Tema I

EXIGENCIAS COMPUTACIONALES DEL PROCESAMIENTO DIGITAL DE LA INFORMACION

Tutor: Manuel Fernández Barcell
Centro asociado de Cádiz
http://prof.mfbarcell.es

TEMA 1: EXIGENCIAS COMPUTACIONALES DEL PROCESAMIENTO DIGITAL DE LA INFORMACIÓN

· Contexto
· Conocimiento Previo Necesario
· Objetivos del Tema
· Guía de Estudio
· Contenido del Tema
1.1. Procesamiento Digital de la Información
1.2. Funciones Combinacionales y Secuenciales Necesarias
1.3. Variables y Operadores Lógicos: Algebra de Boole
1.4. Funciones Lógicas: Formas Canonicas
1.4.1. Forma Normal Disyuntiva
1.4.2. Forma Normal Conjuntiva
1.5. Otras Representaciones Completas (NAND, NOR)
1.6. Análisis y Síntesis
1.7. Introducción a la Minimización
1.8. Problemas
· Preparación de la Evaluación
· Referencias Bibliográficas

Objetivo 1: *Distinguir claramente entre el procesado analógico y digital, entendiendo que son dos formas diferentes de representar la información (los datos) y de operar con ellos para obtener otros datos.*

Objetivo 2: *Conocer los postulados y teoremas básicos del Álgebra de Boole y saber demostrar los teoremas.*

Objetivo 3: *Saber representar funciones lógicas usando distintos tipos de operadores (AND, OR, NOT; sólo NAND; sólo NOR) y saber pasar de una representación a otra. Por ejemplo, de (AND, OR, NOT) a sólo NAND o de NAND a NOR o de términos mínimos (suma de productos) a términos máximos (productos de sumas).*

Objetivo 4: *Saber analizar un circuito lógico. Es decir, saber pasar del esquema de un circuito a la expresión o expresiones lógicas que enlazan las variables de entrada al circuito con las variables de salida.*

Objetivo 5: *Saber sintetizar un circuito lógico. Es decir: (a) saber pasar de un conjunto de especificaciones funcionales a una tabla de verdad, (b) de una tabla de verdad a una o más funciones lógicas y (c) de estas funciones al circuito que las satisface.*

Objetivo 6: *Saber minimizar funciones lógicas. Es decir, dada una cierta función lógica, encontrar otra equivalente (con la misma tabla de verdad) pero con menos términos o con términos con menos variables. Este objetivo está enlazado con el objetivo 2, porque el proceso de minimización se basa en el uso adecuado y repetido de los postulados y teoremas del Álgebra de Boole. Sin embargo, para alcanzarlo es conveniente usar el conocimiento adicional del método de Karnaugh que recoloca los distintos términos de forma tal que hace evidente el proceso de minimización para funciones de hasta 4 ó 5 variables.*

1.1 Procesamiento digital de la información

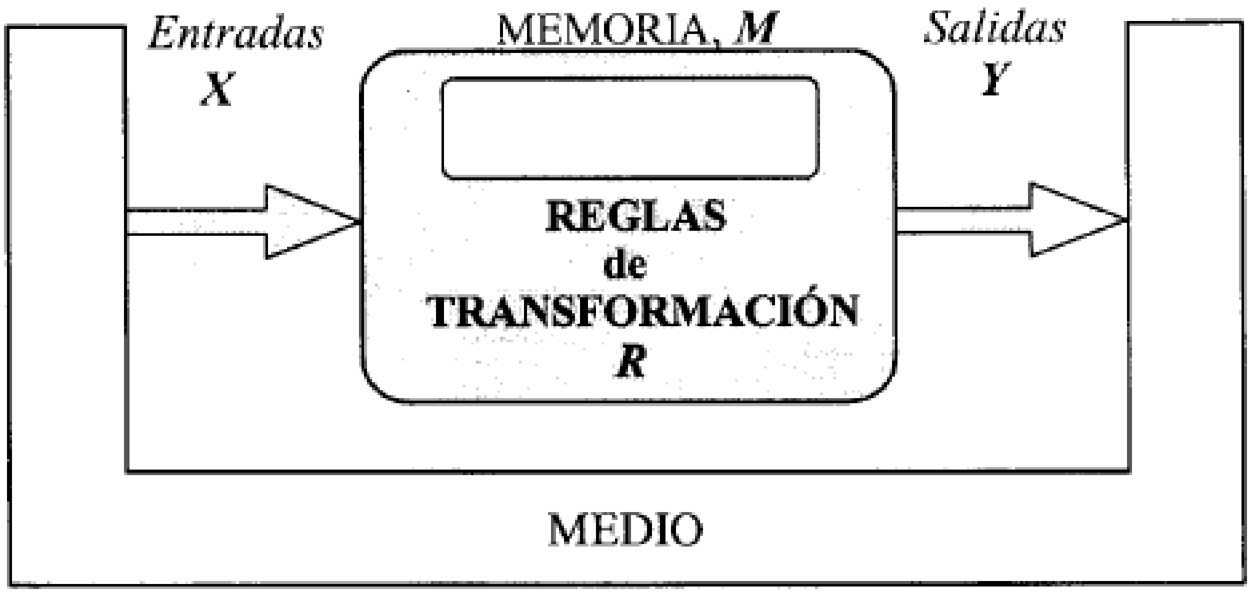


Figura 1.1. Modelo computacional básico.

Computación analógica y digital

$$R_A^1 : y(t) = A \cdot x(t)$$

$$R_A^2 : y(t) = A_1 \cdot x_1(t) + A_2 \cdot x_2(t)$$

$$R_A^3 : y(t) = A_1 \cdot x_1(t) + B \int x_2(t) \cdot dt + C \frac{dx_3(t)}{dt}$$

Expresión de computación analógica: operaciones de multiplicación, integración etc.

Expresiones de computación digital: sumas lógicas, productos lógicos, etc

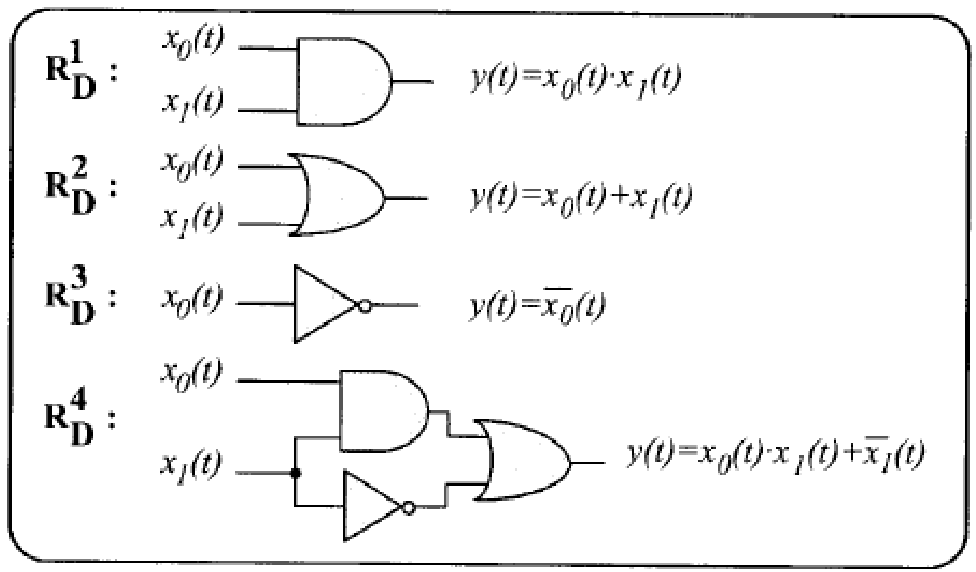


Figura 1.2. Expresiones de computación digital.

Operadores digitales representados con tablas de verdad

x_1	x_0	$\overline{x_1}$	$x_1 \cdot x_0$	$x_1 + x_0$	$x_1 \cdot x_0 + \overline{x_1}$
0	0	1	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	1	1	1

Figura 1.3. Tablas de verdad que especifican en extenso los operadores lógicos usados en el ejemplo a partir de las cuatro configuraciones de valores posibles en las dos variables de entrada, (x_0, x_1).

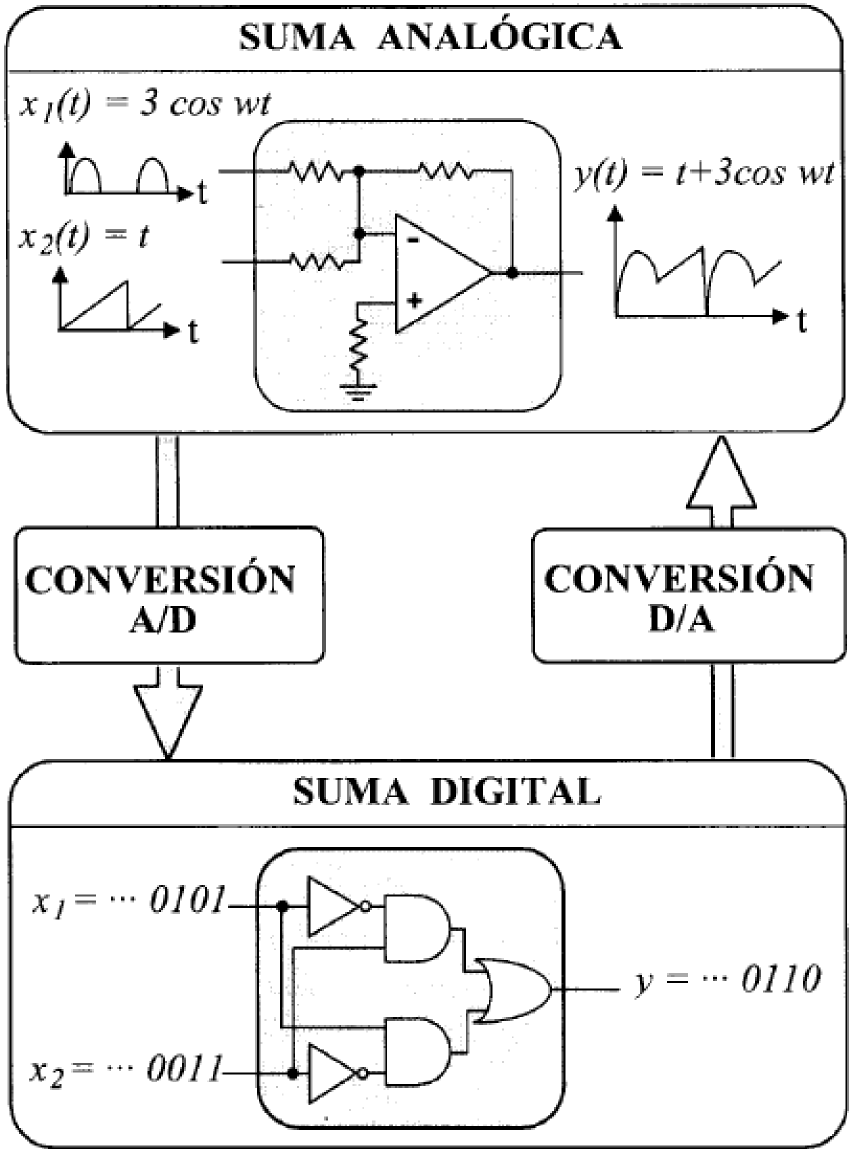


Figura 1.4. Ilustración cualitativa de la suma analógica comparada con la suma digital

1.2 Funciones combinacionales y secuenciales necesarias

1. *Descripción de la computación en lenguaje natural de forma clara, completa, precisa e inequívoca.*
2. *Traslación de esa descripción a un conjunto de especificaciones funcionales en un lenguaje lógico formal.*
3. *Reescritura de esa descripción formal en términos del modelo computacional* de la *figura 1.1.* Es decir, *en términos de entradas, estados de memoria, salidas y reglas de transformación,* $\{R_D\}$, que representan la dinámica interna de la computación especificando cómo se producen el nuevo estado y las salidas a partir de las entradas y del estado anterior.
4. *Síntesis modular del sistema en términos de un conjunto completo de operadores mínimos.*

Todas las funciones necesarias para el procesamiento digital de la información, es decir todas las reglas R_D, pueden incluirse en dos grandes apartados:

a) *Funciones Combinacionales.*

b) *Funciones Secuenciales.*

Son funciones de *lógica combinacional* todas aquellas funciones en las que para obtener el valor de la salida en un cierto instante sólo necesitamos conocer el valor de las entradas en ese mismo instante. Son funciones de decisión, sin "memoria". El modelo matemático soporte de esta parte de la electrónica digital es el *Álgebra de Boole*, de la que más adelante incluiremos un resumen. Ejemplos de este tipo de funciones son entre otras:

a.1. *Operaciones aritmético-lógicas.*

a.2. *Funciones de ruta de datos: multiplexos y demultiplexos.*

a.3. *Circuitos cambiadores de código.*

Funciones secuenciales: La salidas no dependen solo de las entradas, también del estado

Modelo matemático: Teoría de autómatas

Representar, analizar y sintetizar

¿Qué es representar?. Representar una función lógica combinacional es encontrar un procedimiento para *describir de forma completa* la función. Sea cual fuere la configuración de valores en las variables de entrada, la representación debe permitir conocer el valor de la salida. Hay esencialmente dos formas de representación: en *extenso* y en *intenso*. Tabla de verdad y función lógica

¿Qué es analizar?. Analizar un circuito en lógica combinacional es encontrar la representación de las funciones lógicas que lo describen.

¿Qué es sintetizar?. Es el *proceso inverso al de análisis*. Partimos ahora de una función lógica, $y=f(x_1,x_2)$, ó de un conjunto de N funciones lógicas de M variables, $y_k(t)=f_k[x_1(t),x_2(t),\ldots,x_M(t)]$, $k=1, 2, \ldots, N$, y buscamos el circuito que realiza físicamente a esa función de forma que reproduce la misma relación entre las variables de entrada y las de salida para todas y cada una de las combinaciones de valores posibles.

1.3 Variables y operadores lógicos. Algebra de Boole

- Definición del Algebra de Boole
  - George Boole (1854) desarrolló una herramienta matemática que se utiliza para el estudio de computadores.
  - Es una estructura matemática que se construye a partir de un conjunto de elementos sobre los que se definen unos operadores que permiten realizar operaciones en ellos, estableciendo unos postulados o axiomas que relacionan tanto al conjunto de elementos como al conjunto de operadores
  - Para la construcción de un álgebra de Boole, se parte de una estructura algebraica (B, +, ·), formada por un conjunto de elementos B y dos operaciones definidas en el mismo, denominadas + y · (suma y producto).
  - Operaciones:
    - + (suma lógica); * (producto lógico); complementación
  - La aplicación en computadores es del tipo binario ⇒ 0/1
    - El estado de un elemento del circuito lógico viene representado por una variable que puede valer “1” o “0”.

