Titulación: Grado en Ingeniería Informática
Asignatura: Fundamentos de Computadores

Bloque 2: Sistemas combinacionales
Tema 4: Algebra de Boole y funciones lógicas

Pablo Huerta Pellitero

ÍNDICE

BIBLIOGRAFÍA

- Román Hermida, Ana Mº del Corral, Enric Pastor, Fermín Sánchez
  "Fundamentos de Computadores" , cap 2
  Editorial Síntesis
- Thomas L. Floyd
  "Fundamentos de Sistemas Digitales", cap 4
  Editorial Prentice Hall
- Daniel D. Gajski
  "Principios de Diseño Digital", cap 3,4
  Editorial Prentice Hall
- M. Morris Mano
  "Diseño Digital", cap 2
  Editorial Prentice Hall

INTRODUCCIÓN

- El Álgebra de Boole fue creada por el matemático británico George Boole ("An Investigation of the Laws of Thought", 1854).
- Se trata de un formalismo matemático sencillo para representar el conocimiento y realizar cálculos.
- Inicialmente se planteó como un formalismo más para realizar cálculos en Lógica Proposicional.
- En esta asignatura vamos a utilizar el Algebra de Boole como herramienta matemática para diseñar circuitos electrónicos digitales.

DEFINICIÓN FORMAL

- Un Algebra de Boole bivaluada es un conjunto B en el cual:
  - $\forall$ a $\in$ B , a = 0 ó a = 1.
  - 1 > 0
  - Todo elemento tiene un complementario. (Funcion NOT, ’)
  - Define la operación producto lógico (AND, · , *):

    La función AND de dos o más variables vale ‘1’ cuando todas las variables valen ‘1’, y vale ‘0’ en caso contrario.
  - Define la operación suma lógica (OR, +):

    La función OR de dos o más variables vale ‘1’ cuando alguna de ellas vale ‘1’, y vale ‘0’ en caso contrario.

a	not(a)
0	1
1	0

a	b	a·b
0	0	0
0	1	0
1	0	0
1	1	1

a	b	a+b
0	0	0
0	1	1
1	0	1
1	1	1

TEOREMAS Y PROPIEDADES DEL ALGEBRA DE BOOLE

- Principio de dualidad: para todo teorema o ley del Algebra de Boole, existe otro **teorema dual** también válido que se obtiene:
  - Sustituyendo los operadores AND por operadores OR.
  - Sustituyendo los operadores OR por operadares AND.
  - Sustituyendo los ‘1’ por ‘0’, y los ‘0’ por ‘1’.

- Teorema de expansión, o de descomposición de funciones:

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

$$f(a,b,c,...) = [a + f(0,b,c,...)] \cdot [\overline{a} + f(1,b,c,...)]$$

TEOREMAS Y PROPIEDADES DEL ALGEBRA DE BOOLE

Propiedad asociativa	a+(b+c) = (a+b)+c = a+b+c	a·(b·c) = (a·b)·c = a·b·c
Propiedad conmutativa	a+b = b+a	a·b = b·a
Propiedad distributiva	a+(b·c) = (a+b) · (a+c)	a·(b+c) = a·b + a·c
Elemento neutro	0+a=a	1·a=a
	1+a=1	0·a=0
Teoremas de identidad	a+a’=1	a·a’=0
Teoremas de idempotencia	a+a=a	a·a=a
Teorema de involución	(a’)’=a	
Teoremas de absorción	a+a·b = a	a·(a+b) = a
	a+a’·b=a+b	a·(a’+b)=a·b
	a·b+a·b’ = a	(a’+b’) ·(a’+b) = a’
Teoremas del consenso	a·b+a’·c+b·c = a·b+a’·c	(a+b)·(a’+c)·(b+c) = (a+b)·(a’+c)
Teoremas de De Morgan	(a+b)’ = a’·b’	(a·b)’ = a’+b’

FUNCIONES DE CONMUTACIÓN

- Una función de conmutación de ‘n’ variables, también llamada función lógica o función booleana es una aplicación en la que:
  - El conjunto de origen es el producto cartesiano $\{0,1\}^n$
  - El conjunto destino es $\{0,1\}$

$$f : \{0,1\}^n \rightarrow \{0,1\}$$

  - Ejemplo: $f(a,b) = \{(00,1),(01,1),(10,1),(11,0)\}$

    o asumiendo el orden natural binario:

$$f(a,b) = \{1,1,1,0\}$$

- El conjunto de origen contiene 2^n elementos, y el conjunto destino tiene 2 elementos.
- Existen por tanto 2^{2^n} funciones de conmutación diferentes de ‘n’ variables.

