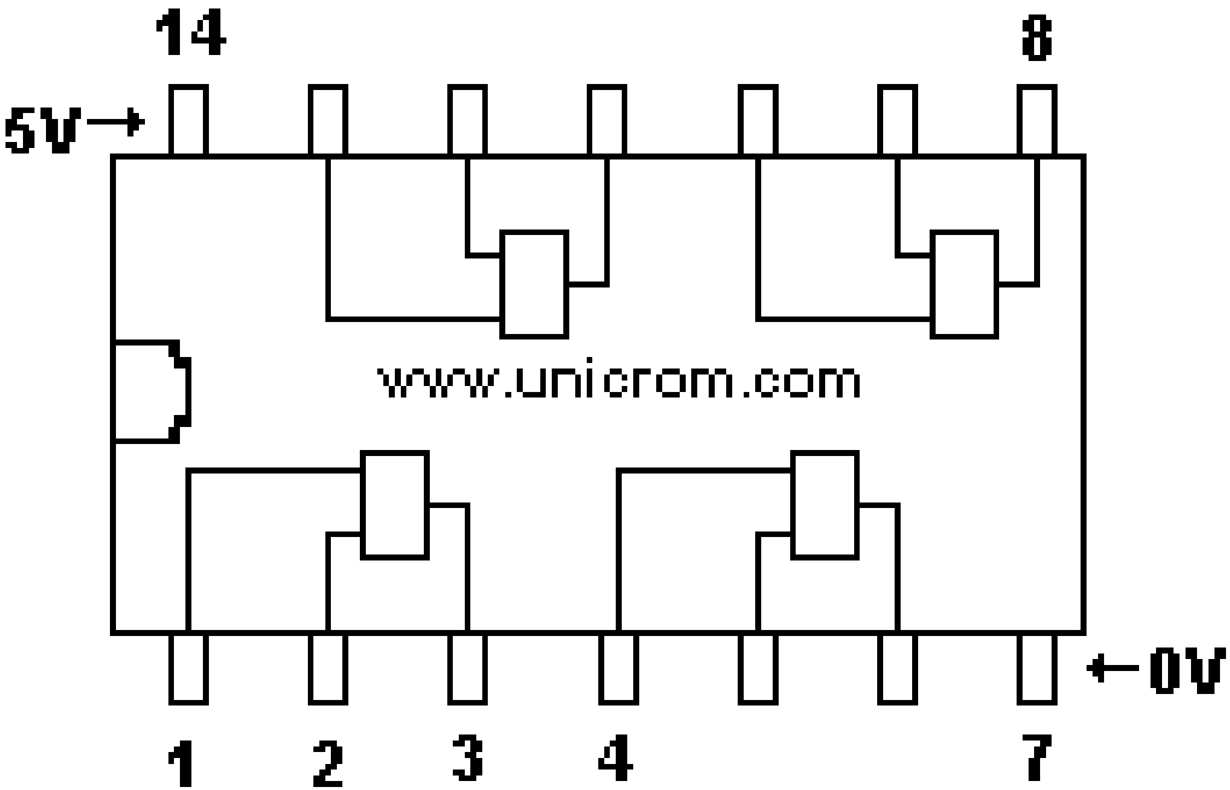
  Lógica Transistor Transistor.
- La familia de los circuitos integrados digitales TTL tienen las siguientes características:
- - La tensión o voltaje de alimentación es de + 5 Voltios, con $V_{mín}$ = 4.75 Voltios y $V_{máx}$ = 5.25 Voltios.
- Por encima del voltaje máximo el circuito integrado se puede dañar y por debajo del voltaje mínimo el circuito integrado no funcionaría adecuadamente.

# TTL

- - Su realización (fabricación) es con transistores bipolares multiemisores, como se observa en el gráfico más abajo.



# CI



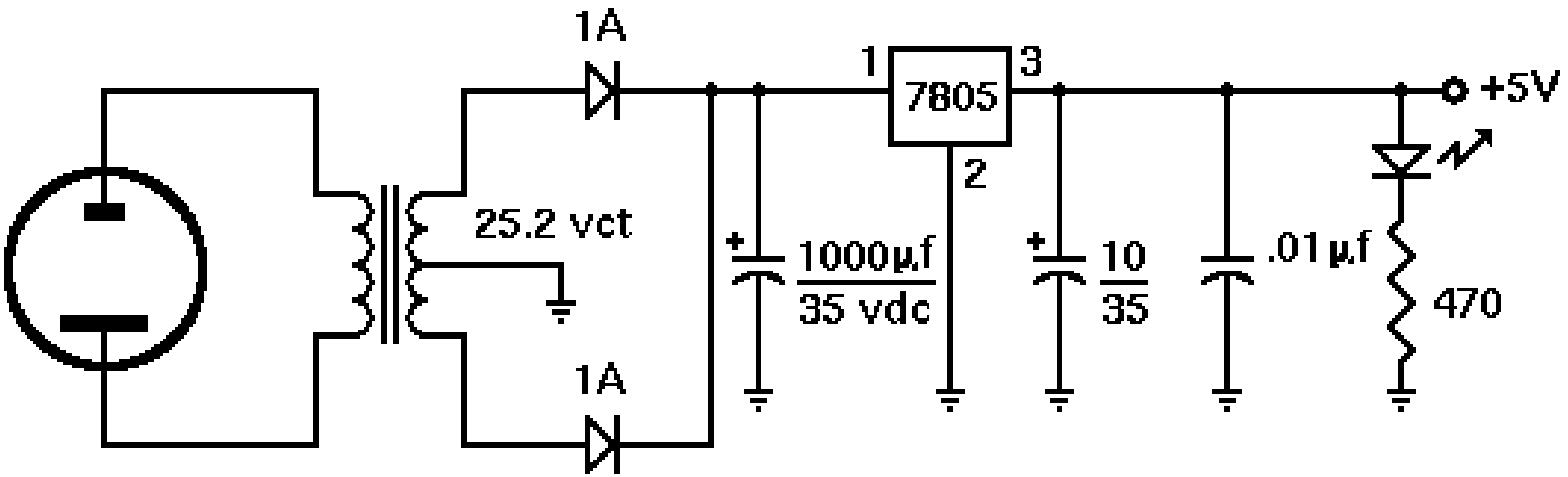
# Tipos de circuitos TTL

| Tipos de circuitos TTL según su número de serie y tipo | |
|---|---|
| **No de Serie** - Tipo | |
| 7400 | compuertas NAND |
| 7403 | compuertas NAND open collector |
| 7408 | compuertas AND |
| 7432 | compuertas OR |
| 7486 | compuertas EXOR |

