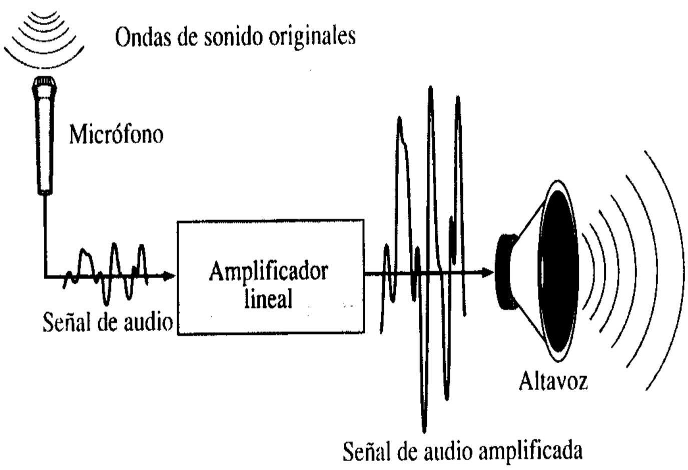
Tema 1. Puertas Lógicas y Algebra de Boole

CONTENIDOS

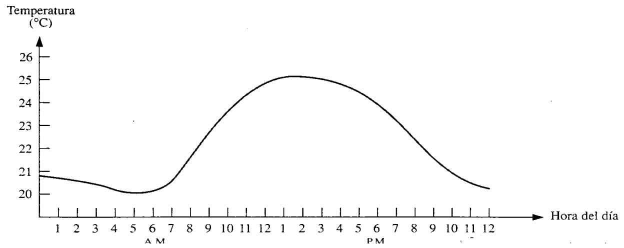
CONTENIDOS (II)

Dpto. de Sistemas Electrónicos y de Control

Señal Analógica

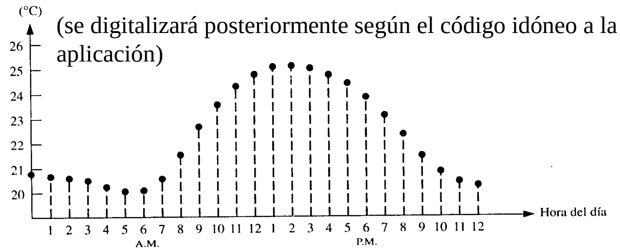


Magnitud Analógica

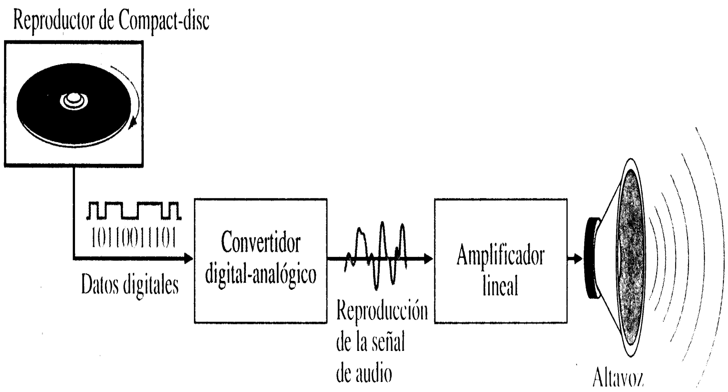


Magnitud Muestreada (Analógica Discreta)

(se digitalizará posteriormente según el código idóneo a la aplicación)



Ejemplo de Sistema Digital



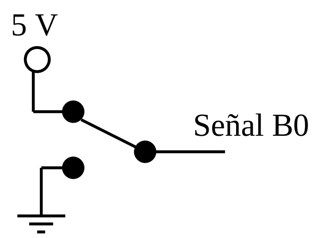
Bit: Unidad Mínima de Información

$V_{H(máx)}$ **ALTO (HIGH) "1"**
$V_{H(min)}$
$V_{L(máx)}$ **BAJO (LOW) "0"** $V_{L(mín)}$

