CIRCUITOS COMBINATORIOS

1 Introducción

En este capítulo presentaremos los elementos básicos para la implementación en hardware de las funciones lógicas y ejemplos de cómo se pueden sintetizar funciones más complejas en base a operaciones más simples disponibles como "bloques".

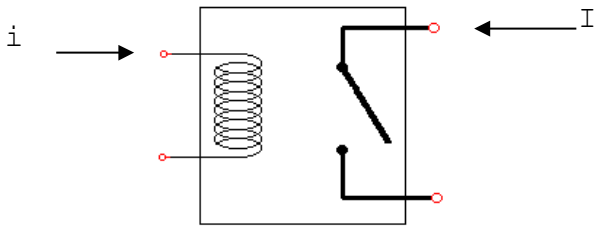
En general trabajaremos con los "Circuitos Combinatorios", que los definimos como aquellos cuya salida está determinada, en todo instante, por el valor actual de sus entradas.

Antes de presentar la forma de representar las funciones lógicas básicas, veremos brevemente como se implementan las mismas en términos electrónicos.

2 Transistores

Ya dijimos que el Modelo Circuital del Álgebra de Boole permite la construcción mediante "llaves eléctricas" de un álgebra isomórfica con el Modelo Binario. Para ello necesitamos que estas llaves puedan ser controladas (es decir accionadas) por los valores que adoptan las variables (que en el modelo están representados por 0 = no circula corriente y 1 = circula corriente).

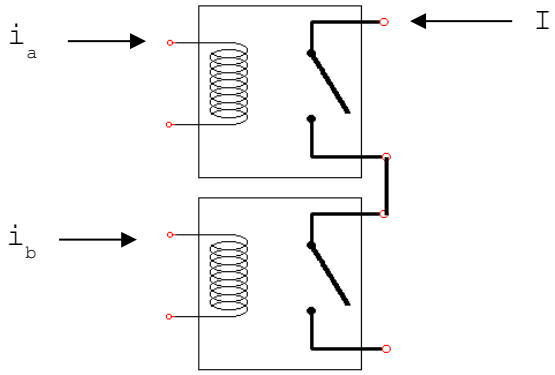
Para ello el primer mecanismo que se usó (de hecho en la computadora MARK I en la década del 40) fue el relé (llave electromagnética). El diagrama de un dispositivo de estas características es:



Al circular una corriente `i` por el circuito de la izquierda la bobina genera un campo electromagnético que acciona el interruptor, el cuál permite el pasaje de la corriente `I` por el circuito de la derecha. Este esquema corresponde a un relé "normalmente abierto". Existen también los relés "normalmente cerrados", en los cuales el mecanismo electromagnético funciona a la inversa: la inducción de la bobina provoca la apertura del interruptor.

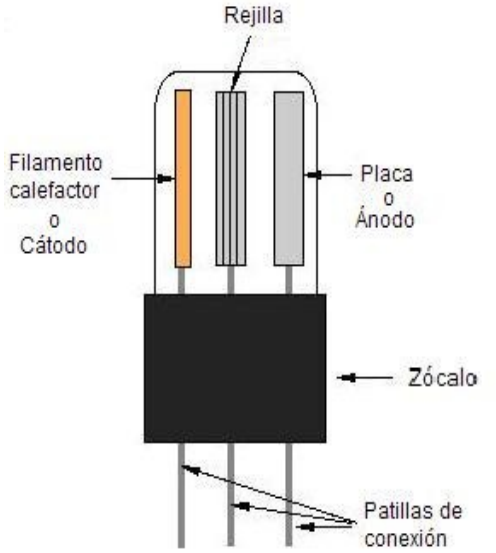
Si en un sistema con relés tomamos el estado "circula corriente" como 1 y "no circula corriente" como 0, es directo observar que un relé normalmente cerrado implementa la función NOT.

En ese modelo también es sencillo de observar que el siguiente circuito implementa la función AND entre dos corrientes i_a e i_b:



Los relés fueron luego reemplazados por las válvulas de vacío. Estos dispositivos fueron los primeros completamente electrónicos utilizados en una computadora: la ENIAC.

