

73
24.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS "ARAGÓN"**



**"ESTUDIO Y APLICACIONES
DEL
DISEÑO DE SISTEMAS DIGITALES"**

TESIS PROFESIONAL

**QUE PARA OBTENER EL TITULO DE :
INGENIERO MECÁNICO ELECTRICISTA**

P R E S E N T A

MARÍA ROSALBA PATRICIA MORÁN CRUZ

San Juan de Aragón, Edo. Méx.

México, 1997.

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

La grandeza del hombre se puede medir por lo que deja plantado para que se desarrolle, y si puso o no a pensar a otros en nuevas direcciones con un vigor que persistiera después de él.

H.g.Wells.

AGRADECIMIENTOS.

A DIOS:

Por su gran amor, apoyo, ayuda y por permitirle
me la gran felicidad de ser siempre yo misma.

A MI HIJITA QUERIDA: Alin Aidé Acevedo Morán.

Por que es la luz, esperanza y energía que me
impulsan a superarme.

A MI QUERIDO ESPOSO: Luis Acevedo P.

Por su presencia, recuerdos y cariño.

MUY ESPECIALMENTE:**A MIS PADRES: Gpe. Morán H. (q.p.d) y Eustolia Cruz E.**

Por su amor, impulso, insistencia y por esa
gran fe que tuvieron para conmigo.

A MIS HERMANOS: José Luis, Javier, Lupita, Judith

y Edwin. Por su apoyo y cariño.

A MIS AMIGOS: Miguel, Pancho, Rosalba, Leticia, etc.

Por su amistad incondicional e invaluable.

**A TODAS AQUELLAS PERSONAS, que contribuyeron a
la realización de esta tesis.**

"GRACIAS"

Patricia Morán Cruz

Hay abundante inspiración en este amplio pasaje de mi vida y de ahí exhorto a aquellos hombres que no nacieron solo para consumir los frutos de la vida.

Y no me temo asegurar que un día, gustaréis de las eruditas meditaciones y seréis honra y decoro en lo que en mi cabe suponer necesidad y afán.

Por ello exclamo; una libre, exaltada y real dedicación a un tributo de tu modesta independencia y creatividad, a ti: PAPA; Gpe. Morán Hdz. (q.p.d), a ti que supiste inspirarme para aliviar mi expresión guardada. "GRACIAS"

Patricia Morán Cruz.

Prologo

El Propósito del diseño lógico con cualquier o a base de cualquier metodología para desarrollarlo, es lograr la mayor simplificación de la función de salida para obtener un circuito de bajo costo, y se piensa que al utilizar el menor numero de circuitos, es preferible debido a que cuesta menos, sin embargo no siempre resulta cierto cuando se utilizan circuitos integrados

En la mayoría de los casos la aplicación de un procedimiento de diseño alterno puede producir un circuito combinacional para una función dada aun mejor que el obtenido al seguir el método de diseño clásico, y el utilizar un posible procedimiento alterno depende de un problema particular y del ingenio del diseñador

Como existen varios componentes MSI que producen determinadas funciones lógicas, habria que determinar si lo que deseamos que desarrolle nuestro diseño se encuentra dentro de alguno de estos componentes MSI preferiblemente antes que utilizar las compuertas para la reducción considerable del circuito, si no, debemos entonces, realizar un método para introducir un elemento MSI en el circuito de diseño, precisamente para reducir el circuito final

Los sistemas digitales pueden ser de dos tipos; o secuenciales (los que usan elementos de Memoria, además de compuertas lógicas), o los combinacionales (combinación entre compuertas lógicas), y se debe determinar que tipo de sistema podemos utilizar en base a los conocimientos que se proporcionan en la presente tesis.

ÍNDICE

CONTENIDO.

	Pag.
Introducción	ii
CAPITULO I	
Panorama Actual del Diseño de sistemas digitales	I
Introducción	2
1.1. Utilización de ctos. Integrados	3
A).- Circuitos de Telecomunicaciones	3
B).- ACIA	4
C).- Circuitos de Línea	4
D).- Manejadores de y para periféricos	4
F).- Manejadores de visualizadores	5
G).- SCI, UART, USRT	5
J).- Utilización de multiplexores en un sistema.	5
K).- Circuitos de Integración a gran escala (Periféricos).	6
L).- Memorias de Microcomputadoras	7
a) - RAM, b) - ROM, c) - PROM, d) - EPROM y EEPROM	9
M).- Microprocesadores	10
N).- Coprocesadores Matemáticos	11
1.2. Interfaz entre circuitos digitales	12
a.- TTL y CMOS	14
b.- Conversor D/A	15
c.- Conversor A/D.	17
CAPITULO II	
Estudio de componentes básicos del Diseño Lógico	20
Introducción	21
2.1.- Puertas lógicas.	22
2.2.- Niveles de Voltaje	32
2.3.- Transformación de expresiones booleanas a diagramas lógicos	33
2.4.- Teoremas Booleanos	36
CAPITULO III	
Técnicas de Minimización de circuitos Digitales.	44
Introducción	45
3.1.- Conceptos tradicionales	46
3.2.- Teoremas de reducción.	46

3.3.- Minimización para una función de salida.	47
3.3.1.- Mapas de Karnaugh.	47
3.3.2.- Método de Quine Mc Cluskey.	56
3.4.- Minimización para varias funciones de salida.	60
3.4.1.- Método de Quine Mc Cluskey.	62
3.4.2.- Implementación de funciones booleanas empleando puertas NAND y NOR.	64

CAPITULO IV

Circuitos Combinacionales a Mediana Escala de Integración.	67
Introducción	68
4.1.- Circuitos para control de tareas y diagramas de tiempo	69
4.2.- Sumadores	70
4.3.- Restador	78
4.4.- Decodificador	79
4.5.- Multiplexor	84
4.6.- Comparador	88
4.7.- Verificadores de Paridad	90

CAPITULO V

Estudio y Aplicaciones de Familias Lógicas.	95
Introducción	96
5.1.- Elementos básicos de los circuitos integrados.	97
5.2.- Circuitos Integrados.	102
5.3.- Familias Lógicas	103
a.- Características	103
b.- Tipos.	103
5.4.- TTL y sus características.	109
5.5.- ECL y sus características	112
5.6.- MOS y CMOS y sus características.	113

CAPITULO VI

Varios Tópicos de Diseño de Sistemas Digitales.	117
Introducción	118
6.1.- Diseño de Controladores con Cartas ASM	119

a).- Flip-Flop JK	123
b).- Flip-Flop SR	125
c).- Flip-Flop T	127
d).- Flip-Flop D	128
e).- Registro de Corrimiento	130
f).- Contadores	134
6.2.- Tecnología en MSI.	136
a).- Diseño de controles programables	138
b).- Diseño de circuitos MSI.	139
Conclusiones	147
Glosario	148
APÉNDICES.	
"A"	151
Introducción.	152
a).-Representación numérica.	153
b).- Conversión de base M a base 10.	153
c).- Conversión de base 10 a base M	154
c.1.-Sistemas binarios, octal y hexadecimal	155
c.2.-Conversión de octal a binario	156
c.3.-Conversión de binario a hexadecimal	156
d).- Operaciones aritméticas con números no signados.	157
e).- Sistemas de numeración complementarios.	157
f).- Representaciones binarias.	158
f.1.- Operaciones binarias con números signados.	159
"B"	
a).- Códigos.	161
b).- Bit, Bytes, Palabras.	161
c).- Códigos binarios, BCD, reflejados, exceso 3, Gray.	162
Bibliografía	165

INTRODUCCIÓN

La actualización en los centros educativos trae consigo programas, planes de estudios y practicas que garantizan el mas alto nivel académico para la institución, esto hace posible que cada vez mayor numero de alumnos se integren a los cambios que demanda el país para su desarrollo.

Es así como se presenta el siguiente trabajo de investigación, como una breve introducción al desarrollo de Sistemas digitales.

El desarrollo de los sistemas digitales ha evolucionado, en la estructura de modelado de un sistema digital (que contiene: unidad de memoria, unidad de control, unidad para transformación de datos).

La aplicación de los circuitos combinatoriales se explica como una unidad de transformación donde no es necesario que exista una unidad de memoria.

En el primer capitulo se presenta el diseño de sistemas digitales, tomando en cuenta la estructura básica que la ha llevado hasta esta etapa; esto no se podría llevar a cabo si no se conocen tales conceptos como son: las compuertas, los circuitos secuenciales, circuitos combinatoriales apoyados con tecnologías SSI, MSI, LSI y técnicas de minimización de circuitos, es así como se desarrolla el segundo, tercero y cuarto capítulos que son dedicados exclusivamente al estudio de estos conceptos, básicos, ilustrados con algunos ejemplos prácticos.

El capitulo quinto esta enfocado a la aplicación de las diferentes familias, en este diseño, haciendo énfasis en que el objetivo principal de este capitulo es la aplicación de sistemas con Familias Logicas, y sus características de funcionamiento, así como sus diferencias.

El ultimo capitulo se enfoca a varios tópicos de diseño de sistemas digitales, proponiendo un trabajo o proyecto donde se diseña, modela, programa y construye un sistema digital, donde se utilicen los componentes y técnicas de diseño vistas, empleando controladores automaticos con cartas ASM.

CAPITULO I

PANORAMA ACTUAL DEL DISEÑO DE SISTEMAS DIGITALES

OBJETIVO:

Conocer y desarrollar un panorama básico sobre el diseño de circuitos digitales.

CONTENIDO:

- 1.1.- Utilización de circuitos Integrados*
- 1.2.- Interfaces entre circuitos digitales*
 - a.- TTL y CMOS*
 - b.- Conversores A/D y D/A*

INTRODUCCIÓN

La Electrónica digital es una tecnología en rápido crecimiento, los circuitos digitales se usan en gran cantidad de nuevos productos de consumo, equipos industriales y de control de oficina y ahora aun más en redes de comunicación

Este uso expansivo de los circuitos digitales es el resultado del desarrollo de circuitos integrados a bajo costo y la aplicación de tecnología de computadoras y visualizadores.

Algunas familias lógicas se fabrican utilizando tecnología bipolar. Estos CI contienen partes comparables a los transistores bipolares, diodos, y resistores discretos, en otro grupo de familias digitales de CI, se utiliza la tecnología metal-óxido semiconductor (MOS)

Actualmente la familia bipolar TTL (lógica transistor transistor), es la más popular. Y la más moderna y utilizada es la CMOS (Metal-óxido semiconductor complementario).

Los circuitos integrados han simplificado muchísimo el diseño de muchos complejos circuitos analógicos y digitales.

Los que se dedican a trabajar con CI deben afrontar la tarea de seleccionar los circuitos integrados y su diseño, deben consultar muchos catálogos de fabricantes y un reducido número de notas de aplicación con el fin de determinar la configuración óptima, de los circuitos integrados y del circuito que se diseña.

En los catálogos, las hojas de datos especifican los parámetros de operación en condiciones extremas y normales, pero no sirven para utilizarse como una guía de selección por que no se pueden realizar comparaciones para valorar a los CI.

A medida de nuestras necesidades se han ido creando las inquietudes de poder encontrar dos cosas, que se le de igual importancia a las aplicaciones de los CI y a la selección de dispositivos. Las configuraciones preferidas y soluciones prácticas fáciles de obtener probadas para diseños reales, se pueden obtener mediante la selección de dispositivos y con los ejemplos de diseño que ayuden a realizar un diseño para la aplicación determinada.

1.1.-UTILIZACIÓN DE CIRCUITOS INTEGRADOS

A) Circuitos de Telecomunicación.

Han proporcionado a la Industria de Telecomunicaciones , los fabricantes de CI, una gran variedad de productos.

El equipo más familiar para el usuario de un sistema telefónico es el aparato receptor transmisor. Esto conlleva a las operaciones de marcar , con el disco y el teclado, recibir y transmitir, que son básicamente las funciones de cualquier terminal, el cuál contiene los siguientes dispositivos:

- Receptor {Microtelefono}
- Transmisor
- Red de voz
- Interruptor de horquilla
- Timbre
- Marcador { Disco y Teclado }
- Rectificador en puente (utilizado en selectores digitales).

Las centrales telefónicas tienen equipos con diferentes niveles de tecnología , una central digital puede dividirse en dos grandes secciones:

- La tarjeta de línea (*line card*) y los conmutadores digitales, la tarjeta de línea lleva a cabo las funciones *BORSCHT*. La palabra *BORSCHT* es una forma de representación de las funciones que realiza la tarjeta de línea.

B.- Alimentación por batería , suministra la corriente correspondiente de c.c. para el equipo usuario.

- O - Protección contra sobre voltaje, protege la tarjeta de línea de daños transitorios por alto voltaje inducido por rayos y/o relámpagos, que caen cerca de la línea de transmisión
- R - Llamado (*ring-ring*) controla la señal de llamado inducida en la línea del usuario
- S - Supervisión para detectar diversas condiciones en la línea
- C - Codificación - codifica la voz del usuario a datos digitales , esta función la realiza el filtro *PCM* y el *CYRILIC*.
- H - Filtrado efectúa la conversión de 2 a 4 hilos que es la que se requiere entre la línea de dos hilos del usuario y la sección de codificación de 4 hilos
- T - Prueba , realiza las pruebas en la línea y en la tarjeta de línea de detección de fallas

El conmutador digital consta de una memoria de acceso aleatorio (RAM) y una red de dispositivos lógicos y de conmutación

El conmutador digital es el que controla el estado de todas las líneas de los usuarios e inicia cualquier interconexión requerida entre ellos .

B).-ACIA

De uso menos general que los UART, son los dispositivos con estructura de bus desarrollados por algunos fabricantes de microprocesadores, estos dispositivos pueden facilitar en gran manera la conexión al microprocesador, ya que su bus de datos y las señales de sincronización necesarias han sido diseñadas específicamente para el microprocesador.

C).-Circuitos de línea

Son circuitos diseñados para transmitir señales de información digital por diferentes líneas de transmisión, los circuitos de línea son manejadores de línea, receptores de línea y transeceptores. Los dispositivos pueden ser de uso general o específicamente de uso en configuración normalizada de transmisión de datos.

D).-Manejadores de Periféricos

Suelen ser circuitos que generalmente contienen 2 o mas transistores capaces de manejar voltajes y corrientes grandes, con todos los resistores de limitación de corriente y de polarización, necesarios para permitir la conexión directa del dispositivo a TTL, Mos o CMOS con un mínimo de componentes externos, además algunos manejadores incluyen algunas compuertas lógicas internas. (AND, NAND, OR, NOR) , que permiten realizar ciertas funciones lógicas en el propio circuito de interfaz.

E).- Manejadores para periféricos

Los manejadores para periféricos son CI que permiten la interconexión de niveles lógicos TTL, MOS, CMOS, con componentes de voltaje y corriente mas elevados, como lamparas, relevadores o motores.

F).-Manejadores de visualizadores

Es una interfaz con microcomputadora en CI (microprocesador), los puertos de salida de un microprocesador o con otros sistemas lógicos digitales y suministra los niveles pertinentes de corriente y voltaje requeridos por el tipo de visualizador usado.

G).- SCSI

Se pronuncia escosi y significa interfaz para sistema de computadora pequeña. (small computer system interface).

H).-UART

Receptor- transmisor sincrónico/asincrónico universal, CI LSI utilizado para la comunicación asincrónica de datos.

I).-USRT

Los primeros circuitos integrados receptores/transmisores sincrónicos, fueron básicamente los UART, modificados para operar en modo CCP.

J).-Utilización de Multiplexores en un Sistema.

Diversos sistemas electrónicos usan visualizadores alfanuméricos , y estos son un indicio de que un sistema electrónico contiene circuitería digital.

Por ejemplo .- un contador de 0 a 99 se utilizará para ilustrar la forma de como multiplexar un visualizador.

Los contadores son controlados por relojes de baja frecuencia (1Hz) , y los dos contadores de década se conectan a un multiplexor, así, es que decodifican alternativamente a los visualizadores , (en este caso *LED* de 7 segmentos) , el reloj multiplexador generará una salida de una frecuencia alta (100Hz), dicha señal genera la visualización de las unidades y decenas del contador alternativamente.

Ahora bien el multiplexaje reduce el consumo de potencia del visualizador , no requiere de un decodificador extra, entonces el multiplexar señales con los visualizadores ahorra energía.

K).-Circuitos de Integración a gran escala**-Periféricos LSI**

La Revolución de los microprocesadores significó un profundo cambio de dirección en el diseño de sistemas. En el lugar de construirlo, con cientos o miles de circuitos integrados y lógica cableada, los sistemas pudieron reducirse a la combinación formada por unos cuantos programas y un microprocesador.

El éxito de esta revolución fue la considerable disminución en el número de componentes y la facilidad, por lo menos en teoría, para efectuar cualquier cambio por medio de la programación.

El microprocesador con la ayuda de algunos componentes externos, se vio que podría controlar sus periféricos como:

- Teclados
- Visualisadores
- Impresoras
- Discos
- ELS (serie y paralelo)
- Unidades de cinta

El reto ha sido permitir que el microprocesador controle todos estos periféricos y pueda también ejecutar los programas, que es su responsabilidad principal.

Otra consideración, con relación a la utilización de microprocesadores para el control de periféricos, es que en los sistemas de producción, los cambios no son tan sencillos como parece. En la práctica, la programación puede resultar tan hermética como la lógica cableada.

Ahora, muchas de las funciones requeridas de control se encuentran más allá de la capacidad de la mayor parte de los microprocesadores excepto por los de "rebanadas de bits". Pero el uso de estos como controladores no es práctico por su costo, y cuando se les compara con controladores realizados con circuitos SSI/MSI se ve esta diferencia

Muchos de los controladores LSI actuales como los *UART*, tienen la arquitectura de circuitos estándar SSI/MSI, que surgieron paralelamente o antes que los microprocesadores

Así, la revolución del microprocesador no origina otra revolución sino una evolución, la evolución de los periféricos, en vez de ser realizador de las tareas de control

Los propios controladores de periféricos pueden ser programables e inclusive pueden ser más rápidos y complejos que el microprocesador.

L).-Memorias de Microcomputadoras.

Cuando surge un nuevo producto los diseñadores deben elegir entre utilizar dispositivos analógicos o digitales. Claro que nuestra cuestión es utilizar dispositivos digitales.

Si la unidad es de entrada, procesa o saca datos alfanuméricos, la elección es claramente digital. También, si la unidad tiene cualquier tipo de memoria, programa almacenado, la elección es claramente digital.

La circuitería digital se ha vuelto muy popular, sin embargo los sistemas electrónicos complejos contienen dispositivos analógicos y digitales.

La memoria de las microcomputadoras es un ejemplo de aplicación de los dispositivos de almacenamiento de datos denominados memorias.

La memoria interna de un sistema está compuesta básicamente de la memoria semiconductora llamada RAM.

Los datos y la mayoría de los programas se almacenan normalmente en dispositivos magnéticos de almacenamiento masivo denominados discos flexibles.

En realidad cada dispositivo como el; teclado, monitor, unidad de disco y CPU tienen dispositivos minúsculos llamados memorias, estos dispositivos pequeños de memoria toman la forma de registros y cerrojos.

Los dispositivos de almacenamiento RAM y ROM vienen en forma de CI y están montados en tarjetas de circuitos impresos.

Entonces en un computadora es usual tener al menos un CI ROM y muchos RAM.

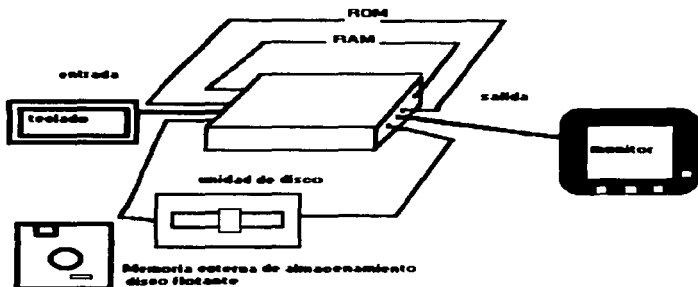


Fig. 1.1.- Dispositivos de almacenamiento de una computadora

Existen diferentes tipos de memorias como son:

a).- Memoria de Acceso Aleatorio. RAM.

Es un sistema de computo electrónico por así decirle, que permite que un conjunto electrónico almacene datos en forma de señales eléctricas, las acciones de almacenar y recuperar datos, son ejecutados por el CPU, la memoria es solo un lugar de almacenamiento de datos, es temporal ya que solo se almacena mientras este presente la corriente eléctrica, esta memoria es de lectura y escritura, es volátil, ya que solo cuando permanece con energía almacena información, pero cuando no tiene energía pierde toda la información almacenada anteriormente. Existen dos tipos de RAM la dinámica que es utilizada normalmente en dispositivos de almacenamiento primario de macros, computadoras, minicomputadoras y computadoras personales y la RAM estática es utilizada en microcomputadoras que requieren poca capacidad de almacenamiento, versiones de alta velocidad para memoria buffer de microcomputadoras, versiones de bajo consumo de energía para computadoras portátiles.

b).- Memoria ROM

Se trata de una memoria de solo lectura, esta formada por una pastilla (chip), y esta reservada para grabar ciertos datos o información esenciales para almacenarse permanentemente como en el caso del encendido de una PC, solo permite que se lea la información contenida dentro de la misma, no permite que se sea modificada.

Esta clase de memoria es utilizable en almacenamiento de programas para computadoras personales, almacenamiento de conjuntos de caracteres para despliegues visuales e impresoras.

c).- Memoria PROM.

En este tipo de memorias el tiempo de desarrollo y los costos son mas bajos, es mucho mas fácil corregir errores de programas y actualizar los productos cuando las Prom pueden ser programadas una vez como una ROM, pero su ventaja es que puede hacerse en cantidades limitadas y pueden ser programadas en laboratorios o tiendas.

D).- Memoria EPROM y EEPROM.

Son del mismo tipo que la Prom pero la Eprom tiene integrada una ventanita de cuarzo con la cual se le puede borrar la información haciéndole pasar un az de luz ultravioleta. Y la EEPROM es la Prom pero eléctricamente borrrable, su ventaja es que puede permanecer dentro de la tarjeta de circuito mientras se borra y se reprograma.

La PROM es utilizada en microprogramas con instrucciones de control para minicomputadoras; aplicaciones militares y en automóviles. La capacidad de reprogramación de la EPROM hace mas fácil corregir errores durante el desarrollo de los programas. En las aplicaciones de ROM y EPROM que requieren modificaciones ocasionales a los programas o los datos es donde se utiliza la EEPROM.

M).-MICROPROCESADORES

Existen algunos microprocesadores como 8086, 8088, 286, 386 e incluso 486, que ya se han ido volviendo obsoletas, ya que hace una década la PC de locura era la IBM 8088, la cual no ofrece espacio más que para ser actualizada.

Al principio de los ochenta aparece la IBM XT pero es un modelo demasiado lento y es demasiado vieja para aceptar la mayoría de los componentes nuevos.

A mediados de los ochenta aparece la tecnología AT 286, que es 5 veces más rápida que la XT y puede ser escalable a 3866 con una tarjeta madre nueva.

A fines de los ochenta se propone la 386, los diseñadores de chips dieron la pauta con el chip 386 de aquí se realizó una nueva generación que son la 486 y la pentium 586, en estos casos se podrá añadir varios componentes, sin causar reacciones contraproducentes con los reemplazos y las mejoras más comunes son; que incluyen unidades de disco duro, mayor memoria y tarjetas de video rápidas

Existen otras innovaciones en el mundo de la electrónica, como son los ratones y digitalizadores y módem.

Ahora bien algo parecido al chip del CPU existe el coprocesador matemático, es una especie de calculadora, al utilizar respuestas rápidas del coprocesador matemático, el CPU opera más rápido, se utiliza para agilizar el calculo matemático, pero los chips 486DX ya incluye el coprocesador matemático, las 486Sx no lo tienen.

Además existe un componente que tiende a utilizar tarjetas de SCSI, se trata de la unidad de disco compacto, las unidades SCSI utilizan una tarjeta controladora, son la opción mas costosa, pero se usan porque pueden elaborar algunos asuntos, como por ejemplo la SCSI podrá dirigir la unidad de disco compacto, donde se tendrá la posibilidad de ser conectado con una unidad de respaldo de cinta o a un CD ROM.

Los fabricantes de disco compacto lograron colocar información de muy diferentes maneras, fabricando diferentes reproductores para acoplarse con los diferentes estándares.

Al revelar fotografías se puede utilizar este procedimiento para grabar imágenes en disco compacto y las unidades compatibles con KODAK Photo CD podrán desplegar dichas imágenes en el monitor.

Existe otra pero muy costosa unidad de disco llamada WORM que permite escribir información en el disco pero solo una vez, tal vez a futuro se vuelva más accesible al público para su utilización.

Existen algunos microprocesadores como 8086, 8088, 286, 386 e incluso 486, que ya se han ido volviendo obsoletas, ya que hace una década la PC de locura era la IBM 8088, la cual no ofrece más espacio para ser actualizada.

N).-Coprocesadores matemáticos

También podemos hablar acerca del coprocesador matemático, es útil cuando se realizan operaciones matemáticas intensas, incluyendo gráficas, aplicaciones científicas y problemas de hojas de cálculo.

Existen microprocesadores que ya traen incluido el coprocesador como por ejemplo: el microprocesador 80486DX. Pero hay otros microprocesadores que necesitan del chip, que funge como coprocesador matemático, este circuito se conecta en la tarjeta madre¹ de la PC.

¹ La tarjeta madre es una tarjeta de circuito que contiene el o los chips principales como son el del microprocesador, etc.

1.2.-INTERFACES ENTRE FAMILIAS.

Una interfaz es un método que trata de conectar dos dispositivos electrónicos de complejidad similar como por ejemplo, las compuertas lógicas.

Se asegura que dos puertas de una misma familia se pueden conectar entre si, sin ningún problema, esto sucede tanto en TTL como en circuitos CMOS.

Pero para conectar circuitos diferentes, debemos considerar que operan a diferentes tensiones, o bien sus requerimientos son diferentes, pero existen interfaces (técnicas sencillas de interconexión), que podemos aplicar.

