ELECTRÓNICA DIGITAL

1. INTRODUCCIÓN

2. SISTEMAS DE NUMERACIÓN Y CÓDIGOS

3. ALGEBRA DE BOOLE

4. FUNCIONES LÓGICAS

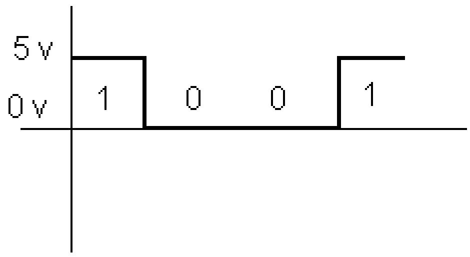
5. SIMPLIFICACIÓN DE FUNCIONES LÓGICAS

1. INTRODUCCIÓN

En un sistema digital, las señales eléctricas que se utilizan tienen dos niveles de tensión, que pueden ser, por ejemplo, 5 voltios o 0 voltios. Los dispositivos electrónicos que se utilizan en estos sistemas, generalmente pueden permanecer en uno de estos dos estados posibles de tensión de manera indefinida, siempre que se mantenga la alimentación. De esto se deduce que un sistema digital puede describirse como un sistema binario y los dos niveles de tensión utilizados se pueden asignar a los valores binarios “0” y “1”.

0	FALSO	APAGADO	SIN TENSIÓN	INTERRUPTOR ABIERTO
1	VERDADERO	ENCENDIDO	CON TENSIÓN	INTERRUPTOR CERRADO

Según se asignen los niveles de tensión, podemos tener lo que se denomina lógica positiva y lógica negativa



LÓGICA POSITIVA

La lógica negativa sería al revés: 0 lógico = 5 v y 1 lógico = 0 v.

2. SISTEMAS DE NUMERACIÓN Y CÓDIGOS

2.1 Sistema binario

Utiliza dos símbolos: el 0 y el 1. Cada símbolo (0 lógico, 1 lógico) es un bit. Sistema binario de 1 bit

Binario	Decimal
0	0
1	1

Sistema binario de 2 bits

Binario	Decimal
0 0	0
0 1	1
1 0	2
1 1	3

Sistema binario de 3 bits

Binario	Decimal
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

Para el caso de 4 bits:

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

Para el 16 necesitaríamos otro bit más: $10000_2 = 16_2$.

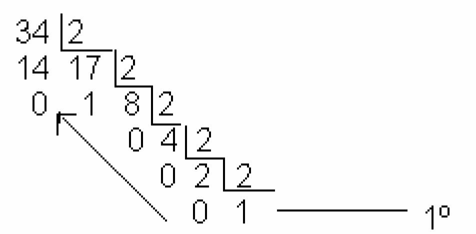
Conversión:

- De binario a decimal

Ejemplo:

$10100_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 20_{10}$

- De decimal a binario



Resultado: $34_2 = 100010_2$.

Sistema hexadecimal

Tiene 16 símbolos: del 0 al 9 y de la A a la F

Binario	Decimal	Hexadecimal
0 0 0 0	0	0
0 0 0 1	1	1
0 0 1 0	2	2
0 0 1 1	3	3
0 1 0 0	4	4
0 1 0 1	5	5
0 1 1 0	6	6
0 1 1 1	7	7
1 0 0 0	8	8
1 0 0 1	9	9
1 0 1 0	10	A
1 0 1 1	11	B
1 1 0 0	12	C
1 1 0 1	13	D
1 1 1 0	14	E
1 1 1 1	15	F

- De binario a hexadecimal:

Se realiza agrupando de cuatro en cuatro bits y si falta se completa con ceros, por último se utiliza la tabla.

Ejemplo:

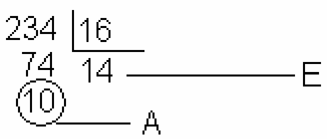
$1011111_2 = 5F_{16}$.