Postulados

- Se dice que es un álgebra de Boole si cumple los siguientes axiomas, también conocidos como **postulados de Huntington**.

P.1. *Las operaciones (+ y ·) son operaciones cerradas*. El resultado de aplicarlas a cualesquiera de las variables del conjunto, producirá variables del conjunto.

P.1.a: Si $X \in B$ é $Y \in B$, entonces $(X+Y) \in B$
P.1.b: Si $X \in B$ é $Y \in B$, entonces $(X \cdot Y) \in B$

[1.3]

P.2. *Existen elementos neutros para ambas operaciones* ("0" para la suma y "1" para el producto).

P.2.a: $X + 0 = X$
P.2.b: $X \cdot 1 = X$

[1.4]

Postulados algebra de Boole

P.3. *Ambas operaciones son conmutativas*:

P.3.a: $X + Y = Y + X$

P.3.b: $X \cdot Y = Y \cdot X$

[1.5]

P.4. *Ambas operaciones son distributivas*, una respecto de la otra:

P.4.a: $X + Y \cdot Z = (X + Y) \cdot (X + Z)$ (La suma es *distributiva* respecto al producto)

P.4.b: $X (Y + Z) = X \cdot Y + X \cdot Z$ (El producto es *distributivo* respecto a la suma)

[1.6]

P.5. *Complementariedad*. Para cada variable X existe su complementaria $\overline{X}$ y entre ambas se cumplen las siguientes condiciones:

P.5.a: $X + \overline{X} = 1$

P.5.b: $X \cdot \overline{X} = 0$

[1.7]

Tablas de verdad

•Se representa el valor que toma la función para cada una de las posibles combinaciones de sus variables
•Recoge todas las combinaciones de las variables de entrada y los valores que toman las salidas..

a	b	c		*b+c*	*a*(b+c)*	*a*b*	*a*c*	*(a*b)+(a*c)*
0	0	0	→	0	0	0	0	0
0	0	1	→	1	0	0	0	0
0	1	0	→	1	0	0	0	0
0	1	1	→	1	0	0	0	0
1	0	0	→	0	0	0	0	0
1	0	1	→	1	1	0	1	1
1	1	0	→	1	1	1	0	1
1	1	1	→	1	1	1	1	1

Ejemplo. Demostrar las leyes de De Morgan mediante las tablas de verdad, para funciones de dos variables. (a + b)’ = a’ * b’

a	b		*a+b*	*(a+b)’*	*a’*	*b’*	*a’ * b’*
0	0	→	0	**1**	1	1	**1**
0	1	→	1	**0**	1	0	**0**
1	0	→	1	**0**	0	1	**0**
1	1	→	1	**0**	0	0	**0**

Demostración de postulados por inducción completa

X	*Y*	*Z*	*Y+Z*	*X(Y+Z)*	*XY*	*XZ*	*XY+XZ*
0	0	0	0	***0***	0	0	***0***
0	0	1	1	***0***	0	0	***0***
0	1	0	1	***0***	0	0	***0***
0	1	1	1	***0***	0	0	***0***
1	0	0	0	***0***	0	0	***0***
1	0	1	1	***1***	0	1	***1***
1	1	0	1	***1***	1	0	***1***
1	1	1	1	***1***	1	1	***1***

son iguales

Figura 1.5. Tablas de verdad para demostrar la distributividad del producto respecto de la suma.

Demostración mediante diagramas de Venm

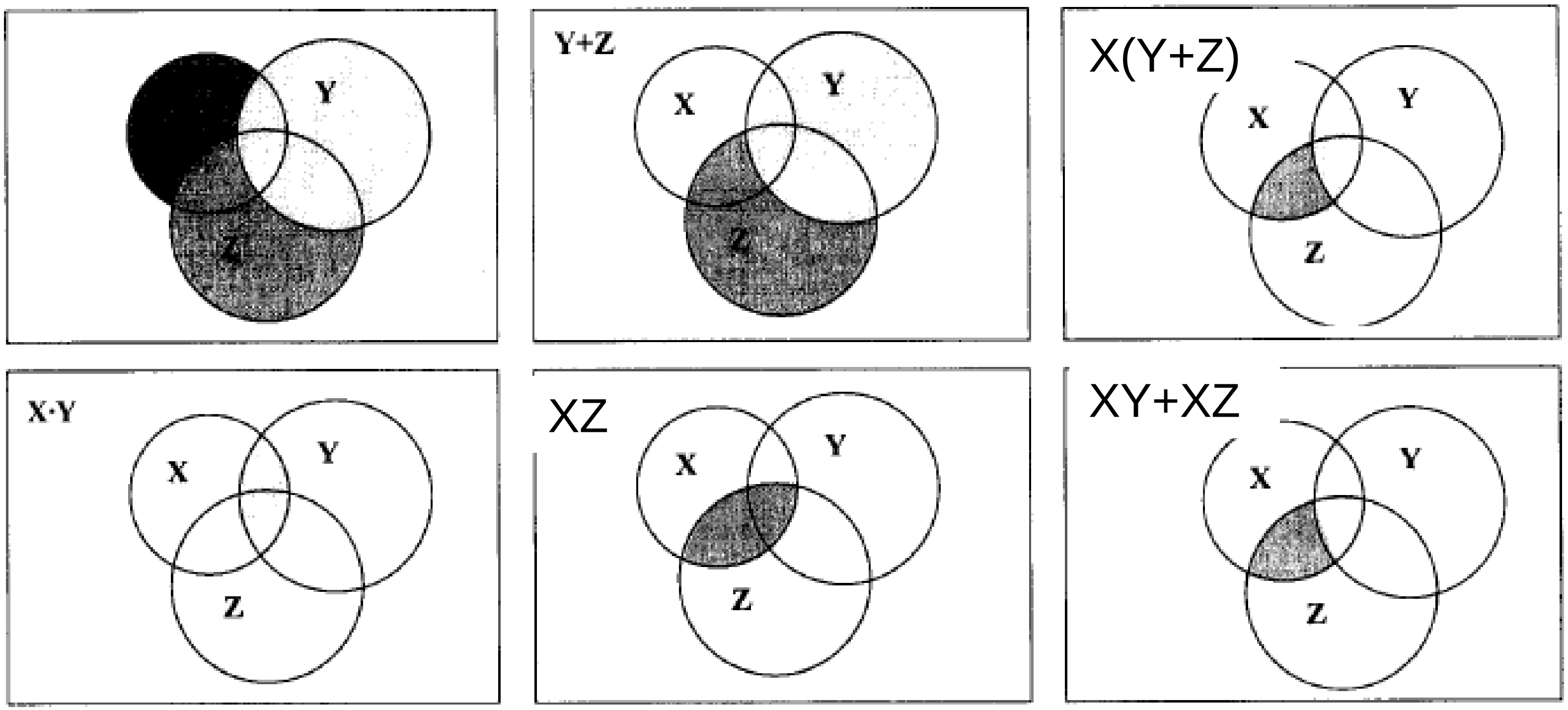


Figura 1.8. Demostración de distributividad del producto respecto de la suma mediante diagramas de Venn.

Teoremas del Algebra de Boole

Teoremas		
T.1: Doble complementación	$\overline{\overline{X}} = X$	Ley de involución
T.2: Idempotencia	$X+X=X$ $X \cdot X = X$	
T.3: Absorción	$X+X \cdot Y = X$ $X \cdot (X+Y) = X$	
T.4: Adyacencia	$X \cdot Y + X \cdot \overline{Y} = X$ $(X+Y) \cdot (X+\overline{Y}) = X$	
T.5: Teoremas de DeMorgan:	$\overline{X+Y} = \overline{X} \cdot \overline{Y}$ $\overline{X \cdot Y} = \overline{X} + \overline{Y}$	

Leyes y Teoremas del Algebra de Boole.

Principio de dualidad.

- Cada identidad deducida de los anteriores axiomas, permanecerá valida si los elementos 0 y 1 y los operadores + y * se cambian entre sí.

Ley de idempotencia. Para cualquier elemento del algebra de boole se verifica que: $\forall\, a \in B$:

$$\left.\begin{aligned} a * a &= a \\ a + a &= a \\ a * 0 &= 0 \\ a + 1 &= 1 \end{aligned}\right\}$$

Estas 2 propiedades, junto con el postulado de existencia de elemento neutro, definen la suma y el producto lógico.

a	**+**	**b**	**S**
0	+	0	0
0	+	1	1
1	+	0	1
1	+	1	1

a	*****	**b**	**S**
0	*	0	0
0	*	1	0
1	*	0	0
1	*	1	1

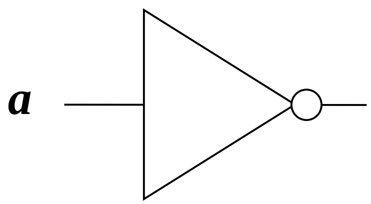
Teorema de expansión de Shannon. Toda función se puede descomponer

f(a, b, c, …) = **a** * f(**1** + b + c + …) + **a’** * f(**0** + b + c + …)

Siendo f (1, b, c, …) la función resultante de sustituir, en la función original, todas las ***a*** por 1, y las ***a’*** por 0. El segundo término, f (0, b, c, …) es la función resultante de sustituir las ***a’*** por 0 y las ***a*** por 1.

Puertas básicas

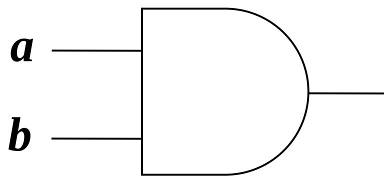
Inversor o Negador (NOT):



$f = a'$

NOT	
a	*f*
0	**1**
1	**0**

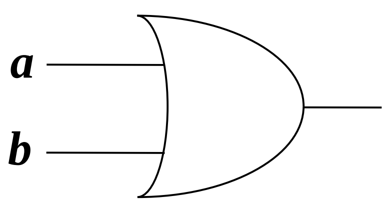
Puerta AND



$f = a \cdot b$

AND		
a	b	*f*
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

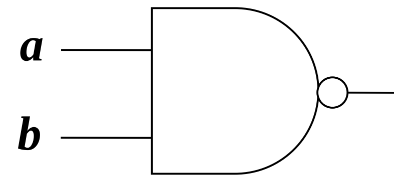
Puerta OR



$f = a + b$

OR		
a	b	*f*
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

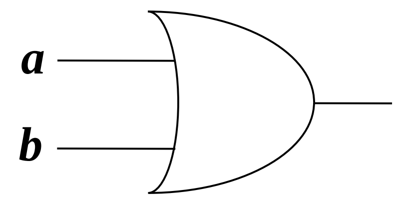
Puerta NAND



$f = \overline{a \cdot b}$

NAND		
a	b	*f*
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**

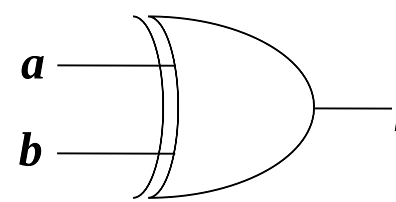
Puerta NOR



$f = \overline{a + b}$

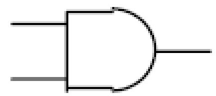
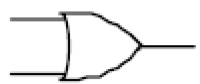
NOR		
a	b	*f*
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**

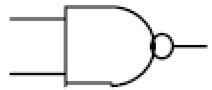
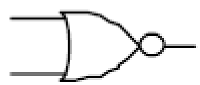
Puerta XOR

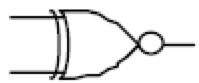
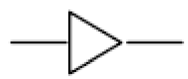


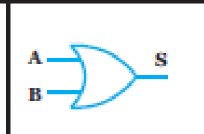
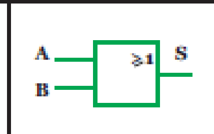
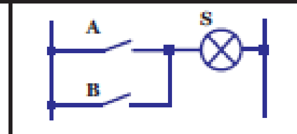
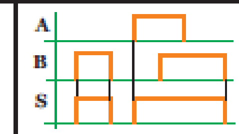
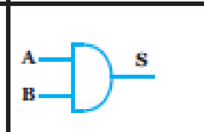
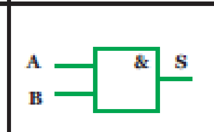
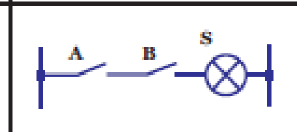
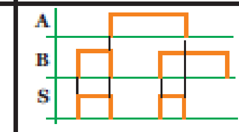
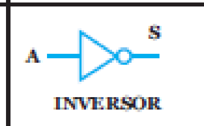
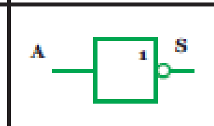
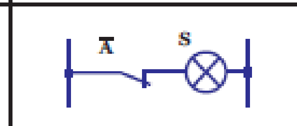
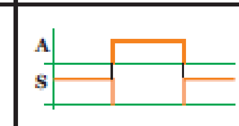
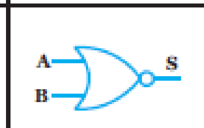
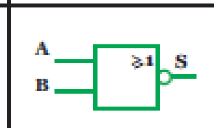
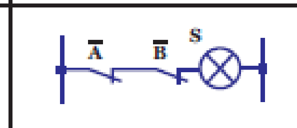
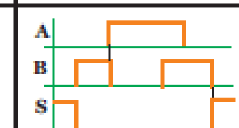
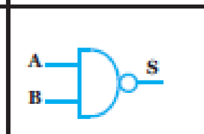
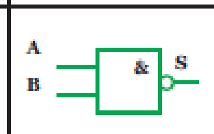
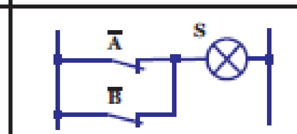
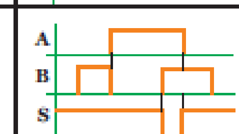
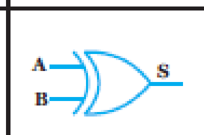
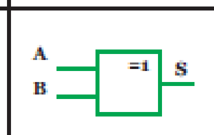
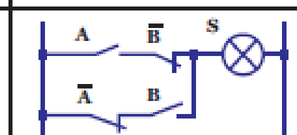
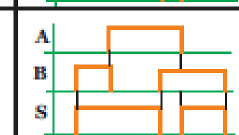
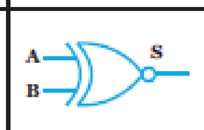
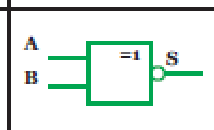
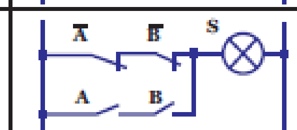
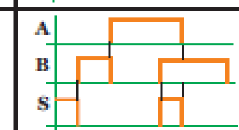
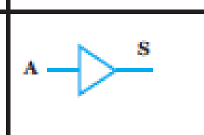
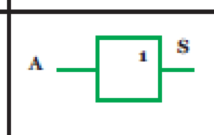
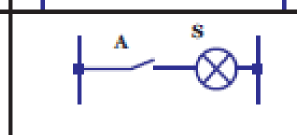
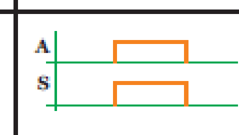
$f = \overline{a \oplus b}$