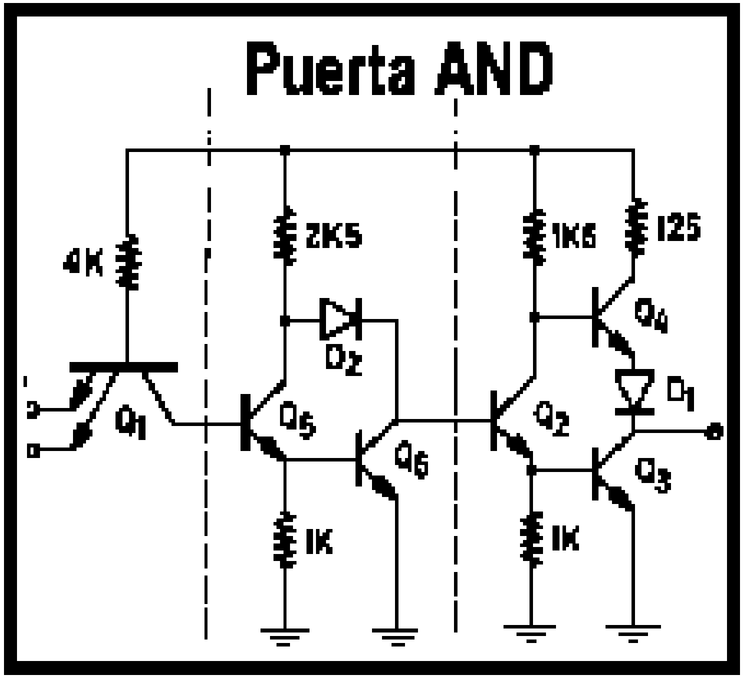
www.unicrom.com

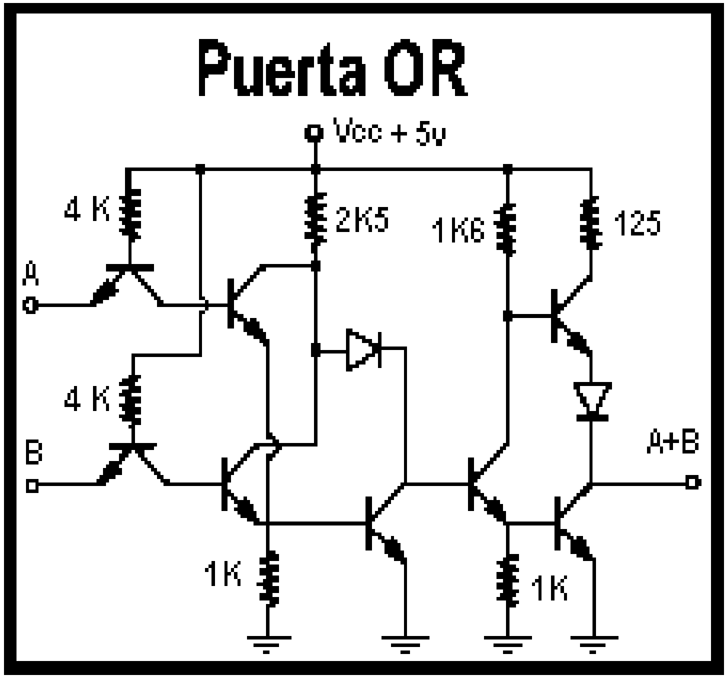
| Series especiales | |
|---|---|
| **74 LS XX** | Low Power Schottky (Tipo schottky de bajo consumo) |
| **74 S XX** | High Speed (alta velocidad) |
| **74 HC XX** | High Speed - C-MOS (Tipo C-MOS, alta velocidad) |

# Power Supply for Logic Circuits



# Compuerta AND y Compuerta OR



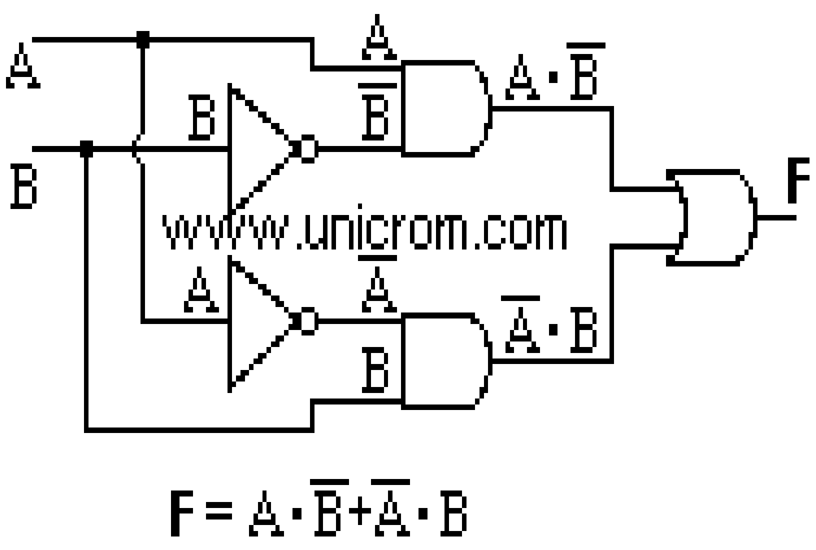



# Circuitos Combinacionales

- Un **circuito combinacional**, es un circuito cuya salida depende solamente de la "**combinación**" de sus entradas en el momento que se está realizando la medida en la salida.
- Analizando el circuito, con compuertas digitales, se ve que la salida de cada una de las compuertas que se muestran, depende únicamente de sus entradas.

# Circuitos Combinacionales

- La salida F (salida final o total del circuito) variará si alguna de las entradas A o B o las dos a la vez cambian.




| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# Aplicación:

# Diseño de un controlador para un robot seguidor de línea

# Especificaciones

- **Objetivo**: Diseñar un circuito digital, capaz gobernar un microbot, haciendo que éste siga una línea negra pintada sobre un fondo blanco.
- **Sensores**: El microbot está dotado de dos sensores digitales capacez de diferenciar el color negro del blanco.
- La salida de estos sensores es ’0’ cuando leen blanco y ’1’ cuando leen negro.

# Tablas de verdad

- Denominaremos a este bit como **C:**

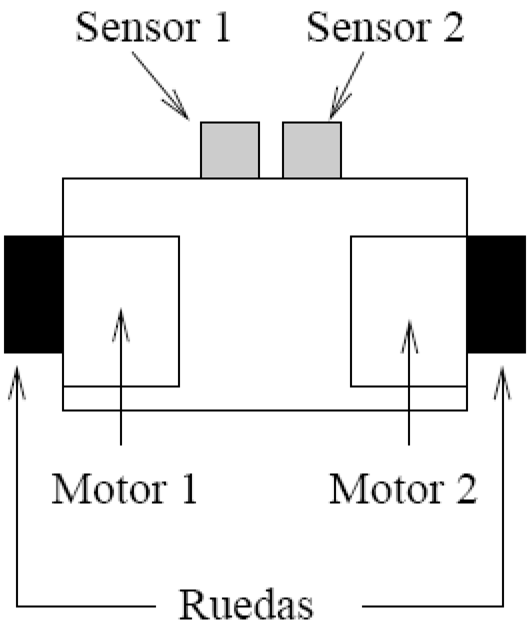
| **Sensor** | **C** |
|---|---|
| Color Blanco | 0 |
| Color Negro | 1 |

- **Motores:** Dos motores de corriente continua que son controlados cada uno mediante dos bits, denominados **S y P,** descritos mediante la siguiente tabla de verdad:

| **P** | **S** | **Motor** |
|---|---|---|
| 0 | 0 | Parado |
| 0 | 1 | Parado |
| 1 | 0 | Giro derecha |
| 1 | 1 | Giro izquierda |

# El robot:

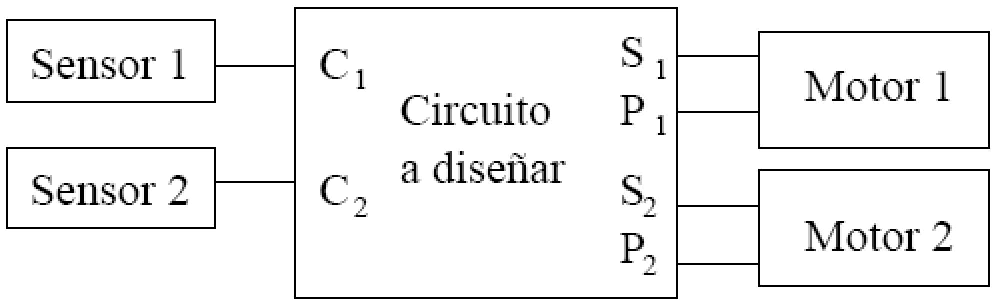
- El esquema del robot es el siguiente (visto desde arriba):



# Algoritmo:

- El algoritmo para seguir la línea negra es muy sencillo. Mientras los dos sensores detecten negro, el robot deberá avanzar. Cuando el sensor de la derecha detecte blanco y el de la izquierda negro, el robot girará a la izquierda y cuando ocurra el caso contrario girará a la derecha. Si ambos sensores leen blanco permanecerá parado.