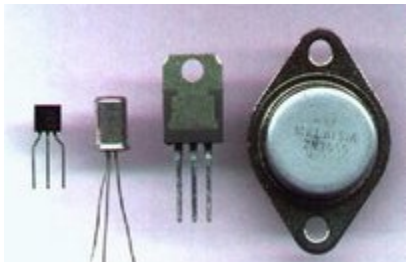
Tomado de Virtual Valve Museum (http://en.wikipedia.org/wiki/Image:Triode.jpg)

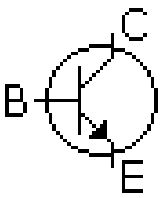


Tomado de Wikipedia

En el caso de la válvula de vacío tipo "Triodo" (como la mostrada en el diagrama) la diferencia de potencial entre la "rejilla" y el "cátodo" controlan el pasaje de electrones entre el "ánodo" y el "cátodo", con lo cual el triodo es una llave controlada por diferencia de potencial (ó "tensión", ó "voltaje").

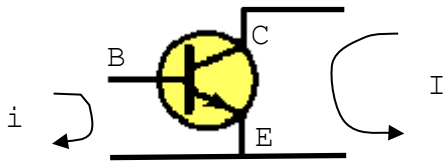
A fines de la década del 50 se inventa el transistor, el cuál, con algunas variantes, es el que se utiliza hasta ahora y que ha permitido alcanzar densidades de millones de unidades por centímetro cuadrado.





Tomado de Wikipedia (http://es.wikipedia.org/wiki/Imagen:Transistor-photo.jpg)

El funcionamiento de un transistor bipolar aplicado a la lógica digital es básicamente el de una llave electrónica muy similar a un relé.



Los transistores se pueden definir, en principio, como "amplificadores de corriente" ya que al circular una corriente i entre la base (B) y el emisor (E) del transistor se produce la circulación de una corriente I entre el colector (C) y el emisor (E), cumpliendo la relación:

$$I = h_{FE} * i$$

siendo h_{FE} una constante típica de cada modelo de transistor. Esta relación de proporcionalidad es válida en el denominado modo de funcionamiento "lineal" y es utilizada por ejemplo en los dispositivos que trabajan con señales analógicas, como equipos de audio, televisores, etc. El funcionamiento en modo "lineal" depende, básicamente, del circuito donde el transistor se coloca.

Sin embargo en las aplicaciones de lógica digital a los transistores se los hace trabajar en el modo de funcionamiento "saturación", donde la corriente I alcanza su valor máximo (determinado por otros parámetros del circuito).

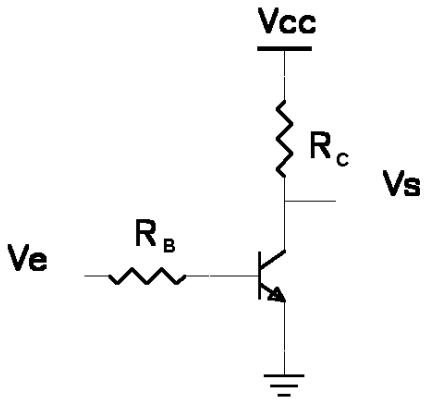
Por otra parte, y al igual que en las aplicaciones analógicas, la magnitud eléctrica que se considera es la diferencia de potencial eléctrico (o "voltaje") en lugar de la corriente. Dada la conocida relación entre diferencia de potencial y corriente eléctrica dada por la Ley de Ohm:

$$\Delta V = R * I$$

Como muchas veces se consideran las diferencias de potencial ΔV con respecto a una referencia común (denominada "tierra lógica") es que se aplica la relación anterior en la forma:

$$V = R * I$$

Entonces veamos como queda el transistor cuando agregamos resistencias en su base y en su colector:



El modo de funcionamiento de saturación se alcanza cuando el valor de V_e (diferencia de potencial o voltaje de entrada) es tal que la corriente entre base y emisor (I_{BE}) genera una corriente entre colector y emisor (I_{CE}) tal que la caída de potencial en la resistencia R_C colocada entre el colector y la fuente de alimentación coincide con la diferencia de potencial de ésta. Esto equivale a decir que la diferencia de potencial entre colector y emisor es 0. En realidad en la práctica dicha diferencia nunca puede llegar a ser menor que un cierto valor denominado "voltaje de saturación colector emisor" (V_{CEsat}), por lo que la condición de saturación se da cuando se cumple la relación:

$$V_{CC} - V_{CEsat} = R_C * h_{FE} * I_{BE}$$

El valor de V_{CEsat} es aproximadamente 0,2 Volts, pero a los efectos se toma como 0.

Por su lado la corriente I_{BE} circula cuando $V_e > V_{BE}$ siendo V_{BE} una diferencia de potencial que se produce entre base y emisor de un transistor y cuyo valor un parámetro característico de los mismos y vale aproximadamente 0,6 Volts. Por ley de Ohm la corriente de base cumple:

$$V_e - V_{BE} = R_B * I_{BE}$$

despejando I_{BE} y sustituyendo resulta:

$$V_{CC} - V_{CEsat} = R_C * h_{FE} * (V_e - V_{BE}) / R_B$$

En definitiva si se eligen correctamente los valores de las resistencias de base y colector para que el sistema funcione en saturación para cuando el valor de la entrada sea $V_e = V_{CC}$ se cumplirá que:

si $V_e = V_{CC}$ entonces $V_s = 0$

Por otra parte si no circula corriente entre base y emisor tampoco circulará corriente entre colector y emisor. Esto significa que no circulará corriente por la resistencia de colector y, de acuerdo a la ley de Ohm, no habrá entonces caída de potencial entre los extremos de la misma, por lo que se cumplirá que:

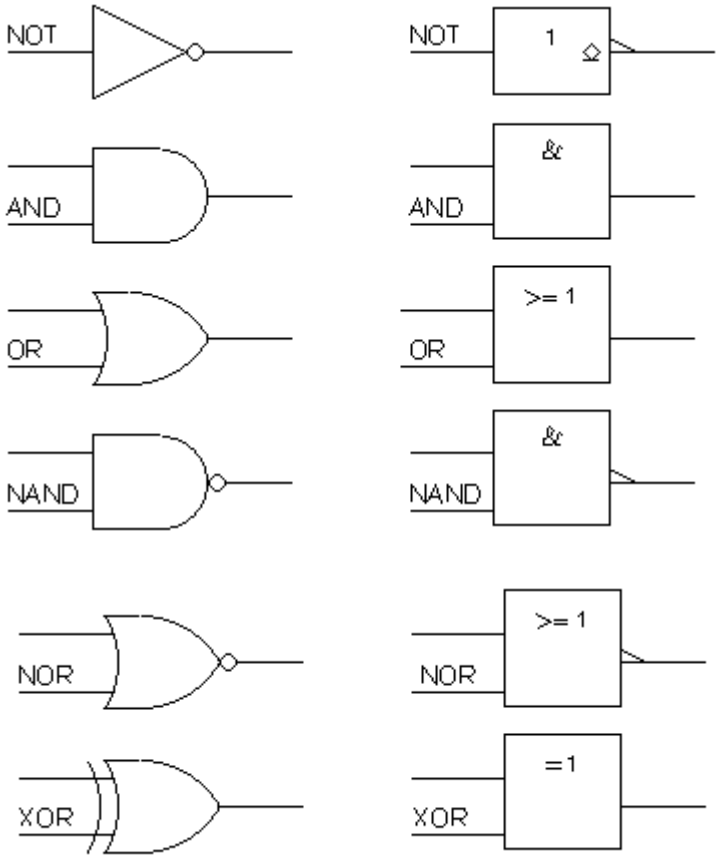
si $V_e = 0$ entonces $V_s = V_{CC}$

Si consideramos que el voltaje 0 corresponde a un 0 lógico y el voltaje igual al de la fuente V_{CC} corresponde a un 1 entonces vemos que el circuito anterior implementa la operación NOT.

3 Compuertas

La implementación en circuitos de las conectivas binarias básicas se denominan habitualmente "compuertas". Así tendremos compuertas AND, OR, NAND, NOR, XOR y NOT.

Las compuertas se representan por símbolos gráficos. Existen dos grandes familias de símbolos, los tradicionales (símbolos "redondeados") y los propuestos por el estándar ANSI/IEEE 91-1984 / IEC Publication 617-12 (símbolos "rectangulares").



4 Circuitos Lógicos Combinatorios

Los circuitos lógicos combinatorios, o simplemente circuitos combinatorios, son circuitos elaborados a partir de compuertas o de otros circuitos del mismo tipo (usados como "bloque constructivo") cuya salida es una función lógica de sus entradas y por tanto las salidas actuales sólo dependen del valor actual de las entradas.