NIVEL 0 = de $V_{Lmín}$ a $V_{Lmáx}$

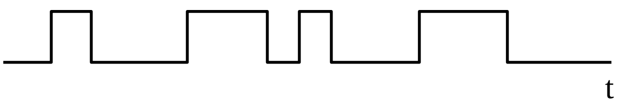
NIVEL 1 = de $V_{Hmín}$ a $V_{Hmáx}$

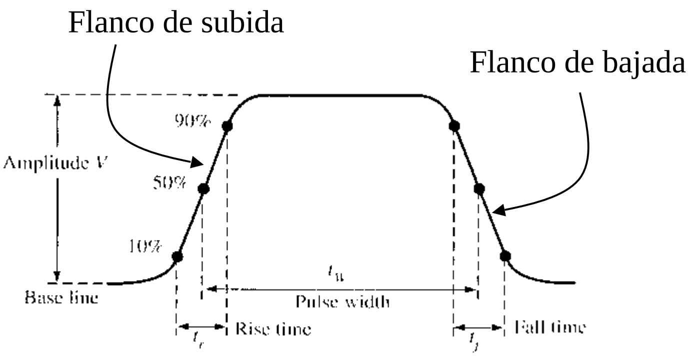
Ejemplo:



Dpto. de Sistemas Electrónicos y de Control

Forma de onda digital





Teoría básica de la codificación (I)

Con 1 Bit se pueden codificar dos objetos, propiedades o situaciones diferentes.

Blanco	0
Negro	1

Abierto	0
Cerrado	1

Teoría básica de la codificación (II)

Con 2 Bits se pueden codificar cuatro objetos, propiedades o situaciones diferentes.

Blanco	00
Negro	01
Azul	10
Verde	11

Español	00
Inglés	01
Francés	10
Japonés	11

Teoría básica de la codificación (III)

Con 3 Bits se pueden codificar ocho objetos, propiedades o situaciones diferentes.

Planta Baja	000
1ª Planta	001
2ª Planta	010
3ª Planta	011
4ª Planta	100
5ª Planta	101
6ª Planta	110
7ª Planta	111

Teoría básica de la codificación (y IV)

Con “n” Bits se pueden codificar hasta 2^n objetos, propiedades o situaciones diferentes.

Por ejemplo: para codificar los siete días de la semana necesitaremos 3 bits

Lunes	000
Martes	101
Miércoles	100
Jueves	010
Viernes	001
Sábado	011
Domingo	111

Códigos de numeración (I)

DECIMAL

- Nº Símbolos: 10 (0..9)
- Código base: 10
- Modelo:

$$\boxed{\ldots + d_2 \cdot 10^2 + d_1 \cdot 10^1 + d_0 \cdot 10^0 + d_{-1} \cdot 10^{-1} + d_{-2} \cdot 10^{-2} + \ldots}$$

- Ejemplo: 128,32 $\Rightarrow$ 128,32 !

Dpto. de Sistemas Electrónicos y de Control

Códigos de numeración (II)

BINARIO

- Nº Símbolos: 2 (0 y 1)
- Código base: 2
- Modelo:

$$\boxed{\ldots + d_2 \cdot 2^2 + d_1 \cdot 2^1 + d_0 \cdot 2^0 + d_{-1} \cdot 2^{-1} + d_{-2} \cdot 2^{-2} + \ldots}$$

- Ejemplo: 011,11 $\Rightarrow$ 3,75 !

Códigos de numeración (III)

HEXADECIMAL

- Nº Símbolos: 16 (0...9, A, B, C, D, E, F)
- Código base: 16
- Modelo:

$$\boxed{...+d_2\cdot 16^2+d_1\cdot 16^1+d_0\cdot 16^0+d_{-1}\cdot 16^{-1}+d_{-2}\cdot 16^{-2}+...}$$

- Ejemplo: 9AD $\Rightarrow$ 9x256+10x16+13=2477 !

