UD2 DISEÑO DE CIRCUITOS CON PUERTAS LÓGICAS

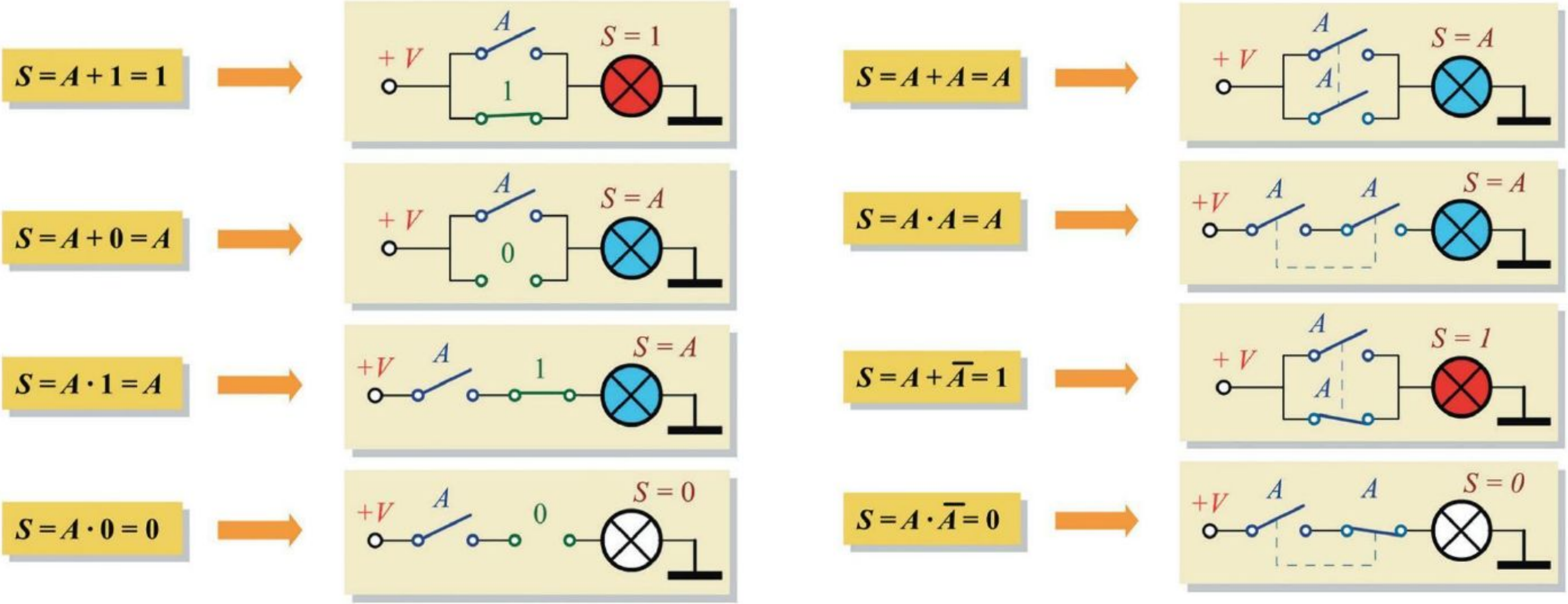
1.- Álgebra de Boole

El álgebra de Boole es una teoría matemática que nos va a permitir operar con números binarios.

El álgebra de Boole fue desarrollada en 1847 por George Boole, para resolver cuestiones de lógica deductiva, en las cuales se utilizan dos soluciones posibles, <<verdadero>> o <<falso>>. Más adelante, este álgebra se utilizó para diseñar los circuitos de conmutación de telefonía que utilizaban relés. La llegada de los circuitos digitales hizo que el álgebra de Boole se convirtiese en indispensable para su diseño y análisis.

Postulados:

Para explicar los postulados del álgebra de Boole nos vamos a ayudar del circuito eléctrico de contactos equivalentes.



Propiedades

Cuando se invierten los dos términos de una igualdad, esta permanece igual:

$$S = A + B \text{ es igual a } \overline{S} = \overline{A + B}$$

$$S = A \cdot B \text{ es igual a } \overline{S} = \overline{A \cdot B}$$

Propiedad conmutativa

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

Propiedad asociativa

$$A + B + C = A + (B + C) = (A + B) + C$$

$$A \cdot B \cdot C = A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

Propiedad distributiva

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

Teoremas

Teorema 1

a) $A + A \cdot B = A$

Demostración:

$$A + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A$$

b) $A \cdot (A + B) = A$

Demostración:

$$A \cdot (A + B) = A \cdot A + A \cdot B = A + A \cdot B = A$$

Teorema 2

a) $A + \overline{A} \cdot B = A + B$

Demostración: Dado que el término $B + \overline{B} = 1$, si lo multiplicamos por el término A, la expresión no varía:

$$S = A + \overline{A}B = A(B + \overline{B}) + \overline{A}B = AB + A\overline{B} + \overline{A}B$$

Dado que el término $AB = AB + AB$, si ahora sumamos a la expresión anterior un término AB, esta no varía:

$$S = AB + A\overline{B} + \overline{A}B = AB + A\overline{B} + \overline{A}B + AB$$

Reagrupando:

$$S = A + B$$

b) $A \cdot (\overline{A} + B) = A \cdot B$

Demostración:

$$A \cdot (\overline{A} + B) = A \cdot \overline{A} + A \cdot B = 0 + A \cdot B = A \cdot B$$

- Teorema 3: **Leyes de Morgan**

Este teorema es de gran utilidad en la simplificación y conversión de funciones:

a) $\overline{A + B + C + \ldots + N} = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \ldots \cdot \overline{N}$

b) $\overline{A \cdot B \cdot C \cdot \ldots \cdot N} = \overline{A} + \overline{B} + \overline{C} + \ldots + \overline{N}$

Simplificación algebraica de funciones lógicas

Con la ayuda de los postulados, propiedades y teoremas del álgebra de Boole es posible simplificar una función lógica hasta su mínima expresión, con lo que se consigue construir circuitos lógicos más sencillos y económicos.

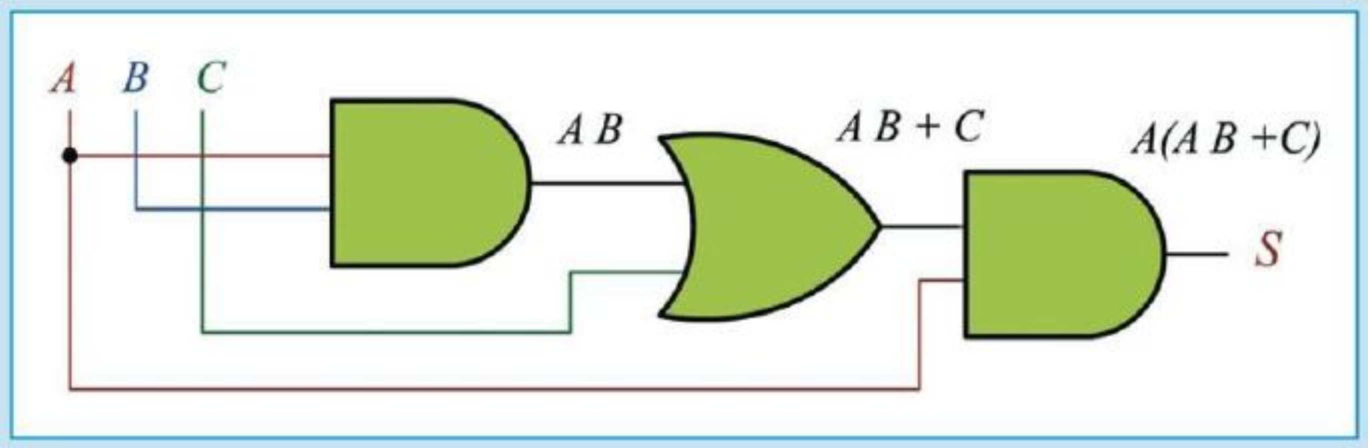
Simplifica la siguiente función y realiza los diagramas lógicos antes y después de la simplificación:

