Tecnología Computacional y Estructura de Datos

Unidad II

Álgebra de Boole

El matemático inglés George Boole, en el año 1847 publicó una pequeña obra llamada “The Mathematical Analysis of Logic” planteando una técnica para representar la lógica proposicional mediante la matemática. A partir de los trabajos de Claude Shannon en 1938 esta técnica se manifestó como la más útil para implementar dispositivos que funcionen en base a expresiones lógicas, siendo ampliamente utilizada en la actualidad para la implementación de circuitos electrónicos digitales.

Las variables en el álgebra de boole toman dos valores, que equivalen a su valor de veracidad, 1 para verdadero y 0 para falso. Por esta causa el álgebra de boole es un álgebra binaria.

Operaciones

Las operaciones básicas del álgebra de Boole son las siguientes:

Operación de suma: se representa con el operador + (similar al de la suma matemática) y corresponde a la conjunción de la lógica proposicional.

$$r=p+q$$

Operación Producto: se representa con el operador . (punto) y corresponde a la disyunción de la lógica proposicional.

$$r=p.q$$

Operación de Complemento: se representa con una línea que se ubica sobre la expresión complementada, también se suele representar con un apóstrofo ('). Corresponde a la negación de la lógica proposicional.

$$r=\bar{p}$$

Postulados y Teoremas

Identidad: al operar una variable p con una constante, se obtiene como resultado el mismo valor de la variable. La constante se denomina elemento neutro. Los valores de los elementos neutro para la suma y el producto son:

$$p+0=p$$
$$p.1=p$$

Propiedad Conmutativa: el orden de las variables operadas no altera el resultado obtenido

$$p+q=q+p$$
$$p.q=q.p$$

Propiedad Asociativa: El resultado de una operación entre 3 o más miembros es independiente de como estos se agrupen.

$$p+(q+r)=(p+q)+r=p+q+r$$
$$p.(q.r)=(p.q).r=p.q.r$$

Propiedad Distributiva: la suma es distributiva con respecto al producto y viceversa.

$$p+(q.r)=(p+q).(p+r)$$
$$p.(q+r)=(p.q)+(p.r)$$

Complementación: La operación entre una variable y su complemento da como resultado el valor neutro.

$$p+\bar{p}=1$$
$$p.\bar{p}=0$$

Absorción: Existe una constante *k* que anula la variable para cada operación. Llamaremos k^1 a la constante de absorción de la suma y k^0 a la del producto.

$$p+k^1=k^1 \rightarrow p+1=1$$
$$p.k^0=k^0 \rightarrow p.0=0$$

Idempotencia: la operación entre una variable y si misma da como resultado la variable.

$$p+p=p$$
$$p.p=p$$

Ley de Involución: La doble complementación da como resultado la variable original.

$$\bar{\bar{p}}=p$$

Ley de Cancelación: En una operación suma de dos variables por el producto de una de ellas anula la otra variable. Lo mismo para el producto y la suma.

$$p+(p.q)=p$$
$$p.(p+q)=p$$

Ley de De Morgan: El complemento de un producto es igual a la suma de los complementos de cada variable. Lo mismo para el complemento de la suma con respecto al producto de los complementos.

$$\overline{p.q}=\bar{p}+\bar{q}$$
$$\overline{p+q}=\bar{p}.\bar{q}$$

Ley del Complemento: el complemento de la constante de absorción del producto es la constante de absorción de la suma y viceversa.

$$\bar{k}^1=k^0$$
$$\bar{k}^0=k^1$$

En todas las expresiones que hemos visto en este capítulo utilizamos p, q y r como variables proposicionales, pero es costumbre denominarlas en el álgebra de boole como variables a, b, c, etc.

Forma canónica

Se denomina forma canónica de una función booleana a una función en la cual aparecen todas las variables expresadas como:

Maxitérminos: en forma directa como suma de los productos de las variables puestas a 1 cuando el resultado de la función es 1, o

Minitérminos como productos de las sumas de las variables puestas a 0 cuando el resultado de la función sea 0.

Veamos un ejemplo: tenemos la función booleana $f(a,b,c)=c.(a+(\overline{a+b}))$, hagamos pues la tabla de verdad:

a	b	c	$a+b$	$\overline{a+b}$	$a+(\overline{a+b})$	$c.(a+(\overline{a+b}))$
0	0	0	0	1	1	0
0	0	1	0	1	1	1
0	1	0	1	0	0	0
0	1	1	1	0	0	0
1	0	0	1	0	1	0
1	0	1	1	0	1	1
1	1	0	1	0	1	0
1	1	1	1	0	1	1

En la tabla hemos sombreado las filas donde el resultado de la función es verdadero, la forma canónica de la función está dada por la suma de las expresiones donde el resultado es 1: $\bar{a}.\bar{b}.c+a.\bar{b}.c+a.b.c$

Tomemos otro ejemplo, supongamos la función $f(a,b,c)=\bar{a}+a.(b+\bar{c}.\bar{b})$ y desarrollemos la tabla de verdad:

a	b	c	$\bar{b}.\bar{c}$	$b+\bar{b}.\bar{c}$	$a.(b+\bar{b}.\bar{c})$	$\bar{a}+a.(b+\bar{c}.\bar{b})$
0	0	0	1	1	0	1
0	0	1	0	0	0	1
0	1	0	0	1	0	1
0	1	1	0	1	0	1
1	0	0	1	1	1	1
1	0	1	0	0	0	1
1	1	0	0	1	1	0
1	1	1	0	1	1	1

Ahora si armamos la expresión canónica de maxitérminos como en el ejemplo anterior esta nos quedará con 7 términos:

$$f(a,b,c)=\bar{a}.\bar{b}.\bar{c}+\bar{a}.\bar{b}.c+\bar{a}.b.\bar{c}+\bar{a}.b.c+a.\bar{b}.\bar{c}+a.\bar{b}.c+a.b.c$$

en cambio es más simple expresarla en la forma canónica de minitérmino con el único término que da como resultado 0:

$f(a,b,c)=\bar{a}+\bar{b}+c$

Observe en el término resultante, que se toma la suma de los complementos de todas las variables de entrada donde el resultado es 0.

Veamos un último ejemplo: $f(a,b)=a.b.+\bar{a}.\bar{b}=(\bar{a}+b).(a+\bar{b})$ estas dos formas canónicas son equivalentes. Hagamos la tabla de verdad:

a	b	$a.b$	$\bar{a}.\bar{b}$	$a.b.+\bar{a}.\bar{b}$	$\bar{a}+b$	$a+\bar{b}$	$(\bar{a}+b).(a+\bar{b})$
0	0	0	1	1	1	1	1
0	1	0	0	0	1	0	0
1	0	0	0	0	0	1	0
1	1	1	0	1	1	1	1

Ahora demostremos utilizando el álgebra que las formas canónicas son equivalentes:

$f(a,b)=a.b+\bar{a}.\bar{b}=(\bar{a}+b).(a+\bar{b})$

Distribuyo la segunda expresión: $(\bar{a}+b).(a+\bar{b})=\bar{a}.a+\bar{a}.\bar{b}+b.a+b.\bar{b}$

Veamos que podemos simplificar: por complementación, los términos $\bar{a}.a=0$ y $b.\bar{b}=0$, entonces nos queda:

$f(a,b)=0+\bar{a}.\bar{b}+b.a+0$

Pero todavía podemos seguir simplificando: por la ley de absorción del producto con la constante 0, y de paso ordenando los términos nos queda:

$f(a,b)=a.b+\bar{a}.\bar{b}$

Lo que es exactamente lo mismo que el primer término de la igualdad.

