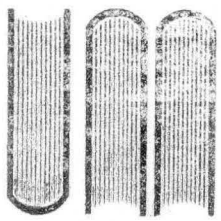
$ 500 =

12 FEB.

Aprobado: A. C. Fernández B.

UNIVERSIDAD DE MONTERREY

DIVISION DE CIENCIAS NATURALES Y EXACTAS



Titulo:

PROGRAMADOR DE MEMORIAS EPROM

REPORTE DEL PROGRAMA DE EVALUACION FINAL

QUE PRESENTA

Autor:
ALBERTO MOLLO MAMANI

EN OPCION AL TITULO DE

INGENIERO EN COMPUTACION ADMINISTRATIVA
Y DE PRODUCCION

MONTERREY, N. L. DICIEMBRE DE 1984

A DIOS TODO PODEROSO

A MIS PADRES:
VICENTE MOLLO H.
TORIBIA V. DE MOLLO
QUE CON SU GRAN SACRFICIO
ME DIERON ESTA QUE ES:
MI PROFESION.
CON MUCHO AMOR RESPETO
Y ADMIRACION

A MIS HERMANOS:
ING. FELIX
SONIA - EDGAR
SEBASTIAN-ROSSE MARY
SAUL
POR EL GRAN ANIMO QUE ME
BRINDARON

A MI ACESOR:
ING. ALFONZO FERNANDEZ P.
POR LA GRAN AYUDA QUE ME
BRINDO EN LA REALIZACION
DE ESTE TRABAJO
! GRACIAS !.

A MIS CATEDRATICOS

A MIS COMPANEROS
Y AMIGOS POR SU
AYUDA EN TODO ASPECTO

C O N T E N I D O

INTRODUCCION

Las Eprom´s (Erasable Programmable Read Only Memory) aparecieron en el mercado alrededor de los años setenta y han estado imponiéndose con insitada fuerza, ya que las nuevas generaciones de ingenieros, particularmente los que están pendientes de las innovaciones tecnológicas, han comprendido que los Eprom´s están destinados a revolucionar el mundo de la electrónica.

En efecto, los Eprom´s han abierto y aun están abriendo nuevos caminos y facilitando el diseño de sistemas complejos, ofreciendo una flexibilidad y posibilitando la introducción de nuevas opciones en los sistemas ya construidos.

Entre las muchas ventajas que proporcionan estos dispositivos es que los costos de memoria han venido reduciéndose, y la capacidad de alamacenamiento se ha incrementado senciblemente. Esto debido a que decenas de circuitos integrados son reemplazados por unos pocos CHIP'S. La reducción de cableado y minimización del circuito supone una mejor confiabilidad, menor consumo de potencia, por último una gran facilidad para la reparación y mantenimiento de los equipos.

Los Eprom's están siendo utilizados en numerosas y muy variadas aplicaciones. Son mas comúnmente usados para almacenar programas ensambladores, monitores, editores, control de microprocesadores o para almacenar tablas de consulta donde un largo número de entradas y salidas son necesarias (por ejemplo, conbertir un código binario a otro, definir funciones matemáticas o controlar la secuencia de circuitos).

El desarrollo del proyecto es presentado de la siguiente manera:

En el primer capítulo se expone el objetivo, los alcances y limitaciones que persigue el proyecto.

En el segundo capítulo se presenta una descripción de tipos y capacidades de memorias, muy particularmente de las memorias Eprom's.

En el capítulo tercero trata de todo el material necesario para llevar a cabo la programación de Eprom's.

El cuarto capítulo habla de los componentes generales del programador de Eprom's y su construcción.

El capítulo quinto cubre lo referente al software de aplicación. Descripción del sistema y la definición del manejo de la memoria de la Apple II.

Finalmente el sexto capitulo se presenta el procedimiento de operación.

CAPITULO I

FINES DEL SISTEMA

1.1 OBJETIVOS

Objetivo General:

El objetivo del presente proyecto es construir un programador de memorias Eprom's, aprovechando óptimamente los recursos computacionales disponibles para enfrentar con mayor capacidad a las opciones que plantea el futuro.

Objetivos Específicos:

1.- Despertar atravez del proyecto el interés de ingenieros sobre

la utilidad que representa un programador de memorias Eprom's, y como con ella puede aumentar

su productividad.

2.- Minimizar los costos involucrados en el sistema.

3.- Facilidad de uso del sistema para que el usuario pueda programar sin ninguna dificultad.

4.- Flexibilidad de programar en diferentes tipos y capacidades de

Eprom's.

5.- Seguridad en el almacenamiento y la recuperación de informacion de Eprom's.

6.- Adicionar este recurso computacional a la UDEM para aplicaciones futuras.

1.2 ALCANCES Y LIMITACIONES

- Este sistema puede programar Eprom's con datos cargando desde un disckette.

- El sistema tiene la capacidad de verificar un mismo programa que se

encuentre en memoria y en un Eprom´s, para detectar si hubo alguna falla en la programación.

- Nos da la facilidad de listar en pantalla el programa que se encuentra en el espacio de trabajo.

- Nos permite copiar el contenido de un Eprom´s a otro.

- Tiene la facilidad de checar que un Eprom´s este completamente borrado antes de ser programado.

- Tiene la versatilidad de usar cualquier ranura (1-7) en la programación o lectura de Eprom´s.

- Este programador Eprom´s puede programar cualquier tipo de Eprom´s con capacidad de 1k, 2k, 4k u 8k.

CAPITULO II

INTRODUCCION A MEMORIAS

La función de este tema es dar una definición de memoria, informar los diferentes tipos y capacidades de memorias que se utilizan en una microcomputadora. También describir los tipos de EPROM'S que se van a programar.

2.1 DEFINICION DE MEMORIA

Dispositivo en el cual se puede almacenar información. Las memorias se clasifican en memoria principal y memoria secundaria.

MEMORIA PRINCIPAL:

La memoria principal es la que recide dentro de la microcomputadora y la función mas importante de esta memoria es la de proporcionar al microprocesador la información que se va a procesar, asi como las instrucciones sobre como procesar estos datos.

La memoria principal esta formado por dos tipos de memoria ROM y RAM.

MEMORIA DE SOLAMENTE LECTURA (ROM):

Se utiliza para almacenar programas fijos que no pueden ser alterados como las rutinas del sistema.

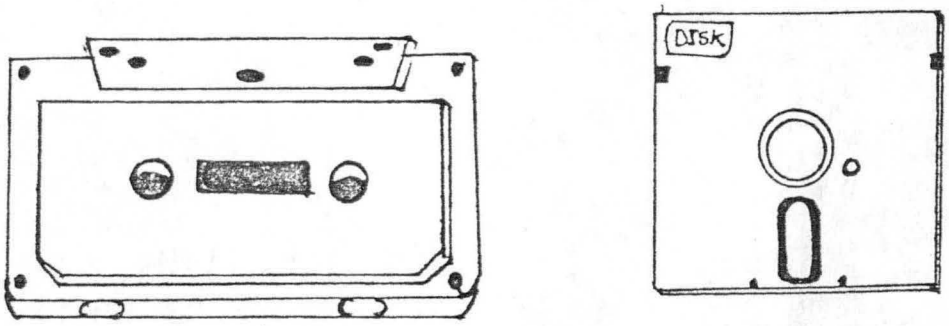
MEMORIAS DE ACCESO AL AZAR (RAM):

Se requiere para el almacenamiento de los programas del usuario no permanentes y para guardar datos.

MEMORIA SECUNDARIA:

El uso de los dispositivos de almacenamiento secundario permiten guardar permanentemente la información que se genera, además la posibilidad de actualizarla. Las mas usadas son los discos y las cintas.

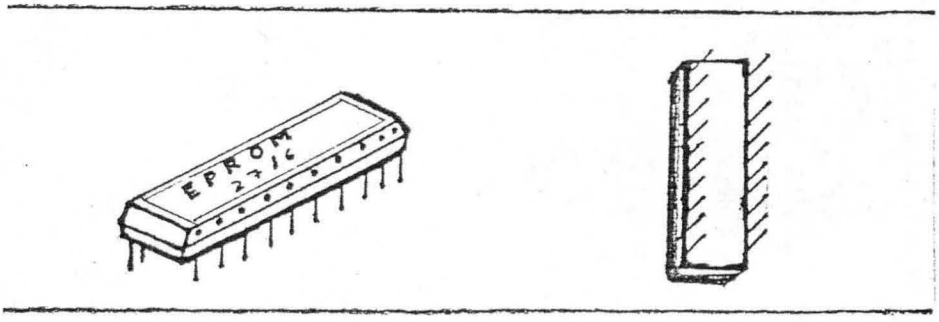
=== MEMORIA SECUNDARIA ===



2.2 DESCRIPCION DE EPROM´S

Las memorias EPROM´S son un tipo de memoria en al cual su contenido no puede ser cambiada facilmente, incluso cuando se apaga el ordenador; pero puede ser borrada mediante una aplicacion de luz ultravioleta.

