Tema 1. Álgebra de Boole y Diseño Lógico

1.0. Documentación

💡 Documentos Tema 1

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/e0561b0f-456c-4a42-9c9f-baacae303fa9/U1_AlgebraBoole_2020-21_(1).pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/53898030-f553-40da-8f59-e6cd8212840e/U1_BooleEnunciados.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9ebc00ca-a6d5-4838-8f20-de374c4c3156/U1_Boole_Soluciones.pdf

1.1. Señales analógicas y digitales

Tanto las señales analógicas como las digitales representan señales físicas. Una **señal analógica** es aquella que toma valores continuos, pudiendo acotarse y encontrando infinitos valores en dicha acotación. Una **señal digital** es aquella que toma valores discretos y se expresan mediante sentencias declarativas (se describen a sí mismas). Sus valores son excluyentes. Las señales digitales binarias toman valores entre dos posibles opciones (0/1, ON/OFF…).

I	B

En un circuito dado:

I	B
0	0
1	1

- Interruptor: abierto (0) o cerrado (1).
- Bombilla: apagada (0) o encendida (1).

1.2. Sistemas numéricos

1.2.1. Sistema decimal - binario

$5374 = 5 \cdot 10^3 + 3 \cdot 10^2 + 7 \cdot 10^1 + 4 \cdot 10^0$

$1101 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13$

$10101 = 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^0 = 21$

2^6	2^5	2^4	2^3	2^2
64	32	16	8	4
0	1	0	1	1

El **lsB (Less Significant Bit)** es el bit con menor peso, el situado más a la derecha, mientras que el **msB (More Significant Bit)** es el bit con mayor peso, el situado más a la izquierda.

1.2.2. Sistema hexadecimal

Hex	Dec	Bin	Hex	Dec
0	0	0000	**8**	8
1	1	0001	**9**	9
2	2	0010	**A**	10
3	3	0011	**B**	11
4	4	0100	**C**	12
5	5	0101	**D**	13
6	6	0110	**E**	14
7	7	0111	**F**	15

$4AF = 4 \cdot 16^2 + 10 \cdot 16^1 + 15 \cdot 16^0 = 1024 + 160 + 15 = 1199$

1.2.3. Unidades

Un **bit** es la unidad mínima dentro del sistema binario. Un **byte** es un conjunto de 8 bits, mientras que un **nibble** es la mitad de un byte, 4 bits.

1.3. Álgebra booleana

El **álgebra de Boole** es la herramienta matemática utilizada para el análisis y la síntesis de los sistemas digitales binarios.

Una **variable booleana** es una señal digital que, en un instante determinado, solo puede tomar una de dos valores. Los valores a tomar son mutuamente excluyentes.

1.3.1. Operaciones y expresiones booleanas

> Una **función lógica** es un circuito que acepta valores lógicos a la entrada y produce un valor lógico a la salida.

Las tablas de verdad describen el comportamiento de funciones lógicas. Son representaciones gráficas que especifican la salida de la función y las posibles combinaciones de entrada.

Lluvia	Amigos	Tareas	Dinero	Transporte
0	1	0	1	1
1	1	0	1	1
0	0	0	1	1

F = ¿Salir de fiesta el viernes?; 0 = No; 1 = Sí

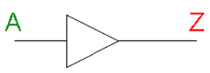
1.3.2. Puertas lógicas

EL AMPLIFICADOR (BUFFER)

- Puerta lógica más sencilla
- Una entrada (A) y una salida (Z)
- Tabla de verdad:

A	Z
1	1
0	0

- Ecuación lógica: **Z = A**
- Representación gráfica:

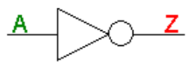


LA PUERTA NOT O INVERSOR

- Una entrada (A) y una salida (Z)
- Tabla de verdad:

A	Z
1	0
0	1

- Ecuación lógica: $Z = \overline{A}$
- Representación gráfica:

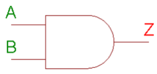


LA PUERTA AND (2 entradas {A,B} y 1 salida {Z})

- Z=1 sólo si las dos entradas están simultáneamente a 1
- Tabla de verdad:

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

- Ecuación lógica: $Z = A \bullet B$
- Representación gráfica:

