Universidad Nacional de la Matanza

Departamento:
Ingeniería e Investigaciones Tecnológicas

Cátedra:

Fundamentos de TIC's

(Tecnologías de la Información y la Comunicación)

e-mail: fundamentos_tics@unlam.edu.ar

JEFE DE CÁTEDRA:
Mg. Daniel A. Giulianelli

INTRODUCCIÓN A ESTRUCTURAS LÓGICAS

COLABORACIÓN:
DOCENTES DE LA CÁTEDRA

CICLO LECTIVO:

2009

Universidad Nacional de la Matanza

DEPARTAMENTO DE INGENIERÍA E INVESTIGACIONES TECNOLÓGICAS

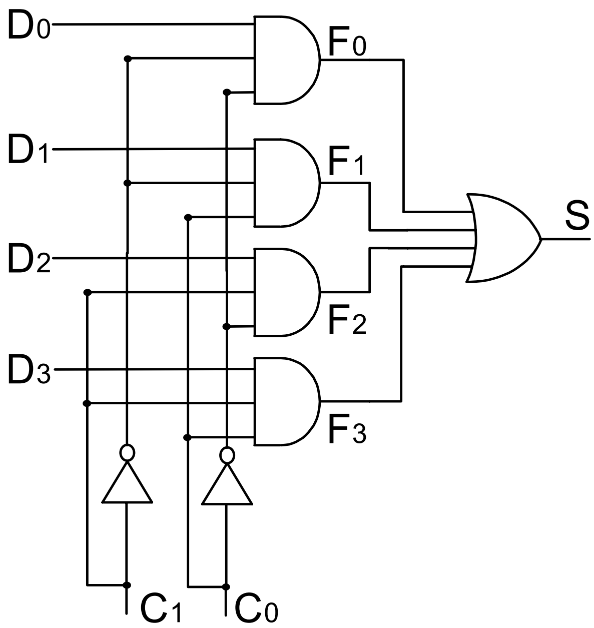
INGENIERÍA EN INFORMÁTICA

FUNDAMENTOS DE TIC´s

UNIDAD 2 parte A

Introducción a circuitos lógicos . . .

- Introducción al álgebra conmutacional.
- Demostración de algunos teoremas del álgebra de Boole.
- Implementación de funciones simples.
- Introducción a los circuitos secuenciales.



C_1	C_0	S	Página
0	0	D_0	3
0	1	D_1	13
1	0	D_2	21
1	1	D_3	29

Colaboración: Ing. Guillermo P. Benéitez

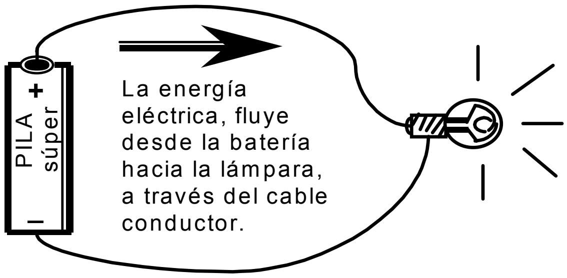
INTRODUCCIÓN AL ÁLGEBRA CONMUTACIONAL

Conmutación y álgebra conmutacional

Introducción:

Algunos esquemas mecánicos, circuitos eléctricos, electrónicos y otros sistemas físicos, nos permiten resolver problemas lógicos.

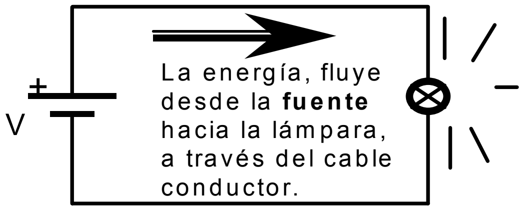
Un circuito eléctrico simple, podrá ayudarnos a entender el concepto.



Representación física

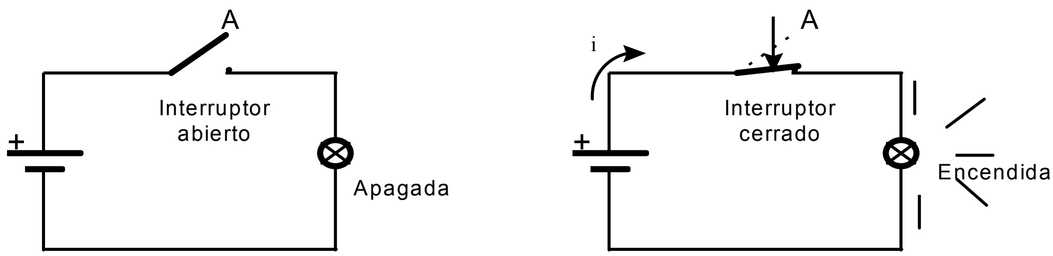
Hemos representado a los conductores eléctricos (cables) mediante trazos curvos.

Este sistema puede simbolizarse por medio de un esquema como el siguiente:



Esquema eléctrico

Para encender o apagar la lámpara, suele emplearse una lámina metálica con un pequeño pliegue que deja al circuito normalmente desconectado[1] en un extremo. Al ejercer presión sobre ella, el extremo abierto se pone en contacto[2] y la corriente eléctrica puede circular. Al dejar de ejercer presión, la lámina metálica se levanta debido a su elasticidad y la corriente se interrumpe nuevamente. A continuación se representan estas situaciones:



Posiciones posibles de un interruptor

El proceso de permitir o no el paso de la corriente eléctrica, por ejemplo cambiando de una posición a otra el interruptor, se denomina **conmutación**[3].

Analicemos la figura anterior:

a) En el circuito de la izquierda, “NO” ejercemos presión sobre el interruptor "A", por lo tanto, la lámpara “NO” se enciende; y
b) En el circuito de la derecha, “SI” ejercemos presión sobre el interruptor "A" y la lámpara “SI” se enciende.

---
[1] Se denomina interruptor de corriente “Normal Abierto”, o sintéticamente “NA”.
[2] El circuito se cierra.
[3] Obsérvese el comportamiento discreto (digital) del interruptor: conduce o no la corriente eléctrica. Idealmente entre esos estados no hay otros estados intermedios.

Que la lámpara encienda o no, depende de la posición del interruptor “A”. Diremos entonces que el encendido de la lámpara es una función de la variable “A”.

A partir de este análisis podremos construir la siguiente tabla:

Interruptor "A" presionado	Lámpara Encendida
NO	NO
SI	SI

O más sucintamente:

A	E
NO	NO
SI	SI

También podemos convenir en hacer las siguientes relaciones:

a) NO ejercer presión sobre el interruptor, lo representamos con "0" (cero) y SI ejercer presión sobre el interruptor lo representamos con "1" (uno);

b) lámpara NO encendida: "0" (cero) y lámpara SI encendida "1".

Así, la tabla anterior se verá:

A	E
0	0
1	1

Esta convención[4] se conoce como “Lógica Positiva”.

“... Una teoría matemática (o lógica, en general) es un conjunto de proposiciones que se siguen según un esquema de *deducción lógica*.

Se entiende por proposición, una expresión de la cual tenga sentido inequívoco decir si es verdadera o falsa.

La determinación del criterio de verdad de las proposiciones es aveces una cuestión extralógica, que pertenece a otro campo del conocimiento.

Por ejemplo: “San Martín murió en Francia”, es una proposición cuya verdad pertenece a la Historia. En cambio, decir: “El enfermo morirá o no morirá” es enunciar una proposición verdadera por su misma estructura lógica[5].

En todo proceso de deducción lógica, o *razonamiento*, las proposiciones de partida forman lo que se llama la *hipótesis*, y la conclusión a la que se llega es la *tesis*.

En un razonamiento válido, la tesis *se deduce* o es una *consecuencia lógica* de la hipótesis; también se dice que la hipótesis *implica* la tesis. ...” [i].

Se simboliza “$P \Rightarrow Q$”, para denotar el concepto fundamental de que cuando el razonamiento es válido, la hipótesis P implica la tesis Q.

Evidentemente, **razonando correctamente** es ilógico partir de una hipótesis “P” **verdadera** y llegar a una tesis “Q” **falsa**. Cualquier otra posibilidad es válida.

Entonces, cuando logremos demostrar la validez de una hipótesis podremos confiar en la verdad de la tesis obtenida por implicación.

Se pueden emplear tablas para representar las combinaciones posibles de hipótesis verdaderas (V) o falsas (F) y tesis verdaderas o falsas.

Se las denomina “Tablas de Verdad”.

Las características de la implicación se pueden aclarar mediante su tabla de verdad:

P	Q	$P \Rightarrow Q$
F	F	V
F	V	V
V	F	F
V	V	V

Aquí hemos representado[6] las cuatro combinaciones posibles[7] de hipótesis y tesis.

[4] Existen dos formas de relacionar los interruptores o la lámpara con los estados 1 y 0. Uno de los casos es el expuesto, conocido como lógica positiva. Cuando es una relación opuesta a la presente, por ejemplo la lámpara apagada es un “1” y prendida es un “0” se dirá que estamos frente a un sistema que emplea lógica negativa.

[5] Toda proposición que resulte siempre verdadera es una tautología.

[6] Se puede establecer una correspondencia entre “Falso → 0” y “Verdadero → 1”, con lo cual esta tabla de verdad adoptaría la nomenclatura de la tabla expuesta anteriormente.

Algunos ejemplos consolidarán este concepto.

Para la primera combinación: P falsa, Q falsa, implicación ($P \Rightarrow Q$) verdadera.

Supongamos la siguiente hipótesis compuesta falsa, P: a) la suma de dos enteros es siempre igual al cociente de los mismos y b) el cociente de dos enteros es siempre una fracción.

Se observa que razonando bien ($P \Rightarrow Q$), se deduce que "la suma de dos enteros es siempre una fracción". Conclusión Q (tesis) que es falsa.

Ahora veamos la segunda combinación que se indica en la tabla de verdad:

P falsa, Q verdadera, implicación ($P \Rightarrow Q$) verdadera.

Supongamos otra hipótesis compuesta falsa, como ser P: a) la suma de dos enteros es siempre igual al cociente de los mismos y b) el cociente de dos enteros es siempre un entero.

Con un razonamiento válido ($P \Rightarrow Q$), se deduce que "la suma de dos enteros es siempre un entero". Conclusión Q (tesis) verdadera. Es importante resaltar que una conclusión acertada donde se ha empleado un razonamiento correcto, no implican necesariamente que la hipótesis sea correcta (como en este y en muchos otros casos).

La tercera combinación es distintiva de la implicación:

P verdadera, Q falsa, implicación ($P \Rightarrow Q$) falsa [8].

Para una hipótesis compuesta verdadera, P por ejemplo: a) la tierra es un astro, y b) las estrellas son astros.

Decir que "la tierra es una estrella", no solo es una conclusión Q (tesis) falsa, sino que no fue una deducción lógica de la hipótesis.

La última combinación posible:

P verdadera, Q verdadera, implicación ($P \Rightarrow Q$) verdadera.

Para una hipótesis compuesta verdadera, como P: a) la tierra es un planeta, y b) todos los planetas son astros.

Se deduce que la tierra es un astro.

Se sugiere buscar otros ejemplos de proposiciones que verifiquen la implicación.

Tomado del álgebra proposicional, se puede generalizar el nombre "Tablas de Verdad" y emplearlo en todas las tablas de esta obra [9].

Cuando dos proposiciones tienen la misma tabla de verdad, estas son equivalentes.

Las proposiciones pueden ser simples[10] como: si **llueve lo necesario** tendremos una buena cosecha, o compuestas: si el sábado **llueve y el auto funciona** iremos al cine.

---

[7] Formadas por las variaciones con repetición de dos objetos V y F con orden igual al número de proposiciones.
[8] Hemos advertido que es ilógica, por eso esta implicación resulta falsa.
[9] No solo en esta, sino en gran número de obras sobre el tema de sistemas y circuitos digitales o electrónica digital se las llama de ese modo "tablas de verdad".
[10] Estas definiciones aclaran lo expuesto en la nota 7.

Álgebra conmutacional:

Existen muchos circuitos distintos con interruptores, algunos de los cuales se presentan muy frecuentemente.

En última instancia, el circuito dependerá de la necesidad concreta.

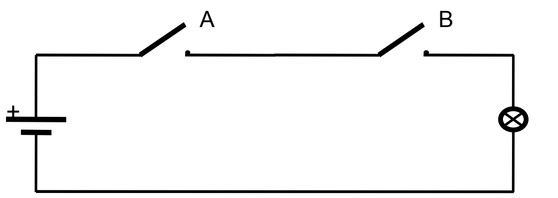
Por ejemplo para que un ascensor arranque, ambas puertas [11] deben estar cerradas.

También podemos decir que cualquiera de las puertas abiertas deberá impedir el funcionamiento del motor del ascensor.

Colocando un interruptor al final del recorrido de cada puerta, los interruptores se abrirán cuando la puerta no esté bien cerrada.

El siguiente circuito permite resolver la lógica planteada.

-1) Si colocamos dos interruptores sobre el mismo conductor, es decir, en serie:



Interruptores en serie

Para que la lámpara encienda (o el ascensor arranque), el interruptor "A" **Y** el "B" deben estar presionados al mismo tiempo. Por lo tanto podrá realizarse la siguiente tabla de verdad:

A	B	E
0	0	0
0	1	0
1	0	0
1	1	1

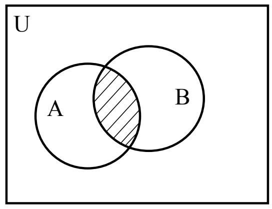
Independientemente del mecanismo que se emplee, existen muchas situaciones en las cuales, para que un evento esperado se cumpla, deben cumplirse simultáneamente determinadas condiciones.

Como ser, se proponen las condiciones que deben cumplirse simultáneamente para aprobar una materia: aprobación de parciales **Y** cumplir con una asistencia del 75%. En álgebra proposicional recibe el nombre de “conjunción”.

