
Circuitos Electrónicos Digitales

Universidad de Sevilla

Tema IV

Circuitos Combinacionales

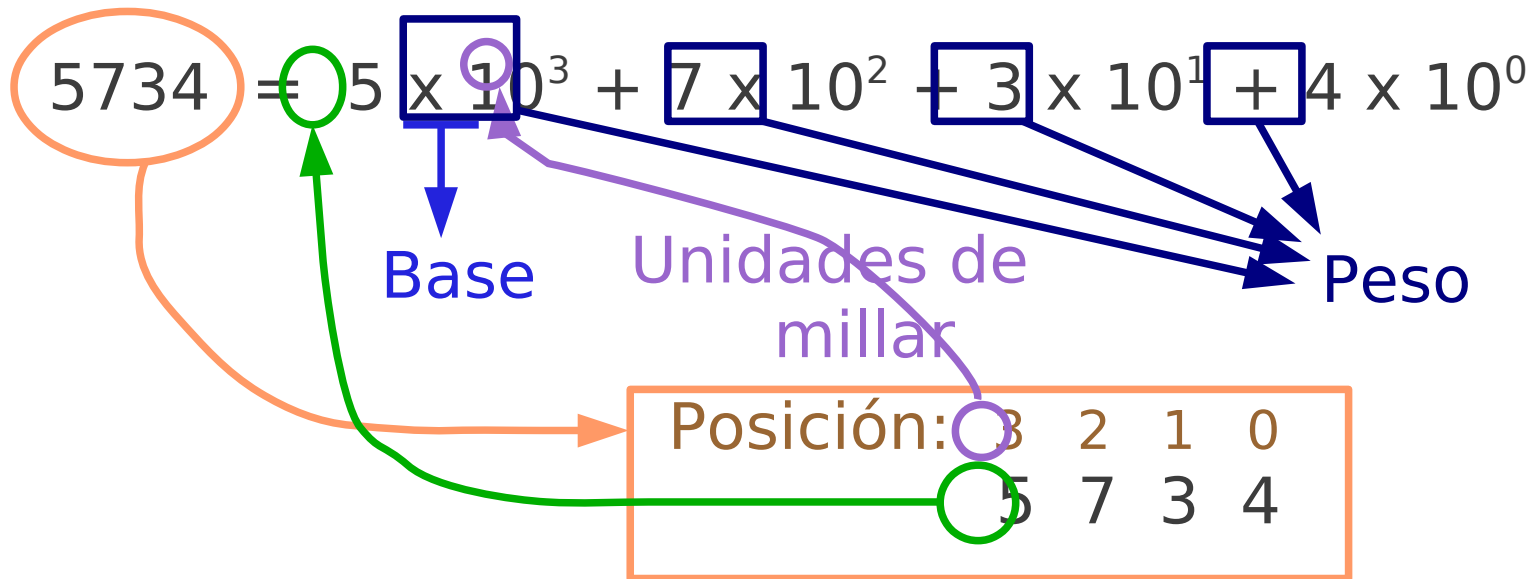
Indice

1. Representación posicional de magnitudes
2. Funciones combinacionales
3. Análisis de circuitos combinacionales
4. Diseño de circuitos combinacionales

Tema IV – Parte I

Representación Posicional de Magnitudes

Representación Posicional de Magnitudes



Base: Número de dígitos distintos que pueden emplearse para representar una magnitud con el sistema utilizado.

Representación Posicional de Magnitudes

Bases interesantes

Base 2: Binario → 0, 1

$$010011_{(2)} = 19_{(10)}$$

Base 8: Octal → 0, 1, 2, 3, 4, 5, 6, 7

$$47_{(8)} = 39_{(10)}$$

Base 16: Hexadecimal → 0, 1, 2, 3, 4, 5, 6, 7, 8,
9, A, B, C, D, E, F

$$2A_{(16)} = 42_{(10)}$$

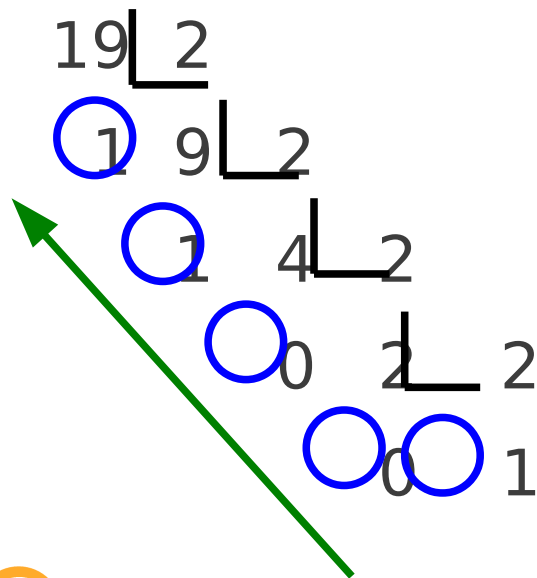
Representación Posicional de Magnitudes

Transformaciones entre bases

Base 2 a Base 10:

$$010011_{(2)} = 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 19_{(10)}$$

Base 10 a Base 2: $19_{(10)} = 10011_{(2)}$



Representación Posicional de Magnitudes

Transformaciones especiales

Base 2 a Base 16:

$$010011_{(2)} = 0001\ 0011_{(2)} = 13_{(16)} \quad \boxed{16} = 2^4$$

Base 16 a Base 2:

$$A\ 7_{(16)} = 1010\ 0111_{(2)}$$

Base 2 a Base 8:

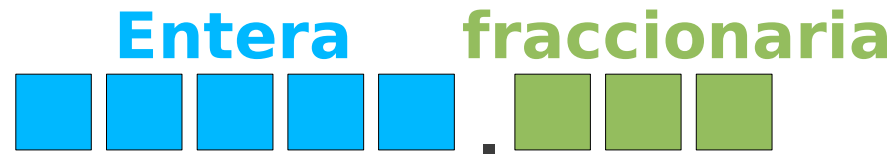
$$010011_{(2)} = 010\ 011_{(2)} = 2\ 3_{(8)} \quad \boxed{8} = 2^3$$

Base 8 a Base 2:

$$3\ 7_{(8)} = 011\ 111_{(2)}$$

Representación Posicional de Magnitudes

Representación parte fraccionaria

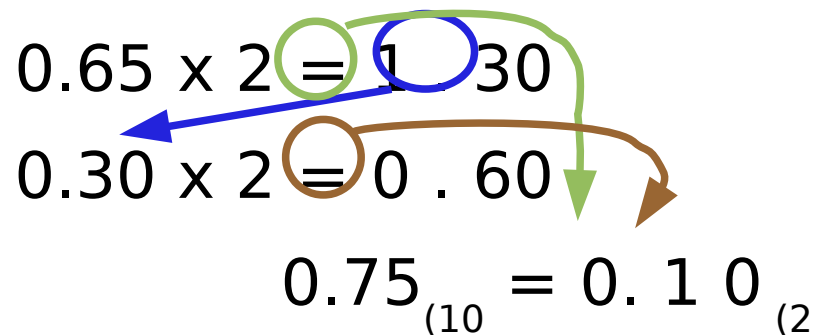


Ejemplo: 0 1 0 1 1 . 1 1 0

Base 2 a Base 10:

$$0.110_{(2)} = 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} = 0.75_{(10)}$$

Base 10 a Base 2:


$$0.65 \times 2 = 1.30$$
$$0.30 \times 2 = 0.60$$
$$0.75_{(10)} = 0.10_{(2)}$$

Códigos Binarios

BCD

7 Segmentos

Gray

Detectores de Errores

ASCII

Códigos Binarios

BCD

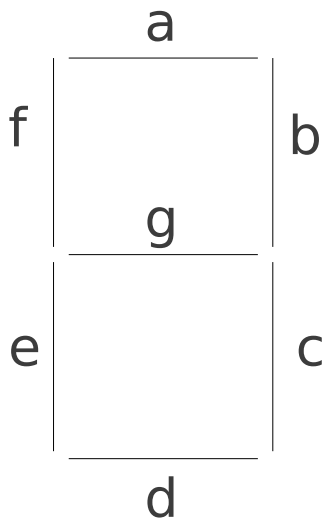
Dígito	Código BCD
0	0000
1	0001
2	0010
3	0011
4	0100

Dígito	Código BCD
5	0101
6	0110
7	0111
8	1000
9	1001

Ejemplo: $16_{(10)} = 0001\ 0110_{(BCD)}$

Códigos Binarios

7 Segmentos



Dígito	Código 7-Seg abcdefg
0	1111110
1	0110000
2	1101101
3	1111001
4	0110011

Dígito	Código 7-Seg abcdefg
5	1011011
6	0011111
7	1110000
8	1111111
9	1110011

Códigos Binarios

Gray

Dígito	Código Gray 1 bit
0	0
1	1

Dígito	Código Gray 2 bits
0	0 0
1	0 1
2	1 1
3	1 0

- - - Espejo

Ejercicio: Construir el código Gray de 3 bits

Códigos Binarios

Detectores de errores

- Bit de Paridad: Se añade un bit (más significativo) al código binario, denominado **bit de paridad**. Puede hacerse de dos formas:
 1. Paridad Par: El número total de 1s debe ser par.
 2. Paridad impar: El número total de 1s debe ser impar.

Código	Bit Paridad Par	Código con Paridad Par	Bit Paridad Impar	Código con Paridad Impar
0000	0	0 0000	1	1 0000
1011	1	1 1011	0	0 1011

Códigos Binarios

ASCII

American Standard Code for Information Interchange (ASCII)

$B_4B_3B_2B_1$	$B_7B_6B_5$							
	000	001	010	011	100	101	110	111
0000	NULL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Tema IV – Parte II

Funciones Combinacionales

Funciones Combinacionales



Definición: Una función de conmutación es una aplicación
 $f: B^n \rightarrow B. f(x_0, x_1, x_2, \dots, x_n)$

x_i son **variables binarias**.

Una función de conmutación es **completamente especificada** cuando asigna un valor (0 o 1) a todos los posibles valores de sus variables. En otro caso, la función es **incompletamente especificada**.

Funciones Combinacionales

Ejemplos:

$X_2 X_1 X_0$	F
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

$X_2 X_1 X_0$	F
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

Representación

Existen varias formas de representar una función de conmutación:

Expresión

Tabla de verdad

Mapa

Circuito

Lenguajes de descripción de hardware (LDH) : Verilog
([Transparencias de introducción al LDH](#))

Funciones Combinacionales

Ejemplo: función de tres variables

Expresión: $f(x,y,z) = xy + xz + yz$

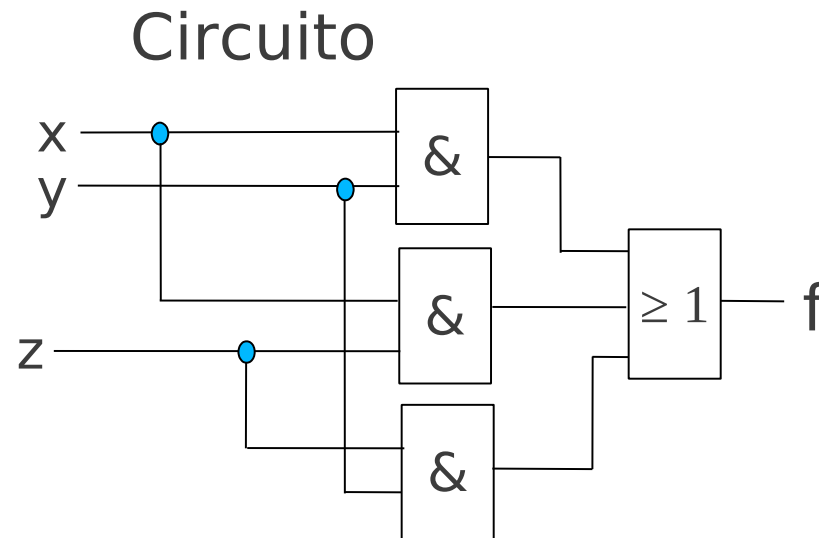
Tabla de verdad

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

Mapa

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

f



Código Verilog

```
module ejemplo(  
    output f,  
    input x,  
    input y,  
    input z,  
)  
  
    assign f = x & y | x & z | y & z;  
endmodule
```

Funciones Combinacionales

Ejemplo: función de 4

variables
Expresión: $g(x,y,z,u) = xyu + xzu' + yz$

$\bar{x}y$ zu	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

Las formas normalizadas son la **suma de productos** y el **producto de sumas**.

Los términos producto siempre determinan los unos de la función y los términos suma los ceros.

Elemento dominante:

- Respecto a operador + $\rightarrow 1$ ($1+x=1$)
- Respecto a operador \cdot $\rightarrow 0$ ($0x=0$)

Ejemplos:

- $f = xy + y' + yz$ Suma de productos
- $g = x(y+z)$ Producto de sumas
- $h = (abc + b'ad + (a+b+c)' + de)'$ No normalizado

Funciones Combinacionales

Forma canónica disyuntiva

Es una **suma de productos** compuesta sólo de mintérminos.

Mintérmino: término producto en el que aparecen todas las variables de la función, complementada o sin complementar, una única vez.

Existen 2^n mintérminos de n variables.

Ejemplo: para 3 variables hay 8 mintérminos que son:

$x'y'z'$, $x'y'z$, $x'yz'$, $x'yz$, $xy'z'$, $xy'z$, xyz' , xyz

Funciones Combinacionales

Forma canónica disyuntiva

Teorema: Dada una lista completa de los minterminos de n variables, si a cada una de las n variables se le asigna el valor 0 o 1, entonces sólo un mintermino de la lista tomará el valor 1 y los otros el valor 0.

Ejemplo:

Para $xyz=110$, sólo el mintermino xyz' toma valor 1, el resto toma el valor 0.

Funciones Combinacionales

Forma canónica disyuntiva

Teorema: Cada función de conmutación completamente especificada puede expresarse en forma canónica de mintérminos.

Teorema: La forma canónica de mintérminos de una función de conmutación completa es única.

Ejemplo

$$g(x,y,z) = x'y'z' + x'yz' + xyz$$

$$g(x,y,z) = 1 \iff x'y'z' = 1 \rightarrow x=y=z=0$$

$$x'yz' = 1 \rightarrow x=z=0 \text{ e } y=1$$

$$xyz = 1 \rightarrow x=y=z=1$$

Funciones Combinacionales

Forma canónica disyuntiva

Notación m: Cada mintérmino se representa de la forma “mX” donde “X” es un número asociado a cada mintérmino de forma que:

1. Se establece un orden entre las variables. Ej.
 (x_1, x_2, x_3)
2. Se asocia un 0 a cada variable complementada
3. Se asocia un 1 a cada variable sin complementar
4. X se obtiene de interpretar en base 2 el código obtenido. Ej: $x_1'x_2x_3' \rightarrow 010 \rightarrow 2 \rightarrow m2$

Ejemplo:

$$f(x_1, x_2, x_3) = x_1'x_2'x_3' + x_1'x_2x_3' + x_1'x_2x_3 + x_1x_2x_3$$

$$f(x_1, x_2, x_3) = m0 + m2 + m3 + m7 = \sum m(0, 2, 3, 7)$$

Funciones Combinacionales

Forma canónica conjuntiva

Es un **producto de sumas** compuesto sólo de maxtérminos.

Maxtérmino: término suma en el que aparecen todas las variables de la función, complementada o sin complementar, una única vez.

Existen 2^n maxtérminos de n variables.

Ejemplo: para 3 variables hay 8 maxtérminos que son:

$$\begin{aligned} &x'+y'+z', x'+y'+z, x'+y+z', x'+y+z, \\ &x+y'+z', x+y'+z, x+y+z', x+y+z \end{aligned}$$

Funciones Combinacionales

Forma canónica conjuntiva

Teorema: Dada una lista completa de los maxtérminos de n variables, si a cada una de las n variables se le asigna el valor 0 o 1, entonces sólo un maxtérmino de la lista tomará el valor 0 y los otros el valor 1.

Ejemplo:

Para $xyz=110$, sólo el maxtérmino $x'+y'+z$ toma valor 0, el resto toma el valor 1.

Funciones Combinacionales

Forma canónica conjuntiva

Teorema: Cada función de conmutación completamente especificada puede expresarse en forma canónica de maxtérminos.

