EQUIPOS MICROPROGRAMABLES

TEMA 2 – ALGEBRA DE BOOLE

IES FONTEXERIA
(MUROS)

Profesor: Javier Fraga Iriso

1. Introducción

En el algebra de Boole las variables (entradas o salidas de un sistema) únicamente pueden tomar dos estados CERO o UNO. Estos estados a nivel eléctrico representan que no hay tensión y que hay tensión, por ejemplo 5 voltios, respectivamente.

Ejemplo:

0 (apagado)

MOTOR

1 (encendido)

2. Función lógica y tabla de verdad

Para representar las posibles combinaciones que pueden tomar las diferentes variables se recurre a dos elementos, la función lógica y la tabla de verdad.

La función lógica es una variable binaria cuyo valor depende de una expresión algebraica representada por otras variables binarias. Generalmente la variable representada por la función es una salida del sistema cuyo valor depende de las entradas.

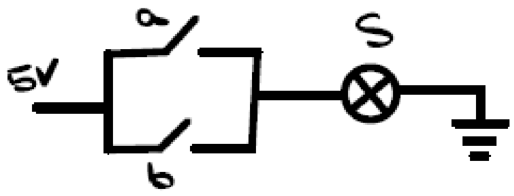
$$\boldsymbol{f\ (a, b, c) = a \cdot b + b \cdot c}$$

La tabla de verdad es cuadro formado por tantas columnas como variables contenga la función y tantas filas como posibles combinaciones binarias puedan producir estas variables, es decir, $n^{\underline{o}}\ filas = 2^{n^{\underline{o}}\ variables}$.

a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

3. Operaciones básicas

3.1. Suma lógica → $S = a + b$

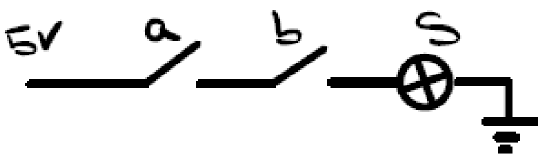
Circuito eléctrico:	Tabla de verdad:
	

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

Una de las dos variables de entrada tiene que encontrarse en el estado UNO para que la salida del sistema tome el estado UNO.

Circuito electrónico: *Puerta OR*	
ASA	**DIN**

3.2. Producto lógico → $S = a \cdot b$

Circuito eléctrico:	Tabla de verdad:
	

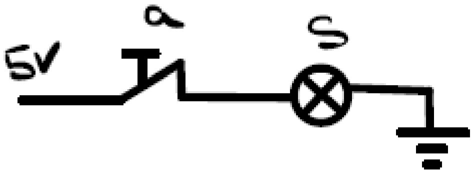
a	0	1	0	1
b	0	0	1	1
S	0	0	0	1

Las dos variables de entrada tienen que encontrarse en el estado UNO para que la salida del sistema tome el estado UNO.

Circuito electrónico: *Puerta AND*	
ASA	**DIN**

3.3. Función negación → $S = \overline{a}$

Circuito eléctrico:	Tabla de verdad:



a	0	1
S	1	0

La variable salida del sistema tomará el estado contrario presente en la variable de entrada.

Circuito electrónico: *Puerta NOT*	
ASA	**DIN**

3.4. Función suma negada → $S = \overline{a + b}$

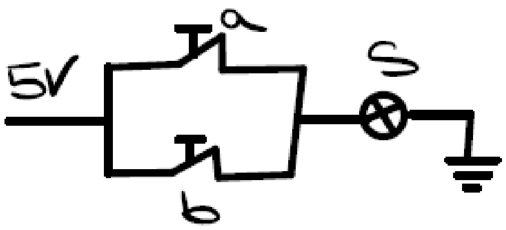
Circuito eléctrico:	Tabla de verdad:

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

La variable de salida se encontrará en estado UNO únicamente cuando ambas variables entrada se encuentren en estado CERO.