Otro caso corresponde a la intersección en la teoría de conjuntos.

La intersección es otro conjunto, el cual está formado por los elementos que pertenecen a uno **Y** al otro simultáneamente.

En la figura adjunta, está representada la intersección entre A y B mediante un diagrama de Euler. Simbólicamente $A \cap B$.



Por estas razones los circuitos que analizamos son llamados **Circuitos Lógicos**.

Esta situación no se presenta solamente en los circuitos eléctricos con interruptores de accionamiento manual, sino también en otras tecnologías.

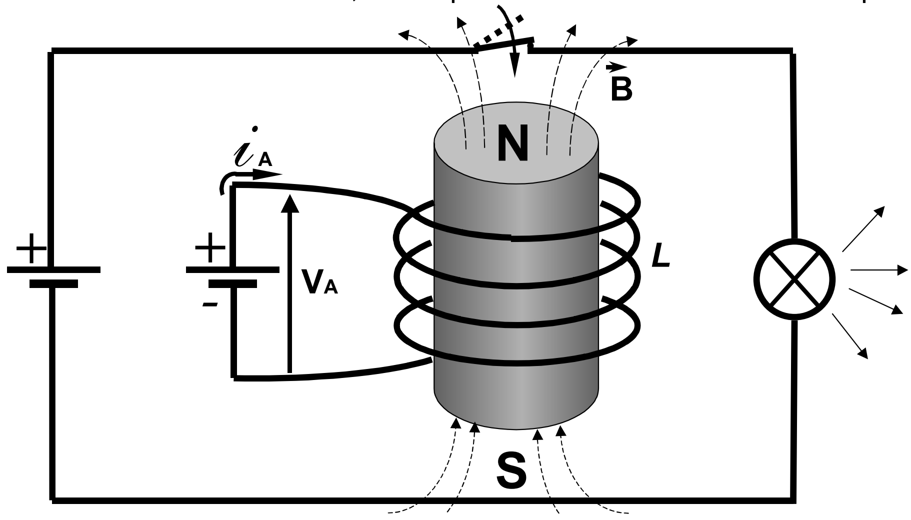
El mismo resultado se podría lograr utilizando un sistema electromagnético.

En un sistema electromagnético, el interruptor podría estar impulsado por la fuerza de atracción magnética, en lugar de hacerlo por fuerza mecánica (manualmente).

La lámina elástica que constituye el interruptor (de metal ferroso) es atraída por el campo magnético (**B**). Este está generado por la corriente “i_A”, que atraviesa una bobina “***L***” con núcleo de hierro – Este es el principio de funcionamiento de relevadores (Relay) y contactores.-

[11] Una puerta se encuentra en la cabina del ascensor y otra en el piso del nivel en el que está detenido.

Se muestra a continuación, una representación ilustrativa de lo expuesto.



En este caso, la presión sobre el interruptor "A", estaría originada por la corriente "i_A".

Corriente "i_A"	Lámpara Encendida
NO	NO
SI	SI

O su correspondiente:

A	E
0	0
1	1

La tecnología cambia, pero la estructura lógica es análoga a la anterior.

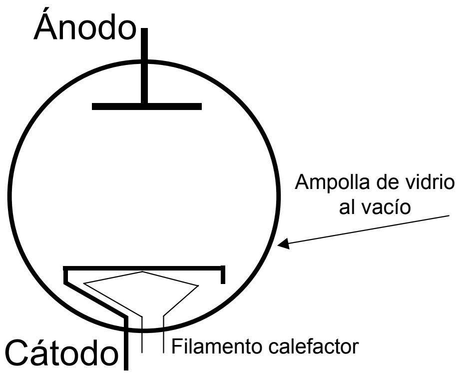
Dispositivos electrónicos

Los dispositivos electrónicos realizan trabajos, controlando corrientes y potenciales eléctricos de circuitos, emisiones de luz y otros efectos, basándose en el control de las cargas eléctricas que constituyen la materia (como electrones).

Dispositivos electrónicos de vacío

Los dispositivos electrónicos de vacío fueron uno de los primeros elementos capaces de controlar electrónicamente el funcionamiento de un sistema.

Se basan en la capacidad de emisión (liberación) de electrones de la superficie de un metal, al elevar su temperatura (emisión térmica) y enfrentarlo a un fuerte potencial positivo (los electrones tienen carga negativa y son atraídos por el potencial opuesto). Este fenómeno se produce dentro de una ampolla de vidrio al vacío (válvulas de vacío).

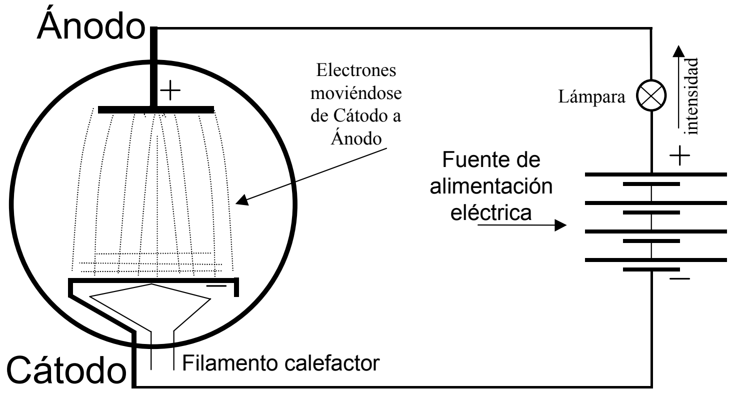


El filamento calefactor conectado a una fuente de energía adecuada, eleva la temperatura (como lo haría una estufa eléctrica), de una placa metálica denominada Cátodo.

Los electrones de la superficie del metal incrementan su velocidad (ya que aumentaron su energía debido a la fuente de calor) y se alejan del Cátodo.

Recordemos que algunos de los electrones que constituyen la materia (átomos, núcleo y electrones), se mueven con cierta libertad por la superficie de los metales.

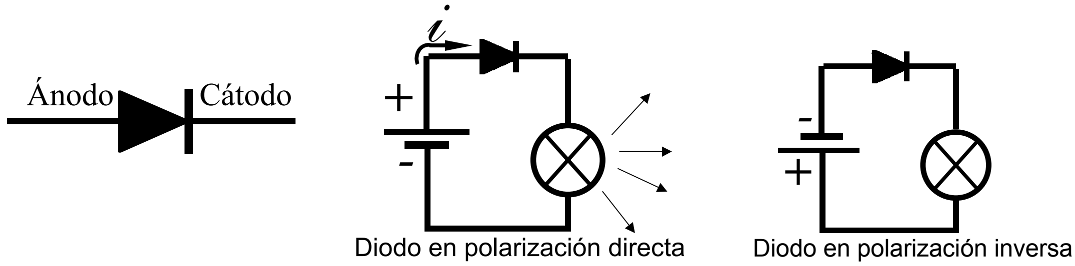
Un elevado potencial eléctrico: negativo en el Cátodo y positivo en otra placa metálica enfrentada (que denominamos Ánodo), permiten alejar los electrones del Cátodo y atraerlos al Ánodo, provocando un movimiento de cargas, es decir, una intensidad de corriente eléctrica en el circuito.



Si en cambio invirtiéramos el potencial eléctrico de la fuente de alimentación (negativo en el Ánodo y positivo en el Cátodo), los electrones (negativos) se verían repelidos por el potencial negativo del Ánodo y no producirían corriente en el circuito.

La válvula de vacío construida de este modo se llama “Diodo”, ó “Diodo valvular”, por su característica de permitir el paso de la corriente en un solo sentido, resultaron de mucha utilidad.

Simbólicamente:



Existen una gran variedad de válvulas de vacío para distintas aplicaciones: triodos, tetrodos, pentodos, tubos de televisión y monitores de computadoras, microondas, amplificación, etc. .

Dispositivos electrónicos de estado sólido

Muchos dispositivos electrónicos, basan su funcionamiento en materiales semiconductores.

Llamamos conductoras, a las sustancias que permiten el paso de la corriente eléctrica (cobre, hierro, etc.) y aisladores a aquellos que la impiden (amianto, el carbono en forma de cristal: diamante, etc.).

Los elementos como el “silicio” y el “germanio” en forma de cristales, se comportan como aisladores cuando están a bajas temperaturas y como conductores a medida que la temperatura aumenta y, por esto, se los denominó semiconductores.

Existen varias formas de lograr la conducción con semiconductores (aparte de aplicarles calor). Por ejemplo, se los puede hacer conducir aplicándoles luz sobre su superficie (fotoconductividad).

Otra manera de controlar las características de la conducción de un semiconductor es reemplazar un pequeña parte del mismo por otra sustancia, por ejemplo reemplazar algunos átomos del cristal de silicio por átomos de aluminio.

Incorporando impurezas de diferentes tipos, se logran construir cristales basados en semiconductores tales que, se comportan como diodos, transistores (capaces de controlar la corriente de un circuito) y circuitos integrados (que contienen complejas estructuras, equivalentes a grandes cantidades de circuitos con diodos y transistores).

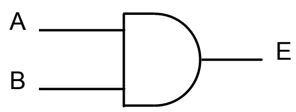
Los dispositivos basados en semiconductores (electrónicos de estado sólido) reemplazan a las válvulas en muchas aplicaciones, sin necesidad de emplear filamentos calefactores internos (ahorrando energía), ocupando menos espacio, disminuyendo las fallas y reduciendo costos, entre otras ventajas.

Existen por lo tanto una gran cantidad de tecnologías para realizar los circuitos lógicos. Algunas emplean válvulas, otras transistores o circuitos integrados, por nombrar solo algunas. En consecuencia, convendrá utilizar una representación que permita esquematizarla, independientemente de la tecnología.

Compuertas lógicas:

Optamos por emplear símbolos conocidos como **Compuertas Lógicas**, que permiten representar operadores lógicos y cuya construcción física podrá realizarse con los dispositivos utilizados en las distintas generaciones de computadores u otros que puedan surgir en el futuro.

Para el caso que nos ocupa, la representación es:



A	B	**E**
0	0	0
0	1	0
1	0	0
1	1	1

Compuerta **Y** (And)

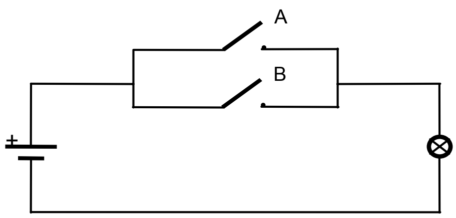
La tabla de verdad se interpreta: para que la salida “E” sea 1, “A **Y** B” deben ser 1.

Esta tabla coincide con el "producto lógico" del álgebra de Boole, donde "**Y**" se reemplaza por el producto lógico representado con un punto "**.**". De esta forma, su tabla de verdad podemos indicarla:

A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

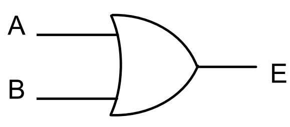
Otra situación diferente, se presenta cuando por ejemplo una alarma debe ser activada desde cualquiera de dos lugares distintos. En este caso presionando uno cualquiera o ambos interruptores debe accionarse la alarma.

-2) Ahora, disponemos dos interruptores en paralelo, es decir, dos líneas (conductores eléctricos) paralelas, cada una de las cuales contiene un interruptor. Esto permite que la lámpara encienda cuando el interruptor “A” **O** el “B” están presionados (indistintamente, incluso pueden estar presionados simultáneamente).



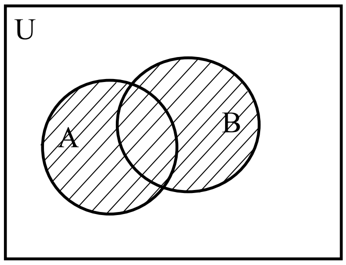
Interruptores en paralelo

Su representación y tabla de verdad se muestran a continuación:



A	B	**E**
0	0	0
0	1	1
1	0	1
1	1	1

Compuerta O (Or)



En teoría de conjuntos está relacionado con la unión de conjuntos.

La unión es otro conjunto, el cual está formado por los elementos que pertenecen a uno **O** al otro.

En la figura adjunta, está representada la unión de A con B mediante un diagrama de Euler. Simbólicamente A $\cup$ B.

La relación con el álgebra proposicional es a través de la disyunción.

En castellano, la letra “o” presenta ambigüedad respecto del significado lógico.

Si decimos: “esto es blanco o negro”, es uno u otro pero no simultáneamente, se excluye la condición de simultaneidad (disyuntiva).

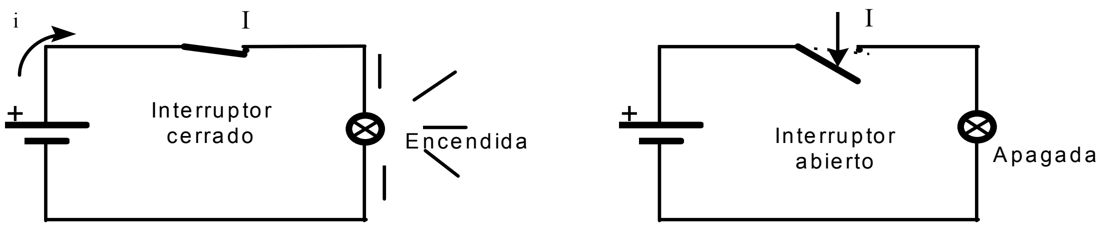
En cambio: “espero tu respuesta, por mail o por teléfono”, implícitamente se acepta la situación de simultaneidad, cualquiera de los medios (mail o teléfono) o los dos al mismo tiempo serán validos. Ahora la “o” es inclusiva[12].

Este último es el sentido en que lo empleamos en este caso y que corresponde a la unión de conjuntos.

La operación lógica O, en álgebra de Boole, equivale a la suma lógica y se representa con el símbolo "+". Así, A **O** B se expresará: A + B (suma lógica de A con B).

