Sistemas automáticos, circuitos digitales y combinacionales: Álgebra de Boole. Puertas y funciones lógicas. Simplificación de funciones lógicas.

INSTITUTO de ENSEÑANZAS a DISTANCIA de ANDALUCÍA

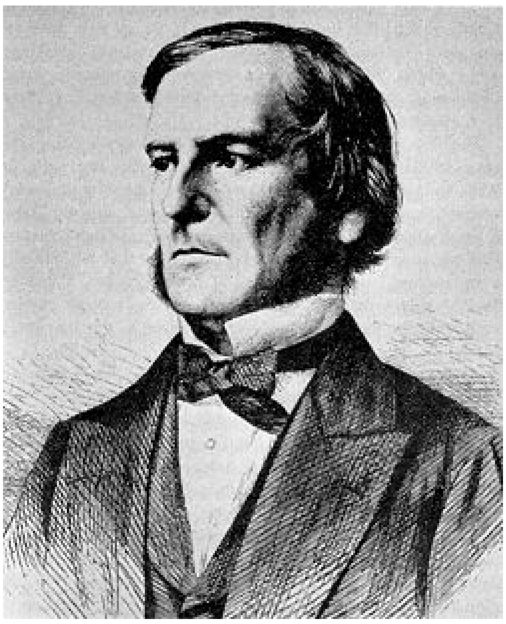
PAC
Preparación Acceso a CFGS

Tecnología Industrial
Contenidos

Sistemas automáticos, circuitos digitales y combinacionales:
Álgebra de Boole. Puertas y funciones lógicas.
Simplificación de funciones lógicas.

El álgebra de Boole se denomina así en honor a **George Boole**, matemático inglés 1815 - 1864, que fue el primero en definirla como parte de un sistema lógico, a mediados del siglo XIX. El álgebra de Boole fue un intento de utilizar las técnicas algebraicas para tratar expresiones de la lógica proposicional.

El álgebra opera con **variables booleanas**, que son aquellas que sólo pueden tomar dos valores (0 y 1), estos valores no representan números si no estados. Ejemplo: pueden simbolizar si un interruptor está abierto (0), o cerrado (1), si conduce o no conduce, si hay tensión o no.



George Boole
Imagen de Hask en **Wikimedia**. **Dominio público**

En la actualidad, esta herramienta se aplica de forma generalizada en el ámbito del diseño de control electrónico de procesos industriales. Estos métodos de control se basan en el uso de sistemas denominados puertas lógicas.

1. Operaciones básicas en el álgebra de Boole

En primer lugar definiremos unos conceptos que necesitamos utilizar:

Función lógica:

Toda aquella variable binaria cuyo valor depende de una **expresión algebraica** constituida por otras variables que se encuentran relacionadas entre si por determinadas operaciones.

Por ejemplo sea la expresión:

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

La interpretación de esta función será que la salida S, tomará el valor "uno" cuando lo hagan las variables A y B o la variable C.

Función canónica:

Expresión lógica en la que todos sus términos contienen todas las variables de entrada, bien afirmadas o bien negadas.

Por ejemplo 1 :

La siguiente función lógica **S**, está compuesta por tres términos, y cada término tiene en su composición las tres variables del sistema (A, B y C) . Está función S, está expresada en **forma canónica**.

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

Por ejemplo 2 :

La siguiente función lógica **S**, está compuesta por tres términos, pero no todos estos términos tienen en su composición las tres variables del sistema (A, B y C). Está función **S**, está expresada en forma **no canónica.**

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

Importante

Tabla de verdad:

Tabla formada por **n** columnas de entradas o variables (**A, B, C**....) y otra más de salida (**f**), y por $2^{\mathbf{n}}$ filas, de modo que cada una de las filas representa cada una de las combinaciones posibles que pueden tener las variables de entrada, y el valor que toma la variable de salida para cada combinación de las variables de entrada.

Los posibles valores que pueden tomar las distintas variables representan estados diferentes de un dispositivo, y están representados por los valores lógicos **"cero"** (apagado, abierto, desconectado, ausencia, nivel bajo,...) o **"uno"** (encendido, cerrado, conectado, presencia, nivel alto,...).

Cuando en una tabla de verdad una variable está representada por el valor "x" quiere decir que su valor no está definido o que es indiferente el valor que tome para la respuesta de la función lógica.

En la figura adjunta se representa la tabla de verdad correspondiente a la función lógica:

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

ENTRADAS		SALIDA
A	B	f
0	0	0
0	1	1
1	0	1
1	1	0

Lógica positiva y lógica negativa:

Las variables lógicas pueden tomar exclusivamente los valores 0 y 1. Estos valores se pueden conseguir eléctricamente por dos procedimientos diferentes, según como se asignen los niveles de tensión se habla de lógica positiva o negativa.

Si se asigna al 1 lógico el valor más positivo de tensión y el 0 lógico al valor de menor tensión, estaremos ante **lógica positiva**.

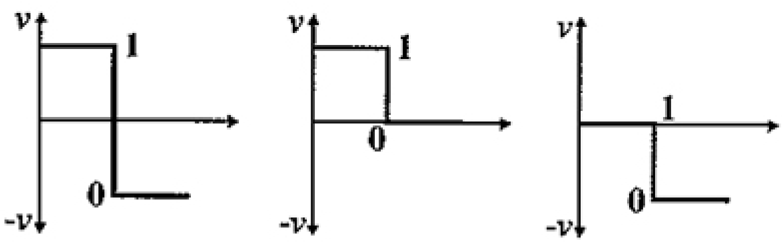


Imagen de elaboración propia

Si se asigna al 1 lógico el valor más negativo de tensión, estaremos ante **lógica negativa**.

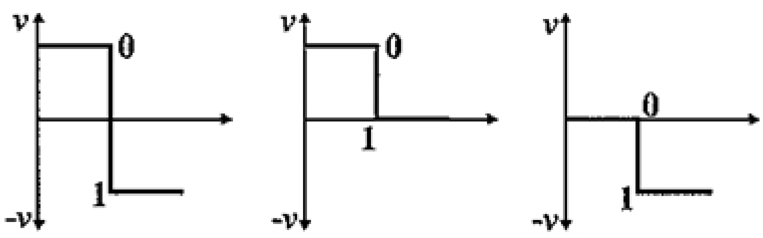


Imagen de elaboración propia

1.1. Operaciones básicas

En el algebra de Boole se definen tres operaciones básicas que son:

- Producto lógico, o intersección.
- Suma lógica o unión.
- Negación, complementación o inversión.

Las operaciones básicas del álgebra de Boole se suelen implementar por medio de circuitos electrónicos, a estos componentes se les llama **puertas lógicas**.

Veamos ahora cada una de ellas por separado:

Producto lógico. Puerta AND. Y. &.

Función representada por la expresión:

$$S = A \cdot B$$

Esta función responde a la tabla de verdad que se acompaña, debe entenderse como: La salida correspondiente a la función lógica producto toma valor 1 solamente cuando también son 1 los valores de las entradas A y B.

dec	A	B	$S = A \cdot B$
0	0	0	**0**
1	0	1	**0**
2	1	0	**0**
3	1	1	**1**

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA (ANSI):

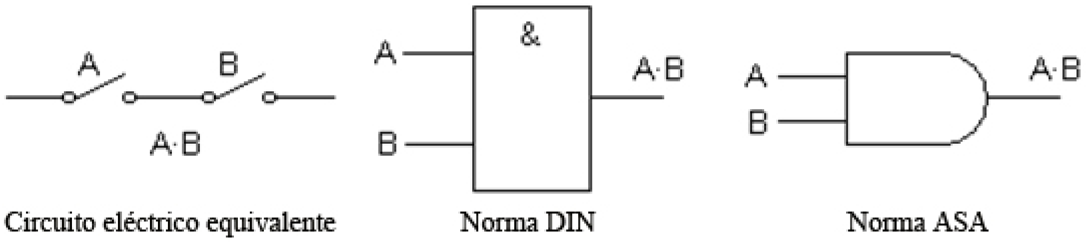


Imagen de elaboración propia

Suma lógica. Puerta OR. O.

