Circuitos Electrónicos Digitales

Universidad de Sevilla

Tema IV

Circuitos Combinacionales

Tema IV

Circuitos Combinacionales

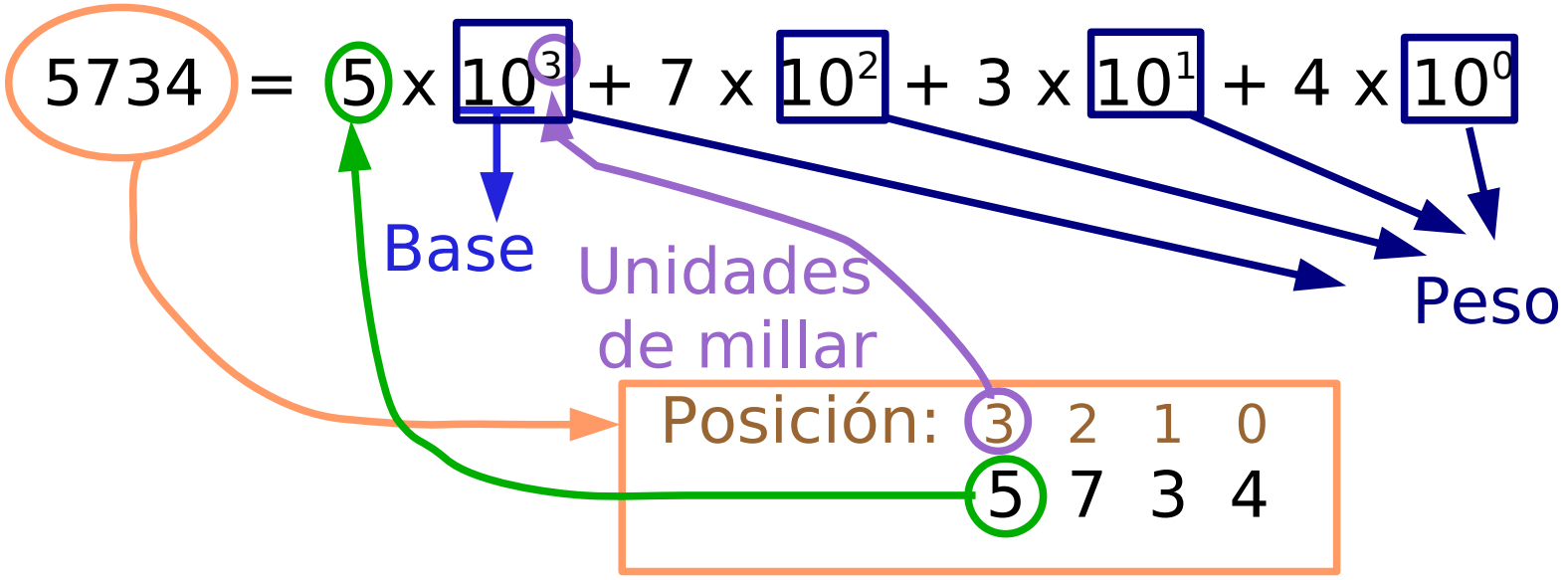
Indice

1. Representación binaria:
    - Representación posicional de magnitudes
    - Códigos binarios
2. Funciones combinacionales
3. Análisis de circuitos combinacionales
4. Diseño de circuitos combinacionales

Tema IV – Parte I

Representación Binaria

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

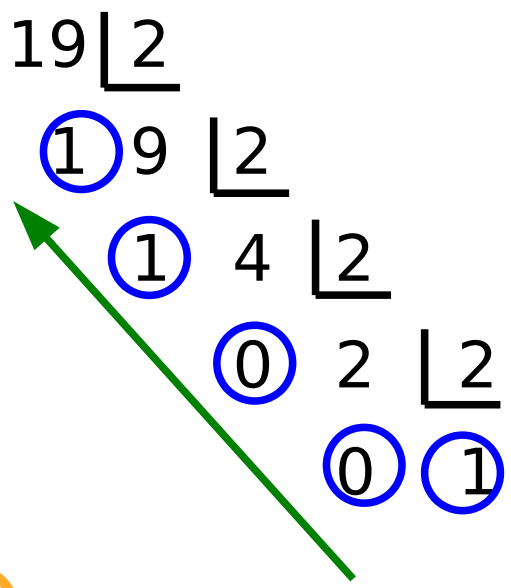
Transformaciones entre bases

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:

$16 = 2^4$

$010011_{(2} = 0001\ 0011_{(2} = 13_{(16}$

Base 16 a Base 2:

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

Base 2 a Base 8:

$8 = 2^3$

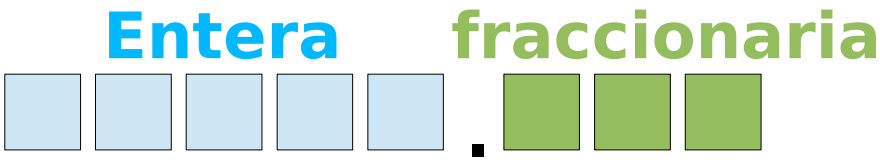
$010011_{(2} = 010\ 011_{(2} = 2\ 3_{(8}$

Base 8 a Base 2:

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

Representación Posicional de Magnitudes

Representación parte fraccionaria

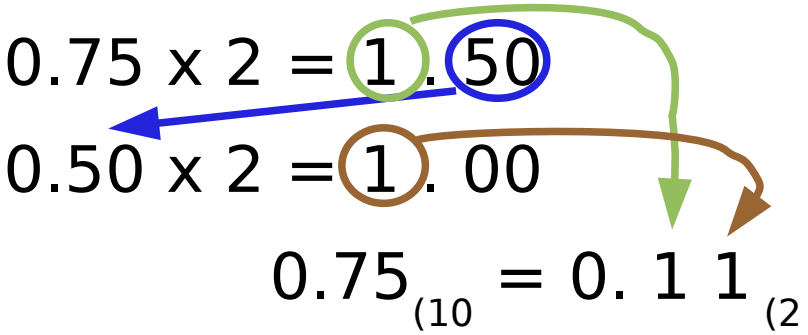


Ejemplo: 0 1 0 1 1 . 1 1 0

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:



$$0.75_{(10} = 0.\,1\,1_{(2}$$

Códigos Binarios

BCD

7 Segmentos

Gray

Detectores de Errores

Códigos alfanuméricos: ASCII, ASCII extendido

Códigos Binarios

BCD

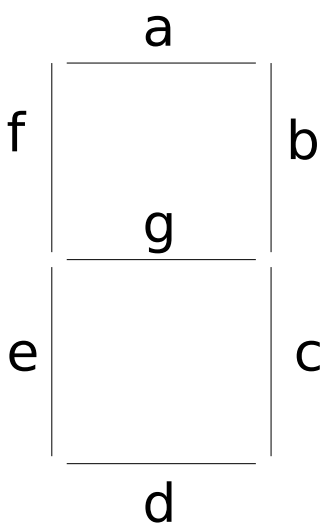
Dígito	Código BCD
0	0000
1	0001
2	0010
3	0011
4	0100

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

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

Códigos Binarios

7 Segmentos



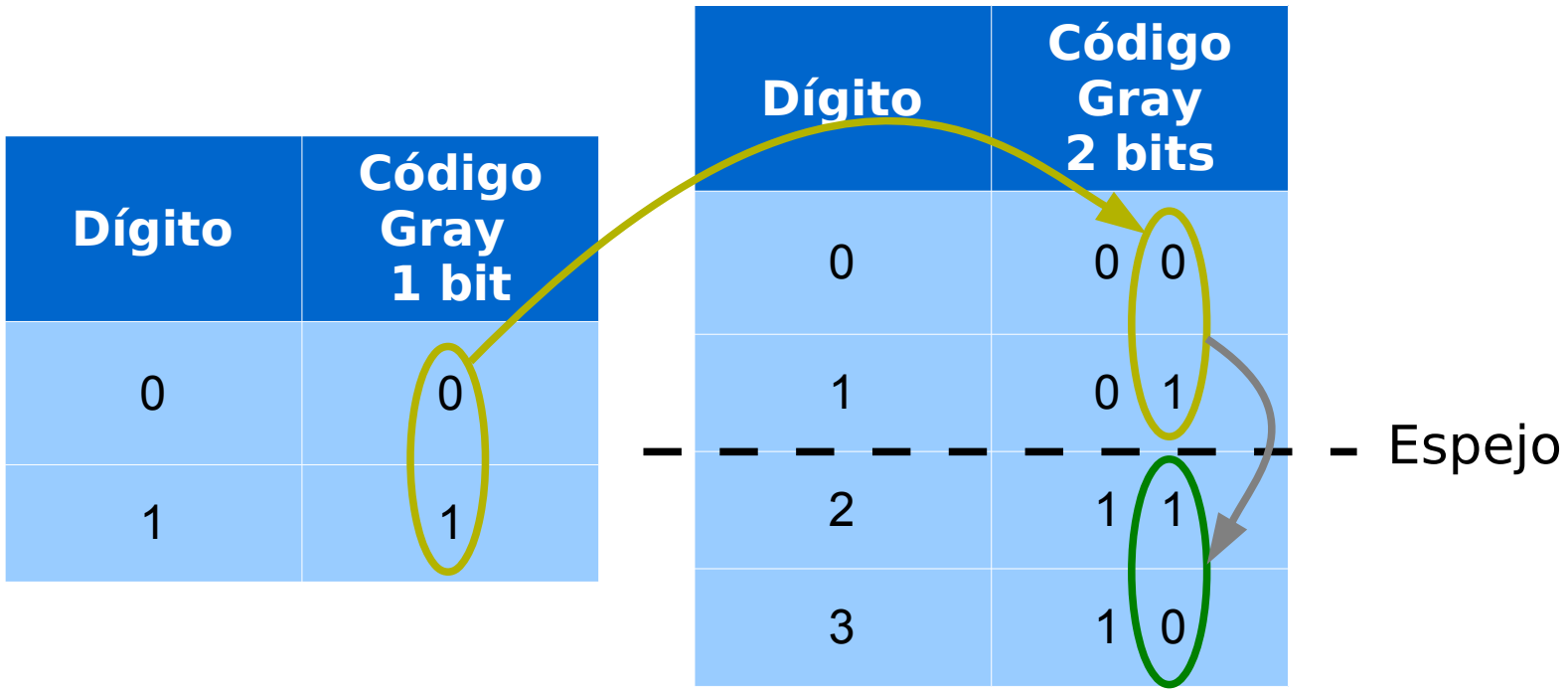
Dígito	Código 7-Seg abcdefg
0	1111110
1	0110000
2	1101101
3	1111001
4	0110011