FUNCIONES DE CONMUTACIÓN

- Para una única variable existen 2^2 funciones de conmutación posibles:

x_0	f_0	f_1	f_2	f_3
0	0	1	0	1
1	0	0	1	1

Nula Inversa Igualdad Unidad

- Para dos variables existen 2^4 funciones de conmutación distintas

$x_1\,x_0$	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Nula And … Xor Or Nor Xnor … Nand Unidad

FUNCIONES DE CONMUTACIÓN

- Las funciones de conmutación se pueden representar de diversas maneras:
  - Tablas de verdad:

a	b	f(a,b)
0	0	0
0	1	1
1	0	1
1	1	0

  - Expresiones de conmutación: son combinaciones de variables, constantes y paréntesis relacionados mediante los operadores booleanos NOT, AND y OR (en ese orden de precedencia).

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

  - Mapas de Karnaugh: es otra representación gráfica, en forma de cuadrícula, en la que cada celda contiene el valor de la función para una combinación determinada de las variables de entrada.

a \ b	0	1
0	0	1
1	1	0

FUNCIONES DE CONMUTACIÓN

- **Expresiones de conmutación equivalentes**: dos EC son equivalentes si representan a una misma función de conmutación.
- Se puede comprobar si dos EC son equivalentes evaluándolas para todos los posibles valores de las variables (obteniendo la tabla de verdad). Si ambas EC llevan a la misma tabla de verdad => son equivalentes.
- También se puede intentar manipular una de las expresiones utilizando las propiedades del Algebre de Boole para llegar a la otra expresión.
- Ejemplo: ¿son equivalentes EC_1 y EC_2?

$$EC_1(x_1,x_0)=x_1 \cdot x_0 + \bar{x}_1$$
$$EC_2(x_1,x_0)=x_0 + \bar{x}_1$$

x_1 x_0	f_1	f_0
0 0	1	1
0 1	1	1
1 0	0	0
1 1	1	1

$\Rightarrow$ Equivalentes

FUNCIONES DE CONMUTACIÓN

$$EC_1(x_2,x_1,x_0) = x_2 \cdot x_1 + x_2 \cdot \bar{x}_1 + x_2 \cdot x_0$$
$$EC_1(x_2,x_1,x_0) = x_2$$

- Ejemplo: ¿son equivalentes EC_1 y EC_2?

$$EC_1(x_2,x_1,x_0) = x_2 \cdot x_1 + x_2 \cdot \bar{x}_1 + x_2 \cdot x_0$$

Por la propiedad distributiva $= x_2 \cdot (x_1 + \bar{x}_1) + x_2 \cdot x_0$

Por la propiedad de complemento $= x_2 \cdot 1 + x_2 \cdot x_0$

Por la propiedad distributiva $= x_2 \cdot (1 + x_0)$

Por la propiedad de elemento neutro $= x_2 \cdot 1 = x_2$

- Se ha visto que una EC representa a una FC, y que ésta se calcula mediante la evaluación de la EC creando la tabla de verdad.

- También se puede recorrer el camino inverso, y dada una FC se ha de poder calcular la EC estándar (forma canónica) que la representa.