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

Ejemplo

$$g(x,y,z) = (x+y+z) (x+y'+z) (x'+y'+z')$$

$$g(x,y,z) = 0 \iff (x+y+z) = 0 \rightarrow x=y=z=0$$

$$(x+y'+z) = 0 \rightarrow x=z=0 \text{ e } y=1$$

$$(x'+y'+z') = 0 \rightarrow x=y=z=1$$

Funciones Combinacionales

Forma canónica conjuntiva

Notación M: Cada maxtérmino se representa de la forma “MX” donde “X” es un número asociado a cada maxtérmino de forma que:

1. Se establece un orden entre las variables. Ej. (x_1, x_2, x_3)
2. Se asocia un 0 a cada variable sin complementar
3. Se asocia un 1 a cada variable complementada
4. X se obtiene de interpretar en base 2 el código obtenido. Ej: $x_1' + x_2 + x_3 \rightarrow 100 \rightarrow 4 \rightarrow M4$

Ejemplo:

$$f(x_1, x_2, x_3) = (x_1' + x_2' + x_3')(x_1' + x_2 + x_3')(x_1' + x_2 + x_3)(x_1 + x_2 + x_3)$$

$$f(x_1, x_2, x_3) = M_7 M_5 M_4 M_0 = \prod(0, 4, 5, 7)$$

Tema IV – Parte III

Análisis de Circuitos Combinacionales

Análisis de Circuitos Combinacionales

Análisis Lógico

Dado un circuito, analizarlo consiste en encontrar:

- la expresión algebraica que implementa,
- su tabla de verdad y/o el k-mapa,
- explicación verbal de su función.

Análisis de Circuitos Combinacionales

Procedimiento

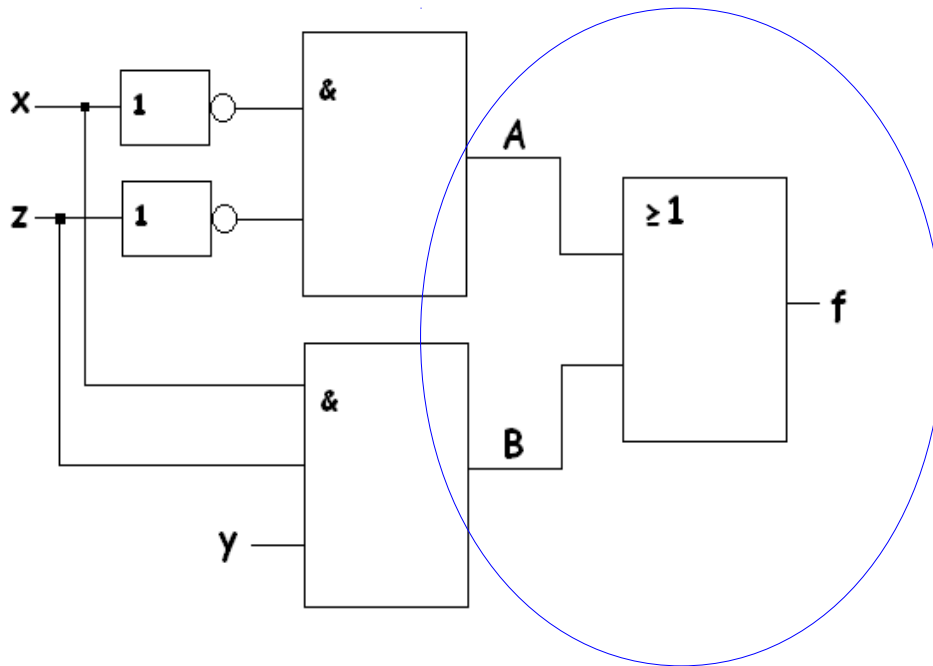
1. Se obtiene la función lógica realizada por las puertas cuyas entradas corresponden a las entradas primarias del circuito.
2. Se obtiene la función lógica realizada en puertas con entradas conocidas (entradas primarias o salidas de puertas ya calculadas).
3. Se repite el paso anterior hasta obtener la función de salida
4. Se simplifica la expresión obtenida y/o se traduce a un mapa o tabla

Análisis de Circuitos Combinacionales

Análisis Lógico

Ejemplo

Circuito:



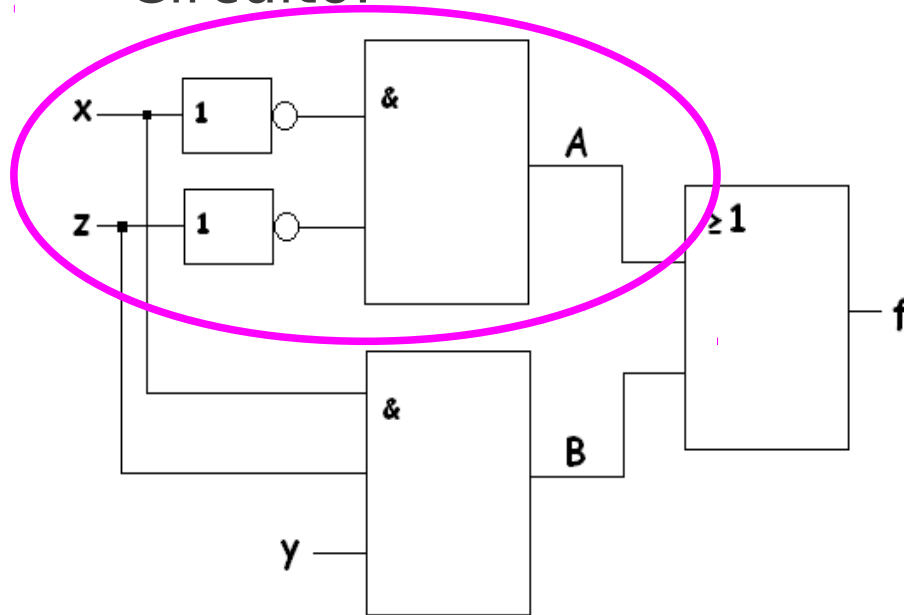
Expresión:

$$f(x,y,z) = A + B$$

Análisis de Circuitos Combinacionales

Análisis Lógico

Circuito:



Ejemplo

Tabla:

Expresión:

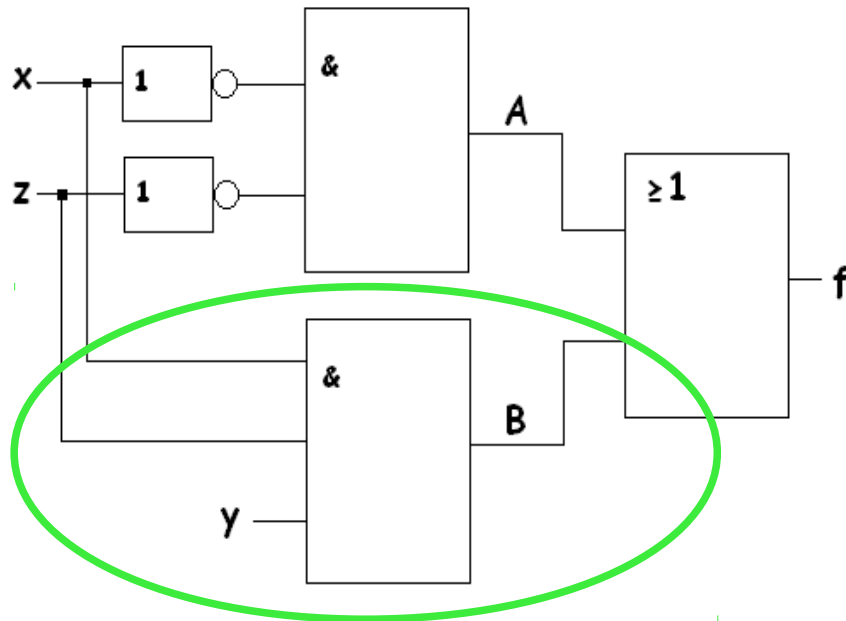
$$f(x,y,z) = A + B$$

$$A = x'z'$$

Análisis de Circuitos Combinacionales

Análisis Lógico

Circuito:



Ejemplo

Tabla:

Expresión:

$$f(x,y,z) = A + B$$

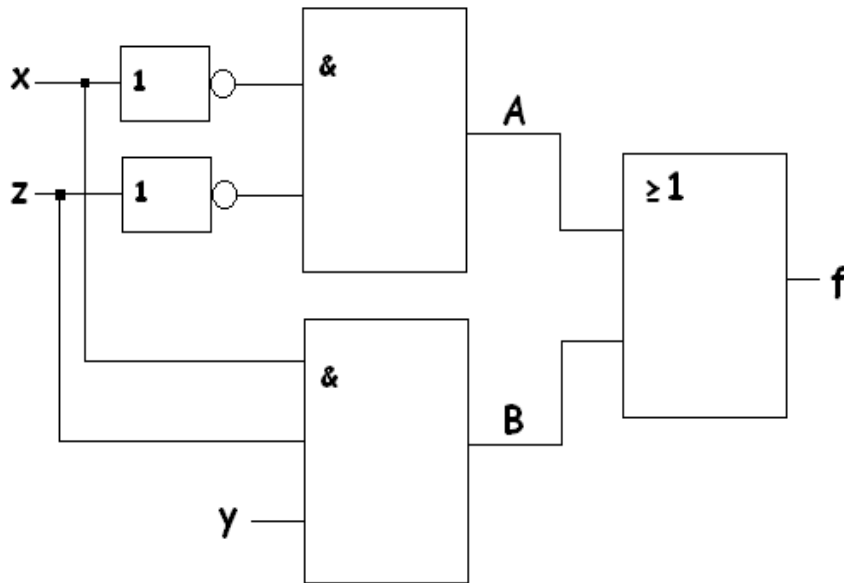
$$A = x'z'$$

$$B = xyz$$

Análisis de Circuitos Combinacionales

Análisis Lógico

Circuito:



Ejemplo

Tabla:

Expresión:

$$f(x,y,z) = A + B$$

$$A = x'z'$$

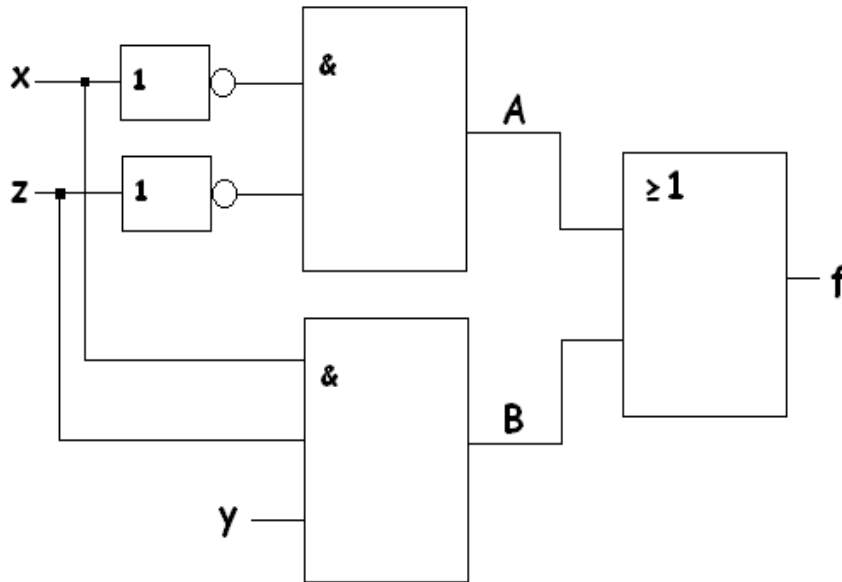
$$B = xyz$$

$$f(x,y,z) = xyz + x'z'$$

Análisis de Circuitos Combinacionales

Análisis Lógico

Circuito:



Ejemplo

Expresión:

$$f(x,y,z) = A + B$$

$$A = x'z'$$

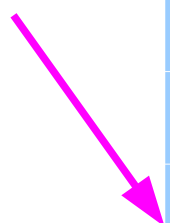
$$B = xyz$$

$$f(x,y,z) = xyz + x'z'$$

Tabla:

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

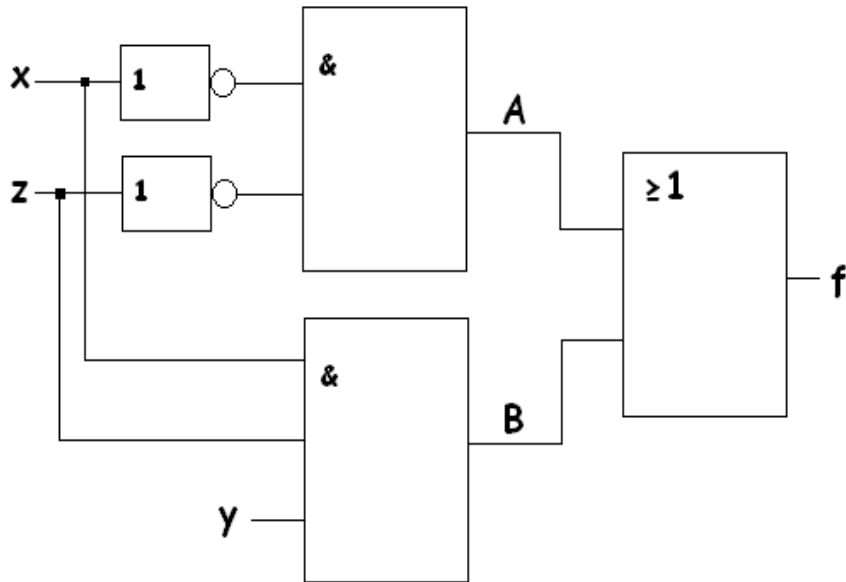
$$f(x,y,z) = 1 \text{ si } \begin{cases} xyz=1 \text{ si } x=y=z=1 \text{ (111)} \\ \text{ó} \\ x'z'=1 \text{ si } x=z=0 \text{ (0-0)} \end{cases}$$



Análisis de Circuitos Combinacionales

Análisis Lógico

Circuito:



Ejemplo

Expresión:

$$f(x,y,z) = A + B$$

$$A = x'z'$$

$$B = xyz$$

$$f(x,y,z) = xyz + x'z'$$

Tabla:

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

$$f(x,y,z) = 1 \text{ si } \begin{cases} xyz=1 \text{ si } x=y=z=1 & (111) \\ \text{ó} \\ x'z'=1 \text{ si } x=z=0 & (0-0) \end{cases}$$

Análisis de Circuitos Combinacionales

Análisis Temporal

Representa la evolución en el tiempo de las entradas y salidas del circuito. A esta representación temporal se la denomina **CRONOGRAMA**.

