LÓGICA DE PROGRAMACIÓN

LÓGICA COMPUTACIONAL

Álgebra de Boole

- **INTRODUCCION**

- Debido a que los computadores trabajan con información binaria, la herramienta matemática adecuada para el análisis y diseño de su funcionamiento es el Álgebra de Boole.

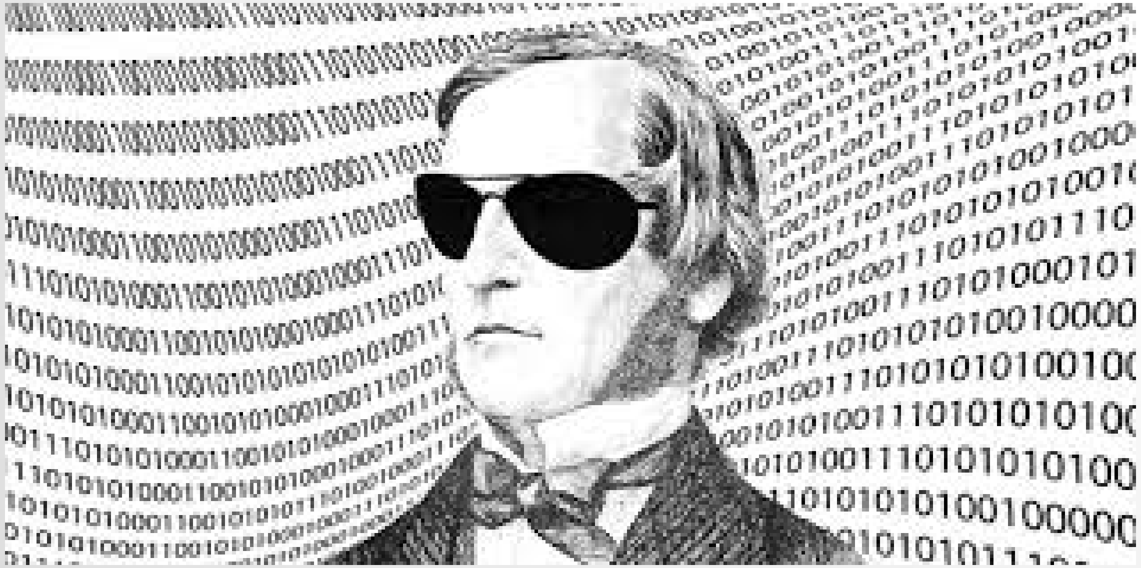
- El Álgebra de Boole fue desarrollada inicialmente para el estudio de la lógica. Ha sido a partir de 1938, fecha en que C.E. Shanon publicó un libro llamado "Análisis simbólico de circuitos con relés"

George Boole

El matemático inglés George Boole nació el 2 de noviembre de 1815 en Lincoln y falleció el 8 de diciembre de 1864 en Ballintemple, Irlanda.

Boole redujo la lógica a una álgebra simple. También trabajó en ecuaciones diferenciales, el cálculo de diferencias finitas y métodos generales en probabilidad.