FORMAS CANÓNICAS

Definiciones previas:

- **Literal**: es una variable suelta, afirmada o negada.
- **Término producto**: es una expresión booleana compuesta por un único literal o por un producto lógico (and) de literales.
    - **Minitérmino (minterm)**: es un término producto que contiene todas las variables de la función, algunas de ellas pueden estar afirmadas y otras negadas.
- **Término suma**: es una expresión booleana compuesta por un único literal o por una suma lógica (or) de literales.
    - **Maxitérmino (maxterm)**: es un término suma que contiene todas las variables de la función, algunas de ellas pueden estar afirmadas y otras negadas
- **Suma de productos (SOP, SdP)**: es una expresión booleana compuesta por un único término producto o por una suma de términos producto.
- **Producto de sumas (POS, PdS)**: es una expresión booleana compuesta por un único término suma o por un producto de términos suma.

FORMAS CANÓNICAS

- Propiedades de los minterms:
  - Un minterm se puede representar como m_i, siendo ‘i’ el número entero que se obtiene de evaluar a positivo el minterm => sustituir las variables afirmadas por ‘1’ y las negadas por ‘0’.

    Ejemplo: suponiendo 3 variables, x_2, x_1 y x_0:

    $$x_2 \cdot \overline{x_1} \cdot x_0 \Rightarrow 101 \Rightarrow 5 \Rightarrow m_5$$

  - Un minterm vale ‘1’ para una única combinación de las variables, que coincide con el índice ‘i’ del minterm. Para el resto de valores de las variables el minterm vale ‘0’.

    Ejemplo: suponiendo 3 variables, x_2, x_1, x_0, el minterm $m_3 = \overline{x_2} \cdot x_1 \cdot x_0$ vale ‘1’ cuando $x_2x_1x_0$ valen ‘3’, es decir cuando x_2 vale ‘0’, x_1 vale ‘1’ y x_0 vale ‘1’.

FORMAS CANÓNICAS

- **Primera forma canónica: suma de productos canónica.**
- La expresión en forma de suma de productos (SdP) canónica de una función de conmutación es una expresión de conmutación en la que aparecen sumados varios minterms.
- Los minterms que se incluyen en la expresión son aquellos que se corresponden con las filas que valen '1' en la tabla de verdad.
- Ejemplo: dada una función f(a, b, c) que tiene la siguiente tabla de verdad:

a	b	c	f(a,b,c)	
0	0	0	0	
0	0	1	1	$\longrightarrow m_1$
0	1	0	1	$\longrightarrow m_2$
0	1	1	1	$\longrightarrow m_3$
1	0	0	1	$\longrightarrow m_4$
1	0	1	0	
1	1	0	1	$\longrightarrow m_6$
1	1	1	0	

Su expresión en forma de SdP canónica es:

$$\sum m(1,2,3,4,6) = m_1 + m_2 + m_3 + m_4 + m_6 =$$

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

FORMAS CANÓNICAS

- Propiedades de los maxterms:
  - Un maxterm se puede representar como M_i, siendo 'i' el número entero que se obtiene de evaluar a negativo el maxterm => sustituir las variables afirmadas por '0' y las variables negadas por '1'.

    Ejemplo: suponiendo tres variables $x_2x_1x_0$:

    $$x_2 + \overline{x_1} + x_0 \Rightarrow 010 \Rightarrow 2 \Rightarrow M_2$$

  - Un maxterm vale '0' para una única combinación de las variables, que coincide con el índice 'i' del maxterm. Para el resto de valores de las variables, el maxterm vale '1'.

    Ejemplo: suponiendo 3 variables, x_2, x_1, x_0, el maxterm $M_3 = x_2 + \overline{x_1} + \overline{x_0}$ vale '0' cuando $x_2x_1x_0$ valen '3', es decir cuando x_2 vale '0', x_1 vale '1' y x_0 vale '1'.

