

UNIVERSIDAD NACIONAL DEL CALLAO

FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICAS

INSTITUTO DE INVESTIGACIÓN



INFORME FINAL DE PROYECTO DE INVESTIGACIÓN

**“TEXTO: INSTRUMENTACIÓN ELECTRÓNICA – PARTE II,
TEORÍA Y PROBLEMAS CON PROGRAMAS
COMPUTACIONALES”**

AUTOR

Mg. ROLANDO JUAN ALVA ZAVALETA

(Período de ejecución: 01/05/12 al 30/04/14)
(Resolución N° 421-2012-Rdel 24 de mayo de 2012)

DEDICATORIA

A la memoria de mis padres,
Enrique y María, porque
siempre están presentes.

CONTENIDO

	Págs.
I. ÍNDICE	1
II. PRÓLOGO	2
III. INTRODUCCIÓN	3
IV. CUERPO DE TEXTO O CONTENIDO	4
4.1. Compuertas lógicas y algebra booleana	4
4.2. Flip flops y dispositivos relacionados	11
4.3. Mapas de Karnaugh	20
4.4. Contadores y registros	28
4.5. Familias lógicas en circuitos integrados	34
4.6. Circuitos lógicos MSI	42
4.7. Interfaz con el mundo analógico	48
4.8. Dispositivos de memoria	55
V. REFERENCIABIBLIGRÁFICAS	61
VI. APÉNDICE	62

II. PRÓLOGO

El presente texto ha sido elaborado con la finalidad apoyar el aprendizaje de los alumnos en el curso de Instrumentación Electrónica II de la Escuela Profesional de Física.

Esta obra desarrolla la electrónica digital, presentando los fundamentos de esta rama de la electrónica con énfasis no solo en la teoría sino también en programas computacionales codificados en Fortran 90 que le dan el soporte en la simulación. Esto último se convierte en una de las aplicaciones de la Física Computacional como línea que se encuentra contemplado en el perfil profesional de los alumnos de la Escuela de Física, los cuales tienen conocimiento en técnicas numéricas y lenguajes de programación que se utilizan en ciencia básica

Por último, pero no menos importante, el alumno de Física encontrará una sencilla guía para aprender electrónica simulando el comportamiento de los sistemas digitales.

III. INTRODUCCIÓN

En el presente trabajo de investigación se desarrolló un texto teórico-práctico de la asignatura de Instrumentación Electrónica II mediante el uso de programas científicos en fortran 90, con el objetivo de hacer didáctica y sistemática la enseñanza y el aprendizaje de temas complejos de diseño de circuitos electrónicos digitales para las mediciones Físicas. También se pretende que los estudiantes desarrollen sus propios instrumentos orientados para satisfacer necesidades experimentales. La importancia del trabajo es que pretende cubrir la necesidad de contar con textos de Instrumentación Electrónica que contengan ejemplos con programas computacionales. La contribución de este trabajo de investigación estriba además de presentar en forma simple la teoría, acompañada de la aplicación a casos concretos que son simulados computacionalmente. Este último caracteriza la diferencia con otros textos que generalmente están en inglés y que no presentan aplicaciones con su correspondiente programa.

Por último, la presente investigación pretende lograr que el estudiante del pregrado de la especialidad de Física tenga a su disposición un material didáctico que favorezca su aprendizaje.

IV. CUERPO DEL TEXTO O CONTENIDO

CAPÍTULO IX

COMPUERTAS LÓGICAS Y ÁLGEBRA BOOLEANA

9.1 Introducción

La electrónica digital se fundamenta en las compuertas lógicas las cuales son unidades electrónicas que hacen posible obtener en la salida de cada una de ellas, una señal lógica que depende de las señales lógicas establecida en sus entradas. Cada compuerta lógica generalmente puede tener una salida digital, sin embargo, estos dispositivos tienen muchas entradas. Conectar compuertas lógicas entre sí permite obtener otros circuitos lógicos con funciones más complejas.

En el mercado existe diversidad de compuertas lógicas, pero cada una tiene un singular funcionamiento lógico. Sin embargo, al ser conectadas entre sí, bajo ciertas pautas de diseño, constituyen un sistema digital, con nuevas características de funcionamiento.

Una compuerta lógica está conformada por un conjunto de transistores incluidos dentro un circuito integrado, que albergan cientos de compuertas. Sin embargo, desde el punto de vista práctico, se considera a una compuerta lógica como una caja negra, en cuyas entradas se establecen valores digitales, dando también un valor digital en su salida.

El funcionamiento de una compuerta lógica se representa a través de una tabla de verdad, la cual expresa que establecida las señales de entrada de cualquiera de las combinaciones posibles nos indica la señal en su salida para cada combinación.

Gracias al avance tecnológico en ciencia de materiales respecto a la fabricación de materiales semiconductores ha hecho posible en la actualidad la fabricación de las

compuertas lógicas, esta a su vez son la base para la fabricación de chips lógicos, hasta procesadores fabricados con miles de estas compuertas.

A continuación, estudiaremos la representación simbólica para cada compuerta lógica, así como la tabla de verdad para cada una de ellas.

9.2 Compuertas lógicas y sus tablas de verdad

El funcionamiento de cada compuerta obedece al tipo de operación que ejecuta dentro del algebra Booleana. Tales operaciones son: AND, OR y NOT. La representación simbólica de cada compuerta lógica básica, el tipo de operación que ejecuta y su tabla de verdad, se presentan a continuación:

9.2.1 Operación AND

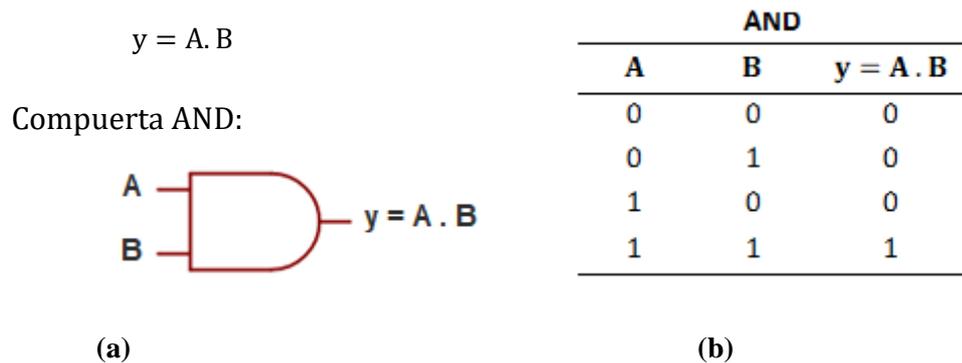


Figura 9.1: a) Símbolo de la compuerta AND b) Tabla de verdad

9.2.2 Operación OR

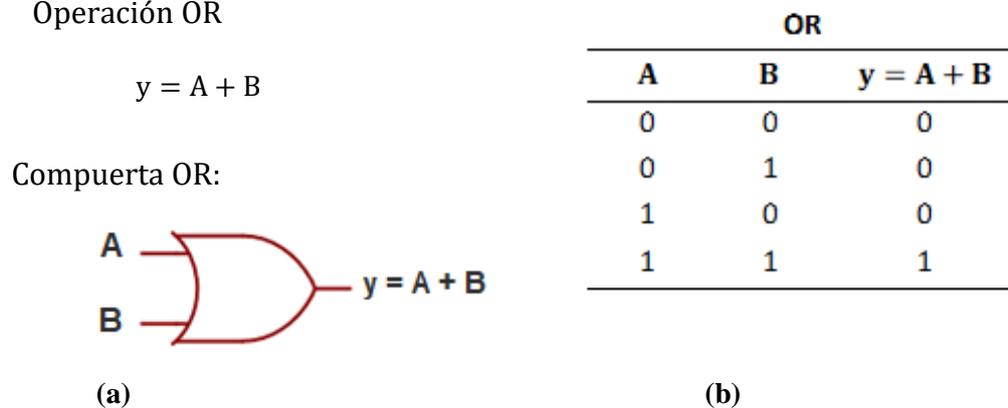
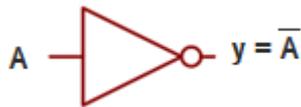


Figura 9.2: a) Símbolo de la compuerta OR b) Tabla de verdad

9.2.3 Operación NOT

$$y = A + B$$

Compuerta NOT:



(a)

NOT	
A	y = \bar{A}
0	1
1	0

(b)

Figura 9.3: a) Símbolo de la compuerta NOT y b) Tabla de verdad

9.3 Compuerta NAND

Otras compuertas son obtenidas a partir de las tres compuertas anteriores. Estas son las compuertas NOR y OR. Sus símbolos, operación y tabla de verdad se presentan enseguida:



a)



b)

NAND		
A	B	y = $\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

c)

Figura 9.4: a) Símbolo de la compuerta NAND b) su circuito equivalente c) y Tabla de verdad

9.3.1 Compuerta NOR

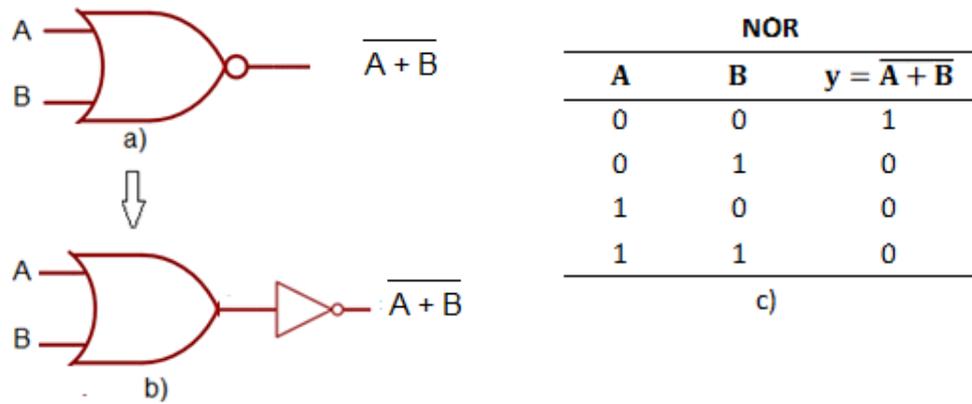
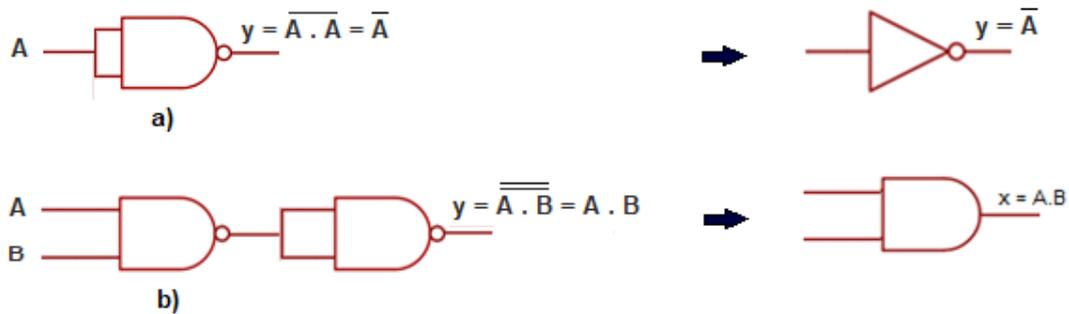


Figura 9.5: a) Símbolo de la compuerta NOR y su circuito equivalente b) Tabla de verdad

9.4 Universalidad de las compuertas NAND.

A continuación, se demuestra la universalidad de las compuertas NAND, ya que, a través de combinar estas compuertas, y fundado en los teoremas de Boole, podemos construir las compuertas básicas señaladas. Esta universalidad (b) también se puede atribuir a las compuertas NOR, dado que también con estas compuertas se puede construir cualquier compuerta básica.

Como consecuencia de esta universalidad se puede decir que cualquier circuito digital puede implementarse ya sea con compuertas NAND o NOR. A continuación, presentamos la universalidad de las compuertas NAND.



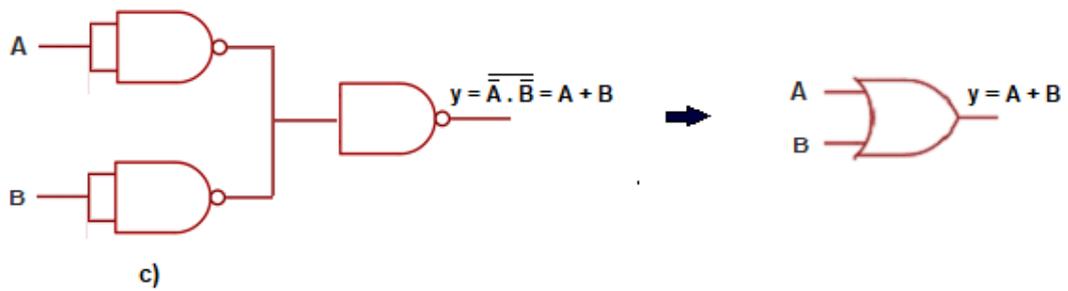


Figura 9.6: Circuitos equivalentes con compuertas NAND de las compuertas básicas: **a)** Inversor **b)** AND **c)** OR

9.5 El Algebra booleana

El álgebra booleana es la matemática que rige el funcionamiento de las compuertas básicas, así como de los sistemas digitales compuesto por ellas. En esta algebra una variable lógica toma el 0 lógico o el valor 1 lógico. En esta algebra existen tres operaciones básicas, las mismas que se ejecutan con su correspondiente compuerta, vistas en el apartado anterior.

Teoremas booleanos	
(1) $x \cdot 0 = 0$	(13) $x + (y + z) = (x + y) + z = x + y + z$
(2) $x \cdot 1 = x$	(14) $x(yz) = (xy)z = xyz$
(3) $x \cdot x = x$	(13a) $x(y + z) = xy + xz$
(4) $x \cdot \bar{x} = 0$	(13b) $(w + x)(y + z) = wy + xy + wz + xz$
(5) $x + 0 = x$	(14) $x + xy = x$
(6) $x + 1 = 1$	(15a) $x + \bar{x}y = x + y$
(7) $x + x = x$	(15b) $\bar{x} + xy = \bar{x} + y$
(8) $x + \bar{x} = 1$	(16) $\overline{(x + y)} = \bar{x} \cdot \bar{y}$ (T. Morgan)
(9) $x + y = y + x$	(17) $\overline{(x \cdot y)} = \bar{x} + \bar{y}$ (T. Morgan)
(10) $x \cdot y = y \cdot x$	

Tabla 9.1: Operaciones del algebra booleana (teoremas)

9.6 Representación alternativa de las compuertas lógicas

Otra consecuencia de los teoremas de Boole es que podemos obtener nuevas representaciones de las compuertas básicas estudiadas en el apartado anterior, para lo cual se utilizará nuevas representaciones para las compuertas AND, OR, NAND, NOR y NOT. A continuación, se presenta la representación alternativa para cada compuerta, tal como se muestra a continuación:

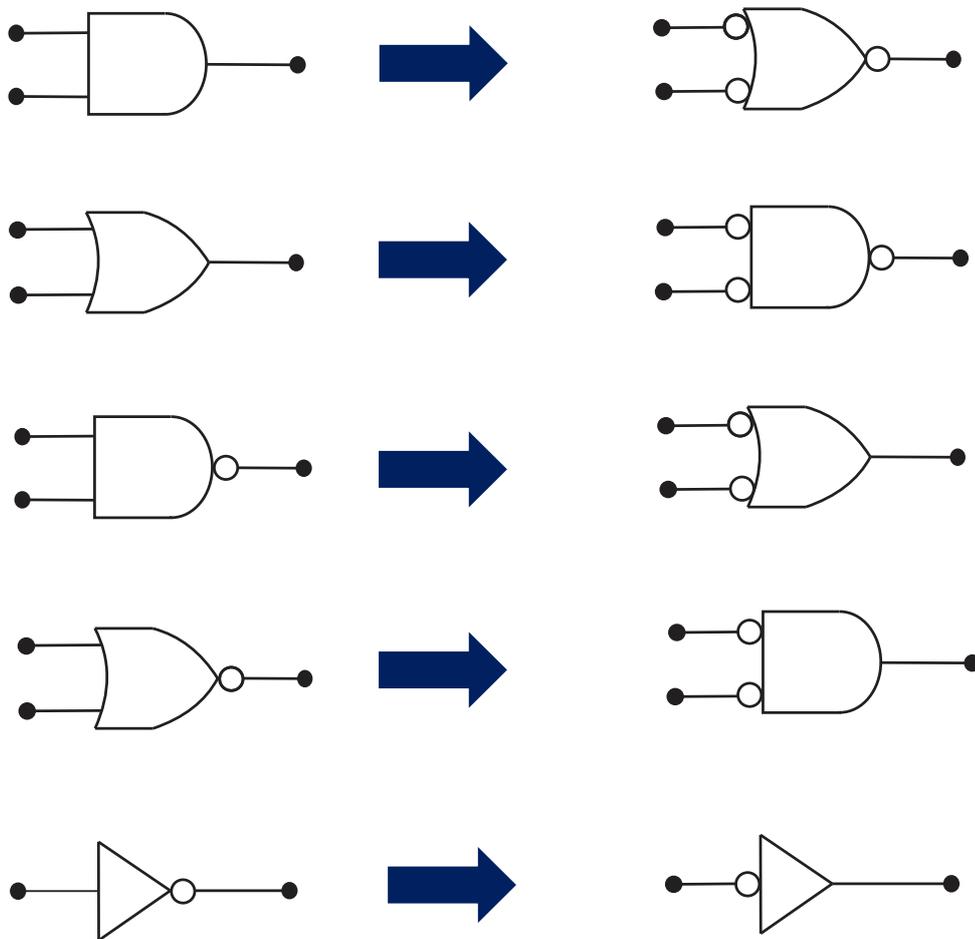


Figura 9.7: Representación alternativa de las compuertas digitales

Los circuitos integrados TTL que se muestran en las figuras 9.1, 9.2, y 9.3 están disponibles en circuitos integrados (CI). Cada CI es *cuádruple*, lo que significa que contiene *cuatro* compuertas idénticas en un solo microcircuito.