=== E P R O M ´ S ===



- FIGURA 2.2 -

EPROM´S luego ROM.

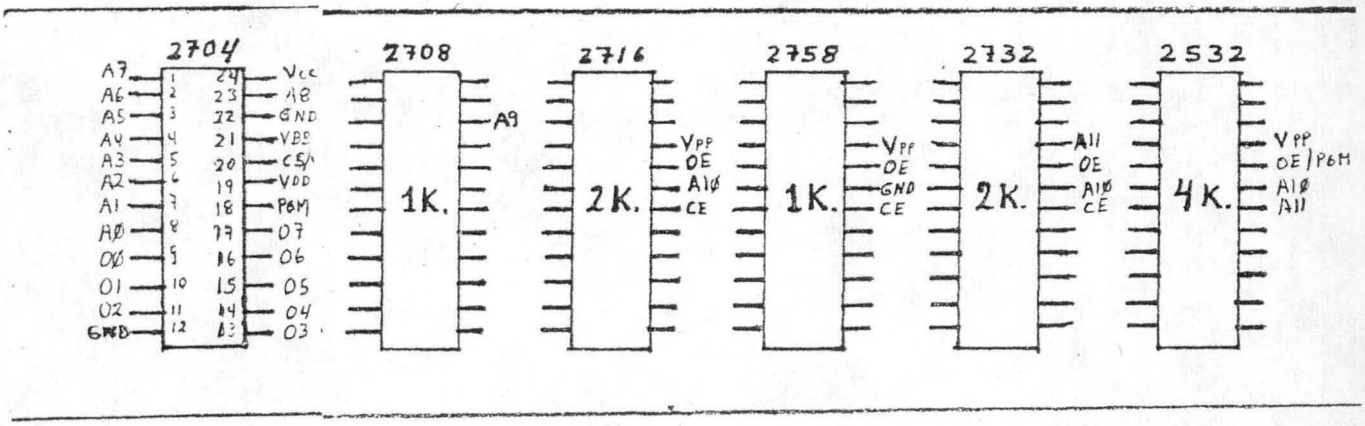
Las EPROM'S es el primer ejemplo que se construyo de memoria que sirvio y aun sirve para experimentar su potencia y rendimiento con el objeto de emprender su fabricacion en serie. Son memorias que se venden a escalas de bajo volumen. Mientras que los ROM'S son memorias fabricadas y vendidas a altas escalas tomando como modelo la EPROM. La cantidad que se pueden fabricar estas memorias es mayor o igual a 5000 ROM'S.

2.3 TIPOS Y CAPACIDADES DE EPROM'S.

Este programamdor de EPROM'S puede programar 2532, 2704, 2708, 2716, 2732, 2758, 2764 y otros tipos de EPROM'S, dando el uso a seleccion de 1K, 2K, 4K y 8K de capacidad.

Los 6 primeros EPROM'S mensionados anteriormente, son de 24 patas y el 2764 es de 28 patas.

==== E P R O M ' S D E 2 4 P A T A S ====



- FIGURA 2.3 -

Como se puede observar en este primer grupo de EPROM'S, la diferencia solamente existe en 4 patas (18, 19, 20 y 21). Esta diferencia es debido a que en algunos de ellos se incrementan las direcciones como en el caso del 2708 con el 2704. y en otros los voltajes necesarios de lectura o programación.

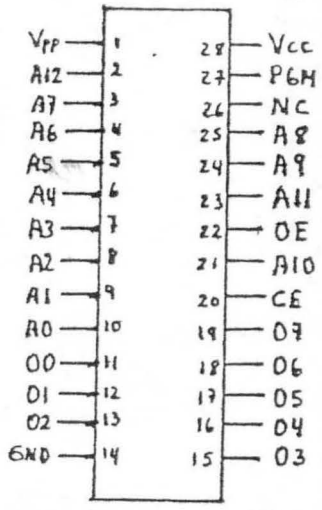
==== T A B L A D E V O L T A J E S ====

EPROM	LECTURA PATAS				PROGRAMACION			
	18	19	20	21	18	19	20	21
2704-2708	0	12	0	-5	25.5	12	12	-5
2758-2716	0	A10	0	5	5	A10	5	25.5
2732	0	A10	0	A11	5	A10	25.5	A11
2532	A11	A10	0	011	A11	A10	5	25.5

- TABLA 2.4 -

La tabla 2.4 muestra la designación de voltaje necesario a las patas en lectura o programación para los EPROM'S del primer grupo.

==== E P R O M ' S D E 2 8 P A T A S ====



- FIGURA 2.5 -

En este tipo de Eprom. la única diferencia del anterior grupo es que tiene una pata más para el direccionamiento (A12).

==== T A B L A D E V O L T A J E S ====

EPROM	LECTURA PATAS						PROGRAMACION					
	1	20	22	26	27	28	1	20	22	26	27	28
2764	5	.8	.8	0	2.0	5	.8	2.0	.8	0	2.0	5

Es la tabla de voltajes del Eprom 2764.

==== N O M B R E S D E P A T A S ====

P A T A S D E S C R I P C I O N

A0 - A13	Lineas de direccionamiento
O0 - O7	Linea de datos
CE - PGM	Chip habilitador de programa
OE	Habilitador
PGM	Programar
Vpp	Programador de voltaje
Vcc	Programador de voltaje
GRD	Conección a tierra
NC	No conectar

- TABLA 2.7 -

========= C A P I T U L O I I I

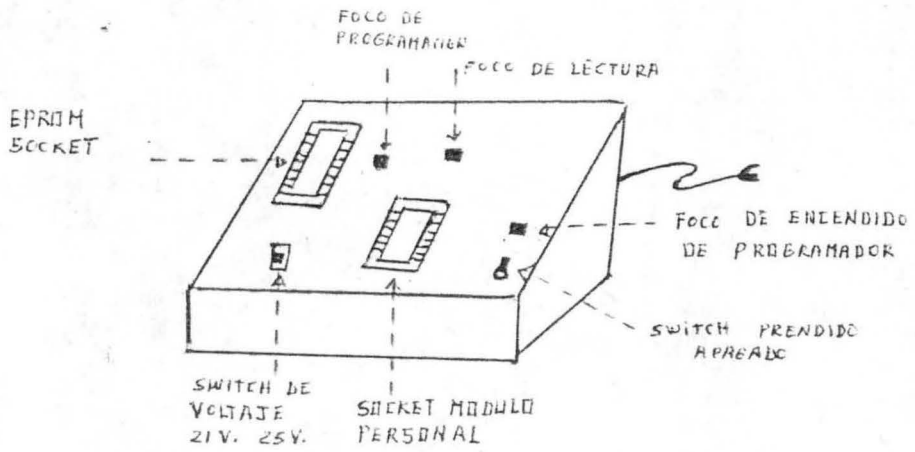
INSTRUMENTOS PARA LA PROGRAMACION DE EPROM´S

La función de este tema es informar los elementos con que cuenta el programador EPROM´S. Describe los componentes generales de hardware y software.

3.1 PROGRAMADOR DE EPROM´S

El programador de EPROM´S (fig. 3.1) es un conjunto de dispositivos electrónicos que realiza las funciones de recibir y transmitir señales de control, direcciones y datos para la programación o lectura de EPROM´S.

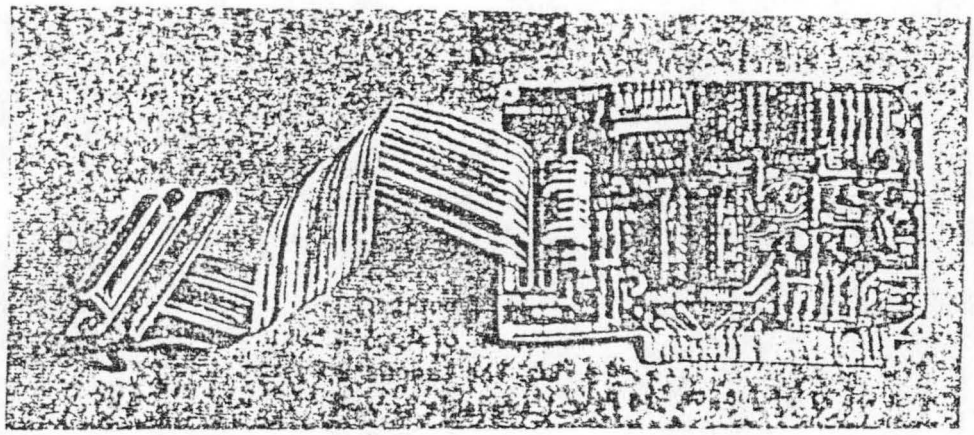
==== P R O G R A M A D O R D E E P R O M ' S ====



- FIGURA 3.1 -

El programador se comunica con el microprocesador por medio de una tarjeta periferico, que está conectado al conector de periferico (Ranura).