Función representada por la expresión:

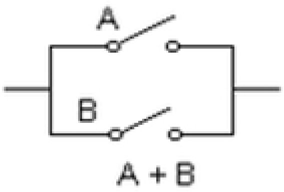
$$S = A + B$$

Esta función responde a la tabla de verdad que se acompaña, debe ser leida como: La salida correspondiente a la función lógica suma toma valor 1 cuando también son 1 cualquiera de las entradas A y B, o las dos simultáneamente.

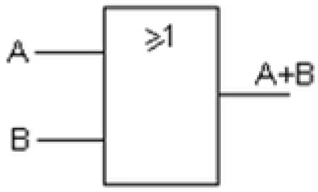
dec	A	B	$S = A + B$
0	0	0	**0**
1	0	1	**1**
2	1	0	**1**
3	1	1	**1**

Imagen de elaboración propia

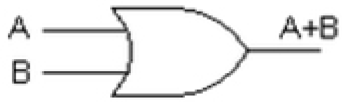
Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA:



Circuito eléctrico equivalente



Norma DIN



Norma ASA

Imagen de elaboración propia

Negación, complementación o inversión. NOT.

Función representada por la expresión:

$$S = \overline{A}$$

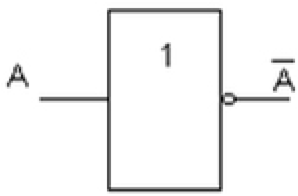
Esta función responde a la tabla de verdad que se acompaña y debe entenderse como: La salida correspondiente a la función lógica complementación toma valor 1 cuando la entrada es 0 y viceversa.

A	$S = \bar{A}$
0	**1**
1	**0**

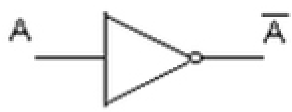
Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muetra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según las normas DIN y ASA:

Circuito eléctrico equivalente



Norma DIN



Norma ASA

Imagen de elaboración propia

2. Otras operaciones lógicas

Importante

Puerta NAND. No Y.

Función representada por la expresión:

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

Esta función responde a la tabla de verdad que se acompaña, debe interpretarse como: La salida correspondiente a la función lógica toma siempre valor 1 salvo cuando A y B tienen a la vez valor 1.

La salida es el producto negado de las variables de entrada. Es decir la función NAND es la asociación de una puerta AND y de la función NOT.

dec	A	B	$S = \overline{A \cdot B}$
0	0	0	**1**
1	0	1	**1**
2	1	0	**1**
3	1	1	**0**

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA:

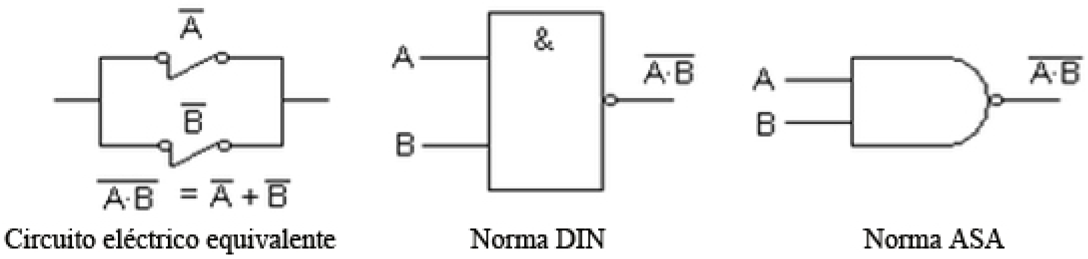


Imagen de elaboración propia

Importante

Puerta NOR. No O.

Función representada por la expresión:

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

Esta función responde a la tabla de verdad que se acompaña, debe interpretarse como: La salida correspondiente a la función lógica toma valor 1 solamente cuando A y B tienen a la vez valor 0.

La salida es la suma negada de las variables de entrada. Es decir la función NOR es la asociación de una puerta OR y de la función NOT.

dec	A	B	$S = \overline{A + B}$
0	0	0	**1**
1	0	1	**0**
2	1	0	**0**
3	1	1	**0**

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA:

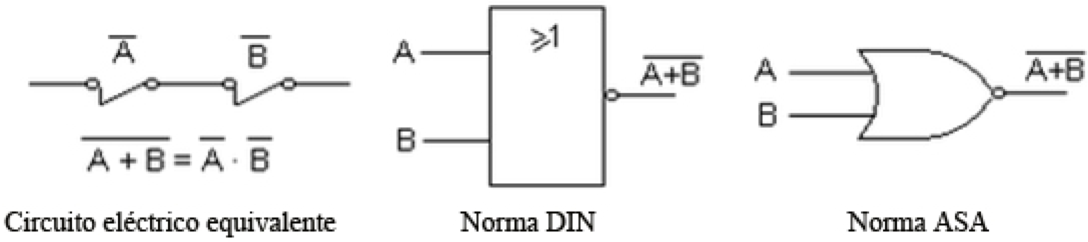


Imagen de elaboración propia

Importante

Puerta OR EXCLUSIVA. XOR.

Función representada por la expresión:

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

Esta función responde a la tabla de verdad que se acompaña, que debemos leer como: La salida correspondiente a la función lógica toma valor 1 cuando A ó B pero no las dos, toman el valor 1, es decir cuando una variable está afirmada y la otra está negada, o viceversa.

dec	A	B	$S = \bar{A} \cdot B + A \cdot \bar{B}$
0	0	0	**0**
1	0	1	**1**
2	1	0	**1**
3	1	1	**0**

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA:

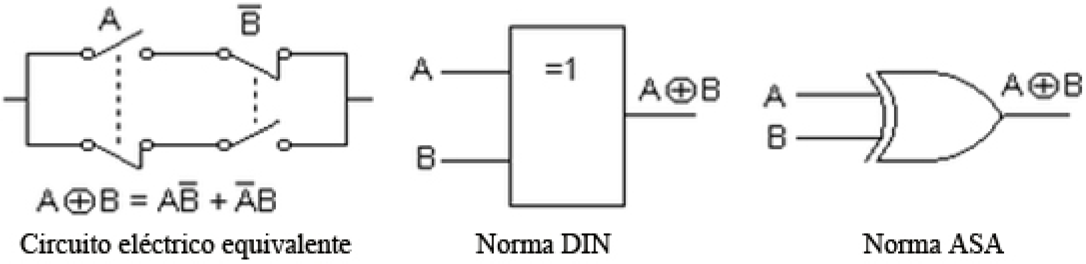


Imagen de elaboración propia

Importante

Puerta NOR EXCLUSIVA. XNOR.

