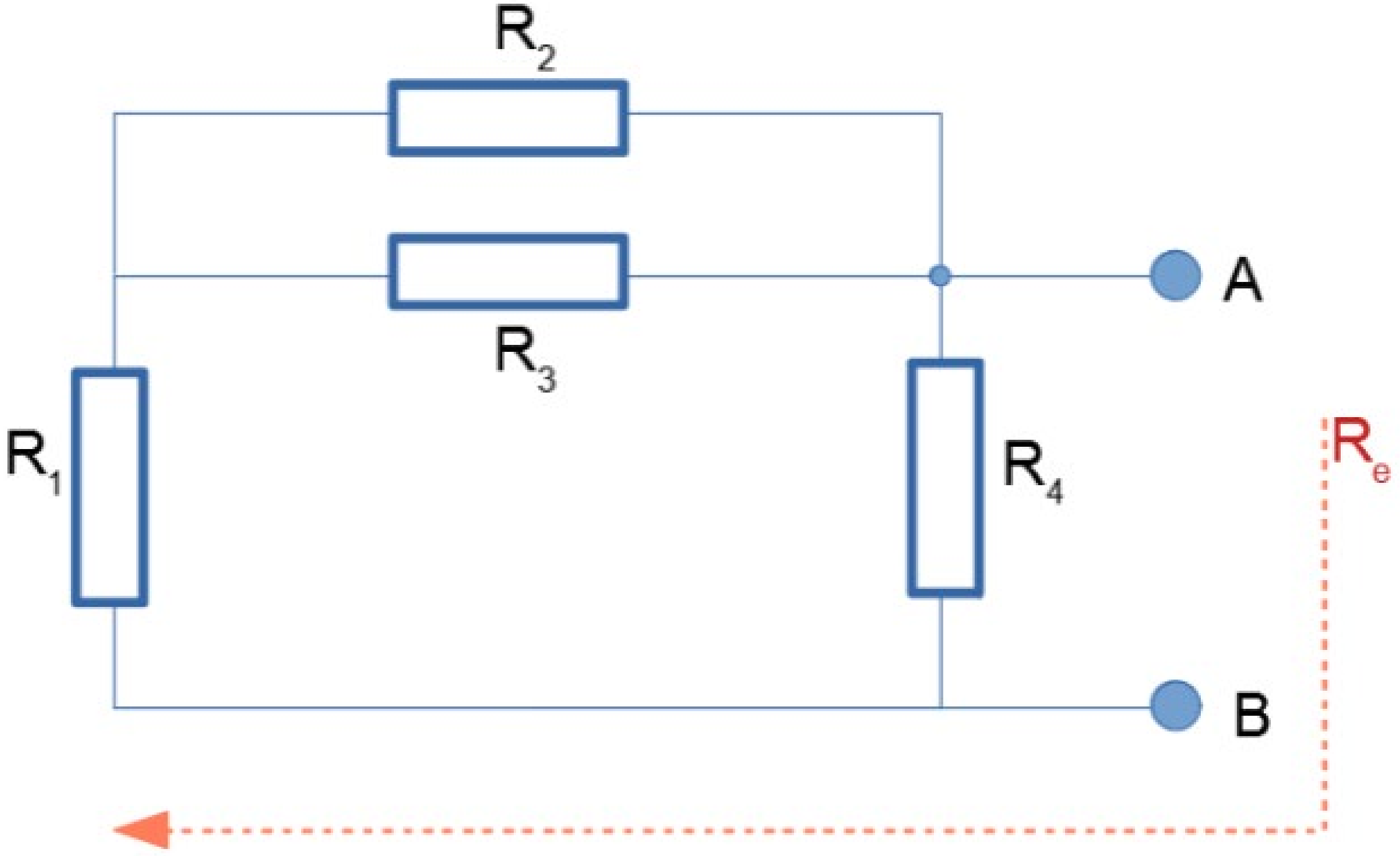
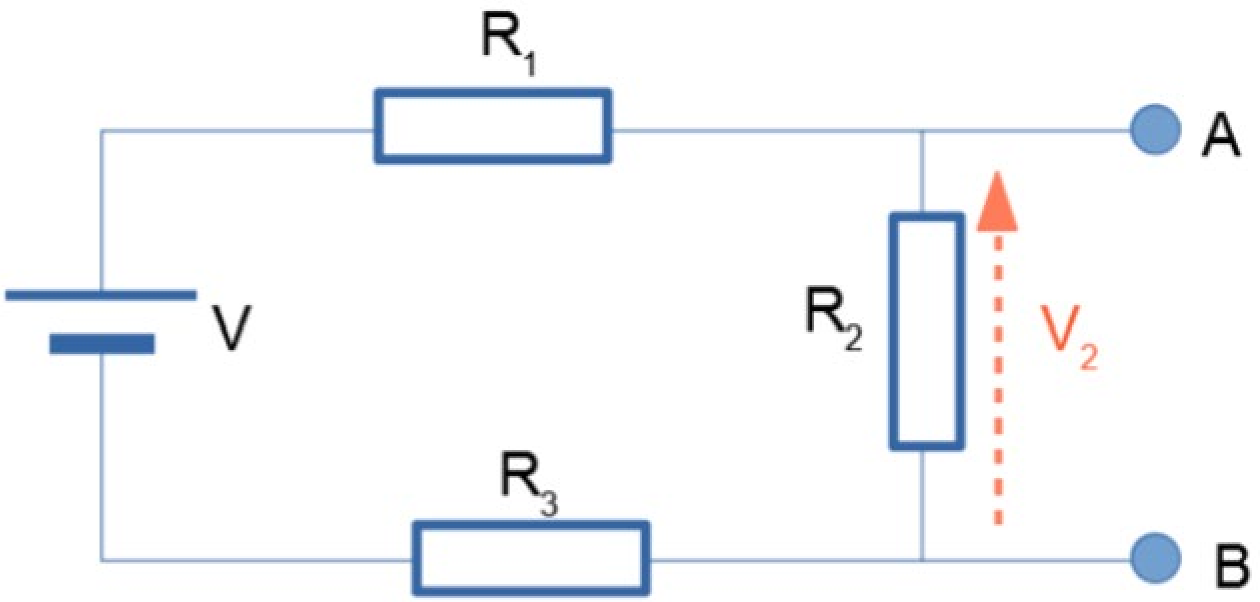
Consideraciones de L2



R_2
R_3
R_1
R_4
A
B
R_e



El objetivo en esta sección será calcular nominalmente el valor de V2 y, luego, simular el circuito para verificarlo.

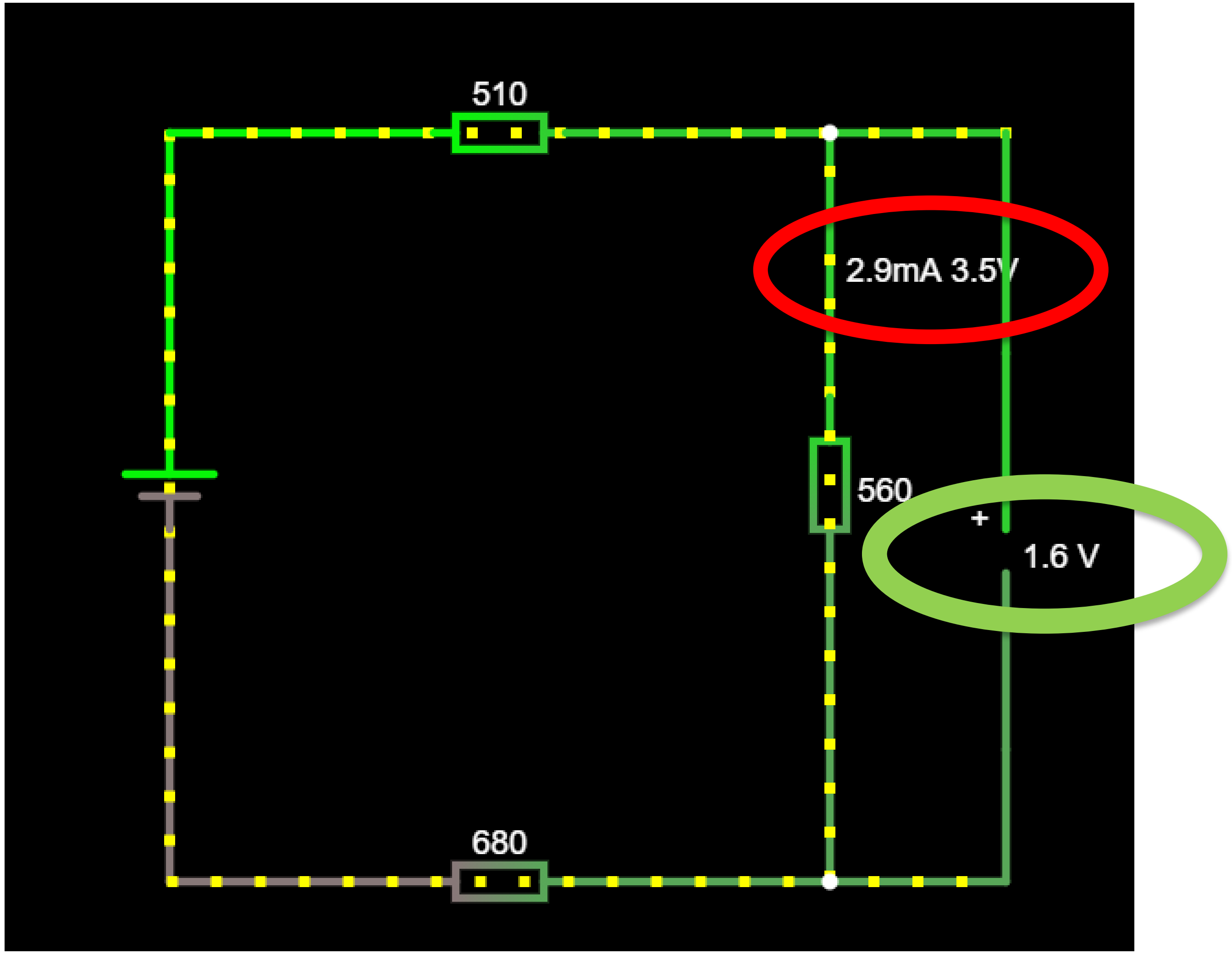
Empecemos:

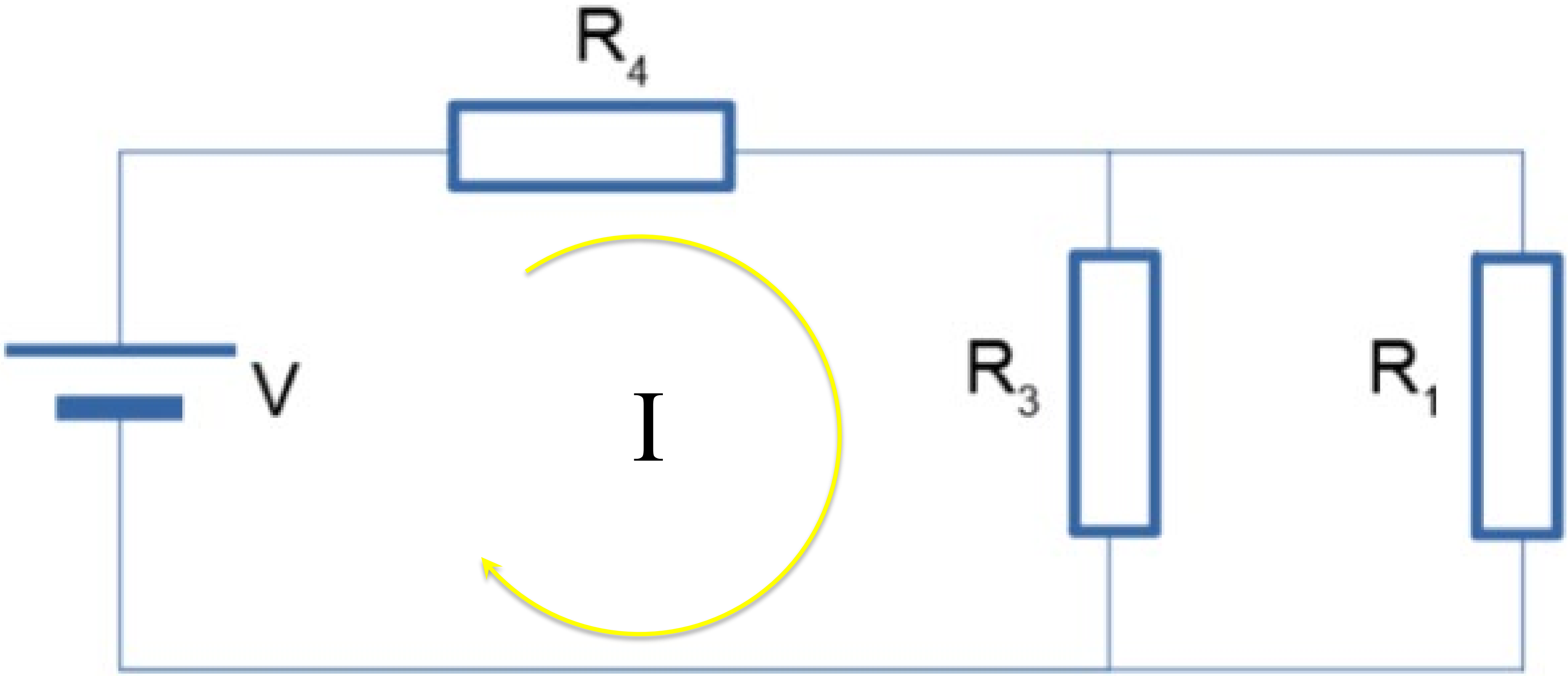
- Anota en el folio: **3. Y una fuente de continua.**
- Con los mismos valores nominales resistencias (e identificaciones) del apartado anterior, calcula V_2 asumiendo que V = 5V.
- Ahora, realiza el montaje en el simulador
- Comprueba que tu cálculo es correcto
- Exporta el circuito como "link", crea una URL corta, y escríbela en tu hoja de resultados.

Ya que estamos aquí:

- Calcula la corriente I2 que circula por la resistencia R2
- Usa un amperímetro en el simulador para comprobar los cálculos

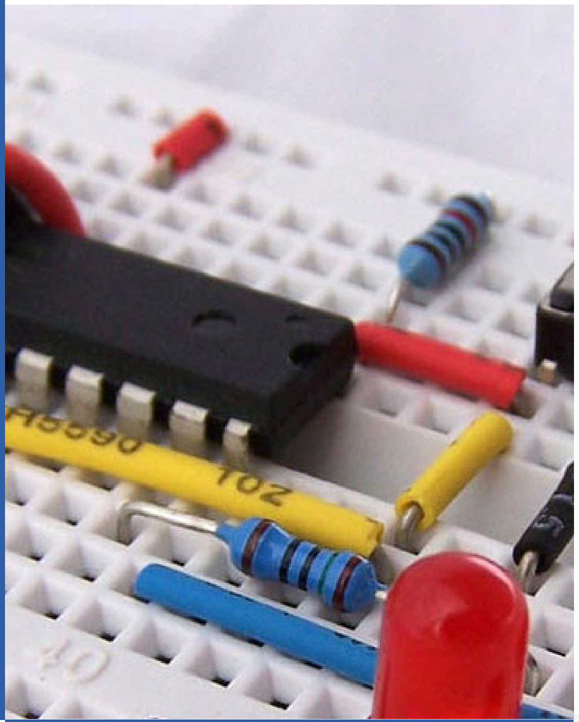

DIVISOR DE CORRIENTE


510
2.9mA 3.5V
560
+
1.6 V
680

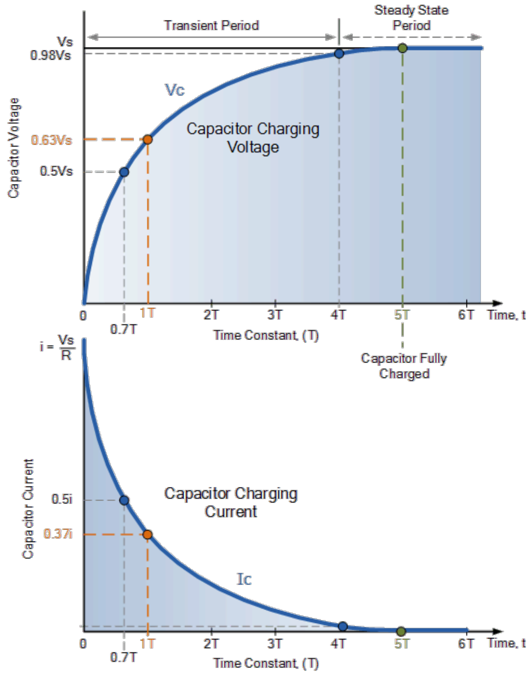


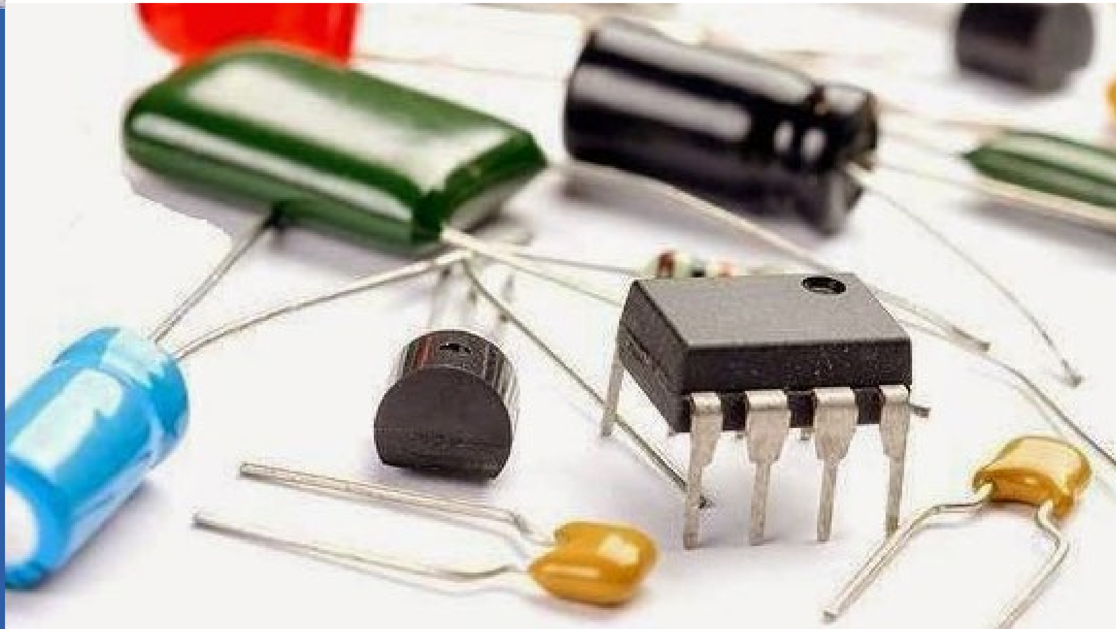
R4
V
I
R3
R1

Electrónica y Tecnología de Computadores



TEMA 1
Sistemas Combinacionales





Francisco.sanchez@ufv.es

Contenidos del tema

- Definiciones previas
- Señales binarias, niveles lógicos y formas de onda digitales
- Especificación de componentes combinacionales
  - Funciones de conmutación y tablas de verdad
  - Expresiones de conmutación
  - Álgebra de Boole
  - Simplificación de expresiones de conmutación : Boole y Karnaugh
- Implementación de circuitos combinacionales. SSI y MSI
- Análisis de circuitos combinacionales
- Otros circuitos combinacionales

Bibliografía básica para este tema

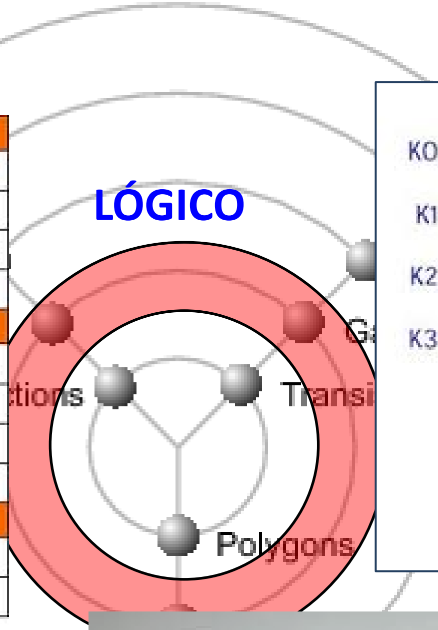
- Fundamentos de Computadores, Ramón Hermida
- Principios del diseño digital, Daniel D. Gajski
- Diseño digital, M. Morris Mano
- Fundamentos de sistemas digitales, Thomas L. Floyd

Niveles y dominios del tema

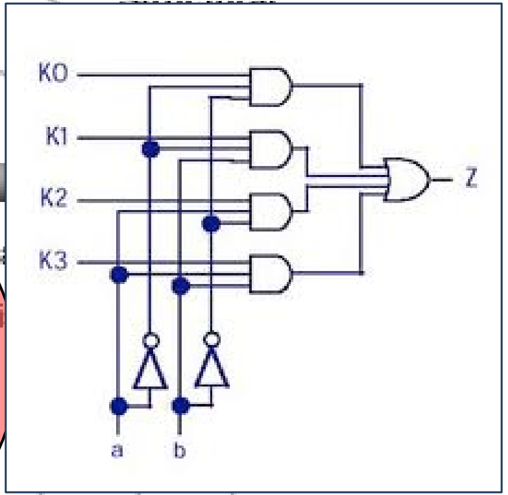
Comportamiento

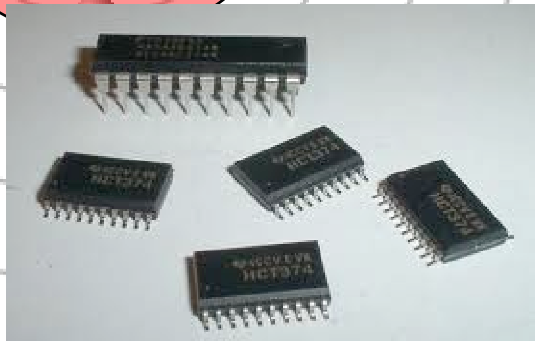
La operación AND o Y	
0 • 0 = 0	0 • 0 = 0
0 • 1 = 0	0 • A = A
1 • 0 = 0	A • 0 = 0
1 • 1 = 1	A • A = A
La operación OR o O	
0 + 0 = 0	A + 0 = A
0 + 1 = 1	A + 1 = 1
1 + 0 = 1	A + A = A
1 + 1 = 1	A + A = 1
La operación NOT o No	
$\overline{0} = 1$	A´´ = A
$\overline{1} = 0$	Nota: A´ = $\overline{A}$