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

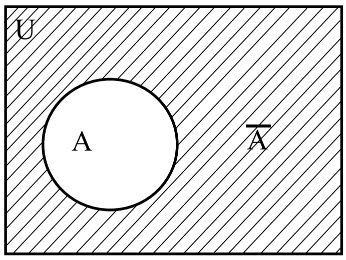
-3) Existe aún un elemento importante en este análisis.



Posiciones posibles de un interruptor inverso al anterior

En el circuito anterior, al ejercer presión sobre el interruptor "I", la lámpara se apaga. Este es un mecanismo inverso al expuesto hasta el momento.

Obviamente existe un elemento equivalente en teoría de conjuntos, el complemento. El complemento de un conjunto es otro conjunto cuyos elementos no pertenecen al conjunto original.



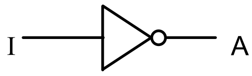
En lógica proposicional, corresponde a "la negación".

[12] En latín, “out - out” es excluyente y “vel” es inclusivo. Un término más preciso sería “yuxtaposición” (en lugar de disyunción), pero no se usa.

Por otra parte, se entiende que este interruptor "I" presenta un funcionamiento opuesto al del interruptor "A", lo cual se indica diciendo: $I = \overline{A}$.

En el álgebra de Boole corresponde a la definición del elemento opuesto.

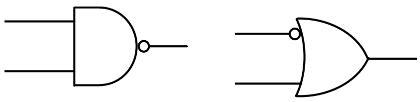
La compuerta lógica que se emplea para realizar la tarea de invertir una variable y su tabla de verdad se muestran a continuación:



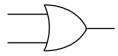
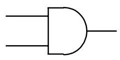
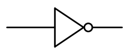
Compuerta Inversora (Not)

I = no A	A
0	1
1	0

Los inversores, también pueden indicarse por medio de un pequeño círculo entre la variable y el resto del esquema. Por ejemplo, asociándolos a compuertas AND u OR:

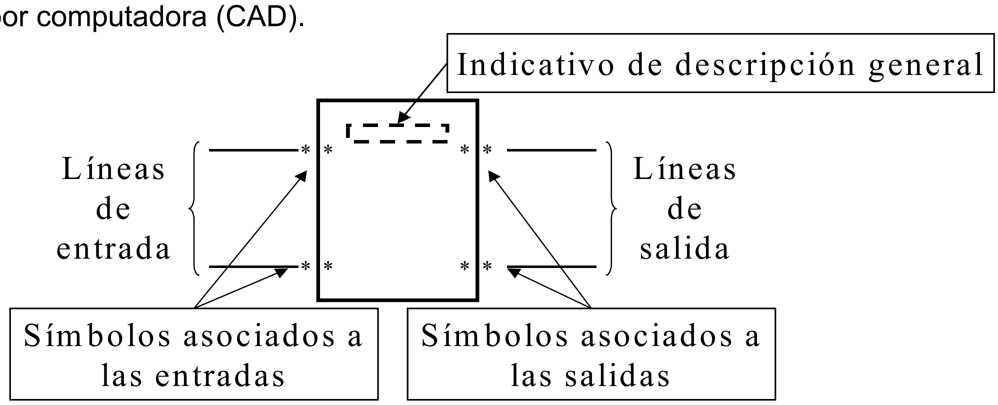


De acuerdo a lo expuesto, realizamos la siguiente tabla, que muestra la relación entre el álgebra de Boole binaria, conjuntos, proposicional, conmutacional y circuitos lógicos.

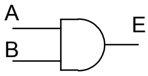
Boole (binaria)	Conjuntos	Proposicional	Conmutación (positiva)	Circuitos lógicos
Elementos	Conjuntos	Proposiciones	Acción, señal	Variables
Suma lógica (+)	Unión ($\cup$)	Disyunción (v), ó	Circuito paralelo	Or 
Producto lógico (.)	Intersección ($\cap$)	Conjunción (^), y	Circuito serie	And 
Elemento opuesto	Complemento	Negación (no)	Inversor	Not 
Neutro de la suma	Conj. Vacío (ϕ)	Falsedad (F)	No a la acción	0
Neutro del producto	Conj. Universal (U)	Certeza (V)	Si a la acción	1

En cuanto a la simbología, conviene aclarar que la representación tradicional que se muestra en este trabajo, es la más empleada en la bibliografía dedicada a profesionales durante la primera etapa de su formación.

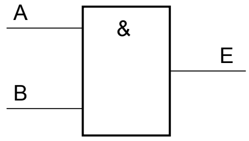
Sin embargo, manuales de datos técnicos y otras publicaciones más avanzadas, utilizan la simbología normalizada (ISO) que se menciona a continuación. Esta última, facilita la representación de bloques funcionales de circuitos integrados digitales y el diseño asistido por computadora (CAD).

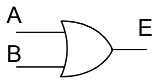


A continuación se muestran algunos ejemplos de la representación normalizada:

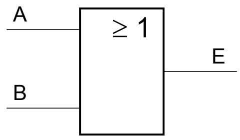
La compuerta AND: 

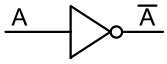
, es equivalente a:



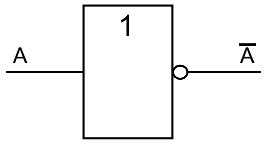
La compuerta OR: 

, es equivalente a:



La compuesta inversora: 

, es equivalente a:



ALGUNAS DEMOSTRACIONES DE TEOREMAS DEL ÁLGEBRA DE BOOLE

Introducción:

Empleando los postulados del álgebra de Boole, demostraremos algunas de las leyes o teoremas[13] necesarios para el desarrollo de circuitos lógicos.

La habilidad en el empleo del álgebra, permite avanzar en la comprensión de las aplicaciones y más tarde redundará en una reducción de costos de los sistemas digitales que, a partir de estas leyes, pueden ser simplificados.

Para que este texto resulte cómodo[14] en el análisis de los teoremas, se transcriben a continuación, los postulados de Huntington.

1) $\exists$ C ^ $\exists$(a, b, ...) $\in$ C / a R b ^ $\exists$(a + b) $\in$ C ^ $\exists$(a . b) $\in$ C

Recordemos que a y b están en relación de equivalencia “a R b”, si se cumplen los principios de identidad “a R a”, simetría “a R b $\Rightarrow$ b R a” y transitividad “a R b ^ b R c $\Rightarrow$ a R c”.

2 a) $\forall$ (a, b) $\in$ C $\rightarrow$ a + b = b + a

2 b) $\forall$ (a, b) $\in$ C $\rightarrow$ a . b = b . a

3 a) $\forall$ (a, b, c) $\in$ C $\rightarrow$ (a + b) + c = a + (b + c)

3 b) $\forall$ (a, b, c) $\in$ C $\rightarrow$ (a . b) . c = a . (b . c)

4 a) $\forall$ (a, b, c) $\in$ C $\rightarrow$ (a + b) . c = a . c + b . c

4 b) $\forall$ (a, b, c) $\in$ C $\rightarrow$ (a . b) + c = (a + c) . (b + c)

5 a) $\forall$ a $\in$ C $\exists$ N1 / a + N1 = a

5 b) $\forall$ a $\in$ C $\exists$ N2 / a . N2 = a

6 a) $\forall$ a $\in$ C $\exists$ $\overline{a}$ / a + $\overline{a}$ = N2 ^ / a . $\overline{a}$ = N1 (6 b)

En algunos casos podemos referirnos a ellos por sus nombres, es decir: 1) Definición, 2) Conmutatividad, 3) Asociación, 4) Distributividad, 5) Existencia del elemento neutro y 6) Existencia del elemento opuesto.

Demostraciones

I) Dualidad:

Cada identidad deducida a partir de los postulados del álgebra de Boole, permanece válida si la operación “+” y “.” y los elementos “0” y “1” se intercambian entre sí.

Se deduce de la simetría de los postulados con respecto a las dos operaciones: suma lógica y producto lógico; y a los dos elementos neutros: N1 (el “0” lógico) y N2 (el “1” lógico).

Por otra parte el lector puede escribir nuevamente los postulados realizando el cambio propuesto por el teorema de Dualidad y comprobar que vuelve a obtener los postulados originales, cambiados de orden.

II) a + 1 = 1:

$1 = a + \overline{a} = a + \overline{a} \cdot 1 = (a + \overline{a}) \cdot (a + 1) = 1 \cdot (a + 1) = a + 1$

Hemos utilizando los postulados: el recíproco de 6a, 5b, 4b, 6a y 5b, respectivamente.

Empleando el concepto de dualidad, se podrá comprobar el teorema: “ a . 0 = 0 ”, aplicando los postulados: recíproco de 6b, 5a, 4a, 6b y 5a, respectivamente.

---

[13] En algunos textos se los denomina también “Principios”.

[14] Significado de algunos símbolos: $\exists\rightarrow$ existe, ^ $\rightarrow$ y, $\in\rightarrow$ pertenece, / $\rightarrow$ tal que, $\forall\rightarrow$ para todo, $\Rightarrow$ $\rightarrow$ implica.

III) Unicidad: $a + a = a$

$a = a + 0 = a + a \, . \, \overline{a} = (a + a) \, . \, (a + \overline{a}) = (a + a) \, . \, 1 = a + a$

Hemos utilizando los postulados: elemento neutro de la suma, producto con el elemento opuesto, distributividad, suma del elemento opuesto y elemento neutro del producto, respectivamente. También aquí podrá utilizarse el concepto de dualidad y demostrar: “ $a \, . \, a = a$ ”.

IV) Absorción: $a + a \, . \, b = a$

$a = 1 \, . \, a = (1 + b) \, . \, a = (a \, . \, 1 + b \, . \, a) = a + a \, . \, b$

Hemos utilizando los postulados: elemento neutro del producto, teorema “II ” aplicado a la variable b, distributividad y producto con el elemento opuesto, respectivamente.

V) Doble negación: $\overline{\overline{a}} = a$

V a) Sea $\overline{a} = x$ y por lo tanto $\overline{\overline{a}} = \overline{x}$.

V b) Considerando el postulado de existencia del elemento opuesto:

$x + \overline{x} = 1$ y $x \, . \, \overline{x} = 0$.

V c) Sustituyendo las variables: $\overline{a} + \overline{\overline{a}} = 1$ y $\overline{a} \, . \, \overline{\overline{a}} = 0$.

V d) Aplicando el postulado de conmutatividad: $\overline{\overline{a}} + \overline{a} = 1$ y $\overline{\overline{a}} \, . \, \overline{a} = 0$.

V e) Que por supuesto satisfacen el postulado de existencia del elemento opuesto:

6 a) $\forall a \in C \; \exists \; \overline{a} \; / \; a + \overline{a} = N2$ ^ / $a \, . \, \overline{a} = N1$ (6 b)

Donde N1 = 0 y N2 = 1 y, por lo tanto esa parte del postulado podría escribirse:

6 a) $a + \overline{a} = 1$ ^ / $a \, . \, \overline{a} = 0$ (6 b) .

Ya que por el postulado 1), todo elemento del álgebra de Boole es equivalente a sí mismo, y por lo tanto $\overline{a} = \overline{a}$, considerando la deducción del punto V d):

$\overline{\overline{a}} + \overline{a} = 1$ y $\overline{\overline{a}} \, . \, \overline{a} = 0$, y el postulado 6 :

$a + \overline{a} = 1$ ^ / $a \, . \, \overline{a} = 0$, se deduce que $\overline{\overline{a}} = a$. Como queríamos demostrar.

VI) De Morgan: $\overline{a + b} = \overline{a} \, . \, \overline{b}$

VI a) Sea $a + b = u$ y por lo tanto “ $\overline{a + b} = \overline{u}$ ”, **si se cumple la igualdad presentada en el teorema de De Morgan, entonces se cumplirá:** “ $\overline{a} \, . \, \overline{b} = \overline{u}$ “. Lo demostraremos empleando el postulado de existencia del elemento opuesto.

VI b) Consideremos ese postulado, aplicado a la variable “u”:

$u + \overline{u} = 1$ y $u \, . \, \overline{u} = 0$

VI c) Sustituyendo las variables: $(a + b) + (\overline{a} \, . \, \overline{b}) = 1$ y $(a + b) \, . \, (\overline{a} \, . \, \overline{b}) = 0$

VI d) Valiéndonos del postulado de distributividad: $(a + b + \overline{a}) \, . \, (a + b + \overline{b}) = 1$

y $(a \, . \, \overline{a} \, . \, \overline{b}) + (b \, . \, \overline{a} \, . \, \overline{b}) = 0$

VI e) Ahora aplicamos el postulado de conmutatividad:

$(a + \overline{a} + b) \, . \, (a + b + \overline{b}) = 1$

y $(a \, . \, \overline{a} \, . \, \overline{b}) + (b \, . \, \overline{b} \, . \, \overline{a}) = 0$

VI f) A continuación utilizamos el postulado de existencia del elemento opuesto:

$(1+b) \cdot (a+1) = 1$

y $(0 \cdot \overline{b}) + (0 \cdot \overline{a}) = 0$

VI g) A continuación utilizamos el teorema demostrado en II) "a + 1 = 1" y "a . 0 = 0", aplicados a las variables a ó b, según corresponda:

$(1+b) \cdot (a+1) = 1 \Rightarrow 1 \cdot 1 = 1$

y $(0 \cdot \overline{b}) + (0 \cdot \overline{a}) = 0 \Rightarrow 0 + 0 = 0$. Como queríamos demostrar.

Funciones de un álgebra de Boole

Una función del álgebra de Boole es una variable binaria[15], cuyo valor depende de una cierta combinación de valores relacionados por las operaciones producto y suma, y donde las variables que intervienen pueden presentarse en forma directa o por medio de su opuesto. Una forma de nombrarla es: $f_{(a, b, c, ...)}$

Por ejemplo: $f_{(b, a)} = a + \overline{b} \cdot \overline{a}$

En este caso, la función $f_{(b, a)}$ vale 1 cuando a = 1, o cuando b = 0 y a = 0. También puede decirse que $f_{(b, a)}$ vale cero, solo si b = 1 y a = 0.

