UNIVERSIDAD DE SAN BUENAVENTURA CALI

Guía

Álgebra Booleana

Tema III: Compuertas

COMPUERTAS LOGICAS

En los sistemas digitales se utilizan, empaquetados como circuitos integrados, ciertos circuitos electrónicos llamados compuertas lógicas, denominados también dispositivos de estado sólido. En ellos no hay interruptores sino entradas al circuito, en forma de alto voltaje (1) y de bajo voltaje (0), esto es exactamente un bit de información, estos datos son procesados por el circuito para obtener una salida que depende de la combinación de las entradas, es decir un bit de salida, o sea, un 0 ó un 1.

Los circuitos lógicos se construyen a partir de ciertas compuertas lógicas básicas, se comenzará estudiando tres tipos de compuertas: compuerta AND, compuerta OR y compuerta NOT (o inversor).

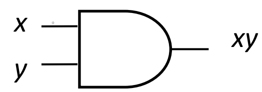
COMPUERTA AND.

Una compuerta AND acepta x y y como datos de entrada, en donde x y y son bits, y produce un dato de salida z, que se denota $z = xy$. El valor de z está determinado por la tabla de valor que aparece a continuación. Es decir, $z = xy = 1$ si, y solo si $x = 1$ y $y = 1$, 0 en otro caso.

Entrada		Salida
x	y	$z = xy$
1	1	1

1	0	0
0	1	0
0	0	0

Una compuerta AND se representa con la figura siguiente:



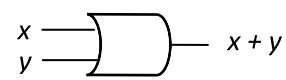
COMPUERTA OR.

Una compuerta OR acepta x y y como datos de entrada, en donde x y y son bits, y produce un dato de salida z, que se denota $z = x + y$. El valor de verdad de z está determinado por la tabla de valor que aparece a continuación; en donde $z = x + y = 1$ si, y solo si $x = 1$ o bien $y = 1$, 0 solo si $x = 0$ y $y = 0$.

Entrada		Salida
x	y	$z = x + y$
1	1	1

1	0	1
0	1	1
0	0	0

Una compuerta OR se representa como se muestra en la figura:



COMPUERTA NOT.

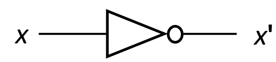
Una compuerta NOT (o inversor) acepta a x como dato de entrada (tiene solamente una entrada), en donde x es un bit, y produce un dato de salida que se denota por x' (o x). El valor de salida x' es el opuesto (complemento a unos) del valor de entrada x; esto es, $x' = 1$ sí $x = 0$ y $x' = 0$ sí $x = 1$.

La tabla de verdad de la compuerta NOT es la siguiente:

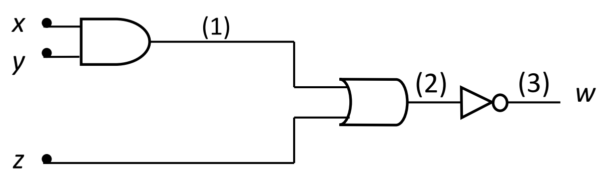
Entrada	Salida
x	x'
1	0

0	1

Una compuerta NOT se representa como aparece en la figura siguiente:



Ejemplo 1. Determine la expresión booleana correspondiente al circuito lógico de la figura y encuentre la tabla lógica de este circuito:



Solución:

Se han enumerado las salidas de cada compuerta para indicar el seguimiento:

1) Se aplica AND a x y y y se obtiene xy.

2) Luego al aplicar OR a (xy) y z se obtiene $(xy) + z$.

3) Entonces el dato de salida se escribe como $[(xy) + z]'$.

La tabla de verdad es la siguiente:

Álgebra Booleana

Entradas				Salida	
x	*y*	*z*	*xy*	(*xy*) + *z*	[(*xy*) + *z*]'
1	1	1	1	1	0
1	1	0	1	1	0
1	0	1	0	1	0
1	0	0	0	0	1
0	1	1	0	1	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	0	0	1

OTRAS COMPUERTAS LOGICAS

Las tres compuertas fundamentales anteriormente presentadas (AND, OR y NOT) son suficientes para describir cualquier ecuación booleana. Toda función lógica puede expresarse con la combinación de estas tres compuertas. Sin embargo, se utilizan otras cuatro compuertas lógicas como simplificación de combinaciones muy usuales de las fundamentales: NAND, NOR, XOR y XNOR.

