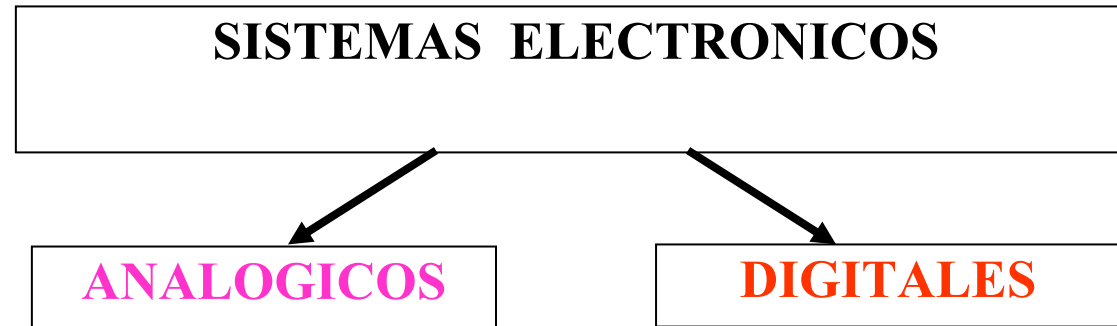


INTRODUCCION A LAS TECNICAS DIGITALES 1



DIGITALES: *INFORMACION CODIFICADA EN DOS UNICOS ESTADOS

*SE BASA EN EL ALGEBRA BOOLEANA

*LOS SISTEMAS DIGITALES PUEDEN CLASIFICARSE EN:

- SISTEMAS CABLEADOS

COMBINACIONALES

SECUENCIALES

MEMORIAS

CONVERTIDORES

SISTEMAS PROGRAMADOS

DISP LÓGICOS PROG

MICROPROCESADORES

MICROCONTROLADORES

PROGRAMA TECNICAS DIGITALES 1

CAPÍTULO 1: ÁLGEBRA DE BOOLE

Teoremas y Postulados

CAPÍTULO 2: FUNCIONES LÓGICAS y MINIMIZACIÓN

Funciones Canónicas – T:V – Formas Standard

Conceptos – Métodos algebraicos y Gráficos

CAPÍTULO 3: SISTEMA DE NUMERACIÓN Y CÓDIGOS

**Diversos sistemas de Numeración – Códigos
detectores y correctores**

CAPÍTULO 4: ARITMÉTICA BINARIA

Sumadores – Restadores – Comparadores – ALU

CAPÍTULO 5: DECODIFICADORES Y DEMUX

Diseño y síntesis circuitales

PROGRAMA TECNICAS DIGITALES 1

CAPÍTULO 6: TECNOLOGÍA

Flia CMOS –PLD – Display

CAPÍTULO 7: CONTADORES Y REGISTROS

Biestables – Contadores – Registro –

CAPÍTULO 8: CIRCUITOS DE TIEMPO

Diseño de osciladores – Temporizadores –

CAPÍTULO 9: CIRCUITOS SECUENCIALES

Diseño y Síntesis de Autómatas

CAPÍTULO 10: MEMORIAS Y BUSES

NOTA: En cada uno de los capitulos se va introduciendo **VHDL**

La simulación y Síntesis se realiza con el programa

BOOLE-DEUSTO

BIBLIOGRAFIA

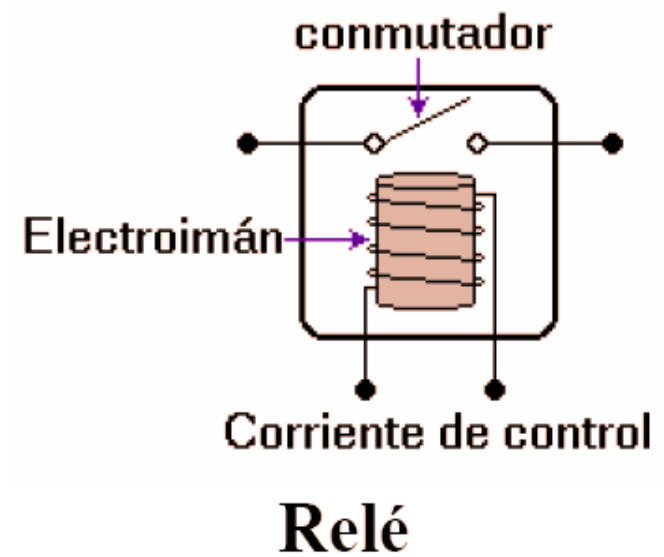
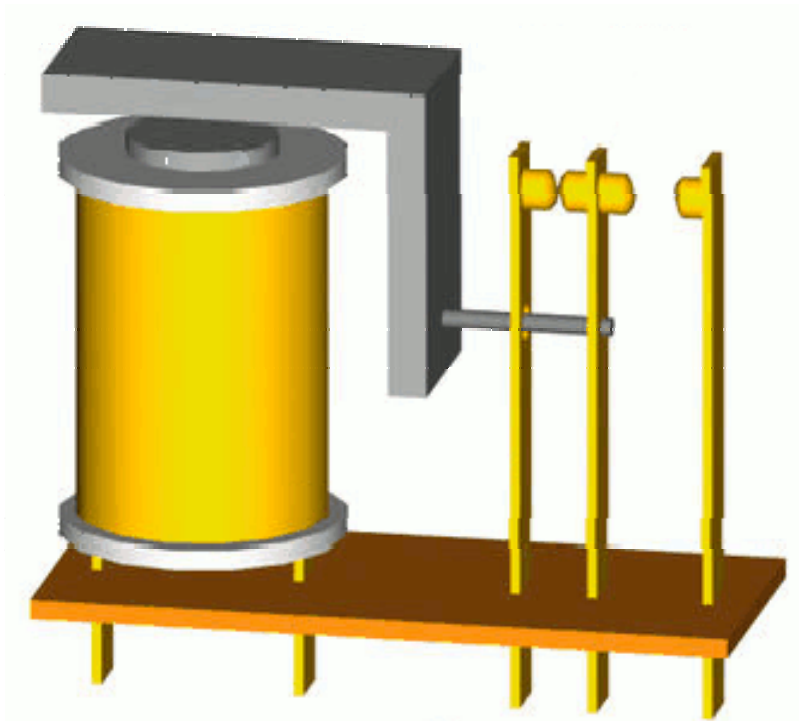
- TECNICAS DIGITALES I : Rodolfo A Cavallero**
- DISEÑO LOGICO DIGITAL CON VHDL : S. Brown – Z Vranesic**
- DISEÑO DE SISTEMAS DIGITALES CON VHDL : S.Perez –E. Soto**
- DISEÑO DIGITAL, PRINCIPIOS Y PRACTICAS : John Wakerly**
- GUIAS DE ESTUDIO DE LA CATEDRA : Página Web de la Cátedra**
- CUADERNILLOS DE CATEDRA: Autogestión**

CRONOGRAMA DE DESARROLLO DE ACTIVIDADES ACADÉMICAS

UNIDAD	TEMA	Clase Nro.															16
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	Álgebra de Boole	x	x														
2	Funciones Logicas y su minimización			x	x	x											
3	Sistemas de Numeración y Códigos					x	x	x									
4	Aritmética Binaria								x	x							
5	Codificadores-Decodificadores Multiplexores/Demux										x	x	x				
6	Tecnología													x	x	x	x

UNIDAD	TEMA	Clase Nro.																
		17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
7	Biestables, Contadores y Registros	x	x	x	x													
8	Circuitos de tiempo					x	x	x	x									
9	Circuitos Secuenciales									x	x	x	x					
10	Memorias y Estructura de Buses													x	x	x	R	