Podemos representar estas posibilidades en una tabla de verdad:

b	a	Valor para $f_{(b, a)} = a + \overline{b} \cdot \overline{a}$
0	0	$f_{(0, 0)} = 1$
0	1	$f_{(0, 1)} = 1$
1	0	$f_{(1, 0)} = 0$
1	1	$f_{(1, 1)} = 1$

O más brevemente:

b	a	$f_{(b, a)}$
0	0	1
0	1	1
1	0	0
1	1	1

Teorema Nº: VII)[16] Para toda función $f_{(a, b, c, ...)}$ se verifica:

$$f_{(a, b, c, ...)} = a \cdot f_{(1, b, c, ...)} + \overline{a} \cdot f_{(0, b, c, ...)}$$

Podemos comprobar este teorema, verificando los valores que adopta la función para las dos posibles condiciones de la variable "a", es decir: para a = 0 y luego para a = 1.

- Para a = 0, $\overline{a} = 1$; por lo tanto en esta condición $f_{(a, b, c, ...)} = f_{(0, b, c, ...)}$.

En el teorema que nos ocupa: $f_{(0, b, c, ...)} = 0 \cdot f_{(1, b, c, ...)} + 1 \cdot f_{(0, b, c, ...)}$

$f_{(0, b, c, ...)} = 0 + f_{(0, b, c, ...)}$

El primer miembro de la derecha es cero por el teorema II, y el segundo es igual a $f_{(0, b, c, ...)}$ por el postulado del elemento neutro del producto.

- Para a = 1, $\overline{a} = 0$; por lo tanto en esta condición $f_{(a, b, c, ...)} = f_{(1, b, c, ...)}$.

[15] Puede adoptar uno de dos valores: "0", ó "1".
[16] En algunos textos se lo conoce formalmente como Teorema de Expansión.

En el teorema: $f_{(1,\, b,\, c,\ \ldots)} = 1 \cdot f_{(1,\, b,\, c,\ \ldots)} + 0 \cdot f_{(0,\, b,\, c,\ \ldots)}$

$f_{(1,\, b,\, c,\ \ldots)} = f_{(1,\, b,\, c,\ \ldots)} + 0$

El primer miembro de la derecha es igual a $f_{(1,\, b,\, c,\ \ldots)}$ por el postulado del elemento neutro del producto, y el segundo es cero por el teorema "II".

Término canónico[17] *de una función del álgebra de Boole* *(Minitérminos y maxitérminos)*

Un producto canónico (o minitérmino), es un producto lógico en el que intervienen la totalidad de las variables (literales: a, b, c, ...) que pertenecen a la función.

Por ejemplo, en la función de tres variables: $f_{(c,\, b,\, a)}$, el producto "c . b . a" es un producto canónico. Del mismo modo "$c \cdot \overline{b} \cdot \overline{a}$", es otro producto canónico ya que posee todas las variables de la función, sean estas en forma directa o negada.

El teorema anterior, se puede emplear para demostrar la validez de la representación de una función del álgebra de Boole como suma de productos canónicos.

El teorema Dual: $f_{(a,\, b,\, c,\ \ldots)} = (a + f_{(0,\, b,\, c,\ \ldots)}) \cdot (\overline{a} + f_{(0,\, b,\, c,\ \ldots)})$, se utiliza para demostrar la validez de la representación de una función del álgebra de Boole como producto de sumas canónicas.

Una suma canónica, es una suma lógica en la que intervienen la totalidad de las variables (literales: a, b, c, ...) que pertenecen a la función. Por ejemplo, "$c + \overline{b} + \overline{a}$" en una función de tres variables, es una suma canónica (un maxitérmino).

Representación de una función como suma de productos canónicos y como producto de sumas canónicas

Empleando el teorema VII, podemos analizar una función de la siguiente manera:

$$f_{(a,\, b,\ \ldots)} = a \cdot f_{(1,\, b,\ \ldots)} + \overline{a} \cdot f_{(0,\, b,\ \ldots)}$$

Aplicamos nuevamente el teorema a los términos de la derecha:

$$f_{(a,\, b,\ \ldots)} = a \cdot (b \cdot f_{(1,\, 1,\ \ldots)} + \overline{b} \cdot f_{(1,\, 0,\ \ldots)}) + \overline{a} \cdot (b \cdot f_{(0,\, 1,\ \ldots)} + \overline{b} \cdot f_{(0,\, 0,\ \ldots)})$$

Utilizando el postulado de distributividad:

$$f_{(a,\, b,\ \ldots)} = a \cdot b \cdot f_{(1,\, 1,\ \ldots)} + a \cdot \overline{b} \cdot f_{(1,\, 0,\ \ldots)} + \overline{a} \cdot b \cdot f_{(0,\, 1,\ \ldots)} + \overline{a} \cdot \overline{b} \cdot f_{(0,\, 0,\ \ldots)})$$

Detenemos el análisis aquí, pero queda claro que podría continuarse con las variables c, d, etc.

Supongamos ahora una función de solo dos variables a y b:

$$f_{(a,\, b)} = a \cdot b \cdot f_{(1,\, 1)} + a \cdot \overline{b} \cdot f_{(1,\, 0)} + \overline{a} \cdot b \cdot f_{(0,\, 1)} + \overline{a} \cdot \overline{b} \cdot f_{(0,\, 0)}$$

Ordenando[18] las variables de mayor a menor (de "Z" a "A"), podremos escribir:

$$f_{(b,\, a)} = b \cdot a \cdot f_{(1,\, 1)} + \overline{b} \cdot a \cdot f_{(0,\, 1)} + b \cdot \overline{a} \cdot f_{(1,\, 0)} + \overline{b} \cdot \overline{a} \cdot f_{(0,\, 0)}$$

Observe que cuando una variable está directa (no negada) en el lugar que le corresponde en la función hay un 1 y cuando la variable está negada hay un 0

La función quedó expresada como suma de productos. Cada producto es canónico (minitérmino) y está acompañado de una función que puede valer 1 o 0.

Si la función vale 1, ese minitérmino forma parte de la expresión. En cambio si vale 0, el minitérmino desaparece de la misma, ya que al multiplicarlo por 0 el resultado es cero.

El valor de $f_{(b,\, a)}$, puede tomarse de su tabla de verdad o calcularse con la expresión que la define (aunque no sea canónica).

[17] Existen dos tipos de términos canónicos: producto canónico y suma canónica. Se los conoce en algunos textos como: "términos mínimos" y "términos máximos", respectivamente.

[18] El ordenamiento de las variables, en una forma u otra, no cambia a la función en sí. Es frecuente que cada autor utilice un mismo ordenamiento en toda su obra. En esta, se intenta ilustrar esta cuestión desde el principio de las demostraciones para que el lector pueda manipularlas, independientemente del ordenamiento que tengan. Ahora bien, es sumamente importante conocer qué ordenamiento se emplea, para las representaciones que se explican a continuación.

Por ejemplo[19]: $f_{(b,\,a)} = b\,.\,a\,.\,1 + \overline{b}\,.\,a\,.\,1 + b\,.\,\overline{a}\,.\,0 + \overline{b}\,.\,\overline{a}\,.\,1$; ya que: $f_{(1,\,1)} = 1$, $f_{(0,\,1)} = 1$, $f_{(1,\,0)} = 0$ y $f_{(0,\,0)} = 1$.

Quedará: $f_{(a,\,b)} = b\,.\,a + \overline{b}\,.\,a + \overline{b}\,.\,\overline{a}$

Para facilitar la nomenclatura y el análisis en funciones de mayor número de variables, representaremos cada minitérmino por medio de un número decimal. Este surge de considerar la combinación de unos y ceros de las funciones, como si fueran binarios.

Por ejemplo:

El término que acompaña a $f_{(0,\,0)}$, “ $\overline{b}\,.\,\overline{a}$ ” lo representaremos con el número 0.

El término que acompaña a $f_{(0,\,1)}$, “ $\overline{b}\,.\,a$ ” lo representaremos con el número 1.

El término que acompaña a $f_{(1,\,0)}$, “ $b\,.\,\overline{a}$ ” lo representaremos con el número 2.

El término que acompaña a $f_{(1,\,1)}$, “ $b\,.\,a$ ” lo representaremos con el número 3.

De los 4 valores posibles, nuestro ejemplo solo tiene a los minitérminos: 0, 1 y 3.

Como los minitérminos se suman entre sí para obtener la función, lo indicaremos:

$f_{(a,\,b)} = \Sigma_2\,(0,1,3)$

El subíndice “2” en la sumatoria, representa el número de variables.

Si la función contara con tres variables, los minitérminos posibles serían 8, considerando las combinaciones posibles de las 3 variables y que cada una puede presentarse en 2 formas, negada o no negada. Con el método empleado, en este caso, representaríamos a cada minitérmino con un número decimal entre 0 y 7. Generalizando diremos 2^n posibilidades, para “n” variables y, el número que representa a cada minitérmino será un valor entre “0” y “$(2^n - 1)$”.

Los maxitérminos pueden analizarse de la siguiente forma:

1º) ¿Cuáles son los minitérminos que no pertenecen a la función ($\overline{f}$)?.

En este caso solo el Nº 2 ($\overline{f} = b\,.\,\overline{a}$) .

2º) complementándolo ($\overline{\overline{f}}$) y aplicando el teorema de “De Morgan” queda:

$f_{(a,\,b)} = (\overline{b} + a)$.

Este término es una suma canónica, es decir un maxitérmino y su representación en la nomenclatura acordada será el número decimal “1”. $f_{(a,\,b)} = \Pi_2\,(1)$.

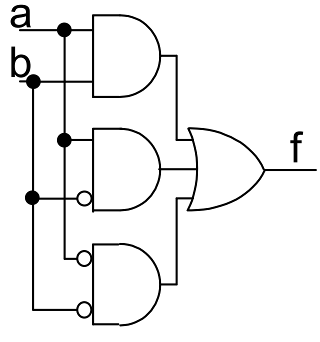
De esta forma, la función tiene dos posibles representaciones canónicas:

$f_{(a,\,b)} = \Sigma_2\,(0,1,3) = \Pi_2\,(1)$.

La primera de las formas “ $f_{(a,\,b)} = \Sigma_2\,(0,1,3)$ ”, es la vista anteriormente:

$f_{(a,\,b)} = b\,.\,a + \overline{b}\,.\,a + \overline{b}\,.\,\overline{a}$

El circuito que le corresponde es:

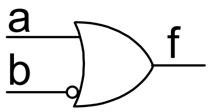


[19] Este ejemplo está tomado de la tabla de la página 15, **Funciones de un álgebra de Boole**. Observe además, que esta función y su tabla de verdad coincide con la “implicación”, explicada al definirse las “proposiciones lógicas”, p. 4.

La segunda forma “ $f_{(a,\,b)} = \Pi_2\,(1)$”, que corresponde a la suma canónica es:

$f_{(a,\,b)} = \overline{b} + a$

Y en esta última, el circuito será:



En ambos casos la función es la misma representada de dos formas distintas pero equivalentes. Resulta obvio que en este caso la segunda resultó mucho más sencilla.

A la hora de realizar el circuito que resuelva una función, será muy importante emplear la versión más simple posible, que además será la más económica, veloz y confiable.

Otro ejemplo aclarará la metodología: Sea la función dada mediante la siguiente tabla:

c	b	a	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Su expresión simbólica será:

$f = \Sigma_3\,(2,3,4,5,6,7)$, dada como suma de minitérminos.

Los términos que no pertenecen a la función:

$\overline{f} = \Sigma_3\,(0,1)$,

el complemento de esto es: $\overline{\overline{f}} = f = \overline{\Sigma_3\,(0,1)}$

y luego aplicando De Morgan será: $f = \Pi_3\,(6,7)$.

Observemos que 6 y 7 son los complementos a 7, de 1 y 0 respectivamente.

En general y para facilitar el cálculo de los maxitérminos, el complemento a $(2^n - 1)$ de los minitérminos que no pertenecen a la función, representa a los maxitérminos.

Como práctica sugerimos enunciar la expresión algebraica en sus dos formas.

Por ejemplo los maxitérminos en la función: $f = \Pi_3\,(6,7)$, permiten escribir la expresión como producto de sumas canónicas:

$f_{(c,\,b,\,a)} = (c + b + \overline{a}) \cdot (c + b + a)$

A raíz de todo lo expuesto, se infiere la necesidad de simplificar las funciones lógicas a fin de lograr circuitos más sencillos y eficientes.

Existen varios métodos de simplificación.

Podremos clasificarlos en tabulares (gráficos) y numéricos.

Los métodos gráficos como los de Veitch o Karnaugh permiten simplificar funciones de pocas variables.

Métodos numéricos como el de Quine - McCluskey, en cambio, suministran una metodología apta para un número de variables mucho mayor (>4).

La simplificación, en el caso de suma de productos, intenta reducir la cantidad de variables que intervienen en cada producto. A su vez baja la cantidad de productos. El análisis Dual permite obtener ventajas análogas en el producto de sumas. Eso sí, resultan productos o sumas no canónicas.

Los métodos de simplificación se basan en la aplicación de los postulados y teoremas. Para hacerlo algebraicamente, primero se identifican y agrupan los términos donde las variables se repiten salvo una de ellas que, aparece en su forma directa en un término y negada en el otro. Entonces se aplica el recíproco del postulado Nº IV. Luego aplicando el postulado Nº VI se reemplaza la variable agrupada por el término neutro que corresponda y por último se aplica el postulado Nº V. Por ejemplo en: $f_{(c,\,b,\,a)} = (c + b + \overline{a}) \cdot (c + b + a)$;

se podrá agrupar $f_{(c,\,b,\,a)} = (c + b) + (\overline{a} \cdot a) = (c + b) + 0 \rightarrow f_{(c,\,b,\,a)} = (c + b)$

La variable que desaparece por efecto de la simplificación, tiene un peso tal en el ordenamiento elegido, que coincide con la diferencia entre los valores numéricos decimales asignados inicialmente a cada término canónico de la función.