Agrupo y completo: $0101_2 \quad 1111_2 = 5_{16} \quad F_{16}$

- De hexadecimal a decimal

$AF3_{16} = A \times 16^2 + F \times 16^1 + 3 \times 16^0 = 10_{10} \times 256_{10} + 15_{10} \times 16_{10} + 3_{10} \times 1_{10} = 2803_{10}$.

- De decimal a hexadecimal



Resultado: $234_{10} = EA_{16}$.

Códigos

Son representaciones mediante combinaciones de símbolos

- Códigos continuos. Si las combinaciones correspondientes a números consecutivos son ***adyacentes*** (aquellas en las que sólo varía un bit)
- Códigos cíclicos. La última combinación es adyacente de la primera.

BCD exceso 3

0	0 0 1 1
1	0 1 0 0
2	0 1 0 1
3	0 1 1 0
4	0 1 1 1
5	1 0 0 0
6	1 0 0 1
7	1 0 1 0
8	1 0 1 1
9	1 1 0 0

BCD 5421

0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	1 0 0 0
6	1 0 0 1
7	1 0 1 0
8	1 0 1 1
9	1 1 0 0

Gray (reflejado)

De 1 bit	De 2 bits (se refleja y se añade 0 por encima y 1 por debajo)
0 1	0 0 0 1 ------ 1 1 1 0

<u>De 3 bits</u>

0 0 0
0 0 1
0 1 1
0 1 0
----
1 1 0
1 1 1
1 0 1
1 0 0

<u>De 4 bits</u>

0 0 0 0
0 0 0 1
0 0 1 1
0 0 1 0
0 1 1 0
0 1 1 1
0 1 0 1
0 1 0 0
----

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

BCD Natural

Peso: 8 4 2 1

0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

BCD Aiken

Peso: 2 4 2 1

0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	1 0 1 1
6	1 1 0 0
7	1 1 0 1
8	1 1 1 0
9	1 1 1 1

UNIDADES BINARIAS

1 byte = 8 bits.

1 Kilobyte (Kb) = 1024 bytes

1Megabytes (Mb) = 1024 Kilobytes

1 Gigabyte (Gb) = 1024 Mb

Ejemplos:

Texto simple aproximadamente 3000 Kb

Sonido de baja calidad (duración 1 minuto) = 1.14 Mb

3. ALGEBRA DE BOOLE.

Hay un tipo de álgebra, desarrollada en el siglo XIX por el reverendo George Boole, un religioso inglés, que es muy adecuada para representar la situación anteriormente descrita.

Esta rama de las Matemáticas se denomina “Algebra de Boole”. Es un álgebra en la que las variables sólo pueden tomar dos valores, 0 ó 1, existiendo una serie de teoremas y reglas asociadas con el álgebra que permiten la manipulación de ecuaciones booleanas.

Un álgebra de Boole es un conjunto de elementos denominados variables booleanas, las cuales sólo pueden adoptar dos valores o estados perfectamente diferenciados. Estos dos estados, que pueden notarse simbólicamente por 0 y 1, están relacionados por dos operaciones binarias denominadas Suma Lógica (+) y Producto Lógico (·), de modo que se cumplen los siguientes postulados:

1. Ambas operaciones son conmutativas:

$a + b = b + a$

$a \cdot b = b \cdot a$

2. Existen dos elementos pertenecientes al álgebra, denominados elementos neutros para cada operación, tales que:

$0 + a = a$

$1 \cdot b = b$

3. Cada operación es distributiva respecto de la otra:

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

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

4. Existencia del elemento neutro

$a + \bar{a} = 1$

$a \cdot \bar{a} = 0$

Este postulado lleva implícita la existencia de una nueva operación llamada Inversión o Complementación

<u>Nota</u>: El elemento complementario o invertido es el estado contrario del dado.

Teoremas fundamentales de un Álgebra de Boole.