==== T A R J E T A D E I N T E R F A C E ====

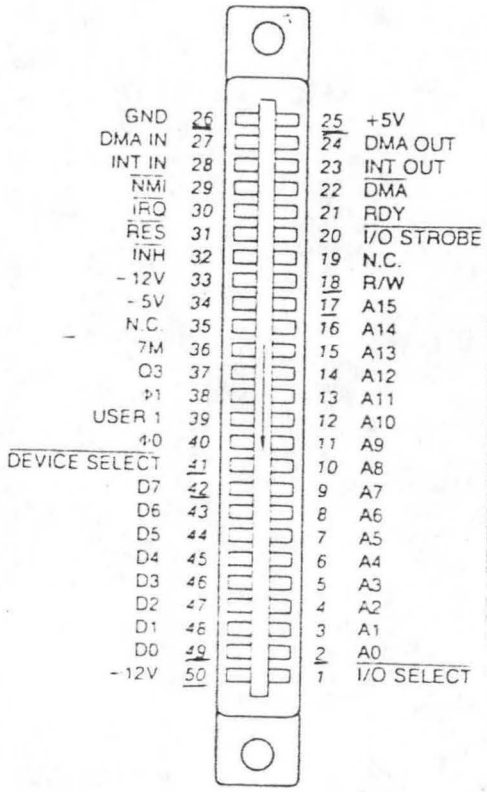


- FIGURA 3.2 -

Esta comunicación es atravez de las lineas de

direccionamiento (A0 - A15). por el bus de datos bidireccional de 8 bit's, una línea de lectura/escritura, 2 líneas para suplir el voltaje, 1 línea de device select y una línea de conección a tierra.

==== C O N E C T O R D E P E R I F E R I C O ====



- FIGURA 3.3 -

La tabla 3.4 da una descripción de señales del conector de periférico usados para la programación.

==== S E Ñ A L E S D E C O N T R O L ====

PATAS	NOMBRE	D E S C R I P C I O N
2 - 17	A0 - A15	Bus de direcciones

18	R/W	Buffer señal de R/W
25	+5V	Suple corriente 5V.
26	GND	Conec tor a tierra.
41	DEVICE SELECT	Esta línea activa sobre cada conec tor periférico cuando el bus de direcciones tiene una dirección entre $COn0 y $COnF.
42 - 49	D0 - D7	Bus bidireccional de datos.
50	+ 12V	Suple corriente de 12v

3.2 SOFTWARE

El manejador de software del programador de EPROM'S esta escrito en lenguaje Basic y lenguaje Ensamblador.

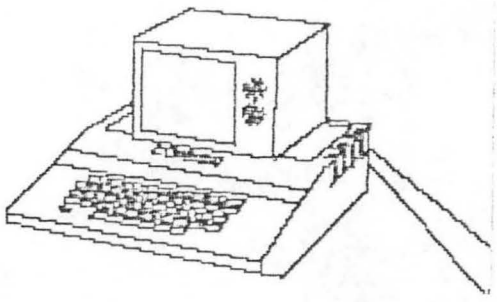
Como el sistema ejecuta una serie de funciones, el software contiene parte del lenguaje Basic para que sea interactivo, osea exista un dialogo entre hombre y maquina y que el usuario pueda entender con claridad que es lo que esta ocurriendo en cualquiera de las opciones del sistema. También contiene parte en lenguaje Ensamblador para aprovechar el área de memoria y optimizar tiempo en las operaciones.

3.3 OTROS INSTRUMENTOS DE HARDWARE Y SOFTWARE

Entre otros instrumentos de importancia para la programación, es la microcomputadora Apple II Plus; adicionados a ella todo el equipo periférico necesario. También se cuenta

con modulos personales para la programacion de diferentes tipos de EPROM'S.

==== A P P L E I I P L U S ====



- FIGURA 3.4 -

Este es la microcomputadora que se tomó para la programación de Eprom's. Consta de una unidad central de procesamiento, dos unidades de diskette, un monitor, un teclado y una impresora.

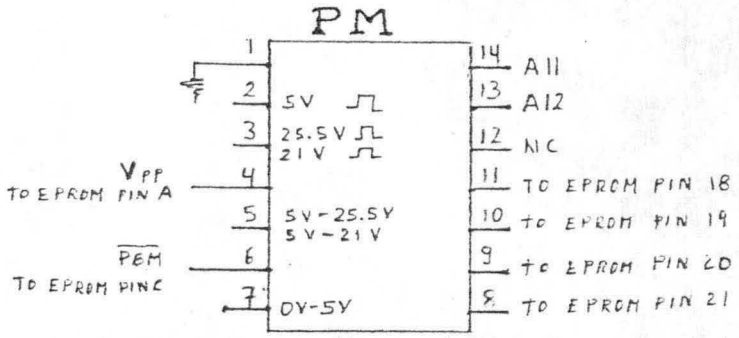
Modulos Personales:

Como hay una similitud en las patas, el circuito esta diseñado para acomodar los varios tipos de Eprom's en solo 4 lineas siendo única para cada Eprom's (18, 19, 20 y 21 lineas).

La conección de estas 4 líneas es hecho convenientemente

por rutas que afecten las señales atravez de módulos de alambrados personales específicos para cada tipo de Eprom.

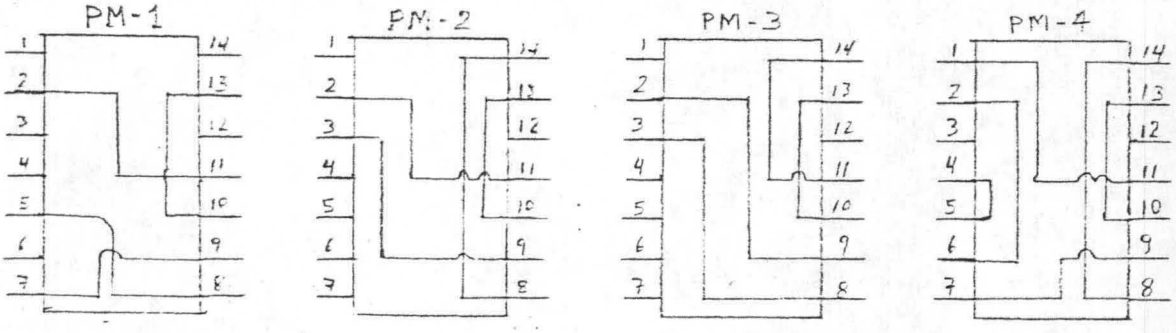
=== MODULO PERSONAL BASICO ===



- FIGURA 3.5 -

Los modulos personales constan de 14 patas de los cuales el 8, 9, 10 y 11 son los que se conectan al socket de Eprom's y mediante estasa las patas 21, 20, 19 y 18 respectivamente.

=== MODULOS PERSONALES A USAR ===



- FIGURA 3.6 -

La figura 3.6 muestra todos los módulos personales que se usarán.

=== RELACION MODULO PERSONAL Y TIPO DE EPROM'S ===

PM-1	2758-A, 2758-B 1K	2716, 2716-1 2K	2716, 2716-C 2K
PM-2	2732, 2732-C 4K	2732-A 4K	
PM-3	2532 4K		
PM-4	2764-6, 2764-3 8K	2764 8K	

- TABLA 3.7 -

La tabla 3.7 muestra el módulo personal que se tiene que usar para un determinado tipo de Eprom. Por ejemplo se usará el modulo 1 cuando se quiera programar o leer el 2758 de 1K, 2716-1 de 2K o el 2716c de 2K.

Además ésta tabla nos enseña el voltaje necesario para la programación para los diferentes tipos de Eprom's. Los que se encuentrán encerrados con una eleipse requierén de un voltaje de 21V. y los restantes de 25V. Para asignar el voltaje se activa un switch de voltaje a 21V. o 25V. en el programador de Eprom's.

En lo que se refiere a otros instrumentos de software, en algunos casos el basic no es lo bastante potente para realizar todas las funciones que necesita el sistema. Esta, desde luego,

es la razón por la cual se hace uso del sistema operativo de disco (DOS) del Apple II y funciones del monitor.