Para estos y otros métodos, la bibliografía es abundante y se recomienda su lectura comprensiva, estudio y ejercitación correspondiente.

El método tabular de Karnaugh se utiliza luego, en este texto en la última parte de la INTRODUCCIÓN A LOS CIRCUITOS SECUENCIALES, para hallar las expresiones simplificadas de los circuitos que allí se estudian.

Si bien, no es el objetivo de este trabajo analizar el método de Quine - McCluskey, sí nos parece oportuno asentar un ejemplo en las líneas siguientes. Esto permitirá clarificar el tema después de estudiarlo en la bibliografía adecuada.

Se trata de un ejemplo para una función de cuatro variables:

Sea la función $f_{(d,\ c,\ b,\ a)} = \sum_4 (0, 1, 2, 4, 5, 8, 9, 10, 11, 15)$;

En una 1ª tabla, se agrupan en orden creciente de acuerdo a la cantidad de unos que poseen, los términos que pertenecen a la función. Así por ejemplo el 5 (0101_2), está dentro del grupo que tiene el número 2 de unos, junto con el 9 (1001_2) y el 10 (1010_2).

Luego se buscan, entre grupos adyacentes, pares de minitérminos cuya diferencia sea una potencia entera <u>positiva</u> de 2. Como ser el “0” del primer grupo con el “1” del segundo grupo ya que $1 - 0 = +1 = 2^0$. De igual modo entre “0” y “2” ($2 - 0 = 2 = 2^1$) y así con los restantes pares de valores que pueden observarse en esa Tabla 2. A cada término de una tabla que fue utilizado para formar parte de la tabla siguiente, se le coloca una “X” en la fila correspondiente (todos los términos de la primera tabla han sido utilizados en la segunda).

Tabla 1

Cant. de unos	Término	Utilizo?
0	0	X
1	1	X
	2	X
	4	X
	8	X
2	5	X
	9	X
	10	X
3	11	X
4	15	X

Tabla 2

Pares	Difiere	Utilizo?
0 - 1	1	X
0 - 2	2	X
0 - 4	4	X
0 - 8	8	X
1 - 5	4	X
1 - 9	8	X
2 - 10	8	X
4 - 5	1	X
8 - 9	1	X
8 - 10	2	X
9 - 11	2	X
10 - 11	1	X
11 -15	4	(A) *

Tabla 3

Pares de pares	Difiere
0 - 1 - 4 - 5	1 - 4 (B) *
0 - 1 - 8 - 9	1 - 8 (C) *
0 - 2 - 8 - 10	2 - 8 (D) *
0 - 4 - 1 - 5	4 - 1 (E) *
0 - 8 - 1 - 9	8 - 1 (F) *
0 - 8 - 2 - 10	8 - 2 (G) *
8 - 9 - 10 - 11	1 - 2 (H) *
8 - 10 - 9 - 11	2 - 1 (I) *

La Tabla 2 queda ordenada cuando para crearla, se recorre metódicamente la Tabla 1.

En la Tabla 2, han quedado cuatro sectores. Cada sector, está formado por pares de términos tomados de la primera tabla, uno de un bloque y otro del bloque adyacente siguiente.

Otros ejemplos afianzarán el método de razonamiento: el segundo sector, se obtiene al tomar un término del segundo grupo de la Tabla 1 (que tienen solo un uno) y otro del tercer grupo (que tienen dos unos): “1-5”, “1-9”, “2-10”, “4-5” etc., y cuya diferencia también es una potencia entera de 2 (“5 -1” = 2^2, “9 -1” = 2^3, etc.). Luego otro sector entre los que tienen dos y tres unos (9 - 11 y 10 - 11) y un sector (último para este caso particular) entre los que tienen tres y cuatro unos (el 11 - 15). Siempre respetando que las diferencias sean potencias enteras de 2.

Para obtener la siguiente tabla (la Tabla 3), se analizan los sectores adyacentes de la tabla que acabamos de analizar (la Tabla 2 en este caso).

Deben buscarse pares que tengan el mismo valor de diferencia, por ejemplo: en el primer sector, el par: “0 _ 1” difiere en 1 y el par “4 _ 5” del sector adyacente siguiente, también difiere en 1. Ahora se calcula la diferencia entre los pares:

```
    4 _ 5
 -  0 _ 1   Se debe verificar que la diferencia entre pares sea una potencia entera de “2”
 ---------
    4 _ 4        (4 = 2^2)
```

En la tabla 3 se han indicado las diferencias calculadas, por ejemplo: 1 - 4 (B) *. Donde el primer valor: “1” es la diferencia dentro de un par de minitérminos de la Tabla 2 y el segundo valor: “4” es la diferencia entre los dos pares (que acabamos de calcular).

De esta forma se va completando la Tabla 3 hasta terminar de buscar esas condiciones en la totalidad de los valores de la Tabla 2.

Observe que el término A no fue incluido en ningún grupo de la Tabla 3, por esta razón está marcado con un asterisco “*”. Se dice que “A” es un Implicante Primo.

Terminada la Tabla 3, se analizan sus componentes buscando términos adyacentes con diferencias iguales (parejas de diferencias en este caso). Al no haber diferencias iguales entre los dos sectores de esta última tabla, se termina el proceso de agrupación. De lo contrario, debería hacerse una "Tabla 4".

Con el mismo criterio empleado para el término A, los términos: B, C, D, E, F, G, H e I, llevan asterisco “*” (implicantes primos), ya que no fueron incluidos en tablas posteriores.

Algunas agrupaciones contienen los mismos minitérminos, tal es el caso de: B y E ya que incluyen: 0, 1, 4 y 5, solo que en distinto orden y por consiguiente resultan equivalentes y dará lo mismo usar uno u otro. Emplearemos el B. De igual modo los términos: C y F (0-1-8-9), D y G (0-2-8-10) y H e I (8-9-10-11). Emplearemos: C, D y H.

Ahora en la siguiente tabla, incluimos a los minitérminos implicados en la función y las agrupaciones de términos que los contienen y no han podido ser incluidas en agrupaciones de orden mayor (implicantes primos).

La tabla de implicantes (colocamos en ella, solo los términos primos distintos entre sí):

Términos	primos	0	1	2	4	5	8	9	10	11	15
A	11 - 15									X	X
B	0 - 1 - 4 - 5	X	X		X	X					
C	0 - 1 - 8 - 9	X	X				X	X			
D	0 - 2 - 8 - 10	X		X			X		X		
H	8 - 9 - 10 - 11						X	X	X	X	
				↑	↑	↑					↑

El análisis de esta tabla nos proveerá de la expresión simplificada de la función.

Existen algunos minitérminos que solo pueden ser resueltos mediante una agrupación determinada, (tienen una sola cruz en su columna y han sido identificados con una flecha en la parte inferior de la taba). Esto nos obliga a emplear esa agrupación, conocida como término esencial. Por ejemplo el minitérmino 15, solo se resuelve en la agrupación “A”.

Términos esenciales son: A: 11 - 15 ($d\ b\ a$); B: 0 - 1 - 4 - 5 ($\overline{d}\ \overline{b}$), y D: 0 - 2 - 8 - 10 ($\overline{c}\ \overline{a}$).

La expresión algebraica que corresponde a cada término esencial se obtiene mediante la simplificación de los minitérminos que le son propios. Veamos uno de ellos, el A:

El minitérmino 11: $d \cdot \overline{c} \cdot b \cdot a$, y el 15: $d \cdot c \cdot b \cdot a$.

La sumatoria es: $d \cdot \overline{c} \cdot b \cdot a + d \cdot c \cdot b \cdot a = d \cdot b \cdot a \cdot (\overline{c} + c) = d \cdot b \cdot a$

Otra forma de obtener el resultado correcto es: la diferencia entre los minitérminos indica la variable que se simplifica. Veamos: 15 - 11 = 4. En el ordenamiento de las variables lógicas que hemos empleado: d c b a , y considerando el peso que le corresponde a cada una: $d \rightarrow 2^3$, $c \rightarrow 2^2$, $b \rightarrow 2^1$ y $a \rightarrow 2^0$, la variable que pesa $4 = 2^2$ es “c” y por lo tanto se simplifica (desaparece de la expresión) y el resto de las variables quedan en la condición en la que se encontraban.

De los dos términos no esenciales “C” y “H”, elegimos uno para dar solución al minitérmino "9", ya que los términos esenciales no lo resuelven.

Elegimos “C”: 0 - 1 - 8 - 9 ($\overline{c}\ \overline{b}$). Observe que desaparecieron las variables d y a, por simplificación ya que sus pesos: 8 y 1 coincide con las diferencias numéricas.

$$\left.\begin{array}{ll} 0 - 1 & \text{Difiere en 1} \\ 8 - 9 & \text{Difiere en 1} \end{array}\right\} \text{Entre ellos difieren en 8}$$

La función simplificada queda: $f = d\ b\ a + \overline{d}\ \overline{b} + \overline{c}\ \overline{a} + \overline{c}\ \overline{b}$.

De haber elegido “H”: 8 - 9 - 10 - 11 ($d\ \overline{c}$), (en lugar de “C”) la función quedaría:

$f = d\ b\ a + \overline{d}\ \overline{b} + \overline{c}\ \overline{a} + d\ \overline{c}$.

IMPLEMENTACIÓN DE FUNCIONES SIMPLES

(Circuitos combinacionales)

Introducción

Las compuertas lógicas presentadas en la primera parte "INTRODUCCIÓN AL ÁLGEBRA CONMUTACIONAL", los postulados del álgebra de Boole y los teoremas demostrados en la segunda "ALGUNAS DEMOSTRACIONES DE TEOREMAS DEL ÁLGEBRA DE BOOLE", brindan la posibilidad de analizar y comprender algunos circuitos frecuentemente utilizados en sistemas automáticos de cómputo de datos y control digital. Por otra parte, la simplificación de funciones permite realizarlos de manera eficiente.

Los circuitos combinacionales, proveen una representación gráfica tal que las aplicaciones lógicas y aritméticas, muchas veces resultan más claras.

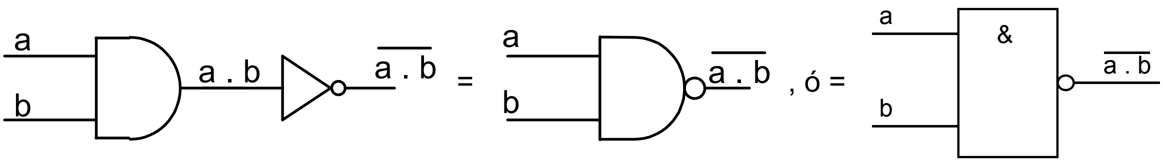
Las funcionalidades de los circuitos combinacionales que se exponen a continuación, se emplean profusamente en aplicaciones prácticas del hardware[20] de sistemas digitales.

Si bien se presentan de manera básica y sencilla, resultan imprescindibles para progresar en materias relacionadas con sistemas de computación.

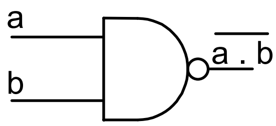
NAND y NOR

Debido a la simplicidad de diseño de circuitos digitales con compuertas del tipo NAND y NOR solamente, varias tecnologías como "lógica de transistor - transistor" (TTL) y otras, propiciaron la construcción de circuitos a partir de ellas.

- La función **NAND** (NO AND), corresponde a la negación del producto lógico. Es decir, se realiza el producto de las variables de entrada y luego se invierte la salida.



Se puede decir que la salida de una compuerta NAND es uno cuando alguna entrada es cero (detecta la presencia de al menos un cero). Desde otro punto de vista, la salida es cero si y solo si todas las entradas valen uno simultáneamente.



b	a	$f = \overline{a . b}$
0	0	1
0	1	1
1	0	1
1	1	0

En una NAND, cuando una de las entradas es 1, la otra se invierte a la salida.

Por ejemplo, si b = 1 → f = $\overline{a}$

Asimismo, uniendo ambas entradas (a = b), la salida será su negado $f = \overline{a}$, ó $f = \overline{b}$.

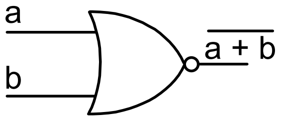
Si colocamos un inversor (negador) a la salida de una NAND obtendremos una AND.

Por otra parte, si negamos cada una de las entradas de una NAND, la función resultante corresponde a una OR. Esto surge de aplicar De Morgan.

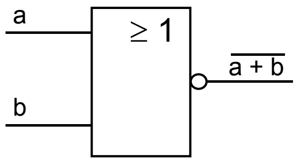
- La función **NOR** (NO OR), corresponde a la negación de la suma lógica. Es decir, se realiza la suma de las variables de entrada y luego se invierte la salida.

[20] Algunas de estas aplicaciones funcionales se implementan en combinaciones de hardware y software. Más aún, algunas funciones se desarrollan y aplican en sistemas que operan de forma analógica, como los Multiplexores.

Se puede decir que la salida de una compuerta NOR es uno cuando todas las entradas son cero simultáneamente (detecta el valor cero en la entrada). Otro enfoque nos permite afirmar que la salida es cero si por lo menos una de las entradas vale uno.



, ó =



b	a	$f = \overline{a + b}$
0	0	1
0	1	0
1	0	0
1	1	0

En una NOR, cuando una de las entradas es 0, la otra se invierte a la salida.

Por ejemplo, si $b = 0 \rightarrow f = \overline{a}$

Aquí también, uniendo las entradas (a = b), la salida será su negado $f = \overline{a}$, ó $f = \overline{b}$.
Si colocamos un inversor (negador) a la salida de una NOR obtendremos una OR.
Negando cada una de las entradas de una NOR, el resultado corresponde a una AND.