1. Teorema de idempotencia

$a \cdot a = a$

$a + a = a$

2. Teorema de las constantes

$a + 1 = 1$

$a \cdot 0 = 0$

3. Teorema de redundancia o de absorción.

$a + a \cdot b = a$

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

4. Teorema de asociatividad

$a + (b + c) = (a + b) + c$

$a \cdot (b \cdot c) = (a \cdot b) \cdot c$

5. Teorema del doble complemento

$\overline{\overline{a}} = a$

6. Teorema de la falsa redundancia, de condiciones aleatorias o del consenso.

$a + \bar{a} \cdot b = a + b$

$a \cdot (\bar{a} + b) = a \cdot b$

7. Teorema de De Morgan

- Para 2 variables:

$$\overline{a + b} = \overline{a} \cdot \overline{b}$$

$$\overline{a \cdot b} = \overline{a} + \overline{b}$$

- Para n variables:

$$\overline{a + b + c + \dots\dots\dots + x} = \overline{a} \cdot \overline{b} \cdot \overline{c} \cdot \dots\dots\dots \cdot \overline{x}$$

$$\overline{a \cdot b \cdot c \cdot \dots\dots\dots \cdot x} = \overline{a} + \overline{b} + \overline{c} + \dots\dots\dots + \overline{x}$$

Funciones de un Álgebra de Boole

<u>1. Función de un Álgebra de Boole.</u>

Es una variable binaria cuyo valor es igual al resultado de una expresión algebraica en la que las variables binarias que intervienen se relacionan entre sí por medio de las operaciones básicas: Suma Lógica, Producto Lógico e Inversión o Complementación.

2. Formas de representar funciones booleanas:

Existen dos formas:

a) *Tabla de Verdad*. Es una representación tabulada en la que se indican todas las combinaciones posibles que pueden adoptar las variables, así como el valor lógico que toma la función para cada posibilidad

Ejemplo: f = f (a, b, c)

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

2^3 = 8 posibles combinaciones

b) *Representación algebraica*. La función expresa como una ecuación algebraica en la que intervienen las variables, bien en su forma directa o en su forma invertida, y relacionadas por las operaciones lógicas: Suma, Producto e Inversión o Complementación.

Ejemplo: $f(a, b, c) = a \cdot \overline{b} \cdot c + a + \overline{c} \cdot b + \overline{a} \cdot c$

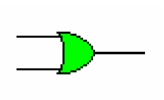
Si a = 0; b = 1; c = 0

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

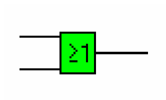
4. FUNCIONES LÓGICAS

a) Suma Lógica (puerta OR). La puerta SUMA LÓGICA o puerta OR es aquella en la que la salida está a 0, sólo cuando todas las entradas están a cero.

Simbología



Símbolo MIL



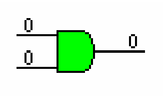
Símbolo CEI

Tabla de Verdad

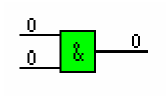
Entrada (a)	Entrada (b)	Salida $f = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

b) Producto Lógico (puerta AND). La puerta PRODUCTO LÓGICO o puerta AND es aquella en la que la salida está a 1, sólo cuando todas las entradas están a 1.

Simbología



Símbolo MIL



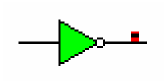
Símbolo CEI

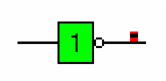
Tabla de Verdad

Entrada (a)	Entrada (b)	Salida $f = a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

c) Inversión o Complementación (puerta NOT). La puerta inversora o puerta NOT es aquella invierte la entrada, es decir, si introducimos un 1 lógico (5 v) obtenemos a la salida un 0 lógico (0 v) y viceversa.