a) $S = A \cdot (A \cdot B + C)$

Simplifica la siguiente función y realiza los diagramas lógicos antes y después de la simplificación:

a) $S = A \cdot (A \cdot B + C)$

El diagrama lógico de la función sin simplificar sería

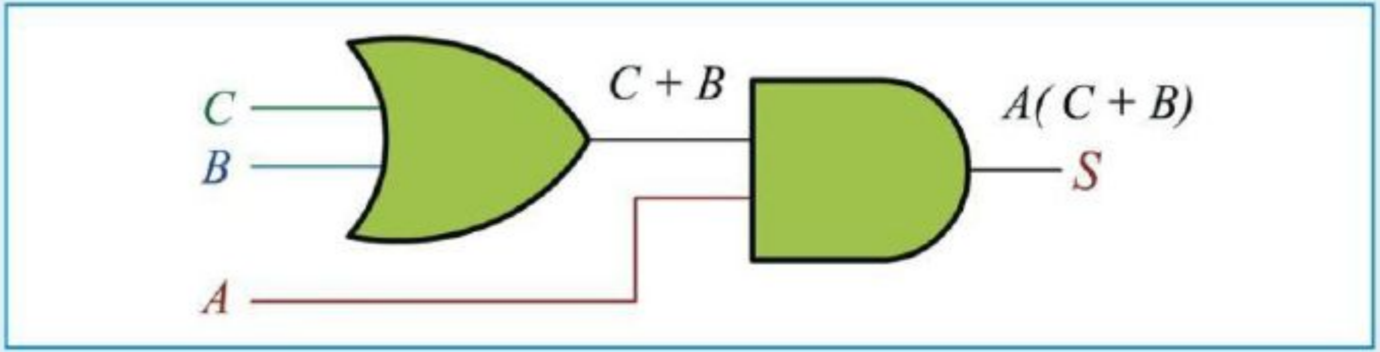


$$S = A \cdot (A \cdot B + C) = A \cdot A \cdot B + A \cdot C = (A \cdot A) \cdot B + A \cdot C$$

Como $A \cdot A = A$, tenemos que:

$$S = A \cdot B + A \cdot C = A \cdot (B + C)$$

El diagrama lógico de la función simplificada es



Simplifica la siguiente función y realiza los diagramas lógicos antes y después de la simplificación:

$$S = A + \overline{A}B$$

Comprueba si la simplificación de las siguientes funciones es correcta

$$\overline{A}B + AB + B\overline{A} = B$$

$$\overline{A}(BC + AB + B\overline{A}) = \overline{A}B$$

$$ABC + CAB + AB + A = A$$

$$AB\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C + A\overline{B} + C = A + C$$

Simplifica la siguiente función y realiza los diagramas lógicos antes y después de la simplificación. Escribe la tabla de la verdad de la función una vez simplificada:

$$S = \overline{\bar{A} + \overline{AB} + BC} + \bar{A}$$

Simplificación de funciones lógicas mediante el mapa de Karnaugh

Dado que la simplificación por el método algebraico resulta largo, complejo y poco sistemático se han ideado otros métodos de simplificación más sencillos como el de Karnaugh.

Mapa de Karnaugh para dos variables

Para dos variables dibujaremos una tabla con $2^2 = 4$ celdas, donde se escribirá el resultado de la función canónica o los términos de la tabla de la verdad que den como resultado un «1» lógico en su salida.

$A \backslash B$	0	1
0	$\overline{A}\,\overline{B}$ 00	$\overline{A}B$ 01
1	$A\overline{B}$ 10	AB 11

Así, por ejemplo, para la siguiente función, la tabla de la verdad y el mapa de Karnaugh serían los que se representan en la Figura

$$S = AB + \overline{A}B$$

En el mapa se escriben solamente el «1» lógico de cada uno de los términos de la función de salida en la celda correspondiente del mapa de Karnaugh.

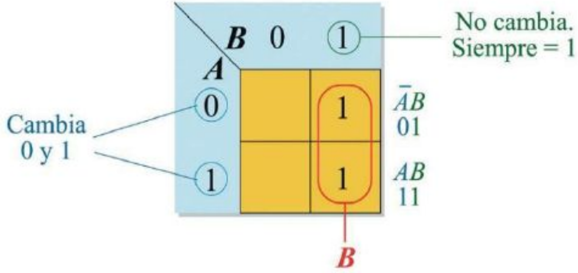
A	*B*	*S*
0	0	0
0	1	1
1	0	0
1	1	1

A \ *B*	0	1
0	$\overline{A}\overline{B}$ 00	1 $\overline{A}B$ 01
1	$A\overline{B}$ 10	1 AB 11

El método de simplificación consiste en agrupar los «1» adyacentes de dos en dos y en sentido horizontal o vertical.

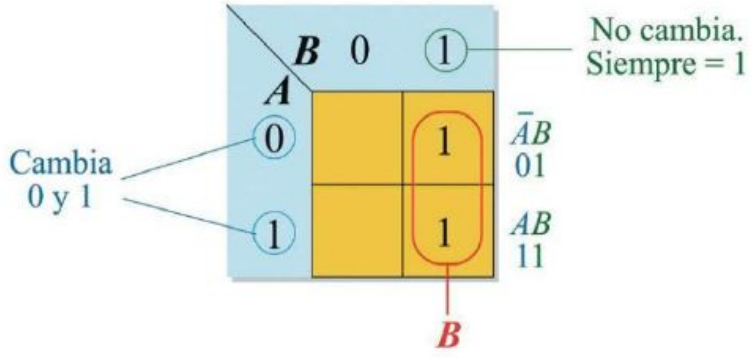
Para una función de dos variables si se consiguen dos «1» adyacentes se puede eliminar una de las variables. Para cuatro «1» adyacentes, el valor de la función es siempre 1.

La variable que se mantiene es aquella que no cambia de valor en la agrupación de unos adyacentes, eliminándose la que cambia.



En nuestro ejemplo de la Figura , la variable que no cambia de valor es la *B* ya que en las dos celdas adyacentes con «1» su valor siempre es 1, sin embargo la variable *A*, toma el valor 0 en una celda y 1 en la otra, por lo que la simplificación de la función queda así

$$S = B$$



Escribe la tabla de la verdad y simplifica la siguiente función de salida:

$$S = A\overline{B} + AB + \overline{A}B$$

Mapa de Karnaugh para tres variables

Para una función de tres variables si se consigue:

- Dos «1» adyacentes se puede eliminar una de las variables.
- Cuatro «1» adyacentes se puede eliminar dos de las variables.
- Ocho «1» adyacentes el valor de la función es siempre 1.