<u>O Exclusiva</u> (Exclusive Or, X-OR)

La “O exclusiva” entrega un uno a su salida, cuando una de sus dos entradas está en uno, pero **<u>no</u>** simultáneamente (se excluye esta condición).

Expresado en forma de función lógica, sería: $f = (b + a) \,\mathbf{.}\, \overline{b . a}$

Esto difiere de la operación “O” (inclusiva) definida en el primer postulado del álgebra de Boole en la cual la salida es uno cuando cualquiera de las entradas es uno, aún si todas tienen el valor uno al mismo tiempo.

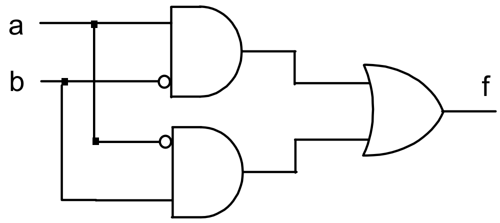
Otra forma de analizar, sería: la salida de una X-OR es uno si una entrada no vale uno y la otra sí, ó viceversa (la primera es uno y la segunda no).

Ahora la expresión sería: $f = \overline{b} . a + b . \overline{a}$

Las dos expresiones expuestas son equivalentes y poseen por lo tanto la misma tabla de verdad.

b	a	f
0	0	0
0	1	1
1	0	1
1	1	0

Una de las formas de implementarla mediante compuertas AND, OR e inversoras es:



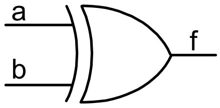
En la tabla de verdad de la X-OR se puede observar que la salida es uno cuando la cantidad de unos de entrada es **“impar”**. Por este motivo se lo emplea como detector de paridad (detecta paridad impar).

Puede, además servir para generar un bit de paridad par. Su salida genera un uno cuando una combinación de entrada es impar y, agregando el bit generado a los bit de entrada, queda un conjunto de bit con paridad par[21].

Otra característica importante es: cuando una de sus entradas es cero (por ejemplo: b = 0), la salida es igual a la otra variable de entrada (f = a) y en cambio, cuando la primera es uno (b = 1), la salida es el opuesto de la segunda ($f = \overline{a}$)[22]. Por este motivo se lo puede utilizar como inversor controlado.

[21] Note que las filas de la tabla anterior (sus tres columnas) tienen una cantidad par de unos, en todos los casos.
[22] Para razonar este comportamiento, divida la tabla a la mitad y observe que: cuando la entrada “b” es “0”, la salida “f” toma los mismos valores que “a”, es decir, si a = 0, f = 0 y si a = 1, f = 1. En la parte inferior de la tabla, cuando la entrada “b” es “1”, la salida “f” toma los valores opuestos a los de “a”, es decir si: a = 0, f = 1 y si a = 1, f = 0.

Debido a su importancia y uso extendido en muchas aplicaciones, a la OR exclusiva se le ha asignado el siguiente símbolo:



, y el signo " $\oplus$ " dentro de las expresiones algebraicas:

$f = b \oplus a$. Se lee: "b OR EXCLUSIVA a".

Multifuciones

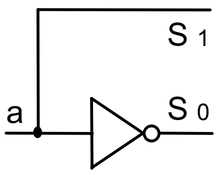
Un sistema que posee varias salidas y por lo tanto responde simultáneamente a varias funciones lógicas, se dice que es una multifunción.

Decodificadores

Un decodificador es un circuito combinacional de varias salidas (multifunción) que convierte la entrada binaria (de uno o más bit), en salidas correspondientes a productos canónicos (minitérminos).

Cada salida tiene, por lo tanto, un solo uno. Este uno se ubica en el minitérmino que coincide con el valor asignado a la salida respectiva (por ejemplo, su equivalente en decimal).

El siguiente circuito lógico, constituye un decodificador elemental. Está formado por una compuerta inversora y una simple línea (o podría darse por medio de una compuerta no inversora). Presenta una entrada "a" y dos salidas "S0" y "S1"

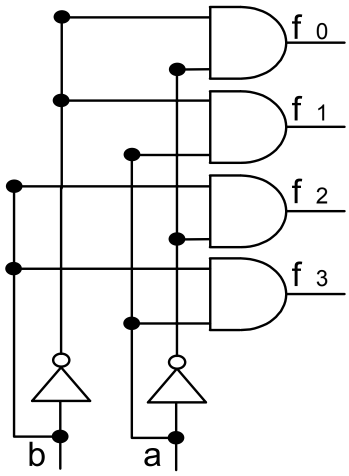


Su tabla de verdad es:

a	S_0	S_1
0	1	0
1	0	1

Las expresiones de las funciones lógicas serán: "$S_0 = \overline{a}$ " y "$S_1 = a$".

Un decodificador un poco más amplio, será el siguiente. Se trata de un decodificador en el cual la entrada está formada por un código binario de dos bit (b a), y cada salida se pone en uno cuando coincide el código binario con el subíndice decimal correspondiente (decodificador binario a decimal).



b	a	f_0	f_1	f_2	f_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$f_0 = \overline{b} \,.\, \overline{a}$

$f_1 = \overline{b} \,.\, a$

$f_2 = b \,.\, \overline{a}$

$f_3 = b \,.\, a$

En la parte inferior del circuito, pueden verse a las entradas "b" y "a", cada una conectada a un decodificador elemental como el analizado en el punto anterior.

Se infiere que, decodificadores de mayores dimensiones, pueden realizarse a partir de circuitos combinacionales simples como, por ejemplo, los expuestos.

Suma aritmética

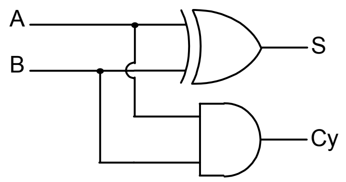
- Un **semi - sumador** aritmético (half adder), es también una multifunción. Cuenta con entradas binarias de un bit "a" y "b" y las salidas: suma "S", y arrastre "C_y" (o acarreo), de acuerdo al siguiente principio:

$$\begin{array}{rr} & a \\ + & b \\ \hline & C_y\,S \end{array}$$

La tabla de verdad se realiza colocando en ella los resultados de la suma binaria: "C_y", y "S", para cada combinación de los valores de "b" y "a".

A continuación se muestra dicha tabla de verdad, el esquema lógico y las expresiones correspondientes a cada salida:

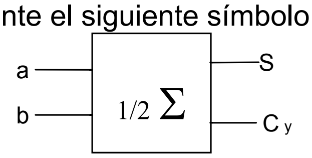
b	a	C_y	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



Se puede observar que el arrastre coincide con el producto lógico "a . b", y la suma aritmética "S" coincide con la función O exclusiva. $C_y = b \cdot a$;

$S = \bar{b} \cdot a + b \cdot \bar{a}$; $S = b \oplus a$

Podemos representar el semi sumador mediante el siguiente símbolo:



Cuando necesitamos sumar números de más de un bit, por ejemplo la suma de dos números de 4 bit: A + B ➔ $a_3 a_2 a_1 a_0 + b_3 b_2 b_1 b_0$, todas las columnas de la suma, con excepción de la primera, requerirán sumar tres bit.

$$\begin{array}{rccccc} & c_{Y3} & c_{Y2} & c_{Y1} & c_{Y0} & \\ & & a_3 & a_2 & a_1 & a_0 \\ + & & b_3 & b_2 & b_1 & b_0 \\ \hline & C_{y\acute{o}} & S_3 & S_2 & S_1 & S_0 \end{array}$$

- El **sumador - total**[23], tiene tres entradas binarias de un bit "a" y "b" y el arrastre de la suma anterior, que lo llamaremos de entrada "c_{Yi}" y las salidas: suma "S", y arrastre "$C_{y\acute{o}}$" al cual nos referiremos como de salida:

$$\begin{array}{rr} & c_{yi} \\ & a \\ + & b \\ \hline & C_{y\acute{o}}\ S \end{array}$$

La tabla de verdad se realiza colocando en ella los resultados de la suma binaria: "$C_{y\acute{o}}$" y "S", para cada combinación de los valores de "c_{Yi}", "b" y "a".

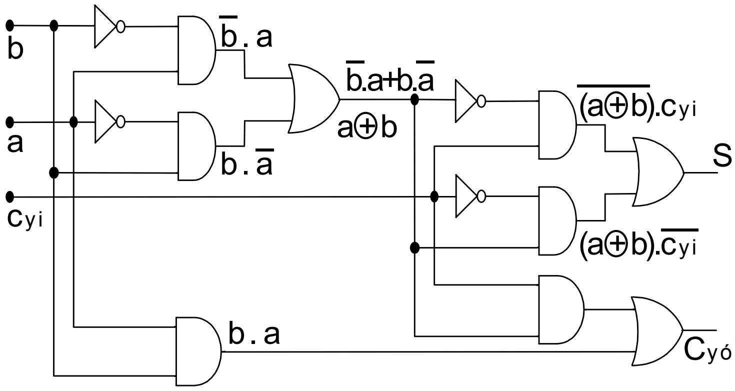
Por ejemplo si:

c_{Yi} = 0, b = 0 y a = 0, la suma será: $C_{y\acute{o}}$ = 0, y S = 0.
- Ya que 0 + 0 + 0 = 0 y por supuesto no produce arrastre.
c_{Yi} = 0, b = 1 y a = 0, la suma será: $C_{y\acute{o}}$ = 0, y S = 1.
- Ya que 0 + 1 + 0 = 1 y tampoco produjo arrastre.
c_{Yi} = 1, b = 0 y a = 1, la suma será: $C_{y\acute{o}}$ = 1, y S = 0.
- Ya que 1 + 0 + 1 = 10_2, en binario (2_{10}, en decimal).
c_{Yi} = 1, b = 1 y a = 1, la suma será: $C_{y\acute{o}}$ = 1, y S = 1.
- Ya que 1 + 1 + 1 = 11_2, en binario (3_{10}, en decimal).

23 Full adder.

A continuación se muestra dicha tabla de verdad, el esquema lógico y las expresiones correspondientes a cada salida:

C_{yi}	b	a	$C_{yó}$	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Podemos analizar esta multifunción a partir de su tabla de verdad, considerando la suma de productos canónicos. Para la suma: $S_{(C_{yi}, b, a)} = \Sigma_3\ (1, 2, 4, 7)$

$S = \overline{C_{yi}} \cdot \overline{b} \cdot a + \overline{C_{yi}} \cdot b \cdot \overline{a} + C_{yi} \cdot \overline{b} \cdot \overline{a} + C_{yi} \cdot b \cdot a$

Agrupamos aplicando el postulado recíproco de Distributividad:

$S = \overline{C_{yi}} \cdot (\overline{b} \cdot a + b \cdot \overline{a}) + C_{yi} \cdot (\overline{b} \cdot \overline{a} + b \cdot a)$;

Aplicando el concepto de X - OR: $S = \overline{C_{yi}} \cdot (b \oplus a) + C_{yi} \cdot (\overline{b \oplus a})$;

Volvemos a aplicar X - OR a "C_{yi}" y a "$b \oplus a$":

$$S = C_{yi} \oplus (b \oplus a);$$

Para el arrastre: $C_{yó\,(C_{yi}, b, a)} = \Sigma_3\ (3, 5, 6, 7)$

$C_{yó} = \overline{C_{yi}} \cdot b \cdot a + C_{yi} \cdot \overline{b} \cdot a + C_{yi} \cdot b \cdot \overline{a} + C_{yi} \cdot b \cdot a$;

Agrupamos aplicando el postulado recíproco de Distributividad:

$C_{yó} = b \cdot a \cdot (\overline{C_{yi}} + C_{yi}) + C_{yi} \cdot (\overline{b} \cdot a + b \cdot \overline{a})$;

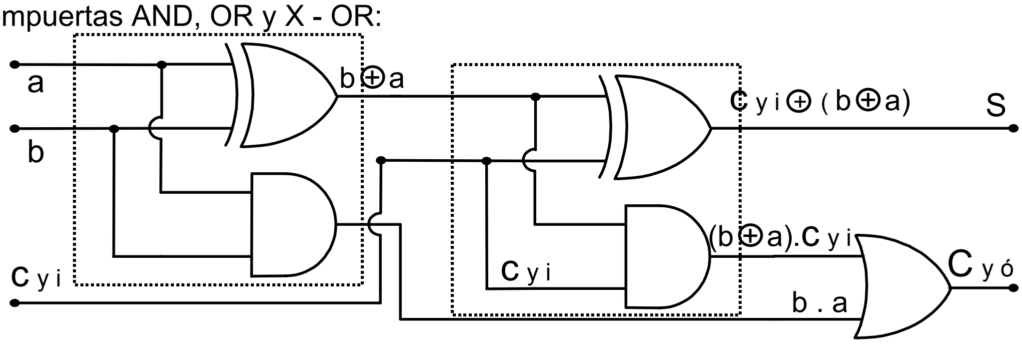
Aplicamos el postulado del elemento opuesto a $\overline{C_{yi}} + C_{yi} = 1$ y el del elemento neutro: $b \cdot a \cdot (\overline{C_{yi}} + C_{yi}) = b \cdot a \cdot 1 = b \cdot a$;

Obtenemos: $C_{yó} = b \cdot a + C_{yi} \cdot (\overline{b} \cdot a + b \cdot \overline{a})$;

Aplicando el concepto de X - OR: :

$$C_{yó} = b \cdot a + C_{yi} \cdot (b \oplus a);$$

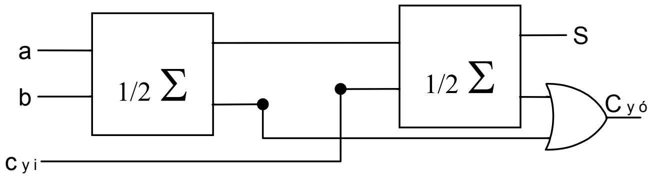
De esta forma, podremos construir el siguiente circuito equivalente al anterior, pero con compuertas AND, OR y X - OR:

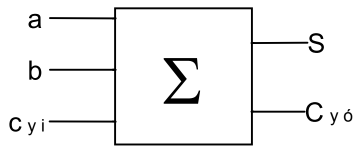


En este esquema podemos descubrir, por simple inspección, la presencia de dos semi - sumadores. El primero suma "a" con "b" y el segundo suma el resultado $b \oplus a$ con C_{yi} .