Función representada por la expresión:

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

Esta función responde a la tabla de verdad que se acompaña, que debemos leer como: La salida toma el valor 1 cuando las dos entradas son 1 o las dos entradas son 0.

dec	A	B	$S = A \cdot B + \bar{A} \cdot \bar{B}$
0	0	0	**1**
1	0	1	**0**
2	1	0	**0**
3	1	1	**1**

Imagen de elaboración propia

Esta operación puede representarse a través de un circuito eléctrico equivalente que se muestra en la siguiente tabla. En ella también se muestran los símbolos de este tipo de puertas según sean normas DIN o ASA:

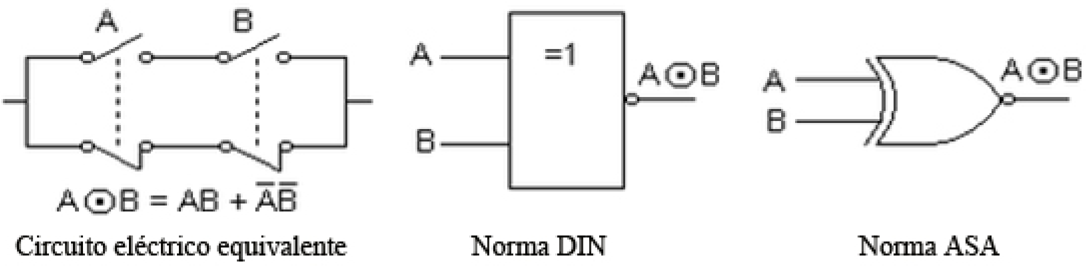
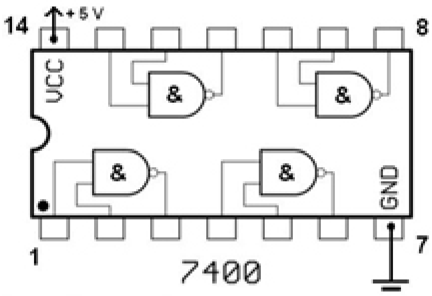


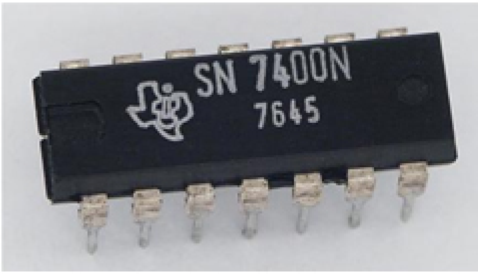
Imagen de elaboración propia

Para saber más

- Comercialmente las puertas lógicas se distribuyen en circuitos integrados (CI), que disponen de catorce patillas o pines.
- Todos ellos presentan una muesca que se debe ver a la izquierda para empezar a contar las patillas desde abajo de la muesca y en sentido contrario a las agujas de un reloj.
- La patilla número 7 se conecta a masa y la número 14 a tensión de alimentación +Vcc, por lo que quedan 12 patillas para utilizar distintas entradas y salidas.
- Cada circuito integrado se identifica mediante un código, que indica el número y el tipo de puertas de que consta.

En la fotografía se muestra un circuito integrado que contiene cuatro puertas NAND de dos entradas, con las conexiones como se puede observar en el esquema adjunto.





Imágenes de elaboración propia

3. Álgebra de Boole, postulados y teoremas

Vamos a enumerar las distintas operaciones, postulados, leyes y teoremas, para familiarizarnos con el manejo del algebra de Boole.

Postulados:

Respecto a la suma	Respecto al producto
$A+0=A$	$A\cdot 0=0$
$A+1=1$	$A\cdot 1=A$
$A+A=A$	$A\cdot A=A$
$A+\bar{A}=1$	$A\cdot\bar{A}=0$
$\bar{\bar{A}}=A$	

Imagen de elaboración propia

Leyes y teoremas:

Leyes/Teoremas	Respecto a la suma	Respecto al producto
Conmutativa.	$A+B=B+A$	$A\cdot B=B\cdot A$
Asociativa	$A+(B+C)=(A+B)+C$	$A\cdot(B\cdot C)=(A\cdot B)\cdot C$
Distributiva	$A\cdot(B+C)=A\cdot B+A\cdot C$	$A+(B\cdot C)=(A\cdot B)+(A\cdot C)$
Cancelación	$(A\cdot B)+A=A$	$(A+B)\cdot A=A$
De Morgan	$\overline{A+B}=\bar{A}\cdot\bar{B}$	$\overline{A\cdot B}=\bar{A}+\bar{B}$

Imagen de elaboración propia

Reflexiona

Comprueba que se cumplen las leyes de DeMorgan, mediante el método de las tablas de verdad, e implementa con los distintos tipos de puertas lógicas los resultados que has obtenido.

Mostrar retroalimentación

Es este un caso particular del **principio de dualidad**.

A toda relación o ley lógica le corresponderá su dual, que se forma permutando las operaciones suma lógica por producto lógico, los 0 por 1 y los 1 por 0, y las variables afirmadas por negadas y viceversa.

4. Obtención de la función lógica a partir de la tabla de verdad

Una vez que hemos obtenido la tabla de verdad de un sistema o problema, existen dos métodos distintos para obtener la función lógica que la representa. En ambos métodos obtenemos una forma canónica. Como recordarás una forma canónica es la expresión de la función lógica sin ninguna simplificación, es decir, en cada término estarán presentes todas las variables de entrada, negadas o sin negar.

Los dos métodos son:

Productos lógicos "Minterms" (implementación por unos)

El proceso consiste en tomar todas las combinaciones de las variables de entrada que provocan que la función lógica presente un 1 en la salida.

Cada una de estas combinaciones de las variables de entrada será un sumando, constituido por el producto de todas las variables de entrada, en el que las variables estarán negadas cuando tomen el valor 0 y estarán afirmadas cuando tomen el valor 1.

La expresión de la función canónica será la suma de todos los productos equivalentes a las combinaciones de las variables de entrada que hacen que la salida sea un 1.

Sumas lógicas "Maxterms" (implementación por ceros)

El proceso consiste en tomar todas las combinaciones de las variables de entrada que provocan que la función lógica presente un 0 en la salida.

Cada una de estas combinaciones de las variables de entrada será un producto, constituido por la suma de todas las variables de entrada, en el que las variables estarán negadas cuando tomen el valor 1 y estarán afirmadas cuando tomen el valor 0. La expresión de la función canónica será el producto de todas las sumas equivalentes a las combinaciones de las variables de entrada que hacen que la salida sea un 0.

Cualquiera de los dos métodos es igual de eficaz y rápido, aunque suele tenerse tendencia a resolver los problemas utilizando el método de minterms, es decir suma de productos, o el método de los unos, aunque realmente deberíamos escoger un método u otro según el que produjese funciones canónicas más cortas.

Parece complicado, verás que es mucho más sencillo de entender si se hace un ejemplo.

5. Implementación de circuitos con puertas NAND y NOR

Para la implementación de circuitos lógicos se pueden utilizar cualquier tipo de puertas. Sin embargo la tendencia más común es implementar un circuito empleando solamente un tipo de puertas. De este modo se abaratan costes.

Este método de implementación solo se puede realizar con puertas NAND o NOR, ya que solo estas dos puertas lógicas son **universales**, es decir se puede realizar cualquier circuito lógico y sustituir cualquier puerta empleando únicamente este tipo de puertas, para ello debemos seguir un cierto protocolo aprovechando que una doble negación es igual a una afirmación ($A = \overline{\overline{A}}$).

Siempre podemos negar una expresión lógica dos veces, tantas veces como necesitemos y ésta quedará inmutable, si luego aplicamos el teorema de DeMorgan a una de las dos negaciones anteriores, conseguimos que un producto negado se convierta en una suma de variables negadas, o bien que una suma negada se convierta en un producto de variables negadas.

Ejemplo de ejecución de cualquier puerta empleando solamente puertas NOR.