Leyes del Álgebra de Boole

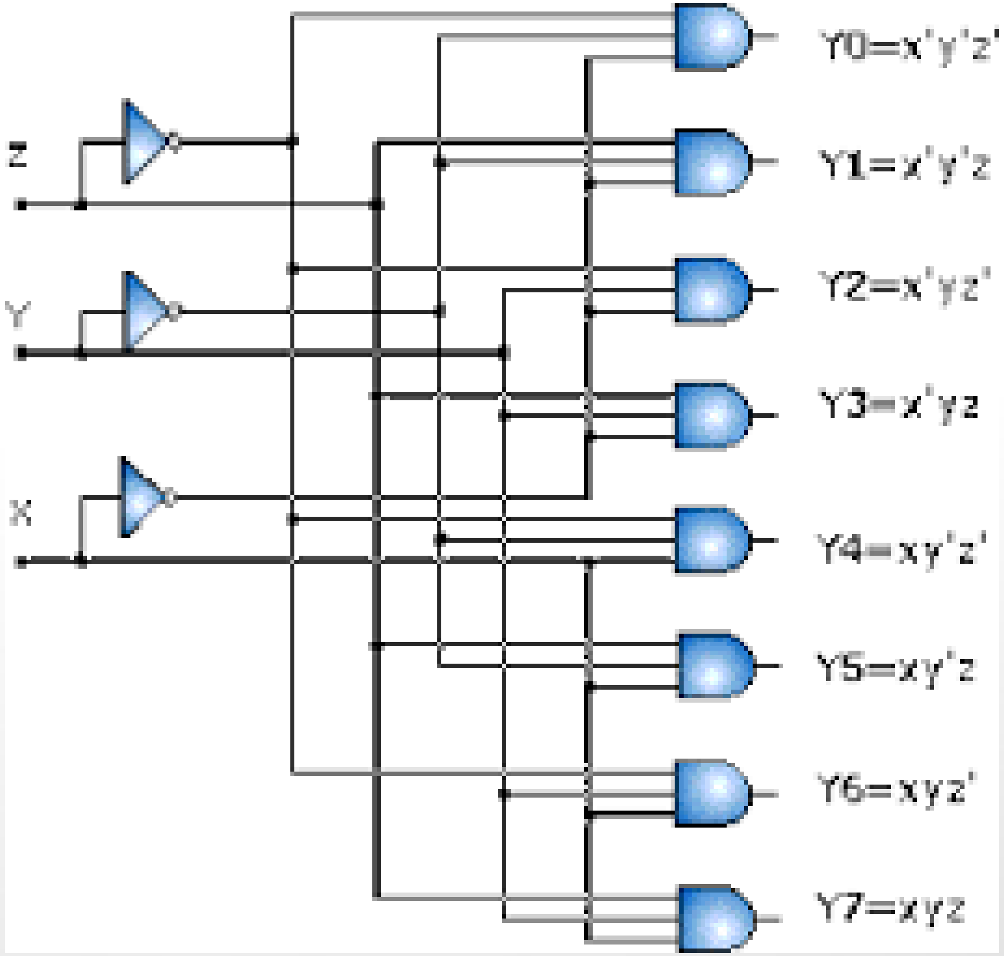


- Hoy en día, esta herramienta resulta fundamental para el desarrollo de los computadores ya que, con su ayuda, el análisis y síntesis de combinaciones complejas de circuitos lógicos puede realizarse con rapidez y eficacia.

Leyes del Álgebra de Boole

- **APLICACION A CIRCUITOS LOGICOS**

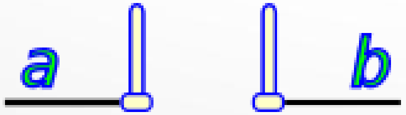
- Dado que los elementos de los circuitos utilizados en los computadores sólo admiten dos estados, las clases y operaciones básicas del Álgebra de Boole deberán particularizarse para este caso.



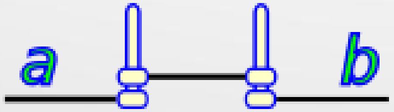
Álgebra de Boole aplicada a la Informática:

Se dice que una variable tiene **valor booleano** cuando, en general, la variable contiene un 0 lógico o un 1 lógico. Esto, en la mayoría de los lenguajes de programación, se traduce en *false* (falso) o *true* (verdadero), respectivamente.
Una variable puede no ser de tipo booleano, y guardar valores que, en principio, no son booleanos; ya que, globalmente, los compiladores trabajan con esos otros valores, numéricos normalmente aunque también algunos permiten cambios desde, incluso, caracteres, finalizando en valor booleano. ..

El 0 lógico
El valor booleano de negación suele ser representado como **false**, aunque también permite y equivale al valor natural, entero y decimal (exacto) 0, así como la cadena "false", e incluso la cadena "0".



El 1 lógico
En cambio, el resto de valores apuntan al valor booleano de afirmación, representado normalmente como **true**, ya que, por definición, el valor 1 se tiene cuando no es 0. Cualquier número distinto de cero se comporta como un 1 lógico, y lo mismo sucede con casi cualquier cadena (menos la "false", en caso de ser ésta la correspondiente al 0 lógico).



Compuerta Lógica

Una **puerta Lógica**, o **compuerta Lógica**, es un dispositivo electrónico que es la expresión física de un operador booleano en la Lógica de conmutación. Cada puerta Lógica consiste en una red de dispositivos interruptores que cumple las condiciones booleanas para el operador particular. Son esencialmente circuitos de conmutación integrados en un chip.

Claude Elwood Shannon experimentaba con relés o interruptores electromagnéticos para conseguir las condiciones de cada compuerta Lógica, por ejemplo, para la función booleana **Y** (AND) colocaba interruptores en circuito serie, ya que con uno solo de éstos que tuviera la condición «abierto», la salida de la compuerta Y sería = 0, mientras que para la implementación de una compuerta **O** (OR), la conexión de los interruptores tiene una configuración en circuito paralelo.
La tecnología microelectrónica actual permite la elevada integración de transistores actuando como conmutadores en redes lógicas dentro de un pequeño circuito integrado. El chip de la CPU es una de las máximas expresiones de este avance tecnológico.

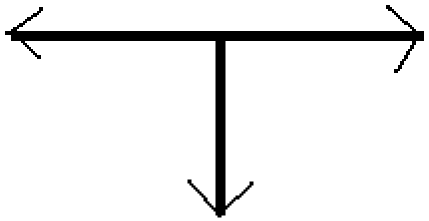
En nanotecnología se está desarrollando el uso de una compuerta Lógica molecular, que haga posible la miniaturización de circuitos.