# Diagrama de bloques y Tabla de verdad




| $C_1$ | $C_2$ | $S_1$ | $P_1$ | $S_2$ | $P_2$ |
|---|---|---|---|---|---|
| 0 | 0 | x | 0 | x | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |

# Ecuaciones booleanas del circuito

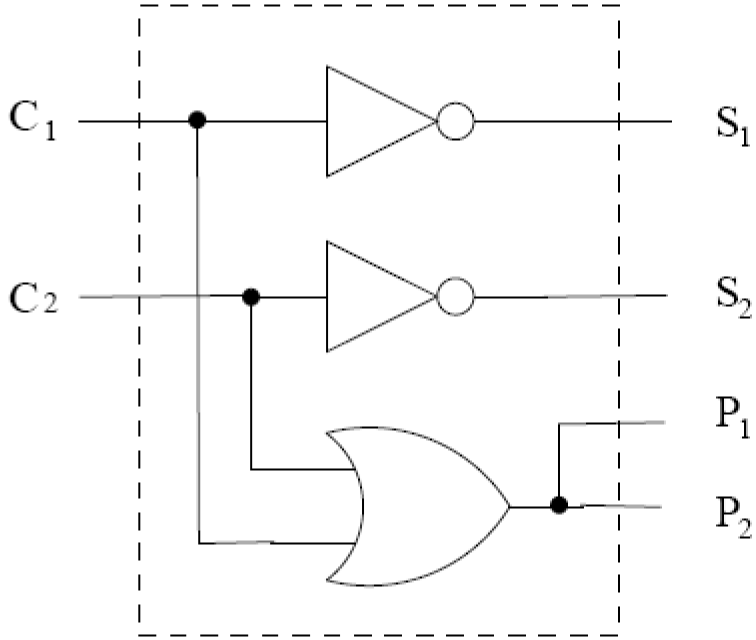
- Puesto que el circuito sólo tiene 2 variables de entrada, es inmediato obtener las expresiones de $S_1, P_1, S_2$ y $P_2$.

$$S_1 = \overline{C_2}$$

$$S_2 = \overline{C_1}$$

$$P_1 = P_2 = C_1 + C_2$$

# Implementación del circuito



# Algebra booleana

- Cuando se trabaja con **circuitos digitales** es muy común que al final de un diseño se tenga un circuito con un número de partes (**circuitos integrados** y otros) mayor al necesario.
- Para lograr que el circuito tenga la cantidad de partes correcta (la menor posible) hay que optimizarlo (reducirlo).
- Un diseño óptimo causará que:
  - El **circuito electrónico** sea más simple
  - El número de componentes sea el menor
  - El precio de proyecto sea el más bajo
  - La demanda de potencia del circuito sea menor
  - El mantenimiento del circuito sea más fácil.
  - Es espacio necesario (en el **circuito impreso**) para la implementación del circuito será menor.
- En consecuencia que el diseño sea el más económico posible.
- Una herramienta para reducir las expresiones lógicas de **circuitos digitales** es la matemáticas de expresiones lógicas, que fue presentada por **George Boole** en 1854, herramienta que desde entonces se conoce como **álgebra de Boole**.
- Las reglas del álgebra Booleana son:
- Nota:
  - (punto): significa producto lógico
  - **+** (signo de suma): significa suma lógica

# Operaciones básicas

| La operación AND o Y | | |
|---|---|---|
| $0 \cdot 0 = 0$ | | $0 \cdot 0 = 0$ |
| $0 \cdot 1 = 0$ | | $0 \cdot A = 0$ |
| $1 \cdot 0 = 0$ | | $A \cdot A' = 0$ |
| $1 \cdot 1 = 1$ | | $A \cdot A = A$ |
| **La operación OR o O** | | |
| $0 + 0 = 0$ | | $A + 0 = A$ |
| $0 + 1 = 1$ | | $A + 1 = 1$ |
| $1 + 0 = 1$ | | $A + A = A$ |
| $1 + 1 = 1$ | | $A + A' = 1$ |
| **La operación NOT o No** | | |
| $\overline{0} = 1$ | | $A'' = A$ |
| $\overline{1} = 0$ | | Nota: $A' = \overline{A}$ |

# Ley Distributiva, ley Asociativa, ley Conmutativa

## Ley Distributiva

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

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



## Ley Asociativa

$$(A \cdot B) \cdot C = A \cdot (B \cdot C) = A \cdot B \cdot C$$

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



## Ley Conmutativa

$$A \cdot B = B \cdot A$$

$$A+B = B+A$$

# Precedencia y Teorema de Morgan

**Precedencia**

$$A \blacksquare B = A \blacksquare B$$

$$A + B \blacksquare C = A + (B \blacksquare C)$$

$$A \blacksquare B + C = (A \blacksquare B) + C$$



**Teorema DeMorgan**

$$\overline{(A \cdot B)} = \overline{A} + \overline{B} \quad \text{(NAND)}$$

$$\overline{(A+B)} = \overline{A} \blacksquare \overline{B} \quad \text{(NOR)}$$

# Mapas de Karnaugh
## Simplificación de funciones booleanas

- Los Mapas de Karnaugh son una herramienta muy utilizada para la simplificación de circuitos lógicos.
- Cuando se tiene una función lógica con su tabla de verdad y se desea implementar esa función de la manera más económica posible se utiliza este método.
- Ejemplo: Se tiene la siguiente tabla de verdad para tres variables.
- Se desarrolla la función lógica basada en ella. (primera forma canónica). Ver que en la fórmula se incluyen solamente las variables (A, B, C) cuando F cuando es igual a "1".
- Si A en la tabla de verdad es "0" se pone A, si B = "1" se pone B, Si C = "0" se pone C, etc.