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

Ejemplo: $16_{(10}$ = 0110000 0011111 $_{(7\text{-seg}}$

Códigos Binarios

Gray



Ejercicio: Construir el código Gray de 3 bits

Códigos Binarios

Detectores de errores

- Bit de Paridad: Se añade un bit denominado **bit de paridad)** al código binario. Puede hacerse de dos formas:

  1. Paridad Par: El número total de 1 debe ser par.
  2. Paridad impar: El número total de 1 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

- Códigos alfanuméricos:

  - Codifican números, letras, signos de puntuación y otros caracteres

    - Código **ASCII**(American Standard Code for Information Interchange)
      - es un código de 7 bits
      - permite codificar 128 caracteres
    - Código **ASCII extendido**
      - es un código de 8 bits
      - permite hasta 256 caracteres distintos

  - La tabla de códigos se puede representar en binario, decimal, octal o hexadecimal

Códigos Binarios

ASCII-bin

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

Códigos Binarios

ASCII-oct/hex/dec

Oct	Hex	Dec	Carácter		Oct	Hex	Dec	Carácter	Oct	Hex	Dec	Carácter	Oct	Hex	Dec	Carácter
0	00	0	NUL	NULl	40	20	32		100	40	64	@	140	60	96	`
1	01	1	SOH	Start Of Heading	41	21	33	!	101	41	65	A	141	61	97	a
2	02	2	STX	Start of TeXt	42	22	34	"	102	42	66	B	142	62	98	b
3	03	3	ETX	End of TeXt	43	23	35	#	103	43	67	C	143	63	99	c
4	04	4	EOT	End of Transmission	44	24	36	$	104	44	68	D	144	64	100	d
5	05	5	ENQ	ENQuiry	45	25	37	%	105	45	69	E	145	65	101	e
6	06	6	ACK	ACKnowledge	46	26	38	&	106	46	70	F	146	66	102	f
7	07	7	BEL	BELl	47	27	39	'	107	47	71	G	147	67	103	g
10	08	8	BS	BackSpace	50	28	40	(	110	48	72	H	150	68	104	h
11	09	9	TAB	horizontal TAB	51	29	41	)	111	49	73	I	151	69	105	i
12	0A	10	LF	new Line Feed	52	2A	42	*	112	4A	74	J	152	6A	106	j
13	0B	11	VT	Vertical Tab	53	2B	43	+	113	4B	75	K	153	6B	107	k
14	0C	12	FF	new page From Feed	54	2C	44	,	114	4C	76	L	154	6C	108	l
15	0D	13	CR	Carriage Return	55	2D	45	-	115	4D	77	M	155	6D	109	m
16	0E	14	SO	Shift Out	56	2E	46	.	116	4E	78	N	156	6E	110	n
17	0F	15	SI	Shift In	57	2F	47	/	117	4F	79	O	157	6F	111	o
20	10	16	DLE	Data Link Escape	60	30	48	0	120	50	80	P	160	70	112	p
21	11	17	DC1	Device Control 1	61	31	49	1	121	51	81	Q	161	71	113	q
22	12	18	DC2	Device Control 2	62	32	50	2	122	52	82	R	162	72	114	r
23	13	19	DC3	Device Control 3	63	33	51	3	123	53	83	S	163	73	115	s
24	14	20	DC4	Device Control 4	64	34	52	4	124	54	84	T	164	74	116	t
25	15	21	NAK	negative acknowledge	65	35	53	5	125	55	85	U	165	75	117	u
26	16	22	SYN	SYNchronous idle	66	36	54	6	126	56	86	V	166	76	118	v
27	17	23	ETB	End of Transmission. Block	67	37	55	7	127	57	87	W	167	77	119	w
30	18	24	CAN	CANcel	70	38	56	8	130	58	88	X	170	78	120	x
31	19	25	EM	End of Medium	71	39	57	9	131	59	89	Y	171	79	121	y
32	1A	26	SUB	SUBstitute	72	3A	58	:	132	5A	90	Z	172	7A	122	z
33	1B	27	ESC	ESCape	73	3B	59	;	133	5B	91	[	173	7B	123	{
34	1C	28	FS	File Separator	74	3C	60	<	134	5C	92	\	174	7C	124	|
35	1D	29	GS	Group Separator	75	3D	61	=	135	5D	93	]	175	7D	125	}
36	1E	30	RS	Record Separator	76	3E	62	>	136	5E	94	^	176	7E	126	~
37	1F	31	US	Unit Separator	77	3F	63	?	137	5F	95	_	177	7F	127	DELETE

Códigos Binarios

ASCII extendido

Caracteres ASCII de control		
00	NULL	(carácter nulo)
01	SOH	(inicio encabezado)
02	STX	(inicio texto)
03	ETX	(fin de texto)
04	EOT	(fin transmisión)
05	ENQ	(consulta)
06	ACK	(reconocimiento)
07	BEL	(timbre)
08	BS	(retroceso)
09	HT	(tab horizontal)
10	LF	(nueva línea)
11	VT	(tab vertical)
12	FF	(nueva página)
13	CR	(retorno de carro)
14	SO	(desplaza afuera)
15	SI	(desplaza adentro)
16	DLE	(esc.vínculo datos)
17	DC1	(control disp. 1)
18	DC2	(control disp. 2)
19	DC3	(control disp. 3)
20	DC4	(control disp. 4)
21	NAK	(conf. negativa)
22	SYN	(inactividad sínc)
23	ETB	(fin bloque trans)
24	CAN	(cancelar)
25	EM	(fin del medio)
26	SUB	(sustitución)
27	ESC	(escape)
28	FS	(sep. archivos)
29	GS	(sep. grupos)
30	RS	(sep. registros)
31	US	(sep. unidades)
127	DEL	(suprimir)

Caracteres ASCII imprimibles					
32	espacio	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

ASCII extendido							
128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

Códigos binarios

- Ejercicio:
  - Buscar información sobre código UNICODE
    - Número de bits
    - Número de caracteres codificados
    - Utilidad/conveniencia
  - Ver la tabla en el enlace: Tabla Unicode

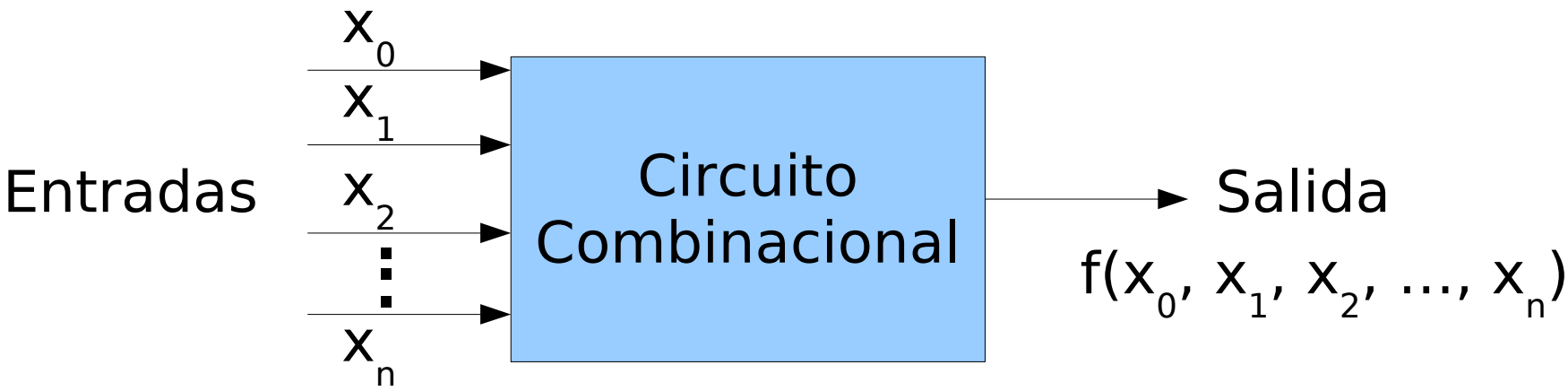
Tema IV – Parte II

Funciones Combinacionales

Tema IV – Parte II

Funciones Combinacionales

Funciones Combinacionales



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

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:

Función compl. especificada

$X_2\ X_1\ X_o$	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

Función incompl. especificada

$X_2\ X_1\ X_o$	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 de Karnaugh
- Circuito
- Lenguajes de descripción de hardware (HDL) : Verilog.