XOR		
a	b	*f*
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

<table>
<tr>
<td>AND (Y)</td>
<td>$F = a \bullet b$</td>
<td></td>
<td>
<table>
<tr><th>a</th><th>b</th><th>$F = a \bullet b$</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
<tr>
<td>OR (O)</td>
<td>$F = a + b$</td>
<td></td>
<td>
<table>
<tr><th>a</th><th>b</th><th>$F = a + b$</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
<tr>
<td>INVER</td>
<td>$F = \overline{a}$</td>
<td></td>
<td>
<table>
<tr><th>a</th><th>$F = \overline{a}$</th></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td>NAND</td>
<td>$F = \overline{a \bullet b}$</td>
<td></td>
<td></td>
<td>
<table>
<tr><td>a</td><td>b</td><td>$F = \overline{a \bullet b}$</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td>
</tr>
<tr>
<td>NOR</td>
<td>$F = \overline{a + b}$</td>
<td></td>
<td></td>
<td>
<table>
<tr><td>a</td><td>b</td><td>$F = \overline{a + b}$</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td>
</tr>
<tr>
<td>O exclusive</td>
<td>$F = a \oplus b$</td>
<td></td>
<td></td>
<td>
<table>
<tr><td>a</td><td>b</td><td>$F = a \oplus b$</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td>NOR exclusive</td>
<td>$F = \overline{a \oplus b}$</td>
<td></td>
<td></td>
<td>
<table>
<tr><th>a</th><th>b</th><th>$F = \overline{a \oplus b}$</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
<tr>
<td>Seguidor Buffer</td>
<td>$F = a$</td>
<td></td>
<td></td>
<td>
<table>
<tr><th>a</th><th>$F = \overline{a}$</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
</table>

Función	Ecuación lógica	Símbolos			Tabla de verdad	Cronograma
		Norma MIL	Norma IEC	Circuito físico		
OR	$S = A + B$				A B | S 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1	
AND	$S = A \cdot B$				A B | S 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1	
NOT	$S = \overline{A}$				A | S 0 | 1 1 | 0	
NOR	$S = \overline{A + B}$ $S = \overline{A} \cdot \overline{B}$				A B | S 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0	
NAND	$S = \overline{A \cdot B}$ $S = \overline{A} + \overline{B}$				A B | S 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0	
XOR	$S = A \oplus B =$ $= A \cdot \overline{B} + \overline{A} \cdot B$				A B | S 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	
XNOR	$S = \overline{A \oplus B} =$ $= \overline{A} \cdot \overline{B} + A \cdot B$				A B | S 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1	
EXCITADOR	$S = A$				A | S 0 | 0 1 | 1	

1.4 Funciones Lógicas: formas Canónicas

F= f(a,b,c,..) $F(a,b,c) = abc + \overline{b}(c + \overline{d})$

Un conjunto de *variables lógicas* conectadas por cualquier combinación de los *operadores básicos* (suma, producto y complementación) constituye una función lógica. Más formalmente, una función lógica, f, de n variables, $(x_1, x_2, \ldots x_n)$, sobre un álgebra de Boole $B = \{0,1\}$, es cualquier aplicación del producto cartesiano de B por sí mismo n veces, B^n, en B.

$$B^n = \overbrace{B \times B \times \cdots \times B}^{n}$$

$$B^n \xrightarrow{f(x_1,\ldots,x_n)} B \qquad [1.11]$$

Los elementos de B^n son configuraciones lógicas de n variables. Para $n=2$, tenemos cuatro configuraciones ($x_1 = 0, x_2 = 0;$ $x_1 = 0, x_2 = 1;$ $x_1 = 1, x_2 = 0;$ $x_1 = 1, x_2 = 1$). En general, para n variables tendremos 2^n configuraciones mutuamente exclusivas. Como para cada una de estas configuraciones la función lógica puede asignarle un "1" ó un "0", tenemos 2^{2^n} funciones lógicas posibles.

Función lógica

- Variable lógica
  - Se define una variable como un símbolo, por ejemplo “a” que representa a cualquiera de los elementos de un conjunto B={0, 1} sobre el que se ha definido un álgebra de Boole.
- El **valor de una función**
  - Se determina sustituyendo las variables por sus valores en la expresión algebraica y aplicando las reglas definidas para las operaciones suma y producto.

Ejemplo: Sea la siguiente:

- f = f (c, b, a) = b a + c b a’ + b’ a + a

Si sustituimos cada variable por el valor que representa {0, 1} obtenemos:

- f = f (c, b, a) = (1 * 1) + (1 * 1 * 0) + (0 * 1) + (1) = 1
- La representación mediante tabla de verdad sería la siguiente:

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

Funciones lógicas equivalentes

Se dice que dos funciones lógicas son **equivalentes,** si ambas tienen la misma tabla de verdad y por lo tanto describen la misma función de conmutación.

Ejemplo:
Sea la anterior función: $f_1 = b\,a + c\,b\,a' + b'\,a + a$

Determinar si la siguiente función es equivalente a f_1:
$f_2 = c\,b + a$

Representamos ambas funciones mediante tablas de verdad y si obtenemos el mismo resultado, entonces son equivalentes:

c	b	a	f_1	f_2
0	0	0	**0**	**0**
0	0	1	**1**	**1**
0	1	0	**0**	**0**
0	1	1	**1**	**1**
1	0	0	**0**	**0**
1	0	1	**1**	**1**
1	1	0	**1**	**1**
1	1	1	**1**	**1**

Ambas funciones son **equivalentes**, ya que obtenemos la **misma** tabla de verdad.

Representación de las funciones lógicas en su forma canónica

Se define ***término canónico***, a la suma o producto en el que aparecen todas las variables de la función, bien sea en su forma directa o negada.

Ejemplo:

Sea la función: $f\,(a, b, c) = \underbrace{abc'} + \underbrace{ab'c'} + ac + b'c$

Terminos canónicos

Los términos en los que aparecen todas las variables definidas en la función son abc' y ab'c'.

Para transformar los otros 2 términos a términos canónicos, habrá que añadirles las variables que faltan de forma que no varíe el resultado.

Para ello añadimos la variable en su forma directa y negada de la siguiente forma:

ac = a**b**c + a**b'**c

b'c = **a**b'c + **a'**b'c

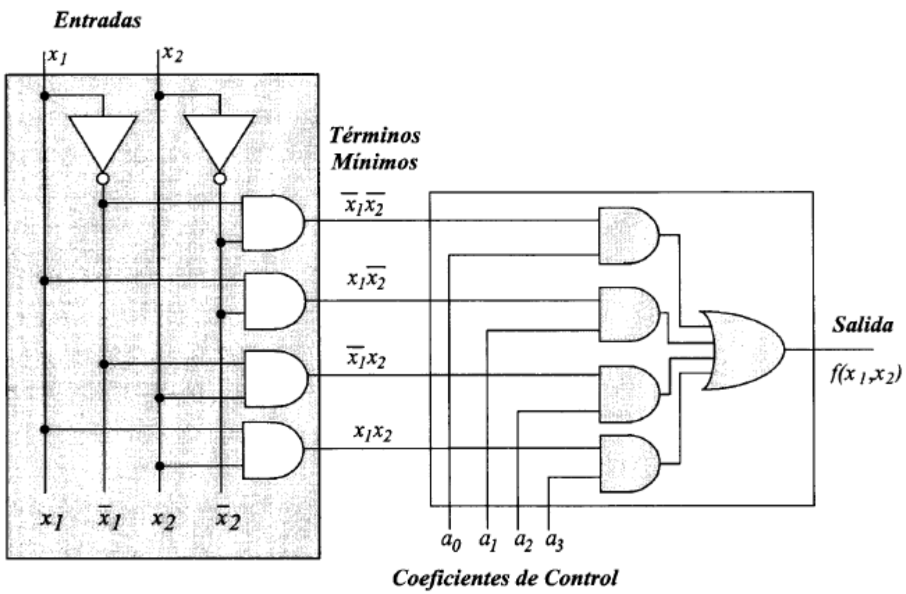
Una función formada <u>**únicamente**</u> por términos canónicos, diremos que se llama ***función canónica***.

Los términos canónicos reciben el nombre de producto canónico (**minitérminos**) o suma canónica (**maxitérminos**) cuando está formado únicamente por productos o por sumas respectivamente.

1.4.1 Forma Normal Disyuntiva

- Expresa una función lógica como suma de productos. Cada producto contiene a todas las variables, negadas o sin negar, sin repetirse ninguna

$$= A\overline{B}CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}D$$



x_1	x_2	m_i	a_i	f_0	f_1	f_2	f_3	....	f_9	....	f_{14}	f_{15}
0	0	m_0	a_0	0	0	0	0	....	1	....	1	1
0	1	m_1	a_1	0	0	0	0	....	0	....	1	1
1	0	m_2	a_2	0	0	1	1	....	0	....	1	1
1	1	m_3	a_3	0	1	0	1	....	1	....	0	1

Término minterm	a	b	c	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	1
7	1	1	1	1

Minterms: Se toman las salidas que son “1” y se expresa como suma de términos producto en los que las variables que son “1” se expresan como literales y las que son “0” como invertidas.

$$F(a,b,c) = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}c + ab\bar{c} + abc \quad \Rightarrow \quad F(a,b,c) = m_1 + m_2 + m_5 + m_6 + m_7 = \sum m(1,2,5,6,7)$$

1.4.2 Forma Normal Conjuntiva

- Expresa una función lógica como productos de sumas. Cada suma contiene a todas las variables, negadas o sin negar, sin repetirse ninguna

$$F(a,b,c) = (a+b+c)(a+\overline{b}+\overline{c})(\overline{a}+b+c)$$

FORMAS CANÓNICAS

x_1 x_2	*miniterms*	*coeficientes*	*Maxterms*	*coeficientes*
0 0	$\overline{x}_1 \cdot \overline{x}_2 = m_0$	a_0	$x_1 + x_2 = M_0 = \overline{m}_0$	A_0
0 1	$\overline{x}_1 \cdot x_2 = m_1$	a_1	$x_1 + \overline{x}_2 = M_1 = \overline{m}_1$	A_1
1 0	$x_1 \cdot \overline{x}_2 = m_2$	a_2	$\overline{x}_1 + x_2 = M_2 = \overline{m}_2$	A_2
1 1	$x_1 \cdot x_2 = m_3$	a_3	$\overline{x}_1 + \overline{x}_2 = M_3 = \overline{m}_3$	A_3
	$f_m = \sum a_i \cdot m_i$ $a_i = 1$ si el *miniterm* está presente $a_i = 0$ sino está presente		$F_M = \prod (A_i + M_i)$ $A_i = 0$ si el *Maxterm* está presente $A_i = 1$ sino está presente	

Representación de las funciones lógicas en su forma canónica

Ejemplo:
Obtener la función canónica a partir de la tabla de verdad siguiente:

c	b	a	*f*
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

Se puede resolver de 2 maneras, utilizando los minitérminos o los Maxitérminos ya que el enunciado no dice nada concreto.

Utilizando ***minitérnimos***:

Sumaremos aquellos en los que la función vale 1, por lo tanto será:

$$f = \sum m_i = m_1 + m_3 + m_4 + m_6 + m_7$$

$$f = 001 + 011 + 100 + 110 + 111$$

$$f = c'b'a + c'ba + cb'a' + cba' + cba$$

Utilizando ***maxitérnimos***:

Multiplicaremos aquellos en los que la función vale 0, por lo tanto será:

$$f = \prod M_i = M_0 * M_2 * M_5$$

a.1: Elegimos las filas en las que ***F = 1*** y ***sumamos los términos mínimos*** (productos de las variables) de las filas correspondientes.

A	***B***	***Minterms***	***Maxterms***	***F=Anticoincidencia***
0	0	$m0 = \overline{A}\,\overline{B}$	$M0 = A + B$	**0**
0	1	$m1 = \overline{A}\,B$	$M1 = A + \overline{B}$	**1**
1	0	$m2 = A\,\overline{B}$	$M2 = \overline{A} + B$	**1**
1	1	$m3 = A\,B$	$M3 = \overline{A} + \overline{B}$	**0**

Así, la expresión de F representada mediante minterms es:

$$F = A \oplus B = m1 + m2 = \overline{A}\,B + A\,\overline{B}$$

b) Representación en la **Forma Normal Conjuntiva** (ver pag. 36) en la que se representa la función mediante producto de sumas (producto de términos máximos). Veamos dos formas de hallar esta representación:

b.1: Elegimos las filas en las que ***F=0*** y multiplicamos los términos máximos (sumas de las variables) de las filas correspondientes.

