Introducción al Diseño Lógico (E0301)

Ingeniería en Computación

Gerardo E. Sager

Clase 4 curso 2024

Clase 4

- Temas a tratar
  - Determinación de Funciones Lógicas a partir de Tablas de Verdad en sus formas “Suma de Productos” (SOP) y “Productos de Sumas” (POS)
  - Conversión entre expresiones de tipo POS y SOP.
  - Simplificación de Expresiones lógicas.
  - Codigos de Gray
  - Algebra Booleana y el mapa de Karnaugh como herramientas para simplificar y diseñar circuitos lógicos.

Tablas de Verdad – SOP

$F=X\cdot Y$		
X	Y	F
0	0	0
0	1	0
1	0	0
1	1	1

$F=X\cdot\overline{Y}$		
X	Y	F
0	0	0
0	1	0
1	0	1
1	1	0

$F=\overline{X}\cdot Y$		
X	Y	F
0	0	0
0	1	1
1	0	0
1	1	0

$F=\overline{X}\cdot\overline{Y}$		
X	Y	F
0	0	1
0	1	0
1	0	0
1	1	0

$F=?$		
X	Y	F
0	0	0
0	1	1
1	0	0
1	1	1

- Las filas donde F= 1, nos dicen que variables intervienen en la función y si intervienen directas o negadas
- Buscamos las filas de la TdV donde F=1
- Hacemos el producto de las variables de cada fila, intervienen negadas si valen cero
- F es la suma de los productos obtenidos
- OJO! esta expresión no es mínima, en el ejemplo de la izquierda:

$$F=X\cdot Y+\overline{X}\cdot Y=Y\left(X+\overline{X}\right)=Y$$

$$\rightarrow F=Y$$

Tablas de Verdad – SOP y POS

$F=?$			
X	Y	F	$\overline{F}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

- Si aplicamos el metodo visto obtenemos:

$$F=X\cdot Y+\overline{X}\cdot Y+X\cdot\overline{Y}=X\left(Y+\overline{Y}\right)+\overline{X}\cdot Y=X+\overline{X}\cdot Y=X+Y$$

- Podemos aplicar un método parecido a la función negada $\overline{F}$
- En este caso se invierten los valores de F y tomamos sólo el termino donde F=0 o sea $\overline{F}=\overline{X}\cdot\overline{Y}$
- Se niega otra vez la expresión y se aplica DeMorgan

$$\overline{\overline{F}}=\overline{\overline{X}\cdot\overline{Y}}=\overline{\overline{X}}+\overline{\overline{Y}}=X+Y$$

$F=?$			
X	Y	F	$\overline{F}$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0

EJEMPLO

- POS

$$F=\overline{\overline{F}}=\overline{\overline{X}\cdot Y+X\cdot\overline{Y}}$$

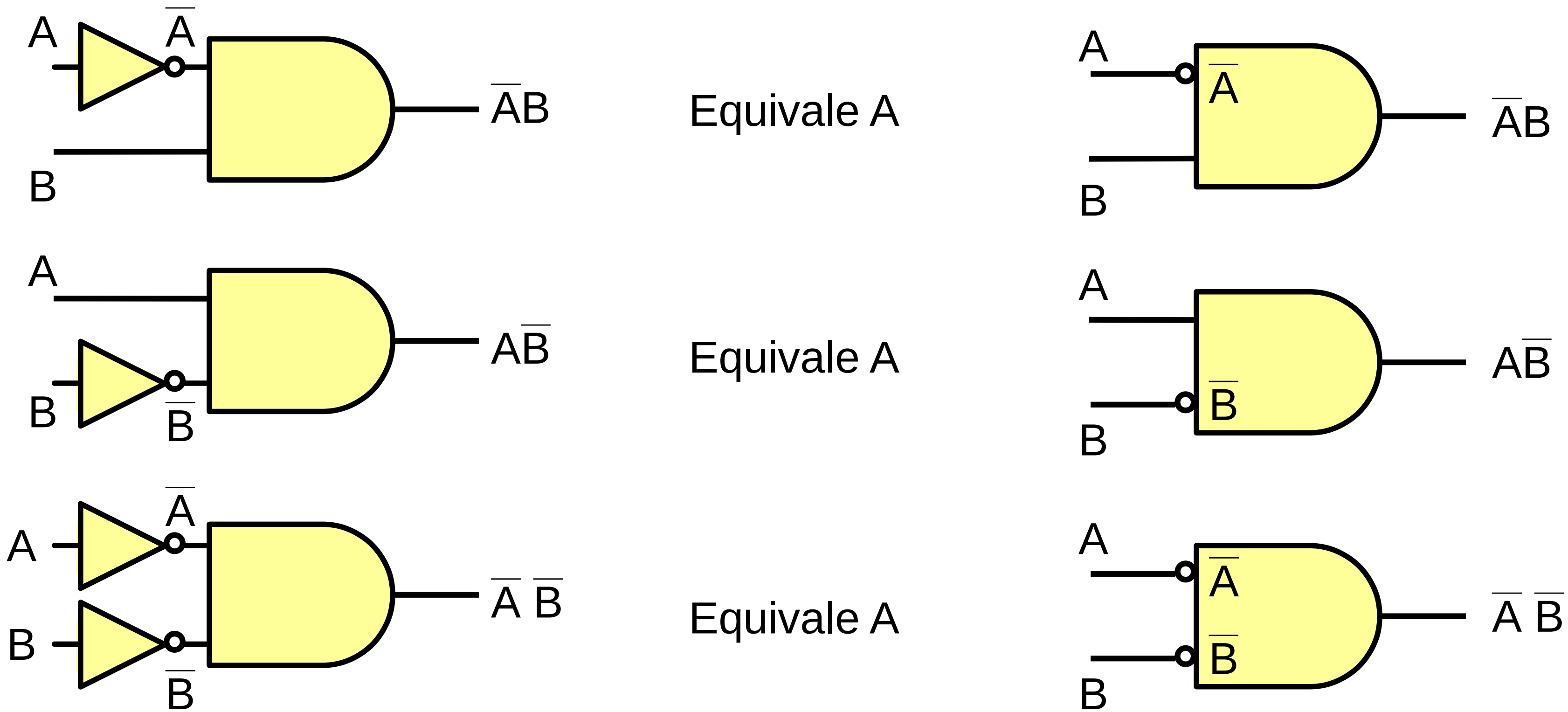
$$F=\overline{\overline{X}\cdot Y}\cdot\overline{X\cdot\overline{Y}}=\left(\overline{\overline{X}}+\overline{Y}\right)\cdot\left(\overline{X}+\overline{\overline{Y}}\right)=\left(X+\overline{Y}\right)\left(\overline{X}+Y\right)$$

- SOP

$$F=\overline{X}\cdot\overline{Y}+X\cdot Y$$

Puede demostrar que ambas expresiones son equivalentes?

Equivalencia de compuertas con entradas negadas



En una compuerta, una entrada con un circulo significa que se activa con la entrada en BAJO.
También puede verse como que la señal aplicada a la entrada se invierte antes de ingresar a la compuerta

Diseño de circuitos Combinatorios

- Procedimiento completo de diseño
  - Interprete el problema y establezca una tabla de verdad para describir su operación.
  - Variante SOP
    - Escriba el término AND para cada caso en que la salida sea 1
    - Escriba la expresión de SOP para la salida.
  - Variante POS
    - Escriba el término AND para cada caso en que la salida sea 0
    - Escriba la expresión de SOP para la salida negada.
    - Niegue lo obtenido y aplique DeMorgan para obtener POS
  - Simplifique si es posible
  - Implemente el circuito de compuertas con la versión final simplificada.

Ejemplo:

- Diseñar un circuito lógico de tres entradas, A, B, y C. La salida debe ser HIGH cuando la mayoría de las entradas es HIGH.

A	B	C	F	SOP	POS
0	0	0	0		$\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	0		$\overline{A}\,\overline{B}C$
0	1	0	0		$\overline{A}B\overline{C}$
0	1	1	1	$\overline{A}BC$	
1	0	0	0		$A\overline{B}\,\overline{C}$
1	0	1	1	$A\overline{B}C$	
1	1	0	1	$AB\overline{C}$	
1	1	1	1	ABC	

Aplicando SOP se obtiene
$F=\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC$
que se puede simplificar a
$F=AB+AC+BC$
(demuéstrelo)

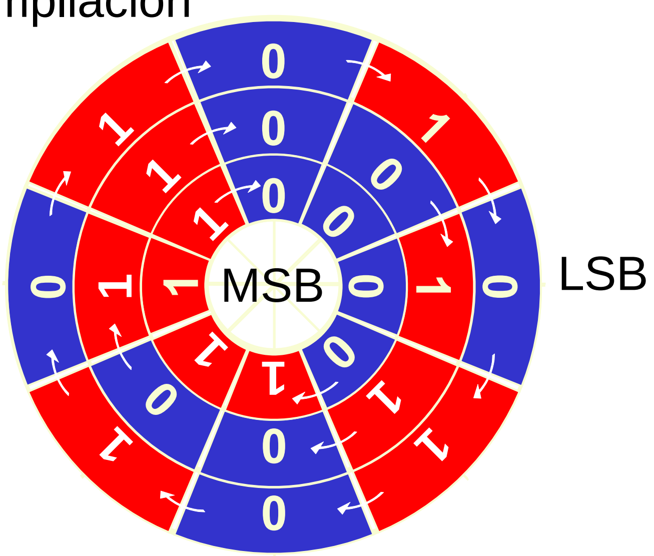
Aplicando POS se obtiene
$\overline{F}=\overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,\overline{B}C+\overline{A}B\overline{C}+A\overline{B}\,\overline{C}$
$\overline{\overline{F}}=\overline{\overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,\overline{B}C+\overline{A}B\overline{C}+A\overline{B}\,\overline{C}}$
$F=(\overline{\overline{A}\,\overline{B}\,\overline{C}})(\overline{\overline{A}\,\overline{B}C})(\overline{\overline{A}B\overline{C}})(\overline{A\overline{B}\,\overline{C}})$
$F=(A+B+C)(A+B+\overline{C})(A+\overline{B}+C)(\overline{A}+B+C)$
que se puede simplificar a
$F=(A+B)(A+C)(B+C)$
(demuéstrelo)

Diseño de Circuitos Combinatorios

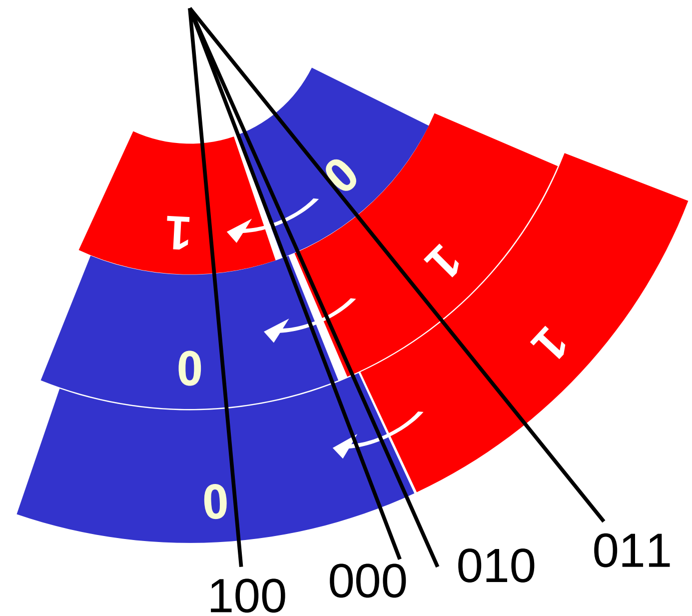
- Observaciones:
  - La primera etapa de la implementación es sencilla, sobre todo para un diseño basado en SOP
  - Para diseños basados en POS hay que realizar un paso más, pero la complejidad es similar.
  - La simplificación es más compleja y depende de la experiencia del diseñador, y su conocimiento de las propiedades dadas por los teoremas vistos anteriormente.
  - Veremos un método más sencillo que nos permite optimizar las expresiones
  - Antes vamos a ver dos temas que ayudarán a la comprensión del método:
    - Codigos de Gray
    - Representación de funciones multidimensionales

Código de Gray

- Para determinar la posición de un eje, tomamos un disco, y lo dividimos en anillos y en sectores y usamos algun tipo de sensor que nos permita determinar si corresponde a un 1 o a un 0.
- Por ejemplo podriamos recortar los sectores donde hay '1's y ubicar sensores de luz bajo el disco y una fuente de luz arriba, cuando se detecta la luz hay un 1 y si no hay un 0
- Si analizamos los dos sectores 011 y 100 contiguos entre sí, vemos que cambian todos los bits al pasar de uno al otro.
- Esto puede producir que por imperfecciones de alineación o de tiempos de respuesta de los sensores, entre 011 y 100, se produzcan valores intermedios como tambien vemos en la ampliación



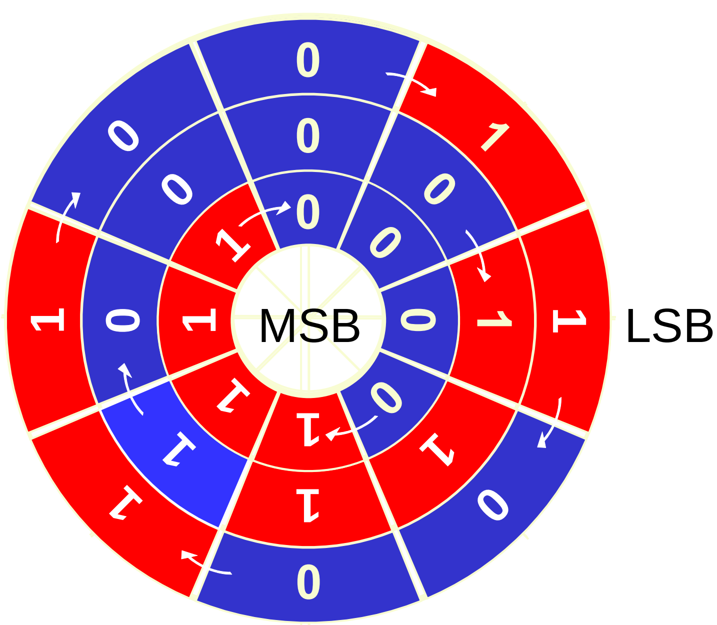
Orden Binario natural



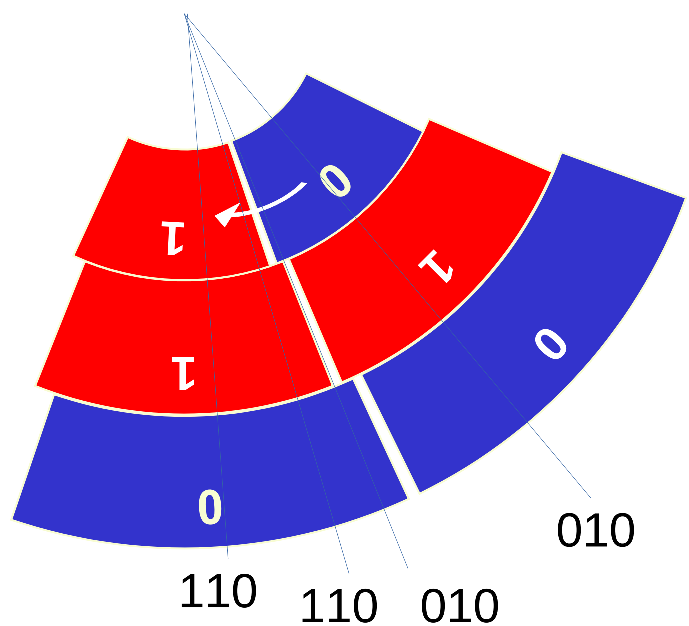
Imperfecciones en la transición generan lecturas erróneas

Código de Gray

- Estos valores no esperados pueden traernos problemas y vemos que una de las causas es que se produzcan multiples cambios de bits al pasar de un sector a otro, en el caso que analizamos hay tres cambios, en otros puede haber dos.
- Si conseguimos que entre sector y sector nunca se cambie más de un bit, eliminamos el problema.
- Necesitamos una manera de numerar los sectores que cumpla con esta propiedad, Esta manera de ordenar los valores, se llama un código de GRAY



Orden Código GRAY



Las imperfecciones en las transiciones NO causan errores

Código de Gray

La construcción del código es simple:

- Se comienza por el LSB, se le da valores. Al resto de los bits se los completa con 0. (a)
- Cuando no pueda hacer más combinaciones, copio abajo la imagen en espejo de lo que tenía hasta ahora, y al bit que sigue hacia el MSB lo pongo en 1 (b)
- Repito el proceso (c)

A	B	C
0	0	0
0	0	1

a)

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

a)

b)

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

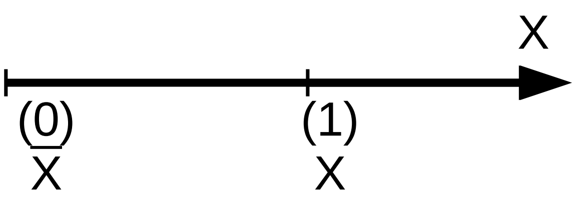
a)

b)

c)

Representacion de las funciones lógicas

Una Variable

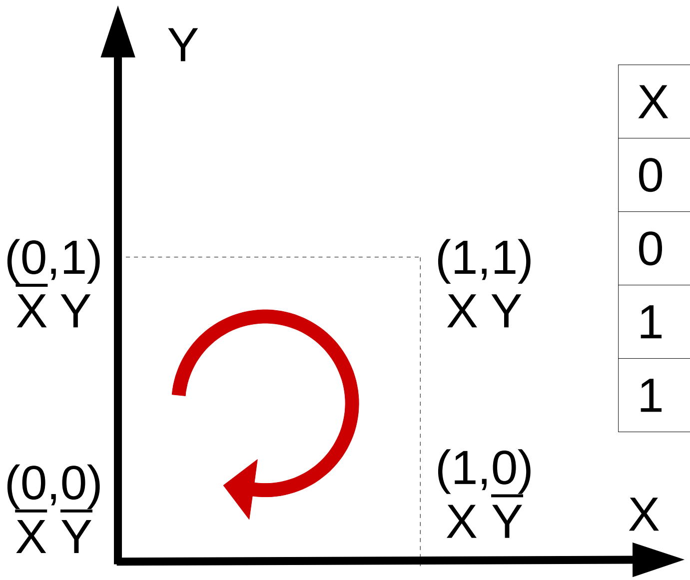


X		F(X)
0	$\overline{X}$	F(0)
1	X	F(1)

Esas funciones pueden tener solamente valores 0 o 1
Podemos tener:

- Función constante, o sea F(X) = 0 o F(x) = 1 para todo X
- Funcion Inversor: invierte la entrada
- Función Buffer: copia la entrada

Dos Variables



X	Y		F(X,Y)
0	0	$\overline{X}\ \overline{Y}$	F(0,0)
0	1	$\overline{X}$ Y	F(0,1)
1	0	X $\overline{Y}$	F(1,0)
1	1	X Y	F(1,1)

Esas funciones también solo pueden adoptar valores 0 o 1.

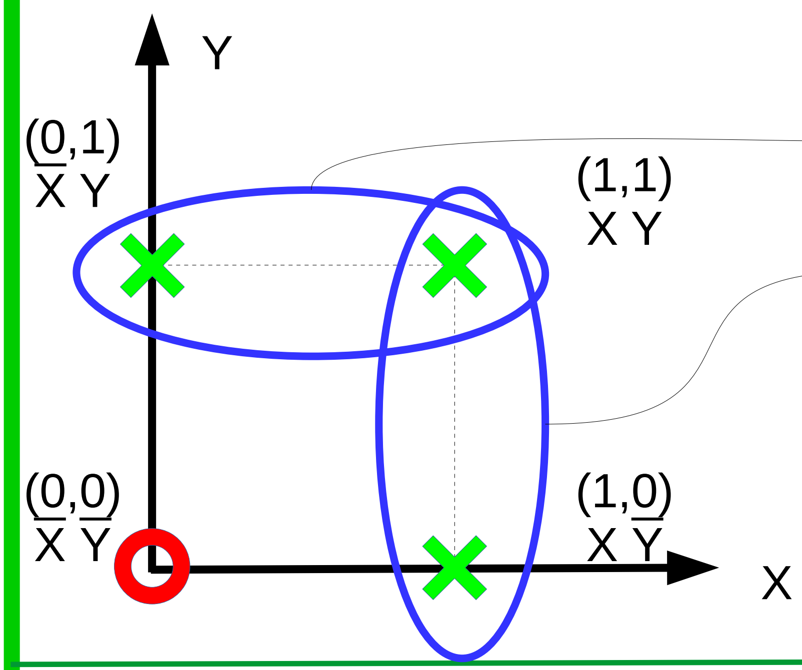
- La variedad de funciones es mayor
- En el grafico,podemos ver que entre vértices adyacentes se mantiene constante el valor de una de las variables
- La flecha muestra como se recorren los puntos adyacentes

Representacion de las funciones lógicas

Ejemplo: usaremos ✗ para simbolizar '1' y ○ para simbolizar '0'

X	Y		F(X,Y)
0	0	$\overline{X}\ \overline{Y}$	0
0	1	$\overline{X}\ Y$	1
1	0	$X\ \overline{Y}$	1
1	1	$X\ Y$	1

Por lo que ya vimos, ya sabemos que la función es X+Y



En el gráfico, podemos agrupar los vértices adyacentes que nos den un valor '1'

- Y vale 1, sin importar el valor de X. Podemos expresarlo con la variable Y
- X vale 1 sin importar el valor de Y Podemos expresarlo con la variable X
- La función entonces sería X+Y

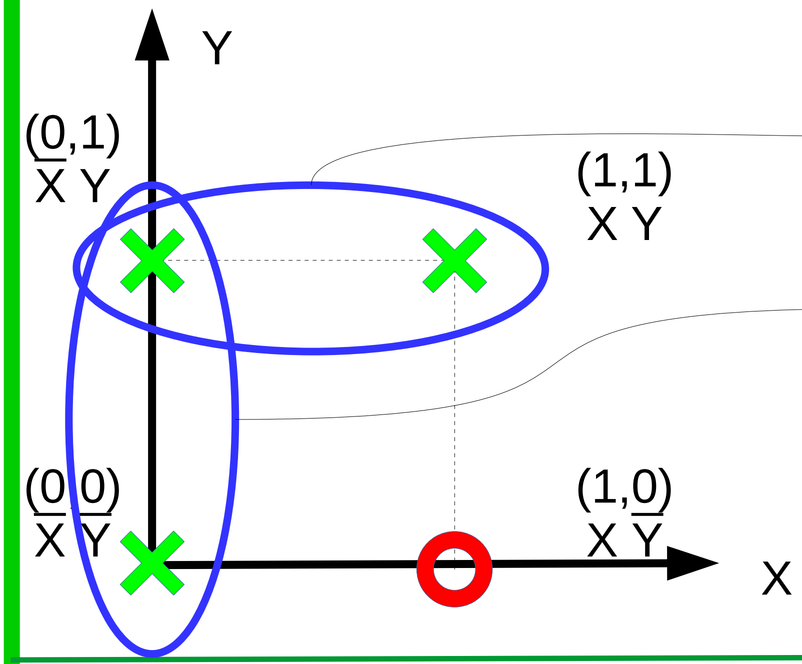
Representacion de las funciones lógicas

Otro Ejemplo:

X	Y		F(X,Y)
0	0	$\overline{X}\,\overline{Y}$	1
0	1	$\overline{X}\,Y$	1
1	0	$X\,\overline{Y}$	0
1	1	$X\,Y$	1

Aplicando lo visto antes:

$F= \overline{X}\,\overline{Y} + \overline{X}\,Y + X\,Y = \overline{X}\,(\overline{Y} + Y) + X\,Y = \overline{X} + X\,Y = \overline{X} + Y$

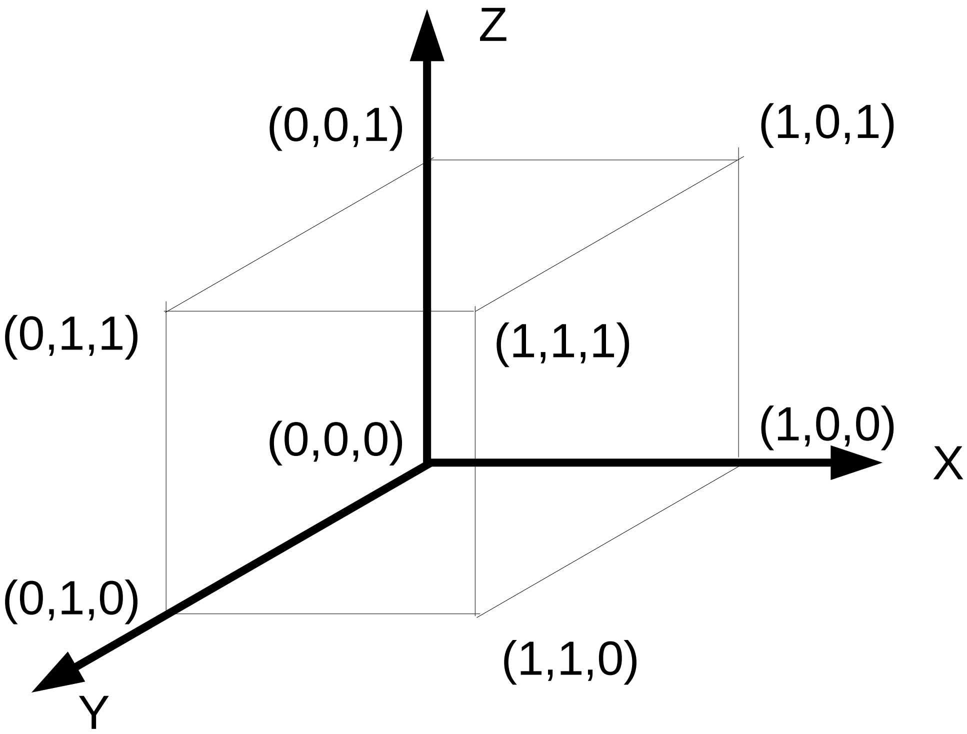


En el gráfico, podemos agrupar los vértices adyacentes que nos den un valor '1'

- Y vale 1, sin importar el valor de X. Podemos expresarlo con la variable Y
- X vale 0 sin importar el valor de Y Podemos expresarlo con la variable $\overline{X}$
- F = $\overline{X}$+Y
- Es mas sencillo obtener la función, pero tenemos que hacer el gráfico para ver las variables adyacentes
- Ahora vamos a ver que pasa con 3 variables

Representacion de las funciones lógicas

X	Y	Z		F(X,Y,Z)
0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$	F(0,0,0)
0	0	1	$\overline{X}\,\overline{Y}\,Z$	F(0,0,1)
0	1	0	$\overline{X}\,Y\,\overline{Z}$	F(0,1,0)
0	1	1	$\overline{X}\,Y\,Z$	F(0,1,1)
1	0	0	$X\,\overline{Y}\,\overline{Z}$	F(1,0,0)
1	0	1	$X\,\overline{Y}\,Z$	F(1,0,1)
1	1	0	$X\,Y\,\overline{Z}$	F(1,1,0)
1	1	1	$X\,Y\,Z$	F(1,1,1)

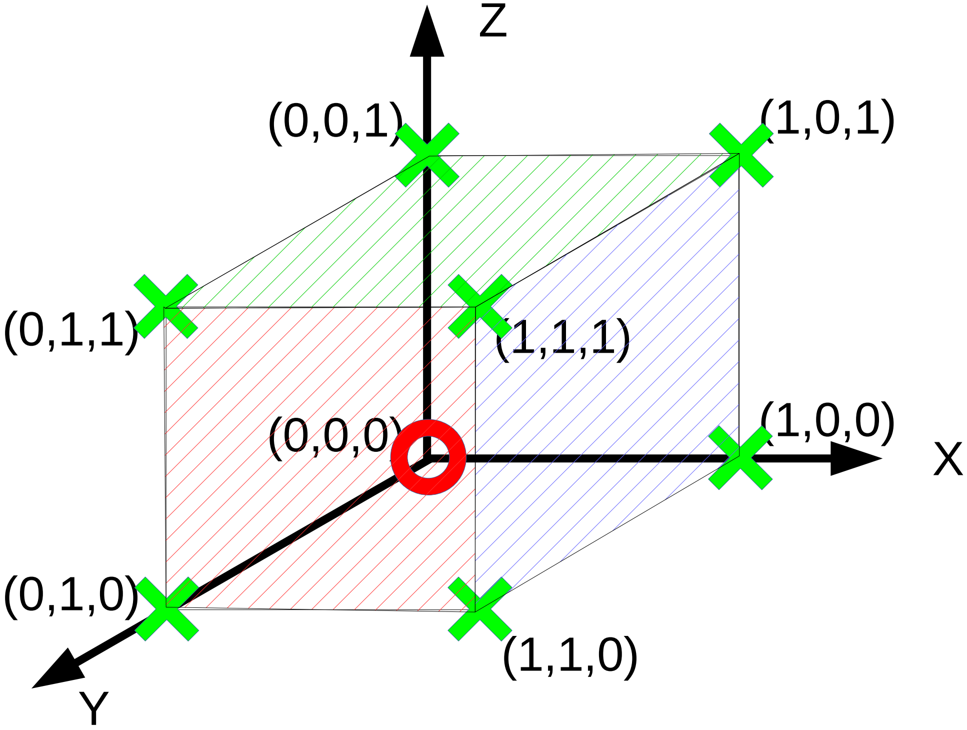


- En el gráfico,podemos ver que siguiendo una arista, dos variables mantienen su valor, y la otra toma los dos valores posibles
- Además podemos ver que en cada cara del cubo formado, una variable se mantiene constante y las otras dos toman todos los valores posibles

Representacion de las funciones lógicas

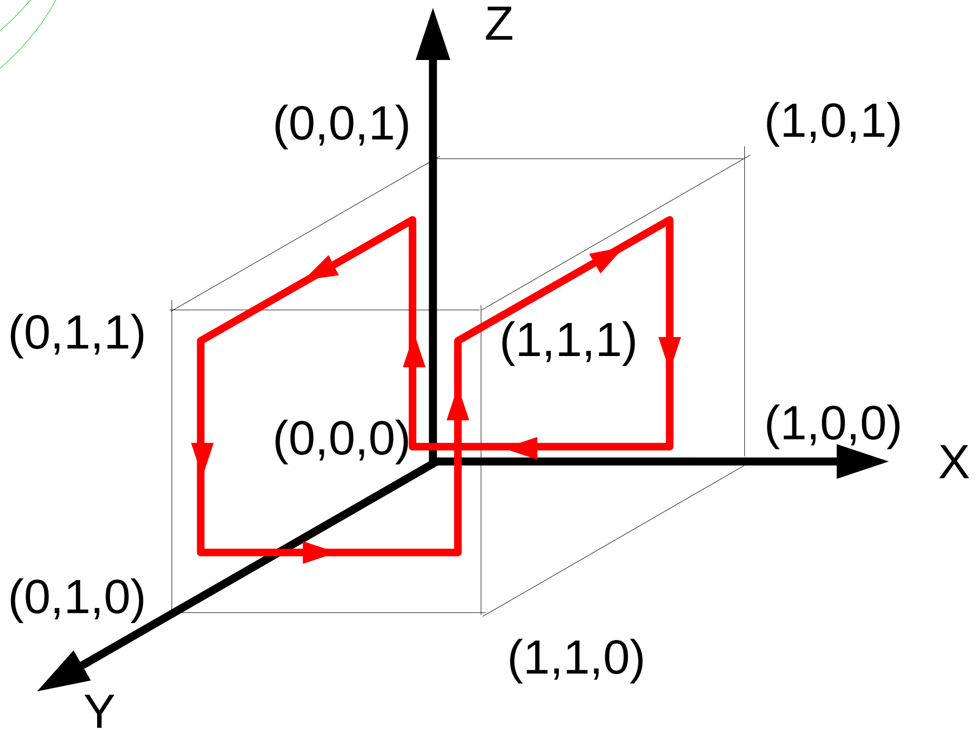
Ejemplo

X	Y	Z		F(X,Y,Z)
0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$	0
0	0	1	$\overline{X}\,\overline{Y}\,Z$	1
0	1	0	$\overline{X}\,Y\,\overline{Z}$	1
0	1	1	$\overline{X}\,Y\,Z$	1
1	0	0	$X\,\overline{Y}\,\overline{Z}$	1
1	0	1	$X\,\overline{Y}\,Z$	1
1	1	0	$X\,Y\,\overline{Z}$	1
1	1	1	$X\,Y\,Z$	1



- Se pueden agrupar las tres caras para cubrir todos los valores de la función donde vale 1.
- Vemos que las caras coinciden con los planos X=1, Y = 1 y Z=1 o sea F=X+Y+Z
- Otra vez, encontrar la función minimizada, es mas sencillo, pero hay que hacer el gráfico

Representacion de las funciones lógicas



X	Y	Z
0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

- Se pueden recorrer todos los vértices, siguiendo el camino marcado en rojo.
- Tambien siguiendo ese camino, se pueden ver cuales son los **puntos adyacentes.**
- Si ordenamos los valores de las variables siguiendo la linea, podemos ver que entre un valor y el siguiente solo cambia el valor de una de ellas.
- Esta manera de ordenar los valores se llama **CODIGO DE GRAY (Más adelante vamos a seguir con esto)**

Mapas de Karnaugh o K-Maps

Se basa en tabular los valores de la función en una tabla de doble entrada, que se construye para que permita visualizar las adyacencias.

Código Gray

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	1	1
0	1	0	1
1	1	0	1
1	1	1	1
1	0	1	1
1	0	0	1

	YZ	00	01	11	10
X		$\bar{Y}\,\bar{Z}$	$\bar{Y}\,Z$	$Y\,Z$	$Y\,\bar{Z}$
0	$\bar{X}$	0	1	1	1
1	X	1	1	1	1

La flecha roja recorre las casillas en el orden que muestra el Codigo de Gray.

	YZ	00	01	11	10
X		$\bar{Y}\,\bar{Z}$	$\bar{Y}\,Z$	$Y\,Z$	$Y\,\bar{Z}$
0	$\bar{X}$	0	1	1	1
1	X	1	1	1	1

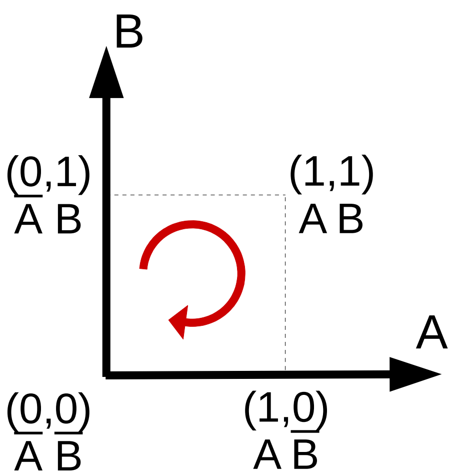
Rojo:Y – Verde: Z – Azul: X

F=X+Y+Z

Mapas de Karnaugh o K-Maps

- Dos Variables

Representamos en K-MAP



1) $z = AB + \bar{A}\,\bar{B}$

	$\bar{B}$	B
$\bar{A}$	1	0
A	0	1

2) $z = \bar{A}\,\bar{B} + \bar{A}\,B + A\,\bar{B}$

	$\bar{B}$	B
$\bar{A}$	1	1
A	1	0

3) $z = \bar{A}\,\bar{B} + \bar{A}\,B$

	$\bar{B}$	B
$\bar{A}$	1	1
A	0	0

Se pueden simplificar las expresiones?

1)No se pueden agrupar términos ni variables. corresponde a una XNOR

2)Si, vemos que pueden agruparse dos valores que representan $\bar{A}$ y otros dos que representan $\bar{B}$ => $z = \bar{A} + \bar{B}$

3)Si, vemos que pueden agruparse dos valores $\bar{A}$ => $z = \bar{A}$

Mapas de Karnaugh o K-Maps

Cuatro variables:

- No podemos representar 4 variables como hicimos para 1, 2 y 3 variables.
- Pero podemos construir el K-MAP
- A partir de la tabla de verdad, se muestra como se ubican los valores de la función en el K-MAP
- Los patrones de colores muestran la correspondencia entre la TdV y el K-MAP

A	B	C	D	F
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

LSB →		00	01	11	10
MSB ↓		$\bar{C}\,\bar{D}$	$\bar{C}\,D$	$C\,D$	$C\,\bar{D}$
00	$\bar{A}\,\bar{B}$	0	1	3	2
01	$\bar{A}\,B$	4	5	7	6
11	$A\,B$	12	13	15	14
10	$A\,\bar{B}$	8	9	11	10

Mapas de Karnaugh o K-Maps

Agrupar Variables en un K-MAP de 3 variables

<u>1 elemento:</u> Si tomamos 1 elemento necesitamos tres variables

	LSB	00	01	11	10
MSB		$\bar{B}\,\bar{C}$	$\bar{B}\,C$	$B\,C$	$B\,\bar{C}$
0	$\bar{A}$	0	1	0	1
1	A	1	0	1	0

$\bar{A}\,\bar{B}\,C$	$A\,\bar{B}\,\bar{C}$
$\bar{A}\,B\,\bar{C}$	$A\,B\,C$

$F = \bar{A}\,\bar{B}\,C + A\,\bar{B}\,\bar{C} + \bar{A}\,B\,\bar{C} + ABC$
Demuestre que F = A XOR B XOR C
XOR y XNOR estan asociados a patrones en damero en el K-MAP

	LSB	00	01	11	10
MSB		$\bar{B}\,\bar{C}$	$\bar{B}\,C$	$B\,C$	$B\,\bar{C}$
0	$\bar{A}$	1	0	0	0
1	A	0	0	1	0

$\bar{A}\,\bar{B}\,\bar{C}$
$A\,B\,C$

$F = \bar{A}\,\bar{B}\,\bar{C} + ABC$

Mapas de Karnaugh o K-Maps

Agrupar Variables en un K-MAP de 3 variables

<u>2 elementos:</u> Si agrupamos dos elementos podemos describirlos con DOS variables

- Para agrupar elementos tienen que ser ADYACENTES.
- Por como construimos el K-MAP, los elementos adyacentes en K-MAP tambien son adyacentes en el espacio de variables.
- Podemos considerar que los extremos están unidos a fines de veridicar la adyacencia

	LSB	00	01	11	10
MSB		$\overline{B}\,\overline{C}$	$\overline{B}\,C$	$B\,C$	$B\,\overline{C}$
0	$\overline{A}$	0	1	1	1
1	A	1	0	0	1

(rojo)	$\overline{A}\,C$
(amarillo)	$B\,\overline{C}$
(azul)	$A\,\overline{C}$

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

	LSB	00	01	11	10
MSB		$\overline{B}\,\overline{C}$	$\overline{B}\,C$	$B\,C$	$B\,\overline{C}$
0	$\overline{A}$	1	0	0	1
1	A	0	1	1	0

(rojo)	$\overline{A}\,\overline{C}$
(amarillo)	$A\,C$

$F = \overline{A}\,\overline{C} + A\,C = A \text{ XNOR } C$
Vemos que tambien hay un patrón alternado de '0's y '1's

Mapas de Karnaugh o K-Maps

Agrupar Variables en un K-MAP de 3 variables

<u>4 elementos:</u> Si agrupamos cuatro elementos podemos describirlos con UNA variables

- Para agrupar elementos tienen que ser ADYACENTES.
- Por como construimos el K-MAP, los elementos adyacentes en K-MAP tambien son adyacentes en el espacio de variables.
- Podemos considerar que los extremos están unidos a fines de verificar la adyacencia

	LSB	00	01	11	10
MSB		$\overline{B}\,\overline{C}$	$\overline{B}\,C$	$B\,C$	$B\,\overline{C}$
0	$\overline{A}$	0	1	1	1
1	A	0	1	1	1

(rojo)	C
(amarillo)	B

(rojo)	C
(azul)	$B\,\overline{C}$

Acá podemos cubrir los '1's agrupando en
a) dos bloques de 4 elementos (rojo y amarillo)
b) uno de 4 y otro de 2 elementos (rojo y azul)

a) $F = C + B$
b) $F = C + B\,\overline{C} = B + C$
Se llega al mismo resultado pero tengo que hacer un paso más.
Me conviene hacer grupos lo más grande posibles

	LSB	00	01	11	10
MSB		$\overline{B}\,\overline{C}$	$\overline{B}\,C$	$B\,C$	$B\,\overline{C}$
0	$\overline{A}$	0	1	1	0
1	A	1	1	1	1

(rojo)	C
(amarillo)	A

F=A+C

Mapas de Karnaugh o K-Maps

Agrupar Variables en un K-MAP de 4 variables

En un K-MAP de 4 variables puedo hacer grupos de 1, 2, 4 y 8 elementos adyacentes

LSB →		00	01	11	10
MSB ↓		$\bar{C}\,\bar{D}$	$\bar{C}\,D$	$C\,D$	$C\,\bar{D}$
00	$\bar{A}\,\bar{B}$	1	1	1	1
01	$\bar{A}\,B$	0	1	1	0
11	$A\,B$	1	0	0	1
10	$A\,\bar{B}$	0	1	1	0

1 elemento: 4 variables
2 elementos: 3 variables
4 elementos: 2 variables
8 elementos: 1 variable

	$\bar{A}\,\bar{B}$		$A\,B\,\bar{D}$
	$\bar{A}\,D$		$A\,\bar{B}\,D$

$$F = \bar{A}\,\bar{B} + \bar{A}\,D + A\,B\,\bar{D} + A\,\bar{B}\,D$$

Mapas de Karnaugh o K-Maps

Agrupar Variables en un K-MAP de 4 variables
En un K-MAP de 4 variables puedo hacer grupos de 1, 2, 4 y 8 elementos adyacentes
Se puede ver que las esquinas también son adyacentes

MSB ↓ \ LSB →	00 $\bar{C}\,\bar{D}$	01 $\bar{C}\,D$	11 $C\,D$	10 $C\,\bar{D}$
00 $\bar{A}\,\bar{B}$	1	1	1	1
01 $\bar{A}\,B$	1	1	1	1
11 $A\,B$	0	1	1	0
10 $A\,\bar{B}$	1	0	0	1

1 elemento: 4 variables
2 elementos: 3 variables
4 elementos: 2 variables
8 elementos: 1 variable

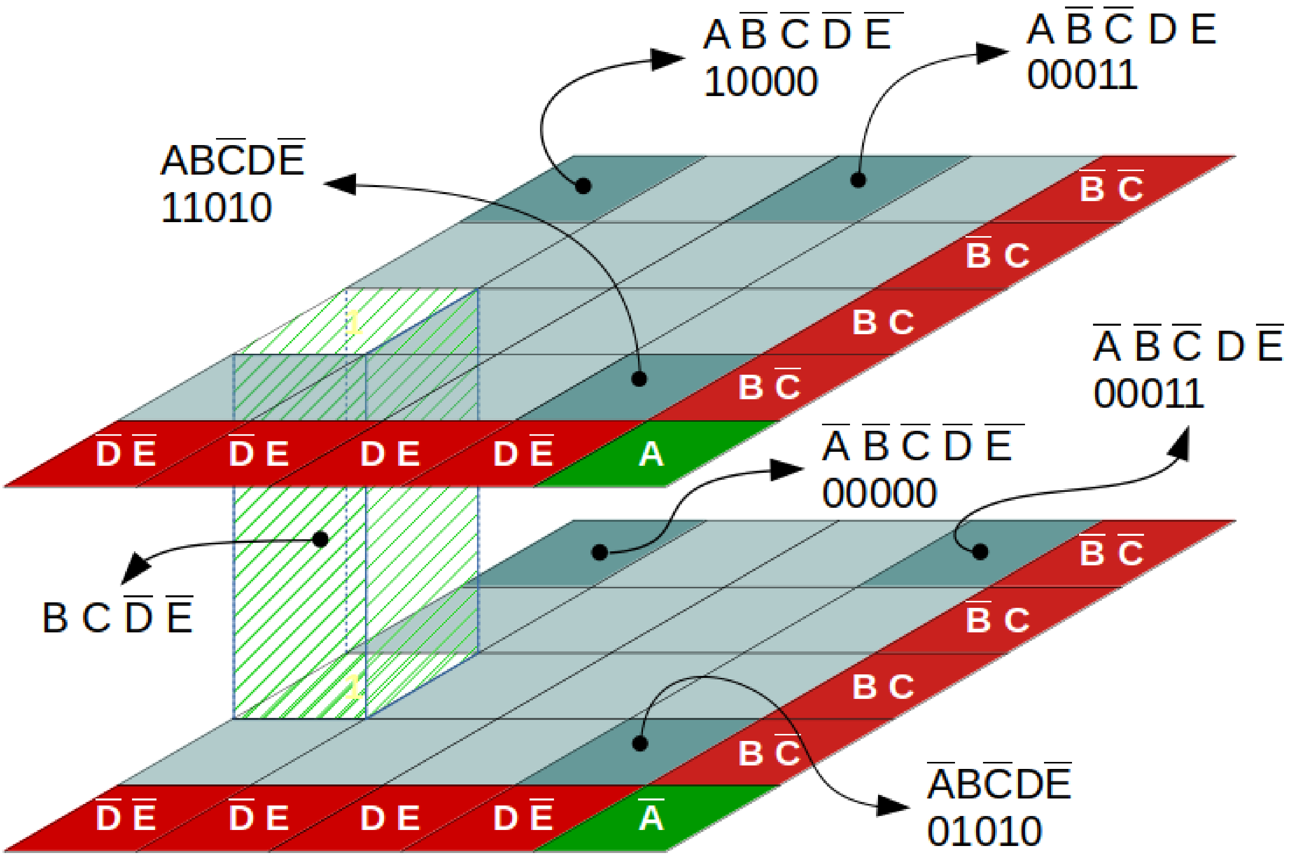
	$\bar{A}$
	$B\,D$
	$\bar{B}\,\bar{D}$

$$F = \bar{A} + B\,D + \bar{B}\,\bar{D}$$

Mapas de Karnaugh o K-Maps

Cinco variables:

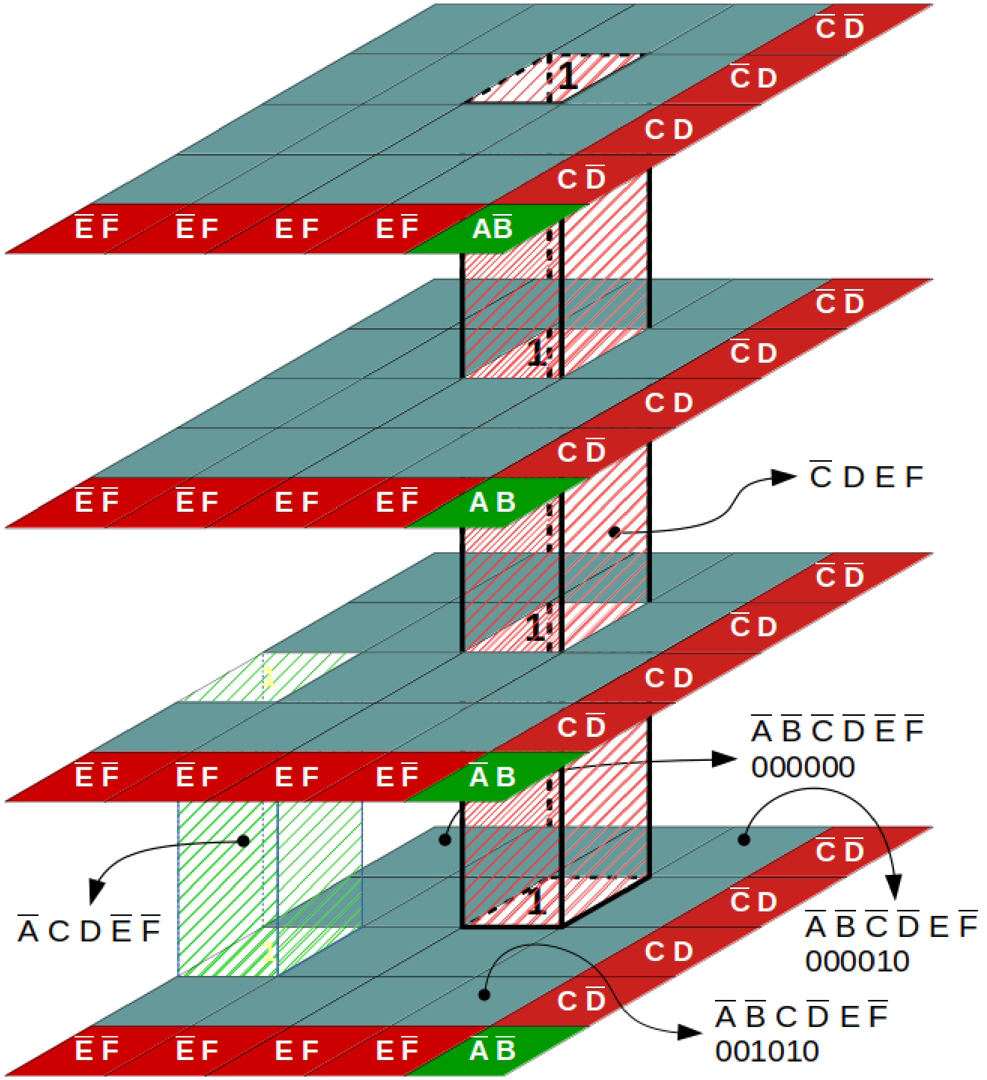
- No podemos representar 5 variables como hicimos para 1, 2, 3 y 4 variables
- Pero podemos construir un K-MAP de 4 variables para cada uno de los valores posibles de la quinta variable.
- En la figura se muestra un ejemplo donde se ve como agrupar valores adyacentes entre los dos K-Maps de 4 variables



Mapas de Karnaugh o K-Maps

Seis variables:

- Usamos la misma idea que para 5 variables.
- Ahora construimos 4 K-MAPs de 4 variables para cada una de las combinaciones posibles de la quinta y sexta variable.
- En la figura se muestra un ejemplo donde se ve como agrupar valores adyacentes entre los cuatro K-Maps de 4 variables



Mapas de Karnaugh o K-Maps

- Proceso de simplificación de K map:
  - Construir el K map, ubicar los 1s como se indica en la tabla de verdad.
  - Marcar 1s que no son adyacentes a ningún otro 1.
  - Marcar 1s que están en pares.
  - Marcar 1s en octetos aún si ya han sido marcados antes
  - Marcar cuartetos que tengan uno o más 1s no marcados antes.
  - Marcar cualquier par necesario para incluir los que ya no estén marcados.
  - Formar el OR (SOP) de términos generados por cada marca

Cuando una variable aparece negada y sin negar dentro de un grupo, esa variable se elimina de la expresión.
Las variables que son la misma para todos los cuadritos dentro del grupo, aparecen en la expresión final.

Mapas de Karnaugh o K-Maps

- Condiciones No-Importa
  - Cuando en una tabla de verdad, alguno de los valores de salida son indiferentes, se marcan con X.
  - En un K-MAP, si un cuadro está marcado con X, puede tomarse como 0 o como 1 según convenga para simplificar la expresión.
- Como llenar un mapa K a partir de una expresión Booleana
  - Si la expresión está expresada como SOP puede llenarse el K-MAP, poniendo 1s en los lugares correspondientes a los Productos y ceros en los restantes