

### III. LÓGICA COMBINACIONAL

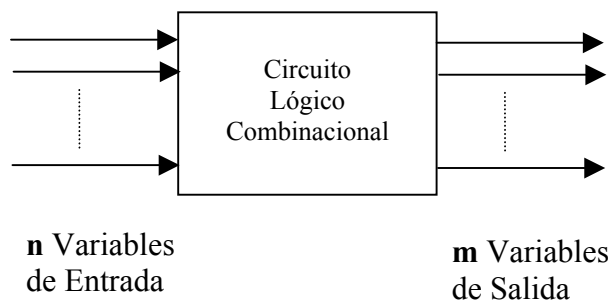
#### 3.1. Introducción

Los circuitos lógicos para sistemas digitales pueden ser **secuenciales** o **combinacionales**.

Un circuito combinacional consiste de un conjunto de compuertas lógicas cuyas salidas en cualquier momento están totalmente determinadas por la combinación presente de las variables de entrada, en el caso de un circuito secuencial, sus salidas son función de sus entradas presentes y del estado de sus elementos de memoria.

El propósito de este capítulo es utilizar el conocimiento adquirido en capítulos anteriores y formular varios procedimientos de análisis y diseño de circuitos combinacionales.

En un circuito combinacional para ' $n$ ' variables de entrada, existen  $2^n$  combinaciones posibles de valores binarios de entrada. Para cada combinación de entrada existe una y sólo una combinación de salida. Un circuito combinacional puede ser descrito por ' $m$ ' funciones de Boole, una para cada salida.



#### 3.2. Definición de Lógicas $\oplus$ y $\ominus$ (Positiva y negativa)

Lógica Positiva: La lógica positiva asocia el estado lógico '1' al estado físico de mayor valor y consecuentemente el estado lógico '0' al estado físico de menor valor.

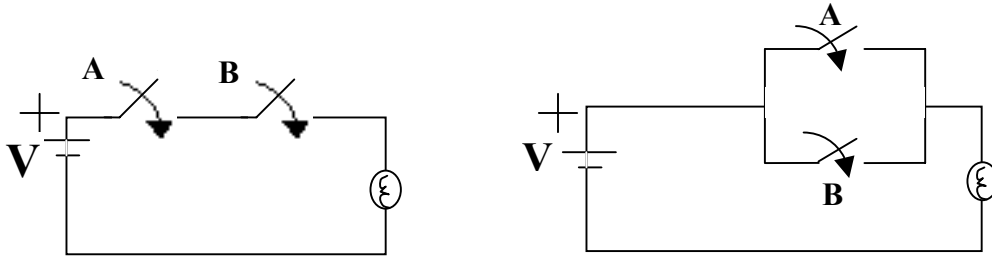
Lógica Negativa: La lógica negativa asocia el estado lógico '1' al estado menor valor y '0' al de mayor valor.

### 3.3. Elementos Físicos

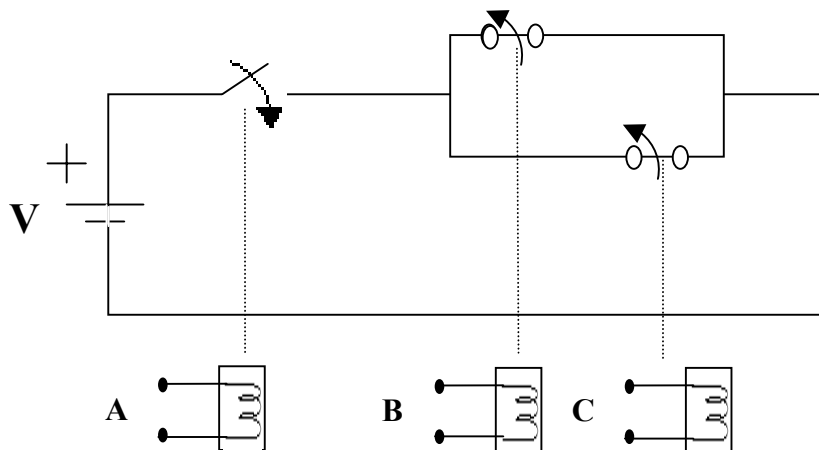
Las compuertas lógicas definidas en el capítulo anterior pueden implementarse por medio diversos elementos físicos. Ellos pueden ser eléctricos, electrónicos o mecánicos.