Para acoplar un TTL estándar con un CMOS se utiliza un resistor de (pull-up) 1K Ω

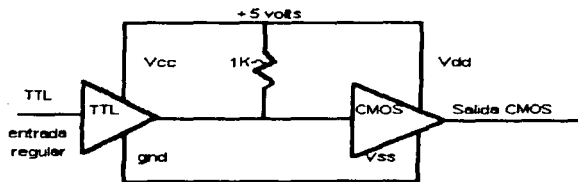


Fig. 1.2.1.- Interfaz entre TTL y CMOS

Para un CI TTL de baja Potencia y CMOS se utiliza un resistor de (pull-up) de 2.2K Ω .

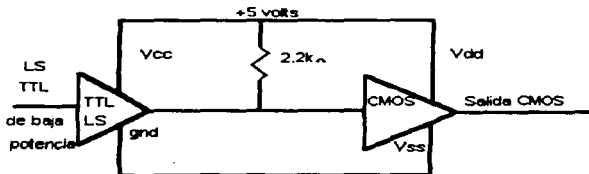


FIG.: 1.2.2.- Interfaz entre TTL de baja Potencia y CMOS

La interfaz de CMOS a TTL es aun más sencillo, ya que la salida del CMOS se puede conectar a una sola entrada del TTL de baja potencia, fig. (3).

La CMOS 74H00 se puede conectar hasta un máximo de 10 entradas TTL de baja potencia, compartiendo la misma fuente de alimentación.

Se requiere de un buffer especial CMOS 4049 entre CMOS y TTL , cuando se requieren de más conexiones , ya que un buffer se puede conectar a 2 entradas TTL.

Ahora bien cuando opera con una fuente diferente de alimentación , se requieren de otros elementos para conectar.

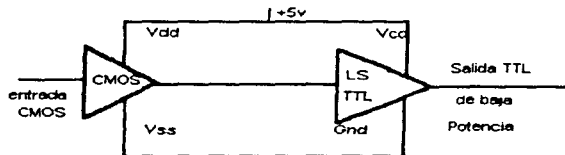


Fig. 1.2.3.- Interfaz CMOS a TTL Schottky

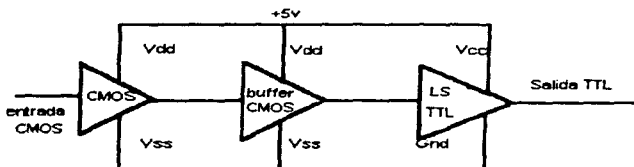


Fig 1.2.4.- Interfaz CMOS a TTL extendida utilizando un buffer de CMOS

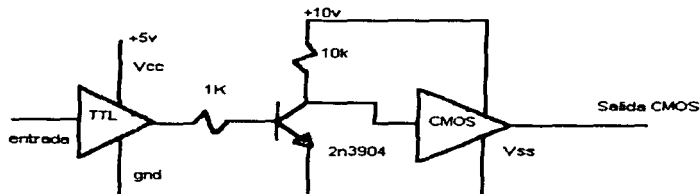


Fig.1 2.5.- Interfaz CMOS a TTL estandar utilizando un buffer de CMOS

a) Interfaz TTL a CMOS utilizando un transistor.

En este caso el transistor y los resistores traducen las salidas TTL de baja tensión a las entradas de alta tensión necesarias para que opere el inversor CMOS, la salida del inversor CMOS es una tensión que varía entre GND (tierra) y + 10v.

El uso tan amplio de circuitos lógicos integrados, como los componentes TTL, MOS, CMOS, en los sistemas de microprocesadores y microcomputadoras han generado la necesidad creciente de interfaz, estos dispositivos toman los niveles bajos de voltaje y corriente de los puertos de salida de la computadora de otros circuitos de bajo nivel, y los adaptan a los niveles mayores de voltaje y corriente propios de los circuitos periféricos como visualizadores, relevadores, lamparas, impresoras, etc.

Tales interfaces se pueden dividir según su aplicación en:

- A) Manejadores para periféricos
- B) Circuitos de línea
- C) Manejadores para visualizadores
- D) Interfaz de memoria

Los visualizadores más comunes son:

- a).- Diodos LED
- b).- Visualizadores fluorescentes al vacío
- c).- Plasma en cc o descarga gaseosa
- d).- Plasma en ca
- e).- Cristal líquido (LCD)

b).- Conversión D/A y A/D

Con frecuencia los sistemas digitales tienen que ser conectados con equipos analógicos.

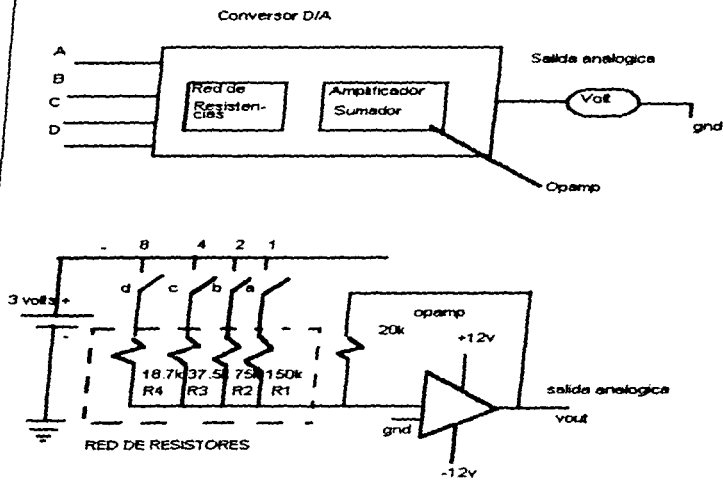
Convertor D/A.- Su trabajo es transformar una entrada digital a una salida analógica.

Pensemos en un convertor D/A que para diferentes combinaciones de entradas ABCD obtenemos, un nivel de voltaje de salida asociado a cada combinación .

Este convertor consta de una red de resistencias y un amplificador sumador , las redes de resistencias, asigna adecuadamente las cargas a las entradas , y el amplificador sumador gradúa la tensión de salida de acuerdo con los valores asociados a una señal digital .

Se utiliza normalmente al amplificador operacional para que funcione como un amplificador sumador en el convertor D/A.

1).- Diagrama a bloques de un:



2).- Diagrama electrico.

FIG. 1.2.6.- Convertor D/A

La resistencia R4 es de mas bajo valor , la R3 es el doble de R4, la R2 es el doble de R3, R1 es dos veces mayor que R2, y son valores muy pequeños para que el convertor sea preciso.

b.1).-CONVERTIDOR A/D

Un convertidor A/D produce una señal digital que representa a la señal analógica después de haberla recibido a la entrada.

En realidad un conversor A/D es mas complejo y mas largo que un D/A, y aunque hay conversores A/D que vienen integrados en un solo CI, existen varios métodos para diseñarlos que sirven para verificar el rendimiento de un Convertidor A/D.

El siguiente diagrama muestra los componentes principales de un CA/D.

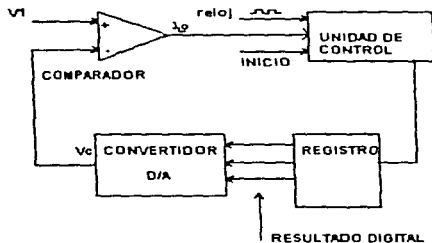


FIG. b.1 Diagrama a bloques de un CA/D.

El Reloj manda la señal que permite operar al CA/D, dentro de la unidad de Control se encuentran circuitos lógicos que generan la secuencia de operaciones adecuadas en respuesta al comando de inicio, que permite la conversión.

Ahora bien el comparador, tiene 2 entradas analógicas y una salida digital, que intercambia estados, dependiendo de que entrada analógica se trate (su valor en amplitud).

La operación de los CA/d mostrados en la figura anterior, es la siguiente:

a).- Cuando "Inicio" se encuentra en un "1" lógico, se comienza la operación de conversión.

b).- Mediante la señal de reloj, la unidad de control modifica continuamente el número binario que se encuentra almacenado en el registro.

c).- Entonces el convertidor A/D convierte el número binario del registro en un voltaje analógico, V_c .

d).- En tanto en el comparador los voltajes V_1 y V_c (que son entradas analógicas los dos) son comparados, mientras $V_c < V_1$ el comparador manda un "1" lógico, cuando $V_c > V_1$, entonces el comparador tendrá una salida igual a "0", con lo cual se suspende el proceso de modificación del número binario en el registro, que es el que permite conocer el resultado digital acorde a la señal analógica de entrada, en este punto, V_c es un valor muy aproximado de V_1 y el número digital del registro que es el equivalente digital de V_c , es así mismo el equivalente digital de V_1 , en los límites de resolución y exactitud del sistema.

Ejemplo:

Determinar para los siguientes datos lo siguiente:

- a).- el equivalente digital,
- b).- tiempo de conversión,
- c).- resolución del conversor.

Datos:

Salida igual $S = 5.11$ volts y una entrada de 8 bits.

El voltaje de umbral (voltaje mínimo para exceder el voltaje V_1 de entrada) es igual a $V_t = 0.1 \text{ m V}$.

El voltaje obtenido es $V_1 = 4.32 \text{ volts}$

Entonces:

El CD/A tiene una entrada de 8 bits, por lo que el número posible de etapas es de : $2^9 - 1 = 511$, de tal manera que el tamaño de la etapa es :

$$5.11 / 511 = .01 \text{ v} = 10 \text{ m volts}$$

Entonces el V_c crece en etapas de 10 mv cuando el contador cuenta hacia arriba desde cero.

Y como $V_1 = 4.32\text{v}$, y $V_t = 1\text{mv}$, entonces V_c tiene que llegar a 4.32 v o mas antes que el comparador cambie a bajo. Por lo que esto requerirá de :

$$4.32 / 10\text{mv} = 432 \text{ etapas.}$$

Entonces al término de la conversión el contador tendrá el equivalente binario de 432, el cual es : 110110000, este es el equivalente deseado para $V_1 = 4.32$ volts, como lo produce este convertidor A/D.

Ocurrieron 432 pulsaciones del reloj a razón de una por microsegundo por lo que el tiempo de conversión será : de 432 μs .

La resolución de este convertidor es igual al tamaño de etapa del convertidor D/A, que es 10 mv. Que en porcentaje es $10\text{mv} / 5.11\text{v} \times 100 \text{ por ciento} = 0.19\% \cong 0.2\%$.

CAPITULO II

ESTUDIO DE COMPONENTES BÁSICOS DEL DISEÑO LÓGICO

OBJETIVO:

Conocer los principios del Álgebra Booleana para poder manipular ecuaciones booleanas y diseñarlas con compuertas.

CONTENIDO:

- 2.1.- Compuertas lógicas
- 2.2.- Niveles de Voltaje y su relación con variables lógicas .
- 2.3.- Transformación de expresiones booleanas a diagramas lógicos .
- 2.4.- Teoremas Booleanos

INTRODUCCIÓN

El Hecho de que se realicen diferentes tipos de circuitos lógicos en sistemas de control digitales, requiere de la manipulación o transformación de información a niveles de voltaje, para realizar estas consideraciones se localizan dos posibles valores lógicos de referencia, los cuales nos permiten utilizar una lógica binaria ¹, y en función de esta procedemos a localizar preferentemente una solución óptima sencilla y más compacta del desarrollo de circuitos.

Entonces especificamos el uso del álgebra booleana como una herramienta ideal para realizar y diseñar circuitos o sistemas digitales.

Este tipo de álgebra es más fácil y sencilla de manejarse en comparación a la lógica ordinaria.

Al utilizar las diferentes composiciones del álgebra booleana nos permite aplicar las compuertas básicas y lógicas capaces de tomar una decisión, también podemos agregar que las familias lógicas utilizan circuitos integrados con compuertas lógicas.

¹ Ver apéndices

2.1.-COMPUERTAS LÓGICAS Y BÁSICAS

Los circuitos lógicos llamados también digitales son elementos básicos para la combinación de sistemas digitales, como aplicaciones podemos mencionar controladores (para sistemas de mecanismos), digitales instrumentos de medición, etc.

Estos circuitos digitales trabajan con una lógica binaria², ya que denominan los elementos de entrada y/o de salida con un "0" o un "1" lógico.

Entonces existe manipulación de información por medio de bits³, donde cada señal representa una variable y se representan diversas combinaciones de los circuitos digitales según convenga.

Es por esto, que debemos definir el uso de estas combinaciones con la ayuda de las tablas de verdad y el álgebra booleana.

Las compuertas lógicas utilizan el álgebra booleana como herramienta para diseñar sistemas digitales. En este tipo de algebra las variables solo pueden tener dos valores posibles "0" o "1".

Estas cantidades lógicas se emplean frecuentemente para representar los niveles de voltaje de entrada y salida de un circuito.

Por lo que podemos decir que estas cantidades lógicas representan el estado de una variable de voltaje, el cual se le denomina nivel lógico.

Ejemplo:

NIVEL LÓGICO	{ 0	1 }
	falso	verdadero
	No	Si
	Bajo	Alto
	Desactivado	Activado
	Abierto	Cerrado

² idem

³ idem

Se emplean símbolos alfabéticos por lo regular, para representar las variables lógicas de entrada salida de un circuito digital en donde la variable toma uno de los dos posibles valores en determinado momento (0 o 1).

Ejemplo:

Variable de entrada $A = 0$

Variable de salida $B = 1$

El álgebra booleana utiliza únicamente tres operaciones aritméticas lógicas básicas.

A) OR se trata de una adición lógica y se identifica por (+)

B) AND se trata de un producto lógico y se simboliza (.)

C) NOT inversión lógica y se simboliza con (-)

TABLAS DE VERDAD

Al hacer uso de las operaciones lógicas se crean ciertas combinaciones de variables de entrada en relación con una salida.

Estas relaciones obtenidas como respuestas a las diversas combinaciones de Niveles lógicos en las entradas las podemos mostrar mediante una tabla de verdad.

Sean A y B entradas de un circuito y Z la salida del mismo:

ENTRADAS		SALIDAS
A	B	Z
0	0	?
0	1	?
1	0	?
1	1	?

Niveles lógicos

La salida en nivel lógico depende de la combinación de las entradas y de cualquiera de las 3 operaciones básicas.

Las tablas de verdad por lo regular tiene una variable de salida (aunque en otros casos se tiene mas de una salida) y 2 o mas variables de entrada.

OPERACIONES LÓGICAS

OR : Tomemos a A Y B como dos variables de entrada independientes, que al combinarse se obtiene una salida Z.

Entonces :

$$Z = A + B \text{ y se lee : Z es igual a A o B.}$$

Con esto podemos realizar la tabla de verdad de esta operación como sigue :

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

Su simbolo es el siguiente :

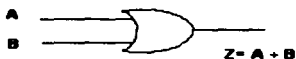


FIG. 2.1.1.- COMPUERTA OR

Si analizamos una compuerta OR de 3 entradas, observando la siguiente tabla de verdad, la salida será "1" lógico en el caso de que una o mas entradas sean 1 y la salida sea igual a "0"0, en el caso de que todas sus entradas sean, 0 como sigue:

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

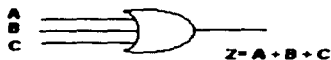


FIG.2.1.2.- COMPUERTA OR (3 ENTRADAS)

COMPUERTA AND

La compuerta AND se identifica mediante el uso de la multiplicación o el producto entre dos o mas variables, entonces para 2 entradas A y B tenemos que la salida será en álgebra booleana :

$$Z = A \cdot B$$

Y su tabla de verdad :

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

Simbolo de la compuerta AND:



FIG. 2.1.3.- COMPUERTA AND

Las tablas de verdad por lo regular tiene una variable de salida (aunque en otros casos se tiene mas de una salida) y 2 o mas variables de entrada.

En este caso siempre que A o B sea "0", la salida o producto será igual a "0", y solamente cuando las dos entradas A y B sean "1", la salida será igual a "1".

Haciendo uso de interruptores para demostrar lo anterior, tenemos:

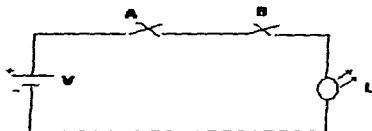


FIG. 2.1.4.- COMPUERTA AND CON INTERRUPTORES

La lampara es la salida, cuando se enciende es porque los dos interruptores están cerrados y son igual a "1".

Entonces para mas de 2 entradas, vemos que la salida de una compuerta AND será igual a "1" solamente cuando todas las entradas sean "1", así:

FIG. 2.1.5.- Tabla de verdad y símbolo para compuerta AND de 3 entradas

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

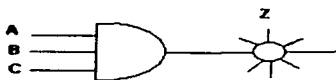


FIG. 2.1.6.- COMPUERTA AND (3 ENTRADAS)

Como ejemplo tomemos en cuenta el control para un equipo de bombeo, con un tanque bajo y un tanque alto:

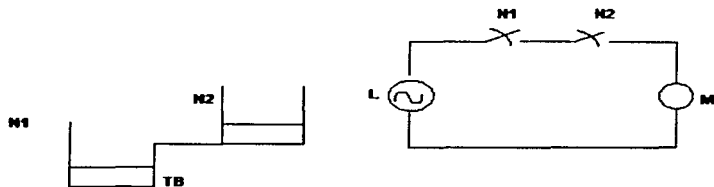


FIG. 2.1.7.- Control de bombeo

Cuando se active el interruptor N1, quiere decir que el tanque bajo esta lleno y se encenderá la bomba para llenar el tanque alto, siempre y cuando el interruptor N2 se active mostrando que el tanque esta vacío. La tabla de verdad que muestra los requerimientos de este ejemplo es :

N1	N2	motor
0	0	0
0	1	0
1	0	0
1	1	1

$$M = N1 \cdot N2$$

Expresándose en álgebra booleana, donde M es la salida y significa que el motor de la bomba funciona solo cuando las dos entradas - N1 y N2 son activadas.

COMPUERTA NOT (INVERSOR)

Este circuito tiene solamente una entrada y su característica es la de invertir la salida, esto es su nivel lógico de salida es siempre contrario al de la entrada, entonces:

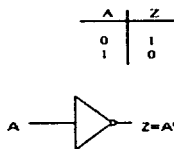


FIG. 2.1.8.- símbolo y tabla de verdad (Compuerta NOT)

donde la expresión $Z = A'$ se lee Z es igual al inverso de A.
Entonces si $A = 0$, la salida $Z = 1$ y si $A = 1$, la salida será $Z = 0$.

Estas compuertas anteriormente mencionadas son la base fundamental para el desarrollo de otros tipos de circuitos combinacionales como son las compuertas : NAND, NOR y OR-EXCLUSIVA.

Al utilizar el circuito inversor en combinación con una compuerta AND, se produce el complemento de la función AND, quedándose como resultado la función AND, quedándose como resultado la función NAND.

Así mismo, combinando la compuerta OR con la función NOT, obtenemos como resultado el complemento de la función OR o sea la compuerta NOR.

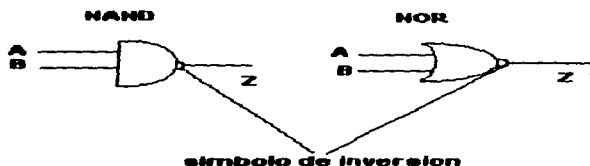


FIG. 2.1.9.- compuertas NAND y NOR

Este tipo de compuertas son mayormente utilizadas en el diseño de circuitos lógicos debido a que son construidos más fácilmente con transistores y además la implementación de los mismos con funciones booleanas es más sencilla al utilizar estas compuertas.

Las tablas de verdad para las compuertas NAND y NOR son:

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

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

Expresadas en Álgebra Booleana como :

$$\text{NAND: } Z = (A \cdot B)'$$

$$\text{NOR: } Z = (A + B)'$$

OR EXCLUSIVO

Este circuito produce una salida $Z = 1$ siempre y cuando las dos entradas estén en niveles opuestos de voltaje y puede representarse mediante la utilización de la combinación de las 3 compuertas básicas NOT, AND y OR de la manera siguiente:

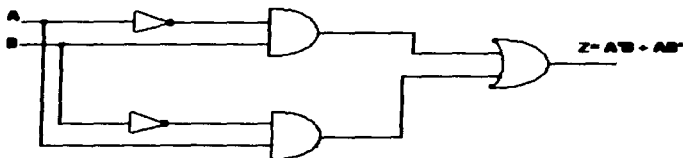


FIG. 2.1.10.- OR-EXCLUSIVA



FIG. 2.1.11.- Símbolo que representa a la compuerta OR-EXCLUSIVA

TABLA DE VERDAD

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

2.2.-NIVELES DE VOLTAJE Y SU RELACIÓN CON VARIABLES LÓGICAS

Como ya sabemos existen solamente dos posibles valores en la lógica binaria, que son el "1" lógico y el "0" lógico, los cuales son resultados, ya sea de entrada o de la salida de un circuito.

Por lo que se hacen ciertas designaciones de niveles de voltaje como un alto H (Hi) y un nivel bajo L (Low) para distinguirse entre dos valores de voltaje, uno debe ser mas grande o de mayor intensidad que el otro.

Al realizar la representación de la lógica "1" con la utilización de un nivel alto de voltaje, se define como lógica positiva.

Al seleccionar un nivel bajo L para representar una lógica "1", se trata de un sistema de lógica negativa. El tipo para representar una lógica "1", se trata de un sistema de lógica negativa. El tipo de lógica no se trata de determinar por medio de la polaridad de la señal, sino de la asignación de valores lógicos de acuerdo a la amplitud de la información.

Al seleccionar la familia lógica del circuito, reconocerán un nivel alto o bajo de voltaje, generalmente se utiliza el valor típico.

También existen algunos circuitos que dependen de la asignación de polaridad, estos se comportan como compuertas NOR y NAND.

Por ello existe un indicador de polaridad, esto indica que puede signarse una lógica negativa a la terminal, entonces el circuito puede funcionar ya sea como una NAND de lógica positiva o como una NOR de lógica negativa.

Estas conversiones de lógica positiva a negativa y viceversa son el resultado del intercambio de los "0" por "1" y los "1" por "0", en las entradas y en las salidas del circuito, produciendo la dualidad de una función. Existen hojas de datos que nos sirven para identificar los niveles de voltaje como se muestra en la siguiente tabla:

FAMILIA IC	VOLTAJE DE SUMINISTRO (V _{CC})	NIVEL DE VOLTAJE MARGEN TÍPICO		NIVEL DE VOLTAJE MARGEN TÍPICO	
TTL	V _{CC} =5	2-4.5	3.5	0-0.4	0.2
ECL	V _{EE} =-5.2	-0.45/-0.7	-0.8	-1.9	-1.8
CMOS	V _{DD} =3-10	V _{DD}	V _{DD}	0-0.5	0
LÓGICA POSITIVA o NEGATIVA		LÓGICA 1		LÓGICA 1	
		LÓGICA 0		LÓGICA 0	



TTL 7400 Compuerta NAND de lógica Positiva

Fig. 2.2.1 Símbolo de la NAND

X	Y	Z
L	L	H
L	H	H
H	L	H
H	H	L

FIG. 2.2.2.- Tabla de Verdad y Símbolo de la Compuerta NAND de lógica positiva.

2.3.-TRANSFORMACIÓN DE EXPRESIONES BOOLEANAS A DIAGRAMAS LÓGICOS.

Sabemos que la operación de un circuito puede definirse por medio de una función booleana y partiendo de esta podemos realizar un diagrama de conexiones del circuito.

Por ejemplo si tuviésemos la función : $F1 = x + y + z$, como función de salida de un determinado circuito, sabemos que podríamos utilizar directamente una compuerta de 3 entradas OR, para obtener dicha función de salida.

Y si tenemos $F2 = x'y + z$, directamente observamos que necesitamos una compuerta AND, un inversor y una compuerta OR para realizar dicha función, de esta manera podemos realizar el diagrama de conexiones de un circuito con una función de salida mas complejo.

Entonces podemos utilizar este método directo para minimizar y reproducir los circuitos en diagramas lógicos.

Por ejemplo:

Si tenemos la función Booleana:

$$F = xy + x'yz + y'z$$

Entonces necesitamos :

1.- 2 inversores para:

$$x' \text{ y } y'$$

2.- Necesitamos 3 compuertas AND, 2 de 2 entradas para xy y $y'z$ y otra de 3 entradas para $x'yz$.