Simbología





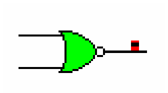
Símbolo MIL Símbolo CEI

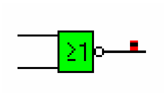
Tabla de Verdad

Entrada (a)	F = Salida (ā)
0	1
1	0

d) Suma Lógica Invertida (NOR). La puerta SUMA LÓGICA INVERTIDA o puerta NOR es una puerta OR a la que se le ha colocado a la salida un inversor, por tanto, la salida está a 1 sólo cuando todas las entradas están a 0. Suma las entradas e invierte el resultado

Simbología





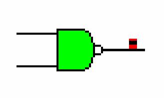
Símbolo MIL Símbolo CEI

Tabla de Verdad

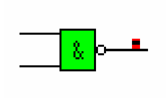
Entrada (a)	Entrada (b)	Salida $f = \overline{a + b}$
0	0	1
0	1	0
1	0	0
1	1	0

e) Producto Lógico Invertido (NAND). La puerta PRODUCTO LÓGICO INVERTIDO o puerta NAND es una puerta AND a la que se le ha colocado a la salida un inversor, por tanto, la salida está a 0 sólo cuando todas las entradas están a 1. Multiplica las entradas e invierte el resultado.

Simbología



Símbolo MIL



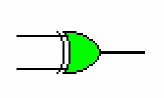
Símbolo CEI

Tabla de Verdad

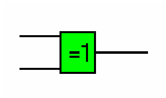
Entrada (a)	Entrada (b)	Salida $f = \overline{a \cdot b}$
0	0	1
0	1	1
1	0	1
1	1	0

f) Suma lógica Exclusiva (puerta X-OR). La puerta SUMA LÓGICA EXCLUSIVA o puerta EXOR es una puerta en la que la salida está a 0 sólo cuando todas las entradas están a igual nivel lógico. La salida está a 1 siempre que una sola de las entradas está a 1.

Simbología



Símbolo MIL



Símbolo CEI

Tabla de Verdad

Entrada (a)	Entrada (b)	Salida $f = a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

4.1. MINTÉRMINOS Y MAXTÉRMINOS

- **Término canónico:** Se entiende por término canónico de una función lógica a todo producto o suma lógica en la que intervienen todas las variables de la función bien en su forma directa o invertida.

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

Término canónico

No son térmimos canónicos

- **Mintérmino (m):** Cuando el término canónico es un producto.

Ejemplo: $f(d, c, b, a) = d \cdot \bar{c} \cdot b \cdot a + d \cdot c \cdot \bar{b} \cdot a + d \cdot c \cdot b \cdot \bar{a}$

- **Maxtérmino (M):** Cuando el término canónico es una suma.

Ejemplo: $f(d,c,b,a) = (d + \bar{c} + b + a) \cdot (d + c + \bar{b} + a) \cdot (d + c + b + \bar{a})$

- **Función expresada en forma canónica:** Cuando una función representada algebraicamente tiene todos sus términos canónicos y del mismo tipo (mintérminos o maxtérminos), decimos que la función está expresada en forma canónica

Está expresda en forma canónica —— $f(d, c, b, a) = d \cdot \bar{c} \cdot b \cdot a + d \cdot c \cdot \bar{b} \cdot a$

No está expresada en forma canónica —— $f(a, b, c) = a \cdot \bar{b} \cdot c + a + \bar{c} \cdot b + \bar{a} \cdot c$

Si una función tiene "n" variables, el máximo número de términos canónicos que puede tener es "2^n", siendo "n" el número de variables de la función.

Existe un criterio generalizado de representación de las funciones canónicas:

- Si una variable está en su forma directa se pone un 1
- Si una variable está en su forma directa se pone un 0

$d \cdot \bar{c} \cdot b \cdot a$

1 0 1 1

m_i	a b c	f
0	0 0 0	0
1	0 0 1	1
2	0 1 0	0
3	0 1 1	0
4	1 0 0	1
5	1 0 1	1
6	1 1 0	0
7	1 1 1	0

$$f(a,b,c) = \overline{a}\cdot\overline{b}\cdot c + a\cdot\overline{b}\cdot\overline{c} + a\cdot\overline{b}\cdot c = \Sigma_3 (1, 4, 5) = \Sigma_n m_i$$

Una función en forma canónica, si se expresa en forma de MINTÉRMINOS, la función toma valor lógico 1 para la combinación de variables que indica cada término canónico. Los mintérminos para los que la función toma valor lógico 0 no van a intervenir en la forma canónica de la función.

