BLOQUE IV: CIRCUITOS Y SISTEMAS LÓGICOS

TEMA 6:

CIRCUITOS DIGITALES

Daniel Gallardo García

ÍNDICE

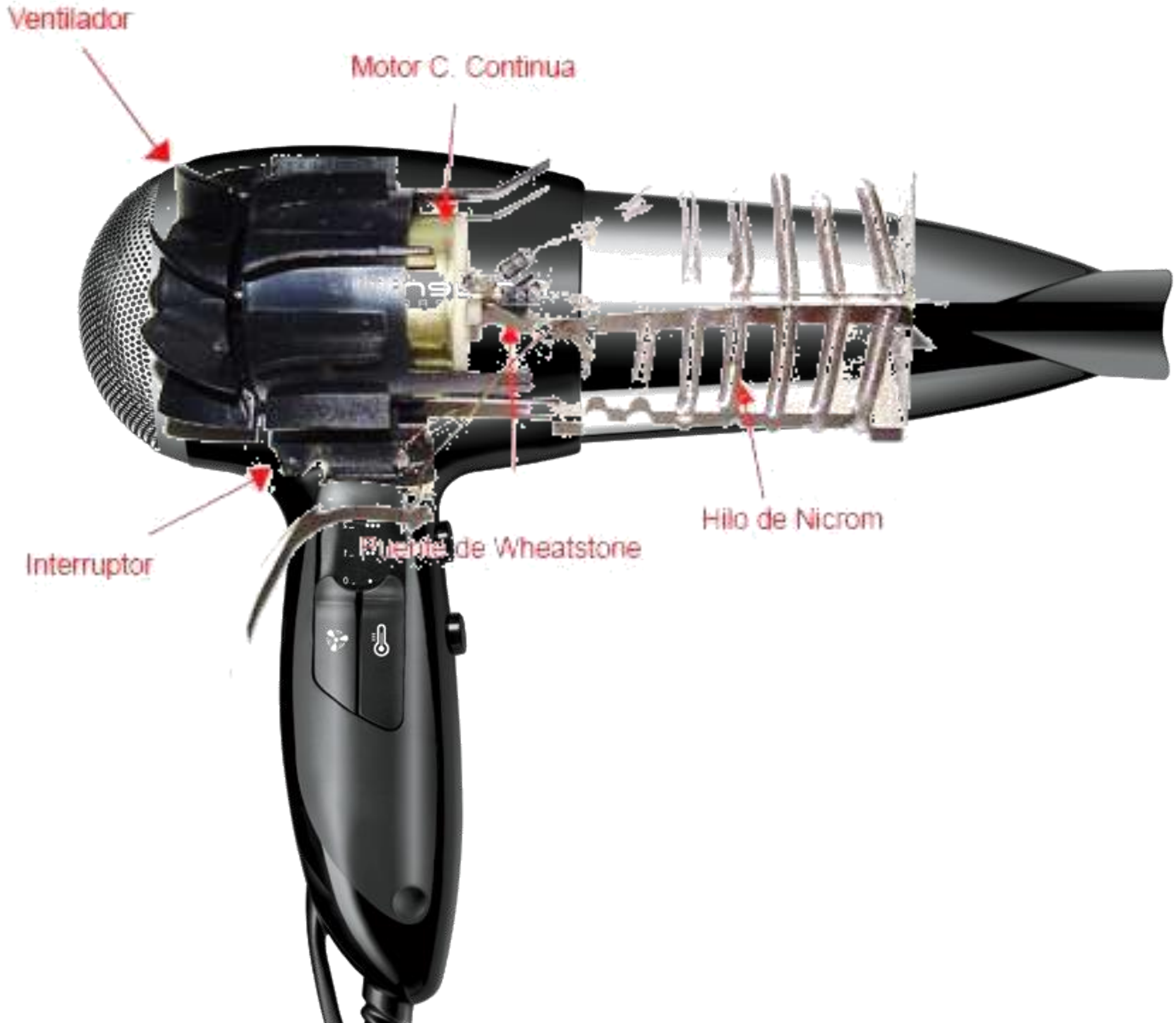
1. ¿QUÉ ES LA ELECTRÓNICA DIGITAL?

No debemos confundir “electrónica” con “electricidad”. Los **circuitos eléctricos** se diseñan con la intención de convertir la energía eléctrica en otro tipo de energía:

La energía eléctrica se transforma en...	A través de...	Ejemplo
Calor	Resistencias	Tostadora, radiador de aceite, plancha...
Luz	Lámparas	Bombillas, lámparas halógenas, tubos fluorescentes...
Movimiento	Motores	Ventilador, Sierra de calar, batidora...

Además, se caracterizan por emplear valores altos tanto en la tensión como en la intensidad (por ejemplo 230 V y 10 A).

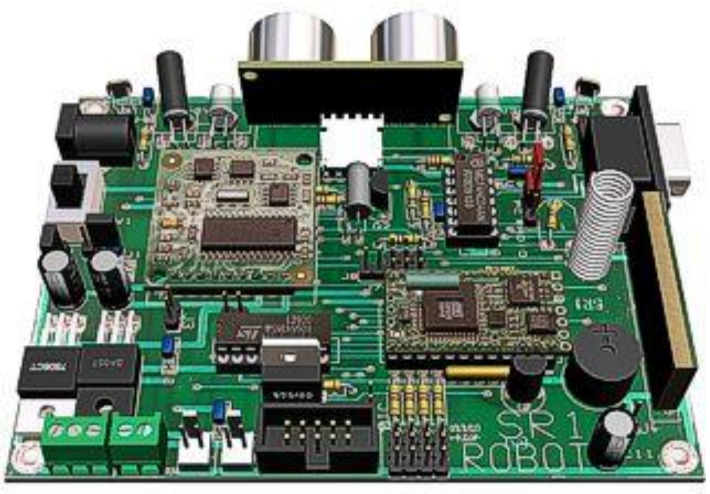
Ejemplos de aparatos eléctricos:


Ventilador
Motor C. Continua
Hilo de Nicrom
Puente de Wheatstone
Interruptor

Los **circuitos electrónicos** se diseñan con la intención de procesar información y proporcionar una respuesta según sea el resultado de dicho procesamiento. Son algo así como **circuitos inteligentes**.