FORMAS CANÓNICAS

- **Segunda forma canónica: producto de sumas canónico.**
- La expresión en forma de producto de sumas (PdS) canónico de una función de conmutación es una expresión de conmutación en la que aparecen multiplicados varios maxterms.
- Los maxterms que se incluyen en la expresión son aquellos que se corresponden con las filas que valen '0' en la tabla de verdad.
- Ejemplo: dada una función $f(X_2,X_1,X_0)$ que tiene la siguiente tabla de verdad:

X_2	X_1	X_0	f(X)	
0	0	0	0	$\rightarrow M_0$
0	0	1	1	
0	1	0	1	
0	1	1	1	
1	0	0	1	
1	0	1	0	$\rightarrow M_5$
1	1	0	1	
1	1	1	0	$\rightarrow M_7$

Su expresión en forma de PdS canónico es:

$$\prod M(0,5,7) = M_0 \cdot M_5 \cdot M_7 =$$

$$= (x_2 + x_1 + x_0) \cdot (\overline{x_2} + x_1 + \overline{x_0}) \cdot (\overline{x_2} + \overline{x_1} + \overline{x_0})$$

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

- Las EC se utilizarán más adelante a la hora de especificar y diseñar circuitos electrónicos digitales. Las EC están directamente relacionadas con el tamaño y complejidad del circuito físico y cuanto más simple sea una EC más simple y barato será el circuito electrónico resultante.
- Ante la necesidad de que las EC sean lo más sencillas posibles surgen distintos métodos de simplificación como los mapas de Karnaugh, las tablas de Quine-McCluskey, etc.
  - En nuestro caso sólo atenderemos a las simplificaciones gráficas mediante mapas o diagramas de Karnaugh.
- Un mapa de Karnaugh es una forma gráfica alternativa para representar una función de conmutación (tabla de verdad) en forma de un conjunto bidimensional de celdas y que permite realizar simplificaciones fácilmente.
- Para una función de ‘n’ variables, existen celdas (una celda para cada minterm) identificadas por el número decimal ‘i’ representante del minterm. 2^n

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

mapa de Karnaugh de 1 variable

X	
0	0
1	1

mapa de Karnaugh de 2 variables

$X_1 \backslash X_0$	0	1
0	0	1
1	2	3

mapa de Karnaugh de 3 variables

$X_2 \backslash X_1X_0$	00	01	11	10
0	0	1	3	2
1	4	5	7	6

mapa de Karnaugh de 4 variables

$X_3X_2 \backslash X_1X_0$	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

Mapa de Karnaugh de 5 variables

$X_3X_2 \backslash X_1X_0$	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$\overline{X_4}$ $(X_4=0)$

$X_3X_2 \backslash X_1X_0$	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

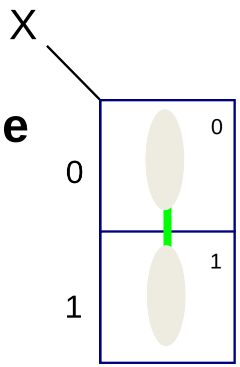
X_4 $(X_4=1)$

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

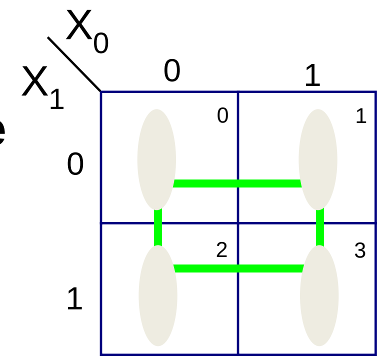
- Se dice que dos celdas de un mapa de Karnaugh son adyacentes cuando entre ellas solamente cambia el valor de una de las variables.
- Son adyacentes las celdas que tienen un lado en común en el mapa.
- También son adyacentes las celdas que están en los extremos del mapa.
- Las siguientes transparencias muestran las adyacencias entre celdas en mapas de Karnaugh de distinto tamaño.