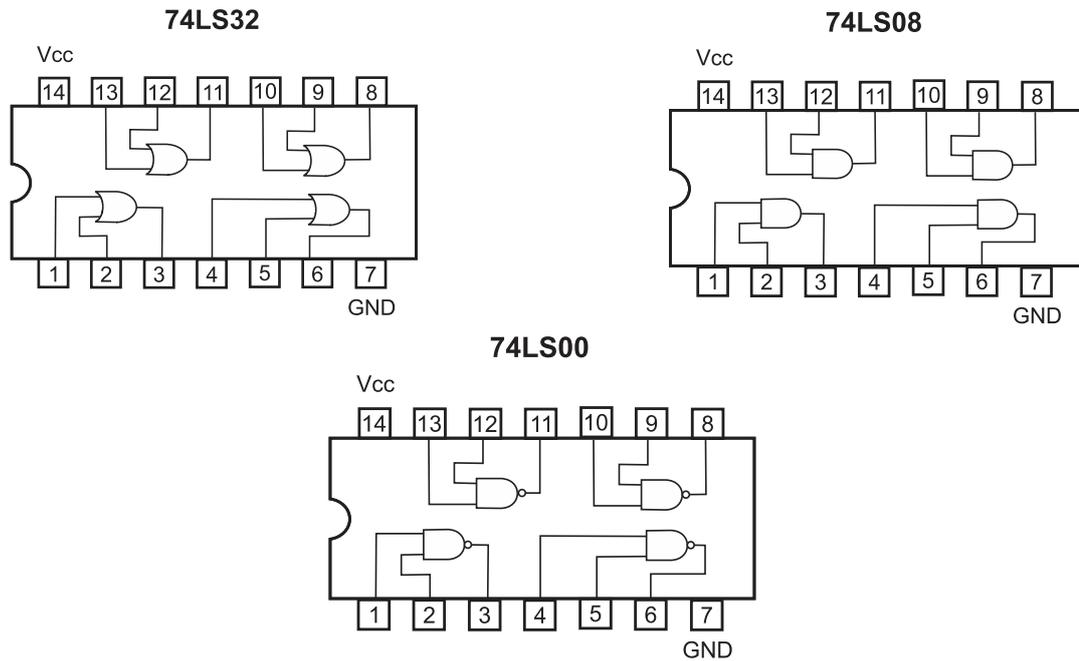


Figura 9.8: Circuitos integrados cuádruples con compuertas OR, AND Y NOR

9.7 Símbolos lógicos del ESTÁNDAR IEEE/ANSI

Desde 1984 también se viene utilizando símbolos lógicos como la Representación Estándar **IEEE/ANSI** 91-1984 para cada uno de los símbolos lógicos. Estos símbolos se representan a través de rectángulos para todas las compuertas y circuitos lógicos. Estos símbolos son:

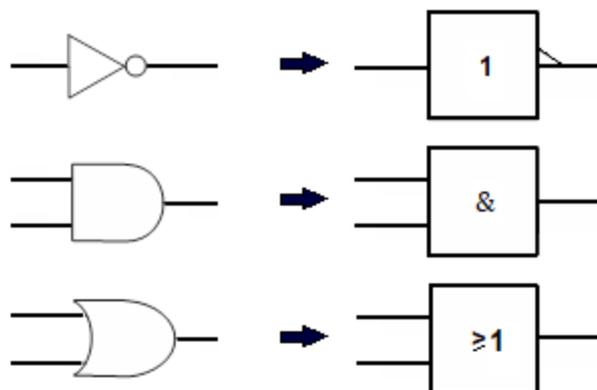


Figura 9.9: símbolos lógicos del estándar IEEE/ANSI

CAPÍTULO X

FLIP – FLOPS Y DISPOSITIVOS RELACIONADOS

10.1 Introducción

Para un adecuado funcionamiento, los circuitos digitales emplean datos binarios. Estos han sido diseñados para realizar funciones como el conteo de pulsos, detección de señales, ejecución de operaciones binarias, entre otros. Sin embargo, según el tipo de funcionamiento de las compuertas digitales, los datos en sus salidas varían en función de los datos presentes en sus entradas. Ahora si las entradas cambian, las salidas lo harán también. Pero a veces es necesario mantener los datos de salida por un determinado tiempo, problemas que ha sido solucionado utilizando las memorias, las cuales básicamente pueden definirse como sistemas de almacenamiento de información que evitan su pérdida.

Los Flip-flops (FF) son la base de una memoria digital. Estas están conformadas por la asociación de compuertas conectadas bajo condiciones de retroalimentación. Si los flipflops se asocian bajo ciertas reglas, pueden almacenar mucha información digital que se pueden maniobrar bajo cierta polarización de estos sistemas.

El circuito en bloque general para un Flip-Flop (FF) se presenta a continuación:

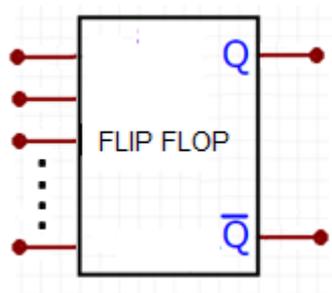


Figura 10.1: Modelo en bloque de un Flip-flop

Como se puede ver en el circuito de bloque de la Figura 10.1, los FF tienen varias entradas y dos salidas, una salida normal Q y una salida \bar{Q} complementada. Las

salidas de los FF poseen siempre dos estados binarios y cuentan además con valores opuestos como se puede observar en la siguiente tabla:

Q	\bar{Q}	Estado de salida
1	0	Estado ALTO o 1 SET (Establecer)
0	1	Estado BAJO o 0 CLEAR (Restablecer o borrar)

Tabla 10.1: Estados de un Flip-Flop

A un FF también se le conoce como: "Registro Básico" por ser el elemento de memoria base para construir sistemas de memoria de mayor capacidad. También se le llama "Multivibrador Biestable", debido a que presentan dos estados estables igualmente probables.

10.2 Flip-Flop construido con compuertas NAND

Un flip flop elemental se puede construir con dos compuertas NAND o dos compuertas NOR cada una de las cuales tienen dos entradas. En la Figura 10.2 se muestran un flip flop formado construido con dos compuertas NAND. A este registro también se le conoce como Registro Básico NAND.

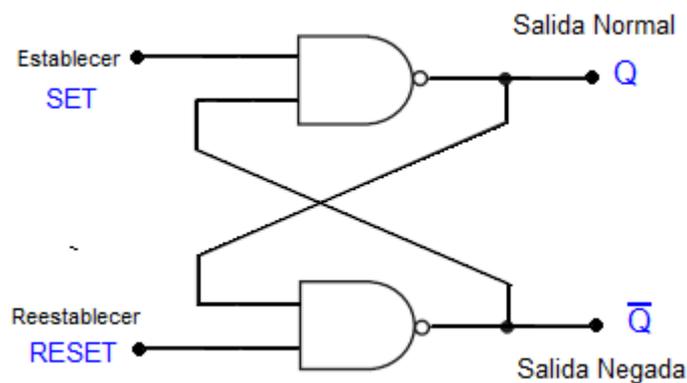


Figura10.2: Elementos internos de un Flipflop utilizando compuertas NAND

A continuación, se presenta dos estados igualmente probables que tiene un registro básico NAND cuando su entrada Establecer=1 y Restablecer=1 flip flop puede estar

en cualquiera de los dos estados: $Q=0$ o $Q=1$. En seguida se presenta la Tabla de verdad correspondiente:

Entrada	Q	Q
Establecer =1	1	0
Restablecer = 1	0	1

Tabla 10.2: Estados de un Flipflop con compuertas NAND

Si las entradas Establecer=1 y Restablecer=1 y, estando el flip flop en cualquiera de sus estados indicados en la Tabla 10.2, este se mantendrá en su estado. Si Establecer=0 y Restablecer=1 esto hará que la salida $Q=1$, manteniéndose en ese estado aun si la entrada Establecer recobre su valor 1. Si Establecer=1 y Restablecer=0, ahora $Q=0$ se mantiene en este estado aun si Restablecer recobre su valor 1. Por último, si se aplica dos pulsos negativos a las dos entradas simultáneamente se observa que ambas salidas asumen el valor $Q = \bar{Q} = 1$, por lo que se considera que es invalido. El resumen de esto se observa en la Tabla 10.3

ACCIÓN	ENTRADAS		Q
No se aplica pulso	Establecer	1	Sin cambio (Puede ser 0 o 1)
	Restablecer	1	
Se aplica Pulso BAJO a SET	Establecer	0	1
	Restablecer	1	
Se aplica Pulso BAJO a RESET	Establecer	1	0
	Restablecer	0	
Se aplica Pulso BAJO a SET y RESET	Establecer	0	Inválido
	Restablecer	0	

Tabla 10.3: Cambios de las salidas del FF con compuertas NAND, según cada selección de entradas

Aquí la S representa SET y R representa RESET. También en algunos textos se usa la letra C de CLEAR en lugar de la R de RESET. Asimismo, esta representación alternativa nos permite establecer que ambas entradas son activas en bajo, como lo demostrado. En la Tabla 10.4 se presenta la tabla de verdad de un flip flop con compuertas NAND

S	R	Salida
1	1	Sin cambio
1	0	0
0	1	1
0	0	Inválido

Tabla 10.4: Registro Básico NAND y su tabla de verdad

La representación alternativa de este FF, debido a la equivalencia entre las compuertas que lo sustituyen, está dado por la siguiente representación alternativa

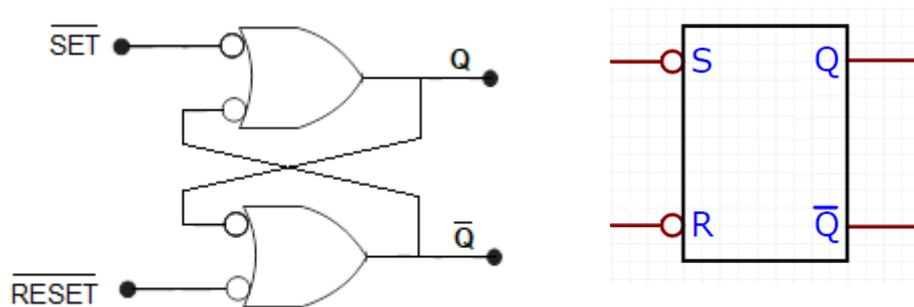


Figura 10.3: a) Representación alternativa del Registro Básico NAND, b) símbolo

10.3 Flip-flop usando compuertas NOR

El flip-flop NAND funciona de igual forma que el flip-flop NOR, solo que las repuestas a las condiciones en sus entradas, seran distintas. La constitución y la

tabla de verdad de este flip flop usando compuertas NOR se presenta en la Figura 10.4.

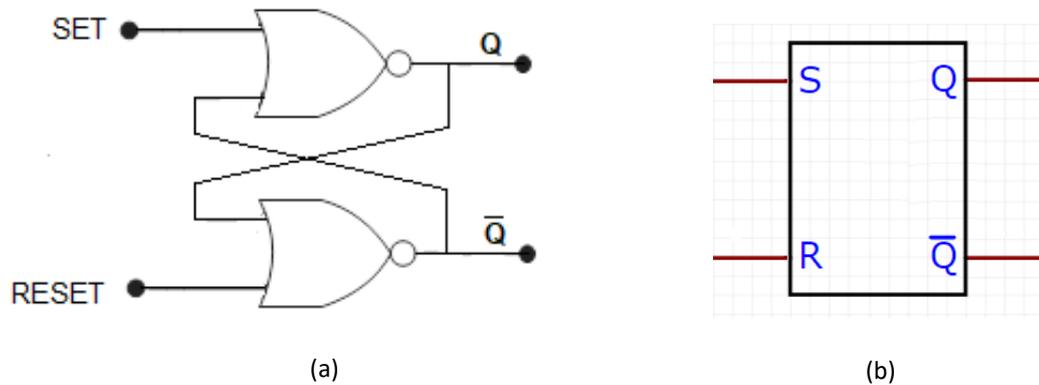


Figura 10.4: a) FF con compuertas NOR b) Circuito en bloque

S	R	Salida
0	0	Sin cambio
1	0	1
0	1	0
1	1	Inválido

(c)

Tabla 10.5: Tabla de verdad del Registro Básico NOR

Los usos más comunes de los Registros Básicos NAND o NOR en sistemas digitales son los siguientes:

- Reemplazan a los interruptores mecánicos suprimiendo el ruido que causan.
- En el encendido o apagado de circuitos digitales y analógicos.
- Son la base para la construcción de sistemas con sensores.
- Otras aplicaciones.

10.4 Señales de Reloj

Todo sistema digital cuenta con dos formas de operación básica. La llamada asíncrona donde las salidas cambian teniendo en cuenta las órdenes de las entradas

y el modo síncrono cuando las salidas cambian de acuerdo con los estados de las entradas, pero solo cuando hay una señal de reloj.

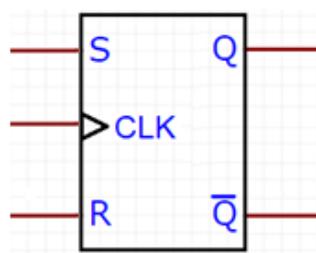
Un circuito digital síncrono trabaja con una señal de reloj que es una onda cuadrada o rectangular.

10.5 Flip-flops sincronizados por Reloj

Este tipo de FF tienen una entrada de reloj a la que se le denomina: RELOJ, CLOCK, CLK o CP. Si esta entrada tiene un círculo se dice que es activa a una TPN y si la entrada no la tiene se dice es activa ante una TPP. Las entradas de control en un FF determinan a que estado ira, mientras que la entrada de reloj determina en qué momento alcanzara dicho estado.

10.6 Flip-flop tipo S-R sincronizado por señal de reloj

A continuación, estudiaremos el funcionamiento del FF tipo S-R sincronizado por una señal de reloj, cuya representación se muestra en la siguiente Figura 10.6. Como la entrada de reloj CLK no tiene un círculo lo cual indica que esta entrada es activa ante una TPP. Se acostumbra solo presentar la salida Q, dado que la otra salida \bar{Q} tiene un valor complementado.