Ejemplo:

$f(a, b, c) = \Sigma_3 (0, 2, 5)$

m_i	a b c	f
0	0 0 0	1
1	0 0 1	0
2	0 1 0	1
3	0 1 1	0
4	1 0 0	0
5	1 0 1	1
6	1 1 0	0
7	1 1 1	0

5. SIMPLIFICACIÓN DE FUNCIONES LÓGICAS

Introducción

Uno de los objetivos del diseñador digital, cuando utiliza puertas discretas, es implementar una función booleana con el mínimo posible de puertas lógicas. Cuantas menos puertas se utilicen, menor será el coste del circuito.

Se puede realizar la simplificación mediante un proceso puramente algebraico, pero esto puede resultar tedioso, y el diseñador nunca estará seguro de haber conseguido la solución más simple al final del proceso.

Un método de simplificación mucho más fácil es dibujar la función sobre una tabla de Karnaugh y con ayuda de una serie de reglas sencillas, reducir la función booleana a su forma mínima. Este método es muy directo hasta con seis variables: por encima de seis variables es mejor utilizar el método de Quine-McCluskey.

Método algebraico de simplificación

Se realiza la minimización aplicando los siguientes pasos:

a) T. de idempotencia

b) Prop. Distributiva.

c) Elemento complementario.

d) Elemento neutro del producto lógico.

Ejemplo:

$f(a, b, c) = abc + abc + abc =$

T. de idempotencia

$= a\cdot b\cdot c + a\cdot b\cdot c + a\cdot b\cdot c + a\cdot b\cdot c =$

P. distributiva

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

Elemento complementario

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

Elemento neutro del producto lógico

$= b\cdot c + a\cdot c$

Con este método no se garantiza la expresión más simple.

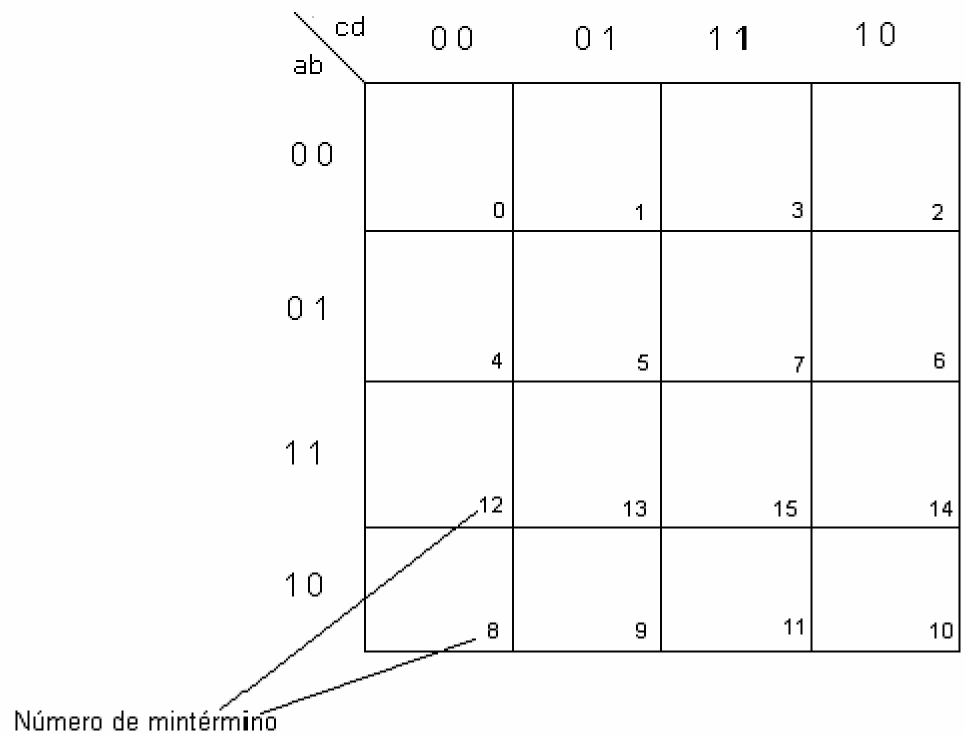
Método gráfico de Karnaugh

Es un método gráfico que se utiliza para minimizar o simplificar funciones lógicas.