Códigos de numeración (IV)

HEXADECIMAL

- Permite representar información binaria agrupada en 4 bits de forma más compacta:

0 (0000)	0
1 (0001)	1
2 (0010)	2
3 (0011)	3
4 (0100)	4
5 (0101)	5
6 (0110)	6
7 (0111)	7

8 (1000)	8
9 (1001)	9
10 (1010)	A
11 (1011)	B
12 (1100)	C
13 (1101)	D
14 (1110)	E
15 (1111)	F

$$\underbrace{0\,0\,1\,1}_{0011}\;\underbrace{1\,1\,1\,0}_{1110}\;\underbrace{1\,0\,1\,1}_{1011}$$

0011 → 3, 1110 → E, 1011 → B

$$\underline{\underline{3\ E\ B}}$$

Códigos de numeración (y V)

BINARY CODED DECIMAL (BCD)

- Cada grupo de cuatro bits representa un código decimal entre 0 y 9.
- Las combinaciones binarias desde 1010 a 1111 no se emplean

$$\underbrace{0\,0\,1\,1}\;\underbrace{1\,0\,0\,1}\;\underbrace{0\,1\,0\,1}$$

0011 1001 0101

$$\underline{\underline{3\;9\;5}}$$

Conversión entre códigos (I)

- Binario $\Rightarrow$ Decimal : suma de pesos

$$\begin{array}{ccccc} 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} \\ 1 & 1 & 0, & 0 & 1 \end{array} \Rightarrow 2^2 + 2^1 + 2^{-2} = 6{,}25!$$

- Decimal $\Rightarrow$ Binario : suma de pesos

1. Número de bits necesarios

   Con “n” bits se pueden representar 2^n números

2. Buscar los pesos según el modelo

Ejemplo: 10,375

$$2^3 + 2^1 + 2^{-2} + 2^{-3} = 10{,}375 \Rightarrow 1\;0\;1\;0{,}0\;1\;1!$$

Conversión entre códigos (y II)

- Hexadecimal ⇒ Decimal

$$4B3F \Rightarrow 4\times 16^3 + 11\times 16^2 + 3\times 16^1 + 15\times 16^0 = 19263!$$

- Decimal ⇒ Hexadecimal
- Ejemplo: 11166

$$\frac{11166}{16^3} = 2{,}726\ (2) \qquad \frac{2974}{16^2} = 11{,}617\ (B) \qquad \frac{158}{16} = 9{,}875\ (9) \qquad 14\ (E)$$

$$11166 != 2B9E$$

Código complemento a 2

VALOR	S+M	C2
-8	-	1000
-7	1111	1001
-6	1110	1010
-5	1101	1011
-4	1100	1100
-3	1011	1101
-2	1010	1110
-1	1001	1111

VALOR	S+M	C2
0	1000/0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111

Intervalos Numéricos

- Con “n” bits podemos representar 2^n números decimales
- Con los rangos siguientes:
  - En binario: desde 0 hasta 2^n -1
  - En C.a.2: desde el $-2^{(n-1)}$ hasta el $2^{(n-1)} -1$

Por ejemplo: con 16 bits podemos representar

2^{16} = 65536 números decimales

En Binario: desde el 0 hasta el 65535

En C.a.2: desde el –32768 hasta el 32767

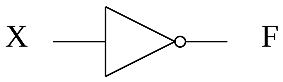
Códigos alfanuméricos: ASCII

DECIMAL VALUE	→	0	16	32	48	64	80	96	112
↓	HEXA-DECIMAL VALUE	0	1	2	3	4	5	6	7
0	0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	‘	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	☻	↕	"	2	B	R	b	r
3	3	♥	‼	#	3	C	S	c	s
4	4	♦	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↨	’	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◙	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	¦
13	D	♪	↔	—	=	M	]	m	}
14	E	♫	▲	.	>	N	∧	n	~
15	F	☼	▼	/	?	O	_	o	△

DECIMAL VALUE	→	128	144	160	176	192	208	224	240
↓	HEXA-DECIMAL VALUE	8	9	A	B	C	D	E	F
0	0	Ç	É	á	░	└	╨	α	≡
1	1	ü	æ	í	▒	┴	╤	β	±
2	2	é	Æ	ó	▓	┬	╥	Γ	≥
3	3	â	ô	ú	│	├	╙	π	≤
4	4	ä	ö	ñ	┤	─	╘	Σ	⌠
5	5	à	ò	Ñ	╡	┼	╒	σ	⌡
6	6	å	û	ª	╢	╞	╓	μ	÷
7	7	ç	ù	º	╖	╟	╫	τ	~
8	8	ê	ÿ	¿	╕	╚	╪	Φ	°
9	9	ë	Ö	⌐	╣	╔	┘	Θ	•
10	A	è	Ü	¬	║	╩	┌	Ω	·
11	B	ï	¢	1/2	╗	╦	█	δ	√
12	C	î	£	1/4	╝	╠	▄	∞	ⁿ
13	D	ì	¥	¡	╜	═	▌	ϕ	²
14	E	Ä	Pt	«	╛	╬	▐	∈	■
15	F	Å	ƒ	»	┐	╧	▀	∩	BLANK 'FF'

Puerta NOT (Inversor lógico)

Símbolo



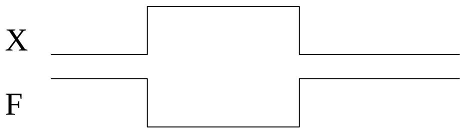
Ecuación

$$F = \overline{X}$$

Tabla de Verdad

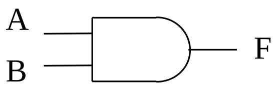
X	F
0	1
1	0

Cronograma



Puerta AND (Producto lógico)

Símbolo



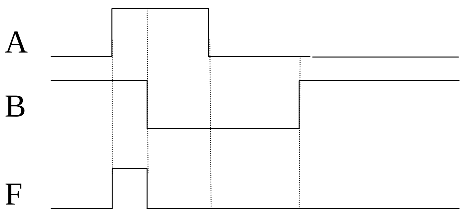
Ecuación

$$F = A \cdot B$$

Tabla de Verdad

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Cronograma

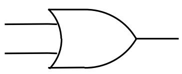


Puerta OR (Suma lógica)

Símbolo

A

B



F

Ecuación

$$F = A + B$$

Tabla de Verdad

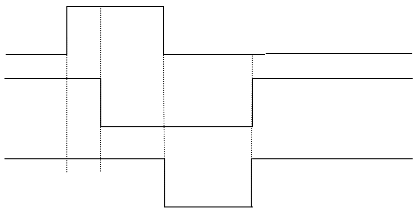
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Cronograma

A

B

F

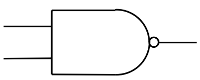


Puerta NAND

Símbolo

A

B



F

Ecuación

$$F = \overline{A \cdot B}$$

Tabla de Verdad

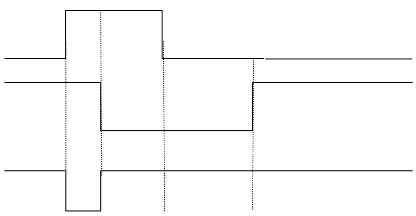
A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Cronograma

A

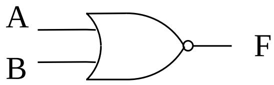
B

F



Puerta NOR

Símbolo



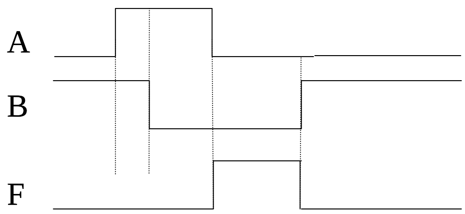
Ecuación

$$F = \overline{A + B}$$

Tabla de Verdad

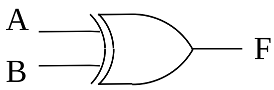
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Cronograma



Puerta XOR (Suma binaria)

Símbolo



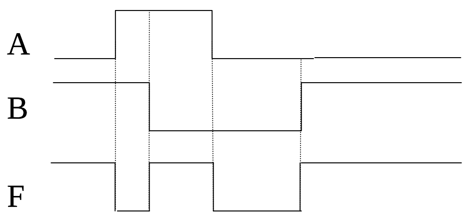
Ecuación

$$F = A \oplus B = \overline{A}B + A\overline{B}$$

Tabla de Verdad

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

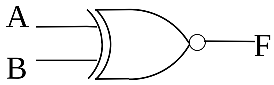
Cronograma



Dpto. de Sistemas Electrónicos y de Control

Puerta XNOR

Símbolo



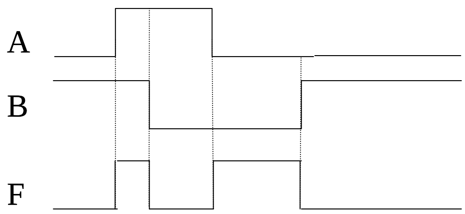
Ecuación

$$F = \overline{A \oplus B} = \overline{A}\,\overline{B} + AB$$

Tabla de Verdad

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Cronograma



I-29

Dpto. de Sistemas Electrónicos y de Control

Álgebra de Boole

Expression	**Dual**
$P2(a): a + 0 = a$	$P2(b): a \cdot 1 = a$
$P3(a): a + b = b + a$	$P3(b): ab = ba$
$P4(a): a + (b + c) = (a + b) + c$	$P4(b): a(bc) = (ab)c$
$P5(a): a + bc = (a + b)(a + c)$	$P5(b): a(b + c) = ab + ac$
$P6(a): a + \bar{a} = 1$	$P6(b): a \cdot \bar{a} = 0$
$T1(a): a + a = a$	$T1(b): a \cdot a = a$
$T2(a): a + 1 = 1$	$T2(b): a \cdot 0 = 0$
$T3: \quad \bar{\bar{a}} = a$	
$T4(a): a + ab = a$	$T4(b): a(a + b) = a$
$T5(a): a + \bar{a}b = a + b$	$T5(b): a(\bar{a} + b) = ab$
$T6(a): ab + a\bar{b} = a$	$T6(b): (a + b)(a + \bar{b}) = a$
$T7(a): ab + a\bar{b}c = ab + ac$	$T7(b): (a + b)(a + \bar{b} + c) = (a + b)(a + c)$
$T8(a): \overline{a + b} = \bar{a}\bar{b}$	$T8(b): \overline{ab} = \bar{a} + \bar{b}$
$T9(a): ab + \bar{a}c + bc = ab + \bar{a}c$	$T9(b): (a + b)(\bar{a} + c)(b + c) = (a + b)(\bar{a} + c)$

$T10(a): f(x_1, x_2, \ldots, x_n) = x_1 f(1, x_2, \ldots, x_n) + \bar{x}_1 f(0, x_2, \ldots, x_n)$

$T10(b): f(x_1, x_2, \ldots, x_n) = [x_1 + f(0, x_2, \ldots, x_n)][\bar{x}_1 + f(1, x_2, \ldots, x_n)]$

-30

Funciones Lógicas (I)

• Es una expresión algebraica de variables booleanas

• Dos funciones lógicas son equivalentes si tienen la misma tabla de verdad

• Pueden tener forma de expresión algebraica, tabla de verdad o cronograma.

Dpto. de Sistemas Electrónicos y de Control

Funciones Lógicas (y II)

Expresión algebraica

Y = (A+B) (C+D)

Cronograma

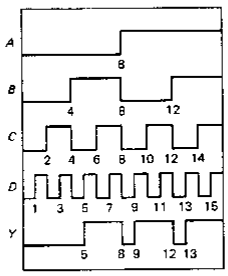


Tabla de Verdad

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Síntesis de Funciones Lógicas

1. Obtención de la expresión canónica:
   - Suma de productos (Miniterms)
   - Producto de sumas (Maxiterms)
2. Simplificación de funciones
   - Algebraica
   - Mapas de Karnaugh, otros

Dpto. de Sistemas Electrónicos y de Control

Suma de Productos

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

1. Se trabaja con los 1s de F
2. Cada 1 es un término producto de las variables de entrada
3. Si la variable X está a 1 $\rightarrow$ X
4. Si la variable X está a 0 $\rightarrow \overline{X}$

$$F = \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + AB\overline{C}$$

Producto de Sumas

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

1. Se trabaja con los 0s de F
2. Cada 0 es un término suma de las variables de entrada
3. Si la variable X está a 1 $\rightarrow \overline{X}$
4. Si la variable X está a 0 $\rightarrow$ X

$$F = (A + B + C) \cdot (A + \overline{B} + \overline{C}) \cdot (\overline{A} + \overline{B} + \overline{C})$$

Ejercicio

Hallar la ecuación de la siguiente tabla de verdad analizando por minterms y maxterms

X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Solución

Trabajando con 1s: $F = \overline{X}\,\overline{Y}\,\overline{Z} + X\overline{Y}Z + XY\overline{Z}$

Trabajando con 0s:

$F = (X+Y+\overline{Z})(X+\overline{Y}+Z)(X+\overline{Y}+\overline{Z})(\overline{X}+Y+Z)(\overline{X}+\overline{Y}+\overline{Z})$

$F = (X+Y+\overline{Z})(X+\overline{Y}+Z)(\overline{X}+Y+Z)(\overline{Y}+\overline{Z})$

$F = (X+YZ+\overline{Y}\,\overline{Z})(\overline{X}+Y+Z)(\overline{Y}+\overline{Z})$ **desarrollando:**

$F = \overline{X}\,\overline{Y}\,\overline{Z} + X\overline{Y}Z + XY\overline{Z}$

Simplificación Algebraica

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$F = A\overline{B}C + AB\overline{C} + ABC =$$

$$= A\overline{B}C + AB\cdot(C+\overline{C}) =$$

$$= A\overline{B}C + AB =$$

$$= A\cdot(\overline{B}C + B) =$$

$$= A\cdot(B+C)$$

P5(b)

P6(a)

P5(b)

T5(a)

Funciones incompletas (II)

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0

A	B	C	D	F
1	0	0	0	1
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	?
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

X

Funciones incompletas (y III)

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0

A	B	C	D	F
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

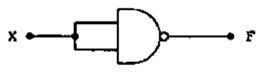
Se sustituyen las “X” por unos o ceros según convenga: mayor simplificación de la expresión lógica, etc.

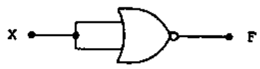
Conjuntos funcionalmente completos (I)

a) NOT y AND (NAND) b) NOT y OR (NOR)

Implementación de la función NOT

$$F = \overline{X} = \overline{X \cdot X} \qquad F = \overline{X} = \overline{X + X}$$

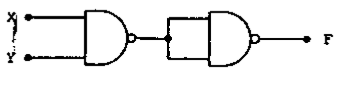


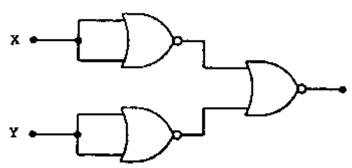


Conjuntos funcionalmente completos (II)

Implementación de la función AND

$$F = X \cdot Y = \overline{\overline{X \cdot Y}} \qquad F = X \cdot Y = \overline{\overline{X \cdot Y}} = \overline{\overline{X} + \overline{Y}}$$



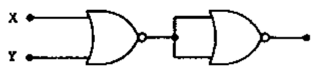


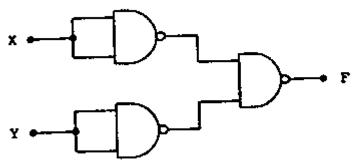
Conjuntos funcionalmente completos (III)

Implementación de la función OR

$$F = X + Y = \overline{\overline{X + Y}}$$

$$F = X + Y = \overline{\overline{X + Y}} = \overline{\overline{X} \cdot \overline{Y}}$$



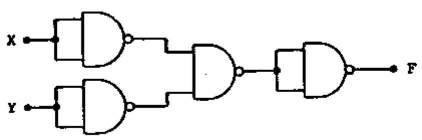


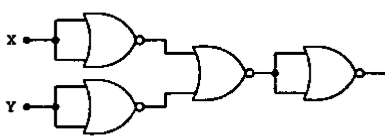
Conjuntos funcionalmente completos (IV)

Implementación de las funciones NAND y NOR

$$F = \overline{X + Y} = \overline{X} \cdot \overline{Y} = \overline{\overline{\overline{X} \cdot \overline{Y}}}$$

$$F = \overline{X \cdot Y} = \overline{X} + \overline{Y} = \overline{\overline{\overline{X} + \overline{Y}}}$$

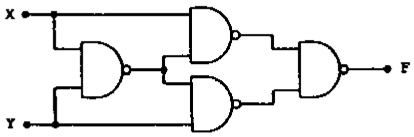




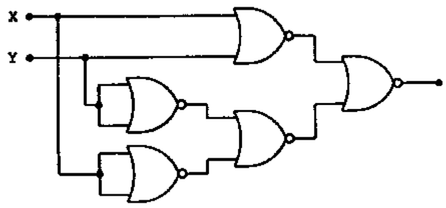
Conjuntos funcionalmente completos (y V)

Implementación de la función XOR

$$F = X \cdot \overline{Y} + \overline{X} \cdot Y =$$
$$= \overline{X} \cdot X + X \cdot \overline{Y} + \overline{Y} \cdot Y + \overline{X} \cdot Y =$$
$$= X(\overline{X} + \overline{Y}) + Y(\overline{X} + \overline{Y}) =$$
$$= X(\overline{X \cdot Y}) + Y(\overline{X \cdot Y}) =$$
$$= \overline{\overline{X(\overline{X \cdot Y})} \cdot \overline{Y(\overline{X \cdot Y})}}$$



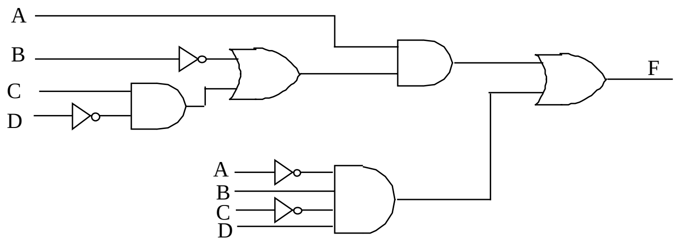
$$F = X \cdot \overline{Y} + \overline{X} \cdot Y =$$
$$= (X + Y)(\overline{X} + \overline{Y}) =$$
$$= \overline{\overline{(X + Y)(\overline{X} + \overline{Y})}} =$$
$$= \overline{\overline{(X + Y)} + \overline{(\overline{X} + \overline{Y})}}$$



Análisis de circuitos con puertas (I)

- Derivación de las funciones booleanas a partir del diagrama lógico
- Derivación directa de la tabla de verdad
- Simulación lógica

Análisis de circuitos con puertas (II)



$$F = A(\overline{B} + C\overline{D}) + \overline{A}B\overline{C}D$$

Dpto. de Sistemas Electrónicos y de Control

Análisis de circuitos con puertas (III)

$$F = A(\overline{B} + C\overline{D}) + \overline{A}B\overline{C}D$$

Es 1 cuando:

• A = 0, B = 1, C = 0 y D = 1 ,

• A = 1 y :
  B = 0 ó
  C = 1 y D = 0

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Análisis de circuitos con puertas (y IV)

$$F = A(\overline{B} + C\overline{D}) + \overline{A}B\overline{C}D =$$
$$= A\overline{B} + AC\overline{D} + \overline{A}B\overline{C}D =$$
$$= A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}D + A\overline{B}C\overline{D} +$$
$$+ A\overline{B}CD + A\overline{B}C\overline{D} + ABC\overline{D} +$$
$$+ \overline{A}B\overline{C}D$$

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Metodología de diseño de circuitos con puertas lógicas (I)

El diseño comienza por la **especificación del problema** y culmina con un **diagrama lógico** o un grupo de **ecuaciones booleanas** a partir de las cuales obtener el diagrama lógico.

Se trata de una metodología sistemática compuesta por un conjunto de pasos.

Metodología de diseño de circuitos con puertas lógicas (II)

Pasos que comprende el procedimiento a seguir:

1. Determinar el número de entradas y salidas requeridas, a partir de las especificaciones dadas.
2. Codificar la información de entrada y de salida, caso de ser necesario.
3. Obtener la relación funcional entre las entradas y las salidas.
4. Expresar esta relación funcional en modo de tabla de verdad.
5. Obtener la expresión algebraica que responde a la relación funcional establecida con anterioridad.
6. Simplificar, por el método que sea, esa expresión algebraica.
7. Implementar mediante puertas lógicas la expresión algebraica que cumple las especificaciones dadas.
8. Verificar la corrección del diseño realizado.

Metodología de diseño de circuitos con puertas lógicas (III)

Ejemplo: Especificación

Se desea diseñar un circuito digital que detecte el exceso de peso de dos tipos de paquetes.

A través de una señal digital se nos indicará el tipo de paquete: grande o pequeño.

Se nos ofrecerá el peso del paquete codificado en binario: un número entero comprendido entre 0 y 7 Kg.

El sistema deberá indicar el sobrepeso cuando un paquete pequeño pese más de tres kilos o cuando un paquete grande pese más de 5Kg.

Metodología de diseño de circuitos con puertas lógicas (IV)

Ejemplo: Número de Entradas y Salidas

- Se necesita una entrada digital (T) para indicar el tipo de paquete: 0 (paquete pequeño) y 1 (paquete grande)
- Se necesitan tres señales digitales para indicar el peso del paquete:

C B A	
0 0 0	0Kg
0 0 1	1Kg
0 1 0	2Kg
0 1 1	3Kg
1 0 0	4Kg
1 0 1	5Kg
1 1 0	6Kg
1 1 1	7Kg

- Se necesita una salida (S) para indicar la existencia o no de sobrepeso.

Metodología de diseño de circuitos con puertas lógicas (V)

Ejemplo: Tabla de Verdad y Expresión Booleana

T	C	B	A	S
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

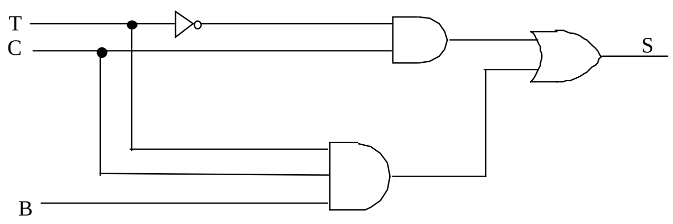
$$F = \bar{T}\cdot C\cdot \bar{B}\cdot \bar{A} + \bar{T}\cdot C\cdot \bar{B}\cdot A + \bar{T}\cdot C\cdot B\cdot \bar{A} +$$

$$+ \bar{T}\cdot C\cdot B\cdot A + T\cdot C\cdot B\cdot \bar{A} + T\cdot C\cdot B\cdot A$$

$$= \bar{T}\,C + TCB = C(\bar{T} + B)$$

Metodología de diseño de circuitos con puertas lógicas (VI)

Ejemplo: Circuito Digital



Dpto. de Sistemas Electrónicos y de Control

Metodología de diseño de circuitos con puertas lógicas (y VII)

Limitaciones de esta metodología

- Circuitos con un número pequeño de entradas y salidas.
- Circuitos de baja complejidad.
- Circuitos combinacionales.