Circuito electrónico: *Puerta NOR*	
ASA	**DIN**

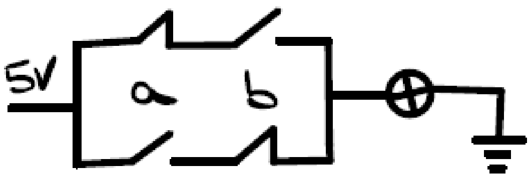
3.5. Función producto negada → $S = \overline{a \cdot b}$

<table>
<tr><th>Circuito eléctrico:</th><th>Tabla de verdad:</th></tr>
<tr><td>
</td><td>
<table>
<tr><td>a</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>b</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>S</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table>
</td></tr>
</table>

Ambas variables de entrada deben de estar en estado UNO para que la variable de salida se encuentre en estado CERO.

Circuito electrónico: *Puerta NAND*	
ASA	**DIN**

3.6. Función OR exclusiva → $S = \bar{a} \cdot b + a \cdot \bar{b} = a \oplus b$

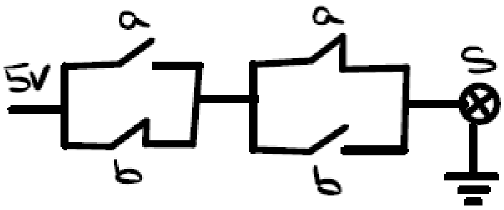
<table>
<tr><th>Circuito eléctrico:</th><th>Tabla de verdad:</th></tr>
<tr><td>
</td><td>
<table>
<tr><td>a</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>b</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>S</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
</table>
</td></tr>
</table>

La variable de salida se encontrará en estado UNO cuando las dos variables de entrada tomen estados diferentes.

Circuito electrónico: *Puerta XOR*	
ASA	**DIN**

NOTA: Cuando existen tres o más variables, da como salida UNO cuando el número de UNOS de las variables de entrada es impar y CERO en el caso contrario.

3.7. Función OR exclusiva negada → $S = \overline{a \oplus b}$

<table>
<tr><th>Circuito eléctrico:</th><th>Tabla de verdad:</th></tr>
<tr><td>
</td><td>

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

</td></tr>
</table>

La variable de salida se encontrará en estado UNO cuando las dos variables de entrada tomen estados iguales.

Circuito electrónico: *Puerta XNOR*	
ASA	**DIN**

4. Postulados

1. $a + 1 = 1$
2. $a + 0 = a$
3. $a \cdot 1 = a$
4. $a \cdot 0 = 0$
5. $a + a = a$
6. $a \cdot a = a$
7. $a + \bar{a} = 1$
8. $a \cdot \bar{a} = 0$
9. $a = \bar{\bar{a}}$

5. Propiedades

5.1. Propiedad conmutativa

$$a + b = b + a$$

$$a \cdot b = b \cdot a$$

5.2. Propiedad asociativa

$$a + b + c = a + (b + c)$$

$$a \cdot b \cdot c = (a \cdot b) \cdot c$$

5.3. Propiedad distributiva

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

$$a + (b \cdot c) = (a + b) \cdot (a + c)$$

6. Teoremas

6.1. Ley de absorción

$$a + a \cdot b = a \rightarrow a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$$

$$a \cdot (b + a) = a \rightarrow a \cdot (b + a) = a \cdot b + a \cdot a = a \cdot b + a = a$$

6.2. Segundo teorema

$$a + \bar{a} \cdot b = a + b \rightarrow a + \bar{a} \cdot b = (a + \bar{a}) \cdot (a + b) = a + b$$

$$b \cdot (a + \bar{b}) = a \cdot b \rightarrow b \cdot \left(a + \bar{b}\right) = b \cdot a + b \cdot \bar{b} = b \cdot a$$

6.3. Leyes de Morgan

$$\overline{a + b} = \bar{a} \cdot \bar{b}$$

$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

7. Expresiones canónicas

Son expresiones algebraicas obtenidas a través de las tablas de verdad que definen el sistema.