Los circuitos combinatorios son, en definitiva, la implementación en hardware de funciones lógicas (funciones booleanas). Los circuitos se representan como "cajas negras" de las cuales se especifican las entradas, las salidas y la función booleana que las vincula.

5 Síntesis de Circuitos Combinatorios

Una forma siempre aplicable (aunque en algunos casos más trabajosa) para construir un circuito lógico parte de la función booleana a implementar representada, por ejemplo, mediante su tabla de verdad. A partir de ella y mediante Diagramás de Karnaugh (u otro método) se realiza la minimización (en dos niveles) de la misma. De esa forma se

llega a una expresión de la función en base a operaciones NOT y ANDs y ORs (multipuerta).

Una alternativa es identificar partes del circuito que se puedan realizar con otros "bloques constructivos" de mayor nivel, tales como sumadores de n bits, multiplexores, decodificadores, comparadores, selectores, etc. Luego se reúnen y conectan adecuadamente estos bloques minimizando, de ser necesario, la lógica de dichas conexiones mediante Diagramás de Karnaugh.

Veremos a continuación un par de ejemplos de aplicación de la metodología.

5.1 Circuito Mayoría

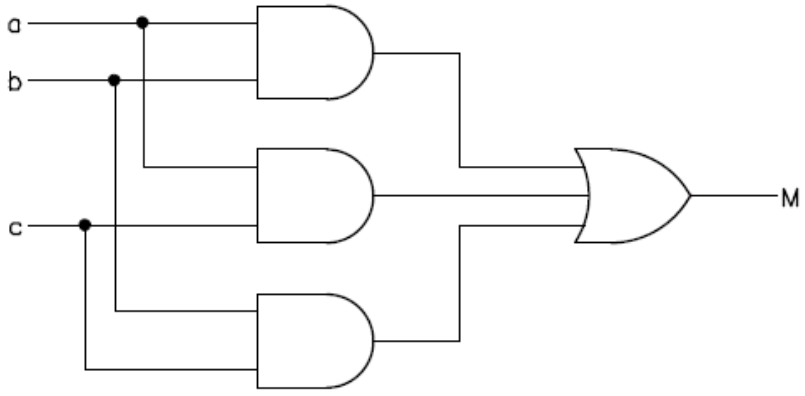
Consideremos la función booleana determinada por la tabla de verdad:

a	b	c	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

La función minimizada queda:

```
M = ab + ac + bc
```

Entonces el circuito en base a compuertas AND, OR y NOT tiene la forma:

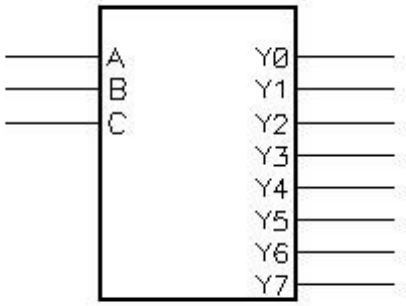


6 Bloques Constructivos

A continuación veremos algunos ejemplos de circuitos combinatorios útiles para ser utilizados como parte de diseños más complejos. De hecho estos circuitos, o algunas variantes de ellos, están disponibles como circuitos integrados independientes, pensando en dicho propósito. Conforman la base del concepto de re-utilización que introdujo la electrónica digital hace más de 4 décadas y que la industria del software imitó con los conceptos de modularidad, re-usabilidad de código y orientación a objetos.

6.1 Circuito Decodificador

El circuito decodificador es un circuito con N entradas y 2^N salidas. Para el caso de N = 3 tiene el siguiente símbolo:



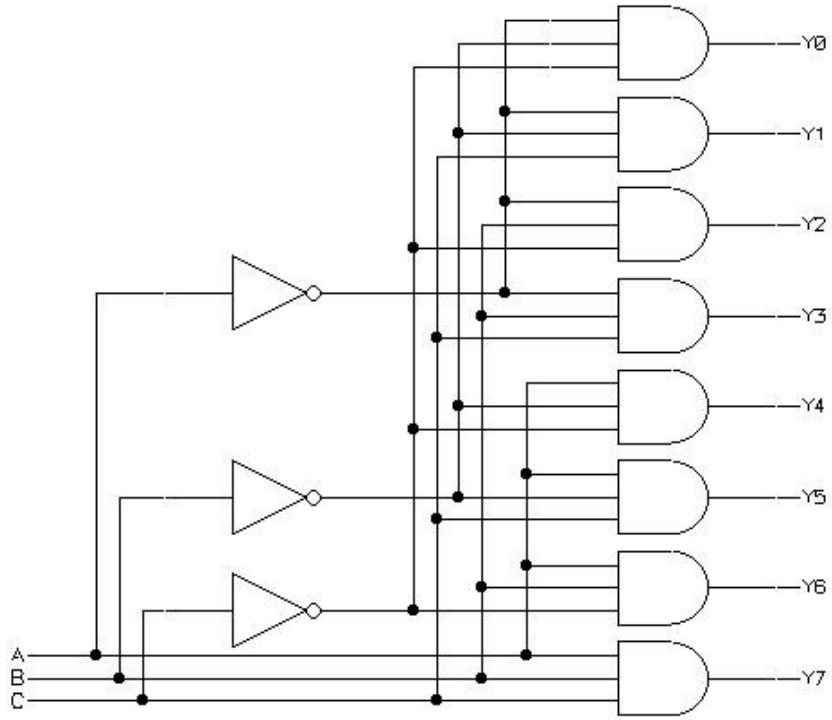
Su tabla de verdad es:

A	B	C	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

es decir dada una combinación de unos y ceros a la entrada todas las salidas estarán en 0 salvo la que corresponda al número binario coincidente con la combinación de entradas.

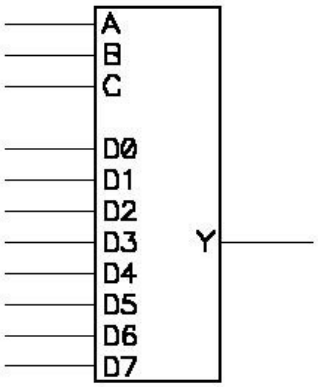
En la práctica este circuito está disponible con lógica negativa (las salidas son 1 salvo la seleccionada) y con entradas adicionales funcionando como un demultiplexor (circuito que veremos más adelante).

El circuito "interno" de un decodificador es simple:



6.2 Circuito Multiplexor

El circuito multiplexor es un circuito con N + 2^N entradas y 1 salida. Para el caso de N = 3 tiene el siguiente símbolo:



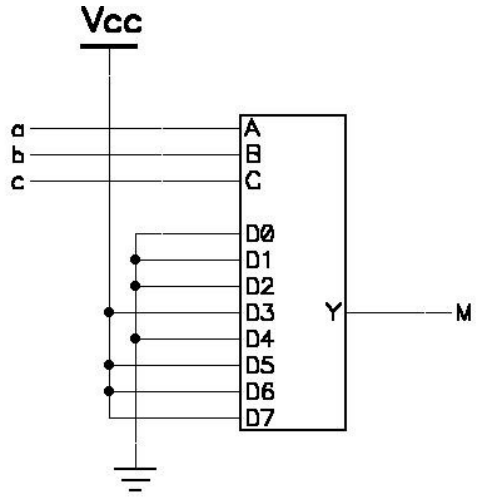
Su tabla de verdad es:

A	B	C	Y
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

es decir que la salida Y toma el valor de la entrada D cuyo índice coincida con el número binario representado por las entradas A, B y C. Las entradas A, B y C se llaman entradas de "control" y las entradas D se llaman de "datos".

Se denomina multiplexor porque es capaz de presentar en una sola variable Y cualquiera de las 2^N entradas. Esto tiene aplicaciones cuando la salida Y es un "canal principal de comunicación", mientras que las entradas D son "sub-canales de comunicación". Este circuito permite que variando en el tiempo las entradas de control logremos dividir la utilización del canal principal entre los sub-canales. Esta técnica se denomina TDM (Time Division Multiplexing). De allí que el circuito se denomine multiplexor.

Una característica distintiva del circuito multiplexor es su aplicabilidad para la realización de funciones lógicas. Consideremos el siguiente circuito basado en un multilexor 8 a 1:



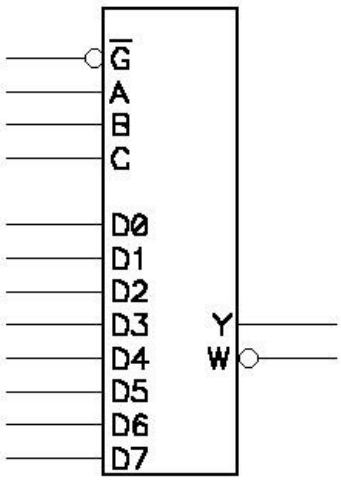
Si escribimos la tabla de verdad de este circuito nos queda:

a	*b*	*c*	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

que coincide con la tabla de verdad del circuito Mayoría (a, b, c).