(a)

S	R	CLK	Q
0	0	TPP	No cambia
1	0	TPP	1
0	1	TPP	0
1	1	TPP	Ambigua

(b)

Figura 10.6: a) Flip Flop tipo SR sincronizado por reloj b) y su tabla de verdad

10.7 Flip-flop sincronizado por señal de reloj tipo J-K

Este FF no presenta una salida ambigua. Este FF tiene dos entradas J, K, CLK y sus salidas complementada y no complementada. Su mayor uso es como un conmutador lo que se logra haciendo las entradas $J = 1$, $K = 1$ lo que convierte en el FF más utilizado en diferentes aplicaciones. El diagrama de este FF y su Tabla de verdad se presentan a continuación:

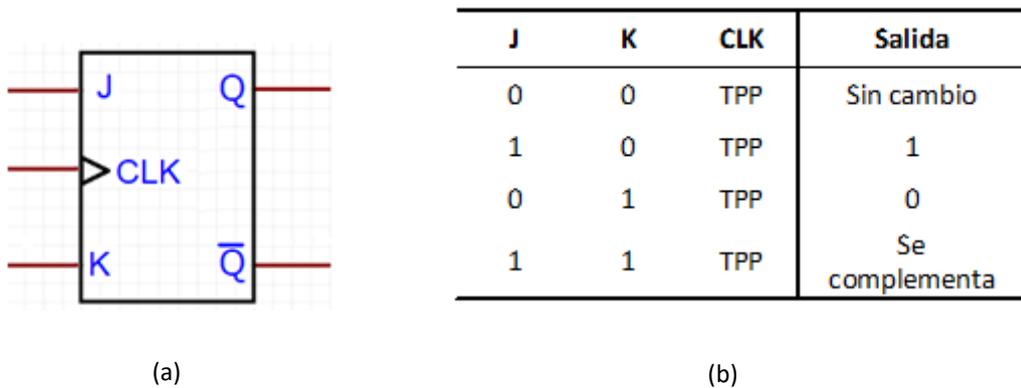


Figura10.6: a) Flip-Flop tipo JK sincronizado por reloj b) y su tabla de verdad

10.8 Flip flop tipo sincronizado por señal de reloj tipo D

Este FF cuenta con solo una entrada de datos además de la entrada de señal de reloj a través de las cuales se controla las salidas. Su operación es muy sencilla: Q cambiara al mismo estado que esté presente en la entrada transición en la entrada de reloj. El diagrama de este circuito digital y su tabla de verdad se presentan.

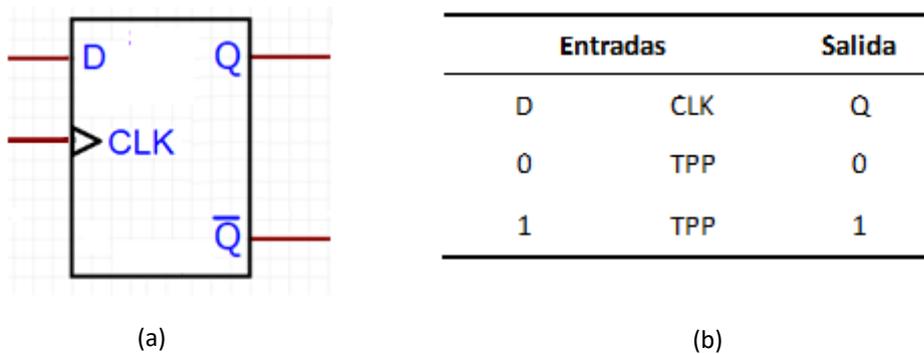


Figura10.7: a) Flip-flop tipo D sincronizado por reloj b) y su tabla de verdad

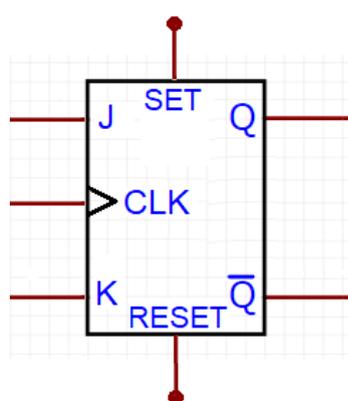
Este flip-flop tiene muchas aplicaciones, dentro de las cuales se encuentra la aplicación en la transmisión de datos al recibir datos de un circuito lógico combinacional. Siempre funciona un FF tipo D por cada salida de los circuitos combinacional.

10.9 Entradas asíncronas en los FF

La mayoría de los FF sincronizados por reloj adicionalmente tienen una o más entradas asíncronas que operan de forma independiente a las entradas sincronas y a la entrada de reloj. Estas entradas son las predominantes. La figura siguiente no muestra los símbolos de los FF tipo J-K y el Tipo D donde aparecen las entradas asíncronas.

10.10 El FF Tipo J-K con entradas asíncronas

La figura muestra un flipflop tipo J-K con dos entradas síncronas, una entrada de reloj activa ante una TPP y dos entradas asíncronas que son activas en ALTO. El diagrama en bloque de esta compuerta y la tabla de verdad correspondiente se presenta en la Figura 10.8. Aquí X representa la condición no importa, lo cual significa puede tomar el valor ALTO O BAJO.



(a)

Entradas					Salida
J	K	SET	RES	CLK	Q
0	0	0	0	TPP	No hay cambio
1	0	0	0	TPP	1
0	1	0	0	TPP	0
1	1	0	0	TPP	Se complementa
X	X	1	0	X	0
X	X	0	1	X	1

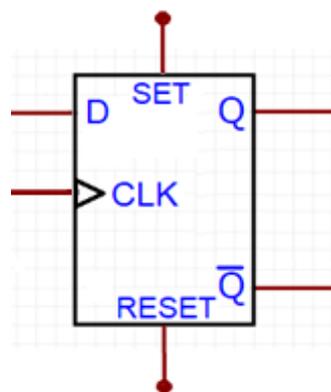
(b)

Figura 10.8: a) Flip-Flop tipo JK sincronizado por reloj, con entradas asíncronas b) y su tabla de verdad

En la tabla de verdad se puede apreciar que el funcionamiento del FF tipo J-K síncrono se sigue conservando, sin embargo, donde además aparece que la salida Q del FF será obligada a que Q asuma el estado BAJO o que alcance el estado ALTO según se lo que se establezca en las entradas SET y RES según la tabla.

10.11 El FF Tipo D con entradas asíncronas

También es gran aplicación FF tipo D, que además de tener su entrada síncrona y entrada de reloj, también tiene dos entradas asíncronas predominantes.



(a)

Entradas			Salida	
D	SET	RESET	CLK	Q
0	0	0	TPP	0
1	0	0	TPP	1
X	1	0	X	0

(b)

Figura10.9: a) Flip-flop tipo D sincronizado por reloj con entradas asíncronas b) y su tabla de verdad

Al igual que en el caso anterior, este flip-flop sigue conservando sus características, sin embargo, las entradas asíncronas son las que determinan de forma predominante el valor de entrada que aparecerá en la salida.

CAPÍTULO XI

MAPAS DE KARNAUGH

11.1 Introducción

Un método efectivo para simplificar un circuito digital es utilizando el método gráfico de Karnaugh el cual nos permite obtener una función lógica reducida para la variable de salida, la cual depende de las variables entrada del circuito. C

11.2 Mapas de Karnaugh y el número de entradas

La forma y la distribución de variables de entrada del circuito en los mapas de Karnaugh se presentan para dos hasta cinco variables digitales de entrada. Estos mapas son los siguientes:

	\bar{B}	B
\bar{A}		
A		

Tabla 11.1 Mapas de Karnaugh para dos variables digitales de entrada

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
\bar{A}				
A				

Tabla 11.2 Mapas de Karnaugh para tres variables digitales de entrada

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$				
$\bar{A}B$				
AB				
$A\bar{B}$				

Tabla 11.3 Mapas de Karnaugh para cuatro variables digitales de entrada

	$\bar{C}\bar{D}\bar{E}$	$\bar{C}\bar{D}E$	$\bar{C}D\bar{E}$	$\bar{C}DE$	$C\bar{D}\bar{E}$	$C\bar{D}E$	$CD\bar{E}$	CDE
$\bar{A}\bar{B}$								
$\bar{A}B$								
AB								
$A\bar{B}$								

Tabla 11.4 Mapas de Karnaugh para cinco variables digitales de entrada

Los mapas anteriores sirven para albergar todos los valores digitales de la salida en una distribución bidimensional de acuerdo con el número de variables de entrada. Para caso específico se tendrá una configuración de unos y ceros según lo precise la tabla de verdad. Con esta técnica los circuitos digitales desde dos hasta cinco entradas se pueden simplificar utilizando estos mapas.

11.3 La adyacencia de los unos lógicos

En un mapa K se ubican los ceros o unos de acuerdo con lo indique la tabla de verdad ya que esta nos dará información de como varía la salida del circuito digital en función de los valores que asumirán las variables de entrada. Se forman grupos que son adyacentes entre sí, los cuales pueden ser grupos de dos unos, grupos de cuatro unos, grupos de ocho unos o grupos de dieciséis unos. Cada agrupación tendrá un conjunto de términos donde aparecen variables de forma complementada y no complementada. Para un grupo de dos unos se elimina una variable. Para un grupo de cuatro se elimina dos variables, para un grupo de ocho se elimina tres variables y así sucesivamente.

11.4 Obtención de un mapa K a partir de una tabla de verdad

Una tabla de verdad se puede obtener a partir de las proposiciones de un problema específico a resolver, para el cual se proyecta diseñar un circuito digital.

Supongamos que partimos una tabla de verdad, sin considerar el problema que se puede resolver con esta. Supongamos que se tiene una tabla de verdad que

corresponde a la tarea que desarrollará un circuito digital con entradas A, B y C con una salida S. Esto se puede observar en la Tabla 11.5.

A	B	C	S	
0	0	0	1	ABC
0	0	1	1	$\bar{A}\bar{B}C$
0	1	0	0	
0	1	1	1	$\bar{A}BC$
1	0	0	1	$A\bar{B}\bar{C}$
1	0	1	0	
1	1	0	0	
1	1	1	0	

Tabla 11.5: Tabla de verdad para un circuito lógico con tres entradas y una salida

Se inicia asignado creando la operación AND de la variable complementada o no de la entrada A, B y C, solo para las filas cuyo valor de la salida es uno lógico. El método se aplica por ejemplo para el uno de la primera fila. En este caso, para lograr que la operación AND de $A = 0$, $B = 0$ y $C = 0$ sea 1 (uno), necesariamente debemos complementar las tres variables de la siguiente forma: $\bar{A} = 1$, $\bar{B} = 1$ y $\bar{C} = 1$ de tal forma que ahora si $\bar{A}\bar{B}\bar{C} = 1$ (ver Tabla 11.5). Para la siguiente fila: $A = 0$, $B = 0$ y $C = 1$, por tanto, para lograr que la salida sea 1, se debe considerar $\bar{A} = 1$, $\bar{B} = 1$ y $C = 1$, consecuentemente $\bar{A}\bar{B}C = 1$. Se continua hasta terminar con las siguientes filas cuya salida es 1, tal como se muestra en la Tabla 11.5.

Para cada expresión se ubica un 1 en el mapa de Karnaugh de tres entradas, tal como se presenta en la Tabla 11.6.

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
\bar{A}	1	1	1	
A	1			

Tabla 11.6: Mapa de Karnaugh con los unos de la tabla de verdad

después se precede a agrupar según el siguiente orden:

- Se encierra con un círculo aquellos unos que no son adyacentes a ningún otro uno.
- Se encierra el uno que es adyacente a solo otro uno (pares) .
- Se encierra los grupos de ocho unos (octetos).
- Se encierra los grupos de cuatro unos (cuádruples).
- Se encierra en grupos los pares que hayan quedado libres

Considerando estos pasos obtenemos las siguientes agrupaciones de pares en el mapa de Karnaugh que se observan en la Tabla 11.7.

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
\bar{A}	1	1	1	
A	1			

Figura 11.7 Mapas de Karnaugh para cinco variables digitales de entrada

Para el primer grupo de color azul se observa las expresiones obtenidas en el extremo de la correspondientes a los unos de las los dos unos es la intersección de la primera columna y las filas. Aquí observamos que involucra cuatro variables como: $\bar{A}\bar{A}\bar{B}\bar{C}$. En este término aparece de manera complementada y no complementada la entrada A por lo que se elimina de este término, quedando como $\bar{B}\bar{C}$. De igual forma para el grupo de color verde observamos que este es la intersección de la primera fila y la segunda y tercera columna. Este grupo involucra a las variables $\bar{A}\bar{B}CBC$ donde aparece de forma complementada la variable B. Por tanto, nos queda el termino $\bar{A}CC = \bar{A}C$. Para finalizar se realiza la operación OR de estos dos términos tal como sigue:

$$S = \bar{B}\bar{C} + \bar{A}C$$

El circuito digital expresado a partir de la expresión reducida obtenida del mapa de Karnaugh es el siguiente:

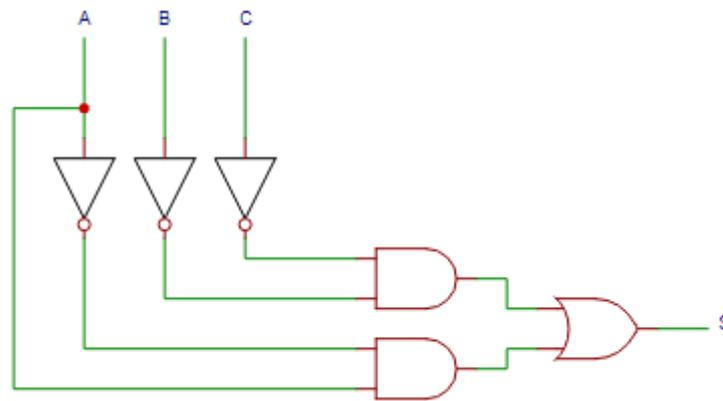


Figura 11.1 Circuito digital del caso resuelto

Como un ejemplo se utilizará de la técnica para resolver el caso de una turbina eléctrica cuyo comportamiento depende de los estados de tres interruptores A, B y C. Projete un circuito digital que permita el funcionamiento de la turbina según las condiciones de funcionamiento que se indican:

- Si los tres interruptores están en estado ON la turbina rota.
- Si se colocan en ON dos interruptores cualesquiera y el tercero en OFF, la turbina rota, pero suena una bocina que alerta.
- Si se coloca en ON a un solo interruptor y los demás están en OFF, la turbina no rota, pero si se emite el sonido de emergencia.
- Si los tres interruptores están en OFF por ende la turbina no rota, ni tampoco la bocina emite sonido.

El presente caso se soluciona escribiendo la tabla de verdad para el circuito donde se considera todos los valores posibles de las entradas y los valores lógicos de sus dos salidas determinadas a partir de las condiciones puntuadas en el presente caso.