========== C A P I T U L O I V

ARQUITECTURA DEL PROGRAMADOR

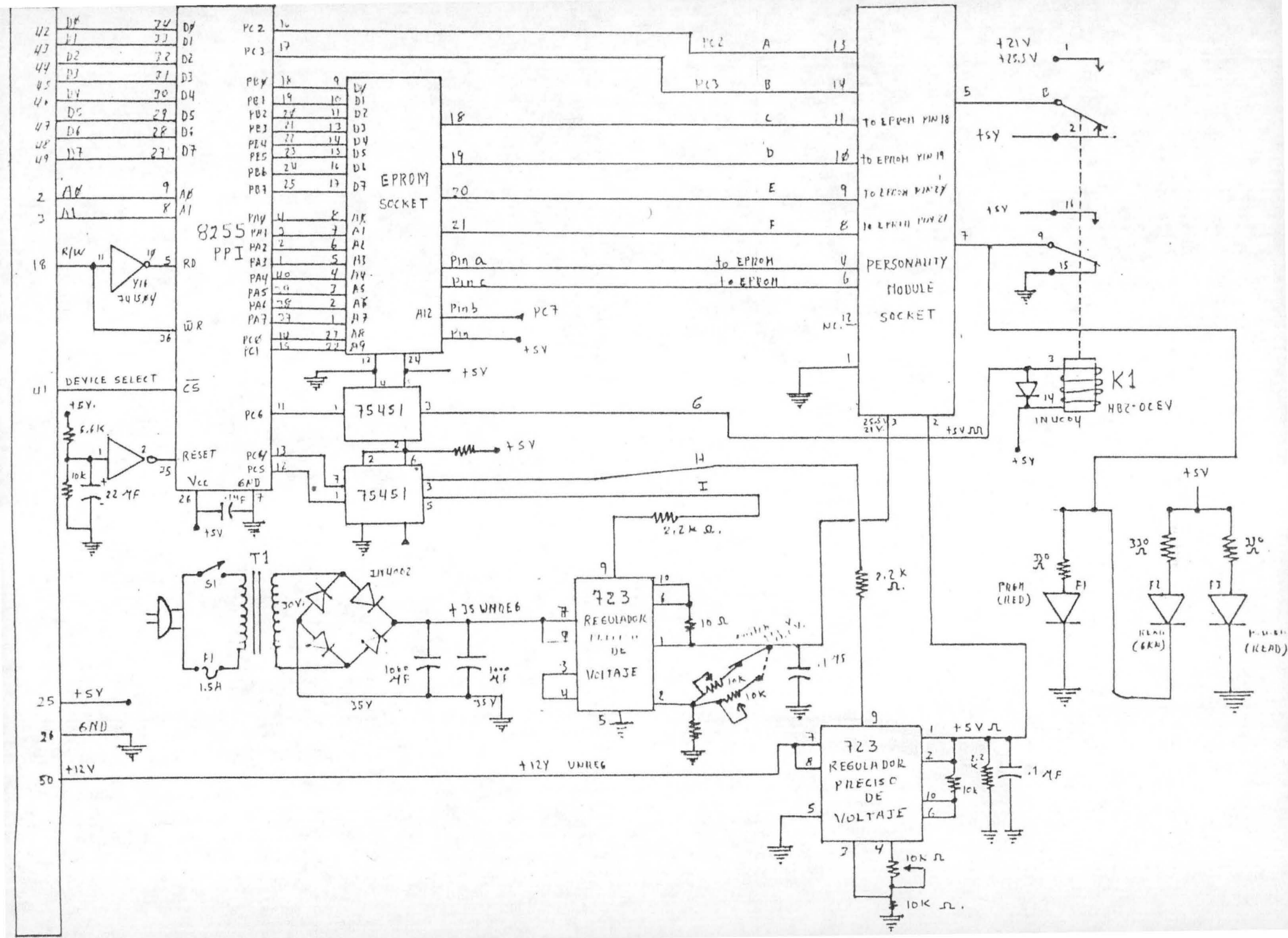
EPROM'S

En este tema se dá una visión general del esquema del programador de Eprom's, e informa cada uno de los elementos con que cuenta.

4.1 DESCRIPCION Y COMPONENTES BASICOS:

Un programador de Eprom's puede ser visto como un conjunto de 4 bloques:

- Interface Periferico programable 8255 (PPI).
- Socket para Eprom (ES).
- Socket para Modulo personal (PMS).
- Fuente de alimentación.

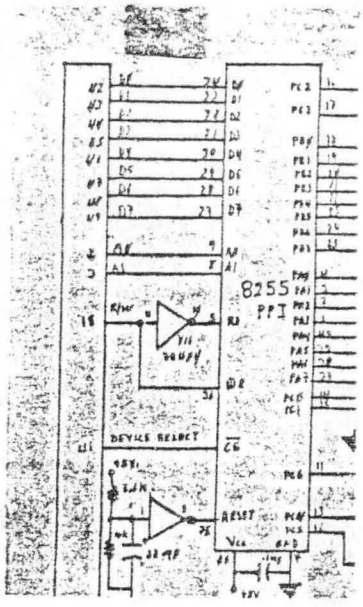

8255 PPI
EPROM SOCKET
PERSONALITY MODULE SOCKET
DEVICE SELECT
RESET
RD
WR
CS
Vcc
GND
75451
75451
723 REGULADOR PRECISO DE VOLTAJE
723 REGULADOR PRECISO DE VOLTAJE
T1
S1
1.5A
30V.
35V
+5V
+12V
GND
+12V UNREG
+5V UNREG
+25.5V
+21V
K1
HB2-DCEV
PROM (RED)
READ (GRN)
POWER (READ)
2.2K Ω.
10 Ω
10K
1K
.1 MF
1000 MF
5.6K
10K
22 MF
.1 MF
PC2
PC3
PC7
Pin a
Pin b
Pin c
Pin n
to EPROM
to EPROM
To EPROM PIN 18
To EPROM PIN 19
To EPROM PIN 20
To EPROM PIN 21

4.2 INTERFACE PERIFERICO PROGRAMABLE 8255 (PPI).

Este periferico es el nucleo para la programacion de Eprom's. La tarea del PPI es recibir y enviar senales de control, direcciones y datos del microcomputador al socket de Eprom's y socket de modulos personales o viceversa.

Este chip se comunica con el CPU atravez de 15 lineas; 8 lineas de bus de datos, 2 de direcciones, 1 de R/W, 1 linea de DEVICE SELECT, 2 lineas para suplir voltaje de 5 y 12V. y 1 linea de coneccion a tierra.

=== E L P. P. I. ===



- FIGURA 4.2 -

En esta aplicacion las lineas de I/O del 8255 estan

organizados en dos puertos de 8 bit's (A y B) y un puerto especial (C) con 8 lineas que pueden ser SET o RESET independientemente.

La parte media superior del puerto C es usado en conjunción con el puerto A para suplir hasta 12 bit's de direcciones. La parte media baja del puerto C controla el relevador K1 y el habilitador de corriente para suplir el voltaje necesario durante la secuencia de programación.

El puerto B es usado para transferir datos entre el Eprom y el CPU. Un puerto adicional de control del 8255 recibe un byte de control la cual configura separar el puerto A, B o C cuando es requerido una entrada o salida. El 8255 queda en una configuración especifica hasta que un nuevo byte de control es enviado por la microcomputadora.

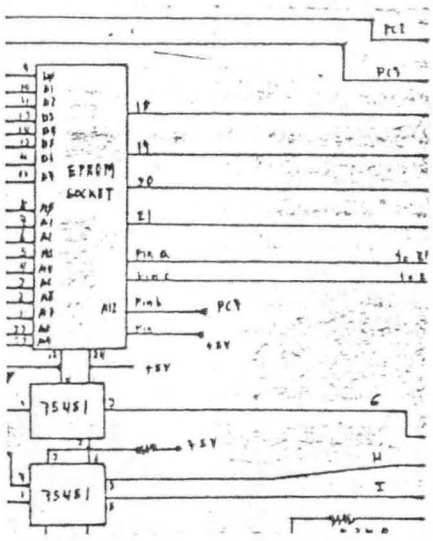
La línea de dirección A0 y A1 son usados para selecionar el puerto.

Cuando el voltaje es aplicado por primera vez, el 8255 automaticamente resetea por la línea 25 (+5V.), inicializando los puertos para ser un modo entrada.

4.3 SOCKET PARA EPROM'S

Este es el socket al cual se conecta el tipo de Eprom's que se desea leer o programar.

=== SOCKET PARA EPROM ===



- FIGURA 4.3 -

Este socket consta de 28 ranuras para la conexion de patas, osea existen Eprom's de 24 patas y Eprom's de 28 patas que son de mayor capacidad; por esta razón se tomó el socket de 28 ranuras.

