42

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

SIMULADOR DE PROMs PARA LA IMPLEMENTACION DE ALGORITMO DE MAQUINA DE ESTADOS

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A N:

LUZ DEL CARMEN GUZMAN ROBLES

MARICELA MORALES HERNANDEZ

DIRECTOR DE TESIS: ING.MARTIN PEREZ MONDRAGON

MEXICO D.F. 1994

TESIS CON
FALLA DE ORIGEN

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 está 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.

A MIS QUERIDOS PADRES

Humbertina y Pedro

Con mucho cariño y amor les dedico esta tesis, porque gracias a ustedes he logrado una de mis más anheladas metas.

Esta tesis es el resultado de un gran esfuerzo tanto mío como de ustedes, ya que la vida no es fácil, hubo obstáculos que pude vencer por el apoyo y confianza que siempre he sentido hacia ustedes.

A ti mamá quiero darte las gracias por todo el cariño y amor que me has brindado y por apoyarme en cada momento y en cada desición que he tomado, ya que ésto ha contribuido que poco a poco vaya consiguiendo las metas que me he trazado en la vida, y quiero que sepas que para mí eres una gran mujer, por tu fuerza y vitalidad que siempre has tenido frente a la vida y que siempre me has transmitido, por eso esta tesis es un fruto de lo que tu has sembrado en mí.

A ti papá te agradezco todo lo que me enseñaste desde niña y sobre todo la confianza y el deseo de superación que me inculcaste. Y aunque ya no estes entre nosotros el cariño y amor que me diste sigue viviendo en mí, y se que tú también te sentirás feliz de que haya logrado esta meta, porque tú siempre vivirás en mi mente y en mi corazón.

Con mucho cariño:
Lucy

A MIS QUERIDOS HERMANOS

Pedro y Roberto

Gracias por el apoyo, cariño, y comprensión que siempre me han brindado, y sobre todo por motivarme a seguir adelante y superarme cada día más. Esta tesis es una de mis metas más anheladas, en la que he puesto todo lo mejor de mí, pues creo es la mejor manera de corresponder a lo que ustedes me han brindado, es por eso que con mucho cariño esta tesis va dedicada a ustedes.

Con mucho cariño:
Lucy

A MARICELA

Por la amistad y el apoyo que me has brindado en todo momento, lo cual ha contribuido en gran manera para la realización de este trabajo, y también por haber sido mi mejor amiga y compañera durante la estancia en la Facultad.

Con afecto y cariño:
Luz

A VICTOR MANUEL

El apoyo y cariño que me has brindado en estos años, así como tu compañía me han motivado a ser cada vez mejor. Gracias por compartir conmigo tanto momentos alegres como tristes y por brindarme tu ayuda, cuando la he necesitado.

Con mucho cariño:
Lucy

A MIS PADRES

ALICIA Y RAFAEL.

Este trabajo lo dedico con todo mi amor a ustedes, mis padres, que han sabido guiarme y apoyarme a lo largo de mi vida con la paciencia y el cariño necesarios, además de que me han enseñado que todo lo que se anhela se consigue a base de esfuerzos y constancia.

A ti' mamá, muy especialmente te doy las gracias por haberme dado la oportunidad de vivir y crecer a tu lado, porque de esa manera he podido conocerte como amiga, también te doy las gracias por todos los hermosos momentos que hemos compartido a lo largo de mi vida. Así que tú sabes muy bien que este trabajo es un reconocimiento a tu labor como MADRE.

A tí papá, que me has brindado siempre tu mano amiga y que has señalado algunos de mis errores que no fui capaz de ver, te doy las gracias por todos esos momentos gratos en mi vida, en especial este momento que es para mí muy importante, que con tu presencia y tus palabras siempre alentadoras llegó a hacerse realidad.

Gracias a ambos por todo lo que me han brindado y por aceptarme como soy. Para mí siempre serán un ejemplo a seguir, porque he visto que su vida y la nuestra la han construido con muchos sacrificios, pero siempre con la mentalidad de salir adelante a pesar de las dificultades que se presentaron.

Con profundo respeto:

Mari.

A MIS HERMANOS HILDA. CESAR. RAFA Y VERO:

Por darme palabras de aliento cuando fue importante sentir su presencia y su apoyo incondicionales, por ayudarme a enfrentar las dificultades que han surgido a lo largo de la realización de este trabajo y por hacerme sentir siempre el cariño de hermanos que existe entre nosotros. Este trabajo es el resultado del esfuerzo de todos juntos.

Con profundo cariño:

Mari.

A MIS ABUELOS SIXTO (†), FORTUNATA, ANTONIO Y FELICITAS:

Por haber creído en mí siempre, por haberme apoyado moralmente en todo momento, por haberme dado palabras de aliento y por haberme dado a unos padres maravillosos. Gracias porque a pesar de la distancia los sentí a mi lado cuando los tiempos eran difíciles.

Con respeto y cariño:

Mari.

A PAN, NICEFORO E ITANDEHUI:

Por el apoyo incondicional que siempre me han brindado, por sus palabras que siempre fueron oportunas para aconsejarme positivamente, por la amistad que existe entre nosotros, han sido parte importante en mi vida y ahora en este trabajo que culmina un esfuerzo de muchos años. Gracias por todo lo que han hecho por mí, siempre lo tendré presente.

Con cariño y aprecio:

Maricela.

A LUZ:

Por todos los momentos buenos y malos que hemos vivido juntas, por el apoyo incondicional que me has brindado, por todo lo que has aportado para que este trabajo llegue a concluirse. Gracias por la paciencia, comprensión y apoyo que son y serán invaluables para mí en todo momento.

Con cariño y aprecio:

Maricela.

A ELIZABETH:

Por el apoyo que me brindaste durante todos estos años que finalmente se concretan con este trabajo. Gracias por haberme acogido con cariño primero como amiga y luego como primas que somos, aprecio el valor de tus palabras en los momentos oportunos. Gracias por tu invaluable amistad.

Con cariño:
Mari.

Agradecemos a las personas que de forma directa e indirecta contribuyeron a la realización de este trabajo. De manera en especial a:

Ing. Melchor Paz Gonzalez
Ing. Miguel Angel Cruz León
Ing. Martín Pérez Mondragón

De igual Manera les damos las gracias a los académicos por sus conocimientos aportados, así como a los compañeros por su valiosa amistad.

Gracias a la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, por habernos dado una oportunidad de superación dentro de sus instalaciones.

Damos gracias a Dios por habernos permitido llegar a alcanzar nuestra meta.

Luz Del Carmen Guzmán Robles
Maricela Morales Hernández

PROLOGO

En estos últimos años el diseño de sistemas digitales, al igual que la computación son temas de vanguardia, que le han sido de gran utilidad a la Ingeniería para poder crear cada vez más productos de mayor calidad y a menor costo.dado que los circuitos digitales se emplean para el diseño de computadoras digitales, calculadoras electrónicas, dispositivos digitales de control, equipo de comunicación digital y entre otras muchas más aplicaciones que requieren hardware digital electrónico, todo esto apoyado con el Diseño Asistido por Computadora (CAD), logrando así la optimización de tiempo, materia prima y recursos tanto humanos como materiales.

La idea por realizar esta tesis, surgió principalmente por los problemas que se presentan al cursar materias relacionadas al diseño de sistemas digitales, y también tomando en cuenta los conocimientos adquiridos en la materia de Temas Especiales , como son: el manejo de paquetería relacionada a los circuitos electrónicos. Por lo que se decidió realizar un trabajo que permitiera conjuntar el diseño de sistemas digitales con software que nos sirviera de apoyo para su mejor realización, y fue así como surgió el diseño "Simulador de memorias EPROM para la implementación de algoritmo de máquina de estados". (SME).

INTRODUCCION

Para la realización de este trabajo denominado "Simulador de memorias EPROM para la implementación de Algoritmo de Máquinas de Estados" se requirió dividirlo en tres partes. La primera parte consta de tres capítulos que describen todos los conceptos fundamentales para el diseño de los sistemas digitales. La segunda parte consta del capítulo cuatro, donde se describe la relación existente entre la computación y el diseño de los sistemas digitales. La tercera parte esta constituida por el diseño e implementación del "simulador de memmorias EPROM. (SME)" con la cual se consolida el presente trabajo, a continuación se menciona el nombre de cada uno de los capítulos así como una breve descripción de los mismos.

En el capítulo I "Clasificación de máquinas de estados", se da un conjunto de definiciones y conceptos básicos de lo que es una máquina de estados, además de describir los tipos y características de la misma con la finalidad de que los temas subsecuentes sean comprendidos en una forma clara y precisa.

El capítulo II "Algoritmo de máquina de estados", describe las diferencias y similitudes entre una carta ASM (Algoritmo de Máquina de Estados) y un diagrama de flujo, así mismo se hacen notar algunas consideraciones que se deben tomar en cuenta durante el desarrollo de un diseño digital. El próposito de este capítulo es proporcionar una idea clara de lo que es una carta ASM y un Diagrama de Flujo, ya que en algunas ocasiones al realizar un diseño no se tiene la seguridad de cual de las dos representaciones debe utilizarse.

En el capítulo III "Implementación de Cartas ASM empleando memorias EPROMs (Memoria Programable Borrable de Sólo Lectura)", se describen las características de una memoria, las técnicas de implementación de cartas ASM empleando este tipo de memorias; así como también se presentan las ventajas y desventajas con respecto a otros dispositivos lógicos programables. Para la implementación de Cartas ASM se presenta la opción de la memoria EPROM sin embargo cabe hacer notar que ésta no es el único dispositivo lógico programable que puede ser utilizado, como podra verse en las paginas de este capítulo. Otros dispositivos igualmente útiles son: GAL (Arreglo Lógico de Compuertas), PAL (Lógica de Arreglo Programable), PLA (Arreglo Lógico Programable), etc. También se incluye en este capítulo los tipos de direccionamiento que pueden ser útiles en el diseño de cartas ASM.

El capítulo IV "Diseño Electrónico Basado en Sistemas CAD|CAM|CAE", presenta la evolución que se ha dado en el desarrollo de esta tecnología así como las definiciones y conceptos básicos relacionados a la misma, describiendo el campo de acción de cada sistema (CAD, CAM, CAE). Se menciona también la integración que han tenido estos tres sistemas dando ejemplos claros de tal integración y aplicaciones concretas. Se da una lista de herramientas computacionales (software) que permiten desarrollar de principio a fin un diseño digital asistido por computadora.

En el capítulo V "Diseño e Implementación del Simulador de Memorias EPROM (SME)", se da una descripción detallada de las fases de diseño utilizadas en la implementación del SME incluyendo tanto el software como el hardware que componen el dispositivo.

En el apéndice A se encontrará la información específica sobre las herramientas computacionales que pueden ser utilizadas en un diseño.

En el apéndice B se describen los dispositivos que se utilizaron para el simulador de EPROMs para la implementación de algoritmo de máquina de estados.

En el apéndice C se presenta el manual de usuario del SME.

INDICE

CAPITULO I

CAPITULO I
I CLASIFICACION DE MAQUINAS DE ESTADOS

I.1 DEFINICION DE MAQUINA DE ESTADO

Un sistema lógico puede estar constituido por varios módulos, los cuales pueden ser representados por un modelo general llamado MAQUINA DE ESTADOS.

Una MAQUINA DE ESTADOS esta constituida por tres elementos, los cuales son : La función de estado-siguiente, el estado y la función de salida. Estos tres elementos permiten describir el comportamiento del módulo en términos de: entradas, salidas y tiempo.

Las entradas y las salidas, que al pasar a través del módulo hacia el mundo exterior y a otros módulos, reciben el nombre también de INSTRUCCIONES.

El ESTADO de una máquina representa la memoria suficiente de condiciones pasadas para determinar su comportamiento del futuro, dicha memoria es comúnmente realizada de circuitos biestables llamados Flip-Flops. En términos de la máquina de estados, un ESTADO representa la suficiente información para poder determinar la salida así como el estado-siguiente de las entradas presentes, lo cual se ilustra en la figura 1.1. Al grupo de flip-flops que forman el estado se les llama REGISTRO DE ESTADO.

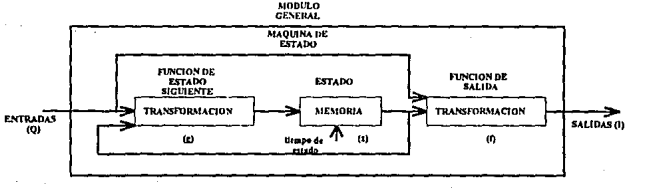


FIGURA 1.1

Un estado se define en forma exclusiva por la combinación de bits que almacena, lo que implica que se tendrán 2^n estados posibles para un registro de estados de "n" flip-flops. A los flip-flops del estado se les llama VARIABLES DE ESTADO y son definidos mediante una declaración de estados. Cada variable de estado recibe un nombre, el cual puede ser A, B ó FF6. El grupo de variables que componen un estado, son ligadas entre sí por una operación de concatenación (~).

Por ejemplo:

Si A, B, C, y D son los nombres asignados de las variables, y proponiendo el ESTADO=D~C~B~C, como una forma de declaración de estados, considerando además que: A=1, B=0, C=1, D=1, se obtiene entonces que nuestro estado en particular queda representado por el código 1101, el cual se obtuvo al situar el valor lógico de cada variable en la posición correspondiente a la declaración de estados propuesta al

inicio del ejemplo, de la misma manera se pueden obtener otros registros en particular.

Cada estado de una máquina tiene un estado-siguiente, el cual es definido por la FUNCION DE ESTADO-SIGUIENTE.

La función de estado-siguiente (g), depende del estado presente (x) y las entradas Q. El tiempo de estado es determinado normalmente por una entrada periódica en el registro de estados. Al final de cada tiempo de estado, el siguiente estado se convierte en el estado presente. Si el tiempo de estado básico es representado por T, y k es un conteo entero, entonces x(kT) representa el estado en el tiempo discreto kT. Empleando esta terminología, la función de estado-siguiente, g, puede ser definida como sigue:

$$x((k-1T)) = g[x(kT), Q(kT)]$$

La notación empleada para definir el estado siguiente, puede ser simplificada utilizando el OPERADOR-RETARDO, el cual es una flecha $\rightarrow$ ó $\leftarrow$ apuntando hacia la dirección del estado-siguiente que va a ser reemplazado.

Por lo tanto, la función de estado-siguiente queda escrita como:

$$x \leftarrow g[x,Q]$$

la cual muestra, que el valor de x es colocado al final de un tiempo de estado, y este nuevo valor queda representado por la expresión g[x,Q], donde: x y Q representan a los valores presentes durante el tiempo de estado. Después del cambio de estado, éste es retardado por la determinación del estado-siguiente, y al operador que realiza todo esto, se le llama OPERADOR-RETARDO.

La FUNCION DE SALIDA es la encargada de generar un conjunto de salidas ó INSTRUCCIONES I, del estado, así como también la información de la entrada para cada estado.

La función de estado-siguiente es semejante, ésta consiste de una operación de transformación llamada f, que tiene la siguiente expresión:

$$[(kT)] = f[x(kT), Q(kT)]$$

donde:

k es el conteo entero

T es el tiempo básico respectivamente, de la misma manera que en la función de estado-siguiente.

La notación utilizada para la función de salida puede ser también simplificada por la utilización del signo (=), el cual es definido como un OPERADOR INMEDIATO que representa una operación en el estado del tiempo presente. Recordando la terminología para la función de salida, en forma simplificada queda expresada como:

$$I = [X, Q]$$

El funcionamiento de una máquina de estados cíclica se lleva a cabo en forma ordenada, alcanza una condición estable durante cada tiempo de estado, kT. El estado, y el estado siguiente así como también las salidas son definidas sólo durante el período del tiempo de estado estable.

La figura 1.2 ilustra la división del tiempo de estado dentro de un período de transición seguido por un período estable. El período de transición es determinado por un circuito de retardos. El tamaño del período estable es determinado calculando la diferencia entre el tiempo de estado y el tiempo de transición.

Observando la figura 1.2 se puede concluir que el tiempo de estado es más grande que el tiempo de transición. La operación de la máquina de estados puede ser visualizada como una serie de pasos que dependen de las salidas para cada tiempo estable.

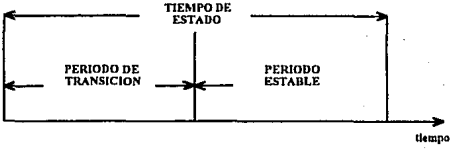


FIGURA 1.2

Por ejemplo:

En la figura 1.3, la notación del tiempo kT es utilizada para representar los cambios y relaciones en las operaciones durante el período estable para tres tiempos de estado sucesivos.

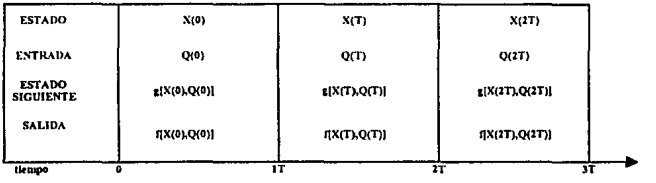


FIGURA 1.3

En la actualidad el diseño de sistemas digitales ha tomado una relevante importancia, por lo tanto existe una gran variedad de información sobre dichos sistemas, la cual se puede obtener de libros, revistas, etc. También es importante mencionar que debido a la gran cantidad de información que se puede obtener de las diversas fuentes, suele suceder que para definir algún concepto se utilizan terminologías distintas, por lo que para reafirmar un poco más el concepto de lo que es una máquina de estados, se plantea otra definición distinta a la descrita al inicio del capítulo.

Una máquina secuencial, es un modelo abstracto que puede ser utilizado para estudiar los circuitos secuenciales. En este modelo, se asume que las entradas están dadas por una secuencia de símbolos seleccionados desde un conjunto finito de símbolos de entrada I, de la misma manera, las salidas están dadas por una secuencia de símbolos seleccionados desde un conjunto finito de salidas Z.

La memoria de la máquina es representada por un conjunto de estados, (uno de estos estados puede ser asumido por la máquina en un instante cualquiera) y por una función de estado-siguiente, la cual define al estado siguiente que la máquina asumirá como una función del estado presente y de la entrada; y una función de salida que indican que la salida es una función del estado presente y la entrada.

En base a lo mencionado anteriormente, se puede decir que una máquina secuencial que tiene un número finito de estados se le denomina MAQUINA DE ESTADOS-FINITOS.

La Máquina de Estados-Finitos M, puede ser expresada como: $M = (I, Z, Q, \delta, \omega)$

donde:

I es el conjunto de símbolos de entrada

Z es el conjunto de símbolos de salida

Q es el conjunto finito de estados

δ es la función de estado-siguiente, el cual es un mapa de IxQ en Q.

ω es la función de salida, el cual es un mapa de IxQ en Z.

Una máquina definida como la antes mencionada es conocida comúnmente como una Máquina Mealy, también otra de las máquinas más conocida es la Máquina Moore que se distingue de la Mealy en que su función de salida esta dada por un mapa de IxQ en Z. Esto se muestra en la figura 1.4.

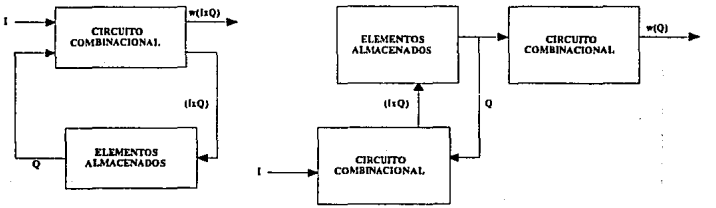


FIGURA 1.4

Por ejemplo:

Si definimos a la máquina de estados finitos como M1, y se le asignan los siguientes valores, se tendrá que:

$$I = \{0,1\} \quad Z = \{0,1\} \quad Q = \{q_0, q_1, q_2, q_3\}$$

$\delta(0,q_0) = q_0 \quad \delta(1,q_0) = q_1$
$\delta(0,q_1) = q_1 \quad \delta(1,q_1) = q_2$
$\delta(0,q_2) = q_2 \quad \delta(1,q_2) = q_3$
$\delta(0,q_3) = q_3 \quad \delta(1,q_3) = q_3$

$\omega(0,q_0) = 0 \quad \omega(1,q_0) = 0$
$\omega(0,q_1) = 0 \quad \omega(1,q_1) = 0$
$\omega(0,q_2) = 0 \quad \omega(1,q_2) = 1$
$\omega(0,q_3) = 0 \quad \omega(1,q_3) = 0$

La tabla de estados para la máquina M1 se ilustra en la figura 1.5.

ESTADO PRESENTE	ESTADO SIGUIENTE valor de entrada		SALIDA PARA ENTRADA	
	0	1	0	1
q0	q0	q1	0	0
q1	q1	q2	0	0
q2	q2	q3	0	1
q3	q3	q3	0	0

FIGURA 1.5

En la tabla de estados, el estado presente, el estado siguiente y las salidas para varias entradas combinacionales son tabuladas en columnas. De esta manera, si hay "n" variables de entrada, entonces se tendrán 2^n columnas de entradas para el estado-siguiente y las salidas.

I.2. TIPOS Y CARACTERISTICAS DE MAQUINAS DE ESTADO.

Existen cuatro lenguajes básicos para la descripción de la operación lógica de módulos de máquinas de estado, estos son: **LA EXPRESION BOOLEANA, LA TABLA DE VERDAD, EL MAPA Y LA CARTA ASM (Algoritmo de Máquina de Estados)**.

La carta ASM describe simultáneamente un algoritmo y una máquina de estado. Estos lenguajes son útiles para las cinco clases de máquina de estado que se describirán posteriormente, por esta razón se da una breve explicación de tres de estos lenguajes que nos son útiles para la descripción de una lógica. Estos lenguajes son los tres primeros citados anteriormente, el cuarto que es LA CARTA ASM tendrá un apartado independiente de los tres primeros dada la importancia que presenta en este trabajo.

♦ EXPRESIONES LOGICAS Y BOOLEANAS.

La lógica es una realidad cotidiana en el mundo que nos rodea. Los conceptos de grupos colectivos se identifican con un palabra dependiendo del tipo de declaración de la que se trate; así las declaraciones que involucran la unión de grupos utilizan la palabra "AND"; las declaraciones condicionales comienzan con las palabras "IF", "WHILE", "FOR" y "WHEN"; las declaraciones alternativas utilizan las palabras "THEN", "ELSE" y "OR"; las declaraciones negativas utilizan la palabra "NOT".

La lógica puede ser usada como una herramienta en el estudio del razonamiento deductivo, también conocido como "Lógica Proposicional".

Uno de los más significativos avances en lógica fue hecho en 1854 cuando Boole postuló que "La estructura del pensamiento lógico podía ser representado mediante símbolos". Su lenguaje lógico simbólico hizo posible la claridad, entendimiento y documentación en el proceso del pensamiento lógico. Los símbolos básicos son: un "1" para VERDADERO y un "0" para FALSO, asimismo existen también muchos símbolos para representar las relaciones entre los hechos de una expresión; las aserciones de estos hechos son representados por letras o grupos de símbolos.

Por ejemplo:

Para que lo mencionado anteriormente se comprenda en forma más clara; la declaración "Cuando los corredores están listos y la pistola dispara, la carrera inicia", puede ser dividida en tres aserciones representadas por las siguientes letras:

A = Los corredores están listos
B = La pistola dispara
C = La carrera inicia

Utilizando un punto (.), para la relación AND, la declaración puede ser representada por las expresiones:

A AND B IGUAL A C
ó
A (.) B = C

De la expresión puede verse que C es verdadero únicamente cuando A y B son ambos verdaderos, que puede ser representado en símbolos, 1(.)1= 1. Las relaciones básicas tal como ésta, son llamadas POSTULADOS. Otras relaciones básicas en lógica son las OR, representadas por una cruz, (+), y el complemento, representado por una barra sobreescrita ($\overline{A}$), que significa NEGACION DE A. A continuación se muestra un conjunto de símbolos lógicos que son utilizados con frecuencia en la lógica booleana.

OPERACION	SIMBOLO	EJEMPLO	RESULTADO
OR LOGICO	+	A + B	1 o 0
AND LOGICO	.	A . B	1 o 0
IGUALDAD	=	A = B	IGUALDAD
COMPLEMENTO LOGICO	$\bar{X}$	$\bar{A}$ $\bar{B}$	1 o 0
SUMA MATEMATICA	(+)	A (+) B	NUMERO
MULTIPLICAR MATEMATICA	(.)	A (.) B	NUMERO
OPERADOR DE RETARDO	$\rightleftarrows$	A $\rightarrow$ B	A TOMA EL VALOR DE B
RESTA MATEMATICA	(-)	A (-) B	NUMERO

En la figura 1.6 se presenta una tabla de postulados para expresiones booleanas, estas relaciones forman la base del ALGEBRA BOOLEANA, que estudia la relación de dos variables valuadas, como las citadas en párrafos anteriores (A,B,etc.).

X = 0 si X ES DIFERENTE DE 1.	X = 1 si X ES DIFERENTE DE 0.
$\overline{0} = 1$	$\overline{1} = 0$
$0 \cdot 0 = 0$	$1 + 1 = 1$
$1 \cdot 1 = 1$	$0 + 0 = 0$
$1 \cdot 0 = 0 \cdot 1 = 0$	$0 + 1 = 1 + 0 = 1$

FIGURA 1.6

♦ TABLA DE VERDAD.

Una tabla de verdad es un lenguaje que nos ayuda a describir todas las relaciones y resultados en una expresión booleana. En la figura 1.7, la tabla 1 muestra una descripción de la expresión A.B=C, la cual fue utilizada en el ejemplo citado en párrafos anteriores, esta tabla describe la función AND entre dos variables, A y B. Las tablas 2 y 3 nos muestran las funciones OR y NOT respectivamente.

VALOR LOGICO DE A	VALOR LOGICO DE B	VALOR LOGICO DE C
0	0	0
0	1	0
1	0	0
1	1	1

1 = VERDADERO 0 = FALSO

TABLA 1

OR LOGICO

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

C = A + B

TABLA 2

COMPLEMENTO

A	C
0	1
1	0

C = A'

TABLA 3

FIGURA 1.7

La Tabla de Verdad presenta una alternativa más de descripción de relaciones lógicas entre variables que tienen un significado y valor en la vida real.

- **EL MAPA DE KARNAUGH.**

El mapa de Karnaugh es una arreglo especial que se presenta en una tabla de tal forma que las entradas representan códigos binarios, éstos códigos difieren únicamente en una variable, es decir, se trata de un código Gray en el cual de posición a posición solo un bit cambia de cero a uno o viceversa. El mapa es en cierto modo el resultado de la visualización del código como un vector con "n" varibles, el cual representa un punto en un espacio de "n" dimensiones, por lo que se puede decir que el mapa de Karnaugh es una representación bidimensional de este espacio. Como una consecuencia de lo expuesto anteriormente los códigos adyacentes son llamados códigos de DISTANCIA UNITARIA.

Un mapa se dibuja usando un proceso de desdoblamiento que proyecta el espacio n_dimensional en un trazo bidimensional bien descrito, cada esquina en el trazo corresponderá a un código o vector, y la información escrita en esa esquina corresponderá a algún valor característico asociado con ese vector. En la figura 1.8 se describe el traslado de un cubo tridimensional, formado por un vector tridimensional [C B A], en un mapa con campos identificados como A, B y C, en el mapa todas las esquinas se situan dentro de un campo designado por una variable que tiene un código cuyo valor de la variable es igual a 1, fuera del campo escogido, la misma variable es igual a 0.

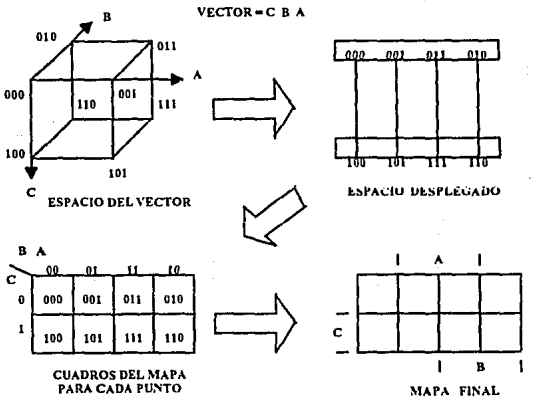


FIGURA 1.8

Por ejemplo:

El mapa de tres variables para el cubo tridimensional tiene a "C" designando el campo de los cuatro cuadros inferiores, en cada cuadro C=1; por otro lado, en los cuatro cuadros superiores C=0; "A" designa los cuatro cuadros del centro, para todos los cuadros en este campo A=1. Por la forma en que el cubo tridimensional es desdoblado, los códigos para los cuadros superiores e inferiores, a la izquierda y derecha de cualquier cuadro tienen una distancia unitaria desde el código en tal cuadro, incluyendo todos los cuadros sobre el final, porque el mapa representa un cubo desdoblado de manera que la columna izquierda está a una distancia unitaria de la columna derecha.

La figura 1.9 muestra las representaciones de mapas de varios cubos n_dimensionales, y se presentan para dar una idea más clara de cómo representar en un mapa de Karnaugh "n" variables.

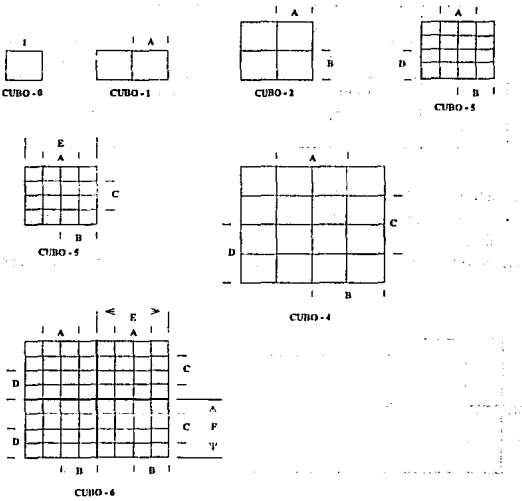


FIGURA 1.9

Existe también la carta ASM, la cual es muy utilizada para la representación de máquinas de estado en una forma clara e inteligible. Esta forma de descripción es definida en el siguiente capítulo del presente trabajo.

♦ TIPOS DE MAQUINAS DE ESTADOS.

Para poder optimizar las características que ofrece el modelo general de una máquina de estados, es preciso fragmentar este modelo partiendo de una estructura básica en la que no desaparezca el concepto de Máquina de Estado; al irle sumando características se llegará finalmente al modelo más general y completo. Con esto se podrá comprender mejor cada uno de los tipos de máquinas con sus correspondientes características.

Partiendo de lo anterior se tienen cinco tipos o clases de máquina de estados con su lógica propia, así que a cada clase se le ha dado un nombre para poder identificarla por su función, a continuación se resumen:

CLASE 0	COMBINACIONAL
CLASE 1	RETARDO
CLASE 2	TRANSICION DE ESTADO DIRECTO Y SALIDA DE ESTADO
CLASE 3	TRANSICION DE ESTADO CONDICIONAL Y SALIDA DE ESTADO
CLASE 4	TRANSICION DE ESTADO CONDICIONAL Y SALIDA DE ESTADO CONDICONAL

• CLASE 0: LOGICA COMBINACIONAL.

DEFINICION: Las máquinas de estado de clase 0 tienen salidas, que son función de las entradas solamente, I=f[Q].

La parte de la máquina de estado general incluida en la clase 0 es la que se muestra en la figura 1.10.

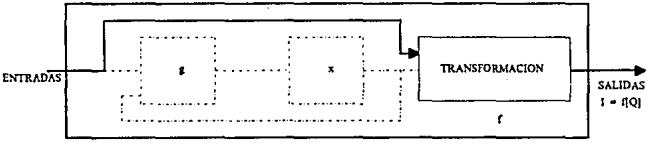


FIGURA 1.10

CARACTERISTICAS: Cualquier máquina de clase 0 se puede describir mediante una carta ASM a través de un bloque ASM. El estado simple de este bloque no tiene otro significado particular que representar que la función de estado siguiente es g[X,Q]= 1. Las cajas de condición describen la función útil. Las salidas utilizables son siempre salidas condicionales porque con un estado simple ninguna salida de estado cambiaría. En la figura 1.11 la caja de decisión simple contiene la función completa para esta máquina,

mientras la figura 1.12 nos muestra una representación equivalente en la cual para cada variable de entrada es dada una caja de decisión separada. En la misma representación IH es dado cuando B + (E . C) es verdadero. De esta manera la carta ASM se convierte en una herramienta muy útil cuyo valor va creciendo en la medida en que la máquina de estado se complica.