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

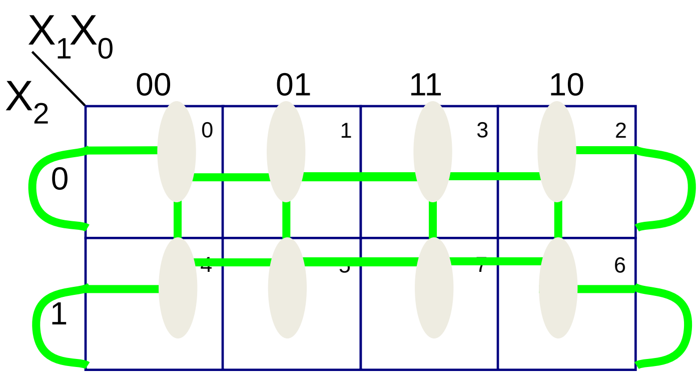
mapa de Karnaugh de 1 variable



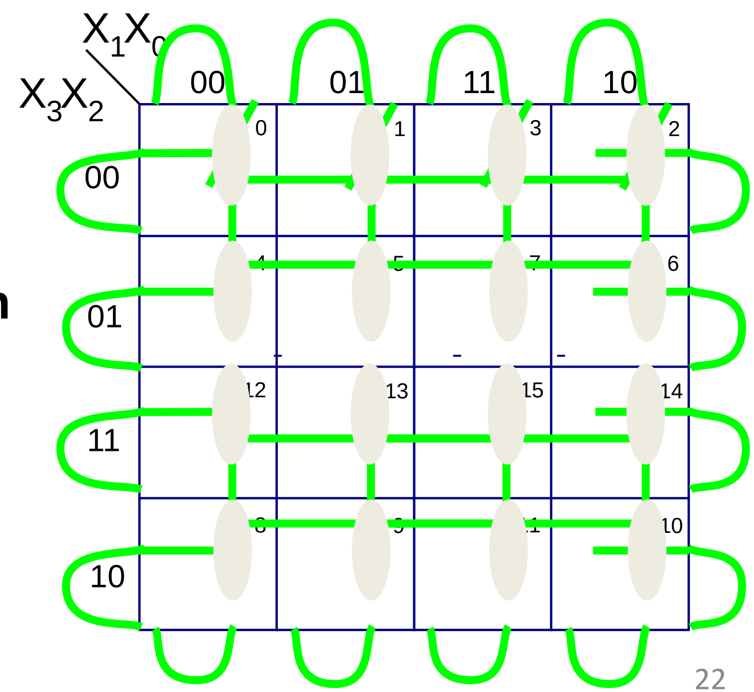
mapa de Karnaugh de 2 variables



mapa de Karnaugh de 3 variables

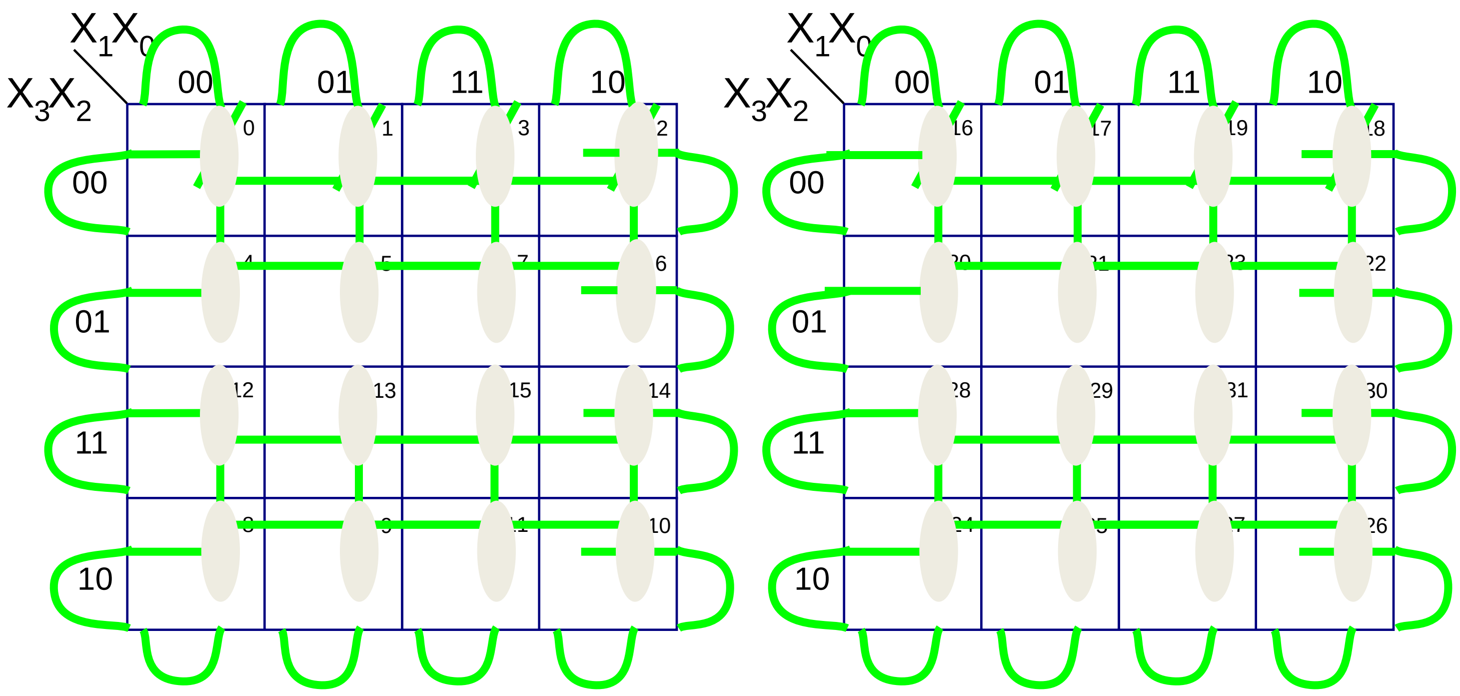


mapa de Karnaugh de 4 variables



SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

Mapa de Karnaugh de 5 variables



$\overline{X_4}$ $(X_4=0)$ X_4 $(X_4=1)$

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

- Para representar una función de conmutación con un mapa de Karnaugh, se coloca en cada celda el valor que tiene la función (‘0’ o ‘1’) para la combinación de las variables de la celda.
- Las celdas están colocadas cíclicamente de forma que cualquier celda que tenga un lado común con otra se diferencia de ella en una única variable.
- Dentro de un mapa de Karnaugh se denomina subcubo a una agrupación de 2^r celdas adyacentes 2 a 2 y en el que todas las celdas tienen el mismo valor (‘0’ o ‘1’).
- Los subcubos están formados por 1, 2, 4, 8, 16 . . . celdas.
- Un subcubo de 2^r celdas con valor ‘1’ en un mapa de Karnaugh de ‘n’ variables es una suma de minterms en la que todos ellos tienen ‘n-r’ variables iguales y ‘r’ variables diferentes. Por el **teorema de absorción**, o **sacando factor común** se pueden simplificar las ‘r’ variables diferentes.

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

- Ejemplo: ‘n’ = 4 variables, ‘r’ = 1. Se simplifica una variable y la expresión resultante está formada por 3 variables.

X_3X_2 \ X_1X_0	00	01	11	10
00	0	1	3	2
01	4	5: **1**	7: **1**	6
11	12	13	15	14
10	8	9	11	10

$$m_5 + m_7 = \overline{x_3} \cdot x_2 \cdot \overline{x_1} \cdot x_0 + \overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0 =$$
$$\overline{x_3} \cdot x_2 \cdot x_0 \cdot (x_1 + \overline{x_1}) = \overline{x_3} \cdot x_2 \cdot x_0$$

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

- Ejemplo: ‘n’ = 4 variables, ‘r’ = 2. Se simplifican 2 variables, y la expresión resultante está formada por 2 variables.