Consiste en una especie de “mapa” en el que los cuadrados corresponden a todos los términos canónicos de la función. Habrá tantos cuadrados como términos canónicos de la función.

Ejemplo:

Una función de 4 variables (a, b, c, d) tendrá $2^4 = 16$ cuadrados con el siguiente mapa:



La tabla o mapa siempre se construye así, de manera que cada elemento del mapa es adyacente con sus contiguos (en horizontal y vertical).

Se va rellenando el mapa de la siguiente forma:

Se pone un 1 en las casillas que corresponden a los términos canónicos para los que la función vale 1, es decir, se pone un 1 en el mintérmino correspondiente

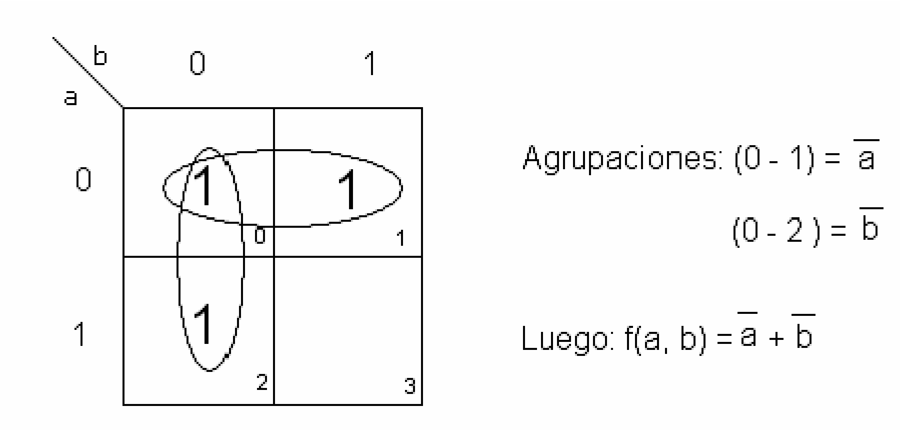
A continuación se realiza el siguiente proceso sistemático:

1. Se toman todos los unos no contiguos a ningún otro.
2. Se agrupan los grupos de 2 unos contiguos, siempre que no puedan formar parte de un grupo de 4 unos.
3. Se toman los grupos de 4 unos contiguos, siempre que no puedan formar parte de un grupo de 8 unos.
4. Etc.

Cuando se hayan hecho todas las agrupaciones posibles, se habrá acabado; obteniéndose de cada agrupación el conjunto de variables que permanecen sólo en una forma (que no cambian), bien directa o invertida.