7.1. Primera expresión canónica (MINTERMS)

Suma de los productos lógicos que dan a la función el valor UNO.

a	b	c	S
0	0	0	1_0 ←
0	0	1	0_1 ←
0	1	0	0_2 ←
0	1	1	0_3 ←
1	0	0	0_4 ←
1	0	1	1_5 ←
1	1	0	1_6 ←
1	1	1	0_7 ←

$$\rightarrow S = \sum_{\underset{\text{nº variables}}{3}} \overbrace{(0,5,6)}^{\text{posiciones}}$$

$$S = \underbrace{\bar{a}\cdot\bar{b}\cdot\bar{c}}_{\text{fila 0}} + \underbrace{a\cdot\bar{b}\cdot c}_{\text{fila 5}} + \underbrace{a\cdot b\cdot\bar{c}}_{\text{fila 6}}$$

7.2. Segunda expresión canónica (MAXTERMS)

Producto de las sumas lógicas que dan a la función el valor CERO.

$$\rightarrow S = \prod_{\underset{\text{nº variables}}{3}} \overbrace{(1,2,3,4,7)}^{\text{posiciones}}$$

$$S = \underbrace{(a+b+\bar{c})}_{\text{fila 1}}\cdot\underbrace{(a+\bar{b}+c)}_{\text{fila 2}}\cdot\underbrace{(a+\bar{b}+\bar{c})}_{\text{fila 3}}\cdot\underbrace{(\bar{a}+b+c)}_{\text{fila 4}}\cdot\underbrace{(\bar{a}+\bar{b}+\bar{c})}_{\text{fila 7}}$$

8. Simplificaciones de funciones – Método algebraico

Para este método no existe ninguna regla o pasos a seguir. Para efectuar la simplificación se recurre a los postulados, propiedades y teoremas del algebra de Boole y mediante su correcta aplicación es posible simplificar funciones.

Ejemplos:

a) $\boldsymbol{a\cdot\bar{b}+a\cdot\left(\overline{b+c}\right)+b\cdot\left(\overline{b+c}\right)}$

Aplicando Ley de Morgan → $a\cdot\bar{b}+a\cdot\bar{b}\cdot\bar{c}+\cancel{b\cdot\bar{b}\cdot\bar{c}}$

Aplicando propiedad distributiva → $a\cdot\bar{b}\cdot(1+\bar{c})=\boldsymbol{a\cdot\bar{b}}$

b) $\boldsymbol{[a\cdot\bar{b}\cdot(c+b\cdot d)+\bar{a}\cdot\bar{b}]\cdot c}$

Desarrollando → $\left[a\cdot\bar{b}\cdot c+\cancel{a\cdot\bar{b}\cdot b}\cdot d+\bar{a}\cdot\bar{b}\right]\cdot c=a\cdot\bar{b}\cdot c\cdot c+\bar{a}\cdot\bar{b}\cdot c$

Tomando $\bar{b}\cdot c$ como factor común → $\bar{b}\cdot c\cdot(a+\bar{a})=\boldsymbol{\bar{b}\cdot c}$

c) $\boldsymbol{\cancel{(a\cdot 0)\cdot(b+b)}+\left(b+\bar{b}\right)\cdot(a+a)+\cancel{(b+1)\cdot(\bar{c}\cdot c)}}=0\cdot b+1\cdot a+1\cdot 0=\boldsymbol{a}$

d) $\boldsymbol{\cancel{(b+1)\cdot a\cdot\bar{a}}+a+c\cdot c+\cancel{b\cdot 0}+c}=a+c+c=\boldsymbol{a+c}$

e) $\boldsymbol{\bar{a}\cdot b\cdot\bar{c}+b\cdot c\cdot(\bar{c}+1)+a\cdot b\cdot\bar{c}\cdot(a+a)}$

Desarrollando → $\bar{a}\cdot b\cdot\bar{c}+b\cdot c+a\cdot b\cdot\bar{c}$

Tomando $b\cdot\bar{c}$ como factor común → $b\cdot\bar{c}\cdot(a+a)+b\cdot c=b\cdot(\bar{c}+c)=\boldsymbol{b}$

f) $\boldsymbol{a\cdot\bar{b}\cdot\bar{c}+a\cdot\bar{b}\cdot\bar{c}\cdot d+a\cdot\bar{b}}$