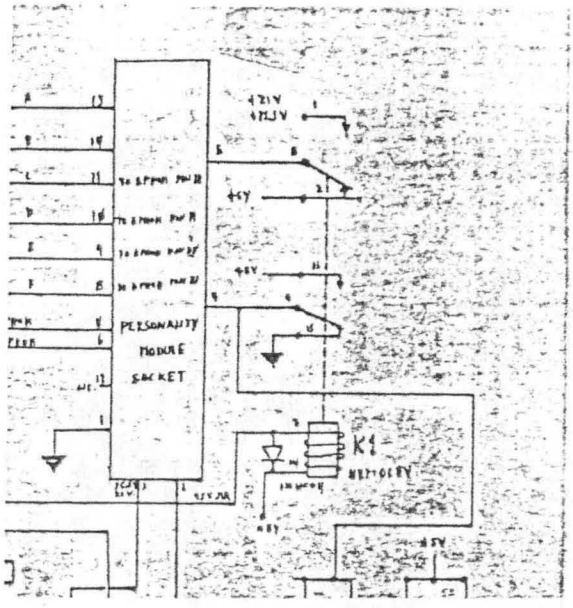
Las lineas bus de datos y de direcciones estan conectadas con el PPI, por la cuál se determina la dirección y la entrada y salida de los datos al Eprom. Las líneas que conectan del socket modulo personal unos son para incrementar direcciones y otros para la programación de voltajes.

4.4 SOCKET MODULO PERSONAL (PMS)

Este es el dispositivo al cuál se conectan los modulos personales.

Este socket consta de 14 ranuras. Las ranuras del 1-7 son para la programación de voltajes, las ranuras 13 y 14 son las que estan conectadas con el 8255 para incremento de direcciones y las ranuras del 8-11, 4 y 6 son las que se conectan con el socket de Eprom, por las cuales se envian direcciones y/o voltajes al socket de Eprom.

=== MODULO PERSONAL ===



- FIGURA 4.5 -

También se puede observar en la figura 4.5 que el abastecimiento de voltaje para algunas de las patas del Eprom depende en que modo se encuentre el 8255 (modo lectura o programacion). El relevador es el encargado de switchear a cualquiera de los dos modos por la linea de control (PC6).

El relevador tambien switchea los focos de control. El foco F1 permanece prendida cuando el 8255 esta en el modo

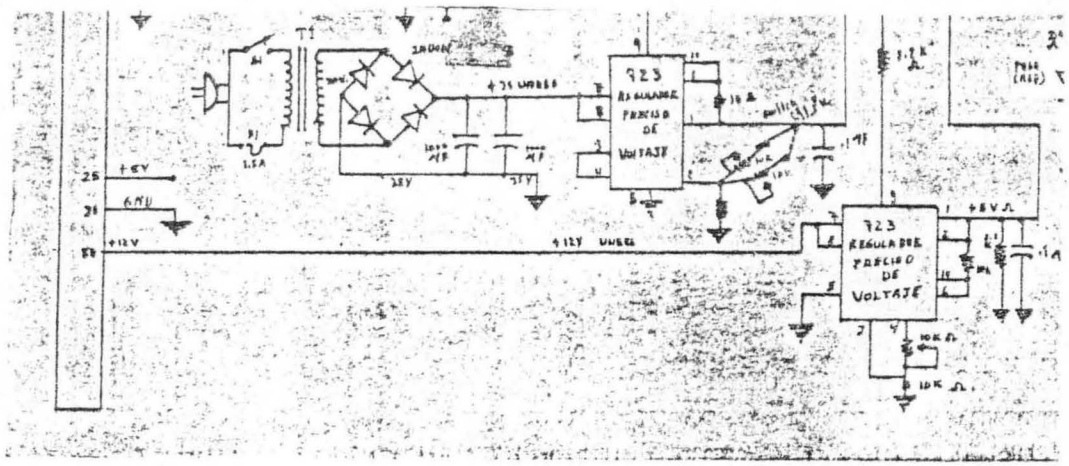
programa, el foco F2 cuando esta en el modo lectura y el foco F3 cuando el programador esta prendida.

4.5 FUENTE DE ALIMENTACION

La fuente de alimentacion se encarga de suplir y distribuir el voltaje necesario a los demas componenetes para llevar a cabo cualquiera de las dos operaciones.

Esta fuente esta integrada por una serie de dispositivos como transformadores de voltaje, diodos, capacitores, resistencias, transistores, relevadores y reguladores de voltaje.

=== FUENTE DE ALIMENTACION ===

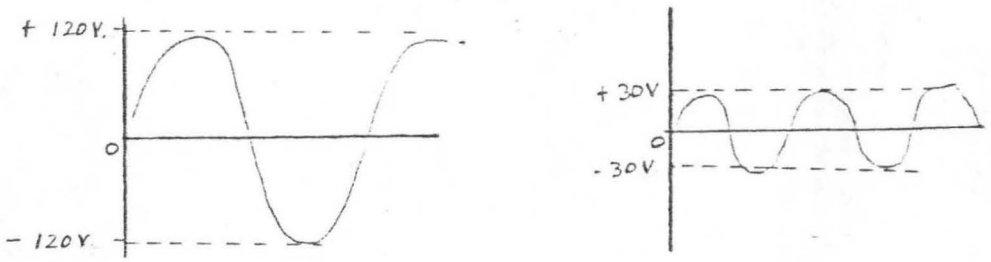


- FIGURA 4.6 -

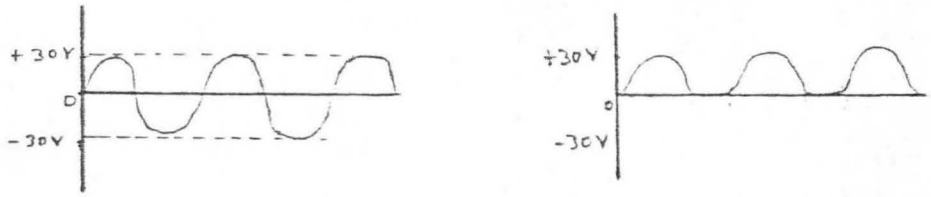
El voltaje que se aprovecha de la microcomputadora atravez de la tarjeta de periferico, no es lo suficiente como para suplir toda la potencia necesaria. Devido a que el programador

de Eprom's requiere hasta 25V. para la impresion o sellado de bytes en el Eprom. se tiene un conector a la corriente de 120V; pero los 120V. es demasiado voltaje para la programacion y ademas éste requiere de corriente continua y no alterna.

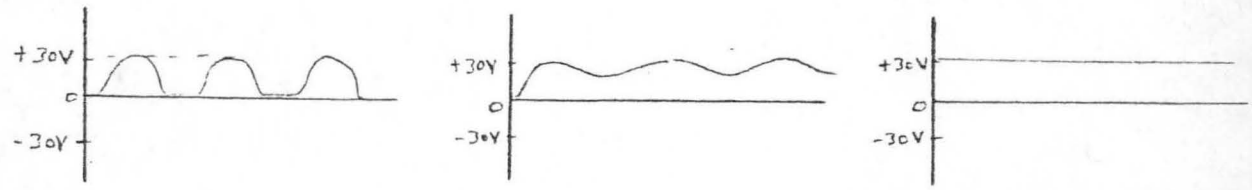
Para esto el transformador de voltaje (Figura 4.6) transforma la corriente alterna de 120V. a 30V.



Luego los diodos rectifican las ondas hasta llegar a lo que se llama "Rectificacion de onda media".



Los condensadores tienen la propiedad de acumular cargas electricas de signos opuestos, y esto hace que las cargas negativas sean aporvechadas.



LLegando de esta forma a obtener corriente continua de

30V. que es lo suficiente para la programación de Eprom's.

Para la impresión de bytes, en algunos Eprom's se requiere un voltaje de 21V. y en otros 25V. El regulador preciso de voltaje es el encargado de regular el voltaje.(Ver apendice A).

4.6 CONSTRUCCION:

Para la construcción del programador se siguió la secuencia del diseño (figura 4.1).

Primeramente se construyó la carroceria el cual contiene todos los dispositivos. Esta carroceria es de una lámina metálica de 25 cm. de largo, 18 cm. de ancho y 15 cm de alto. La parte interna de la carroceria está dividido en 3 niveles (nivel bajo y nivel medio y nivel superior).

Nivel Bajo:

El nivel bajo (base de la carroceria) esta formado por los siguientes dispositivos:

- Transformador de voltaje.
- Los diodos.
- Los capacitores.

Estos elementos estan sujetos firmemente a la carroceria. Para el acabado del primer nivel se hicierón todas las

conecciones requeridas.

Nivel Medio:

El nivel medio contiene a otro grupo de elementos ajustados a una tarjeta. Estos elementos son:

- El P.P.I.
- Los dos reguladores de voltaje (2 chips).
- Los dos habilitadores de voltaje (2 chips).
- El relevador (relay) de voltaje.

La coneccion a los chips fue hecho mediante un enroscado del cable a las patitas. En algunos de los casos se uso soldadura.

Una vez terminado este segundo nivel se hicierón las conecciones necesarias del nivel anterior y este.

Nivel Superior:

En este nivel que es la tapa de la carroceria, están conectados los dispositivos con los cuales se tendrán mayor contacto para la operaciones de programación y lectura de Eprom's.

- Socket de Eprom's.
- Socket de Modulos Personales

- Focos de control.

También en este nivel van ajustados los switch de encendido del programador y el switch para regular el voltaje para la programación.

CAPITULO V

SOFTWARE DE APLICACION

Este tema dá a entender todo lo que concierne al software de aplicación para llevar a cabo las diferentes operaciones con Eprom's.

También nos dá a conocer paso a paso el diseño del sistema software.

5.1 DEFINICION Y MANEJO DE MEMORIA DE LA APPLE II.

Obviamente que para poder programar o leer un Eprom, se tiene que contar con un area de memoria disponible para los programas. La razón por la cual se hace esta separación de area

es para no dañar programas del sistema monitor, o invadir áreas de memoria que esta sistema utiliza. Para esto, se hizo un estudio de la organización de memoria de la siguiente manera:

=== MAPA DE MEMORIA ===

NUMERO DE PAGINA DECIMAL	HEX.	U S O
0	$00	RAM (48K)
.	.	
.	.	
191	$BF	
192	$C0	I/O (2K)
.	.	
.	.	
199	$C7	
200	$C8	I/O ROM (2K)
.	.	
.	.	
207	$CF	
208	$D0	ROM (12K)
.	.	
.	.	
255	$FF	

- FIGURA 5.1 -

La memoria de la Apple II es como un libro de 256 paginas, con 256 localizaciones de memoria en cada pagina. Se divide en tres categorias generales: memoria de L/E (RAM), memoria de solamente lectura (ROM) y posiciones de entrada y salida (E/S). Las diferentes áreas de memoria estan designadas a diferentes funciones.

La memoria ROM esta completamente ocupa por el sistema operativo; mientras que la memoria RAM es usada para almacenar programas y datos. Aunque algunas localizaciones son usadas por

el monitor.

=== MEMORIA R. A. M. ===

NUMERO DE PAGINA DECIMAL	HEX.	U S O
0	$00	PRogramas del Sistema
1	$01	Stack
2	$02	GTNL Buífer de Entrada
3	$03	Localizacion del Vector del monitor
4 . . 7	$04 . . $07	Texto y Graficos de Baja Resolucion Primaria.
8 . . 11	$08 . . $0B	Secundaria
12 a 31	$0C a $1F	
32 a 63	$20 a $3F	Graficas de Alta Resolucion Primaria
64 a 95	$40 a $5F	Secundaria
96 a 191	$60 a $FF	

LIBRE

- TABLA 5.2 -

Como se observa en la tabla 5.2, se tiene espacio libre de 19 páginas en la parte superior y 95 páginas en la parte inferior. Ahora el dos requiere de 42 páginas de memoria, el area de trabajo para los programas 40 páginas y el sistema 40 páginas llegando a necesitar un total de 122 páginas.

Como el espacio libre de memoria no es lo suficiente de

acuerdo al requerimiento del sistema. se hara el uso del area de gráficas quedando quedando definida de la siguiente manera:

=== AREAS QUE USA EL SISTEMA ===

NUMERO DE PAGINA DECIMAL	HEX.	U S O
0	$00	Programas del Sistema
1	$01	Stack
2	$02	GTLN Buffer de Entrada
3	$03	Localizaciones del vector del monitor.
4 . 7	$04 . $07	Texto y Graficas de Alta Resolucion Primaria.
8 . 11	$08 . $0B	Secundaria
12 . 79	$0C . $32	SISTEMA
80 . 112	$50 . $70	AREA DE TRABAJO
113 . 191	$71 . $BF	INTEGER BASIC

- TABLA 5.3 -

5.2 DIRECCIONAMIENTO DE PUERTOS Y BYTES DE CONTROL

El sistema software es el encargado de manipular el direccionamineto de puertos y bytes de control.

Sabemos que el (PPI) es ta organizado de 3 puertos y un puerto de control adicional. Para activar o habilitar un determinado puerto se requiere de una dirección del espacio de I/O para tarjeta de periferico para enviar la señal. La

obtención de estas direcciones es como sigue:

Espacio de I/O Para Tarjeta de Periférico:

Cada ranura tiene asignada 16 localizaciones de memoria, comenzando en la localizacion $c080 hasta $c0ff para propositos de entrada y salida.

=== ESPACIO DE ENTRADA Y SALIDA ===

DIREC.	NUMERO DE SLOT (RAMURA)								
	$0	$1	$2	$3	$4	$5	$6	$7	$8
$C080 $C090 $C0A0 $C0B0 $C0C0 $C0D0 $C0E0 $C0F0	ENTRADA / SALIDA PARA EL NUMRO DE RANURA								

- TABLA 5.4 -

Para la ranura 0, estas 16 localizaciones caen en el rango de memoria $c080 a $c08f, para la ranura 1 esta en el rango de $c090 a $c09f, etc.

Ahora, depende en que ranura se conecte la tarjeta de périferico. Se permite del 1-7. La ranura 0 no puede ser conectada debido a que no es un dispositivo periférico. pr#0 especifica la pantalla de visualización.

=== DIRECCIONES A USAR ===

8255 PUERTO	DIRECCIONES DE PUERTOS CPU I/O						
	PTO-1	PTO-2	PTO-3	PTO-4	PTO-5	PTO-6	PTO-7
A	$C09C	$C0AC	$C0BC	$C0CC	$C0DC	$C0EC	$C0FC
B	$C09D	$C0AD	$C0BD	$C0CD	$C0DD	$C0ED	$C0FD
C	$C09E	$C0AE	$C0BE	$C0CE	$C0DE	$C0EE	$C0FE
CONTROL	$C09F	$C0AF	$C0BF	$C0CF	$C0DF	$C0EF	$C0FF

- TABLA 5.5 -

Una vez definida la ranura, se tiene 16 direcciones de I/O; pero el sistema necesita solamente de 4 direcciones para los 3 puertos y el puerto de control adicional. Se tomo las cuatro ultimas direcciones como se muestra en la tabla 5.5.

Por medio de las direcciones de I/O ya definidas, serán enviados los bytes de control al 8255 para SET o RESET las líneas del puerto C.

=== BYTES DE CONTROL ===

BYTE DE CONTROL	RESPUESTA DE 8255	FUNCION
00	Reset PC0	Reset Eprom ADD A8.
01	Set PC0	Set Eprom ADD A8.
02	Reset PC1	Reset Eprom ADD A9
03	Set PC1	Set Eprom ADD A9
04	Reset PC2	Reset Eprom ADD A10
05	Set PC2	Set Eprom ADD A10
06	Reset PC3	Reset Eprom ADD A11
07	Set PC3	Set Eprom ADD A11
08	Reset PC4	Turn Off 25.5V.
09	Set PC4	Turn On 25.5V.
0A	Reset PC5	Turn Off 5V.
0B	Set PC5	Turn On 5V.
0C	Reset PC6	Relay Program Mode
0D	Set PC6	Relay Read Mode
0E	Reset PC7	Reset Eprom ADD A12
0F	Set PC7	Set Eprom ADD A12
80	Todas las Puertas Modo Salida	Modo Programa
82	Los Puertos A, C de Salida, B de entrada	Modo Lectura

- TABLA 5.6 -

También nos muestra la tabla 5.6 los bytes de control para

el modo lectura y programacion.

5.3 TAREAS DEL SISTEMA SOFTWARE

El sistema software realiza las siguientes tareas:

a) Cargar el programa al espacio de trabajo.

b) Librar el espacio de trabajo a un archivo en disco.

c) Pasar espacio de trabajo a Eprom.

d) Copiar Eprom al espacio de trabajo.

e) Listar el espacio de trabajo.

f) Sacar listado del espacio de trabajo.

g) Comparar espacio de trabajo con Eprom.

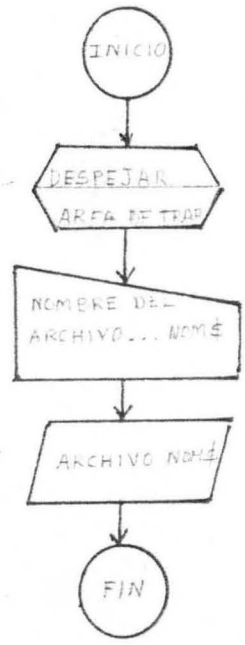
h) Despejar area de trabajo.

Cada tarea del sistema forma un módulo. por lo tanto el sistema consta de 8 módulos.