Circuito Lógico:

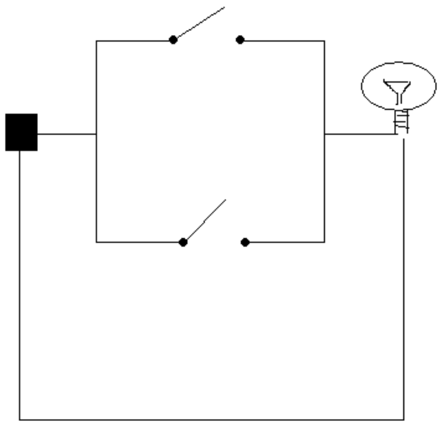
LÓGICA

$p \vee q$

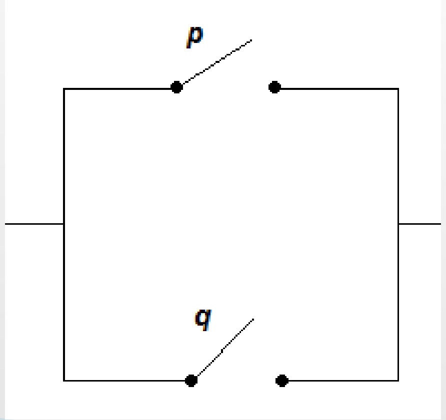
es isomorfa con la



ELECTRÓNICA



y juntas componen



Circuito Lógico

Circuito Lógico:

Son estructuras formales (sistemas abstractos) que representan sistemas para la transmisión de información de toda índole (desde la electricidad hasta datos informáticos) simulando el comportamiento real de un circuito eléctrico.

Un Circuito eléctrico es toda de transmisión de impulsos eléctricos.

Los circuitos eléctricos reales tienen los siguientes elementos:

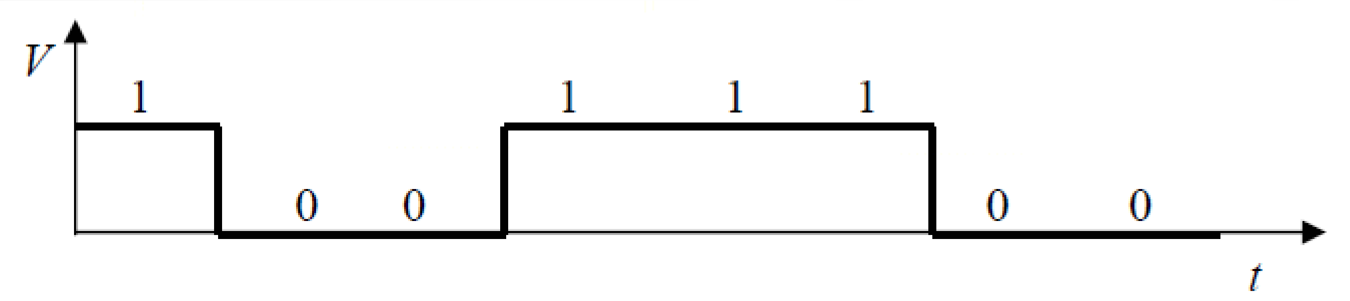
A. Fuente de energía (batería, pila, tomacorriente)
B. Cable de transmisión
C. Interruptores (llamados así porque interrumpen o permiten el flujo de electricidad)
D. Resistencia o receptor de información (foco, lámpara)

La energía parte del polo negativo de la fuente y se transmite por el cable llega hasta el foco (que se prende) y viaja por el cable hasta llegar al polo positivo de la fuente.

Circuito Lógico:

Un circuito lógico es un dispositivo que tienen una o más entradas y exactamente una salida. En cada instante cada entrada tiene un valor, 0 o 1; estos datos son procesados por el circuito para dar un valor en su salida, 0 o 1.

Los valores 0 y 1 pueden representar ciertas situaciones físicas como, por ejemplo, un voltaje nulo y no nulo en un conductor.



Circuito Lógico:

Aunque los **circuitos electrónicos** podrían parecer muy complejos, en realidad se construyen de un número muy grande de circuitos muy simples.

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: (ver gráficos arriba)

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

Los **circuitos lógicos** se pueden representar de muchas maneras. En los circuitos representados con gráficos 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**.

Circuito Lógico (Compuertas Básicas):

Los circuitos lógicos se construyen a partir de ciertos circuitos elementales denominados compuertas lógicas, entre las cuales diferenciaremos:

- Compuertas lógicas básicas: OR, AND, NOT.
- Compuertas lógicas derivadas: NOR, NAND.

Compuerta And: La operación *And* requiere que todas las señales sean simultáneamente verdaderas para que la salida sea verdadera. Así, el circuito de la figura necesita que ambos interruptores estén cerrados para que la luz encienda.