A	***B***	***Minterms***	***Maxterms***	***F=Anticoincidencia***
0	0	$m0 = \overline{A}\,\overline{B}$	$M0 = A + B$	**0**
0	1	$m1 = \overline{A}\,B$	$M1 = A + \overline{B}$	**1**
1	0	$m2 = A\,\overline{B}$	$M2 = \overline{A} + B$	**1**
1	1	$m3 = A\,B$	$M3 = \overline{A} + \overline{B}$	**0**

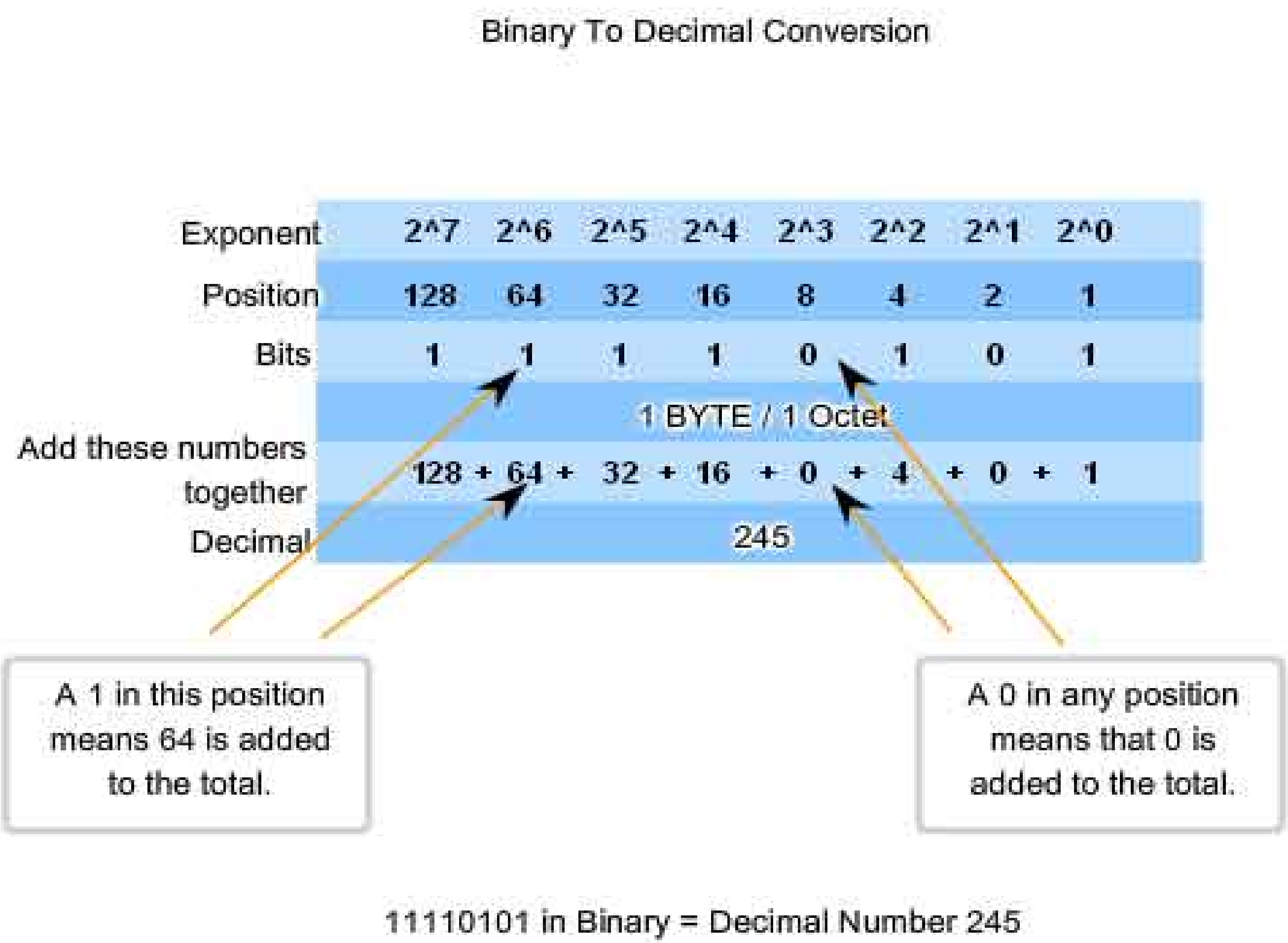
Así, $F = M0 \cdot M3 = (A+B)\left(\overline{A}+\overline{B}\right)$. Esta es la expresión de ***F*** representada mediante ***maxterms***.

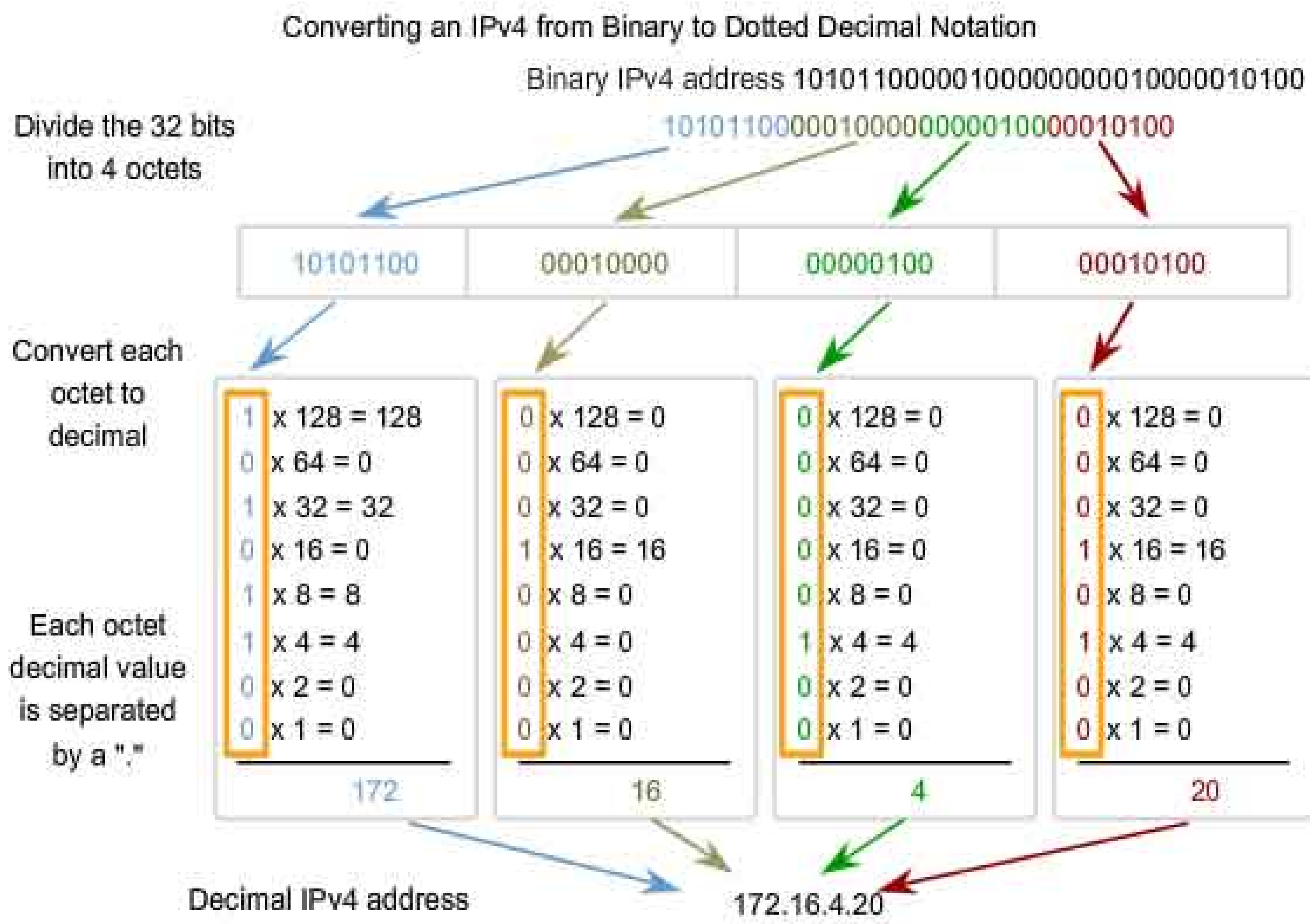
Veamos que esta función también coincide con la función calculada anteriormente. En efecto,

$$F = M0 \cdot M3 = (A+B)\left(\overline{A}+\overline{B}\right) = \overset{0}{A\overline{A}} + A\overline{B} + \overline{A}B + \overset{0}{B\overline{B}} = A\overline{B} + \overline{A}B = A \oplus B$$

Operando

Decimal	*ABCD*	*minterm*	*Maxterm*
0	0000	$\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}$	$A+B+C+D$
1	0001	$\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot D$	$A+B+C+\overline{D}$
2	0010	$\overline{A}\cdot\overline{B}\cdot C\cdot\overline{D}$	$A+B+\overline{C}+D$
3	0011	$\overline{A}\cdot\overline{B}\cdot C\cdot D$	$A+B+\overline{C}+\overline{D}$
4	0100	$\overline{A}\cdot B\cdot\overline{C}\cdot\overline{D}$	$A+\overline{B}+C+D$
5	0101	$\overline{A}\cdot B\cdot\overline{C}\cdot D$	$A+\overline{B}+C+\overline{D}$
6	0110	$\overline{A}\cdot B\cdot C\cdot\overline{D}$	$A+\overline{B}+\overline{C}+D$
7	0111	$\overline{A}\cdot B\cdot C\cdot D$	$A+\overline{B}+\overline{C}+\overline{D}$
8	1000	$A\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}$	$\overline{A}+B+C+D$
9	1001	$A\cdot\overline{B}\cdot\overline{C}\cdot D$	$\overline{A}+B+C+\overline{D}$
10	1010	$A\cdot\overline{B}\cdot C\cdot\overline{D}$	$\overline{A}+B+\overline{C}+D$
11	1011	$A\cdot\overline{B}\cdot C\cdot D$	$\overline{A}+B+\overline{C}+\overline{D}$
12	1100	$A\cdot B\cdot\overline{C}\cdot\overline{D}$	$\overline{A}+\overline{B}+C+D$
13	1101	$A\cdot B\cdot\overline{C}\cdot D$	$\overline{A}+\overline{B}+C+\overline{D}$
14	1110	$A\cdot B\cdot C\cdot\overline{D}$	$\overline{A}+\overline{B}+\overline{C}+D$
15	1111	$A\cdot B\cdot C\cdot D$	$\overline{A}+\overline{B}+\overline{C}+\overline{D}$


Binary To Decimal Conversion
Exponent
2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
Position
128 64 32 16 8 4 2 1
Bits
1 1 1 1 0 1 0 1
1 BYTE / 1 Octet
Add these numbers together
128 + 64 + 32 + 16 + 0 + 4 + 0 + 1
Decimal
245
A 1 in this position means 64 is added to the total.
A 0 in any position means that 0 is added to the total.
11110101 in Binary = Decimal Number 245


Converting an IPv4 from Binary to Dotted Decimal Notation
Binary IPv4 address 10101100000100000000010000010100
Divide the 32 bits into 4 octets
10101100000100000000010000010100
10101100
00010000
00000100
00010100
Convert each octet to decimal
1 x 128 = 128
0 x 64 = 0
1 x 32 = 32
0 x 16 = 0
1 x 8 = 8
1 x 4 = 4
0 x 2 = 0
0 x 1 = 0
172
0 x 128 = 0
0 x 64 = 0
0 x 32 = 0
1 x 16 = 16
0 x 8 = 0
0 x 4 = 0
0 x 2 = 0
0 x 1 = 0
16
0 x 128 = 0
0 x 64 = 0
0 x 32 = 0
0 x 16 = 0
0 x 8 = 0
1 x 4 = 4
0 x 2 = 0
0 x 1 = 0
4
0 x 128 = 0
0 x 64 = 0
0 x 32 = 0
1 x 16 = 16
0 x 8 = 0
1 x 4 = 4
0 x 2 = 0
0 x 1 = 0
20
Each octet decimal value is separated by a "."
Decimal IPv4 address
172.16.4.20

x_1	x_2	X-OR	NAND	NOR
0	0	*0*	*1*	*1*
0	1	*1*	*1*	*0*
1	0	*1*	*1*	*0*
1	1	*0*	*0*	*0*
Representación por		$\Downarrow$	$\Downarrow$	$\Downarrow$
minterms		$\Sigma m(1,2)$	$\Sigma m(0,1,2)$	$\Sigma m(0)$
Maxterms		$\Pi M(0,3)$	$\Pi M(3)$	$\Pi M(1,2,3)$

Figura 1.20. Representación por minterms y por maxterms de las funciones X-OR, NAND y NOR.

Ejercicios **(I)** *Dadas las siguientes funciones de tres variables en representación por minterms, obtener su representación por maxterms*

a) $f_m = \sum m(0,1,2,6,7)$

b) $f_m = \sum m(2,3,4,5,6)$

Solución

a) La expresión en minterms de *f*, (f_m), nos da los "1" de *f*. Busquemos entonces sus ceros que estarán en las líneas complementarias de la tabla de verdad, es decir en (3,4,5). Por consiguiente, podremos escribir directamente que la expresión maxterms de *f* es:

$$f_M = M_3 \, M_4 \, M_5 \qquad [1.32]$$

b) $f_m = \sum m(2,3,4,5,6)$

$\overline{f_m} = \sum m(0,1,7)$

$$f_M = \overline{\overline{f_m}} = \overline{\Sigma m(0,1,7)} = \overline{m_0} \; \overline{m_1} \; \overline{m_7} = M_0 \, M_1 \, M_7 \qquad [1.33]$$

Conversión de expresiones a forma canónicas

Se llama expresión normalizada a la que está formada únicamente por términos canónicos, y todos ellos están representados como suma de productos o producto de sumas.

$$\left.\begin{array}{l} f\ (c, b, a) = c\,b + c'\,b\,a' \\ f\ (c, b, a) = (c + b)(c' + b + a') \end{array}\right\} \text{Normalizadas}$$

$$\left.\begin{array}{l} f\ (c, b, a) = c\,(b + b'\,a) + b \end{array}\right\} \text{No normalizada}$$

Las expresiones no normalizadas se pueden normalizar operando con ellas.

Ejemplo:

La siguiente función no normalizada $f = c\,(b + b'\,a) + b$, se puede normalizar de la siguiente forma:

Hay que expresarla en suma de productos o producto de sumas.

En nuestro caso, si desarrollamos los paréntesis, obtenemos la suma de productos.

$$f = c\ (b + b'\ a) + b = c\ b + c\ b'\ a + b$$

Para convertir funciones no normalizadas a canónicas

En el caso de ***suma de productos***, se multiplican los términos a los que le falta una variable, por dicha variable y su negada.

En el caso de ***producto de sumas***, a cada factor que le falta una variable, se le suma dicha variable y su negada.