- Una vez obtenida la función lógica, se implementa el **mapa de Karnaugh**.
- Este mapa tiene 8 casillas que corresponden a 2n, donde n = 3 (número de variables (A, B, C))
- La primera fila corresponde a A = 0
  La segunda fila corresponde a A = 1
  La primera columna corresponde a BC = 00 (B=0 y C=0)
  La segunda columna corresponde a BC = 01 (B=0 y C=1)
  La tercera columna corresponde a BC = 11 (B=1 y C=1)
  La cuarta columna corresponde a BC = 10 (B=1 y C=0)
- En el **mapa de Karnaugh** se han puesto "1" en las casillas que corresponden a los valores de F = "1" en la tabla de verdad.
- Tomar en cuenta la numeración de las filas de la tabla de verdad y la numeración de las casillas en el **mapa de Karnaugh.**
- Para proceder con la simplificación, se crean grupos de "1"s que tengan 1, 2, 4, 8, 16, etc. (sólo potencias de 2).
- Los "1"s deben estar adyacentes (no en diagonal) y mientras más "1"s tenga el grupo, mejor.

| | A | B | C | F | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 1 | 0 | |
| 2 | 0 | 1 | 0 | 1 | $\rightarrow \bar{A} B \bar{C}$ |
| 3 | 0 | 1 | 1 | 1 | $\rightarrow \bar{A} B C$ |
| 4 | 1 | 0 | 0 | 1 | $\rightarrow A \bar{B} \bar{C}$ |
| 5 | 1 | 0 | 1 | 1 | $\rightarrow A \bar{B} C$ |
| 6 | 1 | 1 | 0 | 1 | $\rightarrow A B \bar{C}$ |
| 7 | 1 | 1 | 1 | 1 | $\rightarrow A B C$ |

www.unicrom.com

$$F = \bar{A} B \bar{C} + \bar{A} B C + A \bar{B} \bar{C} + A \bar{B} C + A B \bar{C} + A B C$$

www.unicrom.com

| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 (0) | 0 (1) | 1 (3) | 1 (2) |
| 1 | 1 (4) | 1 (5) | 1 (7) | 1 (6) |

# Simplificación de Funciones

- **La función mejor simplificada es aquella que tiene el menor número de grupos con el mayor número de "1"s en cada grupo**
- **Se ve del gráfico que hay dos grupos cada uno de cuatro "1"s, (se permite compartir casillas entre los grupos).**
- **La nueva expresión de la función boolena simplificada se deduce del mapa de Karnaugh.**
- **- Para el primer grupo (rojo): la simplificación da B (los "1"s de la tercera y cuarta columna) corresponden a B sin negar)**
  **- Para el segundo grupo (azul): la simplificación da A (los "1"s están en la fila inferior que corresponde a A sin negar)**
- **Entonces el resultado es F = B + A ó F = A + B**
- **Ejemplo:**
- **Una tabla de verdad como la de la, izquierda da la siguiente función booleana:**

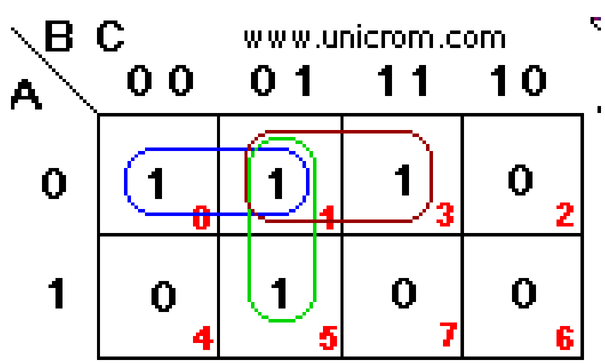
$$F = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + A\,\overline{B}\,C$$

- **Se ve claramente que la función es un reflejo del contenido de la tabla de verdad cuando F = "1"**

- Con esta ecuación se crea el **mapa de Karnaugh** y se escogen los grupos. Se lograron hacer 3 grupos de dos "1"s cada uno.
- Se puede ver que no es posible hacer grupos de 3, porque 3 no es potencia de 2. Se observa que hay una casilla que es compartida por los tres grupos.

| | A | B | C | F |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 0 |






La función simplificada es:

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

Grupo en azul: $\overline{AB}$, grupo marrón: $\overline{A}C$, grupo verde: $\overline{B}C$

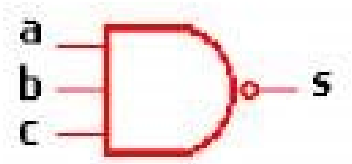
# Leyes de Morgan

**1º Ley:**

El producto lógico negado de varias variables lógicas es igual a la suma lógica de cada una de dichas variables negadas. Si tomamos un ejemplo para 3 variables tendríamos: $\sim(a.b.c) = \sim a + \sim b + \sim c$

# 1ra Ley de Morgan

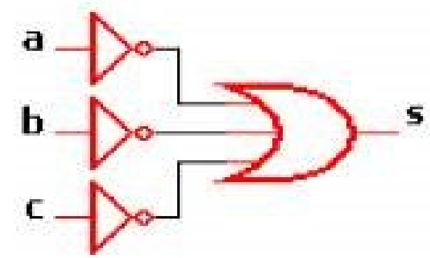
El primer miembro de esta ecuación equivale a una compuerta NAND de 3 entradas, representada en el siguiente gráfico y con su respectiva tabla de verdad.



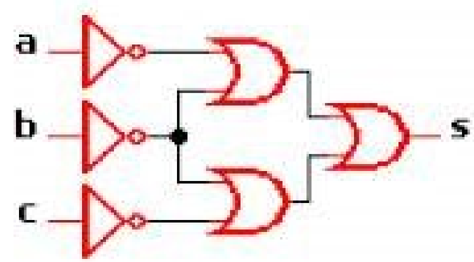

$s = \sim(a \,.\, b \,.\, c)$

| a | b | c | s |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

El segundo miembro de la ecuación se lo puede obtener de dos formas...




o bien...




$s = \sim a + \sim b + \sim c$

| a | b | c | s |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Fíjate que la tabla de verdad es la misma, ya que los resultados obtenidos son iguales. Acabamos de verificar la primera ley.

# Leyes de Morgan

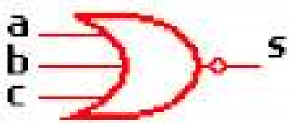
2da Ley

- La suma lógica negada de varias variables lógicas es igual al producto de cada una de dichas variables negadas...

$$\sim (a + b + c) = \sim a \,.\, \sim b \,.\, \sim c$$

# 2da Ley de Morgan

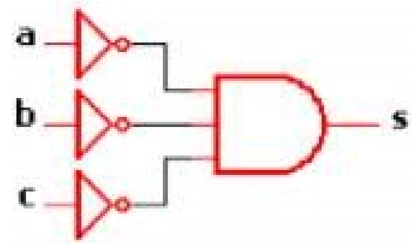
El primer miembro de esta ecuación equivale a una compuerta NOR de 3 entradas y la representamos con su tabla de verdad...



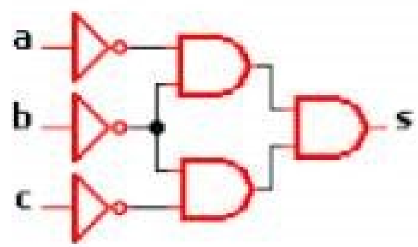

$s = \sim(a + b + c)$

| a | b | c | s |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

El segundo miembro de la ecuación se lo puede obtener de diferentes forma, aquí cité solo dos...




o bien...




| a | b | c | s |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

$s = \sim a \,.\, \sim b \,.\, \sim c$

Nuevamente... Observa que la tabla de verdad es la misma que para el primer miembro en el gráfico anterior. Acabamos así de verificar la segunda ley de Morgan.

Para concluir... Con estas dos leyes puedes llegar a una gran variedad de conclusiones, por ejemplo...

**Para obtener una compuerta AND** puedes utilizar una compuerta NOR con sus entradas negadas, o sea:

$$a . b = \sim( \sim a + \sim b)$$

**Para obtener una compuerta OR** puedes utilizar una compuerta NAND con sus entradas negadas, es decir...

$$a + b = \sim( \sim a . \sim b)$$

**Para obtener una compuerta NAND** utiliza una compuerta OR con sus dos entradas negadas, como indica la primera ley de Morgan...