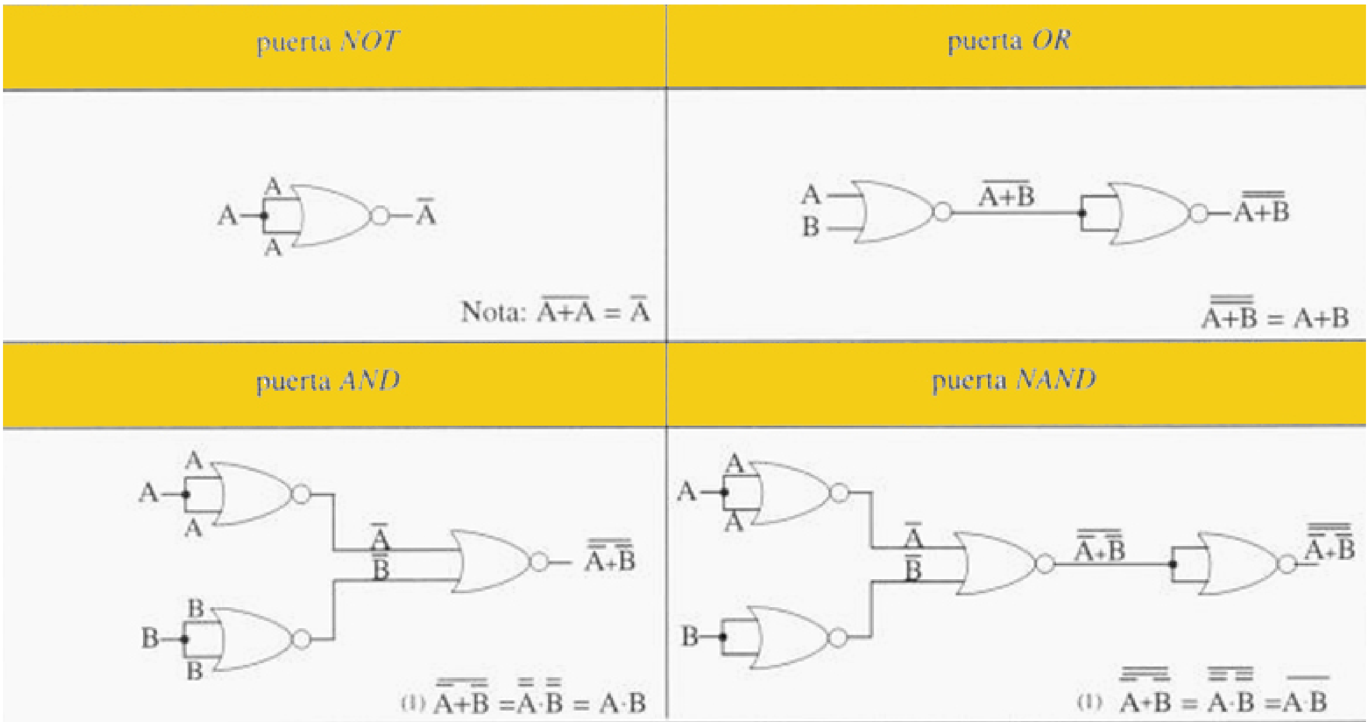


Imagen de elaboración propia

De igual forma podemos utilizar puertas lógicas NAND:

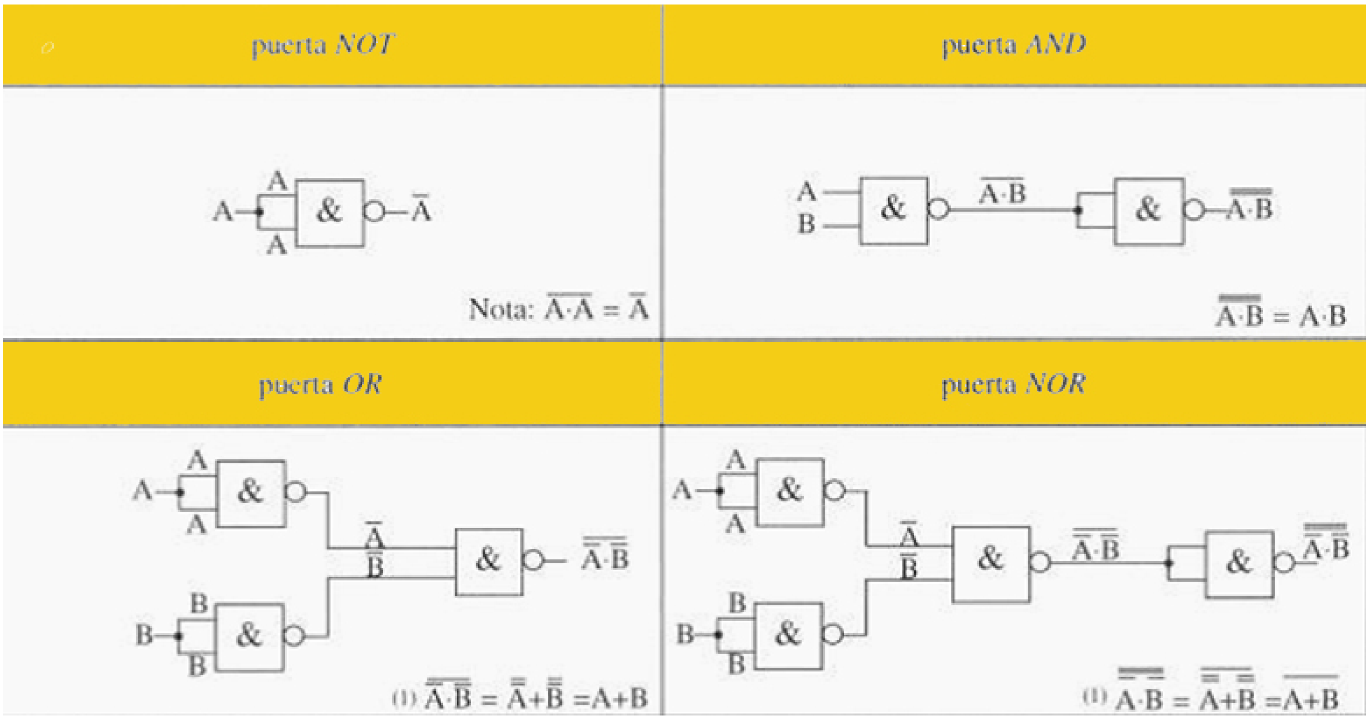


Imagen de elaboración propia

6. Simplificación de funciones lógicas

INSTITUTO de ENSEÑANZAS a DISTANCIA de ANDALUCÍA

2º de Bachillerato

Tecnología Industrial II

Contenidos

Circuitos y sistemas lógicos:
Simplificación de funciones lógicas

Una vez obtenida la función canónica de una expresión lógica, se debe buscar una expresión simplificada de ésta, con el menor número de términos. Con ello se consigue minimizar el número de errores posibles y abarata su implementación.

Existen dos métodos para conseguir simplificar funciones lógicas: la simplificación por el método algebraico y la simplificación por el método de Karnaugh.

Vamos a ver cómo se aplican estos métodos en la simplificación de funciones lógicas.

6.2. Método de Karnaugh

El método de Karnaugh es un método más rápido y eficaz, sobre todo ante problemas complejos, que el método algebraico.

Se basa en la construcción de unas **tablas** con la característica de que entre cada celda y su contigua o adyacente solamente cambia el valor de una variable de entrada.

En este punto vamos a mostrar el funcionamiento del método de Karnaugh. A continuación se adjuntan los mapas de Karnaugh para tres y cuatro variables, indicando en el interior de cada celda a que combinación de variables de entrada corresponde.

Presta atención a la colocación de las filas de cabecera: **00 01 11 10**

C \ AB	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

CD \ AB	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}B\bar{C}\bar{D}$	$AB\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$
01	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}B\bar{C}D$	$AB\bar{C}D$	$A\bar{B}\bar{C}D$
11	$\bar{A}\bar{B}CD$	$\bar{A}BCD$	$ABCD$	$A\bar{B}CD$
10	$\bar{A}\bar{B}C\bar{D}$	$\bar{A}BC\bar{D}$	$ABC\bar{D}$	$A\bar{B}C\bar{D}$

Imagen de elaboración propia

Se colocará un 1 en las celdas correspondientes a las combinaciones de las variables de entrada que hacen que la salida sea un 1.