Conversión de expresiones a forma canónicas

Ejemplo:

La función anterior, es normalizada, pero para transformarla a su expresión canónica realizaremos los siguientes pasos:

$$f = \underbrace{c\,b}_{(a + a')} + c\,b'\,a + \underbrace{b}_{(ca + ca' + c'a + c'a')}$$

Al primer término le falta la variable "a", por lo que multiplicamos el término por dicha variable y su negada.

$$f = c\,b\,(a + a') + c\,b'\,a + b \quad = c\,b\,a + c\,b\,a' + c\,b'\,a + b$$

El último término le faltan las variables "a" y "c", por lo que multiplicamos por cada una de ellas y su negada.

$$f = c\,b\,a + c\,b\,a' + c\,b'\,a + b\,(c'a' + c'a + ca' + ca)$$

$$f = c\,b\,a + c\,b\,a' + c\,b'\,a + c'\,b\,a' + c'\,b\,a + c\,b\,a' + c\,b\,a$$

De esta forma tenemos finalmente la transformación de una expresión normalizada a su expresión canónica.

$$\boldsymbol{f = c\,b\,a + c\,b\,a' + c\,b'\,a + c'\,b\,a' + c'\,b\,a + c\,b\,a' + c\,b\,a}$$

1.- Dada la función de 2 variables, $f = \overline{x_0}\,\overline{x_1} + x_0 x_1$, expresada en su forma normal disyuntiva (suma de minterms) ¿cuál es la representación de esta misma función en su forma normal conjuntiva (producto de maxterms)?.

a) $f = \left(x_0 + x_1\right)\cdot\left(\overline{x_0} + \overline{x_1}\right)$

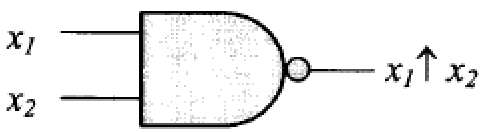
b) $\overline{f} = \left(x_0 + \overline{x_1}\right)\cdot\left(\overline{x_0} + x_1\right)$

c) $f = \left(x_0 + \overline{x_1}\right)\cdot\left(\overline{x_0} + x_1\right)$

d) $f = \left(x_0 + x_1\right)\cdot\left(x_0 + \overline{x_1}\right)$

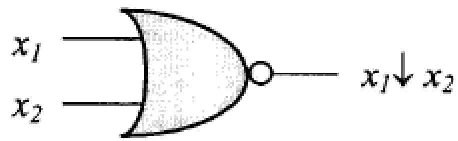
1.5 Otras Representaciones Completas

NAND ($\uparrow$)	NOR ($\downarrow$)



x_1	x_2	$\uparrow$
0	0	*1*
0	1	*1*
1	0	*1*
1	1	*0*

$$x_1 \uparrow x_2 = \bar{x}_1 \bar{x}_2 + \bar{x}_1 x_2 + x_1 \bar{x}_2 =$$
$$= \overline{x_1 x_2} = \bar{x}_1 + \bar{x}_2$$



x_1	x_2	$\downarrow$
0	0	*1*
0	1	*0*
1	0	*0*
1	1	*0*

$$x_1 \downarrow x_2 = \bar{x}_1 \bar{x}_2 = \overline{x_1 + x_2}$$

Figura 1.21. Definición de los operadores NAND y NOR.

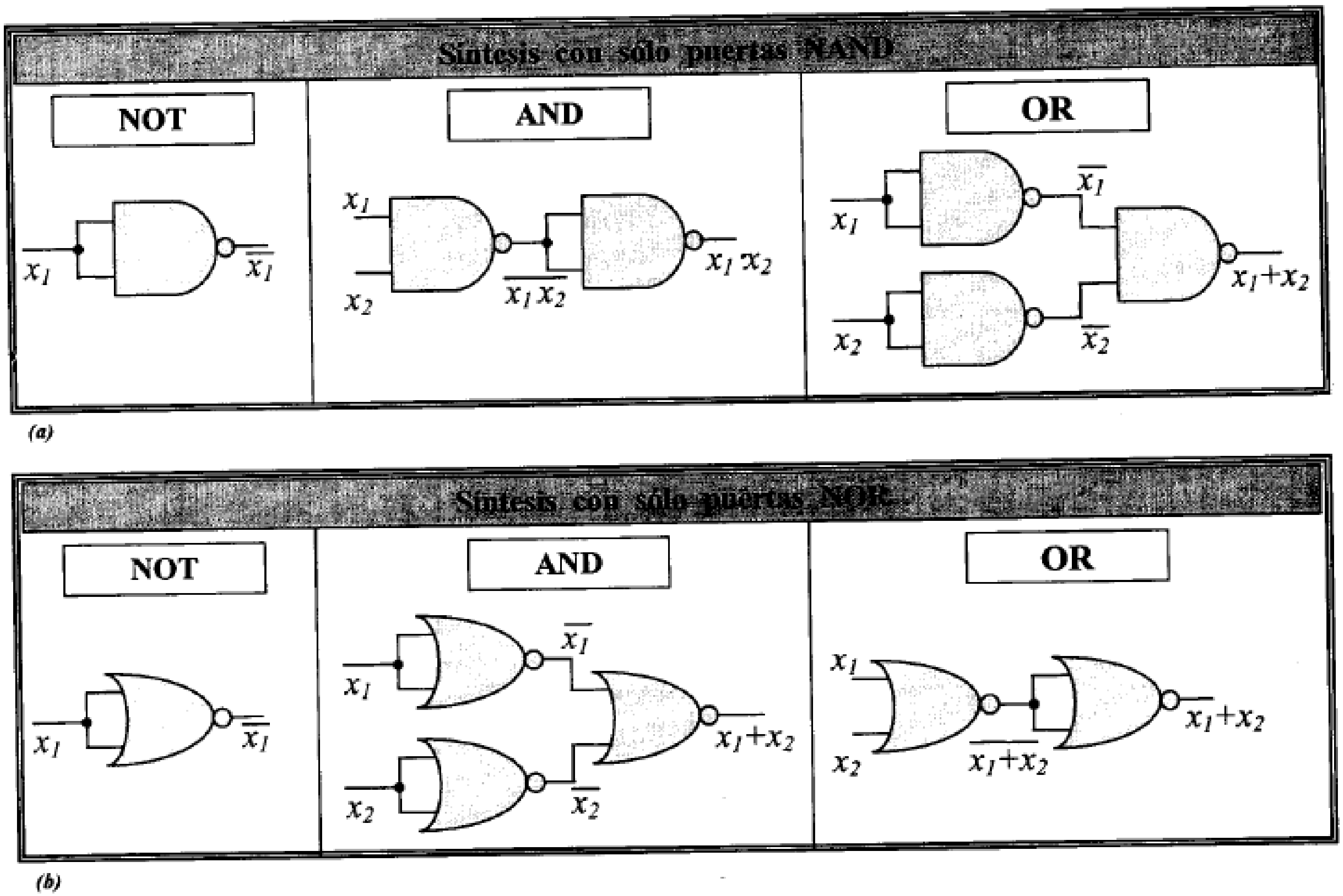


Figura 1.22. Síntesis del conjunto completo (AND, OR, NOT). **(*a*)** Usando sólo operadores NAND. **(*b*)** Usando sólo operadores NOR.

Ejercicio: *Representar con sólo puertas NAND la siguiente función:*

$$A = XY\overline{Z} + XYZ + X(Y + \overline{Z})$$

Para el paso a NAND podemos usar el siguiente procedimiento general:

1. ***Obtener una expresión mínima*** en forma de suma de productos (este apartado lo veremos al comentar el **objetivo 6**).
2. ***Complementar dos veces.*** Al complementar dos veces hemos dejado la función como estaba, por lo que siempre podremos hacerlo.
3. ***Aplicar los teoremas de De Morgan***. La aplicación repetida de los Teoremas de De Morgan debe pararse cuando:
   a) ***Sólo encontramos variables negadas*** ($\overline{X}, \overline{Y}$) que se sintetizan con un inversor, el cual es un caso particular de una puerta NAND en la que se unen las dos entradas.
   b) ***Sólo encontramos negaciones de productos*** ($\overline{X\,\overline{Y}}, \overline{\overline{X}\,Y}, \ldots$).

Solución:

De acuerdo con el procedimiento descrito, primero complementamos dos veces. Así,

$$A = \overline{\overline{A}} = XY\overline{Z} + XYZ + X\left(Y + \overline{Z}\right) = \overline{\overline{XY\overline{Z} + XYZ + XY + X\overline{Z}}} \qquad [1.43]$$

A continuación aplicaremos los teoremas de DeMorgan, resultando:

$$A = \overline{\overline{XY\overline{Z} + XYZ + XY + X\overline{Z}}} = \overline{\overline{XY\overline{Z}}\;\overline{XYZ}\;\overline{XY}\;\overline{X\overline{Z}}} \qquad [1.44]$$

Por último, comprobaremos con la tabla de verdad que ambas representaciones *[1.43]* y *[1.44]* coinciden.

X	Y	Z	$XY\overline{Z}$	XYZ	$X\left(Y+\overline{Z}\right)$	***A [1.43]***	$\overline{XY\overline{Z}}$	$\overline{XYZ}$	$\overline{XY}$	$\overline{X\overline{Z}}$	***A [1.44]***
0	0	0	0	0	0	**0**	1	1	1	1	**0**
0	0	1	0	0	0	**0**	1	1	1	1	**0**
0	1	0	0	0	0	**0**	1	1	1	1	**0**
0	1	1	0	0	0	**0**	1	1	1	1	**0**
1	0	0	0	0	1	**1**	1	1	1	0	**1**
1	0	1	0	0	0	**0**	1	1	1	1	**0**
1	1	0	1	0	1	**1**	0	1	0	0	**1**
1	1	1	0	1	1	**1**	1	0	0	1	**1**

Figura 1.23. Tabla de verdad para demostrar que las expresiones *[1.43]* y *[1.44]* coinciden.

R.1.11: Los pasos a seguir para expresar una función con sólo operadores NAND son:

1º. Obtener la expresión mínima como suma de productos.

2º. Complementar la expresión resultante dos veces, lo que es equivalente a dejarla tal cual.

3º. Aplicar reiterativamente los teoremas de De Morgan hasta obtener la función expresada sólo con variables negadas y con productos negados.

Veámoslo a través de un ejemplo:

Sea la función lógica: $f = XY\overline{Z} + XY + \overline{X}Z$

1º. Minimizamos: $f = XY\overline{Z} + XY + \overline{X}Z = XY(\overline{Z} + 1) + \overline{X}Z = XY + \overline{X}Z$

puesto que $(\overline{Z} + 1) = 1$

2º. Negamos dos veces: $f = XY + \overline{X}Z = \overline{\overline{f}} = \overline{\overline{XY + \overline{X}Z}}$

3º. Aplicamos De Morgan reiteradamente: $f = \overline{\overline{XY + \overline{X}Z}} = \overline{\overline{XY}\;\overline{\overline{X}Z}}$ (aunque en este caso sólo ha hecho falta aplicar De Morgan una vez para obtener la expresión con sólo operadores NAND).

Para expresar una función con ***sólo operadores NOR*** el procedimiento es análogo, sólo que detenemos la aplicación reiterada de los teoremas de De Morgan cuando tengamos la expresión expresada sólo con variables negadas y con sumas negadas.

Veamos ahora la forma de representar la misma función con sólo puertas NOR.

Los 3 pasos anteriores son comunes. Por tanto, partimos de la expresión:

$$f = \overline{\overline{XY}\;\;\overline{\overline{X}Z}}$$

Si seguimos aplicando De Morgan reiteradamente obtenemos:

$$f = \overline{\overline{XY}\;\;\overline{\overline{X}Z}} = \overline{\left(\overline{X} + \overline{Y}\right)\left(X + \overline{Z}\right)} = \overline{\left(\overline{X} + \overline{Y}\right)} + \overline{\left(X + \overline{Z}\right)}$$

Como podemos observar, la función ***f*** está expresada como la función OR (no NOR) de dos funciones NOR. Por lo tanto, como tenemos que expresarla en función sólo de NOR, tendremos que ver la forma de pasar de la OR a la NOR. Esto se consigue directamente negando la expresión dos veces. Así, en este caso conseguimos la función NOR de la NOR de dos funciones NOR. Es decir:

$$f = \overline{\left(\overline{X} + \overline{Y}\right)} + \overline{\left(X + \overline{Z}\right)} = \overline{\overline{\overline{\left(\overline{X} + \overline{Y}\right)} + \overline{\left(X + \overline{Z}\right)}}}$$

Método gráfico

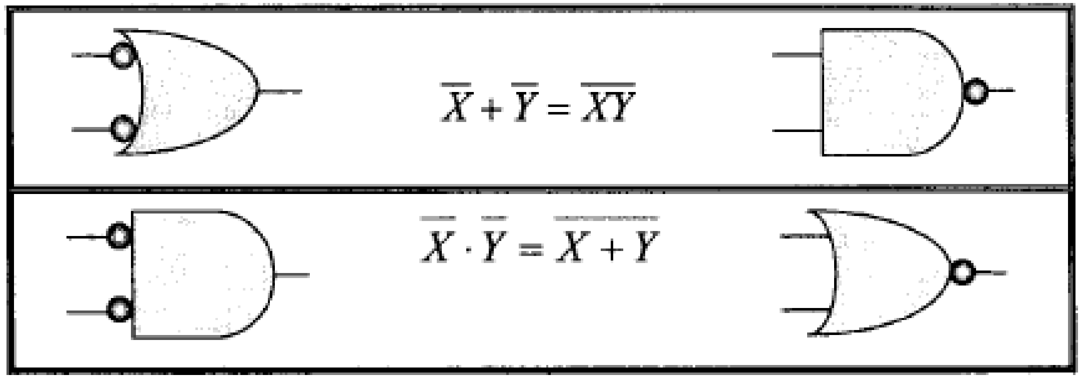


Figura 1.28. Implementación de los teoremas de DeMorgan

Negar las salidas de las puertas de primer nivel y las entradas de las puertas de segundo nivel
Las dos negaciones hace que el resultado no cambie

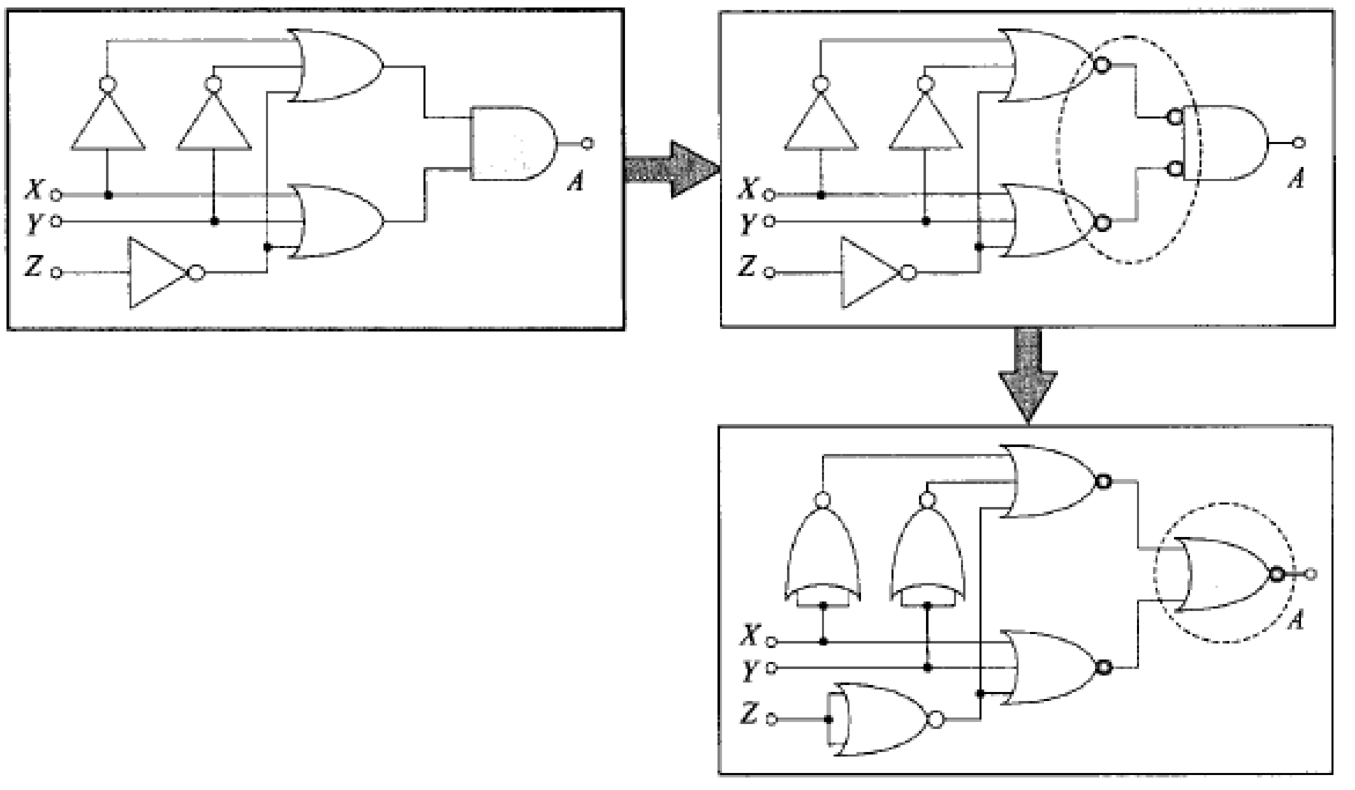


Figura 1.29. Ilustración del método gráfico.