#### a) *Eléctricos: (Interruptores y relés)*

Ej: Interruptores



Ej: Relés



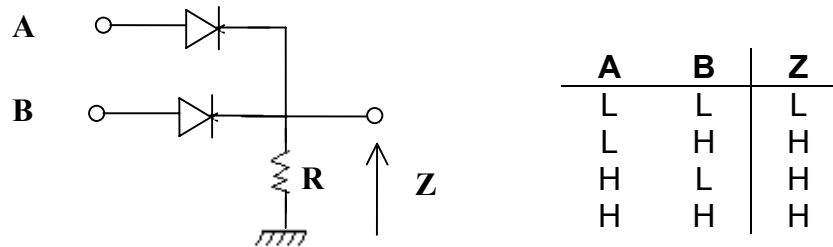
Se encenderá la ampollita cuando **A** este energizado **and** **B** o **C** no esten energizados:

$$\Rightarrow T = A(\bar{B} + \bar{C})$$

**b) Electrónicos:**

Los circuitos más simples para la construcción de compuertas lógicas están compuestos por diodos y resistencias (DL).

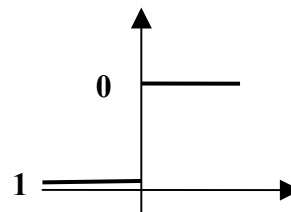
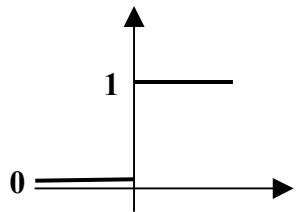
Ej 1:



Dependiendo de la lógica asociada, positiva o negativa, el circuito puede representar una compuerta **OR** o una **AND**

Lógica Positiva

Lógica Negativa



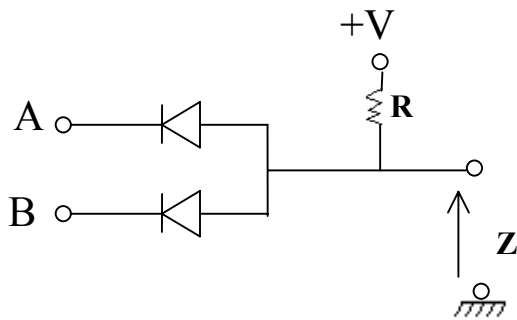
A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

} **OR**  
Lógica Positiva

A	B	Z
1	1	1
1	0	0
0	1	0
0	0	0

} **AND**  
Lógica Negativa

**Ej. 2:**



A	B	Z
L	L	L
L	H	L
H	L	L
H	H	H

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

**AND**  
Lógica Positiva

A	B	Z
1	1	1
1	0	1
0	1	1
0	0	0

**OR**  
Lógica Negativa

También es posible definir una lógica mixta para el ejemplo 1

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

Log. Positiva    Log. Negativa  
**NOR**

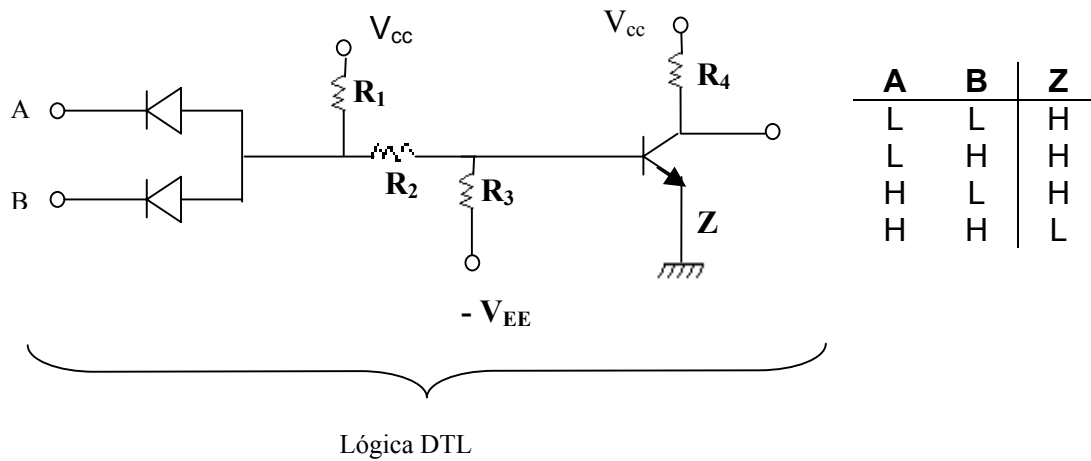
A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

Log. Positiva    Log. Negativa  
**NAND**

} Para el ejemplo 2

Pueden haber circuitos que se comporten como **NOR** o **NAND** con un sólo tipo de lógica tanto para entradas como para salidas.

**Ejemplo:**



**c) Elementos fluidicos**

Tambien es posible obtener compuertas fluidicas.

**Ejemplo.** Sean A y B entradas a un conducto con una salida comun S



**3.3. Fan In, Fan Out**

***Fan-in*** : Especifica el número de entradas a una compuerta, por ej. Una compuerta AND de 4 entradas tiene *Fan-In* 4.

El Fan-In está determinado por la habilidad del circuito para soportar entradas adicionales y realizar su función correctamente.

**Fan-Out** : Especifica el número de "cargas standard" para una salida de una compuerta sin salirse de su función normal. Una carga standard puede ser una entrada a otra compuerta, a un inversor o a otro circuito.

**Ejercicio** : Implementar la siguiente función. Considérese que sólo se dispone de compuertas Fan-In 3.

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

### 3.4. Circuitos Integrados

Un CI es un pequeño cristal de silicio que contiene un número determinado de compuertas. La gran mayoría de ellos pertenece a alguna familia según sean los elementos utilizados en su construcción.

Por ej.:

DTL  
RTL  
TTL  
CMOS, etc.



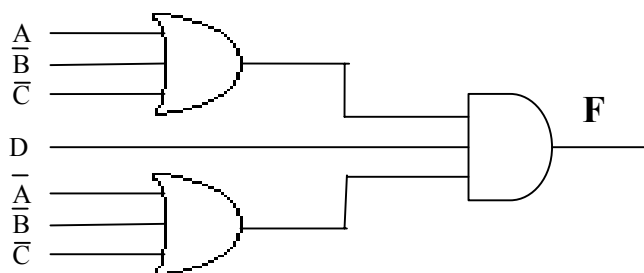
Los más utilizados son los CI pertenecientes a las familias TTL y CMOS

El número de componentes que contiene una pastilla puede variar desde unas pocas compuertas hasta miles de ellas. De acuerdo a este número se habla de CIs SSI (Integración a Pequeña Escala), MSI (Integración a Mediana Escala), LSI (Integración a Gran Escala), VLSI, etc.

### 3.5. Análisis de Circuitos Combinacionales

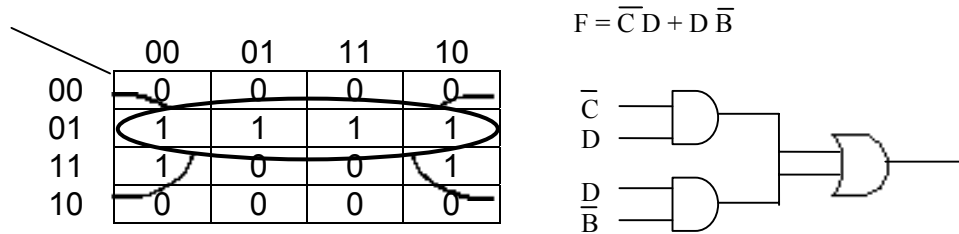
#### 3.5.1. Obtención de su función.

Dado un circuito lógico combinacional determinar la función que lo representa.

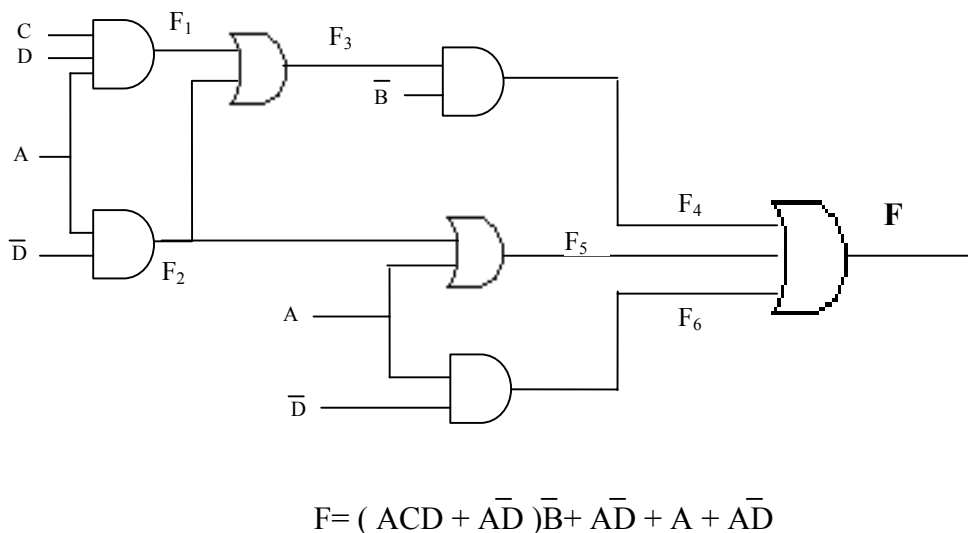


$$F = (A + B' + C')(A' + B' + C')D$$

¿ Existe otro circuito con menos número de puertas que represente la función ?



Eercicio Determinar la función Booleana del siguiente circuito.

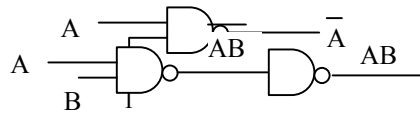


### 3.5.2. Análisis de circuito implementado con compuertas NAND o NOR.

Los circuitos son comúnmente contruidos con compuertas NAND o NOR en vez de AND, OR y NOT debido a las mejores características físicas de las compuertas NAND y NOR (Rapidez, mantención de niveles, menores costos, etc.)

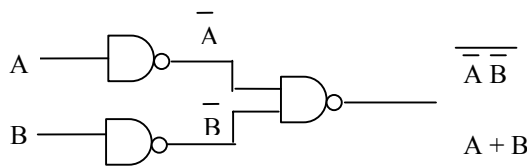
**I.- LÓGICA NAND:** Cualquier función booleana puede implementarse con sólo compuertas NAND.

### Ejemplo



### Inversor

### And

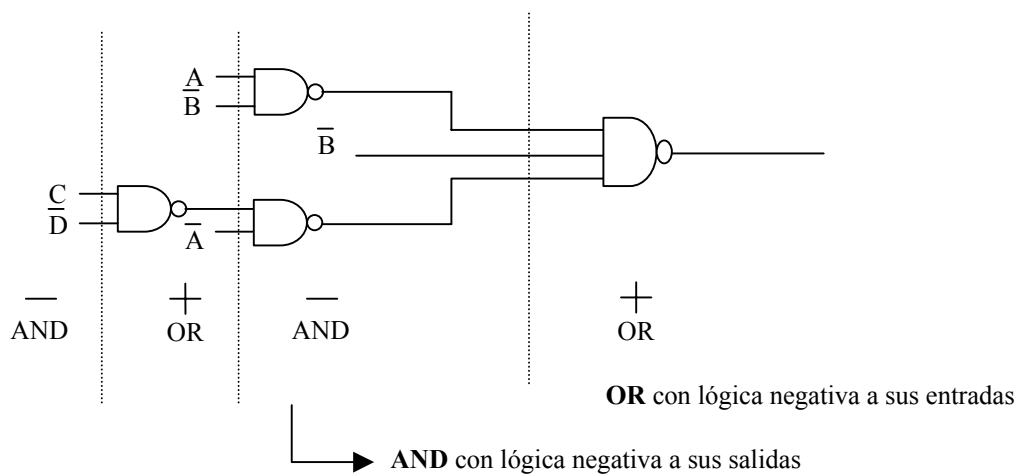


### Or

Desde el punto de vista del análisis de un circuito implementado con compuertas NAND, se puede pasar a uno con compuertas AND y OR como camino alternativo a la utilización sucesiva del teorema DE MORGAN

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

### Ejemplo



- Método**
- 1.- Se separa por niveles a partir de su salida.
  - 2.- Se aplica lógicas positiva y negativa alternativamente.



**Nota:** Aplicar siempre primero lógica positiva a la salida del último nivel.

NAND Física

A	B	$\overline{A \cdot B}$
L	L	H
L	H	H
H	L	H
H	H	L

NAND Lógica

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

Compuerta AND

A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

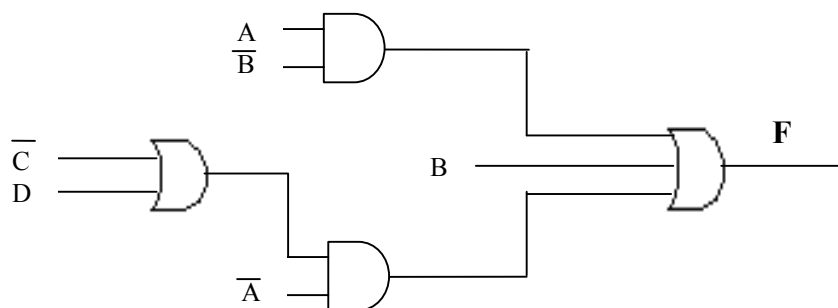
Compuerta OR

A	B	$A + B$
1	1	1
1	0	1
0	1	1
0	0	0



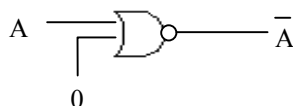
NAND con lógica positiva    lógica negativa

NAND con lógica negativa    lógica positiva

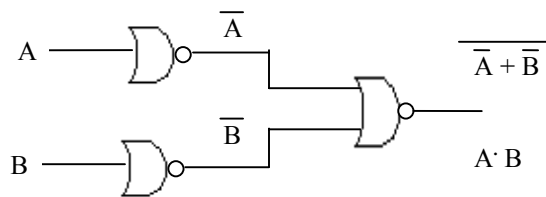


**II. LÓGICA NOR:** Cualquier función booleana puede implementarse con sólo compuertas NOR

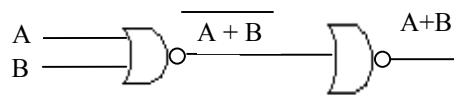
### Inversor



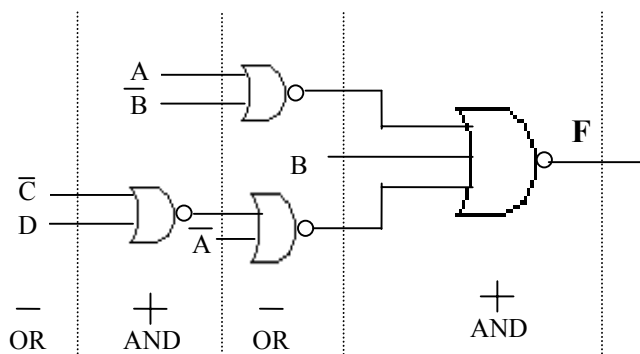
### And



### Or



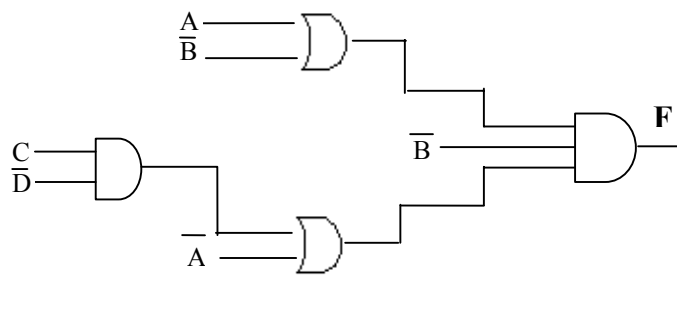
Ej.:



NOR lógico

A	B	$\overline{A+B}$	
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

$\oplus$      $\ominus$     OR



A	B	$\overline{A+B}$
1	1	1
1	0	0
0	1	0
0	0	0

$\ominus$      $\oplus$   
 AND

### 3.6. Diseño de Circuitos Combinacionales

Se deben considerar los siguientes aspectos:

- a) Formulación verbal de un problema.
- b) Determinación de las variables de entrada y salida.
- c) Deducción tabla de verdad.
- d) Minimización  $\left\{ \begin{array}{l} \text{Mapas -K} \\ \text{Q-M} \end{array} \right.$
- e) Implementación física.
- f) Restricciones.

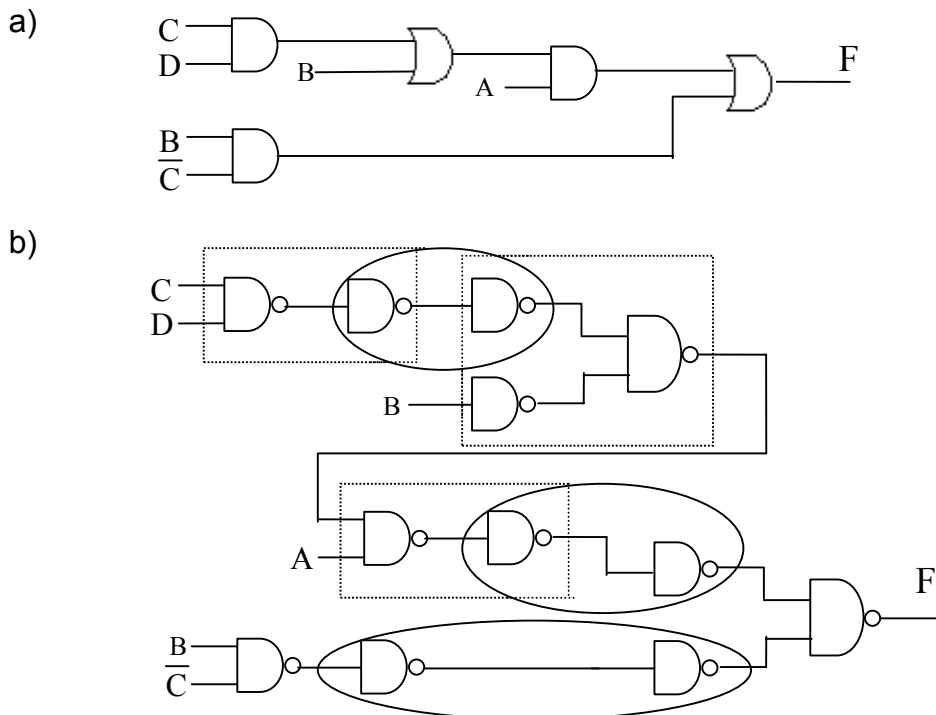
### 3.7. Obtención de un Circuito con Compuertas NAND

- a) Implementar la función con compuertas AND, OR y NOT.
- b) Dibujar un diagrama con compuertas NAND ó NOR. Sustituyendolas por las AND, OR y NOT del circuito.
- c) Remover de a 2 los inversores que se encuentran en cascada.

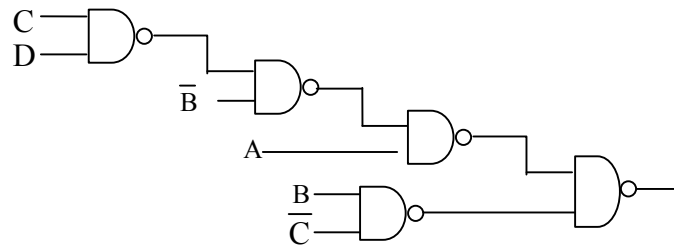
Ej.: Sea la expresión algebraica:

$$1) \quad F = A ( B + CD ) + B\bar{C}$$

- Obtener su circuito con compuertas NAND

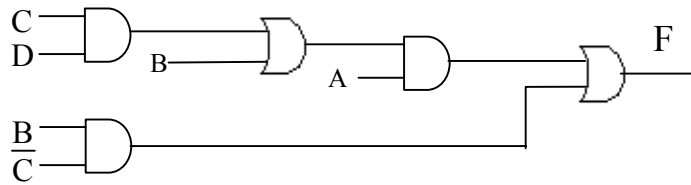


c)

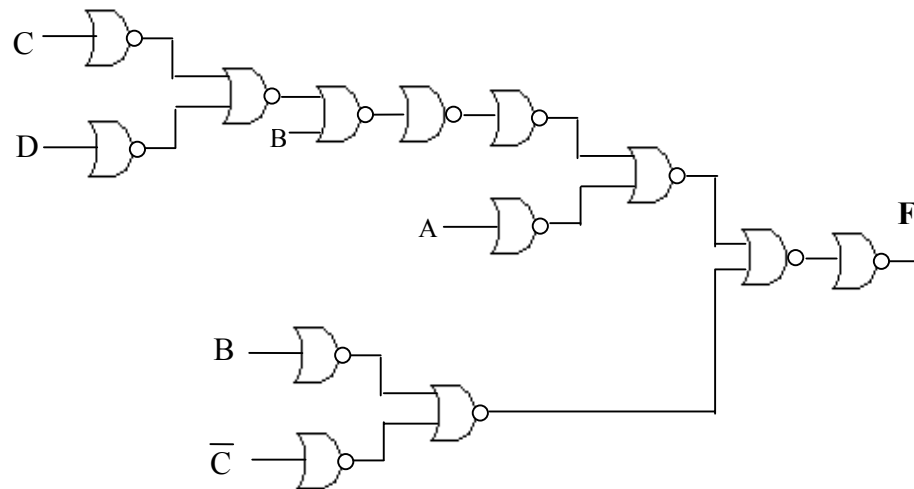


- 2) Sea la misma función  $F = A(B + CD) + \overline{B}\overline{C}$   
 - Implementar su circuito sólo con compuertas NOR

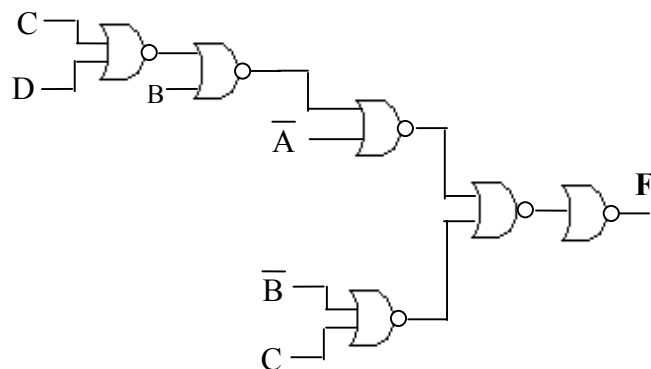
a)



b)



c)



**Ejercicio :** Diseñar un circuito lógico combinacional que sea la implementación de la siguiente función.

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

Solo se dispone de compuertas con Fan-In 3.

### Algunos circuitos combinacionales de interés

#### 1. Sumadores :

+	0	1
0	0	1
1	1	① 0

1 carry

a) Semisumadores: Sumador que suma 2 bit

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

	x	0	1
y	0	0	1
	1	1	0

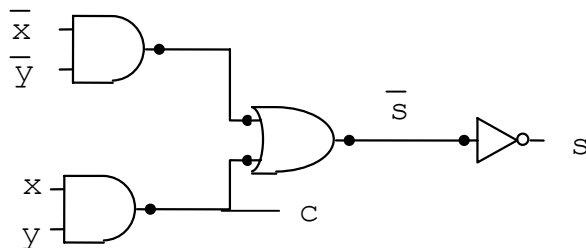
S

$$\Rightarrow S = \bar{X}Y + X\bar{Y} = X \oplus Y$$

$$C = XY$$

$$S = (X + Y)(\bar{X} + \bar{Y}) ; C = XY$$

$$\bar{S} = (X Y) + (\bar{X} \bar{Y}) = C + \bar{X}\bar{Y} \Rightarrow S = \bar{\bar{S}}$$



b) **Sumador Completo(Full Adder)** : Sumador que suma 3 bit. 2 bit significativos más el carry anterior.

$X_i$	$Y_i$	$C_i$	$Z$	$C_{i+1}$	$S_i$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1

		xy			
		00	01	11	10
z	0	0	1	0	1
	1	1	0	1	0

$S_i$

$Z$  = Carry de la etapa anterior

$$S = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + XY\bar{Z} + X\bar{Y}\bar{Z} = S_i$$

$XY$	00	01	11	10
$z$	0		1	
	1	1	1	1

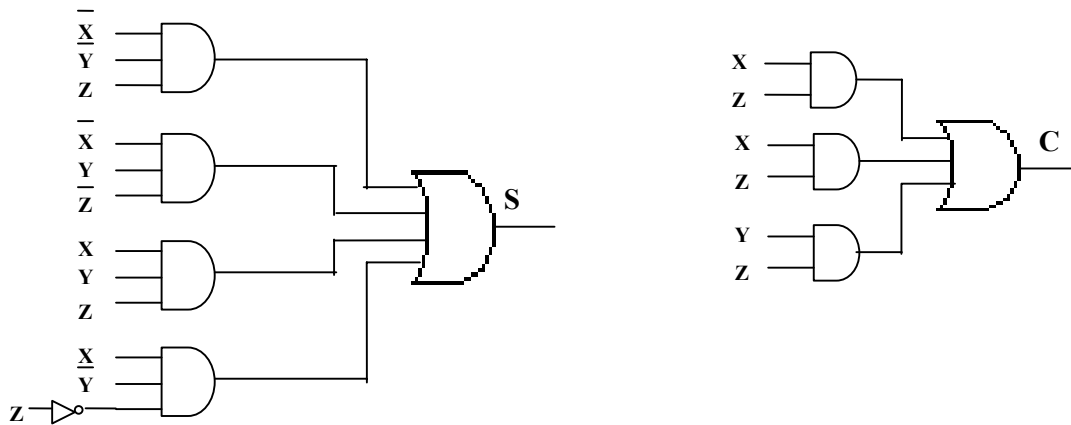
$C_{i+1}$

$$C_{i+1} = XY + YZ + XZ$$

$$X_i Y_i + \bar{X} Y Z + X Y Z + \bar{Y} X Z + Y X Z$$

$$Z ( \bar{X} Y + \bar{Y} X ) + X Y ( Z + 1 )$$

$$C_{i+1} = Z ( X \oplus Y ) + XY$$



Existen varias formas de realizar las funciones  $S_i$  y  $C_{i+1}$  una muy conveniente para  $S_i$  hace uso del OR-EX, ya que se puede demostrar que:

$$S_i = X_i \oplus Y_i \oplus Z_i$$

$$C_{i+1} = (X_i \oplus Y_i) C_i + X_i Y_i$$

$$C_{i+1} = P C_i + X_i Y_i$$

$$P = (X_i \oplus Y_i)$$