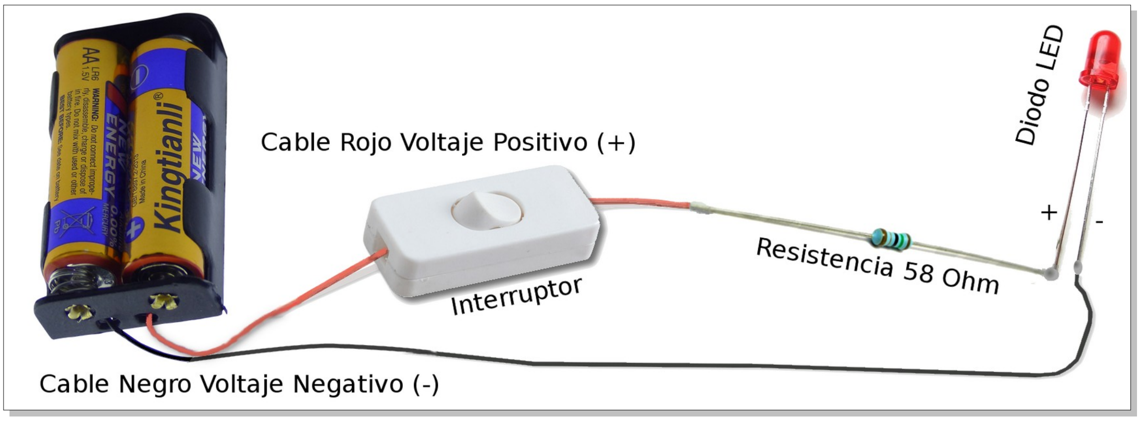
Pero, ¿cuál es la utilidad de expresar una función booleana en su forma canónica? Más adelante veremos que los procedimientos de simplificación necesitan ejecutarse sobre estas.

El Sistema Binario

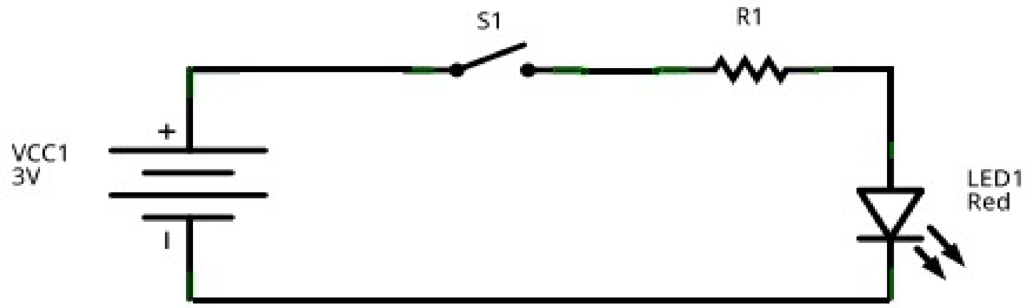
Hemos visto un resumen de lógica en las unidades anteriores y hemos observado que la lógica toma como objeto de estudio la veracidad de expresiones y el razonamiento. Pues bien, concedamos por un momento que un razonamiento no es más que un proceso, automatizable en gran medida que se ejecuta sobre variables lógicas o sea sobre valores indeterminados que pueden tomar un valor de sólo dos posibles: verdadero o falso.

Para representar una variable lógica podemos tomar un objeto que podamos manipular para que tome dos estados distintos, el objeto ideal sería una lámpara que puede estar encendida cuando querramos representar un valor verdadero y apagada cuando querramos representar un falso.

El esquema físico de nuestro artefacto representado por un esquema eléctrico podría ser:



El interruptor representa la variable lógica, por convención tomaremos el interruptor encendido como verdadero y el interruptor apagado como falso. El esquema electrónico del circuito realizado se representa de la siguiente forma:



La interpretación es bastante simple

S1 es el interruptor, cuando está abierto no conduce corriente, por lo tanto el LED no se enciende. Cuando se presiona el interruptor, circula corriente por el circuito encendiendo el LED, en ese caso se dice que el interruptor está **cerrado**, en el caso de que no esté presionado se denomina **abierto**. La resistencia R1 sirve como limitador de corriente para no quemar el LED. En un LED normal consideramos como referencia una caída de voltaje de 2,1 V (volts) y un consumo típico de 1,5 mA (miliamperios), por lo tanto aplicando la ley de ohm *I = V / R* (Intensidad = Voltaje / Resistencia) se puede calcular la resistencia necesaria para que el LED se ilumine.

Despejando la fórmula obtenemos:

$$resistencia=\frac{voltaje\ de\ la\ fuente\ de\ alimentación-voltaje\ del\ LED}{Intensidad\ para\ iluminar\ el\ LED}$$

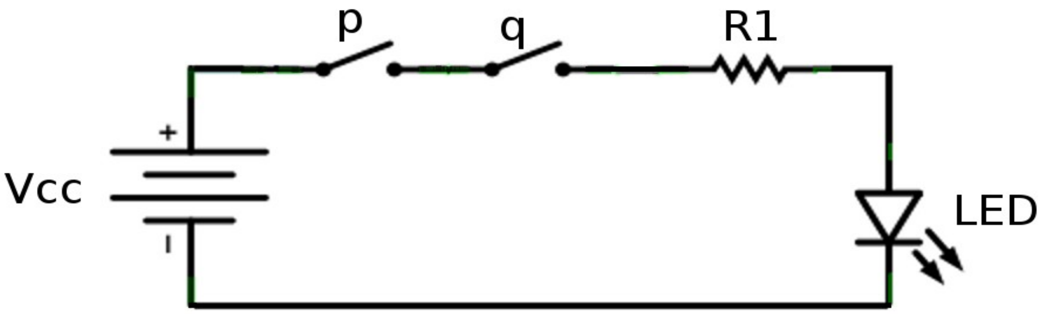
$$resistencia=\frac{3V-2{,}1V}{15mA}=\frac{0{,}9V}{0{,}015A}=60\Omega$$

que produce una resistencia de 60 ohmios, elegimos entonces una resistencia cuyo valor comercial está estandarizado en 58 ohms por ser la más cercana.

Ahora estudiemos las operaciones lógicas modificando nuestro circuito básico.

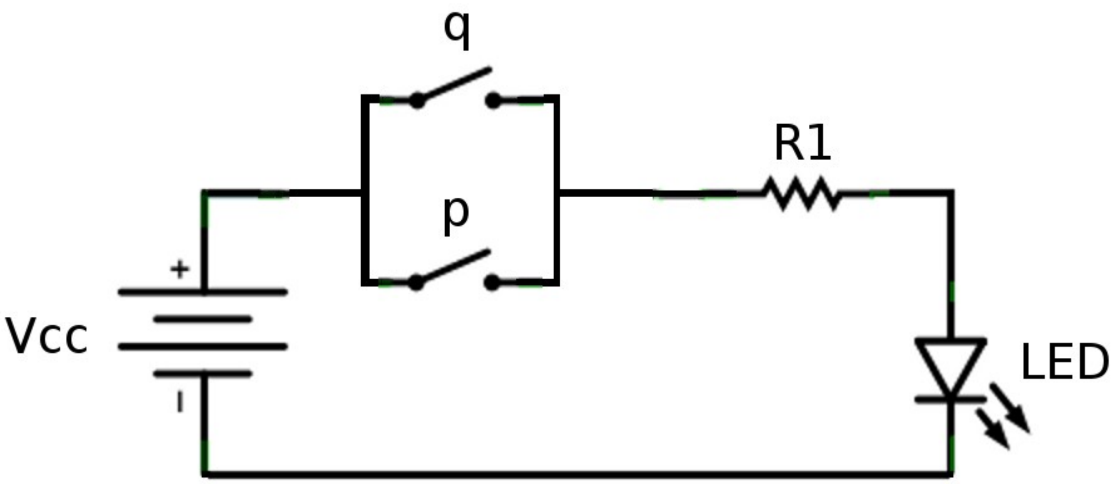
Operación de funciones lógicas en circuitos digitales

Si queremos representar bajo un circuito eléctrico la operación de disyunción, necesitamos dos interruptores, uno para cada variable (o premisa). La forma de modificar el diagrama electrónico se representa a continuación:



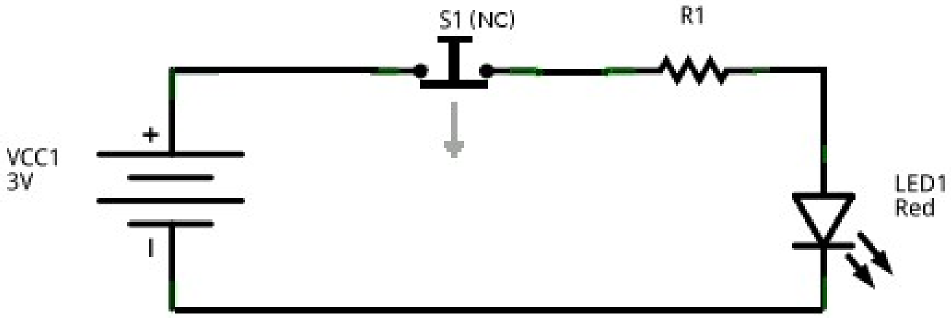
Los interruptores p y q pueden estar pulsados o no, cuando ambos están cerrados (ambos son verdaderos) el LED se enciende (verdadero). Cuando alguno de los dos pulsadores (o ambos) se encuentra abierto (falso), el LED permanece apagado (falso). La organización de los interruptores, uno a continuación del otro (podemos decir en la misma vía de corriente) se denomina en **serie**.