Hay que tener en cuenta que las tablas son contiguas por los flancos extremos, es decir en una tabla de cuatro variables de entrada, las celdas de la columna de la derecha son contiguas a las de la columna de la izquierda, e igualmente ocurre con las casillas de la fila superior y las de la fila inferior.

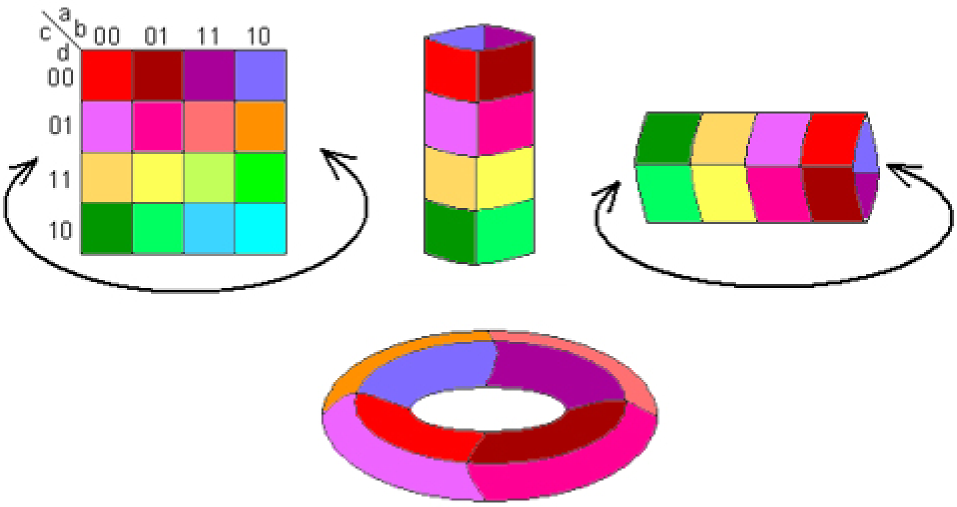


Imagen de elaboración propia

Es decir se puede considerar que se doblan formando un cilindro, como se observa en la figura anterior, y se vuelve a doblar de nuevo formando un toroide.

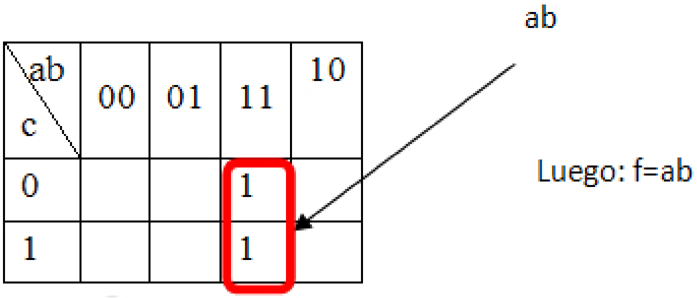
Las celdas que ocupan los extremos también son adyacentes, por lo anteriormente descrito.

El método se basa en **hacer agrupamientos con el mayor número de celdas posible**, siempre que sean potencias de dos (16, 8, 4, 2), tratando que todas las celdas que contengan 1 pertenezcan a agrupaciones de celdas. Consideraciones:

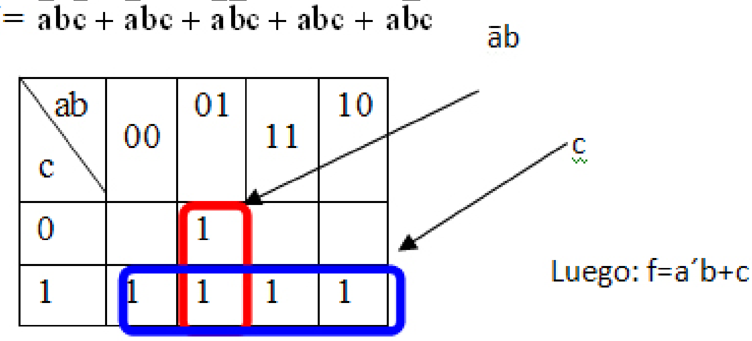
6.3. Ejemplos de simplificación por el método de Karnaugh

A continuación se muestras cinco ejemplos correspondientes a tablas de Karnaugh de tres y cuatro variables:

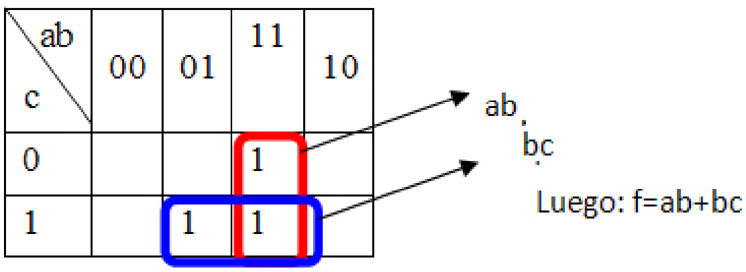
Ejemplo 1. Con 3 variables: $f = ab\bar{c} + abc$



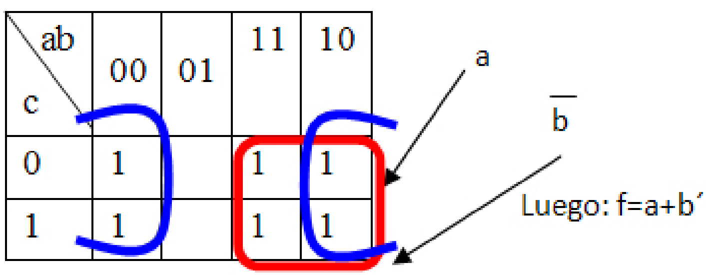
Ejemplo 2. Con 3 variables: $f = \bar{a}b\bar{c} + \bar{a}bc + \bar{a}\bar{b}c + abc + a\bar{b}c$



Ejemplo 3. Con 3 variables: $f = \bar{a}bc + abc + ab\bar{c}$

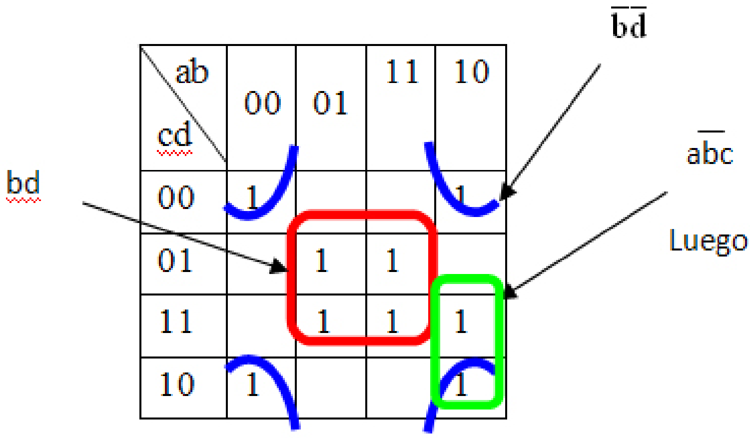


Ejemplo 4. Con 3 variables: $f = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + ab\bar{c} + abc + a\bar{b}\bar{c} + a\bar{b}c$



Ejemplo 5. Con 4 variables:

$$f = \bar{a}b\bar{c}d + \bar{a}bcd + ab\bar{c}d + abcd + \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}c\bar{d} + a\bar{b}\bar{c}\bar{d} + a\bar{b}c\bar{d} + a\bar{b}cd$$



Luego: f=bd+b´d´+ab´c

Recurso propio

- Se debe tratar de que no haya celdas comunes a varias agrupaciones, siempre que sea posible. Pero no está prohibido que haya celdas que pertenezcan a más de una agrupación.
- Las agrupaciones serán horizontales y verticales; las diagonales no están permitidas.
- Aunque si están permitidos las verticales y horizontales que lleguen al final de la fila o la columna, y vuelvan a enlazarse otra vez al inicio, o viceversa.
- Se debe procurar que haya el menor número de agrupaciones con el mayor número de unos posible.
- La función simplificada tendrá tantos términos como agrupaciones o bolsas tenga el mapa de Karnaugh.