En una *compuerta AND con entradas A y B, la salida Y resulta: Y = A∗B*

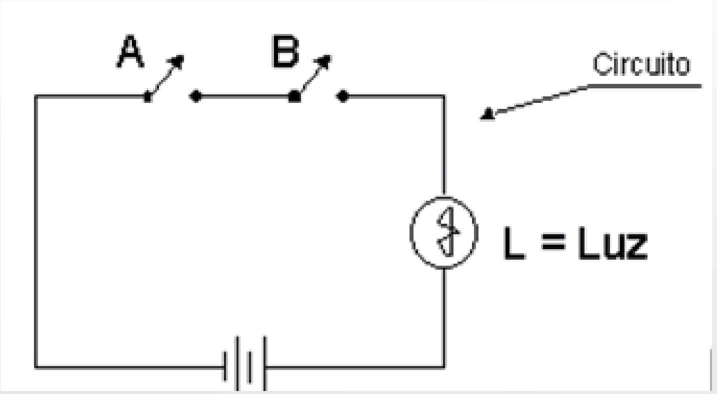


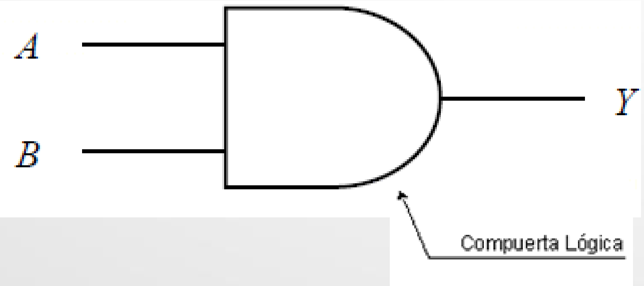
Tabla de Verdad

A	B	L
0	0	0
0	1	0
1	0	0
1	1	1

Circuito Lógico (Compuertas Básicas):

Compuerta And: Los estados posibles del circuito se pueden modelar en la Tabla de Verdad que tiene asociada. Sabemos que los interruptores sólo pueden tener dos estados, abiertos o cerrados, si el interruptor abierto se representa mediante el cero (0 o falso) y el cerrado mediante el valor uno (1 o verdadero) entonces en la tabla de verdad asociada se puede ver la situación que se describía en el párrafo anterior, cuando se decía que la luz sólo prende cuando ambos interruptores están cerrados, es decir, si A = 1 y B = 1 entonces L = 1.

Para efectos de este trabajo, la operación And la representaremos como la función And(A, B), donde A y B serían los parámetros de entrada (los mismos valores de A y B en el circuito) y L = And(A, B), correspondería a la forma de asignación de valor a L. En este caso el parámetro de salida es la misma función And.



Circuito Lógico (Compuertas Básicas):

Compuerta Or: La operación Or tiene similares características a la operación And, con la diferencia que basta que una señal sea verdadera para que la señal resultante sea verdadera. En la figura se puede ver tal situación.

Note que en el circuito los interruptores están en paralelo, por lo cual basta que uno de ellos esté cerrado para que el circuito se cierre y encienda la luz.

En una *compuerta OR con entradas A y B, la salida Y resulta: Y = A+ B*

La operación Or también tiene una representación funcional como Or (A, B) donde A y B serían los parámetros de entrada (los mismos valores de A y B en el circuito) y L = Or(A, B), correspondería a la forma de asignación de valor a L. En este caso, el parámetro de salida es la misma función Or.

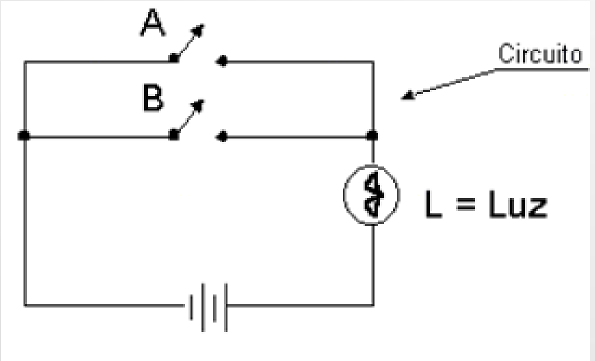
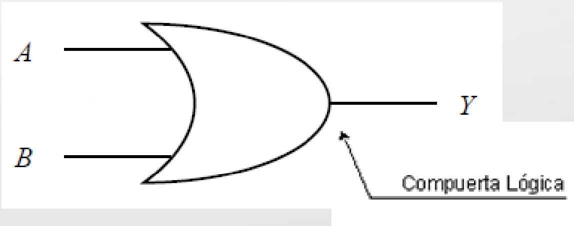


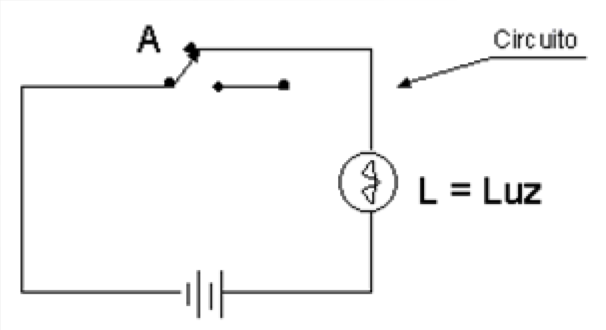
Tabla de Verdad

A	B	L
0	0	0
0	1	1
1	0	1
1	1	1



Circuito Lógico (Compuertas Básicas):

Compuerta Not: La última de la tres operaciones fundamentales, la cual también se conoce como negación, complemento o inversión, es mucho más simple que las anteriores. En la figura se puede observar el circuito, que en este caso tiene la particularidad de que al estar el interruptor abierto la luz enciende, cuando él está en posición de cerrado la luz permanecería apagada.



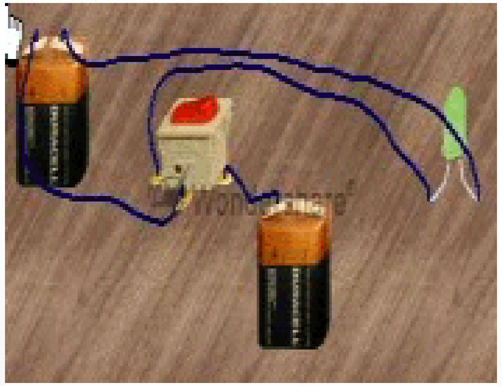
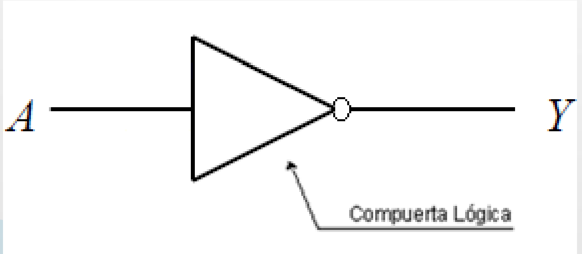


Tabla de Verdad

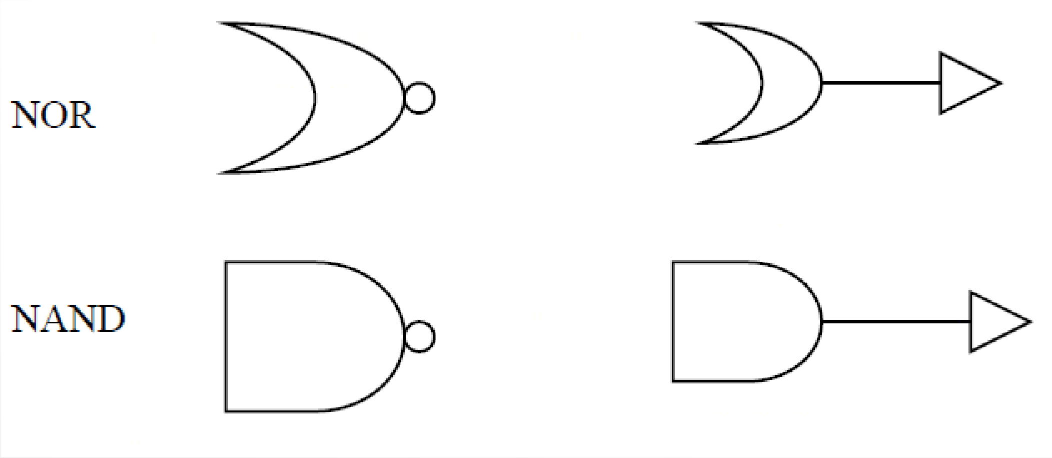
A	L
0	1
1	0

La notación funcional para esta operación será Not(A), donde A corresponde a la señal de entrada y Not(A) corresponde al valor complementario de A. **En una *compuerta NOT con entrada A, la salida Y resulta:*** $Y = \overline{A}$



Circuito Lógico (Compuertas):

Compuertas NOR y NAND: Las compuertas NOR y NAND no son básicas. Una *compuerta NOR equivale a una* compuerta OR seguida de una compuerta NOT. Una *compuerta NAND equivale a una* compuerta AND seguida de una compuerta NOT.



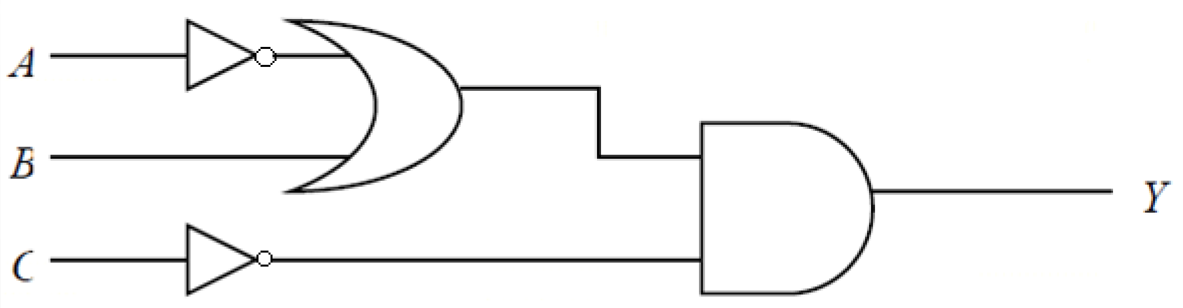
Por lo tanto, cuando las entradas son *A y B, las salidas de estas compuertas resultan:*

- NOR: $Y = A + B$
- NAND: $Y = A * B$

Circuito Lógico:

Los circuitos lógicos se forman combinando compuertas lógicas. La salida de un circuito lógico se obtiene combinando las tablas correspondientes a sus compuertas componentes. Por ejemplo:

$$Y = (\overline{A} + B) * \overline{C}$$



Es fácil notar que las tablas correspondientes a las compuertas OR, AND y NOT son respectivamente idénticas a las tablas de verdad de la disyunción, la conjunción y la negación en la lógica de enunciados, donde sólo se ha cambiado V y F por 0 y 1. Por lo tanto, los circuitos lógicos, de los cuales tales compuertas son elementos, forman un *álgebra de Boole al igual que los enunciados de la lógica de enunciados.*

Circuito Lógico:

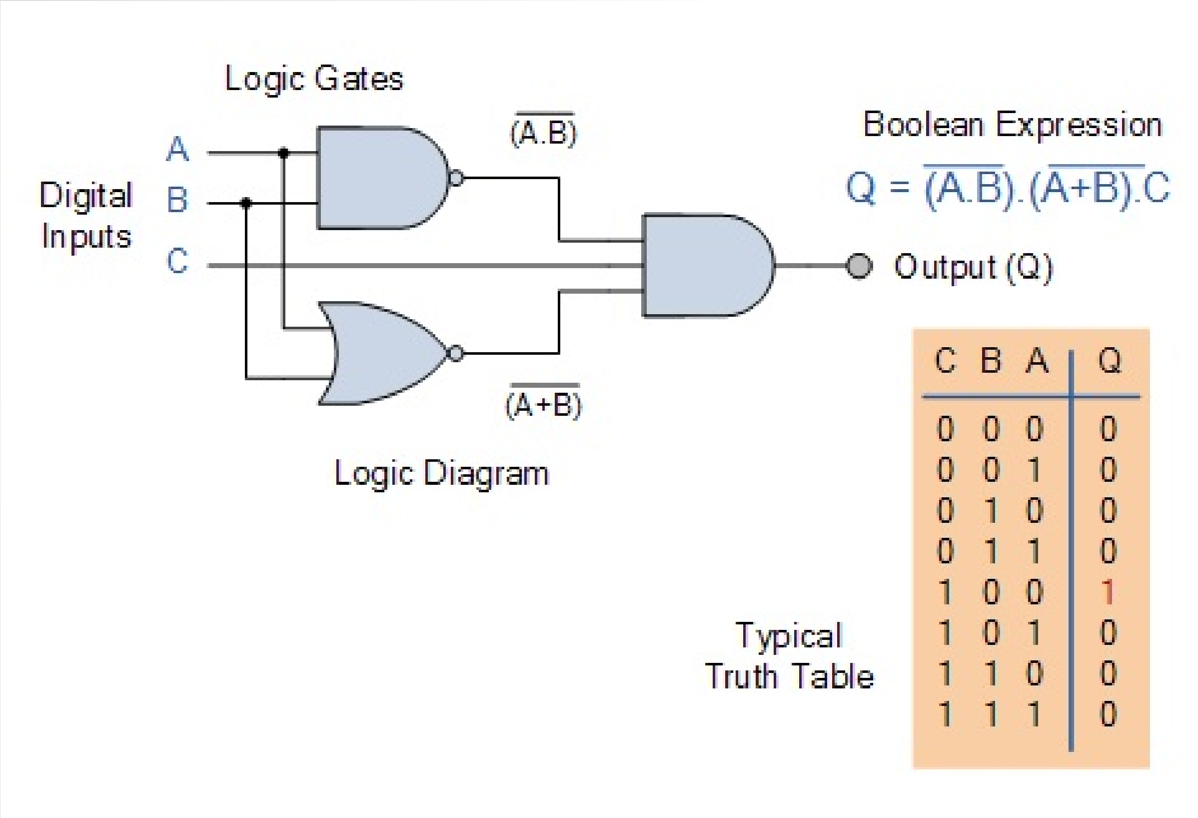
Adoptaremos, entonces, aquí las mismas convenciones adoptadas en el caso del álgebra de Boole:

• Omitimos el símbolo *, usándose en su lugar la yuxtaposición de variables.
• Establecemos que + es más fuerte que * y * es más fuerte que.

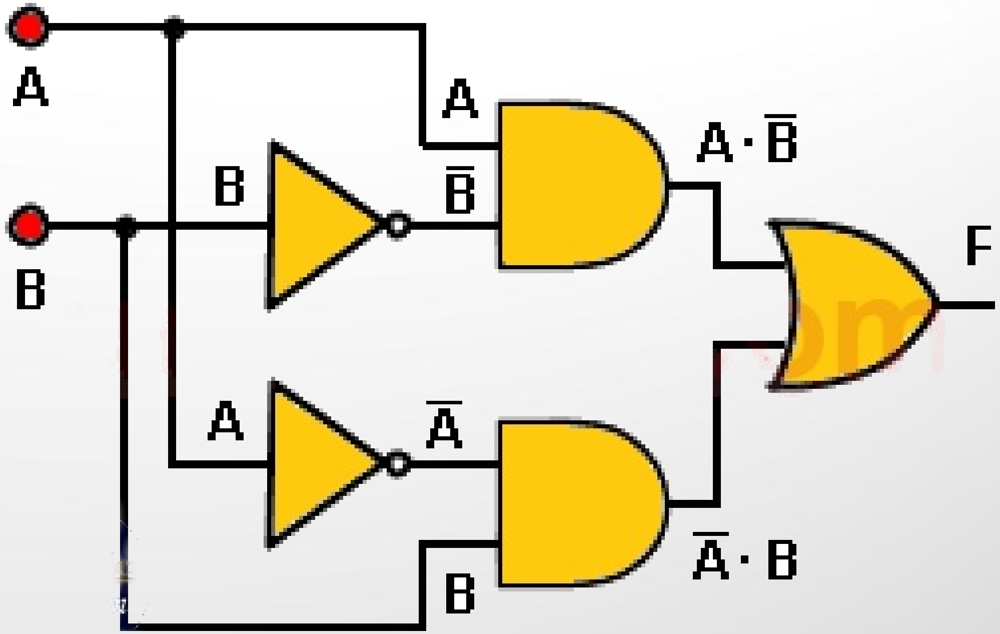
Puesto que tanto el álgebra de Boole es la estructura algebraica tanto de los circuitos como de la lógica de enunciados, la salida de un circuito lógico también puede expresarse en el lenguaje de la lógica de enunciados. Por ejemplo, la salida del circuito anterior resulta:

$$(\overline{A} + B) * \overline{C} \qquad\qquad (\neg p \vee q) \wedge \neg r$$

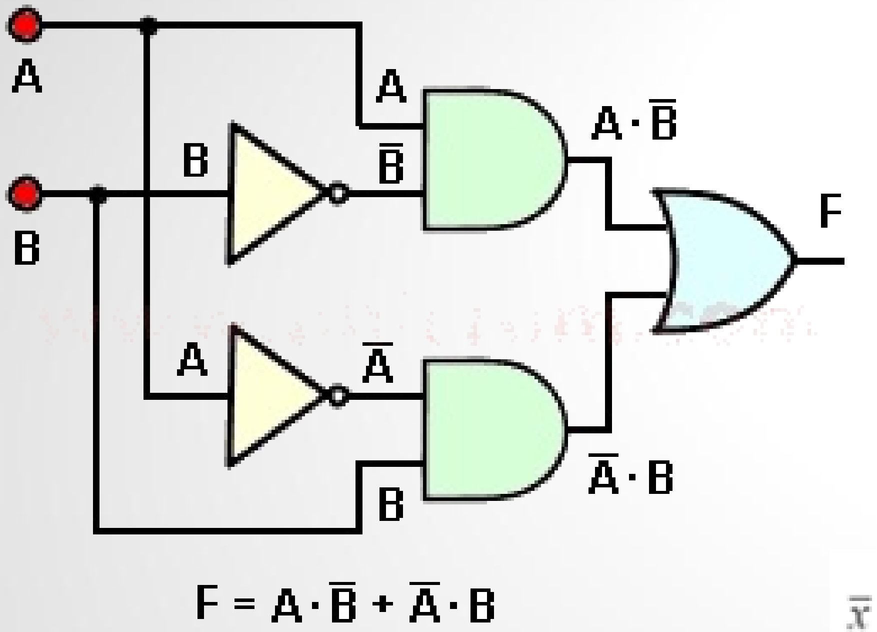
Ejemplos

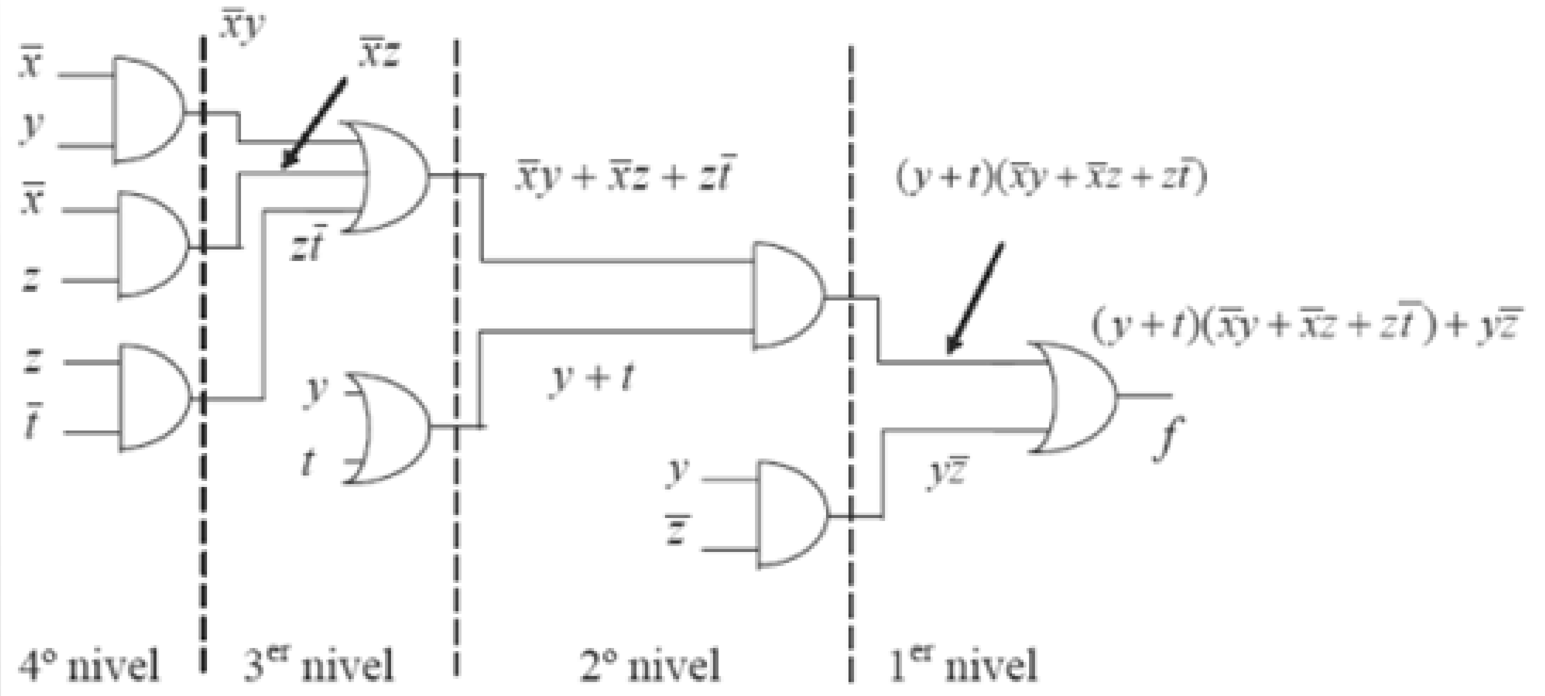


C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

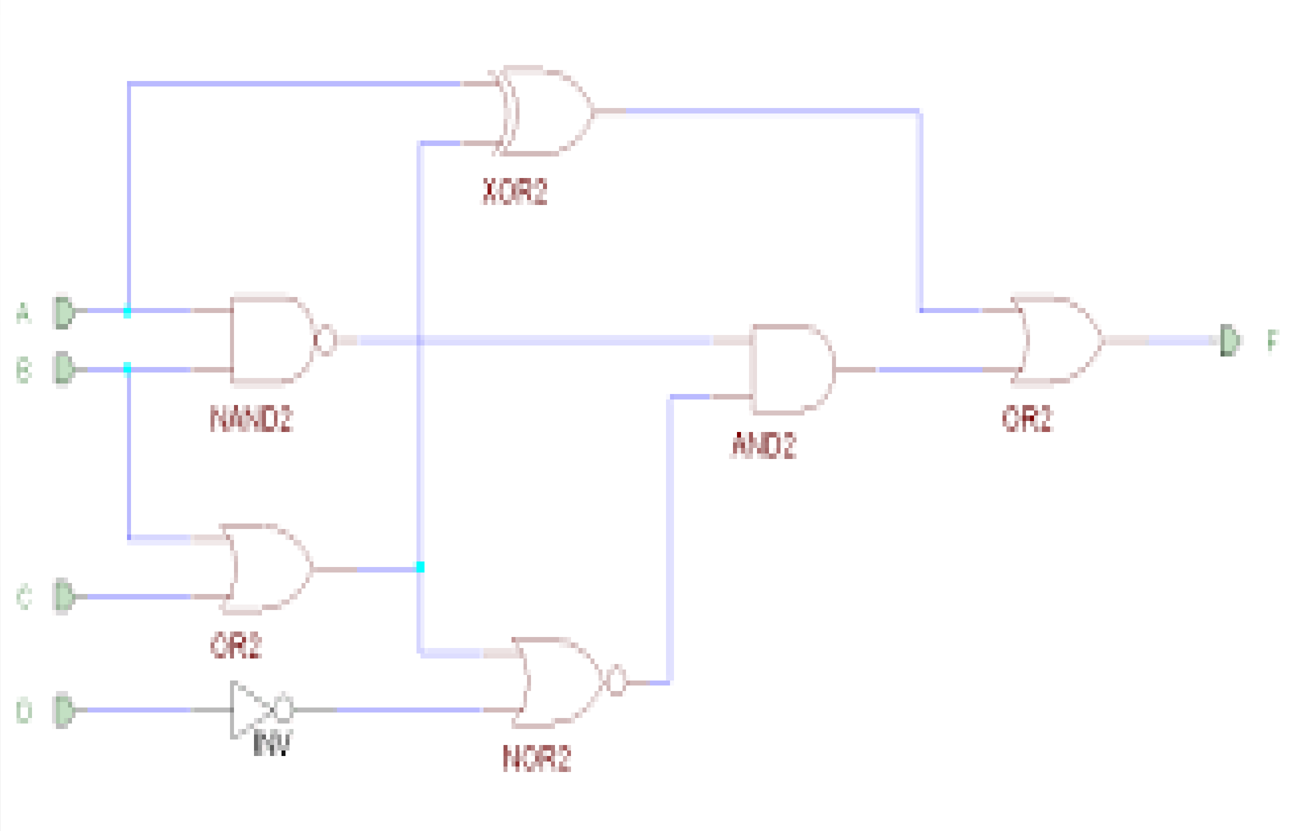


Otros ejemplos





Ejemplo final



REALIZAR LOS EJERCICIOS DE CLASE