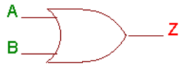


LA PUERTA OR: (2 entradas {A,B} y 1 salida {Z})

- Z = 1 cuando alguna de las dos entradas vale 1
- Tabla de verdad:

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

- Ecuación lógica: $Z = A + B$
- Representación gráfica:



LA PUERTA NAND: (2 entradas {A,B} y 1 salida {Z})

- Z=1 si al menos una de las dos entradas vale 0
- Tabla de verdad:

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

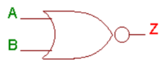
- Ecuación lógica: $Z = \overline{A \cdot B}$
- Representación gráfica:

LA PUERTA NOR: (2 entradas {A,B} y 1 salida {Z})

- Z=0 si al menos una de las dos entradas vale 1
- Tabla de verdad:

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

- Ecuación lógica: $Z = \overline{A + B}$
- Representación gráfica:

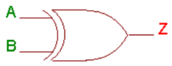


LA PUERTA XOR (OR-Exclusiva): (2 entradas {A,B} y 1 salida {Z})

- Z=1 si y sólo si una de las entradas está a 1
- Tabla de verdad:

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

- Ecuación lógica: $Z = A \oplus B$
- Representación gráfica:



Z=1 si el número de 1s en entrada es impar

LA PUERTA XNOR (NOR-Exclusiva): (2 entradas {A,B} y 1 salida {Z})

- Z=0 si sólo una de las entradas está a 0
- Tabla de verdad:

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

- Ecuación lógica: $Z = \overline{(A \oplus B)}$
- Representación gráfica:

Z=1 si el número de 1s en entrada es par

1.3.3. Leyes y reglas del álgebra de Boole

> Ley de la dualidad: cualquier expresión o identidad en un álgebra de Boole tiene su expresión dual intercambiando el + por la · y los 0 por los 1

Nombre		Teorema		Dual
Identidad	**T1**	B ● 1 = B	**T1’**	B + 0 = B
Elemento nulo	**T2**	B ● 0 = 0	**T2’**	B + 1 = 1
Idempotencia	**T3**	B ● B = B	**T3’**	B + B = B
Involución	**T4**	//B = B		
Complemento	**T5**	B ● /B = 0	**T5’**	B + /B = 1
Prop. Conmutativa	**T6**	B ● C = C ● B	**T6’**	B + C = C + B
Prop. Asociativa	**T7**	(B ● C) ● D = B ● (C ● D)	**T7’**	(B + C) + D = B + (C + D)
Prop. Distributiva	**T8**	(B ● C) + (B ● D) = B ● (C + D)	**T8’**	(B + C) ● (B + D) = B + (C ● D)
Ley de De Morgan	**T12**	$/(B_0 \bullet B_1 \bullet \ldots \bullet B_{n-2} \bullet B_{n-1}) = (/B_0 + /B_1 + \ldots + /B_{n-2} + /B_{n-1})$	**T12’**	$/(B_0 + B_1 + \ldots + B_{n-2} + B_{n-1}) = (/B_0 \bullet /B_1 \bullet \ldots \bullet /B_{n-2} \bullet /B_{n-1})$

T1. Al multiplicar una variable por 1 o sumarle 0, su valor no cambia.

T2. Al multiplicar una variable por 0 su valor siempre será 0, mientras que al sumarle 1 su valor siempre será 1.

T3. Al multiplicar o sumar una variable por si misma, el resultado es la misma variable.

T4. Al invertir una variable dos veces esta no cambia.

T5. Al multiplicar una variable por su complementada su valor será 0, mientras que al sumarlas su valor será 1.

T6. Al multiplicar o sumar dos variables el orden es irrelevante para su valor final.

T7. Al multiplicar o sumar tres o más variables su agrupación es irrelevante para su valor final.

T8. Al sumar multiplicaciones de variables con alguna en común se puede operar como en el álgebra común, mientras que al multiplicar sumas de variables con alguna en común se puede operar como una multiplicación común.