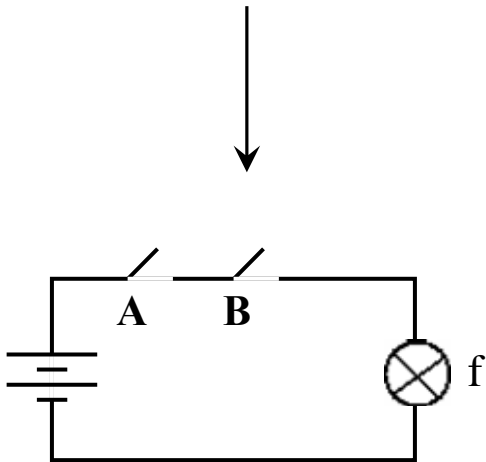
FUNCIONAMIENTO RELE



SISTEMAS DIGITALES

CONBINACIONALES

SECUENCIALES



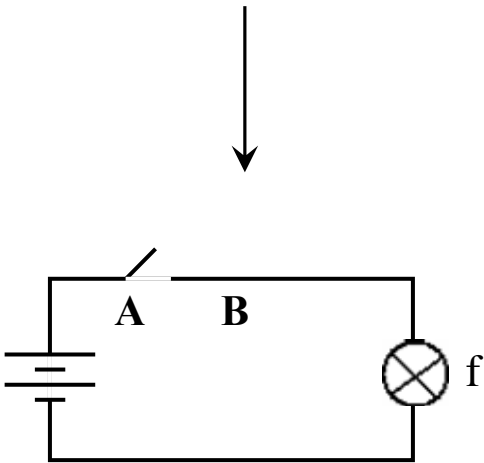
→

A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SISTEMAS DIGITALES

CONBINACIONALES

SECUENCIALES




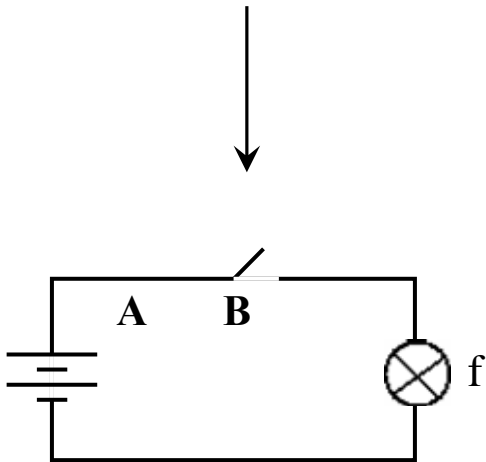
→

A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SISTEMAS DIGITALES

CONBINACIONALES

SECUENCIALES




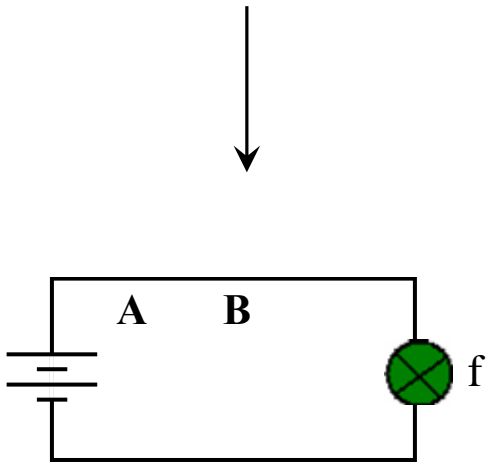
A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SISTEMAS DIGITALES