1.6 Análisis y Síntesis

ANÁLISIS

Analizar un circuito lógico es encontrar la función lógica que calcula, a partir del esquema de conexión de las variables de entrada con los distintos operadores hasta llegar a la variable de salida.

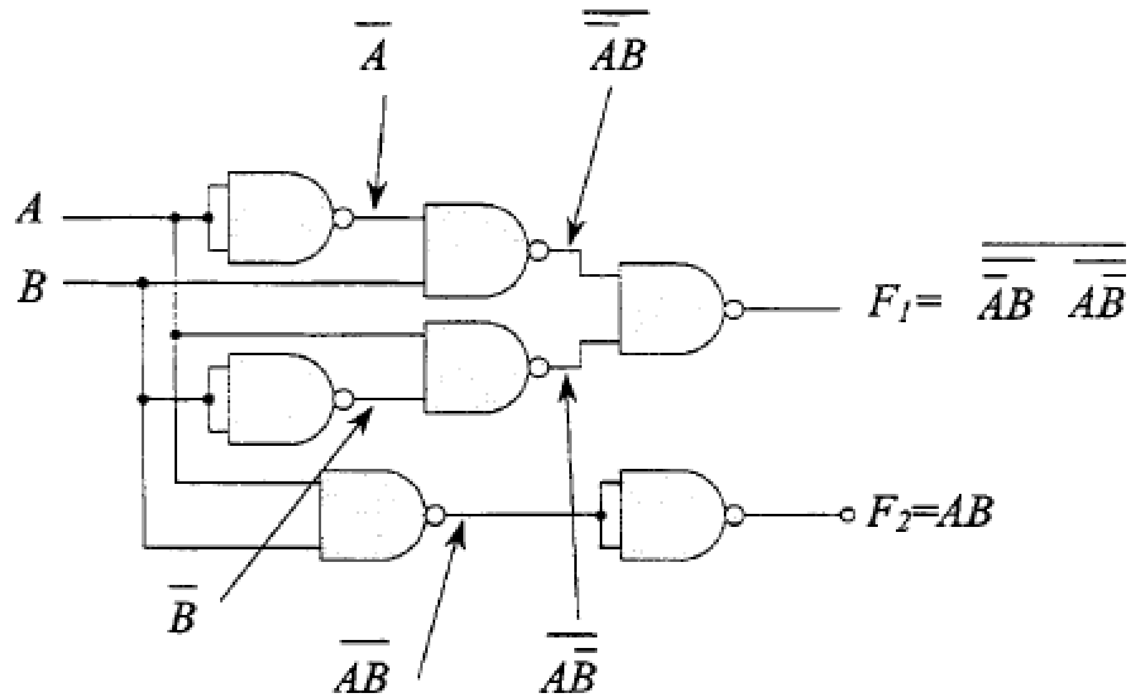


Figura 1.30.

La forma de analizar un circuito es seguir el camino de la señal desde la entrada hasta la salida, anotando las transformaciones que introducen los operadores que se encuentran en el camino

Síntesis

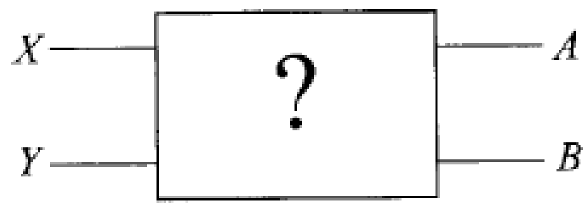
Las dos fases esenciales en todo proceso de *síntesis* son:

Fase a:	*Paso de una descripción en lenguaje natural a una función lógica.*
Fase b:	*Paso de una función lógica a su circuito correspondiente.*

Ejercicio: *Obtener una descripción lógica, incluyendo la selección de variables de entrada y salida, de un circuito con dos entradas y dos salidas tal que la primera salida esté en alta cuando el valor en las dos entradas es el mismo y la segunda salida está en alta cuando no coinciden las entradas. Obtener también el circuito que implementa dichas funciones*

Fase a: *Paso de una descripción en lenguaje natural a una función lógica.*

1. Empezamos asignando dos variables (X,Y) a las entradas y otras dos (A,B) a las salidas.



2. Construimos una tabla de verdad en la que describimos las cuatro configuraciones lógicas posibles en la entrada y ponemos un uno en los términos mínimos que hacen pasar A a alta y en otra columna, los que hacen pasar B a alta.

X	*Y*	*A*	*B*
0	0	*1*	*0*
0	1	*0*	*1*
1	0	*0*	*1*
1	1	*1*	*0*

Si hacemos la síntesis usando los términos mínimos tomamos las configuraciones de entrada que hacen que la salida sea 1. Así, para la función A, tenemos:

$$A = \overline{X}\ \overline{Y} + X\ Y \qquad [1.48]$$

Análogamente para la salida obtenemos: $B = \overline{X}\ Y + X\ \overline{Y}$ [1.49]

Es decir, la función A es la *coincidencia* y la B la *anticoincidencia* ó OR exclusivo.

Así, $A = \overline{X \oplus Y}$ y $B = X \oplus Y$ [1.50]

<u>Fase b</u>: *Paso de una función lógica a su circuito correspondiente.*

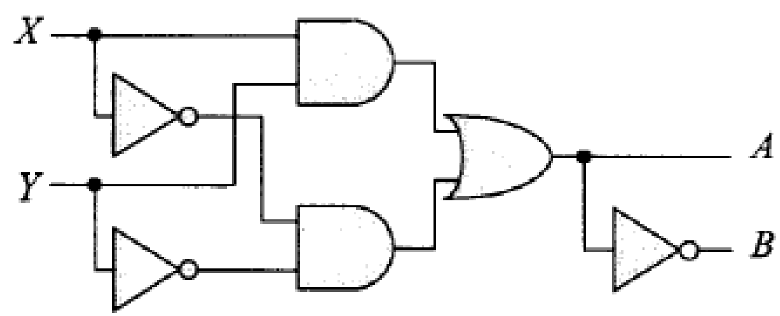


Figura 1.33.

1.7 Introducción a la minimización

Minimizar una función lógica es obtener la expresión más simplificada posible para la misma de forma que el número de operadores necesarios para su síntesis es también mínimo.

a) $$f = AB + C\left(AB + \overline{C}\right) = AB + CAB + \underbrace{C\overline{C}}_{0} = AB\underbrace{\left(C + 1\right)}_{1} = AB \qquad [1.60]$$

b) $$f = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + A\,B\,\overline{C} + A\,\overline{B}\,\overline{C} \qquad [1.61]$$

$$\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + A\,B\,\overline{C} + A\,\overline{B}\,\overline{C} = \overline{A}\,\overline{C}\left(B + \overline{B}\right) + A\,\overline{C}\left(B + \overline{B}\right) = \overline{A}\,\overline{C} + A\,\overline{C} = \overline{C}\left(A + \overline{A}\right) = \overline{C} \qquad [1.62]$$

Mediante las técnicas del álgebra de Boole, simplificar la siguiente expresión:

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

Solución

El método que se aplica no es necesariamente el único posible.

Paso 1. Aplicar la ley distributiva al segundo y tercer término de la expresión del siguiente modo:

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

Paso 2. Aplicar la regla 7 ($BB = B$) al cuarto término:

$$AB + AB + AC + B + BC$$

Paso 3. Aplicar la regla 5 ($AB + AB = AB$) a los dos primeros términos:

$$AB + AC + B + BC$$

Paso 4. Aplicar la regla 10 ($B + BC = B$) a los dos últimos términos:

$$AB + AC + B$$

Paso 5. Aplicar la regla 10 ($AB + B = B$) a los términos primero y tercero:

$$B + AC$$

En este instante, la expresión ya no puede seguir simplificándose. Según se vaya adquiriendo experiencia en la aplicación del álgebra de Boole, se podrán combinar muchos pasos individuales.

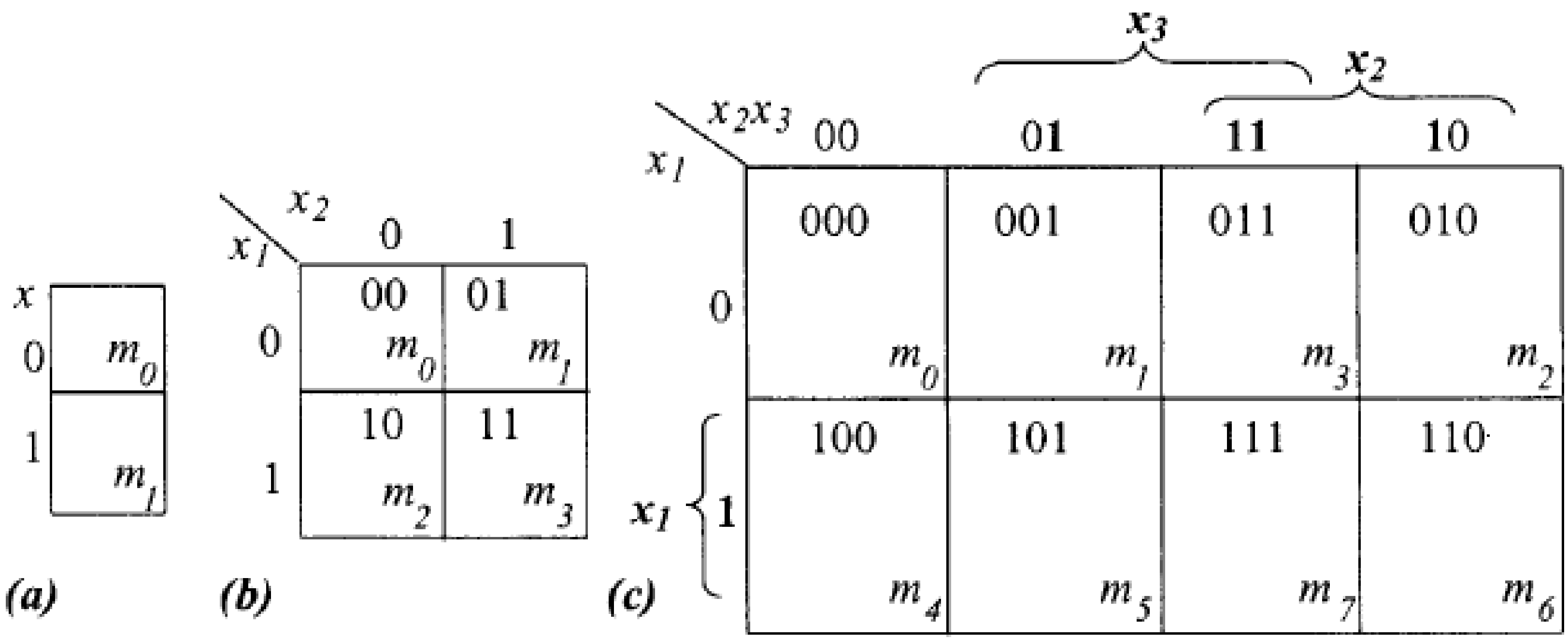


Figura 1.37. Simplificación de funciones usando diagramas de Karnaugh. Ver descripción en el texto.

A \ B	0	1
0	m_0: $\overline{A}\,\overline{B}$	m_1: $\overline{A}B$
1	m_2: $A\overline{B}$	m_3: $\overline{A}\,\overline{B}$

A \ BC	00	01	11	10
0	m_0: $\overline{A}\,\overline{B}\,\overline{C}$	m_1: $\overline{A}\,\overline{B}C$	m_3: $\overline{A}BC$	m_2: $\overline{A}B\overline{C}$
1	m_4: $A\overline{B}\,\overline{C}$	m_5: $A\overline{B}C$	m_6: ABC	m_7: $AB\overline{C}$

AB \ CD	00	01	11	10
00	m_0: $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$	m_1: $\overline{A}\,\overline{B}\,\overline{C}D$	m_3: $\overline{A}\,\overline{B}CD$	m_2: $\overline{A}\,\overline{B}C\overline{D}$
01	m_4: $\overline{A}B\overline{C}\,\overline{D}$	m_5: $\overline{A}B\overline{C}D$	m_7: $\overline{A}BCD$	m_6: $\overline{A}BC\overline{D}$
11	m_{12}: $AB\overline{C}\,\overline{D}$	m_{13}: $AB\overline{C}D$	m_{15}: $ABCD$	m_{14}: $ABC\overline{D}$
10	m_8: $A\overline{B}\,\overline{C}\,\overline{D}$	m_9: $A\overline{B}\,\overline{C}D$	m_{11}: $A\overline{B}CD$	m_{10}: $A\overline{B}C\overline{D}$

a) $f = \overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}\,\overline{Y}\,Z + \overline{X}\,Y\,\overline{Z} + \overline{X}\,Y\,Z + X\,Y\,Z = \sum m(0,1,2,3,7)$

YZ \ **X**	$\overline{Y}\,\overline{Z}$ ⇓ 00	$\overline{Y}\,Z$ ⇓ 01	$Y\,Z$ ⇓ 11	$Y\,\overline{Z}$ ⇓ 10
$\overline{X} \Rightarrow 0$	*1* 0	*1* 1	*1* 3	*1* 2
$X \Rightarrow 1$	*0* 4	*0* 5	*1* 7	*0* 6

$$\boxed{f = \overline{X} + Y\,Z}$$

Figura 1.40. Minimización por diagramas de V-K de la función $f=\Sigma m(0,1,2,3,7)$.