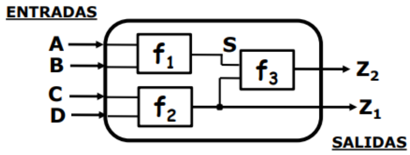
T12. Al complementar la multiplicación de variables, su valor es la suma de las variables complementadas, mientras que al complementar la suma de variables, su valor es la multiplicación de las variables complementadas.

x	y	/(x·y)	x		y		/(x+y)
0	0	1	0	1	0		1
0	1	1	0	1	1		0
1	0	1	1	1	0		0

x	y	/(x·y)	x		y		/(x+y)
1	1	0	1	0	1		0

1.4. Circuitos lógicos

> Los **circuitos lógicos** son las combinaciones de diferentes valores lógicos a la entrada, que hacen que aparezcan distintos valores lógicos a la salida.



- **Lógica combinacional**: el estado de las salidas depende solo del estado de las entradas. Es un sistema sin memoria.
- **Lógica secuencial**: el estado de la salida también depende del estado anterior del sistema. Es un sistema con memoria.

1.4.1. Funciones lógicas

> Una **función lógica** es una expresión booleana que relaciona variables lógicas directas o complementarias a través de operaciones AND y OR.

> Un **minterm** es un producto (AND) de las variables y sus complementos. La función se construye por la suma (OR) de los minterm con salida verdadera.

> Un **maxterm** es una suma (OR) de las variables y sus complementos. La función se construye por el producto (AND) de los maxterm con salida falsa.

A	B	Y	minterm
0	0	0	$\overline{A}\ \overline{B}$
0	1	1	$\overline{A}\ B$
1	0	0	$A\ \overline{B}$
1	1	1	$A\ B$

Y = F(A, B) = /AB + AB

A	B	Y	maxterm
0	0	0	$A + B$
0	1	1	$A + \overline{B}$
1	0	0	$\overline{A} + B$
1	1	1	$\overline{A} + \overline{B}$

Y= F(A, B) = (A+B)(/A+B)

1.4.2. Diagrama de Karnaugh (mapas k)

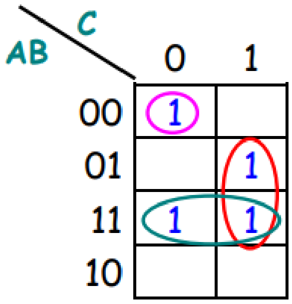
A\B	0	AB\C	0	AB\CD	00		0
0	00	00	000	00	0000		0
1	10	01	010	01	0100		0
		11	110	11	1100		1
		10	100	10	1000		1

BC\DE	00	01	11	10
00	00000	00001	00011	00010
01	00100	00101	00111	00110
11	01100	01101	01111	01110
10	01000	01001	01011	01010
	A=0	A=0	A=0	A=0

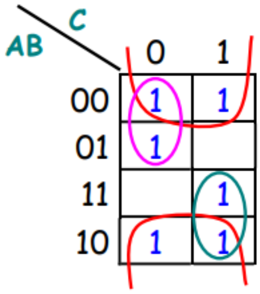
Dos celdas son **adyacentes** si comparten un lado (su distancia Hamming es 1). A su vez, la columna de la izquierda y la derecha son adyacentes, al igual que las filas de arriba y abajo. Entre dos celdas adyacentes solo puede variar un único bit.

La **minimización** es el proceso que genera una expresión que contiene el menor número de variables posibles.

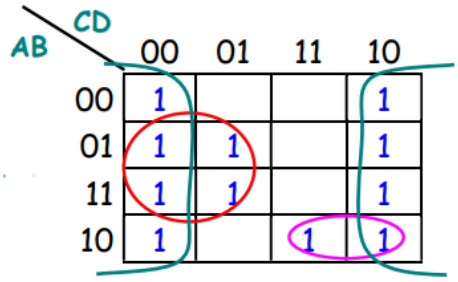
Dado un diagrama de Karnaugh, para minimizar la suma de productos, la agrupación de 1's consiste en buscar grupos de celdas adyacentes con el valor 1. Estos grupos tienen que contener 2^n celdas. Una misma celda puede pertenecer a más de un grupo. Posteriormente, se obtienen las expresiones de cada uno de los grupos y se suman para obtener la expresión final.