Dicha representación puede ser:

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

Análisis de Circuitos Combinacionales

Análisis Temporal

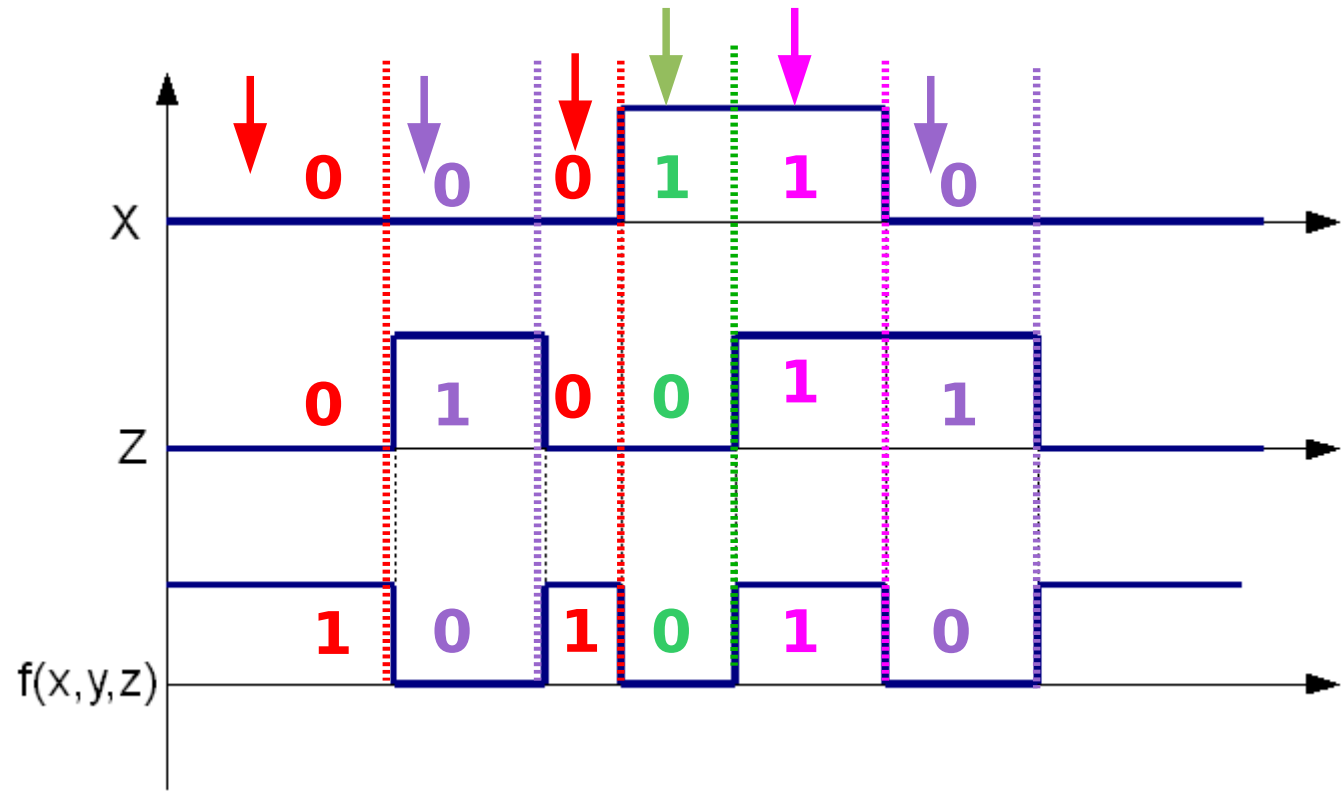
Tabla:

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

Ejemplo

- Cronograma (con $y=1$)

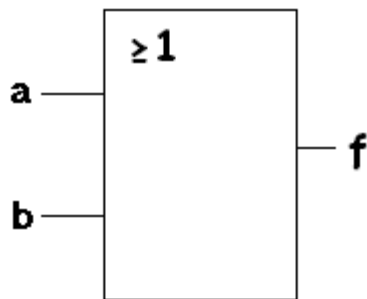
(sin considerar retrasos):



Análisis de Circuitos Combinacionales

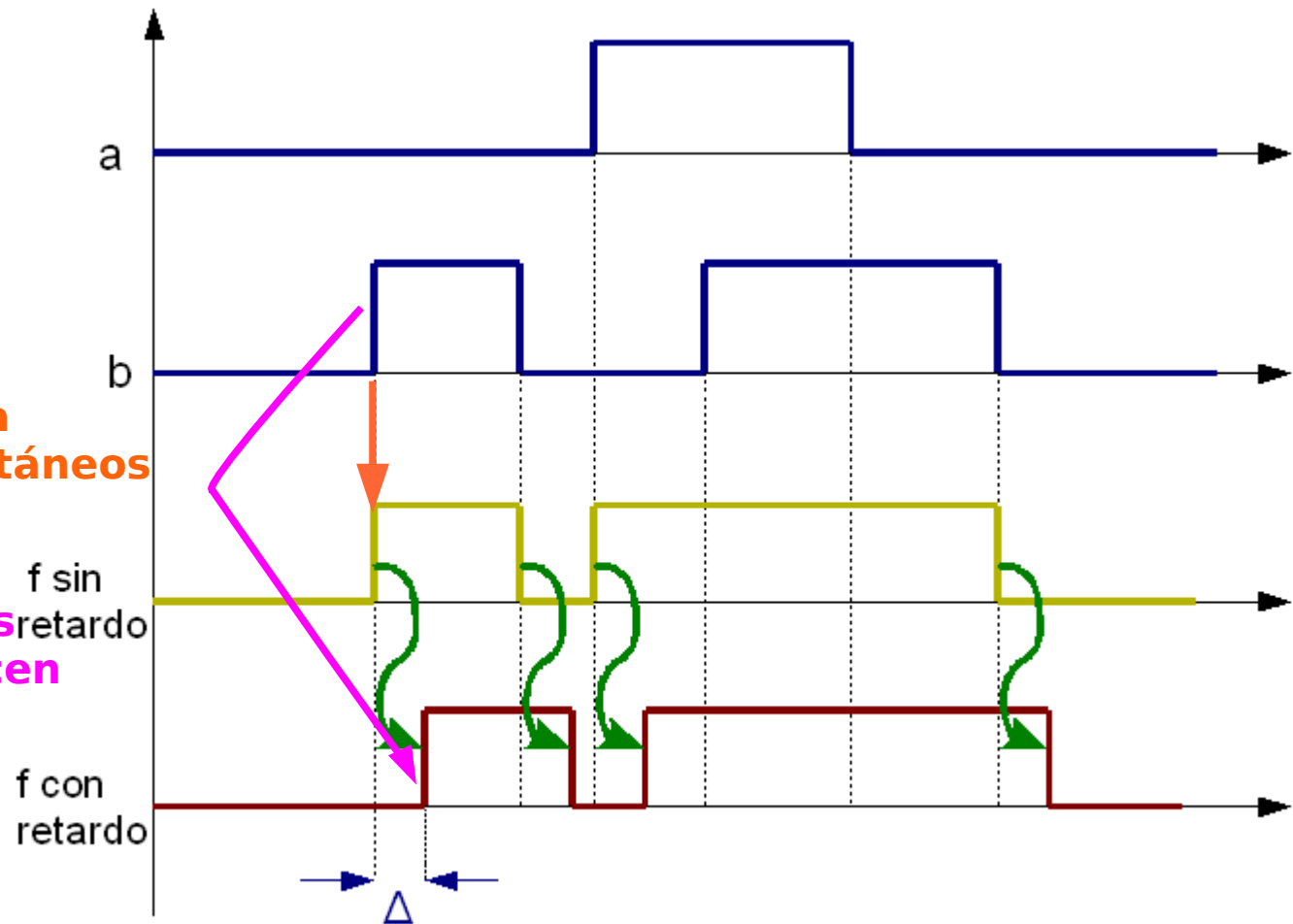
Análisis Temporal

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



Si no hubiera retrasos los cambios en la entrada provocan cambios instantáneos en la salida.

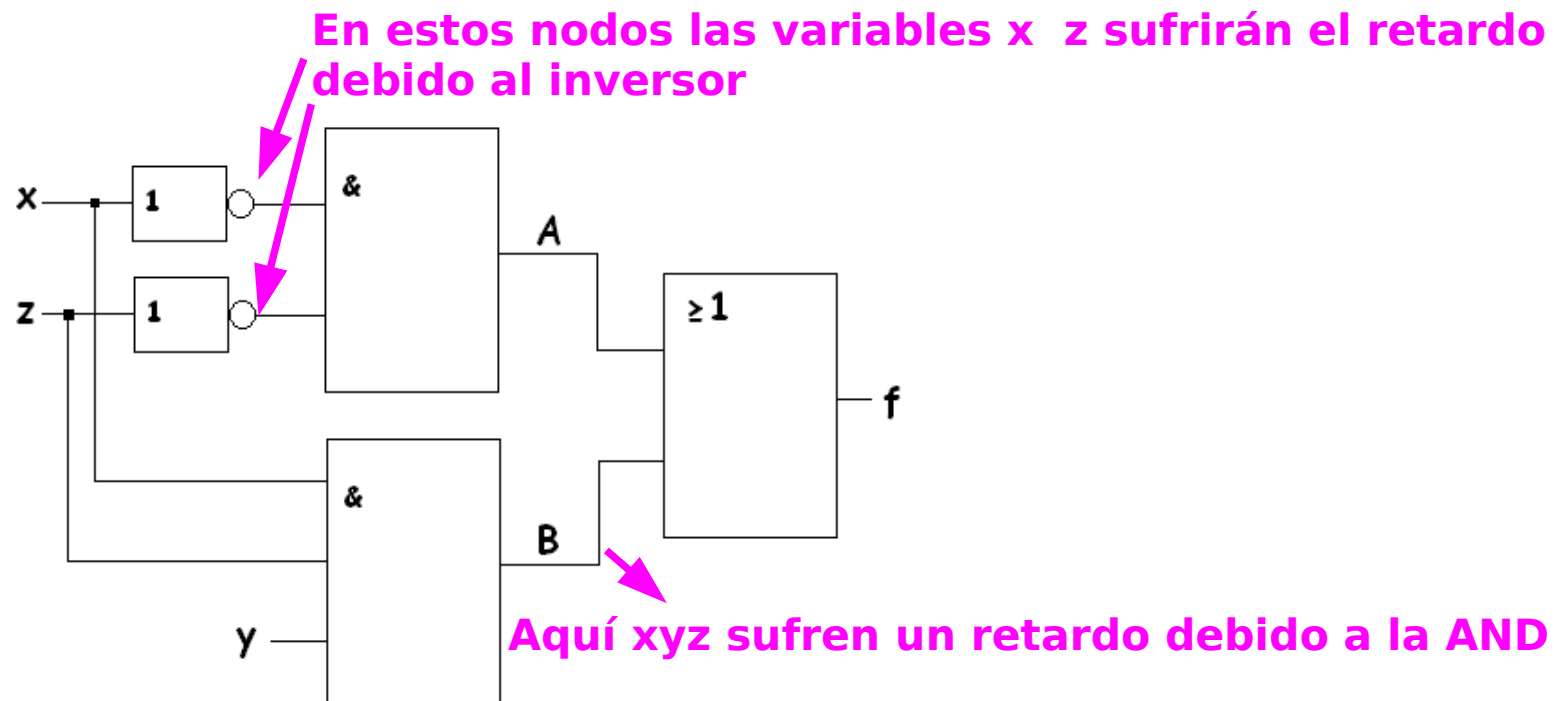
Con retardos, los cambios en las entradas se traducen en la salida un tiempo Δ después



Análisis de Circuitos Combinacionales

Análisis Temporal

Ejemplo $y=1$ y retrasos iguales para todas las puertas

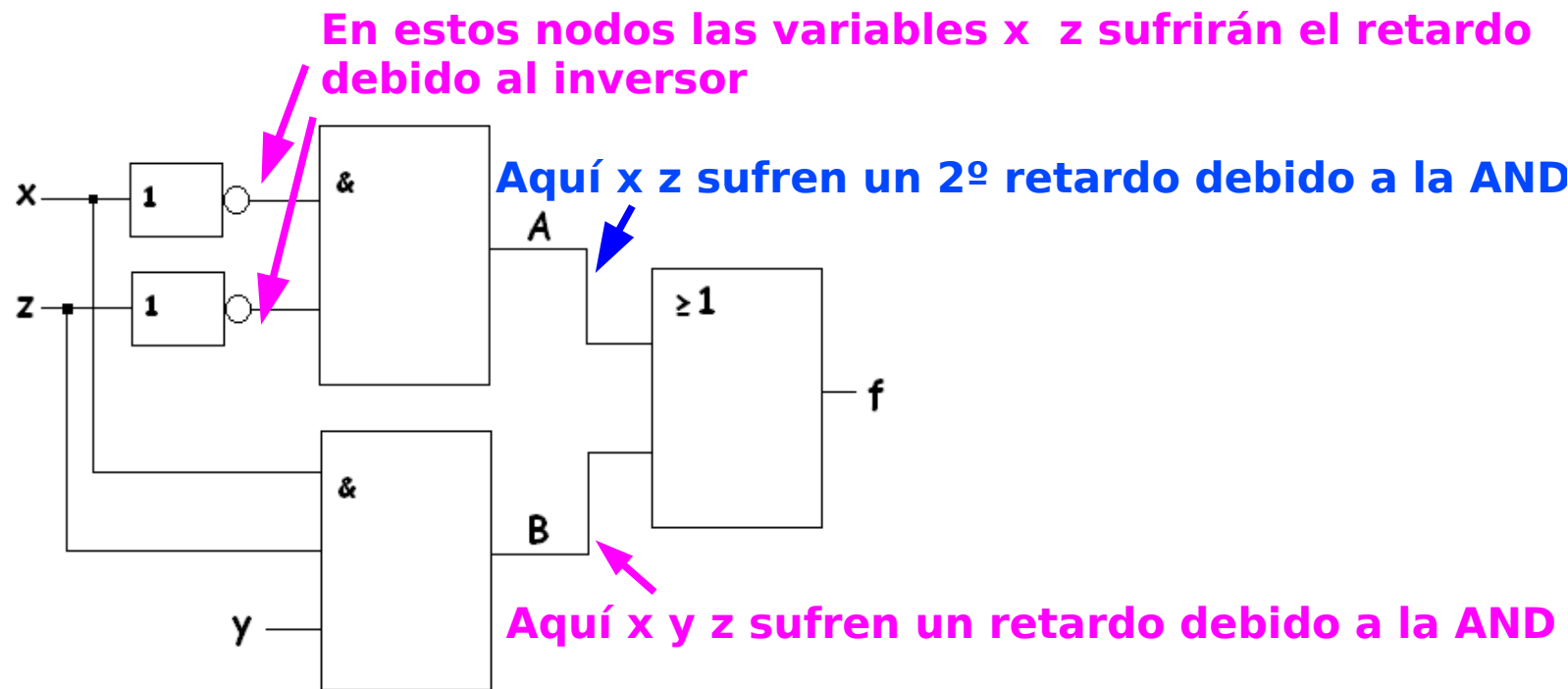


Importante: cuando se considera retardo hay que ir analizando lo que ocurre en cada nodo del circuito de entrada a salida. Una misma variable de entrada puede sufrir retardos diferentes dependiendo del camino

Análisis de Circuitos Combinacionales

Análisis Temporal

Ejemplo $y=1$ y retrasos iguales para todas las puertas

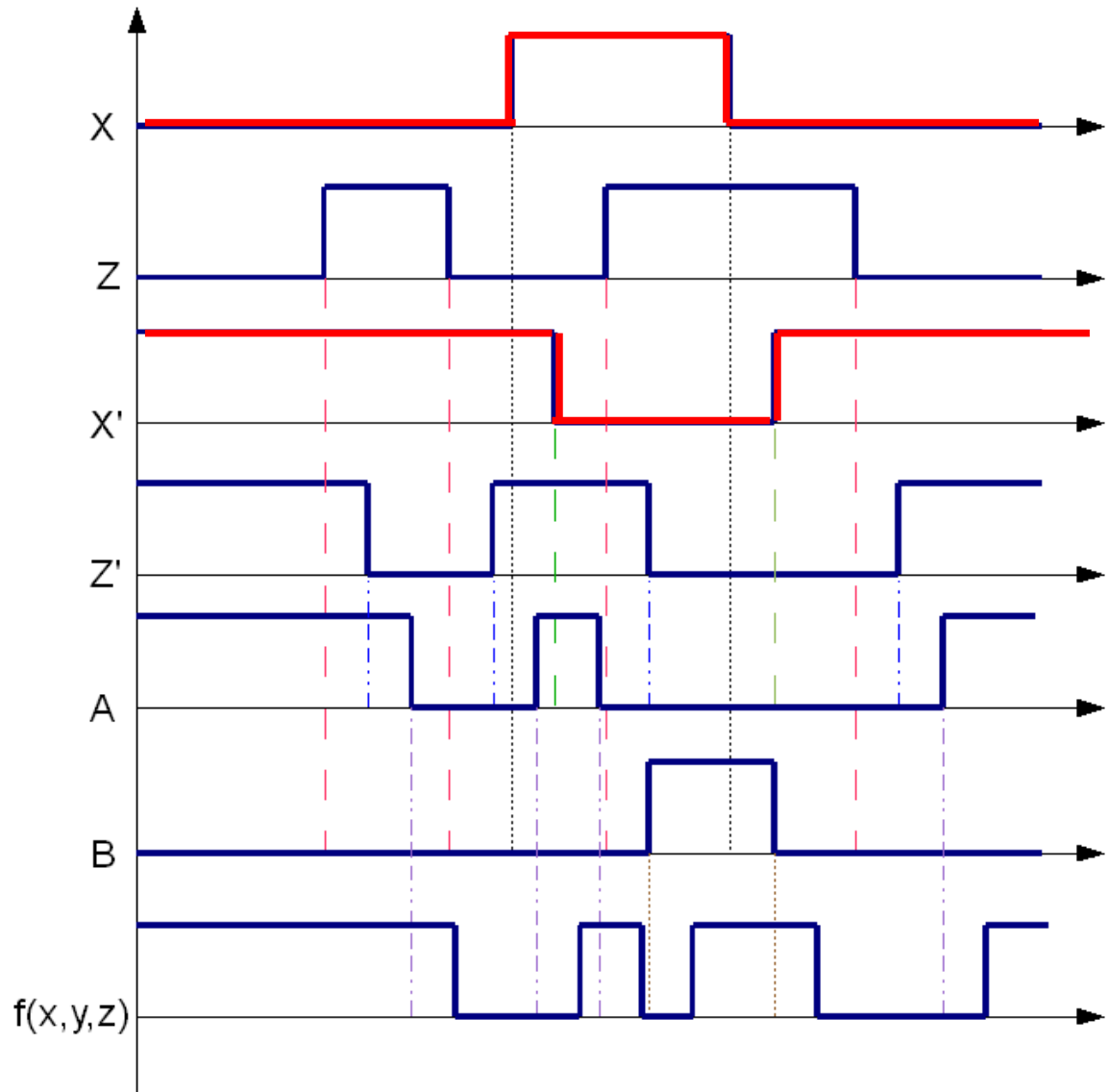
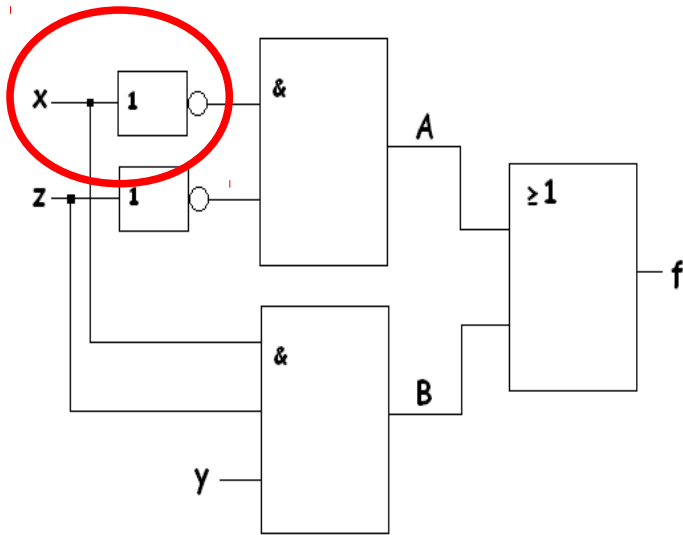