Simplificación con los ceros; el resultado hay que complementarlo

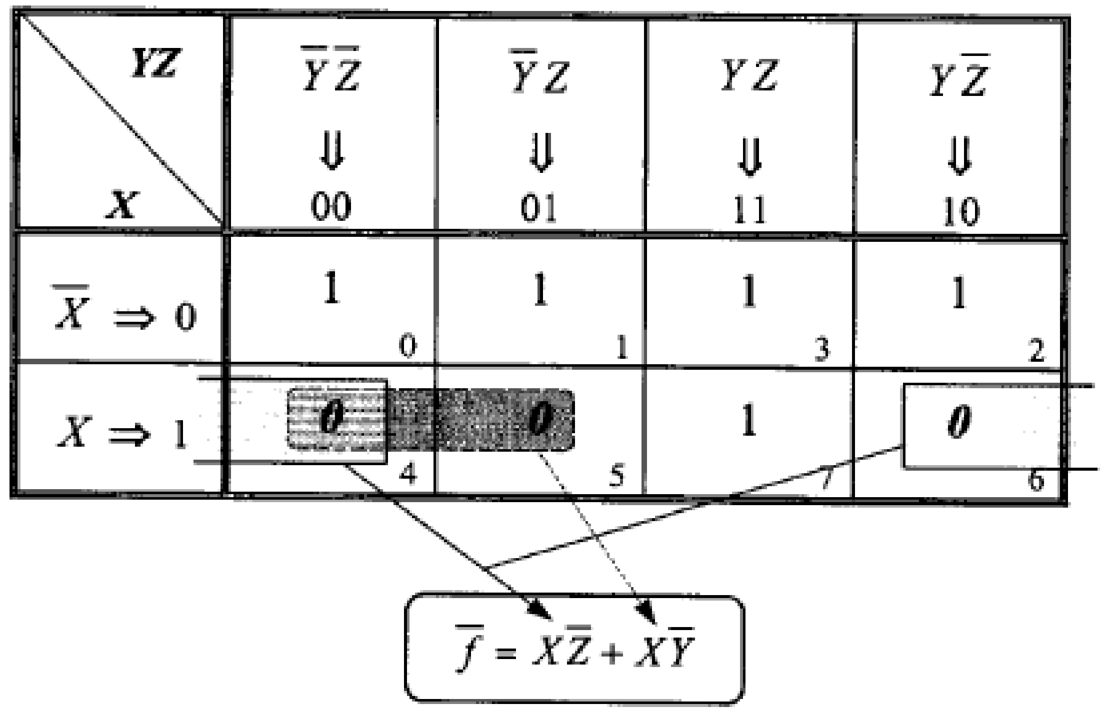


Figura 1.41. Minimización por diagramas de V-K de la función $f=\Sigma m(4,5,6)$.

b) $f = \overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}\,Y\,\overline{Z} + \overline{X}\,Y\,Z + X\,\overline{Y}\,\overline{Z} + X\,Y\,Z = \sum m(0,2,3,4,7)$

YZ \ X	$\overline{Y}\,\overline{Z}$ ⇓ 00	$\overline{Y}Z$ ⇓ 01	YZ ⇓ 11	$Y\overline{Z}$ ⇓ 10
$\overline{X} \Rightarrow 0$	1 [0]	0 [1]	1 [3]	1 [2]
$X \Rightarrow 1$	1 [4]	0 [5]	1 [7]	0 [6]

$$\boxed{f = \overline{Y}\,\overline{Z} + YZ + \overline{X}Y}$$

Figura 1.42. Minimización por diagramas de V-K de la función $f = \Sigma m(0,2,3,4,7)$.

c) $$f = \overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}\,\overline{Y}\,Z + \overline{X}\,Y\,\overline{Z} + \overline{X}\,Y\,Z + X\,\overline{Y}\,\overline{Z} + X\,Y\,\overline{Z} = \sum m(0,1,2,3,4,6) \qquad [1.75]$$

En esta función, dado que el número de "0" es menor que el de "1" la minimizamos tomando los ceros. Así, obtenemos:

X \ YZ	$\overline{Y}\,\overline{Z}$ ⇓ 00	$\overline{Y}\,Z$ ⇓ 01	$Y\,Z$ ⇓ 11	$Y\,\overline{Z}$ ⇓ 10
$\overline{X} \Rightarrow 0$	1 (0)	1 (1)	1 (3)	1 (2)
$X \Rightarrow 1$	1 (4)	***0*** (5)	***0*** (7)	1 (6)

$\overline{f} = X\,Z$

Figura 1.43. Minimización por diagramas de V-K de la función $f = \Sigma m(0,1,2,3,4,6)$

Ahora para obtener la expresión de la función f minimizada deberemos complementar el resultado. Es decir,

$$f = \overline{X\,Z} = \overline{X} + \overline{Z} \qquad [1.76]$$

a) $f = \sum m(0,2,3,4,5,6,7,8,12)$ [1.78]

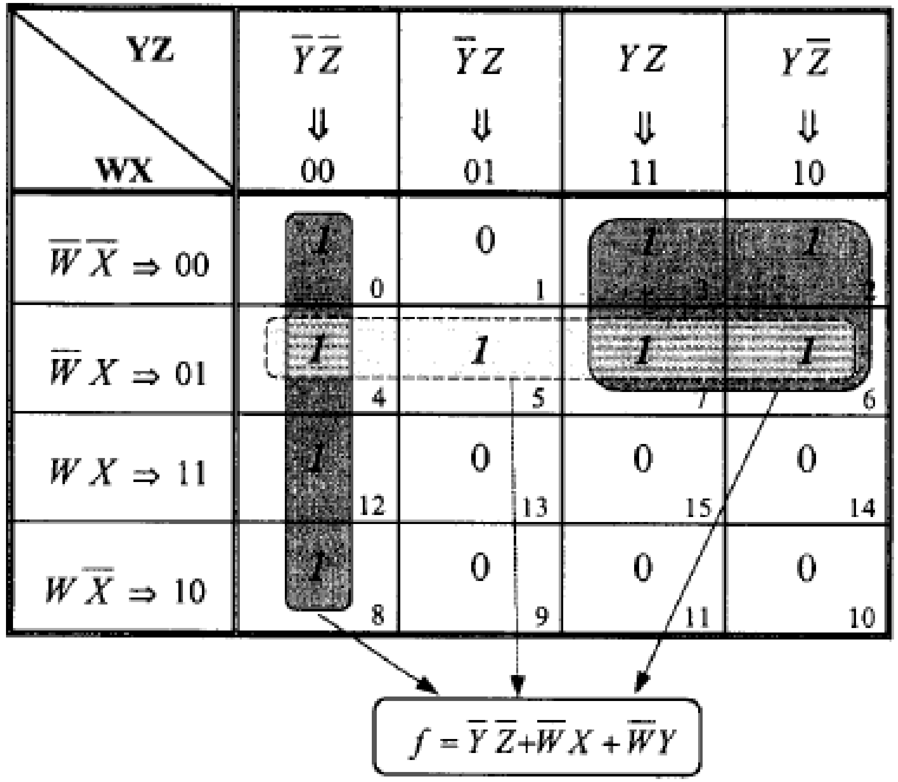


Figura 1.45. Minimización mediante diagrama de V-K de la función de cuatro variables $f = \Sigma m(0,2,3,4,5,6,7,8)$.

YZ \ WX	$\overline{Y}\,\overline{Z} \Downarrow$ 00	$\overline{Y}Z \Downarrow$ 01	$YZ \Downarrow$ 11	$Y\overline{Z} \Downarrow$ 10
$\overline{W}\,\overline{X} \Rightarrow 00$	1 (0)	0 (1)	0 (3)	1 (2)
$\overline{W}X \Rightarrow 01$	1 (4)	0 (5)	0 (7)	0 (6)
$WX \Rightarrow 11$	1 (12)	0 (13)	0 (15)	0 (14)
$W\overline{X} \Rightarrow 10$	1 (8)	0 (9)	0 (11)	1 (10)

$$\boxed{f = \overline{Y}\,\overline{Z} + \overline{X}\,\overline{Z}}$$

Figura 1.46 Minimización mediante diagrama de V-K de la función *f=Σm(0,2,4,8,10,12)*.

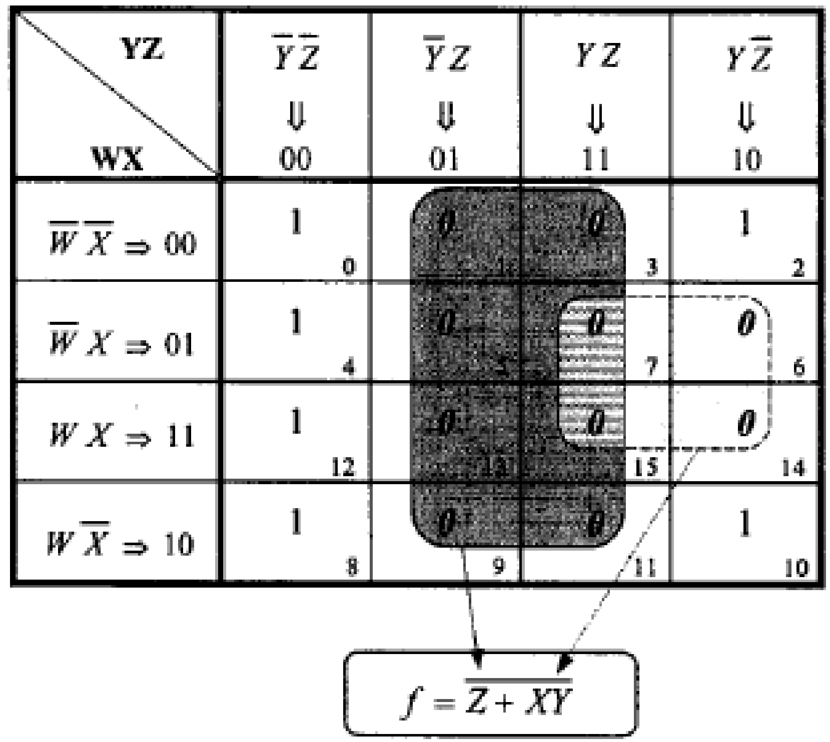


Figura 1.47. Minimización de la función $f=\Sigma m(0,2,4,8,10,12)$ agrupando los ceros.

YZ \ WX	$\overline{Y}\,\overline{Z}$ ⇓ 00	$\overline{Y}Z$ ⇓ 01	YZ ⇓ 11	$Y\overline{Z}$ ⇓ 10
$\overline{W}\,\overline{X} \Rightarrow 00$	0 (0)	*1* (1)	*1* (3)	0 (2)
$\overline{W}X \Rightarrow 01$	0 (4)	0 (5)	0 (7)	*1* (6)
$WX \Rightarrow 11$	***d*** (12)	*1* (13)	*1* (15)	*1* (14)
$W\overline{X} \Rightarrow 10$	***d*** (8)	*1* (9)	***d*** (11)	0 (10)

$$f = WX + \overline{X}Z + XY\overline{Z}$$

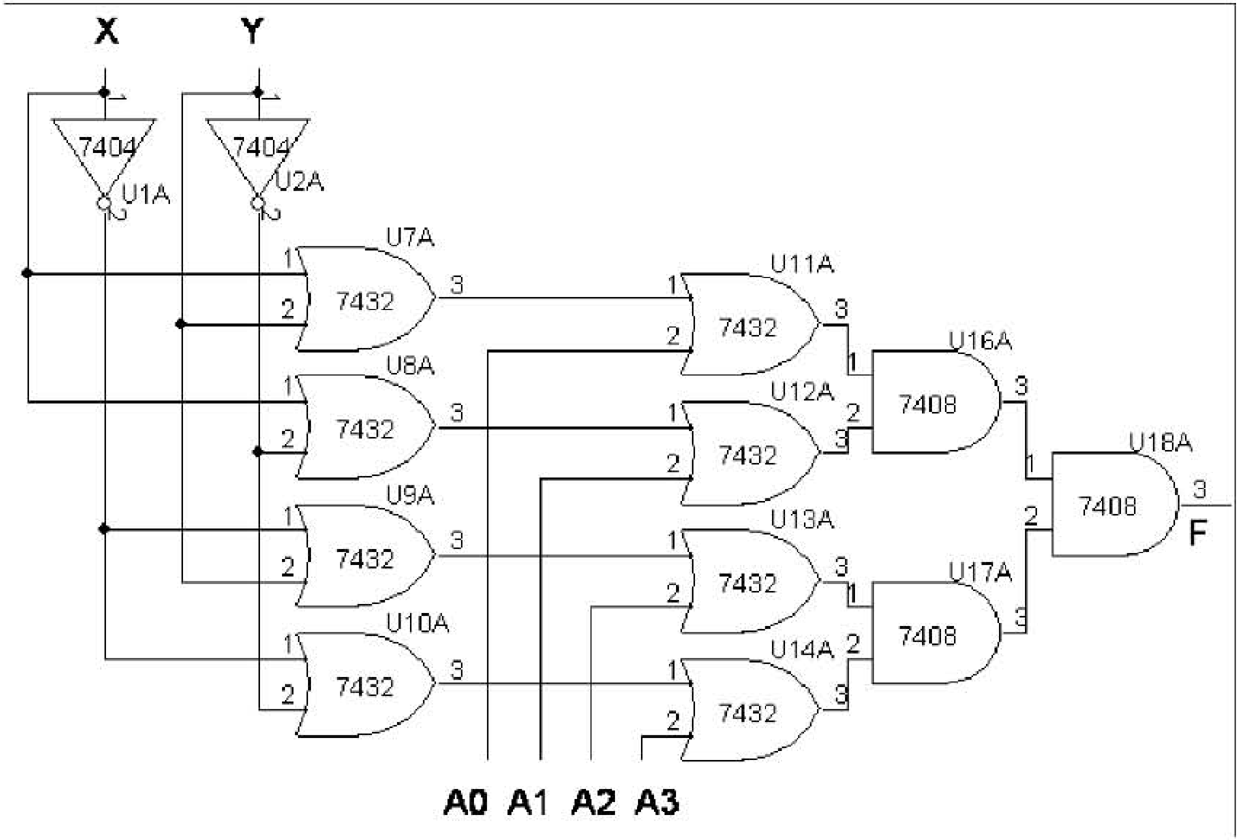
Figura 1.48. Minimización de la función $f=\Sigma m(1,3,6,9,13,14,15)+\Sigma d(8,11,12)$.