Para una conjunción, debemos modificar el circuito en la siguiente forma:



Cuando uno o ambos interruptores se encuentra cerrado el LED se enciende representando el valor verdadero. La única forma en que el LED quede apagado representando el valor falso es cuando ambos pulsadores se encuentran abiertos (en falso). A esta organización de interruptores, cuando se generan vías alternativas de circulación de corriente se la denomina en **paralelo**.

Vamos a introducir ahora un nuevo elemento a nuestro circuito para representar el operador de negación, simplemente consideramos que el interruptor funciona en el sentido inverso de los que hemos representado hasta ahora, todos los interruptores utilizados son del tipo “normal abierto”, esto significa que si no presionamos el interruptor este se encuentra interrumpido. En cambio el interruptor que utilizaremos ahora es del tipo “normal cerrado”, esto significa que el interruptor está siempre en contacto excepto cuando se lo presiona.



Entonces un verdadero en el interruptor S1, provoca que el circuito se abra y en consecuencia apagando el LED que pasa a estado falso.

El transistor

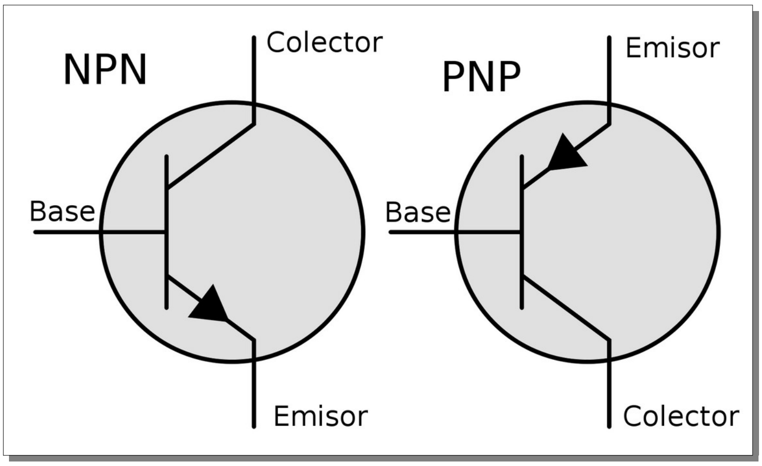
Vamos a realizar un breve comentario acerca de una de las tecnologías fundamentales en la electrónica moderna.

Como vimos en el capítulo anterior, los estados de las variables se simbolizan mediante pulsadores mecánicos, es obvio que estos requieren de la intervención directa del usuario.

Durante el año 1947 en los laboratorios Bell de USA, los científicos John Bardeen, Walter Houser Brattain y William Bradford Shockley diseñaron un dispositivo cuyo objetivo era reemplazar a las válvulas de vacío de tipo triodo. Se trata de un dispositivo semiconductor realizado sobre un sustrato de silicio (los más extendidos) o germanio, que se ha contaminado (dopado) en tres partes adyacentes con distintos materiales específicos provocando 2 superficies de unión entre las partes denominadas junturas.

La parte intermedia (base) controla la corriente que circula entre las partes de los extremos. Una de las partes se denomina colector y la otra emisor. Si hay una corriente incidiendo en la base se habilita el paso de una corriente proporcional entre el colector y el emisor. Por una parte esta proporcionalidad permite que un transistor se utilice como amplificador de señal en un factor determinado por los materiales de construcción que se denomina factor de ganancia. Por otra parte los transistores tienen un límite superior de ganancia llamado saturación, que hace que cualquier señal que alcance el límite superior permita el paso máximo de señal. Cuando un transistor funciona más allá del factor de saturación sirve como simple interruptor.

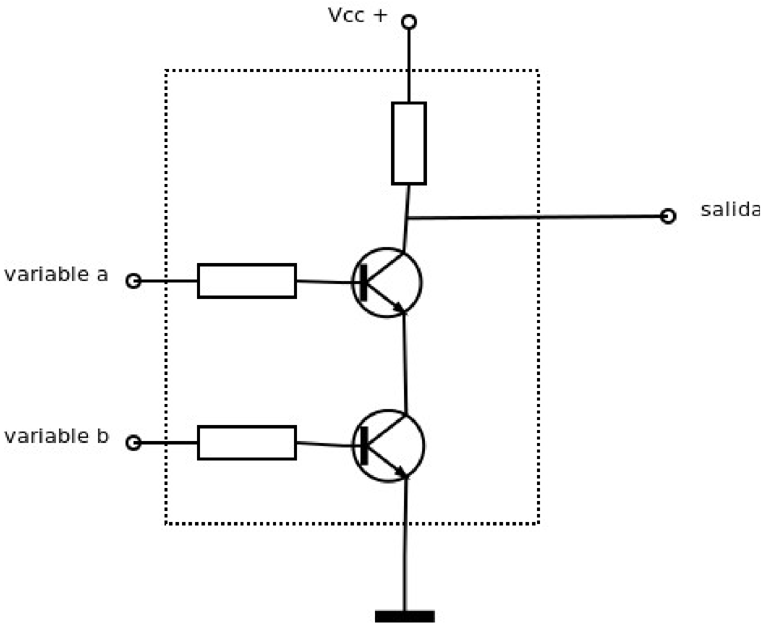
Hay dos tipos principales de transistor entre los transistores más comunes llamados bipolares, el tipo NPN y el tipo PNP. La diferencia principal entre un tipo y otro radica en el sentido en el cual circula la corriente de la base hacia el emisor en modo activo, siendo desde el positivo en el NPN y desde la masa (o también desde el negativo) en el tipo PNP. Por lo general el tipo más común es el transistor NPN



Entonces su función dentro del campo de la electrónica digital es como un interruptor que depende de una señal para habilitar el valor 1 o 0. A las representaciones electrónicas de estos dispositivos lógicos se las denomina compuertas lógicas.

Las compuertas lógicas (logic gates) se nombran por la función que cumplen, en inglés: NOT, AND, OR y XOR, además de compuertas combinadas como NAND, NOR, etc.

El esquema de una compuerta NAND puede ser el siguiente:

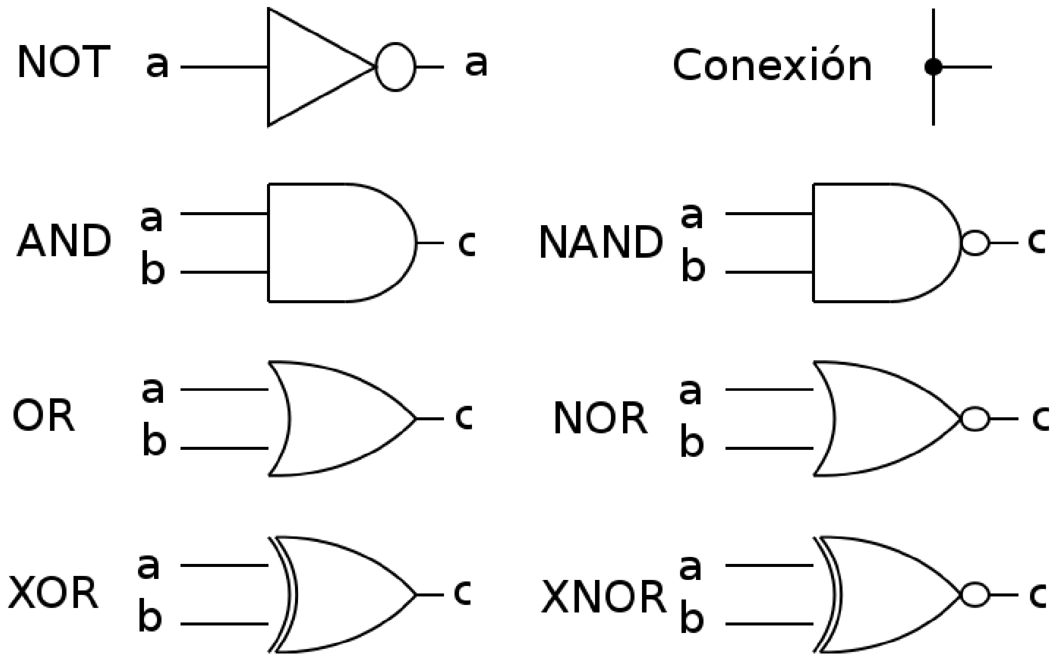


Se puede demostrar que cualquier compuerta lógica se puede armar a partir de compuertas NAND, por lo tanto es el diseño de circuito electrónico más común.

