Tema 4
Circuitos combinacionales

Funciones Combinacionales

Definición:

•Una función combinacional de *n* variables (también llamada función lógica o función de conmutación) es un operador de conmutación de aridad *n*, es decir, una aplicación *f*: $\mathbb{B}^n \rightarrow \mathbb{B}$ siendo $\mathbb{B}$ el conjunto de valores lógicos ($\mathbb{B}=\{0,1\}$).

$f(x_{n-1}, x_{n-2}, \ldots, x_2, x_1, x_0)$, las x_i son **variables binarias**.

•La función de conmutación es **completamente especificada** cuando su dominio es todo $\mathbb{B}^n$. En otro caso, la función es **incompletamente especificada**.

•Los elementos de $\mathbb{B}^n$ que no pertenecen al dominio se denominan **inespecificaciones**.

Funciones Combinacionales

Sea una función de conmutación f: $\mathbb{B}^n \rightarrow \mathbb{B}$ cualquiera, dado que $\mathbb{B}^n$ es un conjunto finito f puede definirse indicando la imagen de cada elemento de su dominio. Para ello se escribe una tabla, denominada **tabla de verdad**, de 2^n filas y dos columnas. En la casilla izquierda de cada fila aparece un elemento distinto de $\mathbb{B}^n$. Si dicho elemento pertenece al dominio de f, en la parte derecha aparece su imagen. En caso contrario aparece un guión o una equis para indicar una indeterminación. A continuación se muestra un ejemplo para n=3:

$X_2\ X_1\ X_o$	$f(X_2,X_1,X_0)$
0 0 0	1
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1

$X_2\ X_1\ X_o$	$f(X_2,X_1,X_0)$
0 0 0	0
0 0 1	0
0 1 0	--
0 1 1	1
1 0 0	1
1 0 1	--
1 1 0	1
1 1 1	--

Funciones Combinacionales

Definición:

Se dice que un conjunto de funciones de conmutación es un **universal** si y sólo si cualquier otra función de conmutación puede expresarse por composición de las mismas. Algunos conjuntos de operadores universales son los siguientes:

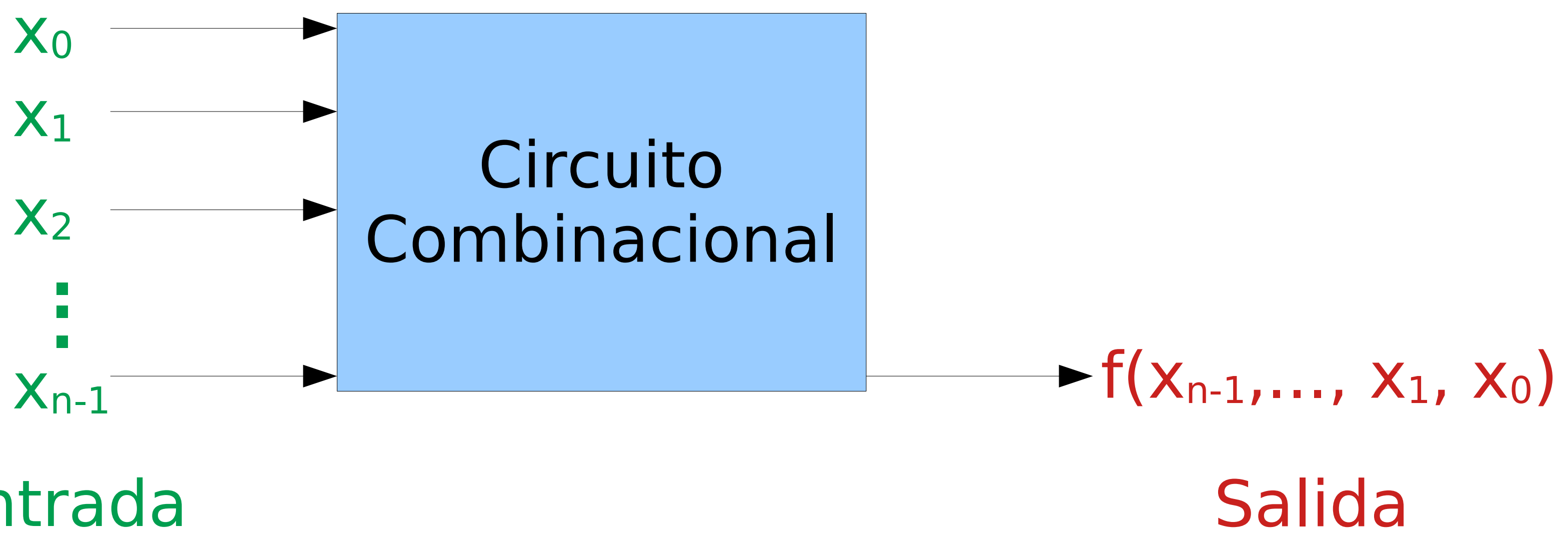
- {AND,OR,NOT}
- {NAND}
- {AND,NOT}
- {NOR}
- {OR,NOT}

Funciones Combinacionales

Definición:

Un circuito combinacional es aquel que:

- Implementa una función de conmutación.
- en cualquier momento en que su entrada sea estable, su salida sólo depende el valor de su entrada en ese momento.



Funciones Combinacionales

Representación

Existen varias formas de representar una función de conmutación:

- Expresión de álgebra de conmutación
- Tabla de verdad
- Mapa de Karnaugh (K-mapa)
- Circuito que implemente la función
- Lenguajes de descripción de hardware (LDH) que represente dicho circuito

Funciones Combinacionales

Ejemplo: función de tres variables

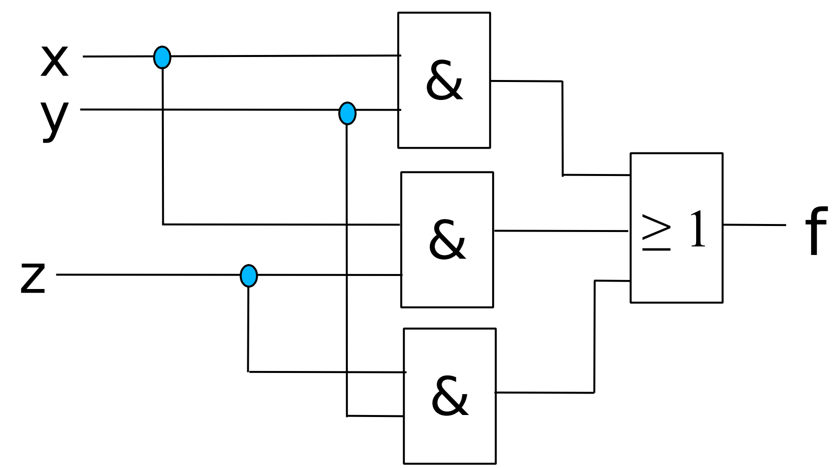
Expresión

$f(x,y,z)=xy+xz+yz$

Tabla de verdad

xyz	f(x,y,z)
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

Circuito



Código Verilog

```
module ejemplo(
 output f,
 input x,
 input y,
 input z)

 assign f = x&y | x&z | y&z;
endmodule
```

K-Mapa

z \ x y	00	01	11	10
0	0	0	1	0
1	0	1	1	1

f

Funciones Combinacionales

Ejemplo: función de 4 variables

Expresión: $g(x,y,z,u) = x\,y\,u + x\,z\,\overline{u} + y\,z$

K-Mapa

zu \ xy	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	0
10	0	1	1	1

g

Código Verilog

```
module g(
 input  x,
 input  y,
 input  z,
 input  u,
 output g);

 assign g = (x&y&u) | (x&z&~u)|(y&z);

endmodule
```

Funciones Combinacionales

Formas normalizadas

Un *literal* de una función es una expresión algebraica que incluye únicamente el nombre de una variable o su complemento.

Un *[término] producto* es una expresión algebraica en la que sólo aparecen literales y el símbolo del operador AND (posiblemente omitido). Análogamente Un *[término] suma* es una expresión algebraica en la que sólo aparecen literales y, posiblemente, el símbolo del operador OR.

Una *forma normalizada* de una función es una definición de la misma como de suma (OR) de productos o como producto (AND) de sumas.

$f(x,y,z) = xy+\overline{y}+yz$	Suma de productos
$g(x,y,z) = x(y+z)$	Producto de sumas
$h(a,b,c,d,e) = a+\overline{(b+c)}+d$	No normalizada

Funciones Combinacionales

Forma canónica disyuntiva

- Un *mintérmino* (o término mínimo) es una función de conmutación completamente especificada que vale 1 únicamente para un elemento del dominio.
- Existen 2^n mintérminos de $\mathbb{B}^n$ que se numeran de forma ascendente empezando por cero en la forma m_0, m_1,..,m_{n-1}. El número asociado a un mintérmino es el que representa en base 2 el elemento de su dominio cuya imagen es 1. Por ejemplo, los mintérminos de $\mathbb{B}^2$ son los siguientes:

x y	m_0(x,y)
0 0	1
0 1	0
1 0	0
1 1	0

x y	m_1(x,y)
0 0	0
0 1	1
1 0	0
1 1	0

x y	m_2(x,y)
0 0	0
0 1	0
1 0	1
1 1	0

x y	m_3(x,y)
0 0	0
0 1	0
1 0	0
1 1	1

Funciones Combinacionales

Forma canónica disyuntiva

- Cualquier mintérmino puede definirse usando un término producto en el que aparezcan ordenadas todas las variables de la función una sóla vez. A dicho término producto también se le llama mintérmino.
- Para obtener el término producto que expresa un mintérmino m_x se hace lo siguiente:
  - Se toma el elemento del dominio v tal que $m_x(v)=1$.
  - Si un bit de v vale 0 entonces el nombre la variable correspondiente aparecerá complementada en el término producto. En caso contrario aparecerá sin complementar
- Ejemplo para $\mathbb{B}^3$: $m_2(0,1,0)=1 \rightarrow m_2(x,y,z)=\overline{x}\,y\,\overline{z}$

Forma canónica disyuntiva

Una forma canónica disyuntiva de una función es una definición de la misma en forma de suma de productos en la que todos ellos son mintérminos y aparecen ordenados una sóla vez.

Teorema: Toda función de conmutación puede expresarse en forma canónica disyuntiva.

Corolario: {AND,OR,NOT} es universal.

Teorema: La forma canónica de disyuntiva de una función de conmutación completamente especificada es única.

Ejemplo: $f(x_1,x_2,x_3) = \bar{x}_1\bar{x}_2\bar{x}_3+\bar{x}_1x_2\bar{x}_3+\bar{x}_1x_2x_3+x_1x_2x_3 =$

$= m_0 + m_2 + m_3 + m_7 = \sum(0,2,3,7)$

Funciones Combinacionales

Forma canónica conjuntiva

- Un ***maxtérmino*** (o término máximo) es una función de conmutación completamente especificada que vale 0 únicamente para un elemento del dominio.
- Existen 2^n maxtérminos de $\mathbb{B}^n$ que se numeran de forma ascendente empezando por cero en la forma M_0, M_1,..,$M_{n\text{-}1}$. El número asociado a un maxtérmino es el que representa en base 2 el elemento de su dominio cuya imagen es 0. Por ejemplo, los maxtérminos de $\mathbb{B}^2$ son los siguientes:

x y	M_0(x,y)
0 0	0
0 1	1
1 0	1
1 1	1

x y	M_1(x,y)
0 0	1
0 1	0
1 0	1
1 1	1

x y	M_2(x,y)
0 0	1
0 1	1
1 0	0
1 1	1

x y	M_3(x,y)
0 0	1
0 1	1
1 0	1
1 1	0

Funciones Combinacionales

Forma canónica conjuntiva

- Cualquier maxtérmino puede definirse usando un término suma en el que aparezcan ordenadas todas las variables de la función una sóla vez. A dicho término suma también se le llama maxtérmino.
- Para obtener el término suma que expresa un maxtérmino M_x se hace lo siguiente:
  - Se toma el elemento del dominio v tal que $M_x(v)=0$.
  - Si un bit de v vale 1 entonces la variable correspondiente aparecerá complementada en el término suma. En caso contrario aparecerá sin complementar
- Ejemplo para $\mathbb{B}^3$: $M_2(0,1,0)=0 \;\rightarrow\; M_2(x,y,z)=x+\overline{y}+z$

Funciones Combinacionales

Forma canónica conjuntiva

Una forma canónica conjuntiva de una función es una definición de la misma en forma de producto de sumas en la que todas ellas son maxtérminos y aparecen ordenados una sóla vez.

Teorema: Toda función de conmutación puede expresarse en forma canónica conjuntiva.

Corolario: {AND,OR,NOT} es universal.

Teorema: La forma canónica de conjuntiva de una función de conmutación completamente especificada es única.

Ejemplo: $f(x_1,x_2,x_3)=(x_1+x_2+x_3)(\overline{x}_1+x_2+x_3)(\overline{x}_1+x_2+\overline{x}_3)(\overline{x}_1+\overline{x}_2+\overline{x}_3) = M_0M_4M_5M_7 = \prod(0,4,5,7)$

Análisis de Circuitos Combinacionales

Análisis Lógico

Dado un circuito combinacional, analizarlo consiste en encontrar:

- una expresión algebraica de la función que implementa
- su tabla de verdad y/o el K-mapa
- una explicación verbal de su función

Análisis de Circuitos Combinacionales

Procedimiento

MIENTRAS no se conozca la expresión de la función de salida

- obtén las expresiones lógicas correspondientes a las salidas de las puertas para las que se conocen las expresiones asociadas a todas sus entradas

FIN MIENTRAS

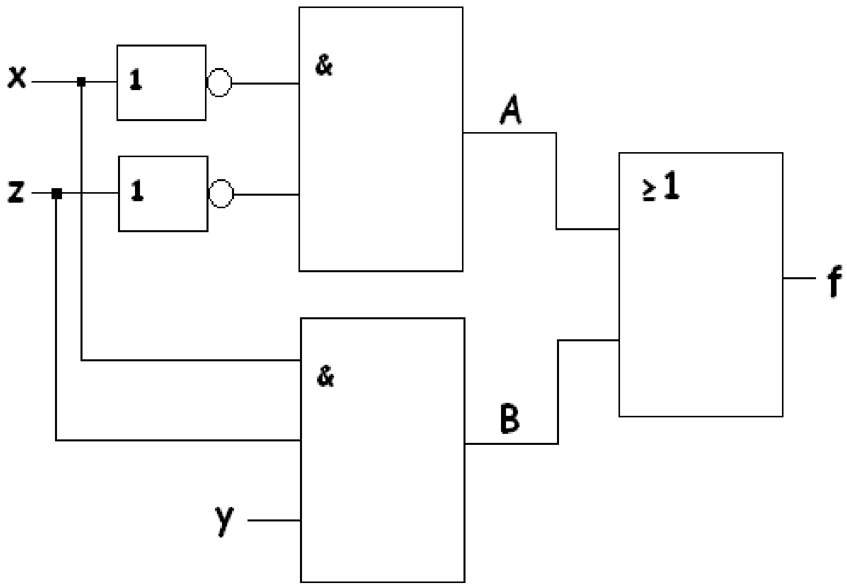
simplifica la expresión obtenida y/o traduce a un mapa o tabla

Análisis de Circuitos Combinacionales

Análisis Lógico

Ejemplo

Circuito:



Expresión:

$B = x\,y\,z$

$A = \bar{x}\,\bar{z}$

$f(x,y,z) = A + B = x\,y\,z + \bar{x}\,\bar{z}$

Tabla:

xyz	f(x,y,z)
000	1
001	0
010	1
011	0
100	0
101	0
110	0
111	1

$$f(x,y,z) = 1 \text{ si y sólo si } \begin{cases} xyz=1, \ (x=y=z=1, (x,y,z)=(1,1,1)) \\ \text{ó} \\ \bar{x}\,\bar{z}=1 \ (x=z=0, \quad (x,y,z)=(0,-,0)) \end{cases}$$

Análisis Temporal

Consiste en obtener un representación de la evolución en el tiempo de las señales de un circuito. A esta representación temporal se la denomina **CRONOGRAMA**.

Dicha representación puede ser:

- Suponiendo que las puertas no tienen retraso.
- Teniendo en cuenta los retrasos propios de las puertas lógicas.

Análisis de Circuitos Combinacionales

Análisis Temporal

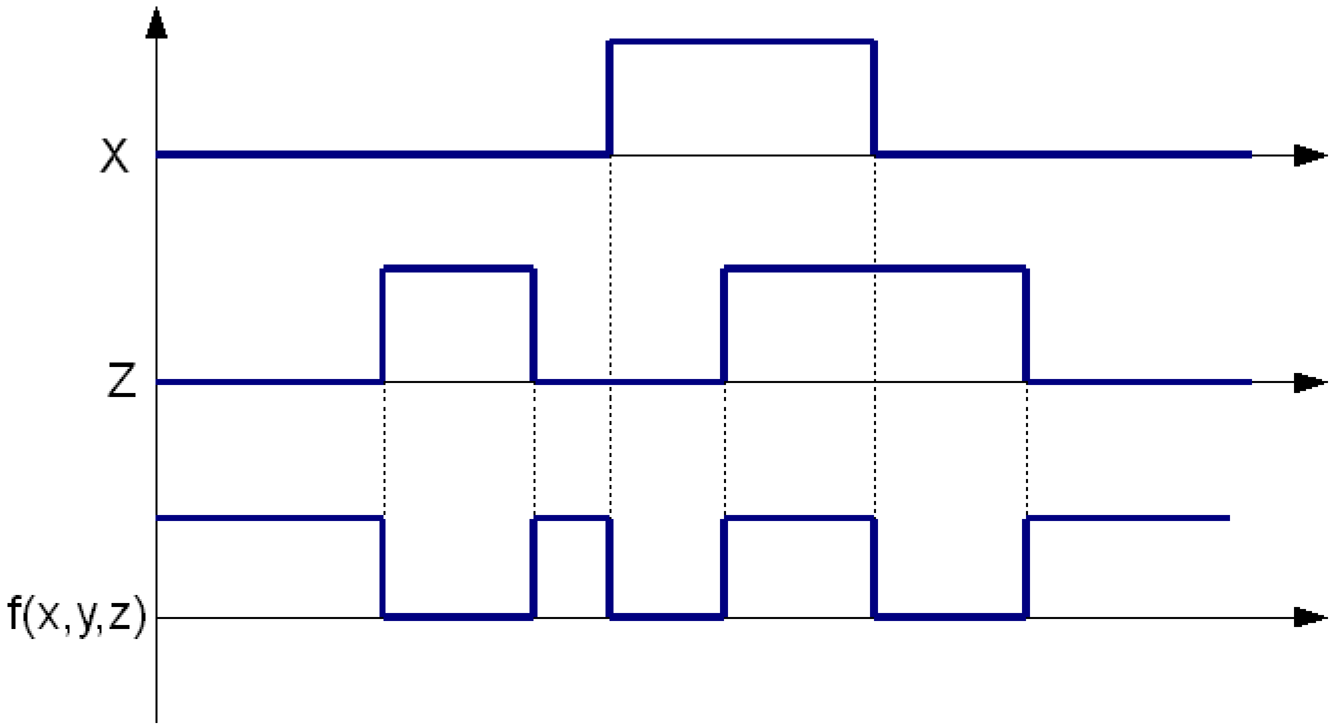
Tabla:

xyz	f(x,y,z)
000	1
001	0
010	1
011	0
100	0
101	0
110	0
111	1

Ejemplo

Cronograma (con y=1)

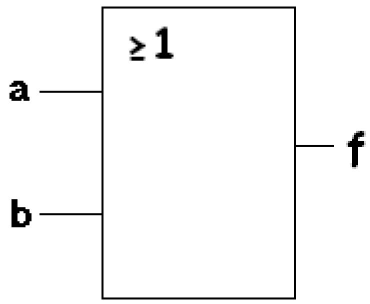
(sin considerar retrasos):

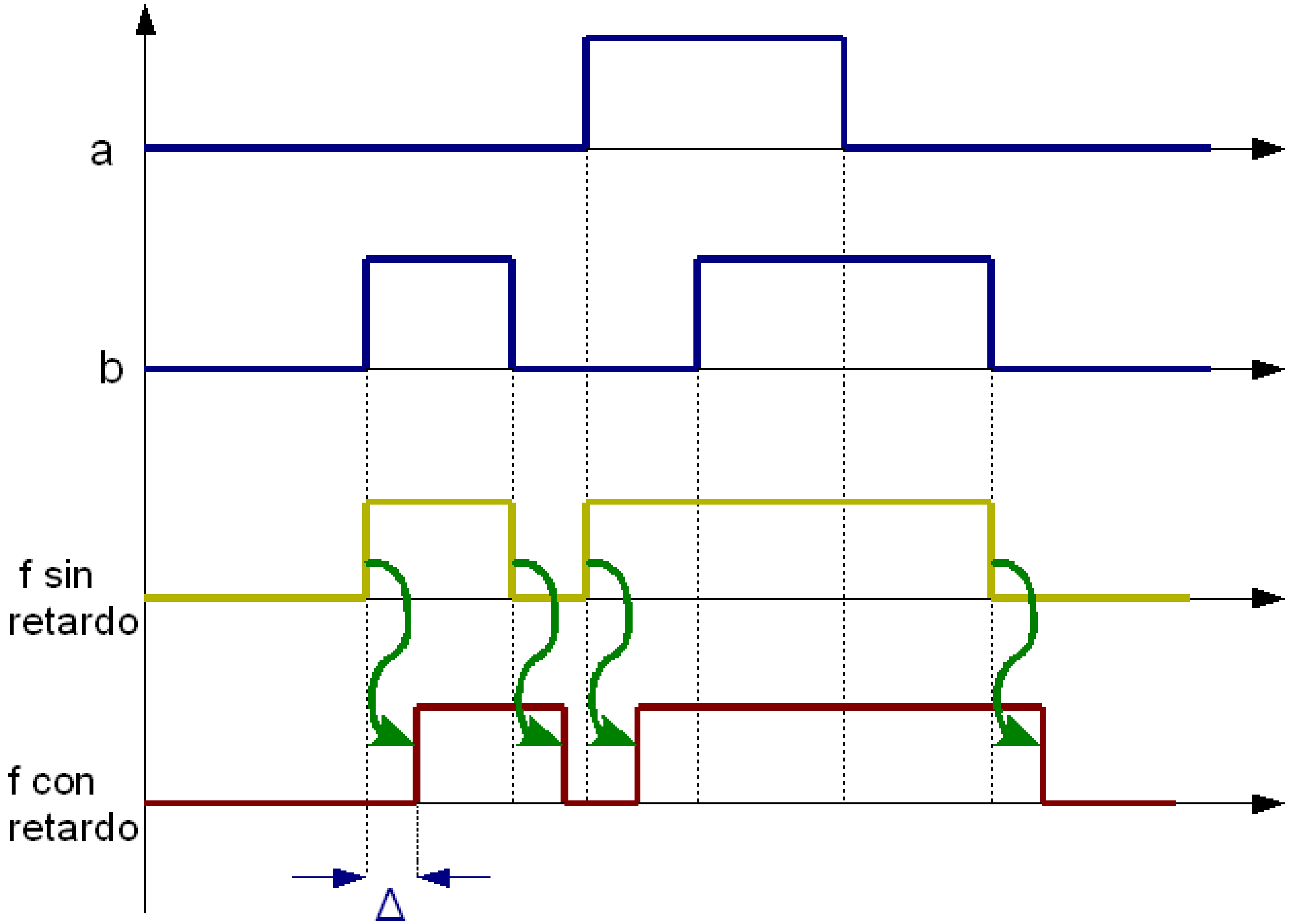


Análisis de Circuitos Combinacionales

Análisis Temporal

Para dibujar el cronograma considerando los retrasos, es necesario desplazar la salida de las puertas tanto como indique el valor del retraso.



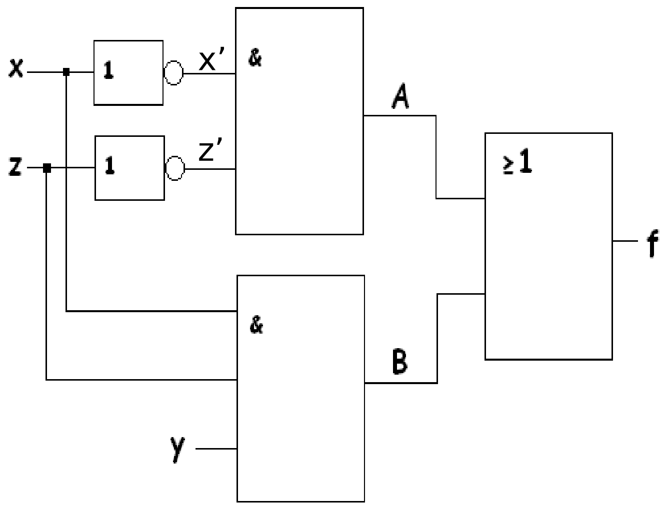


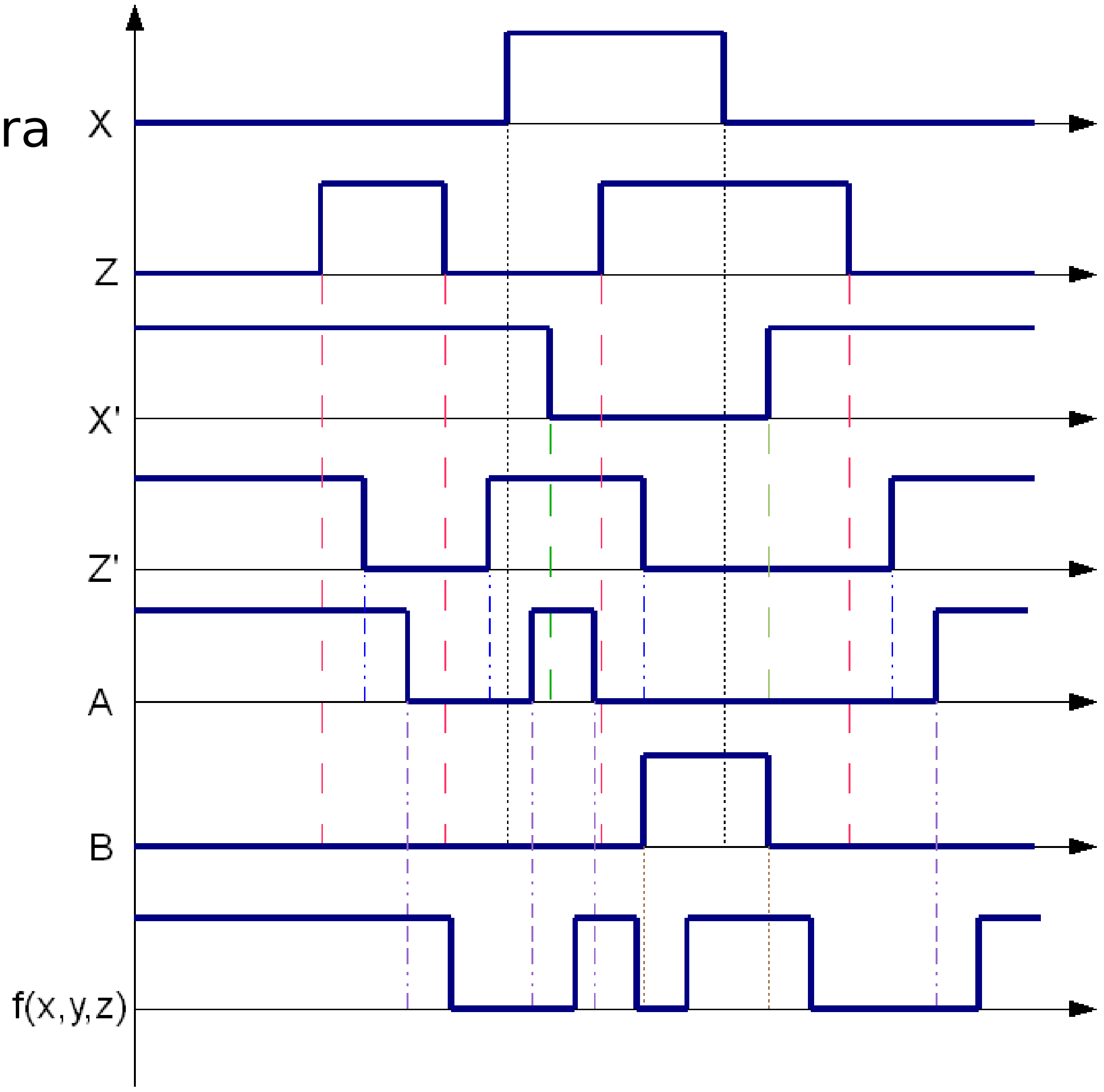
Análisis de Circuitos Combinacionales

Análisis Temporal

Cronograma (con y=1)

(con retrasos igual para todas las puertas)





Análisis Temporal

Azares: Teniendo en cuenta los retrasos de la puertas podemos encontrarnos con la aparición de pequeños pulsos transitorios que hacen que la salida difiera de la esperada, es decir, de la obtenida de forma teórica sin considerar los retrasos.

Ejemplo: Circuito con azar (Δ=1ns)

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

b=c=1,

a → señal periódica (T=10ns)

Diseño de Circuitos Combinacionales

Diseño de Circuitos Combinacionales

Objetivos y conceptos básicos

El diseño (o síntesis) de un circuito combinacional es el proceso inverso al análisis: partiendo de una descripción de una función debemos proporcionar un circuito que la realice.

Diseño de Circuitos Combinacionales

Objetivos y conceptos básicos

El circuito debe ser óptimo en algún sentido. Puede pedirse:

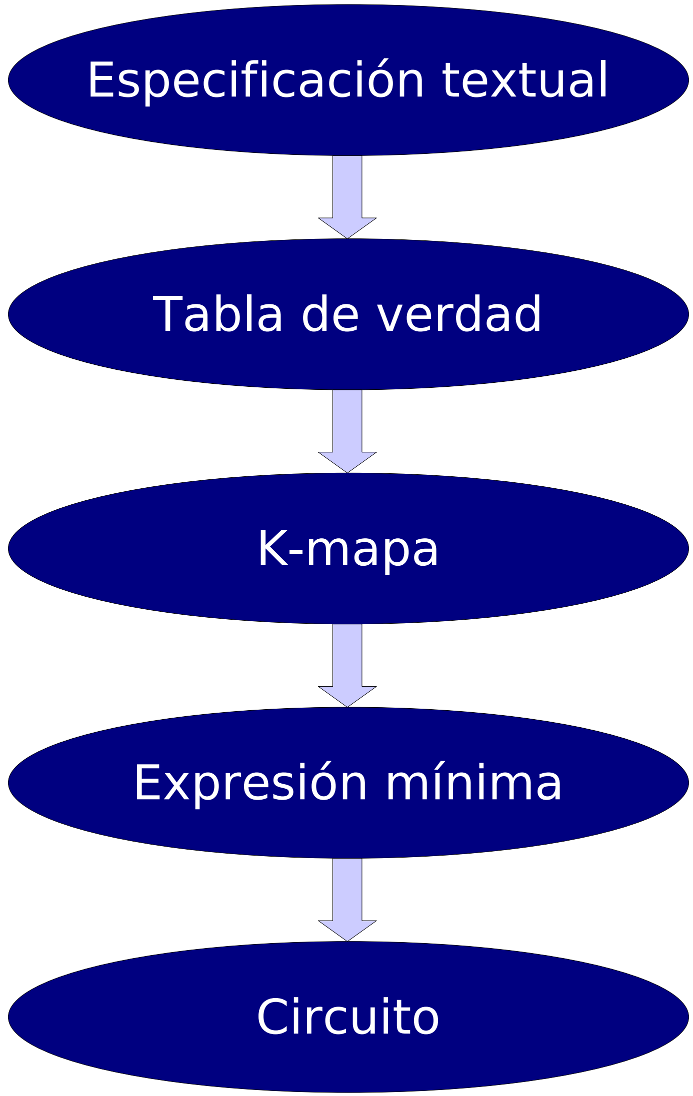
- Optimizar velocidad, es decir, minimizar el retraso
- Optimizar consumo, es decir, minimizar la energía que usa
- Optimizar coste de fabricación, es decir, minimizar número de puertas y conexiones (entradas de puertas)

Veremos un método para optimizar el coste de fabricación. Sólo podrá aplicarse cuando el número de entradas sea pequeño. Tiene las siguientes características:

- Proporciona un circuito en dos niveles de puertas (tres para **simple raíl**). Esto acota el retraso total del circuito.
- Usa puertas AND, OR, NAND y/o NOR
- No considera fan-in ni fan-out como restricciones.

Diseño de Circuitos Combinacionales

Pasos del proceso



Pasos del proceso

Paso 1: Descripción textual -> Tabla de verdad

- Determinar variables de entrada y especificar el significado de sus valores (0 y 1).
- Igual para las variables de salida.
- Obtener la tabla de verdad.

Paso 2: Obtener el K-mapa

- A partir de la tabla de verdad anterior o de la especificación establecida, se obtiene el K-mapa de la función a implementar.

Pasos del proceso

Ejemplo1 (pasos 1 y 2):

Suponga que un circuito debe tener una entrada de cuatro bits $X_3X_2X_1X_0$ y una salida Z. La salida debe valer uno si y solo si el número representado por la entrada en base 2 es primo.

X3X2X1X0	Z
0 0 0 0	0
0 0 0 1	0
0 0 1 0	1
0 0 1 1	1
0 1 0 0	0
0 1 0 1	1
0 1 1 0	0
0 1 1 1	1

X3X2X1X0	Z
1 0 0 0	0
1 0 0 1	0
1 0 1 0	0
1 0 1 1	1
1 1 0 0	0
1 1 0 1	1
1 1 1 0	0
1 1 1 1	0

Diseño de Circuitos Combinacionales

Pasos del proceso

Ejemplo2 (pasos 1 y 2):

Se desea diseñar un circuito combinacional que recibe información del estado de tres bombillas (encendida o apagada) y del estado de un único interruptor (on - off). El circuito debe generar una alarma que se active cuando alguna de las bombillas no esté encendida cuando el interruptor está on, o cuando alguna bombilla esté encendida y el interruptor esté off.

$i\ b_1 b_2 b_3$	a	$i\ b_1 b_2 b_3$	a
0000	0	1000	1
0001	1	1001	1
0010	1	1010	1
0011	1	1011	1
0100	1	1100	1
0101	1	1101	1
0110	1	1110	1
0111	1	1111	0

<u>Entradas</u>: i (interruptor), $b_{1\text{-}3}$ (bombillas),

<u>Salida</u>: a (alarma)

$$b_i = \begin{cases} 0 \ \text{apagada} \\ 1 \ \text{encendida} \end{cases} \qquad i = \begin{cases} 0 \ \text{off} \\ 1 \ \text{on} \end{cases} \qquad a = \begin{cases} 0 \ \text{inactiva} \\ 1 \ \text{activa} \end{cases}$$

Diseño de Circuitos Combinacionales

Minimizado mediante K-mapa

Paso 3: Obtener la expresión mínima en dos niveles

- Usaremos el método del K-mapa
- Permite obtener expresiones mínimas en forma de suma de productos, es decir:
  - Aquellas con el mínimo número posible de términos producto
  - De entre las anteriores, aquellas que presenten un menor número de literales
- Permite obtener expresiones mínimas en forma de producto de sumas, es decir:
  - Aquellas con el mínimo número posible de términos suma
  - De entre las anteriores, aquellas que presenten un menor número de literales
- La explicación del procedimiento para obtener expresiones mínimas requiere definiciones previas.

Diseño de Circuitos Combinacionales

Minimizado mediante K-mapa

Implicante (implicada) de una función de conmutación f

Es una función de conmutación del mismo número de variables definida por un término producto (suma) que vale 0 (1) para todo elemento del dominio de f en el que f vale 0 (1). En el K-mapa lo asociaremos con un grupo de casillas a 1 (0).

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

f(A,B,C,D)

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

$g(A,B,C,D)=\overline{A}\cdot\overline{C}\cdot D$ es implicante de f

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

$h(A,B,C,D)=\overline{B}+C$ es implicada de f

Minimizado mediante K-mapa

Implicante (implicada) de una función f

El <u>orden</u> de un implicante (implicado) es el logaritmo en base 2 del número de entradas en las que vale 1 (0).

El orden (*k)* del implicante (implicado) está relacionado con el número de literales distintos (*l*) que incluye la expresión del término producto (suma) que lo representa y con el número total de variables de la función (*n*) de la forma siguiente: $k=n-l$

Minimizado mediante K-mapa

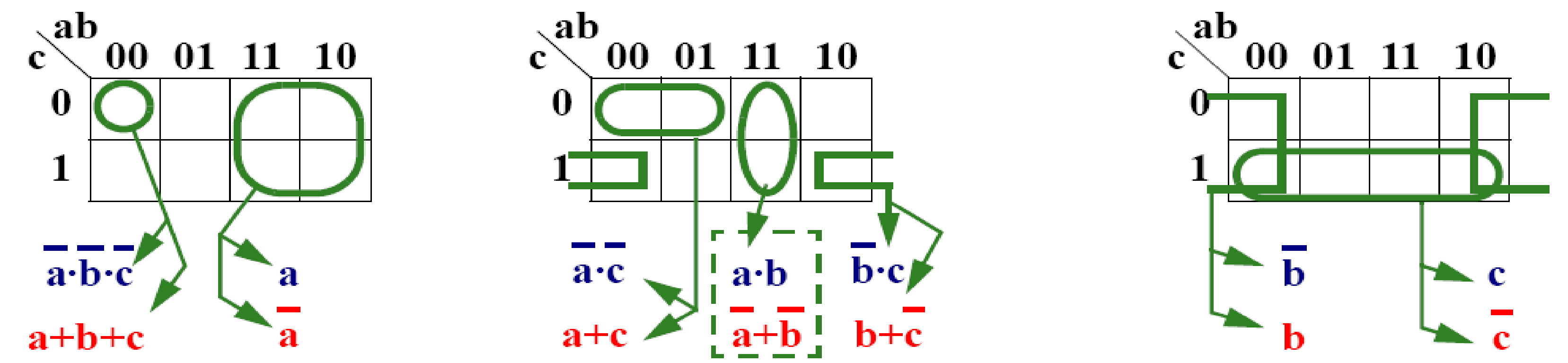
Implicante de una función f

Orden	Nº de 1's	Nº variables	Ejemplo 5 var.	
			Implicante	Cuantas
0	$1=2^0$	n	ab'cd'e	32
1	$2=2^1$	n - 1	ab'd'e	80
2	$4=2^2$	n - 2	ab'e	80
3	$8=2^3$	n - 3	b'e	40
4	$16=2^4$	n - 4	b'	10
5	$32=2^5$	n - 5	1	1
k	$m=2^k$	n - k		

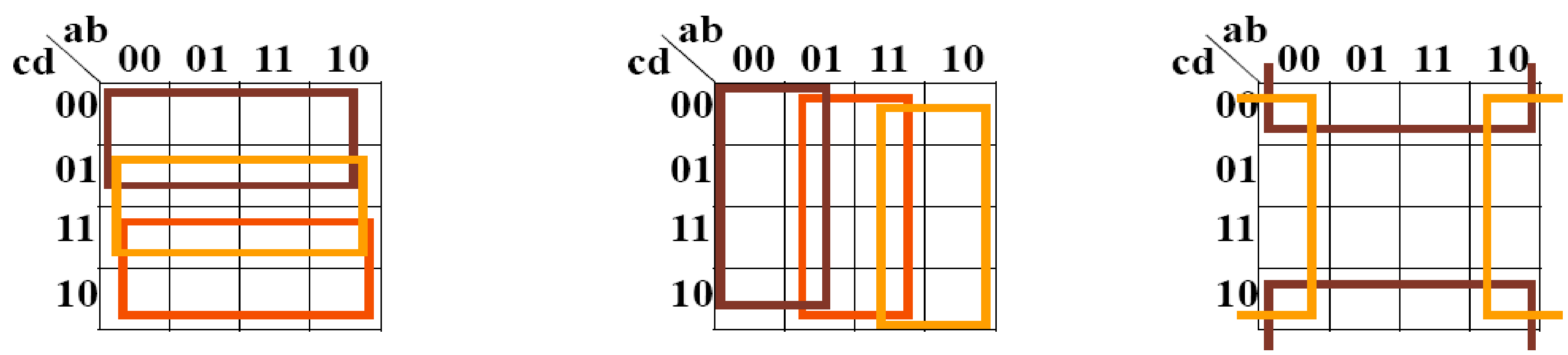
Diseño de Circuitos Combinacionales

Minimizado mediante K-mapa

Algunas términos producto y términos suma



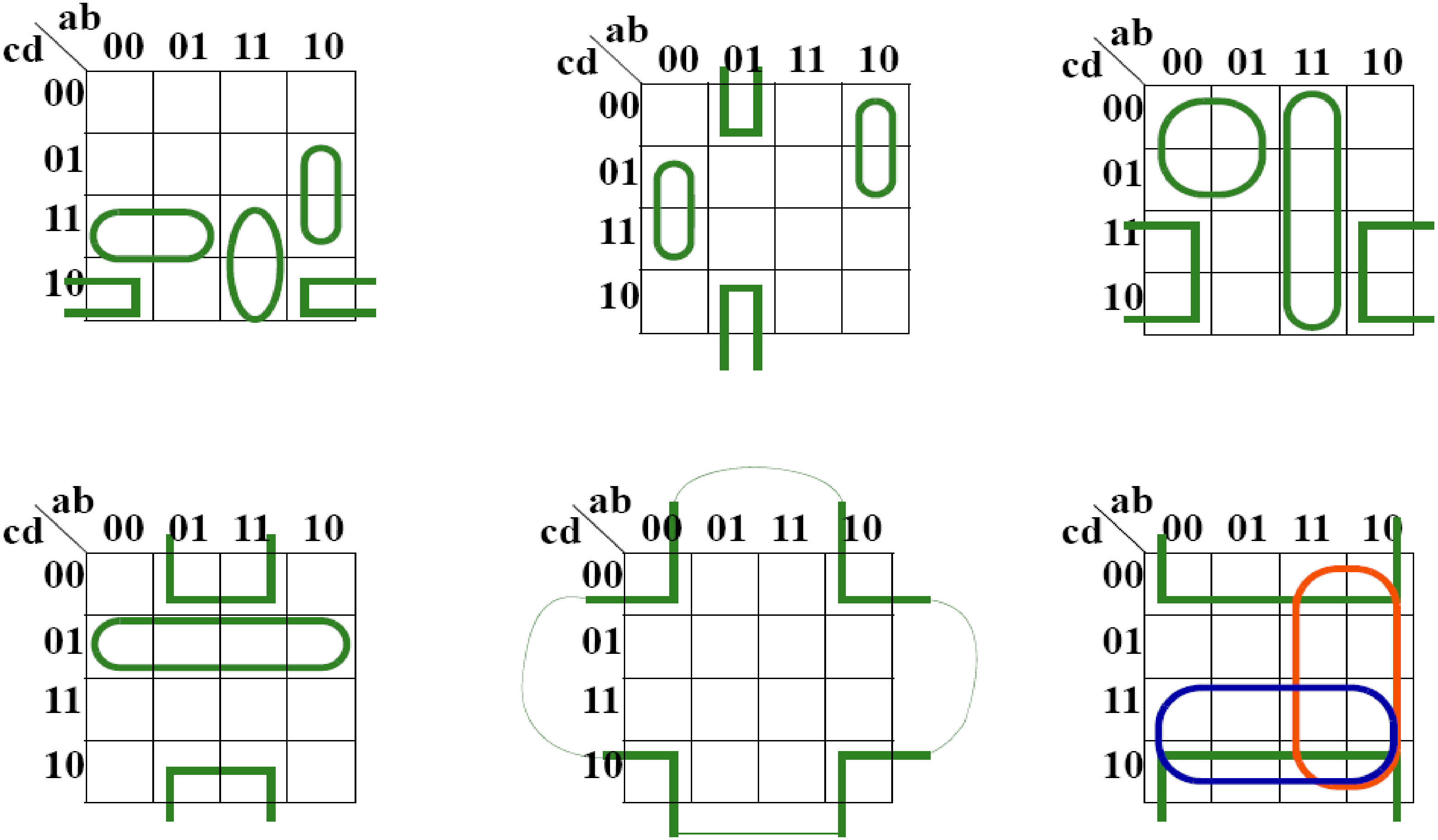
Los mapas de 4 variables contienen varios mapas de 3



Diseño de Circuitos Combinacionales

Minimizado mediante K-mapa

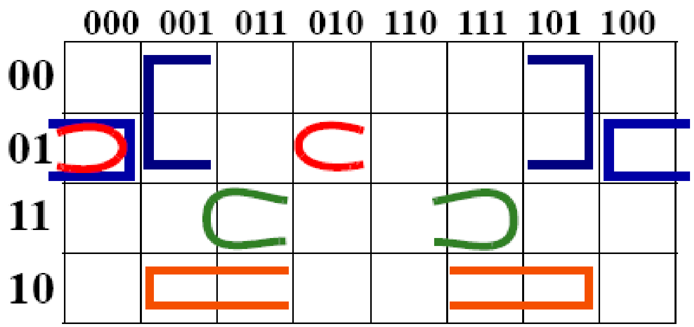
Algunas términos producto y términos suma

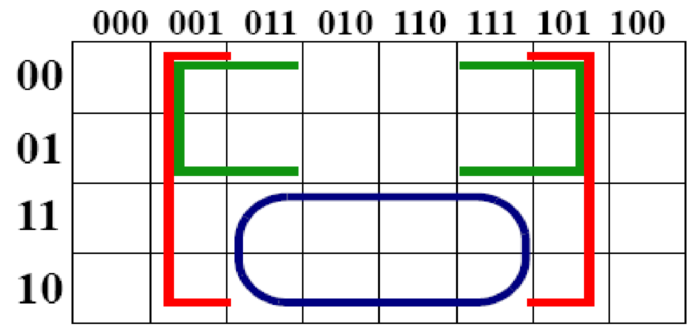


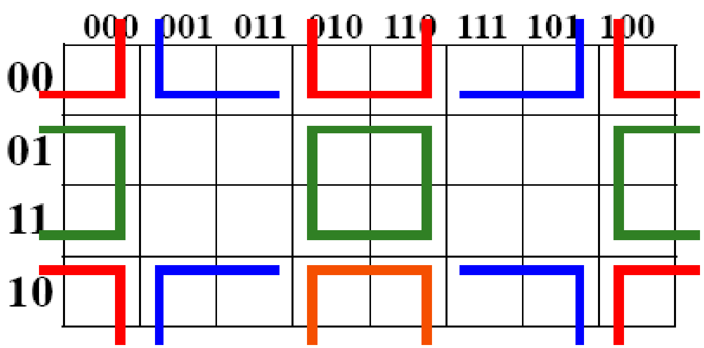
Diseño de Circuitos Combinacionales

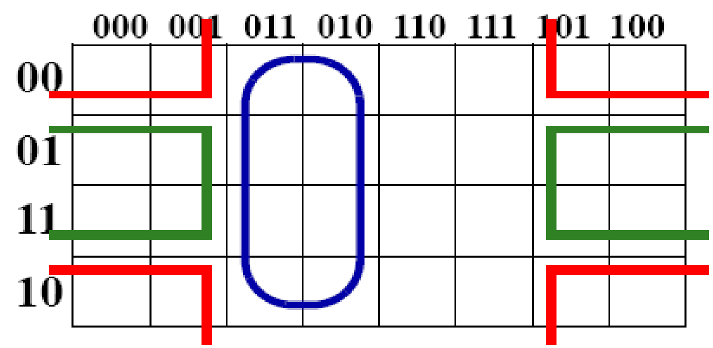
Minimizado mediante K-mapa

Algunas términos producto y términos suma



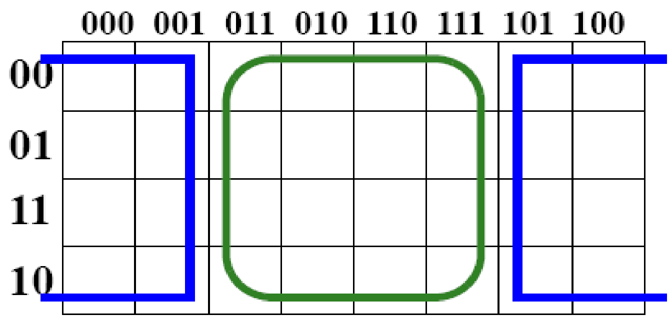


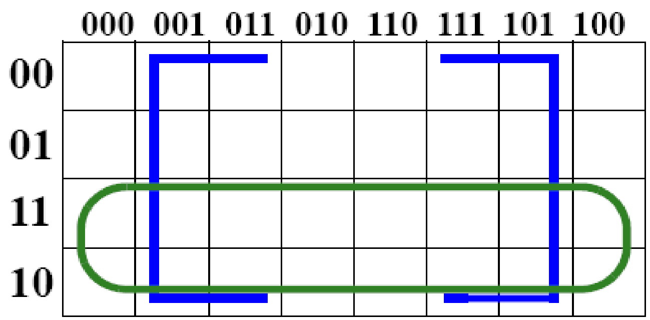


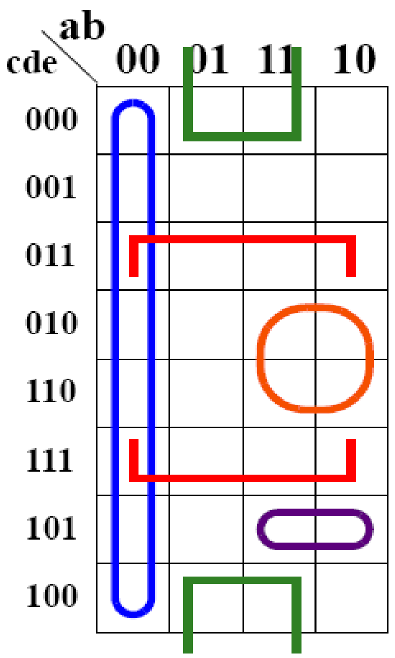


Minimizado mediante K-mapa

Algunas términos producto y términos suma







Diseño de Circuitos Combinacionales

Minimizado mediante K-mapa

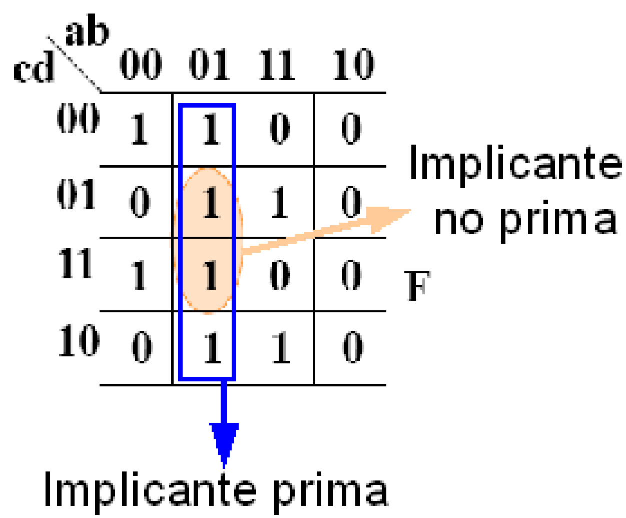
Definiciones: Sea *I* una implicante de una función *F* ...

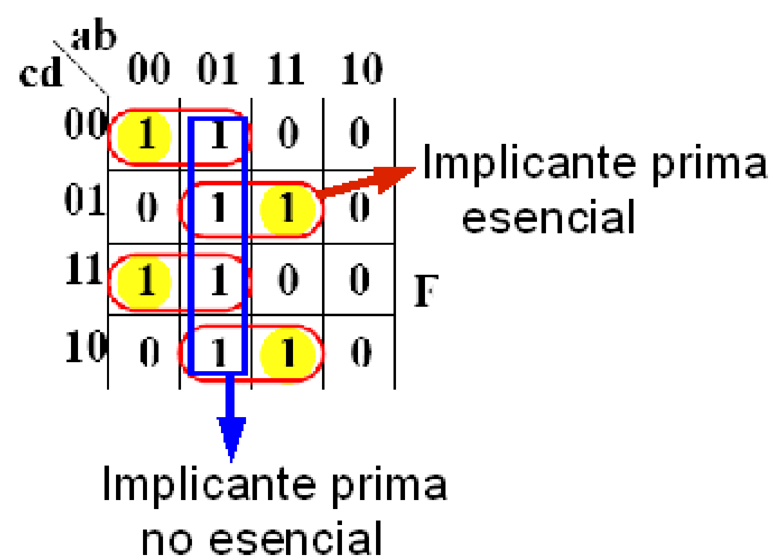
-se dice que *I* es <u>*prima*</u> ⇔ (si y sólo si) no existe otra implicante *Y* de *F* tal que *I* es implicante de *Y*.

-se dice que un mintérmino *m* <u>es</u> de *F* ⇔ el valor *x* tal que *m*(*x*)=1 pertenece al dominio de *F* y *F*(*x*)=1. A *m* se le llamá 1 de *F*.

-se dice que *I* <u>cubre</u> un mimtérmino *m* de *F* ⇔ *m* es implicante de *I*.

-se dice que *I* es <u>esencial</u> ⇔ es prima y, además, existe un mimtérmino *m* de *F* (mimtérmino <u>distinguido</u>) tal que *I* es el único implicante primo que cubre a *m*.





Diseño de Circuitos Combinacionales

Minimizado mediante K-mapa

Definiciones: Sea *I* una implicada de una función *F* ...

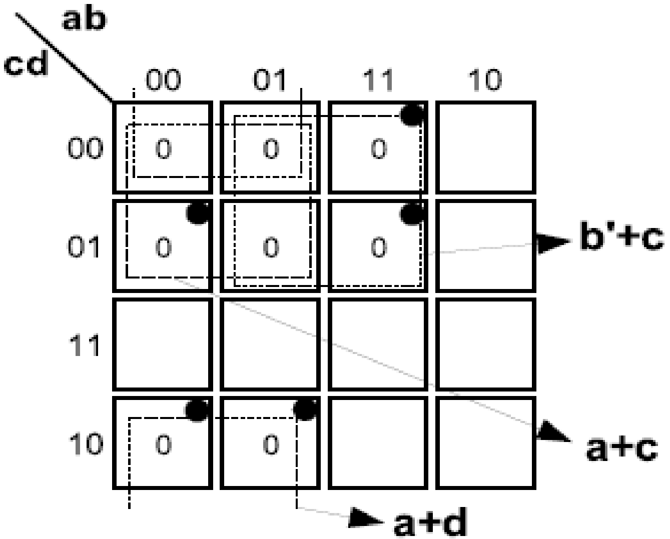
-se dice que *I* es <u>*prima*</u> ⇔ (si y sólo si) no existe otra implicada *Y* de *F* tal que *I* es implicada de *Y*.

-se dice que un maxtérmino *M* <u>es</u> de *F* ⇔ el valor *x* tal que *M*(*x*)=0 pertenece al dominio de *F* y *F*(*x*)=0. A *M* se le llama 0 de *F*.

-se dice que *I* <u>cubre</u> un maxtérmino *M* de *F* ⇔ *M* es implicada de *I*.

-se dice que *I* es <u>esencial</u> ⇔ es prima y, además, existe un maxtérmino *M* de F (maxtérmino <u>distinguido</u>) tal que *I* es el único implicado primo que cubre a *M*.

$$F = \prod(0,1,2,4,5,6,12,13)$$



Minimizado mediante K-mapa

Obtención de una expresión mínima en suma de productos

Una expresión mínima en suma de productos de una función contiene el menor número posible de implicantes y literales. Si *E* es una exresión mínima en forma de suma de productos de *F* se demuestra que:

- Todas las implicantes primas esenciales de *F* aparecen como operandos de la operación OR en *E*.
- Todos los operandos de la operación OR de *E* son implicantes primos de *F*.
- Todos los mintérminos de *F* están cubiertos al menos por uno de los términos producto que aparecen como operandos en la operación OR en *E*.

Directrices para la búsqueda de la expresión mínima:

1) Buscar implicantes primas esenciales. Éstas deben aparecer obligatoriamente en la expresión mínima en s.p.

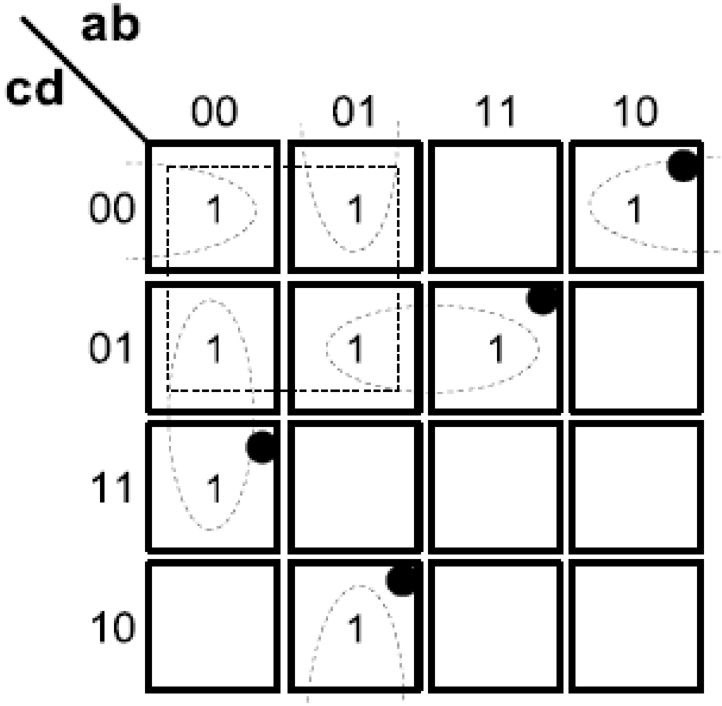
2) Para los mintérminos de la función que queden sin cubrir, probar de forma exaustiva todas las combinaciones de implicantes primas que los cubran en su totalidad.

Obtención de una expresión mínima en producto de sumas

Idénticos criterios que para s.p, pero usando implicadas primas.

Ejemplos de obtención de la expresión mínima

Ejercicio 1.- f = ∑(0,1,3,4,5,6,8,13)



Buscamos las i.p. esenciales

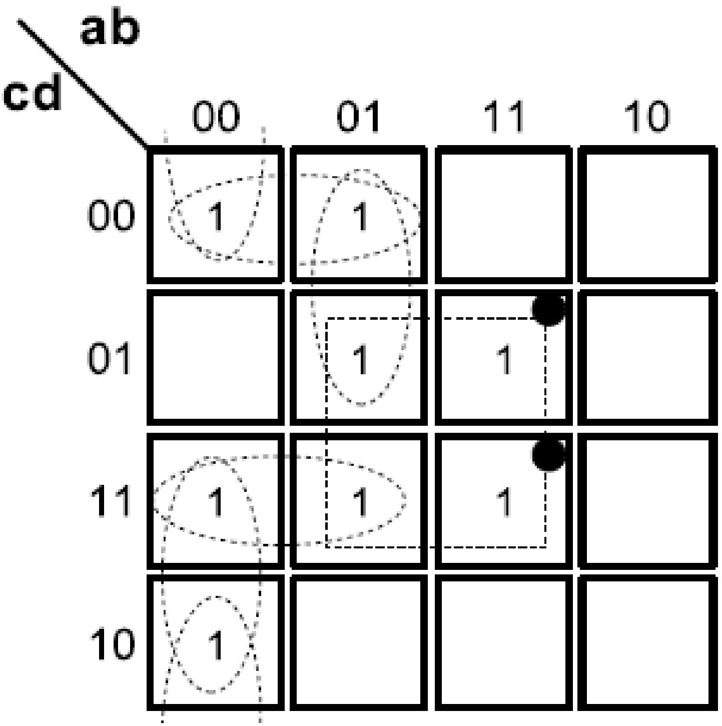
$$\mathbf{f(a,b,c,d) = \bar{a}\,\bar{b}\,d + \bar{b}\,\bar{c}\,\bar{d} + \bar{a}\,b\,\bar{d} + b\,\bar{c}\,d}$$

Fin. Se han cubierto todos los mintérminos sólo con las i.p. esenciales
¿como sería la expresión mínima en s.p. de **f(d,c,b,a)** ?

Diseño de Circuitos Combinacionales

Ejemplos de obtención de la expresión mínima

Ejercicio 2.- f = ∑(0,2,3,4,5,7,13,15)



1)Con las i.p. esenciales (“*bd*”) no se cubren todos los mintérminos.
2)El mintérmino 4 puede cubrirse con las i.p. “$\overline{a}\,\overline{c}\,\overline{d}$“ y **“**$\overline{a}\,b\,\overline{c}$”. Ambas son del mismo, coste pero la primera cubre también al mintérmino 0 que no estaba cubierto. Escogemos “$\overline{a}\,\overline{c}\,\overline{d}$“.
3) Ahora sólo quedan por cubrir los mintérminos 2 y3. Obviamente escogemos “$\overline{a}\,\overline{b}\,c$“.

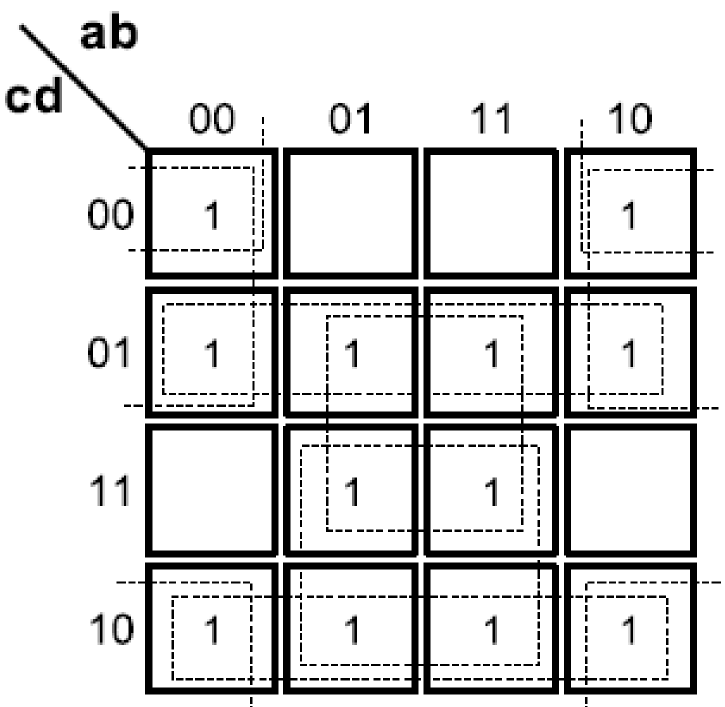
$$\mathbf{f(a,b,c,d) = b\,d + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}\,\overline{b}\,c}$$

Fin. Se han cubierto todos los mintérminos

Diseño de Circuitos Combinacionales

Ejemplos de obtención de la expresión mínima

Ejercicio 3.- f = ∑(0,1,2,5,6,7,8,9,10,13,14,15)



1)No hay i.p. Esenciales.
2)Nos fijamos en el mintérmino 0. Está cubierto por las i.p. Ip “$\bar{b}\,\bar{d}$“ y **“**$\bar{b}\,\bar{c}$”, ambas del mismo coste. **Suponemos** que la expresión mínima incluye a **“**$\bar{b}\,\bar{c}$”.

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

3) Repetimos el proceso suponiendo ahora que la expresión mínima incluye a “$\bar{b}\,\bar{d}$“ y vemos si optenemos una expresión de menor coste.

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

Ambas expresiones tienen el mismo coste. Cualquiera de ellas es solución mínima.

Diseño de Circuitos Combinacionales

Ejemplos de obtención de la expresión mínima

Ejercicio 5.- f = ∑(0,5,6,7,8,13,15,16,20,21,22,23,24,29,31)

de \ abc	000	001	011	010	110	111	101	100
00	1			1	1		1	1
01		1	1			1	1	
11		1	1			1	1	
10		1					1	

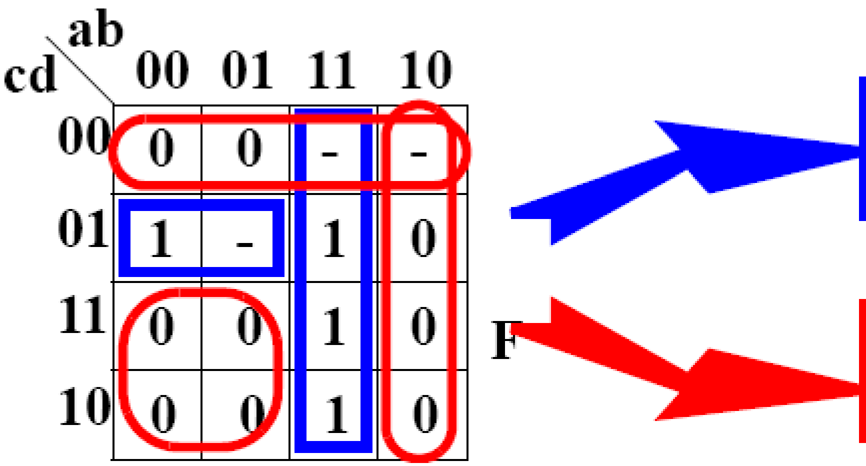
$$f(a,b,c,d,e) = \overline{c}\,\overline{d}\,\overline{e} + c\,e + \overline{b}\,c\,d + a\,\overline{b}\,c$$

Diseño de Circuitos Combinacionales

Minimizado mediante K-mapa

Este procedimiento también sirve para implementar **funciones incompletamente especificadas**. El valor una de una expresión para una entrada que no esté en el dominio de la función (inespecificación) es irrelevante. Por tanto se escoge sencillamente la expresión de menor coste.

Ejemplo: $F = \Sigma\,(1, 13, 14, 15) + d(5, 8, 12)$

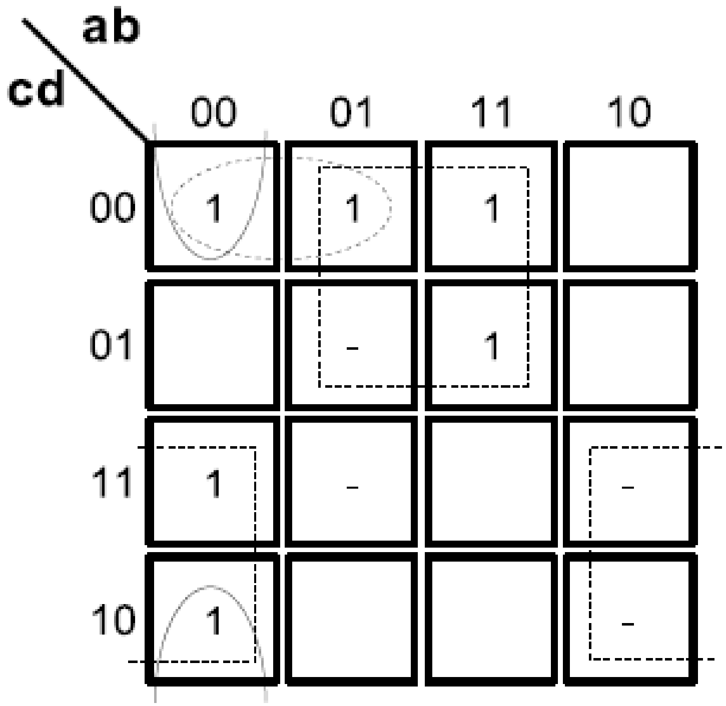


$F_{sp} = a \cdot b + \bar{a} \cdot \bar{c} \cdot d$ $\Rightarrow$ **5 y 12 se hacen 1**

$F_{ps} = (a+\bar{c}) \cdot (c+d) \cdot (\bar{a}+b)$ $\Rightarrow$ **8 y 12 se hacen 0**

Ejemplos de obtención de la expresión mínima

Ejercicio 4.- f = ∑(0,2,3,4,12,13) + d(5,7,10,11)



1)Buscamos las i.p. esenciales usando inespecificaciones como mintérminos y rechazando aquellas formadas sólo por inespecificaciones: **“**$b\,\overline{c}$”.

2)La mejor opción para cubrir el mintérmino 3 es “$\overline{b}\,c$”.

3)Solo queda por cubrir el mintérmino 0. Da igual si usamos “$\overline{a}\,\overline{c}\,\overline{d}$” o “$\overline{a}\,\overline{b}\,\overline{d}$”.

4)No se cubren las inespecificaciones restantes.

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

Fin. Se han cubierto todos los mintérminos

Pasos del proceso

Paso 4:

Si la expresión mínima es una suma de productos se implementa en dos niveles AND-OR o NAND-NAND.

Si es un producto de sumas se implementa en dos niveles OR-AND o NOR-NOR.

Ejemplo

Se requiere un circuito con una entrada de cuatro bits X y una salida G. La salida debe valer 1 si y sólo si el número representado por X en binario natural es mayor que 6.

a) Impleméntelo de forma óptima en dos niveles de puertas NAND.

b) Impleméntelo de forma óptima en dos niveles de puertas NOR.

Haced, al menos, ejercicio 19 de

https://www.dte.us.es/docencia/etsii/gii-ti/cedti/problemas/problemas-4.pdf/at_download/file