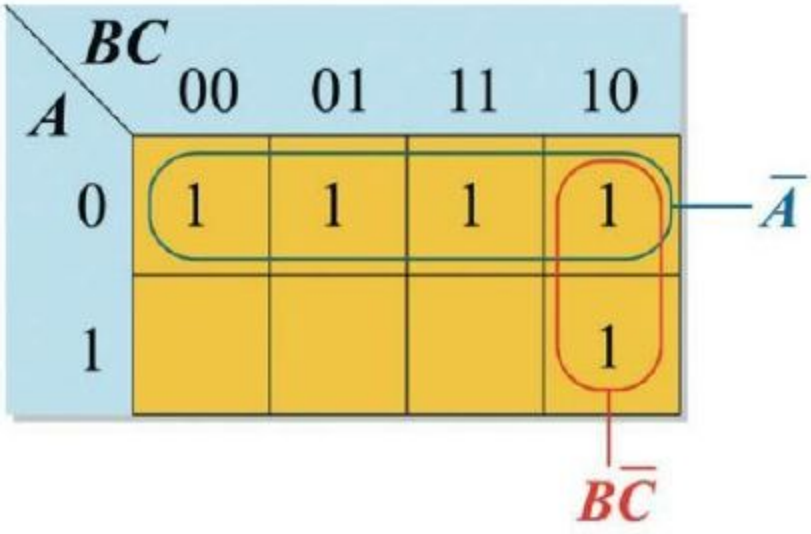
$A \backslash BC$	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$ 000	$\bar{A}\bar{B}C$ 001	$\bar{A}BC$ 011	$\bar{A}B\bar{C}$ 010
1	$A\bar{B}\bar{C}$ 100	$A\bar{B}C$ 101	ABC 111	$AB\bar{C}$ 110

Así, por ejemplo, para la siguiente función, la tabla de la verdad y el mapa de Karnaugh sería el que se representa en la Figura

$$S = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}BC + \overline{A}B\overline{C} + AB\overline{C}$$

A B C	*S*
0 0 0	1
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	1
1 1 1	0

A \ *BC*	00	01	11	10
0	1 $\overline{A}\,\overline{B}\,\overline{C}$ 000	1 $\overline{A}\,\overline{B}C$ 001	1 $\overline{A}BC$ 011	1 $\overline{A}B\overline{C}$ 010
1	$A\overline{B}\,\overline{C}$ 100	$A\overline{B}C$ 101	ABC 111	1 $AB\overline{C}$ 110



La función simplificada queda así:

$$S = \overline{A} + B\overline{C}$$

Escribe la tabla de la verdad y simplifica la siguiente función de salida:

$$S = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + A\overline{B}C$$

Escribe la tabla de la verdad y simplifica la siguiente función lógica:

$$S = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} + ABC$$

Mapa de Karnaugh para cuatro variables

Para una función de cuatro variables si se consigue:

- Dos «1» adyacentes se puede eliminar una de las variables.
- Cuatro «1» adyacentes se puede eliminar dos de las variables.
- Ocho «1» adyacentes se puede eliminar tres de las variables.
- 16 «1» adyacentes el valor de la función es siempre 1.

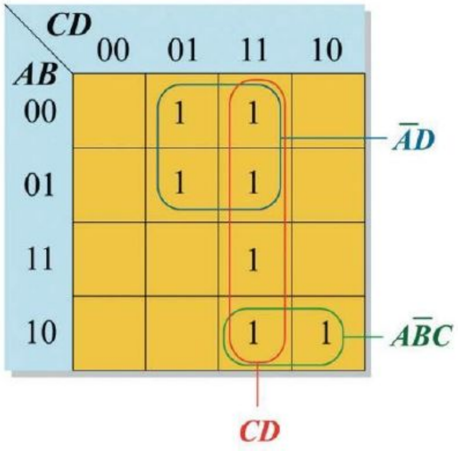
Mapa de Karnaugh para cuatro variables.

$AB \backslash CD$	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$ 0000	$\bar{A}\bar{B}\bar{C}D$ 0001	$\bar{A}\bar{B}CD$ 0011	$\bar{A}\bar{B}C\bar{D}$ 0010
01	$\bar{A}B\bar{C}\bar{D}$ 0100	$\bar{A}B\bar{C}D$ 0101	$\bar{A}BCD$ 0111	$\bar{A}BC\bar{D}$ 0110
11	$AB\bar{C}\bar{D}$ 1100	$AB\bar{C}D$ 1101	$ABCD$ 1111	$ABC\bar{D}$ 1110
10	$A\bar{B}\bar{C}\bar{D}$ 1000	$A\bar{B}\bar{C}D$ 1001	$A\bar{B}CD$ 1011	$A\bar{B}C\bar{D}$ 1010

Así, por ejemplo, para la siguiente función,

$$S = \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BCD + ABCD +$$
$$+ A\overline{B}C\overline{D} + A\overline{B}CD$$

A B C D	S
0 0 0 0	0
0 0 0 1	1
0 0 1 0	0
0 0 1 1	1
0 1 0 0	0
0 1 0 1	1
0 1 1 0	0
0 1 1 1	1
1 0 0 0	0
1 0 0 1	0
1 0 1 0	1
1 0 1 1	1
1 1 0 0	0
1 1 0 1	0
1 1 1 0	0
1 1 1 1	1



La función simplificada queda así:

$$S = \overline{A}D + A\overline{B}C + CD$$

Escribe la tabla de la verdad y simplifica la siguiente función de salida:

$$S = \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}BCD + \overline{A}BC\overline{D} +$$

$$+ A\overline{B}\,\overline{C}D + A\overline{B}CD + AB\overline{C}D + ABCD$$

Escribe la tabla de la verdad y simplifica la siguiente función lógica:

$$S = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}C\overline{D} + \overline{A}B\overline{C}D + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}D +$$
$$+ A\overline{B}C\overline{D} + AB\overline{C}\,\overline{D} + AB\overline{C}D$$

Realiza la tabla de la verdad y simplifica las siguientes funciones:

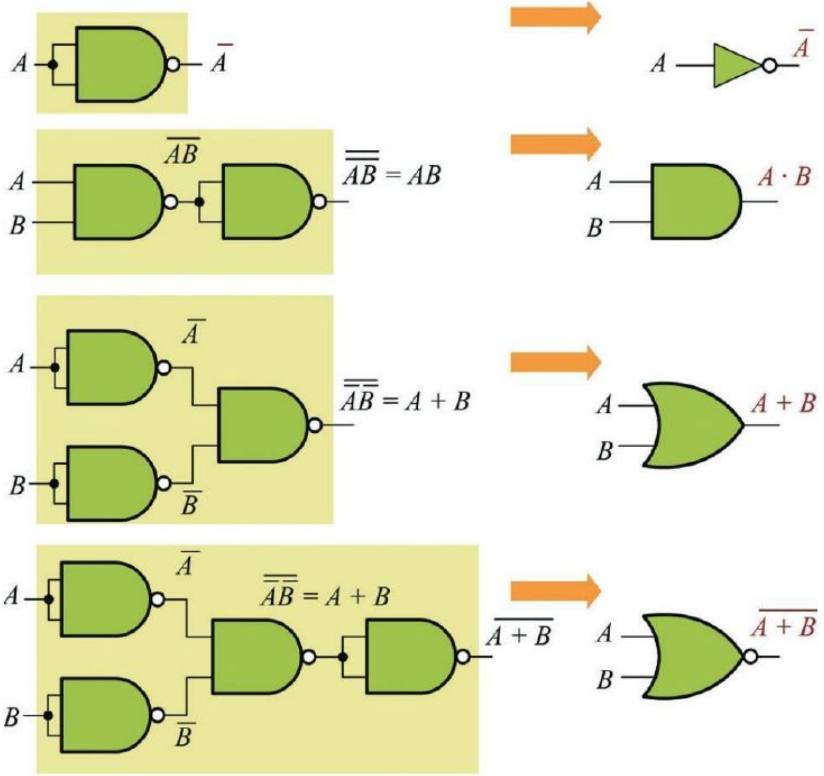
a) $S = \overline{A}B\overline{C} + A\overline{B}C + A\overline{B}\,\overline{C} + ABC$

b) $S = A\overline{B}\,\overline{C}D + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}D +$
$+ \overline{A}B\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}D + A\overline{B}CD + AB\overline{C}D + ABCD$