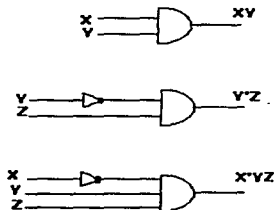


FIG. 2.3.1 Diagrama con compuertas

Y por ultimo necesitamos hacer uso de una compuerta OR de 3 entradas:

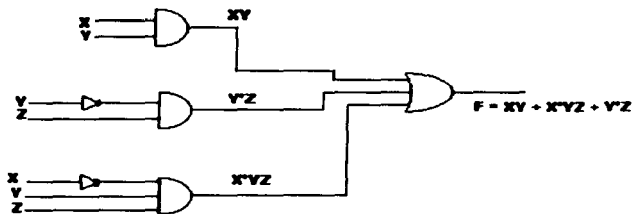


FIG. 2.3.2 Diagrama con compuertas

Ejemplo 2.- Realizar el diagrama del circuito para:

$$F = (x + y)x' + xy$$

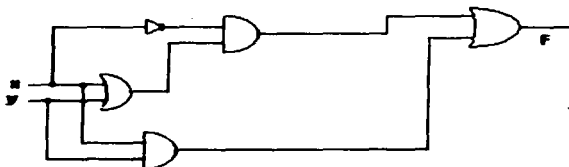


FIG. 2.3.3.- Diagrama con compuertas

Ejemplo 3.- Trazar el diagrama lógico de la siguiente expresión booleana:

$$F1 = (x'y + z)' + x'yz + (xz)'$$

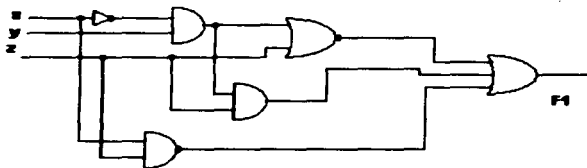


FIG.2.3.4.- Diagrama con compuertas

2.4.- TEOREMAS BOOLEANOS

Podemos hacer uso de diversos teoremas booleanos (reglas), para simplificar las expresiones y los circuitos lógicos, como sigue:

-Teoremas con una sola variable, donde "x" puede tomar el valor de "1" o "0".



a).- $x * 0 = 0$ si $x = 0$; $0 * 0 = 0$ si $x = 1$; $1 * 0 = 0$



b).- $x * x = x$ si $x = 0$; $0 * 0 = 0$, si $x = 1$; $1 * 1 = 1$



c).- $x + 0 = x$ si $x = 0$; $0 + 0 = 0$, si $x = 1$; $1 + 0 = 1$



$$d). - x + x = x \quad \text{si } x = 0; 0+0=0, \text{ si } x=1; 1+1=1$$



$$e). - x \cdot 1 = x \quad \text{si } x=0; 0 \cdot 1=0, \text{ si } x=1; 1 \cdot 1=1$$



$$d). - x \cdot x' = 0; \quad \text{si } x = 0; 0 \cdot 1 = 0, \text{ si } x=1; 1 \cdot 0 = 0$$



$$f). - x + 1 = 1; \quad \text{si } x = 0; 0+1=1, \text{ si } x=1; 1+1=1$$



$$g). - x + x' = 1, \quad \text{si } x=0; 0+1=1, \text{ si } x=1; 1+0=1$$

-Teoremas con múltiples variables:

- 1).- $x + y = y + x$
- 2).- $x \cdot y = y \cdot x$
- 3).- $x + (y + z) = (x + y) + z$
- 4).- $x(yz) = (xy)z$
- 5).- $x(y + z) = xy + xz$
- 6).- $(w + x)(y + z) = wy + xy + wz + xz$
- 7).- $x + xy = x(1 + y) = x(1) = x$
- 8).- $x + x'y = x + y$

Demostración del teorema (8):

$$\text{Para } x=0, y=0;$$

$$x + y = 0+0=0$$

$$0 + 1(0) = 0 + 0 = 0$$

$$\begin{aligned} \text{Para } x=0, y=1 \\ 0 + 1(1) = 0 + 1 = 1 \end{aligned}$$

$$x + y = 0 + 1 = 1$$

$$\begin{aligned} \text{Para } x=1, y=0 \\ 1 + 0(0) = 1 + 0 = 1 \end{aligned}$$

$$x + y = 1 + 0 = 1$$

$$\begin{aligned} \text{Para } x=1, y=1 \\ 1 + 0(1) = 1 + 1 = 1 \end{aligned}$$

$$x + y = 1 + 1 = 1$$

Ejemplo :

Simplificar la siguiente función :

$$\begin{aligned} F &= AB'D + AB''D'' \quad \text{utilizando el teorema (5)} \\ &= AB'(D + D'') \quad \text{utilizando el teorema } x + x' = 1 \\ &= AB'(1) = AB' \end{aligned}$$

Teoremas de DeMorgan

Los teoremas de DeMorgan son muy útiles para la simplificación de expresiones en las cuales se invierte un producto a suma de variables.

$$A). - (x + y)' = x' \cdot y'$$

$$B). - (xy)' = x' + y'$$

EJERCICIOS SOBRE EL CAPITULO:

1.- Después de reducir a su mínima expresión, implementar las funciones con compuertas lógicas y checar tabla de verdad.

$$a). - (A+C+D)(A+C+D')(A+B') \quad \dots\dots \text{reducir a 4 literales.}$$

Aplicando la ley distributiva:

$$(AA+AC+AD'+CA+CC+CD'+DA+DC+DD')(A+C'+D)(A+B') =$$

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

$$(A+AC+A+C+C)(AA+AB'+CA'+C'B+DA +DB') =$$

$$(A+AC+C)(A+AB'+CA'+C'B+DA+DB') =$$

$$\begin{aligned}
 &AA + AAB' + AC\bar{A} + AC'B + ADA + ADB' + ACA + ACAB' + ACC\bar{A} + ACC'B + ACDA + ACDB' + CA \cdot C \\
 &AB' + CC\bar{A} + CC'B + BDA + CDB' + \\
 &A + AB' + AC' + ACB' + AD + ADB' + AC + ACB' + ACD + ACDB' + CA + ACB' + ACD + CDB' + \\
 &A + AB' + A(C' + C \cdot C) + AC(B + B') + AD + ADB' + ACD + ACDB' + ACB' + CDB' + \\
 &A + AB' + AD + ADB' + ACD + ACDB' + ACB' + CDB' + \\
 &A + AB' + AD + ADB' + ACD + CDB'(A + 1) + ACB' + \\
 &A + AB'(1 + D) + AD(1 + C) + ACB' + CDB' + \\
 &A + AB' + AD + ACB' + CDB' + \\
 &A(1 + D) + AB'(1 + C) + CDB' + \\
 &A + AB' + CDB' + \\
 &A(1 + B') + CDB' + \underline{A \cdot B' \cdot C \cdot D}
 \end{aligned}$$

Alambrado:

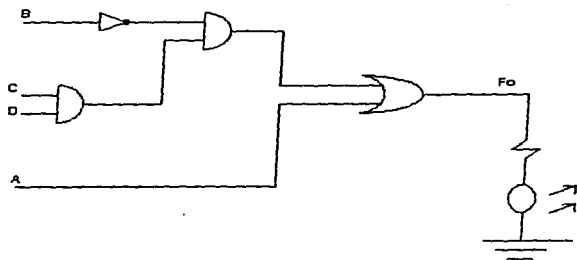


FIG.2.5 DIAGRAMA LÓGICO

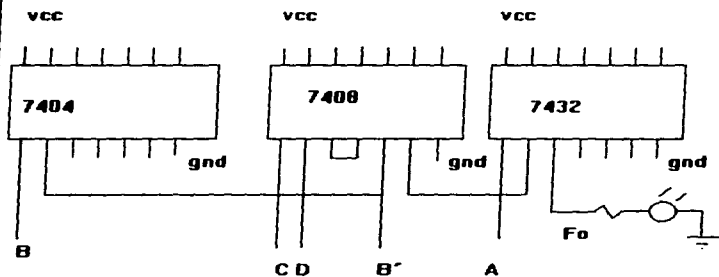


FIG. 2.6 ALAMBRADO

TABLA DE VERDAD

<u>ABCD</u>	<u>Fo</u>
0000	0
0001	0
0010	0
0011	1
0100	0
0101	0
0110	0
0111	0
1000	1
1001	1
1010	1
1011	1
1100	1
1101	1
1110	1
1111	1

$$b).- (A' + AB)' + (B' + AB)'$$

Utilizando leyes de Morgan

$$(A'')(AB)' + (b'')(AB)' = A(AB)' + B(AB)' =$$

Factorizando:

$$(AB)'(A+B) = (A' + B')(A+B) = A'A + A'B + B'A + B'B = A'B + AB' = A \oplus B \dots$$

OR EXCLUSIVA

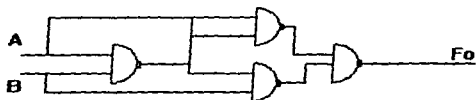


FIG. 2.7.- DIAGRAMA LÓGICO

Tabla de verdad

A	B	Fo
0	0	0
0	1	1
1	0	1
1	1	0

* Realizar también el alambrado con los CI, como resultaría físicamente, como en el ejemplo anterior.

$$C) - [(CD)' + A] + A + CD + AB$$

$$(C'D'+A)+A+CD+AB = CDA'+A+CD+AB = CD(A+1)+A(1+B) = CD+A$$

Alambrado físico y diagrama lógico:

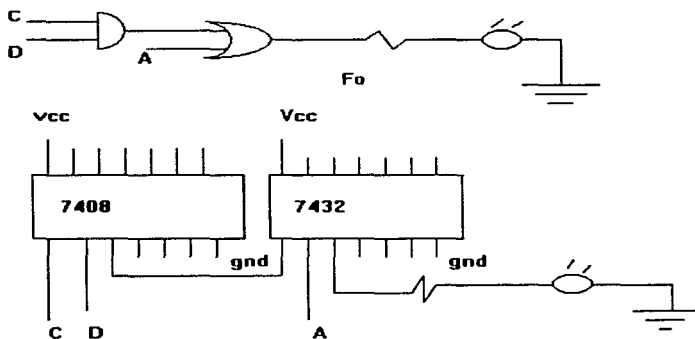


FIG. 2.8.- DIAGRAMA Y ALAMBRADO

TABLA DE VERDAD

ΔCD	F_2
000	0
001	0
010	0
011	1
100	1
101	1
110	1
111	1

d).- $F = [(A'B')'(AB)]' = (A'' + B''')' + (A' + B')' = A'B' + AB = A \otimes B$ NOR EXCLUSIVA

TABLA DE VERDAD

ΔB	F_2
00	1
01	0
10	0
11	1

CAPITULO III

TÉCNICAS DE MINIMIZACIÓN DE CIRCUITOS DIGITALES

OBJETIVO:

Conocer las técnicas básicas de minimización de funciones lógicas, (ec. booleanas).

CONTENIDO :

- 3.1.- Conceptos tradicionales.
- 3.2.- Teoremas de reducción.
- 3.3.- Minimización para una función de salida.
 - 3.3.1.- Mapas de Karnaugh.
 - 3.3.2.- Método de Quine Mc Cluskey.
- 3.4.- Minimización para varias funciones de salida.
 - 3.4.1.- Método de Quine Mc Cluskey.
 - 3.4.2.- Implementación de funciones booleanas empleando compuertas NAND y NOR.

INTRODUCCIÓN

En el capítulo anterior conocimos los teoremas fundamentales del álgebra Booleana, el siguiente paso es utilizar dichos teoremas para el análisis y realización de circuitos combinacionales.

Para obtener con mayor facilidad y/o utilizar menor número de circuitos, se emplean tres métodos básicamente:

- a) Álgebra Booleana
- b) Mapas o Planos
- c) Tabular

Lo importante de estas técnicas es el diseño y en base a esto su simplificación que dependerá de la experiencia que se obtenga de su manejo.

Cada uno de estos métodos pueden ser confiables dependiendo de la habilidad del usuario, del enfoque que se tenga y/o el número de variables utilizadas.

**3.1.-CONCEPTOS TRADICIONALES Y
3.2.- TEOREMAS DE REDUCCIÓN.**

Sea "Z" LA FUNCIÓN DE SALIDA:

$$Z = XY(X' + YW)'$$

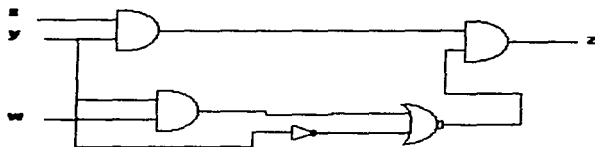


FIG. 3.1.1.-Diagrama de la Funcion de salida

Simplificando:

$$Z = XYW'$$



FIG. 3.1.2.-Diagrama simplificado

La complejidad de la misma depende del método utilizado, por lo que se pueden encontrar formas de una expresión que realicen la misma función.

Veamos la siguiente función:

$$Z = ABC + ABC' + AB'C$$

simplificando:

$$Z + A(B + C) \quad \text{ya que } x + x'y = x + y$$

Pero también podemos reducir los términos de la siguiente manera:

Agregamos ABC a la función original:

$$Z + ABC + ABC' + AB'C + ABC \quad \text{ya que } x+x = x$$

entonces:

$$Z = AB(C + C') + AC(B + B')$$

$$Z = AB + AC$$

$$Z = A(B + C) \text{ obteniéndose la misma expresión.}$$

3.3.-MINIMIZACIÓN PARA UNA FUNCIÓN DE SALIDA.

3.3.1.-MÉTODO DE MAPAS DE KARNAUGH

La representación lógica de un circuito se observa en las tablas de verdad, donde las posibles combinaciones de las variables y la función de salida obtenida se muestran para su implementación que será lo más reducida posible dependiendo del método aplicado, ya sea el de mapas de Karnaugh o el de Quine Mc Cluskey.

El método de mapas de Karnaugh es una representación de un diagrama de cuadros en el cual se expresa una función.

MAPA DE DOS VARIABLES

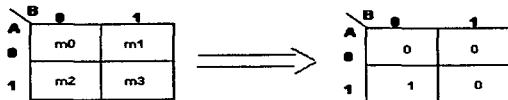
En este mapa únicamente encontramos la combinación de 2 variables ($2 = 4$ miniterminos), por lo que se representan con 4 cuadros, uno por cada minitermino.

		B	
		0	1
A	0	$A'B'$ 0	$A'B$ 1
	1	AB' 2	AB 3

TABLA DE VERDAD

	AB	FS
<i>m0</i>	00	0
<i>m1</i>	01	0
<i>m2</i>	10	1
<i>m3</i>	11	0

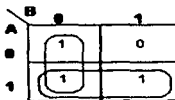
A cada cuadro se le asigna un valor correspondiente de "m" en cuestión con la función de salida cuando esta vale "1".



El encierro del "1" verifica el valor de la salida.

$$F_s = AB' = m_2$$

En el caso de que tuviese el siguiente mapa:



Se hacen los encierros correspondientes de dos (1s) para determinar la función de salida, ya sea vertical u horizontal.

Después se determinan que variables se anulan, al verse involucradas tanto la negación como la variable original, obteniéndose la siguiente función de salida por medio de la Σ de minterminos, tenemos;

$$F_s = m_0 + m_2 + m_3$$

sustituyendo:

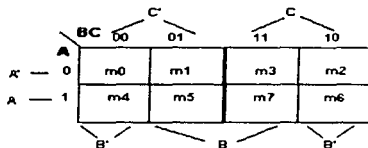
$$F_s = A'B' + AB' + AB \quad \text{ya que: } x + x = x$$

entonces:

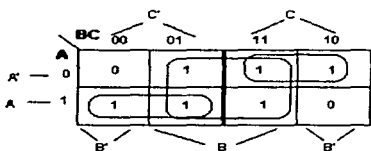
$$F_s = B' + A$$

MAPAS DE TRES VARIABLES.

Cada una de las variables en 4 cuadros con valor "1" y en 4 cuadros con valor "0" aparecen en el diagrama, lo cual nos servirá para identificar en donde se localizan los encierros de "1s" y anular las variables cuando estén involucradas en su forma afirmativa y negada a la vez.



Los encierros se hacen de 1 de 2 y de 4 (1s), según convenga, se determina su posición y la intersección de las variables, se obtienen los términos correspondientes a cada encierro, como se observa a continuación.



El 1er. encierro superior de 2 (1s), esta contenido dentro de A' como en C, pero se encuentra tanto en B' como en una parte de B, por lo que la variable B se anula al obtener la función; por lo que su valor será de ; $A' C$.

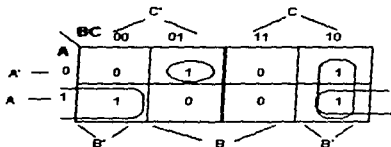
El segundo encierro de 4 (1s) abarca tanto a A' como a A y a C' como a C, pero queda incluido solo en B, por lo que su valor será; B.

El tercer encierro queda dentro de A y dentro de C pero abarca tanto a B' como a B, por lo que valdrá: AC' .

Al final sumando los tres encierros la función de salida quedara:

$$F_s = A'C + B + AC'$$

En el caso de que se tuviese el siguiente mapa:



Cuando los (1s) se encuentran en las esquinas opuestas se pueden agrupar en un solo encierro, entonces el encierro de (1) esta contenido dentro de A' , C' y B , por lo que su valor será de: $A'C'B$.

EL encierro de dos (1s) vertical, esta contenido dentro de C' y B quedando anulada la variable A , por lo que será: BC'

El encierro de los (1s) esquinados, esta contenido dentro de A y B' , anulando la variable C , por lo que queda: AC' .

Sumando los tres encierros tenemos:

$$F_s = A'C'B + BC' + AC'$$

MAPAS DE 4 VARIABLES

La minimización de 4 variables en un mapa es similar a las anteriores, nada más que en la combinación de dichas variables debe tomarse en cuenta que:

- a).- Un cuadro representa un minitermino el cual involucra a las 4 variables.
- b).- Dos cuadros en un encierro representan un término de dos variables.
- c).- 4 cuadros adyacentes representan 2 literales en un término .
- d).- 8 cuadros en un encierro representan una literal.
- e).- 16 cuadros con (1s) forman la salida igual a un nivel alto.

Si se tiene la siguiente función :

$$F_s = \{\sum m_0, m_2, m_3, m_5, m_7, m_8, m_{10}, m_{11}, m_{13}, m_{14}, m_{15}\}$$

		CD			
		00	01	11	10
AB	00	1	1	0	1
	01	0	1	1	0
	11	0	1	1	1
	10	1	0	1	1

Diagrama de Karnaugh para la función F_s con encierros etiquetados:

- a**: Encierro de los dos '1' en la fila AB=00.
- b**: Encierro de los '1' en las columnas CD=00 y CD=10.
- c**: Encierro de los '1' en las columnas CD=01 y CD=11.
- d**: Encierro de los '1' en las columnas CD=11 y CD=10.

Realizando los encierros y sumándolos nos queda:

$$F_s = BD + AC + A'C'D + A'B'D' + AB'D'$$

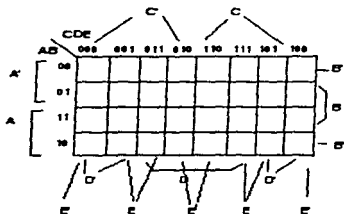
simplificando con álgebra Booleana:

$$F_s = BD + AC + A'C'D + B'D$$

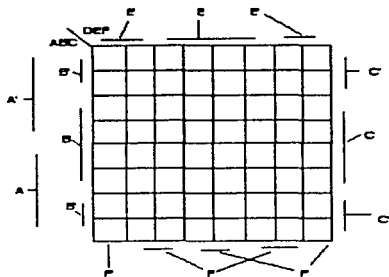
MAPAS DE 5 VARIABLES Y MAS

Estos mapas por el numero de variables que utiliza resulta ser menos practico y mas complejo, ya que aumentan las variables, el numero de cuadros también aumentan, ya que cada cuadro representa un mintermino de la función, pero su desarrollo es equivalente a los anteriores.

Para 5 variables:



Para 6 variables



USO DE LAS CONDICIONES DE NO IMPORTA

Hay ocasiones en que se utilizan circuitos en los cuales existen algunas combinaciones de las variables que nunca serán ocupadas.

Estas combinaciones de no importa que tengan salida ya que no deberán ocurrir, es por esto que se les denomina condiciones de no importa (*). Dada esta condición podemos utilizarla con los métodos de minimización de variables para llegar a la función de salida mas simple.

En los cuadros de los mapas se utiliza (*) para designar una condición de no importa para diferenciarla de los "0s" y los "1s".

Cuando se encuentre un encierro por realizar que tenga 3 (1s) por ejemplo y adyacente a estos, se encuentre un (*), se puede tomar como otro "1" para que el encierro se complete en el mayor numero de (1s) dentro del encierro, para poder simplificar aun mas la función de salida, sin alterar el diseño.

$F_s = (\sum 3,5,6,7,9,13)$ con condiciones de no importa:

$* = (\sum 2,4,8,12,14)$

tenemos:

CD \ AB	00	01	11	10
00	0	*	1	0
01	*	1	1	0
11	*	1	0	*
10	*	1	0	0

Utilizando algunas condiciones de no importa, se reduce el numero de encierros, quedando la función simplificada:

$$F_s = AC' + A'D$$

Si no tomáramos las condiciones de no importa, nuestro mapa quedaría:

	CD	00	01	11	10
AB	00	0	*	1	0
	01	*	1	1	0
	11	*	1	0	*
	10	*	1	0	0

Dándonos un mayor numero de compuertas.

$$F_s = AC'D + A'BD + A'CD$$

3.3.2.-MÉTODO DE QUINE MC-CLUSKEY

Este método es conveniente para encontrar la reducción de la función de salida, para cuando se tiene un mayor numero de variables.

Pensemos en la siguiente función de salida cuya reducción la encontraremos por el método de Quine Mc-Cluskey.

$$F_s = \sum m\{0,2,3,5,7,9,10,13,14,15\}$$

Realizaremos una tabla de 3 columnas en las cuales representaremos con grupos de miniterminos, por el numero de (1s) contenidos en estos, en otra columna la comparación entre cada uno de los términos de un grupo con los que difieren en un bit solamente para cada uno de los términos de los grupos siguientes, y en la ultima columna colocaremos la siguiente comparación de los grupos tomados en la columna b.

Entonces en la columna (a) ponemos la agrupación con los términos que no tengan (1s), la segunda agrupación con los términos que contengan un (1), el tercer grupo será de los miniterminos que tengan 2 (1s) y así sucesivamente hasta terminar con los números de la sumatoria de miniterminos.

COLUMNA a)

<i>m0</i>	0000	1
<i>m2</i>	0010	11
<i>m3</i>	0011	
<i>m5</i>	0101	
<i>m9</i>	1001	111
<i>m10</i>	1010	
<i>m7</i>	0111	11
<i>m13</i>	1101	
<i>m14</i>	1110	
<i>m15</i>	1111	1

Ahora bien, en la columna (b) hacemos la comparación, primero del término "*m0*" con los términos de los grupos siguientes, siempre y cuando difieran de un bit únicamente, luego formaremos otro grupo de comparación con el número siguiente del grupo 11 de la columna (a) de manera similar, con la tercera sección de la misma hacia abajo, los términos que difieran en más de un bit no comparan, cuando se encuentran dos números iguales en sus posiciones menos en un bit, ese bit se quita colocando un guión en su lugar.

Columna (a)	Columna (b)	Columna (c)
<i>m0</i> 0000	<i>m0,m2</i> 00-0	<i>m5,m13,m15</i> 1-1
<i>m2</i> 0010	<i>m2,m3</i> 001	<i>m5,m7,m13,m15</i> 1-1
<i>m3</i> 0011	<i>m2,m10</i> 010	
<i>m5</i> 0101	<i>m3,m7</i> 0-11	
<i>m9</i> 1001	<i>m5,m7</i> 01-1	
<i>m10</i> 1010	<i>m5,m13</i> -101	
<i>m7</i> 0111	<i>m9,m14</i> 101	
<i>m13</i> 1101	<i>m10,m14</i> 1-10	
<i>m14</i> 1110	<i>m7,m15</i> -111	
<i>m15</i> 1111	<i>m13,m15</i> 11-1	<i>m14,m15</i> 111-1

Entonces en la columna (c), de manera similar a lo anterior verificamos cada uno de los números del 1er. grupo con los de abajo en la columna (b) que difieren en una literal, y lo anotamos en otra lista de secciones en la última columna.

El término que resulto de la comparación de m_0 , m_2 , ya no compara, por lo que se expresa así:

El "0" indica que la variable es negada, el "1" que se encuentra en su forma original y el (-) que se encuentra anulada, entonces dependiendo de la posición en la que se encuentren las literales de ABCD será el valor de la variable a utilizar.

$$m_0, m_2, \dots A'B'C' = m_0 + m_2$$

Ahora de la columna (c)

ABCD

-1-1

=BD

-1-1

Quedando:

$$F_s = A'B'C' + A'B'C + B'CD' + A'CD + AC'D + ACD' + ABC + BD$$

Simplificando:

$$F_s = A'B' + B'CD' + A'D + ACD' + ABC + BD$$

Existe otra manera de resolver el método de quine Mc Cluskey, utilizando ahora la sustracción de números decimales en lugar de comparar los binarios.

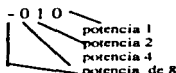
Realicemos la simplificación de la misma función anterior:

$$F_s = m(0,2,3,5,9,10,7,13,14,15)$$

Similarmente a lo anterior, en la columna (a) ordenamos en grupos los términos dependiendo del No. de (1s) involucrados en el número binario por ahora localizados como números decimales:

(a)	(b)	(c)
0	0,2 (2)	5,7,13,15,(2,8)-1-1
2	2,3(1)	5,13,7,15(2,8) -1-1
3	2,10 (8)	
5	3,7 (4)	
9	5,7 (2)	
10	5,13 (8)	
7	9,13 (4)	
13	10,14 (4)	
14	7,15 (8)	
15	13,15 (2)	
	14,15 (1)	

En la columna (b) realizamos la sustracción de los números decimales de un grupo con el siguiente abajo, formando secciones que representen a el resultado, para realizar la sustracción, el numero que se toma para sustraerlo de otro grupo , deberá ser menor que el minuendo en potencias de 2 encerrándose el resultado entre paréntesis, obteniéndose para cada bit que difiera en la comparación en posición del bit que se convierte en guión, esto es que para tomar la comparación 2, 10 (8), indica que 2 en binario es 0010 y 10 en binario es 1010 y el (8) indica la posición del guión en donde la variable no compara, quedando:



posición que no compara entre los números 2 y 10 en binario.

Por ultimo se comparan únicamente los términos con el mismo numero entre paréntesis, estos números difieren en una potencia de 2, de los números del grupo siguiente, además de ser mayores a los primeros, entonces en la columna ⊕ se ponen estos términos que constaran de 4 dígitos binarios, dando como resultado:

La suma de términos que no comparan mas la comparación de la columna (c).

		ABCD
0,2(2)	0000,0010(2)	00-0 -A'B'D'
2,3(1)	0010,0011(1)	001- -A'B'C
2,10(8)	0010, 1010 (8)	-010 -B'CD'
3,7(4)	0011,0111 (4)	0-11 -A'CD
9,13(4)	1001,1101 (4)	1-01 -AC'D
10,14(4)	1010,1110(4)	1-10 -AC'D'
14,15(1)	1110,1111(1)	111- -ABC
5,7,13,14(2,8)	0101,0111,1101,1111 (2,8)	-1-1 -BD

FS = A'B'D' + A'B'C + B'CD' + A'CD + ACD' + ABC + BD

3.4.-MINIMIZACIÓN PARA VARIAS FUNCIONES DE SALIDA

Hay ocasiones en que para ciertas combinaciones de n variables, podemos obtener mas de una función de salida.

Para tal efecto utilizamos cualquiera de los métodos ya señalados alternando la operación con todas las funciones o por separado para cada una de ellas.

Si tenemos :

$$F1 = \sum m \{3,4,6,7,8,10,12,13,15\}$$

$$F2 = \sum m \{0,1,2,4,5,9,11,12,14,15\}$$

$$F3 = \sum m \{1,3,5,6,7,10,11\}$$

Utilizando álgebra Booleana:

$$F1 = BCD + A'BC + BC'D' + AB'D'' + ABC' + A'CD$$

$$F2 = A'B'C' + A'BD' + ABC + AB'D$$

$$F3 = A'B'D + A'BC + AB'(C'D' + CD)$$

CD \ AB	00	01	11	10
00	0	0	1	0
01	1	0	1	1
11	1	1	1	0
10	1	0	0	1

$$F1 = BC'D' + ABD + A'CD + AC'B'$$

CD \ AB	00	01	11	10
00	1	1	0	1
01	1	0	0	1
11	1	0	1	1
10	0	0	0	1

$$F2 = A'B'C' + BD' + CD' + ABC$$

CD \ AB	00	01	11	10
00	0	1	1	0
01	0	0	1	1
11	0	0	0	0
10	1	0	0	1

$$F3 = A'B'D + A'BC + AB'D'$$

Para realizar la implementación de las 3 funciones de salida, buscamos los términos comunes en las operaciones algebraicas para simplificar el número de circuitos empleados para cada función por separado.