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

Tabla 11.8: Tabla de verdad para el circuito lógico con tres entradas y dos salidas

Se construye para cada salida su mapa de Karnaugh. Se inicia considerando solamente la salida para la turbina T que es para el motor:

A	B	C	T	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{A}BC$
1	0	0	0	
1	0	1	1	$A\bar{B}C$
1	1	0	1	$AB\bar{C}$
1	1	1	1	ABC

Tabla 11.9: Tabla de verdad para la salida de la turbina y sus expresiones canónicas

Reemplazando en el mapa de Karnaugh para todos los unos de la tabla Tabla 11.9, se tiene lo siguiente:

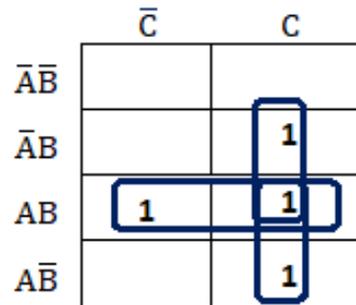


Figura 11.10 Mapas de Karnaugh para la turbina

En este mapa de Karnaugh se agrupa a los unos que solo son adyacentes a solo otro uno. Después se reduce la expresión de cada par eliminando la variable que aparece de forma complementada y no complementada. después se desarrolla la operación OR de estos términos reducidos la cual se asigna a la salida T según se indica a continuación:

$$T = BC + AC + AB$$

Ahora se analiza la salida S que para la lámpara:

A	B	C	S	
0	0	0	0	
0	0	1	1	$\bar{A}\bar{B}C$
0	1	0	1	$\bar{A}B\bar{C}$
0	1	1	1	$\bar{A}BC$
1	0	0	1	$A\bar{B}\bar{C}$
1	0	1	1	$A\bar{B}C$
1	1	0	1	$AB\bar{C}$
1	1	1	0	

Tabla 11.12 Tabla de verdad para la bocina

Al igual que en el caso anterior se obtiene el mapa de Karnaugh despues de haber asignado los unoas de la tabla de verdad según las expresiones indicadas en la Tabla 11.12:

	\bar{C}	C
$\bar{A}\bar{B}$		1
$\bar{A}B$	1	1
AB	1	
$A\bar{B}$	1	1

Figura 11.11 Mapas de Karnaugh para la bocina

Tal como se hizo en el caso anterior se obtiene la ecuación algebraica para la salida de la bocina, la cual es como sigue:

$$S = \bar{A}C + A\bar{B} + B\bar{C}$$

Finalmente, el circuito simplificado para este ejemplo se presenta a continuación:

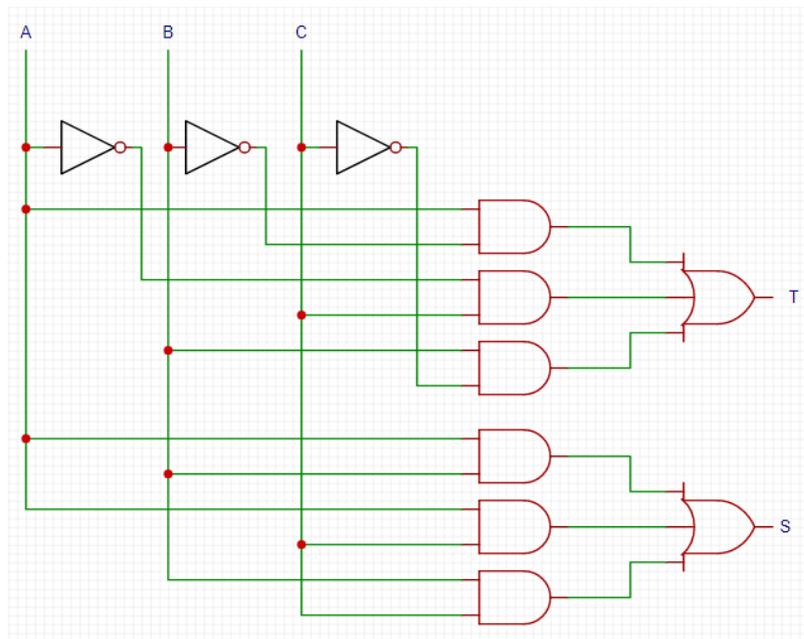


Figura 11.2 Circuito digital que resuelve el caso planteado

CAPÍTULO XII

CONTADORES Y REGISTROS

12.1 Introducción

Se les llama circuitos contadores porque cuentan el número de pulsos expresado en binario de una señal digital. Estos tienen una entrada para la señal digital, normalmente una señal de reloj, y a su vez una salida compuesta por muchas líneas igual al número de bits del número de conteo.

Este circuito digital puede contar pulso de forma ascendente o de forma descendente. Asimismo, pueden operar de forma asíncrona o síncrona. Los circuitos contadores síncronos son los más utilizados porque en ellos se puede controlar el número máximo de conteo.

También, los contadores pueden ser utilizados como divisores de conteo, es decir se puede dividir la frecuencia de la señal digital de entrada. La frecuencia de la señal dividida dependerá de la línea que se tome la salida del contador.

Podemos también conectar la secuencia de FF en serie y en paralelo. Estos últimos tienen un tiempo de respuesta menor que los primeros.

La mayoría de los contadores utilizan FF tipo JK los cuales pueden tener entradas síncronas o no.

12.2 Contadores asíncronos de cuatro bits

Este tipo de contador está compuesto por cuatro flipflops tipo J y K, en conexión en serie o paralelo, que tienen sus entradas J y K conectadas a 1 lógico. Esto permite que cada uno tenga su salida y que se vaya complementado, tras el ingreso de la señal de reloj en su entrada correspondiente. Un circuito digital en serie de este contador se presenta a continuación:

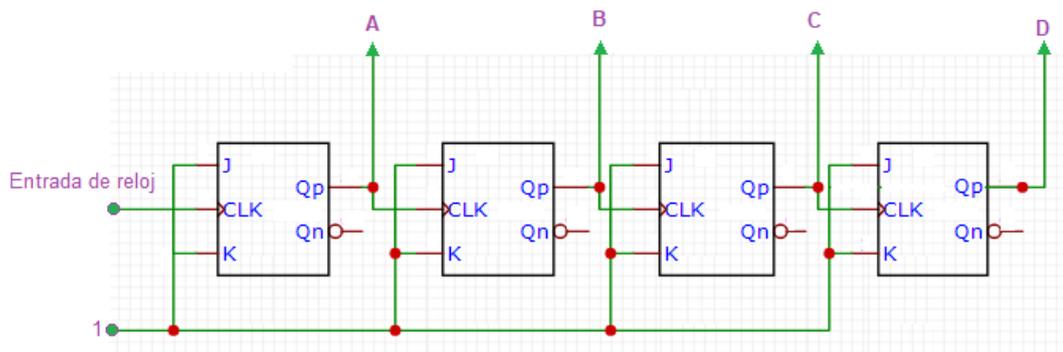


Figura 12.1: Contador asíncrono de cuatro bits.

Este tipo de contadores utiliza flipflops tipo JK que están conectados de tal forma que el pulso de reloj solo se aplica al primer flipflop. Todos los demás flipflop conectados en serie tiene las entradas JK conectadas a 1s lógicos, permitiendo que la señal de salida de cada flipflop se complemente toda vez que las entradas de reloj este frente a una transición de pendiente positiva.

La tabla 12.1 presenta los distintos estados por los que pasa el contador que se pueden medir en A, B, C, D.

Cuenta decimal	Cuenta Binaria				Cuenta Decimal	Cuenta binaria			
	D	C	B	A		D	C	B	A
0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	1	0	0	12	1	1	0	0
5	0	1	0	1	13	1	1	0	1
6	0	1	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1

Tabla 12.1: Secuencia de un contador de 4 bits.

Además, se muestran las gráficas y como varía la señal digital en la salida de cada flipflop. Esta se a presenta continuación.

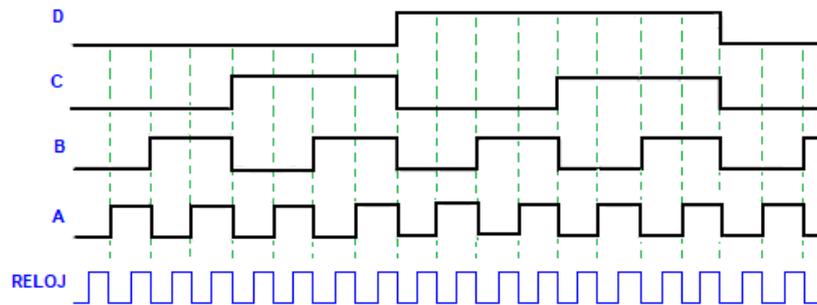


Figura 12.2: Señales en un contador de cuatro bits

12.3 Contadores como circuitos integrados

En la actualidad no se implementa los contadores utilizando FF de forma discreta, sino lo que más se utiliza son contadores en circuitos integrados, los cuales albergan muchos flipflops con conexión interna, y otros con posibilidades de conectar externamente con otras compuertas y/o otros flipflops. Un circuito contador asíncrono en CI muy versátil es el 74192 el cual nos permite contar de manera ascendente o descendente, para lo cual tiene su entrada para cada función, hasta alcanzar en número MOD de 10. Este flipflop también tiene una entrada de borrado colocando todas las salidas a cero lógico. También permite ampliar el número MOD al conectar en cascada con otros contadores en su entrada de arrastre

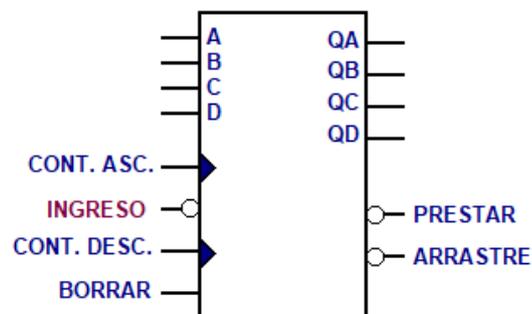


Figura 12.3: Contador digital CI 74192

12.4 Contador binario cuatro bits en circuito integrado

Uno de los contadores de cuatro bits muy utilizados es el 74LS93, el cual utiliza cuatro flipflops tipo JK que funcionan en modo de complementación y no complementación permanente. Este circuito nos presenta tres flipflops que ya están conectados internamente entre sí, dejando al diseñador conectar externamente el cuarto flipflop utilizando una compuerta NAND la cual se conecta según su necesidad. Si el diseñador considera utilizar solo tres flipflops, tendrá la entrada $\overline{CP1}$ de este contador tres bits y si este conecta externamente el cuarto flipflop tendrá como entrada del contador, ahora de cuatro bits, a la entrada $\overline{CP2}$. Asimismo, el CI integrado presenta dos entradas $\overline{MR1}$ y $\overline{MR2}$ que permite cambiar el número MOD del contador.

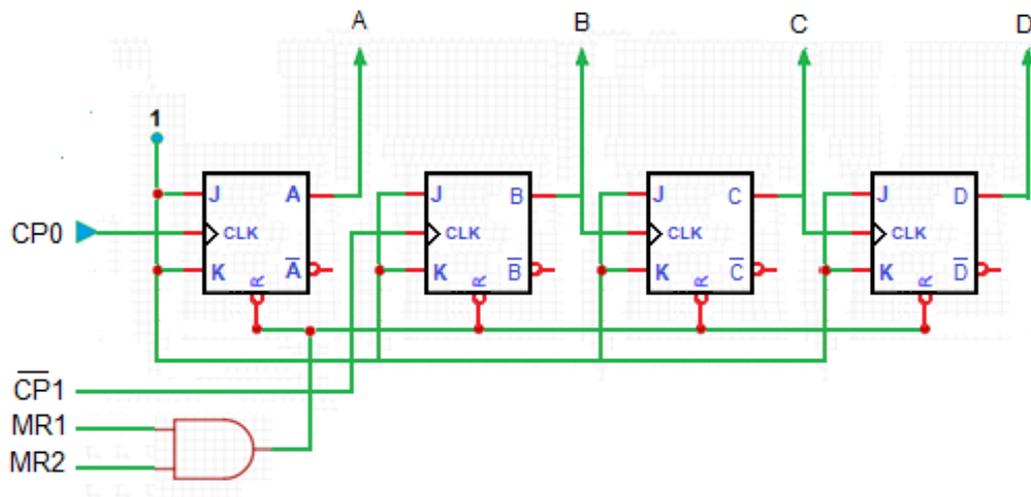


Figura 12. 4: EL contador CI 74LS93

12.5 Contadores paralelos

Los flip-flops son conectados de tal forma que estos accionan al mismo tiempo durante el conteo. Así evitamos el retardo en la propagación de la señal cuando estos están conectados en serie generado, por el tiempo individual de reacción de cada flip-flop.

Se logra cuando todas las entradas de reloj están conectadas entre sí generándose una conexión en común que constituirá como la entrada de datos del contador.

12.6 Contadores síncronos en paralelo

Los contadores que tienen conexión en serie tienen el problema cuando reciben los datos, estos retardan durante la trasmisión de la señal en cada flip-flop, dado que estos no paso de un estado a otros al mismo tiempo tal como cambian los pulsos de entrada. Para evitar estos retrasos se utilizan los flip-flop tal como se ve en la en la figura 12.5.

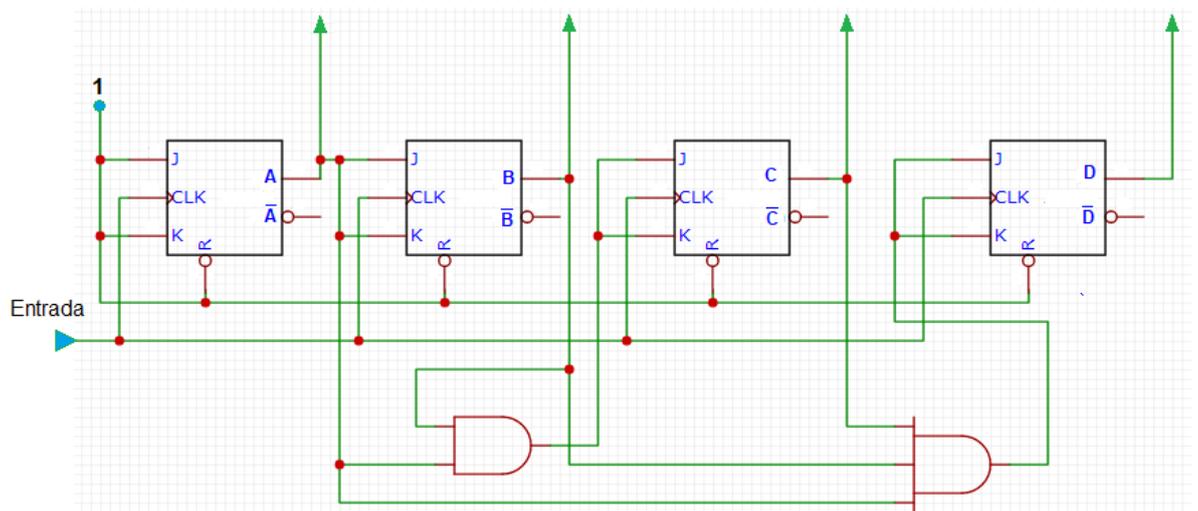


Figura 12.5: Contador binario de cuatro bits

D	C	B	A	n
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	10
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Tabla 12.8: Tabla de conteo del contador en paralelo

La característica más importante de los contadores en paralelo es que la respuesta de estos se basa en la reacción simultánea de todos los flip-flops cuando les ingresa la señal de reloj.

CAPÍTULO XIII

FAMILIAS LÓGICAS EN CIRCUITOS INTEGRADOS

13.1 Familias lógicas

En la fabricación de los circuitos integrados digitales se tiene en cuenta el tipo de tecnología empleada pues esta determina las diferentes características de operación que tienen estos dispositivos, tales como: niveles de tensión, márgenes de ruido, potencia disipada, cargabilidad de entrada y salida, etc.

La construcción de las familias lógicas está basada en una tecnología de transistores determinada. Sus diferentes compuertas presentan distintos comportamientos eléctricos ante las tensiones de entrada, condiciones ambientales existentes, y condiciones de salida. Desde que aparecieron los circuitos digitales siempre se buscó disminuir el tamaño de los circuitos, la velocidad de respuesta, la degeneración por envejecimiento de los componentes, las tolerancias y la disminución de potencia consumida, entre otras características.

13.2 Tecnologías TTL y CMOS

Los parámetros más importantes de las compuertas TTL son el retardo de propagación (ns), la disipación de potencia (m W), y el producto velocidad - potencia (pJ). El producto velocidad-potencia indica un retardo en la propagación con una disipación de potencia determinada.

Características Generales de los Circuitos Digitales

Los circuitos digitales poseen:

- Fan Out: Es el número máximo de salidas que puede soportar una compuerta sin alterar su operación normal.
- Fan In: Es el máximo número de entradas que puede tener una compuerta.

- Tensión de Umbral: Si bien es cierto que las variaciones en el voltaje no son tan críticas en las compuertas digitales, sin embargo, se distinguen dos tensiones de umbral; la primera para el estado lógico cero y la segunda para el estado lógico uno.
- Margen de ruido: valor máximo de ruido que puede soportar la entrada de un dispositivo lógico, sin que cambie el estado de la salida. Tanto para estado lógico cero como para el estado lógico uno existe un margen de un ruido.

13.3 Potencia disipada: Floyd (2006), define esta potencia como la energía por unidad de tiempo que consume r una compuerta. La potencia disipada versus la frecuencia en una compuerta TTL se mantiene constante cuando cambia la frecuencia. Contrariamente, la compuerta CMOS cambia al variar la frecuencia

Familia TTL (Lógica de Transistor - Transistor)

Los circuitos de tecnología TTL utilizan transistores bipolares y resistencias de polarización. Estos aceptan una tensión nominal de alimentación de 5 VDC.