Importante: cuando se considera retardo hay que ir analizando lo que ocurre en cada nodo del circuito de entrada a salida. Una misma variable de entrada puede sufrir retardos diferentes dependiendo del camino

Análisis de Circuitos Combinacionales

Análisis Temporal

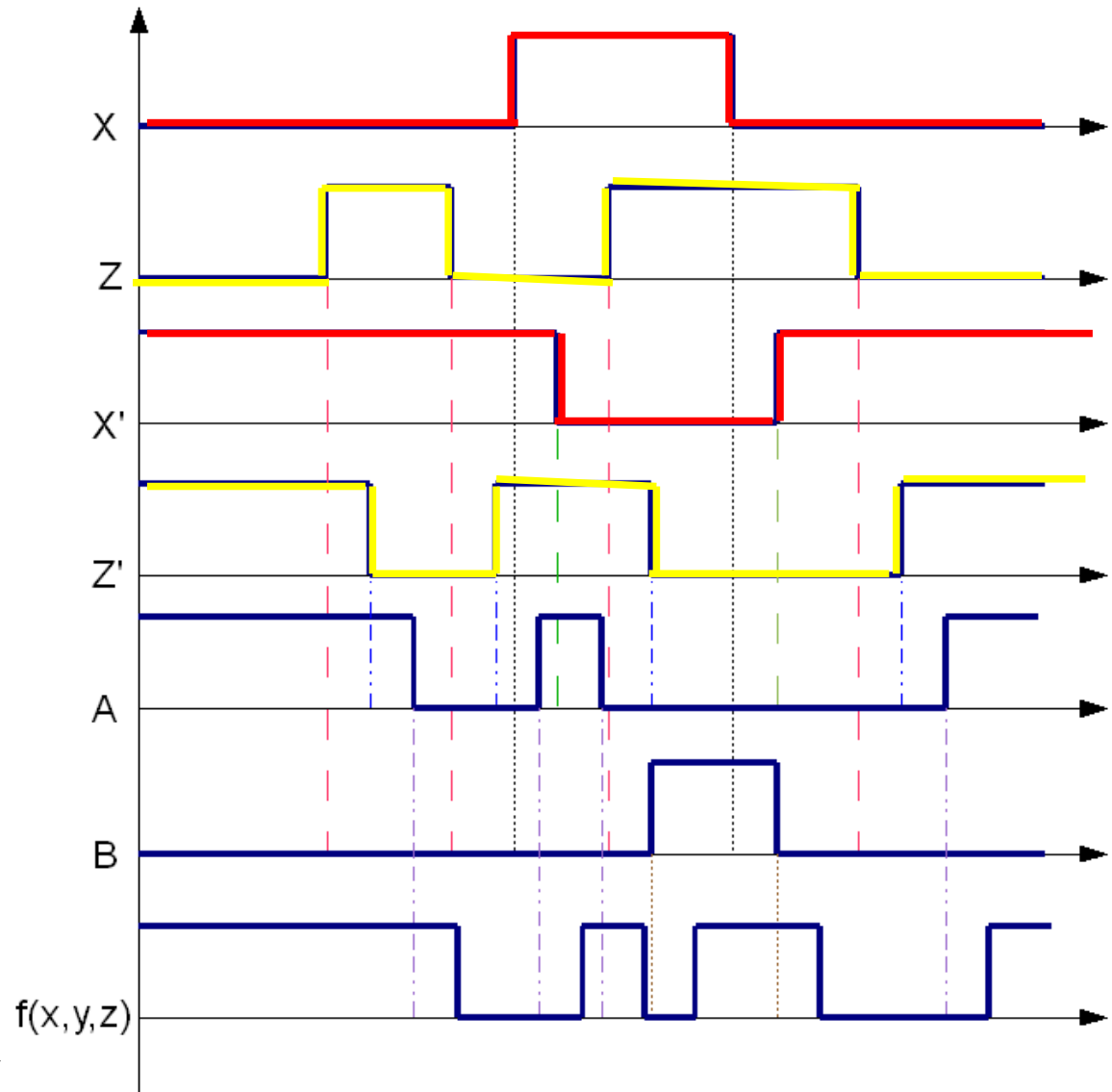
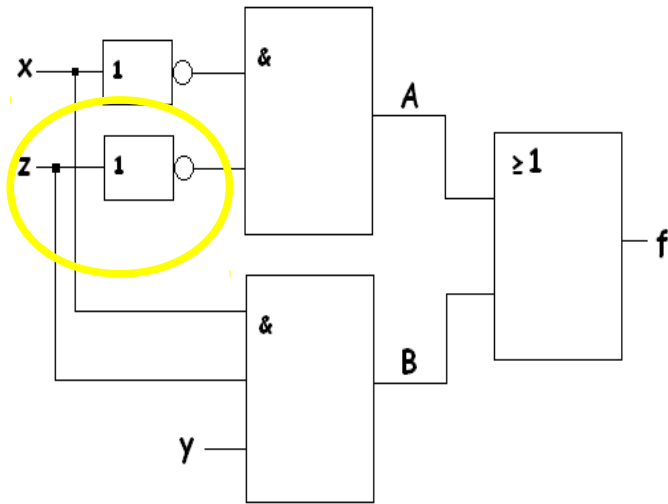
Cronograma (con retrasos igual para todas las puertas) (con $y=1$)



Análisis de Circuitos Combinacionales

Análisis Temporal

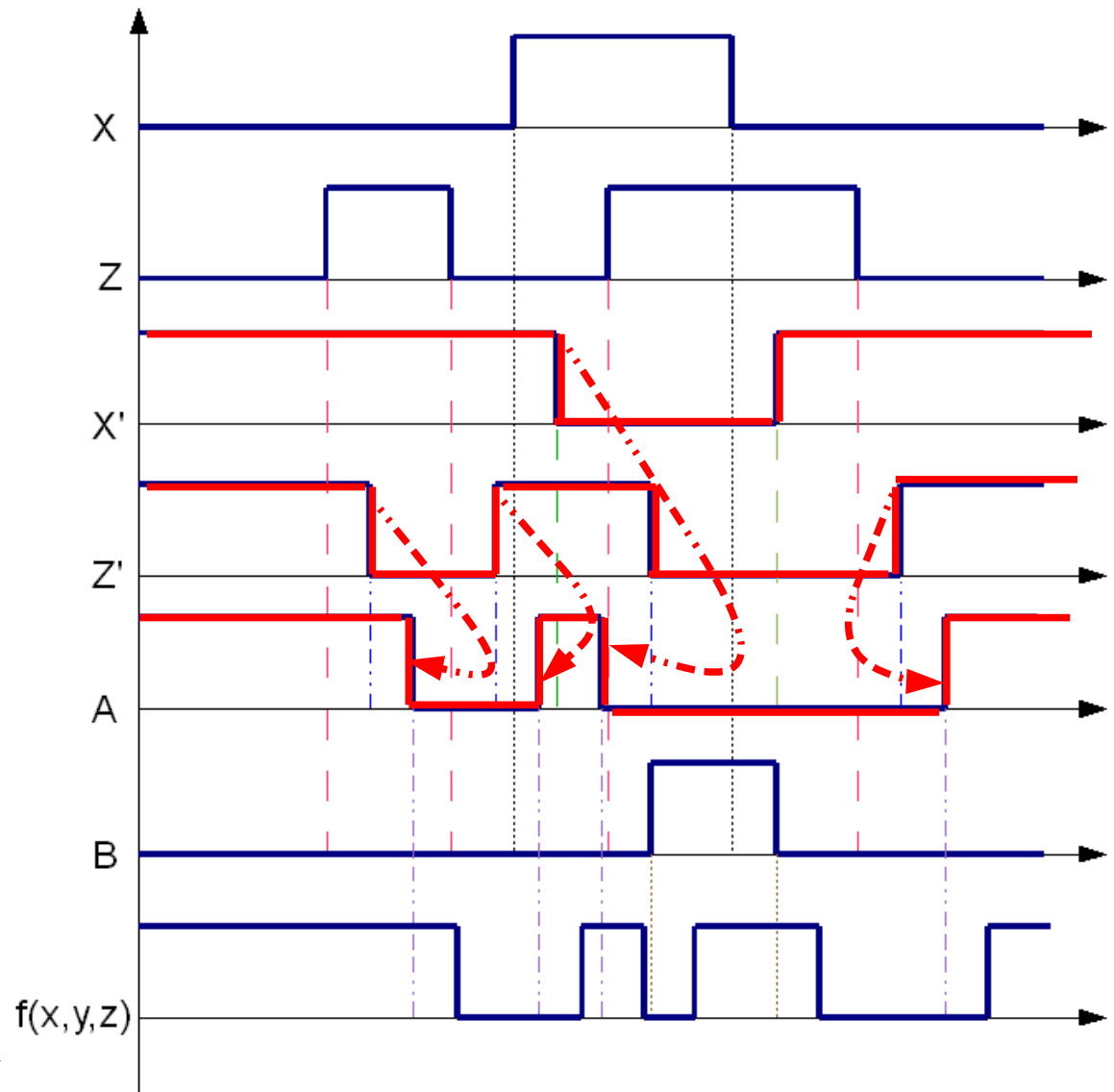
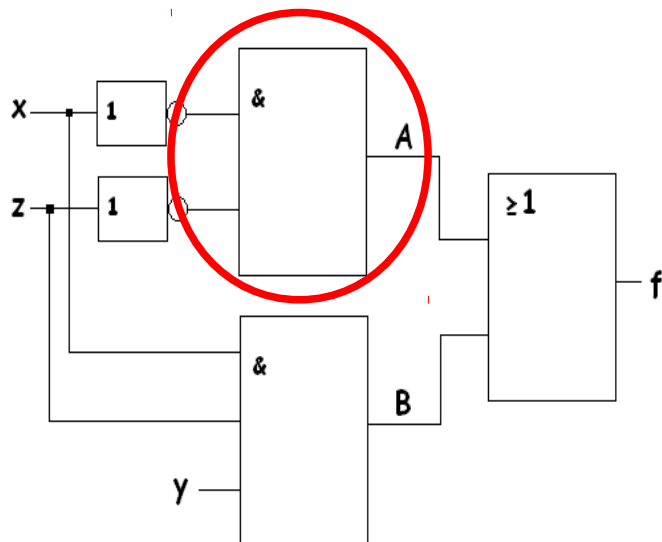
Cronograma (con retrasos igual para todas las puertas) (con $y=1$)



Análisis de Circuitos Combinacionales

Análisis Temporal

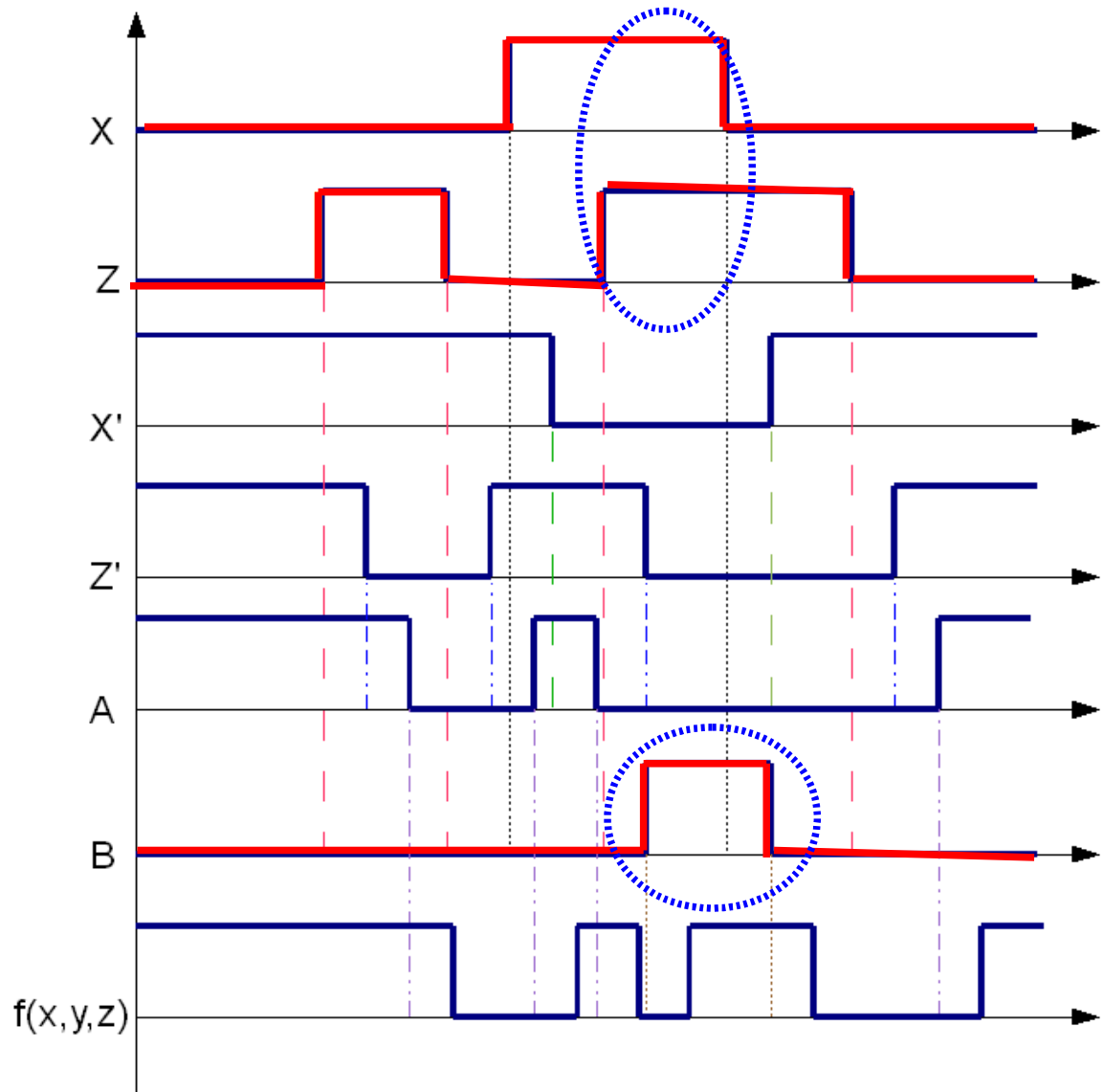
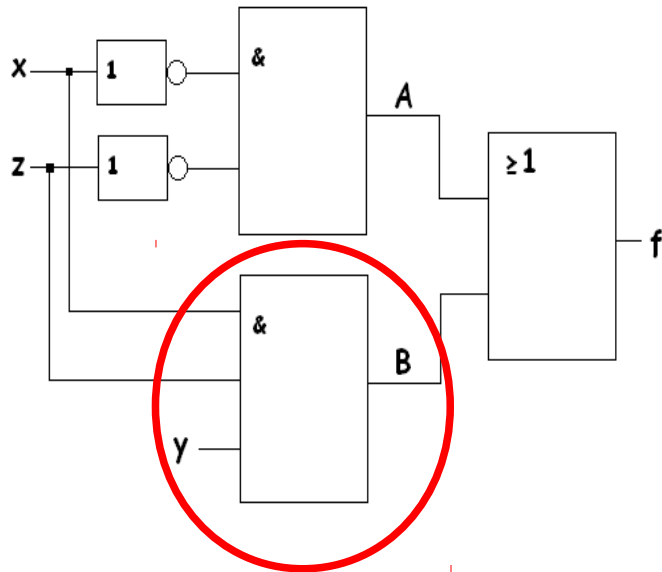
Cronograma (con $y=1$) (con retrasos igual para todas las puertas)