Así el término $A'B'$ que se emplea tanto en $F2$ como en $F3$, se desarrolla una vez para las dos funciones.

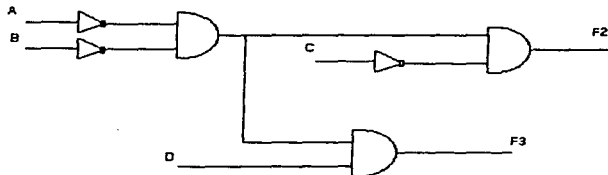


FIG.3.4.1.- Diagrama de simplificación.

3.4.1.-POR EL MÉTODO DE QUINE MC-CLUSKEY:

Para $F1$:

4	(a)	0100	(b)	4,6	01-0
8		<u>1000</u>		4,12	-1-
3		0011		8,10	10-1
6		0110		8,12	<u>1-00</u>
10		1010		3,7	0-11
12		<u>1100</u>		12,13	<u>110-</u>
7		0111		7,15	-111
13		<u>1101</u>			
15		<u>1111</u>			

$$F1 = A'BD' + BC'D + AB'D' + AC'D + A'CD + A'BC + ABC' + BCD + ABD'$$

Para F2 :

(a)	(b)	(c)
0	0,1 (1)	0,2,4,6 0-1
1	0,2 (2)	0,4,2,6 0-0
2	0,4 (4)	4,6,12,14 -1-0
4	1,9	1,12,6,14 -1-0
6	2,6	
9	4,6 (2)	
12	4,12 (8)	
11	6,14 (8)	
14	9,11 (2)	
15	12,14 (2)	
	11,15 (4)	
	14,15 (1)	

$$F2 = A'D' + BD' + A'B'C' + B'C'D + AB'D + ACD + ABC$$

Para F3 :

(a)	(b)	(c)
1	1,3 (2)	
2	3,7 (4)	
3	3,11 (8)	
6	6,7 (1)	
7		
11		

$$F3 = A'B'D + A'CD + B'CD + A'BC + AB'C'D'$$

3.4.2.-IMPLEMENTACIÓN DE FUNCIONES BOOLEANAS EMPLEANDO COMPUERTAS NAND Y NOR.

Después de haber acordado que método es el más conveniente para simplificar una función determinada por álgebra booleana, identificamos con que tipo de circuitos contamos o decimos que tan compacto queremos su implementación, para esto empleamos las compuertas básicas o bien las cambiamos por su equivalente en compuertas NAND y NOR.

Si tenemos :

$$F_s = \sum m \{ 0,3,5,6,7 \}$$

BC	00	01	11	10
A				
0	1	0	1	0
1	0	1	1	1

$$F_s = A'B'C' + BC + AC + AB$$

Negando dos veces a la función:

$$F_s'' = \{ (A'B'C')'(BC)'(AC)'(AB)' \}$$

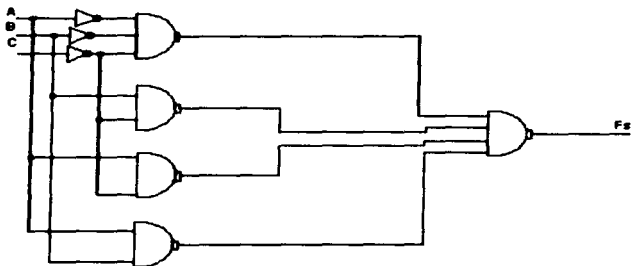


FIG.3.4.2.1.- Diagrama lógico

Entonces para obtener la implementación con compuertas NAND de una suma de productos, negamos 2 veces a la función y por medio del Álgebra Booleana obtenemos el diagrama anterior.

Implementación con NOR

Ahora debemos encontrar la función de salida en productos de suma, y después complementar la función para obtenerla con compuertas NOR únicamente:

$$\text{Si } F_s = \sum m \{ 1,2,4,5,6,7 \}$$

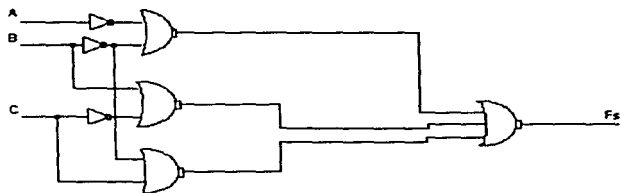


FIG.3.4.2.2

Entonces para lograr la implementación en compuertas NOR es necesario sacar la función de salida en productos de sumas.

CAPITULO IV

CIRCUITOS COMBINACIONALES

OBJETIVO:

Comprender y utilizar los elementos básicos de las técnicas de análisis y síntesis de circuitos combinacionales.

CONTENIDO

- 1).- Circuitos para control de tareas y diagramas de tiempo
- 2).- Sumadores
- 3).- Restador
- 4).- Decodificadores
- 5).- Multiplexores
- 6).- Comparadores
- 7).- Verificadores de paridad

INTRODUCCIÓN

En los capítulos anteriores mencionamos la manipulación de señales eléctricas para obtener una operación específica determinada por medio del álgebra booleana y métodos de reducción de circuitos lógicos.

Al realizar este tipo de transformaciones o procesamiento de información nos damos cuenta que los circuitos obtenidos dan como resultado una función de salida la cual depende totalmente de las diversas combinaciones que hagamos en las variables de entrada según nuestros requerimientos, con todo lo anterior vemos que los circuitos combinatoriales no dependen prácticamente de un estado de tiempo, esto es que en cierta forma un circuito implementado ni toma en cuenta una secuencia de tiempo para realizar las funciones requeridas.

En sí se trata de circuitos básicos ya que reciben la información por medio de códigos binarios y lo procesan de tal manera que la señal de salida sea la adecuada para una determinada secuencia de operación de los circuitos secuenciales, y su representación a la salida es también en forma binaria.

4.1.-CIRCUITOS DE CONTROL DE TAREAS Y DIAGRAMAS DE TIEMPO

Existen ciertos circuitos digitales que dependen de la retroalimentación que se les da, obteniéndose una salida determinada por el estado de tiempo en que se encuentre.

Dicha retroalimentación utilizara pulsos que duren poco, en donde una amplitud del pulso representara un 1 lógico y la ausencia de pulso representara un 0 lógico.

Esta clase de circuitos pueden ser utilizados para producir retardos de tiempo para controlar la temporización de circuitos digitales para realizar una sincronía entre circuitos, etc.

La entrada de un mono estable puede conectarse a la salida del dispositivo detector de errores como un comprobador de paridad.

Y la salida del mono estable se conecta a un solenoide mediante un timbre.

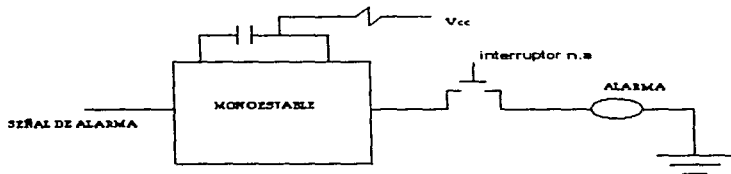


FIG: 4.1.1.- Monoestable

4.2.-SUMADORES

En los sistemas digitales se emplea una función muy específica la cual constituye el desarrollo de las operaciones aritméticas, dentro de las cuales constituye la base de las demás operaciones es la suma.

Anteriormente habíamos visto distintos tipos de sistemas numéricos los cuales son utilizados para que los circuitos lógicos funcionen y procesen debidamente la información, los sistemas de los complementos a 1 y complemento a 2 son los elementos básicos para realizar una suma de dos número binarios, la cual determina la realización de la función de un circuito sumador.

Los sumadores se clasifican en medio sumador y sumador completo.

El medio sumador realiza la adición de dos bits y el sumador completo lleva a cabo la suma de 3 bit, 2 de ellos significativos y el otro como un acarreo.

Las operaciones básicas de los sumadores son: $0 + 0 = 0$; $0 + 1 = 1$; $1 + 0 = 1$; $1 + 1 = 10$, cuando la suma resulta de dos bits el más alto se le denomina acarreo.

a).-MEDIO SUMADOR.

Se trata de un circuito que necesita de dos variables de entrada y dos salidas binarias; las salidas producen tanto el acarreo como la suma de las variables de entrada.

Retomando las operaciones básicas para los sumadores, podemos realizar la siguiente tabla de verdad para identificar la función del medio sumador.

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

→ bit mas significativo

↓
↓
↓

bit menos significativo

Entonces:

$$C = AB \quad \text{y} \quad S = A'B + A'B'$$

Se puede implementar por la suma de productos o con los productos de suma, esto es con implementaciones de compuertas NAND o NOR o fundamentales.

b).-SUMADOR COMPLETO.

Como ya dijimos este circuito realiza la suma de 3 bits y se obtienen las dos salidas; el acarreo y la salida, dos de las variables de entrada son las que representan los bits más significativos para realizar la suma, y la otra variable determina la cuenta de posición más baja en que se encuentra el bit más significativo.

Su tabla de verdad es la siguiente:

A	B	C	Ca	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Por mapas de karnaugh

A \ BC	00	01	10	11
0	0	1	0	1
1	1	0	1	0

A \ BC	00	01	10	11
0	0	0	1	0
1	0	1	1	1

$$S = A'B'C + A'BC' + AB'C' + ABC \quad Ca = BC + AC + AB$$

También la implementación de un sumador completo puede realizarse ya sea con suma de productos o con productos de suma, según se requiera o se desee.

Existen otros 3 tipos de sumadores, el sumador binario, el sumador decimal y el sumador BCD.

1.- *Sumador Completo.*- Un sumador completo tiene una entrada de acarreo, para aceptar las salidas de acarreo de las etapas precedentes como indica la figura 4.2.1), la tabla de verdad es la indicada en la figura b), el sumador suma tres bits a la vez. Por tanto, si $A = 1$, $B = 1$, y $C_i = 0$, la suma $S = 0$ $1+1+0 = 0$ y hay salida de acarreo $C_o = 1$. El sumador completo puede realizarse mediante dos semisumadores y una puerta OR como la figura c):

a.- Dibújese el diagrama lógico de un sumador completo con puertas EX-R, AND, y OR.

b.- Indíquese los 1s y 0s a la salida y entrada de cada puerta cuando $A = C_i = 1$ y $B = 0$.

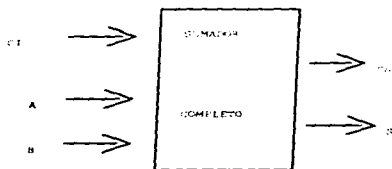


Fig. 4.2.1 sumador completo

ENTRADAS		SALIDAS	
A	B	C_i	C_o S
0	0	0	0 0
0	0	1	0 1
0	1	0	0 1
0	1	1	1 0
1	0	0	0 1
1	0	1	1 0
1	1	0	1 0
1	1	1	1 1

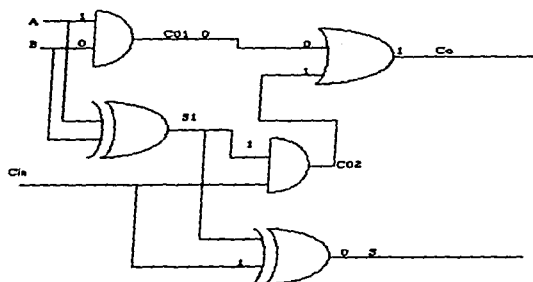
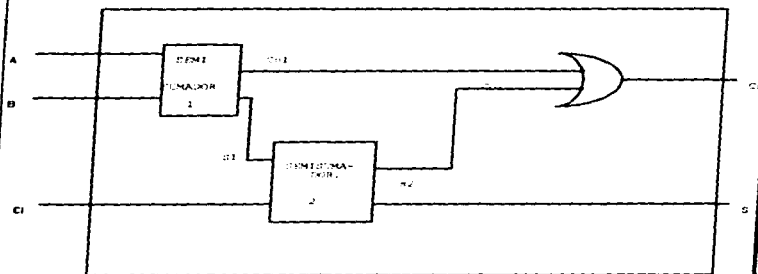


Fig. 4.2.2 Sumador Completo.

e).-SEMISUMADOR

El semisumador suministra la suma de 2 dígitos binarios y el arrastre (si existe), cada etapa de un sumador completo opera con un dígito de cada uno de un sumador completo puede operar con un dígito de cada uno de los 2 números que han de sumarse más al dígito de arrastre de la etapa anterior.

Los sumadores pueden diseñarse para trabajar en serie o en paralelo. Cuando se trata de sumadores en serie, la adición se realiza de forma secuencial, empezando con los bits menos significativos (LSD) y si el sumador es en paralelo se realiza la suma de todos los dígitos simultáneamente.

Los sumadores en paralelo efectúan la suma con mucha mayor rapidez que los sumadores serie, pero son más complicados y por consiguiente más caros.

d).-SUMADOR SERIE

En la suma serie, los dígitos de los números binarios que han de sumarse van siendo presentados al sumador de forma secuencial procedentes, por ejemplo; de una memoria de la calculadora, como se muestra en la figura (4.2.3).

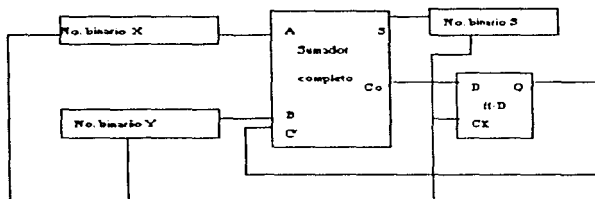


FIG.4.2.3 SUMADOR SERIE

Los números binarios X y Y , y la suma S se conservan en memorias denominadas registros de desplazamiento.

Cada dígito de los números X y Y es desplazado hacia el sumador de forma secuencial, empezando por el menos significativo y bajo el control, estos es que con la secuencia de los impulsos de reloj.

Si se utilizan los impulsos de un mismo reloj para controlar el suministro de datos al sumador, el registro de la suma y el funcionamiento del flip-flop D, el tiempo requerido para la suma, termina cuando se transfiere al terminal de arrastre C , al dígito de dicho arrastre.

Suponiendo que la salida Q del FFD¹ esta inicialmente en "0", al aplicar el impulso de reloj, las cifras LSD de los números binarios x , y , tienen acceso a las entradas A y B del sumador completo. La suma aparece en la salida del sumador y el dígito de arrastre es aplicado a la entrada del FFD.

¹ FFD son las siglas para designar a un Flip-Flop D (abreviatura)

Al siguiente pulso de reloj entran los bits siguientes en significación y se representan a las entradas A y B, la suma anterior se mueve a una posición en el registro.

Ahora bien, la suma de dos números binarios se puede obtener de dos maneras :

- 1) Suma en Serie
- 2) Suma en paralelo

En serie como ya vimos, se utiliza un solo circuito sumador completo y un acumulador para conservar el arrastre de salida generado.

El par de bits en A y B se transfieren en serie, uno a la vez a través del sumador completo para producir la suma a la salida.

El bit de arrastre de salida acumulado, de un par de bits, se usa como bit de arrastre de entrada para el siguiente par de bits.

En Paralelo se utilizan varios circuitos (sumadores completos), y los bits A - B se aplican simultáneamente. El bit de arrastre de salida se conecta con el bit de arrastre de entrada del sumador completo de la posición siguiente a la izquierda. Después de generarse los bits de arrastre, los bits de la suma saldrán de cada sumador completo.

e).-SUMADOR PARALELO.

Un sumador Paralelo es una función digital que produce una suma aritmética de dos números binarios en paralelo, esta se produce conectando varios sumadores completos en cascada con la salida de arrastre de un sumador completo conectado con el arrastre de entrada del siguiente sumador completo.

Con un sumador en paralelo puede obtenerse una respuesta casi instantánea, para obtener el resultado hay que utilizar otras unidades adicionales . El sumador en paralelo necesita un sumador completo por cada uno de los dígitos de los números que se suman .

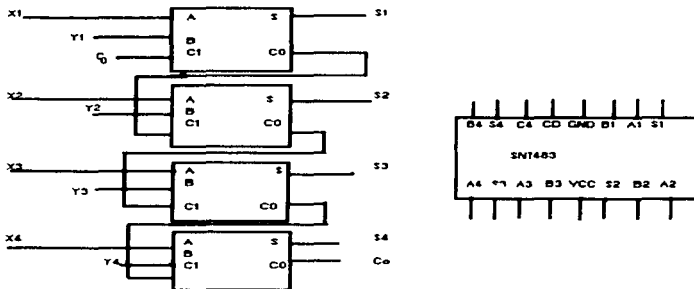


FIG: 4.2.4 Sumador Paralelo

Y podemos obtener la suma en paralelo con un solo CI, el SN7463 (Sumador Completo) de 2 bits, tiene 4 terminales para cada sumando y 4 terminales para cada bit de suma, 2 terminales para los arrastres entrada y salida.

Para un sumador Completo de 4 bits tenemos el circuito integrado de la serie TTL 74283.

Ejercicio:

1.- Sumar 46 y 22 realizando la adición en binario:

$$101110_2 + 10110_2 = 1000100_2 \quad 46_{10} + 22_{10} = 68_{10}$$

Las redes lógicas que permiten realizar la adición binaria están basadas en el semisumador como lo indica la fig. siguiente:

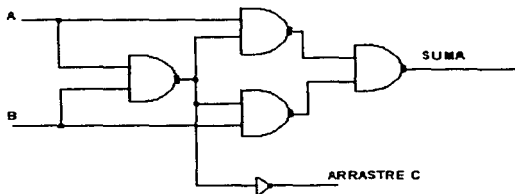


FIG. 4.2.5.- SEMISUMADOR

AB	SC
00	00
01	10
10	10
11	01

4.3.-RESTADORES

Cuando se quiere realizar la resta de 2 números binarios, únicamente hay que agregar al minuendo el complemento del número que se sustrae.

a).-Medio Restador

CIRCUITOS COMBINACIONALES

Este circuito combinacional sustrae 2 bits y realiza la diferencia, esto es que lo resta de $x-y$, para cuando $x \geq y$ sí:

0-0 = 0 1-1 = 0 1-0 = 1 el resultado es el bit de diferencia.

Para cuando $x < y$, tenemos 0-1 y se tomará un bit de la etapa más alta, al realizar esto se deberá agregar 2 al bit minuendo, así; 2-1 = 1
 Entonces su tabla de verdad será:

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

B informa a la siguiente etapa que se a tomado un "1"
 D (resta)

4.4.-DECODIFICADORES

Varios sistemas digitales y analógicos como los instrumentos de medición, los multímetros, necesitan que la salida de información que miden pueda ser visualizada por el usuario, para tal efecto podemos usar los decodificadores en la entrada de estos sistemas de visualización, ya que un decodificador binario convierte números binarios en caracteres.

En este tipo de circuitos observamos la manipulación de datos de los códigos alfanuméricos, además los decodificadores pueden ser utilizados en direccionamiento de memorias de otros dispositivos, su diagrama lógico puede verse a continuación, para un decodificador de 2 a 4.

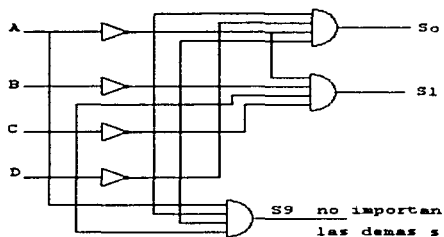


DIAGRAMA LOGICO DEL CIRCUITO 7442

FIG:4.4.1 Diagrama Lógico del 7442 (con compuertas)

Este circuito opera mediante la habilitación de un enable, esto quiere decir que solamente se decodificará en determinados momentos, la información.

Un decodificador convierte a 2^n salidas de n entradas, la salida que resulta con valor igual a "1" representa la habilitación del equivalente del código binario de entrada.

Así si habilitamos el minitérmino número 1 con sus valores correspondientes para cada una de sus entradas la salida obtenida será S1.

Para dos entradas:

A	B	S0	S1	S2	S3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	1	0	0
1	1	0	0	1	0

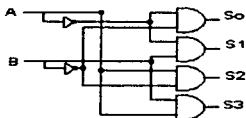


Fig. 4.4.2.- Decodificador de 2 a 4.

Pero también podemos introducir la información codificadas en BCD y obtenerla en decimal. Para esto necesitaremos 10 salidas cada una para identificar a un número decimal (0 al 9), la utilización del código BCD requiere de 4 bits por cada número a representar.

Entonces : 4 entradas - 10 salidas utilizadas , el decodificador que debemos diseñar será de 4- 16 líneas con 6 de ellas no utilizadas .

Haciendo uso del método de mapeo:

Las 6 salidas no utilizadas dependiendo del usuario se toman en cuenta o no como condiciones de no importa.

AB	CD			
	00	01	11	10
00	D0	D1	D3	D2
01	D4	D5	D7	D6
11	0	0	0	0
10	D8	D9	0	0

También podemos realizar el diseño de un decodificador de BCD a 7 segmentos, para que se visualicen los números equivalentes de BCD a decimal, es lo que se utiliza normalmente para visualizar las lecturas numéricas en instrumentos de medición.

Entonces necesitamos 7 salidas diferentes con cuatro entradas de datos para hacer prender los segmentos como lo muestra la figura siguiente.:

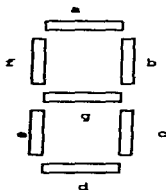


Fig. 4.4.3.- TII de 7 segmentos

Realizando la tabla de verdad :

ABCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1

los demás son condiciones de no importa

		00	01	11	10
00	CD	1	0	1	1
01		1	1	1	1
11		1	*	*	*
10		1	1	*	*

Realizando los mapas para cada segmento:

Para

$$\begin{aligned}
 a &= C'D' + B + A + C \\
 b &= B' + C'D' + CD \\
 c &= C' + D + BC \\
 d &= A + B'D' + CD' + B'C + BC'D \\
 e &= CD' + B'D' \\
 f &= A + C'D' + BC' + B'D' \\
 g &= A + BC' + B'C + CD'
 \end{aligned}$$

El decodificador quedará :

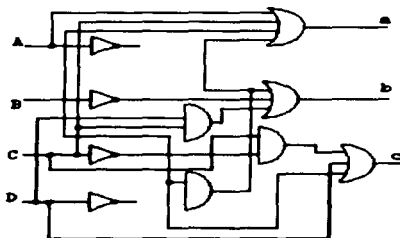


FIG. 4.4.2 Resultado del decodificador con compuertas

O bien podemos implementarlo con compuertas NAND y NOR

$$a'' = [(C'D)'] (B'A'C)'$$

O bien directamente con un CI 7446 o 7447 conectado directamente a cada uno de los segmentos.

Ejemplo:

Diseñar un circuito Sumador Completo con un decodificador .

$$Si \ S = A'B'C + A'BC' + AB'C' + ABC$$

$$Ca = AB + AC + CB$$

con un decodificador de 3x8 tenemos:

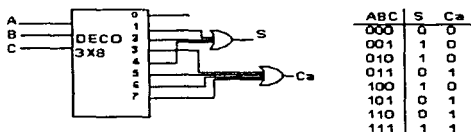


Fig. 4.4.3.- Sumador Completo con un deco de 3x8.

4.5.-MULTIPLEXORES

Pensemos que necesitamos enviar 3 mensajes distintos en serie por una sola línea de transmisión, en lugar de utilizar 3 líneas diferentes para transmitir los datos, utilizamos un conmutador (multiplexor), se reduce el tiempo para recibir los datos y por consiguiente la velocidad de transmisión, ya que depende dela selección de datos que se de para obtener la información de cada mensaje por separado.

Un multiplexor es un selector de datos porque selecciona una de muchas entradas a una sola salida.

Si tenemos que controlar el empleo de 4 bombas de agua, que dependiendo de la selección que tengamos, obtendremos la indicación de encendido o apagado de cada una de las bombas, entonces:

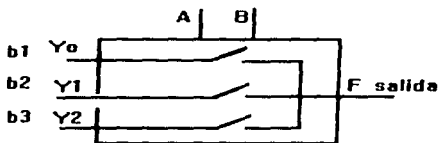


FIG: 4.5.1 Multiplexor

En este caso se trata de un MUX de 4 entradas, que son la señal de encendido o apagado de las bombas y 2 líneas de selección, las cuales determinan que información será guiada a la línea de salida.

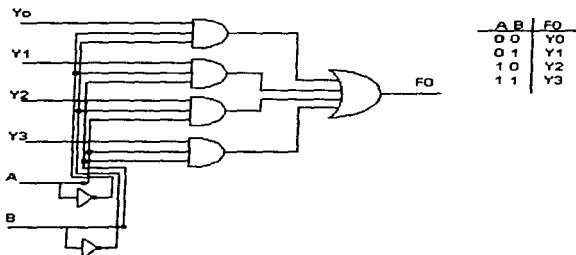


Fig.-4.5.2.- Diagrama de un MUX de 4 a 1 con 2 líneas de selección, alambrado con compuertas.

El tamaño del Mux a emplear se especifica por 2^n entradas con n líneas de selección a una sola salida.

Para extender un Mux con un mayor número de entradas, uniendo varios Mux y seleccionando una de 2 líneas de entrada, se emplea una entrada de control S (strobe) que sirve para seleccionar una de las dos entradas y un E (enable) para habilitar o desactivar el circuito.

Para implementar un Mux de 2^{n-1} a 1 de una función Booleana, procedemos así:

$$\text{Si } F(ABCD) = \{\sum 2,3,5,7,9,12,15\}$$

Donde A es la 1a de las n variables;
 BCD son las $n-1$ variables restantes.

Las $n-1$ variables serán las líneas de selección del MUX.

Se realiza una lista de entradas mostrando los minitérminos donde A es negada y donde es afirmada, para cuando la función de salida es 0 o 1.

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	...	I_{15}
A'	0	1	2	3	4	5	6		7
A	8	9	10	11	12	13	14		15

Se encierran los minitérminos para cuando la $F = 1$.

Si en una columna los dos minitérminos se encuentran encerrados se aplica a la entrada I_n correspondiente, un 1.

Si en una columna aparece un encierro se aplica a A' o A según corresponda en el renglón a la entrada I_n .