En el siguiente video se explica detalladamente el método:

En el apartado siguiente se muestran ejemplos de simplificación por el método de Karnaugh.

Fuentes para el profesorado

Descargar CMAP.

Resumen

Importante

1. Operaciones básicas

Operación	Símbolo	Salida
Producto lógico	$a \cdot b = c$	La salida c toma el valor 1 si a y b también lo son.
Suma lógica	$a + b = c$	La salida c toma el valor 1 si a, b o ambas toman el valor 1
Negación	$\bar{a} = b$	La salida b toma el valor 1 si a toma el valor 0

Importante

2. Otras operaciones lógicas

Operación	Símbolo	Salida
NAND	$S = \overline{A \cdot B}$	La salida toma el valor 1 si A y B no toman simultáneamente el valor 1.
NOR	$S = \overline{A + B}$	La salida toma el valor 1 si A y B toman a la vez el valor 0
XOR	$S = \overline{A} \cdot B + A \cdot \overline{B}$	La salida toma el valor 1 si una entrada es 0 y la otra es 1.
XNOR	$S = A \cdot B + \overline{A} \cdot \overline{B}$	La salida toma el valor 1 si las dos entradas son 0 ó 1 simultáneamente

Importante

3. Álgebra de Boole

Postulados

Respecto a la suma	Respecto al producto

Respecto a la suma	Respecto al producto
$A+0=A$	$A\cdot 0=0$
$A+1=1$	$A\cdot 1=A$
$A+A=A$	$A\cdot A=A$
$A+\bar{A}=1$	$A\cdot\bar{A}=0$
$\bar{\bar{A}}=A$	

Teoremas

Leyes/Teoremas	Respecto a la suma	Respecto al producto
Conmutativa.	$A+B=B+A$	$A\cdot B=B\cdot A$
Asociativa	$A+(B+C)=(A+B)+C$	$A\cdot(B\cdot C)=(A\cdot B)\cdot C$
Distributiva	$A\cdot(B+C)=A\cdot B+A\cdot C$	$A+(B\cdot C)=(A\cdot B)+(A\cdot C)$
Cancelación	$(A\cdot B)+A=A$	$(A+B)\cdot A=A$
De Morgan	$\overline{A+B}=\bar{A}\cdot\bar{B}$	$\overline{A\cdot B}=\bar{A}+\bar{B}$

Importante

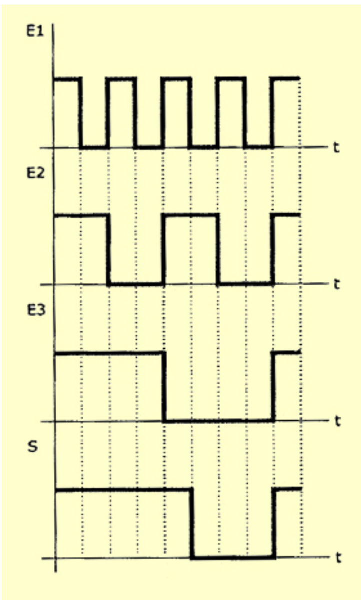
4. Función lógica a partir de la tabla de verdad

Una vez obtenida la tabla de verdad de un sistema, existen dos métodos para obtener la función lógica en la forma canónica que la representa:

- Minterms: implementación por unos.
- Maxterms: implementación por ceros.

5. Cronogramas

Es un método gráfico en el que se representa por niveles altos y bajos (1 y 0) que van tomando las variables. Permite obtener la tabla de verdad y la función lógica.



Importante

6. Implementación de circuitos con puertas NAND y NOR

Se puede implementar un circuito empleando solamente un tipo de puertas abarata costes.

Este método de implementación solo se puede realizar con puertas NAND o NOR, ya que solo estas dos puertas lógicas son universales

Importante

Una vez obtenida la función canónica de una expresión lógica, se debe buscar una expresión simplificada de ésta, con el menor número de términos. Con ello se consigue minimizar el número de errores posibles y abarata su implementación.

Existen dos métodos para conseguir simplificar funciones lógicas.

Simplificación por el método algebraico:

Consiste en utilizar todos los postulados, leyes y teoremas enunciados anteriormente. Con este método se consiguen simplificaciones óptimas, pero hay que tener cierta práctica, no obstante cuando las funciones tienen una expresión grande el procedimiento algebráico puede provocar el que cometamos errores por ser complejo y pesado de utilizar, sobre todo si quien lo realiza no es un experto.

Simplificación por el método de Karnaugh:

Más rápido y eficaz, sobre todo ante problemas complejos. El método se basa en la construcción de unas tablas con la característica de que entre cada celda y su contigua o adyacente solamente cambia el valor de una variable de entrada.

Ejercicios resueltos

Ejercicio resuelto

Obtén la función lógica que se corresponde con la siguiente tabla de verdad:

A	B	C	F	
0	0	0	1	*
0	0	1	1	*
0	1	0	1	*
0	1	1	0	
1	0	0	0	
1	0	1	1	*
1	1	0	0	
1	1	1	1	*

Imagen de elaboración propia

Mostrar retroalimentación

En primer lugar aplicaremos el método de **minterms**, o por el método de los "unos", o por suma de productos.

Para ello observamos todas las combinaciones de las variables de entrada que hacen que la salida sea un 1 (señaladas con un asterisco) y obtendremos como función canónica:

$$F=\overline{A}\cdot\overline{B}\cdot\overline{C}+\overline{A}\cdot\overline{B}\cdot C+\overline{A}\cdot B\cdot C+A\cdot\overline{B}\cdot C+A\cdot B\cdot C$$

Es decir la suma de los productos de las cinco combinaciones de las variables de entrada que dan un 1 en la salida, en donde las variables aparecen negadas si son 0 y afirmadas sin son 1.

Vamos a repetir el ejercicio, pero ahora lo haremos por el método de **maxterms**, o por el método de los "ceros", o por productos de sumas.

Para ello observamos todas las combinaciones de las variables de entrada que hacen que la salida sea un 0 (sin macar con asterico) y obtendremos como función canónica:

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

Es decir, el producto de las sumas de las tres combinaciones de las variables de entrada que dan un 0 en la salida, en donde las variables aparecen negadas si son 1 y afirmadas sin son 0.

En este ejemplo sería más sencillo obtener la función canónica por el segundo método, ya que da como resultado una función canónica más corta.

Las dos funciones son equivalentes, ya que se obtienen de la misma tabla de verdad.

Ejercicio resuelto

Obten la función canónica en el siguiente ejercicio:

Descarga pdf

Ejercicio resuelto

La expresión adjunta corresponde a una determinada función lógica:

$$F = A\cdot\overline{B} + B\cdot C + \overline{A}\cdot C$$

Se desea expresar esta función lógica en forma de productos negados (puertas NAND) o en forma de sumas negadas (Puertas NOR)

Mostrar retroalimentación

La función permanece inmutable si la negamos dos veces:

$$F = A\cdot\overline{B} + B\cdot C + \overline{A}\cdot C = \overline{\overline{A\cdot\overline{B} + B\cdot C + \overline{A}\cdot C}}$$

Aplicando el teorema de Morgan a la negación:

$$F = \overline{\overline{A\cdot\overline{B} + B\cdot C + \overline{A}\cdot C}} = \overline{\overline{A\cdot\overline{B}}\cdot\overline{B\cdot C}\cdot\overline{\overline{A}\cdot C}}$$

Así habremos conseguido expresar la función en forma de productos negados, es decir de modo que se pueda implementar con puertas NAND.

Si lo que quisiéramos es expresar la función en forma de sumas negadas, (puertas NOR), actuaríamos del siguiente modo:

Se niega toda la función dos veces, y se niegan dos veces todos los productos que aparecen en la expresión lógica:

$$F = A\cdot\overline{B} + B\cdot C + \overline{A}\cdot C = \overline{\overline{\overline{\overline{A\cdot\overline{B}}} + \overline{\overline{B\cdot C}} + \overline{\overline{\overline{A}\cdot C}}}}$$

Y aplicando el teorema de DeMorgan a una de las negaciones de los productos:

$$F = \overline{\overline{\overline{\overline{A}\cdot B} + \overline{\overline{B}\cdot\overline{C}} + \overline{A\cdot\overline{C}}}}$$

Y así obtenemos la expresión lógica en función de sumas negadas, que es lo se pretendía.

Ejercicio resuelto

Implementar la función lógica que se adjunta empleando únicamente puertas NAND. Repetir el ejercicio empleando únicamente puertas NOR.

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

Mostrar retroalimentación

En primer lugar obtendremos la solución **utilizando puertas NAND**.

Se aplica una doble negación a toda la expresión y además se niega dos veces la suma:

$$F=A(C+B\cdot D)=\overline{\overline{A\overline{\overline{(C+B\cdot D)}}}}$$

Aplicamos ahora el teorema de Morgan a una de las negaciones de la suma y obtenemos:

$$F=\overline{\overline{A\overline{\overline{(C+B\cdot D)}}}}=\overline{\overline{A\overline{(\overline{C}\cdot\overline{B\cdot D})}}}$$

Que al implementarse quedará:

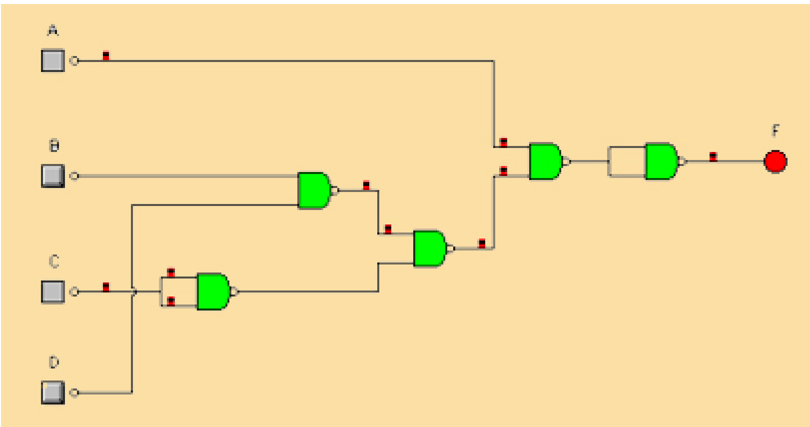


Imagen 25. Recurso propio.

Resolveremos ahora el problema **utilizando puertas NOR**.

Negamos dos veces toda la expresión y el producto interior del paréntesis:

$$F=A(C+B\cdot D)=\overline{\overline{A(C+\overline{\overline{B\cdot D}})}}$$

Aplicando el teorema de DeMorgan por un lado a una de las negaciones que abarca a toda la expresión y por otro al producto interior, tendremos:

$$F=A(C+B\cdot D)=\overline{\overline{A(C+\overline{\overline{B\cdot D}})}}=\overline{\overline{A}+\overline{(C+\overline{\overline{B}+\overline{D}})}}$$

Que al implementarse quedará:

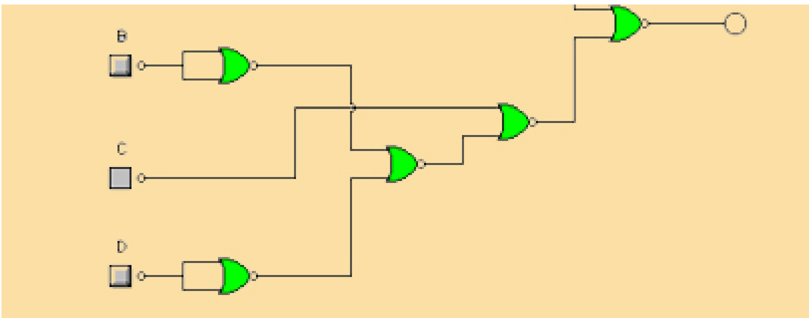


Imagen 26. Recurso propio.

Ejercicio resuelto

Resuelve el siguiente problema implementando puertas NOR

Descarga pdf

Ejercicio resuelto

A partir de la función lógica adjunta:

- Obtener la máxima simplificación posible.
- Implementar la función utilizando cualquier tipo de puertas lógicas de dos entradas.
- Implementar la función empleando únicamente puertas NAND.

$$F = \overline{A}B(\overline{C}+\overline{D})+BC+\overline{A}\cdot\overline{D}$$

Mostrar retroalimentación

Como podemos ver se trata de una función de cuatro variables de entrada, que no está expresada en forma canónica, en primer lugar vamos a obtener su función canónica, en este caso vamos a utilizar el método de la tabla de verdad, para llegar hasta la función canónica:

A	B	C	D	$\overline{A}$	$\overline{C}$	$\overline{D}$	$A\cdot B$	$\overline{C}+\overline{D}$	$A\cdot B\cdot(\overline{C}+\overline{D})$	$C\cdot B$	$\overline{D}\cdot\overline{A}$	$C\cdot B+\overline{D}\cdot\overline{A}$	F
0	0	0	0	1	1	1	0	1	0	0	1	1	1

0	0	0	1	1	1	0	0	1	0	0	0	0	0
0	0	1	0	1	0	1	0	1	0	0	1	1	1
0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	0	1	0	0	1	1	1
0	1	0	1	1	1	0	0	1	0	0	0	0	0
0	1	1	0	1	0	1	1	1	1	1	1	1	1
0	1	1	1	1	0	0	1	0	0	1	0	1	1
1	0	0	0	0	1	1	0	1	0	0	0	0	0
1	0	0	1	0	1	0	0	1	0	0	0	0	0
1	0	1	0	0	0	1	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	1	1	1	1	0	0	0	1
1	1	0	1	0	1	0	1	1	1	0	0	0	1
1	1	1	0	0	0	1	1	1	1	1	0	1	1
1	1	1	1	0	0	0	1	0	0	1	0	1	1

Imagen 32. Recurso propio.

Tomando las salidas que generan 1, tenemos:

$$F=\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}+\overline{A}\cdot\overline{B}\cdot C\cdot\overline{D}+\overline{A}\cdot B\cdot\overline{C}\cdot\overline{D}+\overline{A}\cdot B\cdot C\cdot\overline{D}+\overline{A}\cdot B\cdot C\cdot D+A\cdot B\cdot\overline{C}\cdot\overline{D}+A\cdot B\cdot C\cdot\overline{D}+A\cdot B\cdot C\cdot D$$

Por lo que al introducir los términos de la función canónica en el mapa de Karnaugh, éste quedará:

AB / CD	00	01	11	10
00	**1**	**1**	**1**	
01			**1**	
11		**1**	**1**	
10	**1**	**1**	**1**	

Imagen 33. Recurso propio.

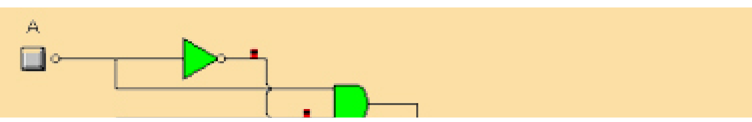
Se pueden hacer tres bolsas de cuatro celdas cada una, con lo que todas las celdas pertenecerán a alguna agrupación, aunque hay alguna celda que pertenece a más de una bolsa.

La función simplificada constará de tantos términos como bolsas hemos tomado para simplificar (3), de dos variables, ya que las bolsas son de $4=2^2$ celdas cada una. (En cada bolsa desaparecen 2 variables).