Análisis de Circuitos Combinacionales

Análisis Temporal

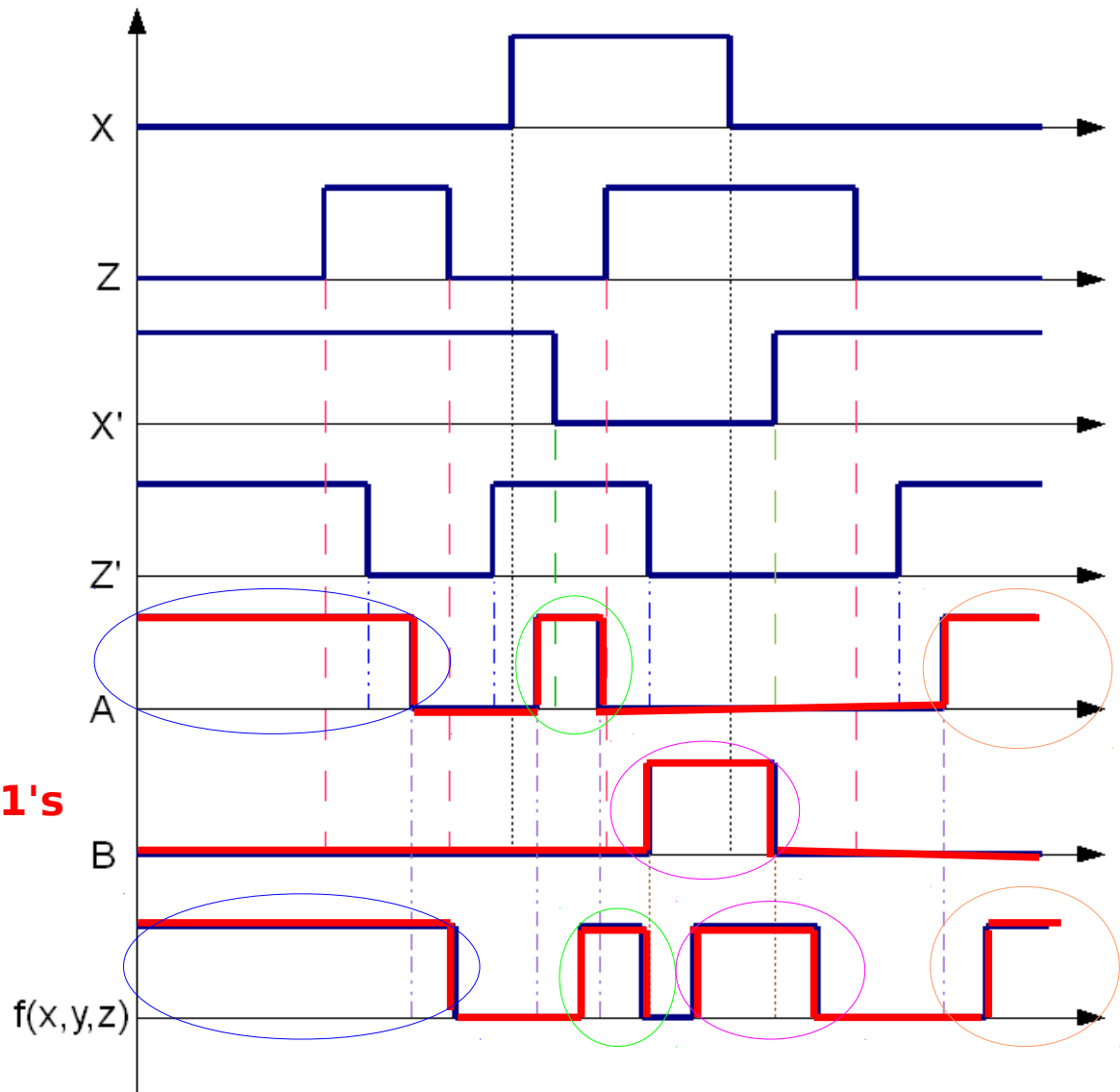
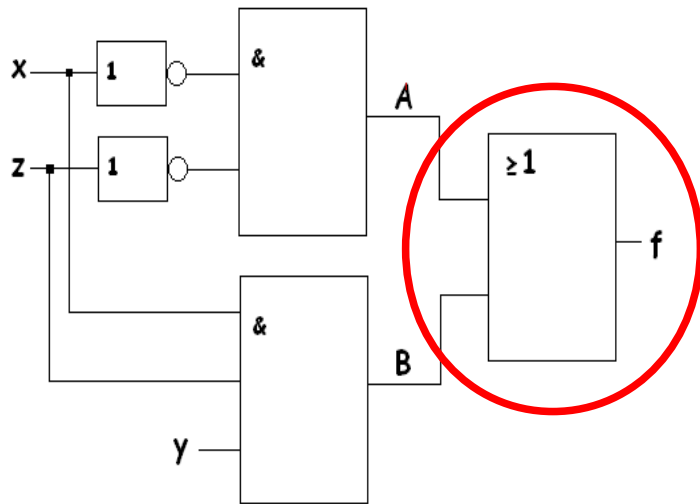
Cronograma (con retrasos igual para todas las puertas) (con $y=1$)



Análisis de Circuitos Combinacionales

Análisis Temporal

Cronograma (con retrasos igual para todas las puertas) $y=1$ (con todas las puertas)



Al ser una puerta OR todos los 1's de A y B lo serán de f, pero un tiempo después

Análisis de Circuitos Combinacionales

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.

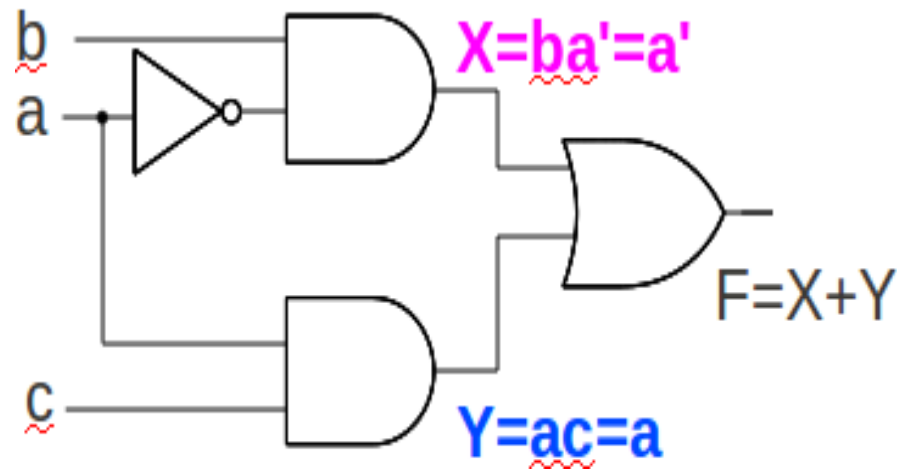
Ejemplo: Circuito con azar ($\Delta=1\text{ns}$)

$$F = a b + a'c$$

$$b=c=1,$$

$a \rightarrow$ señal periódica ($T=10\text{ns}$)

Análisis de Circuitos Combinacionales



ANÁLISIS

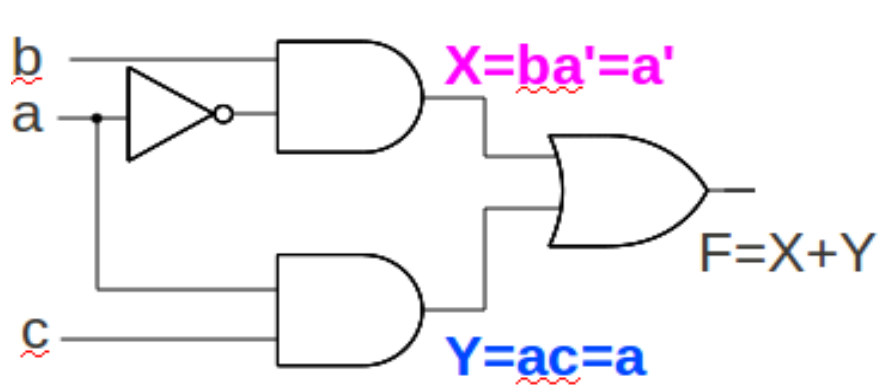
$$b = c = 1$$

a señal periódica de $f = 100 \text{ Mhz}$

$$T = 1/100000000 = 10^{-8} \text{ sg} = 10 \text{ ns}$$

Supongamos un retardo de la Puerta $\Delta = 1 \text{ ns}$

Análisis de Circuitos Combinacionales



ANÁLISIS

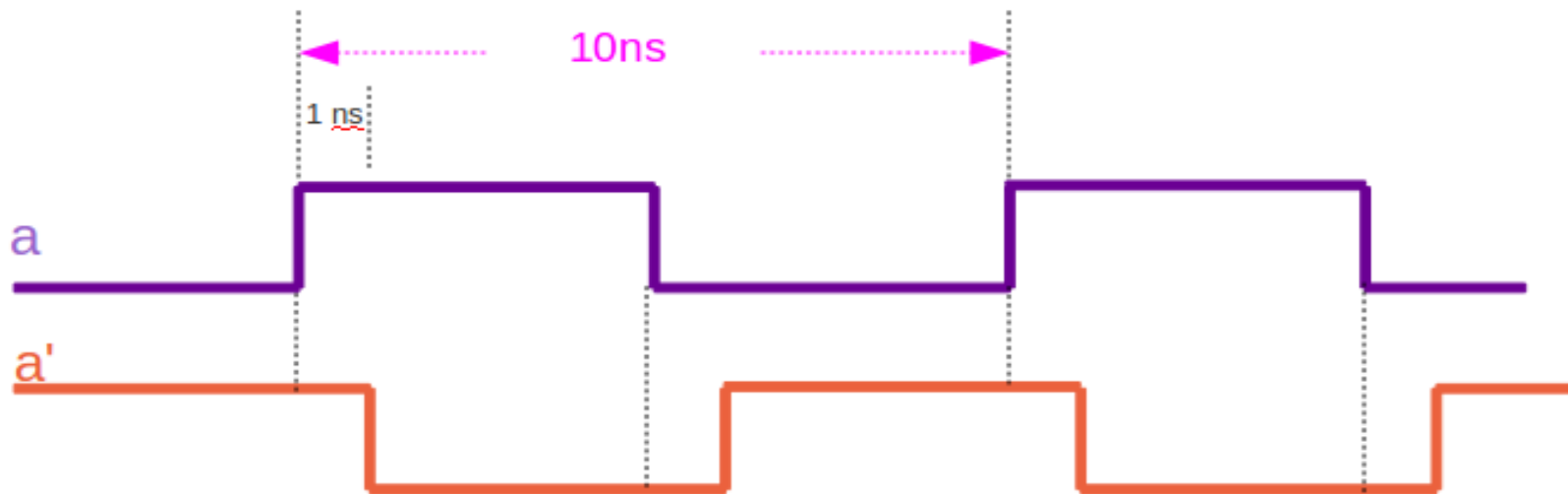
$b = c = 1$

a señal periódica de $f = 100 \text{ Mhz}$

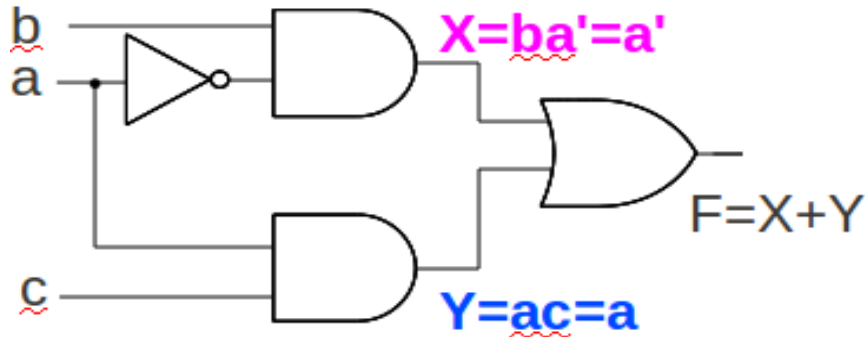
$T = 1/100000000 = 10^{-8} \text{ sg} = 10 \text{ ns}$

Supongamos un retardo de la Puerta $\Delta = 1 \text{ ns}$

Cronograma: Tensión frente a tiempo



Análisis de Circuitos Combinacionales



ANÁLISIS

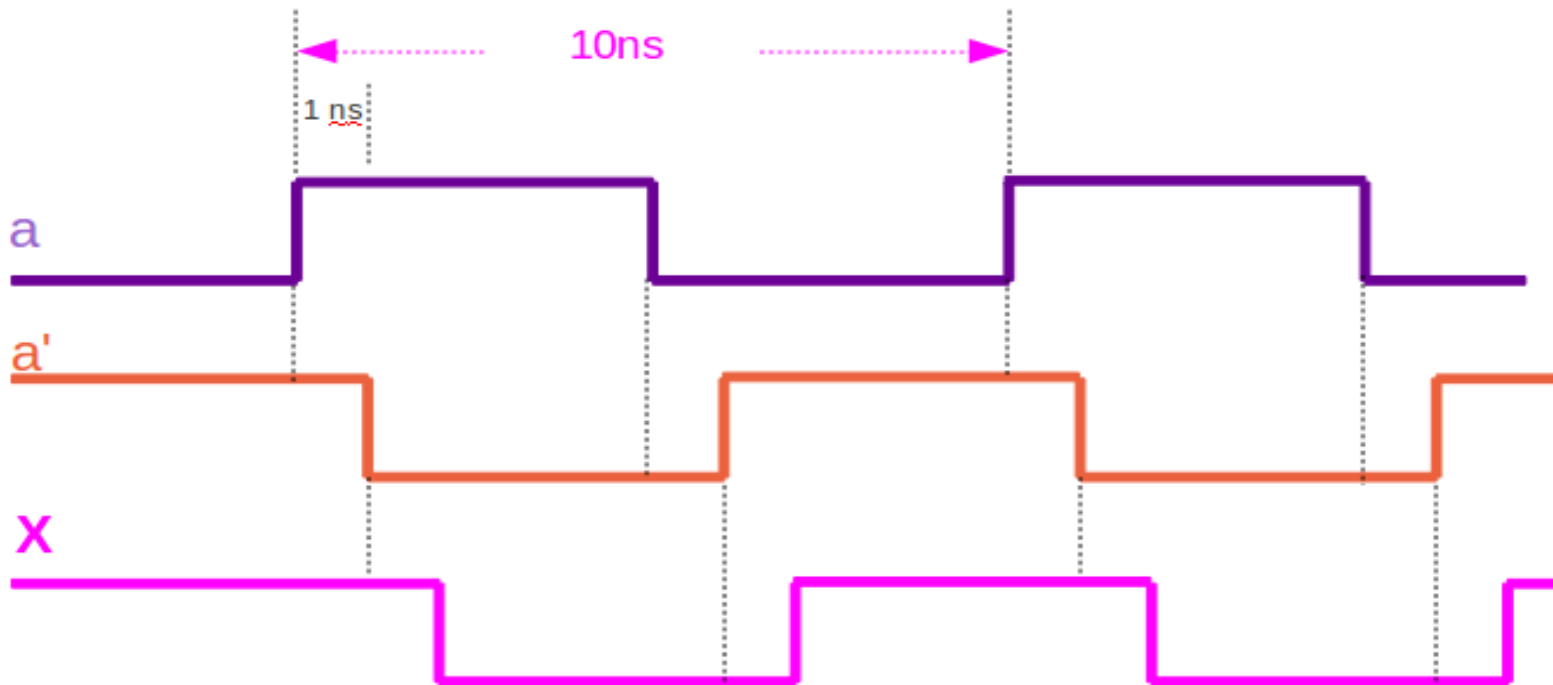
$b = c = 1$

a señal periódica de $f = 100 \text{ Mhz}$

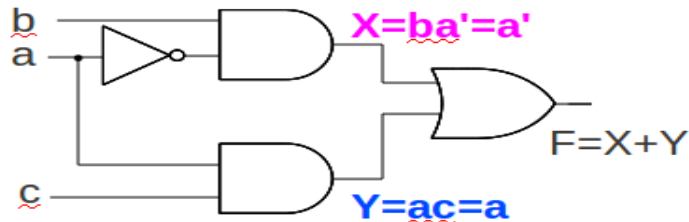
$T = 1/100000000 = 10^{-8} \text{ sg} = 10 \text{ ns}$