X_3X_2 \ X_1X_0	00	01	11	10
00	0	1	3	2
01	4	5: **1**	7: **1**	6
11	12	13: **1**	15: **1**	14
10	8	9	11	10

$$m_5 + m_7 + m_{13} + m_{15} =$$

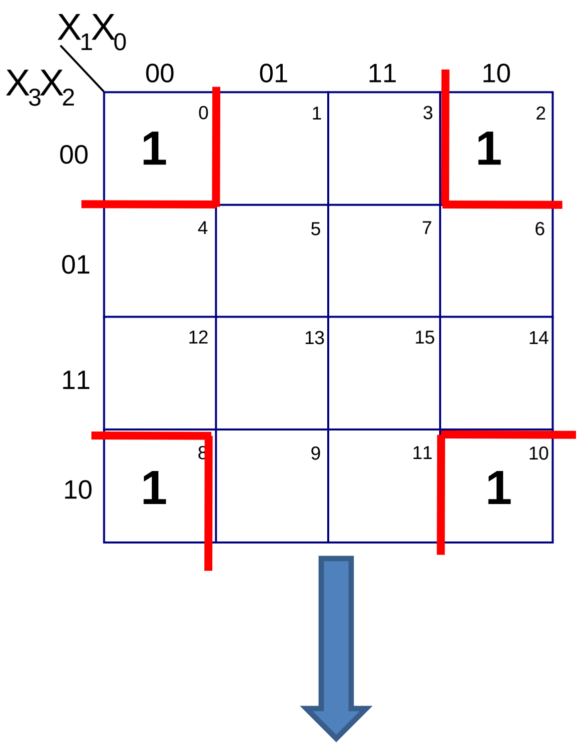
$$= \overline{x_3} \cdot x_2 \cdot \overline{x_1} \cdot x_0 + \overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0 + x_3 \cdot x_2 \cdot \overline{x_1} \cdot x_0 + x_3 \cdot x_2 \cdot x_1 \cdot x_0 =$$

$$= x_2 \cdot x_0 \cdot (\underbrace{\overline{x_3} \cdot \overline{x_1} + \overline{x_3} \cdot x_1 + x_3 \cdot \overline{x_1} + x_3 \cdot x_1}_{= 1}) = x_2 \cdot x_0$$

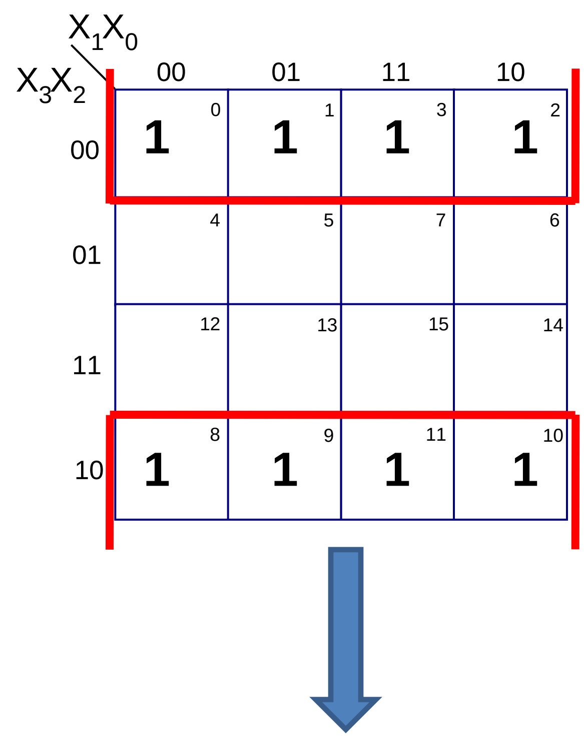
SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

- Para saber la expresión asociada a un subcubo de ‘1’s no es necesario escribir la expresión, sacar factor común y simplificar, si no que de forma visual se puede obtener el valor de la expresión:
  - Según el tamaño del cubo sabremos cuantas variables se van a simplificar:
    - 2 celdas => se simplifica 1 variable.
    - 4 celdas => se simplifican 2 variables.
    - 8 celdas => se simplifica 3 variables.
    - Etc
  - Las variables que se simplifican son aquellas que **NO** tienen el mismo valor en todas las celdas del subcubo.
  - Las variables que aparecen en la expresión simplificada son aquellas que **SI** tienen el mismo valor en todas las celdas del subcubo.
  - En la expresión simplificada las variables aparecerán:
    - Negadas si la variable vale ‘0’ en todas las celdas del subcubo.
    - Afirmada si la variable vale ‘1’ en todas las celdas del subcubo.