Algebra de Boole
Ecuaciones booleanas



Estructural



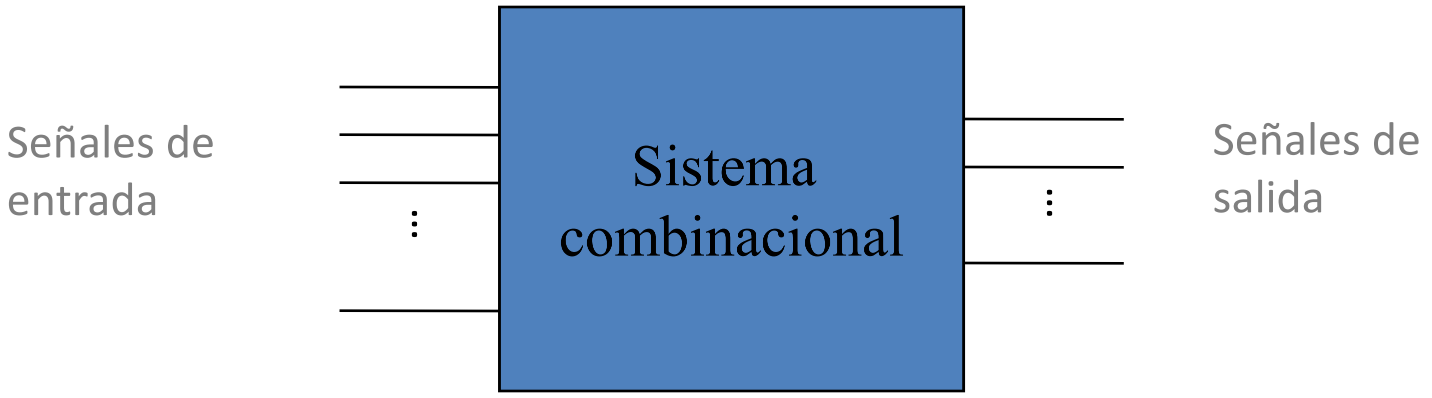


Geométrico

Definiciones previas

Sistema combinacional

Sistema digital binario en el que el valor de sus señales de salida en un momento dado depende sólo de los valores de sus señales de entrada



Definiciones previas

Magnitudes

- **Analógica**: aquella que toma valores en un rango continuo
- **Digital**: aquella que toma valores en un rango discreto
- **Digital binaria**: aquella que sólo puede tomar dos valores distintos (‘0’, ‘1’, es lo que llamamos bit)

Sistema electrónicos

- **Analógico:** sistema formado por circuitos electrónicos cuyas señales son analógicas
- **Digital**: sistema formado por circuitos cuyas señales son digitales

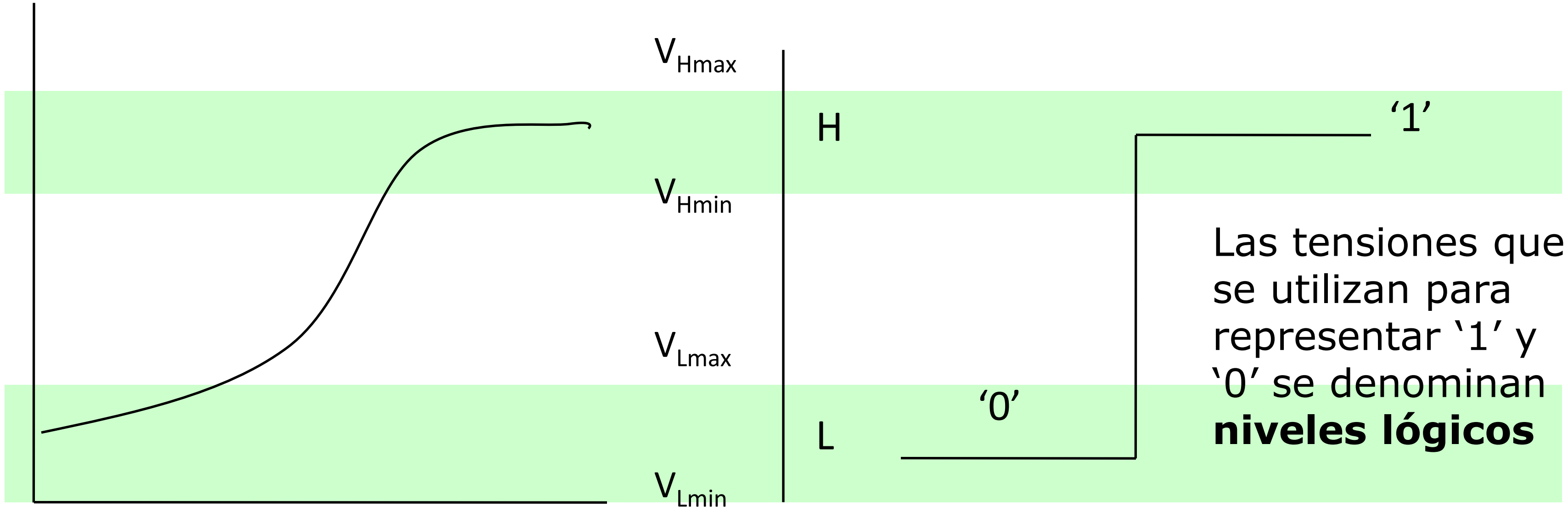
Señal: variación en una determinada magnitud (corriente eléctrica, tensión,…) a lo largo del tiempo

Señales binarias, niveles lógicos y formas de onda digitales

- ¿Cómo puede una señal ser binaria?

Las señales de un circuito electrónico **siempre son analógicas**. Pero la respuesta del circuito sólo se produce para dos rangos de valores de esa señal.

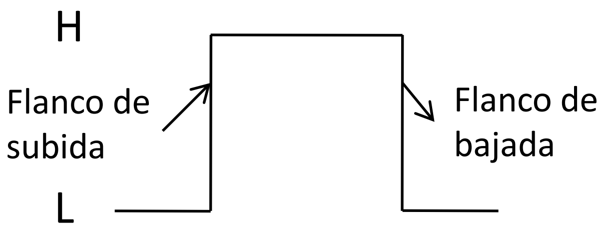
- En concreto, las señales de los circuitos electrónicos digitales representan los valores de tensión en cada instante de tiempo en un punto del circuito. Los circuitos sólo reaccionan ante dos estados de esa señal: alto o bajo.
- Un valor alto (H) de la tensión se corresponde normalmente con un ‘1’ lógico, mientras que un valor bajo (L) se corresponde con un ‘0’ lógico



Las tensiones que se utilizan para representar ‘1’ y ‘0’ se denominan **niveles lógicos**

Dígitos binarios, niveles lógicos y formas de onda digitales

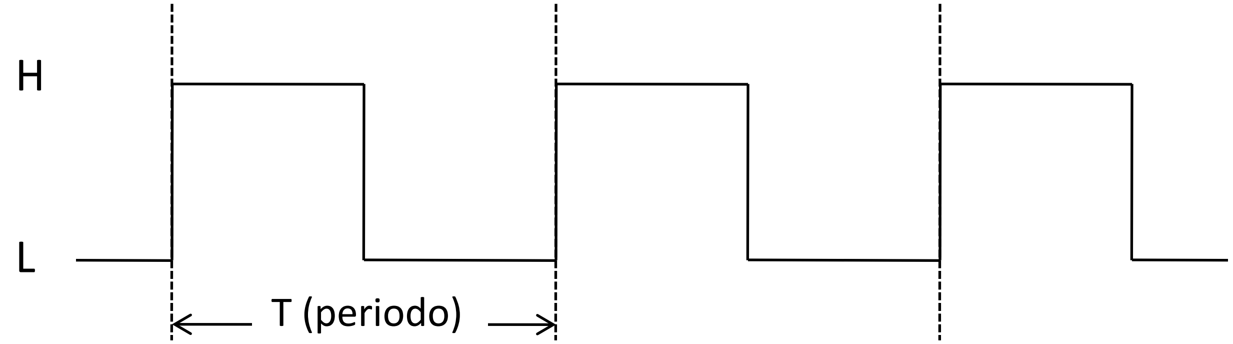
- Por lo tanto, una señal en un circuito digital binario consiste en un nivel de tensión que varía entre los estados alto (H) y bajo (L).
- Una señal binaria está compuesta por una serie de impulsos (a veces llamados pulsos)
  - Positivos: cuando pasa del estado L al H y luego al L de nuevo



  - Negativos: cuando pasa del H al L y luego al H de nuevo

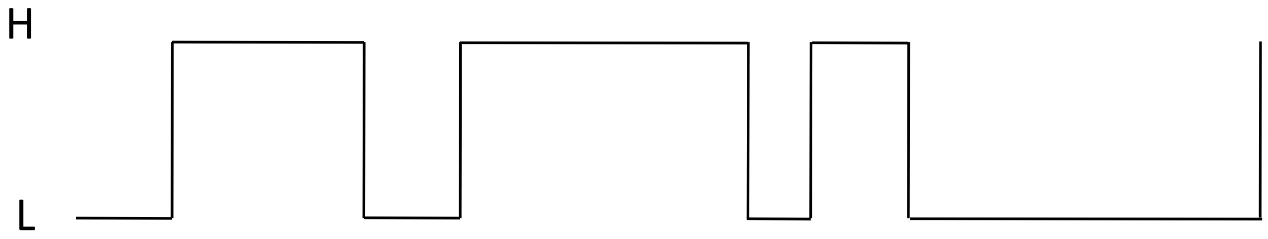
Dígitos binarios, niveles lógicos y formas de onda digitales

- Las señales digitales pueden ser
  - Periódicas: están formadas por un tren de impulsos que se repiten a intervalos fijos de tiempo



F (frecuencia) = 1/T (se mide en hercios o “hertz”, Hz)

  - No periódicas



Especificación de componentes combinacionales

- Especificación: descripción del comportamiento del componente, sin precisar cómo está construido
- Especificación mediante **funciones de conmutación**

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

  - Función de conmutación: describe el valor que toma cada señal de salida para cada posible configuración de los valores de entrada.
  - Pueden definirse mediante tablas de verdad.

Dominio x_2 x_1 x_0	Decimal Equivalente	Rango $f(x_2,x_1,x_0)$
0 0 0	0	0
0 0 1	1	0
0 1 0	2	1
0 1 1	3	1
1 0 0	4	1
1 0 1	5	1
1 1 0	6	1
1 1 1	7	0

Definición de funciones de conmutación mediante tablas de verdad

- Funciones de conmutación de 1 variable

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

- Funciones de conmutación de 2 variables

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}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Definición de funciones de conmutación mediante tablas de verdad

- Funciones incompletamente especificadas: el valor de la función es irrelevante en ciertos puntos del dominio

x_2	x_1	x_0	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	d
1	1	0	0
1	1	1	d

Definición de funciones de conmutación mediante expresiones de conmutación

- **Álgebra de conmutación:** Es el conjunto de los elementos {0,1} con la estructura inducida por las operaciones cerradas suma lógica, producto lógico y negación.
  - **Suma lógica (+):** se define como

$$0+0 = 0$$
$$0+1 = 1$$
$$1+0 = 1$$
$$1+1 = 1$$

OR

  - **Producto lógico(·):** se define como

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

AND

  - **Negación lógica (¯):** se define como

$$\overline{0} = 1$$
$$\overline{1} = 0$$

NOT

Definición de funciones de conmutación mediante expresiones de conmutación

- Las expresiones de conmutación son otra forma de definir funciones de conmutación alternativa a las tablas de verdad.
  - Para toda FC existen una o varias EC equivalentes.
  - Para toda EC, es posible calcular la tabla de verdad de la correspondiente FC simplemente calculando el valor de la EC para todos los posibles valores de sus variables.

Ejemplo: tabla de verdad de la FC que representa una EC

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

x2	x1	x0	FC
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Definición de funciones de conmutación mediante expresiones de conmutación

- Una **expresion de conmutación** (EC) es una cadena de texto en la que pueden aparecer:
  - Símbolos para denotar variables binarias
  - Los operadores "+", "·" y "¯"
  - Paréntesis

  La cadena se construye de manera recursiva:
  - **Las constantes 0 y 1 son EC**
  - **Una variable binaria es una EC**
  - **Si *A* es una EC ⇒ $\overline{A}$ *es EC***
  - **Si *A* y *B* son EC ⇒ *A*+*B* y *A*·*B* también son EC**
  - Número finito de aplicaciones de las reglas anteriores siempre produce una EC
- Ejemplo

$$EC(x_2, x_1, x_0) = x_2 + \overline{x}_1 \cdot x_0$$

$$EC(x_3, x_2, x_1, x_0) = x_2 \cdot (\overline{x}_3 + x_1 \cdot \overline{x}_0)$$

Definición de funciones de conmutación mediante expresiones de conmutación

- **Valor de la *EC*** en el punto *X* de su dominio se calcula sustituyendo cada variable en *EC* por el valor que toma en dicho punto y aplicando las operaciones correspondientes a esos valores

Ejemplo:

$$EC(x_2, x_1, x_0) = \overline{x}_2 + x_2 \cdot \overline{x}_1 + x_1 \cdot x_0$$

Valor de la EC en (0,0,1) = $\overline{0} + 0 \cdot \overline{0} + 0 \cdot 1 = 1$

Valor de la EC en (1,1,0) = $\overline{1} + 1 \cdot \overline{1} + 1 \cdot 0 = 0$

Definición de funciones de conmutación mediante expresiones de conmutación

- **Valor de la *EC*** en el punto *X* de su dominio se calcula sustituyendo cada variable en *EC* por el valor que toma en dicho punto y aplicando las operaciones correspondientes a esos valores

  Ejemplo:

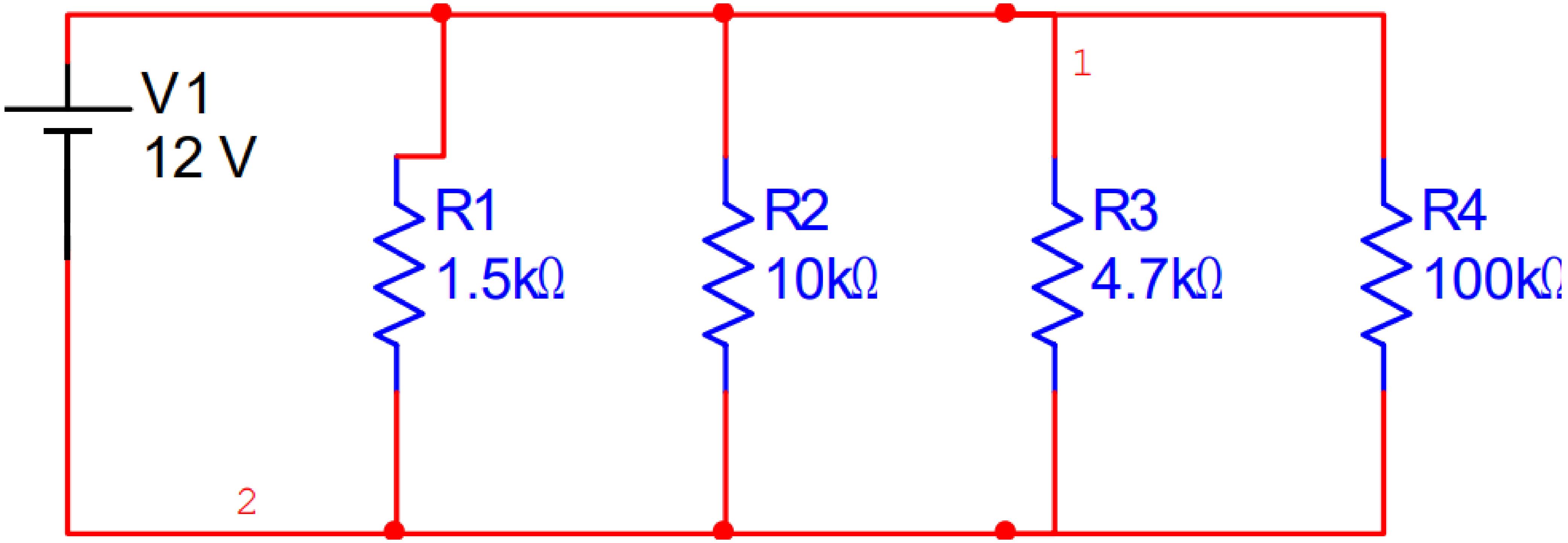
$$EC(x_2, x_1, x_0) = \overline{x}_2 + x_2 \cdot \overline{x}_1 + x_1 \cdot x_0$$

Valor de la EC en $(0,0,1) = \overline{0} + 0 \cdot \overline{0} + 0 \cdot 1 =$

Valor de la EC en $(1,1,0) = \overline{1} + 1 \cdot \overline{1} + 1 \cdot 0 =$

Ejercicio 1

Encontrar la corriente que circula por el circuito mostrado, suponiendo que se tiene una fuente de 12V.



solución 1

Paso 1: En un circuito en paralelo el voltaje se mantiene constante entre cada división o rama, por lo que a partir del voltaje y resistencia se puede calcular la corriente que circula por cada rama mediante la ley de ohm.

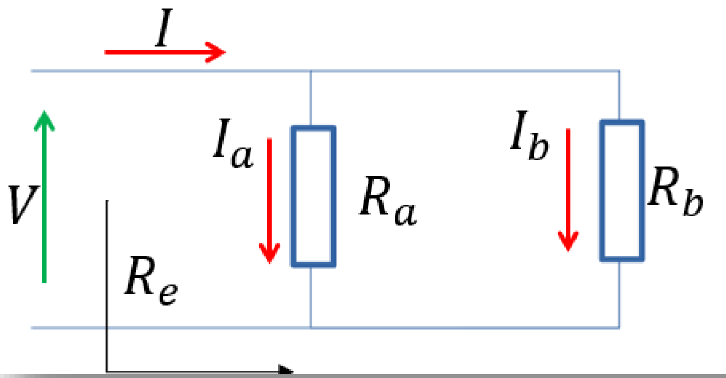
$$I_1 = \frac{V}{R1} = \frac{12V}{1.5k\Omega} = 8mA$$

$$I_2 = \frac{V}{R2} = \frac{12V}{10k\Omega} = 1.2mA$$

$$I_3 = \frac{V}{R3} = \frac{12V}{4.7k\Omega} = 2.55mA$$

$$I_4 = \frac{V}{R4} = \frac{12V}{100k\Omega} = 0.12mA$$

Cuando dos resistencias se encuentran en paralelo la tensión a la que están sometidas es
y, por tanto, la resistencia equivalente es:



$$\boxed{I = I_a + I_b}$$

$$\frac{V}{R_e} = \frac{V}{R_a} + \frac{V}{R_b}$$

$$\frac{1}{R_e} = \frac{1}{R_a} + \frac{1}{R_b}$$

Paso 2: Puesto que la corriente total es la suma de las corrientes individuales obtenemos la corriente que circula en el circuito:

$$Itotal = I_1 + I_2 + I_3 + I_4$$

$$Itotal = 8mA + 1.2mA + 2.55mA + 0.12mA$$

$$Itotal = 11.87mA$$

solución 2

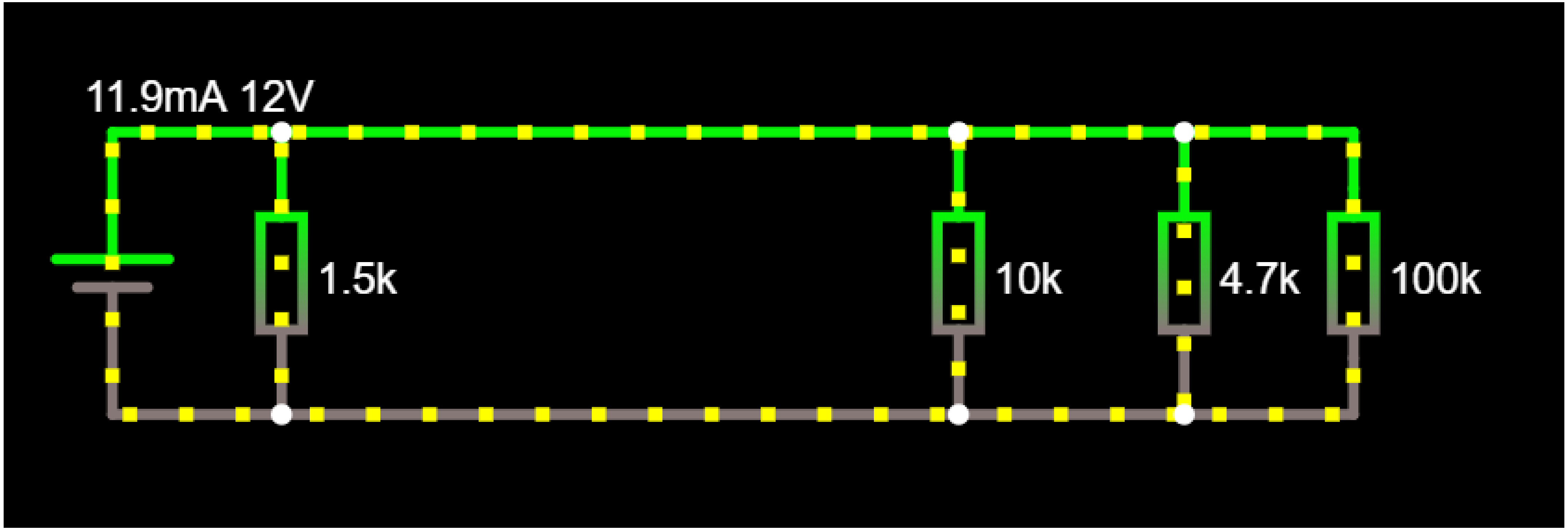
Paso 1: utilizando la suma de recíprocos calculamos la resistencia total.

$$Rtotal = \frac{1}{\frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3}+\frac{1}{R4}}$$

$$Rtotal = \frac{1}{\frac{1}{1.5k\Omega}+\frac{1}{10k\Omega}+\frac{1}{4.7k\Omega}+\frac{1}{100k\Omega}} = 1.01k\Omega$$

Paso 2: Ahora utilizando la ley de Ohm calculamos la corriente total.