Supongamos un retardo de la Puerta $\Delta = 1 \text{ ns}$

Cronograma: Tensión frente a tiempo



Análisis de Circuitos Combinacionales



ANÁLISIS

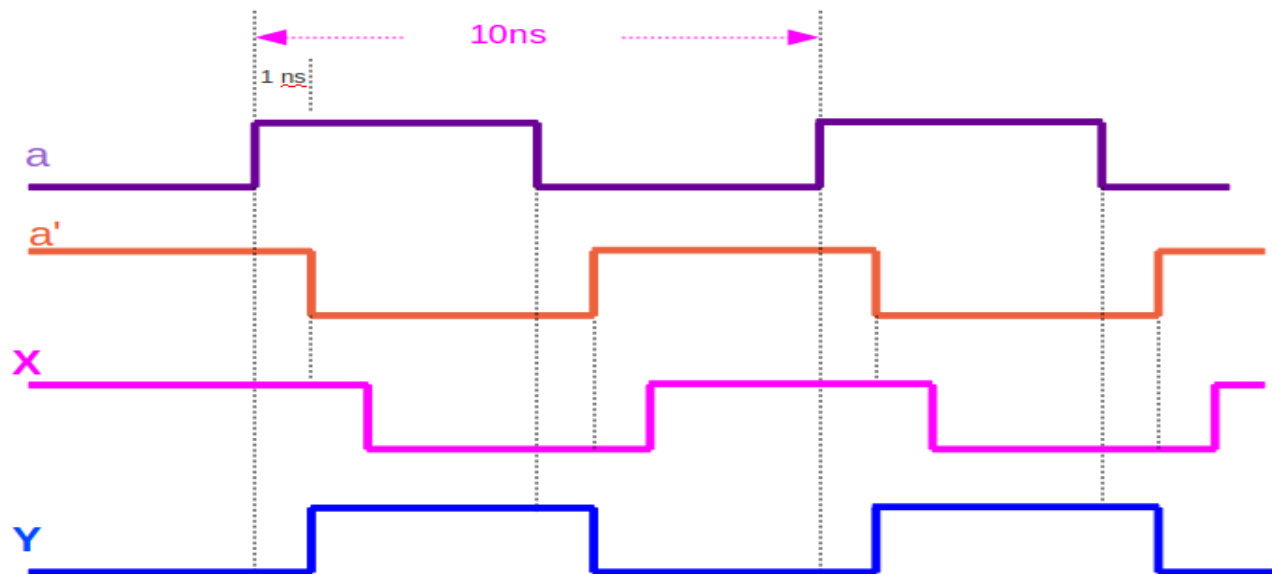
$b=c=1$

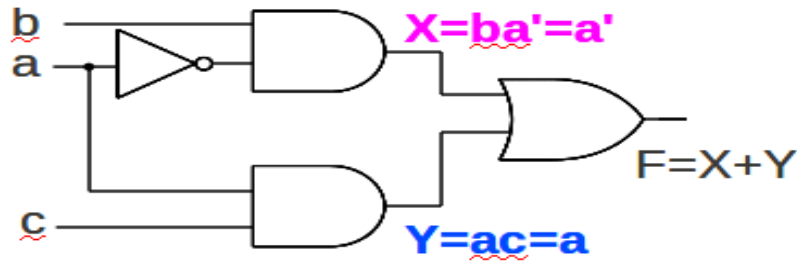
a señal periódica de $f=100$ Mhz

$T=1/100000000=10^{-8}$ sg=10ns

Supongamos un retardo de la Puerta $\Delta=1$ ns

Cronograma: Tensión frente a tiempo





ANÁLISIS

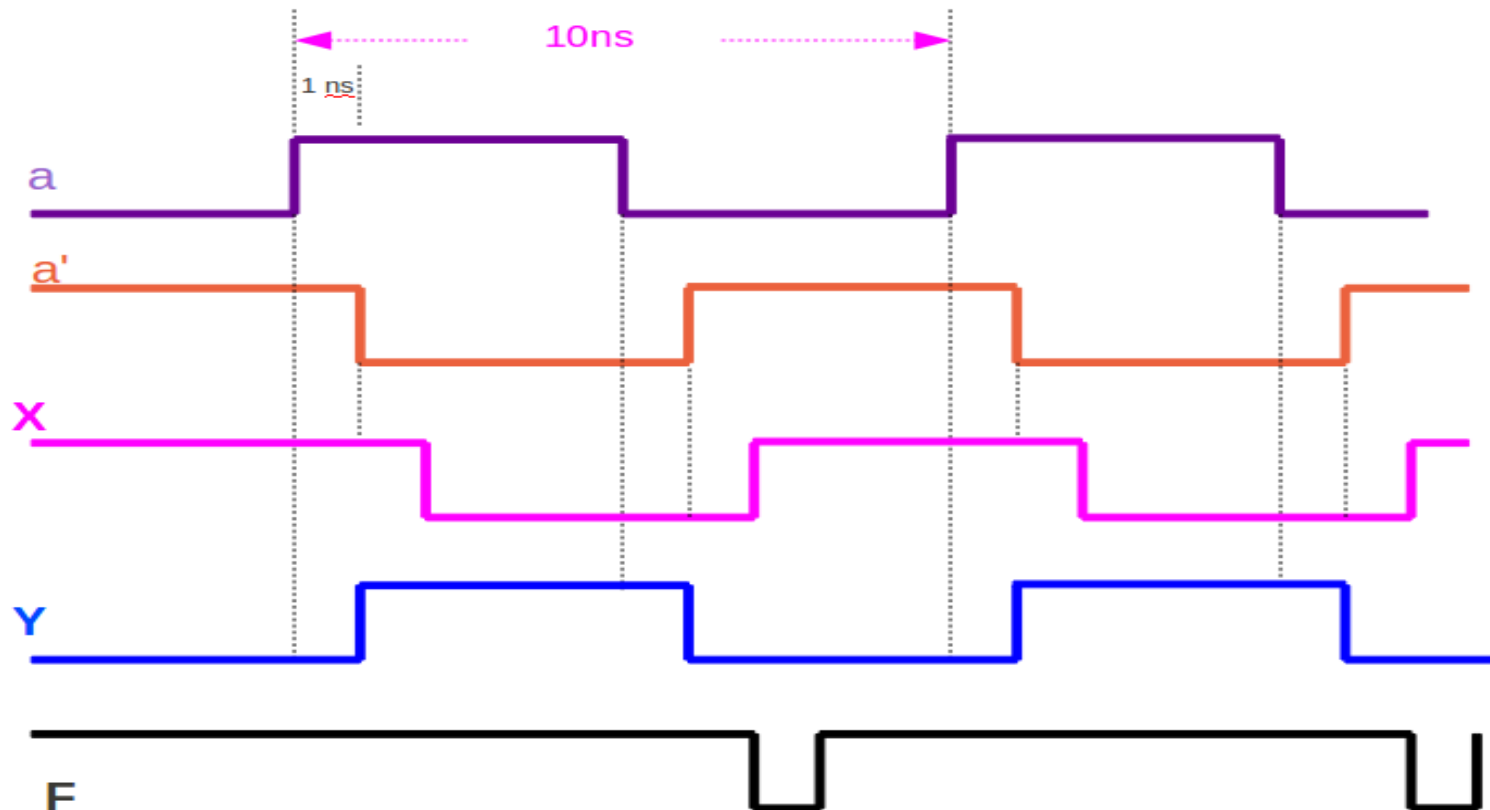
$b=c=1$

a señal periódica de $f=100 \text{ Mhz}$

$T=1/100000000=10^{-8} \text{ sg}=10\text{ns}$

Supongamos un retardo de la Puerta $\Delta=1\text{ns}$

Cronograma: Tensión frente a tiempo



Comportamiento ideal $F=a+a'=1$

Comportamiento real debido a los retardos: F en algunos intervalos de tiempo vale 0, son azares.

Tema IV - Parte IV

Diseño de Circuitos Combinacionales

Diseño de Circuitos Combinacionales

Objetivos y conceptos básicos

El diseño (o síntesis) de un circuito es el proceso inverso al análisis: partiendo de una descripción inicial de la tarea que realiza el circuito, habrá de obtener:

- la tabla de verdad,
- el K-mapa,
- la ecuación booleana,
- el circuito.

Diseño de Circuitos Combinacionales

Objetivos y conceptos básicos

El circuito debe ser óptimo, para ello se ha optado por considerar dos criterios de diseño:

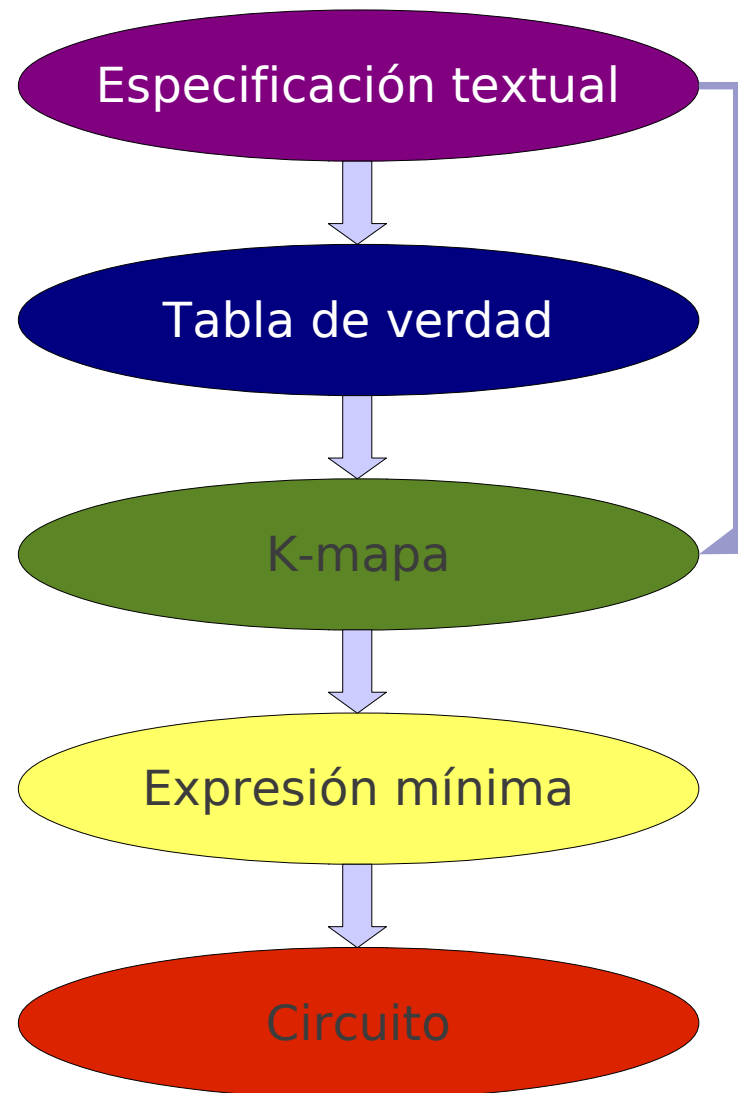
- Reducir el número de puertas
- Reducir el número de conexiones

Restricciones:

- Estructura en dos niveles (tres para simple raíl)
- Uso de puertas AND, OR, NAND y NOR
- No considerar fan-in ni fan-out como restricciones.

Diseño de Circuitos Combinacionales

Pasos del proceso



Diseño de Circuitos Combinacionales

Pasos del proceso

Paso 1: Descripción textual -> Tabla de verdad

Determinar variables de entrada y especificar 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.

Diseño de Circuitos Combinacionales

Pasos del proceso

Ejemplo1 (pasos 1 y 2):

Suponga que los números entre 0 y 15 están representados en binario con cuatro bits: $X_3X_2X_1X_0$, donde X_3 es el bit más significativo. Diseñe un circuito que de salida $Z = 1$ si y sólo si el número $X_3X_2X_1X_0$ es primo.



$X_3X_2X_1X_0$	Z
0 0 0 0	0
0 0 0 1	1
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	0

$X_3X_2X_1X_0$	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.



Entradas: tres bombillas, interruptor Salida: Alarma

(b_1, b_2, b_3, i)		a	
$b_i =$	$\begin{cases} 0 & \text{apagada} \\ 1 & \text{encendida} \end{cases}$	$i =$	$\begin{cases} 0 & \text{off} \\ 1 & \text{on} \end{cases}$
		$a =$	$\begin{cases} 0 & \text{inactiva} \\ 1 & \text{activa} \end{cases}$

i	b_1	b_2	b_3	a	i	b_1	b_2	b_3	a
0	0	0	0	0	1	0	0	0	1
0	0	0	1	1	1	0	0	1	1
0	0	1	0	1	1	0	1	0	1
0	0	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	0	0	1
0	1	0	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	0

Diseño de Circuitos Combinacionales

Diseño con K-mapa

Paso 3: Obtener la expresión mínima en dos niveles

- Nos basaremos en el método del K-mapa
- Expresión mínima como suma de productos
 - Nos fijamos en los 1's del K-mapa, o mintérminos, que son términos producto.
 - Agrupamos los mintérminos para conseguir términos productos con menor número de variables (**implicantes**).
- Expresión mínima como producto de sumas
 - Nos fijamos en los 0's del K-mapa, o maxtérminos, que son términos suma.
 - Agrupamos los maxtérminos para conseguir términos sumas con menor número de variables (**implicadas**).

Diseño de Circuitos Combinacionales

Diseño con K-mapa

Paso 3: Obtener la expresión mínima en dos niveles (cont)

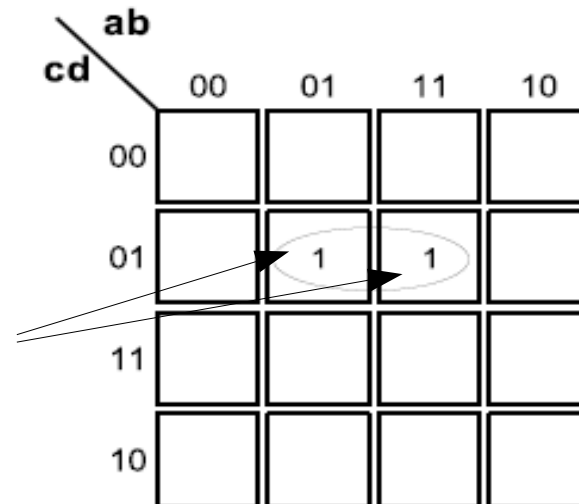
El agrupamiento de **1's (0's)** para construir términos productos (o *términos suma*) con menor número de variables es posible gracias:

1) La adyacencia entre las celdas de un K-mapa (sólo cambian un bit, por efecto del código Gray)

2) Si un término producto se expresa como **p q** (o un término suma como **p+q**), otro adyacente a él, que varíe un bit, sería **p q'** (o **p+q'** para el término suma), por tanto la suma de los dos: $pq + pq' = p(q + q') = p$ - o el producto de ambos $(p+q)(p+q') = p + q q' = p$ -. Es decir, se elimina la variable que aparece complementada y sin complementar en ambos términos.

Ejemplo: $f = a' b c' d + a b c' d =$
 $b c' d (a' + a) = b c' d$

Mintérminos
adyacentes



Diseño de Circuitos Combinacionales

Diseño con K-mapa

Implicante

Es un 1 o grupo de 1's representado en el K-mapa. Los grupos deben estar formado por una potencia de 2 de 1's, y estos deben ser vecinos.

Los grupos se van formando a partir de grupos de tamaño inmediatamente inferior. Por ejemplo, agrupamos dos 1's vecinos para formar un grupo de dos 1's. Luego, este grupo podemos agruparlo con otro vecino formado por otros dos 1's, para obtener un grupo de 4.

El número de 1's del grupo determina el orden del implicante.

El orden del implicante está relacionado con el número de variables que posee la expresión del término producto que lo representa.

Diseño de Circuitos Combinacionales

Diseño con K-mapa

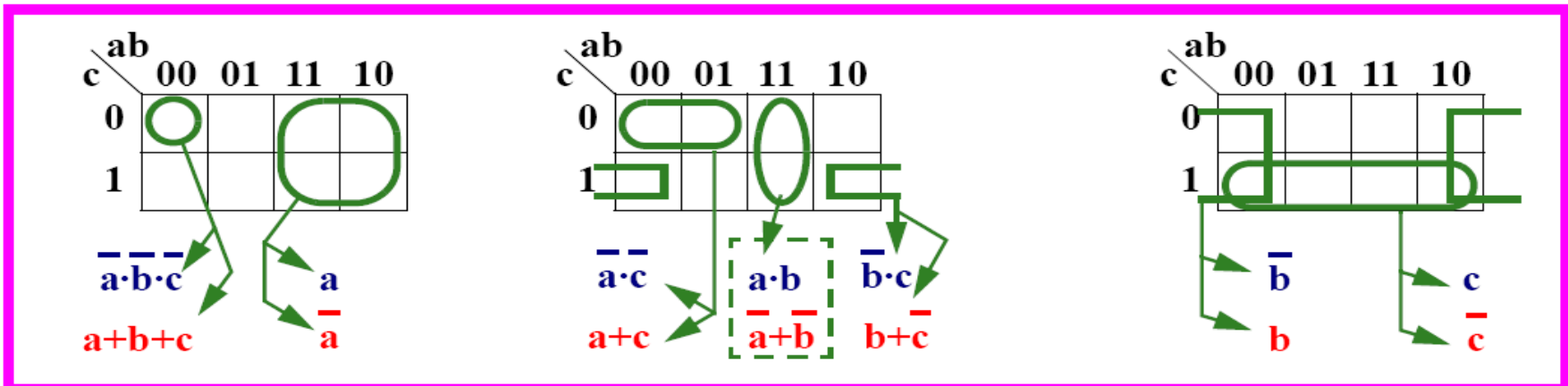
Implicante

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		

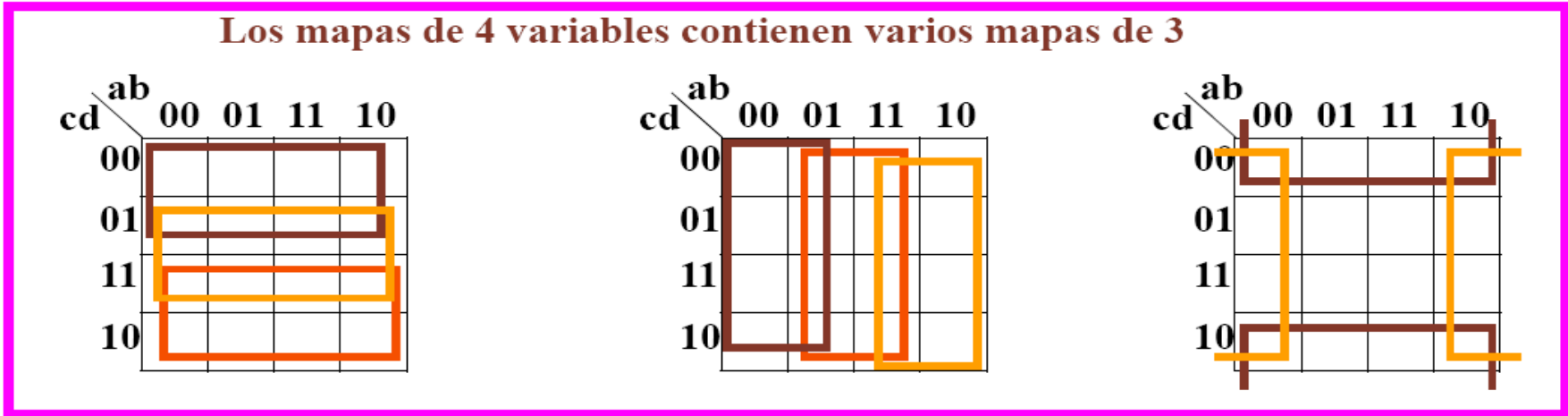
Diseño de Circuitos Combinacionales

Diseño con K-mapa

Agrupaciones posibles



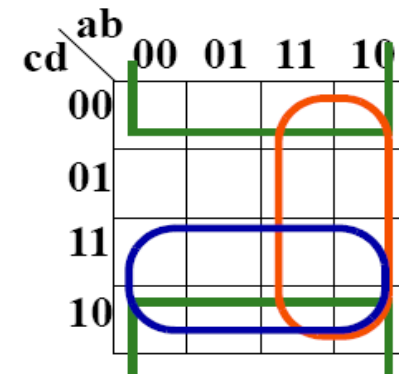
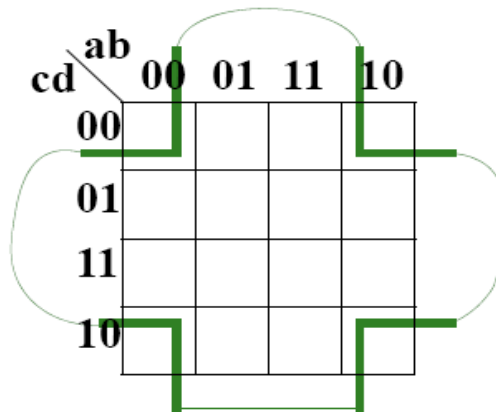
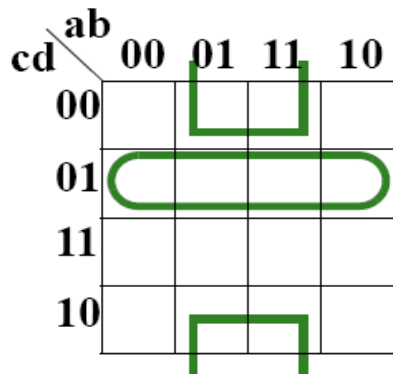
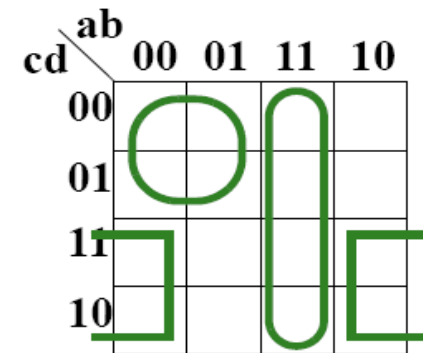
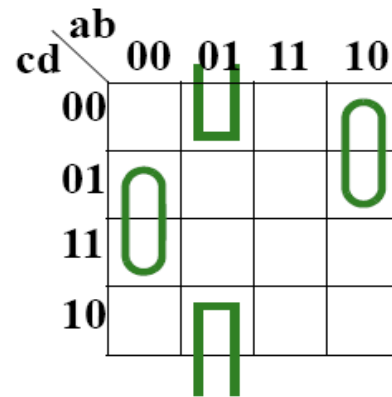
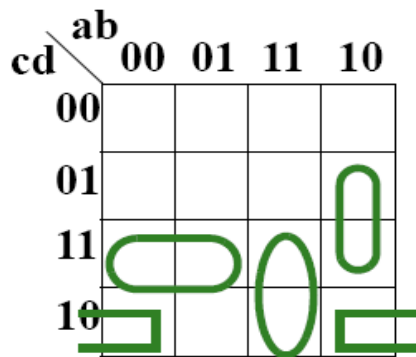
Los mapas de 4 variables contienen varios mapas de 3



Diseño de Circuitos Combinacionales

Diseño con K-mapa

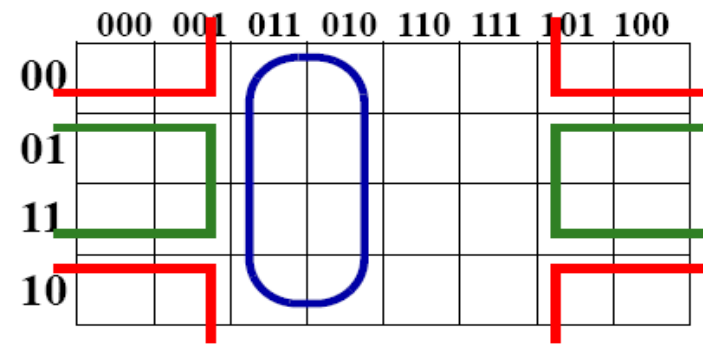
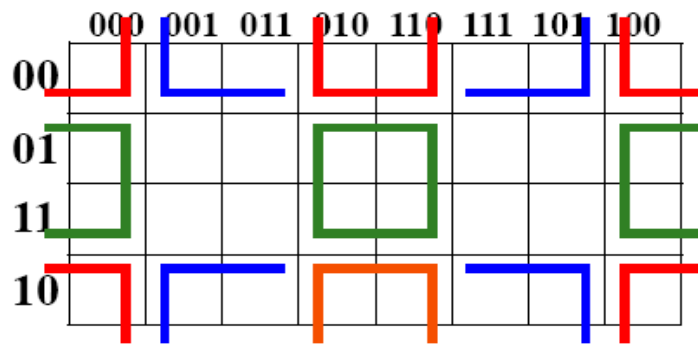
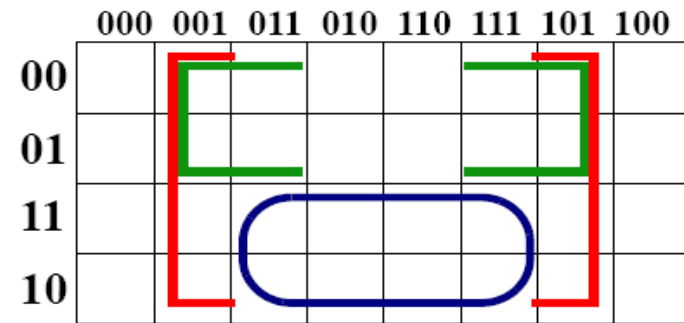
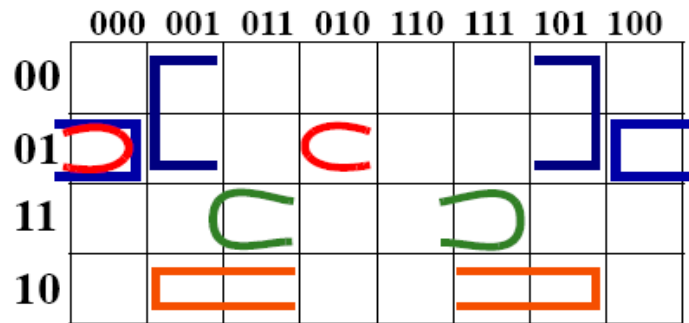
Agrupaciones posibles



Diseño de Circuitos Combinacionales

Diseño con K-mapa

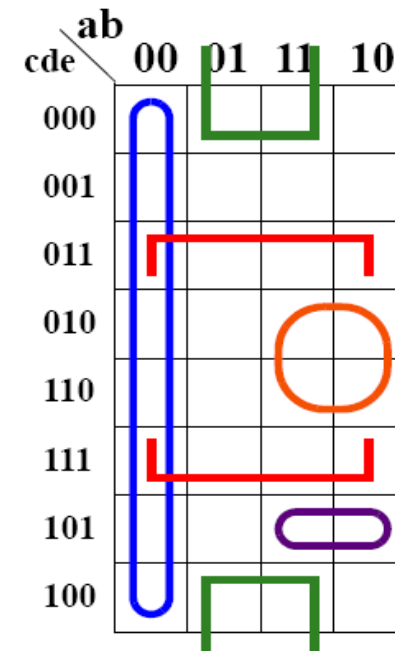
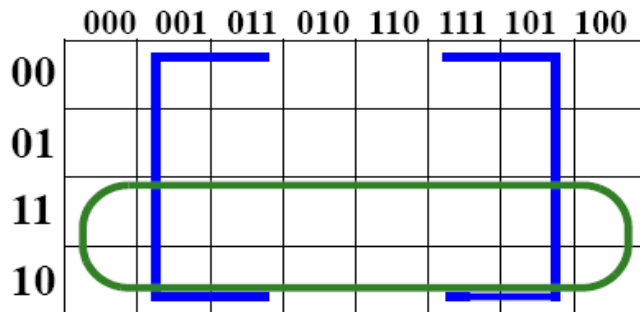
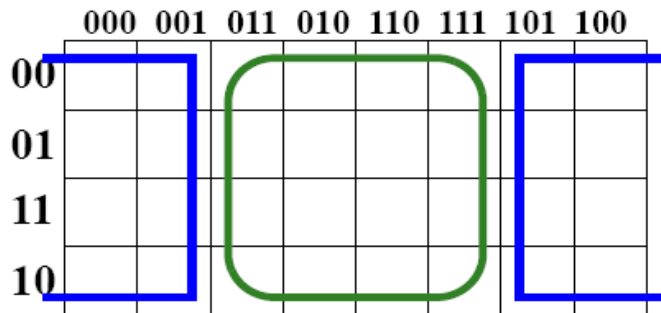
Agrupaciones posibles



Diseño de Circuitos Combinacionales

Diseño con K-mapa

Agrupaciones posibles



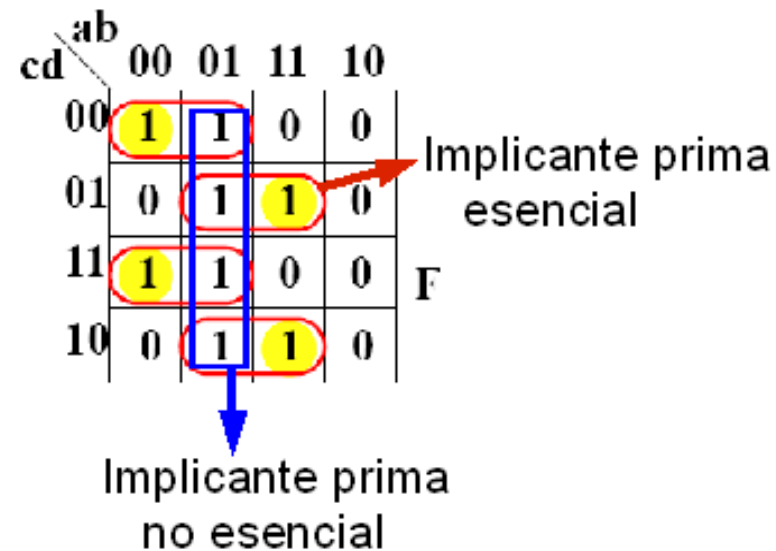
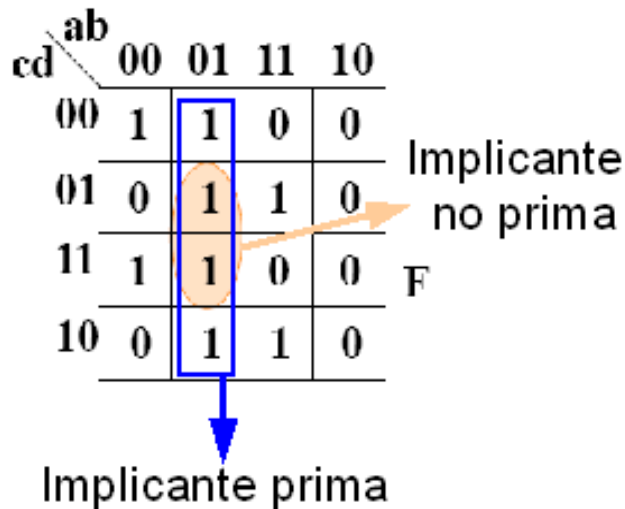
Diseño de Circuitos Combinacionales

Diseño con K-mapa

Definiciones

Una Implicante se dice que es **prima** si no está cubierta por ninguna otra implicante de la función.

Una implicante prima se dice que es **esencial** si cubre algún mintermino no incluido en ninguna otra implicante prima. Al mintermino se le denomina **distinguido**.



Diseño de Circuitos Combinacionales

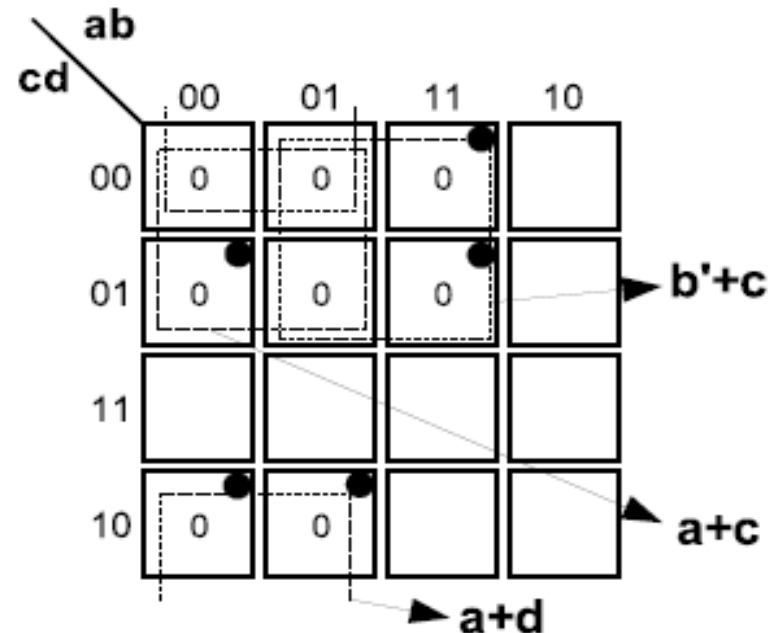
Diseño con K-mapa

Definiciones

Una Implicada (o término suma) se dice que es **prima** si no está cubierta por ninguna otra implicada de la función.

Una implicada prima se dice que es **esencial** si cubre algún maxtérmino no incluido en ninguna otra implicada prima. Al maxtérmino se le denomina **distinguido**.

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



Diseño de Circuitos Combinacionales

Diseño con K-mapa

Funciones incompletamente especificadas

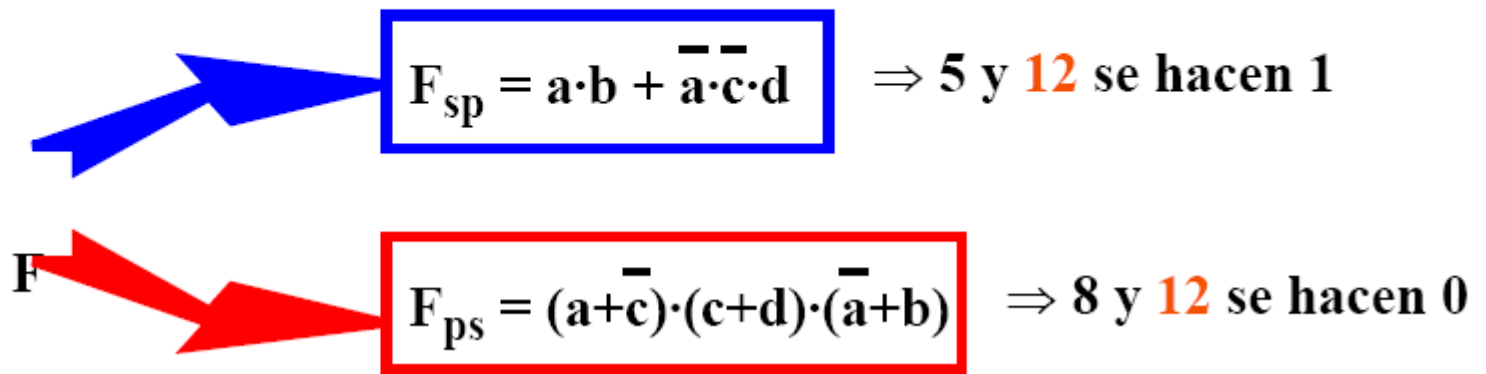
Las casillas con inespecificación se usan como mejor nos convenga:

Se pueden incluir para formar grupos mayores.

No es necesario cubrirlas todas.

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

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



Diseño de Circuitos Combinacionales

Diseño con K-mapa

Expresión mínima en s.p.

La suma mínima se obtiene usando el menor número de implicantes primas obtenidas del K-mapa y que permitan cubrir todos los minterminos del mismo.

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 minterminos sin cubrir, procederemos uno por uno, a analizar cuáles son las implicantes primas que permiten su cubrimiento y como regla general se escogerá aquella que, a igualdad de número de literales, tiene un cubrimiento adicional de minterminos mayor.
- 3) Repetir el punto 2 hasta que se cubra todo el K-mapa

Consideraciones finales:

- a) Las inespecificaciones no se cubren
- b) Si no se pudiese aplicar los puntos 1 y 2, se deberán tomar suposiciones de cubrimiento y evaluar, al final, cuál de todas ellas se traduce en un menor coste.

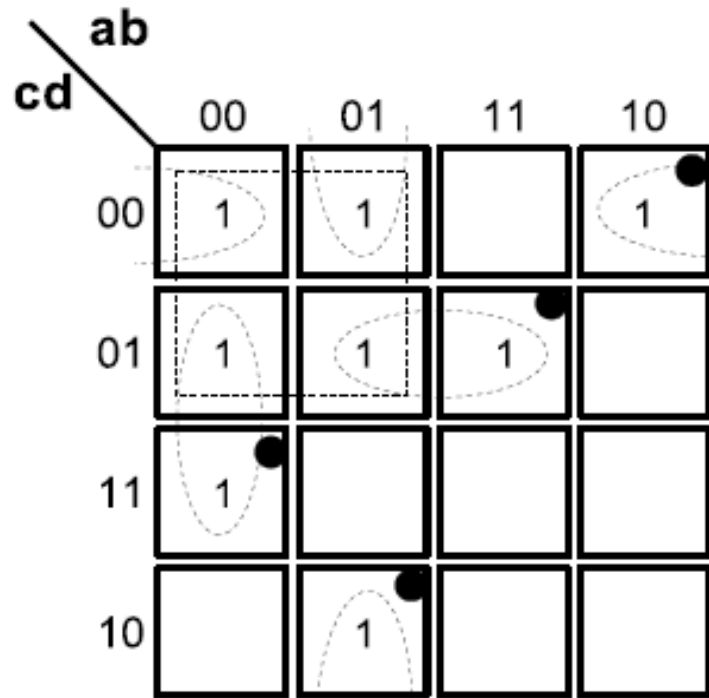
Expresión mínima en p.s.

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

Diseño de Circuitos Combinacionales

Ejemplos de obtención de la expresión mínima

Ejercicio 1.- $f = \Sigma(0,1,3,4,5,6,8,13)$



Buscamos las IP esenciales

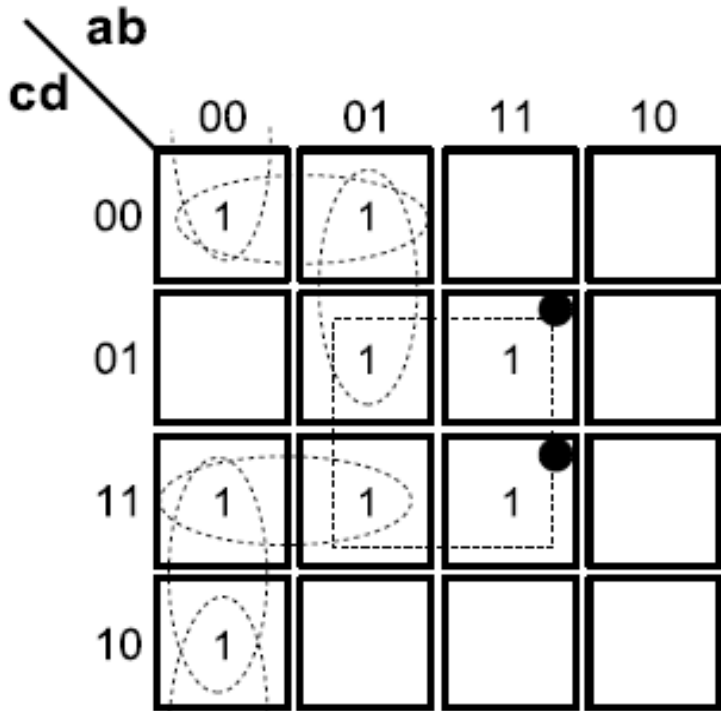
$$f = a'b'd + b'c'd' + a'bd' + bc'd$$

Fin. Se han cubierto todos los mintérminos sólo con las IP esenciales

Diseño de Circuitos Combinacionales

Ejemplos de obtención de la expresión mínima

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



1) Buscamos las Ip esenciales : **bd** con la que no se cubren todos los minterminos

2) Nos fijamos en el mintermino 4.

Está

cubierto por las Ip **a'c'd'** y **a'bd**.

Ambas del

mismo coste pero la primera cubre también

al mintermino 0 que no estaba cubierto por

bd. Escogemos **a'c'd'**

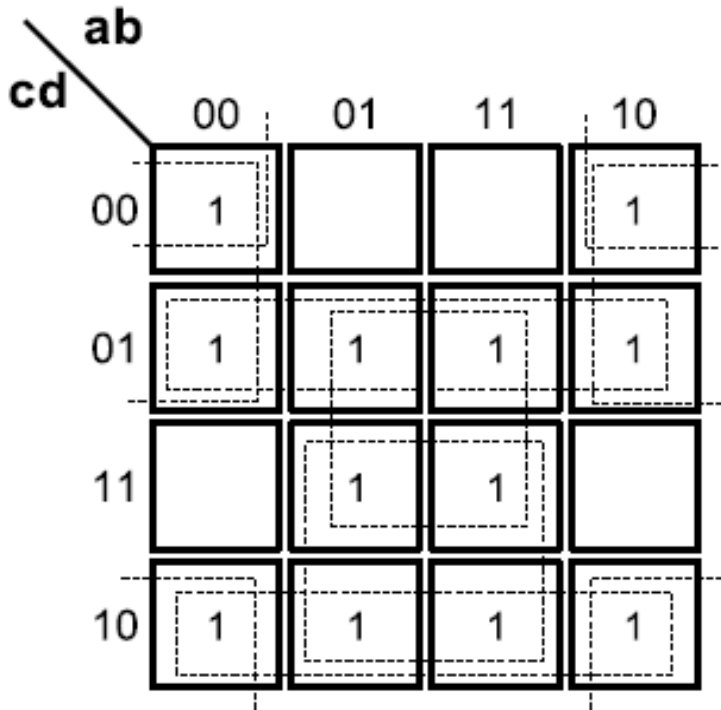
3) Ahora sólo quedan por cubrir los minterminos 2 y 3. Obviamente

Fin. Se han cubierto todos los minterminos escogemos **a'b'c**

Diseño de Circuitos Combinacionales

Ejemplos de obtención de la expresión mínima

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



1) Buscamos las I_p esenciales : ¡NO HAY!

2) Nos fijamos en el mintermino 0.

Está cubierto por las I_p $b'd'$ y $'b'c'$, ambas del mismo coste.

Suponemos que la expresión mínima está formada por $b'c'$.

$$f = b'c' + bd + cd'$$

B) Repetición del proceso en el caso en

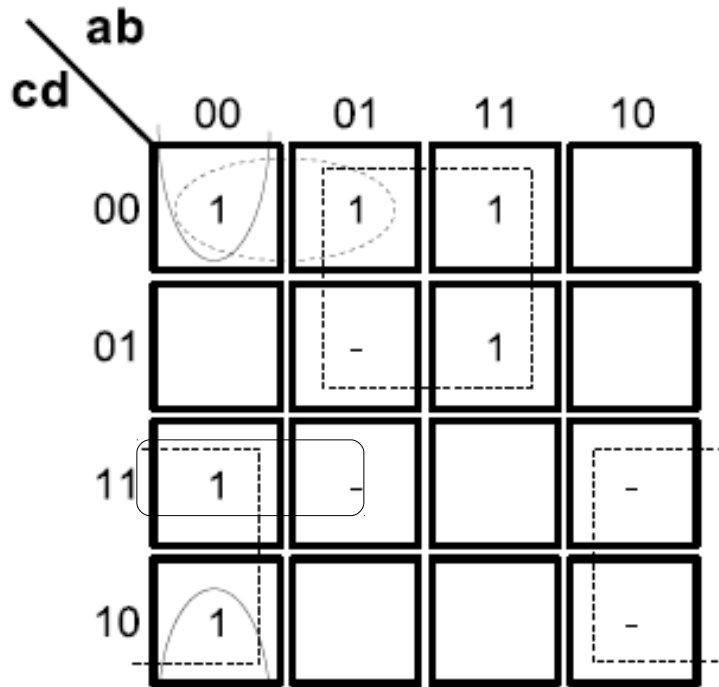
que el mintermino 0 hubiera sido cubierto por $b'd'$.

Fin. Se han cubierto todos los minterminos y ambas expresiones tienen el mismo coste. Cualquiera de las dos representa la solución mínima.

Diseño de Circuitos Combinacionales

Ejemplos de obtención de la expresión mínima

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



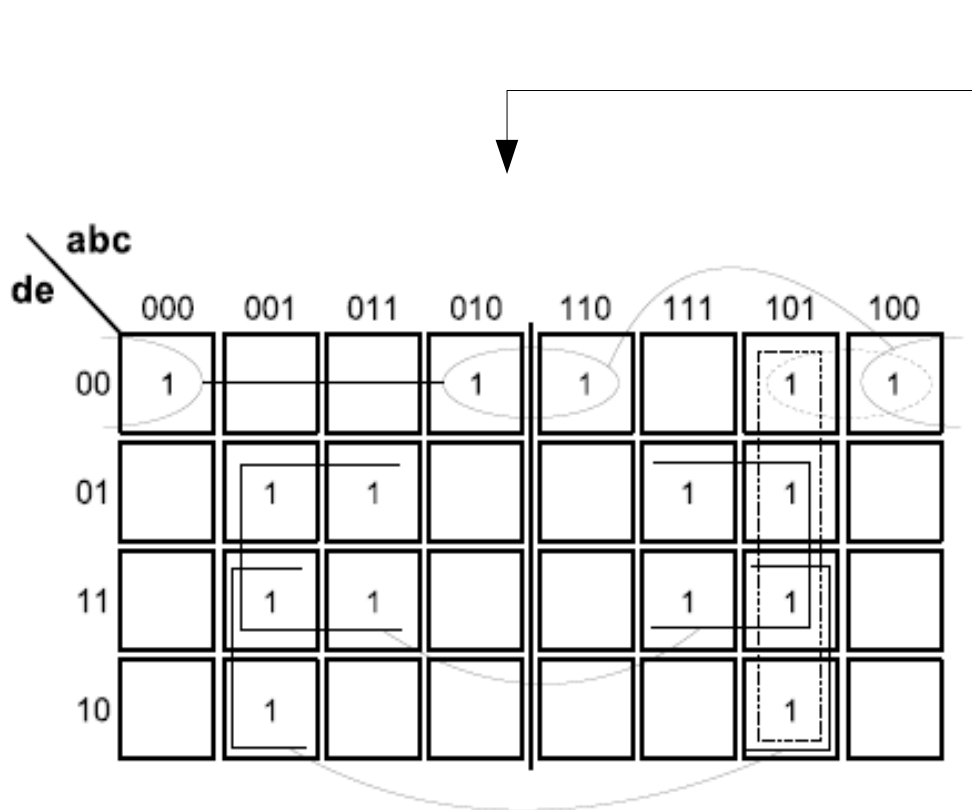
- 1) Buscamos las Ip esenciales usando inespecificaciones como mintérminos y rechazando aquellas formadas sólo por inespecificaciones: bc'
 - 2) Nos fijamos en el mintérmino 0. Está cubierto por las implicantes $a'b'd'$ y $a'c'd'$. La mejor opción es $a'b'd'$
 - 3) Sólo queda por cubrir el mintérmino 3 cuyas implicantes posibles son $a'cd$ y $b'c$. La mejor opción es $b'c$
 - 4) No se busca cubrimiento de inespecificaciones
- $$f = bc' + a'b'd' + b'c$$

Fin. Se han cubierto todos los mintérminos

Diseño de Circuitos Combinacionales

Ejemplos de obtención de la expresión mínima

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



1) La obtención de las implicantes primas en un K-mapa de 5 variables requiere analizar las simetrías entre el sub K-mapa para cuando la variable más significativa (en este ejemplo es **a**) vale 0 y cuando vale 1.

2) Siga los pasos presentados en las transparencias anteriores.

$$f = c'd'e' + ce + b'cd + ab'c$$