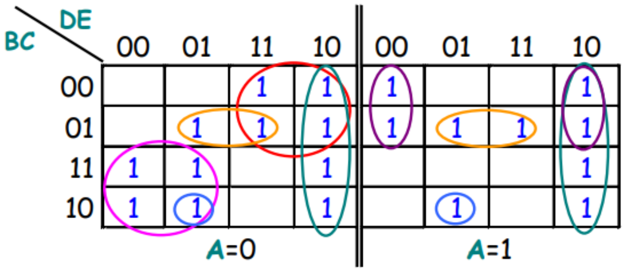
$\overline{A}\,\overline{B}\,\overline{C} + AB + BC$

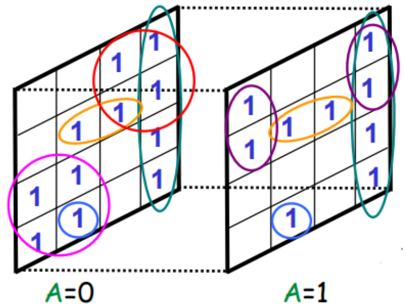


$\overline{A}\,\overline{C} + AC + \overline{B}$



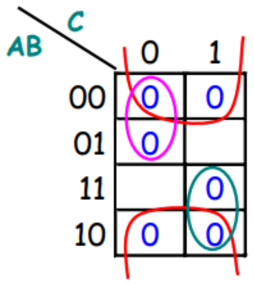
$A\overline{B}C + B\overline{C} + \overline{D}$



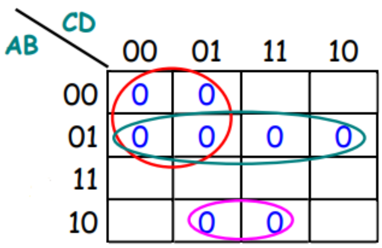


$\overline{A}B\overline{D} + \overline{A}\,\overline{B}D + A\overline{B}\,\overline{E} + D\overline{E} + \overline{B}CE + B\overline{C}\,\overline{D}E$

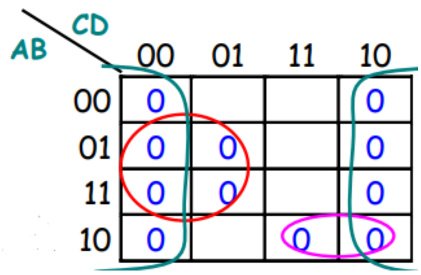
Dado un diagrama de Karnaugh, para minimizar el producto de suma, la agrupación de 0’s sigue las mismas normas que la agrupación de 1’s, cambiando la suma por productos y viceversa.



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

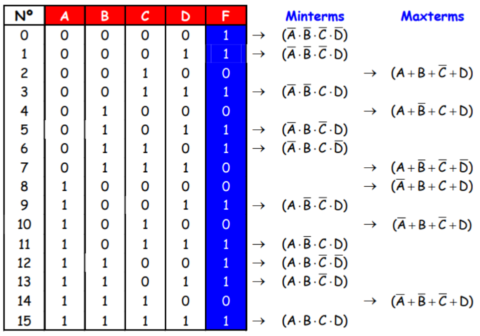


$(\overline{A}+B+\overline{D})\cdot(A+\overline{B})\cdot(A+C)$

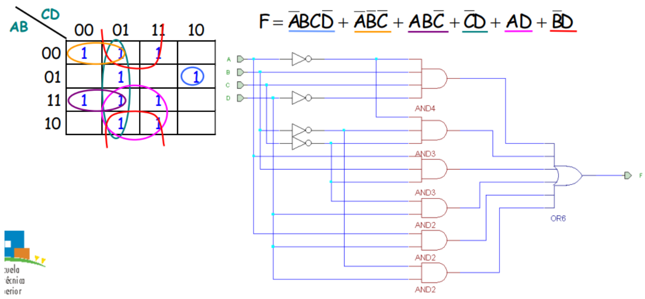


$(\overline{A}+B+\overline{C})\cdot(\overline{B}+C)\cdot D$

El diagrama de Karnaugh también se puede obtener a través de una tabla de verdad o de un circuito electrónico.