Niveles Lógicos TTL

En los circuitos lógicos, se pueden reconocer cuatro rangos lógicos diferentes: V_{EB} , V_{EA} , V_{SB} y V_{SA}

En este caso V_{EB} representa la tensión de entrada válida desde 0 a 0.8 V lo cual significa un nivel lógico 0 (BAJO). El rango de tensión V_{EA} significa la tensión entrada valida desde 2 y 5 V y representa un 1 lógico (ALTO). Además, hay un rango de tensión entre 0.8 y 2 V que no es permitido. Adicionalmente se han planteado rangos de tensiones de salida V_{SB} , V_{SA} en un dispositivo digital. Esto se ve en la figura 13.1

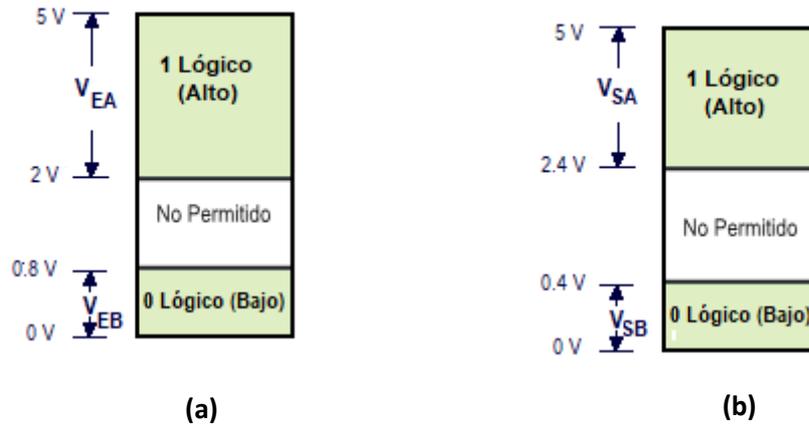


Figura 13.1. Niveles lógico de un circuito TTL (a) entrada (b) salida.

13.4 Circuitos Lógicos CMOS (Metal Óxido Semiconductor Complementario)

La tecnología CMOS es la que más se emplea hoy en día para construir circuitos integrados digitales como, por ejemplo, las memorias y los microprocesadores. Estos circuitos usan una tensión nominal de alimentación de +5 V y +3,3 V.

Niveles Lógicos CMOS

En la figura 132. Se aprecian las tensiones V_{EB} , V_{EA} , V_{SB} , V_{SA} usadas para los dispositivos CMOS de nivel +5 VDC.

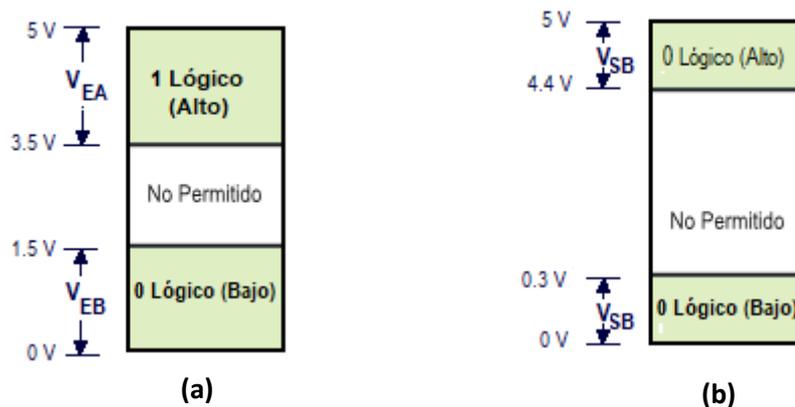


Figura 13.2. Nivel lógico de un circuito CMOS (a) entrada (b) salida

Estos dispositivos funcionan internamente porque se rigen por dos estados bien diferenciados a los cuales se les asigna los valores lógicos de uno o cero. El acuerdo

es asignar valores lógicos uno al interruptor y otro valor lógico de cero al interruptor abierto. Y además en vez de interruptores se usan transistores bipolares o unipolares.

Configuraciones de Salida en las Compuertas con tecnología TTL

Sin considerar lo que desarrollarán las compuertas lógicas con tecnología TTL, su salida tendrá cualquiera de las configuraciones que se indican a continuación:

- a) Colector Abierto
- b) Totém - Pole
- c) Tres Estados

Salida con Colector Abierto de una compuerta TTL

La compuerta básica TTL de tres entradas fue una modificación de la categoría anterior DTL. A esta compuerta la podemos ver en la figura 13.3.

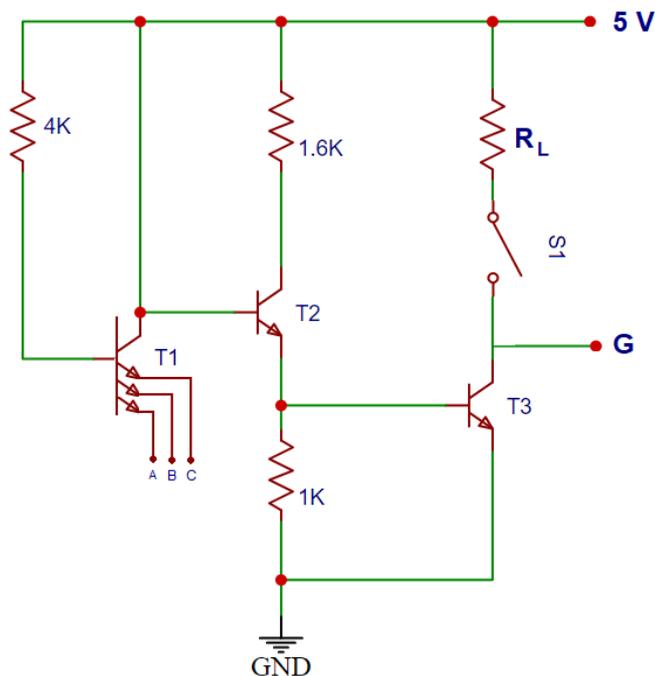


Figura 13.3. Compuerta NAND con tres entradas TTL de colector abierto

Si se quiere llevar a la salida a alto, necesariamente el interruptor S1 debe estar cerrado cuando el transistor T_3 está en corte.

Si el resultado de los niveles lógicos de entrada fuese cero la unión base-emisor en T_1 se polariza de forma directa. Por tanto, el voltaje en base del transistor T_1 es igual a 0.9 Voltios

$$\text{Tensión de entrada} + \text{Tensión } be \text{ en } T_1 = \text{Tensión } b \text{ en } T_1$$

$$0.2 + 0.7 V = 0.9 V$$

En el momento que la suma de las caídas de las tensiones de V_{bc} en T_1 , V_{be} en T_2 , y V_{be} en T_3 , estén sobre a 1.8 V, el transistor T_3 se iniciará en su estado de conducción. Dado que la tensión establecida V_b en toma el valor de 0.9.V, se tendrá que el transistor T_3 se fijará en su estado de corte. Por ese motivo, si se conectase una resistencia al colector, se obtendría que la tensión de salida sería **uno lógico**.

Si ocurriese que todos los niveles lógicos de entrada son 1, entonces los transistores T_2 y T_3 entrarán al estado de saturación puesto que la tensión en la base de Q_1 sobrepasa a la suma de las caídas de tensión V_{bc} en T_1 , V_{be} en T_2 , y V_{be} en T_3 . En consecuencia, el estado de salida alcanzará el **cero lógico**.

13.5 Salida Totém – Pole de una compuerta TTL

Las compuertas poseen una impedancia de salida determinada. Esta impedancia es del tipo resistivo capacitivo. Como se sabe toda capacitor se carga de forma exponencial pasando de un estado bajo a otro alto según, cuando el transistor T_3 de salida pasa del estado bajo al alto.

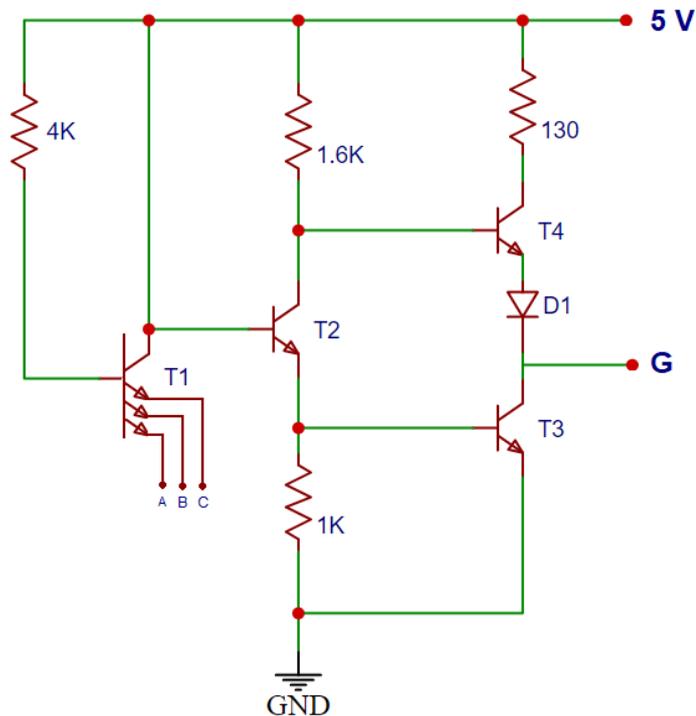


Figura 13.4. Compuerta básica

Cuando la señal en el punto G decae los dispositivos T_2 y T_3 entrarán en saturación idénticamente como sucede en una compuerta de colector abierto

A continuación, la ecuación que se muestra representa el valor de la tensión en el colector de T_2 :

$$\text{Tensión be en } T_3 + \text{Tensión ce en } T_2 = \text{Tensión c en } T_2$$

$$0.7 + 0.2 V = 0.9 V$$

Dado que la salida F toma la tensión V_{ce} en T_3 es $0.2 V$, el transistor T_4 alcanzará el estado de corte por lo siguiente:

$$\text{Tensión be en } T_4 + \text{Tensión en } D_1 = \text{Tensión c en } T_2$$

$$0.7 + 0.7 V > 1.1 V$$

Ya que V_c en T_2 es igual a la tensión V_b en T_4 , el transistor T_4 alcanzará su estado de corte. Aquí se utiliza diodo D_1 para establecer una caída de tensión en la malla y logra así el estado de corte en el transistor T_4 pero con T_3 en su estado de saturación.

Cuando la salida F cambia pasa al estado **uno lógico** a causa de que su entrada pasa a su estado **cero lógicos**, los transistores T_2 y T_3 alcanzarán a su estado de corte. Por ello se mantiene en este estado un breve espacio de tiempo, bajando o su estado bajo debido a que la tensión en el condensador cambia según su respuesta establecido por su constante de tiempo.

Ahora bien, en el instante en que el transistor T_2 pasa a su estado de corte, el transistor T_4 pasa a su estado de conducción a causa de que la tensión en su base se establece a través de la resistencia de $1.6 K\Omega$. La saturación en el transistor T_4 se satura brevemente debido a la corriente proporcionada por el condensador, aumentando la tensión en concordancia con la constante de tiempo RC. Así se tendría que la tensión en la salida sería:

$$5 - \text{Tensión } be \text{ en } T_4 - \text{Tensión en } D_1 = 3.6 V$$

$$5 - 0.7 - 0.7 V = 3.6 V$$

13.6 Compuertas con tres estados

Esta compuerta es construida utilizando diferentes tecnologías. Además, se presenta en las compuertas de tipo totémico que llevan la conexión alamburada de las salidas para formar un bus común.

Presenta los siguientes estados de salida:

- a) bajo nivel (0)
- b) alto nivel (1)
- c) estado flotante (Z)

En la figura 13.5 se muestran los símbolos de las compuertas.

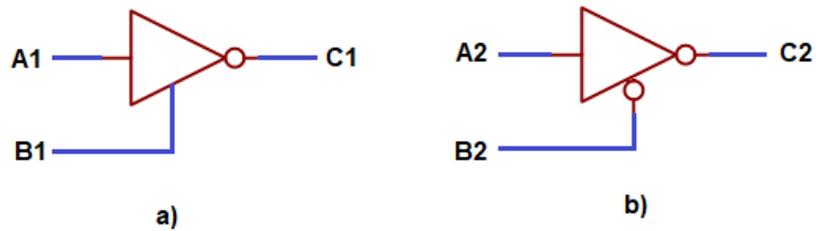


Figura 13.5. a) Compuerta triestado b) Compuerta triestado-inversora

Esta compuerta de tres estados opera comúnmente teniendo la entrada B_1 en el estado alto. Mientras que para el caso de la compuerta inversora que tiene tres estados se inicia cuando la entrada B_2 alcanza el estado bajo. Si la entrada C toma el estado bajo, la salida queda en circuito abierto, pero con alta impedancia, sea cual fuese el valor lógico que se establezca en la entrada A_1 (Ver figura 13.5 a). Además, en el estado Z no es factible que circule corrientes eléctricas. En la tabla 13.1 se presentan los estados de este tipo de compuertas para diferentes valores de entrada.

NO INVERSORA			INVERSORA		
A1	B1	C1	A2	B2	C2
0	0	Z	0	0	0
1	0	Z	1	0	1
0	1	0	0	1	Z
1	1	1	1	1	Z

Tabla 13.1 Tabla de verdad de las compuertas triestados

CAPÍTULO XIV

CIRCUITOS LÓGICOS MSI

14.1 Introducción

Se conoce como circuitos lógicos MSI a los circuitos integrados a media escala, los cuales lo utilizaremos en el presente capítulo para estudiar los circuitos digitales de Decodificación y codificación, multiplexaje y demultiplexaje.

14.2 Decodificadores

Un decodificador es un circuito lógico que tiene sus entradas y salidas. Si se establece un código en binario en las primeras entradas activará solo una de las salidas. Se puede implementar de forma discreta decodificadores de 2, 3, 4 bits, etc, en la entrada.

14.3 Decodificador de 2 bits:

En la figura 14.1 un decodificador de dos bits, con dos entradas de selección, y que al establecer un código en binario estas, solo habilitará una de las salidas, tal como se puede apreciar a continuación en la tabla de verdad de la figura 14.1:

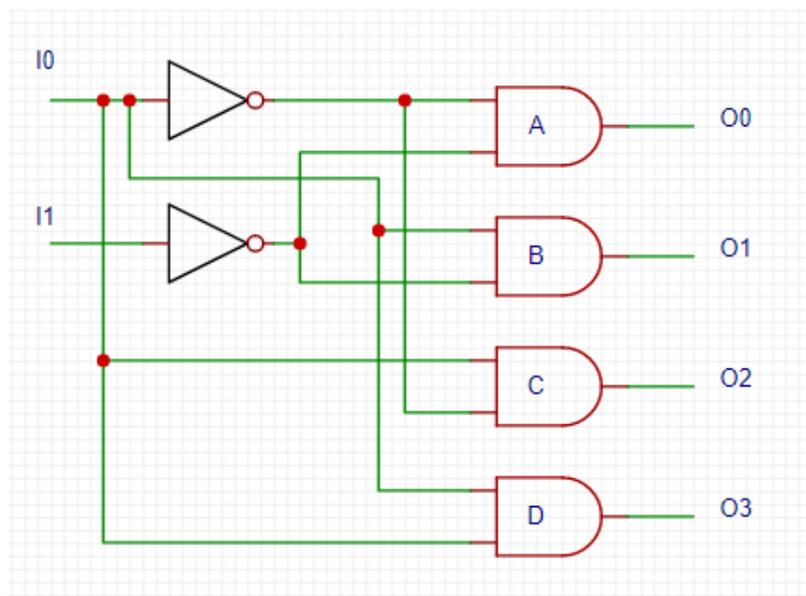


Figura14.1: Decodificador de dos bits

I1	I0	O0	O1	O2	O3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Tabla 14.1: Tabla de verdad del decodificador de dos bits

A este circuito digital se le denominará decodificador completo dado que solo se activará una salida para cada código en binario establecido en sus entradas de habilitación.

14.4 Decodificador de 3 bits:

Si el número de bits de en la entrada del decodificador es N , el número de salidas que le corresponderá a este decodificador será 2^N , por tanto, en el presente caso el número de salidas será $2^3 = 8$. Por tanto, a continuación, en la figura 14.2 se presenta el decodificador de tres entradas, es decir de tres bits, en el cual aparecerá sus 8 salidas.