Funciones Combinacionales

Ejemplo: función de tres variables

Expresión: f(x,y,z) = xy + xz + yz

Tabla de verdad

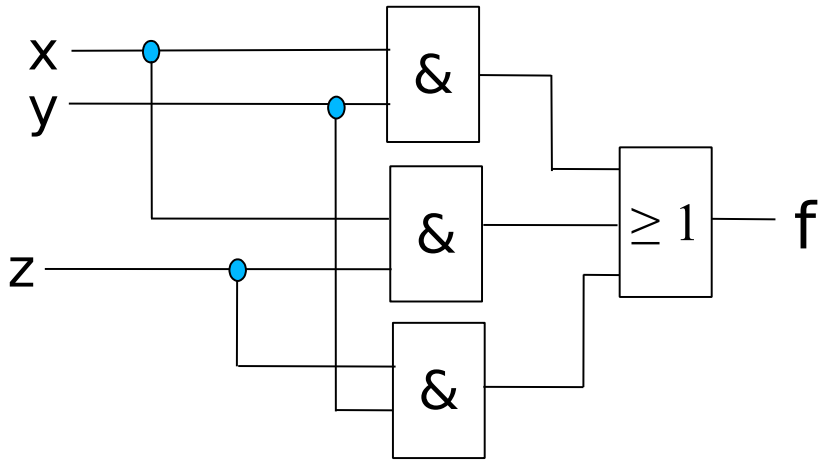
xyz	f(x,y,z)
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

Mapa de Karnaugh

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

f

Circuito



Código Verilog

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

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

Funciones Combinacionales

Ejemplo: función de 4 variables

Expresión: $g(x,y,z,u) = xyu + xz\bar{u} + yz$

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, v, z ,u
          output g
        );

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

endmodule
```

Funciones Combinacionales

Formas normalizadas:

- **suma de productos (sp)**

  $\bar{x}yz + x\bar{w} + \bar{z}w$

  $\bar{x}y$

  $xy + \bar{y} + yz$

- **producto de sumas (ps)**

  $(\bar{y}+w)(x+\bar{z})(\bar{x}+y+w)$

  $x\,(y+z)$

  $(x+\bar{z}+w)$

No normalizadas:

$(abc+b'ad+(a+b+c)'+de)'$ $\qquad$ $xy + z(y+w)$

Funciones Combinacionales

Formas normalizadas:

Para obtener una forma normalizada basta aplicar las leyes de DeMorgan y la propiedad distributiva

- Si una función se expresa en **suma de productos** es fácil obtener sus **unos**

  si un producto es 1 la función vale 1 (elemento dominante de la suma)

- Si una función se expresa en **producto de sumas** es fácil obtener sus **ceros**

  si una suma es 0 la función vale 0 (elemento dominante del producto)

Ejercicio:
obtenga las tablas de verdad de $f = \bar{x}yz + x\bar{w} + \bar{z}w$ / $g=(\bar{y}+w)(x+\bar{z})(\bar{x}+y+w)$

Funciones Combinacionales

Formas canónicas:

- **suma de mintérminos**

el **mintérmino** es un producto que contiene todas las variables

$\bar{x}yzw + xyz\bar{w} + xy\bar{z}w$ (4 variables)

$\bar{x}yz + x\bar{y}z$ (3 variables)

$xy + \bar{x}\,\bar{y} + x\bar{y}$ (2 variables)

- **producto de maxtérminos**

el **maxtérmino** es una suma que contiene todas las variables

$(x+\bar{y}+z+w)(x+y+\bar{z}+w)(\bar{x}+y+z+w)$ (4 variables)

$(x+\bar{z}+y)\,(x+z+\bar{y})$ (3 variables)

$(\bar{y}+\bar{z})\,(y+z)$ (2 variables)

Funciones Combinacionales

Siempre es posible obtener una expresión de la función mediante **suma de mintérminos** y esta es **única:**

- para cada combinación de entradas en la que la función vale 1 se elige el mintérmino que vale 1 para esa combinación

- se suman todos los mintérminos generados

a b c	z
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

$$z = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc = m3 + m5 + m6 + m7$$

notación m

$$z = \Sigma(3,5,6,7)$$

Funciones Combinacionales

Siempre es posible obtener una expresión de la función mediante **producto de maxtérminos** y esta es **única:**

- para cada combinación de entradas en la que la función vale 0 se elige el maxtérmino que vale 0 para esa combinación

- se multiplican todos los maxtérminos generados

a b c	z
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

$z = \Pi(0,1,2,4)$

notación M

$z = (a+b+c)(a+b+\overline{c})(a+\overline{b}+c)(\overline{a}+b+c) = M0\ M1\ M2\ M4$

Tema IV – Parte III

Análisis de Circuitos Combinacionales

Análisis de Circuitos Combinacionales

Análisis Lógico:

Dado un circuito, consiste en encontrar:

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

Análisis Temporal:

Dado un circuito, consiste en representar la evolución en el tiempo de las entradas y salidas del circuito (cronograma**)**

- **análisis ideal:** Suponiendo que las puertas no tienen retrasos.
- **análisis con retrasos:** un modelo simple es suponer que todas las puertas introducen el mismo retraso.

Análisis de Circuitos Combinacionales

Análisis Lógico: 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).

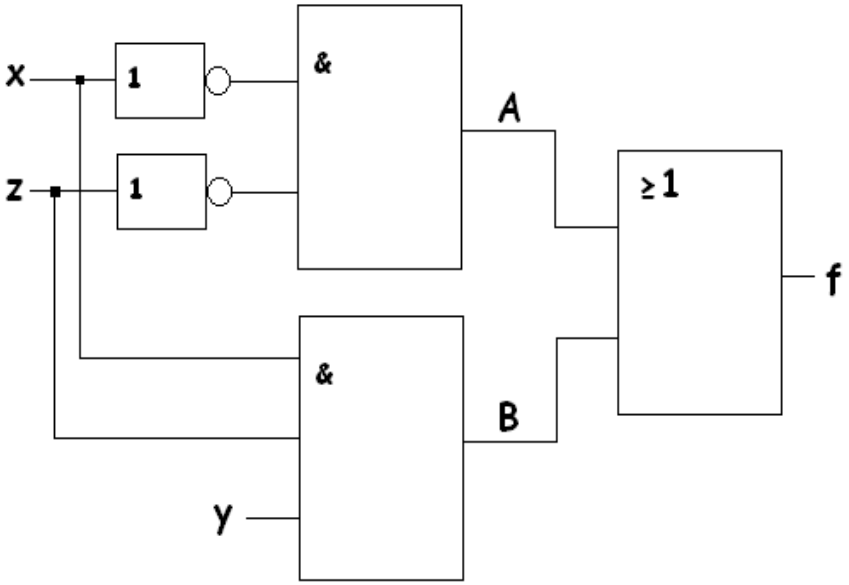
3. Se repite el paso anterior hasta obtener la función de salida

4. Se simplifica la expresión obtenida y/o se representa en un mapa o tabla

Análisis de Circuitos Combinacionales

Análisis Lógico: Ejemplo

Circuito:



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 si

- xyz=1 si x=y=z=1 (111)
- ó
- x'z'=1 si x=z=0 (0-0)

Análisis de Circuitos Combinacionales

Análisis temporal: Procedimiento

- **análisis ideal:** podemos extraer la información de la tabla de verdad o la expresión lógica

- **análisis con retrasos:** debemos obtener todas las ondas que se propagan por el circuito e ir retrasando la onda cada vez que se pasa por una puerta

Análisis de Circuitos Combinacionales

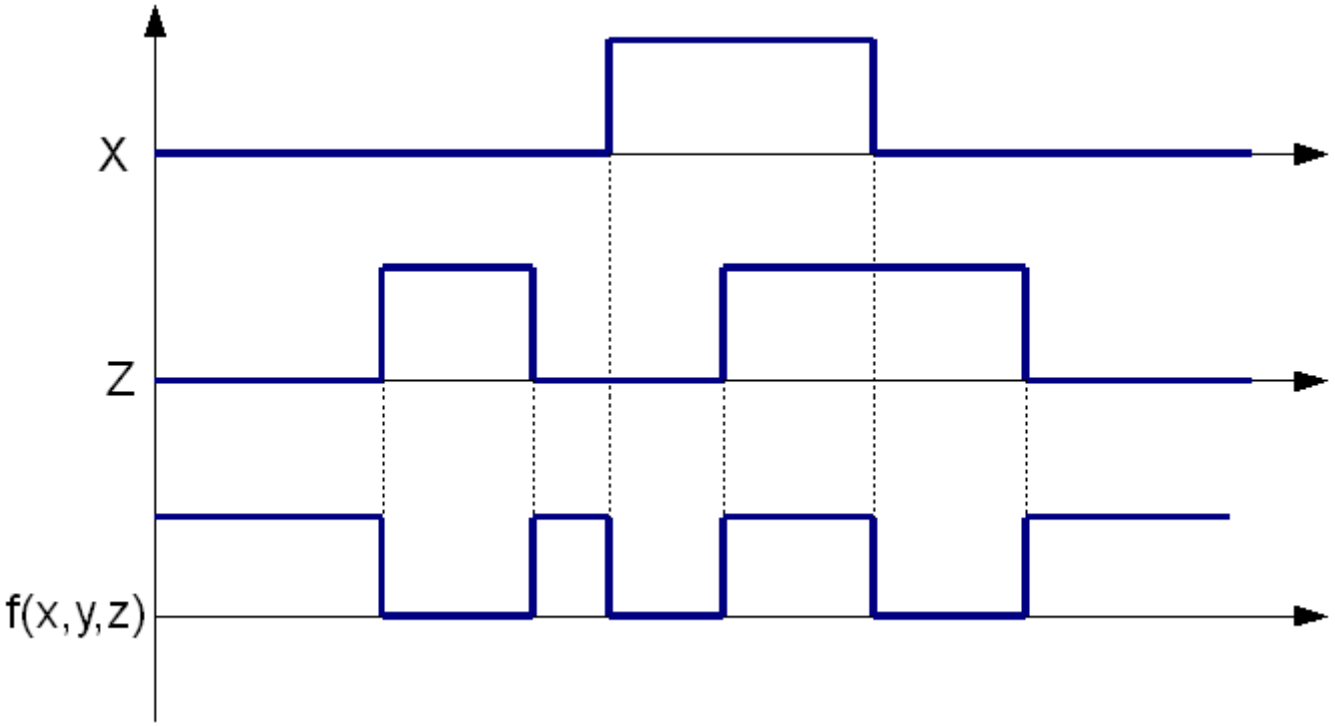
Análisis Temporal (ideal): Ejemplo

Tabla:

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

Cronograma (con y=1):

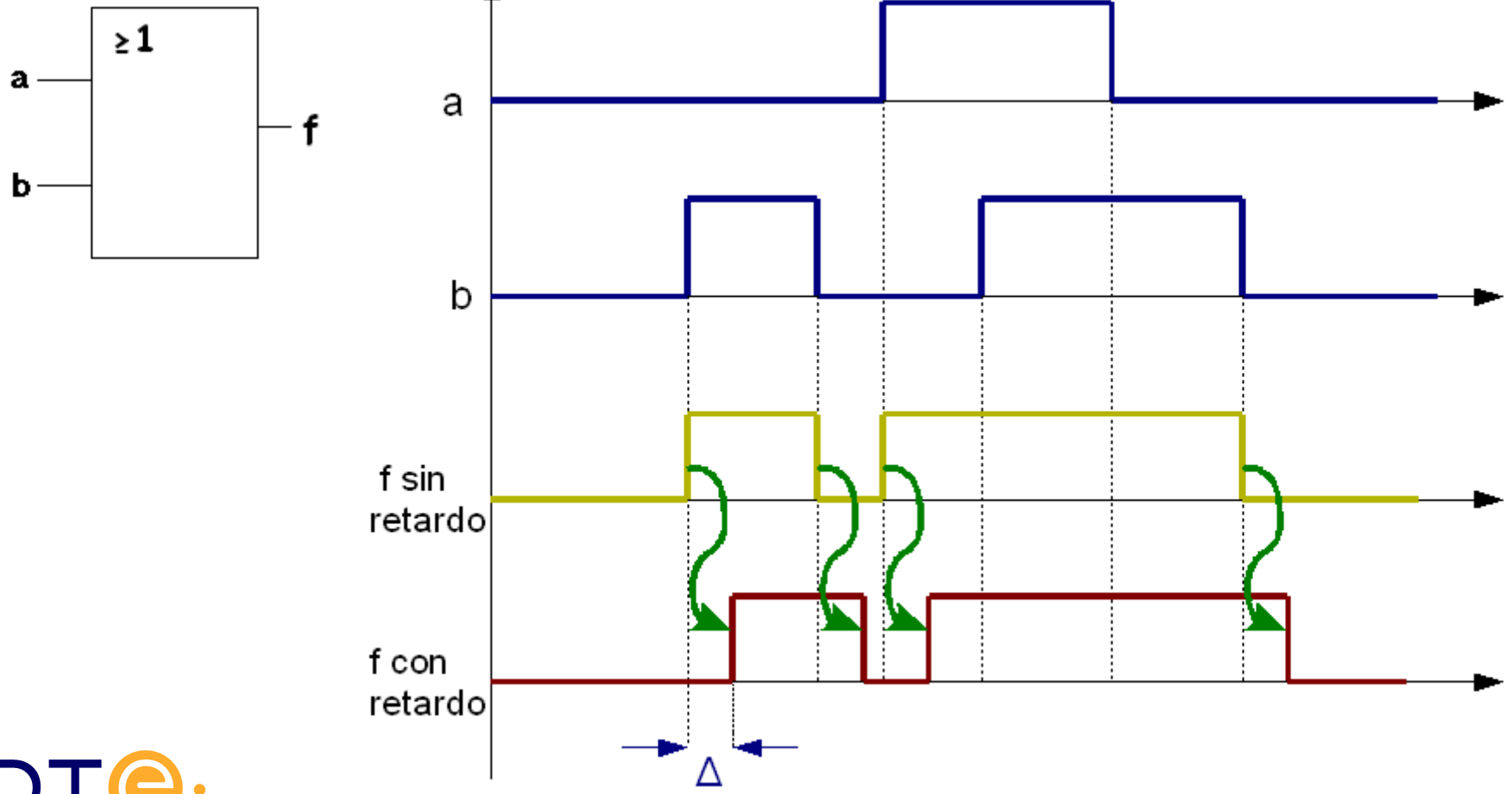
(sin considerar retrasos)



Análisis de Circuitos Combinacionales

Análisis Temporal (con retrasos): Ejemplo

Para cada puerta hay que tener en cuenta el retraso, consideremos una OR que introduce un retraso Δ

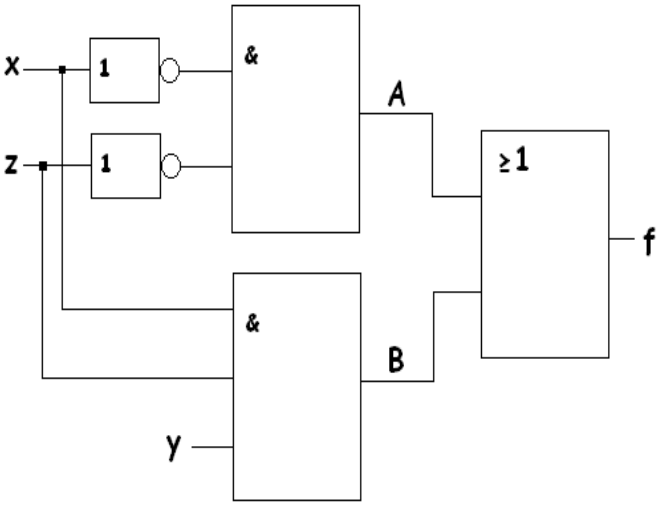


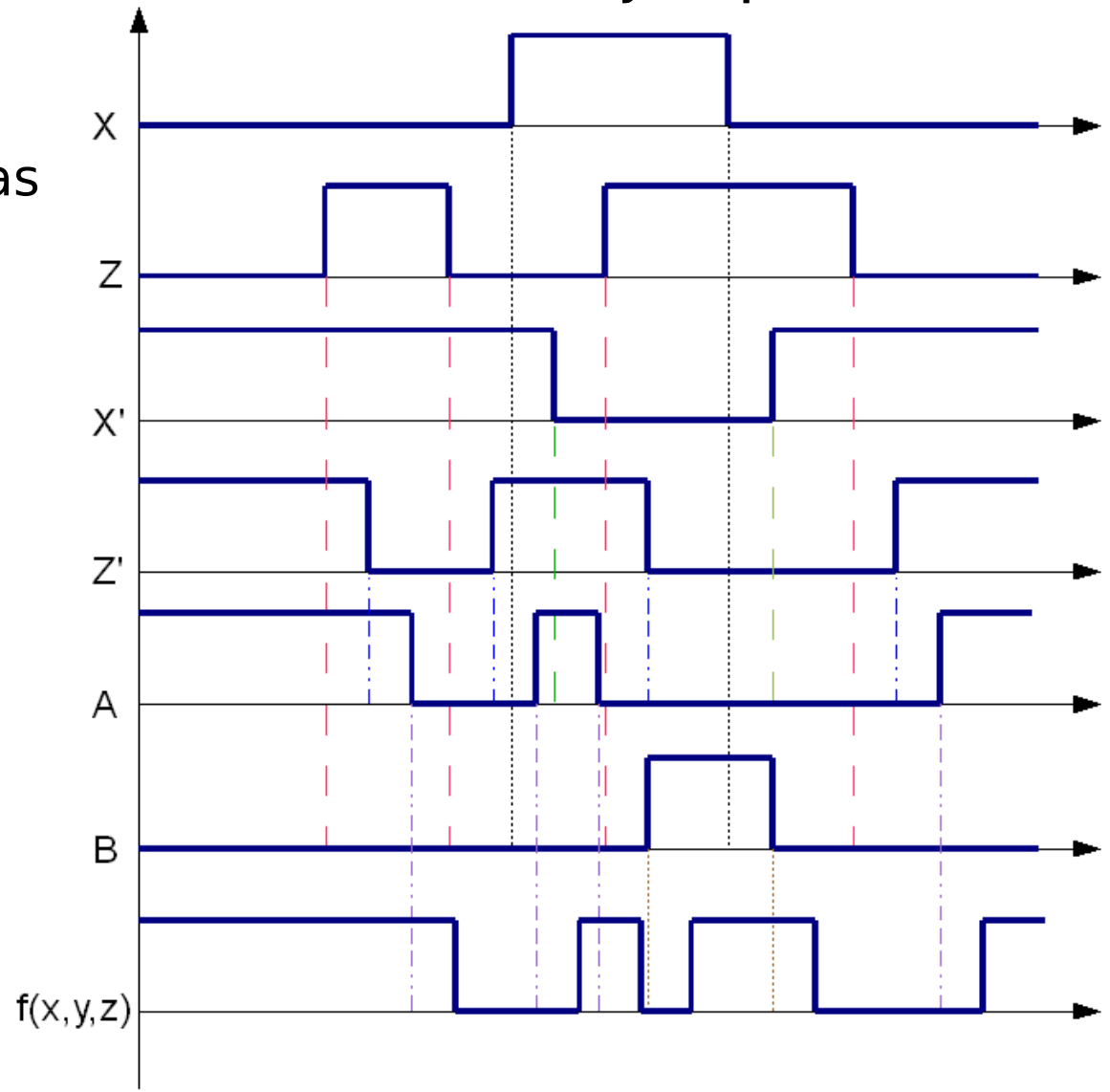
Análisis de Circuitos Combinacionales

Análisis Temporal (con retrasos): Ejemplo

Cronograma (con y=1)

(retraso Δ en todas las puertas)





Análisis de Circuitos Combinacionales

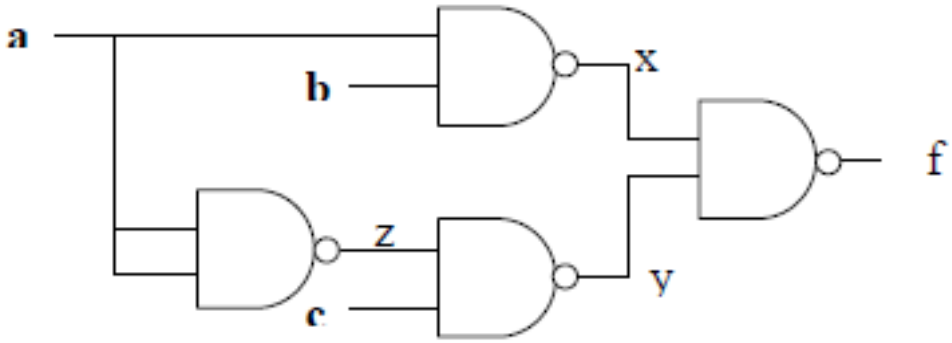
Análisis Temporal (con retrasos): Beneficios

- Permite estimar el retraso de propagación de un circuito que es el tiempo que tarda en proporcionar un valor de salida correcto

- Permite analizar comportamientos no esperados debido a las características temporales: retraso excesivo, **azares**,...

Ejercicio: Analizar temporalmente el circuito de la figura en los casos: 1) análisis ideal y 2) análisis con retrasos considerando Δ=1ns Comparar los resultados obtenidos.

b=c=1,
a → señal periódica (T=10ns)



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:

- se parte de unas especificaciones (descripción de la tarea que ha de realizar el circuito) y se obtiene un circuito que cumple con los requerimientos planteados

- las especificaciones pueden ser de varios tipos, por ejemplo:

    - funcionales

    - temporales (velocidad, azares, ...)

    - de consumo

    - de reusabilidad

    - tecnológicas

Diseño de Circuitos Combinacionales

Objetivos y conceptos básicos

En este tema nos centraremos en el diseño de circuitos:

- que tengan dos niveles (tres para simple raíl)

- que realicen funciones en suma de productos o producto de sumas

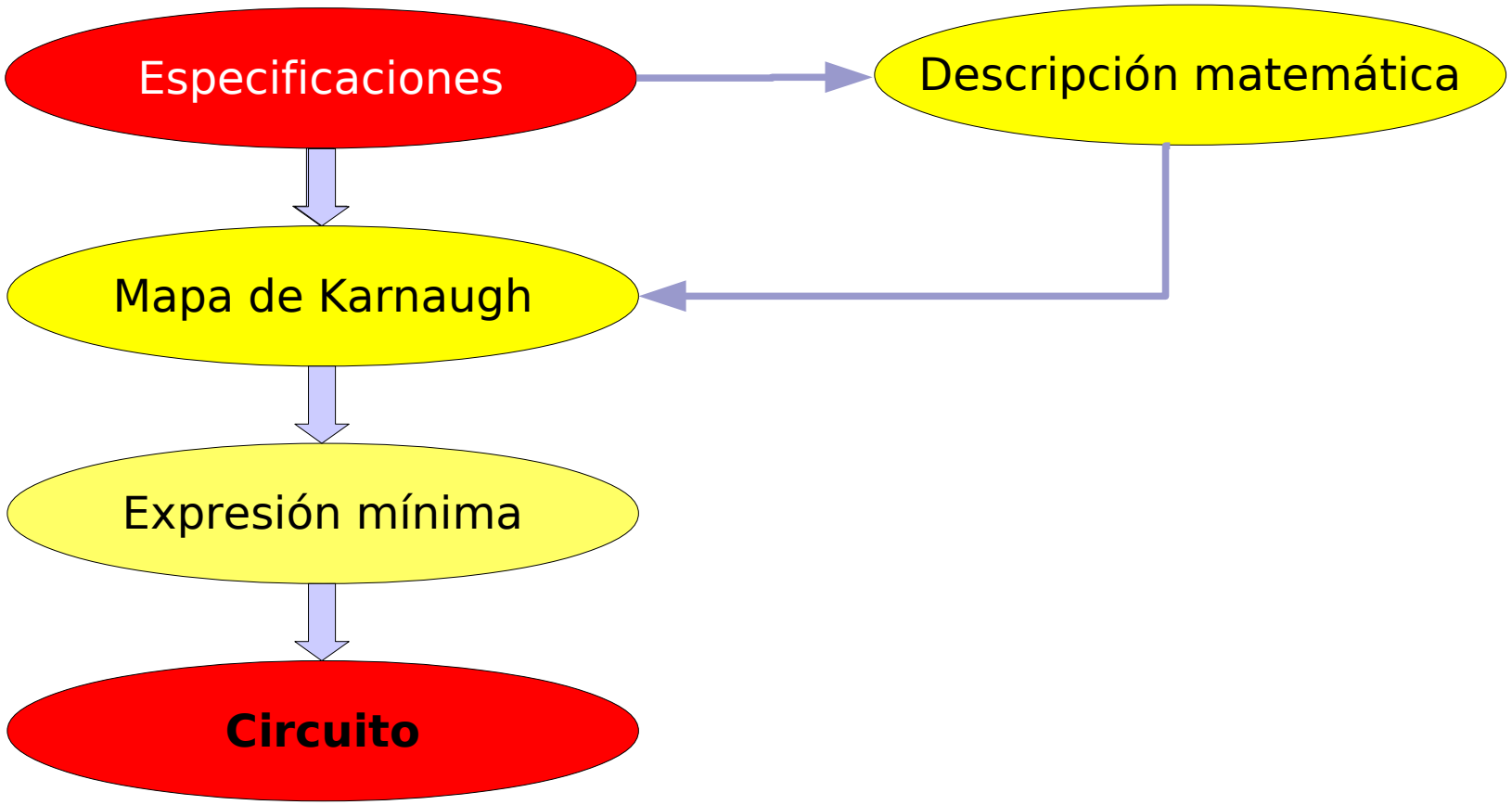
- que sean óptimos: la expresión en sp o ps debe ser lo más simple posible

Por tanto, los criterios de diseño son:

- reducir el número de puertas
- reducir el número de conexiones
- uso de puertas AND, OR, NAND y NOR (e inversores en simple raíl)

Diseño de Circuitos Combinacionales

Pasos del proceso:



Diseño de Circuitos Combinacionales

Pasos del proceso:

Paso 1: Descripción textual -> Descripción matemática

- determinar variables de entrada y salida

- si procede, asignar significado a los valores 0 y 1.

- obtener una descripción matemática: expresión, tabla o mapa de Karnaugh

Paso 2: Obtener el mapa de Karnaugh

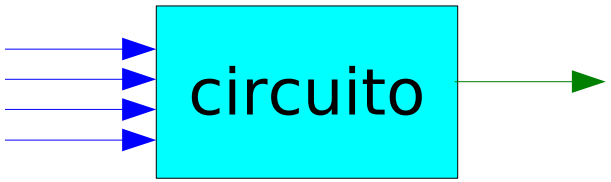
Si en el paso anterior no se obtuvo directamente el mapa sino una expresión algebraica, se tendrá que generar el mapa de Karnaugh a partir de la expresión obtenida

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: X3-X0, donde X3 es el bit más significativo. Diseñe un circuito que de salida Z = 1 si y sólo si el número X3-X0 es primo.



x3x2x1x0	z
0 0 0 0	0
0 0 0 1	0
0 0 1 0	0
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.

Entradas: tres bombillas, interruptor Salida: Alarma

(b_1, b_2, b_3, i) $\qquad$ a

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

i $b_1b_2b_3$	a	i $b_1b_2b_3$	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

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 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 del K-mapa, o maxtérminos, que son términos suma.
  - Agrupamos los maxtérminos para conseguir términos sumas con menor número de variables (**implicadas**).

Diseño con K-mapa

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

El agrupamiento de **1** para construir términos productos con menor número de variables es posible gracias a:

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

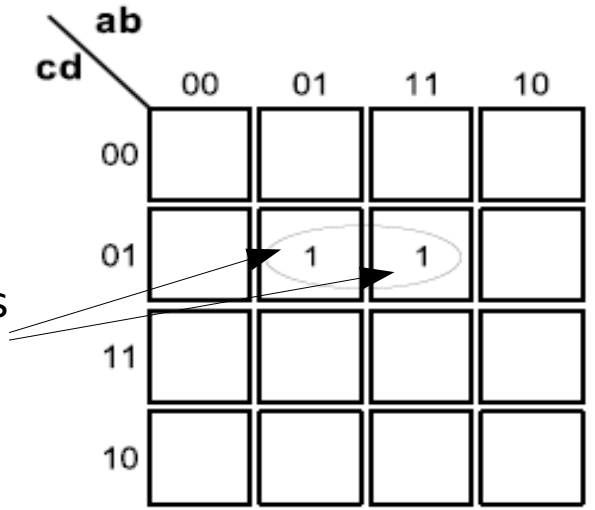
2) Si un término producto se expresa como **p q**, otro adyacente a él, que varíe un bit, sería **p q'**, por tanto la suma de los dos:

$$pq + pq' = p(q + q') = p$$

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



Diseño con K-mapa

Implicante

Es un 1 o grupo de 1 representado en el K-mapa. Los grupos tienen 2^n elementos, y estos deben ser vecinos.

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

El número de 1 del grupo determina el orden de la implicante.

El orden de la 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

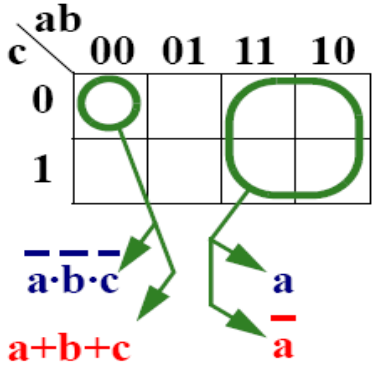
Implicante

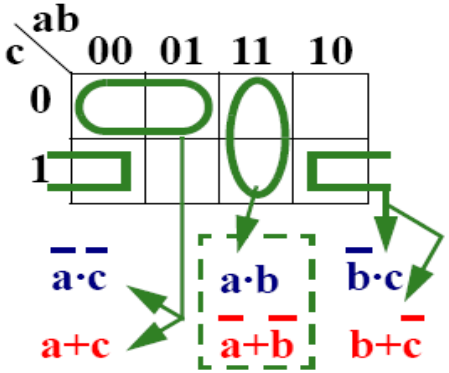
Orden	Nº de 1's	Nº variables	Ejemplo 5 var.	
			Implicante	Cuantas
0	$1=2^0$	n	ab'cd'e	32
1	$2=2^1$	n - 1	ab'd'e	80
2	$4=2^2$	n - 2	ab'e	80
3	$8=2^3$	n - 3	b'e	40
4	$16=2^4$	n - 4	b'	10
5	$32=2^5$	n - 5	1	1
k	$m=2^k$	n - k		

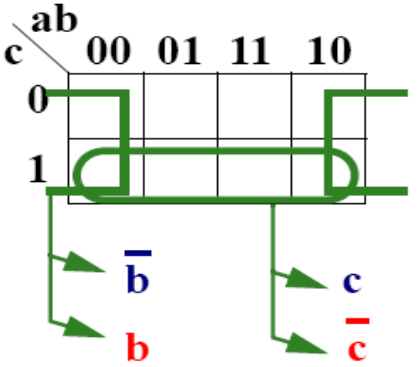
Diseño de Circuitos Combinacionales

Diseño con K-mapa

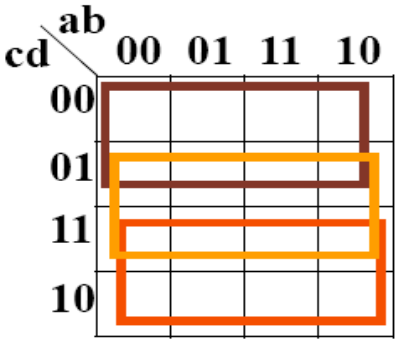
Agrupaciones posibles

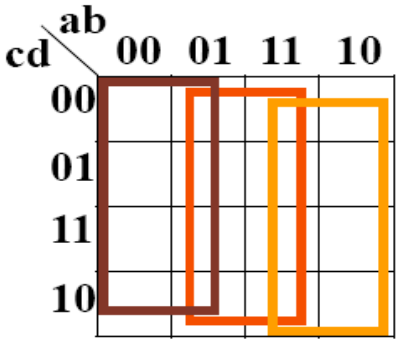


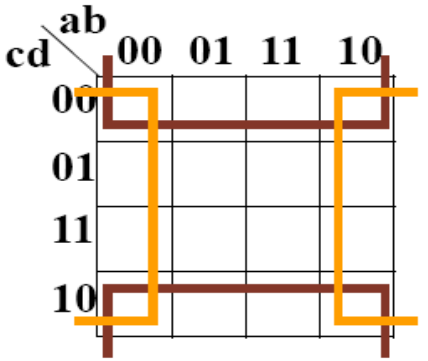




Los mapas de 4 variables contienen varios mapas de 3



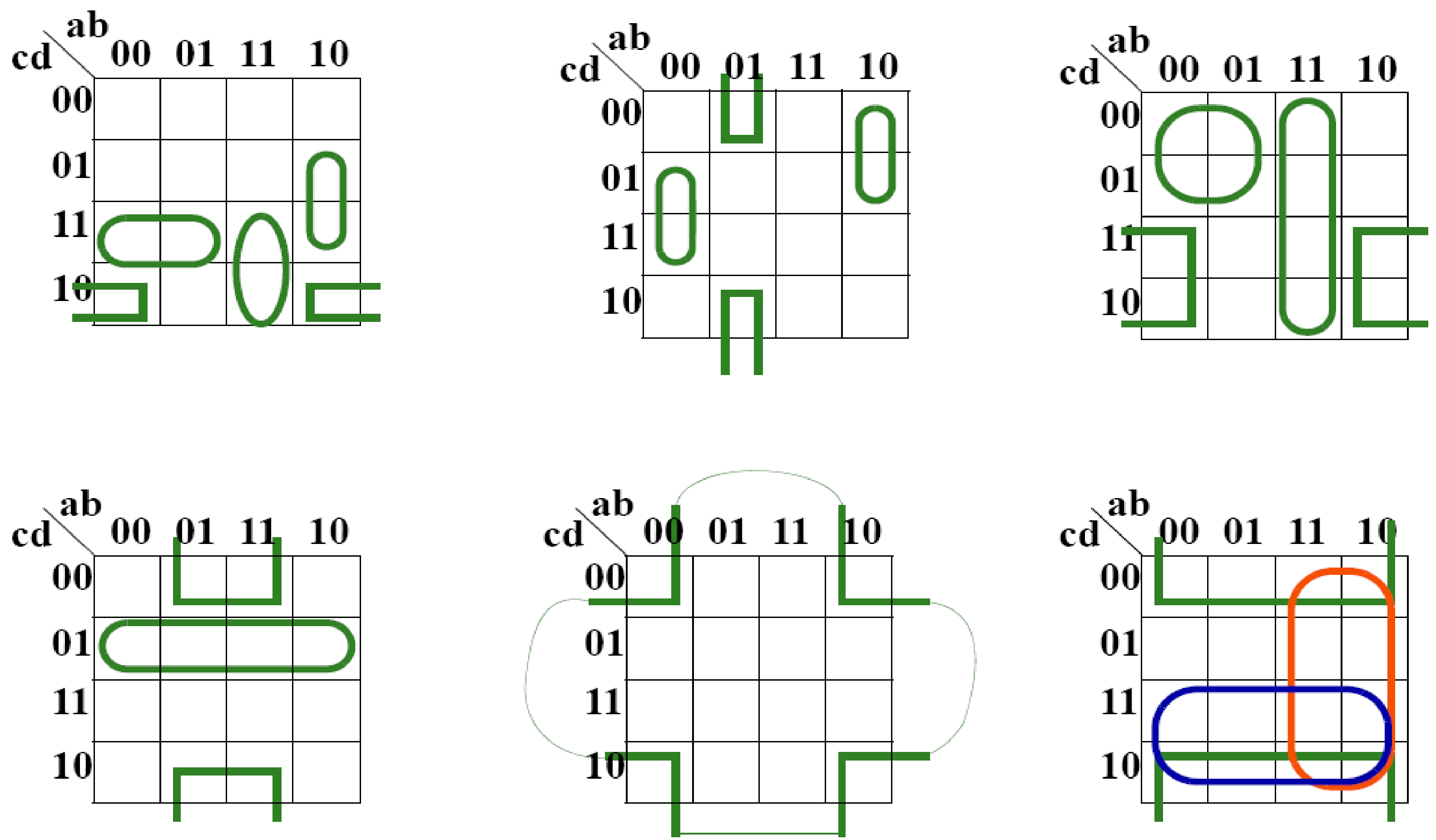




Diseño de Circuitos Combinacionales

Diseño con K-mapa

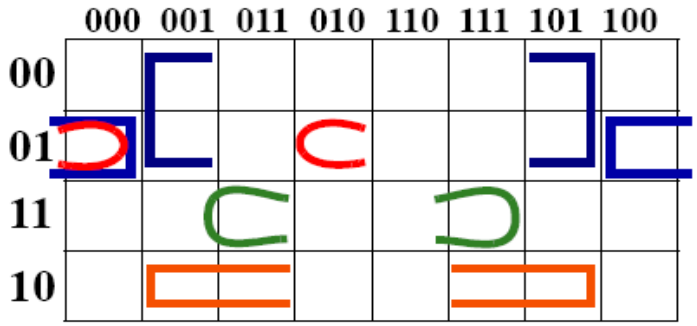
Agrupaciones posibles

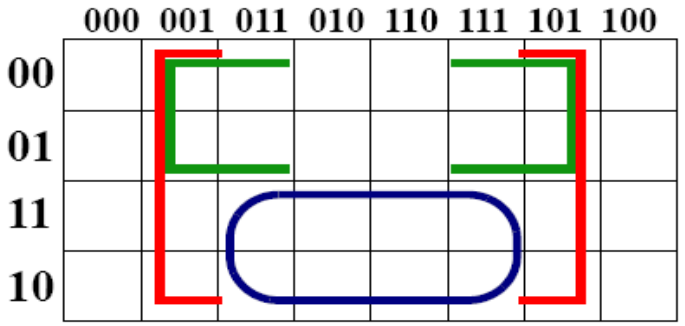


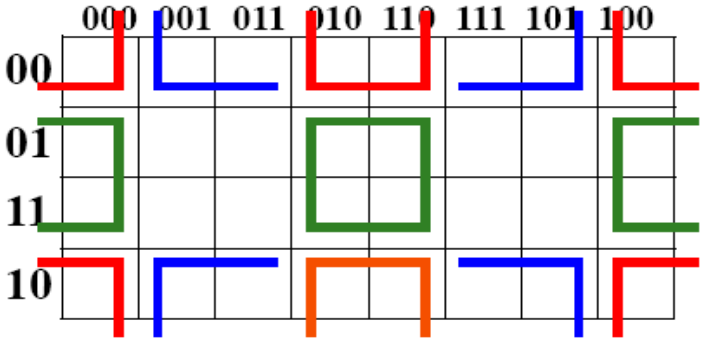
Diseño de Circuitos Combinacionales

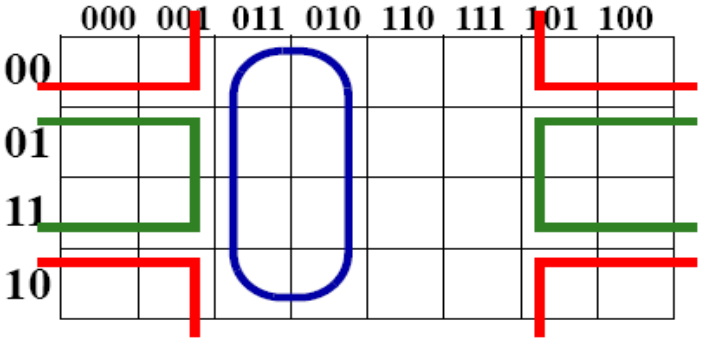
Diseño con K-mapa

Agrupaciones posibles





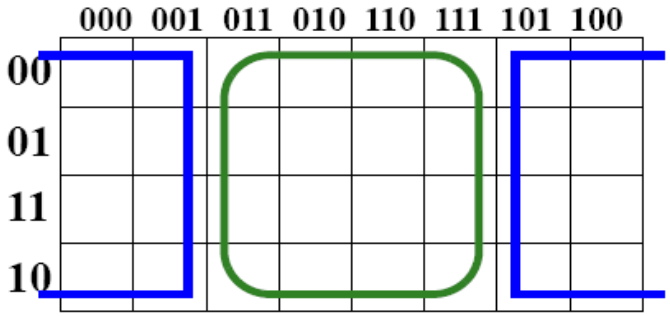


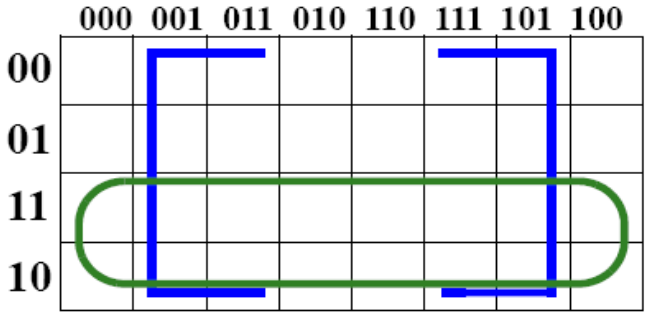


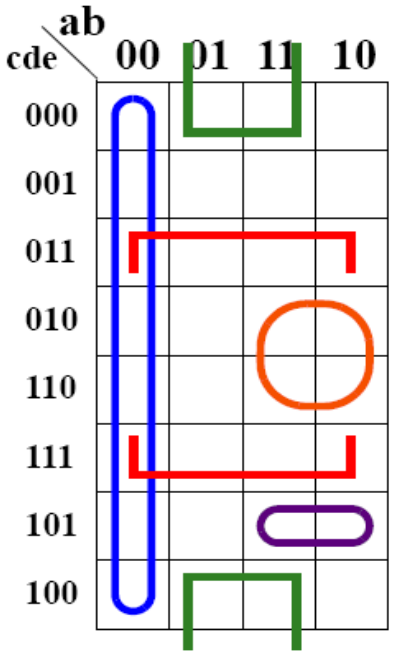
Diseño de Circuitos Combinacionales

Diseño con K-mapa

Agrupaciones posibles







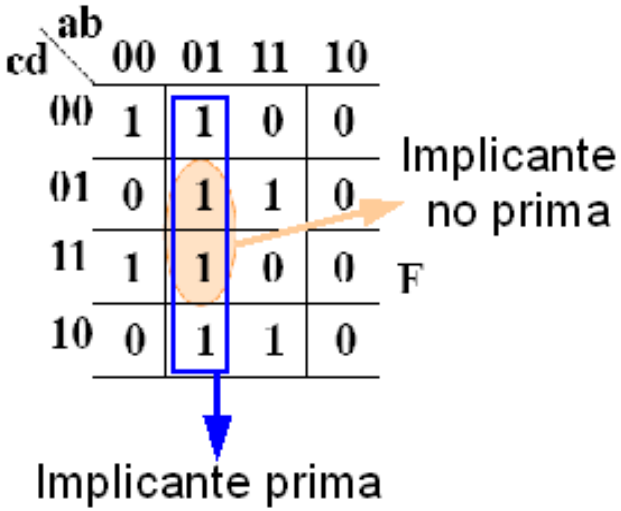
Diseño de Circuitos Combinacionales

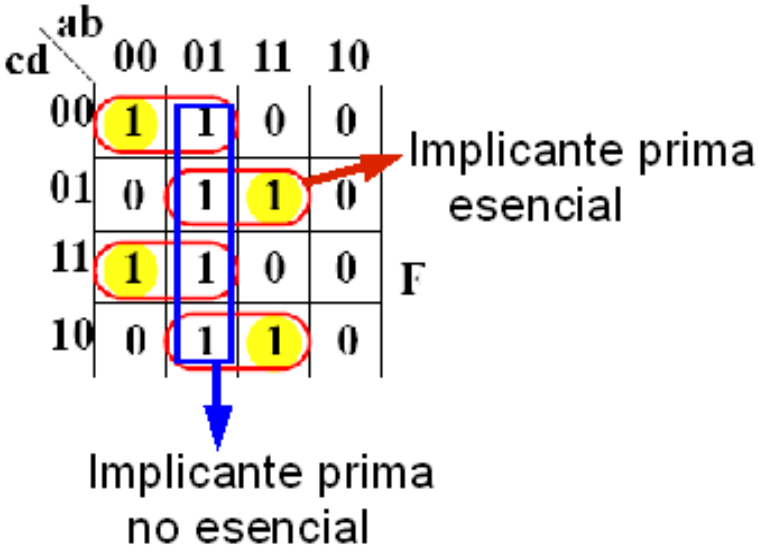
Diseño con K-mapa

Definiciones

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

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





Diseño de Circuitos Combinacionales

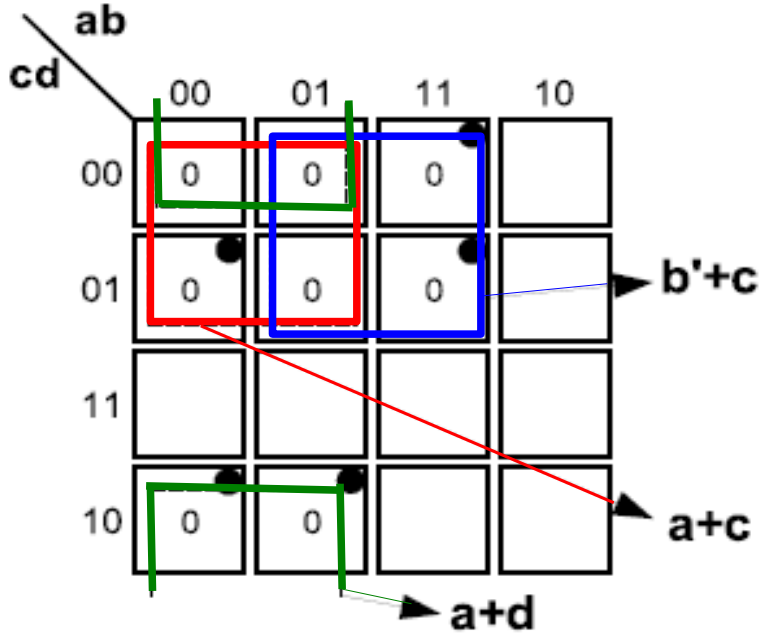
Diseño con K-mapa

La expresión mínima en producto de sumas se obtiene trabajando con las implicadas y los maxtérminos de la misma forma.

Una implicada es **prima** si no está cubierta por ninguna otra implicada de la función.

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

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



Diseño de Circuitos Combinacionales

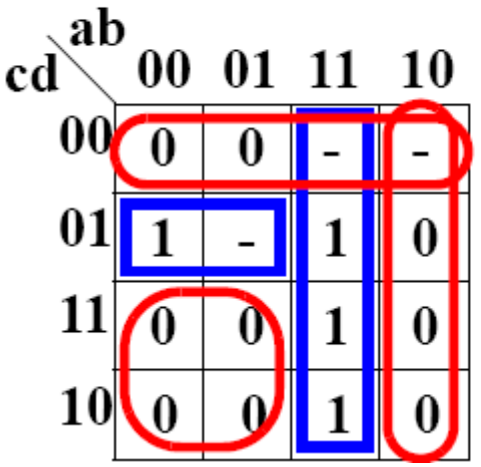
Diseño con K-mapa

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



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

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

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 mintérminos del mismo.

Directrices para la búsqueda de la expresión mínima:

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

2) Para los mintérminos 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 mintérminos 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.

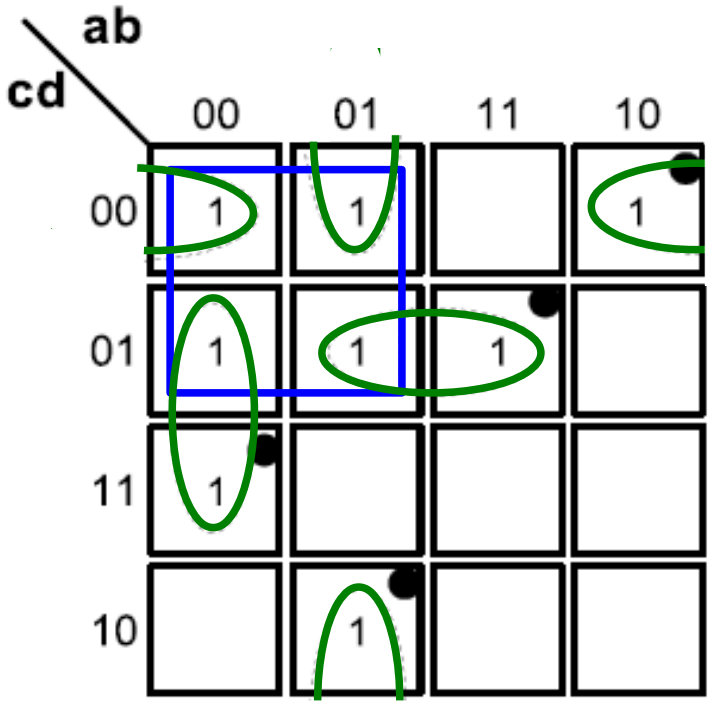
Expresión mínima en p.s.

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

Diseño de Circuitos Combinacionales

Ejemplos de obtención de la expresión mínima

Ejercicio 1.- f = ∑(0,1,3,4,5,6,8,13)



Buscamos las Ip esenciales

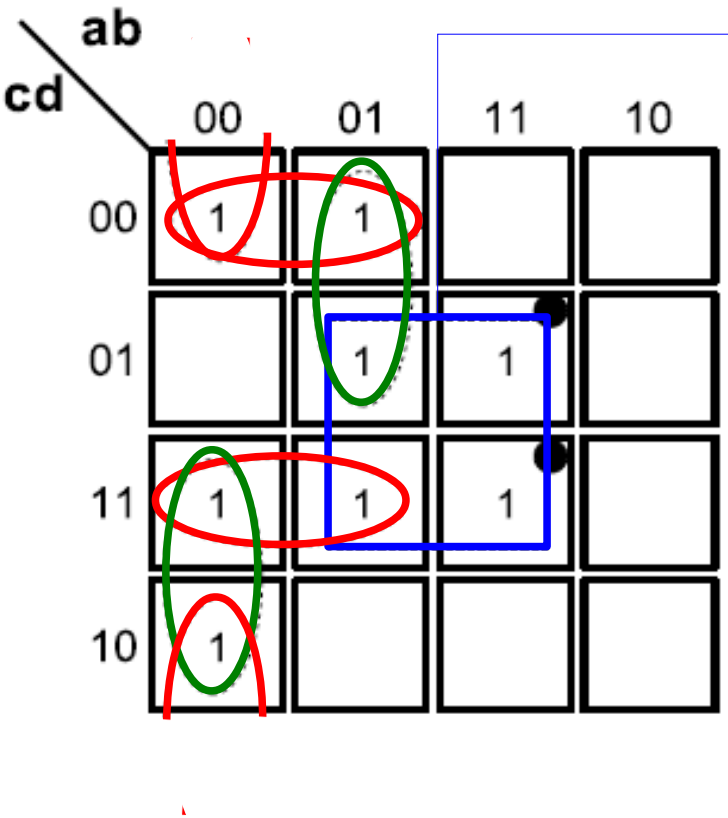
f = a'b'd + b'c'd'+a'bd'+bc'd

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 = ∑(0,2,3,4,5,7,13,15)



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

2)Nos fijamos en el mintérmino 4. Está cubierto por las Ip **a'c'd'** y **a'bd**. Ambas del mismo coste pero la primera cubre también al mintérmino 0 que no estaba cubierto por **bd**. Escogemos **a'c'd'**

3) Ahora sólo quedan por cubrir los mintérminos 2 y3. Obviamente escogemos **a'b'c**

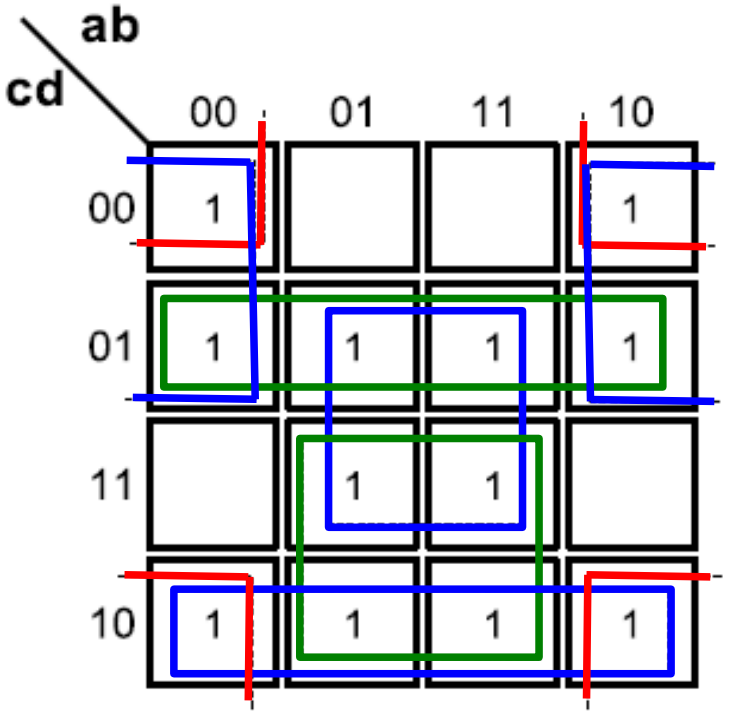
f = bd + a'c'd' + a'b'c

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)Buscamos las Ip esenciales : ¡NO HAY!

2)Nos fijamos en el mintérmino 0. Está cubierto por las Ip **b'd'** y **b'c'**, ambas del mismo coste. **Suponemos** que la expresión mínima está formada por **b'c'**.

f = b'c' + bd + cd'

3) Repetimos el proceso en el caso en que el mintérmino 0 hubiera sido cubierto por b'd'

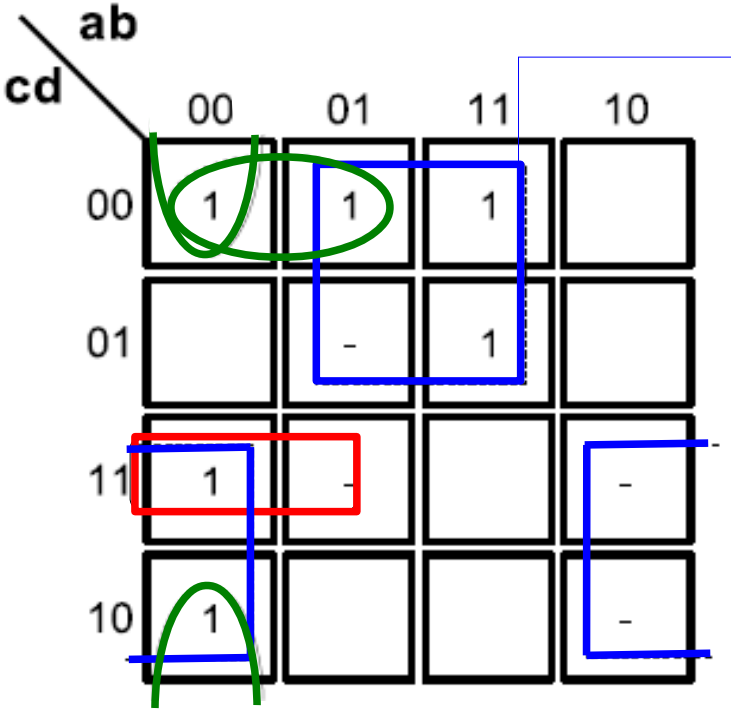
f = b'd' + c'd + bd

Fin. Se han cubierto todos los mintérminos 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 = ∑(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 3, cubierto por **a'cd** y **b'c**. La mejor opción es **b'c.**

3) Sólo queda por cubrir el mintérmino 0. Hay dos posibilidades: las implicantes **a'b'd'** y **a'c'd'**. Cualquier opción es válida.

4) No se busca cubrimiento de inespecificaciones

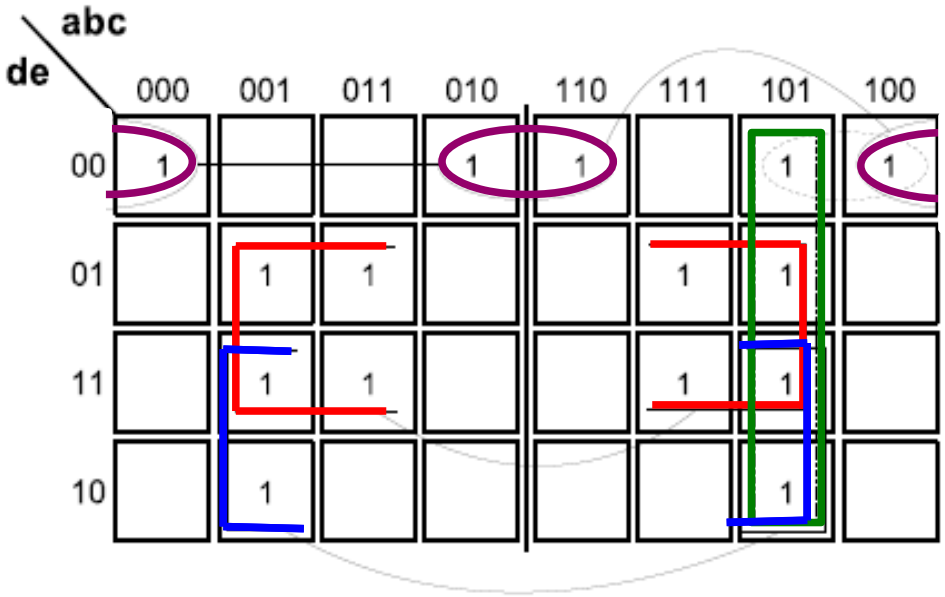
f = bc' + a'b'd' + b'c

Fin. Se han cubierto todos los mintérminos

Diseño de Circuitos Combinacionales

Ejemplos de obtención de la expresión mínima

Ejercicio 5.- f = ∑(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 los sub K-mapas de 4 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' + c e + b' c d + a b'c