Octubre de 2016

Circuitos Logicos

MARIA ALEJANDRA GUIO SAENZ
ALEJANDRO SALAZAR
ALEJANDRO BELTRAN
CAMILO RIVERA

SYGMA

CIRCUITOS LOGICOS

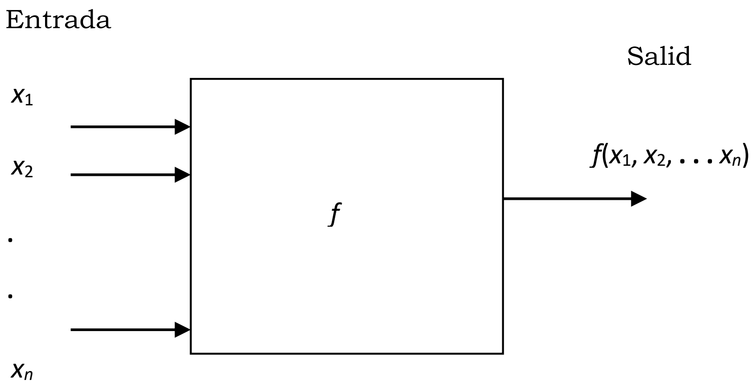
1) FUNCIONES DEL ÁLGEBRA BOOLEANA BINARIA

Sea $B = \{0, 1\}$ y sea $B^n = \{(x_1, x_2, \ldots x_n) \ / \ x_i \in B$ para $i = 1, 2, \ldots, n\}$, entonces se definen funciones f de B^n en B, que se denotan $f : B^n \to B$, como funciones en las que para cualquier n-upla $(x_1, x_2, \ldots x_n) \in B^n$, $f(x_1, x_2, \ldots x_n)$ es 1 o 0. Estas funciones se pueden considerar como funciones de n "variables", donde cada una de ellas toma sólo los valores 1 o 0. Es frecuente enlistarlas en tablas dando cada n-upla posible $(x_1, x_2, \ldots x_n)$ y el valor correspondiente de la función f.

Ejemplo ilustrativo: La tabla que se muestra a continuación representa una función f particular de tres variables; esto es, $f : B^3 \to B$

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	1

El número de combinaciones posibles de ceros y unos de las n-tuplas $(x_1, x_2, \ldots x_n)$ es de 2^n, donde n es el número de variables diferentes. A menudo estas tablas se denominan **tablas de verdad** de f, debido a la analogía con la lógica proposicional.



2) CIRCUITOS DE CONMUTACION (O DE CONMUTADORES)

Un circuito eléctrico de interruptores normalmente contiene alguna fuente de energía (una pila o batería), un dispositivo de salida (por ejemplo una bombillo), y uno o más interruptores o “switches”, todos ellos conectados por alambres. Sólo se considerarán los interruptores que permiten o impiden el paso de la corriente eléctrica que fluye por el circuito, y cuyo funcionamiento es de dos estados cerrado (encendido), on, y abierto (apagado), off.

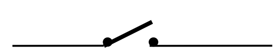
Cuando un interruptor x está en la posición cerrado (on) permite el paso de la corriente, se escribe $x = 1$, y cuando está en la posición abierto (off), no hay paso de corriente, se escribe $x = 0$.

INTERRUPTOR CERRADO (ON)

$x = 1$

INTERRUPTOR ABIERTO (OFF)

$x = 0$



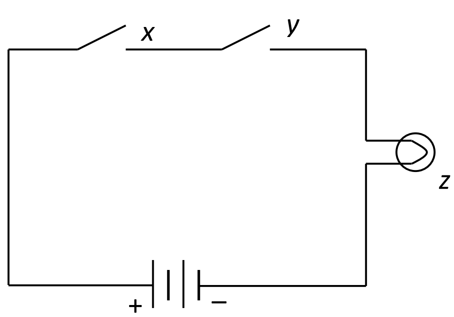
Existen dos formas básicas para interconectar interruptores: conexión en serie y conexión en paralelo.

3) CIRCUITO EN SERIE (CIRCUITO AND).

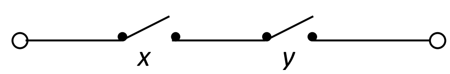
Se caracteriza porque hay paso de corriente a z si, y solo si, los dos interruptores x e y se encuentran cerrados. Es decir, la salida del circuito es 1 si, y solo si $x = 1$ y $y = 1$. Esta configuración se denota por $z = xy$. La tabla de conmutación que presenta las salidas del circuito en serie (AND) es:

x	y	$z = x\,y$
1	1	1
1	0	0
0	1	0
0	0	0

Puesto que estamos interesados solamente en la posición de los interruptores, entonces usaremos la representación simplificada del circuito:



Circuito en serie



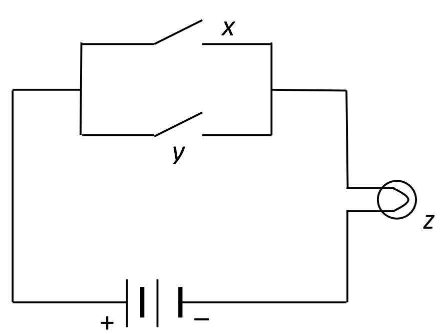
Circuito simplificado

4) CIRCUITO EN PARALELO (CIRCUITO OR).