- Clase ponferrada
  - http://www.intecca.uned.es/portalavip/grabacion.php?ID_Grabacion=58250&ID_Sala=61016&hashData=57bd49de0e2c0342924e76fcc0a0e0a9&paramsToCheck=SURfR3JhYmFjaW9uLElEX1NhbGEs
- Karnaugh
  - http://www.youtube.com/watch?v=DwdyHY3-nGs

SELECTION				ACTIVE-HIGH DATA		
				M = H	M = L; ARITHMETIC OPERATIONS	
S3	S2	S1	S0	LOGIC FUNCTIONS	$\overline{C}_n$ = H (no carry)	$\overline{C}_n$ = L (with carry)
L	L	L	L	$F = \overline{A}$	$F = A$	$F = A$ PLUS 1
L	L	L	H	$F = \overline{A + B}$	$F = A + B$	$F = (A + B)$ PLUS 1
L	L	H	L	$F = \overline{A}B$	$F = A + \overline{B}$	$F = (A + \overline{B})$ PLUS 1
L	L	H	H	$F = 0$	$F =$ MINUS 1 (2's COMPL)	$F =$ ZERO
L	H	L	L	$F = \overline{AB}$	$F = A$ PLUS $A\overline{B}$	$F = A$ PLUS $A\overline{B}$ PLUS 1
L	H	L	H	$F = \overline{B}$	$F = (A + B)$ PLUS $A\overline{B}$	$F = (A + B)$ PLUS $A\overline{B}$ PLUS 1
L	H	H	L	$F = A \oplus B$	$F = A$ MINUS B MINUS 1	$F = A$ MINUS B
L	H	H	H	$F = A\overline{B}$	$F = A\overline{B}$ MINUS 1	$F = A\overline{B}$
H	L	L	L	$F = \overline{A} + B$	$F = A$ PLUS AB	$F = A$ PLUS AB PLUS 1
H	L	L	H	$F = \overline{A \oplus B}$	$F = A$ PLUS B	$F = A$ PLUS B PLUS 1
H	L	H	L	$F = B$	$F = (A + \overline{B})$ PLUS AB	$F = (A + \overline{B})$ PLUS AB PLUS 1
H	L	H	H	$F = AB$	$F = AB$ MINUS 1	$F = AB$
H	H	L	L	$F = 1$	$F = A$ PLUS A†	$F = A$ PLUS A PLUS 1
H	H	L	H	$F = A + \overline{B}$	$F = (A + B)$ PLUS A	$F = (A + B)$ PLUS A PLUS 1
H	H	H	L	$F = A + B$	$F = (A + \overline{B})$ PLUS A	$F = (A + \overline{B})$ PLUS A PLUS 1
H	H	H	H	$F = A$	$F = A$ MINUS 1	$F = A$

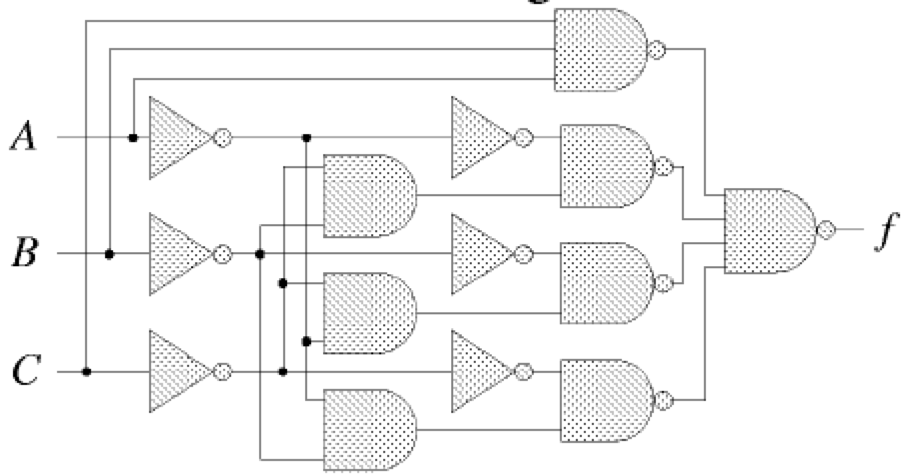
† Each bit is shifted to the next more significant position.

1.- El circuito de la figura corresponde a la implementación de la función universal realizada con términos máximos. ¿Qué función realiza cuando la palabra de programación es: A= (A3 A2 A1 A0) =1001?



a) $F = X \oplus Y$, b) $\boxed{F = \overline{X \oplus Y}}$ c) $F = X + \overline{Y}$, d) Ninguna de las anteriores.

4.- ¿Cuál de las 4 soluciones dadas es la función que realiza el circuito de la figura?.



$\boxed{\textbf{a)} \quad f = A \oplus B \oplus C}$

b) $f = \overline{(A\,B\,C)\,(A\,\overline{B}\,\overline{C})\,(\overline{A}\,B\,\overline{C})\,(\overline{A}\,\overline{B}\,C)}$

c) $f = (A\,B\,C)\,(A\,\overline{B}\,\overline{C})\,(\overline{A}\,B\,\overline{C})\,(\overline{A}\,\overline{B}\,C)$

d) $f = \overline{A \oplus B \oplus C}$

La función XOR de tres variables es:

$$A \oplus B \oplus C = (A \oplus B) \oplus C = (\overline{A}\,B + A\,\overline{B}) \oplus C = \overline{(\overline{A}\,B + A\,\overline{B})}\,C + (\overline{A}\,B + A\,\overline{B})\,\overline{C} =$$

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

Simplificación de funciones.

Ejemplo:

Simplificar la siguiente función aplicando las leyes del Algebra de Boole.

XZ’Y+(XZ’Y+ZX’)(Y(Z+X)+Y’Z+Y’XZ’)

Comenzamos desarrollando la función:

XZ’Y + (XZ’Y+ZX’)(Y(Z+X) +Y’Z+Y’XZ’)

XZ’Y + (XZ’Y+ZX’)(YZ+YX +Y’Z+Y’XZ’)

Eliminamos las variables en una suma que aparezcan en su forma normal y negada:

XZ’Y + (XZ’Y+ZX’)(YZ +YX+Y’Z +Y’XZ’)

XZ’Y + (XZ’Y+ZX’)(Z (Y+Y’)+YX +Y’XZ’)

Desarrollamos el producto:

XZ’Y + (XZ’Y+ZX’) (Z +YX +Y’XZ’)

XZ’Y + XZ’Y (Z +YX +Y’XZ’)+ZX’(Z +YX +Y’XZ’)

XZ’Y + XZ’YZ+XZ’YYX +XZ’YY’XZ’+ZX’Z +ZX’YX+ZX’Y’XZ’

Eliminamos los términos que tengan un producto entre una variable y su negada:

XZ’Y + $\cancel{XZ'YZ}$+XZ’YYX +$\cancel{XZ'YY'XZ'}$+ZX’Z +$\cancel{ZX'YX}$+$\cancel{ZX'Y'XZ'}$

XZ’Y + XZ’YYX+ ZX’Z

Eliminamos las variables repetidas en los términos producto:

XZ’Y + XZ’YYX+ ZX’Z

XZ’Y + XZ’Y+ ZX’

XZ’Y + ZX’

Simplificación de funciones. Mapas de Karnaugh.

Las tablas se rellenan colocando un 1 en las casillas de los términos que cumplen la función en forma de suma de productos.

Para simplificar se toman los mayores grupos posibles formados por potencias de 2 que sean adyacentes.

a	b	c	d	*f(a, b, c)*
0	0	0	0	**1**
0	0	0	1	**0**
0	0	1	0	**1**
0	0	1	1	**1**
0	1	0	0	**1**
0	1	0	1	**0**
0	1	1	0	**1**
0	1	1	1	**1**
1	0	0	0	**1**
1	0	0	1	**0**
1	0	1	0	**1**
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**

Karnaugh de 4 variables

ab \ cd	**00**	**01**	**11**	**10**
00	**1**	**0**	**1**	**1**
01	**1**	**0**	**1**	**1**
11	**0**	**0**	**0**	**0**
10	**1**	**0**	**0**	**1**

Las variables “c” y “a” aparecen en sus formas normal y negada.
Este término queda como ***b’d’.***

La “d” aparece en ambas formas, al igual que la “b” por lo tanto estas dos variables quedarán eliminadas.
Este término queda como ***a’c.***

La variable “b” y la “c” aparecen en sus formas normales y negadas.
Este término queda como ***a’d’.***

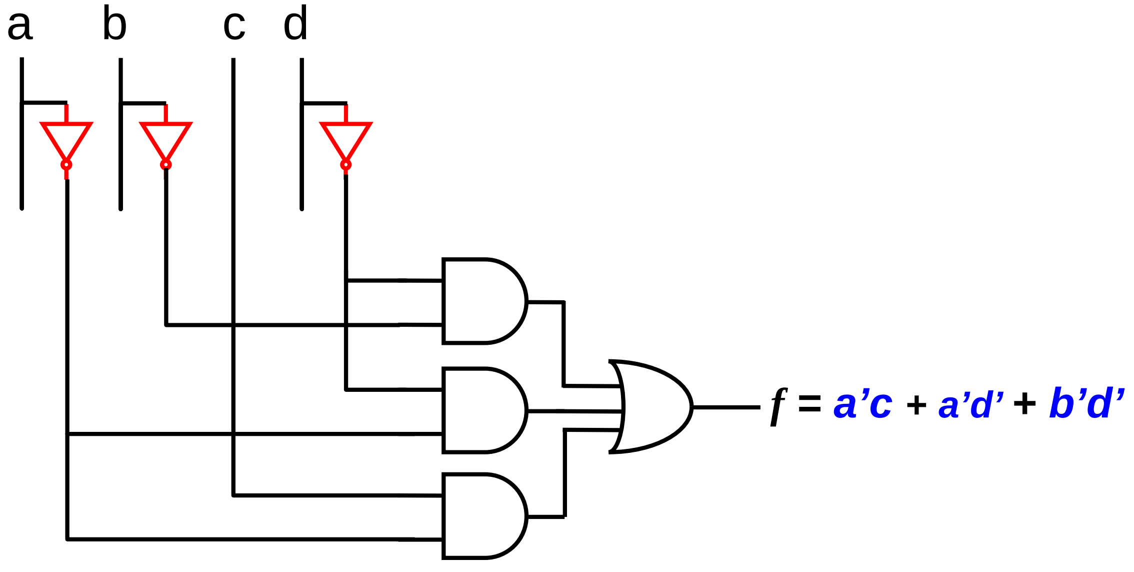
La función simplificada que como:

$$f = a'c + a'd' + b'd'$$

Simplificación de funciones. Mapas de Karnaugh.

El circuito lógico que determina la siguiente función quedará como sigue:

$$f = a'c + a'd' + b'd'$$



Funciones incompletamente definidas

Se dice que una función está ***completamente definida*** si para cada una de las combinaciones posibles de sus variables, la función toma un valor determinado, ya sea 1 o 0.

Aquellas funciones en las que alguna de sus combinaciones de variables toma cualquiera de los valores posibles, es decir, que es indiferente que sea un 1 o un 0, se llaman ***incompletas*** y se representa con una “X”

$$f_{completa}(c, b, a) = \Sigma_3(1, 3, 5, 6, 7)$$

$$f_{incompleta}(c, b, a) = \Sigma_3(1, 3, 6, 7) + X(2, 4, 5)$$

c	b	a	*Completa*	*Incompleta*
0	0	0	**0**	**0**
0	0	1	**1**	**1**
0	1	0	**0**	**X**
0	1	1	**1**	**1**
1	0	0	**0**	**X**
1	0	1	**1**	**X**
1	1	0	**1**	**1**
1	1	1	**1**	**1**

Simplificación de funciones. Mapas de Karnaugh.

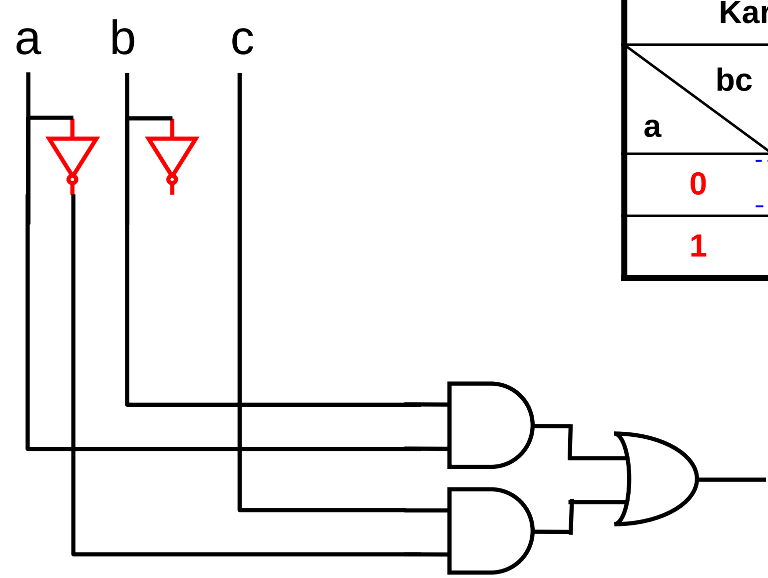
Ejemplo:
Determinar la tabla de verdad y obtener un circuito lógico que satisfaga un sistema digital cuya función incompletamente definida es la siguiente:

$$f = \Sigma_3(0, 6, 7) + X(1, 2, 5)$$

Tabla de verdad:

a	b	c	*f(c, b, a)*
0	0	0	**1**
0	0	1	**X**
0	1	0	**X**
0	1	1	**0**
1	0	0	**0**
1	0	1	**X**
1	1	0	**1**
1	1	1	**1**

Circuito lógico:



$f = ab + a'c$

Simplificación por Karnaugh:

Karnaugh de 4 variables				
bc / **a**	**00**	**01**	**11**	**10**
0	**1**	**X**	**0**	**X**
1	**0**	**X**	**1**	**1**

$$f = ab + a'c$$

Karnaugh de 4 variables				
ab / cd	00	01	11	10
00				
01				
11				
10				

Karnaugh de 5 variables								
abc / de	000	001	011	010	110	111	101	100
00								
01								
11								
10								