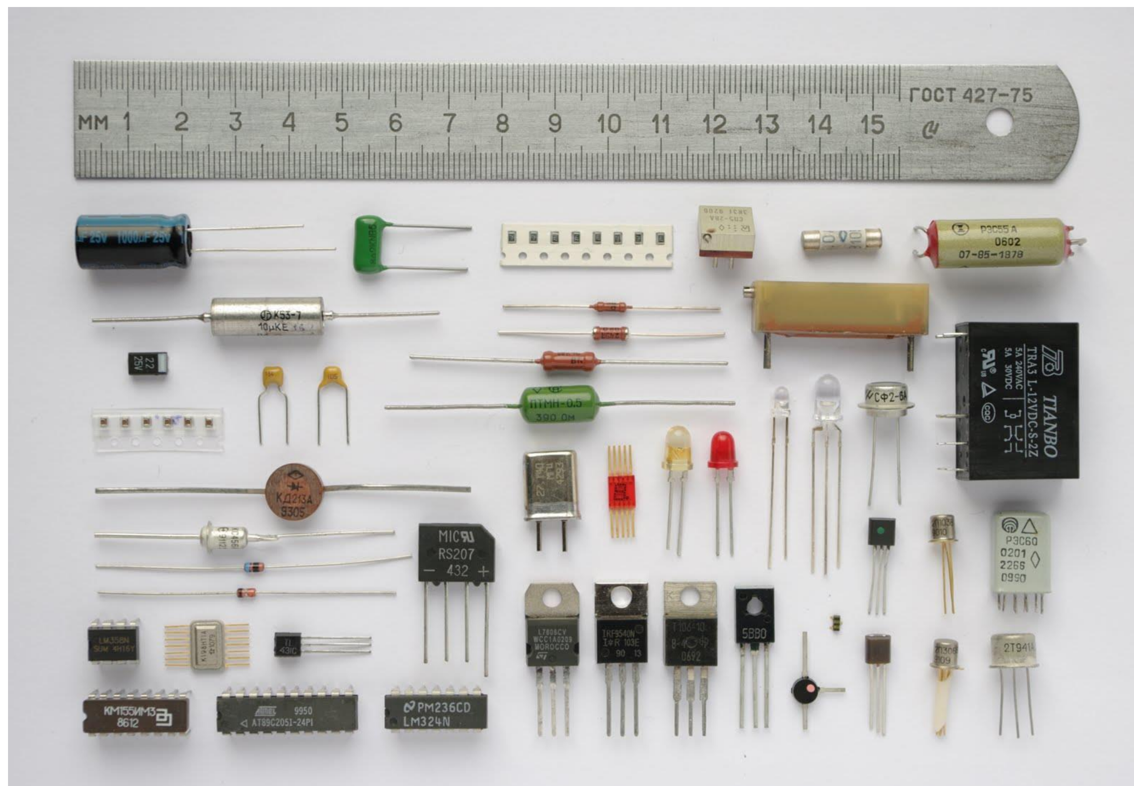
Se caracterizan por emplear valores bajos tanto en la tensión como en la intensidad (por ejemplo 5 V y 50 mA).

Ejemplos son: televisor, ordenador, calculadora, reloj, teléfono móvil, mp3...



Ejemplo de circuito impreso
(sobre una placa de baquelita)

Ejemplos de
componentes
electrónicos



Asimismo, también existen aparatos que combinan claramente ambos circuitos:

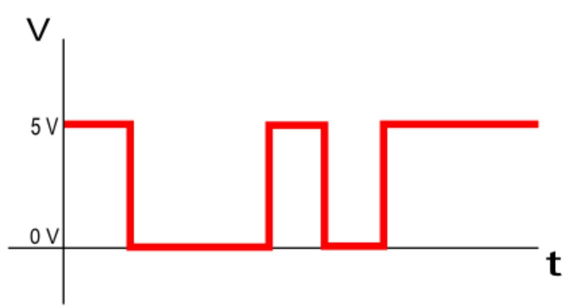
Aparato	Parte eléctrica	Parte electrónica
Lavadora	Motor del bombo	Seleccionador de programas
Ascensor	Motor del ascensor	Programador de movimiento
Alarma	Sirena de alarma	Sensores de movimiento

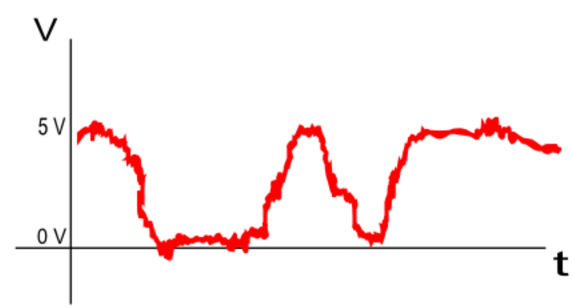
La **electrónica**, atendiendo al **tipo de señal** implicada en la aplicación en la que se destinan, se divide en:

- **Analógica**: la información que portan las señales radica tanto en sus valores concretos como en el instante de tiempo en que se produce.
- **Digital**: la información que portan las señales no radica en su valor concreto, sino en la presencia o no de pulsos (codificación).

Los sistemas digitales presentan como **principal ventaja**:

- La ventaja de utilizar circuitos electrónicos digitales es principalmente su mayor **fiabilidad** en el proceso de transmisión de la información, debido a que se interpreta un “0 lógico” a cualquier valor comprendido entre 0 y 2,4 V, y un “1 lógico” a cualquier valor comprendido entre 2,6 y 5 V. De esta forma, los circuitos electrónicos son muchísimo más inmunes al ruido (interferencias).