CONBINACIONALES




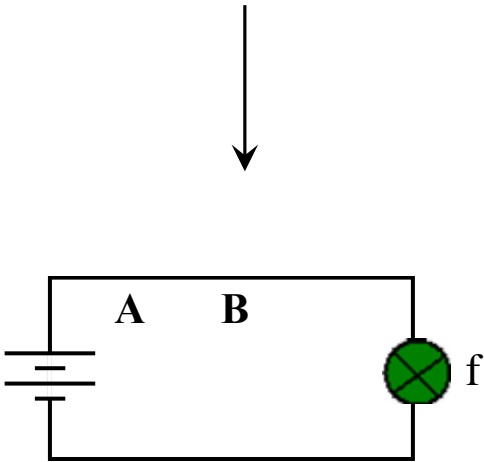
SECUENCIALES



A	B	f
0	0	0
0	1	0
1	0	0
1	1	1


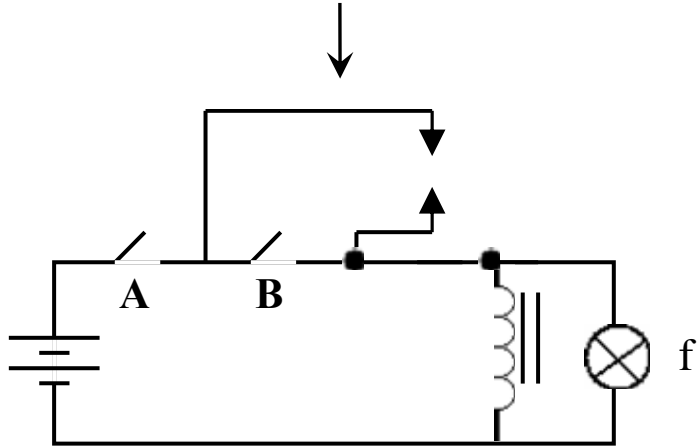
SISTEMAS DIGITALES

CONBINACIONALES




A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SECUENCIALES


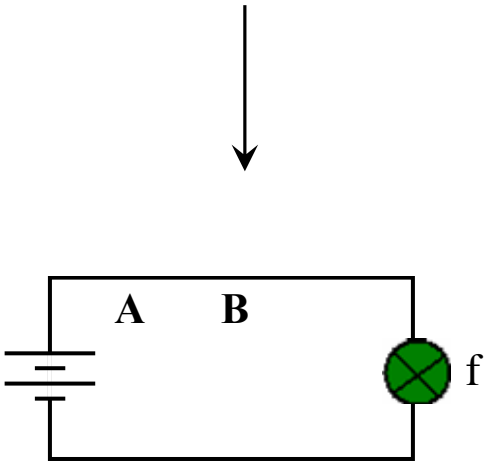


A	B	f
0	0	0
0	1	0
1	0	0
1	1	1




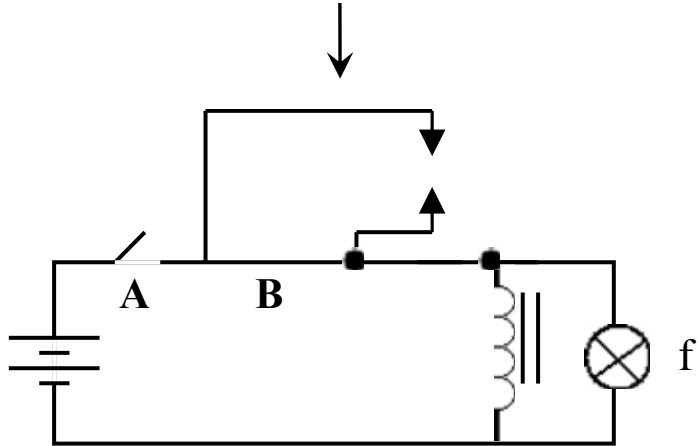
SISTEMAS DIGITALES

CONBINACIONALES




A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SECUENCIALES


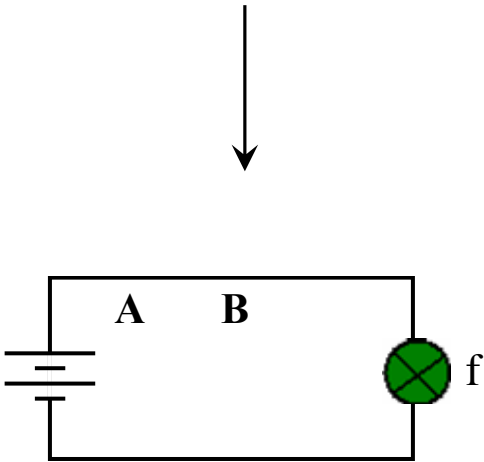


A	B	f
0	0	0
0	1	0
1	0	0
1	1	1




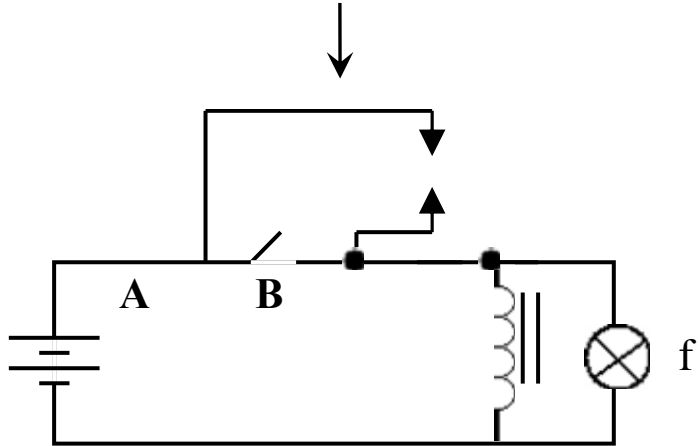
SISTEMAS DIGITALES

CONBINACIONALES




A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SECUENCIALES


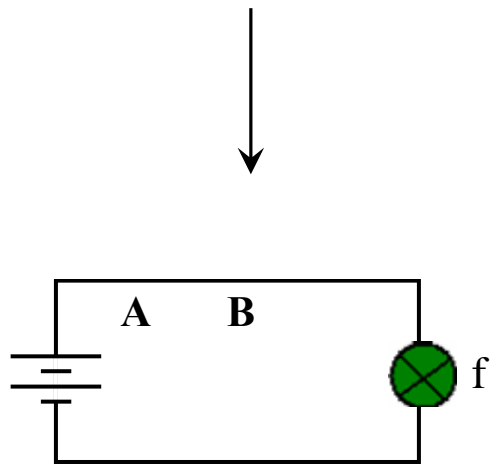


A	B	f
0	0	0
0	1	0
1	0	0
1	1	1




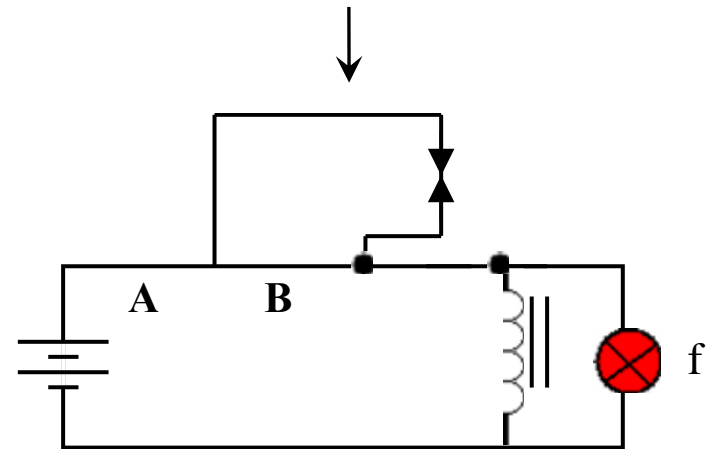
SISTEMAS DIGITALES

CONBINACIONALES




A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SECUENCIALES

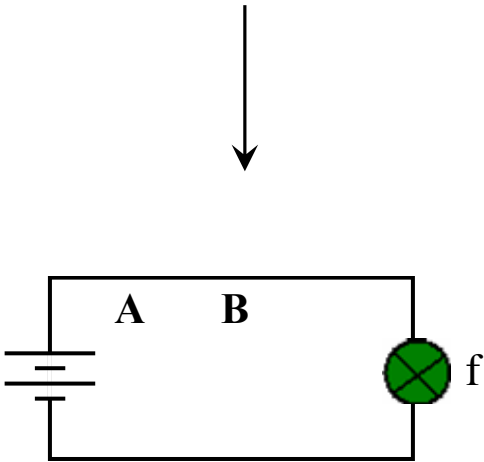


A	B	f
0	0	0
0	1	0
1	0	0
1	1	1



SISTEMAS DIGITALES

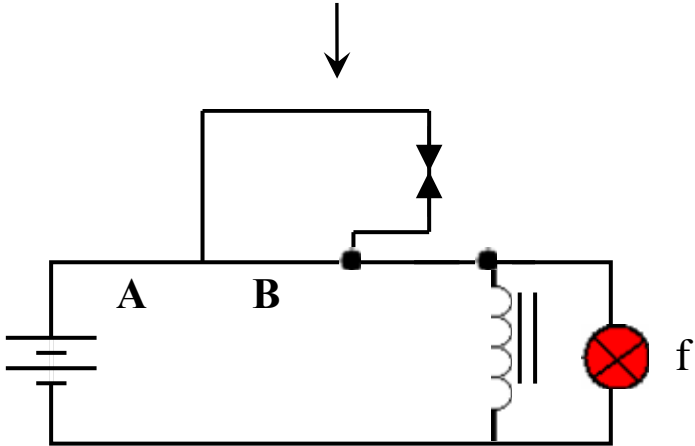
CONBINACIONALES



→

A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SECUENCIALES




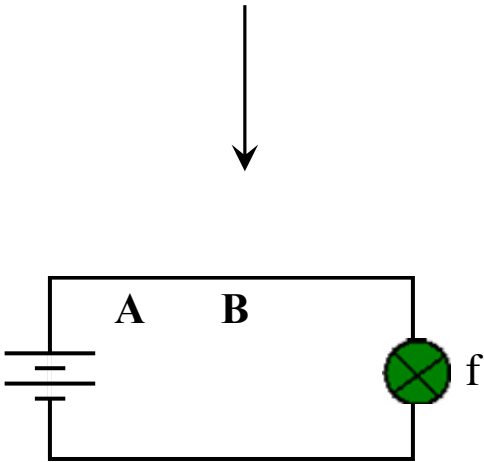
A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