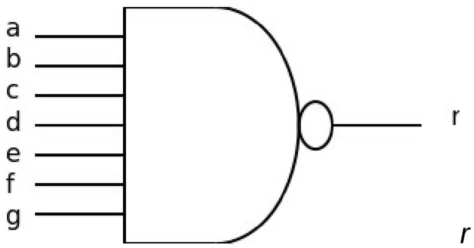
Compuertas lógicas

Como mencionábamos antes las compuertas lógicas son dispositivos electrónicos utilizados para armar circuitos electrónicos que permitan realizar operaciones lógicas con cargas eléctricas.

Las compuertas digitales se representan mediante un diagrama esquemático que permite armar diseños electrónicos complejos teniendo en vista solo la funcionalidad del circuito y no la estructura de componentes físicos. Los símbolos utilizados para los diagramas son los siguientes:

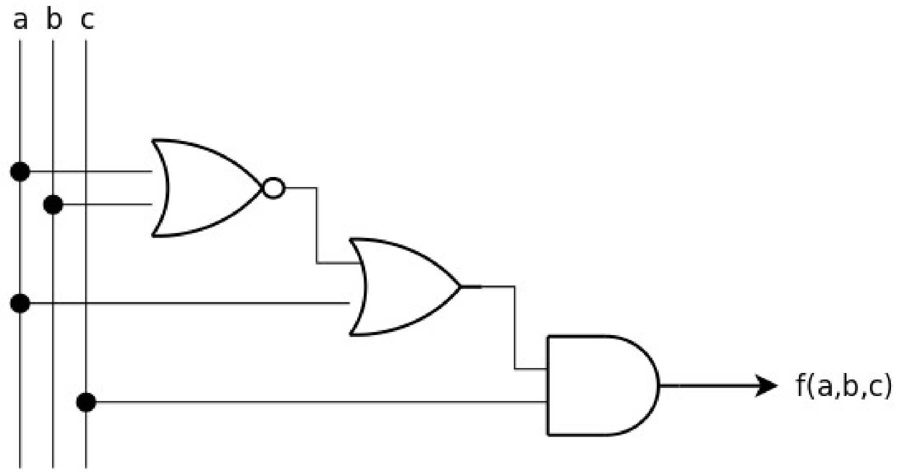


No siempre una compuerta tiene solo 2 entradas, se puede representar con múltiples entradas como por ejemplo esta compuerta NAND de 7 entradas.



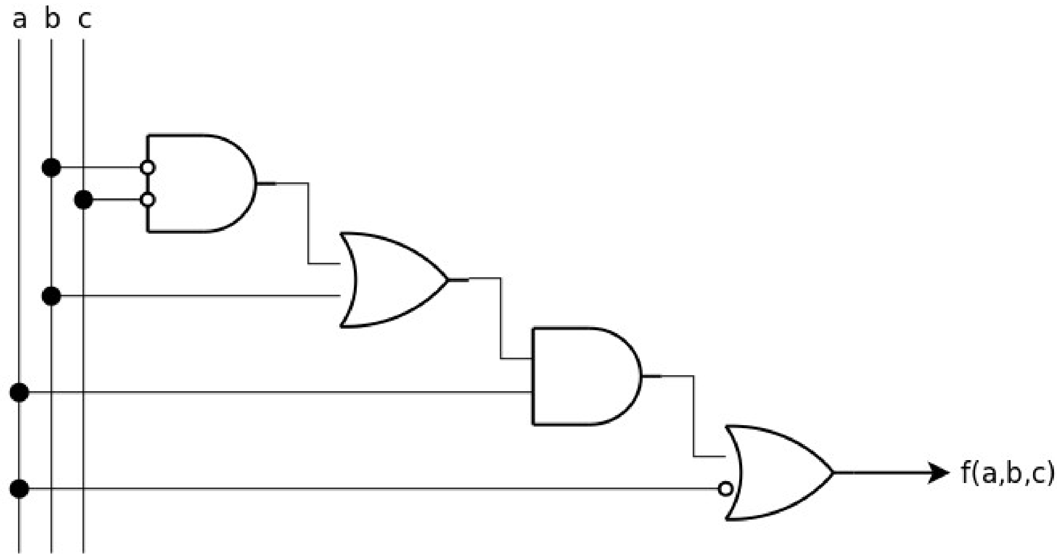
$r = \overline{a.b.c.d.e.f.g}$

Representemos una de las funciones binarias vistas en la unidad anterior: $f(a,b,c)=c.(a+(\overline{a+b}))$, así como lo fuimos desarrollando en la tabla de verdad, la resolución se hace desde las expresiones simples (las más interiores, las que están entre paréntesis) a las de mayor complejidad.

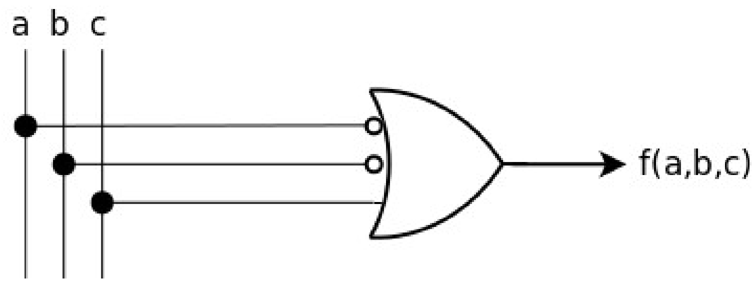


Desarrollemos ahora las otras funciones:

$f(a,b,c)=\bar{a}+a.(b+\bar{c}.\bar{b})$



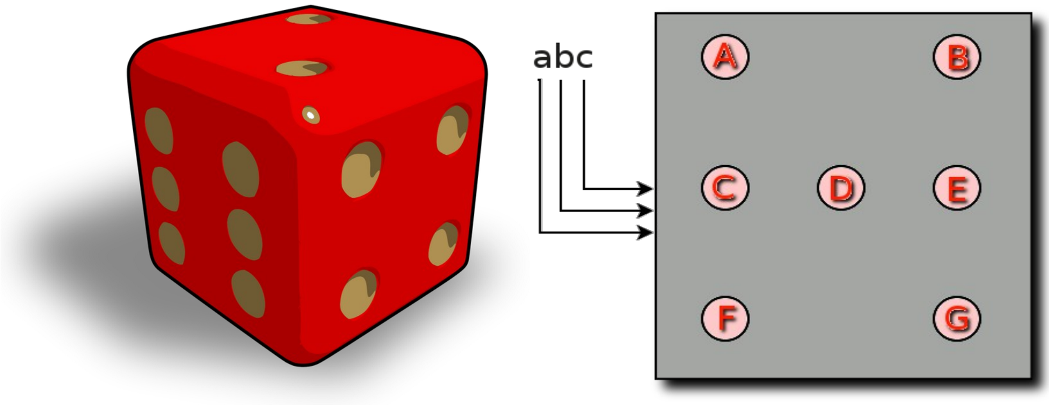
$f(a,b,c)=\bar{a}+\bar{b}+c$



Técnicas de Simplificación de Circuitos Digitales

Vamos a ver un ejemplo de circuito digital y analizaremos su simplificación mediante los mapas de Karnaugh.

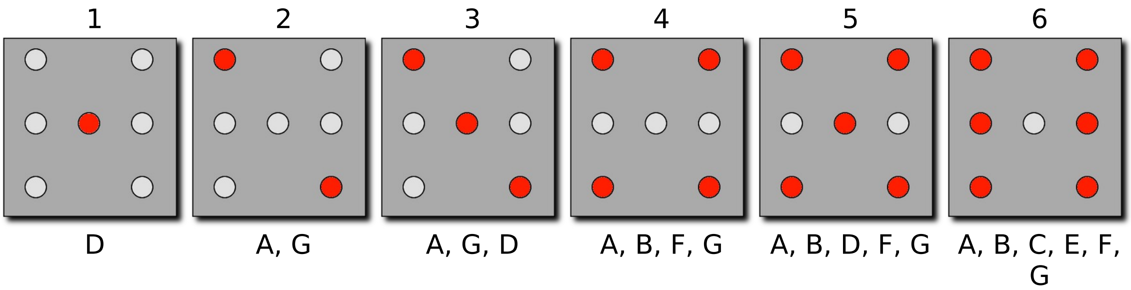
Supongamos que queremos representar un valor binario mediante 7 LEDs que simulan la cara de un dado. Hagamos un esquema de la distribución e identifiquemos cada LED.



El esquema muestra 3 entradas: a, b y c, ya que para representar 6 valores es necesario tener al menos 3 dígitos binarios, con lo que se podrían representar 8 valores en total, desde el 0 hasta el 7. Recordemos que para un dado el 0 y el 7 no son valores representables, entonces los LEDs deberían permanecer todos apagados.

A cada LED le asignamos un nombre de expresión (en mayúsculas) para identificar su esquema lógico.

Cuál sería entonces el funcionamiento esquemático del artefacto para cada valor admitido:



Hagamos entonces la tabla de verdad para analizar con qué combinaciones válidas debería encenderse cada uno de los leds:

Valor	a	b	c	A	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0	0
2	0	1	0	1	0	0	0	0	0	1
3	0	1	1	1	0	0	1	0	0	1
4	1	0	0	1	1	0	0	0	1	1
5	1	0	1	1	1	0	1	0	1	1
6	1	1	0	1	1	1	0	1	1	1
7	1	1	1	0	0	0	0	0	0	0

Cada una de las columnas corresponde a una función binaria que define la condición de encendido de cada LED. Si tomamos solamente la función binaria para el LED A la expresión binaria sería:

$$f(a,b,c)=\bar{a}.b.\bar{c}+\bar{a}.b.c+a.\bar{b}.\bar{c}+a.\bar{b}.c+a.b.\bar{c}=led\,A$$

Podemos simplificarla algebraicamente: tomamos factor común not a y a:

$$A=\bar{a}.(b.\bar{c}+b.c)+a.(\bar{b}.\bar{c}+\bar{b}.c+b.\bar{c})$$

Ahora que tenemos dos subexpresiones, en la primera podemos distribuir el factor b, y en la segunda el factor b negado con lo que nos quedaría:

$$A=\bar{a}.(b.(c+\bar{c}))+a.(\bar{b}.(\bar{c}+c)+b.\bar{c})$$

Entonces por complementación c + $\bar{c}$ se reemplaza por 1 .

$$A=\bar{a}.(b.1)+a.((\bar{b}.1)+b.\bar{c})$$

Simplificando las constantes:

$$A=\bar{a}.b+a.(\bar{b}+b.\bar{c})$$

Ahora esto ya se complica un poco más, volvamos a hacer la distributiva al segundo miembro:

$$A=\bar{a}.b+a.\bar{b}+a.b.\bar{c}$$

Por definición el primer y segundo factor corresponden a una operación xor. Entonces por último la función quedaría escrita como:

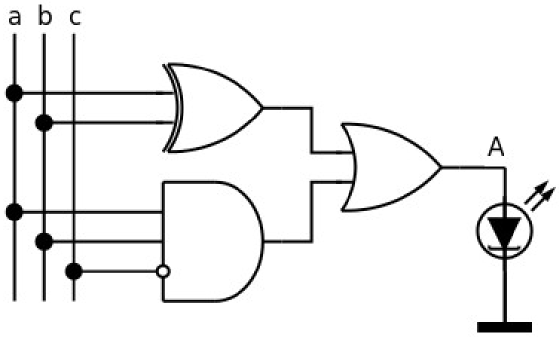
$$A=(a\oplus b)+a.b.\bar{c}$$

Obviamente. el operador ⊕ corresponde a la operación `XOR`.

Corroboremos con la tabla de verdad la equivalencia de la nueva función simplificada:

Valor	a	b	c	$(a\oplus b)$	$a.b.\bar{c}$	A
0	0	0	0	0	0	0
1	0	0	1	0	0	0
2	0	1	0	1	0	1
3	0	1	1	1	0	1
4	1	0	0	1	0	1
5	1	0	1	1	0	1
6	1	1	0	0	1	1
7	1	1	1	0	0	0

Perfecto, las funciones son equivalentes. El circuito entonces quedaría de la siguiente forma:



Pero, el método algebraico requiere de mucha práctica y no siempre tenemos una garantía de llegar a una mínima expresión, eso depende de nuestras capacidades algebraicas.

Mapas de Karnaugh

Inventado por el ingeniero en telecomunicaciones **Maurice Karnaugh**, es un método para simplificar expresiones de hasta 4 variables. La idea es representar sobre un arreglo tabular todos los valores de las variables y su resultado de verdad en las intersecciones de fila y columna. Para esto partimos de la expresión canónica o transcribiendo los valores desde la tabla de verdad.

A continuación desarrollaremos las estructuras de tablas para 2, 3 y 4 variables. Obsérvese que los términos no se ordenan por su valor, sino agrupando en las proximidades los valores donde exista un cambio de una única variable, como por ejemplo utilizando el código gray.

En la clase anterior cuando se definió la forma canónica, se mencionó que podían definirse dos formas canónicas de la misma función binaria:

- La forma más simple denominada forma directa es expresarla como una suma de productos, utilizando las expresiones que dan como resultado 1. A cada uno de los términos de esta expresión se los denomina **maxitermino**.
- La otra forma denominada forma inversa o complementaria se expresa como un producto de la suma de los inversos de las variables en todas las expresiones cuyo resultado sea 0. Cada una de las subexpresiones se denomina **miniterminos**.

Una vez transcriptos los valores de la tabla de verdad al cuadro, buscaremos grupos de 1 o 0 adyacentes, luego de agrupar veremos cual es la menor cantidad de grupos si de ceros o de unos. Los grupos de 0 corresponden a **minitérminos** y los grupos de unos corresponden a **maxitérminos**. Si elegimos los grupos de unos, donde se agrupan 8 "unos" adyacentes se pueden eliminar 3 variables complementarias, donde se agrupan 4 "unos" son dos variables complementarias y donde se agrupan 2 "unos" es una variable complementaria la que se elimina mediante conjunciones. En cambio si elegimos los ceros corresponden a simplificaciones de variables complementarias en disyunciones.

Para 2 variables:

	a	$\bar{a}$
b		
$\bar{b}$		

Para 3 variables:

	ab	$a\bar{b}$	$\bar{a}\bar{b}$	$\bar{a}b$
c				
$\bar{c}$				

Para 4 variables:

	ab	$a\bar{b}$	$\bar{a}\bar{b}$	$\bar{a}b$
cd				
$c\bar{d}$				
$\bar{c}\bar{d}$				
$\bar{c}d$				

Ahora utilizaremos el cuadro para 3 variables y transcribiremos de la tabla de verdad todos los valores de la columna de la función binaria que enciende el LED A:

	ab	$a\bar{b}$	$\bar{a}\bar{b}$	$\bar{a}b$
c	0	1	0	1
$\bar{c}$	1	1	0	1

Según vemos por un lado hay tres grupos de 2 "unos" (color celeste) o 3 maxiterminos y por el otro un grupo de 2 "ceros" más uno solitario (amarillo) que nos daría dos miniterminos. Entonces pareciera más fácil simplificar los ceros.

El grupo de dos ceros corresponde a las expresiones a+b+c y a+b+$\bar{c}$ (recuerde que cada expresión es la suma de los complementos de las variables) y tiene como complementarias a la variable C, por lo tanto la expresión resultante sería a+b, luego el término que está sólo sería $\bar{a}+\bar{b}+\bar{c}$.

$\bar{a}+\bar{b}+\bar{c}$

	ab	$a\bar{b}$	$\bar{a}\bar{b}$	$\bar{a}b$
c	0	1	0	1
$\bar{c}$	1	1	0	1

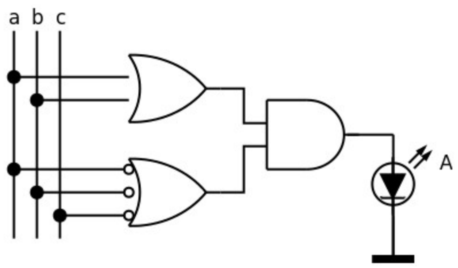
a+b

Entonces la expresión simplificada sería $A=(a+b).(\bar{a}+\bar{b}+\bar{c})$ Pero esta expresión es distinta a la calculada algebraicamente, ¿será entonces una función equivalente?