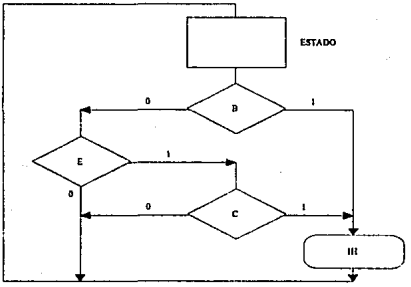


FIGURA 1.11

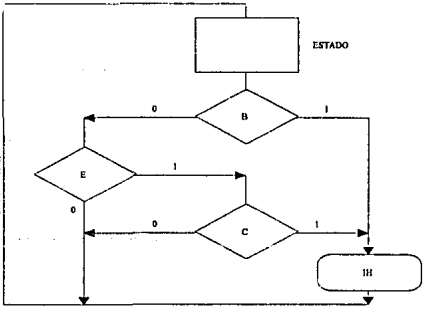


FIGURA 1.12

Para una máquina de Clase 0, no solamente existe la descripción mediante carta ASM, también contamos con la tabla de verdad, esta tabla se diseña listando las salidas condicionales que resultan de todas las posibles combinaciones de entradas; así pues la tabla puede ser generada siguiendo el trazo de

las rutas de liga de la carta ASM para cada combinación de entradas. Cuando nosotros desconocemos las funciones que describen una máquina, la tabla es un buen recurso que nos ayuda a describir estas funciones.

De forma similar podemos utilizar los mapas de Karnaugh para la descripción de la Clase 0, los mapas se forman introduciendo la salida correspondiente a cada combinación de entradas, cada cuadro del mapa corresponde a una combinación de entradas. Del mismo mapa, se puede obtener la ecuación de la máquina de clase 0 en términos de las entradas, a través de algún método de reducción ya sea considerando los 0's ó 1's, de este modo tenemos la expresión booleana, que nos permite tener la misma información que se tiene a través de los mapas y tablas.

- **CLASE 1: MAQUINAS DE RETARDO.**

DEFINICION: La máquina de Clase 1 es conocida como máquina de retardo y se describe como un circuito combinacional con tiempo de retardo de estado sumado.

La ECUACION PARCIAL ESTADO-SIGUIENTE es utilizada para describir el comportamiento del estado-siguiente.

CARACTERISTICAS: La figura 1.13 muestra una máquina de estado de clase 1, ésta consiste de una memoria y dos elementos de transformación. Como la máquina no tiene retorno interno, se puede escribir una ecuación para describir la conducta que involucra únicamente las entradas y las salidas. Esta ecuación es I ←fg[Q] la cual no involucra el estado interno, X. Así pues la máquina de clase 1 puede ser considerada como una función combinacional, I=fg, que produce una salida retardada por un periodo de tiempo que es igual al tiempo que dura un estado, a diferencia de la clase 0 que produce una salida en el tiempo del estado presente.

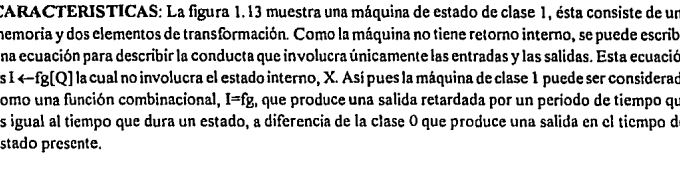


FIGURA 1.13

En la representación de la carta ASM de la máquina de clase 1, las dos funciones f y g son descritas para mostrar que el estado siguiente depende sólamente de las entradas y la salida depende sólamente del estado.

Existe un bloque ASM para cada estado de X. La transformación de las entradas para el estado siguiente es descrita por la estructura de la caja de condición; como existe únicamente una función de estado siguiente y depende sólamente de las entradas, la estructura de la caja de condición es compartida por todos los estados y las salidas son indicadas mediante una lista en cada estado. La carta ASM no es muy utilizada para describir una máquina de clase 1. La figura 1.14 muestra una carta ASM de una máquina de clase 1.

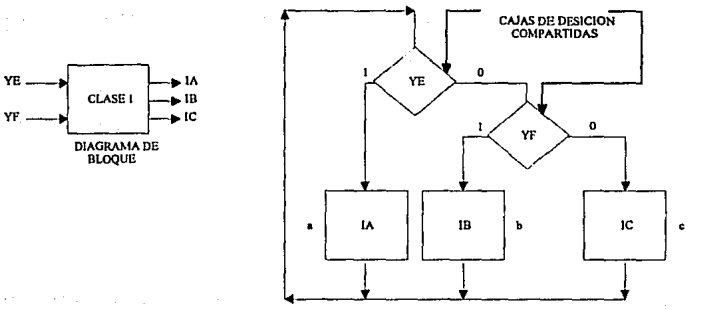


FIGURA 1.14

La tabla de verdad para la máquina de clase 1 es idéntica a la de la clase 0, excepto que las salidas ocurren en el estado siguiente. Como es de suponerse también es posible describir la clase1 mediante mapas, estos relacionan el conjunto de entradas presentes con las salidas del estado siguiente, así que por cada salida se debe dibujar un mapa. Cuando se trata de representar la clase1 con una ecuación, se hace presente el operador retardo para denotar que las salidas ocurren en el estado siguiente; la ecuación es muy útil sobre todo porque describe convenientemente el registro, el cual es una parte importante en la máquina de retardo. Este registro es una colección de elementos de retardo simples que almacena un vector de entrada, en un registro simple, la salida es la misma, que la entrada retardada un período de tiempo igual a la duración de un estado.

Es importante mencionar dos conceptos, señales SINCRONAS y ASINCRONAS. Las señales síncronas son aquellas cuyo comportamiento es bien conocido en instantes discretos de tiempo; mientras las señales asíncronas cambian y pueden ser afectadas en cualquier instante de tiempo. Estos términos están directamente involucrados con las clases de máquina siguientes por el concepto que cada una de ellas maneja.

- **CLASE 2: TRANSICION DE ESTADO DIRECTO Y SALIDA DE ESTADO.**

DEFINICION: Una máquina de clase 2 es una porción de una máquina de estado general que tiene su estado siguiente determinado sólo por el estado presente, $X \leftarrow g[X]$, y las salidas determinadas sólo por el estado presente, $I = f[X]$.

La figura 1.15 muestra una máquina de clase 2, la cual es indicada con líneas sólidas.

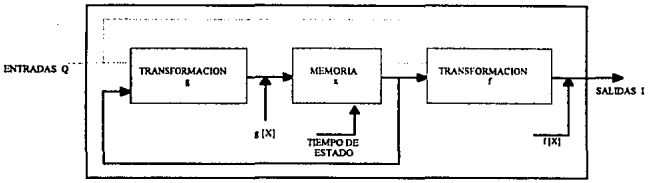


FIGURA 1.15

CARACTERISTICAS: Los contadores síncronos son un ejemplo de máquinas de clase 2. El concepto importante es la idea de una secuencia de estados.

La figura 1.16 muestra una descripción de una carta ASM del contador de décadas.

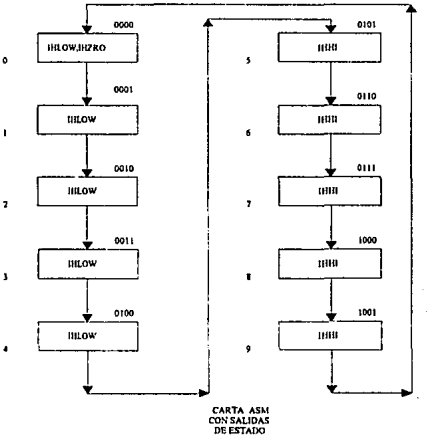


FIGURA 1.16

Es sencillo observar que cada estado tiene un sólo estado siguiente y que los estados están en una SECUENCIA. Una transición desde un estado al siguiente es realizada en cada pulso de reloj, el cual define el tiempo de estado.

Un contador de décadas cuenta en relación al número de entradas de reloj, esto es recordado por el estado; pero, después de contar 10 estados diferentes reinicia su cuenta, repitiéndose así todos los estados. Estas secuencias de repetición de los estados son llamadas CICLOS. La ausencia de algunas cajas de condición en la carta ASM resultan de los requerimientos de la clase 2, ésto es que el estado siguiente es una función únicamente del estado presente.

Es posible también describir la máquina de clase 2 mediante una tabla de verdad y es importante considerar que en una función de estado siguiente, cada estado presente tiene únicamente un estado siguiente. Los mapas para cada una de las variables pueden construirse a partir de la tabla de verdad antes citada; de igual manera se cuenta con herramientas para la obtención de la ecuación representativa en esta clase de máquinas, la reducción a partir de los mapas de Karnaugh no es evidente pero si es posible.

- **CLASE 3: TRANSICION DE ESTADO CONDICIONAL Y SALIDA DE ESTADO.**

DEFINICION: En la máquina de clase 3 el estado siguiente es determinado tanto por el estado presente como por las entradas, g[X,Q], y las salidas están determinadas únicamente por el estado presente, f[X]. Las máquinas de clase 3 son capaces de seleccionar entre secuencias de estado alternativas y es posible desarrollar cualquier algoritmo con ella pues es una máquina completa, la máquina de clase 4 sólo introduce algunas simplificaciones al proporcionarnos ayuda en la descripción de un algoritmo con menos estados.

La figura 1.17 muestra la máquina de clase 3, representada como una parte del modelo general de máquina de estado.

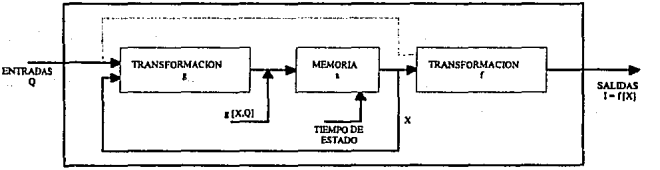


FIGURA 1.17

CARACTERISTICAS: La máquina de transición condicional puede ser descrita por cualquiera de los cuatro lenguajes citados al iniciar este tema. Para poder comprender con claridad las características de la clase 3, citaremos un ejemplo. En la figura 1.18 se muestra una máquina simple de clase 3, la cual cuenta cíclicamente ya sea 5 u 8, dependiendo de la entrada YC8. Si observamos la figura, nos daremos cuenta que existe una transición condicional del estado "a" hacia el "b" ó
"e", todas las demás transiciones son directas.

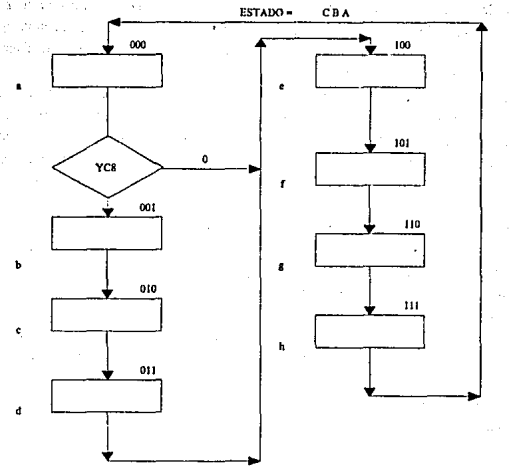


FIGURA 1.18

La tabla de verdad que describe este ejemplo (ver figura 1.19) tiene una columna de entradas, además de las columnas de estado presente y estado siguiente; normalmente se tiene también una columna de salidas pero para este ejemplo no hay descripción de salidas; a cada estado se le debe asignar un código de estado antes de que sea hecha la descripción del contador mediante un mapa de Karnaugh, este proceso es llamado ASIGNACION DE ESTADO.

ENTRADA, Q	ESTADO, X	ESTADO SIGUIENTE, G(X,Q)
YC8	a	b
$\overline{YC8}$	a	e
...	b	c
...	c	d
...	d	e
...	e	f
...	f	g
...	g	h
...	h	a

FIGURA 1.19

Como en los casos anteriores, es posible la representación de la máquina de clase 3 mediante mapas y lo que se hace para obtener estos mapas es utilizar los estados siguientes parciales para cada una de las variables de estado. Por lo tanto para nuestro caso del contador que puede ser alterado, la representación de estos mapas se observa en la figura 1.20.

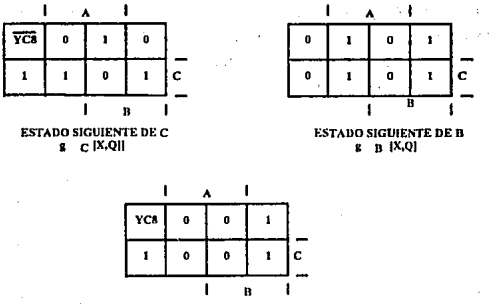


FIGURA 1.20

- **CLASE 4: TRANSICION DE ESTADO CONDICIONAL Y SALIDA DE ESTADO CONDICIONAL.**

DEFINICION: La máquina de clase 4 tiene estados internos que constan de estado siguiente y salidas; ambos están determinados por las las entradas y el estado presente. Este tipo de máquina es igual a una máquina de estado general, la cual es descrita por un diagrama de bloques en la figura 1.21.

FIGURA 1.21

CARACTERISTICAS: Ya que la función de estado siguiente en esta máquina tiene la misma forma que la de la máquina clase 3, la única parte que será descrita aquí es la salida de estado condicional.

Una descripción de carta ASM de la máquina de clase 4 es muy similar a la de la máquina de clase 3 pero existe la diferencia de que en la clase 4 se tienen salidas condicionales en la estructura de liga del bloque ASM; esto significa que un estado puede producir varias salidas diferentes en función de las entradas, I = f[X,Q]; y esta característica con frecuencia nos ayuda a describir un algoritmo con menos estados pero no con menos salidas de estado. La figura 1.22 proporciona una comparación de los efectos de una salida de estado y una salida condicional.

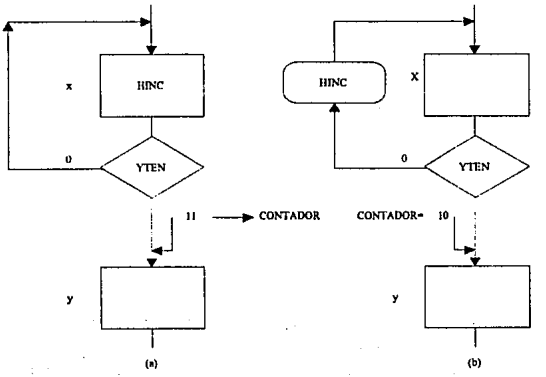


FIGURA 1.22

La ventaja que nos ofrece la salida condicional es que es posible modificar las instrucciones en base a las entradas, con ello tendremos una mayor flexibilidad en la formulación de instrucciones, lo cual nos explica porque un diseño puede con frecuencia ser hecho en menos estados cuando se utilizan salidas condicionales.

RESUMEN DE TIPOS DE MAQUINAS.

CLASE	FUNCIONES DE LA MAQUINA DE ESTADO	NOMBRE DE LA CLASE DE MAQUINA
CLASE 0	$I = f[Q]$ $X <-- I$	SALIDA COMBINACIONAL
CLASE 1	$I = f[X]$ $X <-- g[Q]$	RETARDO
CLASE 2	$I = f[X]$ $X <-- g[X]$	SALIDA DE ESTADO, TRANSICION DE ESTADO DIRECTO.
CLASE 3	$I = f[X]$ $X <-- g[X,Q]$	SALIDA DE ESTADO, TRANSICION DE ESTADO CONDICIONAL.
CLASE 4	$I = f[X,Q]$ $X <-- g[X,Q]$	SALIDA DE ESTADO CONDICIONAL, TRANSICION DE ESTADO CONDICIONAL.

CAPITULO II

CAPITULO II
II ALGORITMO DE MAQUINA DE ESTADOS (ASM)

II.1 CARTA PARA ALGORITMO DE MAQUINA DE ESTADOS (ASM) Y DIAGRAMA DE FLUJO

♦ **CARTA ASM.**

La carta ASM es un tipo especial de diagrama de flujo adecuado para describir las operaciones en un sistema digital. Esta carta, describe la secuencia de eventos; así como también las relaciones de tiempos entre los estados de un controlador secuencial y los eventos que ocurren cuando pasa de un estado al estado-siguiente. Esta adaptada para especificar con precisión la secuencia de control y las operaciones de procesamiento de datos en un sistema digital, tomando en cuenta las restricciones que se pueden presentar en el hardware digital.

El diagrama está compuesto de tres elementos básicos que son: **la caja de estado, la caja de decisión y la caja condicional.**

La ejecución de un algoritmo de máquina de estados (ASM) se lleva a cabo por medio de una secuencia de estados, los cuales se basan en la posición del control en el algoritmo (el estado), y los valores relevantes de las variables de estado.

Un **estado** en la secuencia de control se indica por medio de una **caja de estado**, que tiene una forma rectangular dentro de la cual se escriben operaciones de registro o el nombre de la señal de salida que el control genera mientras se encuentra en este estado. El estado recibe un nombre simbólico, que se coloca en la esquina superior izquierda de la caja. El código binario asignado al estado se coloca en la esquina superior derecha. Esto se ilustra en la figura 2.1.

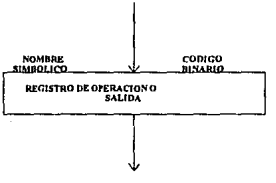


FIGURA 2.1

La **caja de decisión** describe el efecto de una entrada en el subsistema de control. Es una caja en forma de rombo con dos o más trayectorias de salida, como se muestra en la figura 2.2. La condición de entrada que va a probarse está escrita dentro de la caja. Una trayectoria de salida se toma si la condición es cierta y la otra cuando la condición es falsa. Cuando una condición de entrada está asignada a un valor binario, las dos trayectorias se indican por 1 y 0.

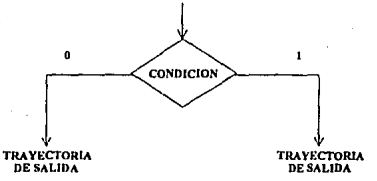


FIGURA2.2

La caja condicional, es de uso exclusivo para la carta ASM. Su representación se muestra en la figura 2.3. Los lados redondeados son la diferencia respecto a la caja de estado. La trayectoria de entrada a la caja condicional va dirigida desde una de las trayectorias de salida de una caja de decisión. Las operaciones de registro o salidas listadas dentro de la caja condicional se generán durante un estado dado siempre que se satisfaga la condición de entrada.

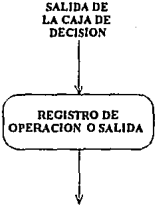


FIGURA2.3

Un bloque ASM es una estructura que consta de una caja de estado y todas las cajas de decisión y condicionales conectadas a sus trayectorias de salida. Un bloque ASM tiene una entrada y cualquier número de trayectorias de salida representadas por la estructura de las cajas de decisión. **Por lo que una CARTA ASM consta de uno o más bloques interconectados.** Cada bloque en una carta ASM especifica las operaciones que van a realizarse durante un pulso común de reloj.

Por ejemplo:

Considerando el bloque ASM de la figura 2.4, su representación gráfica de la transición de estados se muestra en la figura 2.5.

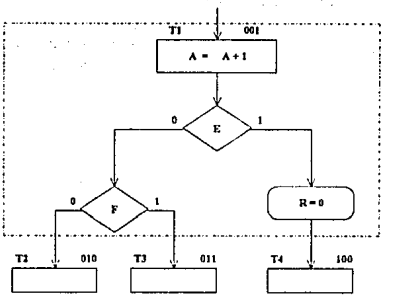


FIGURA 2.4

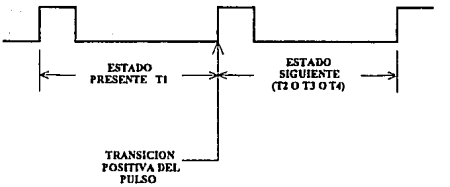


FIGURA 2.5

Esta figura 2.5 permite visualizar y comprender mejor lo mencionado anteriormente, partiendo de un disparo de borde positivo, la primera transición positiva del reloj transfiere el circuito de control al estado T1, mientras se encuentre en éste estado, los circuitos de control verificarán las entradas E y F para poder así generar las señales adecuadas. Las operaciones de incrementar el registro A, limpiar el registro R si E=1 y transferir el control al estado siguiente, T2,T3 ó T4 dependiendo de los valores asignados a E y F; se llevan a cabo durante la siguiente transición positiva del pulso de reloj.

- **COMPARACION ENTRE UNA CARTA ASM Y UN DIAGRAMA DE ESTADOS.**

Una CARTA ASM es muy similar a un diagrama de estado. Cada bloque de estado es equivalente a un estado en un circuito secuencial. La caja de decisión es equivalente a la información binaria escrita a lo largo de las líneas dirigidas que conectan dos estados en un diagrama de estado. Por lo tanto, algunas

veces es conveniente convertir la carta ASM en un diagrama de estado y entonces emplear los procedimientos de circuito secuencial para diseñar el control lógico.

Las ilustraciones siguientes permiten visualizar más claramente las comparaciones anteriores. La figura 2.4 muestra una carta ASM, y la figura 2.6 muestra la carta ASM anterior ahora como un diagrama de estados.

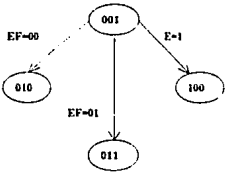


FIGURA2.6

Conceptualmente la carta ASM expresa una secuencia de intervalos de tiempo de una manera precisa, mientras que el diagrama de flujo de software solamente describe la secuencia de eventos y no sus tiempos de duración.

♦ DIAGRAMA DE FLUJO

Un diagrama de flujo es una forma adecuada de especificar una secuencia ordenada de operaciones. Y este consta de bloques conectados por líneas con sentido. Las operaciones se realizan dentro de los bloques especificados y las líneas con flecha entre bloques definen el camino a seguir entre una operación y la siguiente.

Existen básicamente dos tipos de bloques en el diagrama, los cuales son:

a) el bloque rectangular; en el que se especifican las operaciones que deben realizarse y,
b) las cajas en forma de rombo; en donde se toman decisiones basadas en condiciones que se especifican dentro de dichos bloques.

El diagrama de flujo es análogo al diagrama de estados, donde cada uno de los estados es equivalente a un bloque rectangular en un diagrama de flujo, y es éste el lugar donde se realiza una operación determinada. Y las señales lógicas asignadas a los caminos de transición en el diagrama de estados son equivalentes al contenido de los bloques de decisión del diagrama de flujo.

Por ejemplo

En la figura 2.7 se ilustra el diagrama de flujo, que describe el proceso de multiplicación por suma y desplazamiento.

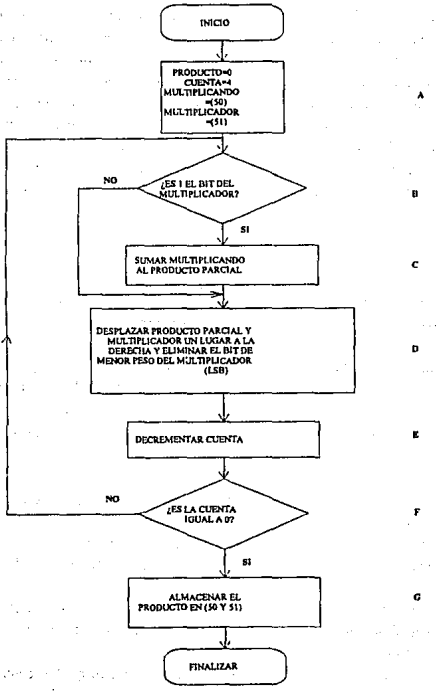


FIGURA2.7

El primer bloque rectangular A, especifica las condiciones iniciales que deben establecerse antes de que el programa empiece a ejecutarse. En éste bloque. lo que se esta indicando es que el contenido del registro del producto se debe poner a cero; y un valor de 4, que indica el número de desplazamientos que van a realizarse, debe colocarse en otro registro. Adicionalmente, el multiplicando y el multiplicador deben colocarse en las posiciones de memoria 50 y 51, respectivamente.

El segundo bloque en forma de rombo B, es una caja de decisión que cuenta con dos salidas, y funciona de la siguiente manera: si el bit del multiplicador es 1 la decisión toma el atributo de "si", apareciendo la siguiente operación en la caja C, en donde se lleva a cabo la operación de suma, sumándose el multiplicando al contenido del registro de producto para dar lugar al siguiente producto; y si el bit del multiplicador es 0, la decisión en "no" y se lleva a cabo un salto que omite a la caja C, permitiendo que los dos caminos se junten y se pueda llevar a cabo la operación de la caja C, en la cual tiene lugar el proceso de desplazamiento. Ya que se ha realizado una suma y un desplazamiento, o bién solo un desplazamiento, el número almacenado en el registro de conteo debe decrementarse, esta operación se lleva a cabo en la caja rectangular E. Después de decrementar el conteo se pasa a la caja de decisión F, y si la cuenta no es cero se ejecuta un salto y se vuelve a pasar por el bucle que contiene las cajas BCDEF. Por el contrario si la cuenta es cero, entra a la caja de operaciones G, en donde se almacena el resultado de la multiplicación en las posiciones de memoria 50 y 51.

Existe un gran número de definiciones con respecto a lo que es un Diagrama de Flujo, las que se mencionan a continuación permitiran comprender un poco más sobre dichos diagramas.

- Un diagrama de flujo representa la información contenida en la tabla de flujo, la cual contiene los estados iniciales y finales del sistema así como los pasos intermedios ó transiciones; ésta representación se lleva a cabo asignando un punto del plano a cada estado interno y se indican mediante flechas las transiciones entre estados, asignándoles los estados de entrada y salida correspondientes a cada transición.

- El diagrama de flujo para un algoritmo de hardware traduce la estipulación en palabras a un diagrama de información que enumera la secuencia de las operaciones junto con las condiciones necesarias para su ejecución. Un diagrama especial de flujo que ha sido desarrollado específicamente para definir algoritmos de hardware digitales se denomina "Diagrama Algoritmico de Máquina de Estado (ASM)".

El diagrama ASM se asemeja a un diagrama convencional de flujo, pero se interpreta en forma diferente. Un diagrama convencional de flujo describe la secuencia de los pasos de procedimiento y las trayectorias de decisión para un algoritmo sin ocuparse de las relaciones en el tiempo.

A diferencia el diagrama ASM describe la secuencia de eventos, lo mismo que las relaciones de tiempos entre los estados de un controlador secuencial y los eventos que ocurren cuando pasa de un estado al siguiente.

II.2. CONSIDERACIONES.

Dentro de la representación diagramática de las cartas ASM, existen algunas consideraciones importantes de mencionar. Una de ellas es la restricción casi obvia que involucra la interconexión de bloques. Esta restricción es que debe haber únicamente un estado siguiente y un conjunto de entradas estables (síncronas) por cada estado presente.

Por ejemplo:

En la figura 2.8 se muestra una transición de estado que es indefinida para un vector de estado simple, porque son especificados dos estados siguientes simultáneamente, por lo que podemos concluir que una máquina simple no puede estar en dos estados simultáneamente como esta carta pareciera implicar. Similarmente, si las cajas de decisión son estructuradas de forma que dos estados siguientes sean indicados simultáneamente para cualquier estado simple y un conjunto de entradas, entonces esta estructura también es indefinida.

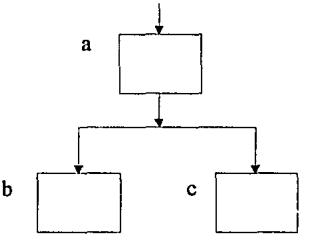


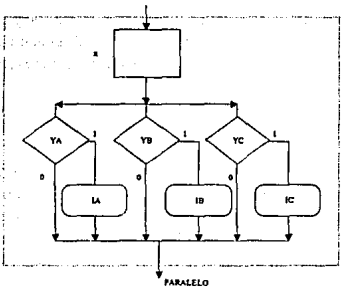
FIGURA 2.8

La restricción de un estado siguiente simple es realmente una extensión lógica de la necesidad de expresar un algoritmo que tiene cada paso bien definido.

Conexiones en serie y paralelo de calificativos múltiples en un bloque ASM surgen como dos formas equivalentes de descripción.

Por ejemplo:

En la figura 2.9 se muestra la descripción de un bloque ASM utilizando cada forma. Ambas descripciones producen la misma operación de máquina. La elección de la forma está basada en la facilidad de la interpretación en una aplicación particular. De cualquier modo, si se tiene duda de la existencia de un estado siguiente simple cuando se utiliza conexión en paralelo, siempre es conveniente regresar a la conexión en serie para verificar la descripción. Una conexión en serie estricta nunca permite más de un estado siguiente para cualquier condición de calificativos.



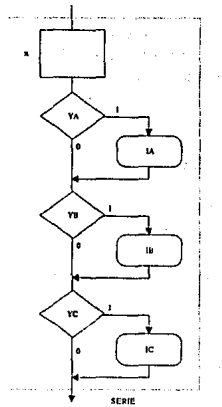


FIGURA2.9

También se debe tener cuidado para evitar algunas estructuras confusas.

Por ejemplo:

En la figura 2.10 se describe un bloque ASM que tiene una caja de condición que se enlaza a sí misma, aunque el concepto es claro, la descripción literal del estado siguiente es confusa cuando YQ1=0. La caja de condición apunta a si misma y ésta no es un estado. Este ejemplo conduce a una restricción más sobre una carta ASM, que puede entenderse como:

a) cualquier ruta representada por un conjunto de cajas de condición debe conducir a un estado,

b) de otra manera el arreglo de cajas de condición y salidas condicionales en un bloque ASM es flexible completamente.

Otra restricción a considerar es que si existe más de una caja de decisión, de las cuales la salida al estado siguiente de cada una de ellas, pasa al mismo estado, esto no es válido, ya que el algoritmo por definición no puede sensar dos variables de entrada simultáneamente.

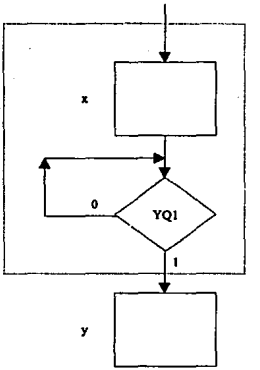


FIGURA 2.10

CAPITULO III

CAPITULO III
III IMPLEMENTACION DE CARTAS ASM (ALGORITMO DE MAQUINA DE ESTADOS) EMPLEANDO MEMORIAS EPROM.

III.1 CARACTERISTICAS DE UNA MEMORIA

La memoria **EPROM**, es un tipo de memoria **ROM** con algunas características adicionales que la hacen ser más versátil, es por esto que a continuación se empezará a describir el funcionamiento de una ROM así como todas sus características y posteriormente se describirá la EPROM agregando sus ventajas de utilización en el diseño de circuitos.

Una ROM es un dispositivo de memoria, en el cual se puede almacenar una cierta cantidad de información binaria. Esta información debe ser especificada por el usuario para posteriormente insertarla en la unidad de memoria y formar así un patrón requerido de interconexión.

Para una interconexión deseada de un diseño en particular, requiere que se rompan ciertos eslabones de la ROM para poder de esta manera formar las trayectorias del circuito necesario. Una de las características de la ROM es que una vez establecido el patrón, éste permanecera fijo.

Por ejemplo:

En la figura 3.1 se muestra un diagrama de bloque de una ROM, el cual consta de "n" líneas de entrada y "m" líneas de salida.

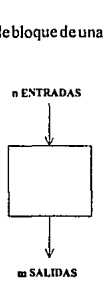


FIGURA 3.1

A cada combinación de bits de las variables de entrada se le da el nombre de DIRECCION, mientras que a la combinación de bits que resulta de las líneas de salida se le denomina PALABRA, esto implica que una dirección será un número binario que representará a un mintérmino de las "n" variables y la cantidad de direcciones distintas que se pueden obtener con las "n" variables de entrada será 2^n.

El número de bits por palabra es igual al número de líneas de salida "m". Una palabra de salida puede seleccionarse por medio de una dirección única, puesto que en una ROM hay 2^n direcciones distintas.

Por lo anterior una ROM se caracteriza por el número de palabras 2^n y el número de bits por palabra "m". Algunas veces la ROM se especifica por el número total de bits que contiene, esto es 2^n x m.