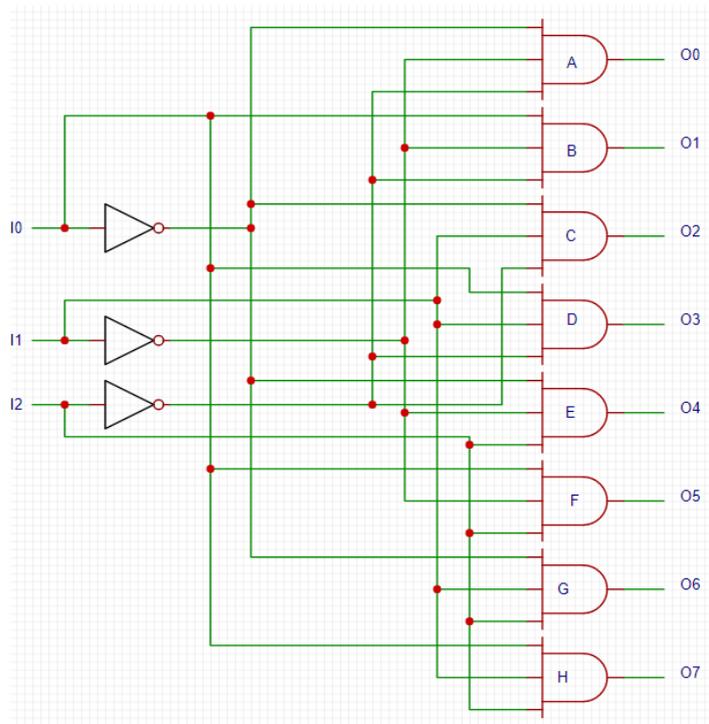


Figura14.2: Decodificador de tres bits

I2	I1	I0	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Tabla 14.2: Tabla de verdad del decodificador de tres bits

Aquí también observamos que en este decodificador se cumple que para código establecido en las entradas se activara una única salida. La tabla de verdad un decodificador de tres se observa en la Figura 14.2.

14.5 Codificadores

Lo codificadores funcionan al inverso del decodificador el cual tiene muchas entradas. Para cada entrada activada se producirá un código de N bits en las salidas. La figura 14.3 presenta el diagrama en bloque de un codificador

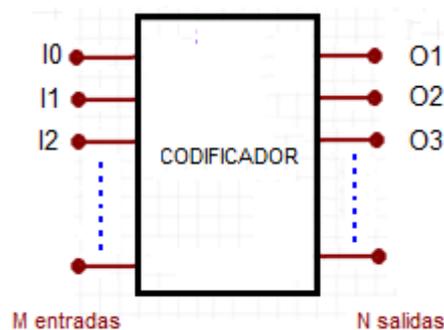


Figura14.3: Diagrama en bloque de un Codificador Tocci (2003)

DEMULTIPLEXORES

Un demultiplexor es un selector digital que conecta la única internamente la su única entrada con solo una de las sus salidas. Con que salida se realizara la conexión

dependerá del código que se plantee en sus entradas de selección. La figura 14-4 y la Tabla 14.53 muestran a un demultiplexor de 1 a 4 y a su tabla de verdad, respectivamente. Aquí se observa que existe una similitud entre un decodificador y un demultiplexor.

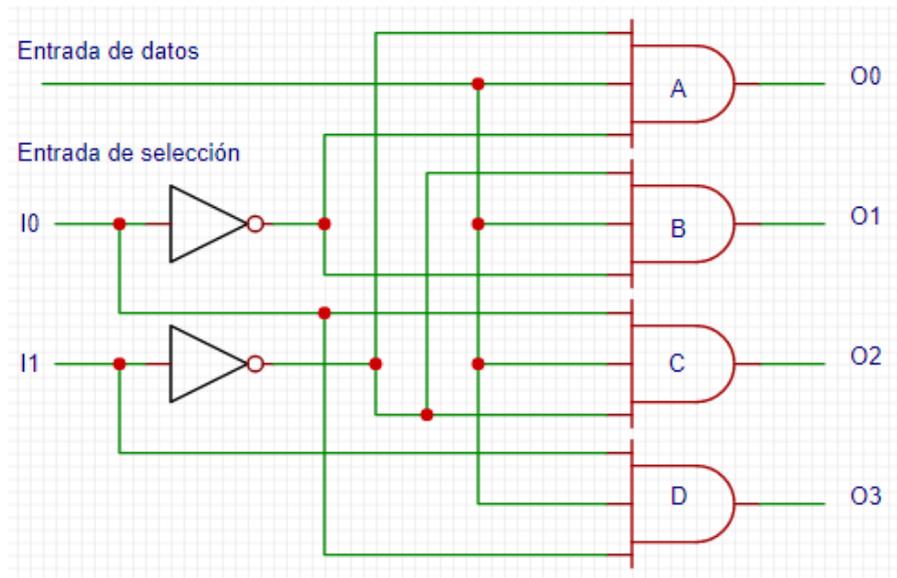


Figura 14.4: Circuito de un demultiplexor de uno a cuatro

Selección		Salidas			
I1	I0	O0	O1	O2	O3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Tabla 14.3: Tabla de verdad de un demultiplexor de una entrada y cuatro salidas.

14.6 Multiplexores

En este caso un multiplexor tendrá muchas entradas y solo una salida, además de sus entradas de selección, como se presenta en la figura 14.8.

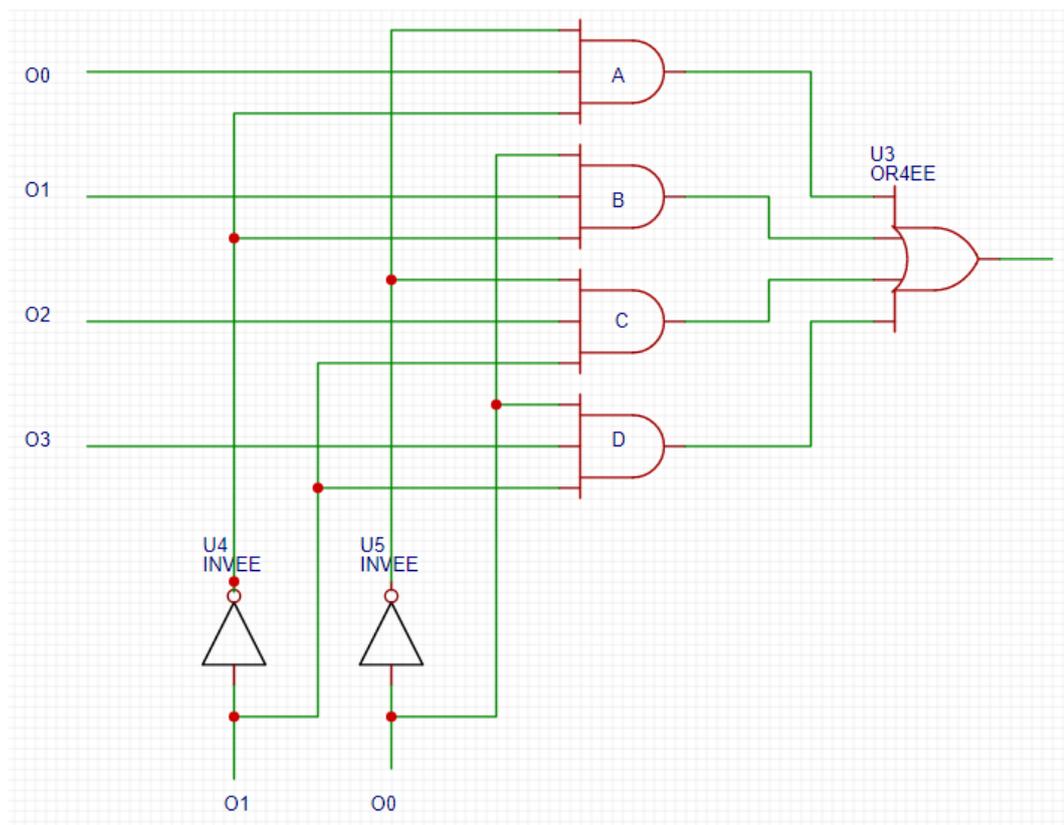


Figura 14.5: Multiplexor de cuatro a uno

En el diseño de este circuito digital se conecta una combinación de compuertas AND y una compuerta OR de varias entradas. Las salidas de las compuertas AND se conectan a una compuerta OR para producir una salida común. Nuevamente aquí el código en las entradas de selección una compuerta AND es la que determina que entrada sera conectada en la única salida.

14.7 Multiplexores y demultiplexores

En el mercado de dispositivos electrónicos se encuentra diferentes circuitos digitales multiplexores y demultiplexores encapsulado en circuitos integrados en la tecnología TTL, tal como cuyo circuito interno se presenta en la Figura 14.8. Por ejemplo, tenemos el circuito digital 74 138 el cual se puede utilizar como multiplexor o de multiplexor. Este CI también tres entradas de selección, además tiene tres entradas \bar{E}_1 , \bar{E}_2 y E_3 donde la primera sirve para el ingreso de datos y las dos siguientes para determinar la activación o no del circuito digital en su totalidad, según la figura 14.9.

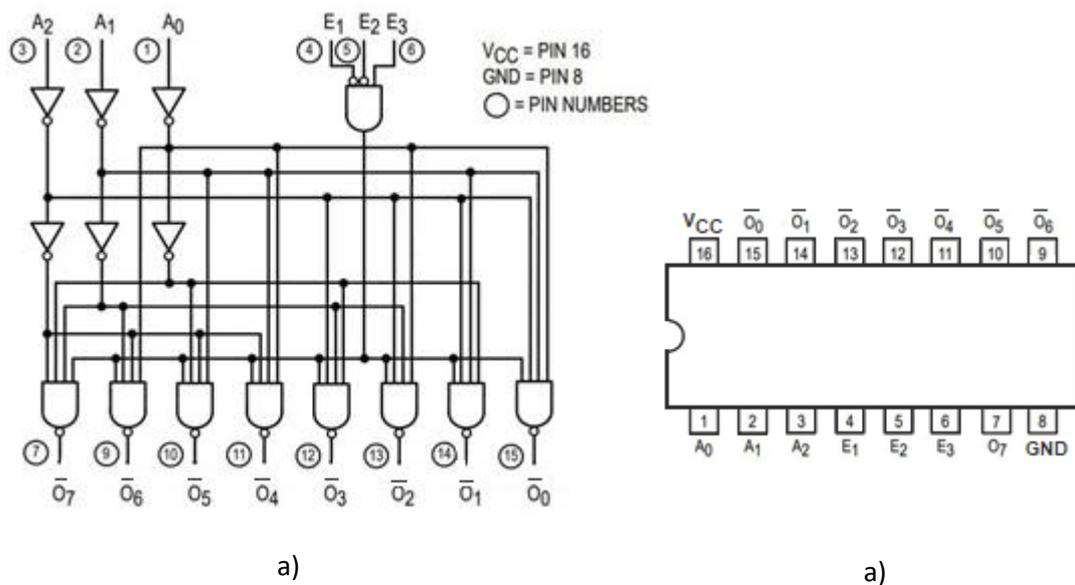


Figura 14.6: a) Circuito interno y b) diagrama en bloque del CI 74LS138

E_1	E_2	E_3	Respuesta
0	0	1	Responde al código en A_2 , A_1 y A_0
1	X	X	Deshabilitada – Todas alto
X	1	X	Deshabilitada – Todas alto
X	X	0	Deshabilitada – Todas alto

Tabla 14.4: Tabla de verdad del CI 74LS Tocci (2003)

CAPÍTULO XV

INTERFAZ CON EL MUNDO ANALÓGICO

15.1 Introducción

Una compuerta digital solo puede asumir dos valores, como 0 lógico (BAJO) o 1 lógico (ALTO). En la práctica estas cantidades, dentro de la lógica TTL, toman cualquier valor dentro de los intervalos que se indican:

$$0V \text{ a } 0.8 V = 0 \text{ lógico}$$

$$2V \text{ a } 5V = 1 \text{ lógico}$$

Para Tocci (2003) las variables físicas son analógicas y pueden asumir algún valor dentro de un rango continuo. Sin embargo, las funciones que realiza un sistema digital las hace utilizando circuitos digitales y además realizando operaciones también digitales. Por ello, si se desea procesar una señal analógica en un sistema digital esta primero tendrá que transformarse en una señal digital.

En la figura 15.1, se presentan los diferentes bloques electrónicos que participan desde la detección de una magnitud física hasta con fines de controlarla. La parte central es el sistema digital de procesamiento que recibe una señal digital obtenida y convertida por el transductor y el ADC. Este sistema digital puede ser una PC.

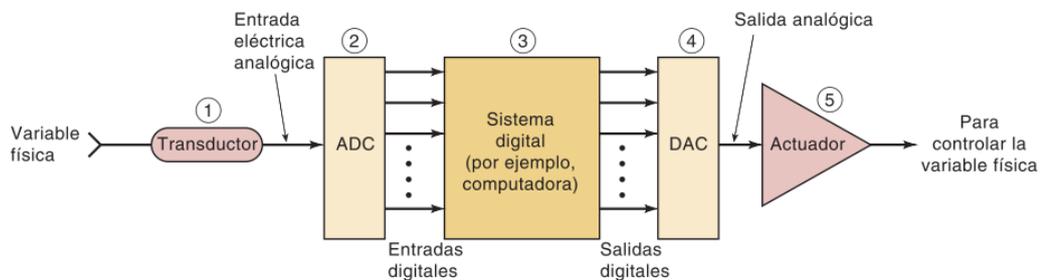


Figura 15.1 Sistema digital y su entorno Tocci (2003)

El bloque N°1 está conformado por el transductor que convierte la variable física a procesar, en una señal de corriente o voltaje eléctrico. Con el avance tecnológico existen diferentes sensores para obtener la señal analógica correspondiente a magnitudes física tales como proximidad, humedad, intensidad de sonido, temperatura, Intensidad de campos magnéticos, Intensidad de radiación, temperatura, entre otros.

El bloque N°2 se encarga de transformar la salida del transductor en una señal digital. Por ejemplo, si la salida del transductor varía entre 800 y 1500 mV, estos valores extremos de señal serán convertidos a los valores digitales 01010000 y 10010110 respectivamente. Para este caso particular, el ADC convierte una variación de 10 mV en una variación de un dígito binario.

El bloque N°3 es el sistema digital el cual tiene como función guardar los datos convertidos para luego procesarlos de acuerdo con el algoritmo de control establecido mediante un programa de computación (conjunto de instrucciones) en ejecución. El programa efectuará cálculos u otras operaciones sobre las “variables binarias”, que representan a las variables analógicas del proceso físico real. Los resultados generan una salida digital que eventualmente servirá para controlar la variable física.

El bloque N°4 muestra el convertidor **digital /analógico**. En él, la salida digital se convierte en analógica para controlar con esta señal, el dispositivo actuador. Por ejemplo, una señal digital que varía desde 00000000 (Hex 00) a 11111111 (Hex FF), es convertida en una señal analógica variable entre 0 Volt y 10 Volt.

El bloque N°5 es el llamado actuador (válvula reguladora, motor eléctrico, servo de posición, etc.), el cual sirve para controlar la variable física.

Los tipos de convertidores que veremos son:

1. **Convertidor analógico- digital (ADC)**. Este circuito recibe en su entrada, generalmente proveniente de un transductor, una señal analógica (eléctrica),

que los convierte en una señal digital. Esta señal se representa a través de varios bits la cual representa el valor de la entrada analógica. Siempre se tendrá que la salida binaria del ADC será proporcional al valor del voltaje analógica de entrada del convertidor.

- 2. Convertidor digital - analógico (DAC).** Este circuito recibe en su entrada una señal digital para convertirla en una señal analógica (generalmente un voltaje o una corriente proporcional).

15.2 Conversión Digital - Analógica

Como muchos de los convertidores analógicos/digital tienen un módulo DAC, resulta entonces conveniente analizar primero, los convertidores digitales/analógico.

La conversión D/A consiste en utilizar un valor representado en digital (como binario o BCD) y convertirlo en un voltaje o corriente que es proporcional al valor de la señal de entrada. La figura 7.2 nos presenta el diagrama en bloque de un convertidor D/A de cuatro bits.