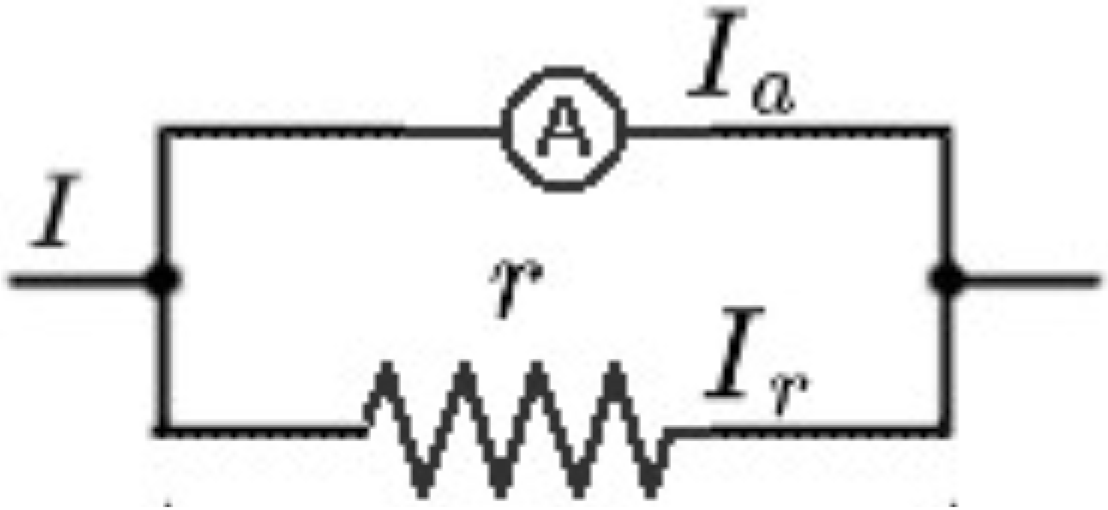
$$I = \frac{V}{Rtotal} = \frac{12V}{1.01k\Omega} = 11.88mA$$



https://tinyurl.com/yxeow7da

Ejercicio 2

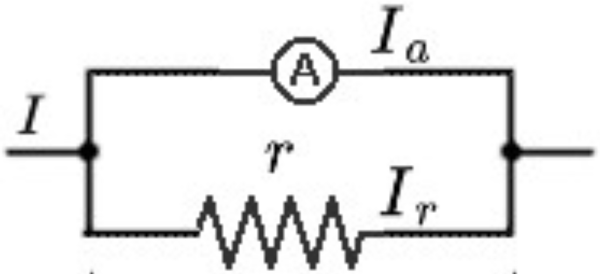
Ra = 2Ω y R = 6Ω. Calculad Ia e Ir




DIVISOR DE CORRIENTE

Ejercicio 3

$$\boxed{I = I_r + I_a}$$



Al estar en paralelo el voltaje es el mismo:

$V_a = V_r \Rightarrow I_a \,.\, R_a = I_r \,.\, R$

$$I_a \cdot 2 = (I - I_a)6 \Rightarrow 8 \cdot I_a = 6 \cdot I \Rightarrow \boxed{I_a = \frac{3}{4} I}$$

Volviendo a la intensidad en circuitos en paralelo…

$$I = I_r + I_a \Rightarrow I_r = I - I_a = I - \frac{3}{4} I = \boxed{\frac{1}{4} I}$$

Definición de funciones de conmutación mediante expresiones de conmutación

- **Valor de la *EC*** en el punto *X* de su dominio se calcula sustituyendo cada variable en *EC* por el valor que toma en dicho punto y aplicando las operaciones correspondientes a esos valores

  Ejemplo:

$$EC(x_2, x_1, x_0) = x_2 \cdot \overline{x}_1 + x_2 \cdot \overline{x}_1 + x_1 \cdot x_0 + x_0$$

Valor de la EC en (0,1,1) =

Valor de la EC en (1,1,0) =

Definición de funciones de conmutación mediante expresiones de conmutación

- **Valor de la *EC*** en el punto *X* de su dominio se calcula sustituyendo cada variable en *EC* por el valor que toma en dicho punto y aplicando las operaciones correspondientes a esos valores

  Ejemplo:

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

Valor de la EC en (0,1,1) = 1

Definición de funciones de conmutación mediante expresiones de conmutación

- **Valor de la *EC*** en el punto *X* de su dominio se calcula sustituyendo cada variable en *EC* por el valor que toma en dicho punto y aplicando las operaciones correspondientes a esos valores

  Ejemplo:

$$EC(x_2, x_1, x_0) = x_2 \cdot \overline{x}_1 + x_2 \cdot \overline{x}_1 + x_1 \cdot x_0 + x_0$$

Valor de la EC en (1,1,0) = 0

Definición de funciones de conmutación mediante expresiones de conmutación

- **Formas normales o canónicas de una EC:** son aquellas en las que en todos sus términos aparecen todas las variables (negadas o sin negar)
- Existen dos formas normales
  - **Forma normal disyuntiva**: suma de productos o minterms (Primera forma canónica)

Ejemplo:

$$EC(x_2, x_1, x_0) = x_2 \cdot \bar{x}_1 \cdot x_0 + x_2 \cdot \bar{x}_1 \cdot \bar{x}_0 + \bar{x}_2 \cdot x_1 \cdot \bar{x}_0$$

  - **Forma normal conjuntiva:** producto de sumas o maxterms (segunda forma canónica)

Ejemplo:

$$EC(x_2, x_1, x_0) = (\bar{x}_2 + \bar{x}_1 + x_0) \cdot (x_2 + \bar{x}_1 + \bar{x}_0)$$

Definición de funciones de conmutación mediante expresiones de conmutación

- Existen diferentes formas de obtener una EC que defina una FC.
- La forma más sencilla es obtener una EC a partir de la tabla de verdad de la FC, siguiendo el siguiente procedimiento:

1. Para cada punto del dominio donde la FC tome el valor 1, generar un producto formado por todas las variables tal que dicho producto se evalúe a ‘1’ en ese punto (cada variable que vale ‘0’ en ese punto aparecerá negada en el producto, y cada variable que vale ‘1’ aparecerá sin negar).
2. Formar una EC sumando todos los productos obtenidos.

La expresión así obtenida es una expresión en **forma normal disyuntiva**

Definición de funciones de conmutación mediante expresiones de conmutación

- Ejemplo: EC en FND obtenida a partir de una tabla de verdad

x2	x1	x0	FC
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

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

- ¿Y la FC?

Definición de funciones de conmutación mediante expresiones de conmutación

- Ejemplo: EC en FND obtenida a partir de una tabla de verdad

	A	B	C	F(A,B,C)
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Primera forma canónica
$S=m0+m3+m5+m7$
$S=(\bar{A}\cdot\bar{B}\cdot\bar{C})+(\bar{A}\cdot B\cdot C)+(A\cdot\bar{B}\cdot C)+(A\cdot B\cdot C)$
Segunda forma canónica
$S=M1\cdot M2\cdot M4\cdot M6$
$S=(A+B+\bar{C})\cdot(A+\bar{B}+C)\cdot(\bar{A}+B+C)\cdot(\bar{A}+\bar{B}+C)$

Teorema de Shannon

❑Teorema de Shannom.

Cualquier función binaria puede expresarse en forma de suma de minterms ó en forma de producto de maxterms. Estas expresiones, que son únicas, reciben el nombre de representaciones canónicas de la función.

$$f(a;b;c) = \sum m(2;5;7) = \bar{a}b\bar{c} + a\bar{b}c + abc$$

$$f(a;b;c) = \prod M(0;1;3;4;6) = \text{?¿?¿?¿?}$$

	a	b	c	f(a;b;c)	
0	0	0	0	**0**	
1	0	0	1	**0**	
2	0	1	0	**1**	$\bar{a}b\bar{c}$
3	0	1	1	**0**	
4	1	0	0	**0**	
5	1	0	1	**1**	$a\bar{b}c$
6	1	1	0	**0**	
7	1	1	1	**1**	abc

	a	b	c	f(a;b;c)
0	0	0	0	**0**
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	**1**

Teorema de Shannon

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \bar{x}_1\bar{x}_2\bar{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \bar{x}_1\bar{x}_2x_3$	$M_1 = x_1 + x_2 + \bar{x}_3$
2	0	1	0	$m_2 = \bar{x}_1x_2\bar{x}_3$	$M_2 = x_1 + \bar{x}_2 + x_3$
3	0	1	1	$m_3 = \bar{x}_1x_2x_3$	$M_3 = x_1 + \bar{x}_2 + \bar{x}_3$
4	1	0	0	$m_4 = x_1\bar{x}_2\bar{x}_3$	$M_4 = \bar{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\bar{x}_2x_3$	$M_5 = \bar{x}_1 + x_2 + \bar{x}_3$
6	1	1	0	$m_6 = x_1x_2\bar{x}_3$	$M_6 = \bar{x}_1 + \bar{x}_2 + x_3$
7	1	1	1	$m_7 = x_1x_2x_3$	$M_7 = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$

Teorema de Shannon

	A	B	C	F(A,B,C)
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Primera forma canónica

$S=m0+m3+m5+m7$

$$S = (\bar{A} \cdot \bar{B} \cdot \bar{C}) + (\bar{A} \cdot B \cdot C) + (A \cdot \bar{B} \cdot C) + (A \cdot B \cdot C)$$

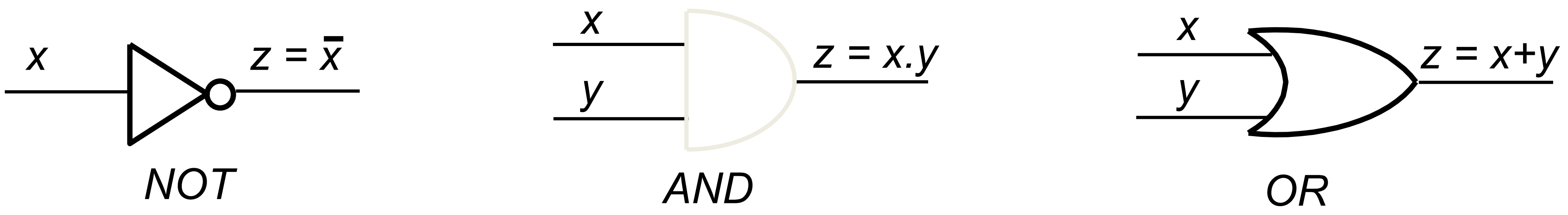
Segunda forma canónica

$S=M1 \cdot M2 \cdot M4 \cdot M6$

$$S = (A + B + \bar{C}) \cdot (A + \bar{B} + C) \cdot (\bar{A} + B + C) \cdot (\bar{A} + \bar{B} + C)$$

Puertas lógicas AND, OR, NOT

- Son dispositivos electrónicos concebidos para realizar operaciones elementales sobre información binaria representada con un cierto convenio (Ej: 0 = 0V, 1 = +5V).
- Las puertas AND, OR y NOT implementan las funciones lógicas del Álgebra de Boole, y constituyen un **conjunto universal de puertas**, dado que cualquier función lógica puede expresarse mediante una EC formada exclusivamente por estos tres tipos de elementos (es decir, cualquier sistema combinacional puede implementarse utilizando únicamente puertas AND, OR y NOT)
- Los símbolos con los que representamos estas puertas son



Síntesis de circuitos AND-OR-NOT a partir de EC

- A partir de una EC es bastante sencillo obtener un circuito utilizando puertas AND, OR y NOT que implemente dicha expresión.
- Basta utilizar una puerta para cada operador que aparezca en la expresión, y conectar sus entradas de manera que se correspondan con los argumentos de cada operador en la EC.
- Existen diferentes versiones de las puertas lógicas, con diferente número de entradas.

- Pero hacer esto directamente es poco eficiente y caro. Antes hay que **SIMPLIFICAR LA EC**

Simplificación de expresiones de conmutación

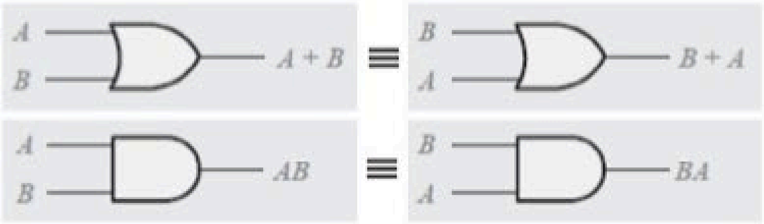
- Álgebra de conmutación es un **álgebra de Boole** y, por tanto, las operaciones suma lógica, producto lógico y negación cumplen las leyes de Boole.

PROPIEDADES	**Conmutativa**	$x+y = y+x$	$x \cdot y = y \cdot x$
	Elemento neutro	$0+x = x$	$1 \cdot x = x$
	Distributiva	$x \cdot (y+z) = (x \cdot y)+(x \cdot z)$	$x+(y \cdot z) = (x+y)(x+z)$
	Asociativa	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	$x+(y+z) = (x+y)+z$
	Complementario	$x+\overline{x} = 1$	$x \cdot \overline{x} = 0$
TEOREMAS	**Idempotencia**	$x+x = x$	$x \cdot x = x$
	Identidad	$x+1 = 1$	$x \cdot 0 = 0$
	Absorción	$x+x \cdot y = x$	$x \cdot (x+y) = x$
	DeMorgan	$\overline{(x+y)} = \overline{x} \cdot \overline{y}$	$\overline{(x \cdot y)} = \overline{x} + \overline{y}$

Propiedades del álgebra de Boole (1)

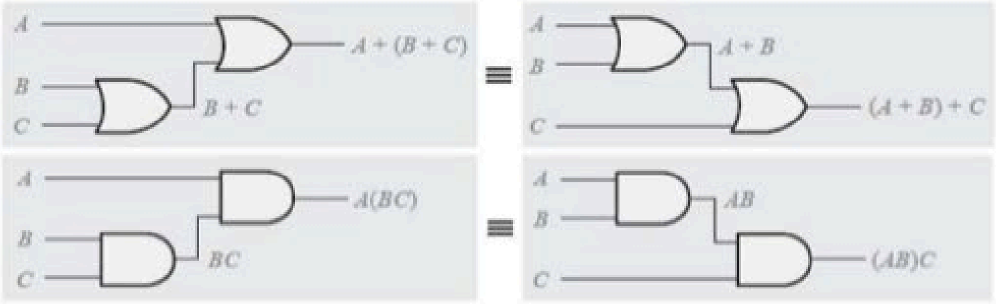
❑Ley conmutativa.

- ➢ Operación “OR”:
  A + B = B + A
- ➢ Operación “AND”:
  A * B = B * A



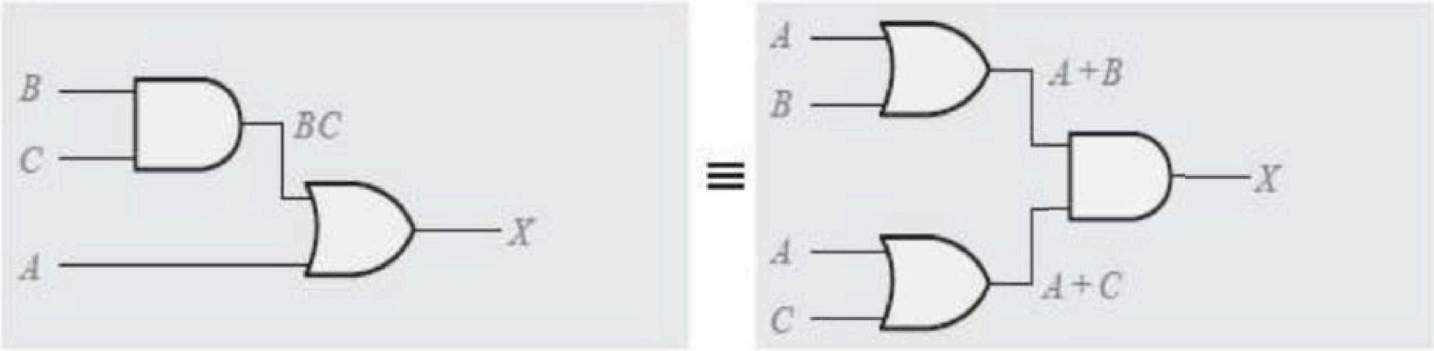
❑Ley asociativa.

- ➢ Operación “OR”:
  A + B + C = (A + B) + C = A + (B + C)
- ➢ Operación “AND”:
  A * B * C = (A * B) * C = A * (B * C)



Propiedades del álgebra de Boole (2)

- **“Ley distributiva” (Regla 12).**
  - Operación: A + (B * C) = (A + B) * (A + C)



A	B	C	A+B	A+C	B*C	A+(B*C)	(A+B)*(A*C)
0	0	0	0	0	0	**0**	**0**
0	0	1	0	1	0	**0**	**0**
0	1	0	1	0	0	**0**	**0**
0	1	1	1	1	1	**1**	**1**
1	0	0	1	1	0	**1**	**1**
1	0	1	1	1	0	**1**	**1**
1	1	0	1	1	0	**1**	**1**
1	1	1	1	1	1	**1**	**1**

Reglas del álgebra de Boole (1)

- **Regla 1:**

$$A + 0 = A$$

- **Regla 2:**

$$A + 1 = 1$$

- **Regla 3:**

$$A \cdot 0 = 0$$

- **Regla 4:**

$$A \cdot 1 = A$$

- **Regla 5:**

$$A + A = A$$

- **Regla 6:**

$$A + \overline{A} = 1$$

- **Regla 7:**

$$A \cdot A = A$$

- **Regla 8:**

$$A \cdot \overline{A} = 0$$

- **Regla 9:**

$$\overline{\overline{A}} = A$$

- **Regla 10:**

$$A + AB = A$$

- **Regla 11:**

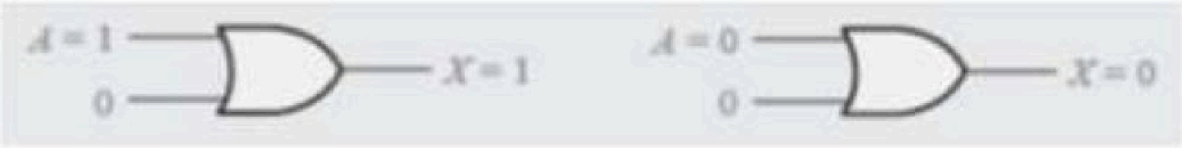
$$A + \overline{A}B$$

- **Regla 12:**

$$(A+B) \cdot (A+C) = A \cdot (B+C)$$

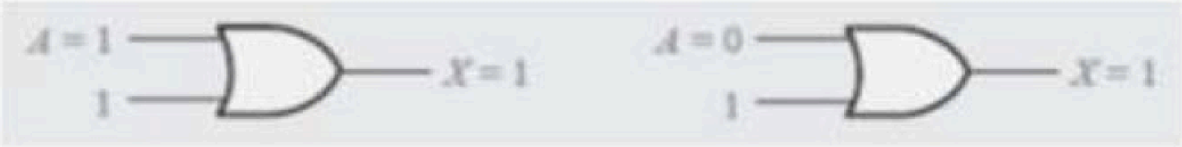
Reglas del Algebra de Boole (2)

- **Regla 1: $A + 0 = A$.**
  - Aplicando la operación OR a una variable cualquiera y a "0", el resultado es siempre igual a la variable.



$X = A + 0 = A$

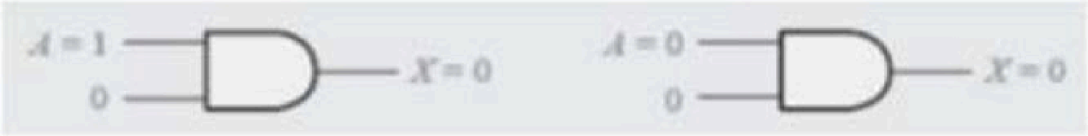
- **Regla 2: $A + 1 = 1$.**
  - Aplicando la operación OR a una variable cualquiera y a "1", el resultado es siempre igual "1".



$X = A + 1 = 1$

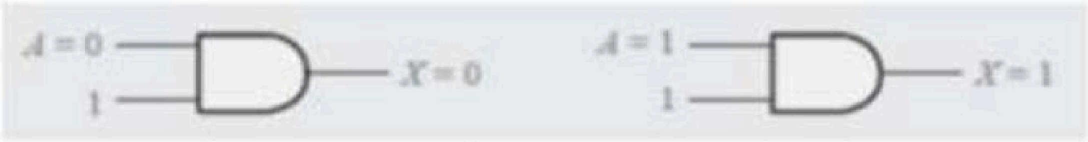
Reglas del Algebra de Boole (3)

- **Regla 3: $A \cdot 0 = 0$.**
  - Aplicando la operación AND a una variable cualquiera y a "0", el resultado es siempre igual "0".



$X = A \cdot 0 = 0$

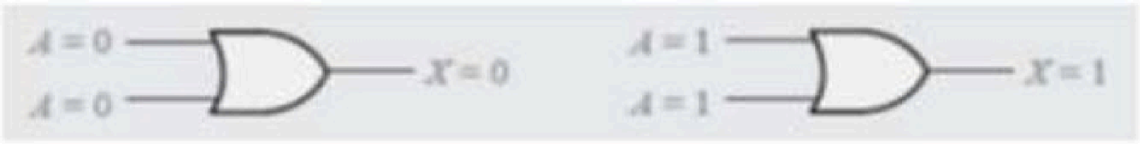
- **Regla 4: $A \cdot 1 = A$.**
  - Aplicando la operación AND a una variable cualquiera y a "1", el resultado es siempre igual a la variable.