Tomando $a\cdot\bar{b}$ como factor común → $a\cdot\bar{b}\cdot(\bar{c}+\bar{c}\cdot d+1)=\boldsymbol{a\cdot\bar{b}}$

g) $\boldsymbol{(a\cdot\bar{c}+c)\cdot(\overline{a+c})\cdot(b+c+a+\bar{a})}=(c+a)\cdot\bar{a}\cdot\bar{c}=c\cdot\bar{a}\cdot\bar{c}+a\cdot\bar{a}\cdot\bar{c}=\boldsymbol{0}$

h) $\boldsymbol{\bar{a}+\bar{b}+\bar{c}+\overline{a\cdot b\cdot c}}=\bar{a}+\bar{b}+\bar{c}+\bar{a}+\bar{b}+\bar{c}=\boldsymbol{\bar{a}+\bar{b}+\bar{c}}$

9. Simplificación de funciones – Método Karnaugh

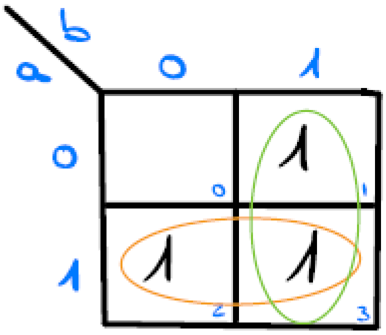
9.1. Dos variables por MINTERMS

PASOS:

1. Situar las variables por orden de peso en la tabla. Las de mayor peso se situarán primero abajo y después a la derecha.
2. Numerar los recuadros
3. Situar los UNOS
4. Agrupar los UNOS adyacentes en grupos lo más grandes posibles, siempre que sean potencias de dos.

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

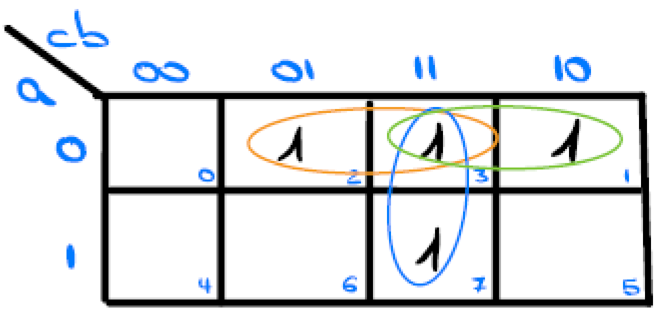
$$S = \sum_{2}(1, 2, 3)$$



$$S = a + b$$

9.2. Tres variables por MINTERMS

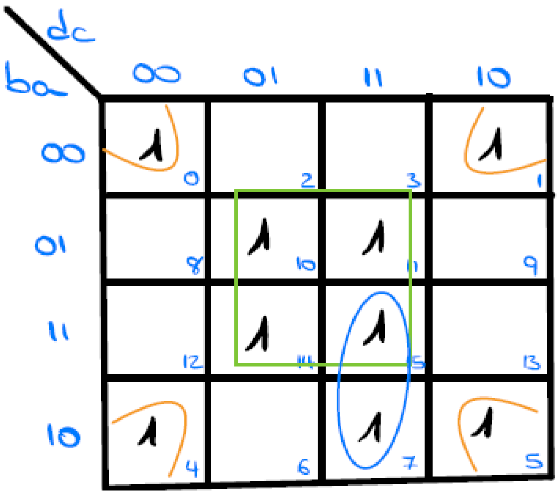
$$f = \sum_{3}(1, 2, 3, 7)$$



$$f = \bar{a} \cdot b + \bar{a} \cdot c + b \cdot c$$

9.3. Cuatro variables por MINTERMS

$$f = \sum_{4}(0, 1, 4, 5, 7, 10, 11, 14, 15)$$



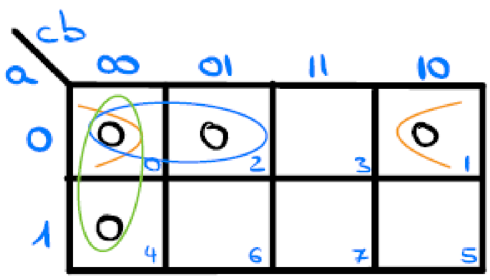
$$f = a \cdot c + \bar{a} \cdot \bar{c} + b \cdot c \cdot d$$

9.4. Simplificación por MAXTERMS

Se realizan los mismos pasos que en MINTERMS pero en este caso se sitúan y posteriormente agrupan los CEROS.

a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$S = \prod_{3}(0, 1, 2, 4)$$