Hagamos la tabla de verdad:

Valor	**a**	**b**	**c**	**(a+b)**	$(\bar{a}+\bar{b}+\bar{c})$	**A**
0	0	0	0	0	1	0
1	0	0	1	0	1	0
2	0	1	0	1	1	1
3	0	1	1	1	1	1
4	1	0	0	1	1	1
5	1	0	1	1	1	1
6	1	1	0	1	1	1
7	1	1	1	1	0	0

Pues bien, hemos encontrado otra función simplificada cuya representación mediante compuertas lógicas es la siguiente:



Hagamos una última interpretación, ¿que tal si tomamos los 3 grupos de unos? Pues entonces obtendríamos una expresión como: $A = a.\bar{b} + \bar{a}.b + a.\bar{c}$

$a.\bar{b}$

	ab	$a\bar{b}$	$\bar{a}\bar{b}$	$\bar{a}b$
c	0	1	0	1
$\bar{c}$	1	1	0	1

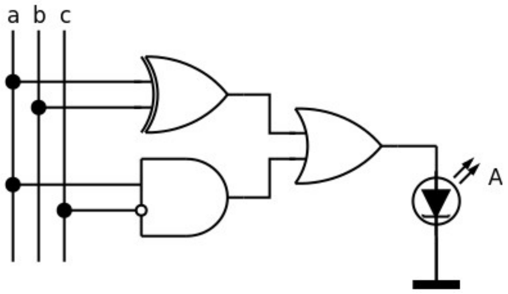
$\bar{a}.b$

$a.\bar{c}$

Habíamos analizado que los 2 primeros factores representan una operación XOR, pero el tercer factor no es similar al que habíamos reducido algebraicamente. ¿Qué habrá pasado entonces? Habría que hacer las tablas de verdad para analizarlo visualmente, cosa que dejamos para que forme parte de la práctica, pero baste decir que al eliminar una variable (a o b) de la última expresión el resultado es una nueva fila que ya está incluida en la expresión del XOR, por eso no afecta al resultado final. Entonces la expresión simplificada queda:

$$A=(a \oplus b)+a\,.\,\bar{c}$$

Y su representación sería:



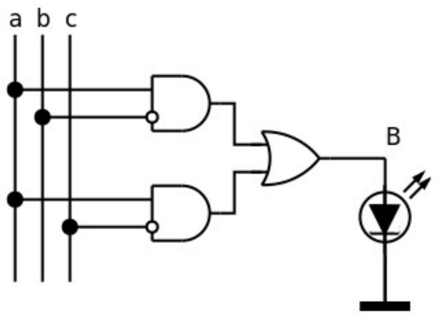
Para seguir practicando hagamos la simplificación del LED B. Recordemos la tabla de verdad:

Valor	a	b	c	A	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0	0
2	0	1	0	1	0	0	0	0	0	1
3	0	1	1	1	0	0	1	0	0	1
4	1	0	0	1	1	0	0	0	1	1
5	1	0	1	1	1	0	1	0	1	1
6	1	1	0	1	1	1	0	1	1	1
7	1	1	1	0	0	0	0	0	0	0

Este parece más simple. Hagamos el mapa de Karnaugh:

	ab	$a\bar{b}$	$\bar{a}\bar{b}$	$\bar{a}b$
c	0	1	0	0
$\bar{c}$	1	1	0	0

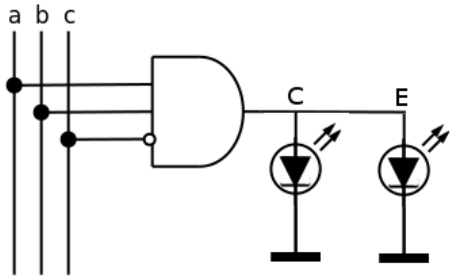
Simplificando son 2 las expresiones resultantes: B= $a.\bar{b}$ + $a.\bar{c}$, hagamos la representación mediante compuertas:



Sigamos con la simplificación para obtener todo el circuito. En el caso del LED siguiente, el C podemos ver que su expresión es similar al caso del LED E, ambos se ponen en estado verdadero (encendiendo el led) cuando el valor de a,b,c corresponde a un número decimal 6, o en binario 110. Entonces hay una única expresión canónica que define la función, sea como miniterm o como maxiterm:

$$C=E=a.b.\bar{c}=\bar{a}+\bar{b}+c$$

Su representación entonces el diagrama será



Así como el LED C tiene una función lógica equivalente al LED E, lo mismo sucede con los LEDs A y G y con los LEDs B y F.

Sigamos con el LED D, el último que falta definir, hagamos el mapa de Karnaugh:

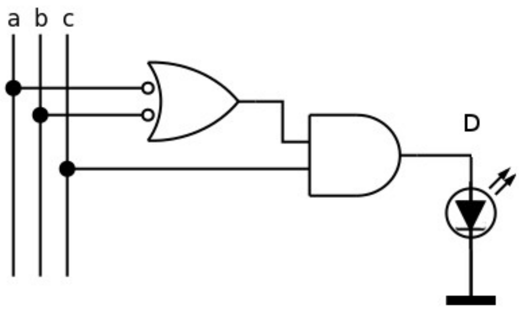
	ab	$a\bar{b}$	$\overline{ab}$	$\bar{a}b$
c	0	1	1	1
$\bar{c}$	0	0	0	0

Según el mapa definido, la expresión simplificada sería: $D=c.(\bar{a}+\bar{b})$

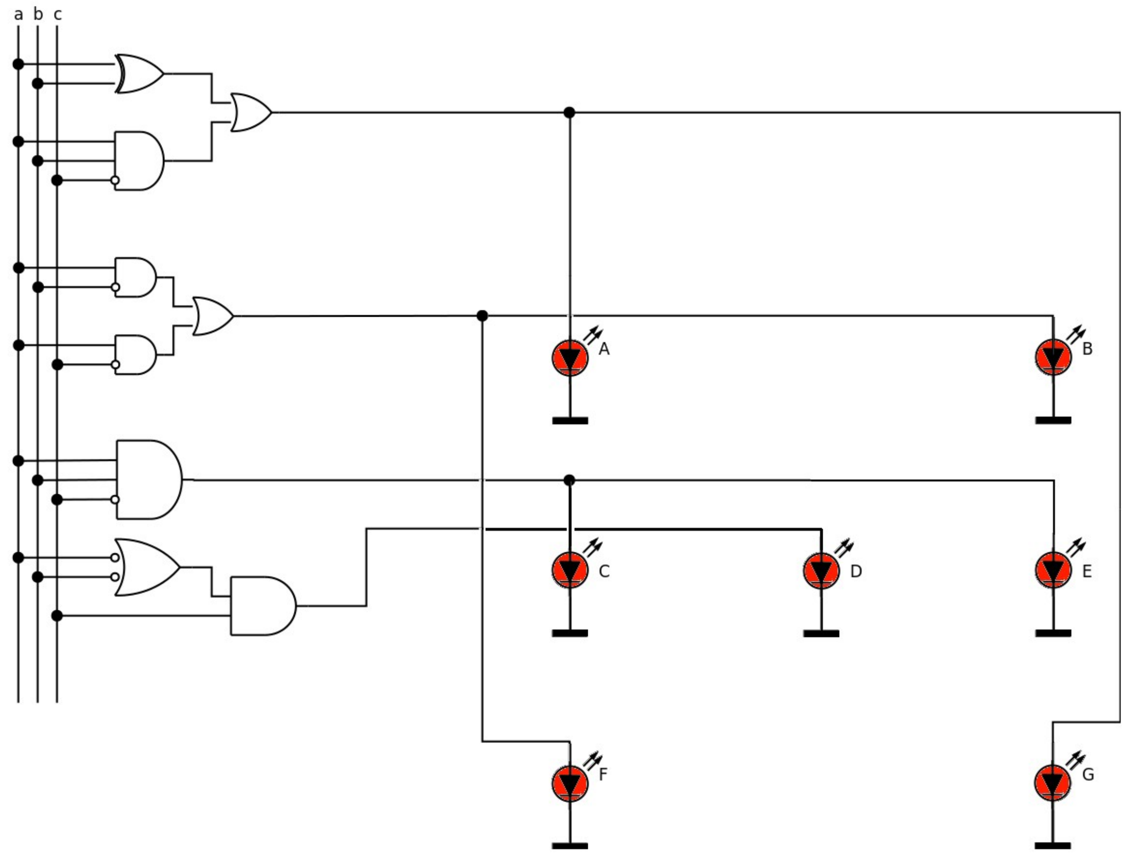
Simplificando algebraicamente quedan las siguientes expresiones:

- tomando los maxiterms $D=\bar{a}.\bar{b}.c+\bar{a}.b.c+a.\bar{b}.c$
- aplicando *c* como factor común $D=c.(\bar{a}.\bar{b}+\bar{a}.b+a.\bar{b})$
- duplicando el subtérmino $\bar{a}.\bar{b}$ y agrupando para poder aplicar factor común $D=c.(\boxed{\bar{a}.\bar{b}+\bar{a}.b}+\boxed{\bar{a}.\bar{b}+a.\bar{b}})=c.(\bar{a}.(\cancel{\bar{b}+b})+\bar{b}.(\cancel{\bar{a}+a}))$
- Finalizando con $D=c.(\bar{a}+\bar{b})$ que es la misma expresión.

Entonces el diagrama de compuertas quedaría así:



Ahora podemos representar todo el esquema lógico del dado electrónico reuniendo sus términos

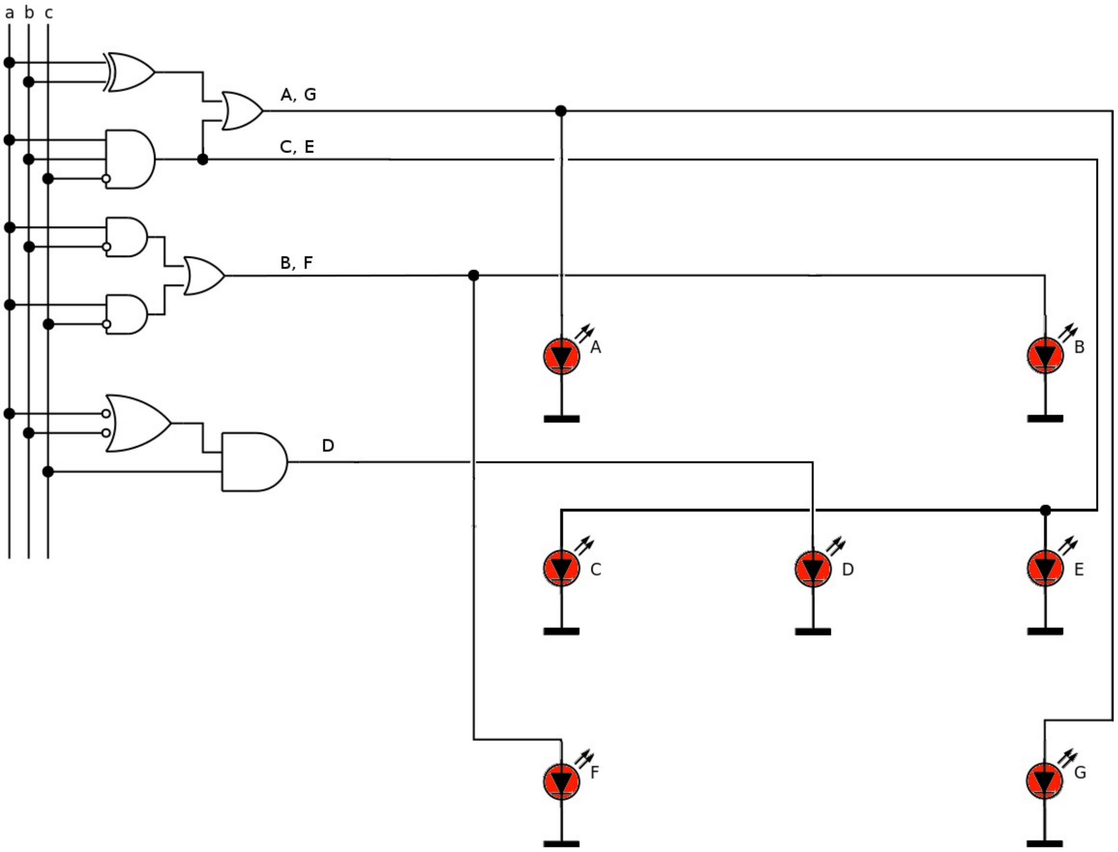


Peroooo...

Todavía queda un último detalle. Si se compara la expresión utilizada para las expresiones A y G, quizás hayan notado que no se utilizó la simplificación hecha por el método del mapa de Karnaugh.

Pero también si son observadores, pueden notar que la expresión de C y E es equivalente a una de las subexpresiones de A y G, por lo tanto podemos reutilizar esa compuerta lógica.

Finalizando, el diagrama quedaría así:


a b c
A, G
C, E
B, F
D
A
B
C
D
E
F
G

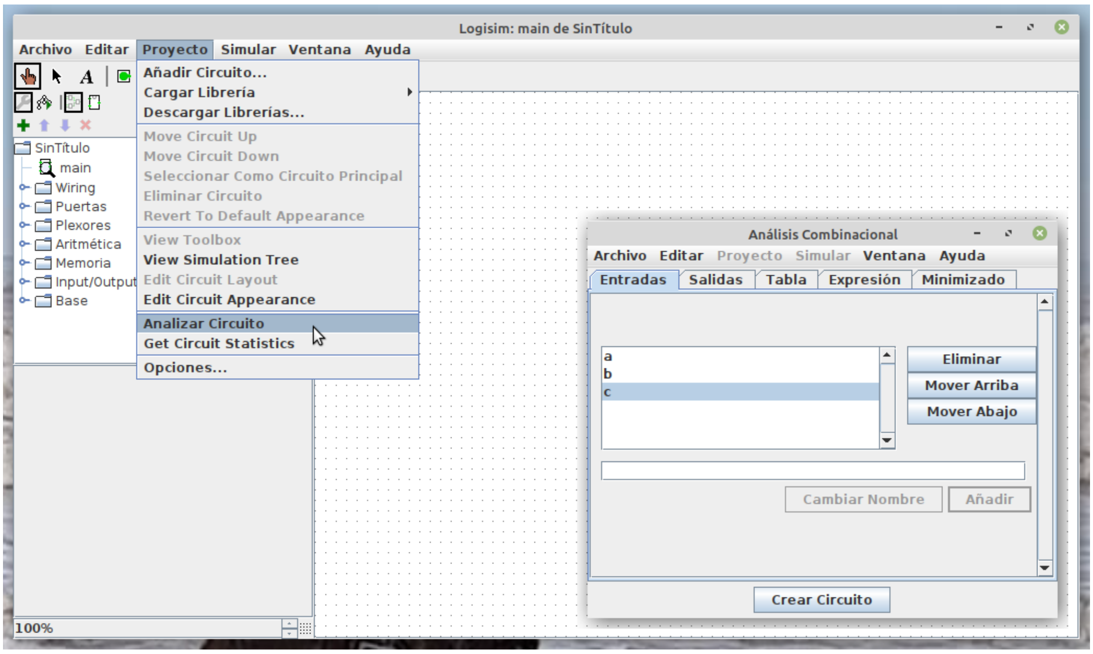
Ejercitación

Imagine que se quiere representar el valor 7 con el dado analizado, el cual tendría todos los LEDs encendidos. Realice la modificación al circuito para contemplar esta nueva expresión.

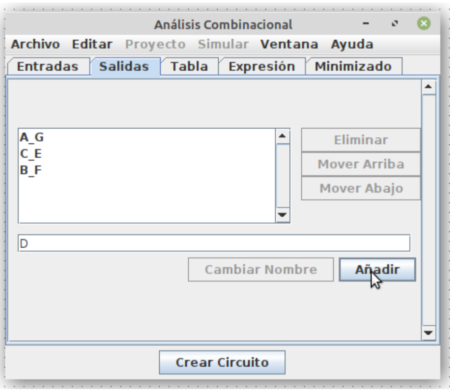
No menciono el caso de querer representar el 0 con todos los leds apagados porque en el diagrama resultaría trivial. ¿Entiende el por qué?