Si no aparecen encierros se aplica a la entrada un 0, así;

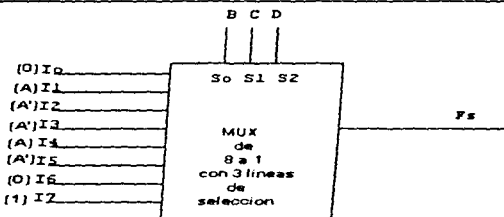


FIG: 4.5.2 Multiplexor (ejemplo)

Ejemplo:

Si $F_o = \{\Sigma 0, 2, 3, 5, 7\}$ alamburar con MUX.

ABC	F_o
000	1
001	0
010	1
011	1
100	0
101	1
110	0
111	1

	I_0	I_1	I_2	I_3
A'	0	1	2	3
A	4	5	6	7

Las líneas de selección serán B y C :

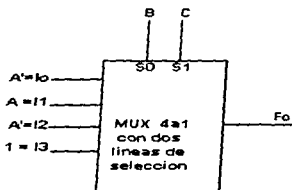


Fig.4.5.3.-Mux de 4 a 1

Con esto vemos que podemos alambrar cualquier función de salida por medio de los multiplexores.

4.6.-COMPARADORES

Los sistemas comparadores se dividen en dos clases :

- 1) Comparadores de identidad . Detectan cuando son idénticas, dos palabras
- 2) Comparadores de magnitud.- Detectan cuál de las dos palabras es mayor, son más complejos y por lo general más lentos.

Todos los comparadores se definen en términos binarios aunque pueden emplearse con el código BCD o cualquier otro código monótono que no cambie.

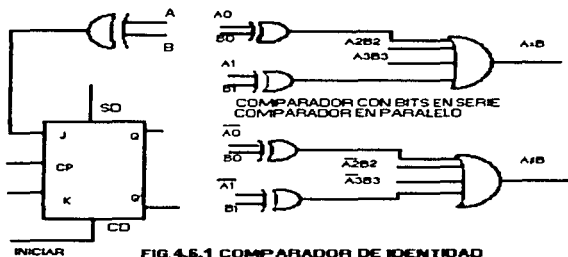
Un circuito OR exclusivo(XOR) y un FF forman el comparador de identidad de tipo serie de la fig.(4.6.1). El Flip flop debe restablecerse al principio.Tan pronto las entradas A y B sean idénticas, la salida de la compuerta XOR será baja y llevará el flip-flop al estado de restablecimiento.

Cuando $A \neq B$, el flip flop se establece (set) y permanece así hasta que inicia un nuevo ciclo, al borrar asíncronamente el flip flop. El estado Q después de que el último bit haya sido temporizado indicará el resultado de la comparación.

$$Q: A \neq B \text{ y } Q: A = B$$

Es obvio que la secuencia de bits no afecta a la comparación de identidad . La comparación de identidad en paralelo es más eficiente cuando se emplean 4 compuertas XOR con salidas hacia una compuerta NAND o NOR.

La configuración NAND es más rápida, pero necesita polaridades opuestas de los dos operandos. Ambos arreglos se muestran en la figura(4.6).



Comparación de Magnitud.- para bits en serie : 1er. bit menos significativo (LSB).

La comparación de magnitud se efectúa entre 3 posibles condiciones $a > b$, $A < B$ y $A = B$, aun cuando por lo general se codifica en 2 señales de salida.

El comparador de magnitud de tipo serie para el 1er. dígito menos significativo es más eficiente cuando se instrumenta con un multiplexor de 4 entradas dobles y un FF doble, o con una compuerta XOR, y un FF D con habilitación como se observa en la siguiente fig:

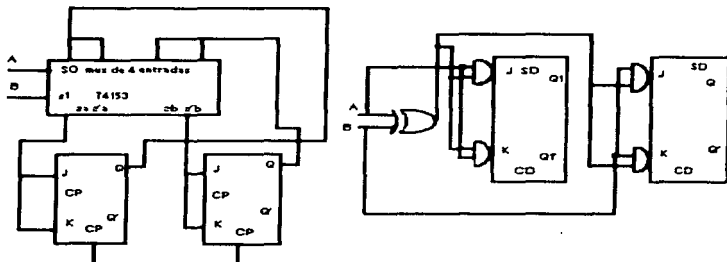


FIG. 4.6.1b REAREGLO

Si se supone la notación de activo alta, Q se establece mediante A^*B^* , mientras que no le afecta;

$$A^*B \text{ o } A^*B^*(A=B)$$

Q2 es establecida por $A \neq B$ y no le afecta $A=B$

Entonces si se comienza por restablecer ambos flip flop sus estados, después del pulso de reloj en el bit más significativo, indican el resultado de la comparación. Un pequeño reareglo del mismo circuito básico puede generar un grupo diferente de salidas, como se indica en fig. (4.6.1 b).

4.7.-VERIFICADORES DE PARIDAD

Dentro de la transmisión de datos en los sistemas digitales, entre dispositivos de entrada y salida, siempre existe la probabilidad de encontrar errores en la transmisión, aun cuando ya exista equipo moderno que trata de eliminarlos, es necesario determinar un método para detectarlos.

El método llamado de "paridad", emplea un bit de paridad, adicional que se agrega a un grupo de código con el que se hace la transmisión, ya sea un 0 o un 1, según la cantidad de unos existentes en el grupo. si la cantidad de "unos" en el grupo de código es par incluyendo el bit de paridad, entonces se le asigna un 1, y si la suma de los "unos" sin incluir el bit de paridad es par entonces se le agrega un 0, obteniéndose el método de paridad par.

El método de paridad impar, realizaría la misma secuencia pero la suma de "unos" en este caso deberá ser impar incluyendo el bit de paridad asignada.

Entonces para el código de grupo 1 1000100
bit de paridad asignado

Es impar la cantidad de "unos" localizados en el grupo de código y si:

0 1001001
bit de paridad asignado

El bit de paridad asignada es 0 para la cantidad de "1s" se a impar.

Detección de errores.

Cuando se transfieren datos digitales de un lugar a otro, hay probabilidad de error debido a fallas en los elementos, o tal ruido. Existen numerosas formas de manejar los errores, algunos sistemas registran el error y envían una orden de retransmisión de datos. En otros la retroalimentación puede ser imposible o extremadamente costosa. En estos casos, el equipo receptor debe no solamente ser capaz de detectar el error, sino también corregirlo.

La detección y corrección de errores esta relacionada con la transmisión de información redundante, esto es de datos repetidos. Esto último requiere bits adicionales de datos y por consiguiente, una disminución en la eficiencia total del sistema de transmisión. En sistemas de transmisión paralelos, se necesitan mas alambres transmisores y receptores, mientras que en sistemas de transmisión en serie, se utiliza más tiempo para transmitir la información redundante.

Todos estos métodos no eliminan por completo los errores de transmisión, aunque el incremento en el porcentaje de bits redundantes, o en la complejidad del sistema de detección de errores, o en algoritmos de conexión, disminuye la probabilidad de no detectar o corregir los errores.

A continuación se muestran otros métodos para reducir los errores:

a).-Generadores del bit de Paridad.

El método más simple y de uso en la eliminación de errores es agregar un bit, llamado bit de paridad, se escoge en forma tal que el número total de unos, en la palabra (contando al bit de paridad) sea non (en el sistema de paridad non) o par (en el sistema de paridad par). Por lo general se prefiere la paridad non, ya que asegura que el último bit sea "1" en cualquier palabra. El receptor se examina la paridad de la palabra. Si un sólo bit en la palabra fue cambiado, el detector indica una paridad errónea.

Sin embargo, si un número para de errores ocurre, este método simple no puede detectarlo. El bit de paridad sólo opera mientras tenga un sólo error.

En el generador de paridad serie de la fig. (4.7.1) se cambia un FF por cada "1" en la palabra y el estado del mismo se inserta al final como bit de paridad. En el estado del receptor, el comparador de paridad tiene un FF equivalente.

Su estado es interrogador después de que llegan los datos. Ambos circuitos se adaptan con facilidad al sistema de paridad par o non.

Para los sistemas en paralelo es necesario generar la suma módulo de dos de muchas entradas simultáneamente. Esto obliga a emplear un arreglo de circuitos XOR en cascada. Los circuitos 74189, 74280, 9348 y 8262, están diseñados para efectuar esta función, son verificadores o generadores de paridad de 8 a 12 entradas utilizadas en la detección de errores y en la aplicación de la corrección correspondiente con datos en paralelo.

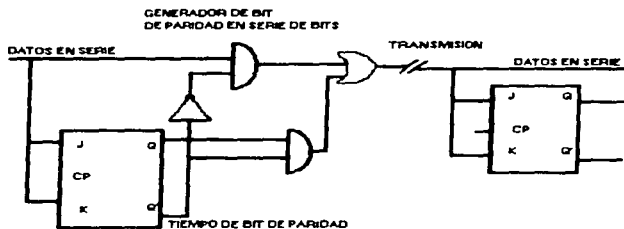


FIG.4.7.1 Generacion del bit de Paridad

Ejercicio:

1.- Realizar y comprobar en el laboratorio, el generador y verificador de paridad dado a continuación:

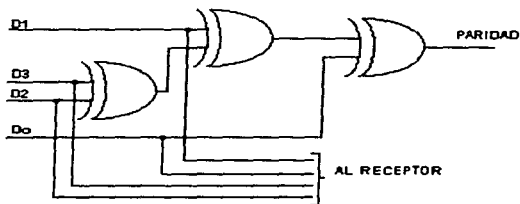


FIG. 4.7.2 VERIFICADOR DE PARIDAD

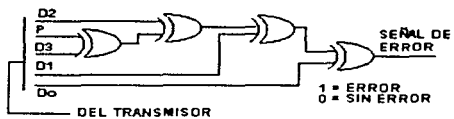


FIG. 4.7.2 VERIFICADOR DE PARIDAD

CAPITULO V

ESTUDIO DE FAMILIAS LÓGICAS

Objetivo:

Comprensión del funcionamiento y limitaciones de las familias lógicas y consideraciones del acoplamiento entre ellas.

CONTENIDO:

- 5.1.- Elementos básicos de los circuitos integrados*
- 5.2.- Circuitos integrados*
- 5.3.- Familias lógicas*
 - a).- Características*
 - b).- Tipos*
- 5.4.- TTL y sus características*
- 5.5.- ECL y sus características*
- 5.6.- MOS Y CMOS y sus características*

INTRODUCCIÓN

Para poder determinar con mayor precisión y seguridad que clase de circuito se debe utilizar para realizar una tarea determinada, se debe de tomar en cuenta los circuitos electrónicos básicos en cada familia lógica digital de CI y analizar su operación eléctrica.

Entonces los CI se clasifican no solamente por su operación lógica sino por la familia de circuitos lógicos específicos a la cual pertenecen , también podemos aclarar que toda base de diseño y principios de funcionamiento para cada una de las familias lógicas es una compuerta AND o una OR, mediante las cuales se realizan circuitos mas complejos.

Además los circuitos integrados según la familia de la que se trate se distinguen comúnmente por la designación numérica como : para TTL (5400 y/o 7400); para la ECL(10 000); para la CMOS (4000) a la que se le denomina serie de la familia en cuestión.

Podemos mencionar que la base de las familias son la RTL y DTL, ya que la DTL fué gradualmente reemplazada por la TTL, y la RTL fué familia comercial que se uso extensamente pero ya es obsoleta.

Ahora bien las familias TTL, ECL y CMOS, tienen un gran número de circuitos MSI y LSI, y la MOS se uso para principalmente para construir diseños con LSI.

Las familias TTL y ECL utilizan transistores bipolares, y la MOS y CMOS usan transistores unipolares MOSFET(transistor de efecto de campo semiconductor de óxido metálico).

5.1.-ELEMENTOS BÁSICOS DE LOS CIRCUITOS INTEGRADOS

Existen diferentes tipos de tecnología de CI:

*SSI (Integración a pequeña escala)

Estos dispositivos contienen máximo hasta doce elementos por cada ci, como el siguiente;

Familia TTL ECG 7400, ECG 74H00, ECG 74L500, ECG 74S00

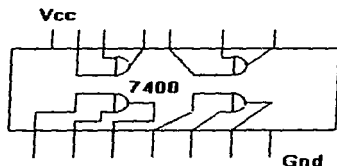


Fig.5.1.1.- 4 compuertas Nand de 2 entradas

*MSI (Integración a mediana escala) Pueden contener de 12 a 99 elementos por ci., como:

Familia TTL ECG 74L5295A

paralelo

4 bit bidireccional
shift registro con 3estados
de salida.

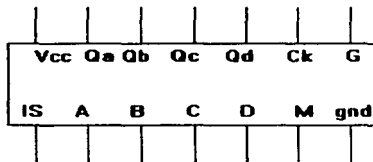


Fig. 5.1.2.-ECG74LS295A

*LSI (Integración a grande escala) de 100 a 9999 elementos por CI, como:

Familia CMOS MC14500B

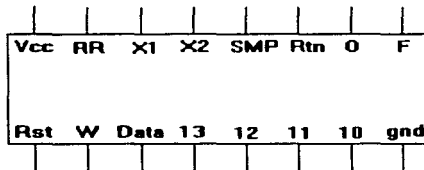


Fig. 5.1.3.-CMOS MC 14500B

*VLSI (Integración a escala muy grande 10000 a 99999 elementos contiene por cada CI)

*ULSI (Integración a escala ultra grande contiene de 100000 ó mas elementos por cada CI)

Ahora bien los CI no pueden manejarse con alta potencia ya que al elevar su temperatura limite provoca descomposturas ; por lo que no pueden manipular como componentes discretos, o elementos que se utilicen para grandes flujos de corriente y de voltaje.

La gran variedad de CI hace necesaria la comprensión de las diferentes familias lógicas, ya que difieren en componentes que forman su circuitería .

La Familia TTL y ECL tiene transistores bipolares como componentes principales. Las Familias PMOS , CMOS Y NMOS , transistores unipolares mosfet.

Los Parámetros más importantes para saber manejar CI son:

••Corriente, voltaje, retardo de propagación , factores de carga, requerimientos de potencia , inmunidad al ruido , suministro y consumo de corriente y velocidad.

- a).- VIH (min) - El nivel alto de entrada mínimo que se requiere para que funcione el circuito.
- VIL (max) - Nivel bajo de entrada máximo para que funcione el CI.
- VOH (min) - Voltaje de salida a nivel alto
- VOL (max) - Voltaje de salida a nivel bajo
- I_{IH} - Corriente de entrada a nivel alto
- I_{IL} - Corriente de entrada a nivel bajo
- I_{OH} - Corriente de salida a nivel alto
- I_{OL} - Corriente de salida a nivel bajo

Si consideramos el siguiente circuito:

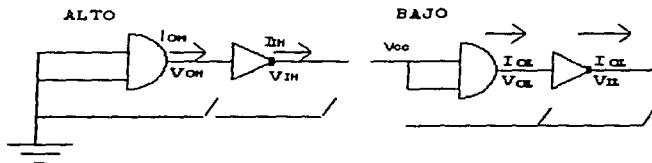


FIG. 5.1.4.-Diagrama con Parámetros de CI.

Para el ECG 7408 y 7404

$V_{cc} = 5v$	$V_{OH} = 2.4v$
$V_{IH} = 2v$	$V_{OL} = 0.4v$
$V_{IL} = 0.8v$	$I_{IH} = 40 \text{ amp}$
$I_{OH} = 0.4 \text{ amp}$	$I_{IL} = 1.6 \text{ amp}$
$I_{OL} = 16 \text{ amp}$	

Si a la salida de la compuerta and se tiene menos de 2.4 volts = V_{OH} , el circuito no aceptara este nivel como un "1" lógico .

Si a la entrada del inversor el voltaje que entra es menor de 2 volts (V_{IH}) no lo aceptará como un nivel alto.

En nivel bajo, si el voltaje de salida de la AND es mayor de $0.4v = V_{OL}$, el circuito lo tomará como un alto, así mismo si el voltaje a la entrada del inversor es mayor que $0.8v = V_{IL}$, será un "1" lógico.

Las corrientes que lleguen por el circuito son determinados por los voltajes que se manejen dentro del mismo .

b).- Todo circuito maneja señales lógicas las cuales tardan un tiempo determinado en recorrerlo, cuando un circuito pasa de un estado lógico a otro tarda un tiempo en obtener respuesta de salida (llamado *retardo de propagación t_p*)

t_{PHL} - cuando cambia de un estado bajo a un alto.

t_{PLH} -cuando cambia de un estado alto a un bajo

Los tiempos de retardo de propagación sirven para determinar la rapidez de los circuitos lógicos.

Entonces si queremos obtener rapidez en la salida de un circuito podemos escoger entre las diferentes series y familias.

Si tenemos 3 circuitos diferentes :

	7400	74LS00	74AS00
t_p	9ns	9.5ns	1.7ns

Entonces la que debemos usar es la 74AS00 por la velocidad de sus respuestas.

b').-FAN OUT (*Factor de carga*) Se le llama al número máximo de entradas lógicas que una salida puede aguantar para seguir trabajando eficazmente . El Factor de carga no debe ser encendido para que los voltajes de salida sean íntegros.

Para un Fan out de 8 se utilizan ocho entradas conectadas a la salida sin alterar el funcionamiento del circuito integrado.

c).- Márgenes de ruido .- Se hace necesario calcular los márgenes de ruido tanto en estado alto como en estado bajo.

Si el margen de ruido en estado alto es mayor que:

$$V_{NH} = V_{OH} (\text{min}) - V_{IH} (\text{min})$$

Puede alterar el voltaje llevándolo por debajo de 2 volts.

Si el margen de ruido en estado bajo es menor que :

$$V_{NL} = V_{IL} (\text{max}) - V_{OL} (\text{max})$$

d).- Requerimientos de potencia, conocido también como disipación de potencia ; los circuitos TTL requieren para poder funcionar cierta potencia eléctrica, la cual viene denominada en los manuales como V_{cc} , y para las familias MOS o CMOS viene indicada como V_{DD} , así:

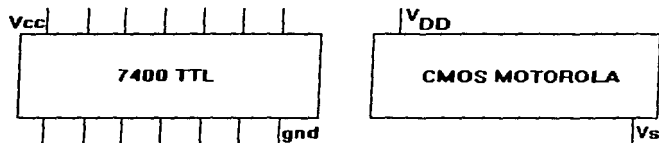


FIG. 5.1.5.- TTL Y CMOS

Para calcular el consumo promedio de potencia que utiliza un circuito, se utiliza la formula:

$$P_{cc} (\text{promedio}) = I_{cc} (\text{prom}) \times V_{cc}$$

$$\text{donde } I_{cc} (\text{prom}) = \frac{I_{ccH} + I_{ccL}}{2}$$

* I_{ccH} - cuando la salida del circuito esta en estado alto

I_{ccL} - cuando la salida del circuito esta en el estado bajo

c).- *Inmunidad al ruido* .- Es la capacidad que tiene un circuito para soportar el ruido sin afectar sus estados lógicos.

5.2.-CI INTEGRADOS

El encapsulado de CI depende de los requerimientos que tengamos, ya que existen circuitos especiales para la cantidad de circuitos internos que contengan, además para evitar los efectos del medio ambiente y la manera de emplearlo.

DIP- (Encapsulado de doble línea)

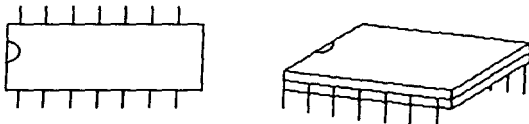


FIG. 5.2.1.-DIP

En esta clase de circuito observaremos la presencia de una muesca o punto , lo cual indica en que posición se encuentra el PIN (1) (a la izquierda).

- Para Montaje de superficie.- Este tipo de encapsulado no tiene una ranura que ayude a localizar la terminal uno, y puede soldarse directamente a la tarjeta de circuito impreso , además son más pequeños que los DIP.

- Encapsulado Plano de Cerámica.- Este tipo de encapsulado esta diseñado para ser inmune a los efectos de la humedad.

Estos encapsulados se emplean en aplicaciones donde el equipo funciona bajo condiciones extremas, y se les llama tipo militar como la serie TTL 5400 que trabaja dentro del rango de (-55 a 125°C) y son los más caros.

5.3.-FAMILIAS LÓGICAS

La lógica TTL fué la primer familia de lógica saturada en circuito integrado, y constituye el estándar de las familias posteriores, ya que ofrece una combinación velocidad, consumo de potencia , fuente de salida y capacidades de observación aconsejables, para la mayoría de aplicaciones.

La familia CMOS, es muy popular, al igual que la TTL , bajan extremadamente su consumo de potencia, tiene alta inmunidad al ruido y pueden funcionar con una fuente de alimentación regulada y económica, además generan menor ruido que la TTL, pero son específicamente sensibles a las descargas estáticas y tensiones transitorias.

La TTL generalmente se utiliza para operar a alta velocidad.

La Familia ECL evita la saturación de los transistores y aumenta su velocidad al conmutar de un estado a otro y opera bajo el modelo CML (lógica en modo de corriente). La Familia TTL opera con transistores en saturación por lo que se limita su velocidad de conmutación , por el retardo en el tiempo de almacenamiento del transistor cuando se satura.

Los circuitos SSI y MSI son requeridos en la familia TTL, en la cuál el circuito en el que se basa es la compuerta NAND.

La salida de un TTL tiene acción de consumo de corriente de estado bajo, ya que recibe corriente del circuito que manejará debido a que lleva al voltaje de salida a un nivel bajo, y se comporta como un estado alto cuando proporciona una corriente de entrada I_{II} (corriente de fuga en polarización inversa), (con valor \pm de $10\mu A$).

El Tipo Tótem conserva baja la disipación de potencia en estado alto, tiene una constante de tiempo corta, para poder manejar cargas capacitivas, pero al pasar del estado bajo a alto, consume una corriente alta ($\pm 40\text{mA}$) de la fuente de alimentación en un periodo de nanosegundos.

Ahora bien, la serie 74 TTL puede manejarse con 4.75 y 5.25 de alimentación a una temperatura entre 0°C y 70°C y es menos costosa, la serie 54 acepta de 4.5 a 5.5v de alimentación a una temperatura de 55° a 125°C y es más cara porque tiene dichas variaciones de temperatura y voltaje.

El nivel máximo que se le puede aplicar a un TTL para que este no se deteriore es de 5.5v y el mínimo es de -0.5v

Un encapsulado TTL de 4 compuertas NAND consume en promedio una potencia = 40 mW por lo que se considera que cada compuerta NAND consume 10mW.

Para saber el retardo de propagación de una NAND TTL standar, sería:

$$t_{pLH} = 11\text{ns} \text{ y } t_{pHL} = 7\text{ns}, \text{ entonces } t_{pd}(\text{prom}) = (t_{pLH} + t_{pHL})/2 = 9\text{ns}$$

Ejemplo: Para determinar la máxima disipación de potencia localizamos los valores de $I_{ccH} = 22$, I_{ccL} , $V_{cc} = 5\text{v}$, entonces:

* Al consultar la hoja de especificaciones para el CI OR cuádruple de 2 entradas, determinar la máxima disipación de potencia en promedio y el máximo retardo de propagación de unas cuatro compuertas.

$$P_{D \text{ prom}} = \frac{(I_{ccH} + I_{ccL})}{2} V_{cc \text{ max}} = \frac{(22\text{mA} + 38\text{mA})}{2} V_{cc \text{ max}}$$

$$P_{D \text{ prom}} = \frac{(60\text{mA})}{2} (5.25\text{v}) = 157.50 \text{ mW}$$

Dividiendo este valor entre 4, porque en el CI existen 4 compuertas:

Para cada compuerta :

$$P_D \text{ prom} = 157.50 \text{mw} / 4 ;$$

Entonces cada compuerta disipara en los peores casos de uso , 39.3mw y basándonos en lo anterior aseguramos un buen funcionamiento del circuito.

El retardo promedio de propagación máximo será:

$$\text{tpd (prom)} = \frac{\text{tp}_{LH} + \text{tp}_{HL}}{2} = \frac{15 + 22}{2} = \frac{37}{2} = 18.5$$

La Serie 74L. (Es de bajo consumo de Potencia), utiliza más resistencias, las cuales reducen la potencia pero tienen mayores retardos de propagación de la señal.

Su disipación de potencia es menor que cualquier otra serie.

Ejemplo.- En la hoja de especificaciones buscar una compuerta y poner su disipación de potencia y su retardo de propagación.

Compuerta OR exclusiva cuádruple de 2 entradas 7486

$$\begin{array}{lll} \text{ICCH} = 75 \text{ ma} & \text{ICCL} = 75 \text{ma} & \text{Vcc} = 5.25\text{v} \\ \text{TPLH} = 10.5 \text{ ns} & \text{tPHL} = 10 \text{ ns} & \end{array}$$

Entonces el máximo consumo de potencia para las 4 compuertas en el ci es :

$$P_{\text{Dprom}} = \frac{(75 \text{ma} + 75 \text{ma})(5.25 \text{v})}{2} = 393.75 \text{ mw}$$

y el retardo de propagación :

$$tPDprom = \frac{(10.5 + 10.)\eta s}{2} = 20.5 = 10.5\eta s$$

La Serie 74H (denominada como de Alta Velocidad).- Esta serie tiene una mayor velocidad de propagación pero tiene un mayor consumo de potencia .

Ejemplo.- Buscar un CI y poner su disipación de potencia y el retardo de propagación :

El CI 74H05 Inversor Hex (6)

$$I_{ccH} = 26mA, \quad I_{ccL} = 58mA, \quad V_{cc} = 5.25v, \quad t_{pLH} = 22\eta s, \quad t_{pHL} = 18\eta s$$

El máximo consumo de potencia será para los 6 inversores en el encapsulado

$$PDprom = \frac{(26mA + 58mA)}{2} (5.25v) = 141.75 \text{ mw}$$

Y el retardo de propagación será:

$$tpd_{prom} = \frac{22\eta s + 18\eta s}{2} = 20\eta s$$

Comparando con 7405:

$$PDprom = \frac{(12+33)}{2} (5.25) = 118.12 \text{ mw} \quad \text{menor disipación de potencia que la anterior}$$

$$tpd_{prom} = 55 + 15 = 35\eta s \quad \text{mayor retardo de propagación.}$$

La Serie 74S Schottky, disminuye el retardo de tiempo al pasar de un estado a otro, elevando la velocidad de conmutación , porque el diodo Schottky, no permite que el transistor este mucho tiempo saturado, es aun más rápido que la serie 74H y tiene la misma disipación de potencia que la 74H.