$X = A \cdot 1 = A$

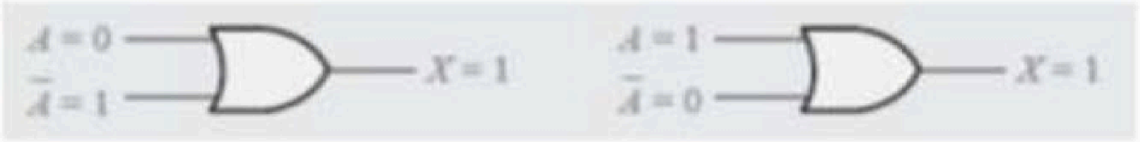
Reglas del Algebra de Boole (4)

- **Regla 5: A + A = A.**
  - Aplicando la operación OR a una variable cualquiera consigo misma, el resultado es siempre igual a la misma variable.



$X = A + A = A$

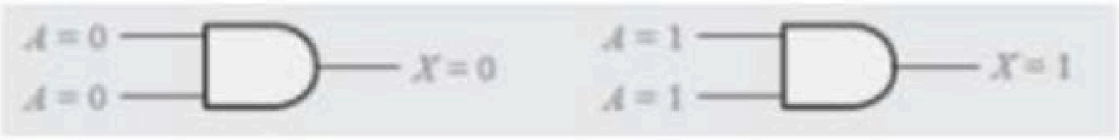
- **Regla 6: $A + \overline{A} = 1$.**
  - Aplicando la operación OR a una variable cualquiera con su complemento, el resultado es siempre igual a "1".



$X = A + \overline{A} = 1$

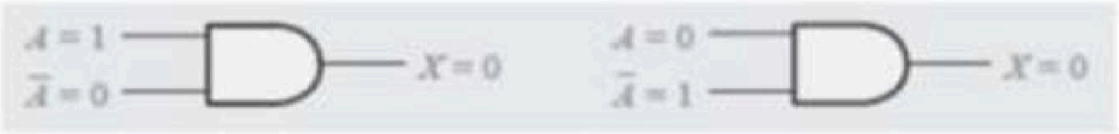
Reglas del Algebra de Boole (5)

- **Regla 7: $A \cdot A = A$.**
  - Aplicando la operación AND a una variable cualquiera consigo misma, el resultado es siempre igual a la misma variable.



$X = A \cdot A = A$

- **Regla 8: $A \cdot \overline{A} = 0$.**
  - Aplicando la operación AND a una variable cualquiera con su complemento, el resultado es siempre igual a "0".

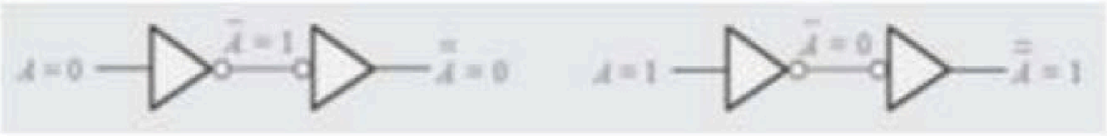


$X = A \cdot \overline{A} = 0$

Reglas del Algebra de Boole (6)

❑Regla 9: $\overline{\overline{A}} = A$.

- El complemento del complemento de una variable es siempre la propia variable.



$\overline{\overline{A}} = A$

❑Regla 10: $A + AB = A$.

- Esta regla se puede obtener aplicando la ley distributiva y las reglas 2 y 4.

$A \cdot (1+B) = A \cdot (1) = A$

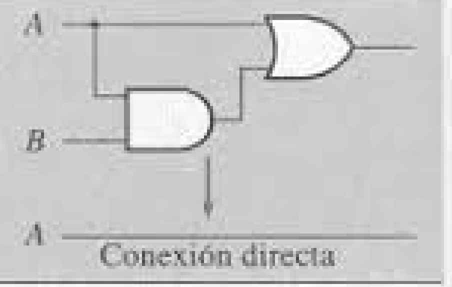
Factor Común

Regla "2"

Regla "4"

A	B	AB	$A + AB$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

igual



Reglas del Algebra de Boole (7)

- **Regla 11: $A + \overline{A}B$** $= A+B$
  - Esta regla se puede obtener aplicando la ley distributiva y las reglas 2 y 4.

    $(A + \overline{A}) \cdot (A+B) = (1) \cdot (A+B) = A+B$

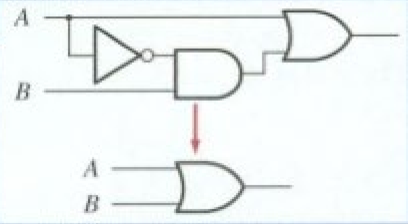
    Distributiva

    Regla "6"

    Regla "2"

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

equal



- **Regla 12: $(A+B)\cdot(A+C) = A\cdot(B+C)$.**
  - Esta regla se puede obtener aplicando la ley distributiva y la regla 7 y 10.

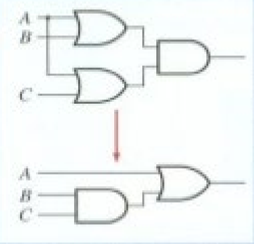
$= AA + AC + AB + BC$ Ley distributiva

$= A + AC + AB + BC$ Regla 7: $AA = A$

$= A + BC$ Regla 10: $A + AB = A$ (aplicada 2 veces)

A	B	C	$A+B$	$A+C$	$(A+B)(A+C)$	BC	$A+BC$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

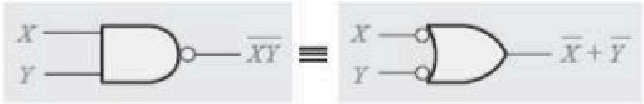
equal



Leyes de DeMorgan

- El complemento de un producto de variables, es igual a la suma de los complementos de las variables.

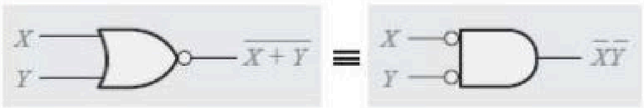
$$\overline{X \bullet Y} = \overline{X} + \overline{Y}$$



Entradas		Salida	
X	Y	$\overline{XY}$	$\overline{X}+\overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

- El complemento de una suma de variables, es igual al producto de los complementos de las variables.

$$\overline{X + Y} = \overline{X} \bullet \overline{Y}$$



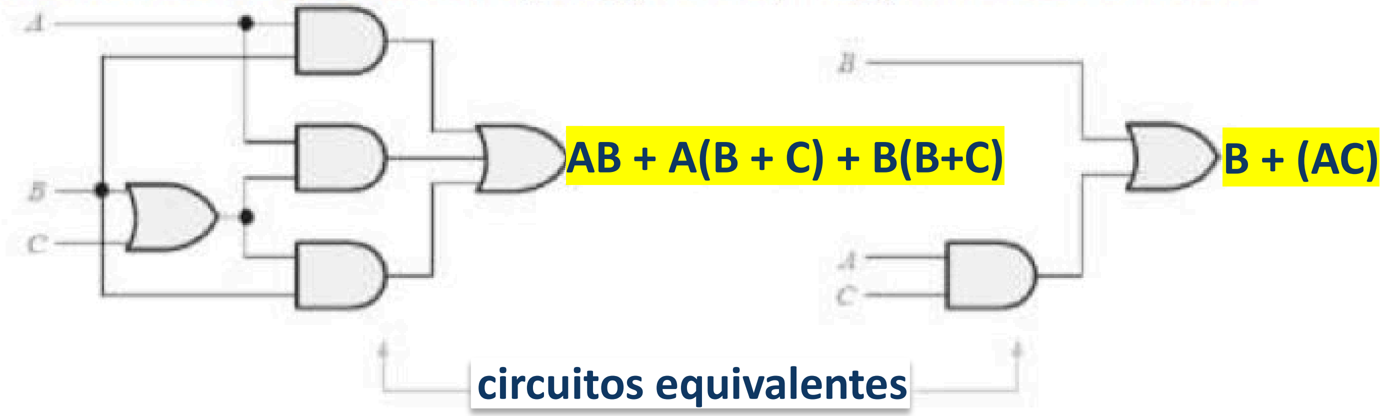
Entradas		Salida	
X	Y	$\overline{X+Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Simplificación de ECs

- Se trata de reducir una expresión a su forma más simple ó cambiarla a una forma más conveniente para conseguir una implementación más eficiente.

- Se trata de utiliza las reglas, leyes y teoremas del álgebra de Boole para manipular y simplificar una expresión.

- Este método requiere un conocimiento del álgebra booleana y experiencia en su aplicación, ingenio y algo de destreza.



Simplificación de expresiones de conmutación

- **Equivalencia entre EC**: dos expresiones de conmutación EC_1 y EC_2 son equivalentes si ambas representan la misma función de conmutación.
- Pueden utilizarse la Leyes de Boole para manipular las EC y obtener otras equivalentes.

Ejemplo:

$$
\begin{aligned}
EC(x_2, x_1, x_0) &= x_2 \cdot \overline{x}_1 \cdot x_0 + x_2 \cdot \overline{x}_1 \cdot \overline{x}_0 + \overline{x}_2 \cdot x_1 \cdot \overline{x}_0 \\
&= x_2 \cdot \overline{x}_1 \cdot (x_0 + \overline{x}_0) + \overline{x}_2 \cdot x_1 \cdot \overline{x}_0 \\
&= x_2 \cdot x_1 \cdot 1 + \overline{x}_2 \cdot x_1 \cdot \overline{x}_0 \\
&= x_2 \cdot x_1 + \overline{x}_2 \cdot x_1 \cdot \overline{x}_0
\end{aligned}
$$

- **Simplificación de EC**: obtener otras equivalentes con un menor número de operadores.
  - Puede hacerse aplicando las leyes de Boole (como ocurría en el ejemplo anterior).

Simplificación de EC: Ejemplo

Simplificar la siguiente expresión utilizando técnicas del álgebra de Boole.

$$AB + A(B+C) + B(B+C)$$

Simplificación de EC: Ejemplo

Simplificar la siguiente expresión utilizando técnicas del álgebra de Boole.

$$AB + A(B+C) + B(B+C)$$

❑Regla 5:

$$A + A = A$$

$$AB + AB + AC + BB + BC$$

$$B + AB + BC + AC$$

$$B\,(1 + A + C) + AC$$

$$\boxed{B + AC}$$

❑Regla 7:

$$A \cdot A = A$$

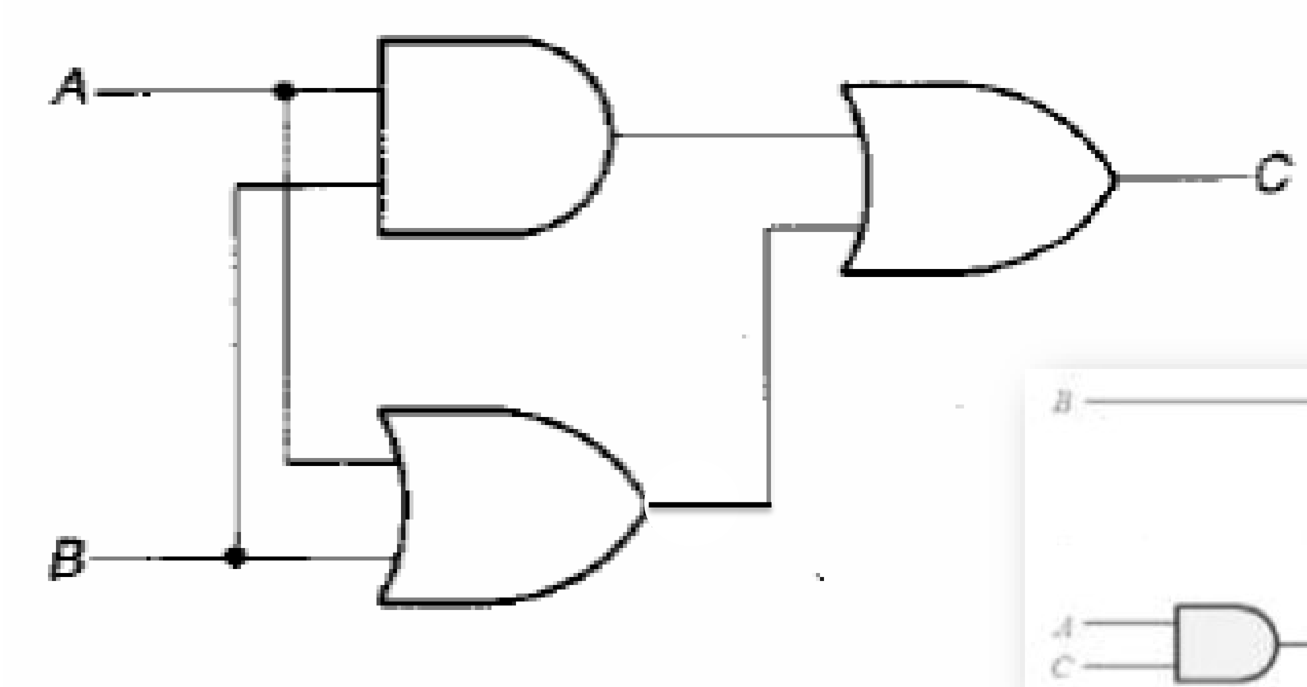
❑Regla 2:

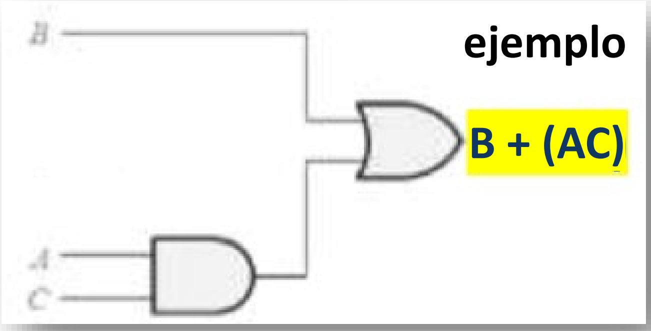
$$A + 1 = 1$$

Ejercicio 1

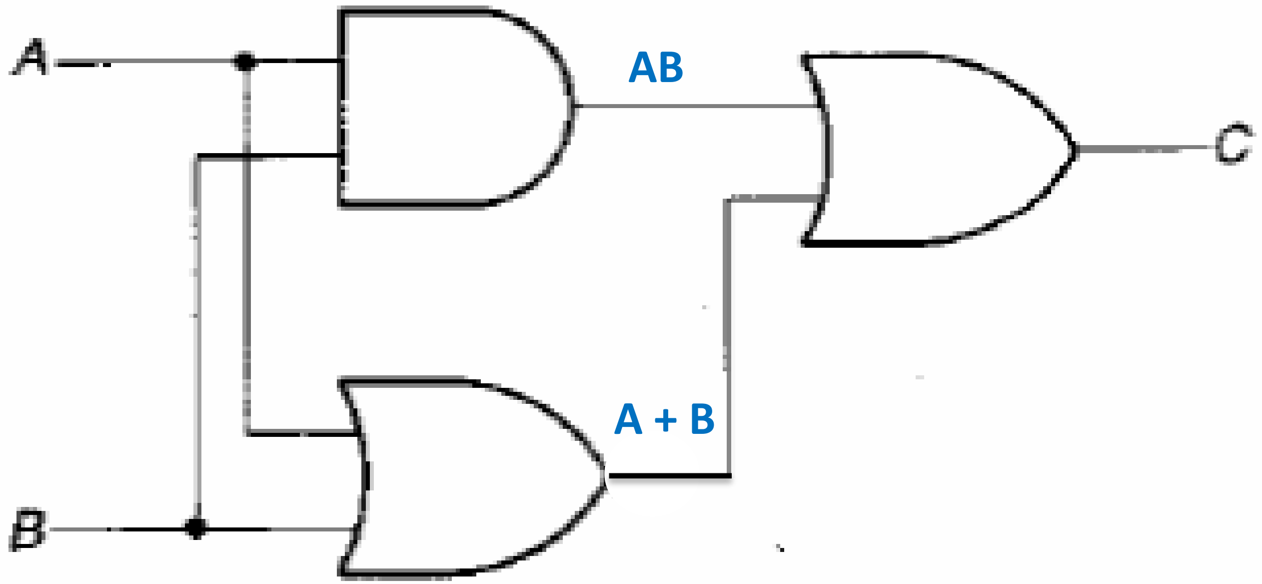
Sacad expresión de conmutación de este circuito:

C = ?





solución

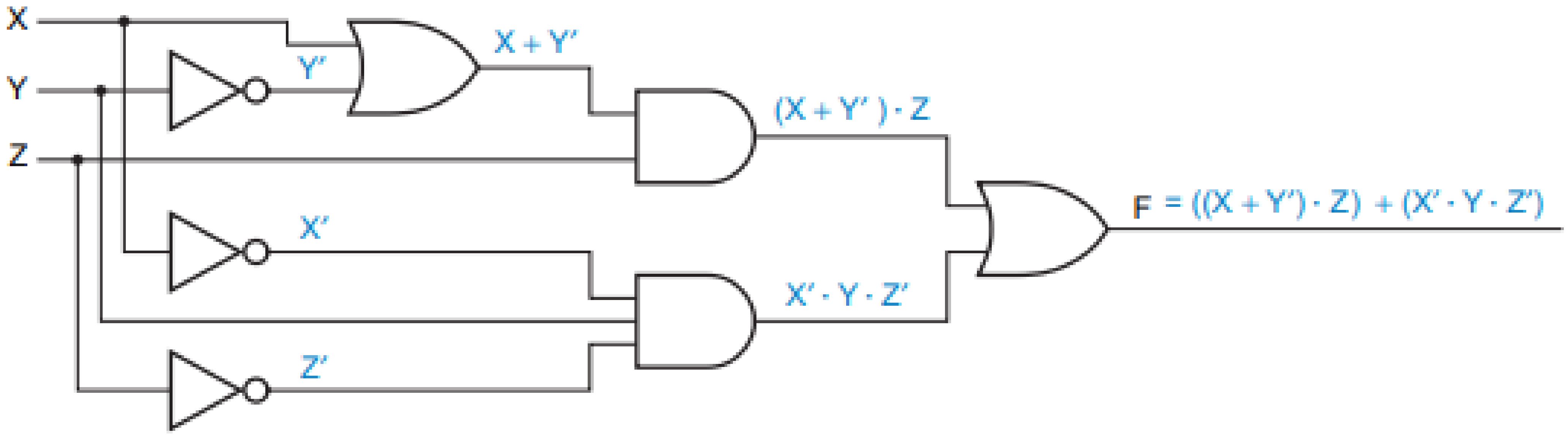


$$C = AB + (A + B)$$

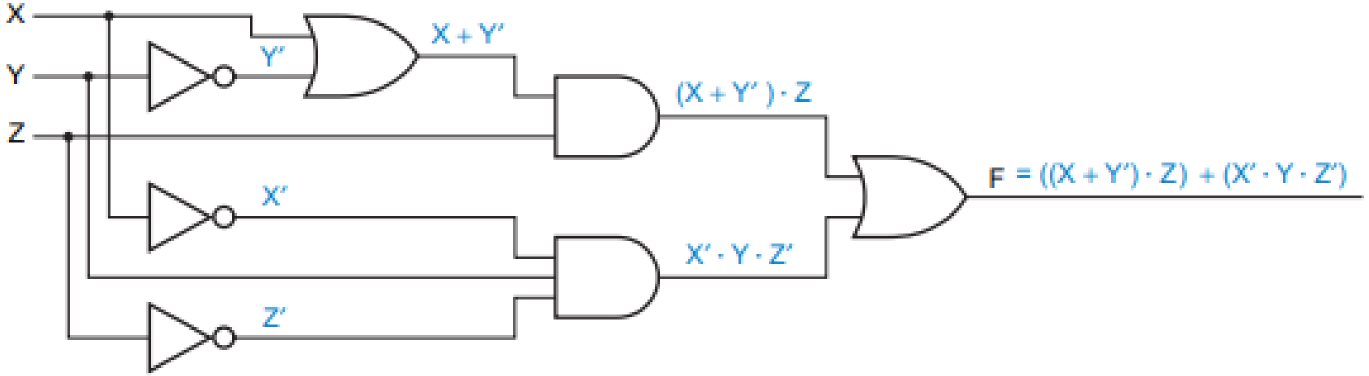
Ejercicio 2

Tabla de verdad de esta expresión (suma de productos)

Nota: Y’ es $\overline{Y}$



solución



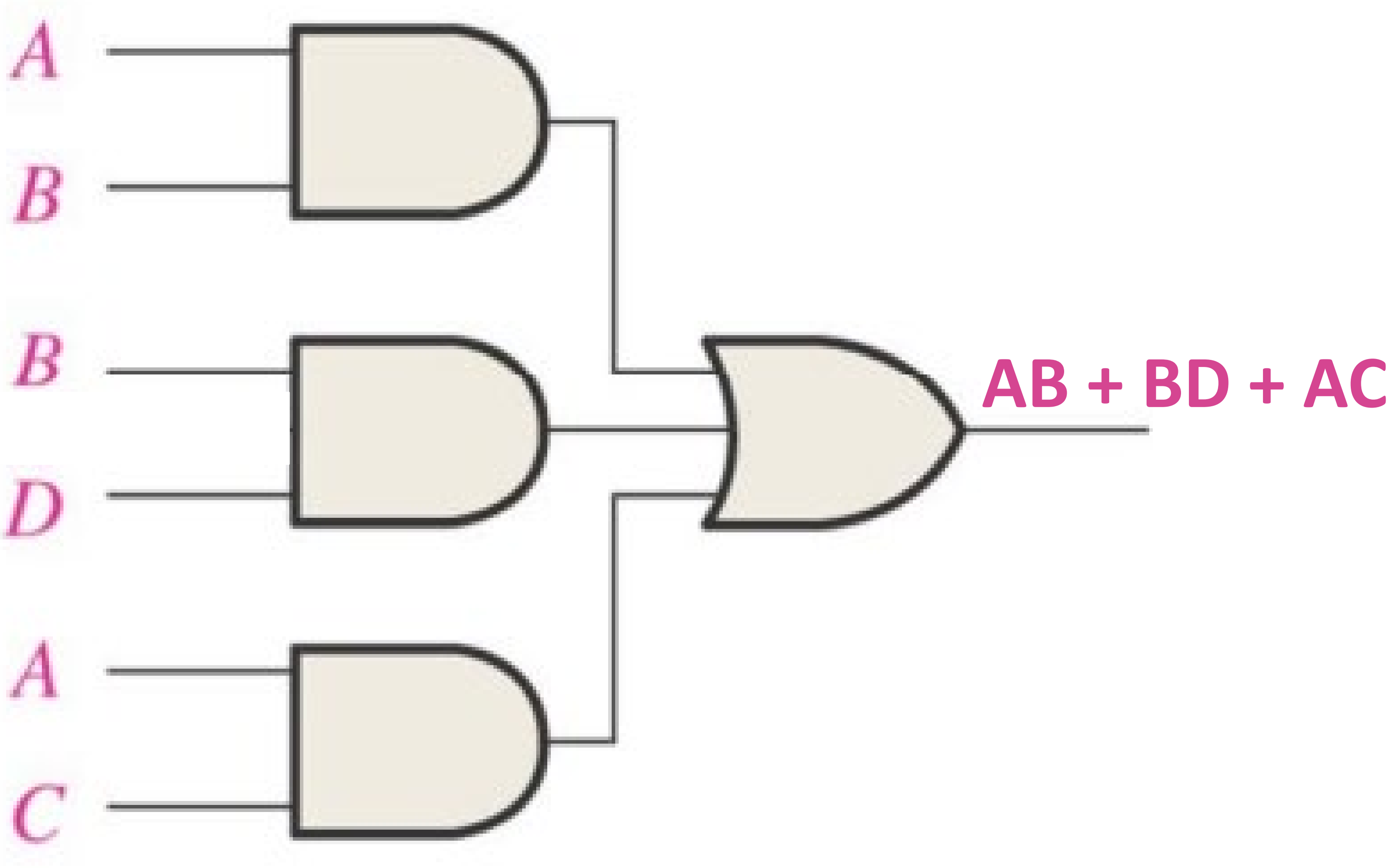
Row	X	Y	Z	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Ejercicio 3

Cread el circuito a partir de esta expresión:

AB + BD + AC

solución



Ejercicio 4

Expresad en forma normal disyuntiva y conjuntiva esta tabla de verdad

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

solución

- Suma de Productos

A	B	C	X	Producto
0	0	0	0	
0	0	1	1	A’B’C
0	1	0	0	
0	1	1	0	
1	0	0	1	AB’C’
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

X = A’B’C + AB’C’ + ABC

- Producto de sumas

A	B	C	X	Suma
0	0	0	0	(A+B+C)
0	0	1	1	
0	1	0	0	(A+B’+C)
0	1	1	0	(A+B’+C’)
1	0	0	1	
1	0	1	0	(A’+B+C’)
1	1	0	0	(A’+B’+C)
1	1	1	1	

X = (A+B+C) (A+B’+C) (A+B’+C’)
(A’+B+C’) (A’+B’+C)

Ejercicio 5

Simplificad la siguiente expresión

$$(X \cdot Y \cdot Z) + (Y \cdot Z) + (X \cdot Y)$$

solución

Factorizar el termino (Y · Z):

$E = (X + 1) \cdot (Y \cdot Z) + (X \cdot Y)$

Dado que A + 1 = 1 según las leyes fundamentales por lo tanto X + 1 = 1:

$E = 1 \cdot (Y \cdot Z) + (X \cdot Y)$

Al realizar la operación tendremos ya simplificada la expresión:

$E = (Y \cdot Z) + (X \cdot Y)$

Aún podemos simplificar la expresión al factorizar Y:

$$E = Y \cdot (Z + X)$$

Simplificación de EC: Ejemplo 2

Simplificar la siguiente expresión utilizando técnicas del álgebra de Boole.