Por ejemplo:

Una ROM de 2048-bits puede organizarse en 512 palabras de 4 bits cada una, lo que significa que la unidad tendrá 4 líneas de salida y 9 líneas de entrada para poder especificar $2^9 = 512$ palabras, y el número total de bits almacenados en la unidad es $512X4 = 2048$.

Internamente una ROM es un circuito combinacional con compuertas AND, las cuales se encuentran conectadas como un decodificador y un número de compuertas OR que es igual al número de salidas. En la figura 3.2 se muestra una memoria ROM de 32 X 4.

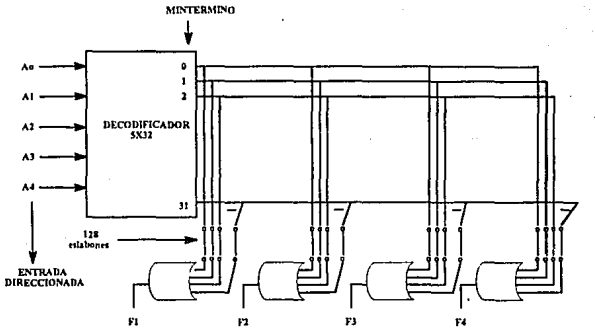


FIGURA3.2

La implementación de la ROM consta de dos niveles realizada en la forma de suma de mintérminos. Cabe aclarar que no necesariamente tiene que ser una de tipo AND-OR, ya que ésta puede ser cualquier otra implementación de mintérminos en dos niveles. El segundo nivel por lo general es una conexión de lógica alambrada, la cual facilita la fusión de eslabones.

Las trayectorias requeridas en una ROM se pueden programar de dos maneras distintas:

a) la primera se le conoce como programación enmascarada que es realizada por el fabricante durante la última etapa del proceso de producción de la unidad.

El procedimiento necesario para llevar a cabo la programación enmascarada de la ROM, requiere primeramente que el diseñador llene la tabla de verdad que él desea que satisfaga la ROM, posteriormente el fabricante realiza la máscara correspondiente para las trayectorias y así producir los 0', y 1', de acuerdo con la tabla de verdad proporcionada por el diseñador. Todo este procedimiento resulta ser costoso, debido a que el fabricante carga honorarios especiales al diseñador por la mascara realizada. Por esta razón, la programación enmascarada sólo se recomienda si van a fabricarse grandes cantidades de la misma configuración para la ROM.

Si la cantidad a fabricarse de ROM con una misma configuración es pequeña, entonces se recomienda utilizar la segunda manera de programar una ROM la cual es la siguiente:

b) Memoria programable de solo lectura, o también conocida como PROM, esta resulta ser más económica.

Los eslabones en este tipo de memoria PROM se rompen por medio de la aplicación de pulsos de corriente aplicados a través de las terminales de entrada. El significado que se les da a los eslabones es: un eslabón roto define un estado binario y un eslabón sin romperse representa al otro estado, ésta forma sencilla de programar la memoria PROM permite al usuario programarla en su propio laboratorio (el cual debe contar con un programador de PROM's) logrando así las relaciones deseadas entre las direcciones de entrada y las palabras almacenadas.

Otra característica de las ROM y PROM es que una vez llevado a cabo el procedimiento de hardware para programarlas éste es irreversible, esto es; los patrones son permanentes y no pueden alterarse, debido a que se estableció un patrón de bit y por lo tanto la memoria no permitiría modificar el patrón de bits.

Otro tipo de memoria ROM es la llamada **PROM borrable o EPROM**, esta puede reestructurarse a su valor inicial (todos 0', o todos 1',) aun cuando se hayan cambiado previamente. Este tipo de memorias puede ser sometido a una luz ultravioleta especial en un período de tiempo (de aproximadamente 15 a 20 minutos), para que la radiación de onda corta realice la descarga en las compuertas internas que sirven como contactos, y después de realizado este borrado, la ROM regresa a su estado inicial y puede volver a ser programada.

Otro tipo de memorias ROM son las llamadas ROM alterables eléctricamente o EAROM, denominadas así porque pueden borrarse con señales eléctricas en lugar de luz ultravioleta.

Las funciones de una ROM, pueden interpretarse en dos formas diferentes. Una función es que mediante la ROM se puede implementar cualquier circuito combinacional, y la otra función es que la ROM se puede considerar como una unidad de almacenamiento que tiene un patrón fijo de cadenas de bits denominadas palabras.

Las ROM se utilizan mucho para implementar circuitos combinacionales complejos de una manera directa desde sus tablas de verdad. También son muy útiles en la conversión de códigos, para implementar funciones aritméticas, así como también para el diseño de unidades de control de sistemas digitales.

III.2 TECNICAS DE IMPLEMENTACION DE CARTAS ASM EMPLEANDO MEMORIAS PROM.

Las memorias PROM son utilizadas para implementar máquinas de estado de clases 2, 3 y 4, esto se consigue almacenando la función de estado siguiente y la función de salida en la memoria, además de diseñar una lógica externa que ejecute la información almacenada. El almacenamiento de la información debe seguir ciertas reglas y una estructura de palabra definida para poder así tener una óptima organización de la memoria.

Existen varias estructuras de palabra, dentro de ellas cabe mencionar las siguientes:

a) DIRECCIONAMIENTO RUTA-LIGA O POR TRAYECTORIA.
b) DIRECCIONAMIENTO ENTRADA-ESTADO.
c) DIRECCIONAMIENTO IMPLICITO.
d) DIRECCIONAMIENTO DE FORMATO VARIABLE.

Cada una de estas estructuras tiene una utilidad específica por las características que las definen, sin embargo en un momento dado cualquiera de estas estructuras puede ser aplicable a todos los tipos de máquinas anteriormente citadas, aún cuando la implementación sea compleja. A continuación se dará una explicación de cada estructura y en que consisten éstas.

♦ **DIRECCIONAMIENTO RUTA-LIGA O POR TRAYECTORIA.**

El direccionamiento por trayectoria está basado en el almacenamiento del estado siguiente y la salida para cada ruta de liga en las cartas ASM. El fragmento de la palabra de la PROM que corresponde al estado siguiente es llamado LIGA; el fragmento de la palabra de la PROM que corresponde a las salidas se le llama INSTRUCCION. Cada dirección es una función del estado presente y de las entradas; tal dirección es llamada una RUTA-LIGA o una TRAYECTORIA. En general cualquier máquina descrita por una carta ASM puede ser implementada con esta estructura.

Por ejemplo:

La figura 3.3 describe un direccionamiento por trayectoria y muestra una posible trayectoria en la carta ASM.

En este tipo de direccionamiento las salidas son trayectorias en lugar de salidas de estado. Las salidas corresponden a salidas condicionales en las rutas de salida de cada bloque de la carta ASM. Cualquier salida en un bloque ASM puede ser expresada como una salida de la carta; esto se consigue moviendo todas las salidas de una carta ASM normal dentro de todas las rutas de liga hacia las salidas de la carta y tomando la suma de las salidas resultantes para cada salida de la carta como las nuevas salidas de la trayectoria. Cada salida de la trayectoria será indicada por una notación adicional sobre el nombre del estado; es decir no existen salidas condicionales.

Este estado puede ser implementado con la PROM como se observa en la figura 3.4, la cual muestra únicamente unas líneas de dirección RUTA-LIGA de la máquina, pues la máquina completa consiste de todas las palabras necesarias para representar todos los estados de la máquina.

El decodificador de direcciones puede ser implementado mediante una reducción combinacional ordinaria y compuertas.

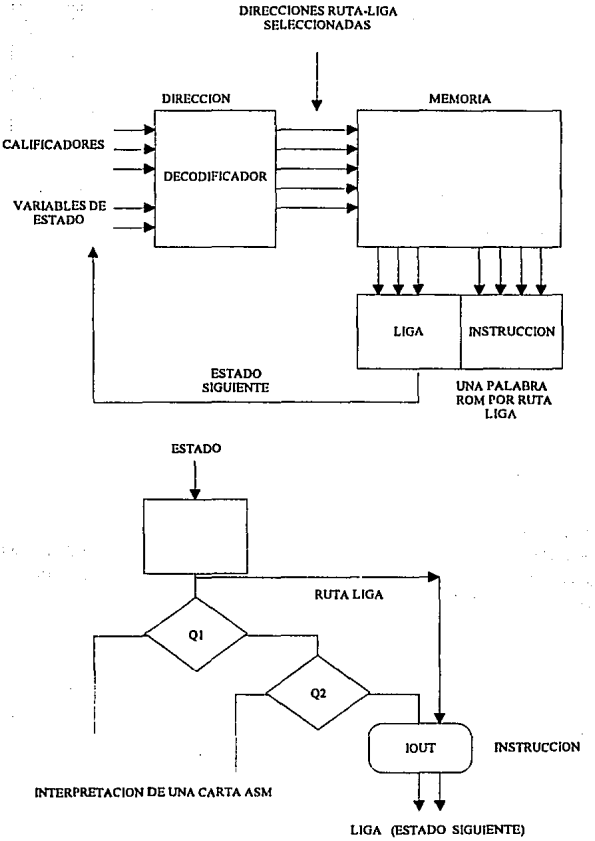

DIRECCIONES RUTA-LIGA
SELECCIONADAS
DIRECCION
MEMORIA
CALIFICADORES
DECODIFICADOR
VARIABLES DE
ESTADO
LIGA
INSTRUCCION
ESTADO
SIGUIENTE
UNA PALABRA
ROM POR RUTA
LIGA
ESTADO
RUTA LIGA
Q1
Q2
IOUT
INSTRUCCION
INTERPRETACION DE UNA CARTA ASM
LIGA (ESTADO SIGUIENTE)

FIGURA3.3

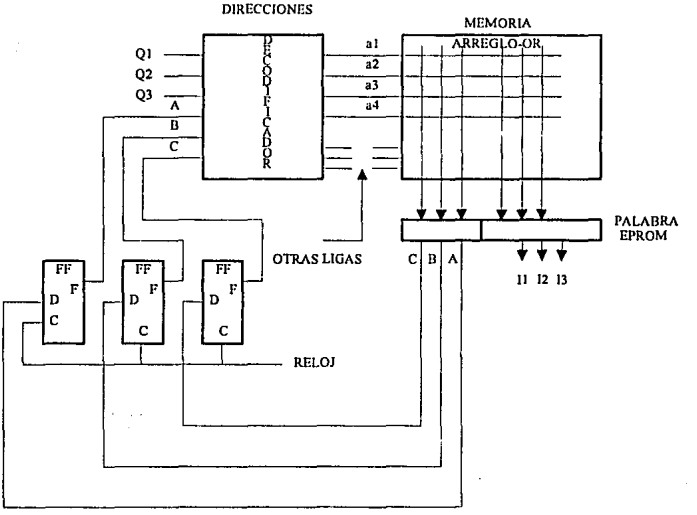
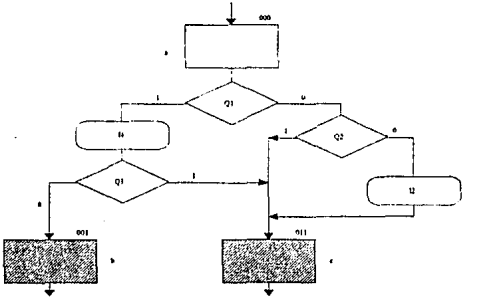
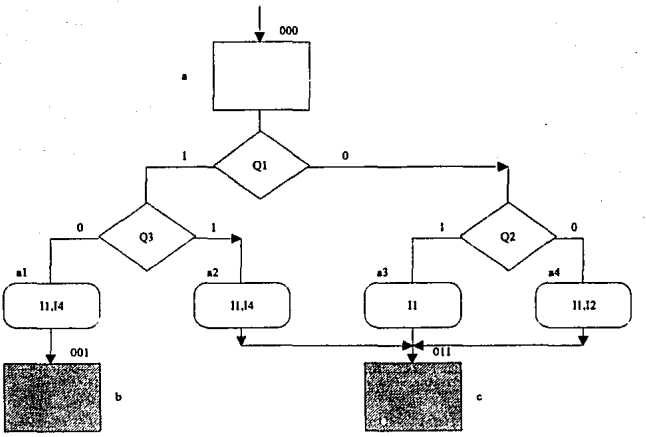


FIGURA3.4

Por ejemplo:

Se puede observar que la parte del decodificador mostrado en la figura 3.5 puede ser implementado como se muestra en la figura 3.6 que aparece debajo de la citada en primer término; debe aclararse que esta solución no es general para cualquier decodificador.





NOTA: ESTADO = C B A.

FIGURA3.5.

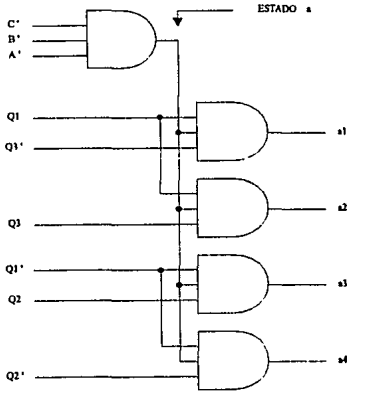


FIGURA3.6

La figura 3.7 proporciona una implementación alternativa del decodificador utilizando un arreglo de compuertas AND, a este arreglo se le llama LIGA-AND; en la figura 3.8 las líneas simples representan las entradas AND colectivas y las flechas indican las salidas de las compuertas AND. También se pueden tener dos tipos de arreglos en el direccionamiento por trayectoria, un arreglo LIGA-AND para las direcciones y un arreglo LIGA-OR para la memoria.

Existe sin embargo un problema con una PROM direccionada por trayectoria, la información para describir la operación de una máquina de estado algorítmica es almacenada en dos tipos diferentes de arreglos, y si existen muchos nombres de variables, el tamaño del arreglo LIGA-AND puede llegar a ser fácilmente del mismo tamaño del arreglo LIGA-OR; además de que son utilizadas unas cuantas de todas las combinaciones posibles para rutas de liga. Así que, es importante contar con una estructura de palabra que permita la descripción de la carta ASM en un sólo tipo de arreglo.

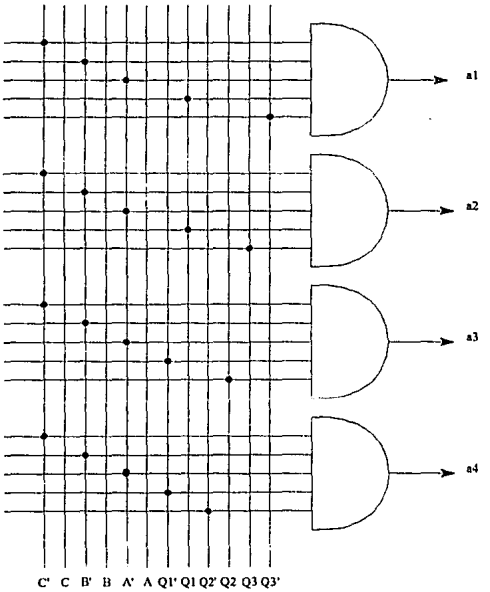


FIGURA 3.7

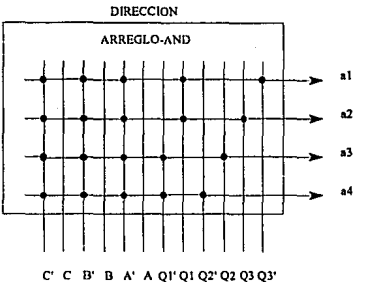


FIGURA 3.8

♦ DIRECCIONAMIENTO ENTRADA-ESTADO.

Al almacenar información en un arreglo LIGA-OR restringimos la descripción de la carta ASM a una máquina de clase 3 con un calificador (entrada) por estado; un segmento de la palabra de la PROM corresponde al calificador seleccionado para cada estado, este segmento es conocido como parte de PRUEBA. La parte de liga tiene ahora dos estados siguientes, el estado siguiente se escoge en base al calificador seleccionado por la parte de prueba. La parte de la palabra de la PROM que corresponde a la instrucción seleccionada, la salida de estado deseada tal como se realiza para las salidas condicionales vistas en los tipos de máquinas de estados. En este tipo de direccionamiento cada dirección de la PROM selecciona una palabra que describe un par CALIFICADOR-ESTADO o ENTRADA-ESTADO, de ahí el nombre de esta técnica de direccionamiento.

En la figura 3.9 se muestran los bloques básicos para este tipo de máquina PROM, además de mostrarse también un estado típico descrito por una palabra de la PROM.

El par ENTRADA-ESTADO tiene un hardware adicional llamado el SELECTOR y el SWITCH. El SELECTOR toma la parte de prueba de la palabra de la PROM y selecciona la entrada correspondiente igual que su salida; el SWITCH selecciona uno de los estados de liga en base al nivel de lógica de la entrada seleccionada y presenta este estado como el estado siguiente para el registro de dirección del estado.

La dirección correspondiente a la variable prueba con valor de "0" (cero) es llamada dirección DIRECTA o FALSA, mientras que para un valor de la variable de prueba de "1" (uno) es llamada la dirección ALTERNA o VERDADERA, estas dos asignaciones son mostradas en la figura anterior y son representadas con las letras F y T en las partes de liga de la palabra de la PROM.

La porción de dirección en el direccionamiento entrada-estado es una decodificación completa simple de "n" lineas de entrada, el decodificador es independiente de la carta ASM ya que sólo nos proporciona los estados posibles suficientes; la carta ASM completa es alamacenada en la porción de memoria de la PROM en una base estado por estado.

Una ventaja que nos ofrece el direccionamiento entrada-estado es que es muy útil cuando se tienen muchas entradas por manejar; sin embargo la desventaja que presenta es que en las cajas de decisión, sólo se puede preguntar simultáneamente por una sola variable, no por varias.

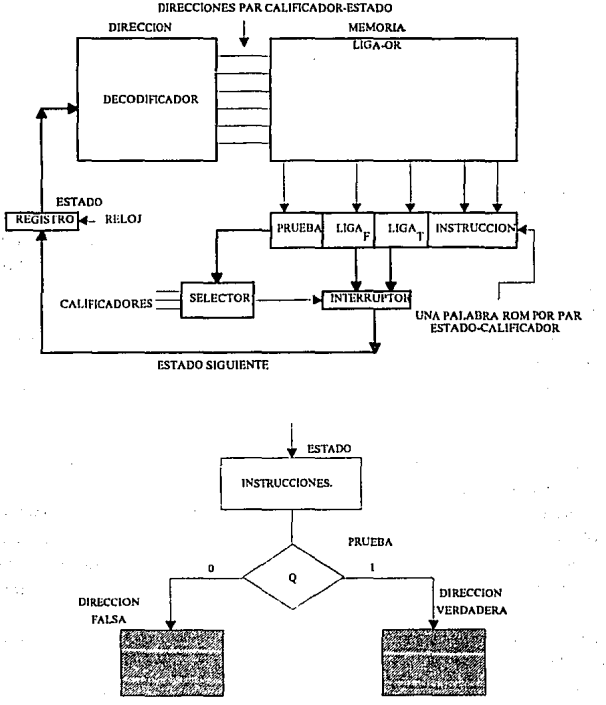
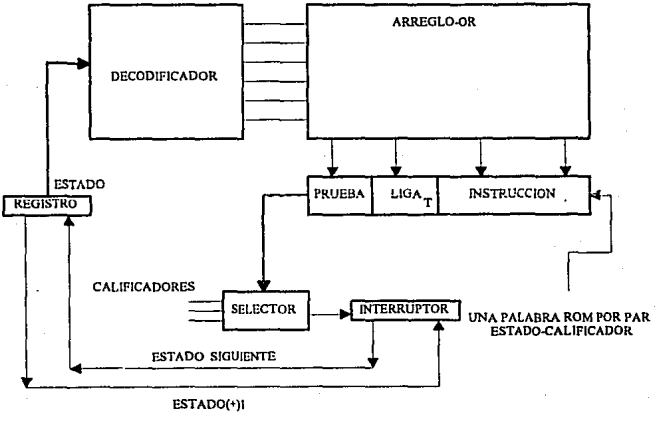


FIGURA 3.9

♦ DIRECCIONAMIENTO IMPLICITO.

Es una variante del direccionamiento entrada-estado y utiliza únicamente un campo de liga en conjunto con el par entrada-estado, la dirección alterna se almacena en este campo y la dirección directa es llamada DIRECCION IMPLICITA y se obtiene sumando 1 (uno) a la dirección presente. La suma es realizada mediante el incremento del registro de dirección aunque en la figura 3.10 se indica mediante una selección del estado (+) 1 por el switch del circuito externo a la PROM. Esta figura también muestra una carta ASM simple con dos estados siguientes y una notación que utiliza el calificador QX sólo con un estado siguiente. La segunda terminologia representa un calificador o entrada que siempre vale "1" para forzar la secuencia de estado a la dirección de verdadero almacenda en la PROM.

En el direccionamiento implícito se requiere de una dirección verdadera forzada para completar las ligas extra causadas por el alcance limitado de la dirección implícita, las ligas extra son llamadas LIGAS_X y son una característica de la carta ASM. Existe un procedimiento simple para encontrar las LIGAS_X en una carta ASM; primero se encuentra el conjunto más pequeño de cadenas ruta-liga que no se intersecten, éstas deben cubrir todos los elementos de la carta ASM, luego se suma una liga-X en cada punto donde se realize un salto a la mitad de una cadena de ligas.



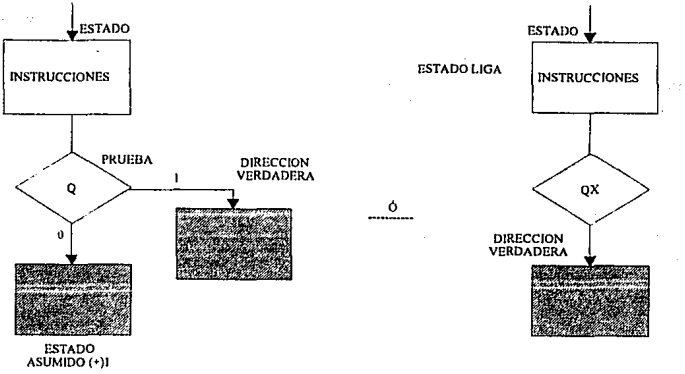


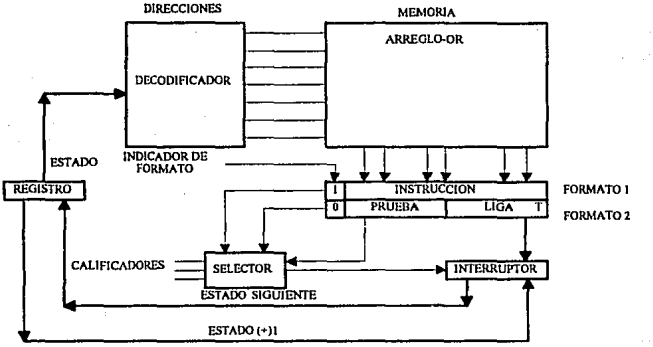
FIGURA 3.10.

♦ DIRECCIONAMIENTO DE FORMATO VARIABLE.

La PROM con direccionamiento por trayectoria y la PROM de dos direcciones son un formato compuesto, cada bit de la palabra de la PROM tiene una función compuesta simple; en una palabra de FORMATO VARIABLE el mismo bit puede tener muchas funciones diferentes dependiendo de la posición, la dirección u otros bits en la palabra. La palabra básica de formato variable tiene dos formatos determinados por un bit de la palabra de la PROM, gráficamente se muestra en la figura 3.11.

El primer formato es la parte de instrucción, mientras el segundo formato es la liga y la parte de prueba; un conjunto compuesto de una instrucción, variable de prueba y la liga ocupa en la memoria dos palabras. En la misma figura 3.11 se observa el diagrama a bloques para el FORMATO VARIABLE y la carta ASM para los dos formatos; el formato 1, seleccionado por un "1" en la posición más izquierda de la palabra de memoria, selecciona las instrucciones de salida del estado, pero como no existe parte de liga se utiliza un direccionamiento implícito para seleccionar la siguiente palabra de la PROM, esta palabra puede contener tanto el formato 1 como el formato 2. En el formato 2 existe la posibilidad de hacer una prueba y saltar a uno de dos estados siguientes; el direccionamiento implícito requiere que uno de estos estados sea sucesivo como se puede verificar en la figura antes mencionada.

La desventaja que existe en este tipo de direccionamiento es que se requieren más tiempos de estado para ejecutar un algoritmo, mientras que comparado con el formato compuesto de dos direcciones es menor el tiempo en la ejecución de las instrucciones.


DIRECCIONES
MEMORIA
ARREGLO-OR
DECODIFICADOR
ESTADO
INDICADOR DE FORMATO
REGISTRO
1
INSTRUCCION
FORMATO 1
0
PRUEBA
LIGA
T
FORMATO 2
CALIFICADORES
SELECTOR
INTERRUPTOR
ESTADO SIGUIENTE
ESTADO (+)1

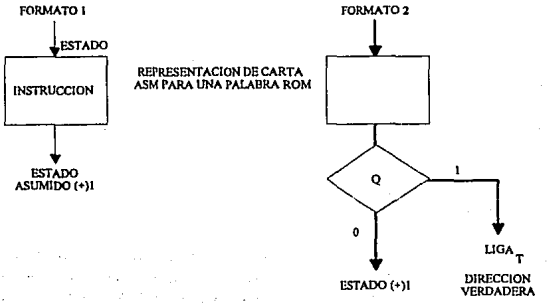

FORMATO 1
ESTADO
INSTRUCCION
ESTADO ASUMIDO (+)1
REPRESENTACION DE CARTA ASM PARA UNA PALABRA ROM
FORMATO 2
Q
1
0
LIGA T
ESTADO (+)1
DIRECCION VERDADERA

FIGURA 3.11.

Las técnicas de direccionamiento antes citadas no son las únicas que existen, ya que son posibles otras alternativas que de una u otra forma cambian las restricciones y condiciones de las funciones f y g; así que cada diseñador puede generar su propia técnica de direccionamiento conociendo las bases de las que son más utilizadas.

III.3. VENTAJAS Y DESVENTAJAS CON OTROS DISPOSITIVOS LOGICOS PROGRAMABLES

♦ **PROMs (Memorias Programables de Solo Lectura).**

La memoria PROM es considerada como el dispositivo más universal de todos los PLD (Dispositivos Lógicos Programables), una PROM con m-entradas y n-salidas es capaz de implementar cualquier conjunto de funciones porque es proporcionado, cada mintérmino. Sin embargo, una PROM puede llegar a ser ineficiente para algunas funciones en las que se requiera un arreglo grande. Las PROM son útiles en las siguientes situaciones:

- Donde el problema es naturalmente especificado en términos de una tabla de verdad, la cual se mapea directamente dentro de las palabras de una PROM, no es necesario ningún proceso de especificación o minimización. De este modo, la función puede ser modificada palabra por palabra, esto ocurre con los controladores microprogramados y aplicaciones de búsqueda en tabla. En estos casos se tiene una desventaja positiva, pues se puede destruir la estructura del problema convirtiéndolo en sumas de productos.
- En la implementación de funciones lógicas se requieren muchos productos para un PLA (Arreglo Lógico Programable), como es el caso de las funciones aritméticas.
- Si se requiere un bloque lógico universal que pueda ser escrito nuevamente, las demandas del término producto no pueden ser predichas, y una PROM es con frecuencia la única solución práctica.
- Las memorias PROM son particularmente útiles en generación de funciones donde es requerida una función arbitraria f(X) de la entrada X (dirección).

Estas memorias están disponibles en muchos tamaños, varias tecnologías y rangos de velocidad. En la tabla 3.1 se resume el rango de memorias que se fabrican actualmente.

♦ **PLAs (Arreglo Lógico Programable).**

El PLA general se muestra en la figura 3.12. La lógica de dos niveles es desarrollada en los arreglos AND y OR, donde cada producto puede ser utilizado en cualquier función de salida, cuya polaridad es controlable. El PLA es también un componente lógico universal que proporciona el número de términos producto suficientes. La ventaja sobre una PROM es que para un número dado de entradas se requiere un arreglo mucho más pequeño, ya que los mintérminos no son generados; por esta razón los PLAs son más rápidos que las PROMs equivalentes en la parte más baja de la tabla 3.1 aunque las memorias más pequeñas tienen velocidades similares a los PLAs. Otra característica que ofrece una ventaja es que la programación de el arreglo de control capacita a cualquier pin de salida para ser utilizado como entrada, esto permite

que el PLA sea empleado en varias configuraciones entrada/salida.

NUMERO DE ENTRADAS.	NUMERO DE PALABRAS.	NUMERO DE SALIDAS. 1 4 8 16
4	16	X
5	32	X X
6	64	X
7	128	X
8	256	X X X
9	512	X X
10	1K	X X X X
11	2K	X X X
12	4K	X X X X
13	8K	X X
14	16K	X X X
15	32K	X
16	64K	X X X X
17	128K	X

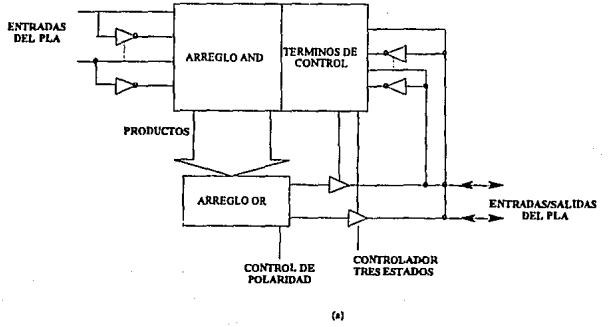
TABLA 3.1. RANGO DE MEMORIAS DISPONIBLES ACTUALMENTE.

Existen aplicaciones donde el número de productos requeridos es muy grande para PLAs estándar, aún después de la minimización; la mayoria de las funciones aritméticas caen en esta categoria. Si consideramos que un PLA es una memoria con un decodificador de dirección programable, pueden comprenderse una clase completa de aplicaciones; así, existen situaciones donde las acciones deben ser tomadas donde ocurren combinaciones de entradas particulares.

Por ejemplo:

Si se requiere arreglar o cubrir ciertas palabras en una memoria de programa, se puede utilizar un PLA para reconocer las direcciones o rangos de dirección a ser arreglados y sustituir la nueva palabra. La figura 3.13 muestra como se conecta el PLA, cada línea de la tabla del programa contiene la dirección o el rango a ser reconocido y la palabra a ser sustituida, esto es:

ENTRADAS (AND).	SALIDAS (OR)
DIRECCIONES	DATO NUEVO



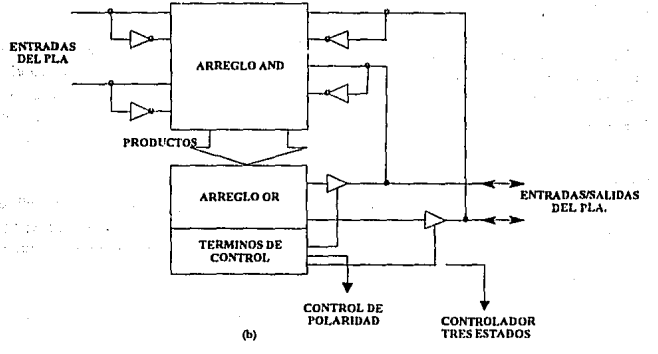


FIGURA 3.12.

(a) PLA SIN CONTROL DE SALIDA (b) SEGUNDA FORMA CON TERMINOS DE CONTROL OR

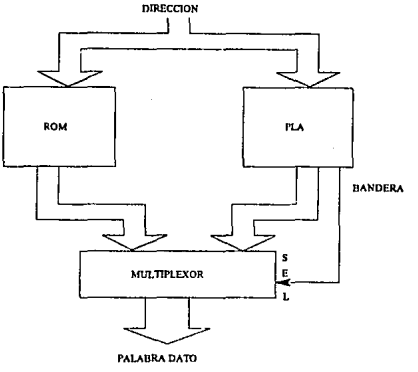


FIGURA 3.13. PLA PARA CORRECCION DE MEMORIA.
NOTA: EL MULTIPLEXOR PUEDE SER REEMPLAZADO CON UN BUS TRES ESTADOS.

Por su universalidad, no existe un número grande de tipos de PLA disponibles, y no llegan a ser obsoletos tan rápidamente como otros componentes más especializados, la aplicación principal ha sido dentro de la tecnología de manufactura.