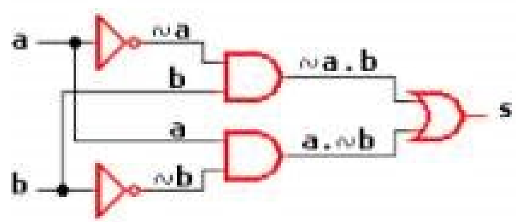
$$\sim (a.b) = \sim a + \sim b$$

**Para obtener una compuerta NOR** utiliza una compuerta AND con sus entradas negadas, eso dice la 2º ley de Morgan, así que... habrá que obedecer...

$$\sim(a + b) = \sim a . \sim b$$

**La compuerta OR-EX** tiene la particularidad de entregar un nivel alto cuando una y sólo una de sus entradas se encuentra en nivel alto. Si bien su función se puede representar como sigue:

Te puedes dar cuenta que esta ecuación te indica las compuertas a utilizar, y terminarás en esto...




| a | b | s |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

$$s = a . \sim b + \sim a . b$$

**Para obtener una compuerta NOR-EX** agregas una compuerta NOT a la salida de la compuerta OR-EX vista anteriormente y ya la tendrás. Recuerda que su función es...

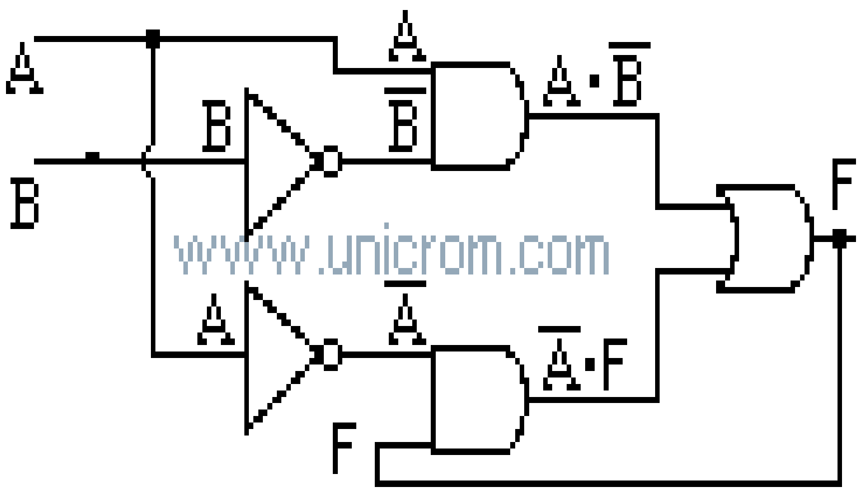
$$s = \sim(a . \sim b + \sim a . b)$$

**Para obtener Inversores (NOT)** puedes hacer uso de compuertas NOR o compuertas NAND, simplemente uniendo sus entradas.


$$s = \sim a$$

# Circuitos Secuenciales

- La diferencia principal entre un circuito combinacional y un **circuito secuencial** es que en el segundo caso hay una realimentación de una señal de salida hacia la entrada.

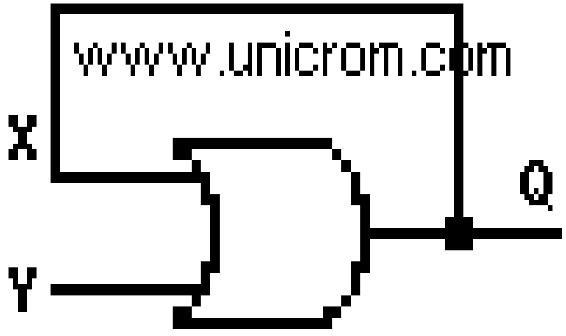


## Biestables o Flip Flops (FF)

- ¿Qué es Realimentación?
- Para poder entender bien el funcionamiento de un **biestable**, hay que tener bien claro el concepto de **realimentación**.
- Cuando una salida es conectada a una entrada del mismo circuito se dice que hay **realimentación**.
- Esta acción de realimentar una salida hacia la entrada causa en muchos casos un **efecto de memoria** (hay la capacidad de almacenar información).

## Datos de inicio:

- X = Q (la entrada X está conectada a la salida Q y no está disponible para ser modificada)
  Y = entrada disponible y esta puesta a "0" (nivel bajo), se puede modificar.
- Secuencia:
  1 - Q = 0 entonces X = 0
  2 - Y = 0
  3 - Y cambia a "1" y causa que...
  4 - Q cambia a "1"
  5 - Y cambia a "0" y ......
  6 - Q se queda en "1" (No cambia a "0", hay el **efecto memoria**)...

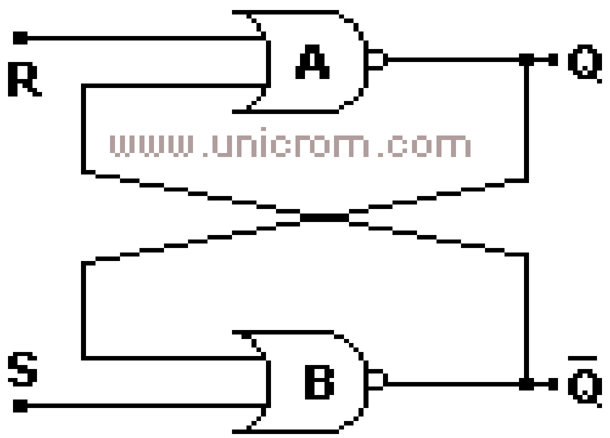


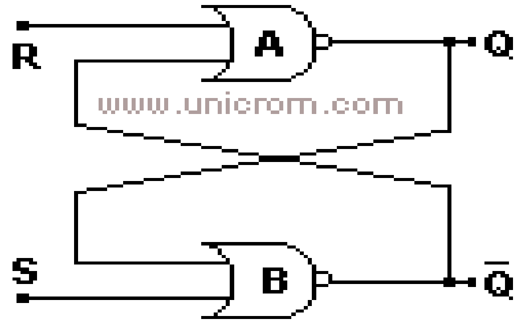
## Entonces

- esto se debe a que Q, que es "1", se realimenta a la entrada X causando que ésta se mantenga en "1" (Un "1" en cualquier entrada causa un "1" a la salida en una compuerta "OR")
- En otras palabras, este circuito recuerda que la entrada Y fue "1". Esta situación se mantendrá así, hasta que se le quite la alimentación.

# El biestable RS implementado con compuertas NOR.

- se llama RS porque sus entradas tiene los nombres SET (poner un "1" en la salida Q) y RESET (reponer o poner a "0" la salida Q)





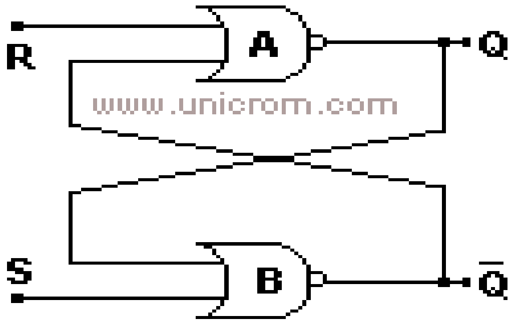

# Funcionamiento

- Caso SET
  1 - Se pone S = "1" y R = "0"
  2 - En la compuerta A, con S = "1" La salida Q = "0"
  3 - Q se realimenta a la entrada de la compuerta B,Q = "0" entonces la entrada también es "0"
  4 - Las dos entradas de la compuerta B están en nivel bajo, lo que causa que la salida Q pase a "1"
  5 - La salida Q se realimenta a la entrada de la compuerta A, y ....
  6 - Las dos entradas de la compuerta Y están en "1", lo que causa que la salida Q permanezca en "0"