Ejemplo.- Inversor Hex 74S05

$$PD \text{ prom.} = \frac{(I_{ccHH} + I_{ccL})}{2} (V_{ccmax}) = \frac{(19.8mA + 54 \text{ mA})}{2} (5.25v) = 193.72 \text{ mw}$$

$$t_{pd \text{ prom}} = \frac{t_{pl,H} + t_{pl,L}}{2} = \frac{7.5 + 7}{2} = \frac{14.5}{2} = 7.25 \text{ ns}$$

La disipación de potencia es mayor que la 7411 y el retardo de propagación disminuye.

La Serie 74LS. (Schottky de bajo consumo de potencia) tiene menor disipación de potencia pero tiene menor velocidad de propagación , se utiliza en aplicaciones donde se requiere alta velocidad con bajo consumo de potencia.

Ejemplo.- Inversor Hex 74LS05

$$P_d \text{ prom} = \frac{(ICCH + I_{ccL}) \cdot V_{cc \text{ max}}}{2} = \frac{(2.3 + 6.6 \text{ mA}) (5.25 \text{ v})}{2} = 23.625 \text{ mw}$$

$$t_{pd \text{ prom}} = \frac{t_{pl,H} + t_{pl,L}}{2} = \frac{22 \text{ ns} + 18 \text{ ns}}{2} = \frac{40 \text{ ns}}{2} = 20 \text{ ns}$$

Serie 74AS (Schottky avanzada) y 74ALS (Schottky avanzada de bajo consumo de potencia)

La primera proporciona un menor consumo de potencia y un menor retardo de propagación que la serie 74S. La segunda mejora aun más a la primera tanto en velocidad como en consumo de potencia. Utiliza un diodo especial para reducir el tiempo de conmutación de un estado a otro.

Ejemplo.- 74AS y 74ALS. ¹

$$\text{Margen de ruido } V_{NIH} = V_{OH}(\text{min}) - V_{IH}(\text{min})$$

$$V_{NL} = V_{IL}(\text{max}) - V_{OL}(\text{max})$$

En el caso de 74AS20 y el 74ALS20.- Dos compuertas NAND positivas de 4 entradas.

$$V_{NIH} = 2.5 \text{ v} - 2 \text{ v} = 0.5 \text{ v}$$

$$V_{NL} = 0.8 \text{ v} - 0.5 \text{ v} = 0.3 \text{ v}$$

¹ = Buscar ejercicio practico en el capitulo de varios topicos, sobre este circuito.

$$V_{NH} = 2.5v - 2v = 0.5v$$

$$V_{NL} = 0.7v - 0.4v = 0.3v$$

Los dos tienen el mismo margen de ruido en el estado Alto y en el estado bajo. Ahora, la TTL AS es la mejor para ser utilizada en altas frecuencias

Entradas Flotantes

Para las entradas flotantes no es conveniente dejar entradas desconectadas en el CI, ya que puede captar señales de ruido, haciendo que el mismo opere inadecuadamente.

Lo mejor es conectar estas entradas flotantes a V_{cc} mediante una resistencia de $1K\Omega$, para asegurar que el nivel lógico sea "1", dicha resistencia sirve para proteger a los transistores en las uniones base emisor de la corriente variable, si el suministro de potencia produce espigas y se pueden conectar a la misma resistencia hasta un máximo de 30 entradas flotantes.

También se puede conectar la entrada flotante a una que si se utilice siempre y cuando no este excedido su factor de carga y además que sea NAND o AND.

Para las compuertas NOR Y OR, las entradas flotantes deberán conectarse a tierra.

5.4.-TTL Y SUS CARACTERISTICAS

FACTORES DE CARGA PARA ENTRADAS INTERCONECTADAS

Para entradas interconectadas el factor de carga será igual a la suma de los factores de carga de cada entrada para las compuertas OR y NOR.

Para las AND y NAND en estado bajo el factor de carga es el mismo que si se conectará solo una entrada, para el estado alto será igual que en el caso de las OR.

Ejemplo Practico:

* (podemos realizarlo en el laboratorio para su comprobación)

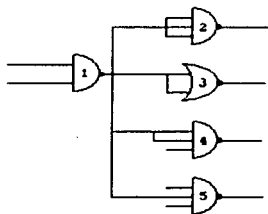


FIG. 5.4.1.-UNIDADES DE CARGA

Para determinar el no. de unidades de carga a la salida de la compuerta (1), se deben analizar cada estado lógico de la siguiente manera:

- Suponiendo que cada entrada se clasifica como 1UL en cada estado;

No. de Compuerta	Edo. Alto	No. de Compuerta	Edo. bajo
compuerta 2	3UL	compuerta 2	1UL
compuerta 3	2UL	compuerta 3	2UL
compuerta 4	2UL	compuerta 4	1UL
compuerta 5	1UL	compuerta 5	1UL

La carga de salida de la compuerta (1) para el estado Alto = 8UL ; para el estado bajo = 5 UL

Cuando hay un desacoplamiento de la fuente de alimentación para una salida TTL tipo tótem, cuando pasa de un bajo a un alto, que toma una espiga de corriente alta de la fuente, se utiliza una filtración con capacitores de radio frecuencia que van de Vcc a tierra para bajar dichas espigas de corriente.

Estos capacitores podrían ser de cerámica de baja inductancia con un valor de $0.01 \mu\text{f}$ cerca de cada circuito tratando de reducir en lo máximo las terminales del capacitor para minimizar la inductancia.

Y para filtrar las variaciones del voltaje de alimentación y de Icc cuando las salidas pasan de un estado a otro, es conveniente utilizar un capacitor de 2 a $20 \mu\text{f}$ de Vcc a tierra.

Salidas de TTL Colector Abierto

La operación AND se realiza conectando todas las salidas a un mismo punto de unión y se le denomina AND alamburada.

Si tuviéramos que realizar la siguiente función:

$$Z = (A \cdot B) \cdot (C \cdot D)' \cdot (E \cdot F)$$

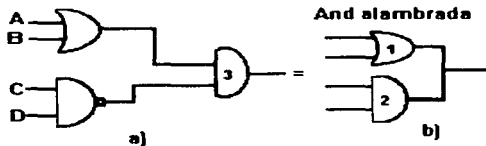


FIG. 5.42.-AND ALAMBRADA

El circuito de la fig.5.4.2(a) realiza la misma operación que el de la fig.5.4.2 (b) ya que para obtener una operación AND se necesita que una o todas sus entradas estén en "0" para obtener un "0" y para que la salida est en "1" es necesario que todas sus entradas estén en alto, así conectando las salidas de las compuertas 1 y 2 se realiza la misma función que si conectáramos la compuerta (3)

Pero en el caso de TTL tipo tótem no se recomienda realizar la AND² alambrada ya que sería perjudicial por la elevada corriente que reciben los transistores de salida de cada compuerta.

Entonces para realizar la operación AND alambrada podemos utilizarlos circuitos TTL de colector abierto , pero debe de conectarse, una resistencia en nivel alto, de Vcc a la salida de colector abierto , con un valor de $R_p = 1k(\text{ohm})$, es considerable , para dar una operación adecuada , pero estos circuitos no deben ser utilizados si se requiere de una buena velocidad de propagación.

Por ejemplo :

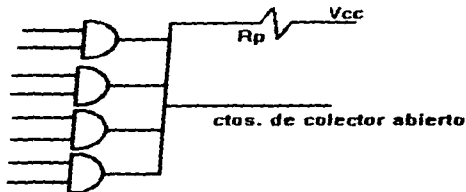


FIG. 5.4.3.-CTOS. DE COLECTOR ABIERTO

²No se debiera hacer que se exceda el valor de la corriente los.

5.5.- FAMILIA ECL Y SUS CARACTERÍSTICAS

LA FAMILIA ECL. Es una familia lógica bipolar que aumenta la velocidad de conmutación al evitar que los transistores trabajen en modo saturado, esto se realiza al traspasar la corriente de un colector a otro.

Su base fundamental es el circuito OR, esta familia es más rápido en su propagación y sus niveles lógicos son para un "1" = 0.8v y para un "0" es de 1.7v , además estos ci producen una salida con su inversor de la misma es difícil conjuntar esta familia con la TTL por la variación en sus niveles lógicos.

■).-CARACTERÍSTICAS DE LA ECL.

- Velocidad muy alta de conmutación , el tiempo de retardo en la propagación es de 2 ns , por lo que se vuelve más rápida que la schottky TTL.
- Los márgenes de ruido son bajos (250mv) por lo que se vuelve insegura esta familia para ser utilizada en medios industriales de excesivo trabajo.
- No es necesario utilizar inversores ya que ECL produce la salida y su complemento.
- Los niveles de voltaje lógicos producidos son para "1" → 0.8 v y para "0" → 1.7v
- Sus factores de carga son como de 25 debido a la baja impedancia en salida emisor -seguidor.
- La disipación de potencia es de un valor mayor a la TTL 74AS (40mw)
- Debido a que la corriente se mantiene constante impide se generen internamente los ruidos.
- Se utiliza en aplicaciones de muy alta frecuencia donde su velocidad es superior
- Es desventaja su voltaje de alimentación y los niveles lógicos que utiliza, ya que no son compatibles con los de otras series.
- ECL requieren menos energía que los TTL estándar.

5.6.- MOS Y CMOS

CI CMOS.- La Familia MOS tiene una menor velocidad de operación con respecto a la ECL, esta dado por la carga que incrementara el tiempo de conmutación, además presentan un alto margen de ruido, también tiene un factor de carga de aproximadamente 50, que es mejor que en la ECL, requiriendo menos consumo de potencia precisamente por las altas resistencias utilizadas y permitiendo con ello que sean mas convenientemente utilizadas para CI LSI y VLSI.

Es en verdad la familia de fabricación del único elemento básico (1 transistor ya sea NMOS o PMOS).

La desventaja de esta familia es que con alguna descarga electrostática puede sufrir serios daños por lo que debe tenerse mucho cuidado al manejarlos.

CMOS.- Familia lógica de Metal Oxido Semiconductor Complementario .

La CMOS es más rápida y consume menos potencia que la MOS, solo que tiene menor densidad de integración y es más compleja para su fabricación con respecto a la MOS. Y mas densa en su integración y menos compleja en su fabricación con respecto a la TTL reduciendo así su costo y además utiliza menos potencia también, aunque más un poco mas lenta que la TTL.

Los CMOS son ideales para funcionar con baterías por su bajo consumo de potencia.

Las Series 4000/14000 CMOS son usadas mas frecuentemente por su amplitud variada de funciones, además tienen una buena capacidad de corriente en sus salidas.

Las series CMOS 74C es ideal para la interconexión con TTL, por su compatibilidad de funciones y configuraciones similares, ya que varias funciones TTL se encuentran en la serie CMOS permitiendo así el reemplazo de TTL por CMOS equivalente, funcionan con 3 a 15v de suministro.

Serie 74HC (CMOS de alta velocidad) tiene aumento en su velocidad de conmutación de 10 veces más que la 74C, esto se puede comparar con los dispositivos 74LS de la TTL.

Ahora bien , la serie 74HCT tiene un diseño de compatibilidad de voltajes con las TTL, esto es que las entradas pueden conectarse a las salidas de TTL, y tienen una alta velocidad , funcionan con voltajes de 2 a 6 volts.

Para utilizar la interconexión de TTL con CMOS , el voltaje de alimentación será de 5 volts para los casos de compatibilidad , si no será necesario utilizar una interfaz para que aumente el voltaje de salida del TTL para la entrada de la CMOS.

Cuando se interconectan dos tipos diferentes de series se debe verificar los requerimientos de potencia del dispositivo de carga en este caso del CMOS.

En el caso de la corriente de entrada, la TTL satisface este problema, pero cuando se trata del voltaje, vemos que es demasiado bajo en algunos casos para algunos CMOS, por lo que deberá conectarse la salida TTL con una resistencia conectada a 5 volts, en nivel alto con lo que dará una entrada al CMOS adecuada al valor de $R_{alto} = 1K \Omega$ a $10 K\Omega$ dependiendo de las corrientes de entrada.

-Cuando se utiliza CMOS con CMOS el nivel lógico bajo puede ser 0 volts y el alto será V_{DD} .

-Los márgenes de ruido son los mismos tanto en alto como en bajo , pero dependen de V_{DD} .

Por ejemplo.- Si utilizamos $V_{DD} = 5$ v el margen de ruido $V_{NH} = V_{NL} = 30\%$ $V_{DD} = 1.5$ v mejor que en el TTL y ECL esto significa que pueden utilizarse en aplicaciones donde exista mucho ruido..

Para tener mejor inmunidad al ruido se requiere mayor disipación de potencia, mientras el CI CMOS permanezca en el mismo estado, sin cambiar, su disipación de potencia se reduce al máximo siempre y cuando este en condiciones de cd, cuando varíe la frecuencia (por lo que cambia de estado el CI) aumentará la disipación de potencia.

Cada vez que una salida pasa de bajo a alto, tiene que suministrarse una corriente de carga con oscilación momentánea a la capacitancia de carga.

Cuando la frecuencia de conmutación aumenta, también aumentarán el consumo de las espigas de corriente, producidas por la fuente V_{DD} , por lo que podemos decir que un CMOS tendrá igual disipación de potencia que un TTL 74LS, en una frecuencia de 2 a 3 Mhz.

La capacitancia de entrada de un circuito representa $5pF$ lo que limitará el factor de carga y dependiendo del mismo aumentará o disminuirá proporcionalmente el tiempo de conmutación.

-con la carga aumenta el tpd en $3 \eta s$.

Si tenemos el siguiente ejemplo en el que se utiliza la compuerta NAND:

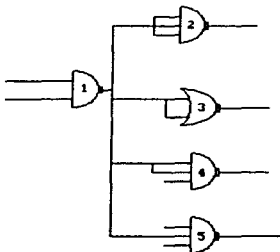


FIG. 5.6.1.- Compuerta NAND

El tpd de la compuerta NAND = $25 \eta s$ sin cargas conectadas a su salida.

Con cargas conectadas a su salida su tpd será: $t_{pd} = 25 \eta s + 3(3 \eta s) = 34 \eta s$

Dependiendo del aumento de la frecuencia que maneje el CI deberá disminuirse el factor de carga, si se quiere operar en frecuencias más grandes, deberá aumentarse también el V_{DD} .

Por la baja resistencia de salida en sus niveles lógicos aumenta la velocidad de conmutación, y aunque la resistencia permite que se cargue más rápidamente la capacitancia de carga.

En el caso de esta serie las entradas que no se utilizan siempre deberán conectarse a un nivel de voltaje o bien a otra entrada según convenga.

Debido a su resistencia (alta) de entrada se encuentra muy propensa a las cargas electrostáticas pero algunos de los CMOS ya están protegidos con diodos zener.

CI Alambrados .

La desventaja que existe en los alambrados con CI CMOS es que los CI convencionales CMOS, no deben interconectarse directamente, ya que se obtendrán los siguientes estados, nos darán ALTO cuando las salidas (ambas) estén en "1", BAJO cuando las salidas (ambas) estén en un "0", y una aproximación de $V_{DD}/2$, cuando una esté en "1" y otra en "0".

Entonces podemos utilizar para estos casos, las salidas de drenaje abierto como el de colector abierto de la TTL.

CAPITULO VI

VARIOS TÓPICOS DE DISEÑO DE SISTEMAS DIGITALES

OBJETIVO:

Conocer y desarrollar prácticamente algunos conceptos teóricos, para su comprobación.

CONTENIDO:

1).- Diseño de Controles automáticos con Cartas ASM utilizando:

a).- Flip-Flop JK

b).- Flip-Flop SR

c).- Flip-Flop T

d).- Flip-Flop D

e).- Registro de Corrimiento

f).- Contadores

2).- Tópicos de diseño en tecnología MSI.

a).- Diseño de Controles Programables

b).- Diseño de circuitos MSI.

INTRODUCCIÓN

La Electrónica digital necesita para su aplicación, desarrollar circuitos en la practica después de haber realizado el calculo teórico.

Así pues debemos, utilizar manuales y catálogos del fabricante de circuitos, para determinar la mejor configuración, la mas sencilla y la mas económica posible.

Y como ya se menciona en capitulos anteriores, según las necesidades especificas, determinan las configuraciones para la solución practica y con la ayuda de los ejemplos citados antes, podemos llegar a realizar un diseño para su aplicación, comprobar la teoría practicamente, y programar las perspectivas de diseño utilizando cartas ASM , y memorias.

6.1.-REDUCCION DE CARTA ASM

La utilización de una carta ASM, reduce un sistema de control automático con sistemas digitales, un diseño con cartas ASM, utiliza la reducción de mapas de karnaugh; utilizando variables suscritas, y nos permite utilizar otros circuitos, como un estado de memoria para transformar la salida por medio de un algoritmo de estados lógicos.

Para realizar estos diseños se utiliza primeramente un algoritmo en el cual se identifican los estados según se prevé funcionara el sistema a controlar, a el cual se le denomina carta ASM, después se realiza una tabla de estados , y se realizan los mapas de karnaugh; mapa de acción , mapa de control de modo y mapa de datos, de los cuales se determinan las ecuaciones para de ahí alambrarse los circuitos con el circuito fundamental, como ; multiplexores, flip-flops, registros, contadores, etc.

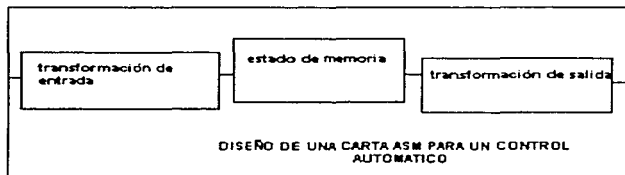


Fig. 6.1 Diagrama a Bloque de una Carta ASM.

Así es, que se pueden diseñar controladores automáticos con condiciones de entrada por medio de las cartas ASM.

Un Controlador .- es un sistema secuencial que produce niveles en tiempos apropiados para realizar operaciones sencillas que en conjunto realizan operaciones complicadas.

Las Características principales de una carta ASM son:

-Tiene 3 elementos de señalización :

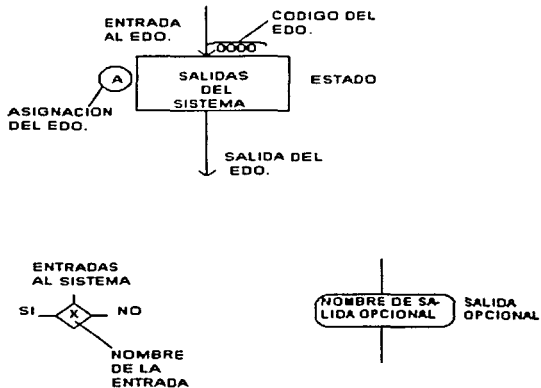


FIG. 6.2 Partes Principales de una Carta ASM

A).-METODOLOGÍA DEL DISEÑO

- 1.- Enunciar el Problema
- 2.- Construcción de la Carta o Algoritmo.
- 3.- Asignación de Estados, y Realización de la Tabla de Verdad
- 4.- Grado de Integración (elección de los circuitos de alambrado)
- 5.- Reducción
- 6.- Construcción. (Alambrado)

Para su grado de Integración del Controlador tenemos:

I).- Circuitos Combinacionales:

- Compuertas
- Decodificadores
- Multiplexores
- Rom, etc.

II).- Circuitos Secuenciales:

- Flip-Flop's
- Registros
- Contadores.

* Esto quiere decir que podemos alambrear el controlador con elementos ya sea MSI o LSI.

Ejemplo:

Alambrear la siguiente carta ASM con:

- a).- Flip- Flop JK.
- b).- Flip-Flop SR
- c).- Flip-Flop T
- d).- Flip-Flop D

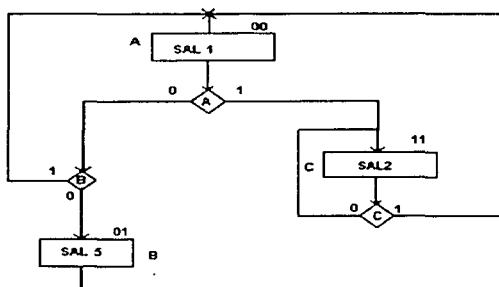


FIG. 6.13.-DIAGRAMA DE LA CARTA ASM

Tabla de verdad de los Flip-Flops

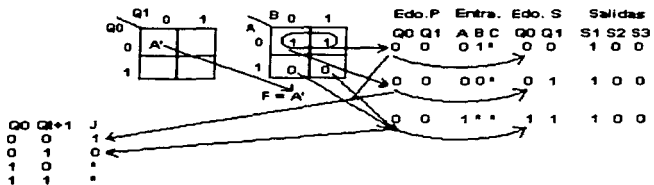
Q	Q _{t-1}	J	K	S	R	T	D
0	0	1	0	0	1	0	0
0	1	0	1	0	1	1	1
1	0	1	0	1	0	1	0
1	1	0	1	1	0	0	1

Tabla de Estados para la Carta ASM

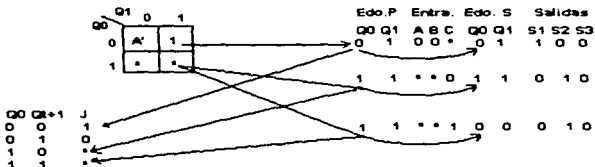
Edo. Presente		Entradas	Edo. Sigiente	Salidas		
Q0	Q1	A B C	Q0 Q1	S1	S2	S3
0	0	0 1 *	0 0	1	0	0
0	0	0 0 *	0 1	1	0	0
0	0	1 * *	1 1	1	0	0
0	1	* * *	0 0	0	0	1
1	0	* * *	* *	0	0	0
1	1	* * 0	1 1	0	1	0
1	1	* * 1	0 0	0	1	0

a).-Empleando Flip-Flop JK, tenemos:

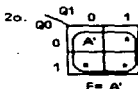
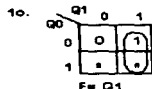
Para cuando Q0=0 y Q1=0, utilizando la tabla del F-FJK, llenamos el mapa de entradas:



Para cuando $Q0=0$ y $Q1=1$, utilizando la tabla del F-FJK, llenamos el mapa de entradas:



Ahora se reduce el mapa empleando el método de variables suscritas como sigue:



$$FT = Q1 + A'$$

- 1o. Se hacen encierros de cuadros adyacentes de "1" lógicos, la variable suscrita se hace igual a cero.
- 2o. La variable suscrita se hace igual a 1 (las otras variables suscritas son igual a 0), los "1"s originales se consideran de no importa (*), finalmente el encierro de la variable suscrita se multiplica por el contenido AND.
- 3o. Se repite el segundo paso para las otras variables suscritas.
- 4o. El resultado final es la suma de todas las funciones individuales.

Entonces el resultado del mapa para J_0 es igual a :

$$J_0 = FT = Q1 + A'$$

Realizando el mapa para obtener K_0 , J_1 , K_1 , tenemos:

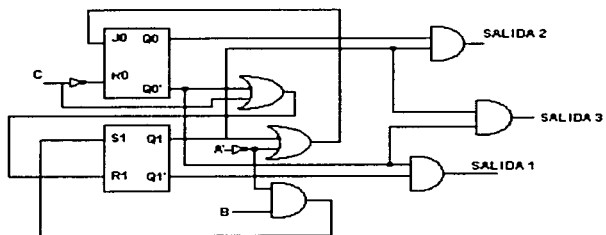
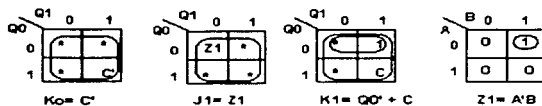
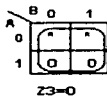
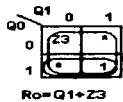
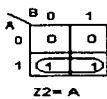
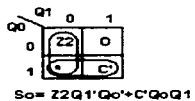


FIG. 6.1.4.- ALAMBRADO FINAL.

b).-Ahora Resolviendo la carta con FF SR, tenemos:



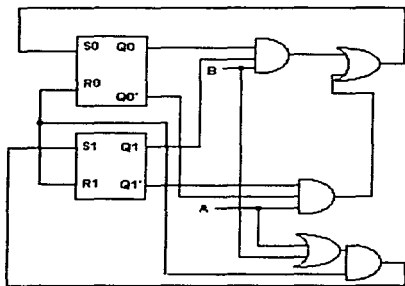
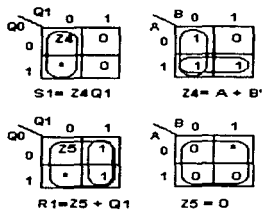
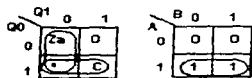


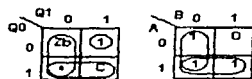
FIG. 6.1.5.- ALAMBRADO FINAL

c).-Resolviendo la Carta con FF T:



$$T_0 = Z_a Q_1' + C Q_0$$

$$Z_a = A$$



$$T_1 = Q_0' Q_1 + Q_1' Z_b + C Q_0$$

$$Z_b = A + B$$

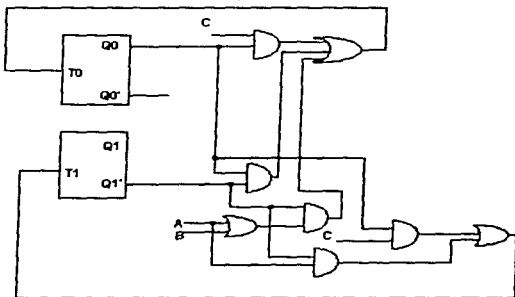


FIG.6.1.6.- ALAMBRADO FINAL

d).-Utilizando FFD:

	Q1	0	1			B	0	1
Q0		0	1	A		0	0	0
		Zc	0			1	1	1
		0	C'					

$$D_0 = Zc Q_0' Q_1' + C' Q_0 Q_1 \quad Zc = A$$

	Q1	0	1			B	0	1
Q0		0	1	A		0	1	0
		Zd	0			1	1	1
		0	C'					

$$D_1 = Zd Q_1' + C' Q_0 \quad Zd = A + B'$$

Y para obtener las salidas en cualquiera de los Flip-Flop's:

	Q1	0	1			Q1	0	1
Q0		0	1			0	0	0
		1	0			1	0	0
		0	0					

$$S_1 = Q_0' Q_1'$$

	Q1	0	1			Q1	0	1
Q0		0	0			0	0	0
		0	1			1	0	1
		1	0					

$$S_2 = Q_0 Q_1$$

	Q1	0	1			Q1	0	1
Q0		0	0			0	0	0
		0	0			1	0	0
		1	0					

$$S_3 = Q_0' Q_1$$

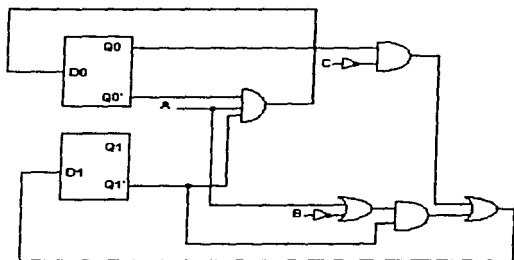


FIG. 6.1.7.- ALAMBRADO FINAL.