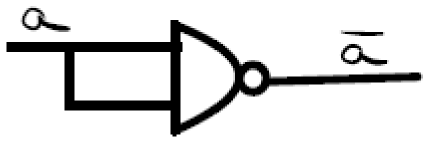


$$S = (b + c) \cdot (a + b) \cdot (a + c)$$

10. Circuitos con puertas NAND

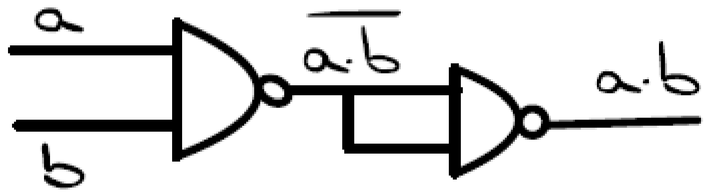
10.1. Circuito inversor (puerta NOT)

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



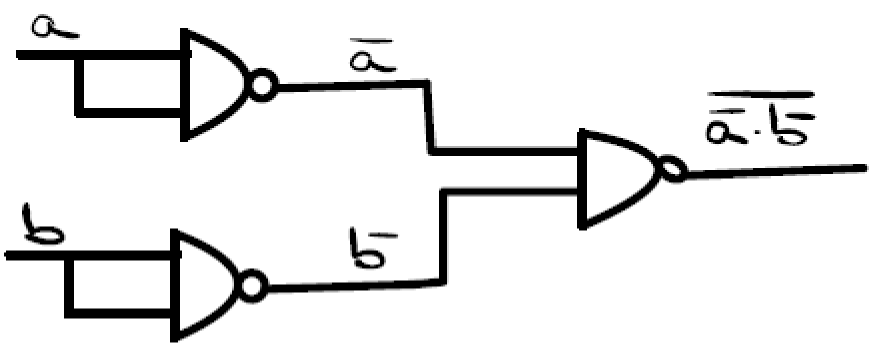
10.2. Producto lógico (puerta AND)

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



10.3. Suma lógica (puerta OR)

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



10.4. Método implementación con puertas NAND

Para la realización de esta explicación se parte de la siguiente función:

$$\boldsymbol{f = a + b \cdot \left(c + \overline{d}\right) + e \cdot f}$$

PASOS:

1 Se complementa o niega dos veces la función completa.

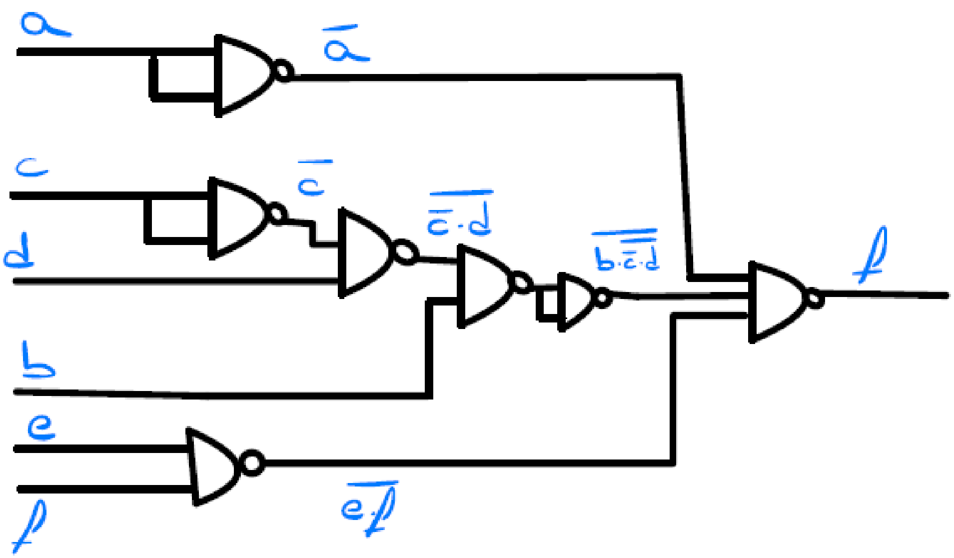
$$\boldsymbol{f = \overline{\overline{a + b \cdot \left(c + \overline{d}\right) + e \cdot f}}}$$