COMPUERTA NAND.

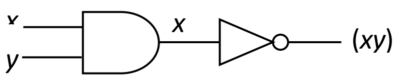
La compuerta NAND (NOT AND), denominada también **operación de Sheffer**, se obtiene cuando a la salida de una compuerta AND se conecta un inversor (NOT); es decir, es la negación de una compuerta AND y acepta x y y como datos de entrada, en donde x y y son bits, y produce un dato de salida z, que se denota como $z = (xy)'$.

La tabla de verdad de la compuerta NAND es la siguiente:

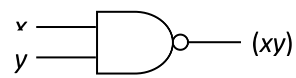
Entrada			Salida
		AND	NAND
x	y	xy	$z = (xy)'$
1	1	1	0
1	0	0	1

0	1	0	1
0	0	0	1

Una compuerta NAND se representa con la figura siguiente:



Compuerta NAND



Compuerta NAND simplificada

COMPUERTA NOR.

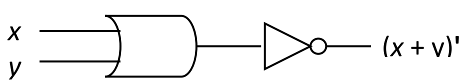
La compuerta NOR (NOT OR), denominada también **operación de Pierce**, se consigue cuando a la salida de una compuerta OR se conecta un inversor (NOT); es decir, es la negación de una compuerta OR y acepta x y y como datos de entrada, en donde x y y son bits, y produce un dato de salida z, que se denota como $z = (x + y)'$.

La tabla de verdad de la compuerta NOR es la siguiente:

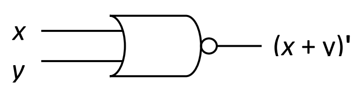
Entrada		Salida
	OR	NOR

x	y	$x + y$	$z = (x + y)'$
1	1	1	0
1	0	1	0
0	1	1	0
0	0	0	1

Una compuerta NOR se representa con la figura siguiente:



Compuerta NOR



Compuerta NOR simplificada

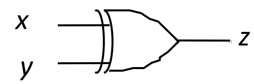
COMPUERTA XOR (OR Exclusiva).

La compuerta XOR, corresponde a la operación lógica disyunción exclusiva. Por lo tanto, la salida es 1 si, y sólo si, exactamente una de las entradas es 1. Realiza la función lógica de salida de la forma $z = xy' + x'y$. Para denotar esta disyunción exclusiva o suma exclusiva se emplea el símbolo $\oplus$; y la función se escribe simplificadamente como $x \oplus y$, así: $f(x, y) = x \oplus y = xy' + x'y$.

La tabla de verdad de la compuerta XOR es la siguiente:

Entrada				Salida XOR
x	y	xy'	$x'y.$	$z = x \oplus y$
1	1	0	0	0
1	0	1	0	1
0	1	0	1	1
0	0	0	0	0

Una compuerta XOR se representa con la figura siguiente:



COMPUERTA XNOR.

La compuerta XNOR, se forma conectando a la salida de la compuerta XOR un inversor. Por lo tanto, la función de salida es $z = f(x, y) = (x \oplus y)'$.

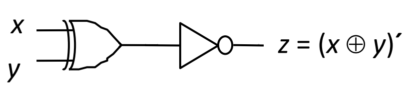
La tabla de verdad de la compuerta XNOR es la siguiente:

Entrada			Salida
		XOR	XNOR
x	y	$x \oplus y$	$z = (x \oplus y)'$
1	1	0	1
1	0	1	0

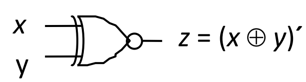
0	1	1	0
0	0	0	1

Observe que la tabla de verdad de la compuerta XNOR es exactamente igual a la tabla de verdad de la equivalencia (o bicondicional); por esta razón, esta compuerta recibe el nombre de **comparador**.

Una compuerta XNOR se representa con la figura siguiente:



Compuerta XNOR



Compuerta XNOR simplificada

Si aplicamos las leyes del álgebra booleana a la expresión $(x \oplus y)'$ se obtiene el siguiente resultado:

$(x \oplus y)' = (xy' + x'y)'$	Definición de la compuerta XOR
$= (xy')'\,(x'y)'$	Leyes de D' Morgan, B9
$= (x' + y)\,(x + y')$	Leyes de D' Morgan, B9
$= (x' + y)\,x + (x' + y)\,y'$	Leyes Distributivas, B2

$= xx' + xy + x'y' + yy'$	Leyes Distributivas, B2
$= 0 + xy + x'y' + 0$	Leyes de Complemento, B4
$= xy + x'y'$	Leyes Modulativas, B3

Esta última expresión es la función booleana que establece la equivalencia entre x y y, por lo tanto: $(x \leftrightarrow y) \equiv (x \oplus y)'$.