Ejemplo 2:

Se trata de un sistema de apertura automática de una puerta, por medio de una clave que el usuario introduce, para que el controlador compare su No. de cuenta con el número de apertura correcto que tiene grabado y permite o no el acceso al usuario.

1.- Primero se despliega un letrero que diga: "Introduzca Clave" si la clave es correcta, entonces presenta otro letrero que dice: "Presione El botón de Apertura", si no, se despliega un letrero que diga: "Trate de Nuevo".

Alambrar y diseñar la carta ASM con:

- Registros
- Contadores

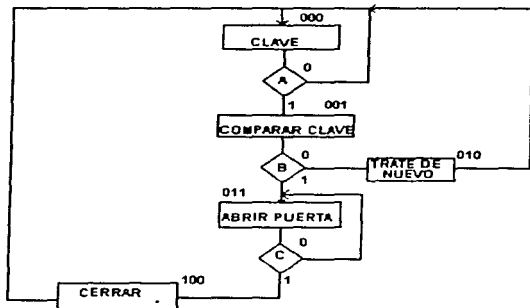


FIG. 6.1.8.- CARTA ASM

c).-Entonces si utilizamos el Registro de corrimiento 74194:

MAPA DE ACCION

		Q1 Q2			
		00	01	11	10
Q0	0	SLIC	Z	BC	BI
	1	BI	-	-	-

Edo. 0 $\frac{000}{001}$ SLICEdo. 1 $\frac{001}{010}$
 $\frac{010}{011}$ Z = SLIC + SLOCEdo. 2 010
000 BIEdo. 3 011 BC por que depende de C
011
100Edo. 4 100 BI
000

S1	S2	ACCION
0	0	Retiene Salida
0	1	SR (Corrimiento dcha)
1	0	SL (" " lqz)
1	1	Carga Paralelo

SLIC corrimiento a la lqz. con carga de 1
SLOC " " " " " " de 0

BI carga incondicional

BC carga condicional

SRIC corrimiento a la dcha. con carga de 1
SROC " " " lqz " " de 0

MAPA DE CONTROL DE MODO:

		Q1 Q2			
		00	01	11	10
Q0	0	A	B	C	1
	1	1	1	1	1

		Q1 Q2			
		00	01	11	10
Q0	0	0	0	1	1
	1	1	1	1	0

$$\leq 1 = Q0 + Q1Q2 + AQ2 + BQ2Q1 + CQ2Q1 \quad \leq 2 = Q0 + Q1$$

* Los mapas se realizan por el metodo de variables suscritas.

Donde observamos que para obtener a s1 :

-En el recuadro (o) y en el estado 000 , si A = 0 entonces retiene la salida o sea el estado 000, y si A = 1 entonces se produce un corrimiento a la izquierda con carga de un 1 ya que pasa al estado 001, por lo que consideramos que depende de la Entrada A.

- Para el recuadro (1): en el estado 001, si la entrada B = 0 entonces produce un corrimiento a la izquierda con carga de 1, si B=1, se produce un corrimiento a la izquierda con carga de cero, entonces depende de la entrada B.

- Para el recuadro (3): en el estado 011; depende de C.

-Para el estado 010 en el recuadro (2) es un 1 por que se trata de una carga incondicional

- En el recuadro (4) del estado 100; se trata de una carga incondicional por lo que vale 1.

En el caso de obtener s0;

-Para el estado 000; como es un corrimiento a la izq. con carga de 1 vale un 0.

-En el estado 001; $Z = SL1c + SLOC = 0 + 0 = 0$

- En 011; como es una carga vale un 1

- En 010; es otra carga por lo que vale un 1.

- En 100; se trata de una carga por lo que es un 1.

MAPA DE ENTRADA DE DATOS

-ENTRADAS EN SERIE

	Q1 Q2			
Q0	00	01	11	10
0	1	B	*	*
1	*	*	*	*

CUANDO SON ENTRADAS SERIE; LOS BRUNCH (CARGAS) SE CONSIDERAN DE NO IMPORTA (*)

EL RESULTADO DE LA ENTRADA EN SERIE "L"

LO OBTENEMOS ASI;

$L = Q2' + B$

-ENTRADAS EN PARALELO

* Para las entradas en paralelo los desplazamientos son de no importa(*)

Q0	Q1	Q2	00	01	11	10
0	*	*	*	*	1	0
1	0	*	*	*	*	*

Ic = Q2

Q0	Q1	Q2	00	01	11	10
0	*	*	*	*	0	0
1	0	*	*	*	*	*

IB = 0

Q0	Q1	Q2	00	01	11	10
0	*	*	*	*	0	0
1	0	*	*	*	*	*

IA = 0

Entonces el alambrado final quedaría así:

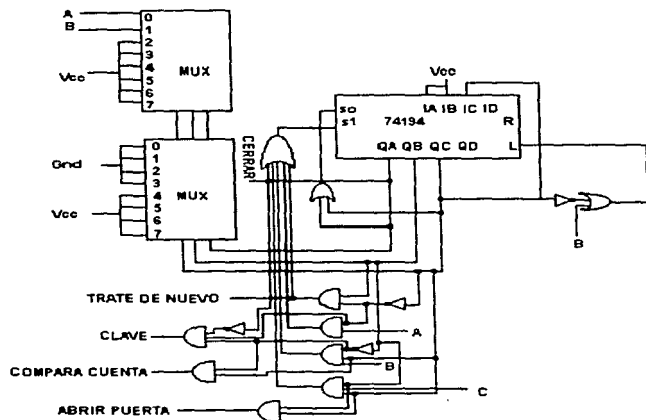


FIG. 6.1.9.- ALAMBRADO FINAL

η.-Con el Contador 74161:

MAPA DE ACCIÓN:

Donde la tabla del comportamiento del circuito es la siguiente tabla de verdad:

CONT	ENABLE	ACCIÓN
0	0	RETIENE
0	1	CONTINJA CUENTA
1	0	CARGA(SALTO, BRINCO)
1	1	-----

donde CC= cuenta condicional
 CI= cuenta incondicional
 BC= carga condicional
 BI= carga incondicional

		Q1Q2			
		00	01	11	10
Q0	0	CC	Z	Z1	B1
	1	BI	*	*	*

$$Z = CC + BC$$

$$Z1 = CC + BC$$

MAPA DE CONTROL DE MODO

		Q1Q2			
		00	01	11	10
Q0	0	0	B	B	1
	1	1	*	*	*

$$CONT = Q0 + Q1'Q2' + BQ2$$

		Q1Q2			
		00	01	11	10
Q0	0	A	B'	B'	0
	1	0	*	*	*

$$ENABLE = Q0'Q1'Q2' + B'Q2$$

Los mapas anteriores se realizan por medio del método de variables suscritas.

MAPA DE ENTRADA DE DATOS:

Q1 Q2	00	01	11	10
Q0 0	*	0	1	0
1	0	*	*	*

$$IC = Q1'Q2$$

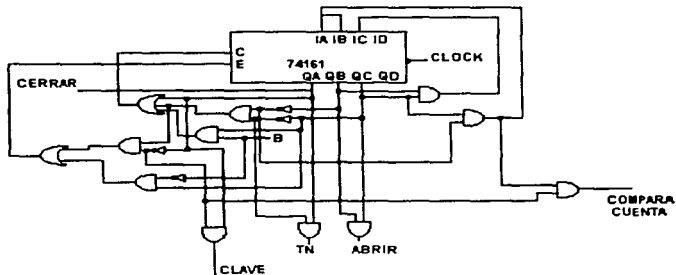
Q1 Q2	00	01	11	10
Q0 0	*	1	0	0
1	0	*	*	*

$$IB = Q1'Q2$$

Q1 Q2	00	01	11	10
Q0 0	*	1	0	0
1	0	*	*	*

$$IA = Q1'Q2$$

EN ESTE CASO LAS CUENTAS SON CONDICIONES DE NO IMPORTA(*)
EN LAS ENTRADAS EN PARALELO.



6.1.10.-ALAMBRADO FINAL

6.2.-TECNOLOGÍA MSI

Los circuitos MSI están disponibles en diferentes tecnologías (TTL, Shottky TTL, shottky de baja potencia, ECL, CMOS y se consideran como la forma lógica de mas aplicaciones. Los MSI ofrecen la ventaja de un favorable alto nivel de su integración, lo que significa bajo costo, tamaño pequeño y poco consumo, mientras se mantiene una alta confiabilidad y absoluta flexibilidad de diseño.

Al utilizar circuitos MSI y compuertas, el diseño puede no solo solucionar un sistema , sino, también obtener un ahorro de 5 a 1 en el costo de los componentes, menor área de circuito impreso y un ahorro en el consumo de potencia, sin mencionar la relación en diseño y tiempo de comprobación en el diseño tradicional de circuitos a pequeña escala de integración.

Los circuitos MSI son empleados actualmente en :

- 1.- Aplicaciones que requieren alta confiabilidad (microcomputadoras, controlador de disco), utilizadas frecuentemente con microprocesadores microprogramables.
- 2.- Aplicaciones especializadas que ofrecen la ventaja de un amplio intervalo de velocidad , de consumo de potencia ,disponibles en componentes MSI con tecnología ECL, TTL y CMOS.
- 3.- Aplicaciones pequeñas o en las que el diseño no justifica el uso de microprocesadores.
- 4.- En circuitos de apoyo o de relleno con microprocesadores, donde los MSI, efectúan la descodificación de direcciones, descodificación de estados, multiplexación de direcciones y datos o funciones similares.

Los circuitos TTL, son los mas antiguos y comunes de los MSI, incluyen la variación Schottky de bajo consumo de potencia que apareció a mediados de los años setenta.

Lo que empezó como un enfoque fragmentado en la gran lucha existente entre las compañías fabricantes de semiconductores. TI¹ tiene la mayor diversidad de circuitos integrados. Fairchild cuenta con las características mas consistentes y mejores, Signetics posee las mas populares, National introdujo la salida de 3 estados.

Los CMOS aparecieron mucho después y fueron durante mucho tiempo el producto RCA especialmente utilizado en aplicaciones militares y aeroespaciales, Motorola, National y Fairchild la significación y también se difundió la aplicación industrial de los CMOS.

La serie de circuitos RCA 4000 no se oriento del todo en sistemas, debido a su inconsistencia de características de polaridad, funciones singulares combinadas con diferentes parámetros entre algunos componentes similares de varios fabricantes.

Pero aun así, se han agregado algunas de las funciones TTL, mas comunes a la familia MSI, CMOS, sin que por esto los fabricantes hayan alcanzado el mismo grado de aceptación que tienen los MSI TTL.

Los ECL son menos conocidos, aun considerando que ofrecen mayor velocidad que los TTL, e incluso que los TTL, schottky, tienen la desventaja de un reducido margen de ruido, que demanda un mayor costo de interconexión y alto consumo de potencia.

Excepto para algunos preescaladores y lazos de amarre por fase, utilizados en radio comunicaciones, también sintonizadores de TV, los ECL se emplean solo en instrumentos complejos y en aplicaciones de prueba.

El primordial proveedor de los circuitos MECL I, II, y III es Motorola, y la de mayor éxito, pero mas lenta que la MECL III, es la MECL 10000 y es mas difícil para emplearse.

Fairchild dio a conocer la línea 100k con respuestas de nanosegundos en los circuitos multiplexadores que pueden utilizarse para simplificar diseños realizados con computas., así:

- Los circuitos 74157, Mux cuádruple de 2 entradas, genera 4 de las 16 funciones de 2 variables.

¹ Texas Instruments.

Los circuitos 74153 y 9309, Mux dobles de 4 entradas generan 2 de las 256 funciones de 3 variables.

El 74151 y 74152 Mux de 8 entradas genera una de las mas de 65536 funciones de 4 variables.

El 74150 Mux de 16 entradas genera una de las mas de 4000 millones de funciones de cinco variables.

En algunas ocasiones el uso de compuertas en ciertos diseños , resulta mas económica pero para funciones aleatorias el empleo de mux resulta mas económico , compacto, flexible y mas fácil de diseñar.

a).-DISEÑO DE CONTROLES PROGRAMABLES

La tendencia a construir equipo de control electrónico va en aumento día a día y aun mas sin hacer uso de circuitos especiales, pero semejantes a la arquitectura de una computadora que utilice circuitos regulares y programas de control en memoria (ROM o RAM).

Actualmente algunas aplicaciones de baja velocidad utiliza minicomputadoras ofrecida actualmente por los fabricantes de semiconductores, hasta algunas calculadoras muy económicas, tienen instrumentada alguna lógica programada en un solo CHIP (CI) con diseño orientado a una ROM.

En realidad el diseño de un pequeño sistema de control comienza con el conocimiento claro del número de salidas y entradas necesarias y sus características eléctricas. Pero la definición exacta de como afectan las entradas de control a las salidas, requieren mas tiempo y ocasiona errores normales.

El diseño lógico clásico solo puede comenzar cuando el diseño del sistema haya sido terminado y posiblemente necesite cambios substanciales, si el diseño de este se modifica debido a errores o nuevos requerimientos.

Sin embargo se puede diseñar, construir y probar un controlador programable tan pronto se definan los requerimientos de entrada y salida casi siempre en forma simultánea con el diseño detallado del sistema.

Existen métodos utilizados comúnmente para la realización de controladores lógicos, estos son:

- a).- Cartas ASM² (Mencionados en el primer tema de este capítulo)
- b).- PLA³

b).-DISEÑO DE CIRCUITOS MSI

Ya no es suficiente reducir el número de compuertas utilizadas, pero si es muy importante seleccionar el CI complejo apropiado que realice la función requerida mas económicamente .

Los niveles mas altos de integración ofrecen ya un menor consumo de potencia y mas confiabilidad de los sistemas .

El diseño lógico amerita una actividad mas exigente, pero igual mas estimulante y satisfactoria, además se debe participar y estar enterado de los avances de los componentes complejos con que se cuenta hoy, y conocer las repercusiones económicas de los semiconductores, circuitos impresos, conexiones, etc, y asegurar el sistema mas económico pero que cumpla con las especificaciones necesarias para su buen funcionamiento .

Podemos actualizarnos con la utilización de las siguientes reglas al diseñar un sistema.

² Maquina de estado algoritmico

³ Arreglo lógico Programable

Utilizar la arquitectura del sistema al rendimiento requerido y a los componentes utilizados.

Convertir los sistemas asíncronos a síncronos, ya que son más fáciles de diseñar, mantener y confiables.

Usar las señales de reloj en contadores y registros con mucho cuidado, ya que existen reacciones ante pulsaciones invisibles.

Ejercicio con Circuitos MSI.

2.- Dibujar el diagrama lógico del control para una cerradura digital con 128 combinaciones de entrada posibles, incluir el circuito de alarma.

• Para las 128 combinaciones de entrada se requerirán 7 entradas ($2^7 = 128$). Sea la combinación correcta la siguiente:

$$A = B = C = L, D = F = G = H.$$

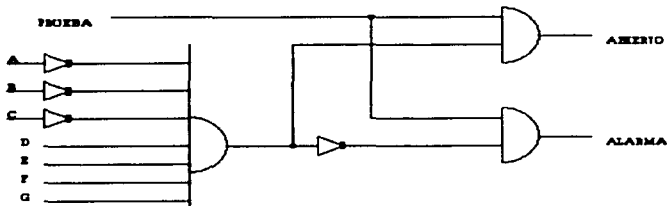


FIG. 6.2.1.-Diagrama del control para una cerradura

b.1.-MEMORIAS ROM

Para cuando se tengan grandes circuitos combinacionales con múltiples entradas y salidas, existe un componente de CI más adecuado, llamado ROM. (Memoria de Solo Lectura).

La ROM es una CI que en un solo encapsulado contiene el decodificador y las compuertas OR, las conexiones entre las salidas del decodificador y las compuertas, se puede especificar para cada configuración particular programando la ROM, por lo que podemos decir, que la ROM se utiliza para configurar un circuito combinacional complejo en un solo CI, para evitar el cableado.

Una ROM tiene n líneas de entrada con m líneas de salida, donde a las combinaciones se les llama dirección, y cada combinación de bits, que sale se le llama palabra.

Una palabra de salida puede ser seleccionada por una dirección única y como hay 2^n direcciones diferentes en una ROM, entonces hay 2^n palabras acumuladas en la unidad.

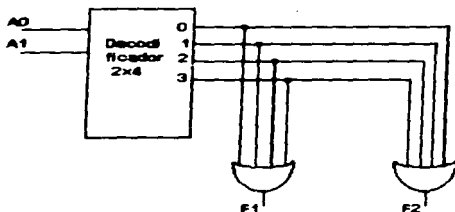


Fig. b.1.1.- Configuración de una ROM con decodificador y compuertas OR

Una de las muchas aplicaciones de la ROM es el diseño de sistemas de computación digitales (Como la configuración de circuitos combinacionales complejos)

Observando el diagrama lógico de la ROM, tenemos que cada salida produce la suma de todos los minterminos de n variables de entrada .

La ROM puede representar la función de Boole de una de las variables de salida en un circuito combinacional determinado , al romper los enlaces de aquellos minterminos que no están en la función.

Dicha ruptura de los enlaces es la programación que se le hará a la ROM, por medio de una tabla del programa ROM , que especifica, la información para los caminos necesarios en la ROM.

Ejemplo de Programación de una ROM

-expresar la programación de una ROM para:

a) $F1(A,B) = \sum(2,3)$

b) $F2(A,B) = \sum(0,3)$

1)

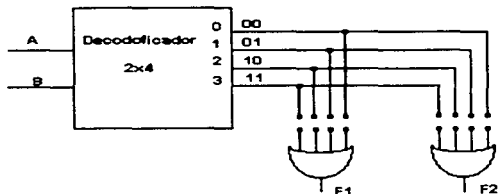


Fig. b.1.2.-Programación de una ROM

2)

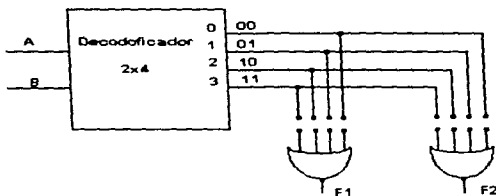


Fig b.1.3.- Configuración de ROM 4x2 con lógica Combinacional

- 1) ROM con decodificador y compuertas OR 2) ROM con decodificador y compuertas NOR

Dicho ejemplo solo muestra el procedimiento y no la situación practica real, ya que la ROM viene en un solo CI encapsulado.

Ahora bien, existen varios tipos de ROM:

1) Programación por Mascara.- la realiza el fabricante durante el ultimo proceso de fabricación del CI. Para fabricar una ROM se requiere que el cliente realice una tabla de verdad de la ROM según sus necesidades, pero dicho procedimiento es muy costoso, resultaría conveniente utilizarla solamente cuando se requieren grandes cantidades de la misma.

2) Memoria Programable (PROM) para pequeñas cantidades utilizados de ROM es mas recomendable este tipo de CI, ya que los enlaces no utilizados en la PROM se rompen por medio de pulsos de corriente a través de las terminales de salida.

Un enlace roto representa un estado binario y uno no roto representa otro estado, esto le permite al usuario programar su ROM directamente en su laboratorio para lograr la relación deseada entre las direcciones de entrada y las palabras almacenadas, existen unidades programadoras especiales para realizar dicho procedimiento.

3) PROM borrable (EPROM) .- estas memorias pueden ser recuperadas a su forma original, esto se realiza exponiendo la EPROM a una luz ultravioleta especial por un periodo de tiempo para que se realicen nuevamente los contactos internos originales, para después de regresar a su estado inicial pueda ser nuevamente reprogramada

4) Existen algunas ROM que puedan ser borradas en lugar de utilizar luz ultravioleta, pueden borrarse con señales electricas y se le denomina ROM eléctricamente alterable o EAROM.

b.2.-PLA

En diversas ocasiones un circuito combinacional tiene condiciones de no importa y si se configura con una ROM, esas condiciones de no importa se convierten en direcciones de entrada que nunca ocurrirán, las palabras en las direcciones de no importa pueden no ser programadas, y por ello se utiliza el PLA ; en los casos en los que existen un número excesivo de condiciones de no importa, resulta más económico utilizar el PLA.

Un PLA es similar a una ROM (en concepto) pero no produce la decodificación completa de las variables y no genera todos los miniterminos como en una ROM.

Dentro de un PLA el decodificador se representa mediante compuertas AND y cada una de estas compuertas puede ser programada para generar un término producto de las variables de entrada, como se muestra a continuación:

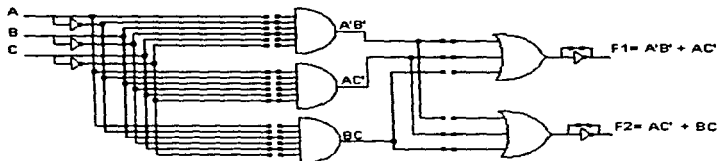


Fig. h.2.1.- Construcción de un PLA con 3 entradas, 3 términos producto y 2 salidas.

TIPOS DE PLA

- 1).- Programable por Mascara.- en este caso el cliente debe entregar una tabla de programación del PLA al fabricante.
- 2).- Programable en el campo FPLA, el cual puede ser programado por el usuario, por medio de ciertos procedimientos recomendados.

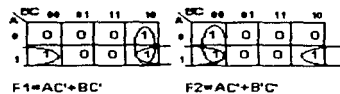
Ejemplo de programación de un PLA.

- Un PLA configura las funciones en la forma de suma de productos .

- 1).- Obtener la tabla de programa del PLA para la siguiente tabla de verdad:

ABC	F1	F2
000	0	1
001	0	0
010	1	0
011	0	0
100	1	1
101	0	0
110	1	1
111	0	0

a) Tabla de verdad



b) simplificación por mapas

Termino producto	entradas A B C	salidas F1 F2
AC'	1 - 0	1 1
BC'	- 1 0	1 -
$B'C$	- 0 0	- 1

c) Tabla de programación del PLA

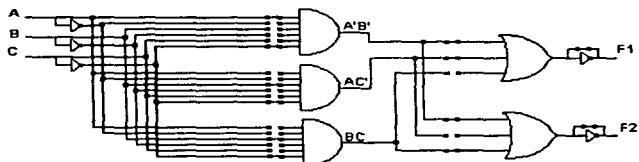


Fig. b.2.2.- Representación de la programación del PLA para el ejemplo.

CONCLUSIONES

Esta tesis esta encausada a la realización de los apuntes de la materia de sistemas digitales, como base de aplicación de los principios de la materia de sistemas digitales.

Además se introduce también tópicos actuales, como uso de microprocesadores, reflejando la practica actual sobre circuitos digitales.

También podemos encontrar especificaciones, interfaces entre familias, y circuitos mas complejos para seguir las tendencias de la industria con circuitos a mayor y alta escala de integración, y de aspectos prácticos, para familiarizarse, con el hardware de la electrónica digital.

Se presupone que el lector ya tiene una base sobre circuitos y sistemas digitales, para desarrollar ciertos problemas , pero aun así, se dispone con un apoyo didáctico en los apéndices para complementar dichos fundamentos.

Además se adhiere al final, un capitulo que involucra varios ejercicios propuestos, resueltos práctica y teóricamente sobre el tema en particular, para recordar y/o reforzar los conocimientos adquiridos en la materia.

En cuanto a la búsqueda de material teórico-practico para resolver problemas de sistemas digitales con cartas ASM (uso de controladores), pienso que no existen las suficientes bases como para estudiarlos y desarrollarlos como autodidacta. Al menos es muy difícil de localizar la suficiente bibliografía traducida al español, y si se encuentra es muy cara.

Particularmente pienso que el diseño de controladores automáticos con cartas ASM, es una manera fácil, rápida y muy practica de utilizarse (aunque muy difícil de encontrar en libros específicamente conocidos) con circuitos de cualquier escala de integración.

GLOSARIO

ACIA.- De uso menos general que los UART, son los dispositivos con estructura de bus desarrollados por algunos fabricantes de microprocesadores, estos dispositivos pueden facilitar en gran manera la conexión al microprocesador, ya que su bus de datos y las señales de sincronización necesarias han sido diseñadas específicamente para el microprocesador.

A/D.- Analógico/digital, conversión de las tensiones y corrientes analógicas procedentes de un sensor para su representación digital en sistemas informáticos.

Álgebra Booleana.- Álgebra de Boole. Álgebra capaz de poner en forma de ecuaciones las proposiciones lógicas, cuyos factores de operación son AND(Y), OR(O), NOT(NO), NOR(NO_O), etc. Desarrollada por Georges Boole.

Amplificadores sensores.- Detectan las señales de entrada diferenciales bipolares del núcleo de la memoria y proporcionan la interfaz entre la memoria son transformados en niveles lógicos compatibles con los niveles lógicos TTL.

Asíncrono.- Dispositivo que no tiene sincronía con el temporizador .

CCP.- Control por carácter

CD ROM.- Unidad lectora de discos compactos.

Circuitos de línea.- Son circuitos diseñados para transmitir señales de información digital por diferentes líneas de transmisión, los circuitos de línea son manejadores de línea, receptores de línea y transceptores. Los dispositivos pueden ser de uso general o específicamente de uso en configuración normalizada de transmisión de datos.

CMOS.- MOS complementario.- Lógica complementaria semiconductor/óxido metálico. Es una tecnología que se caracteriza por poseer muy bajo consumo, requiere un canal P y un transistor de canal N, tiene la velocidad e integración intermedias entre NMOS y PMOS. Dispone de unas características ideales en cuanto a inmunidad al ruido.

Bit.- dígito binario , unidad de información que puede adoptar dos valores o estados distintos.

BYTE.- Término que representa una porción medible de dígitos binarios consecutivos.