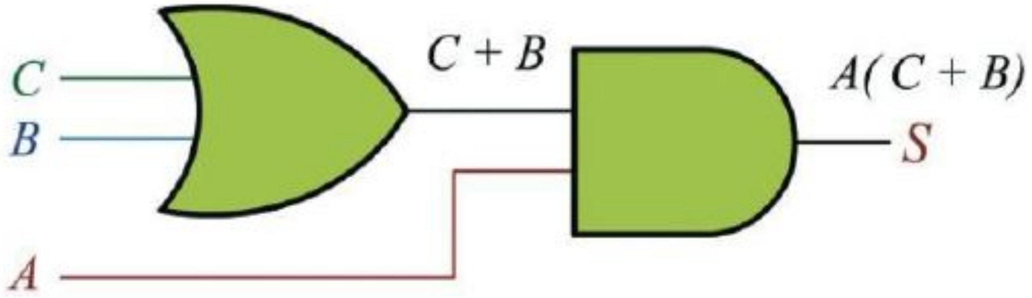
Diseño de circuitos combinacionales con puertas NAND y NOR

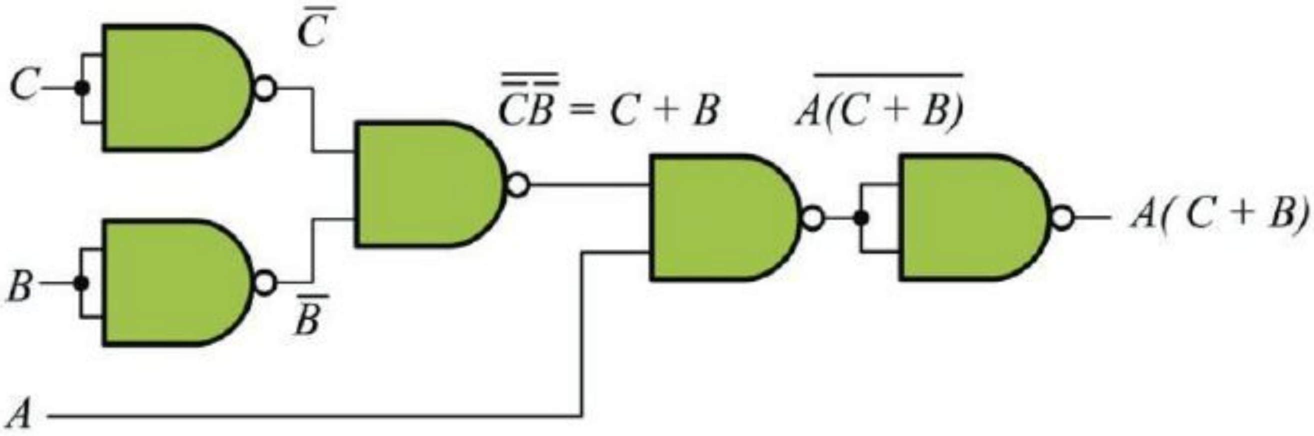
En la práctica resulta muy útil el utilizar en nuestros diseños lógicos solamente un tipo de puerta, como por ejemplo la NAND o la NOR. Aunque esta acción aumente el número de puertas utilizadas tiene sus ventajas, ya que podemos aprovechar todas las puertas que vienen integradas en el chip y no será necesario disponer de todos los tipos de puertas para realizar un diseño.

Equivalencias de diferentes funciones lógicas con puertas NAND



Diseña el circuito lógico de la Figura mediante solo puertas NAND.





$\overline{C}$
C
B
$\overline{B}$
A
$\overline{\overline{C}\,\overline{B}} = C + B$
$\overline{A(C + B)}$
$A(C + B)$

Otro método es aplicar las leyes de Morgan negando dos veces

Convierte la siguiente función lógica para poder ser construida solo con puertas NAND de dos entradas:

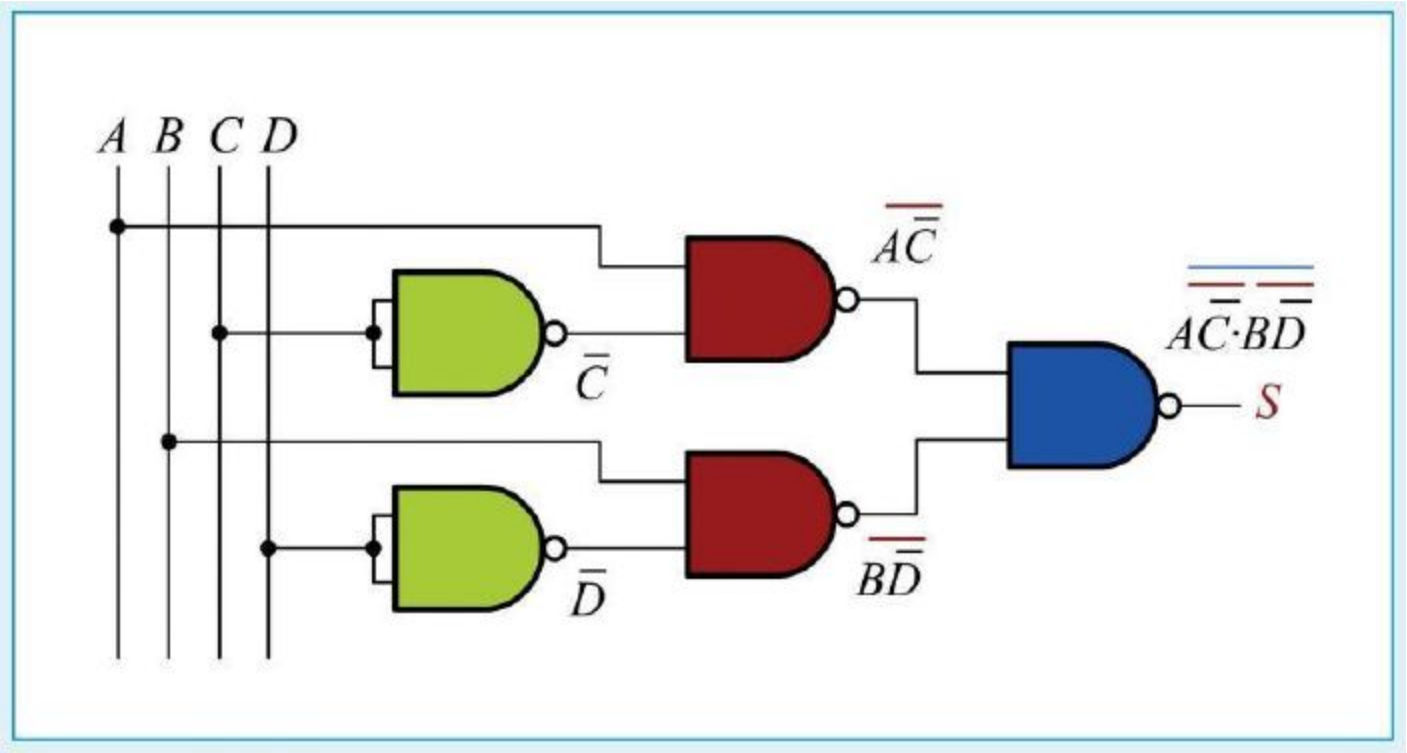
$$S = A\overline{C} + B\overline{D}$$

Si negamos dos veces a toda la función, esta no cambia su valor:

$$S = \overline{\overline{A\overline{C} + B\overline{D}}}$$

Al aplicar el teorema de Morgan, una de las negaciones de sumandos de la función se puede convertir en productos negados:

$$S = \overline{\overline{A\overline{C}} \cdot \overline{B\overline{D}}}$$



A B C D
$\overline{A\overline{C}}$
$\overline{C}$
$\overline{\overline{A\overline{C}} \cdot \overline{B\overline{D}}}$
S
$\overline{D}$
$\overline{B\overline{D}}$

Convierte la siguiente función lógica a puertas NAND de dos entradas:

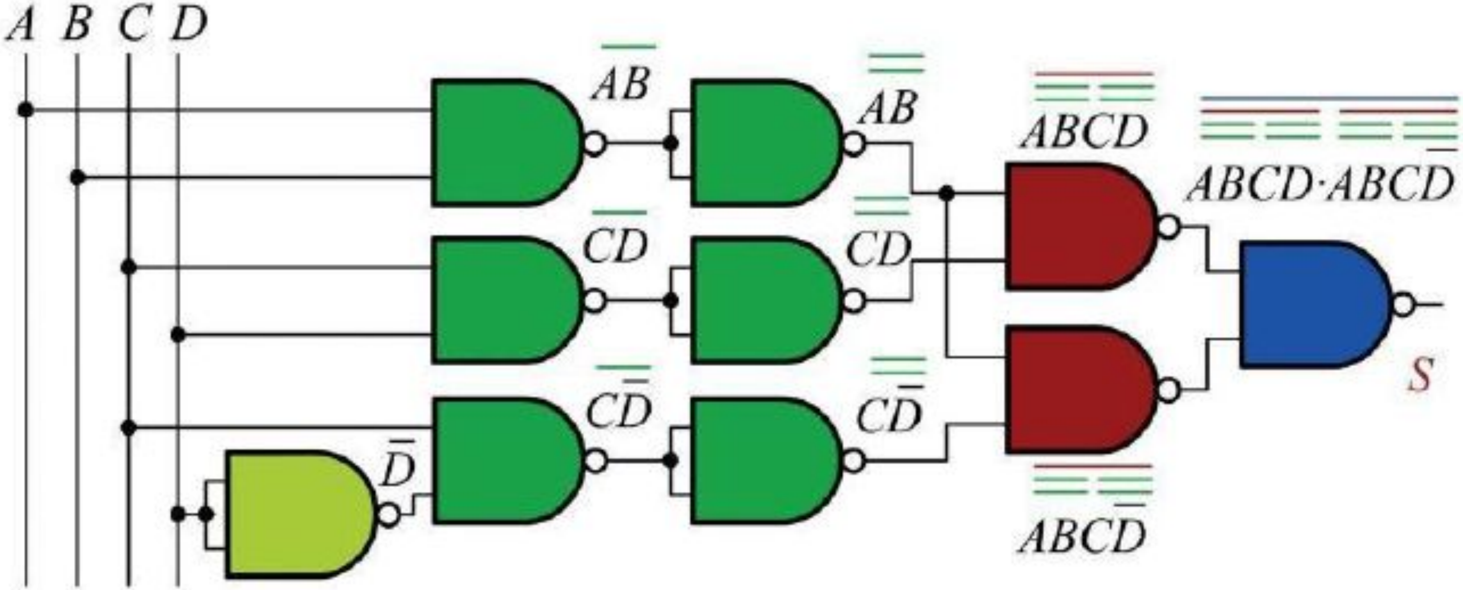
$$S = ABCD + ABC\overline{D}$$

$$S = \overline{\overline{ABCD + ABC\overline{D}}} = \overline{\overline{ABCD} \cdot \overline{ABC\overline{D}}}$$

Para implementar esta función necesitaríamos puertas NAND de cuatro entradas, que podremos convertir en NAND de dos entradas haciendo una doble negación de dos en cada dos de los términos:

$$S = \overline{\overline{\overline{\overline{AB}}\,\overline{\overline{CD}}} \cdot \overline{\overline{\overline{AB}}\,\overline{\overline{C\overline{D}}}}}$$

$$S = \overline{\overline{\overline{\overline{AB}}\,\overline{\overline{CD}}} \cdot \overline{\overline{\overline{AB}}\,\overline{\overline{C\overline{D}}}}}$$



Diseño de circuitos combinacionales

a) Enunciado del problema.

b) Escribir la tabla de la verdad a partir del enunciado.

c) Obtención de la función que se corresponda con la salida que dé como resultado un «1» lógico.

d) Simplificación de la función.

e) Conversión de las funciones, si conviene, para el uso exclusivo de puertas NAND o NOR.

f) Realización del diagrama lógico con puertas.

g) Selección de los circuitos integrados.

h) Montaje práctico del circuito.

A continuación vamos a llevar a cabo el diseño de un sistema de alarma que nos sirva como ejemplo del procedimiento a seguir.

Se dispone de una alarma (S) y de tres sensores A, B y C para su activación (Figura 2.26).

Seguidamente vamos a diseñar un circuito combinacional, de tal forma que la señal de alarma (S) se active cuando se cumplan las siguientes condiciones:

- El sensor A desactivado, el B activado y el C en cualquier posición.
- Los sensores A y B desactivados y el C activado.
- Todos los sensores activados.

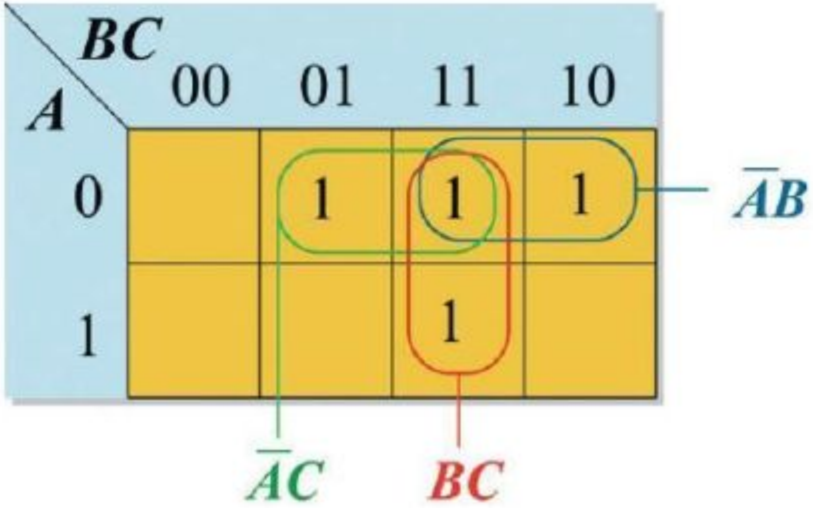
a) La tabla de la verdad para que se cumplan las condiciones del enunciado es la de la Figura