Las entradas D,C,B,A son entradas digitales. Es posible representar $2^4 : 16$ diferentes numeros binarios en esta 4 bits. Para cada numero binario de entrada le correspondera un numero distinto en la salida. El voltaje de salida analogica V_{sal} en voltios se obtiene la señal digital de entrada por un coeficiente de proporcionalidad. El mismo procedimiento se puede utilizar si la salida del D/A fuese una corriente de salida I_{sal} y no una tension.

Entonces tendremos lo siguiente:

$$\text{Voltaje de salida analógica} = K \times \text{Señal digital de entrada}$$

Donde K: es coeficiente característico del Convertidor Digital-Analógico.

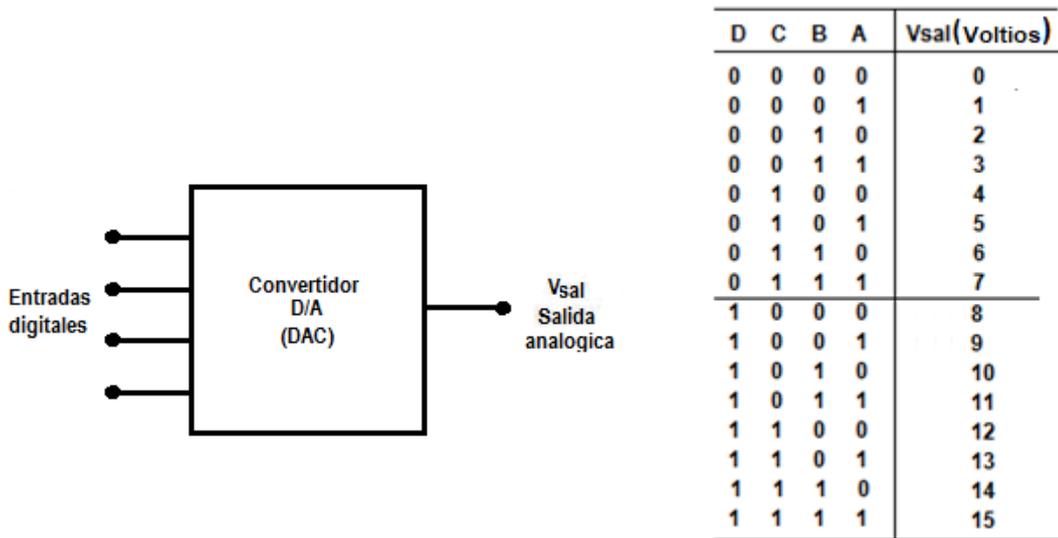


Figura 15.2: Convertidor de 4 bits con salida de voltaje

15.2.1 Factor de ponderación (FP)

El factor de ponderación de cada bit que se presenta en la entrada de un DAC, es el valor que incrementa su salida. Para el caso del ejemplo tenemos:

- A0 → FP= 1 volt (LSB)
- A1 → FP= 2 volt
- A2 → FP= 4 volt
- A3 → FP= 8 volt (MSB)

15.2.2 Resolución del DAC

Se define como el cambio incremental más pequeño de tensión o corriente que se produce en la salida como resultado de un cambio en la entrada digital. Para el caso de nuestro ejemplo la resolución es de 1 volt. La resolución, también denomina “tamaño del escalón”, es siempre igual al “factor de ponderación del bit menos significativo (LSB), que en nuestro caso del ejemplo, corresponde a:

$$A0 \rightarrow FP = \text{resolución} = K = 1 \text{ volt.}$$

La resolución también la podemos obtener mediante:

$$\text{Resolución} = \text{valor de fondo de escala de un DAC} / (2N - 1)$$

Para nuestro caso $N = 4$, tensión de fondo de escala = 15 volt luego:

$$\text{Resolución} = 15 / (2^4 - 1) = 1 \text{ volt}$$

15.2.3 Convertidores DAC con entradas en código BCB

Las entradas analizadas anteriormente estaban en código binario natural. Hay DAC que tienen las entradas en código BCD (decimal codificado en binario). En este código, en realidad las entradas son decimales que están codificados en binario natural. Por ejemplo, si necesitamos representar números decimales desde el 00 al 99, necesitaremos Cuatro bits para las unidades y cuatro bits para las decenas, o sea un total de ocho bits. Recordemos que en el código BCD se utilizan 4 bits para los decimales del 0 al 9, utilizando las 10 primeras combinaciones del código binario natural.

Ejemplo: representar el número decimal 57 en BCD

Decimal	5	7
	↓	↓
BCD	0101	0111

15.3 Aplicaciones de los DAC's

Los DAC se utilizan siempre que la salida de un circuito digital tiene que ofrecer un voltaje o corriente analógico para impulsar o activar un dispositivo analógico.

Algunas de sus aplicaciones más comunes se describen a continuación:

- **Control:** la salida digital de una computadora puede convertirse en una señal de control analógica para ajustar la velocidad de un motor, la temperatura de un horno o bien para controlar casi cualquier variable física.

- **Análisis automático:** En las PCs a través de la programación se puede crear señales analógicas para la aplicación de circuitos analógicos externos para que sean procesado, para lo cual utiliza un DAC. Una vez obtenido la señal analógica resultante, esta nuevamente puede ser convertida a digital esta vez usando el ADC

- **Control de amplitud digital:** Un convertidor de Digital A analógico se puede usar para controlar la amplitud de una señal. Este el método que se utiliza para controlar el volumen digital de un sistema de audio.

- **Convertidores Analógicos-Digitales (ADC):** Muchos tipos de convertidores analógicos-Digitales están contruidos utilizan los Convertidores Digital-Analógicos (DAC)

15.4 Conversión Analógica - Digital

La Función principal de un ADC es la de convertir una señal analógica a digital, contrario a lo realiza un DAC. Esta tare requiere considerar la velocidad con la que convierte los datos, la resolución de esta conversión, los rangos de la señal a convertir, entre otros.

15.4.1 Conversión básica de señales analógicas

En la Figura 15.3 se presenta un diagrama en bloques de convertidor analógico-digital. Su funcionamiento consiste captar a través de sensor una señal analógica como puede ser la temperatura, la presión, la velocidad, una corriente eléctrica, una tensión, el campo magnético, entre otras señales. Esta captación la hace a través un muestreo de la señal que recibe para luego convertirla en una señal digital que será ingresada a una unidad de procesamiento digital, como puede ser una PC.



Para el muestreo de la señal analógica se debe considerar el Teorema de Nyquist que establece que la frecuencia de muestreo debe ser al menos de dos muestras de señal para un periodo. Si se tuviese menos muestras por periodo se presentará el fenómeno de solapamiento (Aliasing)

15.4.2 CARACTERÍSTICAS

Características Estáticas

- a) Resolución:
- b) La linealidad integral y el de linealidad diferencial:
- c) Monotonicidad:
- d) El error de ganancia, el error de Desplazamiento (offset) y el error de cuantificación:
- e) Velocidad:

Características Dinámicas

- a) Tiempos de conversión:
- b) Tiempo de adquisición:
- c) Tiempo de asentamiento:
- d) Slewrate:

CAPÍTULO XVI

DISPOSITIVOS DE MEMORIA

16.1 Introducción

Las memorias son dispositivos que sirven para almacenar información binaria la que después se puede utilizar posteriormente para procesarla. Esta información se ubica en registros de la unidad de memoria. Este procedimiento se puede dar durante el proceso para instantes después sean nuevamente utilizados o al final para que sea almacenada por más tiempo. Una unidad de memoria físicamente es un conjunto de celdas de memoria donde se puede almacenar grandes cantidades de datos binarios.

Se puede distinguir dos tipos de memoria. La primera de ellas es la memoria de acceso aleatorio conocida como RAM (Random acces memory), y la segunda conocida como memoria ROM (read only memory). La primera permite realizar el proceso de escritura dentro de ella o como también de lectura de información que ha sido almacenada en esta memoria. La segunda memoria es de solo lectura ya que en ella se almacena información que se quiere leer las veces que desee.

Las memorias de solo lectura poseen información que ha sido almacenada a través de programación lógica. Por esta razón, una memoria ROM es un dispositivo lógico programable PLD (Programmable logic device). El cual está formado por cientos y hasta millones de puertas lógicas que se encuentran integrados a través de conectores que se pueden fundir por el programador hasta lograr el tipo de configuración del circuito digital deseado

La comunicación entre la unidad memoria y su entorno se efectúa a través buses de entrada y salida de datos, buses de selección de direcciones y buses que palabra se transfiere. En la figura 16.1 se presenta una memoria en bloque. En esta se presenta las n líneas de entrada de datos que serán almacenados en la memoria y n líneas de salida de datos que nos provee la memoria. También observamos k líneas de dirección que determina que palabra será proporcionada por la memoria

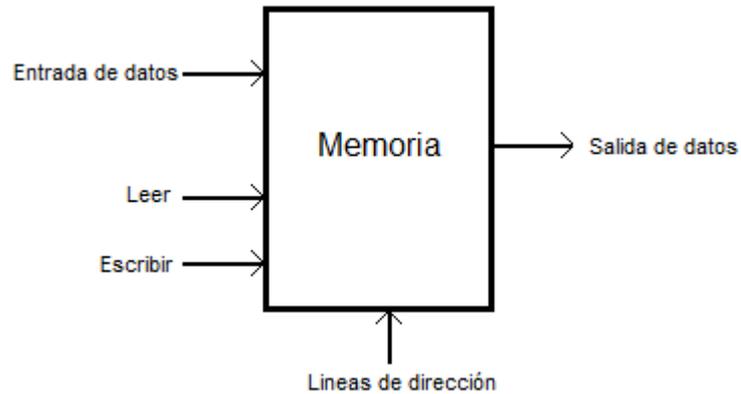


Figura 16.1 Diagrama de bloques de una unidad de memoria

Una unidad se caracteriza por el número de palabras que puede almacenar y el tamaño de cada palabra expresado en bits. A cada palabra de la memoria se le puede identificar con la clave entre 0 y $2^k - 1$ (k es el número de líneas de dirección). Cada palabra almacenada en la memoria puede seleccionarse utilizando los k bits de dirección. Ver figura.

Binario	Decimal
000000000	0
000000001	1
000000010	2
⋮	⋮
111111101	1021
111111110	1022
111111111	1023

Figura 16.2 Dirección de memoria

16.2 Memoria ROM

Esta memoria tiene información que solamente puede leída ya que ha sido grabada por el fabricante. Las características de esta memoria son:

Permanencia. - La información guardada no se borra, ya sea que una PC este encendida o apagada. Tiene el comportamiento como la de un disco duro cuya información almacenada no es volátil.

Seguridad. - Esta característica es importante ya que la información que ha sido almacenada en una memoria ROM no es posible de borrar o modificar por causas provocadas o imprevistas. Los usos de esta memoria son:

Existen varios tipos de ROM como se puede apreciar en la siguiente figura 16.2:

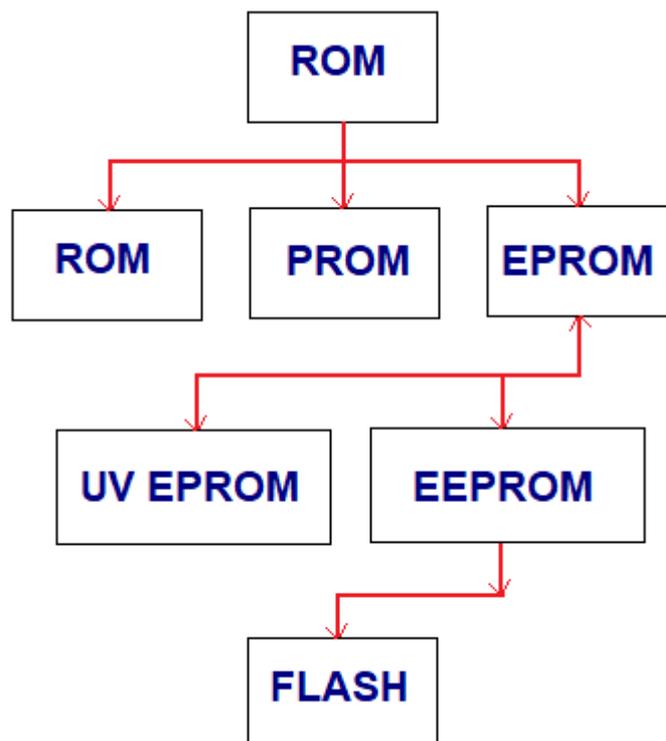


Figura 16.3 Diagrama de bloques de una unidad de memoria

16.3 Memoria PROM

Este tipo es una memoria tipo ROM, de solo lectura programable (Programmable Read Only Memory) es programada por un dispositivo especial. Este procedimiento se realiza a través de la aplicación de pulsos eléctricos de altos voltajes que quemas fusibles de conexión dentro de las componentes internas de esta memoria. Esta memoria solo permite realizar una sola programación.

16.4 Memoria EPROM

Esta memoria EPROM (Electrically Erasable Programmable Read Only Memory) permite ser leída y ser grabada y borrada utilizando una técnica de aplicación de un dispositivo electrónico que nos proporciona voltajes superiores a los usados en circuitos electrónicos. Para el borrado y grabado una EPROM viene provista de un recubrimiento de cristal que se convierte en una ventana a través de la cual se puede lograr este procedimiento.

16.5 Memoria EEPROM

Es una memoria ROM de solo lectura programable y borrrable eléctricamente (Electrically Erasable Programmable Read Only Memory) la cuales se pueden alterar utilizando un software. Esta se utiliza en los computadores para mantener la

Una memoria RAM tiene conectores para la entrada y salida de información, así como entradas para la selección de direcciones y conectores

16.6 Características generales

Las memorias RAM requiere de una serie de señales que sirven para gobernar su modo de funcionamiento:

16.6.1 Bus de datos: es un bus de entrada de memoria a través de los cuales **ingresan datos** para guardarlos después de realizar la escritura y por **salen los datos** después del proceso de lectura. La capacidad de transporte del bus depende del número de bits que tiene la palabra.

16.6.2 Bus de direcciones: es un bus de entrada que sirve para seleccionar la palabra que será procesada a través de la lectura o escritura. La capacidad de este bus dependerá del número de palabras que existan en la memoria.

16.6.3 Bus de control: Este bus es el que controla el uso y el acceso a los buses de datos y direcciones.

16.7 Memoria RAM

Es una unidad de memoria RAM (Random Acces Memory) donde se puede almacenar datos binarios en grupos de bits que se conocen como *palabras*. En una *palabra*, compuesta por ceros y unos se pueden guardar o retirar como un grupo. Debemos recordar que un byte tiene ocho bits. Cada palabra esta compuesta por un múltiplo de ocho bits. De este modo una palabra puede contener dieciséis bits (dos bytes). En una PC almacena datos que se están utilizando durante la ejecución de los programas. Es una memoria volátil que al desconectarse la PC se perderá toda la información guardada. Hay dos tipos de memorias RAM tal como se puede observar en la figura 16.4.

16.8 Memoria SRAM: Es una memoria estática (Static RAM) que se compone por flip-flops, por lo que su almacenamiento de datos depende de que estas estén alimentadas eléctricamente. Tiene poca capacidad de almacenamiento, pero permite una elevada velocidad de lectura. Se clasifican en dos tipos, las que son asíncronas y las síncronas de ráfaga.

16.9 Memoria DRAM Es una memoria dinámica (Synamic RAM) que se compone por condensadores que permite un almacenamiento de datos durante mucho tiempo. La desventaja de esta memoria que tiene baja velocidad de lectura. Esta memoria se divide en: FPM DRAM, EDO DRAM, BEDO DRAM, SDRAM Y DDR SRAM. Estas últimas (Double Data Rate SDRAM) son las que en la actualidad se usan en las PCs. Esta memoria se caracteriza porque envía dos datos por ciclo de la señal de reloj, y no por su frecuencia de esta señal. Una aplicación de estas memorias es en la fabricación de memorias cache.