Se obtiene la función simplificada:

$$F=A\cdot B+\overline{A}\cdot\overline{D}+B\cdot C$$

Que una vez implementada quedaría:



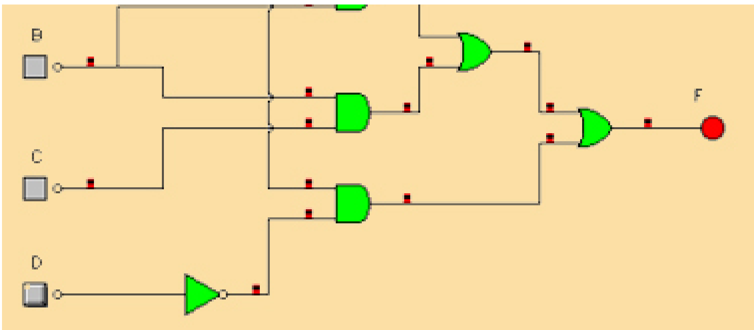


Imagen 34. Recurso propio.

Para implementarlo empleando únicamente puertas NAND de dos entradas, se debe negar dos veces la función y aplicar el teorema de Morgan a una de las negaciones con lo que se obtiene la expresión:

$$F=\overline{A\cdot B+\overline{A}\cdot\overline{D}+B\cdot C}=\overline{A\cdot B}\cdot\overline{\overline{A}\cdot\overline{D}}\cdot\overline{B\cdot C}$$

Que una vez implementada quedará:

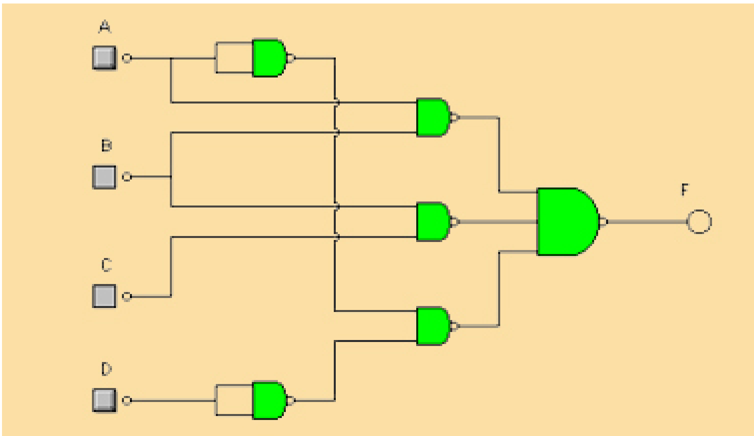


Imagen 35. Recurso propio.

Ejercicio resuelto

Sea un número inferior a diez codificado en binario.
a) Obtén la tabla de verdad de la función F =< 5 (menor o igual de cinco), y escribe su función canónica.
b) Simplifica la función obtenida por el método de Karnaugh.

Mostrar retroalimentación

a) En primer lugar escribimos la tabla de verdad del problema:

Vamos a utilizar una tabla de cuatro bits.

decimal	A	B	C	D	S	decimal	A	B	C	D	S

0	0	0	0	0	**1**	8	1	0	0	0	**0**
1	0	0	0	1	**1**	9	1	0	0	1	**0**
2	0	0	1	0	**1**	10	1	0	1	0	**x**
3	0	0	1	1	**1**	11	1	0	1	1	**x**
4	0	1	0	0	**1**	12	1	1	0	0	**x**
5	0	1	0	1	**1**	13	1	1	0	1	**x**
6	0	1	1	0	**0**	14	1	1	1	0	**x**
7	0	1	1	1	**0**	15	1	1	1	1	**x**

Imagen 35. Recurso propio.

La función tomará el valor 1 en la salida cuando el número introducido es inferior o igual a cinco. Tomará el valor 0 cuando sea mayor que cinco.

A pesar de que existen las combinaciones de entrada correspondientes a los números entre diez y quince, estos valores no se darán nunca, ya que el enunciado del problema limita los datos hasta el decimal diez, por lo tanto a estas combinaciones de las variables de entrada se les asigne el valor x en la salida y se llaman términos indiferentes.

b) Por lo tanto el mapa de Karnaugh será:

AB / CD	**00**	**01**	**11**	**10**
00	1	1	x	
01	1	1	x	
11	1		x	**x**
10	1		x	**x**

Imagen 37. Recurso propio.

En él se han representado las celdas que generan salidas 1 y x. De todas éstas, las que están situadas en la columna de la derecha las trataremos como si fuesen 1, porqué de este modo favorecen la simplificación de la función, mientras que las demás celdas a las que hemos asignado el valor x, nos interesa tratarlas como 0, porque así se favorece la simplificación.

Obteniéndose la función:

$$F = \overline{A}\cdot\overline{C} + \overline{B}\cdot C$$

Imprimible

Mapa imprimible

Aviso Legal

El presente texto (en adelante, el "**Aviso Legal**") regula el acceso y el uso de los contenidos desde los que se enlaza. La utilización de estos contenidos atribuye la condición de usuario del mismo (en adelante, e "**Usuario**") e implica la aceptación plena y sin reservas de todas y cada una de las disposiciones incluidas en este Aviso Legal publicado en el momento de acceso al sitio web. Tal y como se explica más adelante, la autoría de estos materiales corresponde a un trabajo de la **Comunidad Autónoma Andaluza, Consejería de Educación y Deporte (en adelante Consejería de Educación y Deporte**).

Con el fin de mejorar las prestaciones de los contenidos ofrecidos, la Consejería de Educación y Deporte se reserva el derecho, en cualquier momento, de forma unilateral y sin previa notificación al usuario, a modificar, ampliar o suspender temporalmente la presentación, configuración, especificaciones técnicas y servicios de sitio web que da soporte a los contenidos educativos objeto del presente Aviso Legal. En consecuencia, se recomienda al Usuario que lea atentamente el presente Aviso Legal en el momento que acceda al referido sitio web, ya que dicho Aviso puede ser modificado en cualquier momento, de conformidad con lo expuesto anteriormente.

Régimen de Propiedad Intelectual e Industrial sobre los contenidos del sitio web.

Imagen corporativa. Todas las marcas, logotipos o signos distintivos de cualquier clase, relacionados con la imagen corporativa de la Consejería de Educación y Deporte que ofrece el contenido, son propiedad de la misma y se distribuyen de forma particular según las especificaciones propias establecidas por la normativa existente al efecto.

6.1. Método algebraico

El método algebraico consiste en utilizar todos los postulados, leyes y teoremas del Álgebra de Boole que se han visto en el tema anterior.

Con este método se consiguen simplificaciones óptimas.

No obstante cuando las funciones tienen una expresión grande el procedimiento algebráico puede provocar el que cometamos errores por ser complejo y pesado de utilizar, sobre todo si quien lo realiza no es un experto.

A continuación, se puede ver un sencillo ejemplo de su uso.

ej1_simplificacion_metodo_algebraico.pdf 1 / 1

Otro ejemplo de este método es el siguiente:

6.4. Términos indiferentes

En ocasiones, en algunos problemas lógicos, hay combinaciones de las variables de entrada que no se pueden producir, o bien el valor que tome la función para esas combinaciones de las entradas es indiferente para la salida, en ese caso en la tabla de verdad en vez de poner en la salida un 1 o un 0, se pone una x y al simplificar por el método de Karnaugh se considera que el valor es un 1 o un 0 según lo que más favorezca para simplificar. Vamos a verlo con un ejemplo.

Otro ejemplo de simplificación donde aparecen términos indiferentes es el que se plantea a continuación:

ej3_simplificacion.pdf 1 / 2

Mapa Conceptual

Mapa conceptual (pdf - 279067 B).

TI_U4_T3_mapa_conceptual.pdf 1 / 1