Si observamos nuevamente la tabla de verdad resumida del circuito multiplexor constataremos que asignando un valor apropiado a cada una de las entradas del datos D, estamos en condiciones de generar el valor 0 ó 1 correspondiente a cada una de las combinaciones de los bits de control (entradas A, B, C). Esto permite construir cualquier función lógica de **n** variables con un multiplexor de **n** entradas de control.

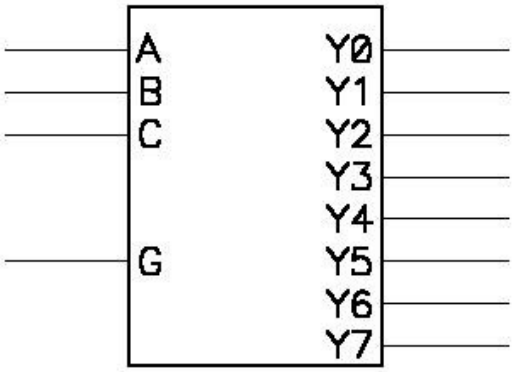
Los circuitos integrados disponibles (como el 74251) disponen de salida "tri-state" (el concepto de tri-state se verá más adelante) controlada por una entrada adicional G y una salida de lógica invertida W:



Por más información ver el documento:
http://www.fairchildsemi.com/ds/DM/DM74ALS251.pdf

6.3 Circuito Demultiplexor

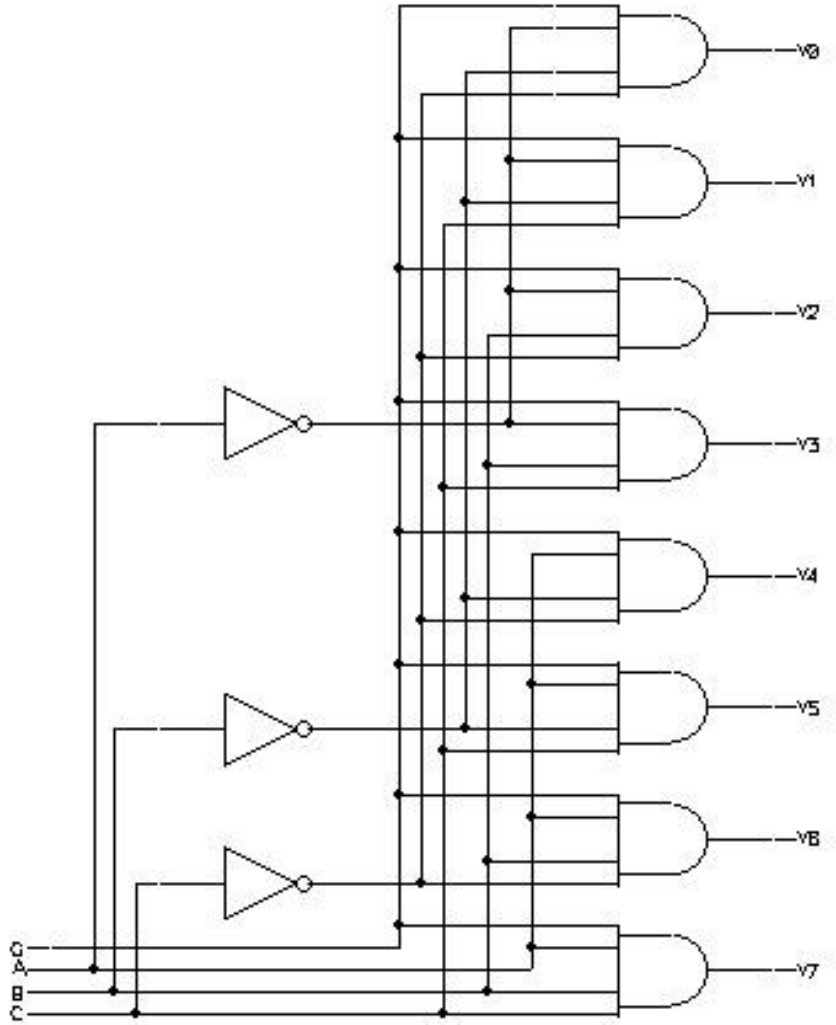
El circuito demultiplexor realiza la tarea inversa al multiplexor. Posee 1 entrada de datos, N entradas de control y 2^N salidas, según el símbolo:



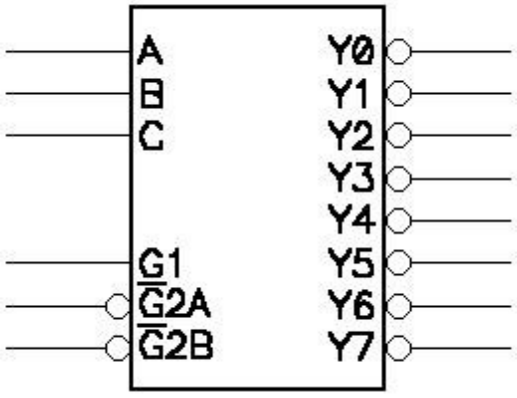
Su tabla de verdad es muy similar a la del decodificador, con la diferencia que el valor de la salida seleccionada depende de la entrada G:

A	B	C	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	G
0	0	1	0	0	0	0	0	0	G	0
0	1	0	0	0	0	0	0	G	0	0
0	1	1	0	0	0	0	G	0	0	0
1	0	0	0	0	0	G	0	0	0	0
1	0	1	0	0	G	0	0	0	0	0
1	1	0	0	G	0	0	0	0	0	0
1	1	1	G	0	0	0	0	0	0	0

El circuito interno equivalente de un demultiplexor es:



En la práctica solamente se consiguen circuitos integrados demultiplexores y no existen decodificadores "puros", ya que con ellos es muy simple construir un decodificador (basta poner la entrada G en 1). Es el caso, por ejemplo, del circuito 74138, el cual tiene las salidas con lógica negada y además posee dos entradas adicionales de lógica negada (que están en AND con la entrada de lógica directa):



Por más información ver el documento:
http://www.fairchildsemi.com/ds/DM/DM74ALS138.pdf

6.4 Circuito Sumador Completo de 1 bit

Veremos a continuación un circuito que puede ser utilizado para construir sumadores de n bits, mediante su conexión en "cascada". Para ello deberemos construir un sumador de dos números de 1 bit cada uno con entrada y salida de acarreo (carry).

La tabla de verdad de este circuito es:

a	b	c_{in}	S	c_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

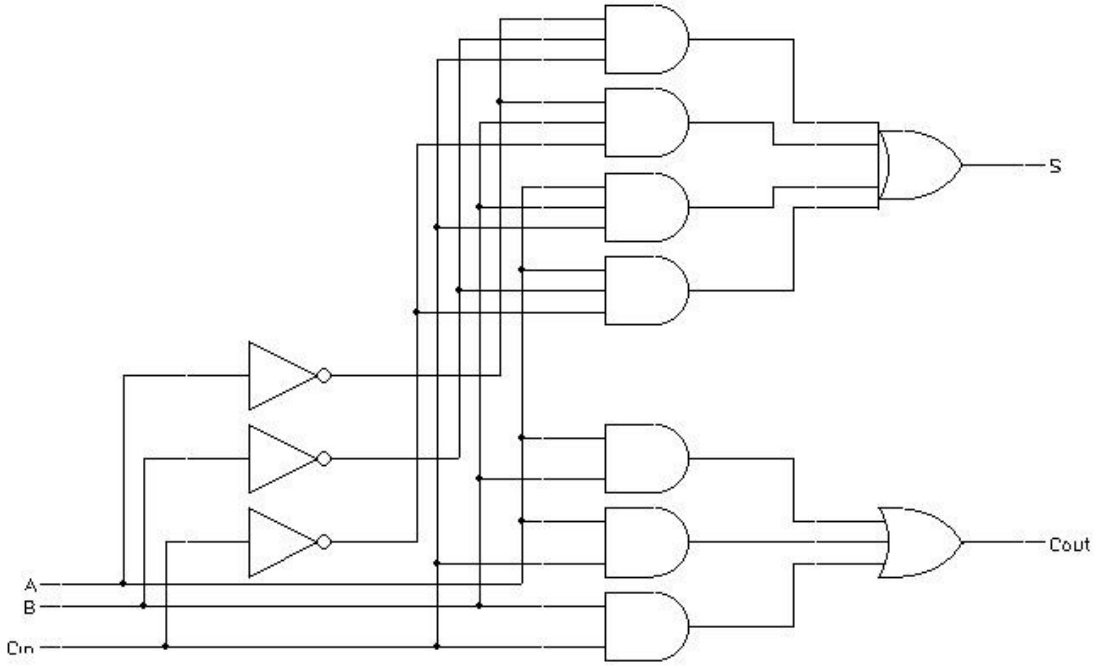
(Nota: observar que c_{out} tiene el mismo diagrama que el circuito Mayoría)