En la tabla 3.2 se muestran los principales parámetros de algunos de los PLAs disponibles comercialmente.

NOMBRE	NO DE ENTRADAS	NO. DE SALIDAS.	NO. DE E'S	MAXIMO NO. DE ENTRADAS	NO. DE PRODUCTOS	NO. DE TERMINOS DE CONTROL	NO. DE PINES	AÑO DE DIVULGACION.
82S16	14	8	-	14	96	-	24	1973
IM5200	14	8	-	14	48	-	24	1975
PLS100/1	16	8	-	16	48	-	28	1975
74S330/1	12	8	-	12	50	-	20	1976
82S106/7	16	8	-	16	48	-	28	1977
PLS152/3	8	-	10	17	32	10 AND	20	1980
TIFPLA839/40	14	6	-	14	32	-	24	1981
100459	16	8 + 8'	-	16	24	-	64	1985
PLS161	12	8	-	12	48	-	24	1985
PLS173	12	-	10	21	32	10 AND	24	1985
PLHS473	11	2	9	20	24	11 OR	24	1985
PEEL253	8	-	10	17	42	10 OR	20	1987
PEEL273	12	-	10	21	42	10 OR	24	1987

TABLA 3.2. ARQUITECTURAS PLA.

- **PAL (Lógica de Arreglo Programable).**

Un dispositivo PAL es un PLA sin arreglo OR. En lugar de éste tiene un conjunto de compuertas OR para las particiones de combinación de los productos. En aplicaciones donde la capacidad de un PLA de compartir términos producto es desperdiciada, la solución utilizando un PAL será generalmente la más económica, y es probable que el PAL tenga un retardo más pequeño.

Una aplicación común para PALs en sistemas de microcomputadoras es la decodificación de direcciones para dispositivos de memoria y dispositivos de entrada/salida. Los decodificadores pueden ser requeridos para reconocer tanto una dirección simple como rangos de direcciones; para reconocer una dirección simple solo se necesita un término y también para la decodificación de algunos rangos es adecuado un término simple.

En la figura 3.14 se muestra la arquitectura de un PAL, como se observa es muy similar al PLA.

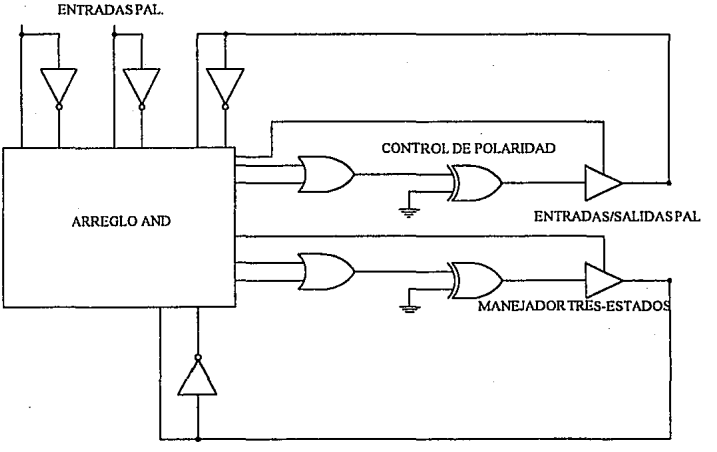


FIGURA 3.14 ARQUITECTURA DEL PAL.

La arquitectura del PAL es muy requerida, la razón es histórica ya que el primer PAL no tenía pines bidireccionales (y además un número compuesto de entradas y salidas), polaridades compuestas en las salidas y un número total limitado de términos producto. Pero compensando esto los PALs son generalmente más económicos y están disponibles en versiones desarrolladas más recientes, es por ello que siempre se vuelve a los primeros dispositivos que aparecieron en el mercado, lógicamente en su presentación actual.

Una característica presente en algunos dispositivos PAL (por ejemplo: 16SP8,20S10), es la guía de términos producto; esta característica permite una mejor distribución de términos producto entre compuertas OR adyacentes.

- **GAL (Lógica de Arreglo Genérico).**

Esta es una familia de dispositivos lógicos programables, borrables eléctricamente, una de las empresas fabricantes es Lattice Semiconductors; el primer miembro de esta familia es el GAL16V8 de 20 pines (ver figura 3.15 donde se presenta el diagrama a bloques del dispositivo), este GAL tiene 8 macroceldas lógicas de salida (OLMC). Las salidas de compuertas AND son alimentadas dentro de una OLMC, donde son sumadas por una compuerta OR (ver figura 3.16). La salida de la compuerta OR maneja una entrada de una compuerta EX-OR. Cada OLMC contiene 4 celdas, SYN, ACO, AC1(n), y EX-OR(n) las cuales pueden ser programadas para seleccionar uno de los cinco modos de operación de la OLMC. El bit ACO, junto con 8AC1(n), selecciona una de las siguientes configuraciones de salida habilitadas:

1. El pin de Entrada/Salida (E/S) en una OLMC es una salida dedicada;
2. El pin de E/S en una OLMC es una entrada dedicada;
3. Los inversores tres-estados en las salidas de todas las OLMC son habilitados por la salida común habilitada OE; o
4. Los inversores tres-estados pueden ser habilitados individualmente por un término producto.

Los bits de control ACO y AC1(n) también determinan la fuente del término de retroalimentación, este término es retroalimentado hacia el arreglo AND vía el multiplexor FMUX.

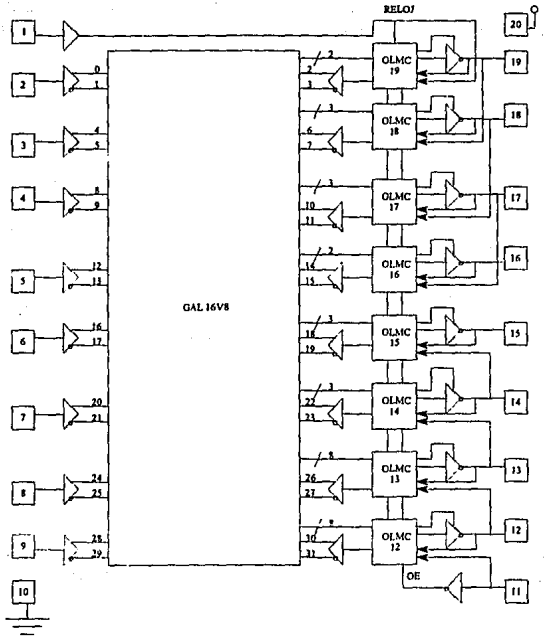


FIGURA 3.15. DIAGRAMA A BLOQUES DEL GAL 16V8.

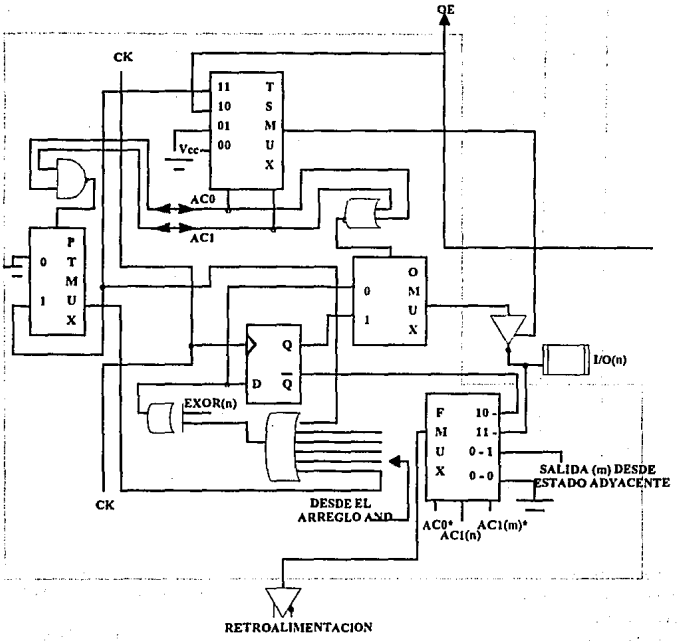


FIGURA 3.16. ESTRUCTURA DE LA MACROCELDA.

Una gran ventaja que ofrece este tipo de dispositivo programable es que puede ser borrado y reprogramado, además de que proporciona cinco posibles modos de operación de las macroceldas, esto es súmamente útil cuando se requieren varias funciones en un circuito y el espacio es pequeño. Los modos de operación anteriormente citados se explican en la tabla 3.3.

MODO	SYN	ACO	ACT(n)	EX-OR (n)	FUNCION
1	1	0	1	-	ENTRADA DEDICADA; LA SALIDA DE LA MACROCELDA ADYACENTE ES RETROALIMENTADA AL ARREGLO AND.
2A	1	0	0	0	SALIDA COMBINACIONAL/ACTIVA EN BAJO; LA ENTRADA HABILITADA DEL INVERSOR TRES-ESTADOS ES CONECTADA A VCC.
2B	1	0	0	1	IGUAL QUE COMO SUCEDE CON LA SALIDA ACTIVA EN ALTO.
3A	1	1	1	0	SALIDA COMBINACIONAL/ACTIVA EN BAJO; LA ENTRADA HABILITADA DEL INVERSOR TRES-ESTADOS ES MANEJADA POR UN TERMINO PRODUCTO DEDICADO Y LA SALIDA ES RETROALIMENTADA AL ARREGLO AND. LA LINEA DE RELOJ ES DESCONECTADA DESDE LA MACROCELDA.
3B	1	1	1	1	IGUAL QUE COMO SUCEDE CON LA SALIDA ACTIVA EN ALTO.
4A	0	1	1	0	SALIDA COMBINACIONAL/ACTIVA EN BAJO; LA ENTRADA HABILITADA DEL INVERSOR TRES-ESTADOS ES MANEJADA POR UN TERMINO PRODUCTO DEDICADO Y LA SALIDA ES RETROALIMENTADA AL ARREGLO AND. LA LINEA DE RELOJ ES CONECTADA A LA MACROCELDA.
4B	0	1	1	1	IGUAL QUE COMO SUCEDE CON LA SALIDA ACTIVA EN ALTO.
5A	0	1	0	0	SALIDA DE REGISTRO/ACTIVA EN BAJO; EL INVERSOR TRES-ESTADOS ES ACTIVADO POR LA LINEA DE HABILITACION.
5B	0	1	0	1	IGUAL QUE COMO SUCEDE CON LA SALIDA ACTIVA EN ALTO.

TABLA 3.3

La expansión de entradas en un FPLA es más difícil de implementar que la expansión de salidas o de término productos, siendo esto último una ventaja para las PROM ya que la implementación con éstas no es complicada.

Cabe hacer notar que cada dispositivo de los aquí mencionados tiene ventajas particulares. Las ROMs son útiles donde la especificación tiene un formato de palabra estructurado o donde se presenta un número grande de productos. Los PLAs son el tipo de dispositivo más flexible ya que todos los productos pueden ser compartidos; sin embargo muchas aplicaciones no requieren de este tipo de dispositivo de tal forma que los dispositivos estructurados PAL pueden llegar a ser más rápidos, más económicos y la mejor opción. El uso de retroalimentación permite el incremento de productos pero redunda en el costo extra del retardo.

Como es lógico concluir, las memorias PROM ofrecen ciertas ventajas sobre otros dispositivos, pero en contraparte existen algunas características especiales que no puede cubrir, es por esto que la decisión de utilizar uno u otro dispositivo queda al criterio del usuario dependiendo también de la aplicación específica que se esté manejando ya sea por los bits que haya que almacenar, por la configuración de entrada/salida, o alguna otra.

CAPITULO IV

CAPITULO IV
IV DISEÑO ELECTRONICO BASADO EN SISTEMAS CAD/CAM/CAE

IV.1 INTRODUCCION

En la actualidad la computación juega un papel muy importante dentro de la sociedad, debido a que ésta cada vez trae consigo mejoras que van desde un notable aumento en la producción, disminución de costos y superación en la calidad del producto.

Estas aplicaciones se han llevado a través de un proceso histórico que tuvo lugar a principios del siglo XVIII cuando aparecieron las primeras máquinas automáticas para la industria textil, más tarde con la segunda guerra mundial apareció la microelectrónica y con ella las microcomputadoras, al mismo tiempo que se introdujeron los robots en la industria para realizar tareas de manipulación, de carga y descarga de piezas de una máquina a otra o bién de una cinta transportadora a una máquina, también podían realizar tareas de soldadura y pintura en la industria automotriz; todo esto controlado por microcomputadoras, con lo cual se logro incrementar la flexibilidad y autonomía de la producción.

Los sistemas CAD (Diseño Asistido por Computadora) comenzaron como una ingeniería tecnológica computarizada, mientras los sistemas CAM (Manufactura Asistida por Computadora) eran una tecnología semiautomática para el control de máquinas que se llevaba a cabo numéricamente; esto dió origen a una nueva tecnología denominada CAD/CAM (Diseño Asistido por Computadora/ Manufactura Asistida por Computadora), que fué pensada para el diseño y fabricación con ayuda de la computadora, es una tecnología que abarca el diseño gráfico, el manejo de bases de datos para el diseño y fabricación, control numérico de máquinas herramientas, robótica y visualización.

Cabe hacer notar que éstas dos disciplinas surgieron separadamente pero se han ido mezclando de tal forma que actualmente los sistemas CAD/CAM se consideran una disciplina única identificable, que día a día se complementa con otras disciplinas como son: el lenguaje natural (asociado con la Inteligencia Artificial), la simulación por computadora, el diseño gráfico, etc.

En el siguiente diagrama (fig.4.1) se ilustran las fases por las que un producto pasa antes de llegar al mercado.

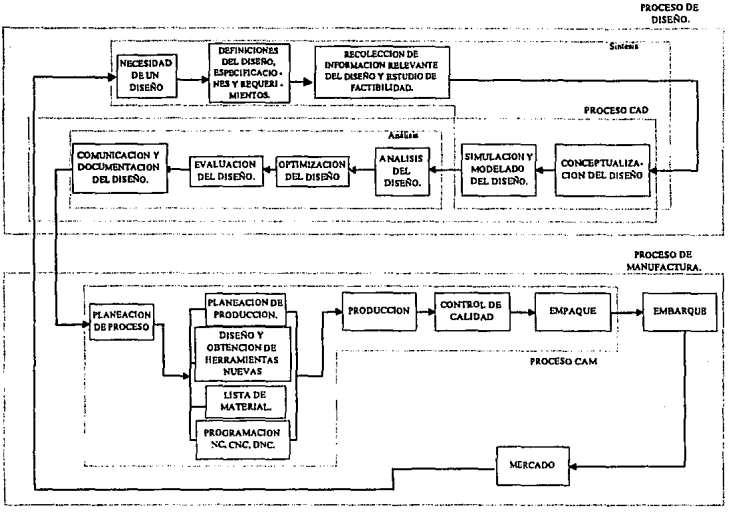


FIGURA 4.1

El ciclo típico de un producto comprende dos procesos principales: el proceso de diseño y el proceso de manufactura; a su vez el proceso de diseño se divide en dos subprocesos: síntesis y análisis, los cuales se explican a continuación.

En el subproceso de síntesis se determina la funcionalidad y filosofía del producto, así como también se concibe la idea del producto dentro de la realidad; la mayor parte de la información generada en éste proceso es cualitativa y por consecuencia difícil de capturar en un sistema de cómputo, la finalidad de este proceso es obtener un diseño conceptual del producto esperado para su presentación posterior.

El subproceso de análisis comienza introduciendo el diseño conceptual en el contexto de las ciencias abstractas de la ingeniería para evaluar la realización del producto esperado, mediante el modelado y simulación del diseño. Para éste proceso es recomendable utilizar un ambiente de cómputo, ya que facilita la selección de alternativas de diseño, obteniendo decisiones en períodos de tiempo más cortos.

El proceso CAD comprende los subprocesos de análisis, modelado, simulación y conceptualización del diseño.

El proceso de manufactura comienza con la planeación y termina con el producto final. Las fases de este proceso sirven como base para definir los contenidos del diseño y la manufactura, en consecuencia las herramientas que un sistema CAD/CAM debe proporcionar a los ingenieros. Para identificar estas

herramientas apropiadamente, se define un proceso CAD y un proceso CAM en relación a los otros procesos; el primero es un subconjunto del proceso de diseño y el segundo un subconjunto del proceso de manufactura.

La ingeniería Asistida por Computadora (CAE) agrupa conceptos tales como los de CAD/CAM y la creación automatizada de dibujos y documentación, enfocándose no solamente a la investigación y diseño previos a la fabricación, sino también a la propia manufactura, como se ilustra en la fig.4.2.

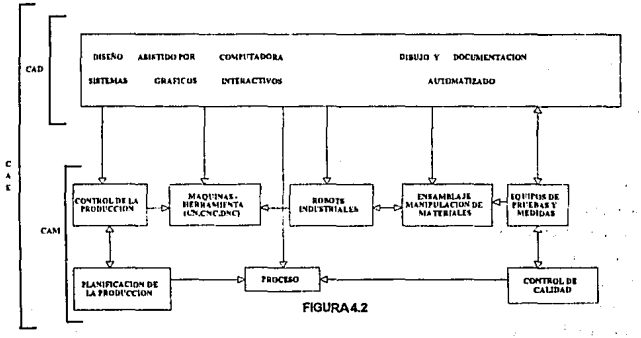


FIGURA 4.2

IV.2 DEFINICION DE CAD, CAM Y CAE

CAD (Diseño Asistido por Computadora), surge de una serie de conferencias dadas por Ivan Sutherland, en el Instituto Tecnológico de Massachusetts durante los primeros años 60's.

En su sentido más moderno, CAD significa proceso de diseño que emplea sofisticadas técnicas gráficas de computadora, apoyadas en paquetes de software para ayudar en los problemas analíticos, de desarrollo, de costo asociados con el trabajo de diseño.

Al emplearse CAD para resolver un problema específico se obtienen las siguientes ventajas:

a) Producción de dibujos más rápida: un diseñador que utiliza un sistema CAD puede producir dibujos unas tres veces más rapidamente de lo que realizaría sobre un tablero de dibujo, ésto acelera la velocidad total del proceso de diseño e introduce el producto al mercado con mayor rapidez. También puede significar una respuesta más rápida a consultas de precios.

b) Mayor precisión de los dibujos: la precisión de un dibujo convencional está en función de la vista del dibujante y del grosor del lápiz con el que se dibuja. Por el contrario, un punto de un dibujo

CAD tiene una precisión exacta, y la técnica conocida como "zooming" permite que una parte de un dibujo CAD se amplíe para mostrar sus componentes con mucho más detalle. Así pues, tanto los detalles como los conjuntos dibujados producidos con CAD son completamente precisos. Con CAD no se requiere nunca una diagramación exacta separada.

c) Dibujos más limpios: la presentación de un dibujo convencional depende por completo del estilo de trabajo y de las herramientas de dibujo de cada dibujante. En cambio, el graficador de un sistema CAD produce líneas y textos de mayor calidad cualquiera que sea el operario del sistema. También muchos dibujos convencionales se estropean a causa de las raspaduras y rastros dejados por las líneas borradas. CAD permite eliminar rápidamente cualquier número de líneas sin dejar rastro alguno en el dibujo final.

d) Dibujos no repetidos: cuando se termina un dibujo o parte de un dibujo, éste se puede almacenar en la computadora para usos posteriores, esto es especialmente importante cuando el dibujo contiene una gama de componentes de características análogas. Se puede accesar cualquier dibujo almacenado para diseñar útiles y montajes, analizar cortes de herramientas y diseñar troqueles, normalmente se requiere un dibujo diferente para cada una de éstas tareas. La memoria de la computadora es también ideal para compilar librerías de símbolos, componentes estándar y formas geométricas.

e) Técnicas especiales de dibujo: además del "zooming", los sistemas CAD poseen técnicas de dibujo más especiales que no son posibles por medios convencionales.

f) Análisis y cálculos de diseños más rápidos: existe una amplia gama de software para llevar a cabo cálculos de diseño en un tiempo mínimo.

g) Estilo de diseño mejorado: las potentes técnicas de modelado por computadora, tales como análisis de elementos finitos, han liberado a los diseñadores de los impedimentos de las restricciones convencionales y les permiten desarrollar formas más innovadoras, estas formas se pueden modificar y optimizar muy rápidamente ahorrando costos.

h) Menores requisitos de desarrollo: las técnicas de análisis y de simulación CAD pueden ahorrar drásticamente el tiempo y dinero invertidos en desarrollos y pruebas de prototipos que constituyen la etapa más costosa del proceso de diseño.

i) Integración del diseño con otras disciplinas: el amplio campo de telecomunicaciones disponibles integrados en redes de computadoras hace posible que el CAD trabaje más estrechamente con otros departamentos de ingeniería.

CAM (Manufactura Asistida por Computadora), se refiere a cualquier proceso de fabricación automática que esté controlado por computadoras. Su origen data de los desarrollos de máquinas controladas numéricamente (NC) del final de los años 40's y principios de los 50's. Se adoptó el término CNC (Control Numérico por Computadora) cuando estas técnicas comenzaron a ser controladas por computadoras al final de los años 50's y durante los 60's. CNC encierra ahora procesos de fabricación automática muy diferentes que incluyen fresado, torneado, oxicorte, corte con láser, troquelado y soldadura eléctrica por puntos. Los

desarrollos paralelos con robots controlados por computadora y fábricas automatizadas dieron lugar a la fabricación completa de unidades, controladas por sistemas informáticos centrales y organizadas bajo una filosofía conocida como FMS (Sistemas de Fabricación Flexibles). El término CAM se utiliza como denominación general para todas las disciplinas antes mencionadas y para cualquier otra tecnología de fabricación controlada por computadora que pueda surgir. Los elementos más importantes de un sistema CAM son:

a) Técnicas de programación y fabricación CNC (Control Numérico por Computadora).
b) Fabricación y ensamblaje mediante robots controlados por computadora.
c) Sistemas de fabricación flexibles (FMS).
d) Técnicas de inspección asistida por computadora (CAI).
e) Técnicas de ensayo asistidas por computadora (CAT).

Las ventajas del sistema CAM están relacionadas con el cumplimiento de los siguientes objetivos:

- Niveles de producción más altos con menor esfuerzo laboral.
- Menor posibilidad de error humano y de las consecuencias de su falta de fiabilidad.
- Mayor versatilidad de los objetos fabricados.
- Ahorro en los costos por incremento de la eficiencia de fabricación e incremento de eficiencia en el almacenamiento y ensamblaje.
- Repetitividad de los procesos de fabricación a través del almacenamiento de los datos
- Productos de mayor calidad

CAE (Ingeniería Asistida por Computadora), es un proceso integrado que incluye todas las funciones de la ingeniería que van desde el diseño propiamente dicho hasta la fabricación. Esto supone, en la práctica, el empleo de sistemas gráficos interactivos combinados con técnicas de modelado geométrico, análisis de estructuras, diseño y dibujo de detalles de piezas, simulación, análisis por elementos finitos y evaluación del comportamiento de los elementos diseñados. En la figura 4.2 antes citada se muestra un diagrama a bloques de lo que es CAE.

El elemento geométrico de un producto es el elemento central dentro del concepto de CAE, y consiste en la representación del mismo en la memoria de la computadora, todos los demás elementos del sistema CAE utilizan ésta descripción geométrica como punto de partida.

Por ejemplo:

El contorno de la pieza puede emplearse para determinar el paso de la herramienta, al mecanizarse ésta mediante un sistema de control numérico, dependiendo del sistema CAD/CAM que se emplee, el modelo

geométrico puede representarse en la memoria en dos o tres dimensiones. Cuando se utiliza la representación en tres dimensiones, el tipo de diseño más comúnmente utilizado es el llamado "croquis", siendo éste el más sencillo, pero que representa el inconveniente de ofrecer poca información acerca de las superficies de la pieza y no distingue entre su interior y su exterior. Por lo tanto su desventaja estriba en las dificultades de interpretación cuando se trata de estructuras complejas. Los inconvenientes que surgen empleando la técnica anterior, se pueden evitar mediante el sombreado de las superficies del modelo, con lo que se consigue un mayor realismo, logrando así que los modelos definan con gran pecisión la geometría de la superficie de la pieza, por lo que resultan ser más útiles para aplicaciones como la del control numérico. Esta técnica tiene también algunos inconvenientes, ya que al "no contener materia", impide el cálculo de magnitudes como el volumen, peso, momentos de inercia, etc.

Existen otras técnicas más, pero la técnica más utilizada para el análisis de una estructura o modelo en la actualidad es la del "Método de Elementos Finitos (FEM)", en ésta técnica se emplea una malla de elementos sencillos para representar la pieza.

En la figura 4.3 se muestra una malla con rejillas de elementos finitos para el empleo de CAE, la computadora utiliza dicha representación para determinar características impuestas por determinadas condiciones de trabajo, como lo pueden ser esfuerzos y deformaciones.

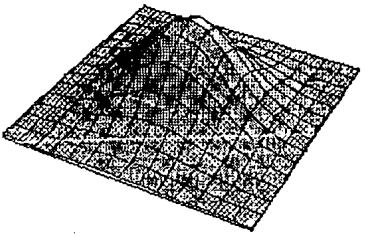


FIGURA 4.3

Un programa de elementos finitos puede caracterizarse por los diferentes problemas mecánicos que es capaz de resolver y por las diferentes geometrías que son susceptibles de análisis.

Desde el punto de vista geométrico pueden presentarse problemas en una, dos o tres dimensiones; y desde el punto de vista de los problemas mecánicos a resolver cabe mencionar los siguientes: análisis térmico, análisis estático, análisis dinámico (cálculo de frecuencias y modos naturales, respuesta transitoria y respuesta estacionaria).

El empleo de la simulación en un sistema CAE permite evaluar el comportamiento de un elemento constituido por diversas piezas, los datos de dichas piezas y su comportamiento es empleado en conjunto para predecir el funcionamiento del producto final. Las simulaciones se llevan a cabo empleando diseños previos y con datos conocidos, cuando se desean nuevos productos, las simulaciones se efectúan a través de un diálogo interactivo, en el que se solicitan datos acerca de parámetros cuyos valores son ajustados de forma interactiva con el diseñador.

Los programas necesarios para un sistema CAE pueden provenir de tres fuentes distintas, las cuales pueden ser:

- el vendedor del sistema,
- los desarrollados en la propia empresa y
- los subcontratados por terceros.

Existe una gran cantidad de ejemplos del empleo de CAE, entre los que se pueden mecionar el diseño y prueba de transmisores diferenciales de presión o el diseño, prueba y fabricación de variadores de velocidad basados en microprocesadores para motores de corriente contínua.

IV.3 INTEGRACION CAD/CAM/CAE.

CAD/CAM es una integracón de las técnicas CAD y CAM dando por resultado un proceso completo, esta integración permite que pueda dibujarse cualquier componente sobre una pantalla de computadora y transferir estos gráficos por medio de señales eléctricas a través de una interfase que los enlace a un sistema de fabricación, en donde los componentes se puedan producir automáticamente sobre una máquina de control numérico por computadora.

Para conseguir la integración del Diseño Asistido por Computadora y la Manufactura Asistida por Computadora es necesaria una compatibilidad que debe enmarcar planes, arquitecturas, bases de datos y elementos específicos de integración. Esta compatibilidad facilita la comunicación entre ambas técnicas.

La fase de planificación de la integración implica un análisis del estado actual y de las espectativas de los sistemas CAD y CAM, las prioridades para facilitar la integración tienen como objetivo mejorar la calidad de los datos, el secuenciamiento de los mismos y la facilidad de su empleo (movimiento), así como una presentación sencilla que mejore el flujo de la información.

En términos de CAD/CAM una arquitectura se refiere a interfases y estructuras de diversos sistemas CAD/CAM empleados por una empresa, el objetivo de la arquitectura de integración es tomar un conjunto de decisiones y establecer las especificaciones que describan la forma en la que los diversos elementos del entorno del CAD/CAM deben interactuar unos con otros. Esta arquitectura describe la existencia de información (bases de datos) y las formas en las que la información se transmitirá de una base de datos a otra.

Uno de los puntos más importantes para la integración CAD/CAM es la elección de las bases de datos y su arquitectura. Por lo que se debe considerar este punto al momento de realizar la integración.

Una extensión evidente del proceso CAD/CAM integrado es la comunicación tanto de CAD/CAM, así como con otros sistemas informáticos, por lo que todas las actividades de ingeniería controladas por computadora se agrupan bajo el concepto general de CAE (Ingeniería Asistida por Computadora), surgiendo así la integración de los sistemas **CAD/CAM/CAE.** Este sistema se define como la integración de computadoras digitales (hardware) y paquetes de programación (software) que pueden crear, modificar, almacenar y desplegar información gráfica. Estos sistemas aplicados al dibujo, diseño y administración permiten incrementar la productividad, reducir tiempos, disminuir costos y mejorar los niveles de calidad de un producto al simplificar y optimizar los procesos de producción, siendo así un factor clave para consolidar y mantener un proceso gradual de modernización y competitividad en la industria.

IV.3.1 APLICACIONES CAD/CAM/CAE.

Debido al gran avance en la tecnologia de integración de los circuitos electrónicos surge la necesidad de reducir el tiempo de diseño para poder lanzar al mercado un producto tecnológicamente vigente, lo cual constituye un reto al que deben enfrentarse los fabricantes de equipo electrónico, este reto se ha ido superando gracias a la ayuda de la computadora, ya que ésta permite agilizar las tareas de diseño. De esta forma nace el concepto CAD, el cual está enfocado al diseño físico, ya sea en el trazado de la tarjeta del circuito impreso, o bien en el de un circuito integrado LSI o VLSI; tomando como base la información del diseño en el sistema CAD.

La tecnologia CAE nace como una herramienta para el ingeniero de diseño, la cual tiene como finalidad ayudar al diseñador en todas las etapas del desarrollo del producto.

CAE se basa en estaciones de trabajo específicas dotadas de terminales, gráficos de alta resolución y software adecuado para la automatización de todas las tareas de diseño, desde la concepción del diseño hasta la realización del diseño físico o trazado, ya sea en el caso de tarjetas de circuito impreso o de circuitos integrados LSI y VLSI. Todo esto es soportado por una base de datos única que permite mantener consistencia durante todas las fases del desarrollo, evitando problemas de concatenación de tareas y de actualizaciones constantes producidas durante la etapa de diseño.

Considerando esta tecnología, el proceso de desarrollo de un producto cambia drásticamente, agilizándose y anulando la posibilidad de introducción de errores. Lo que antes era realización de un diseño esquemático sobre papel para su posterior modelado en una tarjeta de prueba con numerosas iteraciones hasta conseguir un prototipo funcional,lo cual pasa a ser, con el uso del concepto CAE un proceso de captura de esquemáticos en la computadora más la ejecución de un programa de simulación electrónico, sin necesidad de realizar módulos de prueba, se puede comprobar la funcionalidad del diseño capturado así como generar toda la documentación requerida.

IV.4 HERRAMIENTAS COMPUTACIONALES ORIENTADAS A ELECTRONICA

Basándose en el concepto de CAE se han desarrollado una gran cantidad de paquetes de software orientados al diseño de circuitos electrónicos, los cuales exigen cada vez más requerimientos de hardware para su funcionamiento, estos paquetes tienen aplicaciones que van desde una computadora personal hasta una estación de trabajo.

Dado que existe una variedad de productos en el mercado, es importante seleccionar el adecuado, por el costo que esta inversión representa. Por tal motivo, durante el desarrollo del presente trabajo se decidió utilizar los recursos tanto de software como de hardware con los que cuenta la UNAM en la Facultad de Ingeniería.

El software que se seleccionó para el desarrollo del presente trabajo está orientado a computadoras personales y a continuación se da una breve descripción del mismo.

- **palCAD**, el cual esta a disposición de los usuarios en el laboratorio de Diseño Asistido por Computadora de la DEPFI (División de Estudios de Posgrado de la Facultad de Ingeniería). Esta herramienta es útil para el diseño de la carta ASM, minimización de estados de una carta ASM, así como la generación del mapa de fusibles, en el apéndice A se encontrará una descripción más detallada.

- **OrCAD**, se encuentra disponible en el Laboratorio de Computación en la DIMEI (División de Ingeniería Mecánica Eléctrica e Industrial), Laboratorio de Diseño Asistido por Computadora de la DEPFI, Laboratorio de Electrónica, y Laboratorio de Manufactura del CDM (Centro de Diseño Mecánico) de la Facultad de Ingeniería. Este software permite capturar el diseño esquemático de un circuito electrónico, verificando posibles errores en las conexiones (corto circuito, plano de tierra, plano de Vcc, etc.). La información obtenida en este paquete puede ser utilizada para continuar en otra etapa del diseño, en el apéndice A se menciona con más detalle.