A B C	*S*
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	1

b) La función lógica en forma canónica para $S = 1$, será:

$$S = \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + \overline{A}BC + ABC$$

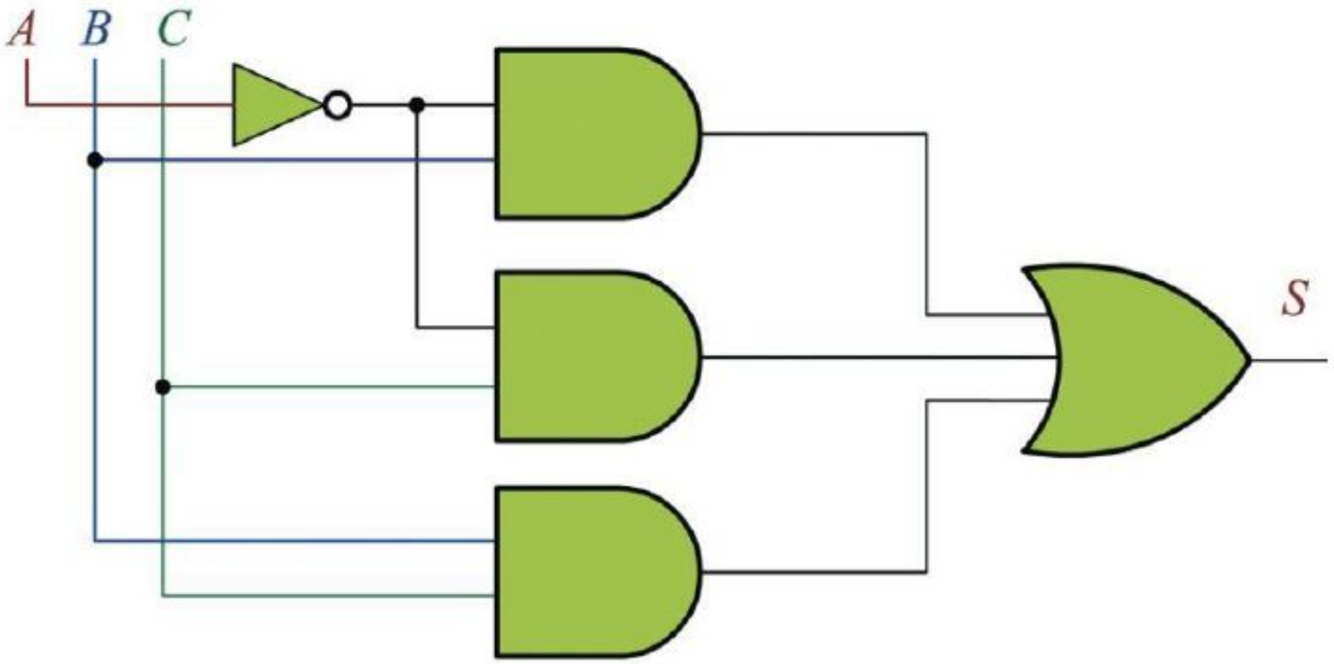
c) Simplificamos la función con el mapa de Karnaugh



La función simplificada queda así:

$$S = \overline{A}B + \overline{A}C + BC$$

d) Para esta función el diagrama lógico es el que se presenta en la Figura



e) Vamos a convertir el circuito para utilizar solo puertas NAND. Para ello aplicamos el teorema de Morgan a la función simplificada.

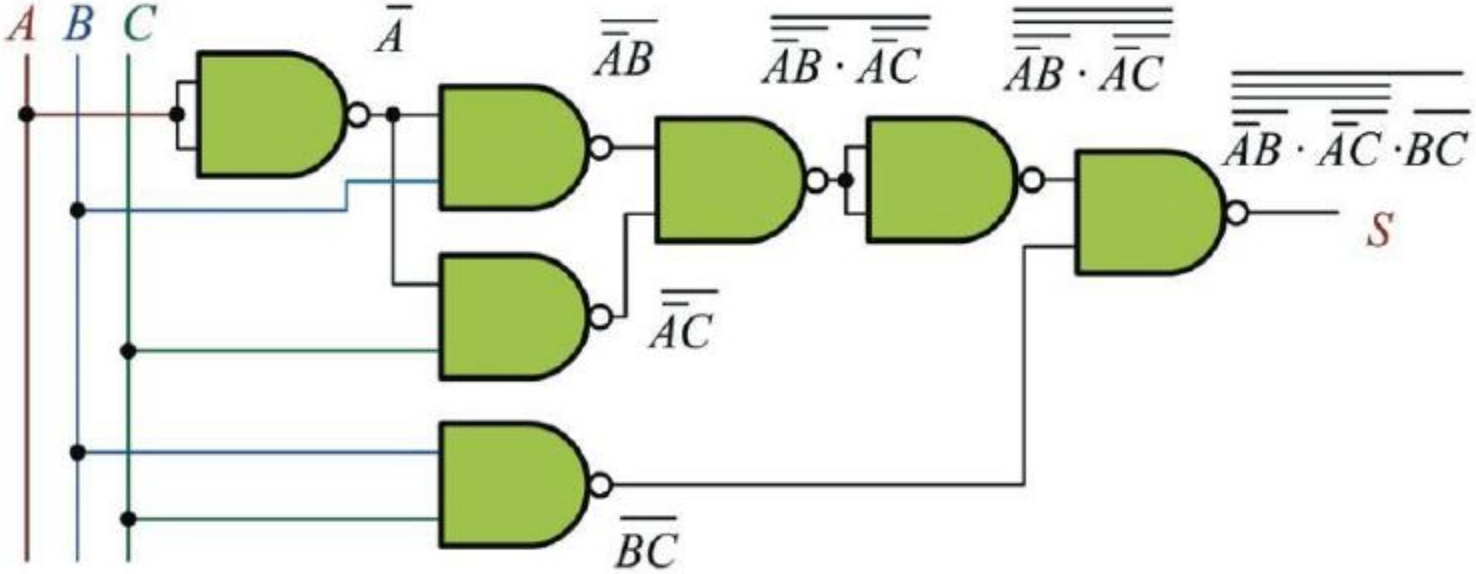
$$S = \overline{\overline{\overline{A}B + \overline{A}C + BC}} = \overline{\overline{\overline{A}B} \cdot \overline{\overline{A}C} \cdot \overline{BC}}$$

Esta función se puede construir ya con puertas lógicas NAND. El inconveniente es que necesitamos mezclar una puerta NAND de tres entradas con puertas NAND de dos entradas.