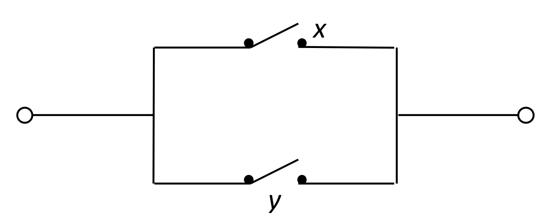
Se caracteriza porque hay paso de corriente a z si, y solo si uno de los interruptores del circuito, x o bien y, o ambos está(n) cerrado(s). Es decir, la salida del circuito es 1 si, y solo si $x = 1$ o $y = 1$, o ambos $x = y = 1$. Esta configuración se denota por $z = x + y$. La tabla de conmutación del circuito en paralelo es:

x	y	$z = x + y$
1	1	1
1	0	1
0	1	1
0	0	0

Al igual que en el circuito en serie, usaremos la representación simplificada del circuito:



Circuito en paralelo

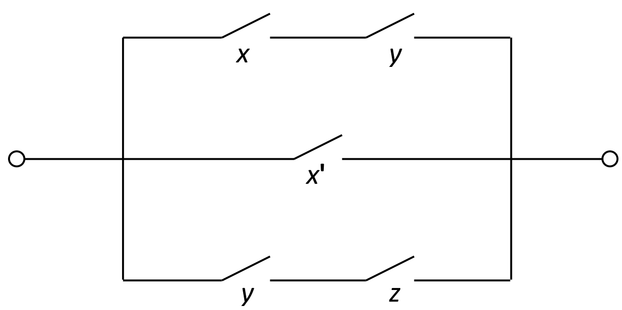


Circuito simplificado

Si en un circuito hay varios interruptores que coinciden en su estado (1 o 0) los designaremos con la misma variable. El símbolo x' (o $\bar{x}$) representa un interruptor que presenta el estado opuesto al que presenta un interruptor denotado por x, se denomina el **inversor de x**. En un circuito junto con cualquier interruptor x podemos incluir cualquier interruptor x' que está abierto cuando x está cerrado, y está cerrado cuando x está abierto:

x	x'
1	0
0	1

Ejemplo 1: Determine una expresión de Boole para el siguiente circuito de interruptores y elabore su correspondiente tabla conmutadora



Solución: Se procede por niveles de arriba a abajo: 1) xy, 2) x', 3) yz.

Al final obtenemos la expresión: $xy + x' + yz$ que corresponde a la expresión de Boole del circuito, donde los niveles son de un circuito en paralelo.

La tabla conmutadora correspondiente es:

x	y	z	xy	x'	yz	$xy + x' + yz$
1	1	1	1	0	1	1
1	1	0	1	0	0	1
1	0	1	0	0	0	0
1	0	0	0	0	0	0
0	1	1	0	1	1	1
0	1	0	0	1	0	1
0	0	1	0	1	0	1
0	0	0	0	1	0	1

5) COMPUERTAS LOGICAS

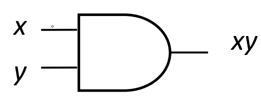
En los sistemas digitales se utilizan, empaquetados como circuitos integrados, ciertos circuitos electrónicos llamados compuertas lógicas, denominados también dispositivos de estado sólido. En ellos no hay interruptores sino entradas al circuito, en forma de alto voltaje (1) y de bajo voltaje (0), esto 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.

6) 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:

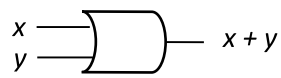


7) COMPUERTA OR.

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:



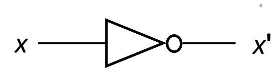
8) COMPUERTA NOT.

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 $\overline{x}$). 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 $x = 1$.

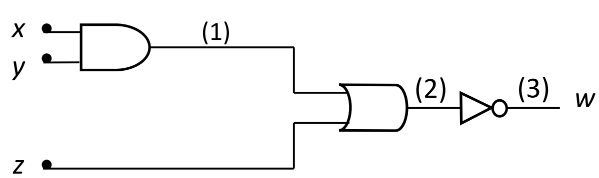
La tabla de verdad de la compuerta NOT es la siguiente:

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:

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

9) OTRAS COMPUERTAS LOGICAS

Se utilizan otras cuatro compuertas lógicas como simplificación de combinaciones muy usuales de las fundamentales: NAND, NOR, XOR y XNOR.

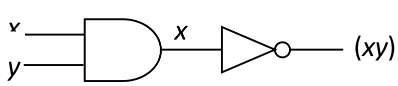
10)COMPUERTA NAND.

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 $z = (xy)'$.

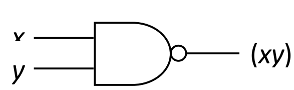
La tabla de verdad de la compuerta NAND es la siguiente:

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

1Una compuerta NAND se representa con la figura siguiente:



Compuerta NAND



Compuerta NAND simplificada

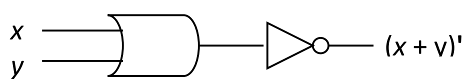
11) COMPUERTA NOR.

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 $z = (x + y)'$.

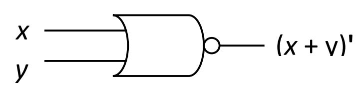
La tabla de verdad de la compuerta NOR es la siguiente:

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

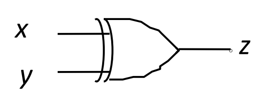
12) COMPUERTA XOR (OR Exclusiva).

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 ⊕; 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 $z = f(x, y) = (x \oplus y)'$.

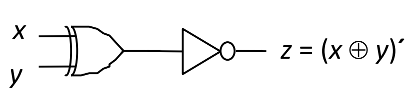
La tabla de verdad de la compuerta XNOR e

s la siguiente:

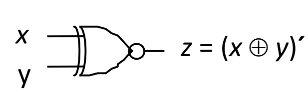
Entrada			Salida
		XOR	XNOR
x	y	$x \oplus y$	$z = (x \oplus y)'$
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	1

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: $(x \leftrightarrow y) \equiv (x \oplus y)'$.