# Funcionamiento

- Caso Reset

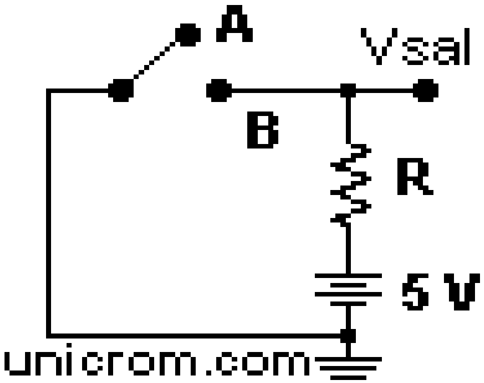
  1 - Las entradas cambian S pase de "1" a "0" y R = pasa de "0" a "1"
  2 - Con R = 1, Q en la compuerta B pasa a "0", y .....
  3 - Este Q se realimenta a la entrada de la compuerta A, y causa ....
  4 - Que la salida Q pase a "1"

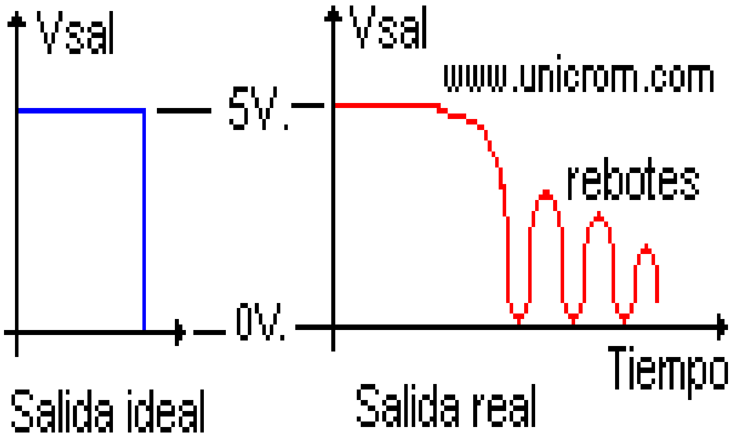


# Circuito eliminador de rebote.

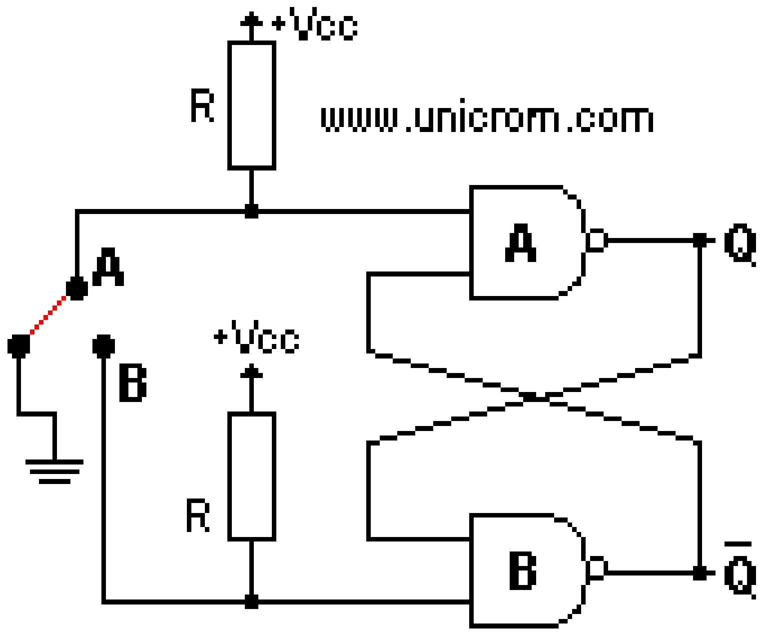
- Este circuito tiene una aplicación muy interesante:
- Cuando se implementa un conmutador con el propósito de alimentar un circuito, ya sea con un nivel bajo "0 V." o un nivel alto "5 V. Es muy difícil lograr que esta señal de entrada sea perfecta. Esto debido a que el conmutador es un elemento mecánico , que a la hora de cerrar produce rebotes.
- Estos rebotes serían similares a los de una pelota que se deja caer y al final se detiene. En un conmutador este fenómeno no es evidente pero si ocurre.

# Salida ideal y salida real de un conmutador



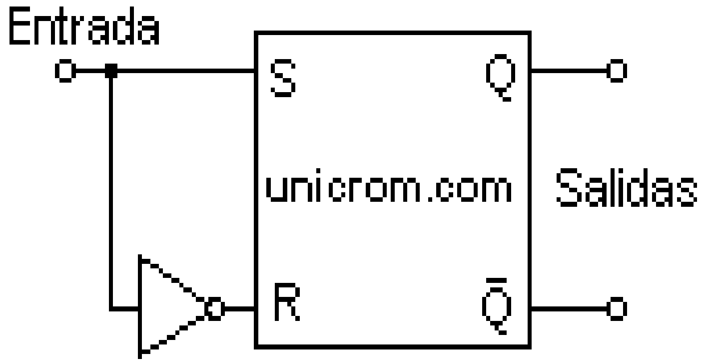



# Circuito eliminador de rebote.



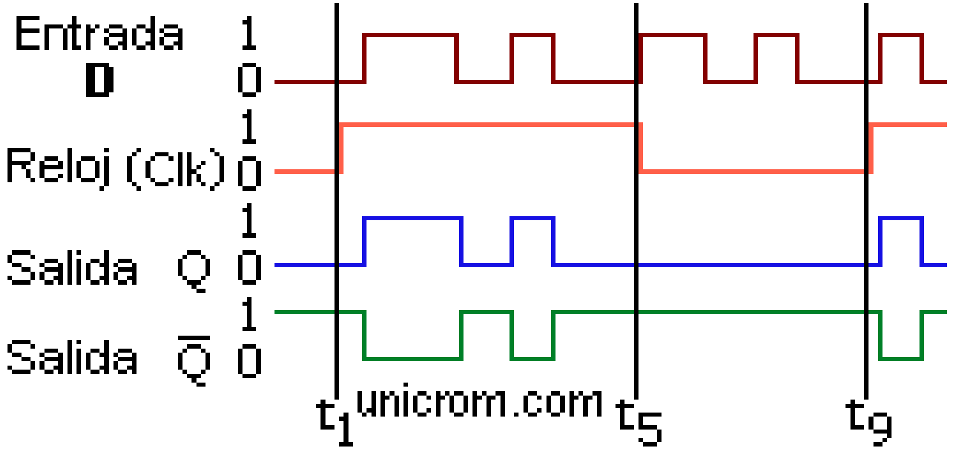
# Biestable tipo D

- Un bistable tipo D sólo tiene una entrada, pero mantiene las mismas salidas que el biestable tipo RS.
- Con un biestable RS se puede implementar un biestable tipo D si se coloca entre las dos entradas R y S un inversor como se muestra.



# Biestables sincrónicos o síncronos

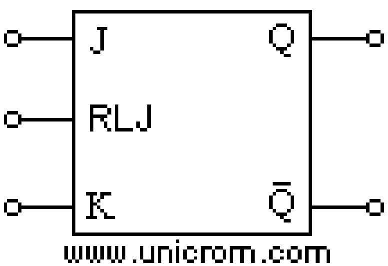
- Los **biestables síncronos** son otro tipo de **biestables**, que utilizan una señal especial (llamada señal de reloj).
- Esta señal establece el ritmo con el cual las señales se transmiten en el **biestable** y entre **biestables**. El circuito de reloj mas simple es un oscilador de onda cuadrada.