El arrastre final, por su parte, se obtiene con la suma lógica de los arrastres parciales, es decir será 1 cuando cualquiera de los arrastres parciales sea 1.

Los últimos párrafos nos permiten representar al sumador total mediante:





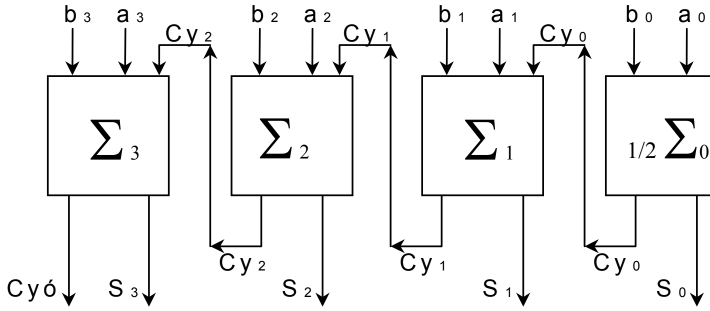
El sumador total mediante semi sumadores. El sumador total como bloque.

Probablemente, este último análisis se acerque a la forma en la que frecuentemente realizamos la operación de suma de tres dígitos. Sumamos los primeros dos y ese resultado lo sumamos al tercero, reservando siempre el arrastre en cada paso.

Para sumar números de varios bit, solemos aplicar la metodología tradicional, que transcribimos y la aplicamos al circuito esquemático simplificado:

$$\begin{array}{cccccc} & c_{Y3} & c_{Y2} & c_{Y1} & c_{Y0} \\ & & a_3 & a_2 & a_1 & a_0 \\ + & & b_3 & b_2 & b_1 & b_0 \\ \hline & C_{y\acute{o}} & S_3 & S_2 & S_1 & S_0 \end{array}$$

Tomando arrastre en serie:



Comparadores

Un comparador es un sistema que permite establecer si un número es mayor, igual o menor que otro. Se trata, además, de una multifunción.

Básicamente podría razonarse para entradas binarias de un bit "A" y "B" (aunque podrían compararse binarios de "n" bit) y por ejemplo dos salidas: igual "=" ("A = B"), y mayor que ">" ("A>B"). La salida "A es igual a B" se pone en uno cuando las entradas A y B son iguales, y la salida "A mayor a B" se pone en uno cuando el valor de A es mayor que el de B. Podría haber otra salida que informara cuando A es menor que B.

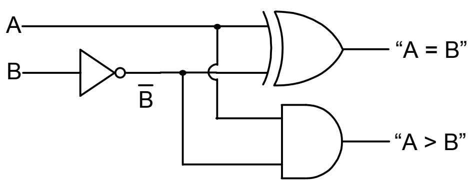
La tabla de verdad de un comparador, se puede deducir a partir el principio expuesto precedentemente:

B	A	A>B	A=B
0	0	0	1
0	1	1	0
1	0	0	0
1	1	0	1

"A > B" = $A \cdot \overline{B}$

"A = B" = $\overline{A} \cdot \overline{B} + A \cdot B$

Igualdad	"="	Mayor que	">"
Si A = B	1	Si A>B	1
Si A ≠ B	0	A<B o A = B	0



El lector podrá deducir la correspondencia entre las expresiones y el esquema lógico, aplicando álgebra de Boole.

El esquema del comparador, nos muestra que contiene un semi - sumador que realiza la suma de las variables en sus entradas, y como la variable B está invertida (complemento a 1), se obtiene la resta mediante la suma del complemento a la base menos uno (de la variable B). Por otra parte este estudio permite reconocer que una comparación no es otra cosa que una resta.

Al restar dos números, el resultado es cero si son iguales. Positivo si el primero es mayor que el segundo y negativo en el caso contrario.

Un concepto importante, es el caso del comparador de igualdad.

B	A	A=B
0	0	1
0	1	0
1	0	0
1	1	1

B	A	$B \oplus A$
0	0	0
0	1	1
1	0	1
1	1	0

Hemos reproducido a la derecha, la tabla de la X - OR.

Se concluye que el opuesto de la X - OR, es un comparador de igualdad. Por este motivo a la X - OR se la conoce también como comparador de desigualdad.

Este efecto, se consigue también negando una sola de las entradas (como se verifica en el circuito anterior, donde se hizo $\overline{B}$).

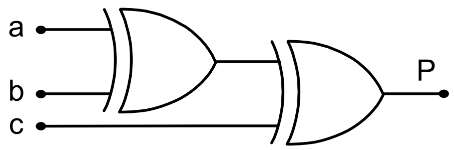
Por lo cual, para la compuerta lógica X - OR se cumple que: $\overline{B \oplus A} = \overline{B} \oplus A = B \oplus \overline{A}$.

Generadores (o detectores) de bit de paridad

Un circuito generador de bit de paridad, permite obtener un bit de paridad par (o impar).

Agregándolo a un código para su transmisión, permitirá detectar los errores posibles de los sistemas teleinformáticos. Los detectores, emplean el mismo concepto para verificar el código recibido. Se trata, en síntesis, de una aplicación de las X - OR.

Por simplicidad, supongamos un código de 3 bit, para el cual queremos generar un bit de paridad par. Esto es, la suma de los unos de un código (incluido el bit de paridad), debe ser par.



c	b	a	P
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

Esta tabla es igual a la suma del Sumador total.

Multiplexores

Un multiplexor es un sistema que permite seleccionar una de varias entradas de datos, mediante un conjunto de líneas de control.

Así, por ejemplo, si tuviéramos varias computadoras y una sola impresora, podríamos seleccionar una de ellas para realizar tareas de impresión, mientras las demás continúan con otros trabajos.

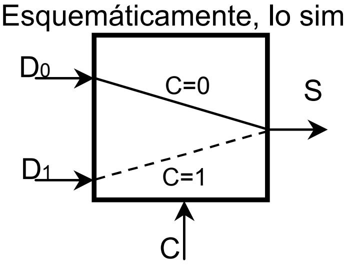
En muchas aplicaciones, aún analógicas, se emplea este concepto. Por ejemplo en un radio grabador, mediante un selector le indicamos al sistema aquello que queremos escuchar: radio o cinta. Incluso seleccionamos radio AM o FM. De esta forma, el sistema electroacústico de salida (los parlantes), quedan conectados a la fuente (datos de entrada) que hemos seleccionado mediante los controles del panel frontal o el control remoto.

Más específicamente, en un sistema digital, y para un sistema con dos posibles entradas de datos, la salida del multiplexor en función de la entrada de control es tal que coincide con la entrada D_0 si el control se encuentra en estado “0” y será igual a D_1 si la entrada de control se encuentra en “1”.

Esquemáticamente, lo simbolizamos:

C	**S**
0	D_0
1	D_1

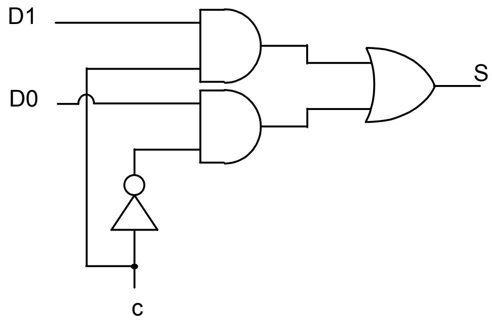
Tabla reducida.



La tabla de verdad y el circuito lógico siguiente, configuran este multiplexor:

C	D_1	D_0	**S**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Tabla completa.



$S = C \,.\, D_1 + \overline{C} \,.\, D_0$

Un multiplexor (MUX), puede tener 2^n entradas de datos para “n” terminales de control.

Un MUX de dos entradas de control tiene la siguiente tabla de verdad reducida:

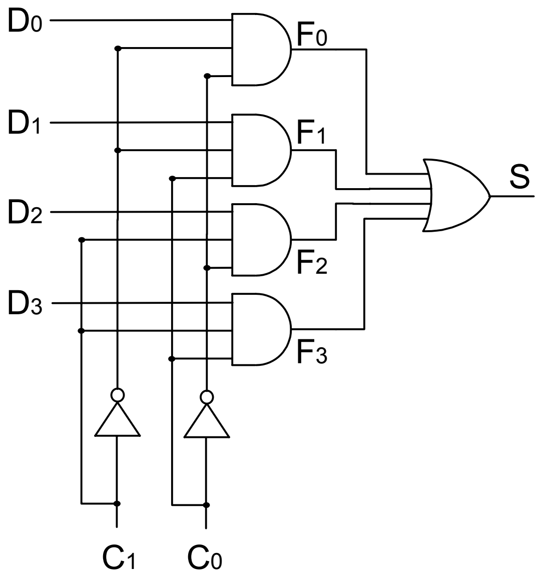
C_1	C_0	S
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

En total, disponemos de 6 entradas (cuatro de datos y dos de control). Si nos proponemos hacer la tabla completa, debemos estar preparados a escribir las 64 combinaciones binarias correspondientes.

De todas formas, la expresión Booleana es fácil de deducir a partir de la tabla reducida:

$$S = \overline{C_1} \,.\, \overline{C_0} \,.\, D_0 + \overline{C_1} \,.\, C_0 \,.\, D_1 + C_1 \,.\, \overline{C_0} \,.\, D_2 + C_1 \,.\, C_0 \,.\, D_3$$

Un circuito posible es el siguiente:



Las funciones F_0 a F_3 indicadas en el circuito, corresponden a cada uno de los sumandos de la expresión anterior.

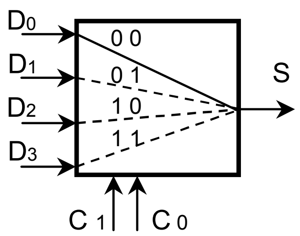
$F_0 = \overline{C_1} \,.\, \overline{C_0} \,.\, D_0$

$F_1 = \overline{C_1} \,.\, C_0 \,.\, D_1$

$F_2 = C_1 \,.\, \overline{C_0} \,.\, D_2$

$F_3 = C_1 \,.\, C_0 \,.\, D_3$

Simbólicamente, podemos dibujar:



Existen además muchas otras funciones, como los transcodificadores que convierten un código en otro (como por ejemplo la conversión de BCD GRAY a BCD 8421). Algunas funciones inversas a las vistas (como los demultiplexores, que dirigen una entrada de datos a diferentes salidas). Por mencionar solo algunas.

INTRODUCCIÓN A LOS CIRCUITOS SECUENCIALES

En un circuito combinacional como los presentados hasta ahora, la salida responde a la combinación de las variables de entrada.

En cambio en un circuito secuencial, la salida es una función de las variables de entrada y de la secuencia de valores que fueron adoptando las variables del sistema.

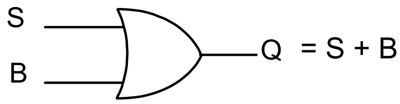
Es por este motivo que un circuito secuencial depende del estado de las variables de entrada y de la historia (secuencia) registrada como estados internos, involucrando el concepto de memoria.

Analicemos estos conceptos desde un ejemplo sencillo.

Imaginemos un sistema de alarma para un automóvil. Este deberá accionar la bocina cuando un intruso abra alguna de sus puertas, el baúl etc. .

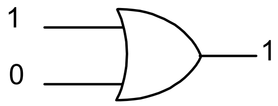
Aprovechando los interruptores dispuestos para el encendido de la luz de cortesía, se puede realizar un sistema que haga sonar la bocina cuando se abra alguna puerta, ó varias.

Esto corresponde a la función lógica “OR”, de tantas variables de entrada como aperturas tenga el vehículo. Por ejemplo para dos variables de entrada “S” y “B” (que podrían ser las puertas delanteras), y suponiendo la convención: puerta cerrada ”0”, puerta abierta “1”, bocina sonando “1”, quedaría una tabla como la siguiente:



S	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

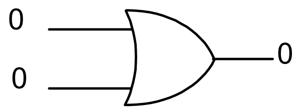
Ahora bien, al abrir la puerta, se activa la alarma



S	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

y el intruso podría salir corriendo cuando escuche el ruido ...

o cerrar la puerta,



S	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

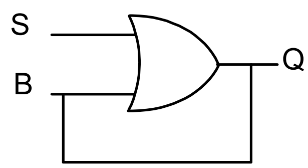
en cuyo caso dejaría de sonar.
(El sistema vuelve al estado inicial).

En estas condiciones sería razonable que, en un segundo intento, el delincuente entre y cierre rápidamente la puerta, perpetrando el delito.

Por este motivo, convendría que el sistema registre la maniobra y mantenga activa su salida (la bocina sonando) permanentemente, a partir de la primera apertura.

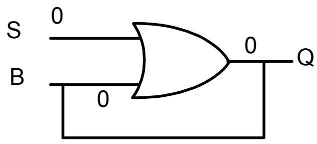
El efecto deseado, puede lograrse conectando la salida de la OR a una de sus entradas. A esta conexión se la denomina "**realimentación**". En este caso, empleamos la entrada “B”. De este modo, cuando la salida tome el valor lógico "1", la entrada adoptará ese mismo valor, manteniendo la salida en "1".

B y Q corresponden ahora a la misma conexión (digamos Q). B = Q .



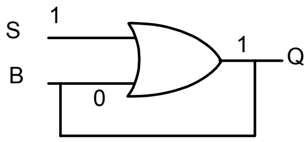
- Observe que ahora, estamos considerando una sola apertura en el vehículo, la “S” -.