Mediante el software **logisim** mencionado en la clase anterior, se puede realizar el esquema inicial del circuito mediante la opción *Análisis Combinacional* del menú *Proyecto*. El proceso tiene unos simples pasos como muestran las imágenes:

- Se añaden las 3 entradas a,b,c en la solapa **Entradas.**

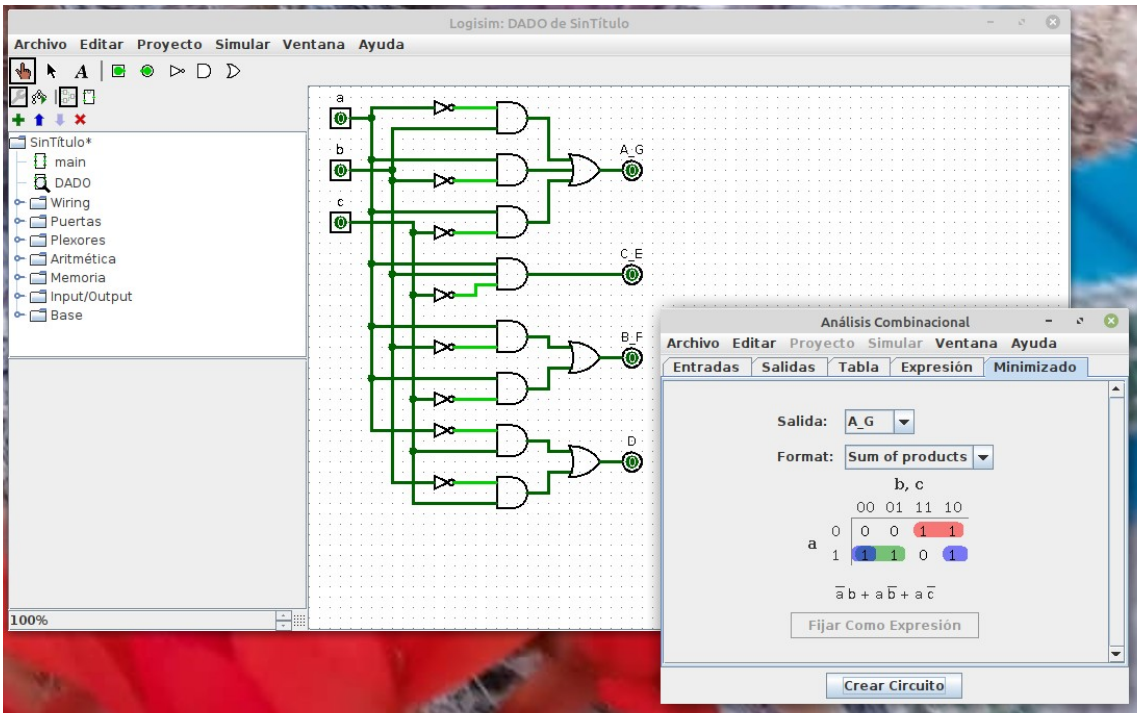


- Se definen las variables de salida (LEDS) en la solapa **Salida**, En la imagen se cambió el orden así el circuito sale expresado de una forma más facil de modificar.



- Se transcribe la tabla de verdad en la solapa **Tabla** respetando las variables de salida cambiadas de orden

- Presionando el botón **Crear Circuito** (se puede dar un nombre, por ejemplo DADO), automáticamente quedarán simplificadas las expresiones canónicas en la solapa **Expresión** y se podrá observar la simplificación mediante Karnaugh de cada variable de salida en la solapa **Minimizado**.

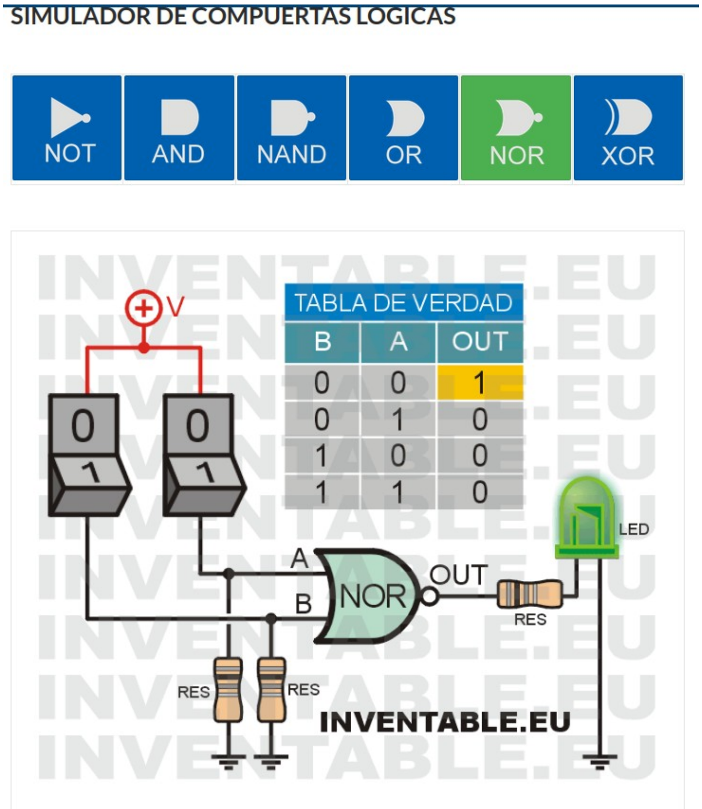


A partir de allí, solamente hay que hacer la tediosa tarea de modificar el circuito para que quede más claro añadiendo los leds que se deben encender de a pares.

Bibliografía adicional

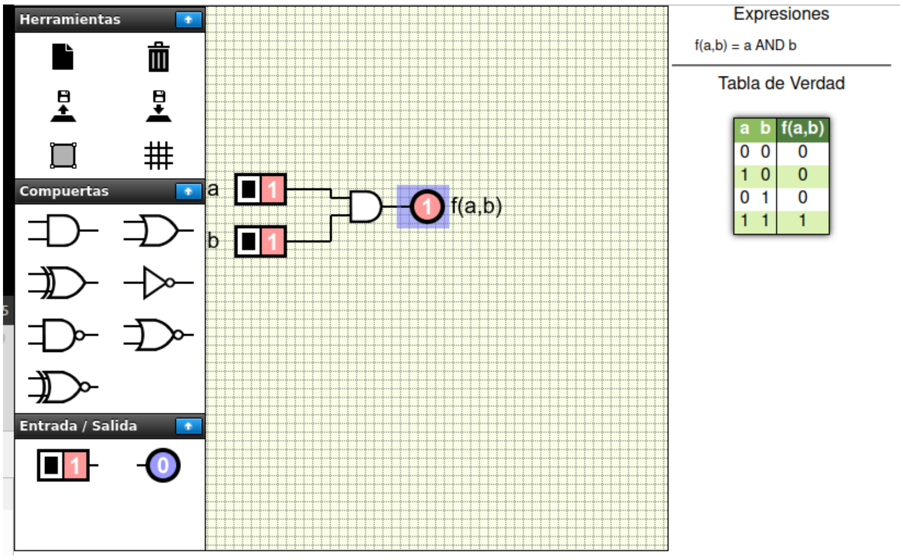
- Simulador de compuertas lógicas:

  https://www.inventable.eu/simulador-compuertas-logicas/



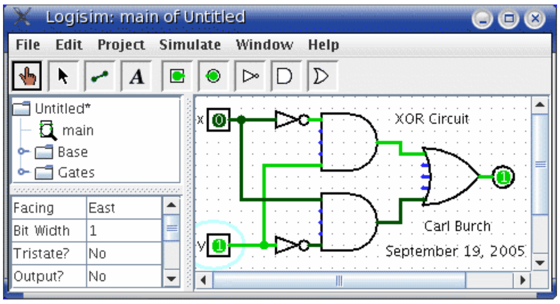
- Simulador de simples circuitos binarios on-line, lamentablemente no permite almacenar los circuitos diseñados:

  http://163.10.22.82/OAS/compuertas_logicas/Simulacion/editor_simple.html



- **Logisim**, aplicación multiplataforma de distribución libre hecha en java para diseñar circuitos digitales elementales.

  http://www.cburch.com/logisim/index.html

- **Karnaugh map generator**, mapa de Karnaugh online: http://www.32x8.com/index.html