Incluso en el caso de que el proceso de un sistema electrónico requiera de una señal analógica, para el procesado de la información se utiliza, en la entrada, **convertidores A/D** (analógico/digital), y a la salida **convertidores D/A** (digital/analógico):

Ejemplo: señales de humo en un poblado indio. Necesidad de codificación

Cuentakilómetros binario:

0000 = 0
0001 = 1
0010 = 2
0011 = 3
0100 = 4
0101 = 5

Pasar de binario a decimal:

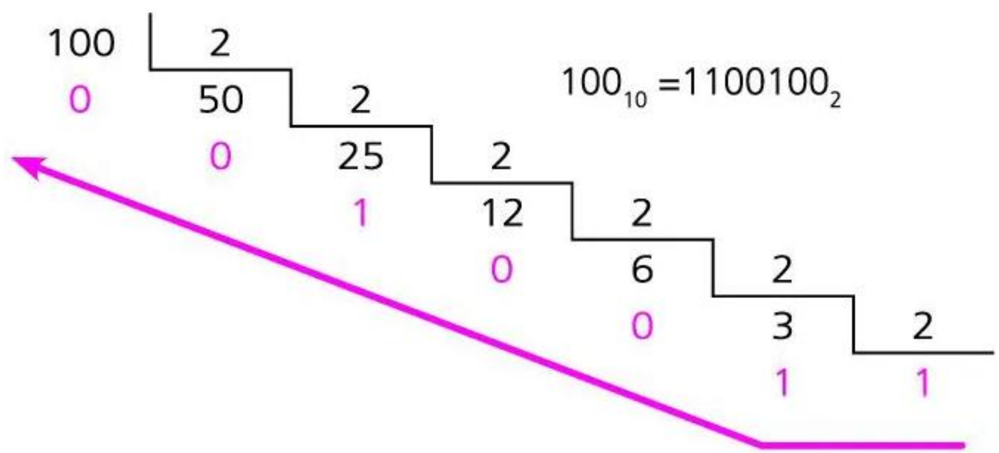
$$110101_2$$

$$1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$32 + 16 + 0 + 4 + 0 + 1 = 53$$

$$110101_2 = 53_{10}$$

Pasar de decimal a binario:



Operaciones con números binarios

```
   111111
   0011101
+  1101011
----------
  10001000
```

```
      11101
 ×      101
 ----------
      11101
     00000
+   11101
 ----------
  100100001
```

```
101010 | 110
       |-----
-110     111
 1001
 -110
  0110
 - 110
   000
```

2. ÁLGEBRA BOOLEANA. ÁLGEBRA DE CONMUTACIÓN

Un sistema matemático consistente en un conjunto B de elementos y dos operaciones binarias cerradas (+) y (·) se denomina álgebra booleana si y solo si se verifican los siguientes postulados:

P1. Las operaciones (+) y (·) son conmutativas:

$$\forall\, a, b \in B \quad \begin{cases} (i) \quad a + b = b + a \\ (ii) \quad a \cdot b = b \cdot a \end{cases}$$

P2. Cada operación (+) y (·) es distributiva sobre la otra:

$$\forall\, a, b \in B \quad \begin{cases} (i) \quad a + (b \cdot c) = (a + b) \cdot (a + \mathrm{c}) \\ (ii) \quad a \cdot (b + c) = (a \cdot b) + (a \cdot c) \end{cases}$$

P3. Existen en B elementos identidad distintos, representados por 0 y 1, relativos a las operaciones (+) y (·) respectivamente:

$$\forall\, a, b \in B \qquad \begin{cases} (i)\;\; 0 + a = a + 0 = a \\ \\ (ii)\;\; 1 \cdot a = a \cdot 1 = a \end{cases}$$

P4. Para cada elemento a $\epsilon\ B$ existe un $\bar{a}\epsilon\ B$ tal que:

$$\forall\, a \in B \quad \exists\, \bar{a} \in B \qquad \begin{cases} (i)\;\; a + \bar{a} = 1 \\ \\ (ii)\;\; a \cdot \bar{a} = 0 \end{cases}$$

Puesto que $\bar{a}$ está unívocamente determinado por *a*, se llama a (¯) el operador negación o complemento.
En algunos sitios también se pude encontrar representado este operador con prima (‘): *a’*

Se asumen los siguientes convenios:

- El símbolo de la operación (·) se omite: $a \cdot b \rightarrow ab$
- Existe jerarquización: (·) precede a (+): $(a \cdot b) + c \rightarrow ab + c$

Este tipo de matemáticas resultó muy práctica para trabajar con los circuitos digitales o de computación. Para ello se introdujo un tipo particular de álgebra booleana, denominada **álgebra de conmutación**, definiendo:

- El conjunto B formado por dos únicos elementos: B = {0, 1}
- Las operaciones suma (+), producto (·) y complemento (¯) definidos por:

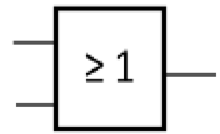
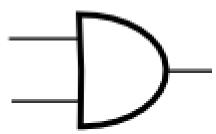
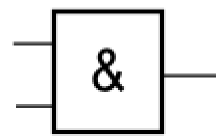
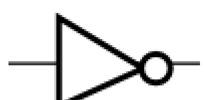
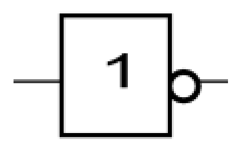
+	0	1
0	0	1
1	1	1

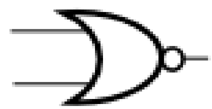
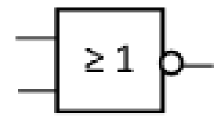
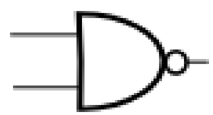
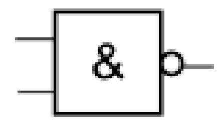
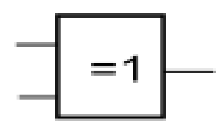
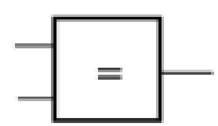
·	0	1
0	0	0
1	0	1

¯	
0	1
1	0

Este álgebra de conmutación es muy útil porque eléctricamente se pueden construir las operaciones (+), (·) y (¯) de forma muy eficiente con puertas lógicas de relativa fácil construcción.