2 Se hacen dos nuevas inversiones a cada uno de los sumandos que aparezcan como factores.

$$\boldsymbol{f = \overline{\overline{a + b \cdot \left(\overline{\overline{c + \overline{d}}}\right) + e \cdot f}}}$$

3 Se opera siguiendo las leyes de Morgan, sustituyendo las sumas presentes en la función por multiplicaciones.

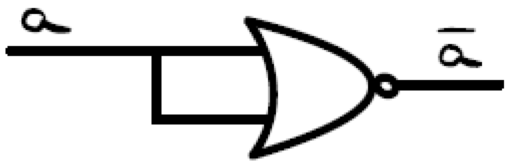
$$\boldsymbol{f = \overline{\overline{a + b \cdot \left(\overline{\overline{c} + d}\right) + e \cdot f}} = \overline{\overline{a} \cdot \overline{b \cdot \overline{\overline{c} \cdot d}} \cdot \overline{e \cdot f}}}$$



11. Circuitos con puertas NOR

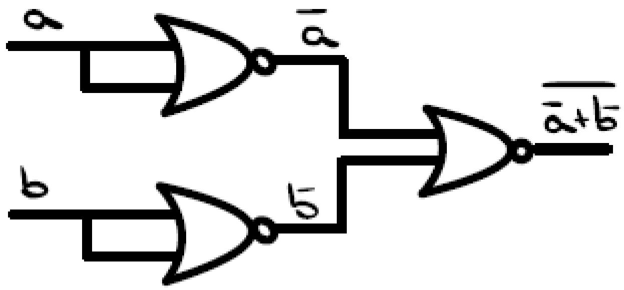
11.1. Circuito inversor (puerta NOT)

$$f = \overline{a} = \overline{a + a}$$



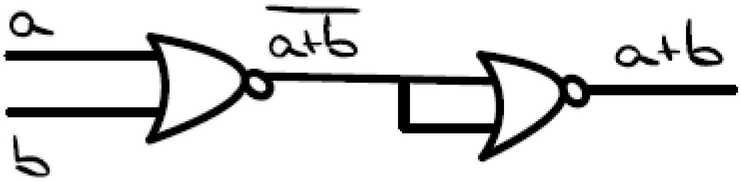
11.2. Producto lógico (puerta AND)

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



11.3. Suma lógica (puerta OR)

$$f = a + b$$



11.4. Método implementación con puertas NOR

Para la realización de esta explicación se parte de la siguiente función:

$$f = a + b \cdot \left(c + \bar{d}\right) + e \cdot f$$

PASOS:

1 Se complementa o niega dos veces la función completa.

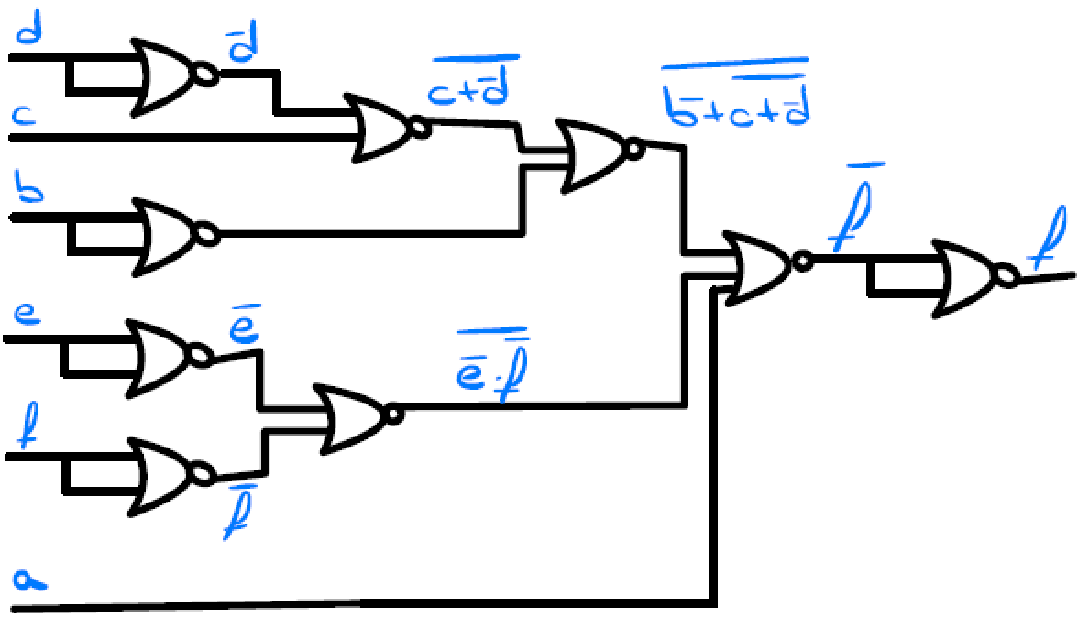
$$f = \overline{\overline{a + b \cdot \left(c + \bar{d}\right) + e \cdot f}}$$