D/A.- Digital /analógico , conversión de una representación digital utilizada en ordenadores, para excitar altavoces, motores, etc.

DIP.- Encapsulado de doble línea de un circuito integrado.

ECL.- Lógica de acople por el emisor. Emplea transistores bipolares y es extraordinariamente rápida

E/S.- Líneas o dispositivos empleados para obtener o visualizar la información en el exterior. Entrada /salida.

Familia Lógica .- Son las características de funcionamiento en las que se basa para trabajar un circuito integrado.

FAN OUT.- Cargabilidad de salida.- Carga eléctrica que puede excitar una salida, Numero de entradas que pueden ser excitadas.

Flip-Flop.- Este circuito puede mantener un estado binario en forma indefinida hasta que recibe otra información para cambiar de estado.

gnd.- Es la tierra a la que debe ser conectado el CI para que funcione correctamente, tiene el valor de un "0" lógico

ICCH.- Consumo de corriente en estado alto.

ICCL.- Consumo de corriente en estado bajo.

IIIH.- Corriente de entrada alto.

IIIL.- Corriente de entrada bajo.

Interfaz.- Conexión común a dos sistemas distintos de calculadores o a dos partes de un mismo sistema, conceptos comunes que relacionan a dos equipos de diferentes funciones.

IOH.- Corriente de salida a nivel alto

IOL.- Corriente de salida a nivel bajo.

LSI.- Alta escala de integración en CI.

Manejadores de Periféricos.- Suelen ser circuitos que generalmente contienen 2 o mas transistores capaces de manejar voltajes y corrientes grandes, con todos los resistores de limitación de corriente y de polarización, necesarios para permitir la conexión directa del dispositivo a TTL, MOS o CMOS con un mínimo de componentes externos, además algunos manejadores incluyen algunas compuertas lógicas internas. (AND, NAND, OR, NOR), que permiten realizar ciertas funciones lógicas en el propio circuito de interfaz.

Manejadora para periféricos.- Los manejadores para periféricos son CI que permiten la interconexión de niveles lógicos TTL, MOS, CMOS, con componentes de voltaje y corriente mas elevados, como lamparas, relevadores o motores.

Manejadora de visualizadores.- es una interfaz con microcomputadora en CI (microprocesador), los puertos de salida de un microprocesador o con otros sistemas lógicos digitales y suministra los niveles pertinentes de corriente y voltaje requeridos por el tipo de visualizador usado.

USRT.-Los primeros circuitos integrados receptores/transmisores sincronías, fueron básicamente los UART, modificados para operar en modo CCP .

Vcc.- Voltaje de polarización , se aplica para hacer funcionar al circuito integrado, tiene el valor de un "1" lógico.

VIH.-Voltaje de entrada alto.

VIL.-voltaje de entrada bajo.

VLSI.-Muy alta escala de integración de CI.

VNH.- Margen de ruido a nivel alto.

VNL.- Margen de ruido a nivel bajo.

VOH.-Voltaje de salida alto

VOL.-voltaje de salida bajo.

WROM.- Escriba solo una vez en el disco .

APÉNDICES

OBJETIVO:

Comprender y utilizar los diferentes sistemas de numeración y códigos utilizados en los sistemas digitales.

CONTENIDO:

- A.- *Sistemas de Numeración*
 - a.- *Representación numérica*
 - b.- *Conversiones de base M a base 10*
 - c.- *Conversiones de base M a base 10*
 - c.1.- *Sistemas binarios, octal y hexadecimal.*
 - c.2.- *Conversiones de binario a octal y viceversa*
 - c.3.- *Conversiones de binario a hexadecimal.*
 - d.- *Operaciones aritméticas con números signados*
 - e.- *Sistemas de numeración complementarios*
 - f.- *Representaciones binarias con números signados*
 - f.1.- *Operaciones binarias con números signados*
- B.- *Códigos*
 - a.- *Bin, Bytes, Palabras*
 - b.- *Códigos binarios, BCD, reflejados, exceso 3, alfanuméricos y Gray.*

INTRODUCCIÓN

Cuando se utiliza equipo de medición, se necesita representar la cantidad de la señal de información que se mide (paso a paso). Por ejemplo si utilizamos un termómetro, para localizar las variaciones de temperatura mediante una escala, digamos de 0° C a 40 °C en la cual el nivel de mercurio ascenderá conforme vaya aumentando la temperatura ambiental y viceversa, en este caso la señal medida estará representada por un conjunto de valores de información de tipo analógico (Por que a medida que aumenta la temperatura , se le asocia un valor correspondiente a su incremento), esto es que se manejan señales continuas.

También se puede obtener la medición de valores discretos de información , por ejemplo; si queremos saber el tiempo de acceso para el encendido de las luces de un semáforo, dependerá del tránsito que tenga la carretera, esto es que puede durar más tiempo encendida la luz verde que la roja o la amarilla, como esto depende de la secuencia y varia con el tiempo , se puede decir que tenemos un tiempo de representación digital de la señal de información que estamos utilizando.(ya que maneja elementos discretos de información).

Entonces, siempre es necesario representar la información, ya sea para comprender o bien para procesar un circuito requerido que opere con la anterior, para esto empleamos diversos sistemas numéricos y las conversiones respectivas para que las entiendan tanto las máquinas como el usuario.

Ahora bien, cuando se utiliza algún tipo de calculadora digital, se realizan diversas operaciones aritméticas y para lograrlo se hace uso de los códigos , ya que para sumar dos números (5+8), se teclan estos y la operación , el teclado funciona como receptor de la información que son los números, después de esta etapa de recepción, se encuentra la etapa en que se convierten estas señales en un representación numérica digital , esto es a un código de señales que pueda interpretar los circuitos para realizar la operación de adición, y después se vuelve a convertir la señal de salida a una señal analógica que será representada en la pantalla de la calculadora por códigos decimales, para que el usuario los identifique.

Así mismo se emplean las representaciones analógicas y digitales en diferentes dispositivos como las computadoras, el equipo de medición o sistemas electromecánicos entre otros, como resulta en el control de acceso de apertura de una puerta, ya que por medio de un código de números operando como clave, se abrirá si es correcta por medio de un mecanismo de accionamiento.

A.- SISTEMAS DE NUMERACION**a).-Representación numérica.**

Para localizar las cantidades de valores numéricos debemos definir el sistema numérico a emplear. Ya que no es lo mismo introducirle información a un circuito electrónico por medio de números decimales, que mandarle los datos en números binarios.

Lo que podemos hacer, es cambiar la presentación de un número en un sistema por otro que sea su equivalente en un sistema requerido.

Así para representar 835.2_{10} con potencias de 10:

$$8 \times 10^2 + 3 \times 10^1 + 5 \times 10^0 + 2 \times 10^{-1} = 835.2_{10}$$

Así tenemos el No. 111.01_2 en potencias de 2 representaría:

$$1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 7.25_{10}$$

b).-Conversión de base m a base 10.

Si queremos convertir 421.2_5 (base 5), a su equivalente a base 10, lo representaremos en potencias de 5 y realizaremos las operaciones:

$$4 \times 5^2 + 2 \times 5^1 + 1 \times 5^0 + 2 \times 5^{-1} = 111.4_{10}$$

Entonces si la base del número en cuestión es M, y lo queremos en base 10, lo representaremos en potencias de M,:

$$A_n (M^n) + \dots + A_0 (M^0) + \dots + A_{-r} (M^{-r}) = \text{Equivalente decimal}$$

\
/

coeficientes
potencias de M

c).-Conversión de base 10 base m.

Para convertir 85_{10} a base 2 (binario), dividimos el número 85 entre la base a la que queremos convertir, en este caso es 2, el residuo formara el número equivalente en base 2:

$85/2 = 42$	Residuo = 1	=A ₀
$42/2 = 21$	Residuo = 0	=A ₁
$21/2 = 10$	Residuo = 1	=A ₂
$10/2 = 5$	Residuo = 0	=A ₃
$05/2 = 2$	Residuo = 1	=A ₄
$02/2 = 1$	Residuo = 1	=A ₅
	Residuo = 1	=A ₆

$$85_{10} = 1010101_2$$

Cuando se trata de una fracción en base 10 a convertir a base M, en lugar de dividir se multiplica por la base a convertir y, el resultado del producto en número entero será el equivalente en base M.

Si:	$0.375_{10} =$	$?_2$
	$0.373 \times 2 =$	$0.75 \quad 0 = A_{-1}$
	$0.75 \times 2 =$	$1.5 \quad 1 = A_{-2}$
	$0.5 \times 2 =$	$1 \quad 1 = A_{-3}$
	$0.375_{10} =$	011_2

c.1).-Sistemas binarios, octal y hexadecimal

El sistema binario es uno de los sistemas mas casuales por su accesibilidad para utilizarse en las máquinas o circuitos digitales.

Este sistema emplea dos posibles valores que son "0" y "1", los cuales identifican, niveles de voltaje, abrir o cerrar interruptores, etc.

Los sistemas numéricos como el octal (base 8), el decimal (base 10) y el hexadecimal (base 16) también tienen muchas aplicaciones además de la facilidad de convertir al sistema binario y viceversa.

El sistema decimal tiene como base 10 dígitos (0 al 9) el octal tiene como base 8 dígitos (0 al 7) y el hexadecimal tiene 16 dígitos (0 al 9) y (A a la f) para poder completar los dígitos utilizados.

Cada número hexadecimal utiliza un grupo de 4 dígitos para representar su equivalencia binario, las letras de A a la f, representan los valores del 10 al 15 en decimal.

Para poder realizar un conteo en un sistema, por ejemplo el octal se cuenta desde 0 hasta 7 regresa al 1 e incrementa un dígito en ascendencia combinatoria de los mismos hasta terminar la secuencia.

La tabla siguiente muestra la relación entre los 4 sistemas.

DECIMAL	BINARIO	BASE 8	BASE 16
0	0000	0	0
1	0001	1	1
.	.	.	.
.	.	.	.
8	1000	10	8
.	.	.	.
.	.	.	.
15	1111	17	f

c.2).-Conversión de base 2 a octal y viceversa.

Para realizar el proceso inverso de conversión de base 8 a base 2, se hacen las asignaciones de los grupos de tres dígitos en base 2 con el equivalente en base 8 a cada grupo en particular.

$$\begin{array}{rcccccccc}
 (3547701)_8 & = & \begin{array}{c} 3 \\ \hline 011 \end{array} & \begin{array}{c} 5 \\ \hline 101 \end{array} & \begin{array}{c} 4 \\ \hline 100 \end{array} & \begin{array}{c} 7 \\ \hline 111 \end{array} & \begin{array}{c} 0 \\ \hline 000 \end{array} & \begin{array}{c} 1 \\ \hline 001 \end{array} \\
 354701_8 & = & 01110110011100001_2
 \end{array}$$

c.3).-Conversión de base 2 a hexadecimal.

Como cada dígito hexadecimal es equivalente a 4 dígitos binarios, para convertir un número binario a su equivalencia hexadecimal, se realizan las particiones en grupos de 4 dígitos binarios de derecha a izquierda después del punto, y se le asigna el valor correspondiente a hexadecimal:

$$\begin{array}{r}
 \text{Sea } 100010110_2 = 7_{16} \\
 \hline
 0001\ 0001\ 0110_2 = 116_{16} \\
 \hline
 1 \quad \quad 1 \quad \quad 6
 \end{array}$$

Para convertir un número de base 16 a base 2, realizamos el proceso inverso, asignándole a cada dígito hexadecimal sus equivalente de 4 dígitos binarios.

$$\begin{array}{rcccccc}
 35A01_{16} = & \begin{array}{c} 0001 \\ \hline 3 \end{array} & \begin{array}{c} 0101 \\ \hline 5 \end{array} & \begin{array}{c} 1010 \\ \hline A \end{array} & \begin{array}{c} 0000 \\ \hline 0 \end{array} & \begin{array}{c} 0001 \\ \hline 1 \end{array} \\
 & = & 00110101101000000001_2
 \end{array}$$

d).-Operaciones con números no signados.

Dentro de las operaciones aritméticas con números sin signo tenemos, que es la adición la que se realiza igual que con números decimales, haciendo uso de las cuatro posibilidades para su realización:

$0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 1 = 10 = 0 +$ corrimiento de 1 a la siguiente posición.
 $1 + 1 + 1 = 11 = 1 +$ corrimiento de 1 a la siguiente posición.

Entonces para:

```

      1111 (15)
+   1010 (10)
-----
1  1001 (25)
  
```

Para desarrollar las operaciones tales como la sustracción, multiplicación y división de números binarios sin signo se emplea básicamente a la adición.

e).-Sistemas de numeración complementarios.

En algunas máquinas binarias se emplea la forma de magnitud verdadera para almacenar a los números después de transformarlos en complemento a 2, por que es fácil de manejarse con los mismos dispositivos provocando un ahorro físico de los mismos, en otras máquinas se utiliza el complemento a 1 o complemento a 2, pero el más confiable es el complemento a 2.

FORMA DE MAGNITUD VERDADERA.- En el No. 17_{10} representado en su forma binaria :

1 10001₂ tiene un bit de signo, 5 bits de magnitud
 -- -----
 magnitud

bit de signo

los cuales son el equivalente binario verdadero del valor que representa

FORMA DE COMPLEMENTO A 1.- En este caso se realiza al cambio de cada bit del número binario por su complemento, cada 0 es cambiado por 1 y cada 1 es cambiado por 0.

$$\begin{aligned} \text{Entonces el complemento a 1 de } -52_{10} &= 1110100_2 \\ &1001011 \text{ C.1} \end{aligned}$$

* El bit de signo no cambia, solamente la magnitud real del número por su complemento a 1.

FORMA DE COMPLEMENTO A 2.

Para obtener el complemento a 2 se toma el complemento a 1 del número binario y se le suma un 1 al bit menos significativo (LSB) como sigue:

$$\begin{array}{r} 11011101 \\ 00100010 \text{ C.1} \\ + \quad \quad 1 \text{ sumar 1 al LBS para obtener :} \\ \hline 00100011 \text{ C.2} \end{array}$$

f).-Representación binaria de números signados.

Para cualquier operación aritmética que opere con números signados es necesario involucrar los positivos y los negativos, esto se lleva a cabo agregando un bit mas al número en cuestión, este es el llamado bit de signo para designar un número negativo se le agrega un "1" y para un positivo se le agrega un "0".

Para representar el No. $+17_{10}$

$$\begin{array}{r} 0 \quad \quad 10001_2 \\ \hline \text{bit de signo} \quad \text{magnitud} \end{array}$$

Para representar el No. -17_{10}

$$\begin{array}{r} \underline{1} \qquad 10001_2 \\ \text{bit de signo} \quad \text{magnitud} \end{array}$$

Para representar la magnitud de un número negativo se pueden tomar cualquiera de las siguientes formas:

- 1.- Forma de magnitud verdadera
- 2.- Forma de complemento a 1
- 3.- Forma de complemento a 2.

f.1).-Operaciones con números binarios signados.

ADICIÓN.

En caso de dos números positivos, la adición es directa:

$$\begin{array}{r} + 9 \ 0 \ 1001 \\ + 4 \ 0 \ 0100 \\ \hline + 13 \ 0 \ 1101 \text{ magnitud} \\ \text{bit de signo} \end{array}$$

Cuando se trata de un No. negativo menor y un No. positivo mayor:

$$\begin{array}{r} + 7 \ 0 \ 0111 \\ - 5 \ 1 \ 1011 \\ \hline + 2 \ 10 \ 0010 \end{array} \quad \text{para } 5_{10} = \begin{array}{r} 0101 \\ 1010 \text{ C.1} \\ + 1 \\ \hline 1011 \text{ C.2} \end{array}$$

el acarreo final se desprecia quedando:

$$0\ 0019 = +2_{10}$$

en el caso de un número positivo y un número negativo mayor:

$$\begin{array}{r} -1010110 \\ +400100 \\ \hline -600100 \end{array} \quad \begin{array}{l} \text{para } 10_{10} = \\ \hline 1010 \\ 0101 \text{ C.1} \\ 1011 \text{ C.2} \end{array}$$

Sustracción en el complemento a 2.

Para restar un número binario (sustraendo) de otro número binario (minuendo) se procede como sigue:

Se altera el signo cambiando el sustraendo a complemento a 2 incluyendo el bit de signo, este resultado se suma al minuendo tomado en su forma verdadera y el resultado será entonces la sustracción de dos números binarios.

Entonces para:

$$\begin{array}{r} - \quad +500101 \text{ minuendo} \\ \quad +311101 \text{ sustraendo} \\ \hline +2100010 \end{array} \quad \begin{array}{l} \text{C.2 de } 3_{10} = 11101 \end{array}$$

La adición se convierte a sustracción utilizando el complemento a 2 de un número (sustraendo) para sumarlo al minuendo.

Multiplicación a complemento a 2.

Si los dos números a multiplicar son positivos, entonces se encuentran en forma verdadera, y así se multiplican como con los decimales, al resultado se le agrega un bit "0" porque será positivo el producto, en el caso de que el producto sea negativo entonces al resultado se le agregara un 1.

$$\begin{array}{r} 0111 \\ x 1110 \\ \hline 0000 \\ 0111 \\ 0111 \\ 0111 \\ \hline 1100010 \end{array}$$

B.- CODIGOS**a).-Bits, Bytes, Palabras.**

Los sistemas digitales utilizan diferentes tipos de códigos y estos a su vez necesitan del empleo de:

- a) Bits
- b) Bytes
- c) Palabras

En muchos procesos industriales se miden señales analógicas y se convierten en señales digitales, para que ciertos circuitos las comprendan, y se proceda a realizar la conversión de la señal digital a una señal analógica, para alimentarla en algún controlador del mecanismo de un proceso industrial indicado. Existen diversas formas de información que continuamente se transfieren y manipulan en el interior de una computadora, la unidad mas elemental de información utilizada es el bit (dígito binario). Consecuentemente la unidad primaria de información es un grupo de bits (palabra de computadora), resulta por tanto, importante describir el tamaño de la palabra de la computadora, por que es la manera como se les distinguen a las anteriores. Entonces, si tenemos una computadora de 16 bits, se trata de una maquina que procesa información en unidades de 16 bits, y además indica el tamaño de la palabra de la unidad de memoria, resultando una memoria que almacena un cierto número de palabras de 16 bits.

El byte es un grupo de 8 bits siendo la unidad universal, el tamaño de palabra se determina por el número de bytes utilizados. Entonces una computadora tiene un tamaño de palabra de 2 bytes si tiene 16 bits.

Una palabra almacenada en una memoria de una computadora puede tener tres tipos de información:

- 1) Instrucciones
- 2) Datos codificados y
- 3) Datos numéricos binarios.

Para observar estas diferencias recordemos el caso de la calculadora en la cual tecleamos la información por medio de instrucciones de la operación aritmética que queremos saber, luego el teclado pasara esta información a otros circuitos que convierten a la anterior en datos codificados en datos numéricos binarios para traducirlos y volverlos a convertir a la salida al código original para su representación en pantalla.

El contenido de cada mensaje, entonces depende del uso exacto del código, para que las instrucciones o datos sean debidamente interpretados.

**b.-CÓDIGOS BINARIOS, BCD,
REFLEJADOS, EXCESO 3,
ALFANUMÉRICOS Y GRAY.**

CÓDIGO BINARIO.

Si queremos hacer la representación de números de decimales a un código binario por medio de 3 bits entonces:

$$n = \text{No. de bits} = 3 \quad \therefore \quad 2^n = 2^3 = 8 \quad \therefore$$

solamente podremos representar del 0 al 7 decimales a su equivalente en binario:

DECIMAL	BINARIO CON 3 BITS
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Dependiendo del número de bits utilizados será la representación lograda

CÓDIGO BCD.

Para realizar una codificación binaria directa, necesitamos hacer uso de 4 bits binarios para representar su equivalente a decimal de un número.

Entonces tenemos 143_{10} , cada dígito decimal se cambiara por su equivalente binario de 4 bits:

<u>1</u>	<u>4</u>	<u>3</u>	Decimal
0000	0100	0011	BCD

Este código utiliza números de 4 bits únicamente de 0000 al 1001.

CÓDIGOS REFLEJADOS.

Cuando se requiere la manipulación de la información en forma continua se utilizan la conversiones analógico - digital o viceversa, entonces para interpretar estas transformaciones se requiere del código reflejado se emplean también 4 bits para representarlo a su equivalente binario, con la diferencia del código reflejado cambia en un bit del número que precede al siguiente, como se muestra en la siguiente tabla:

CÓDIGO REFLEJADO	EQUIVALENTE DECIMAL
0000	0
0001	1
0011	2
0010	3
0110	4
0111	5

Se puede obtener un código reflejado diferente al utilizado en la tabla anterior, ya que para deducirlo solo hay que cambiar un bit de "0" a "1" o "1" a "0", comenzando en cualquier combinación de los mismos, tomando en cuenta que no habrá repetición de la asignación que se le de a un número con respecto de otro.

CÓDIGO DE EXCESO 3.

Este código no es mas que una implementación del código BCD, ya que se realiza un mismo tipo de codificación de decimal a binario, nada mas que antes de tal codificación, se le suma el numero 3 a cada dígito decimal y después se le asigna su equivalente a binario con 4 bits de representación.

Pensemos que, el 35_{10} debe convertirse a exceso 3;

sumamos 3 unidades a cada dígito decimal

$$\begin{array}{r} + 3 \\ \underline{3} \\ 6 \\ \hline 0110 \end{array}$$

$$\begin{array}{r} + 5 \\ \underline{3} \\ 8 \\ \hline 1000 \end{array}$$

equivalente de 4 bits en binario.

CÓDIGO ALFANUMÉRICO.

Existen algunos usos en la manipulación de información, que requieren tanto de carácter especiales como de letras, además de los ya conocidos números en su equivalente binario, entonces se requiere de la utilización de un código binario especial, este se llama código alfanumérico, se trata de un código binario que consta de 10 dígitos decimales 26 letras del alfabeto mayúsculas y 26 minúsculas, y 25 símbolos especiales, entonces al ser codificados en grupo de caracteres alfanuméricos deberá hacerse con un mínimo de 7 bits, por $2^7 = 128$ posibles números binarios con los cuales podríamos hacer la representación de los 87 caracteres anteriormente mencionados como alfanuméricos.

El código (ASCII), es el código alfanumérico mas común y mas utilizado por los fabricantes de microcomputadoras, pues es utilizado en la transmisión de información alfanumérica que existen entre los dispositivos de entrada y salida de una computadora (tales dispositivos podrían ser: la impresora, el monitor, etc.)

Entonces la computadora utiliza el código ASCII para comprender las instrucciones que se le dan, produciéndose únicamente las conversiones entre los caracteres y el código ASCII de 7 bits en binario.

CÓDIGO GRAY

El código Gray no es mas que un tipo mas de los códigos reflejados ya que dependen de la manipulación en el cambio que se efectúa de 0 a 1 en un carácter decimal a un equivalente binario.

Esto es que solo un bit de todo el grupo de bits que son parte de la equivalencia binaria, es cambiado al pasar de una etapa a otra.

Como este código carece de valor, no se puede usar para operaciones aritméticas, si no, más bien es usado en convertidores de información de analógica a digital.

BIBLIOGRAFÍA

- CIRCUITOS ELECTRÓNICOS DISCRETOS E INTEGRADOS
DONALD L. SCHILLING CHARLES BELOVE
ALFA OMEGA. MARCOMBO, 2a. EDICIÓN.

-CIRCUITOS INTEGRADOS DIGITALES Y COMPUTADORAS
BARRY WOOLLARD
PARAMITO 4a. EDICIÓN.

-CIRCUITOS LÓGICOS Y SISTEMAS DE MICROCOMPUTADORAS
A. WIATROWSKI, CLAUDE &
H. HOUSE, CHARLES.
LIMUSA, 1a EDICIÓN, MÉXICO

- DISEÑO DIGITAL
MORRIS MANO
PENTRICE HALL.

-DISEÑO DE SISTEMAS DIGITALES
P. DESCHAMPS, JEAN &
ANGULO JOSÉ M.a.
PARANINFO, 2a. EDICIÓN, ESPAÑA.

-DIGITAL CIRCUITS & LOGICAL DESIGN
C. LEE SAMUEL
PRENTICE HALL, EUA.

-DISPOSITIVOS ELECTRÓNICOS III
MARGARITA GARCÍA
IPN

-ELECTRICIDAD Y ELECTRÓNICA APLICACIONES PRACTICAS
TOMO IV
MC. GRAW HILL

-ELECTRÓNICA
J.M. CALVERT
VOLUMEN III
EDICIONES CIENCIA Y TÉCNICA.

- ELECTRÓNICA DIGITAL BÁSICA.
JHON A. DEMPSEY
REPRESENTACIONES Y SERVICIOS DE ING. S.A., MEX.

-ELECTRÓNICA DIGITAL BÁSICA
EDUTELSA

-INTEGRATED CIRCUIT ENGINEERING
J. GLASER Y J. SUBAL SHERPE
ADDISON-WESLEY READING MASS 1978

-LÓGICA DIGITAL
MORRIS MANO
PRENTICE HALL

- MANUAL DE CIRCUITOS INTEGRADOS .
MOTOROLA.

- MANUAL DE CIRCUITOS INTEGRADOS, SELECCIÓN Y DISEÑO.
ARTHUR B. WILLIAMS
TOMO I, II y III.
MC. GRAW HILL.

-MANUAL ECG PARA TTL

- MODER LOGIC DESIGN
GREEN, DAVID
ADDISON, WESLEY, EEUU

-PRINCIPIOS DIGITALES
ROGER L TOKHEIM
MC. GRAW HILL, 2a. EDICIÓN
SCHAUM.

- PRINCIPIOS Y APLICACIONES DE SISTEMAS DIGITALES.
RONALD J. TOCCI
PRENTICE HALL, 3a. EDICIÓN.

-SISTEMAS DIGITALES
RONALD J. TOCCI
PRENTICE HALL, 2a. EDICIÓN.

-TEORIA DE CONMUTACIÓN Y DISEÑO LÓGICO.
J. HILL, FREDERICK &
R. PETERSON, GERALD.
LIMUSA, MÉXICO. D.F.

- APUNTES Y PRACTICAS (VARIOS), DE LA MATERIA DE CIRCUITOS
LÓGICOS DE LA CARRERA DE ING. MEC. ELÉCTRICA DE LA ENEP
"ARAGÓN".