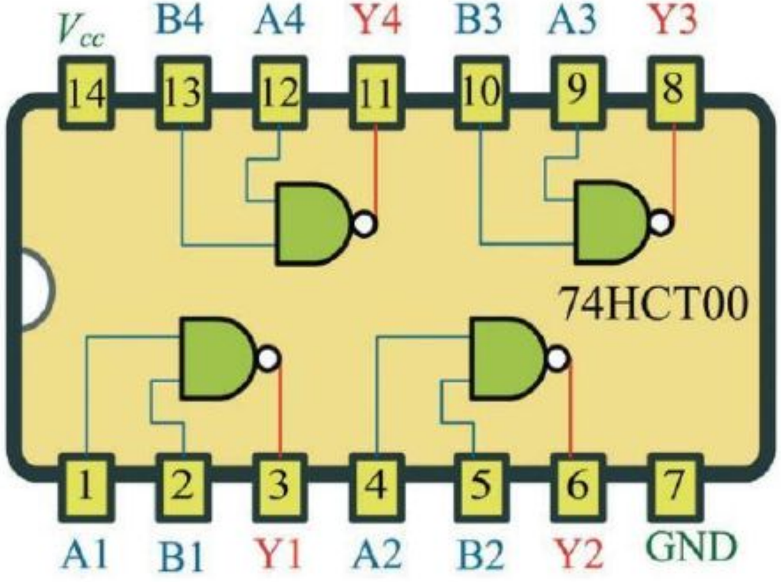
Si queremos utilizar solamente puertas NAND de dos entradas será necesario volver a aplicar el teorema de Morgan.

$$S = \overline{\overline{\overline{\overline{\overline{A}B} \cdot \overline{\overline{A}C}}} \cdot \overline{BC}}$$

El diagrama lógico correspondiente con puertas NAND es el que se muestra en la Figura



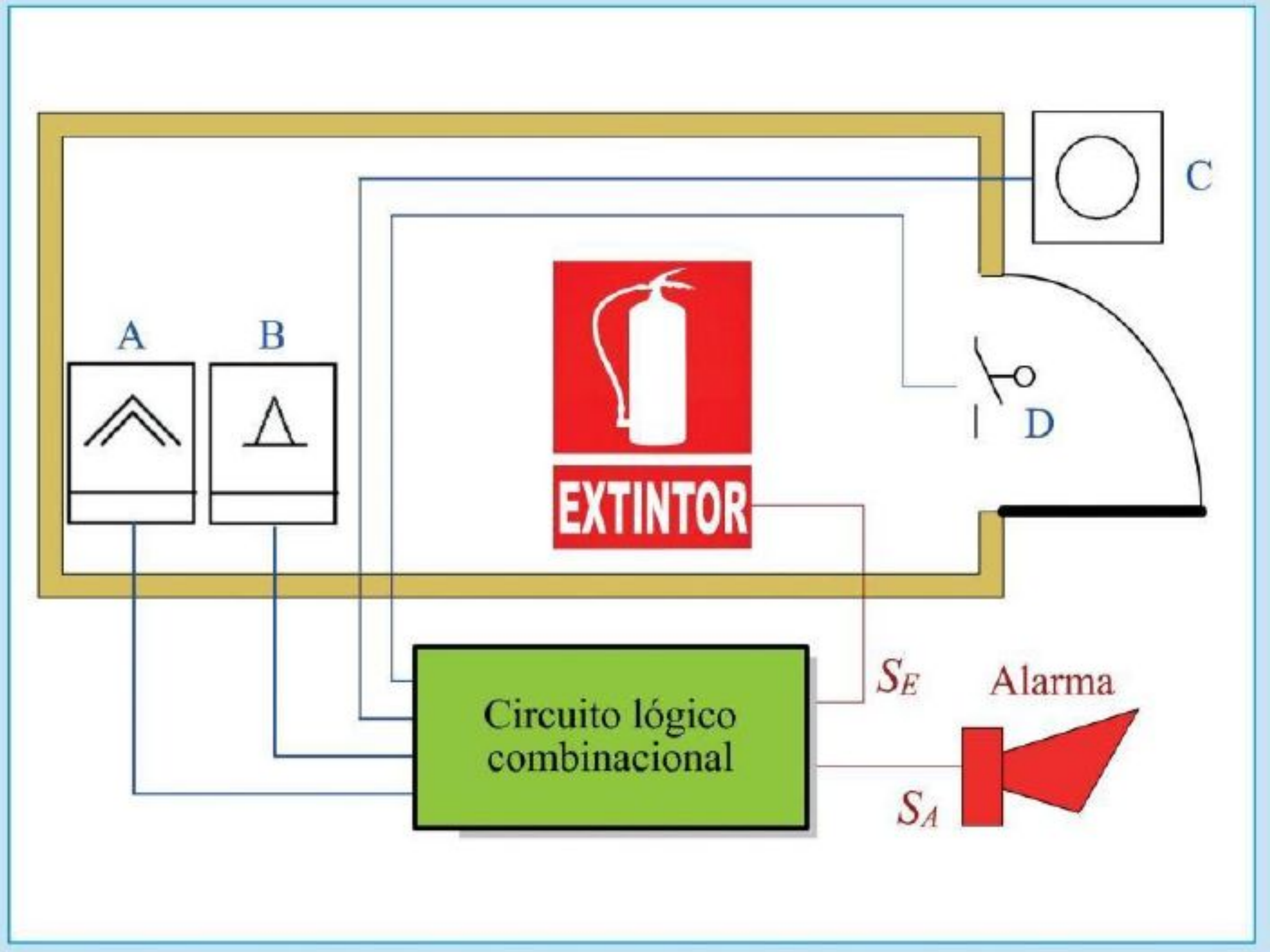
f) Para el montaje práctico de este circuito necesitamos 7 puertas NAND de dos entradas, para lo que se puede utilizar dos circuitos integrados 74HCT00 que contienen 4 puertas NAND cada uno



Se desea diseñar un sistema automático de detección y extinción de incendios para un cuarto de contadores Para ello se dispone de un extintor de gases inertes que se activa mediante una electroválvula. Para la detección automática de incendios se dispone de un detector de humos (sensor A) y de un detector de llama por infrarrojos (sensor B).

La apertura del extintor se producirá en el caso de que cualquiera de estos dos sensores se active. Además, se añade un pulsador (C) en el exterior del cuarto de contadores para tener la posibilidad de producir la activación del extintor de forma manual.

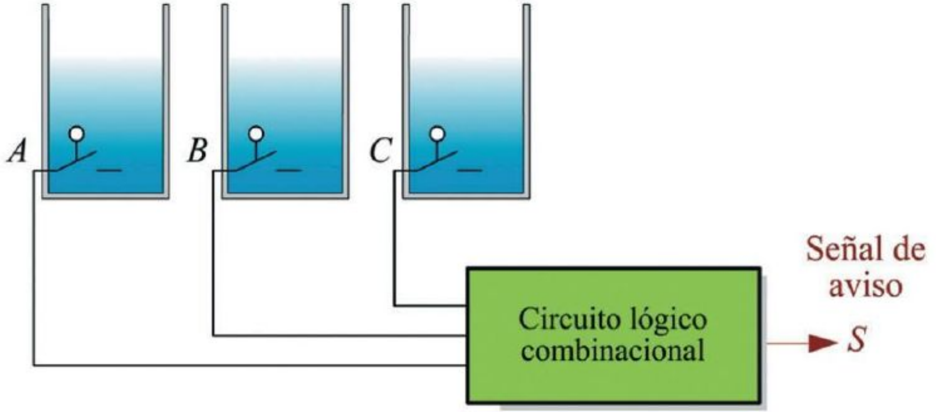
Por razones de seguridad, solo será posible la activación del extintor si la puerta del cuarto está cerrada (sensor D, que con la puerta cerrada emite un 1 lógico). Además, se añade una alarma que se activará en caso de detección automática de humos o de llama, este abierta o cerrada la puerta.