3. PUERTAS LÓGICAS

<table>
<tr><th>Puerta Lógica</th><th>Simbología Americana</th><th>Simbología Europea</th><th>Función</th></tr>
<tr><td>OR
(sumador)</td><td></td><td>
</td><td><table><tr><th>+</th><th>0</th><th>1</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>AND
(multiplicador)</td><td></td><td>
</td><td><table><tr><th>·</th><th>0</th><th>1</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr></table></td></tr>
<tr><td>NOT
(inversor)</td><td></td><td>
</td><td><table><tr><th>—</th><th></th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
</table>

Puerta Lógica	Simbología Americana	Simbología Europea	Función
NOR (sumador negado)			$\begin{array}{c\|cc} \downarrow & 0 & 1 \\ \hline 0 & 1 & 0 \\ 1 & 0 & 0 \end{array}$
NAND (multiplicador negado)			$\begin{array}{c\|cc} \uparrow & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 1 & 0 \end{array}$
X-OR (exclusive OR)			$\begin{array}{c\|cc} \oplus & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array}$
X-NOR (exclusive NOR)			$\begin{array}{c\|cc} \odot & 0 & 1 \\ \hline 0 & 1 & 0 \\ 1 & 0 & 1 \end{array}$

3. TEOREMAS DEL ÁLGEBRA BOOLEANA

Principio de dualidad: Cada teorema o identidad algebraica deducible de los postulados de un álgebra booleana puede transformarse en un segundo teorema o identidad algebraica válida sin más que intercambiar las operaciones (+) y (·) junto con los elementos identidad 0 y 1.

Teorema 1. Para cada elemento *a* en un álgebra booleana se tiene:

$$\begin{cases} (i) \quad a + 1 = 1 \\ (ii) \quad a \cdot 0 = 0 \end{cases}$$

Teorema 2. Ley de **idempotencia**:

$$\begin{cases} (i) \quad a + a = a \\ (ii) \quad a \cdot a = a \end{cases}$$

Teorema 3. Ley de **involución**:

$$\overline{(\overline{a})} = a$$

Teorema 4. Ley de **absorción**:

$$\begin{cases} (i) \quad a + ab = a \\ (ii) \quad a \cdot (a + b) = a \end{cases}$$

Teorema 5. Ley de **DeMorgan**:

$$\begin{cases} (i) \quad \overline{(a + b)} = \overline{a} \cdot \overline{b} \\ (ii) \quad \overline{ab} = \overline{a} + \overline{b} \end{cases}$$

4. FUNCIONES Y FÓRMULAS DE CONMUTACIÓN

FUNCIÓN: una función de conmutación completa de n-variables:

$$f(x_1, x_2, \ldots, x_n)$$

se define como el conjunto de pares ordenados en los que el dominio de la función es B^n y el rango $B = \{0, 1\}$:

$$f: \quad B^n \rightarrow B$$

Se denominan también funciones completamente especificadas o simplemente **función de conmutación**.

Esta definición de función de conmutación de n-variables sugiere varias formas de representación como: tabla de verdad o mapa de Karnaugh.

Tabla de combinaciones o Tabla de verdad: es una tabla con n + 1 columnas, donde las n primeras conforman el conjunto de variables (entradas) y la última el valor de la función (salida) para cada una de las 2^n combinaciones posibles. Por ejemplo, una función de 3 variables sería así:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	f(0, 0, 0)
0	0	1	f(0, 0, 1)
0	1	0	f(0, 1, 0)
0	1	1	f(0, 1, 1)
1	0	0	f(1, 0, 0)
1	0	1	f(1, 0, 1)
1	1	0	f(1, 1, 0)
1	1	1	f(1, 1, 1)

Mapa de Karnaugh: es una representación gráfica de la función, que consta de 2^n casillas dispuestas de tal forma que todas las casillas son adyacentes, es decir, las combinaciones de las variables correspondiente a esas casillas solo varíen en 1 dígito. Se debe tener presente que las casillas de los extremos también son adyacentes.

x_1 \ x_2 x_3	**0 0**	**0 1**	**1 1**	**1 0**
0	f(0, 0, 0)	f(0, 0, 1)	f(0, 1, 1)	f(0, 1, 0)
1	f(1, 0, 0)	f(1, 0, 1)	f(1, 1, 1)	f(1, 1, 0)

FÓRMULA: una fórmula de conmutación de n-variables consiste en un número finito de constantes (0 y 1) y variables (x_1, x_2, …, x_n) conectadas mediante operaciones (+), (·) y (¯) de forma que las operaciones (+) y (·) nunca sean adyacentes.

De estas definiciones (de función y de fórmula) se llega a que cada fórmula de conmutación de n-variables describe a una función de conmutación de n-variables.

Ejemplos:

$$f(x_1, x_2, x_3) = x_1 + (\bar{x}_1 x_2 + x_3)$$

$$f(x_1, x_2) = x_2 + \bar{x}_1 \bar{x}_2$$

<u>FÓRMULAS NORMALES</u>:

Se define:

- Literal: una única variable, complementada o no.
- Término producto: literal o producto de literales.
- Término suma: literal o suma de literales.

Una fórmula de conmutación escrita como un solo término producto o como suma (disyunción) de términos productos se dice que está en **forma normal disyuntiva**, y se llama **fórmula normal disyuntiva**.

Una fórmula de conmutación escrita como un solo término suma o como producto (conjunción) de términos suma se dice que está en **forma normal conjuntiva**, y se llama **fórmula normal conjuntiva**.

REPRESENTACIÓN EN FORMA CANÓNICA:

FORMA CANÓNICA DISYUNTIVA:

Se define **mintérmino** a un término producto en el que cada variable aparece una vez y solo una vez, bien complementada o incomplementada.

Ejemplo con 3 variables: $\bar{x}_1 x_2 \bar{x}_3$

Una fórmula está en **forma canónica disyuntiva** o **de mintérminos** si está compuesta solo de mintérminos.