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

Simplificación de EC: Ejemplo 2

Simplificar la siguiente expresión utilizando técnicas del álgebra de Boole.

$$\overline{\overline{\mathbf{AB}} + \overline{\mathbf{A}(\mathbf{B} + \mathbf{C})} + \overline{\mathbf{B}(\mathbf{B} + \mathbf{C})}}$$

$$(AB) \cdot (A(B + C)) \cdot (B(B + C))$$

$$(AB) \cdot (AB + AC) \cdot (B + BC)$$

$$(AB) \cdot (ABB + ABBC + ABC + ABCC)$$

$$(AB) \cdot (AB + ABC)$$

$$(AB + ABC)$$

$$AB \cdot (1 + C)$$

$$\boxed{\mathbf{AB}}$$

Nota sobre formas canónicas FND

Forma canónica de la suma de productos

- **¿En qué consiste?**

  Que todos los términos de los productos de la suma de productos, contengan todas las variables del dominio.

- **Ejemplo: El dominio de la expresión es el conjunto de variables a, b y c, y la función booleana “$\mathbf{ab\bar{c} + bc + \bar{a}}$”.**
  - El término “$\mathbf{ab\bar{c}}$”, está completo.
  - El término “$\mathbf{bc}$”, le falta la variable “a”, por tanto:
    - Los términos “$\bar{a}bc$” y “abc”.
  - El término “$\mathbf{\bar{a}}$”, le faltan las variables “b” y “c”, por tanto:
    - Para la variable “b”: El término “$\bar{a}b$” y el término “$\bar{a}\bar{b}$”
    - Para la variable “c”: Partiendo de los dos (2) anteriores, los términos “$\bar{a}bc$”, “$\bar{a}b\bar{c}$”, “$\bar{a}\bar{b}c$” y “$\bar{a}\bar{b}\bar{c}$”.

$$\boxed{\mathbf{ab\bar{c} + \bar{a}bc + abc + \bar{a}b\bar{c} + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c}}}$$

Nota sobre formas canónicas FNC

Forma canónica del producto de sumas

- **¿En qué consiste?**

  Que todos los términos de las sumas del producto de sumas, contengan todas las variables del dominio.

- **Ejemplo: El dominio de la expresión es de las variables a, b, c y d, y la función booleana "$(\mathbf{a} + \bar{\mathbf{b}} + \mathbf{c} + \bar{\mathbf{d}}) \cdot (\mathbf{a} + \bar{\mathbf{b}} + \mathbf{c}) \cdot (\bar{\mathbf{b}} + \mathbf{c} + \bar{\mathbf{d}})$".**

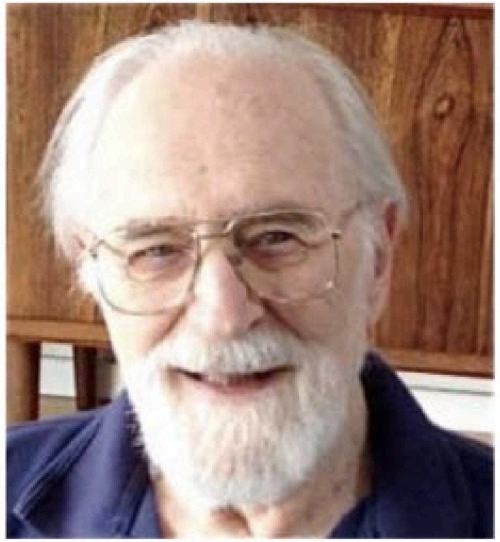
  - El término "$\mathbf{a} + \bar{\mathbf{b}} + \mathbf{c} + \bar{\mathbf{d}}$", está completo.
  - El término "$\mathbf{a} + \bar{\mathbf{b}} + \mathbf{c}$", le falta la variable "d", por tanto:
    - Los términos "$a + \bar{b} + c + d$" y "$a + \bar{b} + c + \bar{d}$".
  - El término "$\bar{\mathbf{b}} + \mathbf{c} + \bar{\mathbf{d}}$", le falta la variable "a", por tanto:
    - Los términos "$a + \bar{b} + c + \bar{d}$" y "$\bar{a} + \bar{b} + c + \bar{d}$".

$$\boxed{(\mathbf{a} + \bar{\mathbf{b}} + \mathbf{c} + \bar{\mathbf{d}}) \cdot (\mathbf{a} + \bar{\mathbf{b}} + \mathbf{c} + \mathbf{d}) \cdot (\bar{\mathbf{a}} + \bar{\mathbf{b}} + \mathbf{c} + \bar{\mathbf{d}})}$$

Simplificación por mapas de Karnaugh

Maurice Karnaugh

- **Físico y Matemático estadounidense del siglo XX.**
- **Trabajó en los laboratorios Bell e IBM.**
- **Publica "The Map Method for Synthesis of Combinational Logic Circuits" (1953).**
  - Presenta y define como trabajar con métodos gráficos con el álgebra booleana.



Simplificación por mapas de Karnaugh

Generalidades

- **Proporciona un método sistemático de simplificación de sentencias booleanas generando expresiones mínimas ("receta de simplificación").**

- **Características.**
  - Útiles para expresiones de dos (2), tres (3), cuatro (4) y cinco (5) variables.
  - Es una matriz de 2^n celdas en la que cada una representa un valor binario de las variables de entrada.
  - El orden de los valores en filas y columnas es tal que celdas adyacentes difieren únicamente en una variable.
  - La simplificación de una determinada expresión consiste en agrupar "adecuadamente" las celdas.
  - Un número mayor de variables exige el uso de un método llamado Quine-McClusky.

Simplificación por mapas de Karnaugh

Tablas de dos (2), tres (3), cuatro (4) y cinco(5) variables

- **Tabla de:**
  - Dos (2) variables.
  - Tres (3) variables.
  - Cuatro (4) variables.
  - Cinco (5) variables.

(a,b)	$\bar{a}$	a
$\bar{b}$	0	0
b	0	0

(a,b,c)	$\bar{a}$		a	
	$\bar{b}$	b	b	$\bar{b}$
$\bar{c}$				
c				

(a,b,c,d)		$\bar{a}$		a	
		$\bar{b}$	b	b	$\bar{b}$
$\bar{c}$	$\bar{d}$				
	d				
c	d				
	$\bar{d}$				

(a,b,c,d,e)		$\bar{a}$				a			
		$\bar{b}$		b		b		$\bar{b}$	
		$\bar{c}$	c	c	$\bar{c}$	$\bar{c}$	c	c	$\bar{c}$
$\bar{d}$	$\bar{e}$								
	e								
d	e								
	$\bar{e}$								

Simplificación por mapas de Karnaugh

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

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

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

Adyacencia en Karnaugh

❑Concepto de adyacencia.

- Las celdas de un mapa de Karnaugh se disponen de manera que sólo cambia una única variable entre celdas adyacentes.
- La **adyacencia** se define por un cambio de una única variable.
- Las celdas que difieren en una única variable son adyacentes.

❑Tipos de adyacencia.

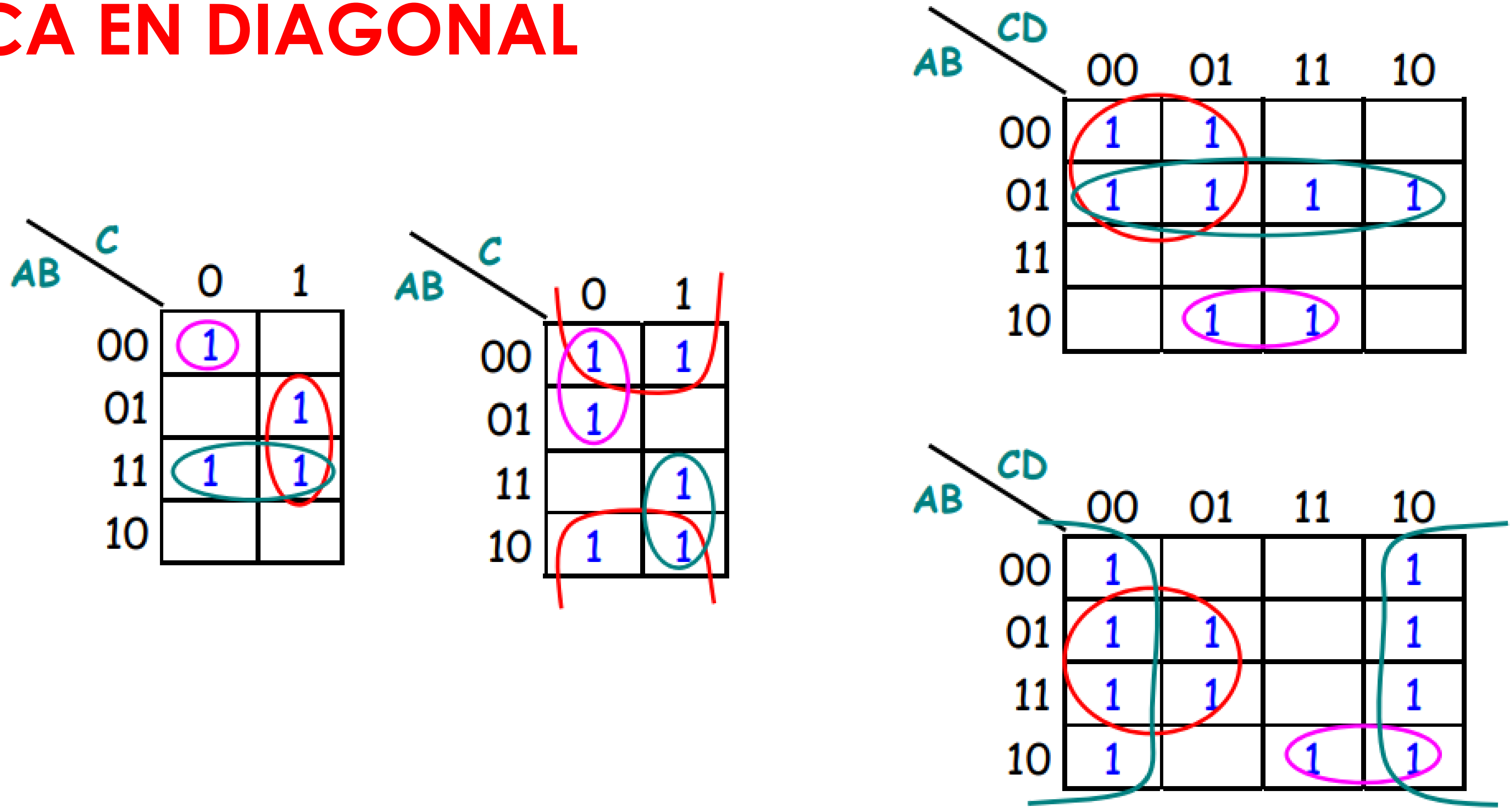
- En el interior (●).
- En los extremos (●).

”esfera”

F(A,B,C,D)		$\bar{A}$		A	
		$\bar{B}$	B	B	$\bar{B}$
$\bar{C}$	$\bar{D}$	0	2	3	1
	D	8	10	11	9
C	D	12	14	15	13
	$\bar{D}$	4	6	7	5

Adyacencia en Karnaugh

NUNCA EN DIAGONAL



Simplificación de FND (minterms)

Procedimiento para obtener la función lógica suma de productos

- **Representar en el mapa de Karnaugh la función algebraica ó tabla de verdad que se desee representar.**
- **Agrupar los "1´s".**
  - Un grupo tiene que contener uno (1), dos (2), cuatro (4), ocho (8) ó dieciséis (16) celdas. **SIEMPRE POTENCIA DE 2**
  - Cada celda del grupo tiene que ser adyacente a una ó mas celdas del grupo sin necesidad de que todas las celdas del grupo sean adyacentes entre sí.
  - Incluir siempre en cada grupo el mayor número posible de unos "1´s".
  - Cada "1" del mapa tiene que estar incluido en al menos un grupo. Los "1´s" que ya pertenezcan a un grupo pueden estar incluidos en otro, siempre que los grupos que se solapen contengan "1´s" no comunes.
- **Simplificar.**
  - Eliminar variables que aparecen complementadas y sin complementar dentro del mismo grupo

Simplificación de FNC (maxterms)

Procedimiento para obtener la función lógica producto de sumas

- **Representar en el mapa de Karnaugh la función algebraica ó tabla de verdad que se desee representar.**
- **Agrupar los “0´s”.**
  - Un grupo tiene que contener uno (1), dos (2), cuatro (4), ocho (8) ó dieciséis (16) celdas.
  - Cada celda del grupo tiene que ser adyacente a una ó mas celdas del grupo sin necesidad de que todas las celdas del grupo sean adyacentes entre sí.
  - Incluir siempre en cada grupo el mayor número posible de ceros “0´s”.
  - Cada “0” del mapa tiene que estar incluido en al menos un grupo. Los “0´s” que ya pertenezcan a un grupo pueden estar incluidos en otro, siempre que los grupos que se solapen contengan “0´s” no comunes.
- **Simplificar.**
  - Eliminar variables que aparecen complementadas y sin complementar dentro del mismo grupo y negar las variables de posición de los 0’s

Ejemplos

Simplificación de una SOPs mediante el mapa de Karnaugh

- **Función booleana normalizada de tres (3) variables.**

$$f(a,b,c) = \bar{a}\bar{c} + ab\bar{c}$$

PRIMERO

Se debe buscar la función booleana estándar.

$$f(a,b,c) = \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + ab\bar{c}$$

Cojo este grupo

(a,b,c)	$\bar{a}$		a	
	$\bar{b}$	b	b	$\bar{b}$
$\bar{c}$	1	1	1	0
c	0	0	0	0

Cojo este grupo

$$\boxed{f(a,b,c) = \bar{a}\bar{c} + b\bar{c}}$$

Eliminar variables EN CADA GRUPO que estén a la vez negadas y sin negar (b y a)

Ejemplos

Simplificación de una SOPs mediante el mapa de Karnaugh

- Función booleana estándar de tres (3) variables.

$$f(a,b,c) = \bar{a}bc + a\bar{b}c + ab\bar{c}$$

(a,b,c)	$\bar{a}$		a	
	$\bar{b}$	b	b	$\bar{b}$
$\bar{c}$	0	0	1	0
c	0	1	0	1

$$\boxed{f(a,b,c) = \bar{a}bc + a\bar{b}c + ab\bar{c}}$$

Ejemplos

Simplificación de una SOPs mediante el mapa de Karnaugh

- Función booleana estándar de tres (3) variables.

$$f(a,b,c) = \bar{a}bc + a\bar{b}c + ab\bar{c}$$

RECORDAD LA ADYACENCIA EN EL INTERIOR

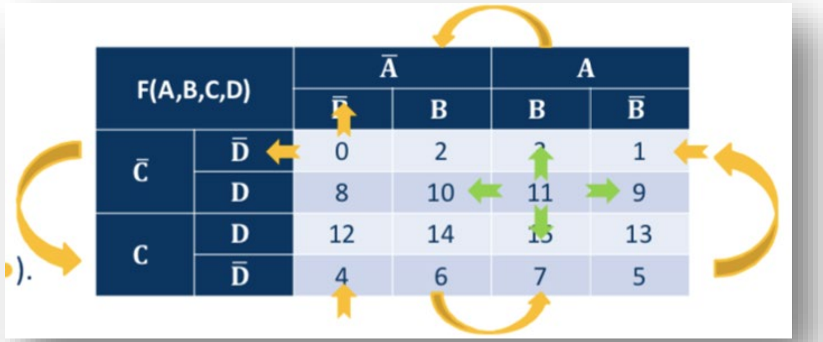
(a,b,c)	$\bar{a}$		a	
	$\bar{b}$	b	b	$\bar{b}$
$\bar{c}$	0	0	1	0
c	0	1	0	1

B	B	$\bar{B}$
2	[illegible]	1
10	11	9
14	[illegible]	13

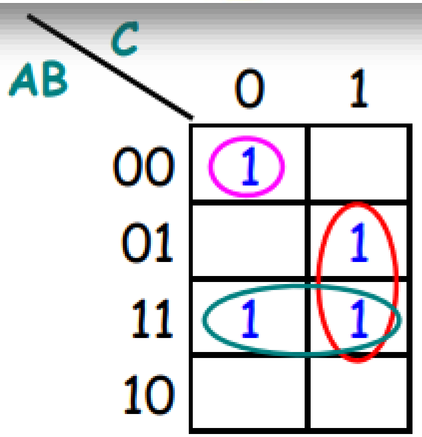
NO PODEMOS FORMAR GRUPOS

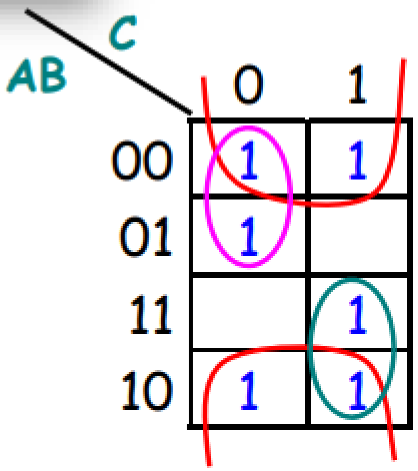
$$f(a,b,c) = \bar{a}bc + a\bar{b}c + ab\bar{c}$$

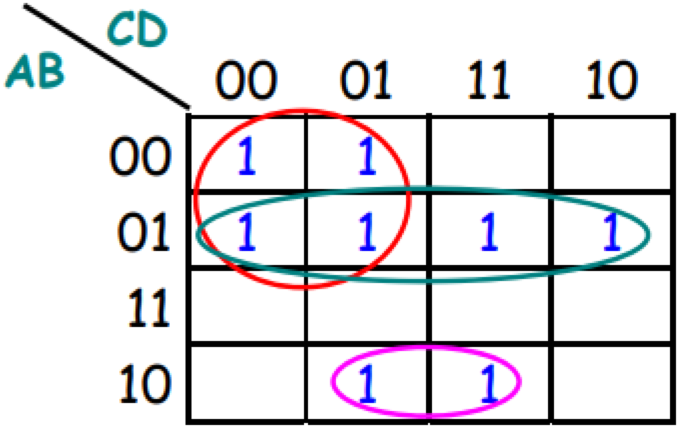
Ejemplos: TIPOS DE ADYACENCIAS

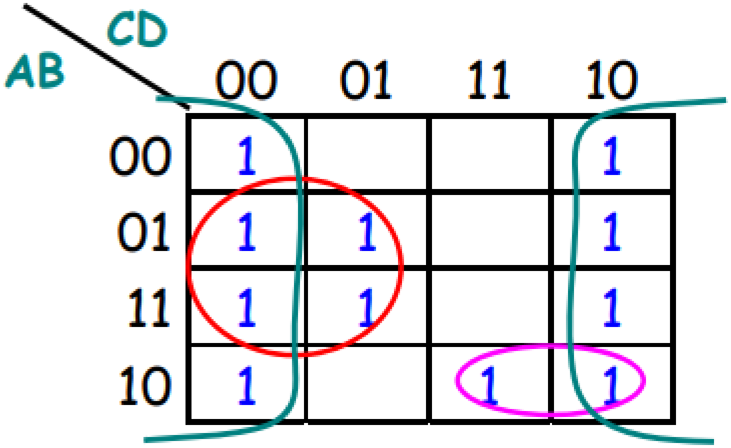


F(A,B,C,D)		$\bar{A}$		A	
		$\bar{B}$	B	B	$\bar{B}$
$\bar{C}$	$\bar{D}$	0	2	3	1
	D	8	10	11	9
C	D	12	14	15	13
	$\bar{D}$	4	6	7	5