- **TANGO-PLUS**, se encuentra disponible en los laboratorios citados anteriormente. Este paquete permite el diseño y producción de tarjetas de circuito impreso (PCB), está constituido por módulos, éstos son:

Esquemático, Diseño de Circuito Impreso, y AutoRuteo, así como interfases con CAM. Las características principales con que cuenta, se describen en el apéndice A.

- **UP600**, se encuentra disponible en el Laboratorio de Computación de la DIMEI, este software permite programar la mayoría de los dispositivos programables que existen en el mercado actual incluyendo PROM's, EPROM's, PAL's, GAL's, etc.. También cuenta con una unidad de hardware básico que proporciona manejadores de pin con voltajes de programación generados para software independiente. Para más detalles consultar el apéndice A.

CAPITULO V

CAPITULO V.
V DISEÑO E IMPLEMENTACION DEL SIMULADOR DE MEMORIAS EPROM (SME)

V.1 INTRODUCCION

Dentro del plan de estudios de la carrera de Ingeniería en Computación de la Facultad de Ingeniería de la UNAM, se encuentran materias tales como Diseño Lógico, Diseño de Sistemas Digitales, Memorias y Periféricos, Organización de Computadoras, etc. En todas estas materias se utilizan memorias EPROM, en especial la memoria 2716, en el diseño con cartas ASM (Algoritmo de Máquina de Estados); con el fin de desarrollar proyectos que conduzcan a un mejor aprendizaje, poniendo en práctica los conocimientos teóricos. Por ello es importante que los alumnos cuenten con todas las posibilidades en cuanto a software y hardware para llevar a cabo un buen desempeño escolar que más tarde redundará en un mejor desempeño profesional.

El diseño del SME se ha dividido en cinco fases, éstas se enlistan a continuación:

FASE I: PLANTEAMIENTO DEL PROBLEMA
FASE II: INVESTIGACION DE RECURSOS Y MATERIAL BIBLIOGRAFICO
FASE III: POSIBLES SOLUCIONES
FASE IV: SELECCION DE UNA SOLUCION
FASE V: DISEÑO E IMPLEMENTACION

V.2 PLANTEAMIENTO DEL PROBLEMA

Los problemas a los que más comúnmente se enfrenta el alumno que diseña sistemas que involucran memorias EPROM, pueden identificarse dentro de los que a continuación se presentan:

1) Pocas fuentes de información sobre diseño basado en Memorias EPROM y tipos de direccionamiento más utilizados. Es decir, la información se encuentra fragmentada en diferentes títulos y éstos son muy solicitados por la misma razón, lo cual de alguna manera impide que el alumno reuna la información necesaria para un mejor desarrollo de sus diseños.
2) Falta de equipo adecuado que permita verificar el contenido de las memorias que previamente han sido grabadas. Ya que el Grabador Universal UP600 existente en el Laboratorio de Computación de la División de Ingeniería Eléctrica, Electrónica y Computación, es el único equipo que permite llevar a cabo las operaciones de grabado y verificación de componentes. Esto además de retrasar los diseños no permite que la retroalimentación y depuración de errores se realize en el menor tiempo posible, es decir, la persona que se encuentra realizando un diseño y solamente requiere verificar el contenido de la memoria, tendrá que esperar turno para poder hacerlo con el Grabador Universal UP600.
3) No existen manuales al alcance del usuario que permitan operar el Grabador Universal UP600 adecuadamente, lo que de alguna manera contribuye a que el tiempo invertido en la grabación se incremente, así como también se presenten errores en la programación de los componentes.

4) Se desconoce el software orientado al diseño de circuitos electrónicos, por lo que estos recursos no son aprovechados en toda su capacidad.

De los problemas expuestos nace la idea de generar un circuito electrónico que permita verificar el contenido de una memoria EPROM 2716, basándose para su diseño en las herramientas que se encuentran disponibles en los laboratorios de la Facultad de Ingeniería, de esta manera se divulgarían dichas herramientas y se pondría a disposición del alumnado no sólo el presente desarrollo sino también el propio circuito para su utilización en proyectos que así lo requieran.

V.3 INVESTIGACION DE RECURSOS Y MATERIAL BIBLIOGRAFICO

Una vez planteado el problema, se inició la búsqueda de información que pudiera ser de utilidad para encontrar posibles soluciones. Se investigó con que recursos se contaba en la Facultad de Ingeniería tanto de software como de hardware, el resultado de esta investigación es el siguiente:

SOFTWARE DISPONIBLE

Grabador Universal UP600 (incluye software y hardware)
palCAD
OrCAD
Tango PCB
Tango ROUTE
Para mayor referencia de cada uno de los paquetes antes mencionados consultar el apéndice B.

HARDWARE DISPONIBLE

Computadoras, las cuales cuentan con la configuración necesaria para que el software funcione adecuadamente
Borrador de Memorias de luz ultravioleta

De igual manera, se recopiló bibliografía relacionada al Diseño de Sistemas Digitales con cartas ASM, memorias EPROM, Diseño Asistido por Computadora (CAD), Referencias Técnicas de Memorias EPROM. Así se tuvo una visión más amplia sobre el problema que se había planteado en la primera fase del desarrollo.

V.4 POSIBLES SOLUCIONES

Para solucionar el problema, se presentaron diferentes alternativas, cada una de ellas ofrecía ventajas y desventajas, a continuación se presenta la descripción y los componentes que podrían utilizarse en cada caso.

1) Diseño de un circuito electrónico basado en memorias EPROM, con este tipo de diseño las técnicas de direccionamiento más frecuentemente utilizadas son: ruta-liga, implícito, formato variable y entrada-estado. Esta opción es sencilla de implementar cuando las características del diseño son simples, sin embargo, a

medida que la complejidad del mismo crece, el número de componentes a utilizar se incrementa considerablemente. Por otro lado, la ventaja que ofrece utilizar meorias EPROM, es que son muy comerciales y su costo no es elevado, además de que dadas sus características de borrable y programable un solo componente puede reutilizarse varias veces.

2) Diseño de un circuito electrónico basado en PAL's y GAL's, con este tipo de componentes el diseño del circuito llega a ser complicado dadas las características limitadas que éstos presentan. Pero una ventaja que podría ser significativa en un momento dado es que son muy comerciales, encontrándose en diferentes presentaciones. La desventaja que se presentaría es que el número de elementos a utilizar se incrementa considerablamente cuando las características del diseño son complejas y con ello la lógica del alambrado se complica porque el número de conexiones entre elementos puede llegar a ser muy grande.
3) Diseño de un circuito electrónico basado en un Microcontrolador, en este diseño es posible implementar circuitos complejos, ya que un microcontrolador ofrece una estructura semejante a un microporcesador, aunque no con las mismas capacidades de este último. Con este tipo de componente el número de elementos requeridos en el diseño se reduce, con lo cual el espacio que llega a ocupar el mismo es menor. En contraparte, para poder realizar un diseño con un microcontrolador se requiere de conocer toda la arquitectura del mismo, así como el conjunto de instrucciones que puede ejecutar, para de esta manera llevar a cabo la programación correcta del mismo.
4) Diseño de un circuito electrónico basado en un Microprocesador, al llevar a cabo un diseño en base a un Microprocesador, se debe considerar el costo elevado de éste, así como también la correcta y adecuada operación del mismo. Un microprocesador está orientado a diseños donde se involucran varios elementos que interactuan entre sí a través de un controlador, en una estructura de este tipo generalmente se encuentran dispositivos de entrada salida, de almacenamiento interno y externo, de procesamiento de datos, etc. Por ello este tipo de diseños son más aplicados a desarrollo y fabricación de microcomputadoras o controladores industriales. Sin embargo, esta opción puede ser incluida como una posible solución y corresponde al diseñador evaluar si es la más adecuada o no.

Con cada una de las opciones presentadas es posible desarrollar el diseño apoyándose en herramientas orientadas al diseño por computadora, con lo que se consigue un trabajo final más limpio, en el menor tiempo posible y a menor costo.

V.5 SELECCION DE UNA SOLUCION

Dadas las características del circuito requerido se ha optado por el diseño de un circuito controlador utilizando como elemento principal un microcontrolador, para ello se ha considerado la disponibilidad de recursos que en la segunda fase de este desarrollo se presentan, entre ellos se cuenta con la literatura relacionada a la arquitectura de un microcontrolador. Además se ha investigado si algún microcontrolador es comercial y si su costo no es elevado (los precios varían entre 50 y 80 nuevos pesos por componente). Por todo lo anterior y aunado a que se cuenta con conocimientos previos sobre diseño con microcontroladores en las secciones siguientes se irá desarrollando la implementación de esta solución.

V.6 IMPLEMENTACION

Para llevar a cabo la implementación del circuito Simulador de Memorias EPROM (SME), se elaboró un diagrama a bloques que esquematiza la interacción entre los elementos que componen el diseño. Este diagrama se presenta en la figura 5.1.

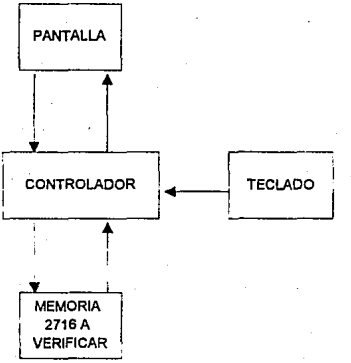


FIGURA 5.1. DIAGRAMA A BLOQUES DEL SME

Los bloques de la figura 5.1 se explican brevemente a continuación con la finalidad de que pueda comprenderse mejor la estructura del SME.

PANTALLA: En este bloque se proporciona un dispositivo de salida que permite visualizar la verificación de la memoria al momento en que ésta está llevandose a cabo. La pantalla además proporciona una guía con la que el usuario puede comprobar la correspondencia entre la información de la memoria y la que se genera con la carta ASM, tomando en consideración el tipo de direccionamiento que se haya utilizado en la carta. Para este caso la guía puede ser útil en los direccionamientos por trayectoria, entrada-estado, implícito y de formato variable.

TECLADO: El bloque del teclado funciona como un elemento de entrada cuya función es controlar las funciones que se tienen disponibles en la verificación de la memoria. Así como también es capaz de interrumpir la verificación si el usuario ya no desea seguir utilizando el SME.

MEMORIA 2716 A VERIFICAR: La memoria 2716 es proporcionada por el usuario y es el elemento que será verificado por el controlador.

CONTROLADOR: Es la parte más importante en este diseño, ya que el controlador practicamente realiza todas las operaciones necesarias para llevar a cabo la verificación de la memoria EPROM 2716, auxiliándose de los bloques PANTALLA y TECLADO para poder tener comunicación con el usuario.

Una vez planteados los bloques que componen el SME, se generó el algoritmo que permitiria la definición más específica de los componentes a utilizar en la implementación, este algoritmo se presenta a continuación.

a) Encendido del SME.

b) En la pantalla debe aparecer la leyenda que se ilustra en la figura 5.2.

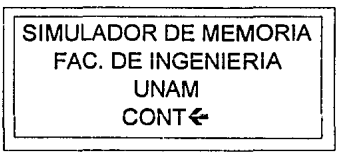


FIGURA 5.2

c) Para continuar se debe presionar una tecla (tecla ENTRAR).

d) Inicia la verificación de la memoria, mostrando en la pantalla la dirección en código hexadecimal y el contenido de la palabra de la memoria en código binario, la estructura de esta pantalla se muestra en la figura 5.3.

FIGURA 5.3

e) Se verifica si el usuario no ha presionado una tecla (tecla PAUSA) la cual indica que se desea ver por más tiempo lo que se está desplegando en la pantalla. Mientras no se presiones esta tecla (PAUSA), el SME continuará verificando cada una de las direcciones de la memoria 2716.

f) Si no se ha presionado la tecla PAUSA, entonces se detecta si el usuario ha interrumpido el proceso de verificación; si lo ha hecho, la pantalla muestra la leyenda ilustrada en la figura 5.4. Y mientras no se interrumpa el proceso de verificación, se mantiene en un ciclo regresando al inciso d hasta que finalize la verificación.

FIGURA 5.4

De la figura 5.4 se observa que se tienen dos opciones: FIN y REINICIO, las acciones que debe seguir el SME para cada opción se explican a continuación.

1) Si se selecciona la opción FIN, la pantalla muestra el mensaje que se ilustra en la figura 5.5, y en este momento el SME puede ser apagado pues su utilización ha finalizado.

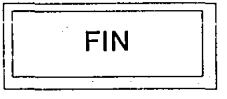


FIGURA 5.5

2) Si se selecciona la opción REINICIO, el SME iniciará una nueva verificación. Por supuesto

previamente se debe proporcionar la memoria que ahora comenzará a verificar. El SME regresa al inciso b.

g) Si no se interrumpió el proceso de verificación y se ha terminado de verificar hasta la dirección 7FF hexadecimal (número de palabras de 8 bits que contiene la memoria 2716), entonces aparece la pantalla mostrada en la figura 5.6, en dicha pantalla se espera a que se presiona la tecla ENTRAR, una vez que ésta se presione, aparecerá el letrero final mostrado en la figura 5.6a, con lo cual finaliza la utilización del SME.

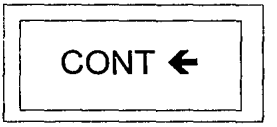


FIGURA 5.6

FIGURA 5.6a

En la figura 5.7 se muestra el diagrama de flujo que representa el algoritmo arriba descrito.

Para poder llevar a cabo la implementación física de este diagrama, se seleccionaron los componentes electrónicos a utilizar, esta selección se hizo en base a las necesidades que presenta el circuito. Tomando en consideración que se ha optado por un circuito electrónico basado en un microcontrolador, el primer elemento seleccionado es el Microcontrolador 8749H de Intel. La pantalla que permitirá visualizar la verificación de la memoria es una Pantalla de Cristal Líquido AND721, la cual está compuesta por una matriz de 4 renglones por 20 columnas para el despliegue de caracteres. El teclado se ha construido para esta aplicación específica de cuatro teclas, las teclas son botones de presión ("push botton"), éstos son muy sencillos de manejar y se encuentran con facilidad en el comercio. La memoria para la cual se ha diseñado el SME es cualquier memoria 2716 compatible con la que fabrica Intel, la presentación del circuito integrado debe ser rectangular, dada la base (de 24 pines), que se proporciona en el SME para que ésta sea colocada.

Para alimentar el SME se requiere de un circuito que reduzca el voltaje de alimentación de un voltaje de

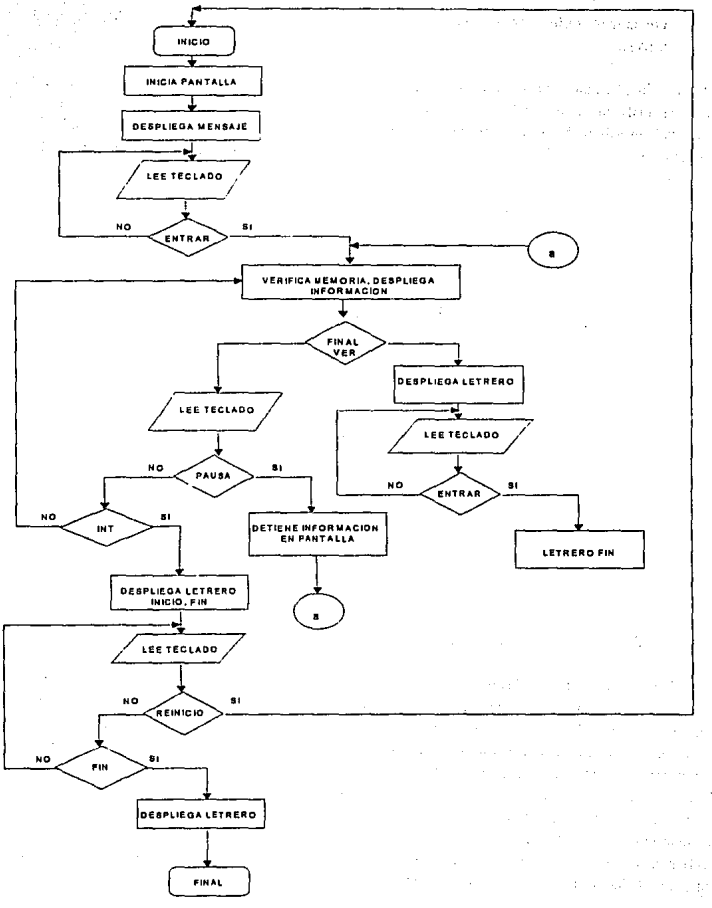


FIGURA 5.7. DIAGRAMA DE FLUJO DEL ALGORITMO DEL SME

110 V aproximadamente, a un voltaje de 5 volts, voltaje con el cual operará el SME.

Para cualquier información a detalle de los circuitos antes mencionados se puede consultar el apéndice B, en donde se encuentra una descripción amplia de los componentes utilizados en el SME.

Una vez definidos los componentes electrónicos a utilizar, se estructuró el diagrama esquemático del mismo, en donde se muestran las conexiones que hay de un ciruito a otro, pero antes de iniciar las conexiones hubo de definirse también que bits servirían de control para los procesos, y cuales para el envío y recepción de datos, en la tabla 5.1 se muestra la organización de la palabra de cada uno de los puertos del microcontrolador.

PUERTO 1	PUERTO 2	BUS DE DATOS
Los bits correspondientes al puerto 1, se utilizan para enviar datos y comandos a la pantalla de cristal liquido.Y además se utilizan para direccionar los ocho bits más significativos de la memoria 2716.	Los ocho bits que forman el puerto 2, se utilizan de la siguiente manera: P20, P21 Y P22 para direccionar los tres bits más significativos de la memoria 2716. P23 para controlar la señal R/W' de la pantalla de cristal líquido. P24 para la señal RS de la pantalla. P25 para la señal E de la pantalla. P26 para la señal E de la memoria. P27 para la señal E de un bus tres estados.	Los ocho bits que componen el bus de datos son utilizados para leer el contenido de la palabra de la memoria 2716. Y también los dos menos significativos son utilizados para leer los datos generados por el teclado.

TABLA 5.1 ORGANIZACION DE LA PALABRA DE LOS PUERTOS 1, 2, Y BUS DEL MICROCONTROLADOR.

Teniendo todos los elementos necesarios para interconectar componentes, se realizó el Diagrama Esquemático del SME, para ello se utilizó la herramienta OrCAD (para mayor detalle ver apéndice A), y el diagrama resultante es el mostrado en la figura 5.8, con lo cual queda completamente definido el SME, en cuanto a la parte electrónica.

Dado que el microcontrolador tiene que ser programado utilizando su propio lenguaje ensamblador, se inició la elaboración del programa que permitiría llevar a cabo las funciones esquematizadas en el diagrama de la figura 5.7. De esta manera la etapa final de la implementación consistiría en vaciar el programa elaborado, a la memoria ROM interna del microcontrolador 8749, para posteriormente realizar las pruebas y correcciones necesarios sobre dicho programa.

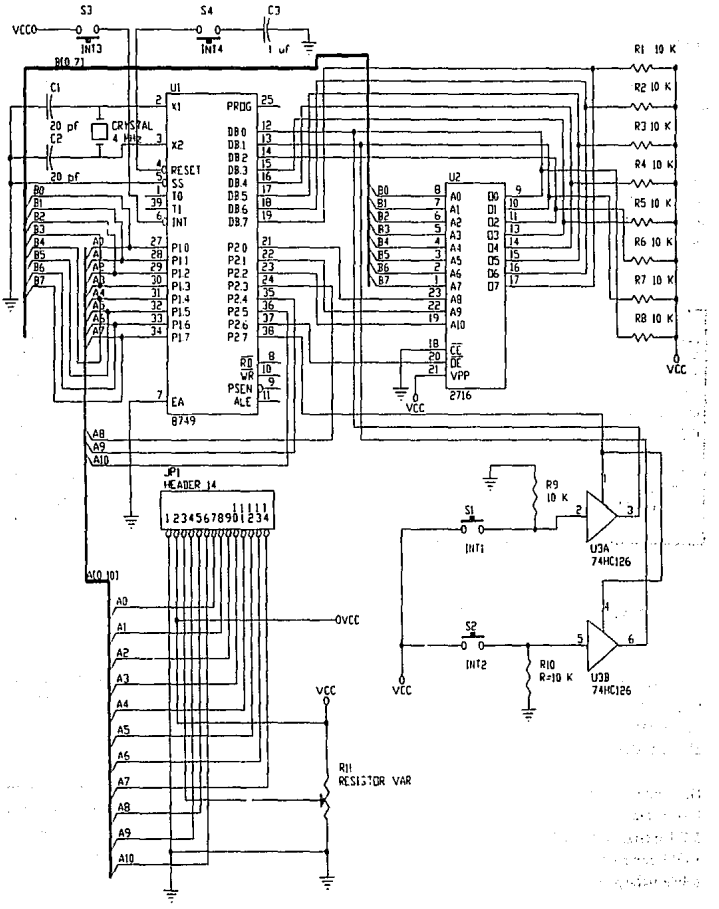


FIGURA 5.8. DIAGRAMA ESQUEMATICO DEL SME

Para generar el código del programa se utilizó el editor del Sistema Operativo MS-DOS versión 5.0, y para ensamblar el código se utilizó el Ensamblador 8048 CROSS ASSEMBLER (XASM48) VERSION 1.64 de Avocet Systems, Inc. de 1983. Antes de ser grabado el programa en el microcontrolador, se simuló en el Simulador que también distribuye la empresa Avocet Systems, Inc. Con ello se pudieron localizar algunos errores en la programación, para poder llevar a cabo correcciones y nuevamente realizar la simulación. Esta herramienta fue de gran utilidad, pues de esta manera no hubo necesidad de realizar muchas pruebas con el circuito físico, sino que sólo se iban realizando las pruebas que se creyeron oportunas e importantes, es decir, pruebas en las que las condiciones reales del circuito eran definitivas y no podían conseguirse con la simulación por computadora.

Finalmente se obtuvo un programa depurado y corregido, el cual se grabó en el microcontrolador, este programa resuelve el problema que se planteó en la primera fase de este diseño y es el que a continuación se presenta. Con ello queda concluido el diseño e implementación del SME.

```
;INICIALIZACION DEL DISPLAY, PROCESO QUE SE EFECTUARA
;TODA VEZ QUE SE INICIE EL SISTEMA.

        ORG000H
        JMPRET0
        ORG003H
        JMPRTEC
        ORG010H
;RETARDO DE APROXIMADAMENTE 15 MILISEGUNDOS.
RET0:   MOVA,#0FFH
RET0A:  DECA
        JNZRET0A
;FUNCTION SET (PRIMERA VEZ)
ETIQ:   MOVA,#03FH
        OUTLP1,A
        MOVA,#0A0H;HABILITA EL DISPLAY Y SE DESHABILITA MEMORIA
        OUTLP2,A
        MOVA,#07FH;RETARDO PARA QUE DETENGA LOS DATOS DE
RETA:   DECA    ;ENABLE, R/W Y RS DEL DISPLAY.
        JNZRETA
        MOVA,#040H;COLOCA EN CERO EL ENABLE DEL DISPLAY
        OUTLP2,A
;RETARDO DE APROXIMADAMENTE 4.1 ms.
RET1:   MOVA,#0FFH
RET1A:  DECA
```

```
            JNZ RET1A
;FUNCTION SET (SEGUNDA VEZ)
ETIQ2 :     MOV A,#03FH
            OUTL P1,A
            MOV A,#0A0H
            OUTL P2,A
            MOV A,#07FH ;RETARDO PARA QUE DETENGA LOS DATOS DEL
RETB :      DEC A    ;ENABLE, R/W Y RS DEL DISPLAY.
            JNZ RETB
            MOV A,#040H ;COLOCA EN CERO EL ENABLE DEL DISPLAY
            OUTL P2,A
;RETARDO DE APROXIMADAMENTE 100 us.
RET2 :      MOV A,#0FFH
RET2A:      DEC A
            JNZ RET2A


;FUNCTION SET (TERCERA VEZ).
ETIQ3 :     MOV A,#03FH
            OUTL P1,A
            MOV A,#0A0H
            OUTL P2,A
            MOV A,#07FH ;RETARDO PARA QUE DETENGA LOS DATOS DEL
RETC :      DEC A    ;ENABLE, R/W Y RS DEL DISPLAY.
            JNZ RETC
            MOV A,#040H ;COLOCA EN CERO EL ENABLE DEL DISPLAY
            OUTL P2,A
;FUNCTION SET (CUARTA VEZ)
ETIQ4 :     MOV A,#038H
            OUTL P1,A
            MOV A,#0A0H
            OUTL P2,A
            MOV A,#07FH ;RETARDO PARA QUE DETENGA LOS DATOS DEL
RETD :      DEC A    ;ENABLE, R/W Y RS DEL DISPLAY.
            JNZ RETD
            MOV A,#040H ;COLOCA EN CERO EL ENABLE DEL DISPLAY
            OUTL P2,A
;ENCENDIDO DEL DISPLAY
ETIQ5 :     MOV A,#00CH
            OUTL P1,A
            MOV A,#0A0H ;HABILITA DISPLAY Y RS=0, R/W=0
            OUTL P2,A
```

```
            MOVA,#07FH ;RETARDO PARA QUE DETENGA LOS DATOS DEL
RETE :      DEC A   ;ENABLE, R/W Y RS DEL DISPLAY.
            JNZ RETE
            MOVA,#040H ;COLOCA EN CERO EL ENABLE DEL DISPLAY
            OUTL P2,A
;LIMPIAR LA PANTALLA DEL DISPLAY
ETIQ6 :     MOVA,#001H
            OUTL P1,A
            MOVA,#0A0H  ;HABILITA DISPLAY Y RS=0, R/W=0
            OUTL P2,A
            MOVA,#07FH ;RETARDO PARA QUE DETENGA LOS DATOS DEL
RETF :      DEC A   ;ENABLE, R/W Y RS DEL DISPLAY.
            JNZ RETF
            MOVA,#040H ;COLOCA EN CERO EL ENABLE DEL DISPLAY
            OUTL P2,A

;ENTRY MODE SET (COLOCA AL DISPLAY EN MODO ENTRADA)
ETIQ7 :     MOVA,#06H ;PUEDE CAMBIAR POR 007H, DEPENDIENDO
            OUTL P1,A ;DEL CORRIMIENTO HACIA LA DERECHA O IZQUIERDA.
            MOVA,#0A0H ;SE ENVIAN DATOS DE LOS BITS DE CONTROL
            OUTL P2,A ;A TRAVES DEL PUERTO DOS.
            MOVA,#07FH ;RETARDO PARA QUE DETENGA LOS DATOS DEL
RETG :      DEC A   ;ENABLE, R/W Y RS DEL DISPLAY.
            JNZ RETG
            MOVA,#040H ;COLOCA EN CERO EL ENABLE DEL DISPLAY
            OUTL P2,A
;PROCEDIMIENTO QUE ENCIENDE EL DISPLAY Y EL CURSOR.
ETIQ8 :     MOVA,#00FH ;EL CURSOR ESTARA PARPADEANDO
            OUTL P1,A
            MOVA,#0A0H
            OUTL P2,A
            MOVA,#07FH ;RETARDO PARA QUE DETENGA LOS DATOS DEL
RETH :      DEC A   ;ENABLE, R/W Y RS DEL DISPLAY.
            JNZ RETH
            MOVA,#040H ;COLOCA EN CERO EL ENABLE DEL DISPLAY
            OUTL P2,A
;PROCEDIMIENTO QUE COLOCA EL CURSOR EN EL PRIMER RENGLON
            MOVA,#080H
            OUTL P1,A
            CALL ESPERA2
            MOVA,#0A0H
            OUTL P2,A
```

```
                NOP
                NOP
                MOV A,#040H
                OUTL P2,A
                NOP
;PROCEDIMIENTO PARA LA PRIMERA PANTALLA
LET1:           EN I    ;HABILITA INTERRUPCIONES EXTERNAS
                SEL RB0
                MOV A,#040H ;COLOCA EN CEROS LOS BITS DE CONTROL
                OUTL P2,A
                CALL VER
                MOV R4, #053H ;DATOS PARA LA LETRA S
                CALL RDISPLAY
                CALL VER
                MOV R4, #049H ;DATOS PARA LA LETRA I
                CALL RDISPLAY
                CALL VER
                MOV R4, #04DH ;DATOS PARA LA LETRA M
                CALL RDISPLAY
                CALL VER
                MOV R4, #055H ;DATOS PARA LA LETRA U
                CALL RDISPLAY
                CALL VER
                MOV R4, #04CH ;DATOS PARA LA LETRA L
                CALL RDISPLAY
                CALL VER
                MOV R4, #041H ;DATOS PARA LA LETRA A
                CALL RDISPLAY
                CALL VER
                MOV R4, #044H ;DATOS PARA LA LETRA D
                CALL RDISPLAY
                CALL VER
                MOV R4, #04FH ;DATOS PARA LA LETRA O
                CALL RDISPLAY
                CALL VER
                MOV R4, #052H ;DATOS PARA LA LETRA R
                CALL RDISPLAY
                CALL VER
                MOV R4, #020H ;DATOS PARA DEJAR UN ESPACIO
                CALL RDISPLAY ;EN BLANCO
                CALL VER
                MOV R4, #044H ;DATOS PARA LA LETRA D
```

```
        CALL RDISPLAY
        CALL VER
        MOV R4,#045H ;DATOS PARA LA LETRA E
        CALL RDISPLAY
        CALL VER
        MOV R4,#020H ;DATOS PARA DEJAR UN ESPACIO
        CALL RDISPLAY ;EN BLANCO
        CALL VER
        MOV R4,#04DH ;DATOS PARA LA LETRA M
        CALL RDISPLAY
        JMP REP1
        ORG 0100H
REP1:   CALL VER
        MOV R4,#045H ;DATOS PARA LA LETRA E
        CALL RDISPLAY
        CALL VER
        MOV R4,#04DH ;DATOS PARA LA LETRA M
        CALL RDISPLAY
        CALL VER
        MOV R4,#04FH ;DATOS PARA LA LETRA O
        CALL RDISPLAY
        CALL VER
        MOV R4,#052H ;DATOS PARA LA LETRA R
        CALL RDISPLAY
        CALL VER
        MOV R4,#049H ;DATOS PARA LA LETRA I
        CALL RDISPLAY
        CALL VER
        MOV R4,#041H ;DATOS PARA LA LETRA A
        CALL RDISPLAY
;PROCEDIMIENTO QUE CAMBIA DE RENGLON
CAMBIA: SEL RB1
        MOV R5,#0C1H
        CALL RENG
LET2:   SEL RB0
        MOV A,#040H ;COLOCA EN CEROS LOS BITS DE CONTROL
        OUTL P2,A
        CALL VER
        MOV R4,#046H ;DATOS PARA LA LETRA F
        CALL RDISPLAY
        CALL VER
        MOV R4,#041H ;DATOS PARA LA LETRA A
```

```
CALL RDISPLAY
CALL VER
MOV R4, #043H ;DATOS PARA LA LETRA C
CALL RDISPLAY
CALL VER
MOV R4, #02EH ;DATOS PARA EL PUNTO
CALL RDISPLAY
CALL VER
MOV R4, #020H ;DATOS PARA DEJAR UN ESPACIO
CALL RDISPLAY ;EN BLANCO
CALL VER
MOV R4, #044H ;DATOS PARA LA LETRA D
CALL RDISPLAY
CALL VER
MOV R4, #045H ;DATOS PARA LA LETRA E
CALL RDISPLAY
CALL VER
MOV R4, #020H ;DATOS PARA DEJAR UN ESPACIO
CALL RDISPLAY ;EN BLANCO
CALL VER
MOV R4, #049H ;DATOS PARA LA LETRA I
CALL RDISPLAY
CALL VER
MOV R4, #04EH ;DATOS PARA LA LETRA N
CALL RDISPLAY
CALL VER
MOV R4, #047H ;DATOS PARA LA LETRA G
CALL RDISPLAY
CALL VER
MOV R4, #045H ;DATOS PARA LA LETRA E
CALL RDISPLAY
CALL VER
MOV R4, #04EH ;DATOS PARA LA LETRA N
CALL RDISPLAY
CALL VER
MOV R4, #049H ;DATOS PARA LA LETRA I
CALL RDISPLAY
CALL VER
MOV R4, #045H ;DATOS PARA LA LETRA E
CALL RDISPLAY
CALL VER
MOV R4, #052H ;DATOS PARA LA LETRA R
```

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

```
        CALLRDISPLAY
        CALLVER
        MOVR4,#049H ;DATOS PARA LA LETRA I
        CALLRDISPLAY
        CALLVE
        MOVR4,#041H ;DATOS PARA LA LETRA A
        CALLRDISPLAY
CAMBIA1: SELRB1
        MOVR5,#09BH
        CALLRENG
LET22:  SELRB0
        MOVA,#040H ;COLOCA EN CEROS LOS BITS DE CONTROL
        OUTLP2,A
        CALLVER
        MOVR4,#055H ;DATOS PARA LA LETRA U
        CALLRDISPLAY
        CALLVER
        MOVR4,#04EH ;DATOS PARA LA LETRA N
        CALLRDISPLAY
        CALLVER
        MOVR4,#041H ;DATOS PARA LA LETRA A
        CALLRDISPLAY
        CALLVER
        MOVR4,#04DH ;DATOS PARA LA LETRA M
        CALLRDISPLAY
CAMBIA2: SELRB1
        MOVR5,#0DFH
        CALLRENG
LET4:   SELRB0
        MOVA,#040H ;COLOCA EN CEROS LOS BITS DE CONTROL
        OUTLP2,A
        CALLVER
        MOVR4,#043H ;DATOS PARA LA LETRA C
        CALLRDISPLAY
        CALLVER
        MOVR4,#04FH ;DATOS PARA LA LETRA O
        CALLRDISPLAY
        CALLVER
        MOVR4,#04EH ;DATOS PARA LA LETRA N
        CALLRDISPLAY
        CALLVER
        MOVR4,#054H ;DATOS PARA LA LETRA T
```

```
                CALL RDISPLAY
                JMP RP2
                ORG 0200H
RP2:            CALL VER
                MOV R4,#02EH ;DATOS PARA EL PUNTO
                CALL RDISPLAY
                CALL VER
                MOV R4,#07FH ;DATOS PARA LA TECLA ENTER
                CALL RDISPLAY
                CALL ENTRAR1
;LIMPIAR LA PANTALLA DEL DISPLAY
ETIQ6A:         MOV A,#001H
                OUTL P1,A
                MOV A,#0A0H  ;HABILITA DISPLAY Y RS=0, R/W=0
                OUTL P2,A
                CALL ESPERA2 ;RETARDO PARA DETENER BITS DE CONTROL
                MOV A,#040H ;COLOCA EN CERO EL ENABLE DEL DISPLAY
                OUTL P2,A
;PROCEDIMIENTO QUE ESCRIBE EL LETRERO PRINCIPAL DE LA PANTALLA
                CALL LETPRI;LLAMA A LA RUTINA DEL LETRERO DIR Y CONT.
;PROCEDIMIENTO QUE ALMACENA EN RAM LOS DATOS PARA EL CONTADOR HEXADECIMAL
                SEL RB1
                MOV R0,#032H;COLOCA EL APUNTADOR R0 EN LA DIR. 32H DE RAM
                MOV @R0,#030H;DATO PARA EL NUMERO 0
                INC R0
                MOV @R0,#031H;DATO PARA EL NUMERO 1
                INC R0
                MOV @R0,#032H;DATO PARA EL NUMERO 2
                INC R0
                MOV @R0,#033H;DATO PARA EL NUMERO 3
                INC R0
                MOV @R0,#034H;DATO PARA EL NUMERO 4
                INC R0
                MOV @R0,#035H;DATO PARA EL NUMERO 5
                INC R0
                MOV @R0,#036H;DATO PARA EL NUMERO 6
                INC R0
                MOV @R0,#037H;DATO PARA EL NUMERO 7
                INC R0
                MOV @R0,#038H;DATO PARA EL #8
                INC R0
                MOV @R0,#039H;DATO PARA EL #9
```

```
            INCR0
            MOV@R0,#041H;DATO PARA EL #A
            INCR0
            MOV@R0,#042H;DATO PARA EL #B
            INCR0
            MOV@R0,#043H;DATO PARA EL #C
            INCR0
            MOV@R0,#044H;DATO PARA EL #D
            INCR0
            MOV@R0,#045H;DATO PARA EL #E
            INCR0
            MOV@R0,#046H;DATO PARA EL #F
;AQUI TERMINA LA CARGA DE DATOS EN RAM.
;PROCEDIMIENTO QUE DIRECCIONA LA MEMORIA DE PRUEBA Y ESCRIBE
;LOS DATOS DE ESTA EN LA PANTALLA DEL DISPLAY.
INICIA:     SELRB0
            MOVA,#040H ;DESHABILITA MEMORIA Y DISPLAY ANTES DE INICIAR
            OUTLP2,A ;CUALQUIER PROCESO.
            MOVR3,#008H;INICIA CONTADOR
            MOVR2,#000H;SOLO IMPORTAN LOS TRES BITS MENOS SIGNIFICATIVOS DE P2
            SELRB1
            MOVR0,#032H;INICIA APUNTADOR
            MOVA,R0
            MOVR2,A
CONTA2:     SELRB0
            MOV A, R2
            MOVR7,A ;CARGA LOS BITS MAS SIGNIFICATIVOS EN R7
            CALLNUM1 ;LLAMA A LA RUTINA QUE DESPLIEGA EL DIG.+ SIGNIF.
            SELRB0
            INCR2
            DECR3
            MOVR0,#0FFH
            MOVR1,#000H
            MOVA,R1 ;LA PRIMERA DIRECCION DE LA MEMORIA QUE ES 0000, SE
            MOVR6,A ;CARGA LOS BITS MENOS SIGNIFICATIVOS EN R6
            SELRB1
            MOVR3,#010H;CONTADOR
            MOVR1,#032H;INICIA APUNTADOR
            MOVR5,#095H;DIR. PANTALLA COLUMNA 2, RENGLON 3
            CALLRENG
            CALLCERO ;ESCRIBE UN CERO
            CALLNUM3;RUTINA QUE DESPLIEGA LA DIRECCION
```

```
            CALL LEEMEM ;LLAMA A LA SUBRUTINA QUE LEE LOS DATOS DE LA MEMORIA
            CALL RTEC1
            SEL RB0
            INC R1
            SEL RB1
            MOV R0,#032H;INICIA APUNTADOR
CONTA1:     SEL RB0
            MOV A, R1
            MOV R6,A ;CARGA LOS BITS MENOS SIGNIFICATIVOS
            CALL NUM2 ;LLAMA A LA RUTINA QUE DESPLIEGA EL 2o. DIGITO
            CALL NUM3 ;LLAMA A LA RUTINA QUE DESPLIEGA EL 3er DIGITO
            CALL LEEMEM ;LLAMA A LA SUBRUTINA QUE LEE LOS DATOS DE LA MEMORIA
            CALL RTEC1
            SEL RB0
            INC R1
            DEC R0
            SEL RB1
            MOV A,R3
            JZ REINI
REINI:      SEL RB0
            MOV A, R0
            JNZ CONTA1
            MOV A, R3
            JNZ CONTA2
            CALL FIN4 ;LLAMA RUTINA DE LETRERO PARA FINALIZAR
            MOV A,#040H ;DESHABILITA SISTEMA
            OUTL P2, A
            MOV A,#0C0H ;HABILITA TECLADO
            OUTL P2,A
ENTRADA:    IN S A, BUS ;LEE TECLADO
            JB1 LETFIN
            JMP ENTRADA
LETFIN:     SEL RB0
            MOV A,#040H ;DESHABILITA EL SISTEMA
            OUTL P2, A
            MOV A,#001H ;LIMPIAR LA PANTALLA
            OUTL P1, A
            MOV A, #0A0H ; HABILITA EL DISPLAY
            OUTL P2,A
            CALL ESPERA2
            MOV A,#040H ;DESHABILITA DISPLAY
            OUTL P2, A
```

```
        SELRB1
        MOVR5,#0C0H;DIRECCIONA EL SEGUNDO RENGLON
        CALL RENG
        SELRB0
        CALL VER
        MOVR4,#046H;LETRA F
        CALL RDISPLAY
        CALL VER
        MOVR4,#049H;LETRA I
        CALL RDISPLAY
        CALL VER
        MOVR4,#04EH;LETRA N
        CALL RDISPLAY
PARA:   NOP
        NOP
        NOP
        JMP PARA  ;FIN DE PROGRAMA, UNA VEZ QUE TERMINA DE LEER
                  ;SE MANTIENE EN ESTE CICLO
REINI:  SELRB1
        MOVR3,#010H
        MOVR1,#032H
        INCR0
        JMP REINI1
        ORG 0300H
LEEMEM: SELRB0
        CALL POSI ;POSICIONA EL CURSOR EN LA TERCERA LINEA LUGAR 11
        MOVA,R7
        OUTL P2,A ;ENVIA LOS BITS MAS SIGNIFICATIVOS DE LA DIRECCION
                  ;HABILITA LA MEMORIA Y DESHABILITA EL DISPLAY
        MOV A,R6
        OUTL P1,A ;ENVIA LOS BITS MENOS SIGNIFICATIVOS DE LA DIRECCION
        INS A,BUS  ;LEE LOS DATOS DE LA MEMORIA POR EL BUS
        MOV R5, A
        MOVA,#040H;DESHABILITA MEMORIA Y DISPLAY
        OUTL P2,A
        MOV A, R5
        JB7 RUT1  ;PASA A LA RURINA QUE DESPLIEGA UN "UNO"
        JMP RUT2  ;PASA A LA RUTINA QUE DESPLIEGA UN "CERO"
ET1:    MOV A, R5
        JB6 RUT3  ;PASA A LA RUTINA QUE DESPLIEGA UN "UNO"
        JMP RUT4  ;PASA A LA RUTINA QUE DESPLIEGA UN "CERO"
ET2:    MOV A, R5
```

```
                JB5 RUT5  ;PASA A LA RUTINA QUE DESPLIEGA UN "UNO"
                JMP RUT6  ;PASA A LA RUTINA QUE DESPLIEGA UN "CERO"
ET3:            MOV A, R5
                JB4 RUT7  ;PASA A LA RUTINA QUE DESPLIEGA UN "UNO"
                JMP RUT8  ;PASA A LA RUTINA QUE DESPLIEGA UN "CERO"
ET4:            MOV A, R5
                JB3 RUT9  ;PASA A LA RUTINA QUE DESPLIEGA UN "UNO"
                JMP RUT10 ;PASA A LA RUTINA QUE DESPLIEGA UN "CERO"
ET5:            MOV A, R5
                JB2 RUT11 ;PASA A LA RUTINA QUE DESPLIEGA UN "UNO"
                JMP RUT12 ;PASA A LA RUTINA QUE DESPLIEGA UN "CERO"
ET6:            MOV A, R5
                JB1 RUT13 ;PASA A LA RUTINA QUE DESPLIEGA UN "UNO"
                JMP RUT14 ;PASA A LA RUTINA QUE DESPLIEGA UN "CERO"
ET7:            MOV A, R5
                JB0 RUT15 ;PASA A LA RUTINA QUE DESPLIEGA UN "UNO"
                JMP RUT16 ;PASA A LA RUTINA QUE DESPLIEGA UN "CERO"
FINLEE:         CALL ESPERA ;RUTINA QUE HACE UN RETARDO PARE PODER VER LOS
                RET
;INICIAN LAS RUTINAS DE CONTROL
RUT1:           CALL UNO
                JMP ET1
RUT2:           CALL CERO
                JMP ET1
RUT3:           CALL UNO
                JMP ET2
RUT4:           CALL CERO
                JMP ET2
RUT5:           CALL UNO
                JMP ET3
RUT6:           CALL CERO
                JMP ET3
RUT7:           CALL UNO
                JMP ET4
RUT8:           CALL CERO
                JMP ET4
RUT9:           CALL UNO
                JMP ET5
RUT10: CALL CERO
                JMP ET5
RUT11: CALL UNO
                JMP ET6
```

```
RUT12: CALLCERO
       JMPET6
RUT13: CALL UNO
       JMPET7
RUT14: CALLCERO
       JMPET7
RUT15: CALL UNO
       JMPFINLEE
RUT16: CALLCERO
       JMPFINLEE
;INICIAN LAS RUTINAS DE DESPLIEGUE
UNO:   SELRB0
       MOVA,#040H ;DESHABILITA TANTO MEMORIA COMO DISPLAY
       OUTL P2,A
       CALL ESPERA2;RETARDO PARA DETENER LOS BITS DE CONTROL
       CALL VER
       MOV R4,#031H;ENVIA DATOS PARA DERSPLEGAR UN "UNO"
       CALL RDISPLAY
       RET    ;REGRESA A LA RUTINA PRINCIPAL LEE
CERO:  SELRB0
       MOVA,#040H ;DESHABILITA TANTO MEMORIA COMO DISPLAY
       OUTL P2,A
       CALL ESPERA2;RETARDO PARA DETENER LOS BITS DE CONTROL
       CALL VER
       MOV R4,#030H;ENVIA DATOS PARA DESPLEGAR UN "CERO"
       CALL RDISPLAY;DESPLIEGA EN LA PANTALLA EL CARACTER
       RET    ;REGRESA A LA RUTINA PRINCIPAL LEE
;RUTINA QUE DIRECCIONA LA POSICION 11 DEL TERCER RENGLON
POSI:  MOVA,#09EH
       OUTL P1,A
       CALL ESPERA2
       MOV A,#0A0H
       OUTL P2,A
       NOP
       NOP
       MOV A,#040H
       OUTL P2,A
       NOP
       RET
;RUTINA PARA CAMBIAR DE RENGLON EN EL DISPLAY
RENG:  SELRB1
       MOVA, R5 ;ENVIA DATOS DE LA DIRECCION
```

```
                OUTL P1,A ; EN EL DISPLAY.
                CALL ESPERA2;RETARDO
                MOV A,#0A0H ; HABILITA EL DISPLAY
                OUTL P2,A ; ENVIA LA INFORMACION POR EL P2
                NOP
                NOP
                MOV A,#040H; COLOCA EN CERO EL ENABLE DEL DISPLAY
                OUTL P2,A
                NOP
                RET
;RUTINA QUE DESPLIEGA EL LETRERO DE DIRECCION Y CONTENIDO.
LETPRI:         SEL RB0
                MOV A,#040H ;DESHABILITA TANTO MEMORIA COMO DISPLAY.
                OUTL P2,A
                CALL VER  ;VERIFICA LA BANDERA BUSY DEL DIPLAY
                MOV A,#0C0H ;DIRECCION DE LA 1A. POS. DEL SEGUNDO RENGLON
                OUTL P1,A
                CALL ESPERA2;RETARDO
                MOV A,#0A0H ;HABILITA DISPLAY
                OUTL P2,A
                NOP
                NOP
                MOV A,#040H ;DESHABILITA DISPLAY
                OUTL P2,A
                NOP
DESLET:         CALL VER  ;VERIFICA BANDERA BUSY
                MOV R4,#044H;DATOS PARA LA LETRA "D"
                CALL RDISPLAY;RUTINA DE DESPLIEGUE
                CALL VER
                MOV R4,#049H;DATOS PARA LA LETRA "I"
                CALL RDISPLAY
                JMP RP4
                ORG 0400H
RP4:            CALL VER
                MOV R4,#052H;DATOS PARA LA LETRA "R"
                CALL RDISPLAY
                CALL VER
                MOV R4,#02EH;DATOS PARA EL PUNTO "."
                CALL RDISPLAY
                CALL VER
                MOV R4,#020H;DATOS PARA UN ESPACIO EN BLANCO
                CALL RDISPLAY
```

```
CALLVER
MOVR4,#020H;DATOS PARA UN ESPACIO EN BLANCO
CALLRDISPLAY
CALLVER
MOVR4,#020H;DATOS PARA UN ESPACIO EN BLANCO
CALLRDISPLAY
CALLVER
MOVR4,#020H;DATOS PARA UN ESPACIO EN BLANCO
CALLRDISPLAY
CALLVER
MOVR4,#020H;DATOS PARA UN ESPACIO EN BLANCO
CALLRDISPLAY
CALLVER
MOVR4,#020H;DATOS PARA UN ESPACIO EN BLANCO
CALLRDISPLAY
CALLVER
MOVR4,#043H;DATOS PARA LA LETRA "C"
CALLRDISPLAY
CALLVER
MOVR4,#04FH;DATOS PARA LA LETRA "O"
CALLRDISPLAY
CALLVER
MOVR4,#04EH;DATOS PARA LA LETRA "N"
CALLRDISPLAY
CALLVER
MOVR4,#054H;DATOS PARA LA LETRA "T"
CALLRDISPLAY
CALLVER
MOVR4,#45H ;DATOS PARA LA LETRA "E"
CALLRDISPLAY
CALLVER
MOVR4,#04EH;DATOS PARA LA LETRA "N"
CALLRDISPLAY
CALLVER
MOVR4,#049H;DATOS PARA LA LETRA "I"
CALLRDISPLAY
CALLVER
MOVR4,#044H;DATOS PARA LA LETRA "D"
CALLRDISPLAY
CALLVER
MOVR4,#04FH;DATOS PARA LA LETRA "O"
CALLRDISPLAY;AQUI TERMINA LA RUTINA PARA DESPLEGAR EL LETRERO.
```

```
            RET
;RUTINA PARA EL DIGITO MAS SIGNIFICATIVO DEL CONTADOR HEXADECIMAL
NUM1:       SELRB1
            MOVA,R2
            MOVR0,A
            MOVA,@R0
            SELRB0
            MOVR4,A
            SELRB1
            MOVR5,#094H
            CALLRENG
            CALLVER
            CALLRDISPLAY
            SELRB1
            INCR2
            RET

;RUTINA PARA EL SEGUNDO DIGITO DEL CONTADOR HEXADECIMAL
NUM2:       SELRB1
            MOVA,@R0
            SELRB0
            MOVR4,A
            SELRB1
            MOVR5,#095H
            CALLRENG
            CALLVER
            CALLRDISPLAY
            RET
;RUTINA PARA EL DIGITO MENOS SIGNIFICATIVO (TERCERO) DEL CONT. HEXADECIMAL
NUM3:       SELRB1
            MOVA,@R1
            SELRB0
            MOVR4,A
            SELRB1
            MOVR5,#096H
            CALLRENG
            CALLVER
            CALLRDISPLAY
            SELRB1
            INCR1
            DECR3
            RET
```

```
;RUTINAS AUXILIARES EN EL DESPLEGADO.
VER:        MOV A,#0A8H ;ENVIA CONDICIONES E=1,RS=0,R/W=1 AL DISPLAY
            OUTL P2,A
            CALL ESPERA2;RETARDO PARA BITS DE CONTROL
VER1:  IN A,P1  ;LEE PUERTO 1 PARA VERIFICAR VALOR DE BUSY FLAG=1
            JC VER1
            RET
RDISPLAY:   SEL RB0
            MOV A,#070H ;BITS DE CONTROL E=1,RS=1,R/W=0 Y EM=1
            OUTL P2,A
            CALL ESPERA2;RETARDO PARA BITS DE CONTROL
ETDIS:  MOV A,R4  ;CODIGO PARA EL CARACTER A DESPLEGAR
            OUTL P1,A  ;SE ENVIA POR EL PUERTO 1
            NOP
            NOP
            NOP
            MOV A,#040H ;COLOCA BITS DE CONTROL EN DESHABILITACION
            OUTL P2,A
            CALL ESPERA2;RETARDO PARA DETERNER BITS DE CONTROL
            RET
;RUTINA DE RETARDO
ESPERA:     SEL RB0
            MOV R4,#050H
ESPERA1:    CALL ESPERA2
            DEC R4
            MOV A,R4
            JNZ ESPERA1
            RET
;RUTINA DE RETARDO PARA LOS DATOS DE HABILITADORES QUE SE CONTROLAN POR P2
ESPERA2:    MOV A,#0FFH
ESP2A:      DEC A
            JNZ ESP2A
            RET
;PROCEDIMIENTO DE ATENCION A INTERRUPCION EXTERNA DEL PIN 6 DEL MICRO
RTEC:       MOV A,#040H ;DESHABILITA SISTEMA
            OUTL P2,A
            CALL LETIN ;RUTINA DE LETRERO
            MOV A,#0C0H ;HABILITA TECLADO
            OUTL P2,A
LEETEC:     INS A,BUS  ;LEE EL TECLADO
            JB1 TFIN  ;IDENTIFICA SI SE PRESIONO FIN
            JMP LEETEC
```

```
;PROCEDIMIENTO PARA LA TECLA FIN.
TFIN:   MOVA,#040H ;DESHABILITA SISTEMA
        OUTLP2,A
        MOVA,#001H ;LIMPIA LA PANTALLA DEL DISPLAY
        OUTLP1,A
        MOVA,#0A0H ;HABILITA DISPLAY
        OUTLP2,A
        NOP
        NOP
        MOVA,#040H ;DESHABILITA DISPLAY
        OUTLP2,A
        SELRB1
        MOVR5,#09BH;DIRECCIONA LA PANTALLA AL PRIMER RENGLON
        CALLRENG
        SELRB0
        MOVA,#040H ;DESHABILITA SISTEMA
        OUTLP2,A
        CALLVER
        MOVR4,#046H;DATOS PARA LA LETRA "F"
        CALLRDISPLAY
        CALLVER
        MOVR4,#049H;DATOS PARA LA LETRA "I"
        CALLRDISPLAY
        CALLVER
        MOVR4,#4EH ;DATOS PARA LA LETRA "N"
        CALLRDISPLAY
TERM:   NOP
        NOP
        NOP
        JMPTERM
        ORG0500H
;RUTINA QUE IDENTIFICA LA TECLA DE PAUSA
RTEC1:  MOVA,#040H;DESHABILITA EL SISTEMA
        OUTLP2,A
        MOVA,#0C0H;HABILITA TECLADO
        OUTLP2,A
        CALLESPERA2;RETARDO
        CALLESPERA2;RETARDO
        INSA,BUS ;LEE EL TECLADO
        JB0PAUSA ;IDENTIFICA LA TECLA PAUSA Y HACE UN RETARDO.
SIGUE1: MOVA,#040H;DESHABILITA TECLADO
        OUTLP2,A
```

```
            RET
;PROCEDIMIENTO PARA LA TECLA PAUSA
PAUSA:      CALL ESPERA
            CALL ESPERA
            CALL ESPERA
            CALL ESPERA
            CALL ESPERA
            CALL ESPERA
            CALL ESPERA
            CALL ESPERA
            CALL ESPERA
            CALL ESPERA
            CALL ESPERA
            CALL ESPERA
            JMP SIGUE1

;RUTINA PARA LA TECLA DE ENTER
ENTRAR1:    MOV A, #040H ;DESHABILITA SISTEMA
            OUTL P2, A
            MOV A, #0C0H ;HABILITA TECLADO
            OUTL P2, A
ENT:        INS A, BUS  ;LEE TECLADO
            JB1 ENT1
            JMP ENT
            RET
;PROCEDIMIENTO PARA LA TECLA ENTRAR
ENT1:       MOV A, #040H ;DESHABILITA TECLADO
            OUTL P2, A
            JMP ETIQ6A
;RUTINA DEL LETRERO PARA LA INTERRUPCION EXTERNA
LETN:       SEL RB0
            MOV A, #040H ;DESHABILITA EL SISTEMA
            OUTL P2, A
            MOV A, #001H ;LIMPIAR LA PANTALLA
            OUTL P1, A
            MOV A, #0A0H ; HABILITA EL DISPLAY
            OUTL P2, A
            CALL ESPERA2
            MOV A, #040H ;DESHABILITA DISPLAY
            OUTL P2, A
            SEL RB1
            MOV R5, #0C0H ;DIRECCIONA EL SEGUNDO RENGLON
```

```
CALLRENG
SELRB0
CALLVER
MOVR4,#045H;LETRAF
CALLRDISPLAY
CALLVER
MOVR4,#049H;LETRAI
CALLRDISPLAY
CALLVER
MOVR4,#04EH;LETRAN
CALLRDISPLAY
CALLVER
MOVR4,#020H;ESPACIOENBLANCO
CALLRDISPLAY
CALLVER
MOVR4,#07FH;TECLAENTER
CALLRDISPLAY
SELRB1
MOVR5,#094H;DIRECCIONAELTERCERRENGLON
CALLRENG
SELRB0
CALLVER
MOVR4,#052H;LETRAR
CALLRDISPLAY
CALLVER
MOVR4,#045H;LETRAE
CALLRDISPLAY
CALLVER
MOVR4,#049H;LETRAI
CALLRDISPLAY
CALLVER
MOVR4,#04EH;LETRAN
CALLRDISPLAY
CALLVER
MOVR4,#049H;LETRAI
CALLRDISPLAY
CALLVER
MOVR4,#043H;LETRAC
CALLRDISPLAY
CALLVER
MOVR4,#049H;LETRAI
CALLRDISPLAY
```

```
        CALLVER
        MOVR4,#04FH;LETRA O
        CALLRDISPLAY
        CALLVER
        MOVR4,#020H;ESPACIO EN BLANCO
        CALLRDISPLAY
        CALLVER
        MOVR4,#07EH;TECLA DE RESET
        CALLRDISPLAY
        RET
FIN4:   MOVA,#040H ;DESHABILITA SISTEMA
        OUTLP2,A
        MOVA,#001H ;LIMPIA LA PANTALLA DEL DISPLAY
        OUTLP1,A
        MOVA,#0A0H ;HABILITA DISPLAY
        OUTLP2,A
        NOP
        NOP
        MOVA,#040H ;DESHABILITA DISPLAY
        OUTLP2,A
        SELRB1
        MOVR5,#09BH;DIRECCIONA LA PANTALLA AL PRIMER RENGLON
        CALLRENG
        SELRB0
        MOVA,#040H ;DESHABILITA SISTEMA
        OUTLP2,A
        CALLVER
        MOVR4,#043H ;DATOS PARA LA LETRA C
        CALLRDISPLAY
        CALLVER
        MOVR4,#04FH ;DATOS PARA LA LETRA O
        CALLRDISPLAY
        CALLVER
        MOVR4,#04EH ;DATOS PARA LA LETRA N
        CALLRDISPLAY
        CALLVER
        MOVR4,#054H ;DATOS PARA LA LETRA T
        CALLRDISPLAY
        JMPFIN41
FIN41:  CALLVER
        MOVR4,#02EH ;DATOS PARA EL PUNTO
        CALLRDISPLAY
```

```
        CALL VER
        MOV R4, #07FH ;DATOS PARA LA TECLA ENTER
        CALL RDISPLAY
        RET
FIN:    END    ;TERMINA PROGRAMA.
```

NOTA: Como etapa final de la implementación del "SME" se elaboró el circuito impreso utilizando la herramienta TANGO-PCB, sin embargo dado que no se encontraron los elementos necesarios para poder obtener la tarjeta impresa con pistas de cobre, no fue posible montar los circuitos en dicha tarjeta. Por lo que tomando en cuenta la información generada en TANGO se realizó un circuito sobre una tarjeta simulando la distribución de pistas.

En la figura 5.9 se muestra la tarjeta impresa que se generó en TANGO.

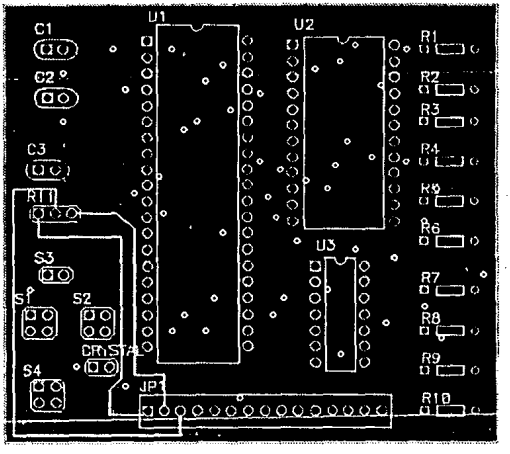


FIGURA 5.9. TARJETA IMPRESA DEL SME.

Para la utilización del SME se cuenta con un MANUAL DE USUARIO el cual es una guía sencilla que permitirá un mejor aprovechamiento (consultar APENDICE C).

CONCLUSIONES

CONCLUSIONES

El objetivo de desarrollar este trabajo fue relacionar el diseño de sistemas digitales con las herramientas de CAD (Diseño Asistido por Computadora), que se encuentran disponibles en la Facultad de Ingeniería; esta relación se llevó a cabo a través del diseño de un circuito que proporcionará a los alumnos una herramienta útil en el diseño de circuitos digitales.

Para el desarrollo del SME (Simulador de Memorias EPROM) se investigó entre los alumnos de la FI que cursan materias relacionadas al diseño digital sobre los conceptos más utilizados al diseñar y desarrollar sus proyectos, entre los más citados se encuentran: diseño de cartas ASM utlizando cuatro tipos de direccionamiento (Ruta-Liga, Entrada-Estado, Implicito y Formato-Variable), la implementación de las cartas ASM con diferentes tipos de componentes donde el más mencionado fue la memoria EPROM 2716 pero no el único ya que también utilizan microcontroladores y PAL's, los diseños se realizan manualmente porque no se tiene conocimiento de las herramientas CAD.

Actualmente la comunidad estudiantil de la FI no cuenta con las herramientas suficientes para el diseño, lo que propicia un desinterés por el desarrollo de sistemas digitales siendo ésto un problema que repercute de alguna manera en el avance tecnológico que ha tenido el país en estos últimos años. Por lo que el SME se presenta como una alternativa para contribuir a que el interés en el diseño digital con sistemas CAD se incremente.

Al iniciar el diseño del SME se presentaron algunos problemas como: escasas fuentes de información, acceso limitado al equipo requerido en la etapa de implementación, los cuales fueron superados paulatinamente a medida que se iba conociendo más sobre el tema. Tomando en cuenta los conocimiento previos que se tenian sobre el diseño digital y los adquiridos se logró que el número de componentes utilizados se redujera al mínimo posible obteniendo con esto una inversión menor, asi como también un dispositivo de operación simple que no requiere de mucho espacio, y que para su utilización no se requiere de un entrenamiento especial, ya que la presentación es amigable para el usuario, y ademas cuenta con un manual de usuario que contiene información sobre el funcionamiento y operación del mismo.

Durante el desarrollo de este proyecto hubo necesidad de involucrarse con componentes electrónicos, diseño y software orientados a los sistemas digitales, con lo cual se tuvo una visión más amplia con respecto a los conceptos que se habian adquirido en la aulas. Es importante resaltar que el costo tanto del software de diseño como del hardware asociado a él es elevado por lo que para su utilización es recomendable contar con una infraestuctura que genere circuitos electrónicos a nivel industrial, ya que a escala pequeña no resulta costeable, sin embargo es necesario tener conocimiento de estas herramientas dada la utilidad que tienen dentro del campo de la ingeniería.

El resultado de esta tesis (SME), se pone a disposición de la comunidad estudiantil con la finalidad de solucionar alguna de las carencias que se tienen en cuanto al equipo.

Se espera que este proyecto no solo le sirva al alumno como una herramienta en sus diseños sino que además contribuya a desarrollar más su ingenio y creatividad, lo cual en un futuro podría traer como consecuencia un incremento en el nivel de desarrollo del país.

APENDICES

APENDICE A.

DESCRIPCION DE HERRAMIENTAS CAD ORIENTADAS A ELECTRONICA

palCAD

palCAD es un paquete de software, enfocado básicamente al diseño de controladores partiendo del concepto de carta ASM; éste permite la minimización de funciones booleanas y la generación del mapa de fusibles en formato JEDET para su posterior utilización en la quema de fusibles del dispositivo PAL (Lógica de Arreglo Programable) ó GAL (Lógica de Arreglo Genérico).

Este paquete, fué desarrollado de tal manera que el usuario no requiera de conocimientos previos de computación, ya que se cuenta con una serie de menúes para la elaboración de las cartas ASM (Algoritmo de Máquina de Estados), donde se describen las secuencias o pasos, para elaborar una aplicación de diseño, y mediante una serie de manipulaciones de los atributos no gráficos obtener las funciones booleanas.

- **CARATERISTICAS**

  Las principales caracteristicas de este paquete son:

  a) Ambiente de trabajo de fácil manipulación

  b) Cuenta con una serie de iconos

  c) La manera de seleccionar una opción puede realizarse de dos formas: utilizando un "Mouse" ó "Teclado de funciones programadas"

  d) Las opciones de entrada del editor gráfico, están dadas por un conjunto de iconos, los cuales proporcionan una mejora en cuanto a espacio en video, ya que la descripción textual correspondiente a las funciones sería mayor.

  e) Dentro del menú se ofrecen los iconos básicos para el diseño de la carta ASM

  f) Además cuenta con acercamientos, traslaciones, borrado de elementos, limpiar el área de despliegue.

  g) Se puede almacenar el archivo de trabajo de la carta ASM en diskette. Así como también se puede realizar una impresión.

El menú principal de palCAD contiene las siguientes opciones:

1) DIRECTORIO: Muestra los archivos contenidos en el directorio de trabajo del Ambiente CAD para Diseño de Sistemas Digitales Orientado a PAL's de donde se puede elegir cualquier archivo existente para realizar modificaciones.

2) GENERACION DE FUNCIONES BOOLEANAS: Esta opción genera las funciones booleanas de la carta ASM previamente editada. Las funciones obtenidas y que presenta el sistema son: estado siguiente, salida en estado presente y salida condicional.

2.1) Formas canónicas o normales, para expresar la función booleana.

2.2) Reducción de funciones booleanas, utilizando el método de Quine-McCluskey

3) HERRAMIENTAS: Son una serie de utilerias que permiten al usuario realizar un respaldo de su información en disco flexible, realizar una impresión de la carta ASM, eliminación de algún archivo, así como también le permite regresar al menú principal del sistema.

4) SALIR DE SESION: Es empleado para abandonar la sesión.

Las características de **Hardware y Software** empleadas en el desarrollo del paquete **palCAD** son:

Equipo:	INB RT-PC
Monitor:	1024x1024 pixeles de resolución
Procesador 6150:	Procesador Motorola 68020 y tres procesadores en paralelo de punto flotante
Dispositivos de control:	Teclado de funciones programadas (LPKF), tableta digitalizadora y ratón
Disco duro:	2 discos de 70 Mb cada uno
Tarjeta gráfica:	Megapel
Memoria RAM:	12 Mb.
Estación de trabajo:	5081
Sistema operativo:	AIX 2.1
Estándar gráfico:	graPHIGS, versión 3.4.
Lenguaje de alto nivel:	C estándar

OrCAD

OrCAD es una herramienta para el diseño esquemático. Su fácil utilización de los menúes permiten la creación, edición, manipulación, almacenamiento e impresión de esquemas electrónicos.

El editor de esquemas para este paquete, es DRAFT, el cual permite crear, editar y salvar hojas esquemáticas. Ademas de que carga los manejadores de los dispositivos periféricos necesarios como: la impresora, el video, y graficador; así como las librerias necesarias.

♦CARACTERISTICAS.

Dentro de las características de este editor "DRAFT" se incluyen las siguientes:

- Acceso a más de 6000 elementos en las librerías
- Partes equivalentes a D'Morgan
- Crear líneas, buses, conectores, etiquetas, etc.
- Rotación y espejo de elementos
- Movimiento, copias y borrado de objetos ó bloques de objetos
- Disponibilidad de una malla visible de puntos
- Paneo automático de la hoja de trabajo
- 5 niveles de acercamiento
- Niveles limitados de jerarquía
- Directorio de librerías
- Busqueda de cadenas
- Opción de orientación a textos (verticales y horizontales)
- Soporta cinco tamaños de hojas de trabajo

Las librerias que contiene **OrCAD** son:

- TTL
- CMOS
- MEMORY
- ECL
- DISCRETAS
- ANALOGICAS
- MICROPROCESADORES
- DISPOSITIVOS PERIFERICOS
- ALTERA
- ETC.

Las utilerias con que cuenta **OrCAD** son:

TREELIST: Rastrea la organización jerárquica de las hojas y despliega su estructura, nombre de la hoja raíz, así como las asociadas a ésta.

ANNOTATE: Actualiza automáticamente cualquier componente dentro de un diseño esquemático, numerándolo de forma correlativa, si éste se ha incluido al diseño esquemático, actualiza el número de pines asociados a dicho componente.

PRINTALL: Imprime una hoja esquemática o grupo de hojas esquemáticas utilizando una impresora.

PLOTALL: Imprime una hoja esquemática o grupo de hojas esquemáticas utilizando un graficador.

PARTLIST: Reporta todos los elementos utilizados en el diseño esquemático ó grupo de hojas esquemáticas (diseño jerárquico).

ERC: Lleva a cabo una verificación de reglas eléctricas comunes, para advertir si existe algún error.

NETLIST: Genera un archivo (NETLIST) adecuado para ser capturado por un programa de colocación de componentes (PCB), en diferentes formatos empleados por programas de PCB de los más usuales.

BACKANNO: Actualiza las referencias correspondientes a componentes que han sido agregados o modificados dentro de un diseño esquemático después de la ejecución de Annotate.

CROSSREF: Muestrea cualquier diseño esquemático, recopilando información sobre los componentes empleados, y crea un archivo de referencia que contiene la localización de cada componente.

CLEANUP: Verifica las conexiones, buses, uniones, etiquetas y cualquier objeto, mostrando mensajes oportunos cuando detecta algún elemento gráfico duplicado, borrándolo.

EXTRACT: Crea archivos fuente, para ser utilizados por el módulo de PLD'S a partir de archivos creados con STD.

FLDATTRB: Modifica los atributos de los componentes que integran un diagrama esquemático, ya sea de bloques, jerárquico o simple, haciendo que éstos sean visibles o invisibles.

FLDSTUFF: Modifica los atributos de aquellos componentes que han sido definidos a través de un archivo tipo texto y que previamente han sido combinados mediante KEY FIELD.

LIBARCH: Permite crear una librería fuente a partir de los componentes empleados en un diseño esquemático.

SIMPLE: Convierte diseños jerárquicos complejos en jerarquías simples.

XFEROVL: Transfiere información de configuración desde una versión a otra del archivo ORCADSDT.OVL. Esto es posible sólo con versiones de OrCAD III superiores a las 3.12.

DECOMP: Esta utilería, es un decompilador de librerías de OrCAD, (archivos con extensión .LIB) a librerías en archivos fuentes.

COMPOSER: Es un compilador de librería; se emplea para convertir la librería de archivo fuente en librería de archivo objeto para ser utilizado por DRAFT.

♦ REQUERIMIENTOS DE HARDWARE

A continuación se detalla una configuración mínima para que el sistema funcione con un rendimiento adecuado:

- Una computadora PC/XT/AT o compatible
- Coprocesador matemático opcional
- 640 Kbytes de RAM
- Sistema operativo MS-DOS versión 2.0 o posterior
- Disco duro de 20 Mbytes, capacidad mínima
- Tarjeta gráfica recomendable, VGA de color
- Uno o dos puertos seriales (para "mouse" y "plotter")
- Un puerto paralelo
- "Mouse"
- Graficador HP o compatible.

TANGO PLUS

TANGO PLUS esta constituido por Tango-PCB PLUS y Tango-Route PLUS, así como también de la parte de diseño esquemático.

Tango-PCB PLUS es un paquete de software con una gran cantidad de características en el arte del trabajo de diseño y producción de tabletas de circuito impreso (PCB), con lo cual se logra un bajo costo.

Con dicho paquete se pueden realizar y obtener el diseño de una tableta en la computadora, y plasmar finalmente un trabajo de gran calidad utilizando un "plotter". El diseño final puede ser almacenado en disco para tenerlo más tarde de referencia o poder realizarle modificaciones.

TANGO-PCB PLUS.

Tango-PCB PLUS corre en las series de computadoras IBM-PC/XT/AT/PS2 y compatibles.

♦ CARACTERISTICAS.

El paquete incluye las siguientes características:

- Un medio ambiente sencillo para aprender a diseñar tabletas para PC, incluyendo menús en la parte superior e inferior, cajas de diálogo, paletas, y ayuda.
- La capacidad para manejar algún tamaño de tableta es de hasta 32x32 pulgadas.

- La memoria expandida (EMS) que soporta, es de hasta 32 Megabytes.
- Soporte para 19 capas, incluyendo la capa de componentes y soldadura, las del centro empezando en la 1 y terminando en la 4, planos de potencia y tierra, tope y base de la malla metálica, tope y base de la mascara de soldadura, tableta, conexiones, observaciones y título.
- La anchura de la pista es variable.
- Tamaños de texto variables con reflexión y rotación.
- Un conjunto único de tres rejillas: absoluta, relativa y visible. La rejilla absoluta y la visible tienen su origen en el punto 0,0 y pueden tomar un valor de 1 a 1000.
- Superficie con un soporte total para montar diseños de tecnología (SMT).
- Librerías de componentes estándar, incluyendo partes SMT.
- Cuenta con un potente bloque de operaciones para mover, copiar, rotar y borrar dentro o fuera de las áreas marcadas de la tableta.
- Soporte para casi todos los dispositivos de video más populares, icluyendo EGA, VGA y otros adaptadores gráficos de color con alta resolución, así como también el MCGA y el adaptador hercules monocromático.
- Soporte para un "host" de "plotters" e impresoras.
- Soporte para "fotoplotters" que acepten archivos de formato Gerber y máquinas de instrucción que acepten archivos de instrucción "Excellon N/C".
- Documentación completa que incluye ilustraciones, manual de referencia, tutorial y una amplia ayuda.

♦ REQUERIMIENTOS DE HARDWARE

Los requerimientos mínimos de hardware para que **Tango PCB PLUS** funcione, son los siguientes:

- Computadora IBM-PC/XT/AT/PS2 o compatible. En la mayoría de los casos Tango-PCB PLUS emplea en sus operaciones cálculos enteros (no de punto flotante).
- 640 Kbyte en RAM
- Sistema operativo MS-DOS ó PC-DOS, versión 2.1 o posterior.
- Dos drives para disco, un drive para diskette (360 Kbyte, 720 Kbyte, 1.2 Mbyte, 1.44 Mbyte, etc) para copiar y salvar programas así como también archivos de datos. Y un drive para operar un disco duro. Un sistema con un sólo drive no es muy conveniente.
- Mouse de Microsoft, Tango-PCB PLUS trabaja con el mouse de Microsoft y con algún otro que sea 100% compatible.
- Adaptadores gráficos de color o monocromáticos. Tango-PCB PLUS está diseñado para utilizarse con las siguientes tarjetas:
  - Hercules (720x348; monocromático)
  - Everex EGA (640x350; 16 colores)
  - IBM EGA (640x350; 16 colores)
  - IBM MCGA (640x480; monocromático)
  - IBM VGA (640x480; 16 colores)
  - Paradise VGA profesional (800x600; 16 colores)

Headland (Video 7) de escritura rápida (800x600; 16 colores.
Headland (Video 7) Vega de lujo (640x480; 16 colores)
Headland (Video 7) VGA (720 x 540; 16 colores)
Headland (Video 7) VRAM (1024x768; 16 colores)

- Monitor a color o monocromático.
- Impresoras. Tango-PCB PLUS soporta las siguientes impresoras y sus compatibles:
  Apple LaserWriter
  Epson series FX/LQ
  Hewlett-Packard DeskJet (150/300 puntos por pulgada)
  Hewlett-Packard PaintJet Plus (150x300 puntos por pulgada)
  IBM Proprinter
  IDS 480
  Okidata series 92 y 192.
  Star Micronix
  Toshiba, en sus series.
- Plotter (plumilla y tinta). Tango-PCB está diseñado para la utilización de los siguientes plotters:
  Calcomp 104X(PCI)
  Hewlett-Packard (HPGL)
  Houston Instruments (DMPL)
  Roland DXY 800.
- Puerto paralelo. Tango-PCB PLUS requiere un dispositivo de seguridad para funcionar. El dispositivo de seguridad evita la utilización del software sin autorización, sin embargo esto no quiere decir que se puedan hacer copias de respaldo. El estándar del dispositivos de seguridad es una tecla externa que conecta al puerto paralelo LPT1 ó LPT2. El puerto paralelo debe ser compatible con IBM.

TANGO-ROUTE PLUS.

Tango-Route PLUS es un ruteador sofisticado que funciona automáticamente, fué diseñado para realizar a mayor velocidad el diseño de una tableta de circuito impreso (PCB), se le denomina ruteo al diseño de las pistas que conectan a los componentes sobre el PCB. Cuando estas pistas se realizan completamente a mano, el proceso se lleva mucho tiempo.

Empezando con un archivo "net list" de conexiones, se puede utilizar Tango-PCB para definir los límites de la tableta y colocar los componentes. Tango-Route podrá entonces utilizar el archivo PCB y la información de conexión en el archivo "net list" para rutear los segmentos de pista sobre la tableta.

Tango-Route es un ruteador multicapas, esto implica que cualquiera de las capas habilitadas puede ser utilizada para la terminación de una ruta, opuesto a la técnica "pares de capas", la cual es la técnica más comúnmente encontrada, donde las capas se colocan de par en par en el programa. Tango-Route PLUS alcanza un alto nivel de terminación mucho más rápido de lo que es posible con un ruteo manual. Este paquete reduce en forma significativa el tiempo de diseño del PCB incrementando su productividad.

Tango-Route PLUS es un ruteador multicapas con algoritmos de ruteo diseñados para obtener una terminación a mayor velocidad. También incluye limpieza de pistas, lo cual combina o elimina

segmentos de pistas, para obtener una tableta más fabricable y un archivo PCB de salida más pequeño.

♦ **CARACTERISTICAS.**

Tango-Route PLUS incluye las siguientes características:

- Trabaja directamente con Tango-PCB y Tango-schematic (también como otros paquetes de captura esquemáticos).
- Rutea tabletas de 8 capas, conteniendo 6 capas de señales más los planos de tierra y potencia. El usuario es el que especifica las vías horizontales ó verticales para cada capa habilitada.
- Utiliza las estrategias de ruteo propietario y laberinto, más las fases de optimización que incluyen la limpieza de pistas y minimización de vías para mejorar la calidad de fabricación y la estética del ruteo.
- Incluye el ruteo del ancho de la pista para tierra y potencia.
- Acomoda las tabletas de hasta 32x32 pulgadas.
- Rutea en 25, 20, 16.7, 12.5 y 10000 rejillas, con capacidad para deshabilitar la rejilla. El tamaño de la pista y de la vía esta basado en la selección de la rejilla.
- Realiza rutas de 90 y 45 grados para densidad alta.
- Despliega gráficamente el proceso de ruteo, con un reporte de estado que contínuamente se actualiza.
- Permite conexiones críticas que son preruteadas con Tango PCB.
- Realiza verificaciones eléctricas y de espacios libres sobre todas las secciones preruteadas.
- Realiza la colocación inteligente de vías para niveles de terminación más altos.
- Permite que todos los pasos del ruteo puedan ser colocadas en encendido ó apagado por el usuario.
- Coloca las conexiones no ruteadas en la capa de conexiones y los lista en el archivo "Look" para acomodar la terminación manual de la tableta.
- Lista los mensajes de estadística y errores del trabajo de ruteo en el archivo "Look".

♦ **REQUERIMIENTOS DE HARDWARE.**

Los requerimientos mínimos de hardware para la ejecución de **Tango-Route PLUS**, son los siguientes:

- Computadora IBM-PC/XT/AT/PS2 o compatible.
- 640 Kbytes en RAM (o más, dependiendo de la aplicación)
- Soporta Memoria Expandida
- MS-DOS o PC-DOS, versión 2.0 o posterior.
- Dos manejadores para disco: uno de 360 Kbytes ó 1.2 Mbytes, para copiar y guardar los programas y archivos de datos. Y uno para operar un disco duro. Un sistema con un solo manejador no es muy conveniente.
- Adaptador gráfico de color o monocromático.
- Monitor a color o monocromático.
- "Mouse"
- Puerto paralelo.

UP600.

El UP600 es un Programador de Dispositivos Universal basado en una IBM-PC/XT/AT capaz de programar la mayoría de los dispositivos programables que existen en el mercado actual incluyendo EPROMS, EEPROMS, PALs, GALs, IFLs, EPLDs, PEELs, PROMs Bipolares, Microcomputadoras. Al mismo tiempo el UP600 es un software de gran calidad que permite programar virtualmente todos los dispositivos programables del futuro que tengan 40 pines o menos. Esto es posible por la estructura del hardware básico de la unidad, la cual proporciona manejadores de pin con voltajes de programación generados D/A (digital a analógico) para software independiente controlado.

♦ CARACTERISTICAS.

- Diseño de hardware universal: cada uno de los 40 pines puede ser colocado a cualquier voltaje entre 5-25 volts con una resolución de 0.1 V. Esto proporciona al UP600 la capacidad de programación de la mayoría de los dispositivos lógicos programables y memorias (PALs, GALs, IFLs, EPLDs, EPROMs, EEPROMs, PROMs bipolares y microcomputadoras de un solo circuito integrado).
- La comunicación con la PC "host" es realizada a través de una tarjeta adaptadora en paralelo diseñada por el fabricante, la cual mapea el registro de control del programador dentro del espacio E/S (Entrada/Salida) de la computadora.
- Programa de control que maneja el menú de forma tal que se presenta al usuario amigablemente.
- El programa de control del UP600 está diseñado de tal forma que permite agregarle partes a las librerías sin requerir el programa principal para ser recompilado. Esto nos proporciona una actualización rápida del software.
- La selección del número de parte y el fabricante asegura el método de programación correcto.
- Los algoritmos Built-in normal, Intelligent I & II, y Quick-pulse proporcionan una programación rápida.
- La facilidad de programación del ancho de palabra permite la programación de palabras de 8/16/32 bits.
- Editor de pantalla completa de buffer de memoria poderoso.
- Soporte de prueba funcional para dispositivos lógicos.
- Editor de pantalla completa para el mapa de fusibles y vectores de prueba.
- Soporta formatos industriales de transmisión de datos estándar: Intel Hex 86/88, Tektronix Hex, Motorola Hex, y JEDEC.
- Prueba funcional para lógica TTL, lógica CMOS, RAMs estáticas y dinámicas.

♦ CONTENIDO DEL PAQUETE UP600.

Hardware.

- Unidad de programación UP600.
- Tarjeta de interfase de PC paralela.

- Cable de interconexión.
- Manual del usuario del UP600.

Software.

- Disco manejador del UP600.
- Disco(s) de librerías.

♦ REQUERIMIENTOS DE LA COMPUTADORA "HOST".

Para utilizar el Programador de Dispositivos Universal UP600 se debe tener la siguiente configuración mínima:

- Una IBM PC/XT/AT o compatible con una memoria de por lo menos 512 Kbytes, y una fuente de poder de 150 watts.
- Unidad de disco flexible de 5 1/4 " de doble densidad ya sea de 360 Kbytes o 1.2 Mbytes y un manejador de disco duro.
- Monitor monocromático o de color.
- PC/MS-DOS versión 2.0 o posteriores.

♦ ESTRUCTURA DEL MENU.

A continuación se presenta el árbol de comandos del menú.

MENU PRINCIPAL
- Part List
- Manufacturer List
- QUICK COPY
- LOAD DEVICE
  - Load Logic Device
  - Load Memory Device
- PROGRAM DEVICE
  - Program Logic Device
  - Program Memory Device
- VERIFY DEVICE
  - Verify Logic Device
  - Memory Device
- MORE COMMANDS
  - Edit Data
    - Edit Logic
      - Edit Fuse Map
      - Edit Vector
      - Fill Fuse Map
      - Clear Vectors

Edit Memory
Edit Memory
Complement Data
Move Data
Fill Memory
Transfer Data
Select Format
Input From Disk
Output To Disk
Device Test

♦ **TECLAS DE FUNCION.**

[↑]	Mueve el cursor al campo o línea previos
[↓]	Mueve el cursor al campo o línea siguientes
[←]	Mueve el cursor al campo, carácter o celda izquierda
[→]	Mueve el cursor al campo, carácter o celda derecha
[Home]	Mueve el cursor a la parte superior de la página
[End]	Mueve el cursor al final de la página
[Ctrl][Home]	Mueve el cursor a la parte superior del buffer
[Ctrl][End]	Mueve el cursor al final del buffer
[Del]	Borra un carácter hacia la derecha
[Backspace]	Borra un carácter hacia la izquierda
[Ins]	Acciona la opción de sobreescritura o inserción
[Esc]	Regresa al nivel previo
[Enter]	Realiza la selección
[F9]	Suspende a DOS

APENDICE B.

DISPOSITIVOS UTILIZADOS PARA LA IMPLEMENTACION DEL SIMULADOR DE MEMORIAS EPROM's.

MICROPROCESADOR 8749.

La arquitectura de este microprocesador se describe en el diagrama a bloques que se ilustra en la figura 1.

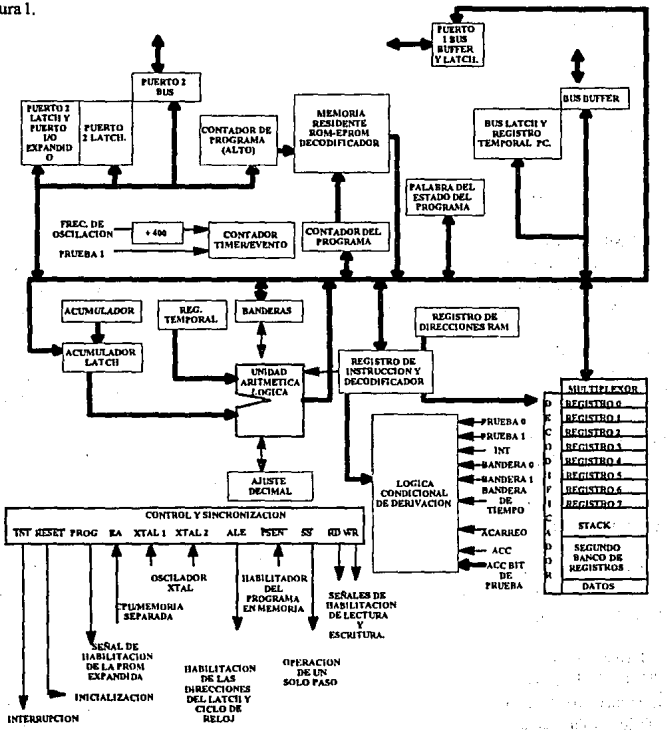


FIGURA 1.

DIAGRAMA A BLOQUES DEL MICROPROCESADOR 8749.

La sección Aritmética contiene las funciones de manipulación de datos básicos, dividiéndolos en los siguientes bloques:

a) Unidad Lógica y Aritmética (ALU)
b) Acumulador
c) Bandera de Acarreo (Carry Flag)
d) Decodificador de instrucciones.

a) UNIDAD ARITMETICA LOGICA

Esta unidad acepta palabras de ocho bits desde una o dos fuentes y genera un resultado también de ocho bits. Esta unidad puede efectuar las siguientes funciones:

1) Sumar con o sin acarreo
2) Funciones Lógicas AND, OR y OR exclusivo
3) Incrementar/Decrementar
4) Rotación izquierda o derecha
5) Cambiar "Nibbles" (4 bits)
6) Ajuste de BCD a Decimal

b) ACUMULADOR

El acumulador es un registro de datos muy importante en el procesador, ya que es una de las fuentes de entrada a la unidad aritmética (ALU) y el destino del resultado de las operaciones efectuadas en esa unidad. Los datos desde o hacia los puertos de I/O y la memoria pasan normalmente a través del acumulador.

c) BANDERA DE ACARREO

Si la operación efectuada por la ALU produce un valor representado por más de ocho bits (sobreflujo del bit más significativo) se enciende la bandera de acarreo en la Palabra de Estado del Programa (PSW).

d) DECODIFICADOR DE INSTRUCCIONES

La función del decodificador de instrucciones es almacenar la porción del código de operación de cada instrucción del programa. Es aquí donde se generan las salidas que controlan la función de cada uno de los bloques de la sección aritmética, estas líneas controlan la fuente de datos y el destino de los registros, así como también la función efectuada en la unidad aritmética (ALU).

La memoria del programa consiste de 2048 palabras de ocho bits, las cuales se direccionan mediante el contador del programa. Esta memoria puede programarse y borrarse por el usuario (memoria EPROM). La memoria del programa puede utilizarse para almacenar constantes así como también el programa de aplicación.

La memoria de datos (RAM) residente está organizada por palabras de ocho bits y tamaños de 128 localidades. Todas éstas localidades se direccionan indirectamente a través de dos registros Apuntadores, los cuales residen en las direcciones 0 y 1. Las primeras ocho localidaes (0-7) del arreglo se designan como registros de trabajo y se direccionan directamente mediante algunas interrupciones. Estos registros se utilizan para almacenar resultados intermedios.

♦ LOGICA DE SALTOS CONDICIONALES.

La Lógica de Salto Condicional se utiliza para habilitar varias condiciones internas y externas con la finalidad de poder ser verificadas por los usuarios del programa. Mediante la utilización de la instrucción de salto condicional las condiciones que se mencionan en la tabla 1, pueden efectuar un cambio en la secuencia de ejecución del programa.

♦ INTERRUPCIONES.

La interrupción se efectúa con un nivel bajo para permitir una "Función OR alambrada" de varias fuentes de interrupción conectadas a la terminal de entrada.

La línea de interrupción se muestrea cada ciclo de instrucción y cuando se detecta se origina una llamada a subrutina tan pronto como finalizan todos los ciclos de la instrucción actual, con instrucciones de dos ciclos la línea de interrupción se muestrea únicamente durante el segundo ciclo. La localidad 3 de la memoria de programa usualmente contiene un salto incondicional para dar el servicio de interrupción en forma de subrutina en cualquier sitio del programa. El final de un servcicio de interrupción por subrutina se señala por la ejecución de una instrucción de retorno o por un comando de restauración del estado RETR.

El sistema de interrupción es de un sólo nivel, en el cual cuando una interrupción se detecta, todos los requerimientos de interrupción posteriores serán ignorados hasta que la ejecución de un RETR rehabilite la lógica de interrupciones, esto ocurre al principio del segundo ciclo de la interrupción RETR.

La entrada de una interrupción externa puede habilitarse o deshabilitarse mediante el programa utilizando las instrucciones ENI y DIS I. Las interrupciones se deshabilitan durante el reinicio (reset) y permanecen así, hasta que sean habilitadas por el usuario del programa. Si se deshabilitan interrupciones permanentemente, INT puede utilizarse como otra entrada de prueba tal como T_0 y T_1.

♦ TEMPORIZADOR/CONTADOR.

El 8749 contiene un contador binario de 8 bits para ayudar al usuario en el conteo de eventos externos y también para generar tiempos de retardo exactos, sin interferir al procesador durante estas funciones, en ambos modos la operación del contador es la misma, la única diferencia es la fuente de entrada al contador.

El contador puede leerse y preseleccionarse con dos instrucciones MOV, las cuales transfieren el contenido del acumulador al contador y viceversa. El contenido del contador no se afecta por reinicio (reset) y puede inicializarse mediante programación. El contador puede ser parado por medio de la instrucción de reinicio (reset) o una instrucción STOP TCNT y permanecer en ese estado hasta que vuelva a ser inicializado como contador de tiempo (timer) mediante la instrucción START CNT, una vez que se ha inicializado el contador

8 y 9 de la memoria RAM, el apuntador de pila se incrementa en uno para apuntar a las localidaes 10 y 11 anticipándose así a otra llamada "call". La interacción de subrutinas pueden continuar hasta 8 veces sin sobrecargar la pila. El final de la subrutina, definido por una instrucción de retorno (RET o RETR), origina que el apuntador de pila se decremente y que el contenido del par de registros direccionados sea transferido al contador del programa.

♦ PALABRA DEL ESTADO DEL PROGRAMA (PSW).

La PSW es de 8 bits, la cual puede ser cargada a/o desde el acumulador. Esta palabra esta compuesta por una colección de FLIP-FLOPS, los cuales pueden ser leídos o escritos como un todo.

La habilidad con que se cuente para escribir la PSW permitirá restaurar fácilmente el estado de la máquina. La figura 4 muestra la Palabra de Estado del Programa (PSW).

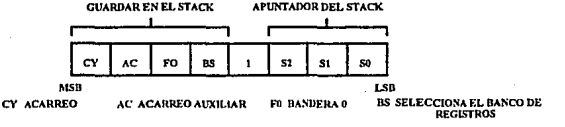


FIGURA4

Cuando existe una llamada a subrutina los cuatro bits superiores (los de más peso) de la PSW se salvan y se restauran opcionalmente mediante un retorno con la instrucción RETR.

Las definiciones para la PSW son las siguientes:

BITS 0-2	APUNTADOR DE PILA (S0,S1,S2)
BIT 3	NO SE UTILIZA (EN UNO CUANDO SE LEE)
BIT 4	BANCO DE REGISTROS DE TRABAJO 0=BANCO 0, 1=BANCO 1
BIT 5	BANDERA 0 (FLAG 0) CONTROLADA POR EL USUARIO, PUEDE COMPLEMENTARSE O BORRARSE Y, PUEDE SER PROBADA CON LA INSTRUCCION DE SALTO CONDICIONAL (JF0)
BIT 6	ACARREO AUXILIAR (AC), BIT DE ACARREO GENERADO POR UNA INSTRUCCION "ADD" Y UTILIZADA POR LA INSTRUCCION DE AJUSTE DECIMAL "DAA"
BIT 7	BANDERA DE ACARREO, INDICA QUE LA OPERACION PREVIA HA PROVOCADO UN SOBREFLUJO EN EL ACUMULADOR

TABLA1.

♦ CONTADOR DE PROGRAMA Y PILA (STACK).

El contador de programa es un contador independiente, mientras que el contador de pila (stack) se implementa utilizando pares de registro del arreglo de memoria de datos. Solo se utilizan 11 bits del contador de programa para direccionar las 2048 palabras de la memoria del programa del 8749, mientras que los bits más significativos pueden utilizarse para trabajar memorias de programas externos, como se ilustra en la figura 2. El contador de programa se inicializa en cero activando la línea de reinicio (reset)

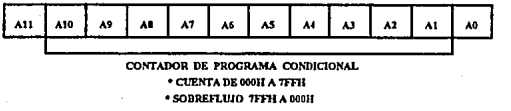


FIGURA 2.

Una interrupción o una llamada a subrutina (CALL) origina que el contador de programa se almacene en uno de los 8 pares del "stack", como se ilustra en la figura 3.

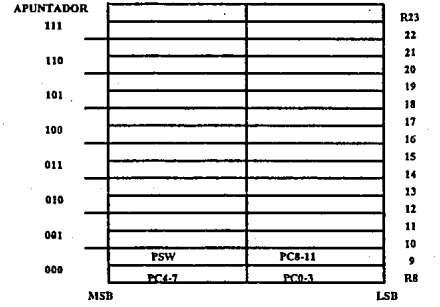


FIGURA 3.

El par a utilizar se determina con un apuntador de pila, de 3 bits, el cual es parte de la Palabra de Estado del Programa (PSW)

El área del stack está definida en las localidades de RAM 8-23, y se utilizan para almacenar el contador de programa y los 4 bits del PSW. Cuando se inicializa el apuntador de pila en 000, éste apuntará a las localidades

♦ LINEAS DE ENTRADA Y SALIDA DEL 8749.

Este microprocesador cuenta con 27 líneas que pueden utilizarse para funciones de entrada y salida o también como puertos bidireccionales, de las cuales tres pueden utilizarse como entradas de "prueba" las cuales pueden servir para alterar las secuencias del programa a través de instrucciones de salto condicional.

El 8749 cuenta con dos puertos (identificados como puerto 1 y puerto 2), los cuales tienen una longitud de 8 bits, y sus características que son idénticas se describen a continuación:

a) Son de tipo Latch, por lo que los datos permanecen invariables hasta que sean reescritos.
b) Como puertos de entrada estas líneas no son tipo Latch, es decir, las entradas deberan estar presentes hasta que sean leidas por una instrucción de entrada.
c) Las entradas son totalmente TTL compatibles y sus salidas pueden manejar una carga estándar.

Las líneas de los puertos 1 y 2 se denominan cuasi-bidireccionales dado que el circuito de salida se encuentra estructurado especialmente para permitir que cada línea sirva como entrada y salida o ambas cosas a través de salidas que estén estáticamente fijas (Latched).