Nº	A	B	C	D	F	Minterms	Maxterms
0	0	0	0	0	1	→ $(\bar{A}\cdot\bar{B}\cdot\bar{C}\cdot\bar{D})$	
1	0	0	0	1	1	→ $(\bar{A}\cdot\bar{B}\cdot\bar{C}\cdot D)$	
2	0	0	1	0	0		→ $(A+B+\bar{C}+D)$
3	0	0	1	1	1	→ $(\bar{A}\cdot\bar{B}\cdot C\cdot D)$	
4	0	1	0	0	0		→ $(A+\bar{B}+C+D)$
5	0	1	0	1	1	→ $(\bar{A}\cdot B\cdot\bar{C}\cdot D)$	
6	0	1	1	0	1	→ $(\bar{A}\cdot B\cdot C\cdot\bar{D})$	
7	0	1	1	1	0		→ $(A+\bar{B}+\bar{C}+\bar{D})$
8	1	0	0	0	0		→ $(\bar{A}+B+C+D)$
9	1	0	0	1	1	→ $(A\cdot\bar{B}\cdot\bar{C}\cdot D)$	
10	1	0	1	0	0		→ $(\bar{A}+B+\bar{C}+D)$
11	1	0	1	1	1	→ $(A\cdot\bar{B}\cdot C\cdot D)$	
12	1	1	0	0	1	→ $(A\cdot B\cdot\bar{C}\cdot\bar{D})$	
13	1	1	0	1	1	→ $(A\cdot B\cdot\bar{C}\cdot D)$	
14	1	1	1	0	0		→ $(\bar{A}+\bar{B}+\bar{C}+D)$
15	1	1	1	1	1	→ $(A\cdot B\cdot C\cdot D)$	



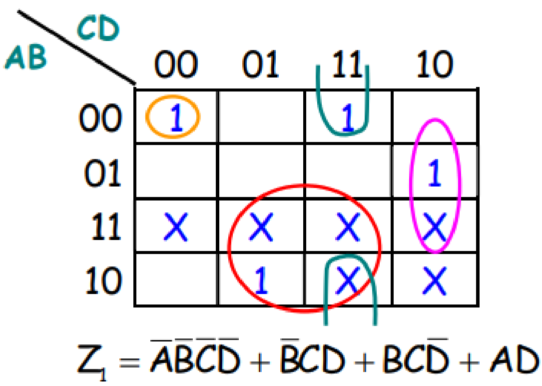
AB\CD	00	01	AB\CD	00		01
00	1	0	00	1	0	0
01	0	0	01	1	1	0
11	1	0	11	1	0	0
10	0	1	10	1	0	0

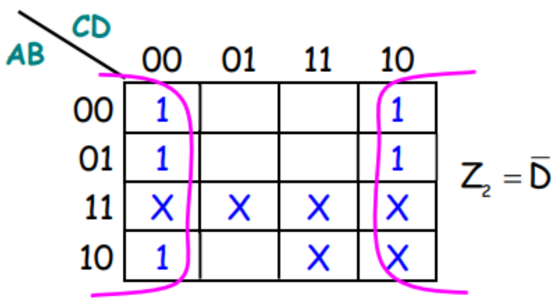
$\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}BC\bar{D} +$
$AB\bar{C}\bar{D} + ABCD + A\bar{B}\bar{C}D$

$\bar{D}$

1.4.3. Funciones incompletamente especificadas

Cuando hay valores reales no considerados por la función, el resultado puede ser indeterminado, x, (ni 0 ni 1). A la hora de minimizar la expresión, estas x pueden ser tratadas como 0 o 1 indistintamente.





1.4.4. Convertir cualquier circuito a puertas NAND

Dada la función $F = A\bar{B} + BC + \bar{A}C$. Primero se niega la función dos veces: $F = \overline{\overline{A\bar{B} + BC + \bar{A}C}}$; y después se aplica la ley de De Morgan: $F =$

$\overline{\overline{A\bar{B}} \cdot \overline{BC} \cdot \overline{\bar{A}C}}$. Por último, se representa la función lógica como un circuito utilizando únicamente puertas NAND.

Una expresión del tipo $\overline{X \cdot Y}$ se representa como una puerta NAND, mientras que una expresión del tipo $\bar{Z}$ se representa como una puerta NAND con una única entrada, funcionando como un inversor.