→

f
0
0
1
1

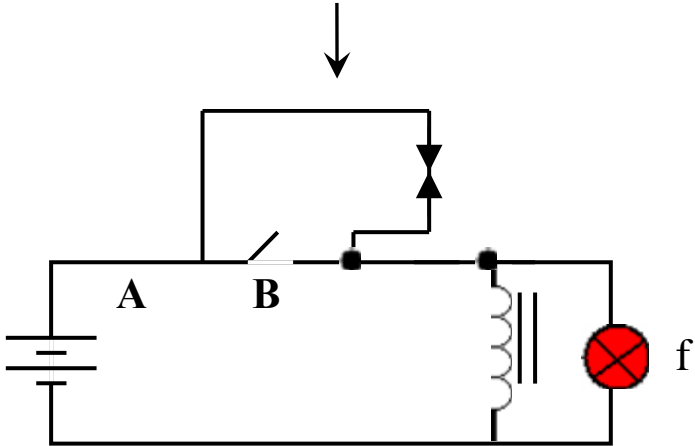
SISTEMAS DIGITALES

CONBINACIONALES




A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SECUENCIALES



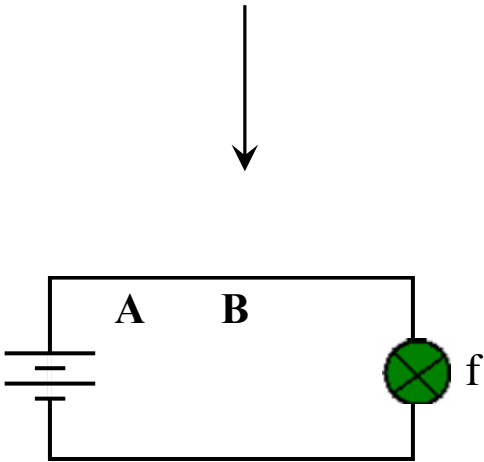
A	B	f
0	0	0
0	1	0
1	0	0
1	1	1



f
0
0
1
1

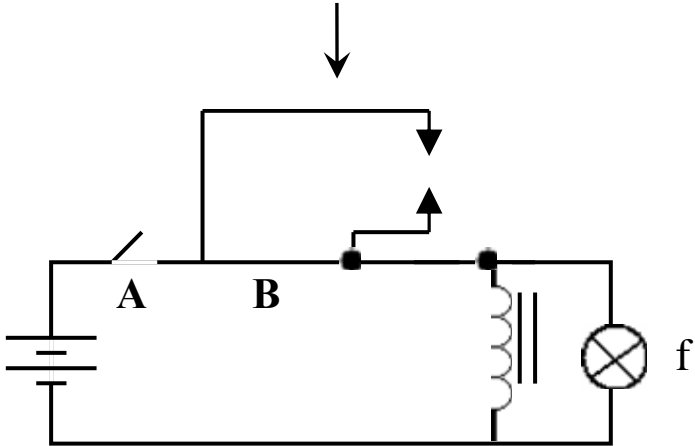
SISTEMAS DIGITALES

CONBINACIONALES



A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SECUENCIALES


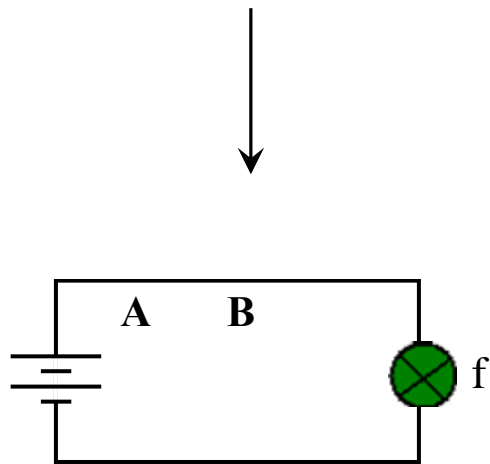


A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

f
0
0
1
1

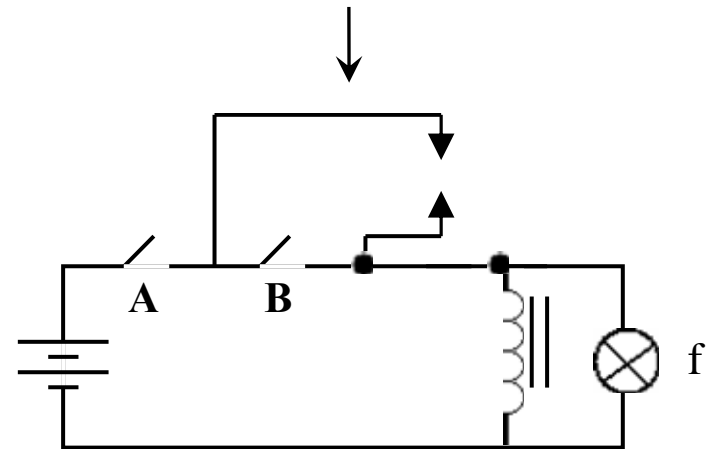
SISTEMAS DIGITALES

CONBINACIONALES




A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

SECUENCIALES



A	B	f
0	0	0
0	1	0
1	0	0
1	1	1



f
0
0
1
1

ALGEBRA DE BOOLE

CLASE: Es un conjunto compuesto por **ELEMENTOS** susceptibles de poseer ciertas propiedades y tener entre ellos, o con elementos de otros conjuntos, ciertas relaciones

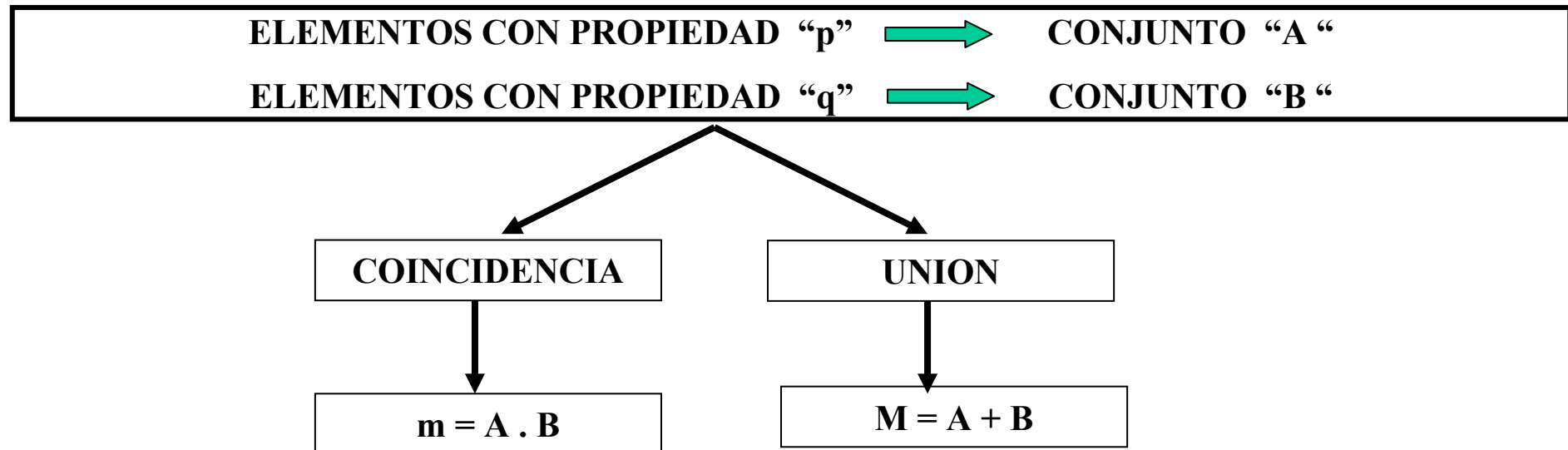
CLASE UNIVERSAL : ($U = 1$)

Es el conjunto de todos los elementos que se van a tratar,

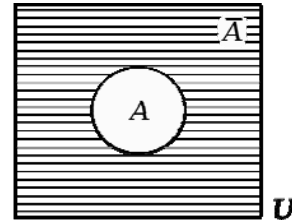
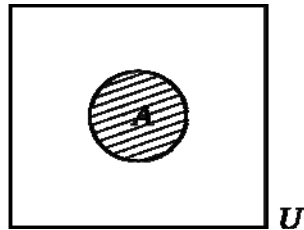
CLASE VACIA: (0)

No contiene ningun elemento

OPERACIONES BOOLEANAS

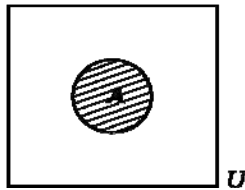


DIAGRAMAS DE VENN (Definamos un Universo)

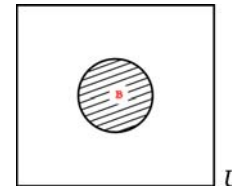


DEFINAMOS DOS CONJUNTOS “A” Y “B”

“A” → HOMBRES BAJOS

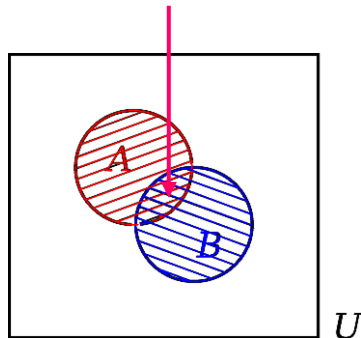


“B” → HOMBRES OBESOS



COINCIDENCIA

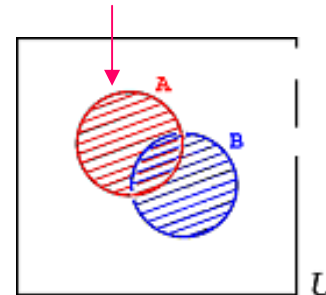
$$m = A \cdot B$$



$m =$ BAJOS Y OBESOS

UNION

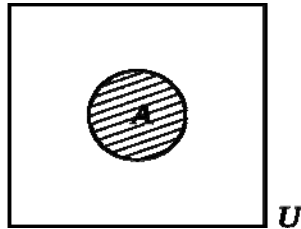
$$M = A + B$$



$M =$ BAJOS O OBESOS

POSTULADOS DEL ALGEBRA DE BOOLE

DIAGRAMA DE VENN



$$A + 1 = 1$$

$$A + 0 = A$$

$$A + \bar{A} = 1$$

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot \bar{A} = 0$$

$$\bar{\bar{A}} = A$$

(DUALIDAD)

PROPIEDAD CONMUTATIVA

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

PROPIEDAD DE INVARIANCIA

$$A + 0 = A$$

$$A \cdot 1 = A$$

PROPIEDAD DISTRIBUTIVA

$$A(B + C) = AB + AC$$

$$A + BC \dots N = (A+B)(A+C) \dots (A+N)$$

PROPIEDAD DEL COMPLEMENTO

$$A + \bar{A} = 1 = U$$

$$A \cdot \bar{A} = 0$$

TEOREMAS DEL ALGEBRA DE BOOLE

PRINCIPIO DE DUALIDAD: SE OBTIENE UNA EXPRESION DUAL SI:

(COMPROBAR EN POSTULADOS)

$(+) \longleftrightarrow (.)$

$(0) \longleftrightarrow (1)$

ABSORCION

$$A + ABCD\dots N = A$$

$$A . (A+B+C+\dots N) = A$$

ASOCIATIVA

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

$$A.B.C = (A.B).C$$

NEGACION

$$\overline{\overline{A}} = A$$

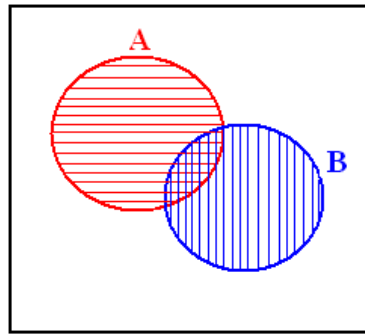
$$\overline{0} = 1, \quad \overline{1} = 0$$

DE MORGAN

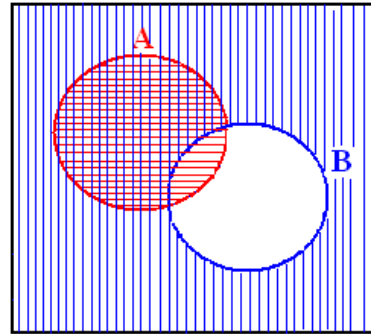
$$\overline{A + B} = \overline{A} . \overline{B}$$

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

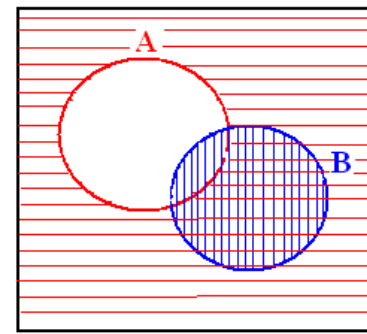
TEOREMA DE DEMORGAN → VENN



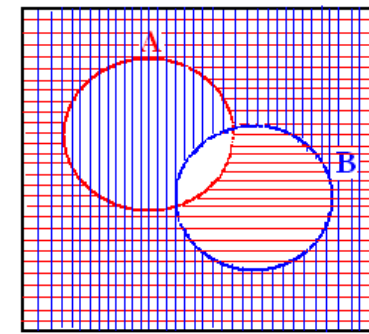
$$m_3 = A \cdot B$$



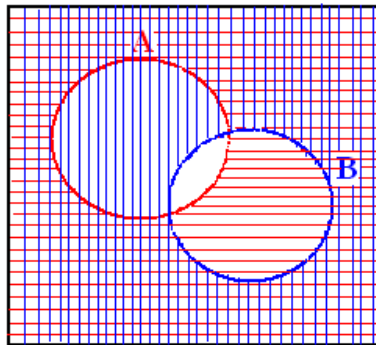
$$m_2 = A \cdot \bar{B}$$



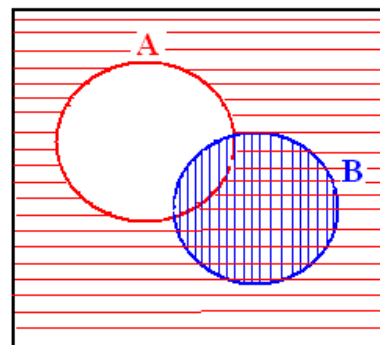
$$m_1 = \bar{A} \cdot B$$



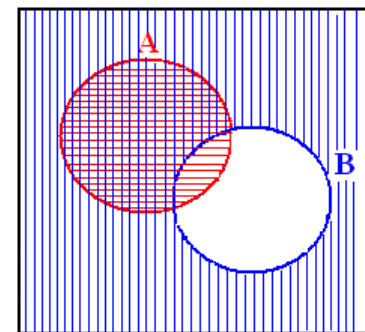
$$m_0 = \bar{A} \cdot \bar{B}$$



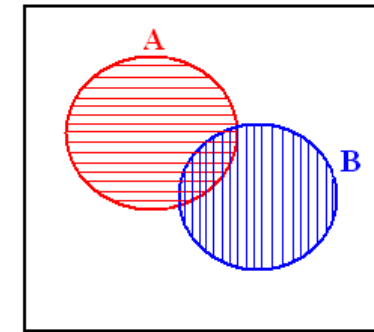
$$M_0 = \bar{A} + \bar{B}$$



$$M_1 = \bar{A} + B$$

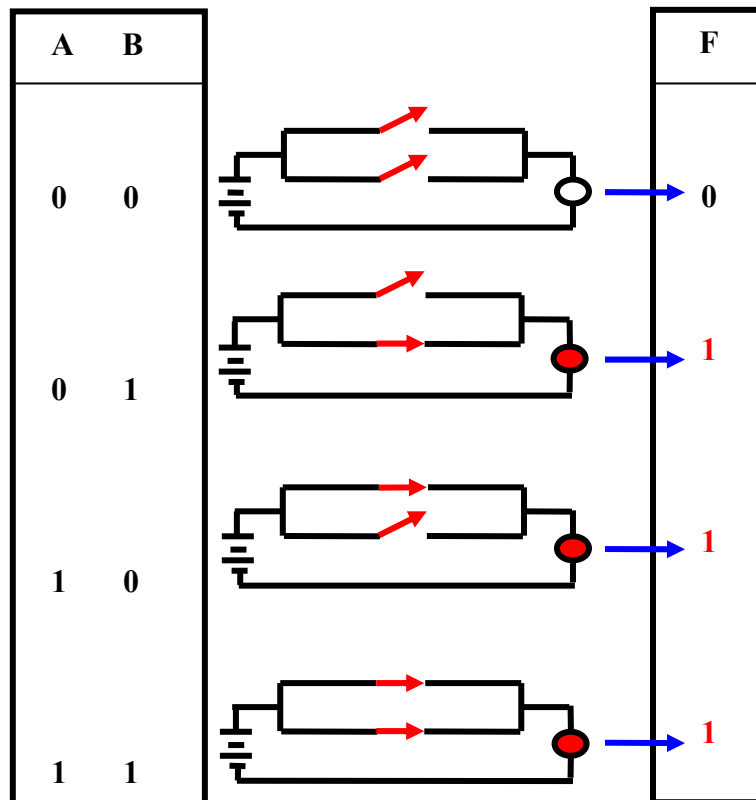
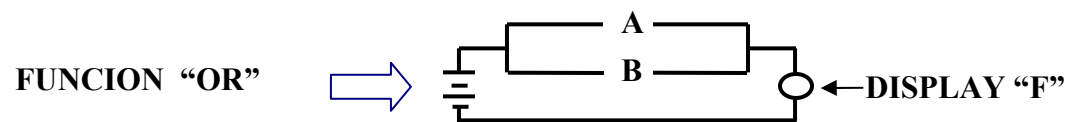


$$M_2 = A + \bar{B}$$

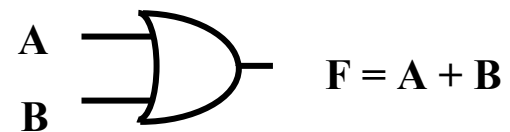


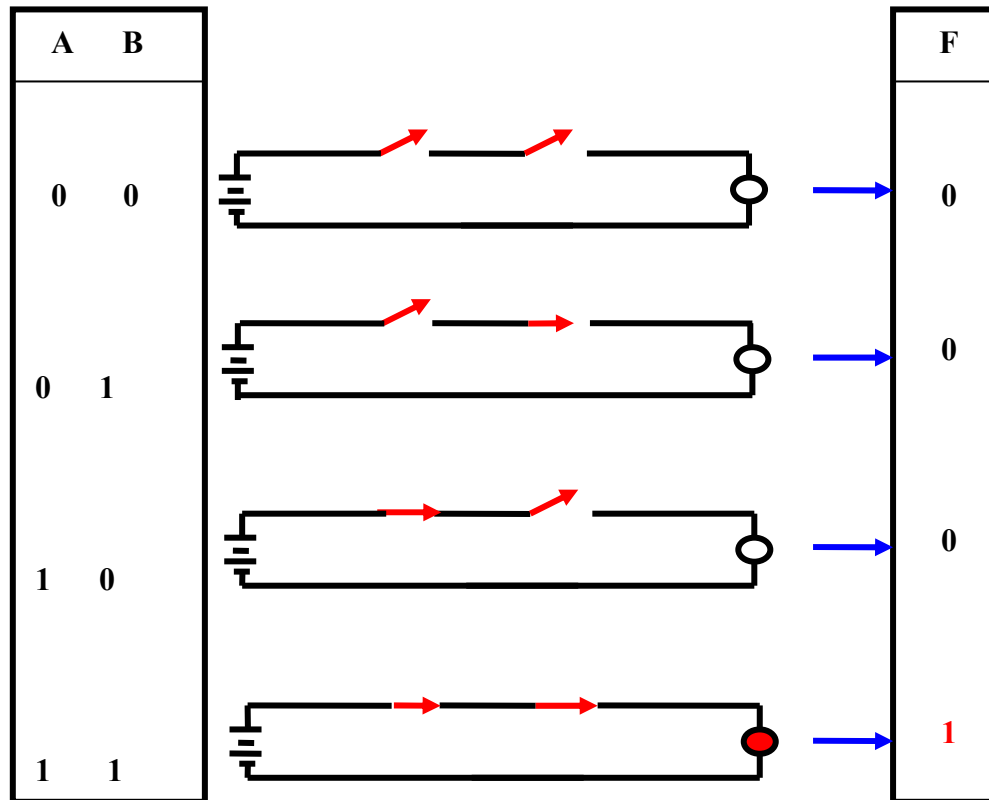
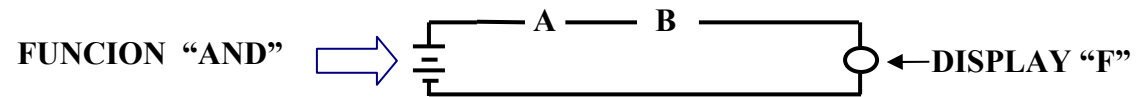
$$M_3 = A + B$$

Observe: $\overline{m_3} = M_0$ con lo que $\overline{A \cdot B} = \bar{A} + \bar{B}$

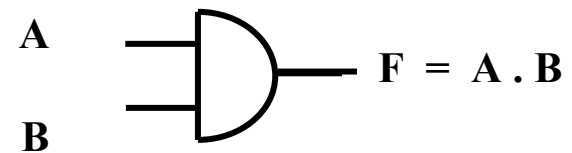


SIMBOLO LÓGICO "OR"

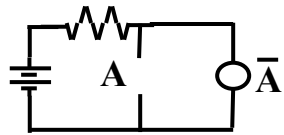




SIMBOLO LOGICO AND



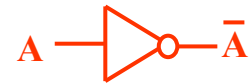
CIRCUITO NOT



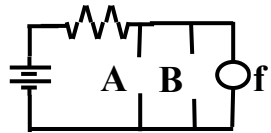
TV

A	f
0	1
1	0

SIMBOLO LOGICO - NOT



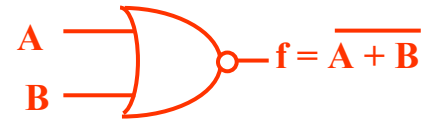
CIRCUITO NOR



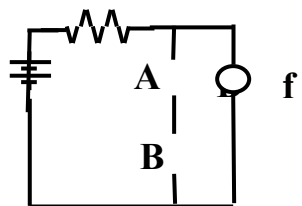
TV

A	B	f
0	0	1
0	1	0
1	0	0
1	1	0

SIMBOLO LOGICO - NOR



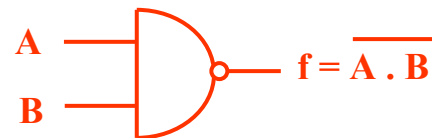
CIRCUITO NAND



TV

A	B	f
0	0	1
0	1	1
1	0	1
1	1	0

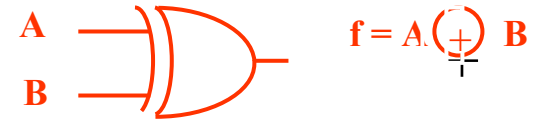
SIMBOLO LOGICO -NAND



TV

A	B	f
0	0	0
0	1	1
1	0	1
1	1	0

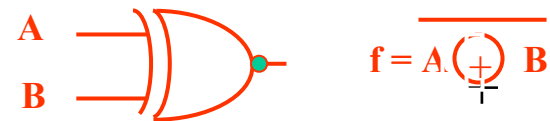
SIMBOLO LOGICO –O-EXCLUSIVA



TV

A	B	f
0	0	1
0	1	0
1	0	0
1	1	1

**SIMBOLO LOGICO –O-EXCLUSIVA NEGADA
(COMPARADOR DE IGUALDAD)**



VHDL

El lenguaje de programación **VHDL** (**V**ery **H**igh Speed Integrated Circuit **H**ardware **D**escription **L**anguage) es un lenguaje que describe el comportamiento del circuito, es decir describe el hardware

En la Fig 1 se observan los tres estilos de descripción

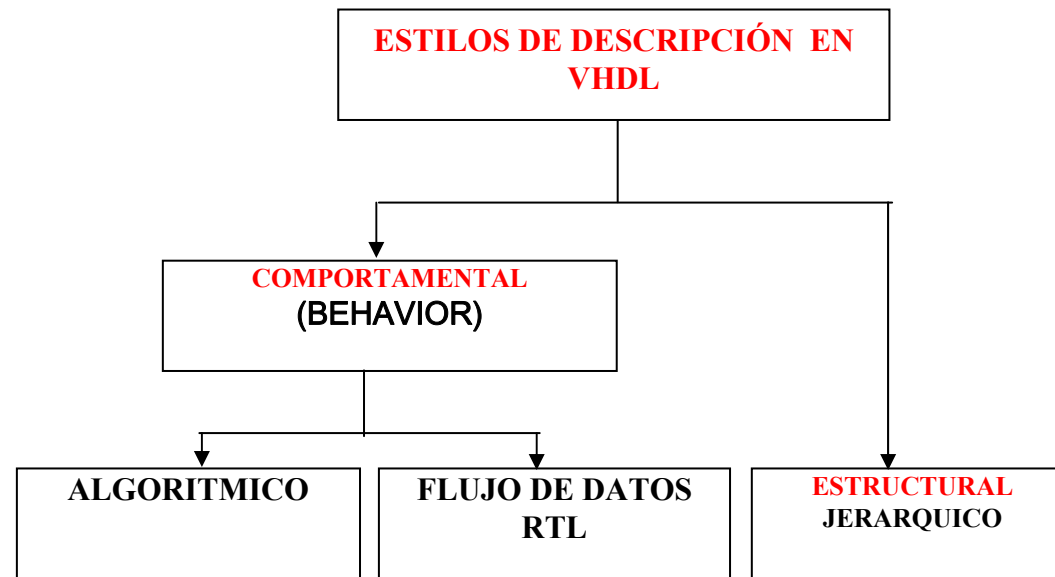


Fig. 1 Estilos de descripción VHDL

VHDL – Lenguaje para síntesis y modelado de circuitos – Fernando Pardo y Jose Boluda
Editorial Alfaomega

VHDL - David Maxinez - Editorial C.E.C.S.A

Diseño de sistemas con VHDL – Editorial Paraninfo

VHDL

ESTRUCTURA BASICA DE UN ARCHIVO FUENTE VHDL

ENCABEZAMIENTO	Library <nombre_libreria> Use <nombre_librería>.<nombre_paquete>. all
ENTIDAD	Entity <nombre_entidad> is <listado de puertos> --Declaración de pines end <nombre_entidad>;
ARQUITECTURA	Architecture <nombre_arquitectura> of <nombre_entidad> is --Declaracion de señales internas --Declaracion de tipos de datos definidos por el usuario --Declaracion de componentes en caso de instanciación begin --Cuerpo de la arquitectura --Se define la funcionalidad del diseño con: --Asignaciones concurrentes --Procesos --Instanciación de componentes end <nombre_arquitectura>;

VHDL

ENTIDAD Y ARQUITECTURA

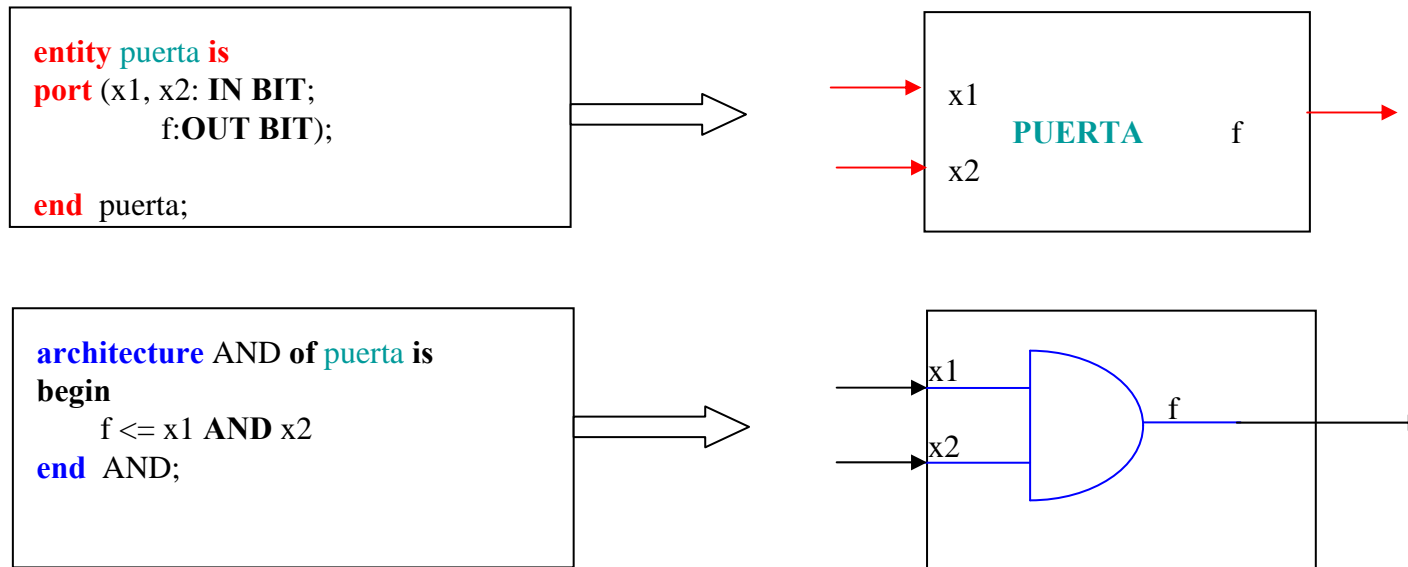


Fig. 4 Entidad y Arquitectura

Metodología para la resolución de problemas lógicos

Diseño de un circuito combinacional

Cuando se diseña un circuito lógico combinacional, debemos realizar los siguientes pasos :

1. Construcción de la Tabla de Verdad.
2. Desarrollo de la expresión lógica
3. simplificación de la expresión lógica
4. Implementación eficiente

Ejemplo 1: SISTEMA DE ALARMA

Un supermercado tiene dos cajeros y cada uno de ellos dispone de un interruptor de pedal conectado a un sistema de alarma. Cuando se pulsa uno de estos interruptores, o ambos, se prende una luz en la gerencia. Plantee el sistema lógico correspondiente.

Definición de las variables de entrada y su comportamiento.

Cajero 1 Interruptor "A"

A=0 interruptor abierto.

A=1 interruptor cerrado.

Cajero 2 Interruptor "B"

B=0 interruptor abierto.

B=1 interruptor cerrado.

2 Definición de las variables de salida y su comportamiento.

Luz de gerencia Lámpara “F”

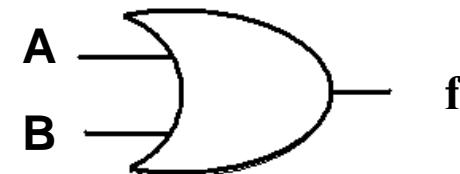
f=0 alarma desactivada.

f=1 alarma activada

3. Tabla de verdad y expresión lógica

Entradas		Salida
A	B	f
0	0	0
0	1	1
1	0	1
1	1	1

4 Implementación



METODOS DE OBTENCION DE LA FUNCIÓN COMO SUMATORIA DE MINTERM

Para obtener una función de la TV se suman los MINTERM que hacen uno (1) la función y se los suma. Volvamos al sistema de alarma:

mi	A	B	f
m0	0	0	0
m1	0	1	1
m2	1	0	1
m3	1	1	1

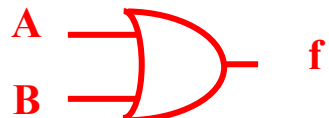
$$\begin{aligned}
 f &= \bar{A}.B + A.\bar{B} + A.B \\
 &= m1 + m2 + m3 \\
 &= \sum 1, 2, 3
 \end{aligned}$$

SIMPLIFICANDO: $f = \bar{A}.B + A.\bar{B} + A.B$

$$= A(B + \bar{B}) + \bar{A}.B \longrightarrow P2$$

$$= A + \bar{A}.B \longrightarrow P4$$

$$= (A + \bar{A}). (A + B) \longrightarrow P2$$

$$f = A + B \longrightarrow$$


COMO PRODUCTO DE MAXTERM

Para obtener una función expresada en MAXTERM se toman los términos que hacen **cero (0)** la TV y se niegan sus variables (*Luego a través de la función complemento lo vamos a demostrar*).

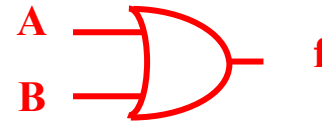
Veamos el ejemplo del sistema de alarma:

Mi	A	B	f
M3	0	0	0
M2	0	1	1
M1	1	0	1
M0	1	1	1



Se niegan las variables

$$f = \overline{\overline{A}} + \overline{\overline{B}} = A + B = M3$$



FUNCION COMPLEMENTO: \overline{f}

$\overline{f} = \Sigma m_i \longrightarrow$ QUE HACEN CERO LA FUNCIÓN
 $\overline{\overline{f}} = f = \Pi M_j$

Ejemplo:

Mj	mi	A	B	C	f
M7	m0	0	0	0	0
M6	m1	0	0	1	1
M5	m2	0	1	0	1
M4	m3	0	1	1	0
M3	m4	1	0	0	1
M2	m5	1	0	1	0
M1	m6	1	1	0	0
M0	m7	1	1	1	1

$$\overline{f} = m0 + m3 + m5 + m6$$

NEGANDO EL COMPLEMENTO

$$\overline{\overline{f}} = \overline{m0 + m3 + m5 + m6}$$

POR DE MORGAN

$$f = \overline{m0} \cdot \overline{m3} \cdot \overline{m5} \cdot \overline{m6}$$

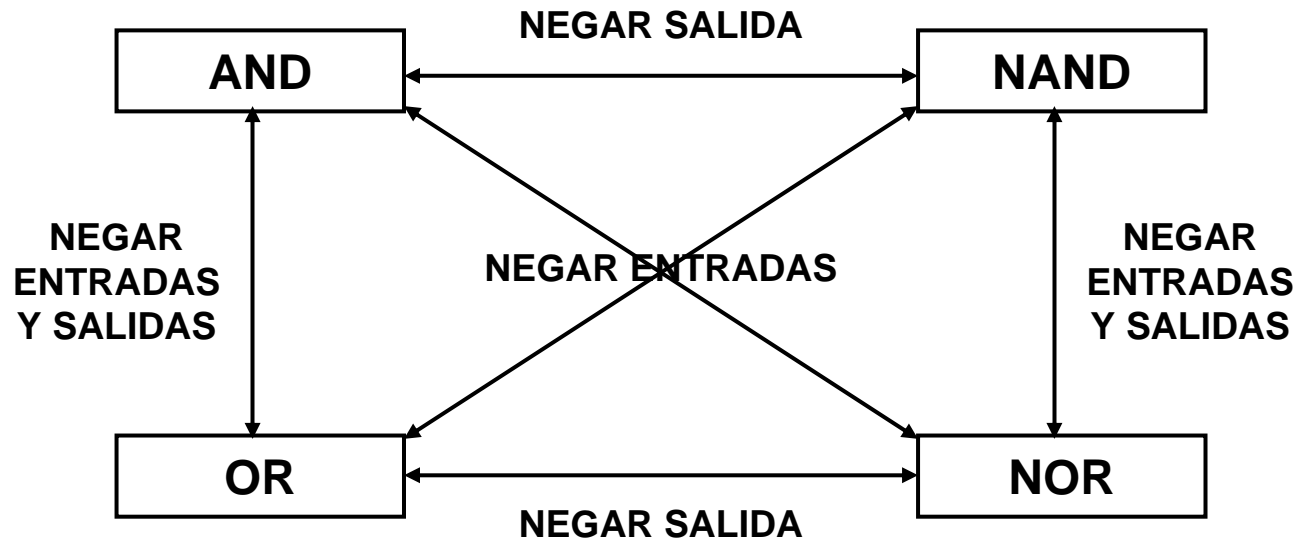
$$f = (\overline{\overline{A.B.C}}) (\overline{\overline{A.B.C}}) (\overline{\overline{A.B.C}}) (\overline{\overline{A.B.C}})$$

APLICANDO DE MORGAN NUEVAMENTE

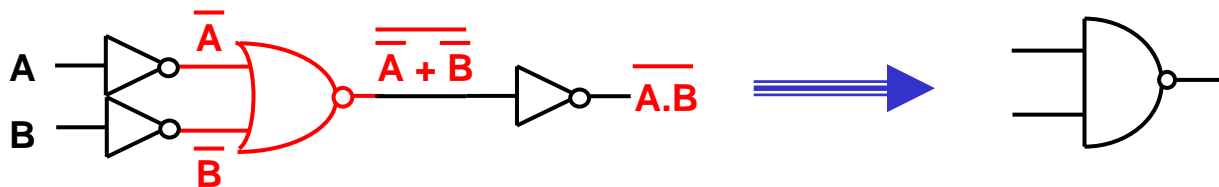
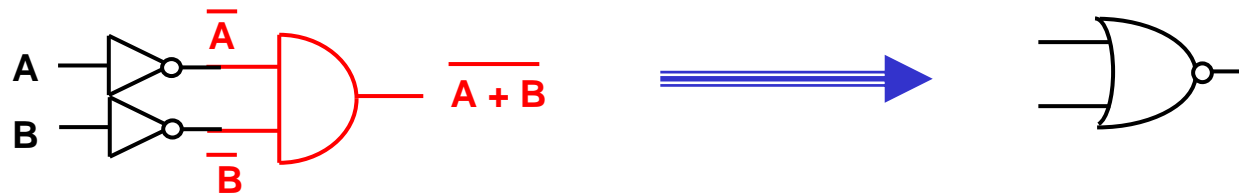
$$f = (A + B + C) (A + \overline{B} + \overline{C}) (\overline{A} + B + \overline{C}) (\overline{A} + \overline{B} + C)$$

$$f = M7 \cdot M4 \cdot M2 \cdot M1$$

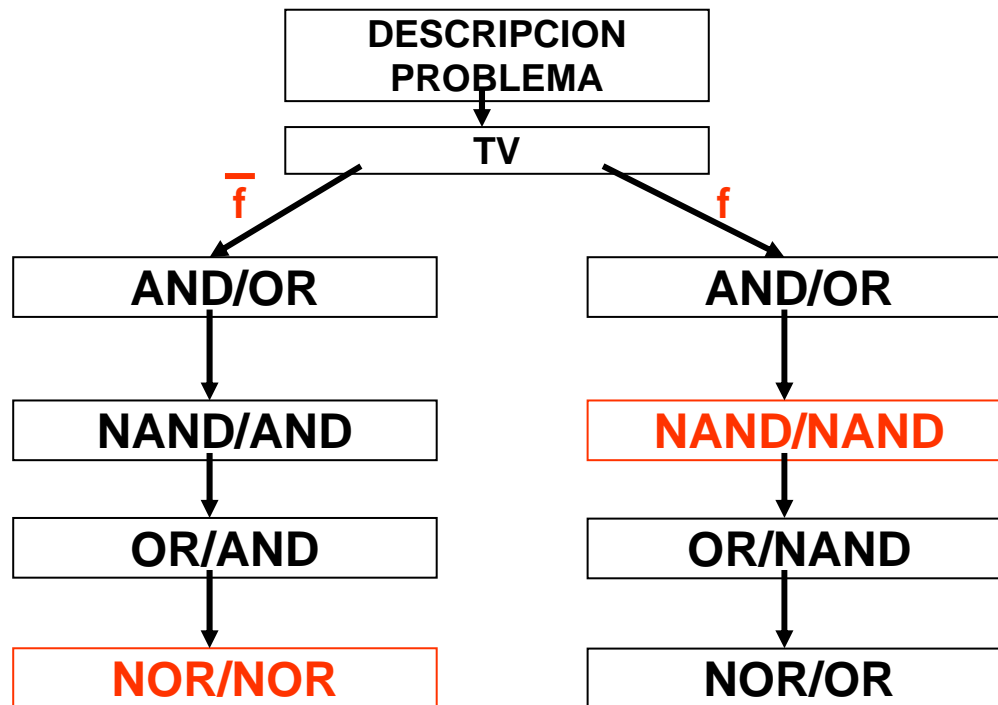
RELACIÓN ENTRE LOS OPERADORES LÓGICOS



EJEMPLOS



FORMAS STANDART DE UNA FUNCIÓN



EJEMPLO: $f = m1 + m2$

$$= \bar{A}B + A\bar{B} \quad \longrightarrow \text{AND/OR}$$

$$f = f = \overline{\bar{A}B + A\bar{B}}$$

$$f = \overline{\bar{A}B} \cdot \overline{A\bar{B}} \quad \longrightarrow \text{NAND/NAND}$$

$$f = \overline{(A + B)(\bar{A} + \bar{B})} \quad \longrightarrow \text{OR/NAND}$$

$$f = \overline{(A + B)} + \overline{(\bar{A} + \bar{B})} \quad \longrightarrow \text{NOR/OR}$$

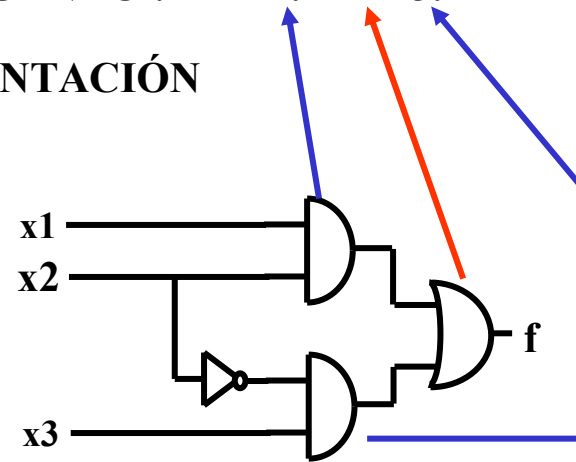
EJEMPLOS OBTENCION FUNCIONES Y VHDL

X3	X2	X1	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$$f = m3 + m4 + m5 + m7$$

$$\text{SIMPLIFICANDO : } f = x2.x1 + x3.\overline{x2}$$

IMPLEMENTACIÓN



A continuación escribimos el código VHDL correspondiente

ENTITY ejemplo1 **IS**

PORT (x1, x2, x3 : **IN** BIT;
f : **OUT** BIT);

END ejemplo1;

ARCHITECTURE LogicFunc **OF** ejemplo1 **IS**

BEGIN

f <= (x1 AND x2) OR (NOT x2 AND x3);

END LogicFunc;

ANÁLISIS DE CÓDIGO VHDL

ENTITY ejemplo2 **IS**

PORT (x1, x2, x3, x4 : **IN** BIT;
f,g : **OUT** BIT);

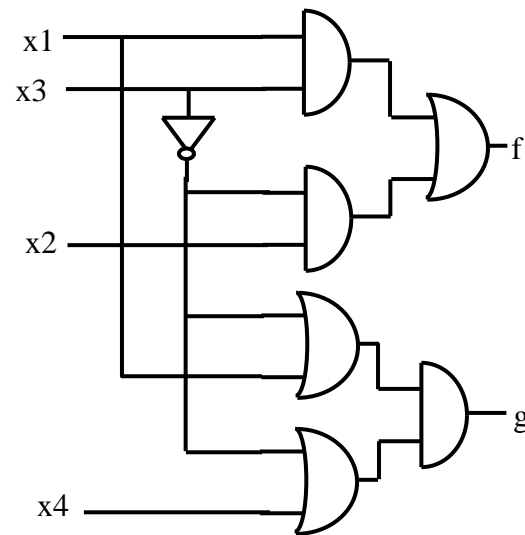
END ejemplo2;

ARCHITECTURE LogicFunc **OF** ejemplo2 **IS**
BEGIN

f <= (x1 AND x3) OR (NOT x3 AND x2);

g <= (NOT x3 OR x1) AND (NOT x3 OR x4);

END LogicFunc;



EJEMPLO DE CODIGO VHDL

Para **STD_LOGIC** hay un numero de valores legales, pero los mas importantes son : **0, 1, z** (alta impedancia) y **-** (condiciones no importa)

```
LIBRARY ieee;
```

```
USE ieee.std_logic_1164.all;
```

```
ENTITY ejemplo4 IS
```

```
    PORT (x1, x2, x3 : IN  STD_LOGIC;
```

```
          f          : OUT STD_LOGIC);
```

```
END ejemplo4;
```

```
    ARCHITECTURE LogicFunc OF ejemplo4 IS
```

```
    BEGIN
```

```
        f <= (NOT x1 AND NOT x2 AND x3) OR
```

```
            (x1 AND NOT x2 AND NOT x3) OR
```

```
            (x1 AND NOT x2 AND x3) OR
```

```
            (x1 AND x2 AND NOT x3);
```

```
    END LogicFunc;
```