Analizamos el sistema a partir del estado inicial, en que la alarma se encuentra desactivada y la puerta cerrada (todo en “0”).



S	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Al abrir la puerta, la entrada correspondiente (“S”) se pondrá en “1” provocando el cambio de la salida a “1”.

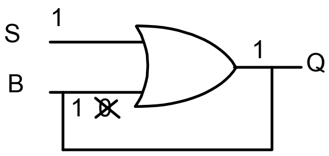


S	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

S	B *(Q_i)*	Q_f
0	0	0
0	1	1
1	0	1
1	1	1

B y Q difieren momentáneamente durante este estado de transición. Inicialmente “B = 0” (que a partir de ahora lo llamaremos estado inicial Q_i), luego “1” el estado final (Q_f). Al principio Q_i está en “0” y rápidamente cambia a “1” Q_f. Cuando en S se coloca “1”, también transcurre un breve lapso (tiempo de demora) hasta que la salida Q adopta ese estado.

Este cambio de estado en la salida, se reflejará inmediatamente a la entrada, pasando al estado resaltado en la tabla adjunta.

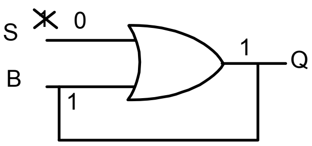


S	B *(Q_i)*	Q_f
0	0	0
0	1	1
1	0	1
1	1	1

Observe que la compuerta OR presenta “1” en la salida cuando una o más de sus entradas están en “1” por lo tanto, ...

S	B *(Q_i)*	Q
0	0	0
0	1	1
1	0	1
1	1	1

aunque se cierre la puerta (“S = 0”) que antes provocó el encendido de la alarma ...

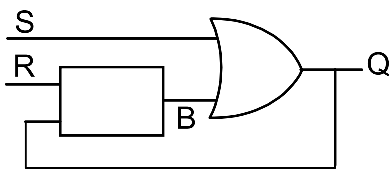


S	B *(Q_i)*	Q_f
0	0	0
0	1	1
1	0	1
1	1	1

la salida seguirá en estado ”1” pues la otra entrada ahora tiene un “1” (“Q =1”). Concepto de memoria.

El sistema descripto proporciona una solución al problema de encender y mantener la alarma, pero tiene el inconveniente de que una vez activada, seguirá sonando indefinidamente.

Para desactivarla, se podría agregar un circuito lógico en la realimentación, de manera de poder volver al estado inicial. Esto es, poner un cero en el punto indicado como “B = Q” a la entrada de la compuerta OR.

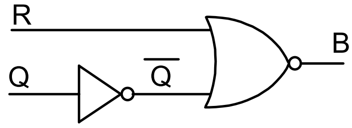


Podemos definir el funcionamiento de ese circuito lógico en la realimentación, diciendo que: a) debe presentar un "0" en los casos en que el usuario desee reiniciar el sistema (R=1, Reset) ó, b) cuando sin resetear (R = 0) la alarma no esté activada (Q = 0). Su tabla es:

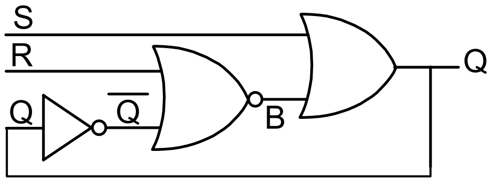
R	Q	B
0	0	0
0	1	1
1	0	0
1	1	0

Corresponde a la función NOR de las variables R con NO Q

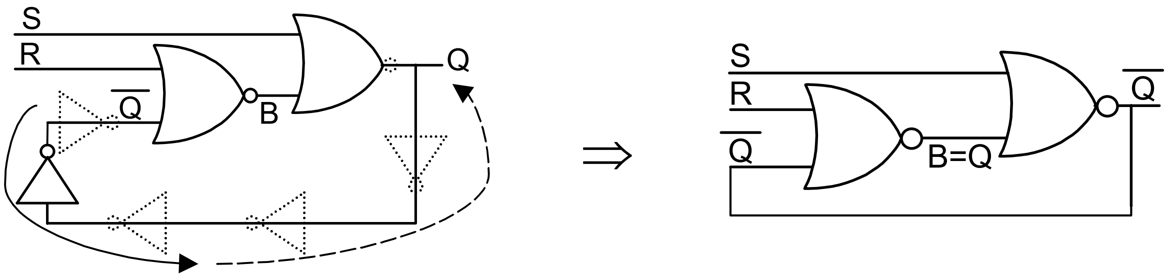
Por lo tanto: B = $\overline{R + \overline{Q}}$



El esquema completo quedaría como se indica a continuación:



Ahora, desplazamos el inversor a través de la línea de realimentación desde la entrada de la NOR hacia la salida de la OR, quedando así, dos compuertas NOR:



Adjuntamos la tabla de verdad completa.
En ella presentamos las dos variables de entrada al sistema R y S, la variable de estado interno B (Q) y la variable de salida Q.
Se observa en el esquema, una salida $\overline{Q}$. La salida original Q, se obtiene del punto B = Q .
Los estados indicados con "X" (estado lógico no relevante), corresponden a la condición en que S y R valgan "1", lo cual representa el absurdo de querer apagar la alarma cuando intentan robar la unidad.

S	R	B(Q_i)	Q_f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

En esta tabla, se presenta el caso de que pueden existir estados no definidos.

En general, si una función presenta estados no definidos y se puede elegir libremente el estado lógico resultante, diremos que se tratará de una "**Función Incompleta**".

Analicemos brevemente la tabla de verdad.

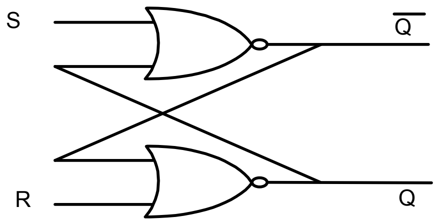
En las dos primeras combinaciones (filas), el estado final de la salida Q_f coincide con el estado anterior (inicial) Q_i . Diremos que el circuito retiene (almacena o memoriza) el estado binario que tenía, sin alterarlo.

En las dos combinaciones siguientes, el estado final de la salida Q_f , es cero independientemente del estado anterior Q_i. Esta es la operación de Reset. Esto se logra gracias a que la entrada R se encuentra en uno y la S en cero.

Las siguientes dos, Q_f está en uno, sin importar el estado en el que se hallaba antes. En este caso lo llamamos Set. Para "setear" (poner en uno) la salida del circuito, debemos asegurar un uno en la entrada Set y un cero en el Reset.

Por ello, al circuito secuencial así formado se lo conoce como **biestable R S**, reset - set, ya que **posee dos estados estables**. También se lo llama **flip flop R S**. Por definición los flip flop, poseen dos salidas: Q y $\overline{Q}$ que se obtuvieron aquí, de la salida de cada una de las NOR.

Por esta razón, en la mayoría de los textos, la forma en la cual se lo encuentra dibujado es:



El último par de combinaciones, se prohiben ya que además de lo expuesto en párrafos anteriores, en este caso impiden el cumplimiento de la definición del flip flop e incluso pueden producir efectos no deseados en su funcionamiento.

En el diseño por métodos tabulares, se colocan en los estados no definidos, una “X”. Así el diseñador podrá elegir entre “0” y “1”, resultando un circuito más simple. En las siguientes figuras, se muestran las tablas de Karnaugh para la simplificación.

S \ R Q_i	0 0	0 1	1 1	1 0
0	0 (0)	1 (1)	0 (3)	0 (2)
1	1 (4)	1 (5)	X (7)	X (6)

S \ R Q_i	0 0	0 1	1 1	1 0
0	0 (0)	1 (1)	0 (3)	0 (2)
1	1 (4)	1 (5)	X (7)	X (6)

S \ R Q_i	0 0	0 1	1 1	1 0
0	0 (0)	1 (1)	0 (3)	0 (2)
1	1 (4)	1 (5)	X (7)	X (6)

En la segunda tabla, para simplificar por medio de maxitérminos, se ha supuesto que las “X” adoptan el valor cero. La expresión simplificada es: $Q_f = (S + Q_i) . \overline{R}$.

Aplicando el teorema de doble negación y luego De Morgan:

$Q_f = \overline{\overline{(S + Q_i) . \overline{R}}}$; $Q_f = \overline{\overline{(S + Q_i)} + \overline{\overline{R}}}$ y finalmente: $Q_f = \overline{\overline{(S + Q_i)} + R}$ (todas NOR)

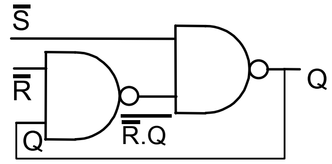
Esta última expresión coincide con el último circuito de la página anterior[24].

En la tercera tabla en cambio, para simplificar por medio de minitérminos, se ha supuesto que las “X” adoptan el valor uno. La expresión simplificada es: $Q_f = S + \overline{R} . Q_i$

Aplicando el teorema de doble negación y luego De Morgan:

$Q_f = \overline{\overline{S + \overline{R} . Q_i}}$, de esta forma: $Q_f = \overline{\overline{S} . \overline{\overline{R} . Q_i}}$ (todas NAND)

A esta última expresión le corresponde el circuito siguiente:



Se observa que las entradas Set y Reset están invertidas. Esto implica que para poner en uno la salida, el $\overline{S}$ debe estar en cero y $\overline{R}$ en uno. Para poner en cero la salida, $\overline{R}$ debe esta en cero y $\overline{S}$ en uno. Si ambas entradas $\overline{S}$ y $\overline{R}$ están en uno, el sistema memoriza (mantiene la salida en su estado anterior). Si ambas estuvieran en cero, se dará el estado prohibido.

Después de este aprendizaje, resultará sintético y esclarecedor realizar la siguiente tabla de verdad reducida de un flip flop R S, donde se muestran las cuatro combinaciones posibles de las variables de entrada R y S y la función de salida final Q_f.

S	R	Q_f
0	0	Q_i
0	1	0
1	0	1
1	1	X

Observe que cuando S y R valen “0”, la salida Q_f mantiene el valor anterior de Q (Q_i). Cuando S=1 la salida pasa a “1” y cuando R=1 la salida será “0”. Esto concuerda con la tabla completa. También queda claro el estado prohibido correspondiente a S y R simultáneamente en 1.

[24] El lector puede deducir directamente del circuito la expresión algebraica y así comprobar la validez de este análisis.

A esta altura de nuestro estudio, podremos reconocer las expresiones algebraicas y explicar el funcionamiento de otros circuitos posibles para un flip flop R S asincrónico[25].

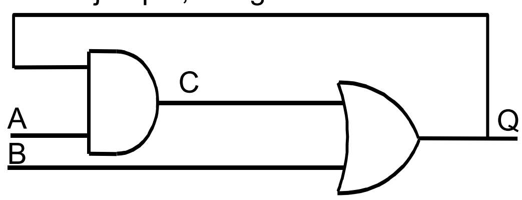
A continuación, se propone un método de análisis.

Consideraremos las características del flip flop R S mediante las cuales: a) se logra poner en uno a la salida con una cierta combinación de las entradas; b) otra combinación logrará poner en cero la salida y c) una tercera combinación mantiene el estado que tenía (memoria de un bit).

Presentado el circuito, se deduce la expresión algebraica. Luego, por medio de las características mencionadas más arriba (o empleando la tabla de verdad característica del flip flop R S) buscamos reconocer las entradas Set y Reset.

En general, veremos el esquema como un sistema con entradas y salidas:

Por ejemplo, el siguiente circuito:



- Recorremos de izquierda a derecha el circuito y encontramos $C = A . Q$
- Continuamos hasta la salida, $Q = B + C$.
- Sustituyendo el valor encontrado para C, en la expresión de Q obtenemos: $Q = B + (A . Q)$.

Ahora bien, como sabemos, la variable de salida realimentada implica que en un momento inicial tendrá un cierto valor y luego un valor final. Ya en párrafos precedentes los hemos indicado con subíndices "i" y "f" respectivamente. Por lo tanto resultará:

$Q_f = B + (A . Q_i)$. Que es la expresión algebraica que representa a este esquema.

Estamos en condiciones de reconocer la función de cada entrada, A y B.

- Observamos que A está conectada a una compuerta AND (producto lógico) y sabemos por los postulados del Álgebra de Boole (más precisamente el "5 b") que el elemento neutro del producto es 1.
- La entrada B, se encuentra conectada a una compuerta OR (suma lógica) y el elemento neutro de la suma es 0.

Un 1 en B pondrá en 1 a la salida, independientemente del estado del resto de las variables.

* Sintéticamente: $B = 1 \Rightarrow Q = 1$. Esto corresponde al Set. Podemos entonces hacer la equivalencia B = S.

Si B está en 0, la salida depende del producto $A . Q_i$, pues: $0 + A . Q_i = A . Q_i$. Si al mismo tiempo, A se encuentra en 1 (neutro del producto) resultará $1 . Q_i = Q_f$.

* Es decir, se conserva el estado anterior (memoria de un bit), con B = 0 y A = 1.

En cambio si B = 0 (neutro de la suma) y A = 0, la salida será cero $Q_f = 0 + 0 . Q_i = 0$.

Sintéticamente: $B = 0$ y $A = 0 \Rightarrow Q = 0$. Esto corresponde al Reset.

* Podemos entonces hacer la equivalencia $A = \overline{R}$, ya que la operación de Reset se logra con A = 0 lo cual corresponde al caso general R = 1 .

---

[25] Los circuitos secuenciales pueden ser sincrónicos o asincrónicos. Los circuitos sincrónicos necesitan una entrada adicional para sincronizar su funcionamiento. Nos hemos ocupado del tipo asincrónico, porque es más breve su análisis.

---

[i] Tomado de “ANÁLISIS MATEMÁTICO” de Rey Pastor, Pi Calleja y Trejo; 8ª edición, volumen I, ξ 1 - 2.