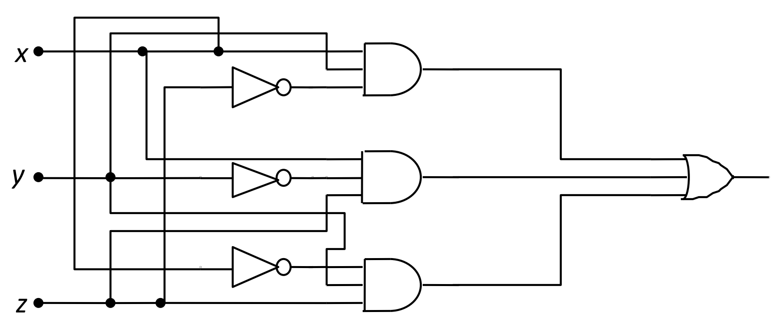
Ejemplo 2. Diseñe un circuito lógico de tres entradas x, y y z cuya salida es 1 cuando, y sólo cuando, exactamente dos de las entradas tienen el valor 1.

Solución: Es conveniente diseñar una tabla con tres entradas y una salida $f(x, y, z)$ evaluando esta con 1 si, y sólo si, dos de las entradas tienen el valor 1, de acuerdo al requerimiento del problema.

Entradas			Salida
x	y	z	$f(x, y, z)$
1	1	1	0
1	1	0	1
1	0	1	1
1	0	0	0

0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

La segunda fila de la tabla que tiene salida 1 origina la combinación xyz', observe que si $x = y = 1$ y $z = 0$ origina la salida 1, de acuerdo a la condición del problema. Así mismo es posible tomar la combinación $xy'z$ para la tercera fila y $x'yz$ para la quinta fila. A continuación se hace la suma de los términos para obtener la expresión booleana: $f(x, y, z) = xyz' + xy'z + x'yz$. Un posible diseño del circuito es el siguiente:



Ejemplo 3. Construir un circuito lógico combinatorio, de acuerdo a la siguiente tabla lógica, usando compuertas AND, OR y NOT.

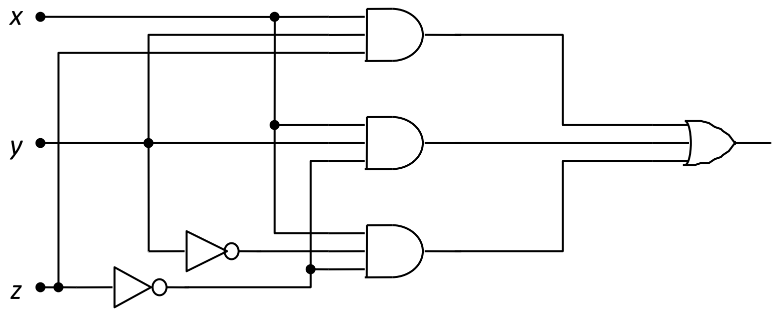
Entradas			Salida
x	*y*	*z*	*f*(*x*, *y*, *z*)
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Solución:

La forma suma de productos (o forma normal disyuntiva) de f es:

$$f(x, y, z) = xyz + xyz' + xy'z'.$$

El circuito lógico correspondiente aparece en la siguiente figura:



El circuito de la figura tiene seis compuertas; es posible diseñar un circuito equivalente que desempeñe la misma función con menos compuertas. El problema de encontrar el "mejor" circuito se llama **problema de minimización**. Para determinar un circuito lógico más simple y equivalente al de la figura se simplificará la expresión booleana que lo representa, como sigue:

$xyz + xyz' + xy'z' = xy\,(z + z') + xy'z'$

$= xy\,(1) + xy'z'$

$= xy\,(1+z') + xy'z'$

$= xy + xyz' + xy'z'$

$= xy + xz'\,(y + y')$

$= xy + xz'\,(1)$

$= xy + xz'$

$= x\,(y + z')$

El circuito lógico correspondiente a la expresión $x\,(y + z')$, solo requiere tres compuertas, como se ilustra en la figura: