

# **CAPÍTULO 3**

## **Procedimiento del Diseño**

---

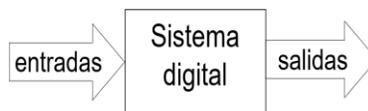
# Procedimiento de diseño

El diseño de cualquier sistema digital comienza con el enunciado del proyecto el cual indica las restricciones que se tendrán y termina con el diagrama del circuito lógico. Este procedimiento requiere cubrir los siguientes pasos:

- 1.- Especificaciones del proyecto.
- 2.- Diagrama de bloques.
- 3.-Tabla de verdad.
- 4.- Función booleana.
- 5.-Diagrama lógico.

1.- **Especificaciones del proyecto:** en este punto se describen de forma clara y concisa los requisitos, las instrucciones y las restricciones que el proyecto tiene.

2.- **Diagrama de bloques:** se determinan las variables de entrada y las funciones de salida con las que se cuenta en el proyecto. En la figura 3.1 se observa el sistema digital como caja una negra con las entradas y las salidas que se pudieran tener.



**Figura 3.1 Diagrama de Bloques.**

3.-**Tabla de verdad:** una tabla de verdad es el elemento utilizado para visualizar el funcionamiento de cualquier sistema combinacional. Dicho sistema cambia sus valores de salida dependiendo de los valores en sus entradas. La tabla está dividida en dos partes. La primera parte maneja las variables de entrada, y todas sus posibles combinaciones.

El número de combinaciones posibles para una función de  $n$  variables vendrá dado por  $2^n$ . La segunda parte maneja las funciones cuyos valores indican el comportamiento del sistema ante las posibles combinaciones de entrada. En la figura 3.2 se observa la estructura de una tabla de verdad.

		Tabla de verdad					
VARIABLES	{	Entradas			Salida	}	FUNCIÓN
		A	B	C	F		
Combinaciones de las variables	{	0	0	0	0	}	Términos de la función
		0	0	1	1		
		0	1	0	1		
		0	1	1	0		
		1	0	0	1		
		1	0	1	0		
		1	1	0	0		
		1	1	1	1		

**Figura 3.2 Elementos que contiene una tabla de verdad**

*Función booleana.*

Se denomina función booleana a aquella función matemática cuyas variables son binarias y están unidas mediante uno o varios de los operadores del álgebra de Boole (+, ·). Dicha función relaciona las salidas del sistema con las entradas. La información de una tabla de verdad puede ser representada mediante una función lógica. Existen distintas forma de representar una función lógica, entre las que podemos destacar las siguientes:

- ✓ Forma canónica
- ✓ Forma normal
- ✓ Minitérminos ó maxitérminos.

*Importancia*

Implementar una ecuación booleana a través de un circuito lógico, reducir la ecuación a través de algebra booleana e implementar la ecuación reducida que debe ser equivalente a la función de la ecuación original

*Forma canónica.*

Si se tiene una tabla de verdad, es posible representar su función lógica en forma canónica, utilizando el símbolo  $\sum$  afuera de un paréntesis que contiene el número en decimal de las combinaciones que a su salida tengan un uno lógico, o el símbolo  $\pi$  afuera de un paréntesis que contiene las combinaciones que a su salida tengan un cero lógico, ambas funciones son equivalentes, cuando representan a la misma función.

La figura 3.3 ejemplifica una tabla de verdad y las ecuaciones en forma canónica equivalentes, lo que significa que con la implementación del circuito lógico de una de ellas es suficiente.

entradas		salida
a	b	f
0	0	0
0	1	0
1	0	1
1	1	1

Forma canónica tomando en cuenta los unos en la función  $f = \sum (2,3)$   
 Forma canónica tomando en cuenta los ceros en la función  $f = \pi (0,1)$

**Figura 3.3 Representaciones de forma canónica de una tabla de verdad**

*Forma normal.*

Si se tiene una tabla de verdad, es posible representar su función lógica en forma normal, utilizando un paréntesis que contiene los elementos en decimal de las combinaciones que a su salida tengan un uno lógico, un paréntesis que contiene las combinaciones que a su salida tengan un cero lógico, ambas funciones son equivalentes, cuando representan a la misma función. La figura 3.4 ejemplifica una tabla de verdad y las ecuaciones en forma normal equivalentes, lo que significa que con la implementación del circuito lógico de una de ellas es suficiente

entradas		salida
a	b	f
0	0	0
0	1	0
1	0	1
1	1	1

Forma normal tomando en cuenta los unos de la función  $f = (m_2, m_3)$   
 Forma normal tomando en cuenta los ceros en la función  $f = (M_0, M_1)$

**Figura 3.4 Representaciones en forma normal de una tabla de verdad**

*Minitérminos y maxitérminos de la función lógica.*

La función booleana puede ser representada de dos formas:

- ✓ Producto de Sumas (maxitérminos)
- ✓ Suma de Productos (minitérminos)

*Producto de Sumas (POS).*

Expresión booleana que reúne las variables de entrada que a su salida tengan un cero lógico, mediante una compuerta OR (suma), y todas las combinaciones las agrupa mediante una compuerta AND (producto).

*Suma de Productos (SOP).*

Expresión booleana que reúne las variables de entrada que a su salida tengan un uno lógico, mediante una compuerta AND (producto), y todas las combinaciones las agrupa mediante una compuerta OR (suma).

La figura 3.5 ejemplifica una tabla de verdad y la representación de las funciones booleanas en maxitérminos y en minitérminos equivalentes, lo que significa que con la implementación del circuito lógico de una de ellas es suficiente

entradas		salida
a	b	f
0	0	0
0	1	0
1	0	1
1	1	1

Función booleana representada mediante minitérminos (SOP)  
 $f = \bar{a}\bar{b} + ab$

Función booleana representada mediante maxitérminos (POS)  
 $f = (a+b)(a+\bar{b})$

**Figura 3.5 Representaciones en forma de maxitérminos y de minitérminos de una tabla de verdad**

*Diagrama Lógico.*

Es la representación gráfica de la unión de los símbolos de los circuitos integrados que se requiere para implementar algún sistema digital.

### 3.1 Compuertas Lógicas.

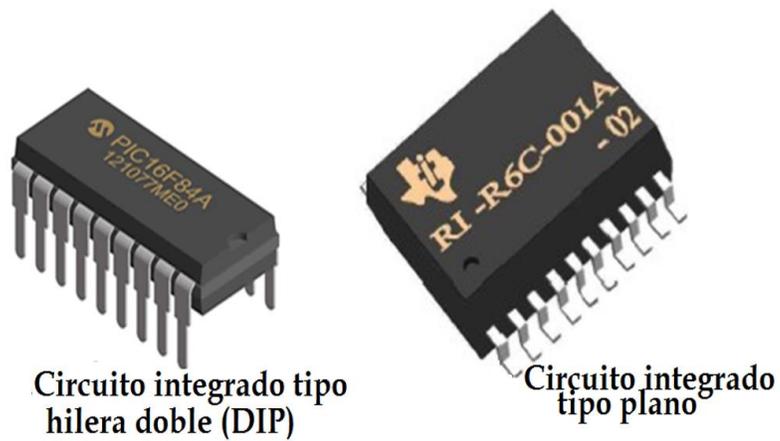
La compuerta lógica es la implementación física y electrónica de las compuertas lógicas. Es un dispositivo digital que trabaja con dos estados lógicos, tanto en sus entradas como en sus salidas.

La manipulación de información binaria se hace mediante circuitos lógicos denominados compuertas. Las compuertas son bloques de hardware que producen señales binarias “1” (con energía) o “0” (sin energía), cuando se satisfacen los requisitos de entrada lógica. Cada compuerta tiene un símbolo electrónico asociado, y su operación puede describirse mediante una función booleana.

#### 3.2.1 Circuitos integrados.

Por otra parte, las compuertas lógicas se encuentran disponibles comercialmente en forma de circuitos integrados. Un circuito integrado, también denominado *chip*, está compuesto por un material semiconductor de silicio, que en su interior, contiene componentes eléctricos tales como transistores, diodos, resistencias y capacitores. Los diversos componentes, están interconectados para formar un circuito electrónico dentro un empaque, por lo general de plástico, con terminales externas de salida/entrada, para conformar así el circuito integrado.

Los fabricantes de circuitos integrados, ponen en disponibilidad dos clases de *chips*. El circuito de doble hilera para aplicaciones electrónicas de tipo *through-hole*, y los circuitos de tipo plano, para aplicaciones de montaje de superficie. La figura 3.6 ejemplifica los dos tipos de circuitos integrados comercialmente disponibles.



**Figura 3.6 Tipos de circuitos integrados**

### **3.2.2 Símbolos gráficos de las compuertas.**

Las compuertas básicas son tres AND, OR y NOT, y sus compuertas complementarias son NAND, NOR. También existe una compuerta suplementaria XOR y su compuerta complementaria XNOR. La figura 3.7 muestra el símbolo, nombre, tabla de verdad y número del circuito integrado de estas 7 compuertas:

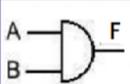
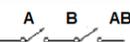
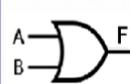
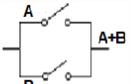
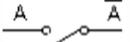
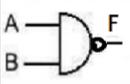
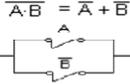
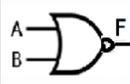
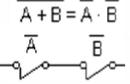
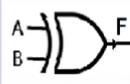
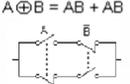
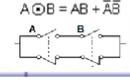
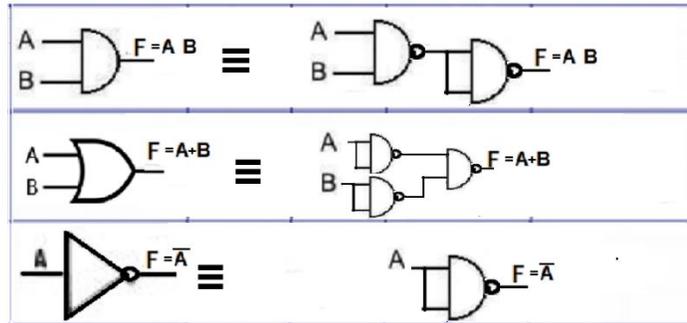
Símbolo	Nombre	Función	Tabla de verdad	Num de chip	Operación										
	AND	$F = AB$	<table border="1"> <tr><td>AB</td><td>F</td></tr> <tr><td>00</td><td>0</td></tr> <tr><td>01</td><td>0</td></tr> <tr><td>10</td><td>0</td></tr> <tr><td>11</td><td>1</td></tr> </table>	AB	F	00	0	01	0	10	0	11	1	7408	
AB	F														
00	0														
01	0														
10	0														
11	1														
	OR	$F = A + B$	<table border="1"> <tr><td>AB</td><td>F</td></tr> <tr><td>00</td><td>0</td></tr> <tr><td>01</td><td>1</td></tr> <tr><td>10</td><td>1</td></tr> <tr><td>11</td><td>1</td></tr> </table>	AB	F	00	0	01	1	10	1	11	1	7432	
AB	F														
00	0														
01	1														
10	1														
11	1														
	NOT	$F = \overline{A}$	<table border="1"> <tr><td>A</td><td>F</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	F	0	1	1	0	7404					
A	F														
0	1														
1	0														
	NAND	$F = \overline{AB}$	<table border="1"> <tr><td>AB</td><td>F</td></tr> <tr><td>00</td><td>1</td></tr> <tr><td>01</td><td>1</td></tr> <tr><td>10</td><td>1</td></tr> <tr><td>11</td><td>0</td></tr> </table>	AB	F	00	1	01	1	10	1	11	0	7400	
AB	F														
00	1														
01	1														
10	1														
11	0														
	NOR	$F = \overline{A + B}$	<table border="1"> <tr><td>A+B</td><td>F</td></tr> <tr><td>00</td><td>1</td></tr> <tr><td>01</td><td>0</td></tr> <tr><td>10</td><td>0</td></tr> <tr><td>11</td><td>0</td></tr> </table>	A+B	F	00	1	01	0	10	0	11	0	7402	
A+B	F														
00	1														
01	0														
10	0														
11	0														
	XOR	$F = A \oplus B$	<table border="1"> <tr><td>A⊕B</td><td>F</td></tr> <tr><td>00</td><td>0</td></tr> <tr><td>01</td><td>1</td></tr> <tr><td>10</td><td>1</td></tr> <tr><td>11</td><td>0</td></tr> </table>	A⊕B	F	00	0	01	1	10	1	11	0	7486	
A⊕B	F														
00	0														
01	1														
10	1														
11	0														
	XNOR	$F = \overline{A \oplus B}$	<table border="1"> <tr><td>A⊕B</td><td>F</td></tr> <tr><td>00</td><td>1</td></tr> <tr><td>01</td><td>0</td></tr> <tr><td>10</td><td>0</td></tr> <tr><td>11</td><td>1</td></tr> </table>	A⊕B	F	00	1	01	0	10	0	11	1	74266	
A⊕B	F														
00	1														
01	0														
10	0														
11	1														

Figura 3.7 Símbolos y datos de las compuertas

### 3.2.3 Universalidad de las compuertas NAND y NOR

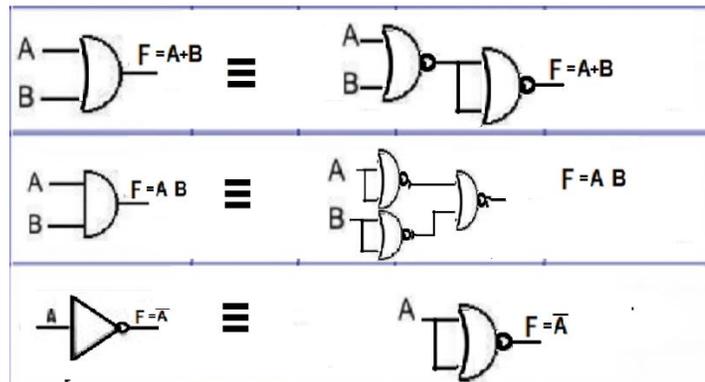
Las compuertas NAND y NOR son llamadas compuerta universales, debido a que emulan el comportamiento de las compuertas básicas. Además, mediante estas compuertas es posible construir cualquier función lógica.

La figura 3.8 muestra la forma de interconectar compuertas NAND's para poder emular el comportamiento de las compuertas básicas.



**Figura 3.8** Universalidad de la compuerta NAND

La figura 3.9 muestra la forma de interconectar compuertas NOR's para poder emular el comportamiento de las compuertas básicas.



**Figura 3.9** Universalidad de la compuerta NOR

### 3.2 Familias lógicas y escalas de integración

Una familia lógica es un grupo de circuitos integrados que están fabricados con una tecnología común y tienen estandarizadas sus características de entrada y de salida; por lo que son compatibles entre sí y no requiere de etapas adicionales de acoplamiento.

Las características más importantes de un circuito integrado son su velocidad, su consumo de potencia, su inmunidad al ruido y su confiabilidad. La velocidad mide la rapidez de respuesta de las salidas de un circuito digital a cualquier cambio en sus entradas. El consumo de potencia mide la cantidad de corriente o de potencia que consume un circuito digital en operación. La inmunidad al ruido mide la sensibilidad de un circuito digital al ruido electromagnético ambiental. La confiabilidad mide el período útil de servicio de un circuito digital.

Las dos familias más populares son la TTL y la CMOS, y es como hablar de grano grueso y grano fino. La familia TTL soporta más el uso rudo, y la familia CMOS elimina más el ruido en sus salidas.

## FAMILIA LÓGICA TTL

Los circuitos integrados TTL implementan su lógica interna, exclusivamente basándose en transistores NPN y PNP, diodos y resistencias y está disponible en dos versiones: la serie 54 y la serie 74. La primera se destina a aplicaciones militares y la segunda a aplicaciones industriales y de propósito general. La familia TTL o bipolar se divide en las siguientes categorías o subfamilias básicas: TTL estándar. TTL Schottky (S). TTL de baja potencia (L). TTL Schottky de baja potencia (LS). TTL de alta velocidad (H). TTL Schottky avanzada (AS). TTL Schottky de baja potencia avanzada (ALS).

## FAMILIA LÓGICA CMOS

Los circuitos integrados CMOS, utiliza transistores MOSFET complementarios canal N y canal P como elementos básicos de conmutación. Los circuitos integrados digitales fabricados mediante tecnología CMOS se pueden agrupar en las siguientes categorías o subfamilias básicas: CMOS estándar. CMOS de alta velocidad (HC). CMOS compatible con TTL (HCT). CMOS equivalente a TTL (C). Familia CMOS estándar. La familia CMOS estándar comprende principalmente los dispositivos que se designan como 40XX (4012, 4029, etc.) y 45XX (4528, 4553, etc.). Existen dos series generales de dispositivos CMOS designadas "A" y "B". Los dispositivos de la serie "A" se designan con el sufijo "A" o simplemente no lo traen impreso (4011A = 4011). Todos los dispositivos de la serie "B" llevan el sufijo B.

## DIFERENCIA ENTRE LAS DOS FAMILIAS.

La familia CMOS, puede trabajar dentro de un rango de voltaje entre 3 a 15 volts mientras que la familia TTL en un rango de 4.75 a 5.25 volts. La capacidad de carga de la familia CMOS es de 400 y la de la familia TTL de 10. Es mucho mayor el rechazo al ruido de la familia CMOS, sin embargo es muy lenta pues trabaja a 50 Mhz mientras que la familia TTL trabaja a 250 Mhz, en su serie LS.

Los circuitos integrados se pueden clasificar en niveles de integración dependiendo del número de compuertas o transistores que tengan.

- ✓ *SSI Small Scaled Integration* (pequeña escala de integración); en esta clasificación entran los circuitos integrados que contengan de 1 a 12 compuertas.
- ✓ *MSI Medium Scaled Integration* (mediana escala de integración); en esta clasificación entran los circuitos integrados que contengan de 13 a 99 compuertas.
- ✓ *LSI Large Scaled Integration* (gran escala de integración); en esta clasificación entran los circuitos integrados que contengan de 100 y hasta 999 compuertas.
- ✓ *VLSI Very Large Scaled Integration* (muy grande escala de integración); en esta clasificación entran los circuitos integrados que contengan de 1000 y hasta 9999 transistores.

- ✓ ULSI *Ultra Large Scaled Integration* (gran escala de integración); en esta clasificación entran los circuitos integrados que contengan más de 10000 transistores.

Para reforzar los conocimientos estudiados en este capítulo, a continuación se presentan algunos ejemplos sobre el diseño de sistemas digitales básicos.

**Ejemplo 3.1.** Obtener el circuito lógico de un sistema detector de números pares, únicamente con el uso de compuertas básicas. El sistema debe contar con dos entradas A y B, cada una de un bit. Únicamente cuando ambas entradas sean iguales se deberá activar la salida para encender una LUZ.

*Solución:*

*Diagrama de Bloques.*



*Tabla de Verdad.*

A	B	Luz
0	0	1
0	1	0
1	0	0
1	1	1

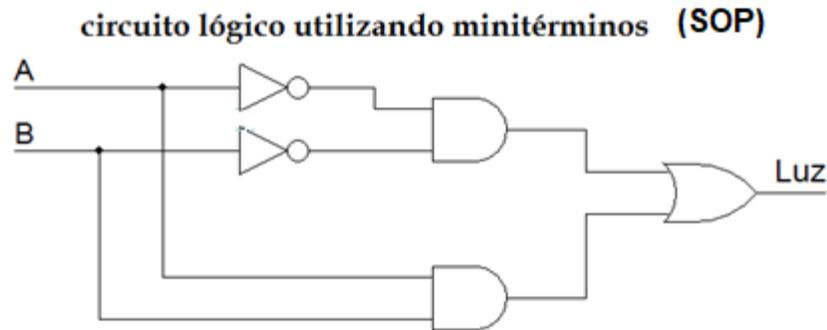
*Funciones booleanas:*

$$(SOP) \text{ Luz} = \overline{A}B + A\overline{B}$$

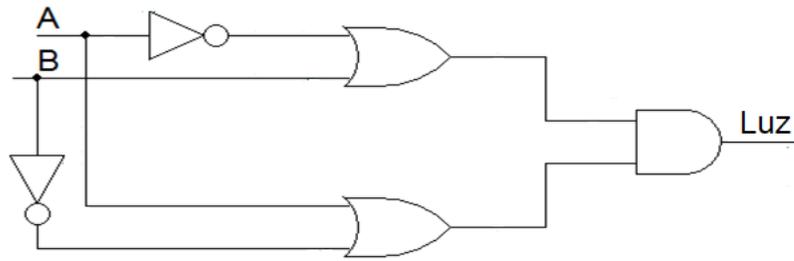
$$(POS) \text{ Luz} = (A + \overline{B})(\overline{A} + B)$$

Nota:

Las dos ecuaciones booleanas son equivalentes, por lo que los dos circuitos lógicos también son equivalentes. Se recomienda implementar el circuito lógico que tenga menos elementos debido a que requiere menor hardware.



**circuito lógico utilizando MAXitérminos (POS)**



**\*\* COMENTARIOS:**

Las fotografías son de mi autoría. Únicamente la figura 3,3 la saque de

<http://www.monografias.com/trabajos104/circuitos-integrados-y-familias-logicas/img3.png>

- Poner referencia de las fotografías e imágenes utilizadas, o en su caso, rehacer diagramas y figuras con mayor resolución y claridad.
- Sustituir las capturas de pantalla de texto por formato de texto/ecuación
- Justificar texto y verificar que los márgenes sean consistentes a lo largo del documento.
- Aclarar los conceptos señalados en rojo.