Ejemplo: $f(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3 + x_1 x_2 \bar{x}_3$

También se puede utilizar la **notación m**: para ello se asigna el 0 a la variable complementada y el 1 a la variable sin complementar, y codificamos cada mintérmino con su número decimal. El ejemplo anterior quedaría así:

$$f(x_1, x_2, x_3) = m_0 + m_2 + m_6 = \sum m(\,0, 2, 6)$$

<u>Primer Teorema de expansión</u>:

Permite obtener la forma canónica de mintérminos por uso repetido del siguiente teorema:

$$f(x_1, x_2, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) =$$

$$= x_i\ f(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) + \bar{x}_i \cdot f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$$

Teorema: cada función de conmutación completa puede escribirse como:

$$f(x_1, x_2, \dots, x_n) = \sum_{i=0}^{2^n-1} f_i \cdot m_i$$

Donde:

$$f(0) = f(0,0, \dots, 0) \qquad m_0 = \bar{x}_1 \bar{x}_2 \cdots \bar{x}_n$$

$$f(1) = f(0,0, \dots, 1) \qquad m_1 = \bar{x}_1 \bar{x}_2 \cdots x_n$$

… …

FORMA CANÓNICA CONJUNTIVA:

Se define **maxtérmino** a un término suma en el que cada variable aparece una vez y solo una vez, bien complementada o incomplementada.

Ejemplo con 3 variables: $\bar{x}_1 + x_2 + \bar{x}_3$

Una fórmula está en **forma canónica conjuntiva** o **de maxtérminos** si está compuesta solo de maxtérminos.

Ejemplo:

$$f(x_1, x_2, x_3) = (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) \cdot (\bar{x}_1 + x_2 + \bar{x}_3) \cdot (x_1 + x_2 + \bar{x}_3)$$

También se puede utilizar la **notación M**: para ello se asigna el 0 a la variable sin complementar y el 1 a la variable complementada, y codificamos cada maxtérmino con su número decimal. El ejemplo anterior quedaría así:

$$f(x_1, x_2, x_3) = M_7 + M_5 + M_1 = \prod M(1, 5, 7)$$

Segundo Teorema de expansión:

Permite obtener la forma canónica de maxtérminos por uso repetido del siguiente teorema:

$$f(x_1, x_2, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) =$$

$$= [x_i + f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)] \cdot [\bar{x}_i + f(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)]$$

Teorema: cada función de conmutación completa puede escribirse como:

$$f(x_1, x_2, \dots, x_n) = \prod_{i=0}^{2^n-1} (f_i + M_i)$$

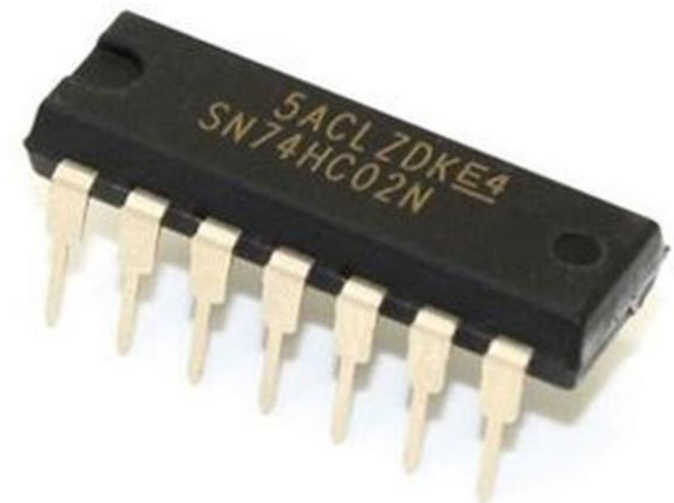
Donde:

$$f(0) = f(0,0, \dots, 0) \qquad M_0 = x_n + x_n + \cdots + x_n$$

$$f(1) = f(0,0, \dots, 1) \qquad M_1 = x_n + x_n + \cdots + \bar{x}_n$$

… …

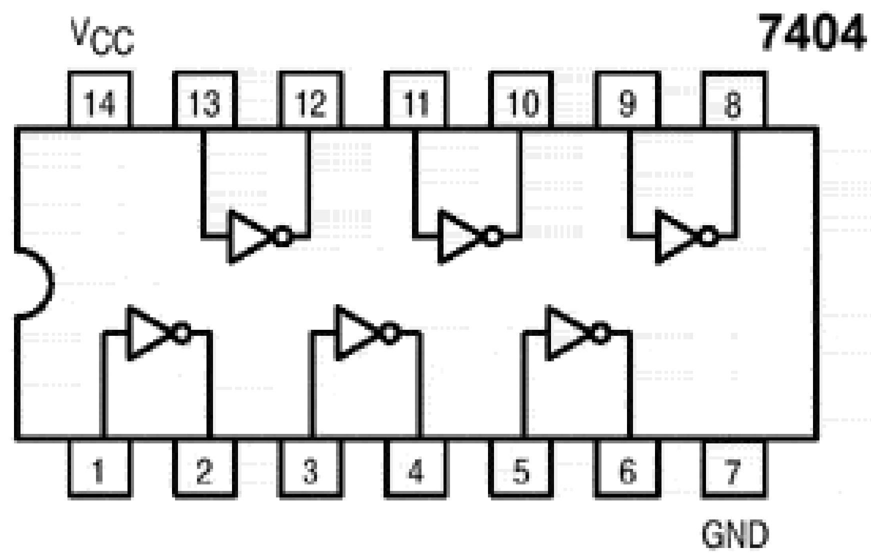
5. CIRCUITOS INTEGRADOS COMERCIALES: 74XX



Las puertas lógicas se pueden adquirir comercialmente en forma **integrado** o **chip**: una “cucaracha con 8 patitas”. También es cierto que cada vez su uso va reduciéndose, bien porque el precio de fabricar un circuito digital personalizado (que realiza la totalidad de la función que hemos diseñado) se ha abaratado notablemente, o bien porque cada vez se utilizan más los circuitos programables (lógica programada, en contraposición de la lógica cableada que supone el uso de utilizar puertas lógicas).