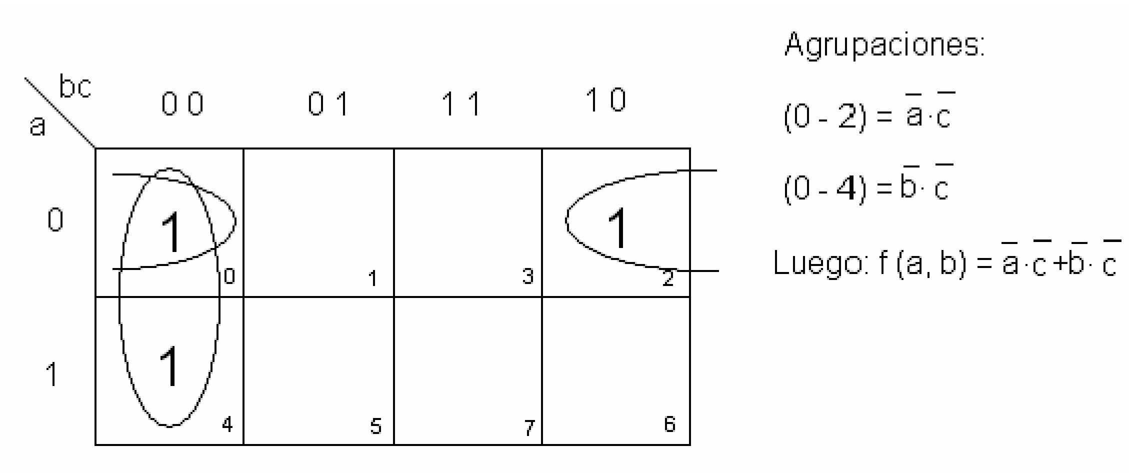
Ejemplo de la simplificación de una función de 2 variables:

$$f(a,b) = \overline{a}\cdot\overline{b} + \overline{a}\cdot b + a\cdot\overline{b} = \sum_2 (0,1,2)$$



Ejemplo de tres variables

$$f(a,b,c) = \overline{a}\cdot\overline{b}\cdot\overline{c} + \overline{a}\cdot b\cdot\overline{c} + a\cdot\overline{b}\cdot\overline{c} = \sum_3 (0,2,4)$$



Ejercicios de 2 y 3 variables:

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

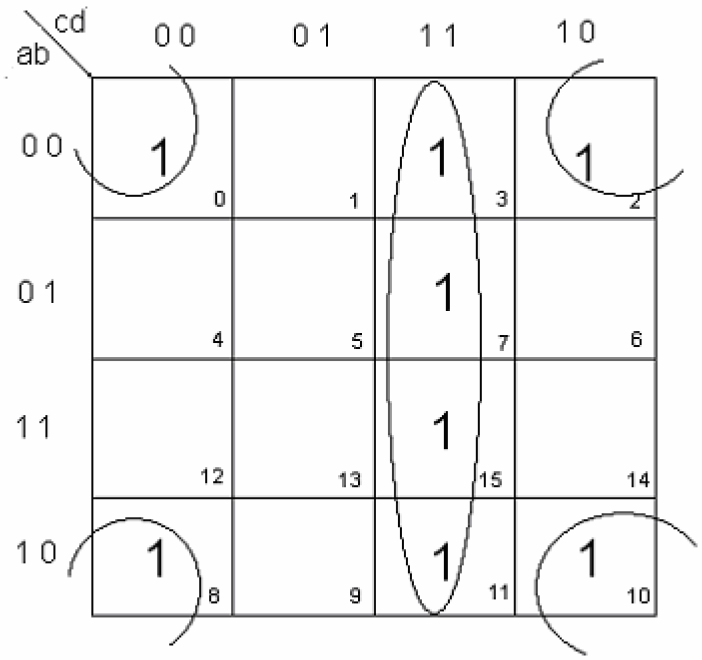
$f(a,b,c) = \overline{a} \cdot \overline{b} \cdot \overline{c} + \overline{a} \cdot b \cdot \overline{c} + \overline{a} \cdot \overline{b} \cdot c + a \cdot b \cdot c$

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

<u>Ejemplo de 4 variables</u>

$$f(a,b,c,d) = \overline{a}\cdot\overline{b}\cdot\overline{c}\cdot\overline{d} + \overline{a}\cdot\overline{b}\cdot c\cdot\overline{d} + \overline{a}\cdot\overline{b}\cdot c\cdot d + \overline{a}\cdot b\cdot c\cdot d + a\cdot\overline{b}\cdot\overline{c}\cdot\overline{d} + a\cdot\overline{b}\cdot c\cdot\overline{d} + a\cdot\overline{b}\cdot c\cdot d + a\cdot b\cdot c\cdot d =$$
$$= \Sigma_4(0,\ 2,\ 3,\ 7,\ 8,\ 10,\ 11,\ 15)$$