C
A
B
EXTINTOR
D
S_E
Alarma
Circuito lógico combinacional
S_A

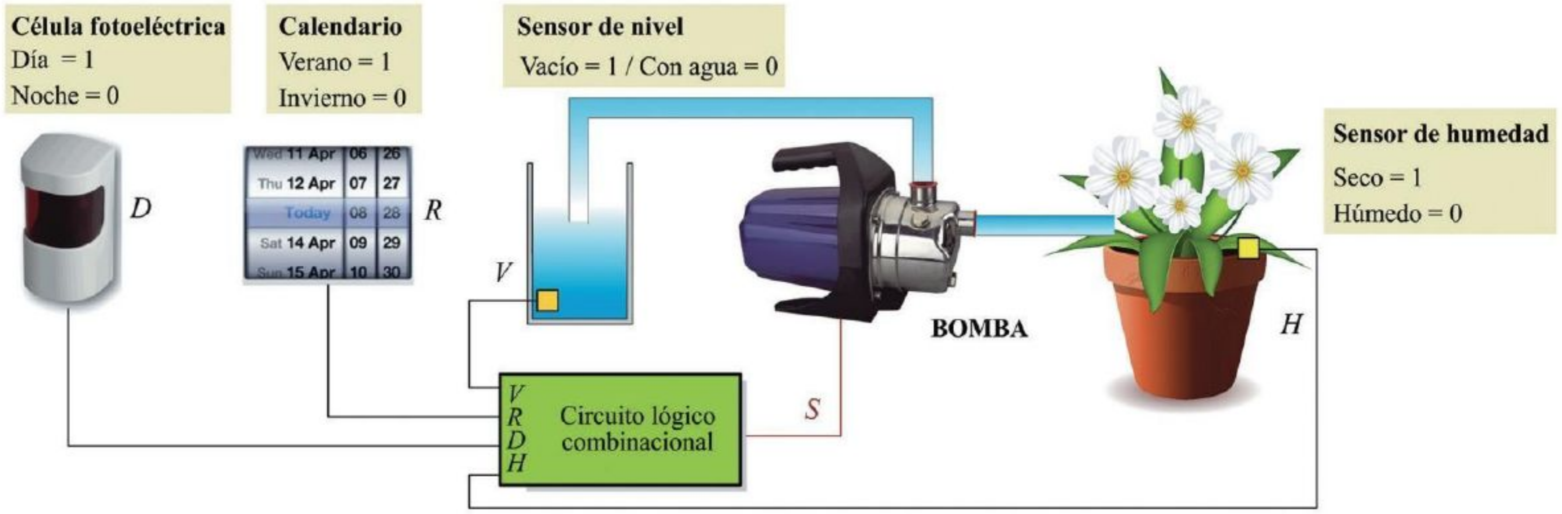
Diseño de circuito lógico para planta depuradora

Una planta depuradora cuenta con tres depósitos de agua. En el fondo de cada depósito se dispone de un sensor de nivel que se activará cuando se encuentre vacío. A continuación vamos a diseñar el circuito lógico con puertas NAND de dos entradas, de tal forma que se active una señal de aviso cuando los sensores indiquen que dos depósitos están vacíos



Diseño de circuito lógico para riego automático

Ahora vamos a diseñar el circuito lógico con puertas NAND de dos entradas que automatice un sistema de riego como el mostrado en la Figura



El circuito deberá accionar la bomba de riego solamente cuando se cumplan las siguientes condiciones:

- Solo se riega si la tierra está seca.
- Para evitar que la bomba se estropee funcionando en vacío, esta nunca se accionará cuando el depósito de agua esté vacío.
- Si hay restricciones en el riego (época de verano), solo se podrá regar de noche.
- En el resto del año (si no hay restricciones) se podrá regar de día y de noche.

Simplifica las siguientes funciones

a) $S = A\overline{B} + \overline{A}\,\overline{B} + \overline{A}B$

b) $S = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + A\overline{B}C + \overline{A}B\overline{C} + AB\overline{C}$

c) $S = ABC + \overline{A}\,\overline{B}C + \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}$

d) $S = \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BCD + ABCD$

e) $S = \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}C\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} +$
$+ ABCD + A\overline{B}C\overline{D}$

Convierte las siguientes funciones para que puedan ser realizadas solo con puertas NAND de dos entradas. Una vez realizada la conversión dibuja los diagramas lógicos.

a) $S = \overline{A} + B\overline{C}$

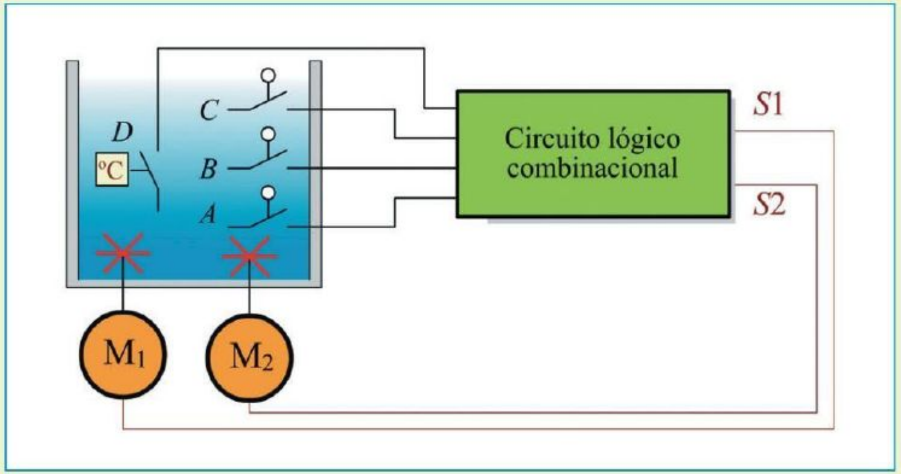
b) $S = \overline{A}C + AD + \overline{B}D$

c) $S = \overline{A}D + A\overline{B}C + CD$

Una trituradora posee dos niveles de trituración, de tal forma que cuando el depósito de trituración está a un nivel medio (sensor *B*) se conecta el motor n.º 1 (*S*1).

Cuando se alcanza un nivel de llenado alto (sensor *C*) se conectan a la vez los motores n.º 1 y n.º 2 (*S*2). Si el nivel de llenado está por debajo del mínimo (sensor *A*) los dos motores se desconectan. Por otro lado, los motores de la trituradora solo funcionarán si se activa un sensor de temperatura (sensor *D*) que indica que la mezcla se realiza a 50 ºC . Escribe la tabla de la verdad y diseña el circuito lógico combinacional con puertas lógicas para el funcionamiento de los dos motores de la trituradora.

Nota: El sensor A proporciona un «1» lógico cuando detecta el depósito lleno. Los sensores *B* y *C* proporcionan un «1» lógico cuando se alcanza dicho nivel.



Se dispone de una alarma (*S*) de cuatro sensores *A*, *B*, *C* y *D* para su activación

Diseña el circuito combinacional de tal forma que la señal de alarma (*S*) se active cuando se cumplan las siguientes condiciones:

- El sensor *A* desactivado, el *B* activado, el *C* desactivado y el *D* en cualquier posición.
- Los sensores *A* y *B* desactivados y el *C* y el *D* activados.
- El sensor *A* activado, el *B* desactivado, el *C* desactivado y el *D* activado.
- Todos los sensores activados.