En el algebra de boole existen básicamente 3 operaciones básicas: NOT, AND Y OR.

Tablas de verdad

Una tabla de verdad es un medio para describir la manera en que la salida de un circuito lógico depende de los niveles lógicos que haya en la entrada del circuito.

La tabla enumera todas las combinaciones posibles de niveles lógicos que se encuentren en las entradas *A* y *B* con su nivel de salida correspondiente x.

Para la construcción de la tabla se asignan únicamente los siguientes valores:

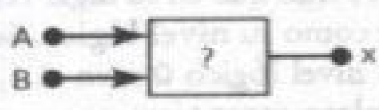
0 LÓGICO	1 LÓGICO
Falso	Verdadero
Desactivado	Activado
Bajo	Alto
No	Sí
Interruptor abierto	Interruptor cerrado

Entradas / Salida

A	B	x
0	0	1
0	1	0
1	0	1
1	1	0

A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

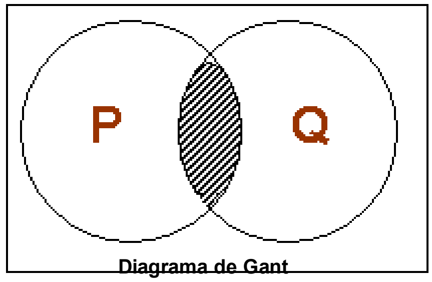
(b)

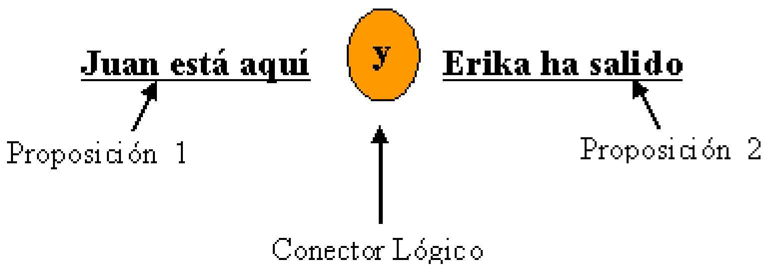


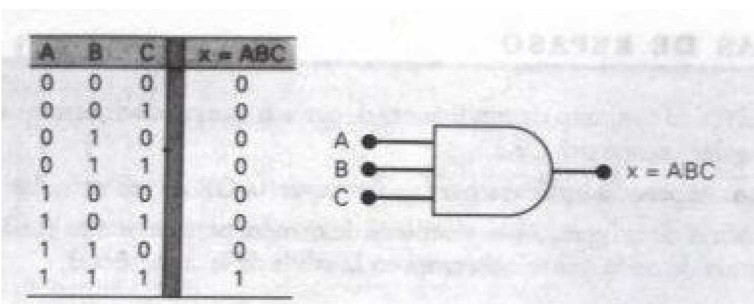
(a)

AND (Conjunción): Las dos o más proposiciones deben ser ciertas. El símbolo de la conjunción es una ^, a veces también se utiliza el símbolo & ó ● que representan la operación lógica “Y”. Por ejemplo, A ● B se lee “A Y B”.

P	Q	P^Q
1	1	1
1	0	0
0	1	0
0	0	0







A	B	C	x = ABC
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

Figura 3-8 Tabla de verdad y símbolo para una compuerta AND de tres entradas.

OR (Disyunción): Al menos una de las proposiciones debe ser cierta. El símbolo de la disyunción es una V, a veces también se utiliza el símbolo | Ó + que representar la operación lógica “O”, A + B se lee “A O B”.

P	Q	PvQ
1	1	1
1	0	1
0	1	1
0	0	0

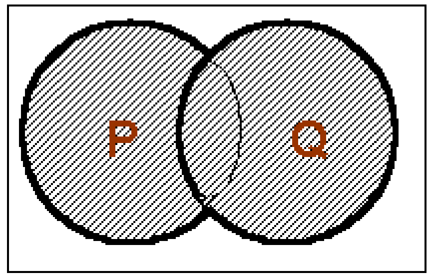
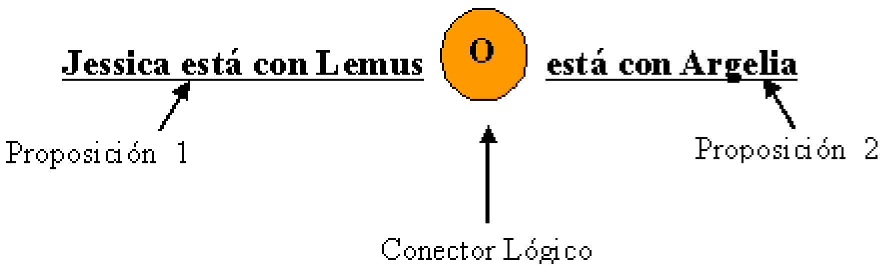
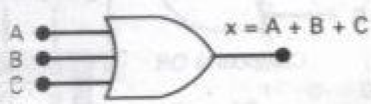


Diagrama de Gant





A	B	C	x = A + B + C
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

Figura 3-3 Símbolo y tabla de verdad para una compuerta OR de tres entradas.

NOT (Negación): Si una proposición es cierta la negada es falsa. Si una proposición es falsa la negada es cierta. El símbolo de la negación es ¬ ó $\overline{A}$

La operación NOT difiere de las operaciones OR y AND en que ésta puede efectuarse con una sola variable de entrada. Por ejemplo, si la variable *A* se somete a la operación NOT, el resultado x se puede expresar como

$$x = \overline{A}$$

donde la barra sobrepuesta representa la operación NOT.

P	¬P
1	0
0	1

$\overline{1} = 0$ ya que NOT 1 es 0

$\overline{0} = 1$ ya que NOT 0 es 1

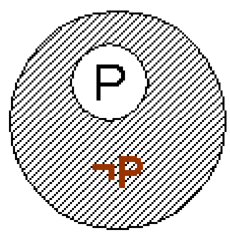
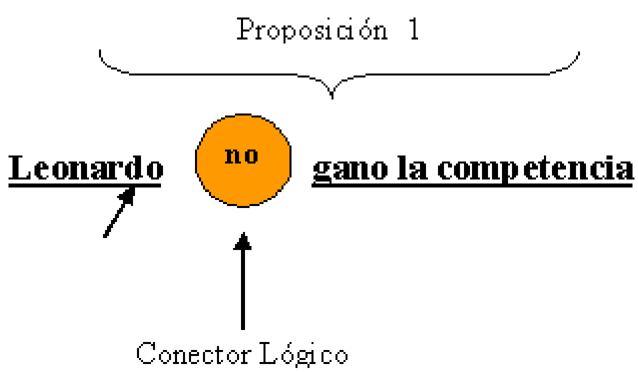


Diagrama de Gant



A	$x = \overline{A}$
0	1
1	0

(a)

(b)

Repaso

¿Con que valores de entrada obtendremos una salida de 1 en la compuerta AND?

¿Con que valores de entrada obtendremos una salida de 0 en la compuerta AND?

¿Con que valores de entrada obtendremos una salida de 1 en la compuerta OR?

¿Con que valores de entrada obtendremos una salida de 0 en la compuerta OR?

Completa las siguientes tablas de verdad

OR	*AND*	*NOT*
$0 + 0 =$	$0 \cdot 0 =$	$\overline{0} =$
$0 + 1 =$	$0 \cdot 1 =$	$\overline{1} =$
$1 + 0 =$	$1 \cdot 0 =$	
$1 + 1 =$	$1 \cdot 1 =$	

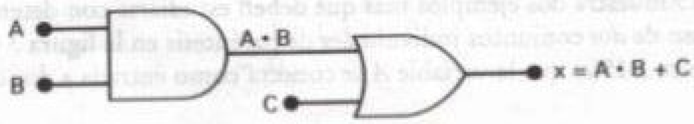
DESCRIPCIÓN ALGEBRAICA DE CIRCUITOS LÓGICOS

Cualquier circuito lógico, sin importar que tan complejo sea, puede describirse completamente mediante las operaciones que se definieron anteriormente, ya que los circuitos de las compuertas OR, AND y NOT son los elementos básicos de los sistemas digitales.

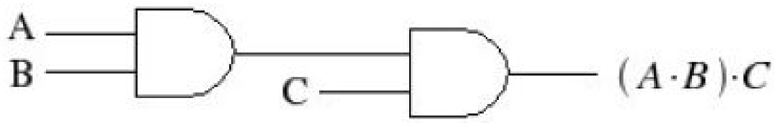
La expresión para la salida de la compuerta AND se escribe $A \cdot B$. Esta salida AND se conecta como entrada a la compuerta OR junto con C, otra entrada. La compuerta OR opera con sus entradas de forma tal que su salida sea la suma OR de las entradas. Así, podemos expresar la salida OR como $x = A \cdot B + C$. (Esta expresión final podría escribirse mejor como $x = C + A \cdot B$, ya que no importa qué término de la suma OR se escriba primero).

En ocasiones, puede existir confusión con respecto de cuál operación se efectúa primero. La expresión $A \cdot B + C$ se puede interpretar de dos formas distintas: (1) $A \cdot B$ se opera con OR con C, o bien (2) A se opera con AND con el término $B + C$. Para evitar esta confusión, se entenderá que si una expresión contiene las operaciones AND y OR, las operaciones AND

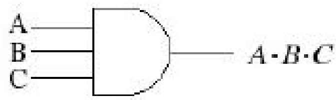
Figura 3-12 Circuito lógico con su expresión booleana.



- Las compuertas pueden tener más de una o dos entradas. Por ejemplo la ecuación de conmutación $F(A, B, C) = A \cdot B \cdot C$ puede ser representada por:



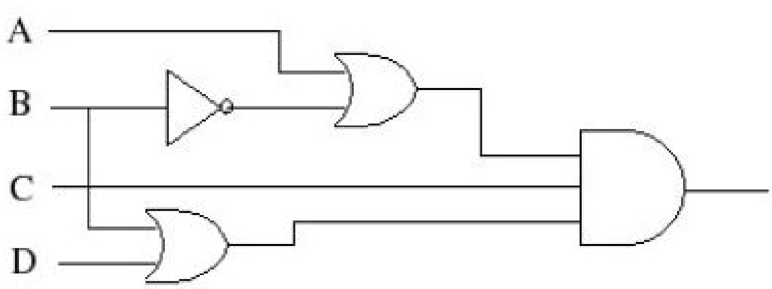
O bien por:



Ejemplo de compuertas

Representar la siguiente ecuación mediante compuertas lógicas.

$$F(A, B, C, D) = (B + D) \cdot (A + \overline{B}) \cdot C$$



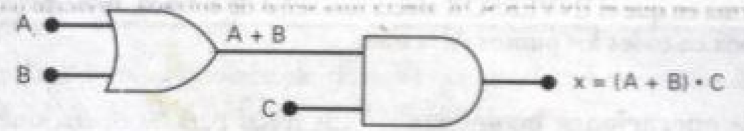


Figura 3-13 Circuito lógico cuya expresión requiere paréntesis.

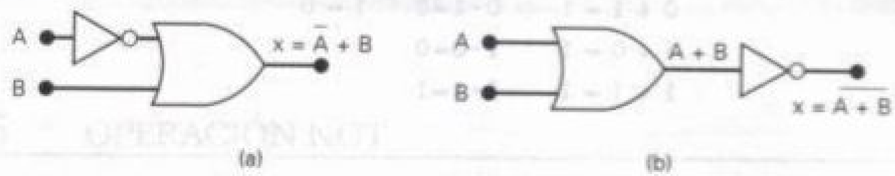


Figura 3-14 Circuitos que utilizan INVERSORES.

se efectúan primero, a menos que haya *paréntesis* en la expresión, en cuyo caso, la operación dentro del paréntesis se realizará primero. Esta es la misma regla que se emplea en el álgebra ordinaria para determinar el orden de las operaciones.

Para ilustrarlo más ampliamente, consideremos el circuito de la figura 3-13. La expresión para la salida de la compuerta OR es simplemente $A + B$. Esta salida sirve como entrada en la compuerta AND junto con otra entrada, C. De este modo expresamos la salida de la compuerta AND como $x = (A + B) \cdot C$. Observe el uso del paréntesis aquí para indicar que A y B se operan con OR *primero*, antes de que su suma OR realice la operación AND con C. Sin los paréntesis esta expresión se interpretaría incorrectamente, ya que $A + B \cdot C$ significa que A se opera con OR con el producto $B \cdot C$.

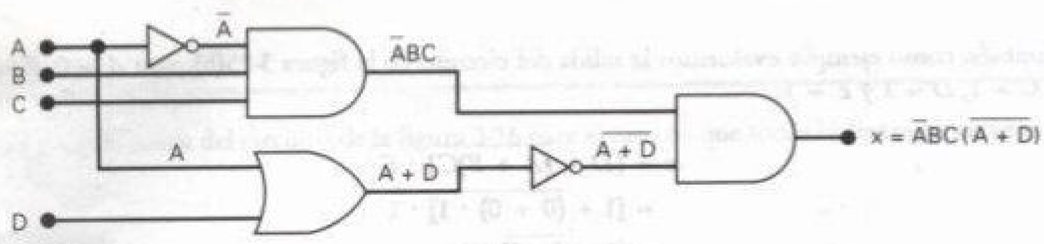
Operador	Gráfo
Parentesis	( )
Exponencial	**
Multi, divide	*, /
Div y mod	Div, mod
Más y menos	+, -

Reglas de prioridad

3-7 EVALUACIÓN DE LAS SALIDAS DE LOS CIRCUITOS LÓGICOS

Una vez que se obtiene la expresión booleana para la salida de un circuito, el nivel lógico de la salida se puede determinar para cualquier valor de las entradas del circuito. Por ejemplo, suponga que deseamos conocer el nivel lógico de la salida x para el circuito de la figura 3-15(a) en el caso donde $A = 0$, $B = 1$, $C = 1$ y $D = 1$. Como sucede en el álgebra ordinaria, el valor de x se puede determinar sustituyendo los valores de las variables en la expresión y efectuando las operaciones que se indican de la siguiente manera:

$$
\begin{aligned}
x &= \overline{A}BC(\overline{A + D}) \\
&= \overline{0} \cdot 1 \cdot 1\,(\overline{0 + 1}) \\
&= 1 \cdot 1 \cdot 1 \cdot (\overline{0 + 1}) \\
&= 1 \cdot 1 \cdot 1 \cdot (\overline{1}) \\
&= 1 \cdot 1 \cdot 1 \cdot 0 \\
&= 0
\end{aligned}
$$



(a)

También como ejemplo evaluemos la salida del circuito de la figura 3-15(b) para $A = 0$, $B = 0$, $C = 1$, $D = 1$ y $E = 1$.

$$
\begin{aligned}
x &= [D + \overline{(A + B)C}] \cdot E \\
&= [1 + \overline{(0 + 0) \cdot 1}] \cdot 1 \\
&= [1 + \overline{0 \cdot 1}] \cdot 1 \\
&= [1 + \overline{0}] \cdot 1 \\
&= [1 + 1] \cdot 1 \\
&= 1 \cdot 1 \\
&= 1
\end{aligned}
$$

En general, siempre deben seguirse los siguientes lineamientos cuando se evalúa una expresión booleana:

1. Primero, realice todas las inversiones de términos simples; es decir $\bar{0} = 1$ o bien $\bar{1} = 0$.
2. Luego efectúe todas las operaciones dentro de los paréntesis.
3. Efectúe una operación AND antes de una OR a menos que los paréntesis indiquen lo contrario.
4. Si una expresión tiene una barra sobre ella, efectúe las operaciones de la expresión primero y luego invierta el resultado.

Para practicar, determine la salida de los dos circuitos de la figura 3-15 en el caso de que todas las entradas sean 1. Las respuestas son $x = 0$ y $x = 1$, respectivamente.

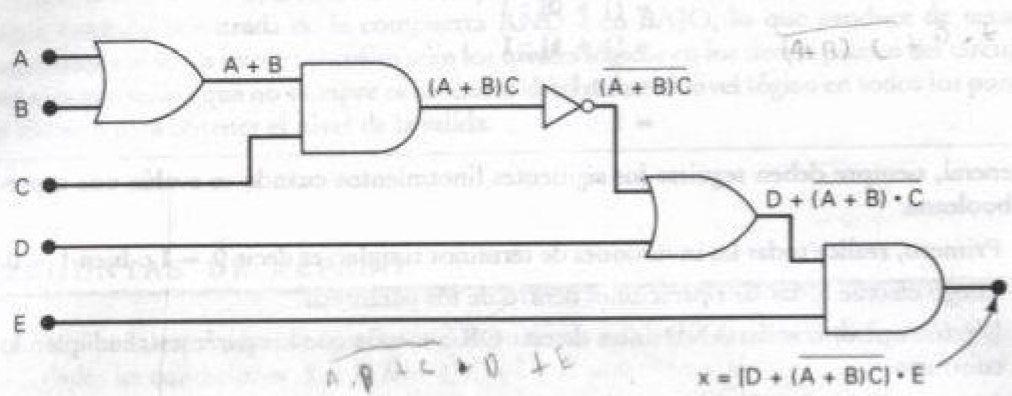


Figura 3-15 Más ejemplos.