Ejemplos

Simplificación de una SOPs mediante el mapa de Karnaugh

- Función booleana estándar de cuatro (4) variables.

$$f(a,b,c,d) = \bar{a}bcd + ab\bar{c}d + a\bar{b}cd + a\bar{b}\bar{c}d$$

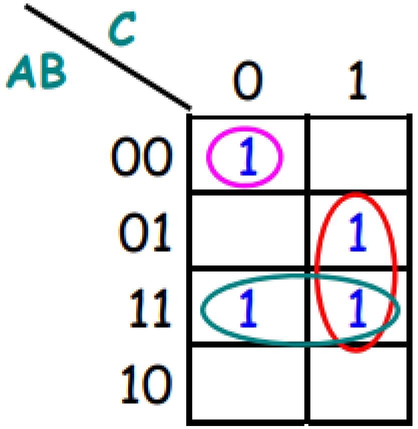
me quito b

(a,b,c,d)		$\bar{a}$		a	
		$\bar{b}$	b	b	$\bar{b}$
$\bar{c}$	$\bar{d}$	0	0	0	0
	d	0	0	1	1
c	d	0	1	0	1
	$\bar{d}$	0	0	0	0

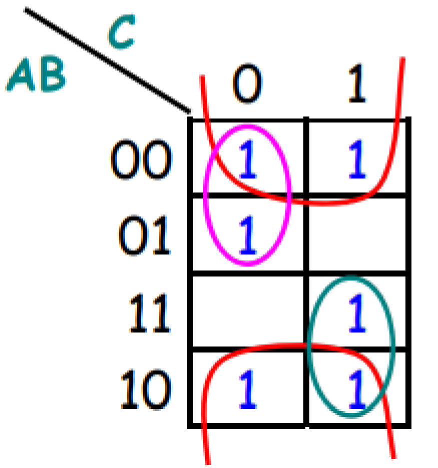
me quito c

$$f(a,b,c) = \bar{a}bcd + a\bar{c}d + a\bar{b}d$$

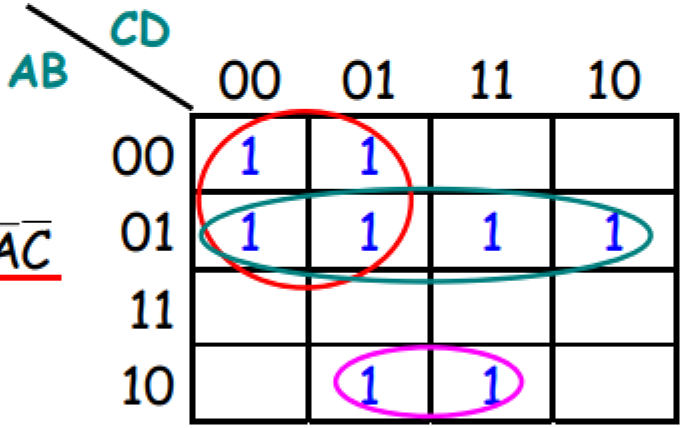
Ejemplos SIMPLIFICACIONES



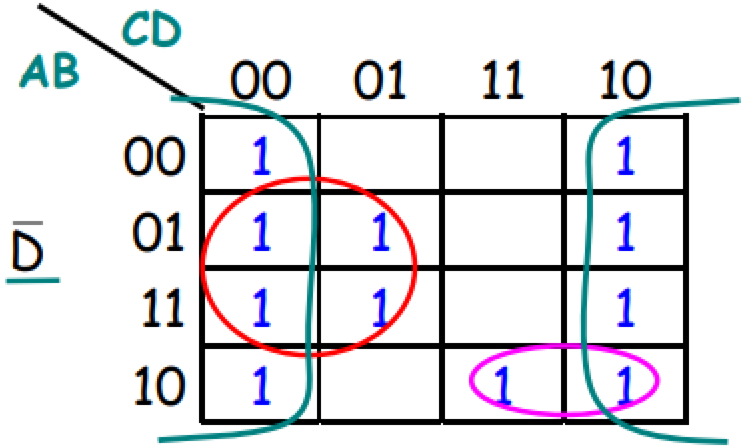
$\overline{A}\,\overline{B}\,\overline{C} + AB + BC$



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



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



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

Ejemplos SIMPLIFICACIONES

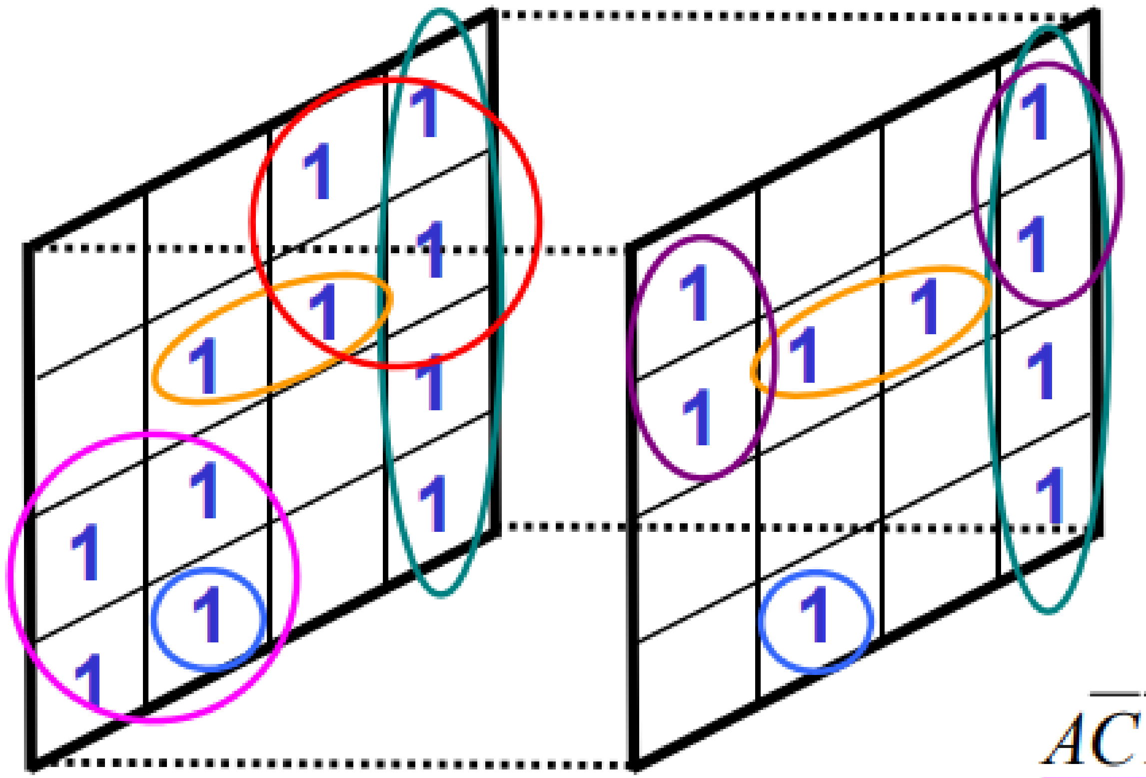
AB \ CD	00	01	11	10	00	01	11	10
00			1	1	1			1
01		1	1	1	1	1	1	1
11	1	1		1				1
10	1	1		1		1		1
	E=0				E=1			

Simplificar F(A,B,C,D,E)= ∑(2,3,5,8,9,10,12,13,14,16,18,20,21,22,23,25,26,30)

$$A\bar{C}\bar{E} + \bar{A}D\bar{E} + \bar{A}\bar{D}E + C\bar{D} + \bar{A}BD + A\bar{B}\bar{C}D$$

Ejemplos SIMPLIFICACIONES

Simplificar F(A,B,C,D,E)= ∑(2,3,5,8,9,10,12,13,14,16,18,20,21,22,23,25,26,30)



$$A\bar{C}\bar{E} + \bar{A}D\bar{E} + \bar{A}\bar{D}E + C\bar{D} + \bar{A}BD + A\bar{B}\bar{C}D$$

Ejemplos SIMPLIFICACIONES

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

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

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

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

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

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

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

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

Ejemplos – Ejercicio 1

Simplificación de SOPs y de POSs

Nº	A	B	C	D	F
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

Ejemplos – Ejercicio 1

1. Minterms y Maxterms

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	→	$(A\cdot B\cdot C\cdot D)$		

Ejemplos – Ejercicio 1

2. Karnaugh

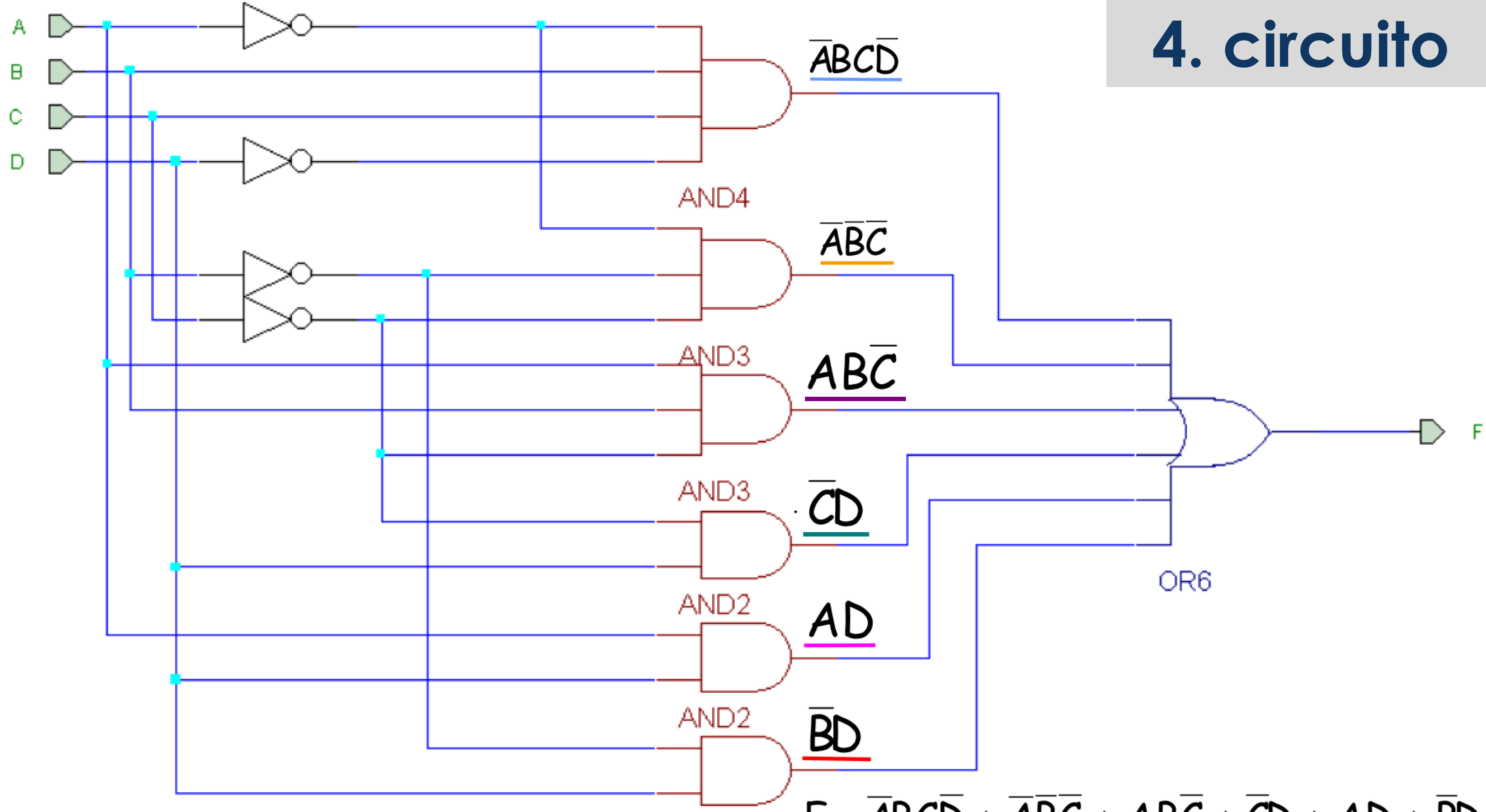
AB \ CD	00	01	11	10
00	1	1	1	
01		1		1
11	1	1	1	
10		1	1	

3. Simplificar

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

$$F = \overline{A}BC\overline{D} + \overline{A}\,\overline{B}\,\overline{C} + AB\overline{C} + \overline{C}D + AD + \overline{B}D$$

4. circuito



$$F = \bar{A}BC\bar{D} + \bar{A}\bar{B}\bar{C} + AB\bar{C} + \bar{C}D + AD + \bar{B}D$$

Síntesis de circuitos combinacionales

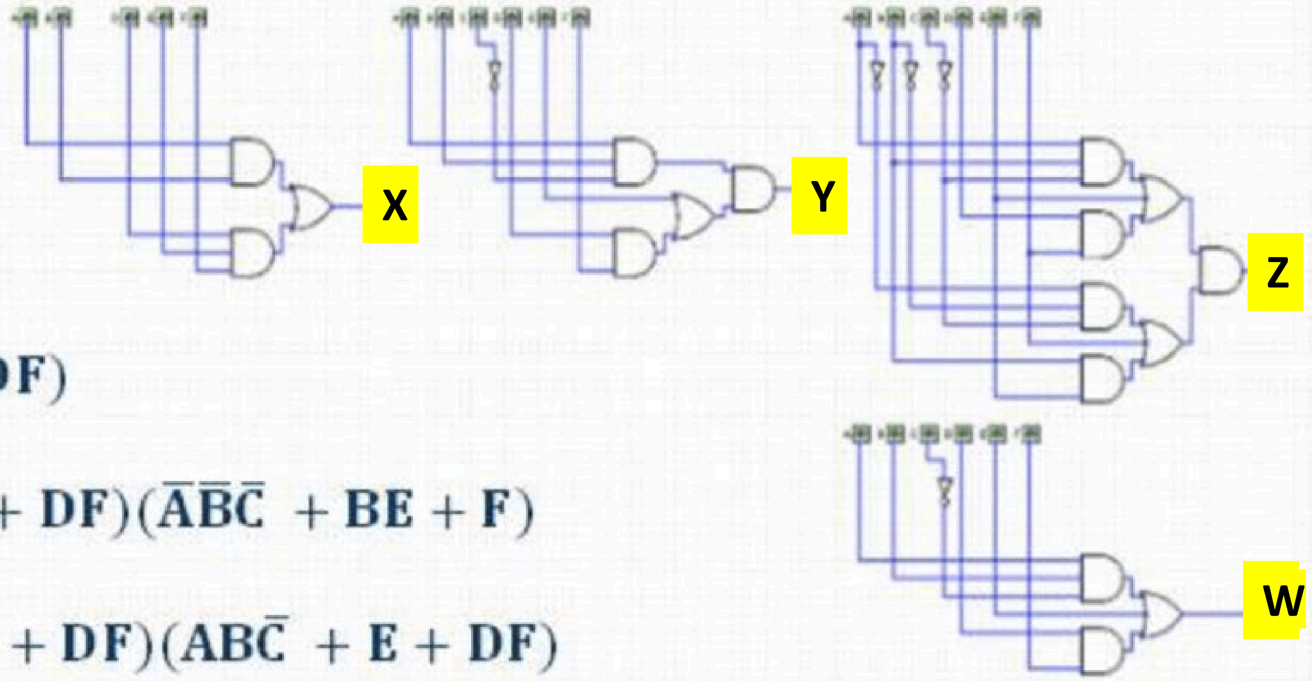
Obtención del circuito lógico a partir de una expresión booleana

- ¿Cual es el orden para realizar las operaciones en cualquier ecuación?

- Buscar los términos y las operaciones entre ellos.

- Ejemplos.

  - $X = AB + DEF$
  - $Y = AB\bar{C}\,(E + DF)$
  - $Z = (AB\bar{C} + E + DF)(\bar{A}\bar{B}\bar{C} + BE + F)$
  - $W = (AB\bar{C} + E + DF)(AB\bar{C} + E + DF)$



Síntesis de circuitos combinacionales

Implementación de la lógica combinacional

Obtención del circuito lógico a partir de la tabla de verdad (i)

- **¿Qué se pide?**
- **Identificar los términos de SoP´s y PoS´s.**
- **Si fuese una SoP´s, ¿Cuántos productos?**
- **Si fuese un PoS´s, ¿Cuántos sumas?**

A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
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	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	0

Síntesis de circuitos combinacionales

Implementación de la lógica combinacional

Obtención del circuito lógico a partir de la tabla de verdad (ii)

- ¿Qué se pide?
- Identificar los términos de SoP´s y PoS´s.
- Si fuese una SoP´s, ¿Cuántos productos?
- Si fuese un PoS´s, ¿Cuántos sumas?

A	B	C	D	X
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Síntesis de circuitos combinacionales

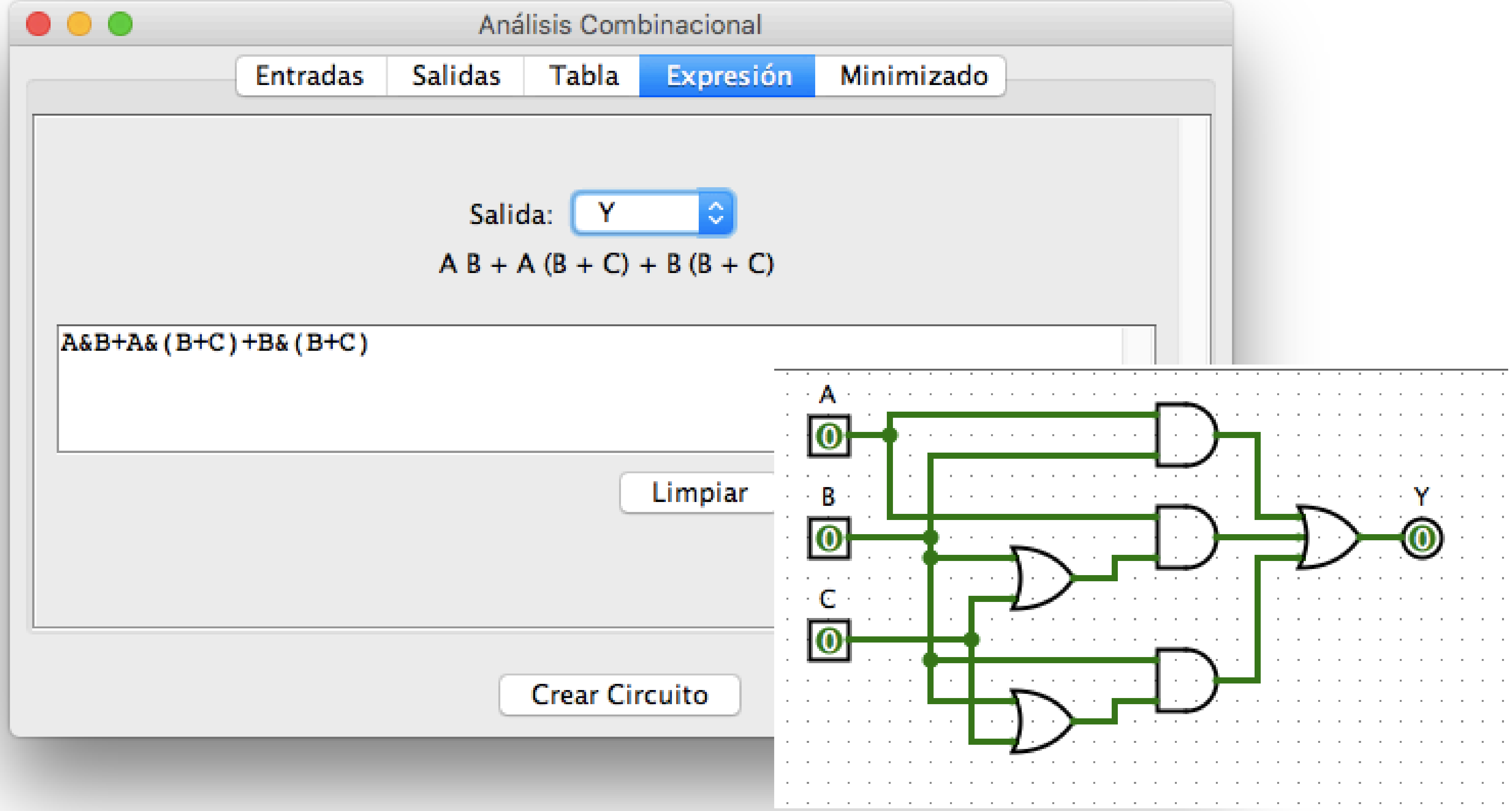
SI VAIS A USAR KARNAUGH . . .

NO SIEMPRE ES MEJOR ELEGIR EL QUE TIENE MENOS ELEMENTOS!!!