$$f(a,b,c,d) = \overline{a} \cdot \overline{b} \cdot \overline{c} \cdot \overline{d} + \overline{a} \cdot \overline{b} \cdot c \cdot \overline{d} + \overline{a} \cdot \overline{b} \cdot c \cdot d + \overline{a} \cdot b \cdot c \cdot d + a \cdot \overline{b} \cdot \overline{c} \cdot \overline{d} + a \cdot \overline{b} \cdot \overline{c} \cdot d +$$
$$+ a \cdot \overline{b} \cdot c \cdot d + a \cdot b \cdot c \cdot d = \sum (0,2,3,7,8,10,11,15)$$



Agrupaciones:

$(0 - 2 - 8 - 10) = \overline{b}\cdot\overline{d}$

$(3 - 7 - 11 - 15) = cd$

$f(a,b,c,d) = \overline{b}\cdot\overline{d} + cd$

Ejercicios de 4 variables:

$f(a,b,c,d) = \sum_4 (3,5,8,9,10,12,13,14)$

$f(a,b,c,d) = \sum_4 (0,2,4,6,8)$

$f(a,b,c,d) = \sum_4 (0,1,2,6,8,9,10)$

EJERCICIOS

1. Dada la siguiente función de tres variables en forma de mintérminos:

   $f(a,b,c)=\sum_3(0,2,3,7)$

   Se pide:

   a. Simplificarla por Karnaugh
   b. Implementación con puertas NAND de dos entradas e inversoras.
   c. Implemetación con puertas NOR de dos entradas e inversoras.

2. Un jurado está formado por tres jueces A, B, y C. Cada juez emite su voto a favor oprimiendo un botón enfrente de él. Expresa en términos minterm la función que se ponga a 1 cuando se produzca una mayoría a favor y a 0 en cualquier otro caso.

3. Un zumbador debe accionarse para dar una señal de alarma “f” cuando cuatro relés (d, c, b, a) cumplan las siguientes condiciones:

   - “a” y “b” excitados, “c” y “d” en reposo.
   - “a” excitado, “b”, “c” y “d” en reposo.
   - “c” excitado, “a”, “b” y “d” en reposo.
   - “a” y “c” excitados, “b” y “d” en reposo.

   Se pide:

   a. Tabla de verdad y función simplificada.
   b. Esquema con puertas lógicas.

4. Dada la función $f(a,b,c,d)=\sum_4(2,3,5,7,10,11,15)$ se pide:

   a. Simplificarla por Karnaugh
   b. Implementar la función mediante puertas NAND de dos y tres entradas

5. Para poner en marcha un motor se requieren tres interruptores “c,b,a” de tal forma que el funcionamiento del mismo se produzca únicamente en las siguientes condiciones:

   - Cuando esté cerrado solamente “c”
   - Cuando estén cerrados simultáneamente “a” y “c” y no lo esté “b”.
   - Cuando estén cerrados simultáneamente “a” y “b” y no lo esté “c”.

   Se pide:

   a. Construir la tabla de verdad
   b. Implementar el circuito con puertas NAND de dos entradas
   c. Implementar el circuito con puertas NOR de dos entradas

6. Un circuito lógico recibe como entradas un número decimal (de 0 a 9) codificado en binario (4 entradas de un bit). La salida será 1 siempre que el número decimal sea menor o igual a 5. Se pide:

   a) Función lógica y tabla de verdad.

   b) Simplificación por Karnaugh y circuito con puertas lógicas de la función simplificada.

7. Un pequeño taller dispone de tres máquinas, ***M1, M2** y **M3***, que en marcha consumen, respectivamente, 3, 6 y 9 kW. Para indicar un consumo excesivo, una señal de alerta ***S*** actúa cuando se superan los 10 kW. Se pide:

a) Obtener la tabla de verdad y la función lógica simplificada.

b) Dibujar el circuito lógico correspondiente a la función lógica simplificada.