Utilizando el método de tomar los lugares donde la función vale 1 y escribir los productos de las entradas dependiendo de su valor (negando la entrada si es 0) y luego sumar estos productos las ecuaciones quedan:

$$S = \bar{a}\bar{b}c_{in} + \bar{a}b\bar{c}_{in} + a\bar{b}\bar{c}_{in} + abc_{in}$$

$$c_{out} = ab + ac_{in} + bc_{in}$$

El circuito que se obtiene sería entonces:



Este es un caso donde se puede observar que el método de tomar los lugares donde la función vale 1 y escribir la función como suma de productos permite hallar un circuito mínimo en dos niveles, pero no necesariamente el circuito que tenga el menor número de compuertas.

Si observamos la expresión de S:

$$S = \overline{\overline{a}}\,\overline{\overline{b}}\,c_{in} + \overline{\overline{a}}\,b\,\overline{c_{in}} + a\,\overline{\overline{b}}\,\overline{c_{in}} + a\,b\,c_{in}$$

y trabajamos con ella resulta:

$$S = (\overline{\overline{a}}+a)\,b\,c_{in} + (\overline{\overline{a}}+a)\,\overline{\overline{b}}\,\overline{c_{in}}$$

El segundo paréntesis corresponde a la expresión del xor entre a y b. El primero a la negación de dicho exor:

$$S = \overline{(a \oplus b)}\,c_{in} + (a \oplus b)\,\overline{\overline{c_{in}}}$$

Podemos observar en la expresión anterior que ahora queda el exor entre el exor de a y b con c_{in}.

$$S = (a \oplus b) \oplus c_{in}$$

Por otra parte la expresión del carry de salida es:

$$c_{out} = ab + ac_{in} + bc_{in}$$

que podemos agrupar como:

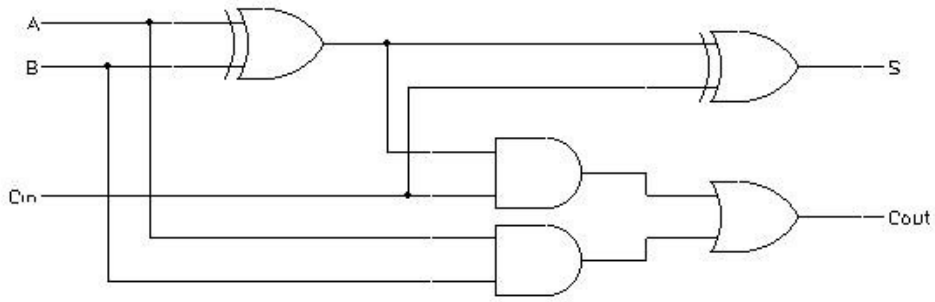
$$c_{out} = ab + (a+b)\,c_{in}$$

y esto es equivalente a:

$$c_{out} = ab + (a \oplus b)c_{in}$$

(esto se debe a que el único caso donde a+b es distinto de a⊕b es cuando a y b son 1 simultáneamente y allí ab es 1 y por tanto el or general da 1 de todos modos).

El circuito que resulta de estas nuevas expresiones es:



que, como se puede observar directamente, posee muchas menos compuertas que la versión anterior. Como nada es "gratis" también se observa que el circuito tiene un nivel más de compuertas (normalmente el nivel de compuertas "not" no se cuenta, por lo que el circuito original tiene 2 niveles y este tiene 3).

Esto desde el punto de vista lógico no presenta dificultades. Pero en la práctica existe un fenómeno (tiempo de setup o propagación) que desaconseja el incorporar más niveles de compuertas en los circuitos.

Veamos como quedaría un sumador de 4 bits en base a sumadores de 1 bit: