UNIVERSIDAD ORT
Uruguay

ALGEBRA DE BOOLE

- Forma canónica
- Puertas lógicas
- Mapa de Karnaugh

ALGEBRA DE BOOLE

- George Boole fue un lógico y matemático británico.
- Escribió los libros:
  - *"The Mathematical Analysis of Logic" (1847)*
  - *"An Investigation of the Laws of Thought" (1854).*
- Desarrolló la lógica Simbólica mediante la cual las proposiciones pueden ser representadas mediante símbolos y la teoría que permite trabajar con estos símbolos,
  - ***Entradas: variables o proposiciones***
  - ***Salidas: respuestas***
- Dicha lógica cuenta con operaciones lógicas que siguen el comportamiento de reglas algebraicas y pueden ser tratadas mediante herramientas matemáticas.

ALGEBRA DE BOOLE

- Las proposiciones lógicas (frases o predicados de la lógica clásica) son aquellas que únicamente pueden **tomar valores Verdadero/Falso**, o preguntas cuyas únicas ***respuestas*** posibles sean ***Sí/No***.
- Conjunto de reglas de la Lógica Simbólica se le denomina Álgebra Booleana.
- Todas las variables y constantes del Álgebra Booleana, admiten sólo uno de dos valores en sus entradas y salidas: ***Sí/No, 0/1 o Verdadero/Falso***.
- Estos valores bivalentes y opuestos pueden ser representados por números binarios de un dígito denominado bit (Álgebra del Sistema Binario).

ALGEBRA DE BOOLE

- Todas las operaciones pueden representarse mediante elementos físicos:
  - Mecánicos
  - Eléctricos
  - Neumáticos o electrónicos que admiten entradas binarias o lógicas y que devuelven una respuesta (salida) también binaria o lógica
- Sus estados pueden ser:
  - Abierto/Cerrado, en el caso de interruptores
  - Encendida/Apagada si se refiere a una bombilla
  - Cargado/Descargado, si se tratase de un condensador
  - Nivel Lógico 0/Nivel lógico 1, para producir una salida lógica de un circuito semiconductor
  - Entre OTRAS

EXPRESIONES BOOLEANAS

Están compuestas de letras mayúsculas (A, B, C, D, …) y cada una de ellas representa la señal de un sensor.

El valor de las señales o de la función solo puede ser 0 o 1, falso o verdadero.

Además de letras, pueden existir los valores 0 o 1.

Las letras de expresiones booleanas pueden estar conectadas por medio de los operadores lógicos: $\wedge$ (y), $\vee$ (o) y $\neg$ (negación). El operador "y" es una multiplicación lógica, el "o" es una suma lógica y la negación es el complemento.

Ejemplo: $F(A, B, C) = A'B + ABC + C(B' + A)$

FORMAS CANÓNICAS

- Se conoce como término canónico de una función lógica a todo producto o suma en la cual aparecen todas las variables en su forma directa o inversa.
- Existen 2 formas canónicas:
  - Minitérminos
  - Maxitérminos
- La forma canónica también se conoce como fusión equivalente, ya que es una forma alternativa o equivalente de representar los diferentes resultados de la tabla de verdad.

FORMAS CANÓNICAS - MAXITÉRMINOS

Es una expresión lógica de “n” variables que consiste únicamente en la disyuncion lógica (OR) y el operador complemento o negación (NOT).

FORMAS CANÓNICAS - MAXITÉRMINOS

Para escribir la forma canónica de una función lógica, se realiza el siguiente procedimiento:

1. Escribir el maxitérmino de cada uno de los “0” en la tabla de verdad.
    1. Se considera que un 1 en la tabla de verdad, corresponde a un valor negado (a’ = 1, siendo “a” la variable original).
    2. Se considera que un 0 en la tabla de verdad, corresponde a un valor no negado (a = 0, siendo “a” la variable original).
2. Realizar el producto de todas las sumas (maxitérminos) halladas.

FORMAS CANÓNICAS – EJEMPLO MAXITÉRMINOS

X	Y	F(X,Y)	Maxitérmino
0	0	1	-
0	1	0	X+Y’
1	0	0	X’+Y
1	1	1	-

Forma canonica: $F(X,Y) = (X + Y')(X' + Y)$

FORMAS CANÓNICAS - MINITÉRMINOS

Es una expresión lógica de "n" variables consistentes únicamente en el operador conjunción lógica (AND) y el operador complemento o negación (NOT).

FORMAS CANÓNICAS - MINITÉRMINOS

Para escribir la forma canónica de una función lógica, se realiza el siguiente procedimiento:

1. Escribir el minitérmino de cada uno de los "1" en la tabla de verdad.
    1. Se considera que un 0 en la tabla de verdad, corresponde a un valor negado (a' = 0, siendo "a" la variable original).
    2. Se considera que un 1 en la tabla de verdad, corresponde a un valor no negado (a = 1, siendo "a" la variable original).
2. Realizar la suma de todos los productos (minitérminos) hallados.

FORMAS CANÓNICAS – EJEMPLO MINITÉRMINOS

X	Y	F(X,Y)	Minitérmino
0	0	1	X'Y'
0	1	0	-
1	0	0	-
1	1	1	XY

Forma canonica: $F(X,Y) = X'Y' + XY$

EXPRESIONES CANONICAS - EJEMPLO MINITÉRMINO

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

$$F = A'B'C'D + A'B'CD + AB'C'D + AB'CD' + AB'CD$$

$$F = \boxed{A'B'C'D} + \boxed{A'B'CD} + \boxed{AB'C'D} + \boxed{AB'CD'} + \boxed{AB'CD}$$

EXPRESIONES CANONICAS - EJEMPLO MAXITÉRMINO

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

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

EXPRESIONES CANONICAS – EJERCICIO

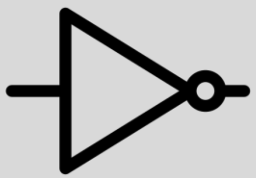
A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

OPERACIONES LOGICAS

- Las puertas lógicas son el bloque fundamental de construcción de todos los circuitos lógicos digitales, de forma que las funciones lógicas se implementan interconectando compuertas.
- Para representar las puertas lógicas se utilizan
  - Símbolos distintivos: corresponden a los utilizados comúnmente por la industria digital.
  - Símbolos rectangulares: IEEE/ANSI 91-1984

PUERTAS LOGICAS

NOT

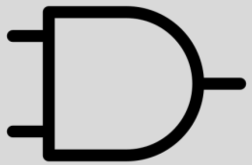


X	F
0	1
1	0

$$F = X'$$

Produce un nivel logico de salida opuesto al suministrado como entrada.

AND



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

$$F = XY$$

Produce un nivel alto a la salida soló cuando todas sus entradas están en un nivel alto.

PUERTAS LOGICAS

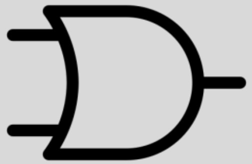
XOR

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

$$F = X'Y + XY'$$

Produce una salida verdadera solo cuando sus entradas son diferentes.

OR



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

$$F = X + Y$$

Produce una salida verdadera cuando cualquira de sus entradas está en un nivel alto.

PUERTAS LOGICAS

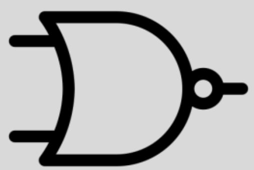
NAND

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

$$F = (XY)'$$

Produce una salida falsa solamente si todas sus entradas son verdaderas.

NOR



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

$$F = (X + Y)'$$

Produce una salida verdadera solamente si todas sus entradas son falsas.

TEOREMAS

	Teorema		Dual
1	$0A = 0$	Nulo	$1 + A = 1$
2	$1A = A$	Neutro	$0 + A = A$
3	$AA = A$	Idempotencia	$A + A = A$
4	$AA' = 0$	Complemento	$A + A' = 1$
5	$AB = BA$	Conmutativa	$A + B = B + A$
6	$ABC = A(BC)$	Asociativa	$A + B + C = A + (B + C)$
7	$(AB\ldots Z)' = A' + B' +\ldots+ Z'$	Ley de De Morgan	$(A + B +\ldots + Z)' = A'B'\ldots Z'$
8	$AB + AC = A(B + C)$	Distributiva	$(A + B)(A + C) = A + BC$
9	$AB + AB' = A$	Combinación	$(A + B)(A + B') = A$
10	$A + AB = A$	Cobertura/Absorción	$A(A + B) = A$
11	$A + A'B = A + B$		$A(A' + B) = AB$
12	$CA + CA'B = CA + CB$		$(C + A)(C + A' + B) = (C + A)(C + B)$
13	$AB + A'C + BC = AB + A'C$	Consenso	$(A + B)(A' + C)(B + C) = (A + B)(A' + C)$

SIMPLIFICACIÓN ALGEBRAICA DE 3 VARIABLES

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

$$F = AB\overline{C} + A\overline{B}\overline{C} + A\overline{B}C + A\overline{C} + C \longrightarrow$$ Aplic prop. distributiva

$$F = A\overline{C}(B + \overline{B}) + A\overline{B}C + A\overline{C} + C \longrightarrow$$ Aplic complemento $(B+\overline{B})=1$

$$F = A\overline{C} + A\overline{B}C + A\overline{C} + C \longrightarrow$$ Aplic idempotencia $(A\overline{C}+A\overline{C})=A\overline{C}$

$$F = A\overline{C} + A\overline{B}C + C \longrightarrow$$ Aplic prop. distributiva

$$F = A\overline{C} + C(1 + A\overline{B}) \longrightarrow$$ Aplic dominacion (1+AB)=1

$$F = A\overline{C} + C \longrightarrow$$ Aplic prop. distributiva

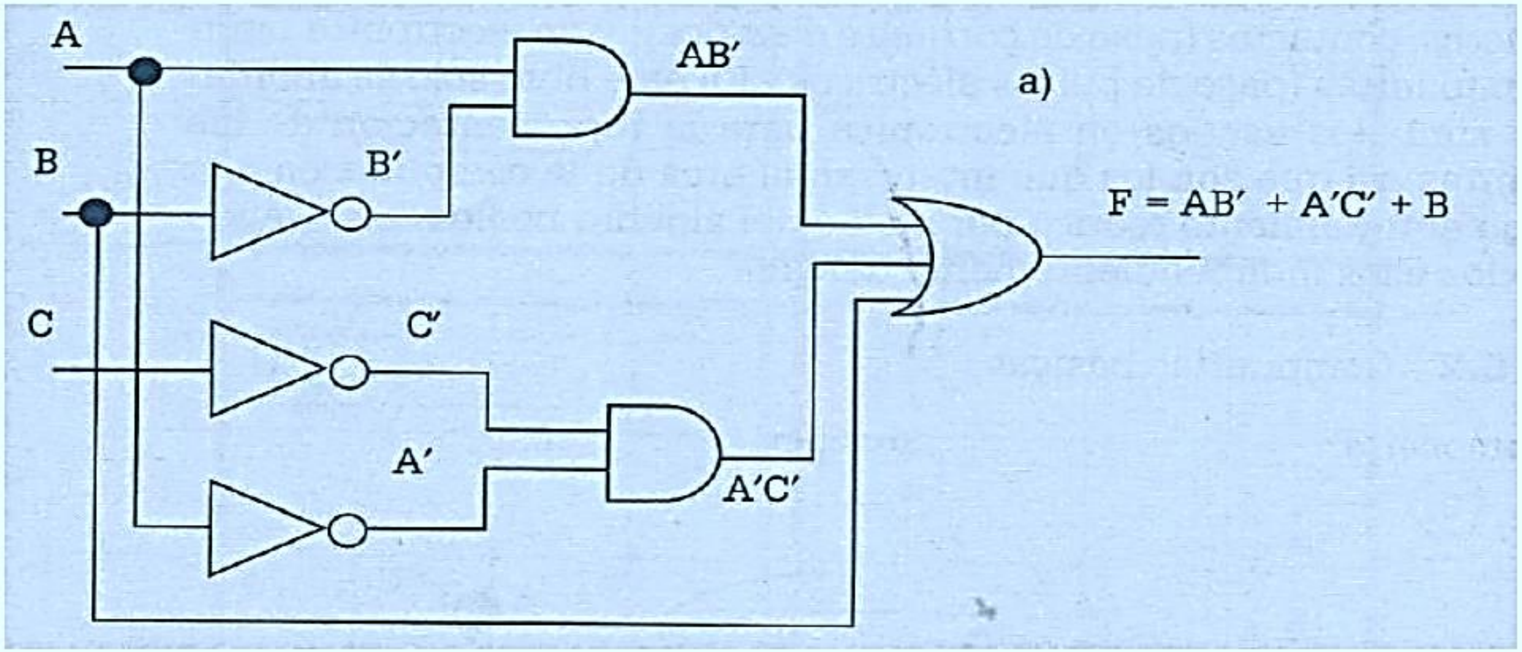
$$F = (A + C)(\overline{C} + C) \longrightarrow$$ Aplic complemento $(\overline{C}+C)=1$

$$\boxed{F = A + C}$$

PUERTAS LÓGICAS – EJEMPLO 1

Ejemplo: Representar en compuerta lógicas la siguiente expresión:

$F = AB' + A'C' + B$



PUERTAS LÓGICAS – EJEMPLO 2

Supóngase que, partiendo del ***enunciado*** verbal de un determinado problema, se tiene la siguiente ***expresión*** booleana*:*

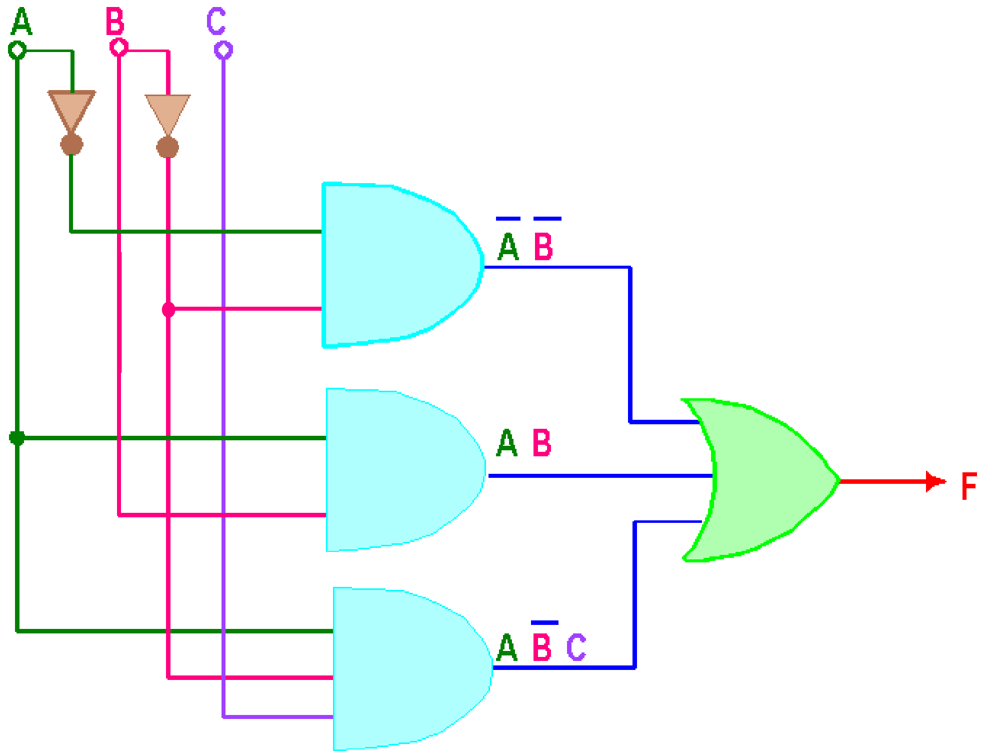
$$F(A, B, C) = A' B' + A B + A B' C$$

Deseamos obtener el diagrama del circuito lógico que realice esta ***función***.

- Las variables **A**, **B** y **C** serán las entradas del circuito y **F** será la salida.
- De la expresión observamos que se tienen tres términos, cada uno de los cuales requiere de una compuerta **Y:**
  - las dos primeras de dos entradas
  - una tercera de tres entradas.
- La salida de cada una de estas compuertas es la entrada de una compuerta **O**. A la salida de esta compuerta se tendrá la ***función*** de salida.
- Pero antes, por cada variable negada, se requiere que ésta pase por un ***inversor***.

PUERTAS LÓGICAS – EJEMPLO 2

$$F(A,B,C) = A'B' + AB + AB'C$$



PUERTAS LÓGICAS – EJEMPLO 3

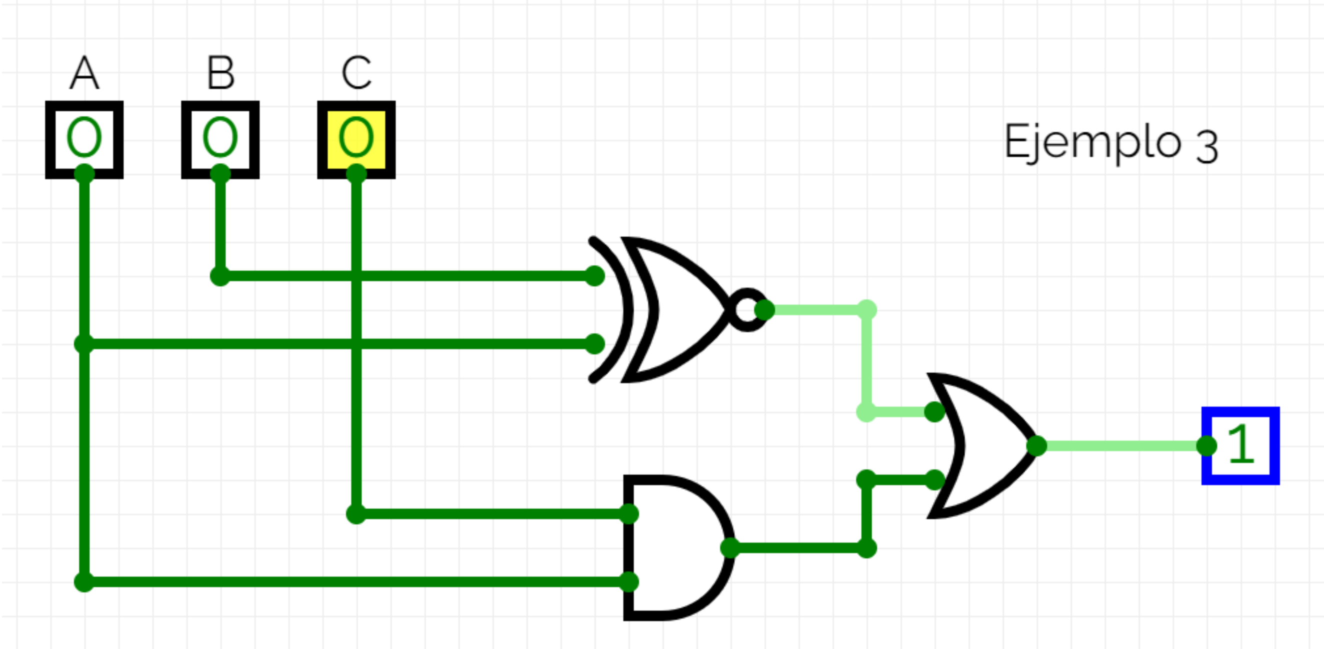
El circuito anterior puede reducirse utilizando los postulados y teoremas. Realizamos lo que se denomina *SIMPLIFICACIÓN DE FUNCIONES BOOLEANAS*

- **F = A'B' + AB + AB'C** = A'B'+AB(C+C')+AB'C = A'B'+**ABC**+**ABC**+ABC'+AB'C
- F = A'B'+(**ABC**+ABC')+ (**ABC** +AB'C) = A'B'+AB(C+C')+ AC(B+B)
- **F = A' B' + A B + A C**

Ahora la expresión queda con tres compuertas Y de dos entradas cada una, pero observamos que los dos primeros términos forman la O EXCLUSIVA NEGADA, por lo tanto, la función queda: **F = (A ⊕ B)' + A C**

PUERTAS LÓGICAS – EJEMPLO 3

$$F = (A \oplus B)' + AC$$



MAPA DE KARNAUGH

- Método para minimizar expresiones booleanas.
- Consiste en la representación bidimensional de la tabla de verdad de la función a simplificar.
- Tabla con 2^n casillas, con n = nº de variables.
- El mapa puede realizarse por minitérminos o maxitérminos.
  - Minitérminos: Se agrupan los “1” en el mapa.
  - Maxitérminos: Se agrupan los “0” en el mapa.

MAPA DE KARNAUGH – REGLA SIMPLIFICACIÓN

- Se deben agrupar los “1” o “0” adyacentes con grupo rectangulares.
- Los grupos deben contener únicamente 2^n elementos (1, 2, 4, 8, 16, …).
- Cada grupo genera un término en la expresión simplificada.
- En el término de cada grupo aparecen las variables comunes a todos los “1” o “0” del grupo.
- Los grupos se construyen desde los grupos de mayor número de elementos a los grupos más pequeños.
- Cada valor lógico puede pertenecer a múltiples grupos.
- Cada grupo debe tener al menos un único elemento.

MAPA DE KARNAUGH - 1 VARIABLE

A	0	1
$f =$	$f(0)$	$f(1)$

MAPA DE KARNAUGH - 2 VARIABLE

$f(A,B) =$

A \ B	0	1
0	f (0,0)	f (0,1)
1	f (1,0)	f (1,1)

MAPA DE KARNAUGH - 3 VARIABLE

$f(A,B,C) =$

A \ BC	00	01	11	10
0	$f(0,0,0)$	$f(0,0,1)$	$f(0,1,1)$	$f(0,1,0)$
1	$f(1,0,0)$	$f(1,0,1)$	$f(1,1,1)$	$f(1,1,0)$

MAPA DE KARNAUGH - 4 VARIABLE

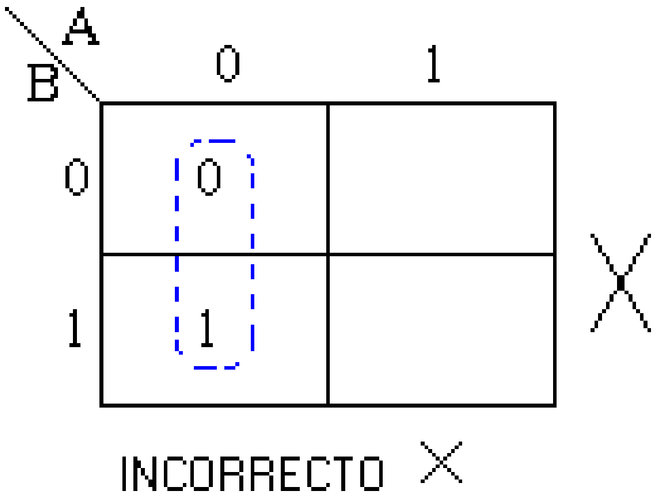
f (A,B,C,D) =

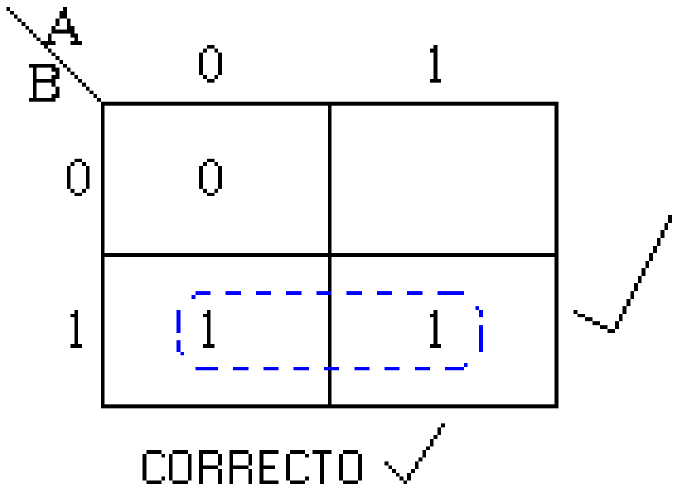
AB \ CD	00	01	11	10
00	f (0,0,0,0)	f (0,0,0,1)	f (0,0,1,1)	f (0,0,1,0)
01	f (0,1,0,0)	f (0,1,0,1)	f (0,1,1,1)	f (0,1,1,0)
11	f (1,1,0,0)	f (1,1,0,1)	f (1,1,1,1)	f (1,1,1,0)
10	f (1,0,0,0)	f (1,0,0,1)	f (1,0,1,1)	f (1,0,1,0)

MAPAS DE KARNAUGH - MINITÉRMINOS

1. LAS AGRUPACIONES SON EXCLUSIVAMENTE DE UNOS.

Esto implica que ningún grupo puede contener ningún cero.

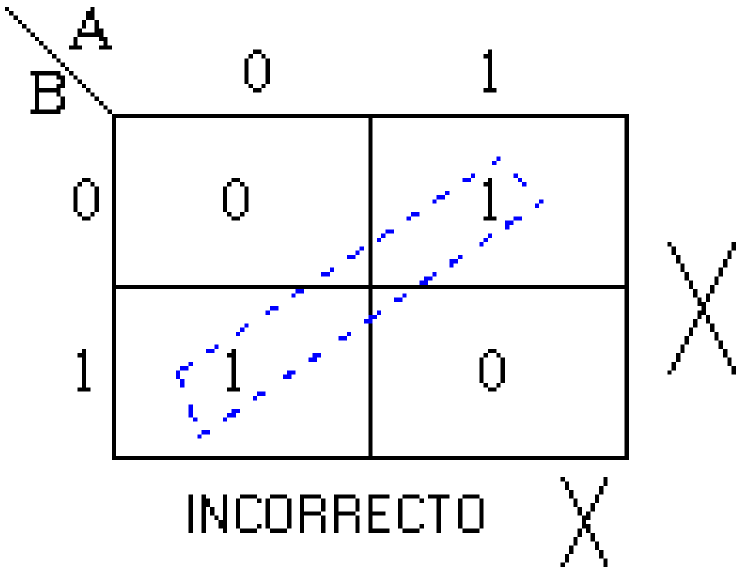


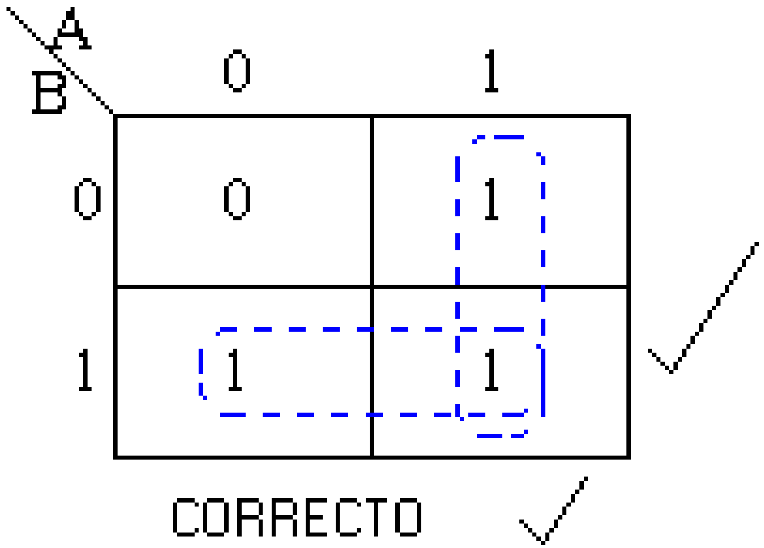


MAPAS DE KARNAUGH - MINITÉRMINOS

2. LAS AGRUPACIONES ÚNICAMENTE PUEDEN HACERSE EN HORIZONTAL Y VERTICAL.

Esto implica que las diagonales están prohibidas.

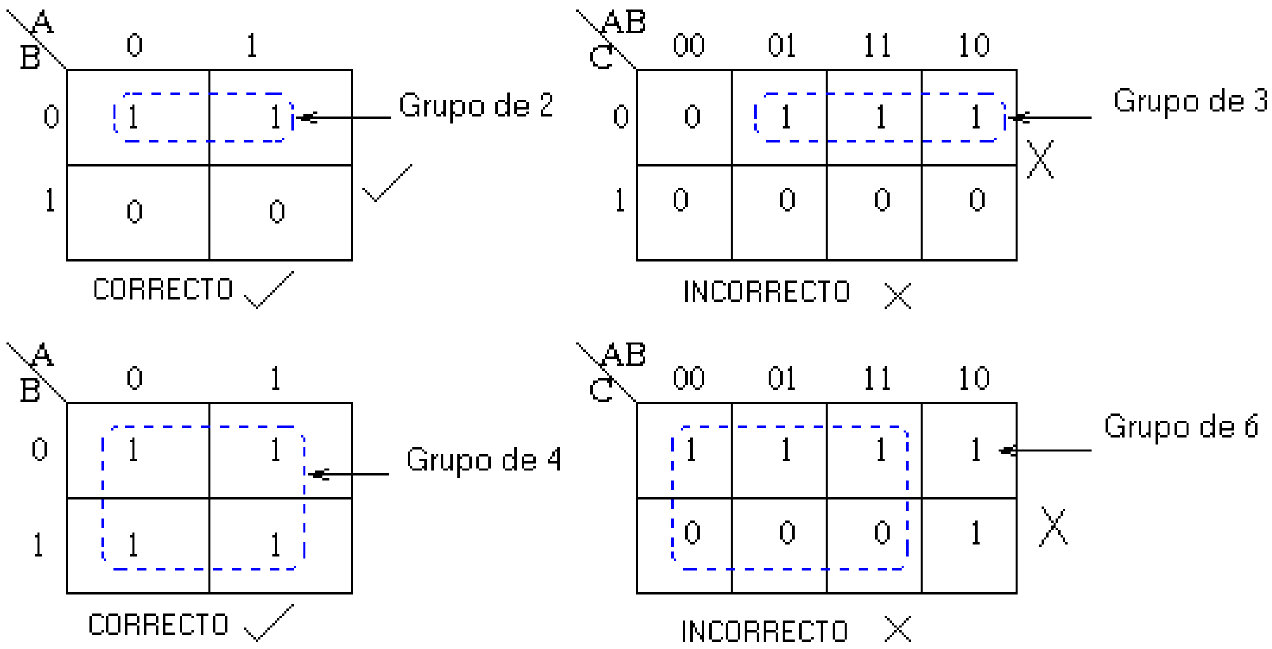




MAPAS DE KARNAUGH - MINITÉRMINOS

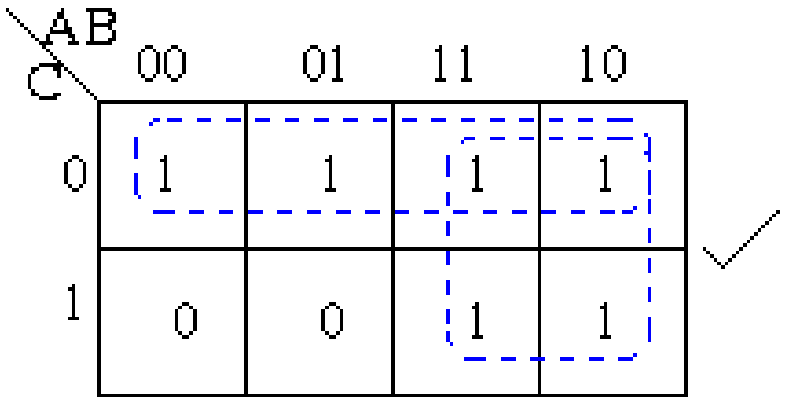
3. LOS GRUPOS HAN DE CONTENER 2^n ELEMENTOS.

Es decir que cada grupo tendrá 1, 2, 4, 8, ... número de unos.

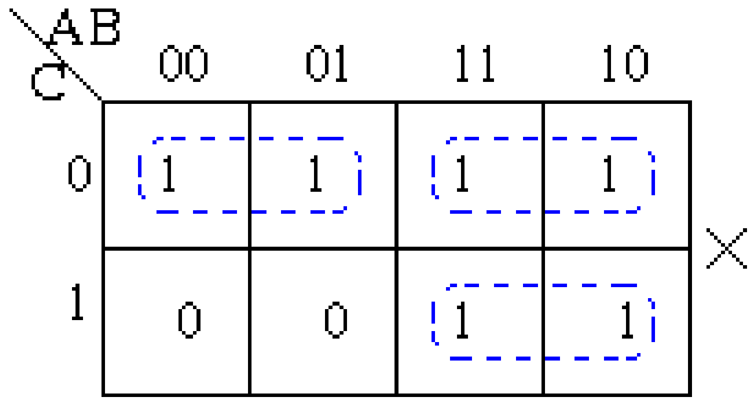


MAPAS DE KARNAUGH - MINITÉRMINOS

4. CADA GRUPO HA DE SER TAN GRANDE COMO SEA POSIBLE.



CORRECTO



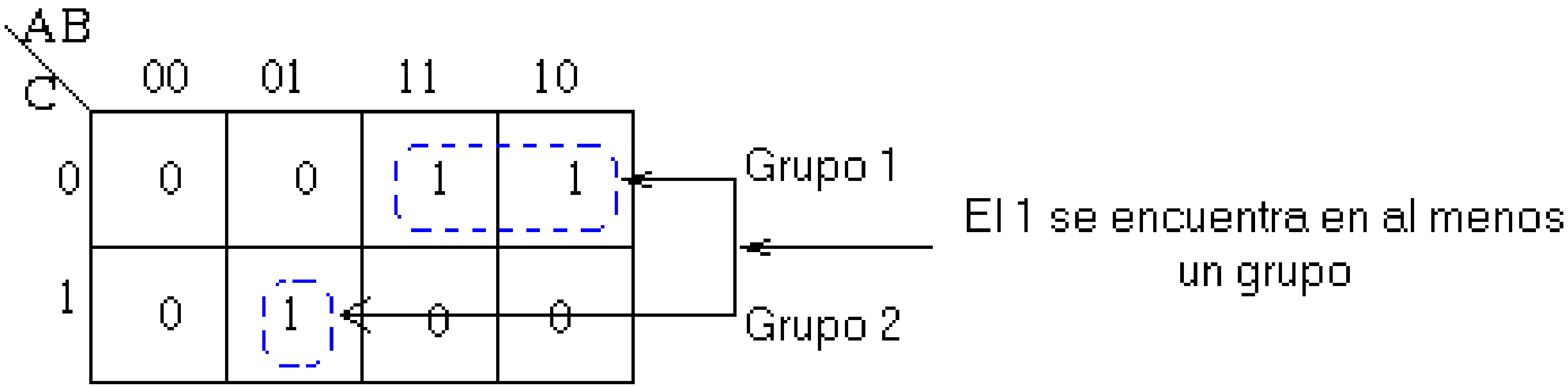
INCORRECTO

No se ha incumplido ninguna regla
pero el resultado no está optimizado

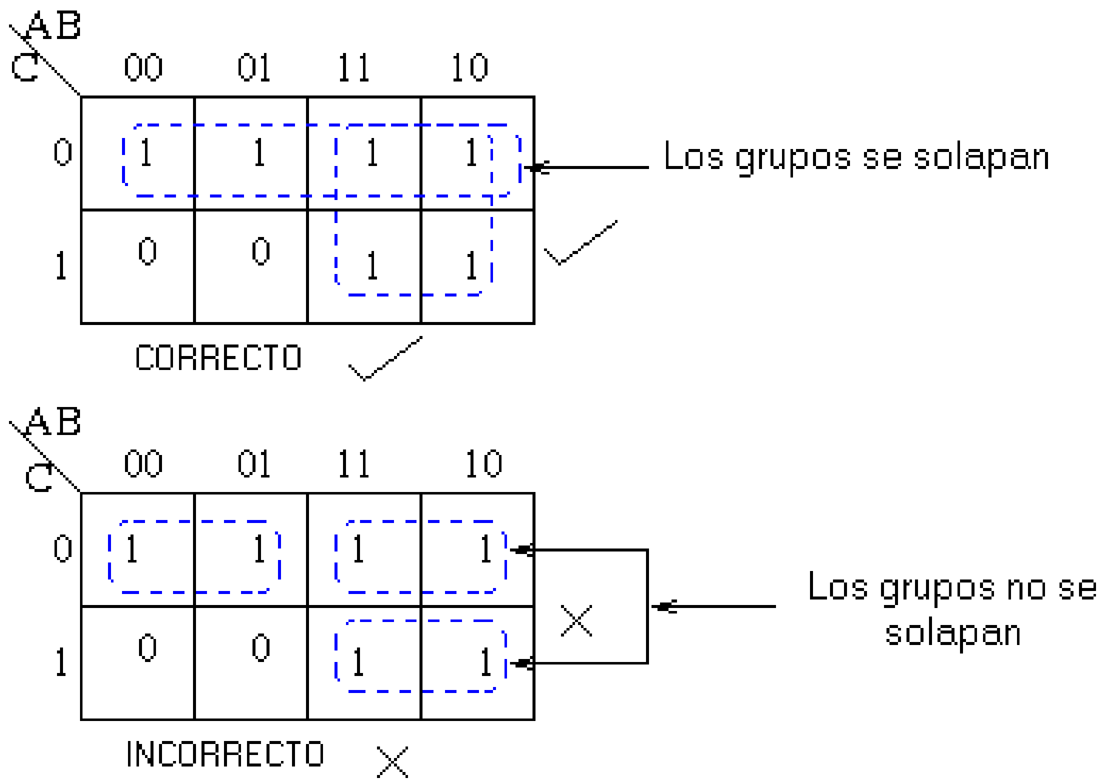
MAPAS DE KARNAUGH - MINITÉRMINOS

5. TODOS LOS UNOS TIENEN QUE PERTENECER COMO MÍNIMO A UN GRUPO.

Aunque pueden pertenecer a más de uno.

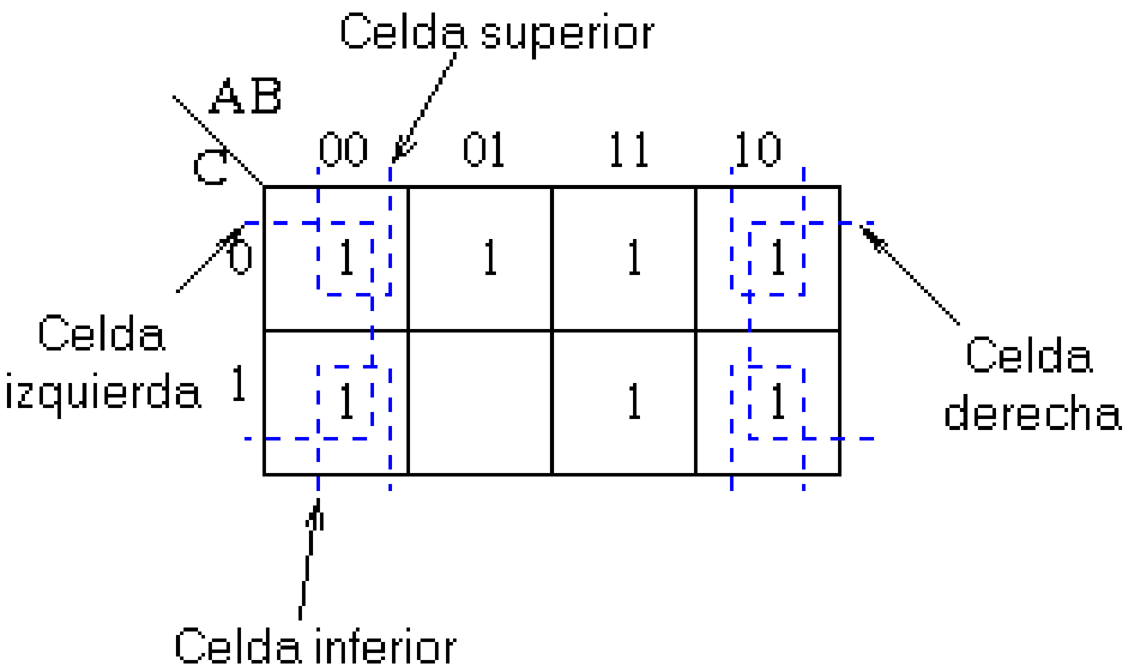


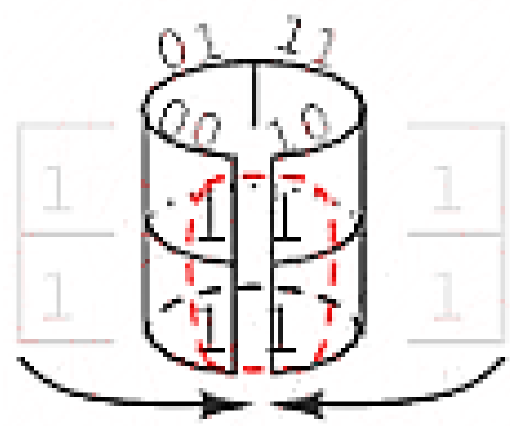
6. PUEDEN EXISTIR SOLAPAMIENTO DE GRUPOS.



MAPAS DE KARNAUGH - MINITÉRMINOS

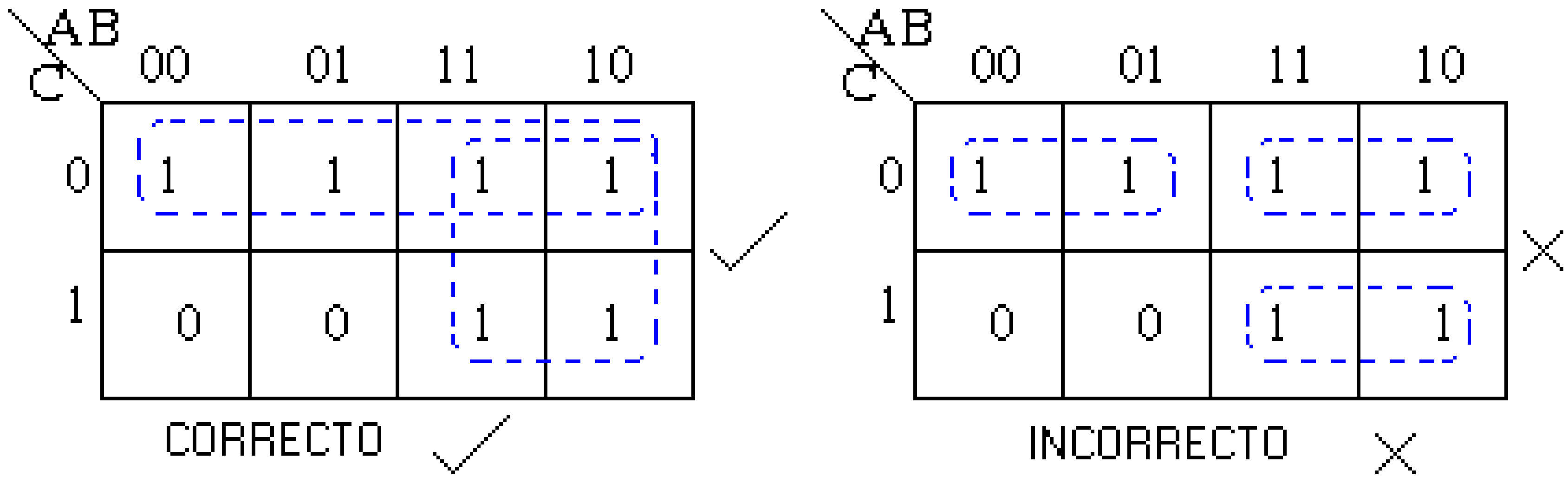
7. LA FORMACIÓN DE GRUPOS TAMBIÉN SE PUEDE PRODUCIR CON LAS CELDAS EXTREMAS DE LA TABLA.





MAPAS DE KARNAUGH - MINITÉRMINOS

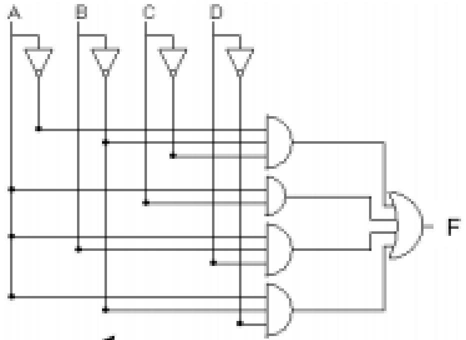
8. TIENE QUE RESULTAR EL MENOR NÚMERO DE GRUPOS POSIBLES SIEMPRE Y CUANDO NO CONTRADIGA NINGUNA DE LAS REGLAS ANTERIORES.



MAPA DE KARNAUGH - EJEMPLO

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

AB \ CD	00	01	11	10
00	1	1		
01				
11		1	1	1
10	1		1	1

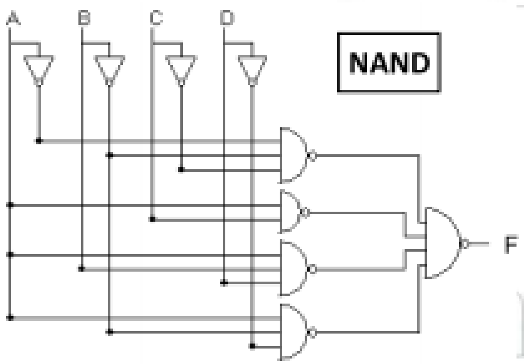


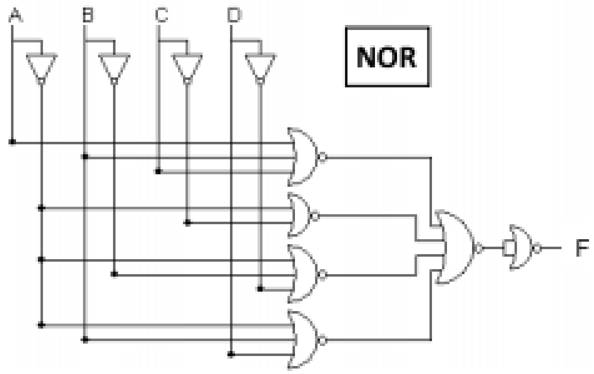
$$F(A,B,C,D) = \boxed{\overline{A}\,\overline{B}\,\overline{C} + ABD + A\overline{B}\,\overline{D} + AC} =$$

$$= \overline{\overline{\overline{A}\,\overline{B}\,\overline{C} + ABD + A\overline{B}\,\overline{D} + AC}} =$$

$$= \boxed{\overline{\overline{\overline{A}\,\overline{B}\,\overline{C}} \bullet \overline{ABD} \bullet \overline{A\overline{B}\,\overline{D}} \bullet \overline{AC}}} =$$

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





MAPA DE KARNAUGH – EJEMPLO COMPARACIÓN 1

AB	CD			
	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	0	1	1
10	1	0	0	0

Minitérminos: $F = AB'C'D' + ABC'D' + A'BCD' + A'BCD + ABCD' + ABCD$

Numero de compuertas usada: $6 + 5 + 5 + 4 + 4 + 3 + 5 = 27$

MAPA DE KARNAUGH – EJEMPLO COMPARACIÓN 1

AB	CD			
	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	0	1	1
10	1	0	0	0

Maxitérminos: $F = (A + C)(C + D')(A + B)(B + C')$

Numero de compuertas usada: $1 + 2 + 1 + 2 + 3 = 9$

MAPA DE KARNAUGH – EJEMPLO COMPARACIÓN 2

AB	CD			
	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	1	1	0	1
10	1	1	1	1

Maxitérminos: $F = AC' + AB' + BCD'$

Numero de compuertas usada: $2 + 2 + 3 + 2 = 9$

MAPA DE KARNAUGH – EJEMPLO COMPARACIÓN 2

AB	CD			
	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	1	1	0	1
10	1	1	1	1

Minitérminos: $F = (A + B)(A + C)(B' + C' + D')$

Numero de compuertas usada: $1 + 1 + 5 + 2 = 9$

MAPA DE KARNAUGH – EJEMPLO COMPARACIÓN CONCLUSIÓN

- En el ejemplo 1 se observa que la simplificación por maxitérminos utiliza más compuertas que la simplificación por minitérminos.
- En el ejemplo 2 es indiferente, ambas usan la misma cantidad de compuertas.
- Conclusión, que tipo de agrupación usar va a depender del problema en cuestión. Siempre hay que recordar que cuantas menos puertas lógicas se utilicen, más simple y económico será el sistema.

Enlace al simulador

- https://circuitverse.org/simulator/embed/ejemplos-06c807a0-da62-4873-8f94-9c8e278415da?theme=default&display_title=false&clock_time=true&fullscreen=true&zoom_in_out=true"