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN



$$\overline{x_2} \cdot \overline{x_0}$$



$$\overline{x_2}$$

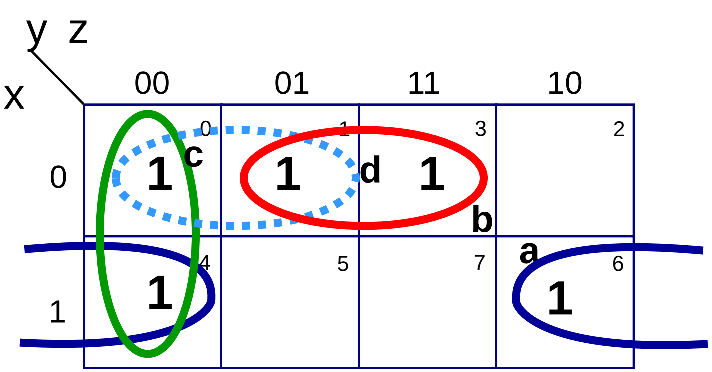
SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

- Una función de conmutación se puede expresar como la suma de los términos producto que corresponden a los subcubos necesarios para cubrir todas las celdas del mapa de Karnaugh que toman valor ‘1’.
- El objetivo es englobar todos los ‘1s’ del mapa con el mínimo número de subcubos (menos términos), y con subcubos los más grandes posible (términos con menos variables).
- Procedimiento:
  1. Cubrir con subcubos aquellos 1s que no puedan ser cubiertos combinados con otras celdas con 1s.
  2. Cubrir aquellas celdas que tengan una sola celda adyacente con un subcubo para dos 1s. Cubrir las restantes celdas que sólo puedan asociarse de dos en dos.
  3. Combinar aquellos 1s que den lugar a subcubos de 4 celdas pero no puedan formar parte de otros subcubos de 8 celdas o más.
  4. Repetir con cubos cada vez más grandes hasta que no haya 1s sin cubrir.

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

- Ejemplo: calcular la expresión como SdP simplificada de la función:

$$f(x,y,z)=\sum m(0,1,3,4,6)$$



- No existen 1s solos.
- La celda 6 sólo se puede agrupar con la celda 4. (subcubo a)
- La celda 3 sólo se puede agrupar con la celda 1. (subcubo b)
- La celda 0 se puede agrupar con la celda 4 (subcubo c) o con la celda 1 (subcubo d). Sólo se debe coger uno de los dos subcubos.
- No quedan 1s sin coger => FIN

- Soluciones:

Cogiendo los cubos a,b y c:

$$f(x,y,z)=x\cdot\overline{z}+\overline{x}\cdot z+\overline{y}\cdot\overline{z}$$

Cogiendo los cubos a,b y d:

$$f(x,y,z)=x\cdot\overline{z}+\overline{x}\cdot z+\overline{x}\cdot\overline{y}$$

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

- Una función de conmutación también se puede expresar como un producto de los términos suma que corresponden a los subcubos necesarios para cubrir todas las celdas del mapa de Karnaugh que toman valor ‘0’.
- La expresión asociada a un subcubo de ‘0s’ es:
  - Un término suma.
  - NO aparecen las variables que NO tienen el mismo valor en todas las celdas del subcubo.
  - SI aparecen las variables que SI tienen el mismo valor en todas las celdas del subcubo.
  - Las variables de la expresión simplificada aparecen:
    - Negadas, si tienen valor ‘1’ en todas las celdas del subcubo.
    - Afirmadas, si tienen valor ‘0’ en todas las celdas del subcubo.

SIMPLIFICACIÓN DE EXPRESIONES DE CONMUTACIÓN

- Ejemplo: calcular la expresión como PdS simplificado de la función:

$$f(x,y,z)=\prod M(2,5,7)$$

x \ y z	00	01	11	10
0	1 [0]	1 [1]	1 [3]	0 [2]
1	1 [4]	0 [5]	0 [7]	1 [6]

- La celda 2 no se puede agrupar en ningún subcubo.
- La celda 7 sólo se puede agrupar con la celda 5.
- No hay más ‘0s’ sin cubrir => FIN

- Solución:

$$f(x,y,z)=(x+\overline{y}+z)\cdot(\overline{x}+\overline{z})$$