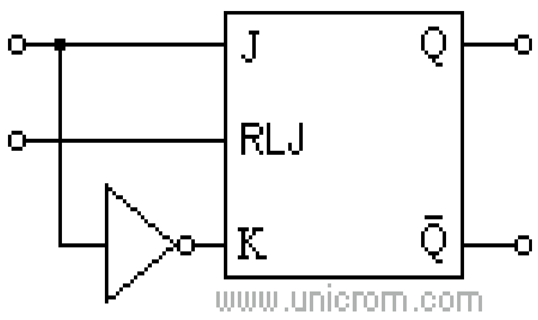



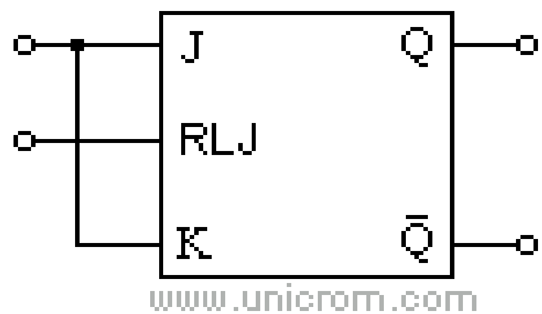
# Biestable JK, universal, tipo D, tipo T

- El biestable JK es también llamado "biestable universal" debido a que con él, se pueden implementar otros tipos de biestable, como el **biestable tipo D** o el **biestable tipo T**.
- En el siguiente diagrama se presenta la representación de un **biestable tipo JK** y las conexiones adicionales que hay que hacer para poder implementar un biestable tipo D y un **biestable tipo T**

# JK, D y T



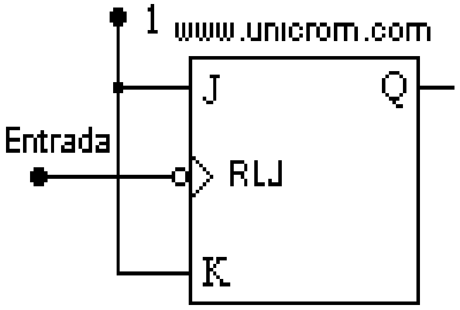

biestable JK

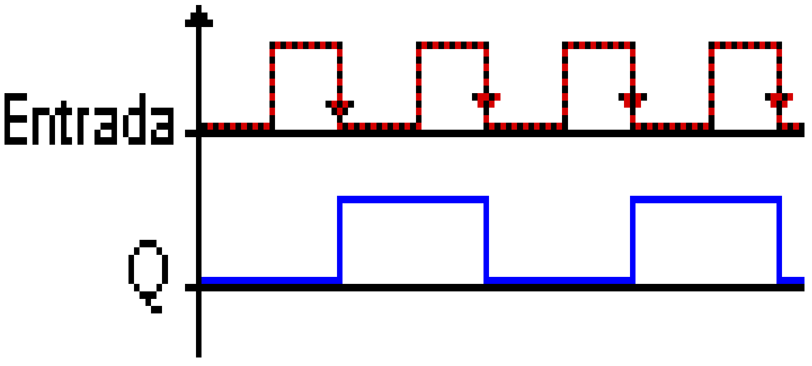
biestable tipo D

biestable tipo T

# Biestable tipo T

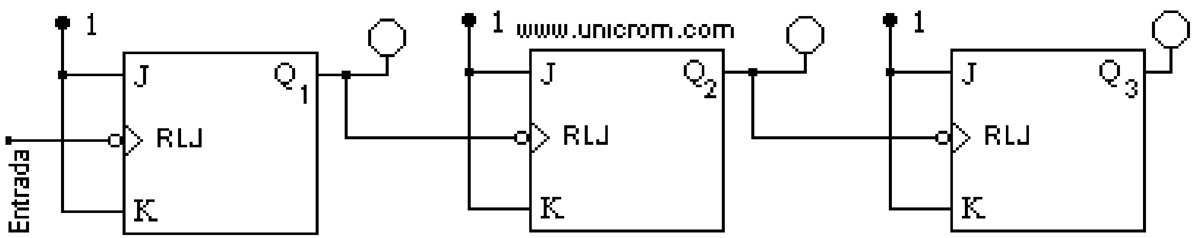
- En el siguiente gráfico un **Biestable JK** está cableado como **FF tipo T** (tienen las dos entradas unidas).

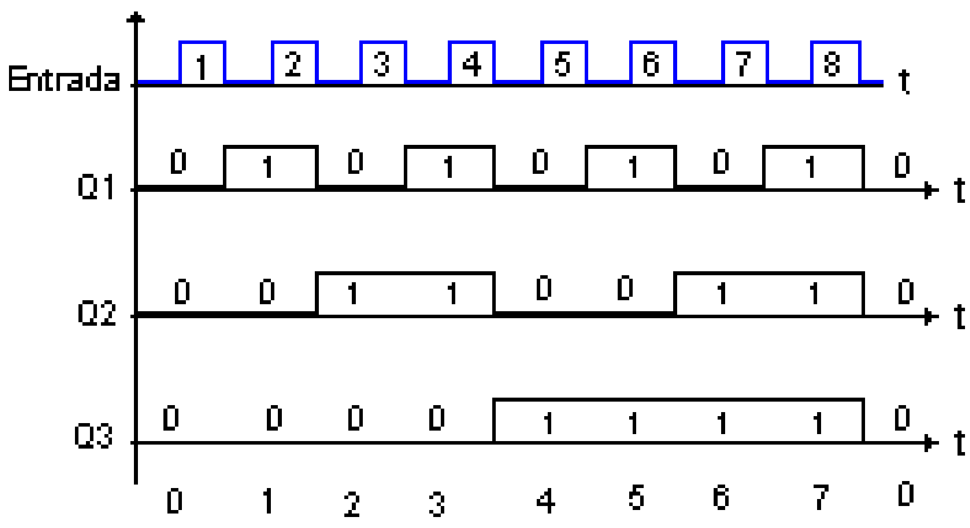


# Aplicación

- Contador asincrónico con biestable tipo T con FF JK



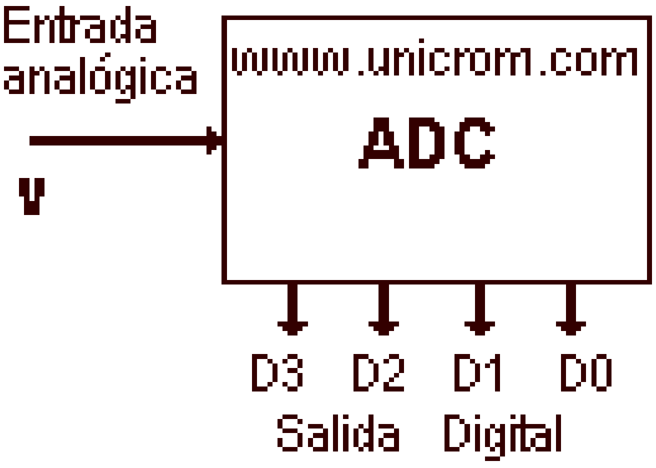



# Convertidor Analógico - Digital: ADC

- Las **señales analógicas**, pueden variar lentamente como la temperatura o muy rápidamente como una señal de audio.
- Lo que sucede con las señales analógicas es que son muy difíciles de manipular, guardar y después recuperar con **exactitud**.
- Si esta información analógica se convierte a **información digital**, se podría manipular sin problema.
- La información manipulada puede volver a tomar su valor analógico si se desea con un DAC (convertidor Digital a Analógico)