Para proporcionar tiempos de conmutación rápidos de una transición de 0 a 1, se conmuta momentáneamente un dispositivo de relativa baja impedancia. Cuando se escribe un "0" en la linea, el dispositivo de baja impedancia proporciona una elevación leve permitiendo que la capacidad de corriente en el dispositivo TTL se reduzca, puesto que el transistor es un dispositivo de baja impedancia deberá aplicarse un "1" a cualquier línea que vaya a ser utlizada como entrada. El reinicio (reset) inicializa todas las líneas a una impedancia alta (estado "1").

El Bus es un puerto de 8 bits, el cual es realmente bidireccional, con entradas y salidas asociadas a señales de habilitación (strobes), este Bus puede servir como un puerto de salida estáticamente controlado (Latch), o como un puerto de entrada sin cerrojo (non-Latching). Como puerto estático los datos son escritos y mantenidos fijos (Latched) utilizando la instrucción OUTL o leidos utilizando la instrucción INS, estas instrucciones generan pulsos en las líneas de "strobe" de salidas correspondiente a RD y WR, por lo tanto, en el modo de puerto estático, éstas generalmente no se utilizan. Como un puerto bidireccional las instrucciones MOVX se utilizan para leer y escribir en el puerto, una escritura en el puerto genera un pulso en las líneas de salida WR y el dato de salida es válido durante el flanco de caída del pulso de WR. Una lectura del puerto genera un pulso en la línea de salida RD y el dato de entrada podrá ser leído durante la caída del pulso de RD. Cabe hacer notar que cuando no se escribe ni se lee, las líneas del Bus están en un estado de alta impedancia.

♦ ENTRADAS DE PRUEBA e INT.

Las terminales de contacto T_0 y T_1 e INT sirven como entrada y pueden evaluarse (testable) con instrucciones de salto condicional.

éste se incrementará hasta su máximo conteo (FF) y se sobrecargará hasta cero, continuando nuevamente su conteo hasta que sea parado mediante la instrucción STOP CNT o la de reinicio (reset).

♦ CIRCUITOS DE RELOJ Y TEMPORIZADOR.

La generación de los inetrvalos de tiempo para el 8749 está contenida en el CI, con la excepción de la frecuencia de referencia, la cual puede ser un cristal (XTAL), una fuente externa que utilice un reloj o un resonador cerámico.

♦ REINICIO (RESET).

Esta entrada es un disparador Schmitt-trigger, el cual tiene un dispositivo de acoplamiento (pull_up), que consiste de un capacitor externo con valor de 1 F, el cual proporciona un pulso interno de reinicio de suficiente longitud, para garantizar que todo el circuito sea reiniciado. Si el pulso de reinicio se genera externamente, el RESET deberá mantenerse con un valor bajo al menos 10 milisegundos después de que la fuente de alimentación esté estabilizada. Se requieren únicamente 5 ciclos de máquina si el circuito está alimentado y el oscilador se ha estabilizado. Las terminales ALE y PSEN estarán activadas durante el reinicio siempre que el habilitador EA sea igual a 1.

El reinicio (reset) en la máquina efectúa las siguientes funciones:

1) Sitúa al contador de programa en 0
2) Sitúa el apuntador de pila (stack) en cero
3) Selecciona el banco de registro 0
4) Selecciona el banco de memoria 0
5) Sitúa el bus en estado de alta impedancia (excepto cuando EA=5V)
6) Sitúa los puertos 1 y 2 en un modo que permite la entrada de datos
7) Deshabilita las interrupciones (las de tiempo y las externas)
8) Detiene el temporizador
9) Borra la bandera del temporizador
10) Borra las banderas F_0 y F_1
11) Deshabilita la salida del reloj en T_0.

♦ OPERACION DE UN SOLO PASO.

Esta operación le permite al usuario revisar programas, ya que le permite controlar el procesador para que ejecute una instrucción únicamente. Cuando se detiene, la dirección de la siguiente instrucción por ejecutar aparece en el Bus y en la mitad inferior del puerto 2, por lo que el usuario puede seguir el programa a través de cada una de las instrucciones. El contenido de las líneas del Bus se pierden durante este proceso, por lo tanto, se debe conectar un "latch" para reestablecer la capacidad de entrada/salida (I/O).

♦ PUESTA A TIEMPO

El 8040 opera en el modo denominado "de un solo paso", a continuación se describe este modo:

1) Al procesador se le detiene aplicando una señal de bajo nivel sobre SS.
2) El procesador responde deteniéndose durante el tiempo de búsqueda de la dirección de la siguiente instrucción, si una instrucción de doble ciclo está en progreso cuando se recibe el comando de un solo paso, ambos ciclos se completan antes de detenerse.
3) El procesador reconoce que ha entrado la situación de paro al colcocar ALE en un nivel alto. En este estado la dirección de la siguiente instrucción por ejecutar se coloca en el Bus y en la parte media inferior del puerto 2.
4) Por lo que SS es un puerto en alto, para colocar al procesador fuera del modo inactivo, permitiéndole buscar la siguiente instrucción. La salida de este estado de inactividad lo indica el procesador colocando ALE en estado bajo.
5) Para detener el procesador en la siguiente instrucción SS deberá colocarse nuevamente en estado bajo, inmediatamente después que ALE va hacia un nivel bajo también. Si SS se deja en estado alto el procesador permanecera en modo inactivo (Run). Borra la bandera del temporizador

♦ CONJUNTO DE INSTRUCCIONES.

A continuación se presenta el conjunto de instrucciones con el cual opera el microcontrolador 8749. Este conjunto se subdivide en instruciones con el acumulador, de entrada/salida, con registros, con etiquetas, con subrutinas, con banderas, con carga de datos, con el contador/temporizador, de control y la de no operación, como puede observarse en las tablas 2, 3, 4, 5, 6, 7, 8, 9, 10 y 11.

ACUMULADOR			
Mnemónico	**Descripción**	**Bytes**	**Ciclos**
ADD A,R	Suma el registro a A	1	1
ADD A, @R	Suma la memoria de datos a A	1	1
ADD A,#dato	Suma inmediata a A	2	2
ADDC A,R	Suma el registro con acarreo	1	1
ADDC A,@R	Suma la memoria de datos con acarreo	1	1
ADDC A,#dato	Suma inmediata con acarreo	2	2
ANL A,R	Realiza la operación AND de R con A	1	1
ANL A,@R	Realiza una AND con la memoria de datos	1	1
ANL A,#dato	Realiza una AND inmediata con A	2	2
ORL A,R	Realiza una OR de R con A	1	1
ORL A,@R	OR de la memoria de datos con A	1	1
ORL A,#dato	OR inmediato con A	2	2
XRL A,R	OR Exclusivo de R con A	1	1
XRL A,@R	OR Exclusivo de la memoria de datos con A	1	1
XRL A,#dato	OR Exclusivo inmediato con A	2	2
INC A	Incrementa A	1	1
DEC A	Decrementa A	1	1
CLR A	Limpia A	1	1
CPL A	Complementa A	1	1
DA A	Ajuste Decimal a A	1	1
SWAP A	Intercambia nibbles de A	1	1
RL A	Rota A a la izquierda	1	1
RLC A	Rota A a la izquierda hacia el bit de acarreo	1	1
RR A	Rota A a la derecha	1	1
RRC A	Rota A a la derecha hacia el bit de acarreo	1	1

TABLA 2. CONJUNTO DE INSTRUCCIONES QUE SE UTILIZAN CON EL ACUMULADOR

ENTRADA/SALIDA			
Mnemónico	**Descripción**	**Bytes**	**Ciclos**
IN A,P	Entrada de un puerto a A	1	2
OUTL P,A	Salida de A hacia un puerto	1	2
ANL P,#dato	AND inmediata del dato con el puerto	2	2
ORL P,#dato	OR inmediato del dato con el puerto	2	2
INS A,BUS	Entrada del BUS a A	1	2
OUTL BUS,A	Salida de A hacia el BUS	1	2
ANL BUS,#dato	AND inmediata del dato con el BUS	2	2
ORL BUS,#dato	OR inmediato del dato con el BUS	2	2
MOVD A,P	Entrada del Puerto de Expansión a A	1	2
MOVD P,A	Salida de A hacia el Puerto de Expansión	1	2
ANLD P,A	AND de A con el Puerto de Expansión	1	2
ORLD P,A	OR de A con el Puerto de Expansión	1	2

TABLA 3. CONJUNTO DE INSTRUCCIONES DE ENTRADA/SALIDA

REGISTROS			
Mnemónico	**Descripción**	**Bytes**	**Ciclos**
INC R	Incrementa el registro	1	1
INC @R	Incrementa la memoria de datos	1	1
DEC R	Decrementa el registro	1	1

TABLA 4. CONJUNTO DE INSTRUCCIONES DE REGISTROS

ETIQUETAS			
Mnemónico	Descripción	Bytes	Ciclos
JMP addr	Salto incondicional	2	2
JMPP @A	Salto indirecto	1	2
DJNZ R,addr	Decrementa el registro y salta	2	2
JC addr	Salta si el Acarreo es igual a 1	2	2
JNC addr	Salta si el acarreo es igual a 0	2	2
JZ addr	Salta si A es cero	2	2
JNZ addr	Salta si A no es cero	2	2
JT0 addr	Salta si T0 es igual a 1	2	2
JNT0 addr	Salta si T0 es igual a 0	2	2
JT1 addr	Salta si T1 es igual a 1	2	2
JNT1addr	Salta si T1 es igual a 0	2	2
JF0 addr	Salta si F0 es igual a 1	2	2
JF1 addr	Salta si F1 es igual a 1	2	2
JTF addr	Salta si la bandera del temporizador es verdadera (1)	2	2
JNI addr	Salta si INT es igual a 0	2	2
JBb addr	Salta si el bit del acumulador es verdadero (1)	2	2

TABLA5. CONJUNTO DE INSTRUCCIONES DE ETIQUETAS.

SUBRUTINAS			
Mnemónico	Descripción	Bytes	Ciclos
CALL addr	Salta a una subrutina	2	2
RET	Regresar	1	2
RETR	Regresar y restaurar el estado	1	2

TABLA6. INSTRUCCIONES DE SUBRUTINAS.

BANDERAS			
Mnemónico	Descripción	Byte	Ciclos
CLR C	Limpia la bandera de acarreo	1	1
CPL C	Complementa el acarreo	1	1
CLR F0	Limpia la bandera 0	1	1
CPL F0	Complementa la bandera 0	1	1
CLR F1	Limpia la bandera 1	1	1
CLP F1	Complementa la bandera 1	1	1

TABLA7. INSTRUCCIONES DE BANDERAS.

CARGA DE DATOS			
Mnemónico	**Descripción**	**Bytes**	**Ciclos**
MOV A,R	Carga el registro hacia A	1	1
MOV A,@R	Carga la memoria de datos hacia A	1	1
MOV A,#dato	Carga inmediata hacia A	2	2
MOV R,A	Carga del acumulador hacia el registro	1	1
MOV @R,A	Carga de A hacia la memoria de datos	1	1
MOV R,#dato	Carga inmediata hacia el registro	2	2
MOV @R,#dato	Carga inmediata hacia la memoria de datos	2	2
MOV A,PSW	Carga PSW hacia A	1	1
MOV PSW,A	Carga A hacia PSW	1	1
XCH A,R	Intercambia A y el R	1	1
XCHD A,@R	Intercambia en nible de A y la memoria de datos	1	1
XCH A,@R	Intercambia A y la memoria de datos	1	1
MOVX A,@R	Carga la memoria de datos externa hacia A	1	2
MOVX @R,A	Carga A hacia la memoria de datos externa	1	2
MOVP A,@A	Carga de A a la página actual	1	2
MOVP3 A,@A	Carga de A hacia la página 3	1	2

TABLA 8. INSTRUCCIONES DE CARGA DE DATOS

Mnemónico	Descripción	Bytes	Ciclos
NOP	No operación	1	1

TABLA 9. INSTRUCCION DE NO-OPERACION

TEMPORIZADOR/CONTADOR			
Mnemónico	Descripción	Bytes	Ciclos
MOV A,T	Lee Temporizador/Contador	1	1
MOV T,A	Carga Temporizador/Contador	1	1
STRT T	Inicia Temporizador	1	1
STRT CNT	Inicia Contador	1	1
STOP TCNTI	Detiene Temporizador/Contador	1	1
EN TCNTI	Habilita Temporizador/Contador	1	1
DIS TCNTI	Deshabilita Temporizador/Contador	1	1

TABLA 10. INSTRUCCIONES DEL TEMPORIZADOR/CONTADOR

CONTROL			
Mnemónico	Descripción	Bytes	Ciclos
EN I	Habilita interrupción externa	1	1
DIS I	Deshabilita interrupción externa	1	1
SEL RB0	Selecciona el banco de registros 0	1	1
SEL RB1	Selecciona el banco de registros 1	1	1
SEL MB0	Selecciona el banco de memoria 0	1	1
SEL MB1	Selecciona el banco de memoria 1	1	1
ENT0 CLK	Habilita la salida de reloj en T0	1	1

TABLA 11. INSTRUCCIONES DE CONTROL

PIN No.	SEÑAL	FUNCION
20	VSS	GND
26	VDD	+5 volts en operación normal
40	VCC	Alimentación principal, +5 volts durante operación y programación
25	PROG	Señal de habilitación de la PROM expandida
27-34	P10-P17 PUERTO 1	Puerto cuasi-bidireccional de 8 bits
21-24 35-38	P20-P23 P24-P27 PUERTO 2	Puerto cuasi-bidireccional de 8 bits
12-19	DB0-DB7 BUS	Puerto bidireccional que puede ser escrito o leído en sincronía con las señales (RD)' y (WR)'
1	T0	Entrada que puede ser examinada utilizando la instrucción de transferencia condicional JT0 y JNT0
39	T1	Entrada que puede ser examinada utilizando las instrucciones JT1 y JNT1
6	(INT)'	Entrada de interrupción
8	(RD)'	Salida que se activa durante la lectura del BUS
4	(RESET)'	Entrada que es utilizada para iniciar el procesador
10	(WR)'	Salida habilitada durante una escritura al BUS
11	ALE	Habilitación de dirección que ocurre durante cada ciclo
9	(PSEN)'	Habilitador de almacenamiento de programa que ocurre durante el acceso a una memoria externa
5	(SS)'	Entrada de operación de un solo paso que es utilizada con la señal ALE
7	EA	Entrada de acceso externo la cual forza que la memoria de programa haga referencia a una memoria externa
2	XTAL1	Entrada del cristal oscilador
3	XTAL2	Entrada del cristal oscilador

TABLA 12. CONFIGURACION DE PINES DEL 8749H

MODULO LCD COMPACTO AND 721

El diagrama a bloques de este módulo se muestra en la figura 5.

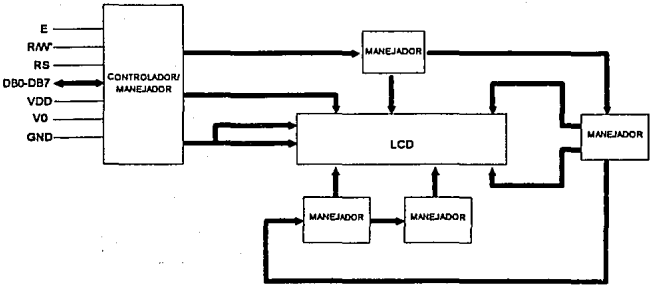


FIGURA 5. DIAGRAMA A BLOQUES DEL LCD AND 721

CARACTERISTICAS.

- Módulo de despliegue integrado
- Alta resolución
- Voltaje bajo, fuente de 5 volts
- Rango de temperatura de operación de 0 - 50 grados centígrados
- Formato de caracter de 5X7 puntos y línea de cursor
- Interfase directa con CPU's de 8 ó 4 bits
- Once comandos para control

La asignación de pines se muestra en la tabla 13.

PIN No.	SEÑAL	FUNCION
1	GND	0 volts
2	VDD	5 volts
3	V0	de 0 volts a VDD
4	RS	Entrada de datos en alto, entrada de comando en bajo
5	R/W	Lectura de datos en alto, escritura en bajo
6	E	Señal de habilitación
7	DB0	Primer bit del bus de datos
8	DB1	Segundo bit del bus de datos
9	DB2	Tercer bit del bus de datos
10	DB3	Cuarto bit del bus de datos
11	DB4	Quinto bit del bus de datos
12	DB5	Sexto bit del bus de datos
13	DB6	Séptimo bit del bus de datos
14	DB7	Octavo bit del bus de datos

TABLA 13. ASIGNACION DE PINES DEL LCD

COMANDO	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DESCRIPCION
Limpia pantalla	0	0	0	0	0	0	0	0	0	1	Limpia toda la pantalla y regresa el cursor a la posición inicial (dirección 0)
Regresa al inicio	0	0	0	0	0	0	0	0	1	X	Regresa el cursor a la posición inicial
Colocar modo entrada	0	0	0	0	0	0	0	1	I/D	S	Coloca la dirección de movimiento del cursor y especifica si el cursor se recorre o no cuando despliegue
Control de encendido -apagado del despliegue	0	0	0	0	0	0	1	D	C	B	Coloca en encendido-apagado toda la pantalla, el cursor y el parpadeo del cursor
Corrimiento del cursor y del despliegue	0	0	0	0	0	1	S/C	R/L	X	X	Mueve el cursor y recorre el despliegue sin cambiar los contenidos del la DDRAM
Coloca Función	0	0	0	0	1	DL	N	F	X	X	Coloca la longitud de los datos de las líneas de despliegue y el tipo de caracter
Coloca la dirección RAM	0	0	0	1	A C G	A C G	A C G	A C G	A C G	A C G	Coloca la dirección en la CGRAM. Los datos son enviados y recibidos después de esta operación
Coloca la dirección DDRAM	0	0	1	A D D	A D D	A D D	A D D	A D D	A D D	A D D	Coloca la dirección de la DDRAM
Lee la bandera BUSY y la dirección	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Lee la bandera BUSY la cual indica que el módulo esta en operación interna y lee los contenidos del contador de direcciones
Escribe datos a la CG o DDRAM	1	0	WD	WD	WD	WD	WD	WD	WD	WD	Escribe datos desde la DDRAM o CGRAM
Lee datos de la CC o DDRAM	1	1	RD	RD	RD	RD	RD	RD	RD	RD	Lee datos desde la DDRAM o CGRAM

TABLA 14. LISTA DE COMANDOS DEL MODULO LCD

MEMORIA EPROM 2716

En la figura 6 se muestra el diagrama a bloques de la memoria EPROM 2716.

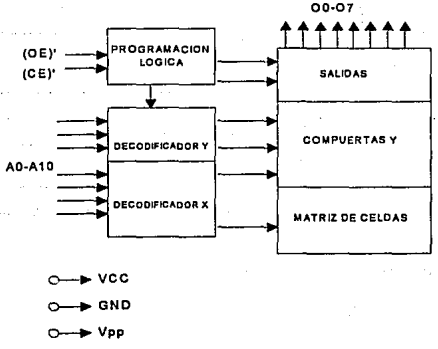


FIGURA 6. DIAGRAMA A BLOQUES DE LA MEMORIA EPROM 2716

CARACTERISTICAS.

- Acceso rápido
- Alimentación de +5 volts
- Baja disipación de potencia
- Configuración de pines compatible con las EPROM de Intel Universales
- Requerimientos de programación simples
- Entradas y salidas compatibles con TTL durante lectura y programación
- Completamente estática

La configuración de pines se muestra en la tabla 15.

PIN No.	SEÑAL	FUNCION
12	GND	0 Volts
24	VCC	Alimentación +5 volts en operación normal
21	VPP	Voltaje de programación
1-8 22-23 19	A0-A10	Direcciones
18	(CE)'	Habilitador del circuito
20	(OE)'	Habilitador de salidas
9-11 13-17	O0-O7	Salidas

TABLA 15. CONFIGURACION DE PINES DE LA MEMORIA EPROM 2716

BUFFER TRES ESTADOS 74HC126A

Es un componente CMOS que contiene cuatro elementos individuales como se muestra en el diagrama a bloques de la figura 7.

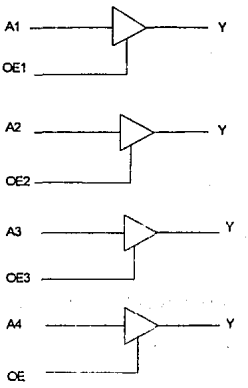


FIGURA 7. DIAGRAMA A BLOQUES DEL BUFFER 74HC126A

CARACTERISTICAS.

- Capacidad de salida de 15 cargas LSTTL
- Interfase directa con circuitos CMOS, NMOS y TTL
- Rango de voltaje de operación de 2 - 6 volts
- Corriente de entrada baja (1 microAmpere)
- Alta inmunidad al ruido
- Retardos de propagación mejorados

En la tabla 16 se muestra la configuración de pines para este circuito.

PIN No.	SEÑAL	FUNCION
7	GND	0 Volts
14	VCC	2-6 Volts
2,5,9,12	A1-A4	Entradas
1,4,10,13	OE1-OE4	Habilitadores
3,6,8,11	Y1-Y4	Salidas

TABLA 16. CONFIGURACION DE PINES DEL BUFFER 74HC126A

FUENTE DE PODER DE +5 VOLTS

La fuente de alimentación utilizada en este diseño es de +5 volts fija, voltaje requerido para alimentar todo el circuito. El diagrama esquemático de la fuente se presenta en la figura 8.

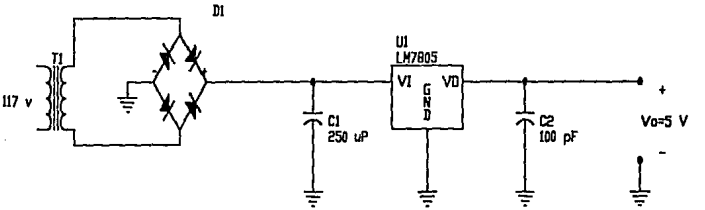


FIGURA 8. DIAGRAMA ESQUEMATICO DE LA FUENTE DE +5 V

TECLADO

El teclado del SME se diseñó con botones de presión (push botton), dado que la configuración es sencilla. Existen dos tipos de botones de presión:

1) Normalmente-abierto, el cual al momento de ser presionado cierra el circuito
2) Normalmente-cerrado, el cual al ser presionado abre el circuito

El teclado que se muestra en la figura 9, está compuesto de estos dos tipos de botones.

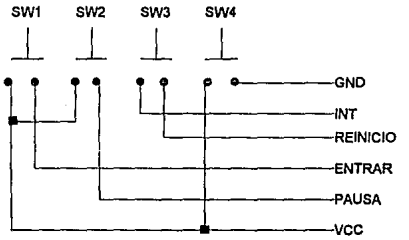


FIGURA 9. DIAGRAMA DEL TECLADO PARA EL SME

APENDICE C

MANUAL DE USUARIO DEL SME

INTRODUCCION

Este manual es una guía que le servirá al usuario para poder poner en funcionamiento el SME. Aquí se presentan los pasos básicos a seguir para su óptimo aprovechamiento.

Esta guía se encuentra estructurada en dos secciones principales. La primera de ellas contiene una descripción general del SME así como los requerimientos para su funcionamiento.

La segunda sección contiene una serie de pasos que describen las diferentes funciones y posibilidades de trabajo con el SME.

SECCION I
DESCRIPCION GENERAL DEL SME

El SME está constituido por los siguientes componentes:

1) Microcontrolador INTEL 8749H
2) Bus tres estados NATIONAL 74HC126N
3) Teclado
4) Pantalla de cristal líquido AND 721
5)Fuente de alimentación

🕮 NOTA: La descripción detallada de los componentes electrónicos se presenta en el apéndice B.

En la figura 1 se muestra el diagrama esquemático del SME.

REQUERIMIENTOS

1) Fuente de alimentación de 117 volts
2) Memoria EPROM 2716, proporcionada por el usuario

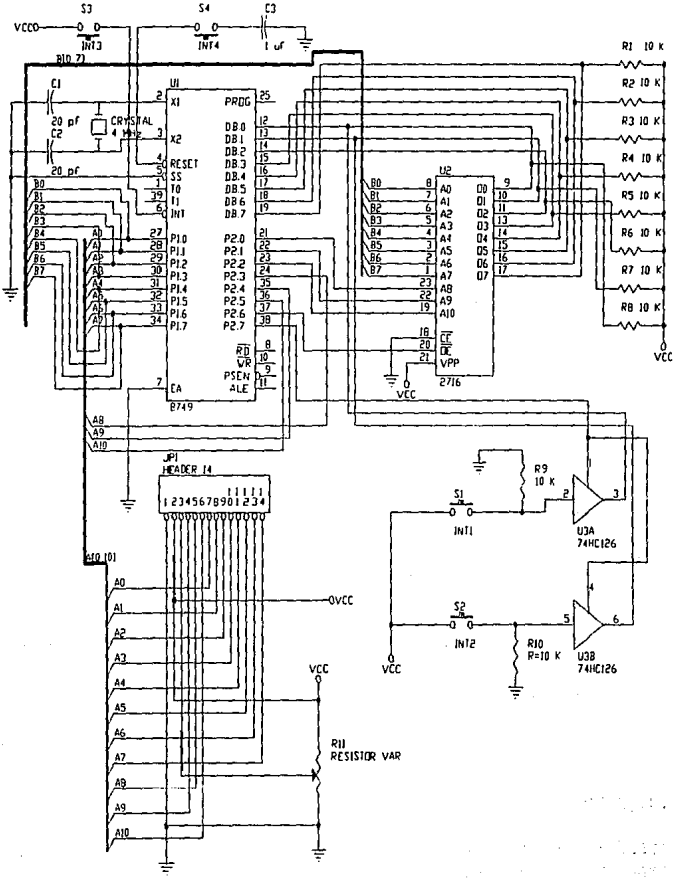
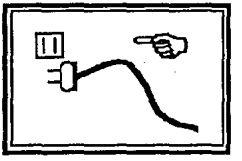


FIGURA 1. DIAGRAMA ESQUEMATICO DEL SME

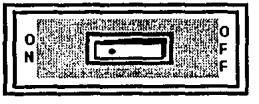
SECCION II
FUNCIONES DEL SME

Para poder verificar el contenido de cualquier memoria EPROM 2716, es necesario que lleve a cabo los siguientes pasos:

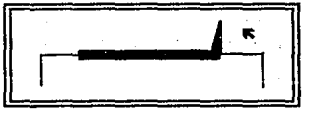
1) Conecte el SME a la fuente de alimentación



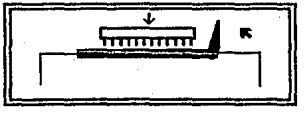
2) Accione el interruptor de encendido



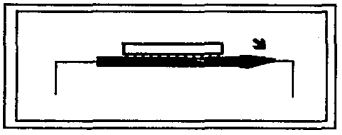
3) Asegúrese que la palanca de la base se encuentre en posición vertical, de tal manera que la memoria puede ser introducida en la base.



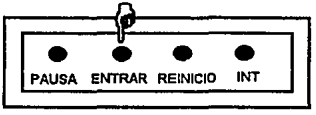
4) Coloque la memoria EPROM 2716 en la base



5) Coloque la palanca de la base en posición horizontal, asegurando que la memoria quede fija.



6) Oprima la tecla de ENTRAR



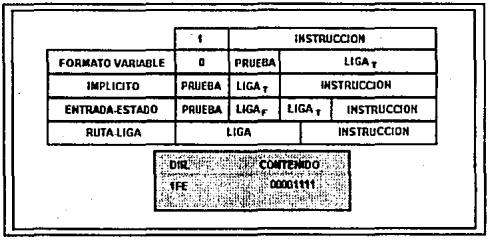
7) A partir de este momento la información desplegada en la pantalla será la dirección y el contenido correspondiente a cada una de las localidades de la memoria . El contenido de estas localidades será interpretado de acuerdo al tipo de direccionamiento que se haya utilizado en el diseño de la carta ASM, como a continuación se enlista:

a) Si se utilizó Direccionamiento por trayectoria la interpretación de los 8 bits será la correspondiente a los letreros de color azul, que se encuentran localizados en la parte superior de la pantalla como se muestra en la figura de abajo.

b) Si se utilizó Direccionamiento entrada-estado la interpretación de los 8 bits será la correspondiente a los letreros de color rojo, que se encuentran localizados en la parte superior de la pantalla como se muestra en la figura de abajo.

c) Si se utilizó Direccionamiento implícito la interpretación de los 8 bits será la correspondiente a los letreros de color verde, que se encuentran localizados en la parte superior de la pantalla como se muestra en la figura de abajo.

d) Si se utilizó Direccionamiento de formato variable la interpretación de los 8 bits será la correspondiente a los letreros de color negro, que se encuentran localizados en la parte superior de la pantalla como se muestra en la figura de abajo.



8) Durante el proceso de despliegue se cuenta con las siguientes opciones:

a) Con la tecla PAUSA usted podrá detener la última información desplegada por un período de tiempo más largo para su lectura.

b) Usted podrá finalizar la utilización del SME oprimiendo la tecla INT y luego la tecla ENTRAR, o bien, iniciar la lectura de otra memoria oprimiento la tecla de INT y luego la tecla REINICIO.

BIBLIOGRAFIA

BIBLIOGRAFIA.

1) **CAD/CAM Theory and Practice.**
IBRAHIM ZEID.
EDITORIAL MC. GRAW HILL.
1991, U.S.A.

2) **SISTEMAS CAD/CAM/CAE**
Diseño y Fabricación por Computador
JOSE MOMPIN POBLET
SERIE: MUNDO ELECTRONICO
EDITORIAL EDITORES BOIXAREU
1988

3) **CAD_CAM**
BARRY HAWKES
EDITORIAL PARANINFO
1989; MADRID, ESPAÑA

4) **DISEÑO DIGITAL.**
M. MORRIS MANO.
EDITORIAL PRENTICE HALL.
PRIMERA EDICION EN ESPAÑOL.
1987, MEXICO.

5) **DIGITAL SYSTEM DESIGN**
Using Programmable Logic Devices.
PARAG K. LALA
EDITORIAL PRENTICE HALL.
1990, U.S.A.

6) **DIGITAL SYSTEMS DESIGN**
with Programmable Logic.
MARTIN BOLTON
EDITORIAL ADDISON-WESLEY
PUBLISHING COMPANY.
1990, GRAN BRETAÑA.

7) **DIGITAL HARDWARE DESIGN**
JOHN B. PEATMAN.
EDITORIAL MC GRAW HILL.
1980, SINGAPORE.

8) **DESIGNING LOGIC SYSTEMS**
Using State Machines.
CHRISTOPHER R. CLARE
MC GRAW HILL BOOK COMPANY
1973.

9) **ARQUITECTURA DE COMPUTADORAS Y PROCESAMIENTO PARALELO.**
KAI HWANG/FAYE A. BRIGGS
EDITORIAL MC GRAW HILL

10) **TANGO ROUTE PLUS**
Manual del Usuario.
ACCEL TECHNOLOGIES INCORPORATION.
MAYO 16, 1990

11) **TANGO PCB PLUS**
Manual del Usuario
ACCEL TECHNOLOGIES INCORPORATION
MAYO 16,1990

12) **LS/S/TTL LOGIC DATABOOK**
NATIONAL SEMICONDUCTOR
1989, U.S.A.

13) **8-BIT EMBEDDED CONTROLLERS DATABOOK**
INTEL
1990, U.S.A.

14) **OrCAD DIGITAL SIMULATION TOOLS REFERENCE GUIDE**
L.P. AND GOGESCH MICRO SYSTEMS, INC.
1990, U.S.A.

15) **AMBIENTE CAD PARA DISEÑO DE SISTEMAS DIGITALES ORIENTADO A PAL's. (TESIS)**
SERGIO AMBRIZ MAGUEY
MARTIN PEREZ MONDRAGON
1990, MEXICO.

16) **UP600 USER'S MANUAL (VERSION 2.0)**

17) **HIGH SPEED CMOS LOGIC DATA**
TECHNICAL INFORMATION CENTER
MOTOROLA INCORPORATION
1989, U.S.A.

18) **MEMORY DATABOOK**
INTEL
SEPTEMBER, 1989
U.S.A.

19) **MANUAL DE MODULOS LCD**
AND
1990, USA.