2 Se hacen dos nuevas inversiones a cada uno de los productos que aparezcan como sumandos.

$$f = \overline{\overline{a + \overline{\overline{b \cdot \left(c + \bar{d}\right)}} + \overline{\overline{e \cdot f}}}}$$

3 Se opera siguiendo las leyes de Morgan, sustituyendo los productos presentes en la función por sumas.

$$f = \overline{\overline{a + \overline{\bar{b} + \overline{c + \bar{d}}} + \overline{\bar{e} + \bar{f}}}}$$



12. Sistemas Multifunción

Se produce cuando aparecen varias funciones de salida para las mismas variables.

Ejemplo: Un catamarán dispone de dos motores y son controlados por tres pulsadores. El primer pulsador se encuentra en la sala de máquinas y al ser pulsado los motores se paran en caso de estar funcionando. Los otros dos pulsadores se encuentran en la sala de control y cada uno de ellos define en qué estado se encuentra cada uno de los motores, entendiéndose que al estar pulsado uno de ellos el motor correspondiente se encuentra encendido.

ENTRADAS

- **a** → pulsador máquinas
  - **0** → NO accionado
  - **1** → accionado (motores OFF)
- **b** → pulsador motor 1
  - **0** → NO accionado
  - **1** → accionado (M1 ON)
- **C** → pulsador motor 2
  - **0** → NO accionado
  - **1** → accionado (M2 ON)

SALIDAS

- **M1** → motor 1
  - **0** → OFF
  - **1** → ON
- **M2** → motor 2
  - **0** → OFF
  - **1** → ON

a	b	c	M1	M2
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

$$S_{M1} = \overline{a} \cdot b$$

$$S_{M2} = \overline{a} \cdot c$$

13. MINTERMS redundantes

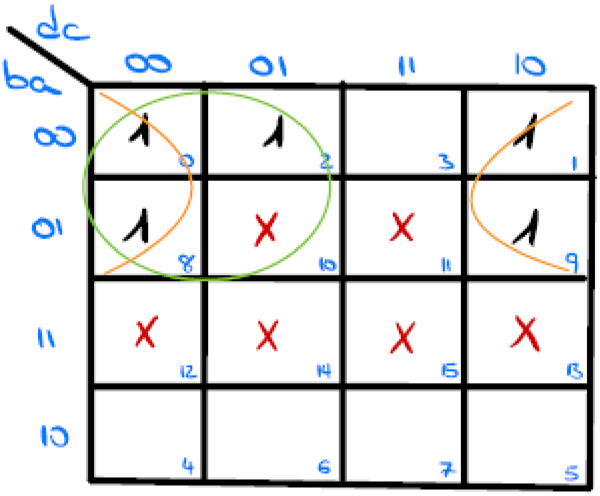
Representan situaciones que no se pueden dar a la entrada, pero podrán ser utilizadas para simplificar por Karnaugh.

Ejemplo: En un registro de 4 bits se almacena información en código BCD. Realiza el circuito lógico mínimo que detecte que el número contenido en el registro es mayor que 7 o inferior a 3.

	a	b	c	d	S
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	X
11	1	0	1	1	X
12	1	1	0	0	X
13	1	1	0	1	X
14	1	1	1	0	X
15	1	1	1	1	X

MINTERMS

REDUNDANTES



$$S = \overline{b} \cdot \overline{c} + \overline{b} \cdot \overline{d}$$