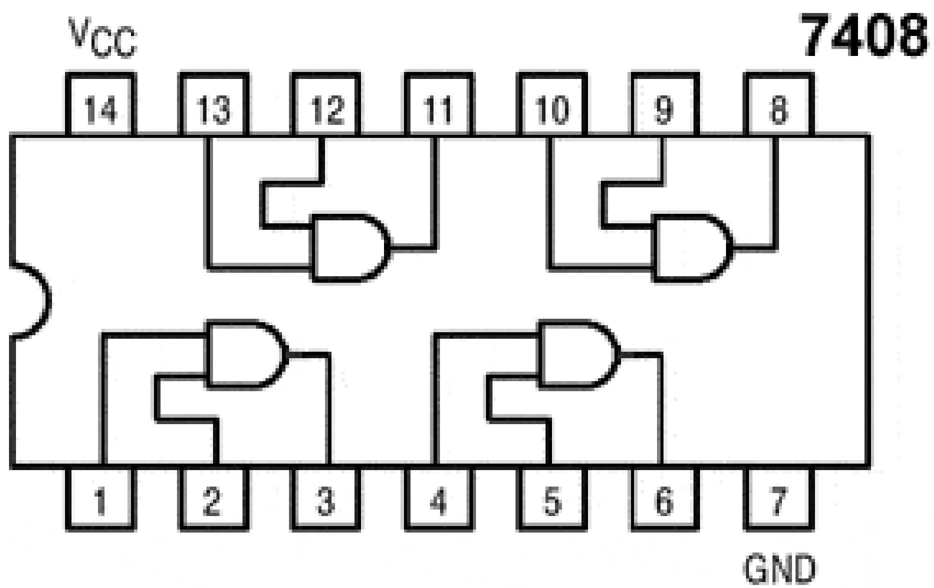
7404: Puertas NOT

Contiene 6 puertas NOT. Las ocho patas corresponden al siguiente datasheet:

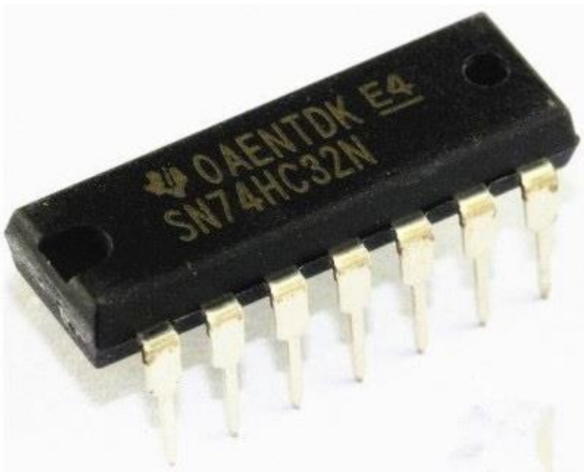


7408: Puertas AND

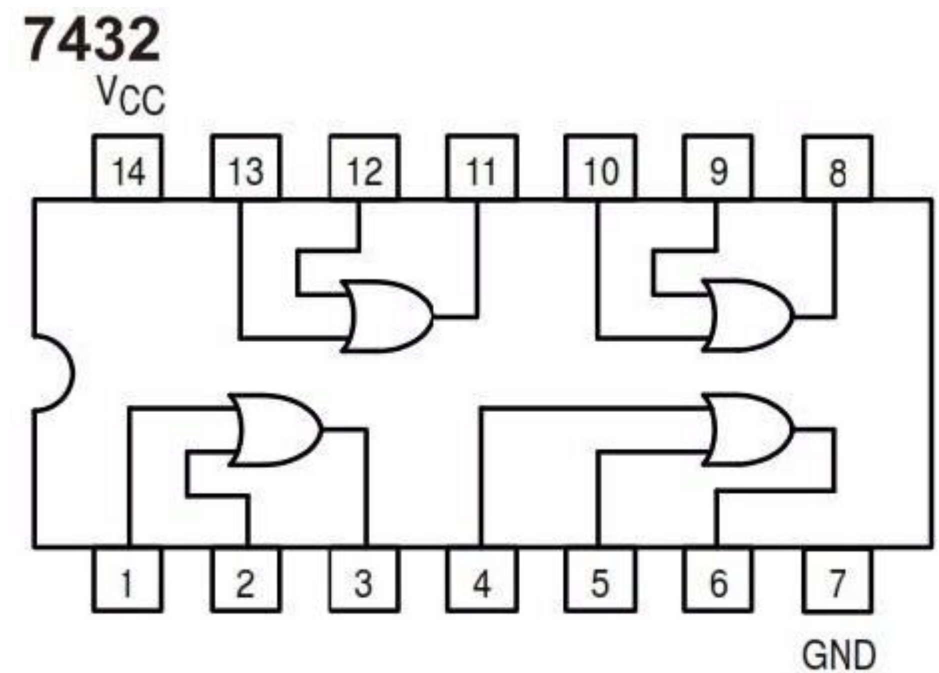
Contiene 4 puertas AND. Las ocho patas corresponden al siguiente datasheet:



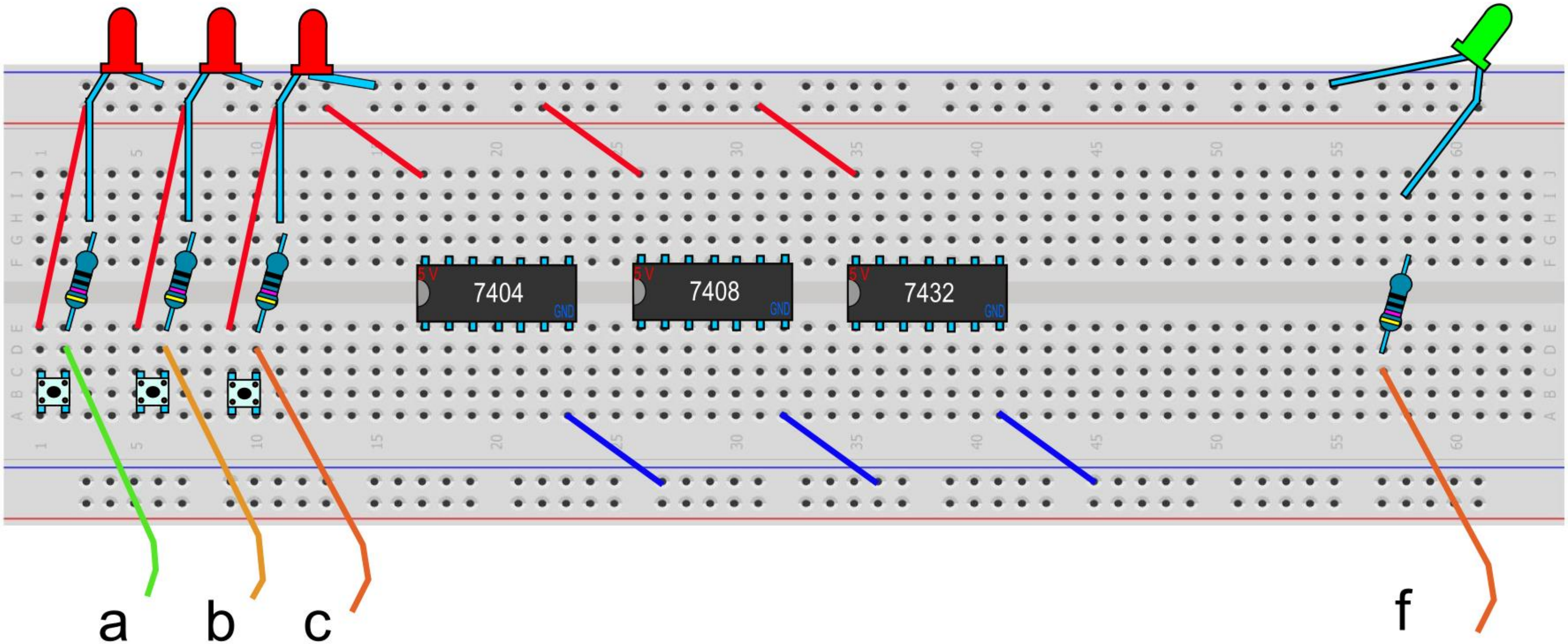
7432: Puertas OR



Contiene 4 puertas OR. Las ocho patas corresponden al siguiente datasheet:



Con la ayuda de una placa de prototipos (protoboard) se pueden construir circuitos digitales de manera muy fácil. Solo debemos conectar las distintas puertas lógicas de manera correcta:



6. DISEÑO DE CIRCUITOS DIGITALES

Un circuito digital no es más que la implementación física de una función lógica. A la hora de diseñar dicho circuito se debe perseguir un coste mínimo de puertas lógicas, lo cual nos lleva a la necesidad de **técnicas de simplificación**.

Se adoptará como criterio de más peso el obtener una expresión en forma de suma de productos o de producto de sumas que contenga el menor número de términos y de literales (pues es la que permitirá un ahorro en componentes y su posterior consumo eléctrico), dando lugar a la **suma minimal** o al **producto minimal**, respectivamente.

MÉTODO DE MAPA DE KARNAUGH:

Este método parte del mapa de Karnaugh y del hecho de la suma de dos términos adyacentes tiene la propiedad:

$$ab + \bar{a}b = (a + \bar{a}) \cdot b = 1 \cdot b = b$$

Así, sobre un mapa de Karnaugh, dos términos productos adyacentes pueden combinarse, puesto que solo difieren en un bit.

De igual forma, también podrían combinarse en gurpos de 4, de 8, etc. La regla general es que si en 2^i términos productos, todos excepto i variables son idénticas, entonces los 2^i términos pueden combinarse y las i variables que cambian eliminarse.

A estos grupos obtenidos de estas combinaciones, así como a los términos productos resultantes de la simplificación se les denominan **implicantes prima**.

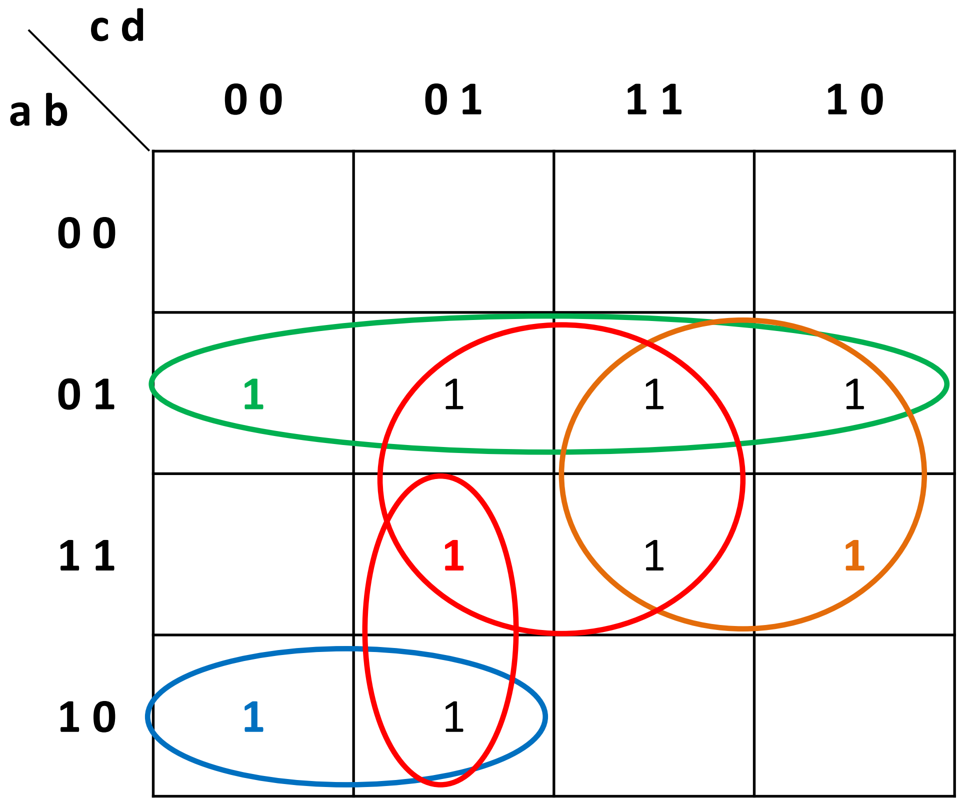
El proceso a seguir es:

- Representar sobre el mapa de Karnaugh la función: se asigna un 1 a aquellas casillas cuya combinación hace 1 a la función, y un 0 al resto (los 0 no se suelen poner).
- Se van obteniendo los implicantes primas de la forma:
  - Se encirculan todos los 1 que no se pueden agrupar con ningún otro.
  - Se encirculan los grupos de dos 1 que no pueden formar grupos de cuatro.
  - Se encirculan los grupos de cuatro 1 que no pueden formar grupos de ocho.
- Las casillas incluidas en solo un implicante prima se denominan **celdas distinguidas**, y las implicantes prima que las contienen, **esenciales**; y deben estar incluidas en la correspondiente suma minimal.
- Seguidamente nos fijamos en los 1 no cubiertos por los implicantes prima esenciales, y se escoge la implicante prima que lo contenga con menor número de literales.

ab \ cd	0 0	0 1	1 1	1 0
0 0				
0 1	1	1	1	1
1 1		1	1	1
1 0	1	1		

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

Implicantes prima esenciales



c d
a b
0 0
0 1
1 1
1 0
0 0
1
1
1
1
1 1
1
1
1
1 0
1
1

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

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

También se puede simplificar la función mediante el **agrupamiento de 0** con el mismo procedimiento, llegando así a un producto minimal. Se debe cuidar el hecho que en este caso el 0 es asignado a la variable sin complementar y el 1 a la variable complementada.

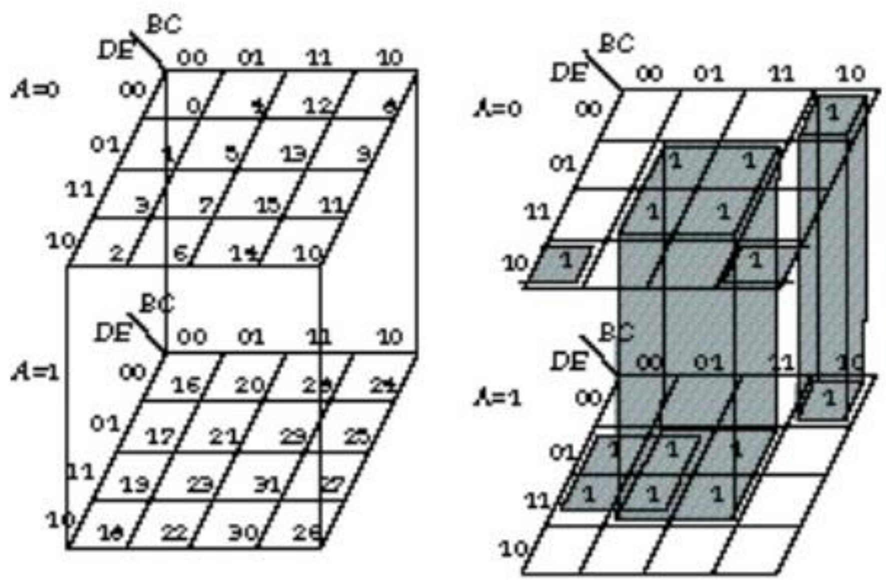
a b \ c d	0 0	0 1	1 1	1 0
0 0				
0 1	0	0		
1 1	0	0	0	
1 0		0	0	

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

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

Para la correcta simplificación de la función se deben hacer ambos casos y comprobar con cuál se obtiene el menor número de términos y de literales.

Este método es útil hasta con 5 variables, empleando dos mapas de 4x4 y teniendo presente que ambos son adyacentes (es como si estuvieran superpuestos).



7. CONJUNTOS COMPLETOS

Hemos visto que cualquier función de conmutación puede implementarse mediante una red de puertas NOT, AND y OR. Cabe preguntarse si son necesarias las tres.

Está claro que se necesitan inversores para realizar funciones arbitrarias; pero si se consideran solo puertas AND e inversores vemos que sobran las puertas OR haciendo uso del teorema de DeMorgan:

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

Análogamente podemos omitir las puertas AND usando únicamente puertas OR e inversores:

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

Un **conjunto de operaciones** se dice **funcionalmente completo** si y solo si cualquier función de conmutación puede expresarse mediante operaciones de dicho conjunto.

Algunos conjunto completos son:

<u>Puertas NAND</u>: $a \uparrow b = \overline{a \cdot b}$

Como inversor: $a \uparrow a = \bar{a}$

Como multiplicador: $(a \uparrow b) \uparrow 1 = ab$

Como sumador: $\bar{a} \uparrow \bar{b} = a + b$

El proceso que se debe seguir para implementar cualquier función con puertas NAND es:

- Aplicar a la expresión una doble inversión.
- Si la función es un producto, las dos negaciones debe dejarse tal cual. Si es una suma, se elimina una de ellas mediante la aplicación del teorema de DeMorgan.
- Se continúa complementando doblemente los términos hasta que todos las sumas y productos se conviertan en productos negados.

$$ab + cd = \overline{\overline{ab + cd}} = \overline{\left(\overline{ab}\right) \cdot (\overline{cd})}$$

$$(a+b)\cdot(c+d) = \overline{\overline{(a+b)\cdot(c+d)}} = \overline{\overline{(\overline{\overline{a+b}})\cdot(\overline{\overline{c+d}})}} =$$

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

<u>Puertas NOR</u>: $a \downarrow b = \overline{a + b}$

Como inversor: $a \downarrow a = \bar{a}$

Como multiplicador: $(a \downarrow b) = ab$

Como sumador: $(a \downarrow b) \downarrow 0 = a + b$

El proceso que se debe seguir para implementar cualquier función con puertas NOR es:

- Aplicar a la expresión una doble inversión.
- Si la función es un suma, las dos negaciones debe dejarse tal cual. Si son productos, se elimina uno de ellos mediante la aplicación del teorema de DeMorgan.
- Se continúa complementando doblemente los términos hasta que todos las sumas y productos se conviertan en sumas negadas.

$$(a+b)\cdot(c+d) = \overline{\overline{(a+b)\cdot(c+d)}} = \overline{\big(\overline{a+b}\big) + (\overline{c+d})}$$

$$ab+cd = \overline{\overline{ab+cd}} = \overline{\overline{(\overline{\overline{ab}}) + (\overline{\overline{cd}})}} = \overline{\overline{\big(\overline{\bar{a}+\bar{b}}\big) + (\overline{\bar{c}+\bar{d}})}}$$