# Ejemplos de Convertidores Analógico - Digital

- Si se tiene un convertidor analógico - digital (CAD) de 4 bits y el rango de voltaje de entrada es de 0 a 15 voltios

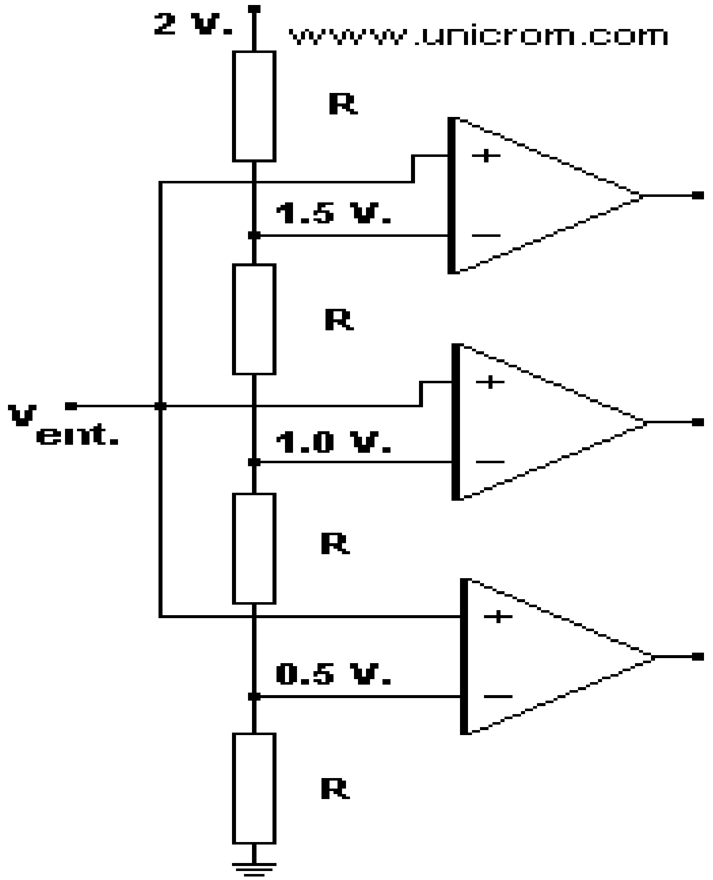


# tabla de conversión para el ADC

| Entrada analógica | Salida digital de 4 bits | | | |
|---|---|---|---|---|
| Voltios | D3 | D2 | D1 | D0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 |

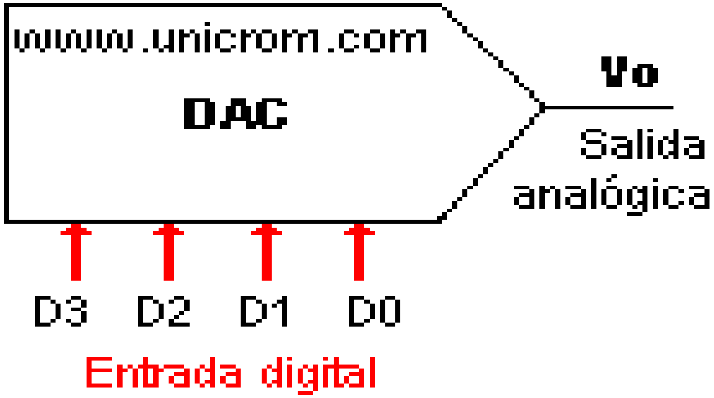
# CAD con comparadores

- una entrada analógica que puede variar entre 0 y 2 voltios y se desea convertir ésta en una salida digital de 2 bits.



## Convertidor Digital – Analógico (DAC)

- En la siguiente figura se representa un CDA de 4 bits. cada entrada digital puede ser sólo un "0" o un "1‟. El voltaje de salida analógica tendrá uno de 16 posibles valores dados por una de las 16 combinaciones de la entrada digital.



# tabla de conversión para el DAC

| Entrada digital | | | | Salida analógica |
|---|---|---|---|---|
| D3 | D2 | D1 | D0 | Voltios |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0.5 |
| 0 | 0 | 1 | 0 | 1.0 |
| 0 | 0 | 1 | 1 | 1.5 |
| 0 | 1 | 0 | 0 | 2.0 |
| 0 | 1 | 0 | 1 | 2.5 |
| 0 | 1 | 1 | 0 | 3.0 |
| 0 | 1 | 1 | 1 | 3.5 |
| 1 | 0 | 0 | 0 | 4.0 |
| 1 | 0 | 0 | 1 | 4.5 |
| 1 | 0 | 1 | 0 | 5.0 |

# Exactitud de un DAC

- Si se tiene diferentes tipos de DAC y todos ellos pueden tener una salida máxima de 15 voltios, se puede ver que la resolución y exactitud de la salida analógica es mayor cuando más bits tenga.

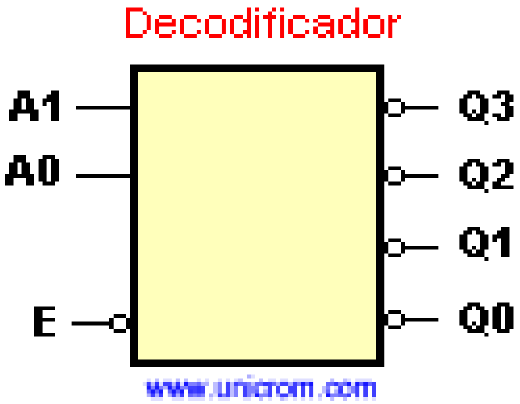
| # de bits del DAC | Resolución |
|---|---|
| 4 bits | 15 voltios / 15 = 1Voltio |
| 8 bits | 15 voltios / 255 = 58.8 miliVoltios |
| 16 bits | 15 voltios / 65536 = 0.23 milivoltios |
| 32 bits | 15 voltios / 4294967296 = 0.0000035 mv |

## Decodificadores

- El decodificador es un dispositivo que acepta una entrada digital **codificada en binario** y activa una salida.
- Este dispositivo tiene varias salidas, y se activará aquella que establezca el código aplicado a la entrada.
- Con un código de n bits se pueden encontrar $2^n$ posibles combinaciones. Si se tienen 3 bits (3 entradas) serán posibles $2^3 = 8$ combinaciones.

# Decodificadores

- Una combinación en particular activará sólo una salida.
- Por ejemplo: Para activar la salida Q2 hay que poner en la entrada el equivalente al número 2 en binario ($10_2$).



# Tabla de verdad de un decodificador

| Entradas | | | Salidas | | | |
|---|---|---|---|---|---|---|
| E | A0 | A1 | Q0 | Q1 | Q2 | Q3 |
| 1 | X | X | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |

# Multiplexor

- El multiplexor (MUX) es un circuito combinacional que tiene varios canales de datos de entrada y un canal de salida.
- Sólo un canal de la entrada pasará a la salida y este será el que haya sido escogido mediante unas **señales de control**.
- Ejemplo: Si utiliza un **multiplexor** de 4 canales de entrada.

## Multiplexor

- Una de los cuatro canales de entrada será escogido para pasar a la salida y ésto se logra con ayuda de las **señales de control o selección**.