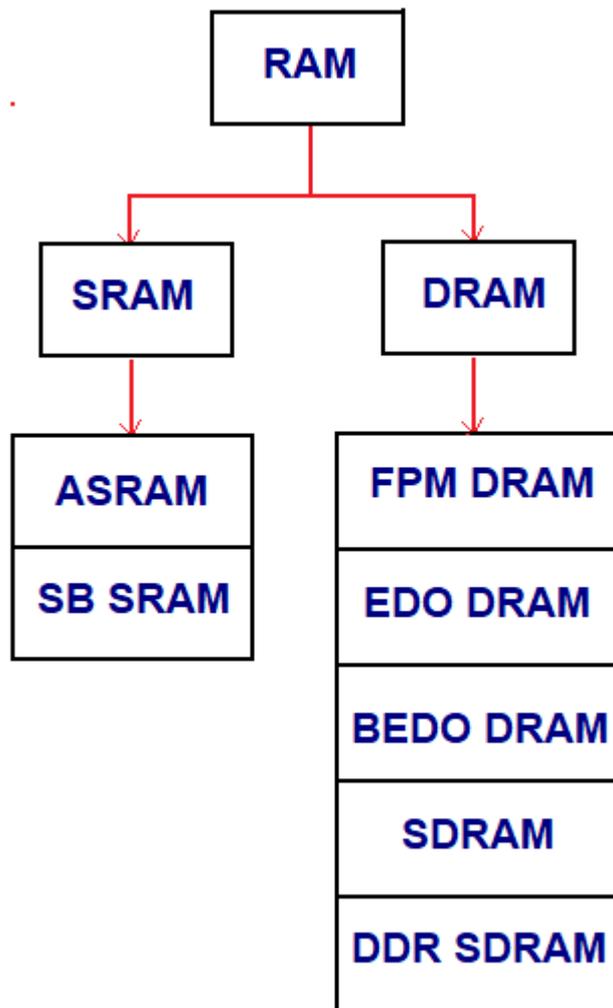


Figura 16.4 Tipos de memorias RAM

V. REFERENCIAS BIBLIOGRÁFICAS

1. Floyd T. (2006). *Fundamentos de sistemas digitales*. (9na ed.). Madrid: Pearson.
2. Garza J. (2006). *Sistemas y electrónica digitales*. Madrid: Pearson.
3. Huircan, J. (2013) *Conversores análogo-digital y digital-análogo. Conceptos básicos*. Recuperado de: <http://repository.unad.edu.co/bitstream/10596/5272/1/ad03.pdf>
4. Nakamura, S. (1992) *Métodos numéricos aplicados con software*. México: Prentice-Hall Hispanoamérica.
5. Pérez García, M. (2014). *Instrumentación Electrónica*. Madrid: Editorial Paraninfo.
6. Press, W. H. S.A., Teukolsky, W., Vetterling, T. y Flannery, B.P. (1992). *Numerical Recipes in Fortran. The Art of Scientific Computing*. (2da ed.) New York: Cambridge.
7. Santiago, E. Acha, S., Rioseras, M., y Lozano, M. (2006). *Electrónica Digital. Lógica Digital Integrada. Teoría, Problemas y Simulación*. et al. (2006). *Electrónica digital, lógica digital integrada: teoría, problemas y simulación*. Barcelona: Editorial Microinformática.
8. Tocci Ronal J. (2013). *Sistemas digitales. Principios y aplicaciones*. (6ta ed.). Prentice-Hall Hispanoamericana.
9. López, J. (2020). *Modulo 5. Técnicas digitales. Sistemas de instrumentos electrónicos*. España: Ediciones Paraninfo.
10. Martín, J., Arias, J., Bidarte, U., Ibáñez, P., Lázaro, J. y Zuloaga, A. (2006). *Electrónica Digital*. España: Delta Publicaciones.
11. Morris Mano, M. (2003). *Diseño digital*. México: Pearson educación.

VI. APÉNDICE

APENDICE 1: Programa computacional diseñado en Fortran para determinar todas las operaciones de las compuertas básicas.

```
PROGRAM COMPUERTAS
INTEGER A,B,X
CHARACTER(len=5):: OPER !,OPER1
WRITE(*,*)'*****'
WRITE(*,*)'DADO LOS VALORES DE LAS VARIABLES LOGICAS A Y B'
WRITE(*,*)'ESTE PROGRAMA CALCULA LAS OPERACIONES SIGUIENTES'
WRITE(*,*)'*****'
WRITE(*,*)'INGRESE EL TIPO DE COMPUERTA'
WRITE(*,*)'AND'
WRITE(*,*)'OR'
WRITE(*,*)'NAND'
WRITE(*,*)'NOR'
WRITE(*,*)'INV'
WRITE(*,*)'=====
WRITE(*,*)
WRITE(*,*)'ELIJA LA OPERACION LOGICA: AND,OR,NAND,NOR o INV'
READ(*,*)oper

INV=INV
operaciones: SELECT CASE (oper)
  CASE ('AND')
    !COMPUERTE AND
    WRITE(*,*)'INGRESE LAS VARIABLES LOGICAS A Y B'
    READ(*,*)A,B
    IF (A==0.AND.B==0)THEN
      X=0
      WRITE(*,*)X
    ELSE IF (A==0.AND.B==1)THEN
      X=0
      WRITE(*,*)X
    ELSE IF (A==1.AND.B==0) THEN
      X=0
      WRITE(*,*)X
    ELSE
      X=1
    WRITE(*,*(I2)')X
    END IF
  CASE ('OR')
    !COMPUERTE OR
    WRITE(*,*)'INGRESE LAS VARIABLES LOGICAS A Y B'
    READ(*,*)A,B
    IF (A==0.AND.B==0)THEN
```

```

        X=0
        WRITE(*,*)X
    ELSE IF (A==0.AND.B==1)THEN
        X=1
        WRITE(*,*)X
        ELSE IF (A==1.AND.B==0) THEN
            X=1
            WRITE(*,*)X
    ELSE
        X=1
        WRITE(*,(I2'))X
    END IF
CASE ('NAND')
!COMPUERTE NAND
WRITE(*,*)'INGRESE LAS VARIABLES LOGICAS A Y B'
READ(*,*)A,B
IF (A==0.AND.B==1)THEN
    X=0
    WRITE(*,*)X
ELSE IF (A==0.AND.B==1)THEN
    X=0
    WRITE(*,*)X

ELSE IF (A==1.AND.B==1) THEN
    X=0
    WRITE(*,*)X
ELSE
    X=0
    WRITE(*,(I2'))X
END IF
CASE ('NOR')
!COMPUERTE NOR
WRITE(*,*)'INGRESE LAS VARIABLES LOGICAS A Y B'
READ(*,*)A,B
IF (A==0.AND.B==0)THEN
    X=1
    WRITE(*,*)X
ELSE IF (A==0.AND.B==1)THEN
    X=0
    WRITE(*,*)X

ELSE IF (A==1.AND.B==0) THEN
    X=0
    WRITE(*,*)X
ELSE
    X=0
    WRITE(*,(I2'))X

```

```

        END IF
CASE ('INV')
    !INVERSOR
    WRITE(*,*)'INGRESE LAS VARIABLES LOGICAS A'
    READ(*,*)A
    IF(A==0)THEN
        X=1
        WRITE(*,*)X
    ELSE
        X=0
        WRITE(*,*)X
    END IF
CASE DEFAULT
    WRITE (*,*) 'ESTA OPERACION LOGICA NO ESTA CONTEMPLADA EN EL
PROGRAMA'
END SELECT operaciones
END

```

APENDICE 2: Programa computacional diseñado en fortran para determinar las tablas de verdad de los Flips flops.

```

PROGRAM FLIP_FLOPS
CHARACTER(len=5):: OPER ,OPER1
WRITE(*,*)'*****'
WRITE(*,*)'DADO LOS VALORES DE LAS VARIABLES LOGICAS A Y/O B'
WRITE(*,*)'ESTE PROGRAMA CALCULA LAS OPERACIONES DE LOS FLIP FLOPS'
WRITE(*,*)'*****'

```

```

WRITE(*,*)'INGRESE EL TIPO DE FLIP FLOP'
WRITE(*,*)'JK'
WRITE(*,*)'RS'
WRITE(*,*)'=====
WRITE(*,*)
WRITE(*,*)'ELIJA EL FLIP FLOP: JK O RS'
READ(*,*)oper

```

INV=INV

operaciones: SELECT CASE (oper)

CASE ('JK')

```

!COMPUERTE AND
WRITE(*,*)'INGRESE LAS VALORES LOGICOS PARA J Y K'
READ(*,*)J,K
IF (J==0.AND.K==0)THEN
    Q=0
    WRITE(*,*)'LA SALIDA Q NO PRESENTA CAMBIO'
ELSE IF (J==0.AND.K==1)THEN
    Q=0
    WRITE(*,*)'EL VALOR LOGICO EN Q'
    WRITE(*,*)Q
ELSE IF (j==1.AND.k==0) THEN
    Q=1
    WRITE(*,*)'EL VALOR LOGICO EN Q'
    WRITE(*,*)Q

```

```

WRITE(*,*)Q
ELSE
    WRITE(*,*)'LA SALIDA SE COMPLEMENTA'
END IF

```

CASE ('RS')

```

WRITE(*,*)'EL FLIP FLOP ES ASINCRONO'
WRITE(*,*)'INDIQUE SI USA COMPUERTAS NOR O NAND'
READ(*,*)OPER1

```

IF(OPER=='NOR')THEN

```

WRITE(*,*)'INGRESE LAS VARIABLES LOGICAS S Y R'
READ(*,*)S,R
IF (S==0.AND.R==0)THEN
    WRITE(*,*)'LA SALIDA Q NO PRESENTA CAMBIO'
ELSE IF (S==0.AND.R==1)THEN
    Q=0
    WRITE(*,*)Q
ELSE IF (S==1.AND.R==0) THEN
    Q=1

```

```

        WRITE(*,*)Q
    ELSE
        WRITE(*,*)'LA SALIDA INDESEABLE'
    END IF
ELSE
    WRITE(*,*)'INGRESE LAS VARIABLES LOGICAS S Y R'
    READ(*,*)S,R
    IF (S==0.AND.R==0)THEN
        WRITE(*,*)'LA SALIDA INDESEABLE'
    ELSE IF (S==0.AND.R==1)THEN
        Q=1
        WRITE(*,*)Q
    ELSE IF (S==1.AND.R==0) THEN
        Q=0
        WRITE(*,*)Q
    ELSE
        WRITE(*,*)'LA SALIDA Q NO PRESENTA CAMBIO'
    END IF
END IF
CASE DEFAULT
    WRITE (*,*) 'EL FLIP FLOP NO ESTA CONTEMPLADO EN EL PROGRAMA'
END SELECT operaciones
END

```

APENDICE 3: Programa computacional diseñado en fortran para determinar los valores digitales en la salidas de decodificador de 1 a 8 dada los valores digitales en la entrada de este circuito.

```

PROGRAM DECODIFICADOR
INTEGER::C,B,A,Q7, Q6, Q5, Q4, Q3, Q2, Q1, Q0
WRITE(*,*)'*****'
WRITE(*,*)'ESTE PROGRAMA DETERMINA LOS VALORES DE SALIDA'
WRITE(*,*)'DE UN DECODIFICADOR DE TRES A 8 LINEAS (o 1 DE )'
WRITE(*,*)'*****'
WRITE(*,*)'INGRESE LAS ENTRADAS A,B Y C DEL DECOFICADOR'
WRITE(*,*)'=====
WRITE(*,*)'INGRESE EL VALOR DE A'
READ(*,*)A
WRITE(*,*)'INGRESE EL VALOR DE B'

```

```

READ(*,*)B
WRITE(*,*)'INGRESE EL VALOR DE C'
READ(*,*)C
WRITE(*,*)'-----'
WRITE(*,*)'LAS SALIDAS DEL DECODIFICADOR SON:'
WRITE(*,*)
IF(C==0.AND.B==0.AND.C==0)THEN
    Q7=0;Q6=0;Q5=0;Q4=0;Q3=0;Q2=0;Q1=0;Q0=1
    WRITE(*,*)
    WRITE(*,*)'Q7=',Q7
    WRITE(*,*)'Q6=',Q6
    WRITE(*,*)'Q5=',Q5
    WRITE(*,*)'Q4=',Q4
    WRITE(*,*)'Q3=',Q3
    WRITE(*,*)'Q2=',Q2
    WRITE(*,*)'Q1=',Q1
    WRITE(*,*)'Q0=',Q0
ELSE IF (C==0.AND.B==0.AND.A==1)THEN
    Q7=0;Q6=0;Q5=0;Q4=0;Q3=0;Q2=0;Q1=1;Q0=0
    WRITE(*,*)
    WRITE(*,*)'Q7=',Q7
    WRITE(*,*)'Q6=',Q6
    WRITE(*,*)'Q5=',Q5
    WRITE(*,*)'Q4=',Q4
    WRITE(*,*)'Q3=',Q3
    WRITE(*,*)'Q2=',Q2
    WRITE(*,*)'Q1=',Q1
    WRITE(*,*)'Q0=',Q0

ELSE IF (C==0.AND.B==1.AND.A==0)THEN
    Q7=0;Q6=0;Q5=0;Q4=0;Q3=0;Q2=1;Q1=0;Q0=0
    WRITE(*,*)
    WRITE(*,*)'Q7=',Q7
    WRITE(*,*)'Q6=',Q6
    WRITE(*,*)'Q5=',Q5
    WRITE(*,*)'Q4=',Q4
    WRITE(*,*)'Q3=',Q3
    WRITE(*,*)'Q2=',Q2
    WRITE(*,*)'Q1=',Q1
    WRITE(*,*)'Q0=',Q0
ELSE IF (C==0.AND.B==1.AND.A==1)THEN
    Q7=0;Q6=0;Q5=0;Q4=0;Q3=1;Q2=0;Q1=0;Q0=0
    WRITE(*,*)
    WRITE(*,*)'Q7=',Q7
    WRITE(*,*)'Q6=',Q6
    WRITE(*,*)'Q5=',Q5
    WRITE(*,*)'Q4=',Q4

```

```

WRITE(*,*)'Q3=',Q3
WRITE(*,*)'Q2=',Q2
WRITE(*,*)'Q1=',Q1
WRITE(*,*)'Q0=',Q0
ELSE IF (C==1.AND.B==0.AND.A==0)THEN
Q7=0;Q6=0;Q5=0;Q4=1;Q3=0;Q2=0;Q1=0;Q0=0
WRITE(*,*)
WRITE(*,*)'Q7=',Q7
WRITE(*,*)'Q6=',Q6
WRITE(*,*)'Q5=',Q5
WRITE(*,*)'Q4=',Q4
WRITE(*,*)'Q3=',Q3
WRITE(*,*)'Q2=',Q2
WRITE(*,*)'Q1=',Q1
WRITE(*,*)'Q0=',Q0
ELSE IF (C==1.AND.B==0.AND.A==1) THEN
Q7=0;Q6=0;Q5=1;Q4=0;Q3=0;Q2=0;Q1=0;Q0=0
WRITE(*,*)
WRITE(*,*)'Q7=',Q7
WRITE(*,*)'Q6=',Q6
WRITE(*,*)'Q5=',Q5
WRITE(*,*)'Q4=',Q4
WRITE(*,*)'Q3=',Q3
WRITE(*,*)'Q2=',Q2
WRITE(*,*)'Q1=',Q1
WRITE(*,*)'Q0=',Q0
ELSE IF (1==0.AND.1==0.AND.A==0)THEN
Q7=0;Q6=1;Q5=0;Q4=0;Q3=0;Q2=0;Q1=0;Q0=0
WRITE(*,*)
WRITE(*,*)'Q7=',Q7
WRITE(*,*)'Q6=',Q6
WRITE(*,*)'Q5=',Q5
WRITE(*,*)'Q4=',Q4
WRITE(*,*)'Q3=',Q3
WRITE(*,*)'Q2=',Q2
WRITE(*,*)'Q1=',Q1
WRITE(*,*)'Q0=',Q0
ELSE
Q7=1;Q6=0;Q5=0;Q4=0;Q3=0;Q2=0;Q1=0;Q0=0
WRITE(*,*)
WRITE(*,*)'Q7=',Q7
WRITE(*,*)'Q6=',Q6
WRITE(*,*)'Q5=',Q5
WRITE(*,*)'Q4=',Q4
WRITE(*,*)'Q3=',Q3
WRITE(*,*)'Q2=',Q2
WRITE(*,*)'Q1=',Q1

```

```
        WRITE(*,*)'Q0=',Q0
END IF
WRITE(*,*)'-----'
END
```