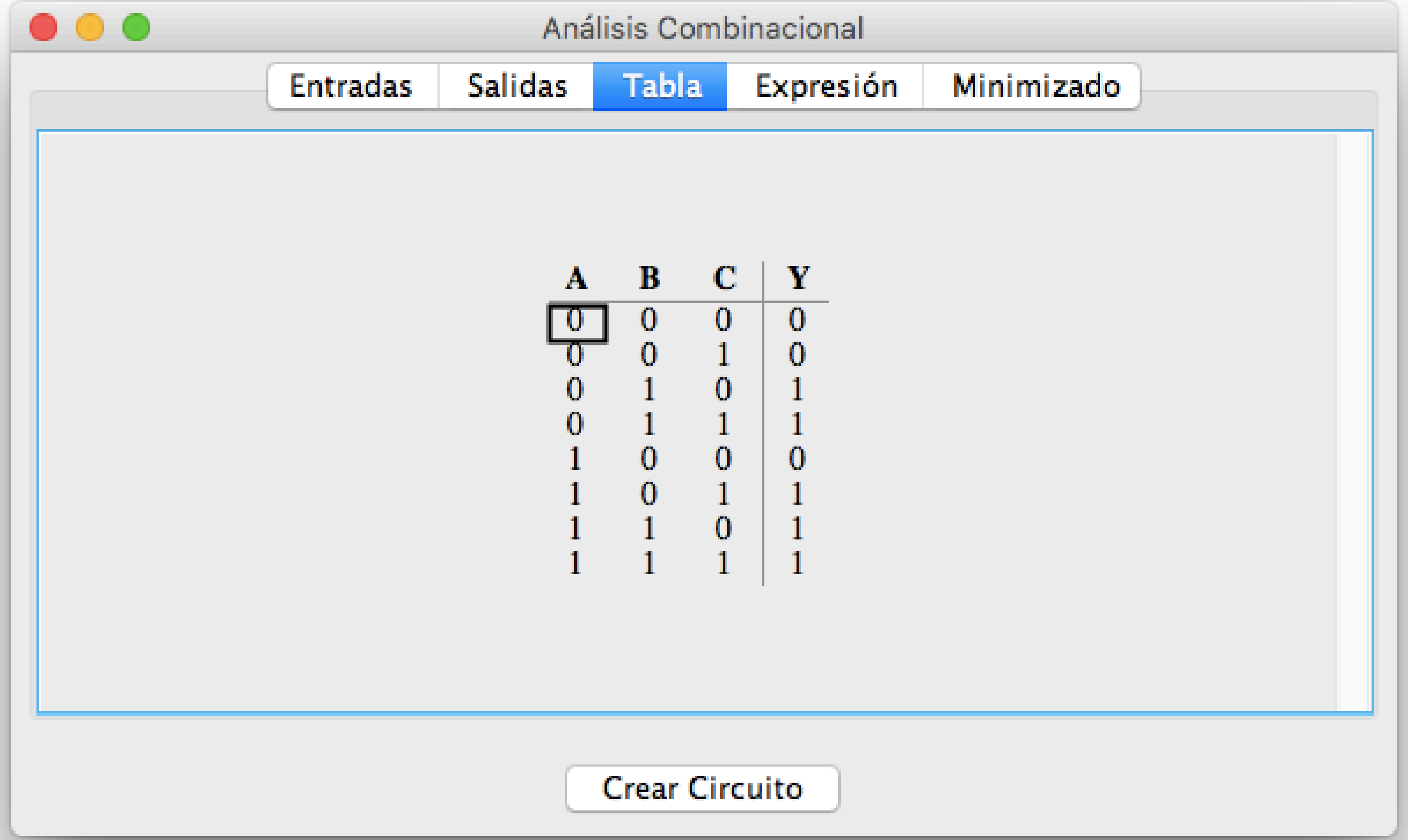
A	B	C	D	X
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

¿por qué?

Simplificación con Logisim

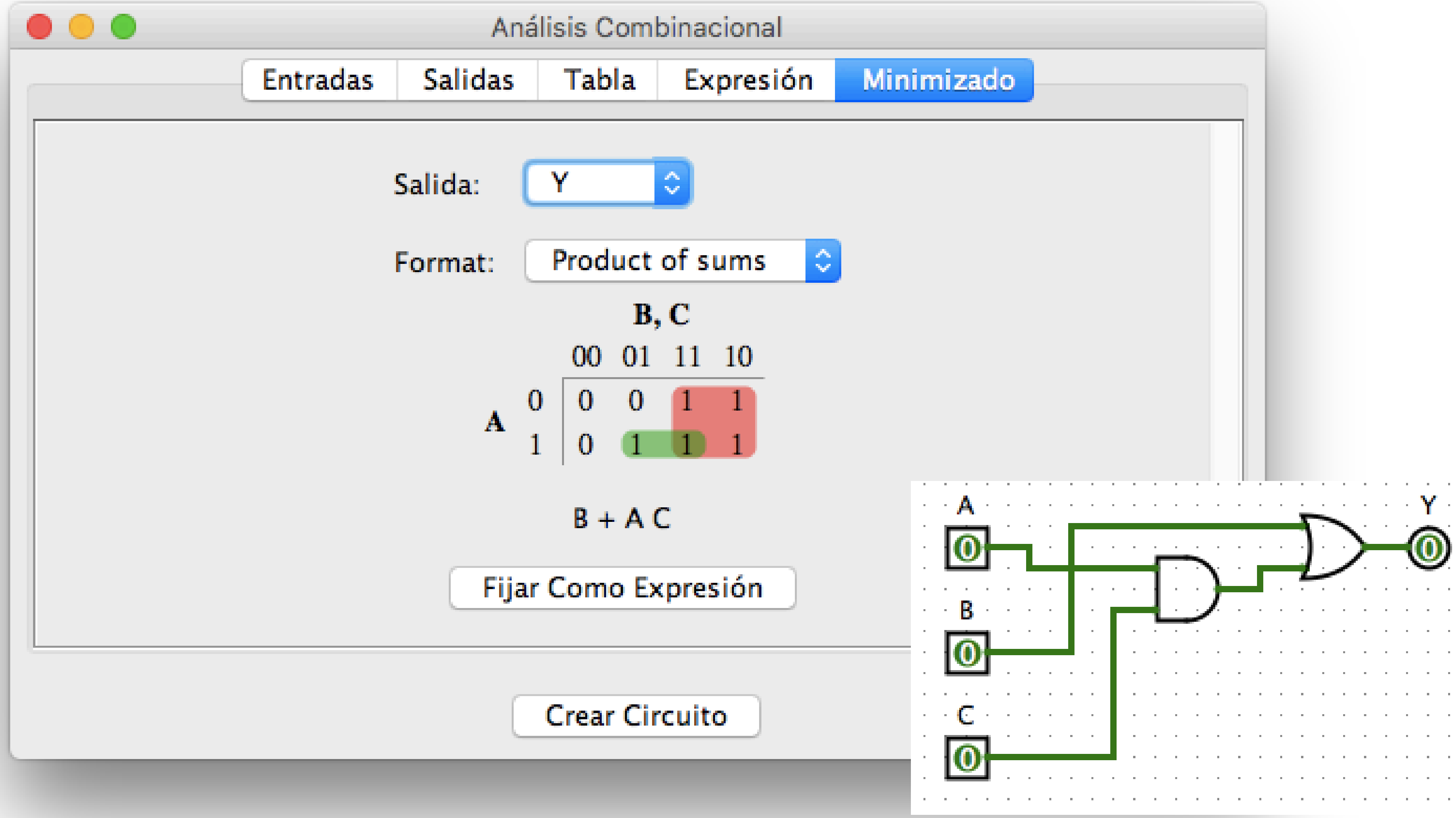


Simplificación con Logisim



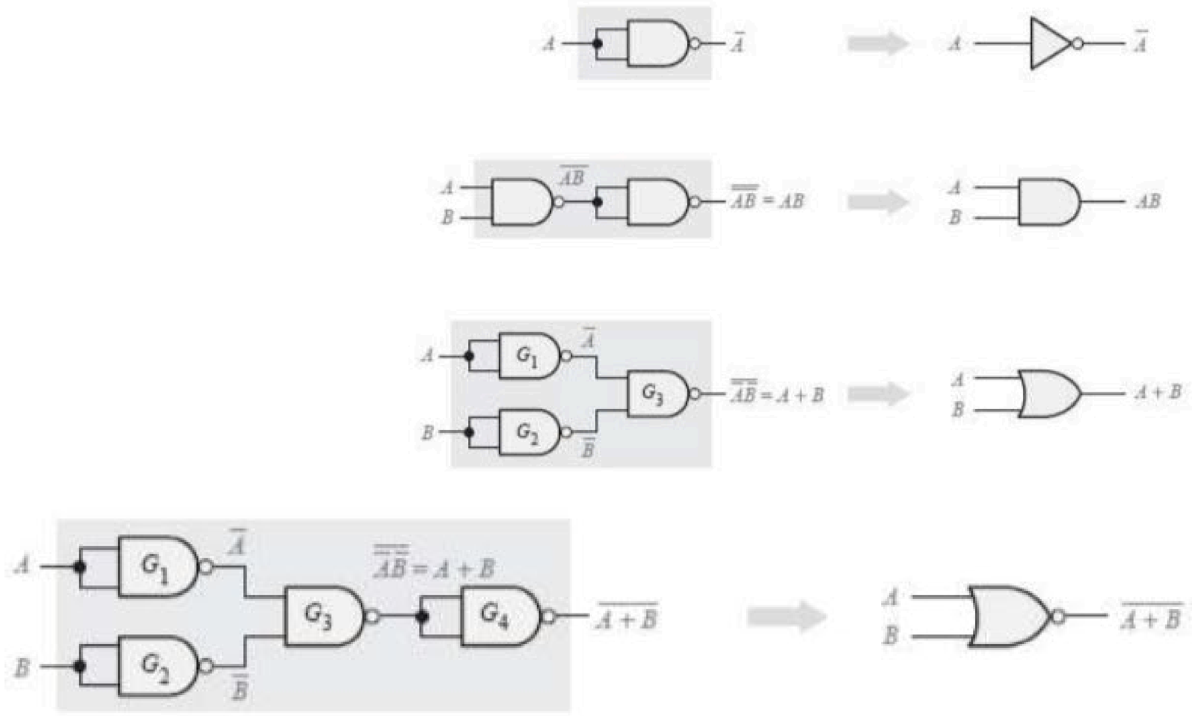
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Simplificación con Logisim

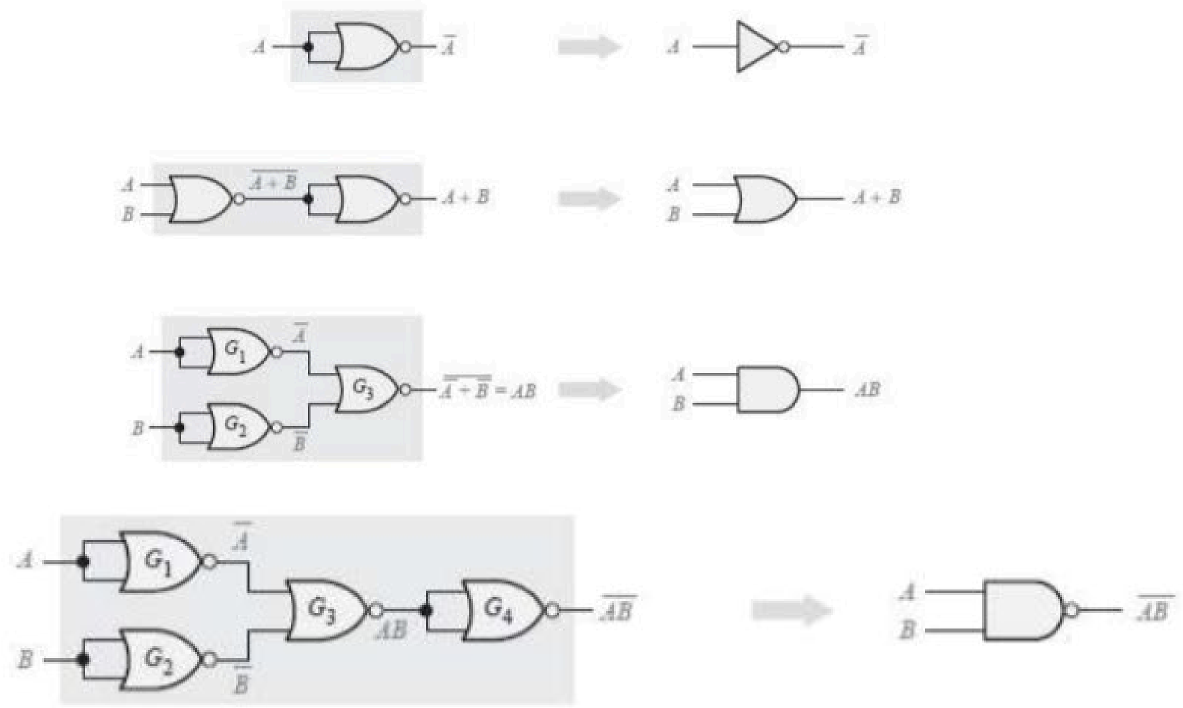


NAND y NOR: Conjuntos universales

❑La puerta NAND.



❑La puerta NOR.

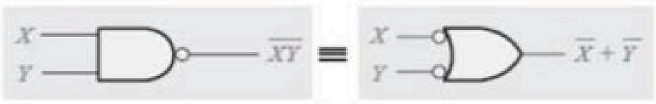


Síntesis con NAND y NOR

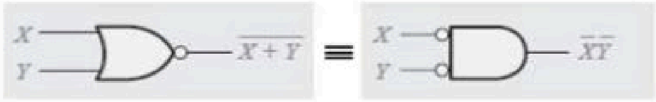
- **La idea:**
  Se trata de lograr implementar sólo con puertas NAND y NOR cualquier función.

- **Lo mejor:**
  Lograr implementar sólo con puertas NAND ó sólo con puertas NOR cualquier función.

- **¿Y como?, utilizando el Teorema DeMorgan.**
  - El complemento de un producto de variables, es igual a la suma de los complementos de las variables.



  - El complemento de una suma de variables, es igual al producto de los complementos de las variables.



Diseño con NAND

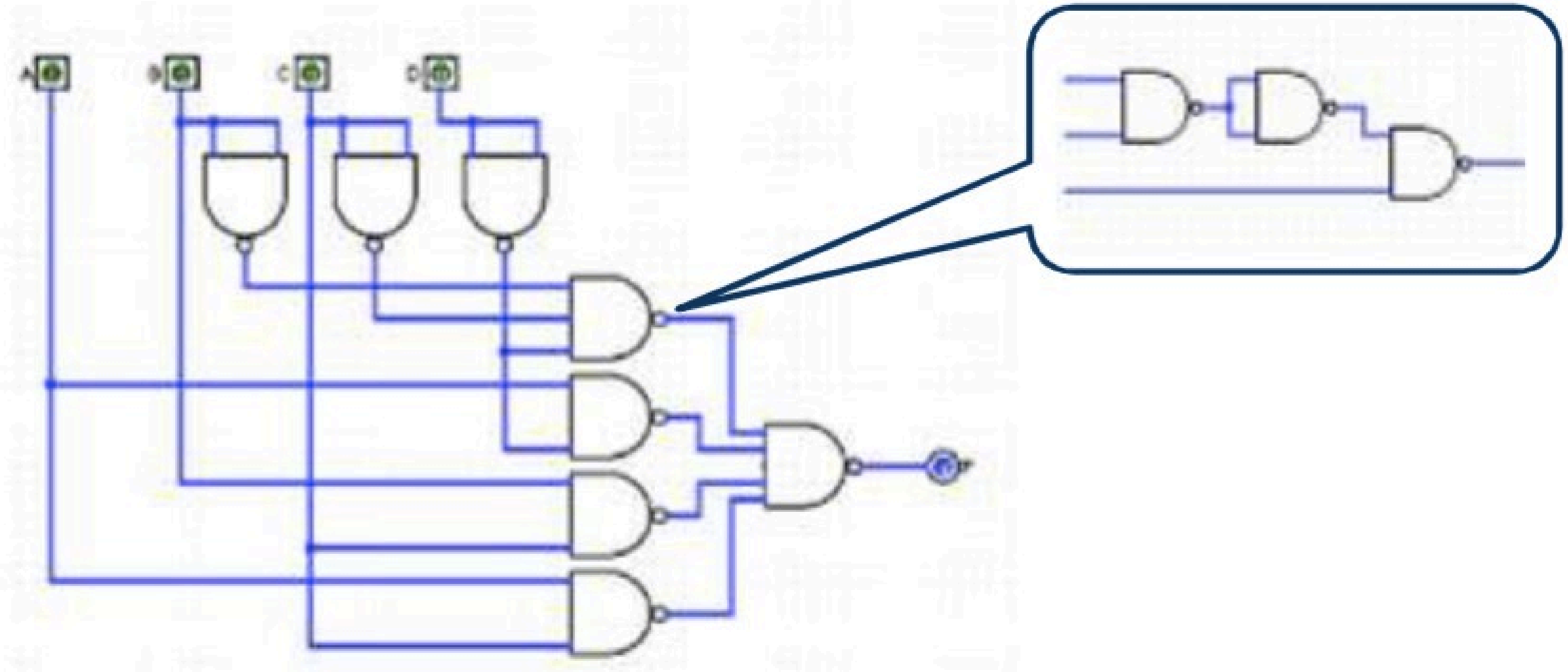
- **Diseñar un circuito que implemente la función booleana mostrada utilizado puertas NAND:**

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

Diseño con NAND

$$\mathbf{F(A,B,C,D) = \bar{B}\bar{C}\bar{D} + A\bar{D} + BC + AC}$$

$$F(A,B,C,D) = \overline{\overline{\bar{B}\bar{C}\bar{D}} \cdot \overline{A\bar{D}} \cdot \overline{BC} \cdot \overline{AC}}$$



Diseño con NOR

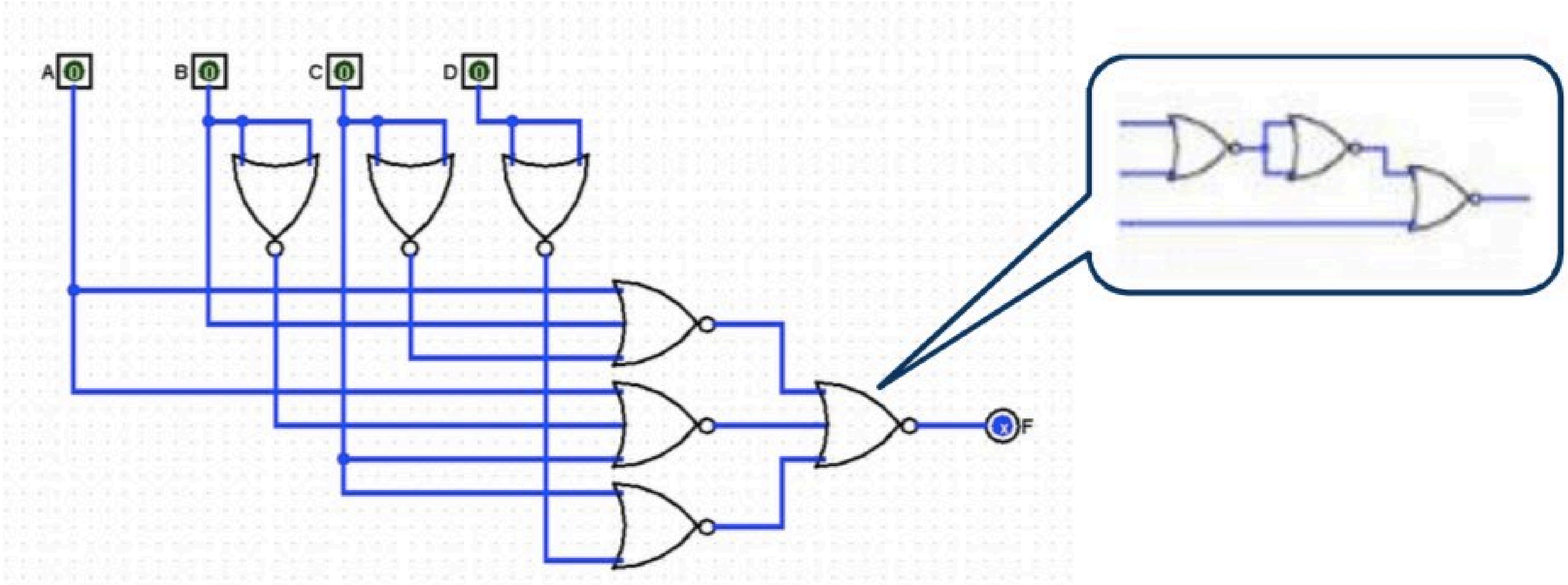
- Diseñar un circuito que implemente la función booleana mostrada utilizado puertas NOR:

$$F(A, B, C, D) = (A + B + \bar{C}) \cdot (A + \bar{B} + C) \cdot (C + \bar{D})$$

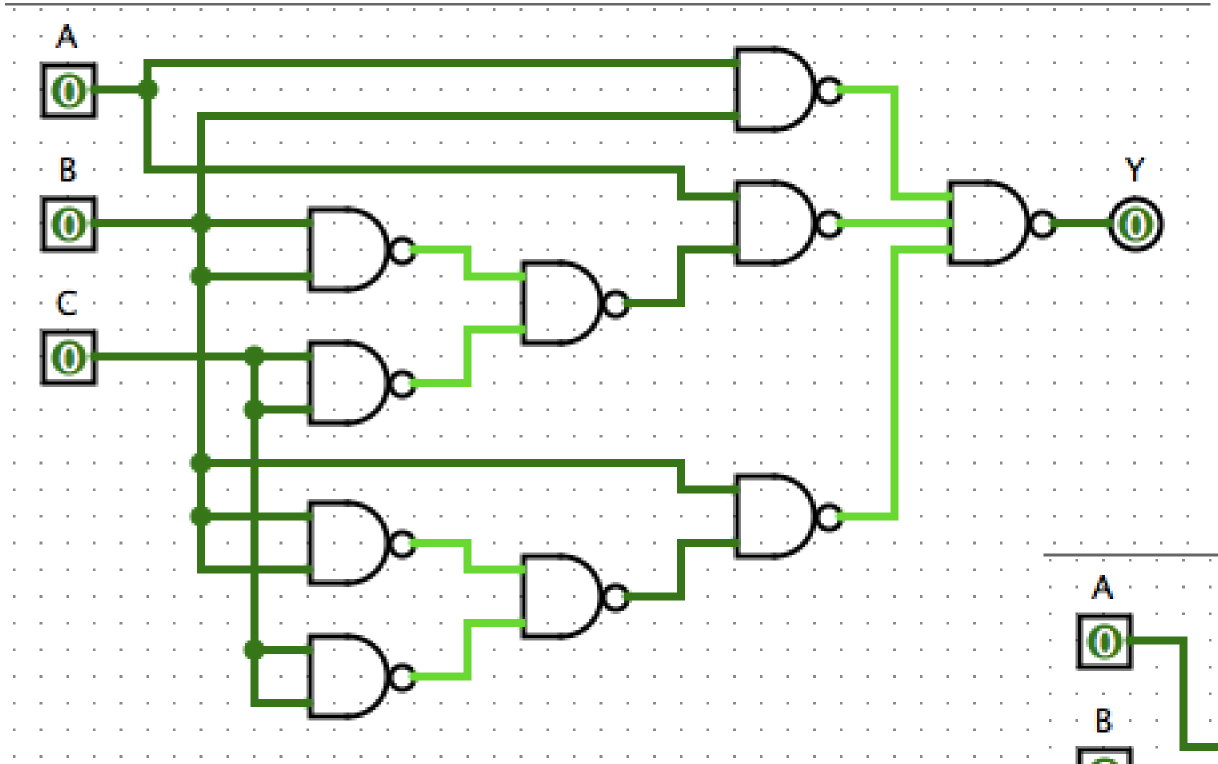
Diseño con NOR

$$\mathbf{F(A, B, C, D) = (A + B + \bar{C}) \cdot (A + \bar{B} + C) \cdot (C + \bar{D})}$$

$$F(A, B, C, D) = \overline{\overline{(A + B + \bar{C})} + \overline{(A + \bar{B} + C)} + \overline{(C + \bar{D})}}$$



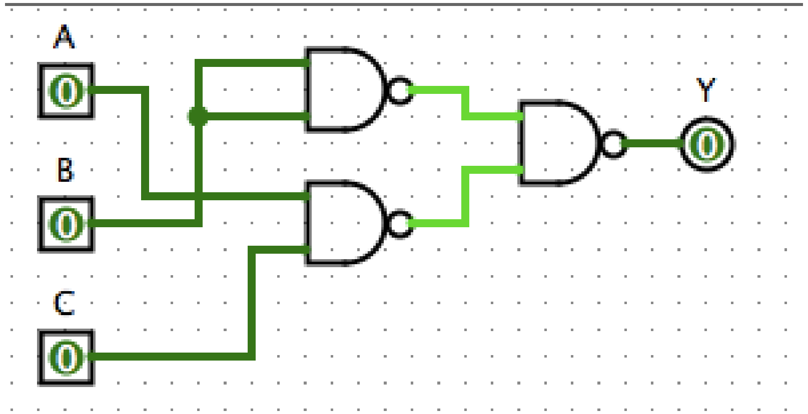
Diseño con NAND en Logisim



AB + A(B+C) +B(B+C)

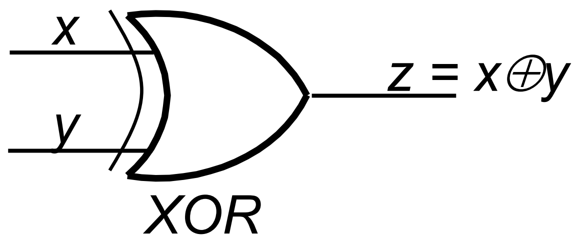
¿Ecuación con DeMorgan?

$\boxed{B + AC}$

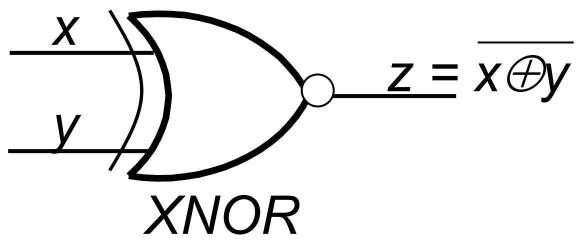


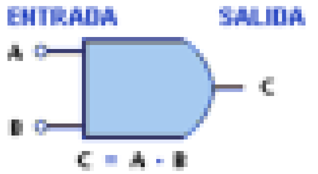
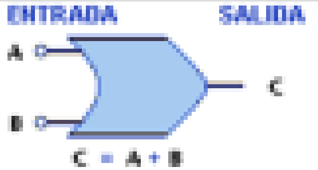
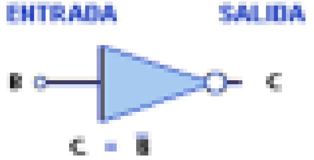
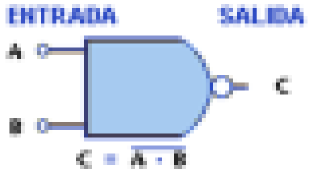
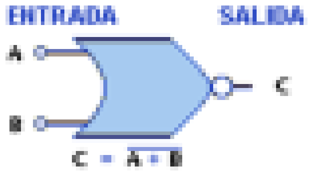
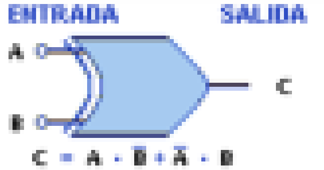
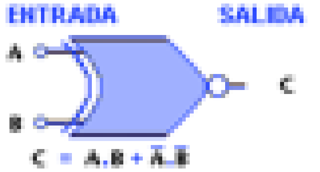
Otras puertas lógicas

- **Puertas XOR**:
  - Implementan la función OR-exclusiva, que devuelve ‘1’ sólo si una de las entradas vale ‘1’ y la otra ‘0’.



- **Puertas XNOR**:
  - Implementan la función complementaria a la XOR.

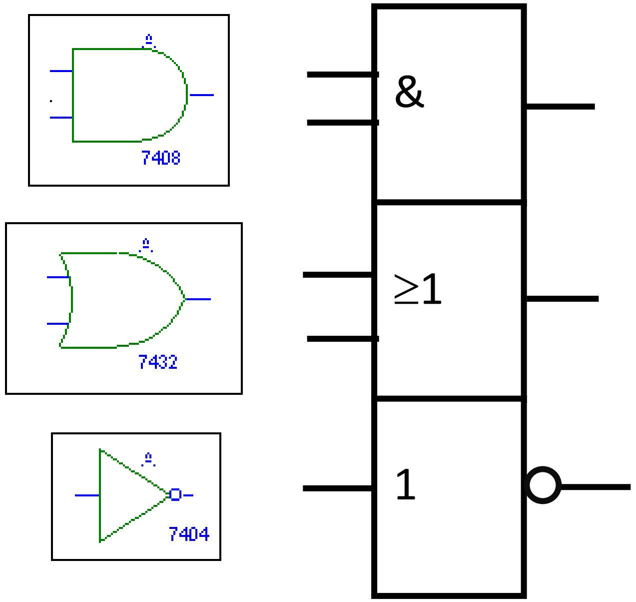


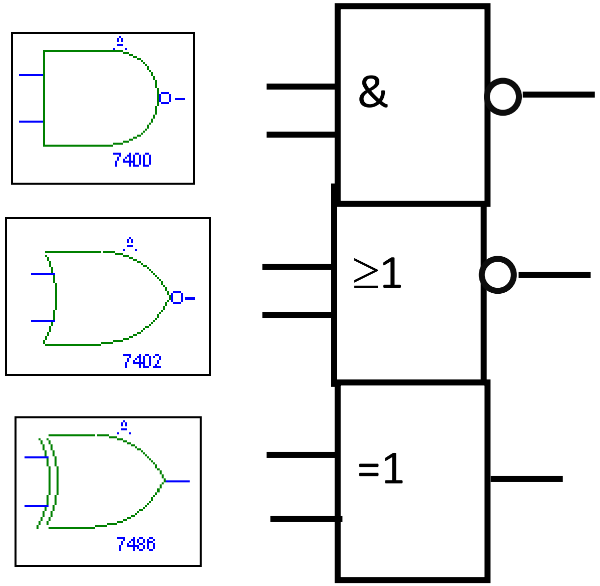
Nombre	Simbolo Grafico	Función Algebraica	Tabla de Verdad
AND		F = A.B	ENTRADA SALIDA A B C 0 0 0 0 1 0 1 0 0 1 1 1
OR		F = A + B	ENTRADA SALIDA A B C 0 0 0 0 1 1 1 0 1 1 1 1
INVERSOR		F = B'	ENTRADA SALIDA B C 0 1 1 0
NAND		F = (A.B)'	ENTRADA SALIDA A B C 0 0 1 0 1 1 1 0 1 1 1 0
NOR		F = (A + B)'	ENTRADA SALIDA A B C 0 0 1 0 1 0 1 0 0 1 1 0
XOR		F = A.B' + A'.B	ENTRADA SALIDA A B C 0 0 0 0 1 1 1 0 1 1 1 0
XNOR		F = A.B + A' B'	ENTRADA SALIDA A B C 0 0 1 0 1 0 1 0 0 1 1 1

Resumen de símbolos de puertas lógicas

Estándar ANSI/IEEE 91-1984:

Standard Graphic Symbools for Logic Functions



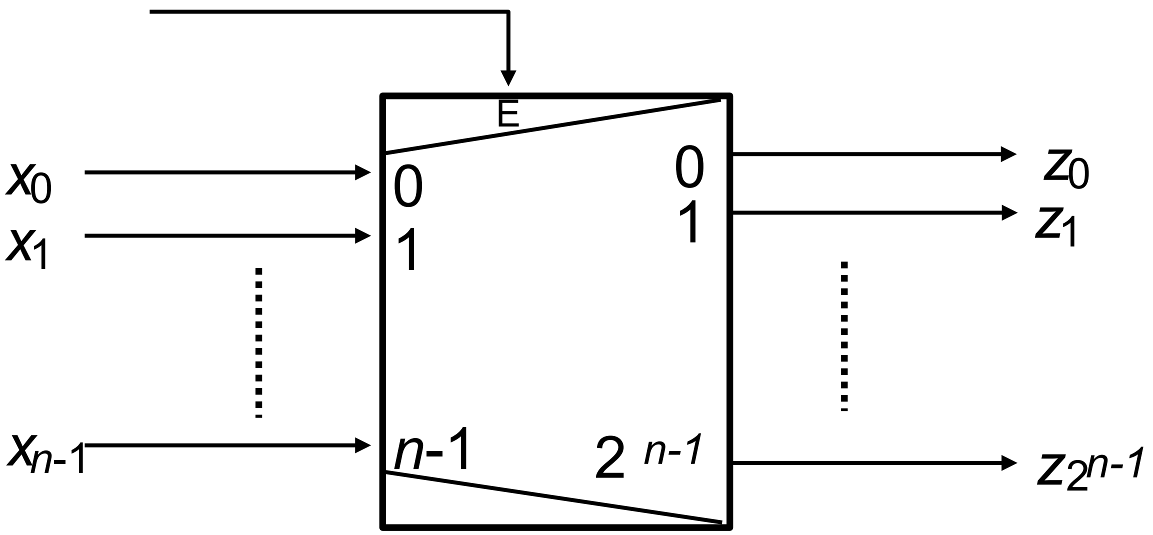


ANSI: American National Standards Institute

IEEE: Institute of Electrical and Electronic Engineers

Otros componentes combinacionales

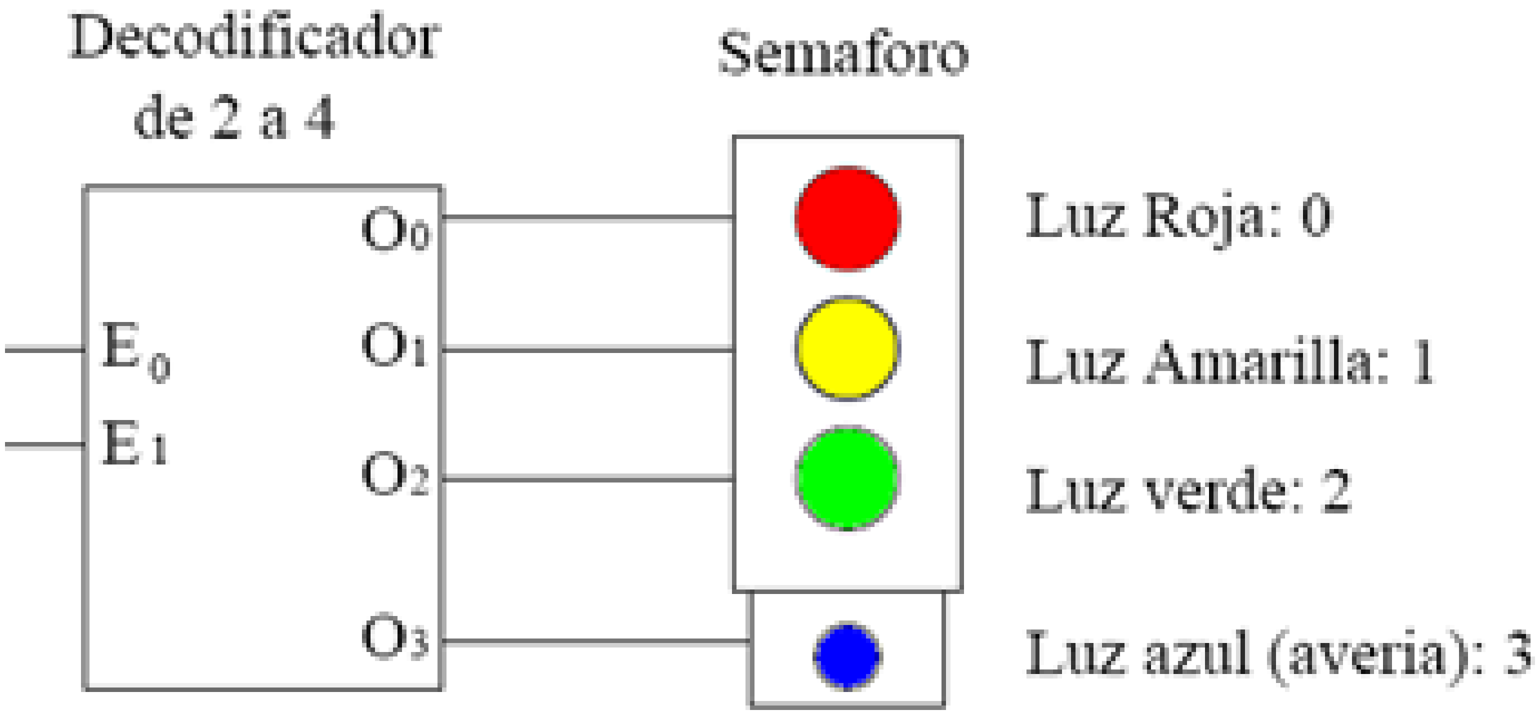
- **Decodificadores de N entradas (n entradas y 2^n salidas)**
  - Activan una **única** de las 2^N salidas en cada momento, dependiendo del valor de las entradas de datos (x_{n-1}, … x_0) y de la entrada de capacitación (E, enable)
  - Asumiendo que todas las entradas y las salidas se activan a '1', entonces el comportamiento es el siguiente:
  - $z_i = 1$ si E=1 y $X = i$; $z_i = 0$ en cualquier otro caso.



¿Encoders de prioridad?

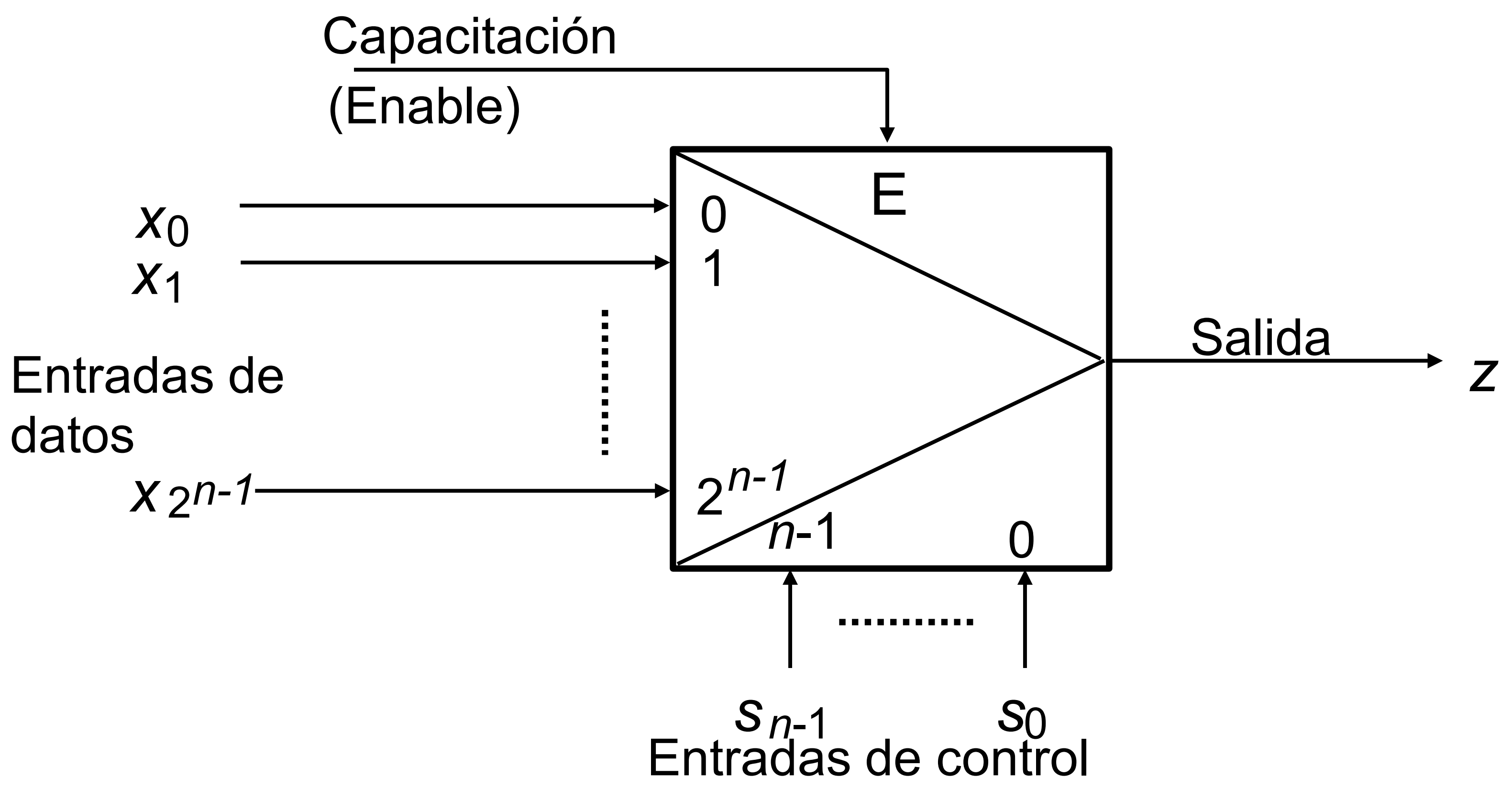
Otros componentes combinacionales

- **Decodificadores de N entradas**



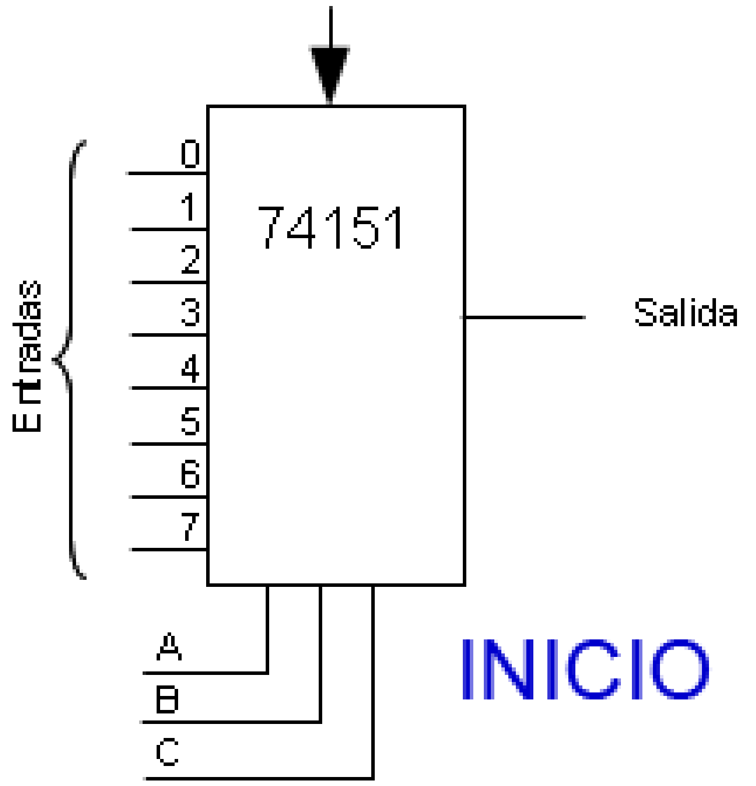
Otros componente combinacionales

- **Multiplexores de 2^N entradas**
  - Disponen de una única salida, que toma en cada momento el valor de la entrada de datos cuyo número se selecciona mediante las entras de control (siempre que la señal E esté activa)



Otros componente combinacionales

- **Multiplexores de 2^N entradas**



INICIO

Implementación de circuitos combinacionales con decodificadores

- Los decodificadores tienen la propiedad de que cada una de las salidas representa un punto de la función.
- Si todas las señales se activan a '1', entonces cada salida representa un producto canónico.
- Por ejemplo, para un decodificador de 3 entradas (x, y, z)
  - La salida 0 representa el $\overline{x}\cdot\overline{y}\cdot\overline{z}$
  - La salida 6 representa el $x\cdot y\cdot\overline{z}$
- Es muy sencillo, por lo tanto, implementar un circuito combinacional a partir de una EC en forma normal disyuntiva (o suma de productos canónica): basta añadir puertas OR que se conecten a las salidas del decodificador que representan los productos canónicos a sumar.

Implementación de circuitos combinacionales con multiplexores

- Pueden utilizarse multiplexores para implementar circuitos combinacionales de manera muy sencilla:
  - Conectamos las entradas del sistema a las entradas de control del multiplexor
  - Conectamos cada una de las entradas de datos del multiplexor a ‘0’ o ‘1’, respetando la tabla de verdad de la salida del sistema.
- Los multiplexores también constituyen un conjunto universal de elementos, dado que puede implementarse cualquier circuito utilizando únicamente multiplexores.

Valores metalógicos

- Si bien un sistema digital binario tiene dos valores de trabajo establecidos: 0 y 1, podemos encontrar estos dos valores adicionales:
  - X: Cortocircuito
  - Z: Alta impedancia (equivalente a circuito abierto)

  - El valor Z se usa frecuentemente para compartir líneas de comunicaciones (buses) entre varios dispositivos lógicos.