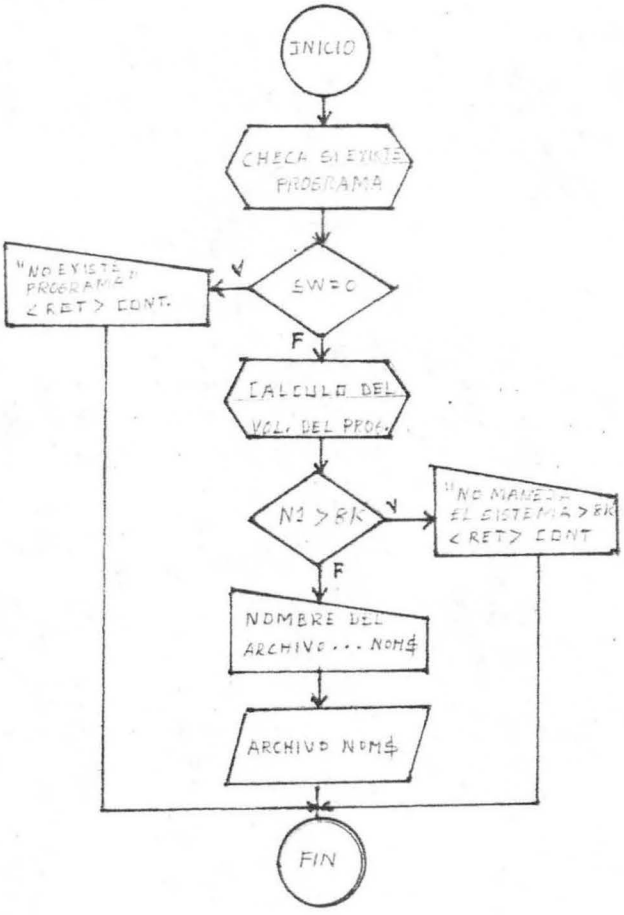
5.4 DESCRIPCION Y DIAGRAMA DE FLUJO DE LOS MODULOS:

Algunos de los módulos del sistema estan escritos parte en lenguaje Basic y parte en lenguaje Ensamblador como es el caso del modulo (c). Mientras que otros estan escritos solamente en Basic como el modulo (a).

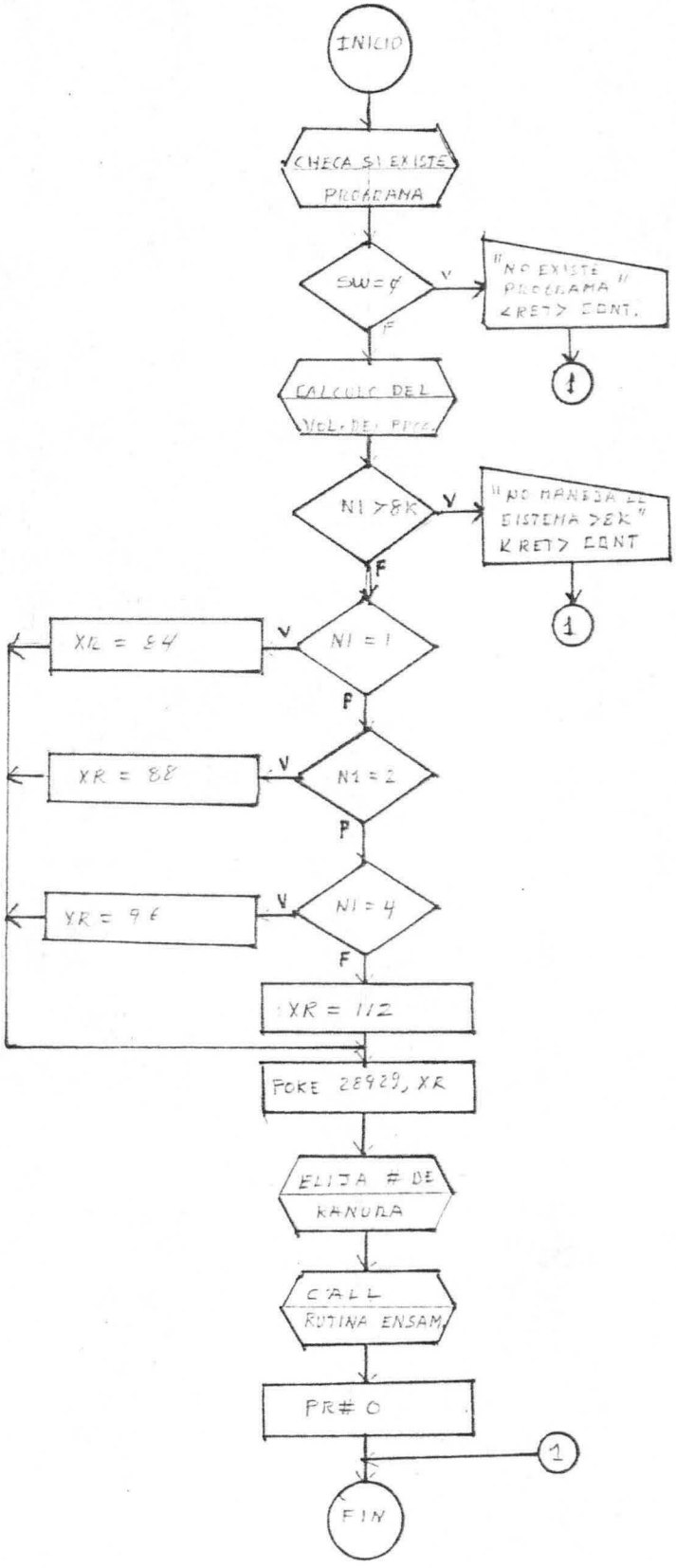
a) CARGAR EL PROGRAMA AL ESPACIO DE TRABAJO;



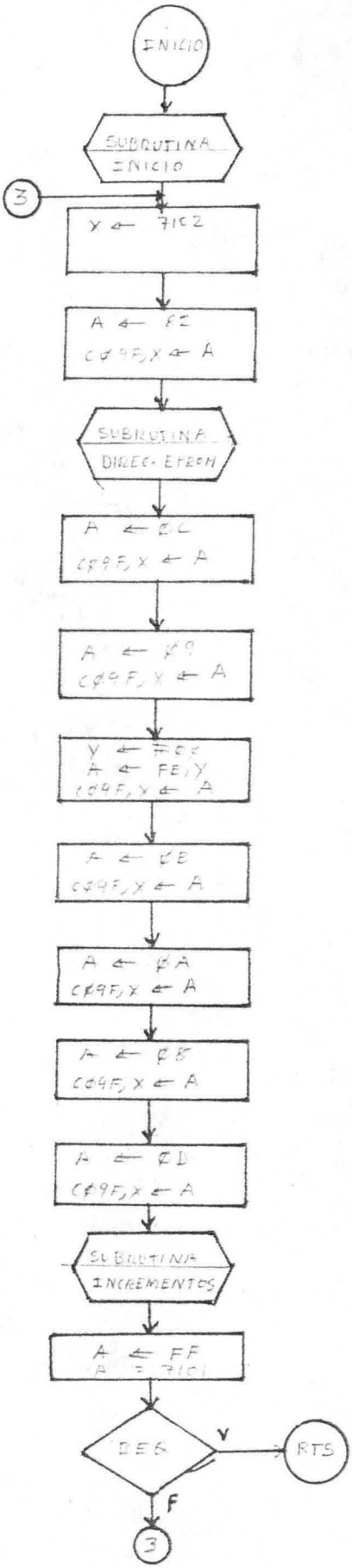
b) LIBRAR ESPACIO DE TRABAJO A ARCHIVO EN DISCO:



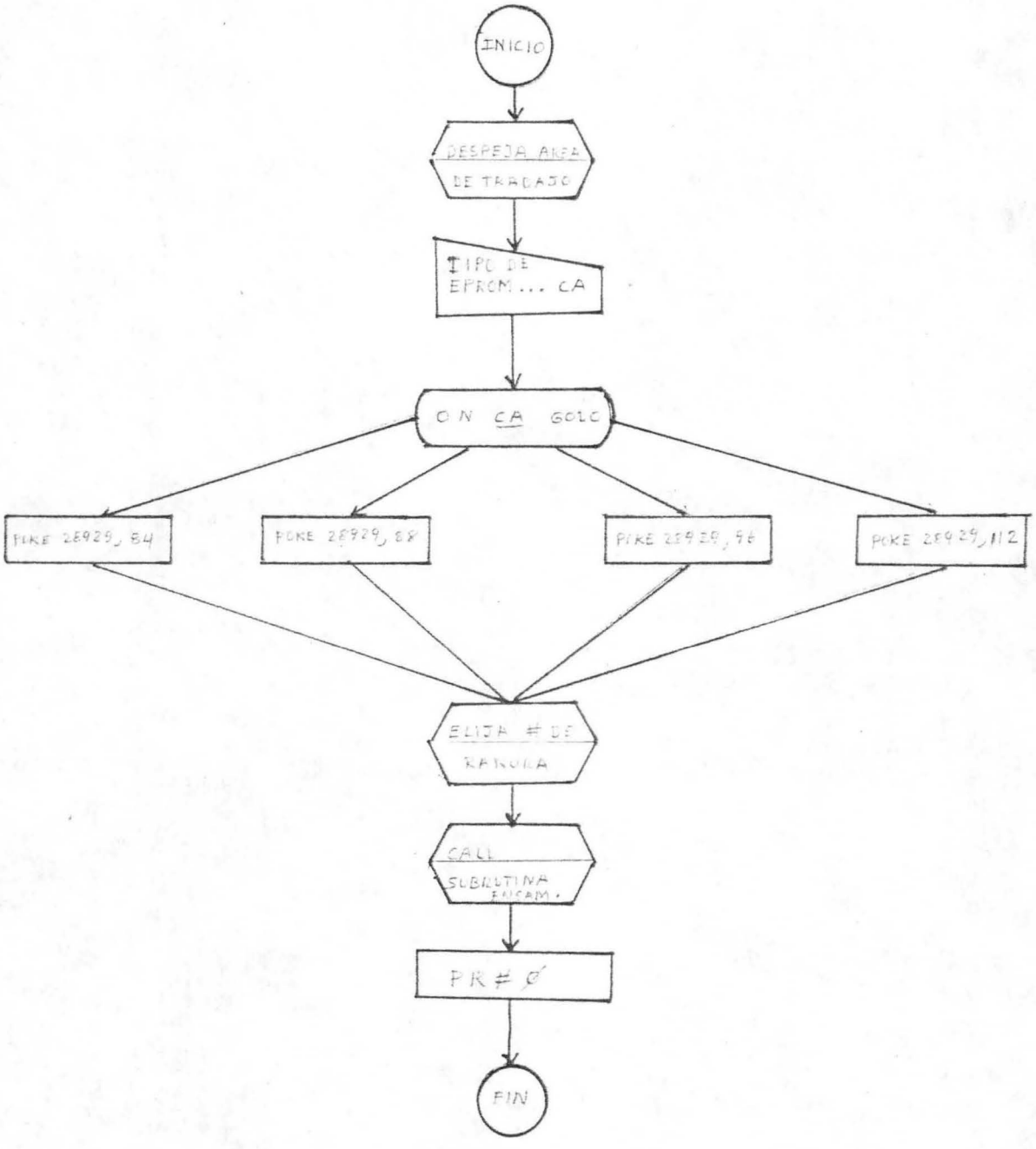
c) PASAR ESPACIO DE TRABAJO A EPROM:



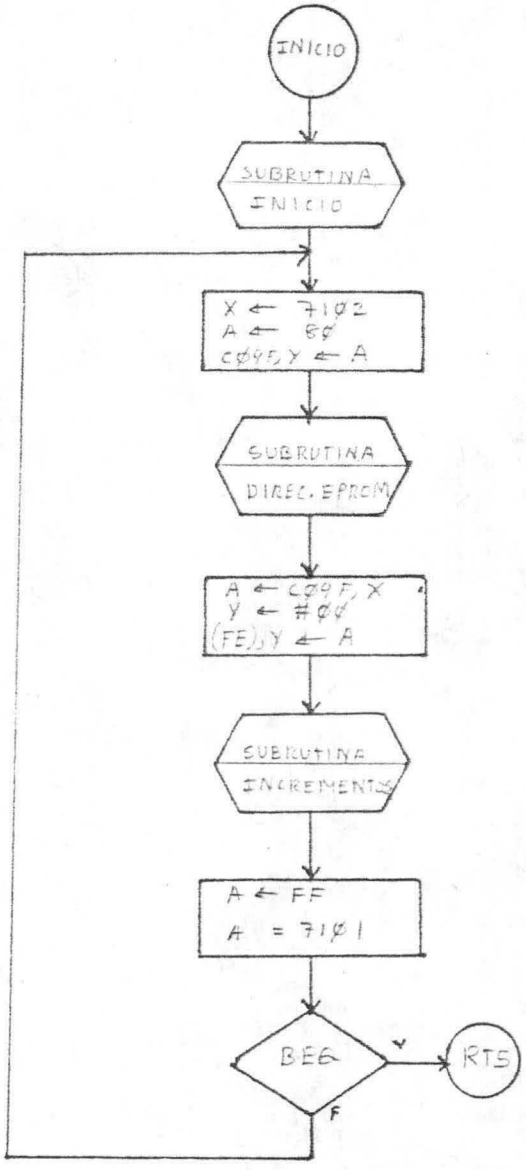
Rutina Ensamblador:



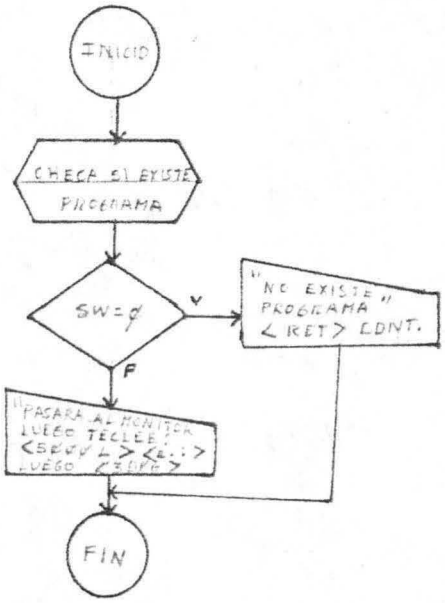
a) COPIAR EPROM AL ESPACIO DE TRABAJO:



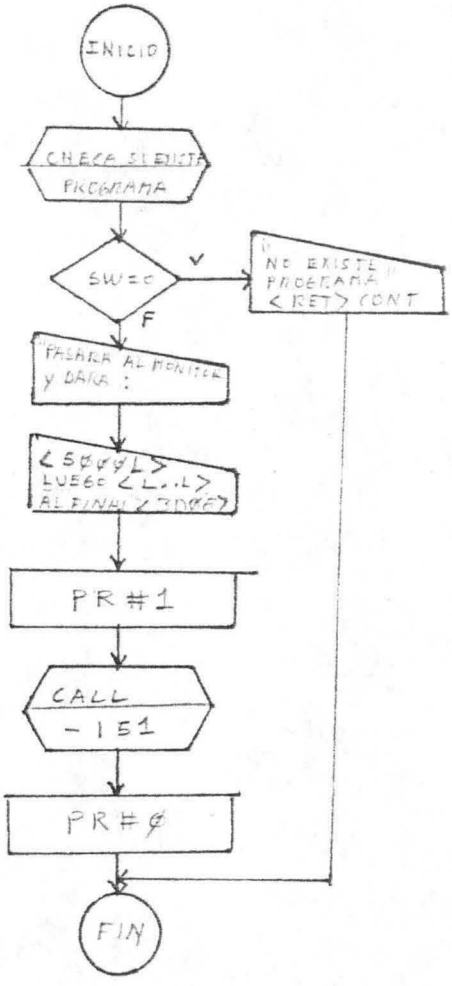
Rutina Ensamblador:



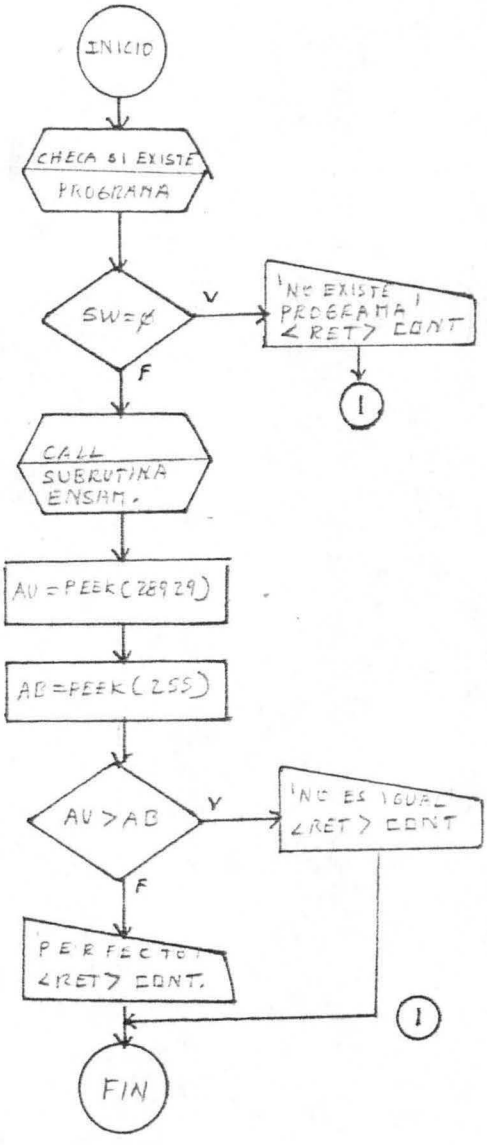
e) LISTAR ESPACIO DE TRABAJO:



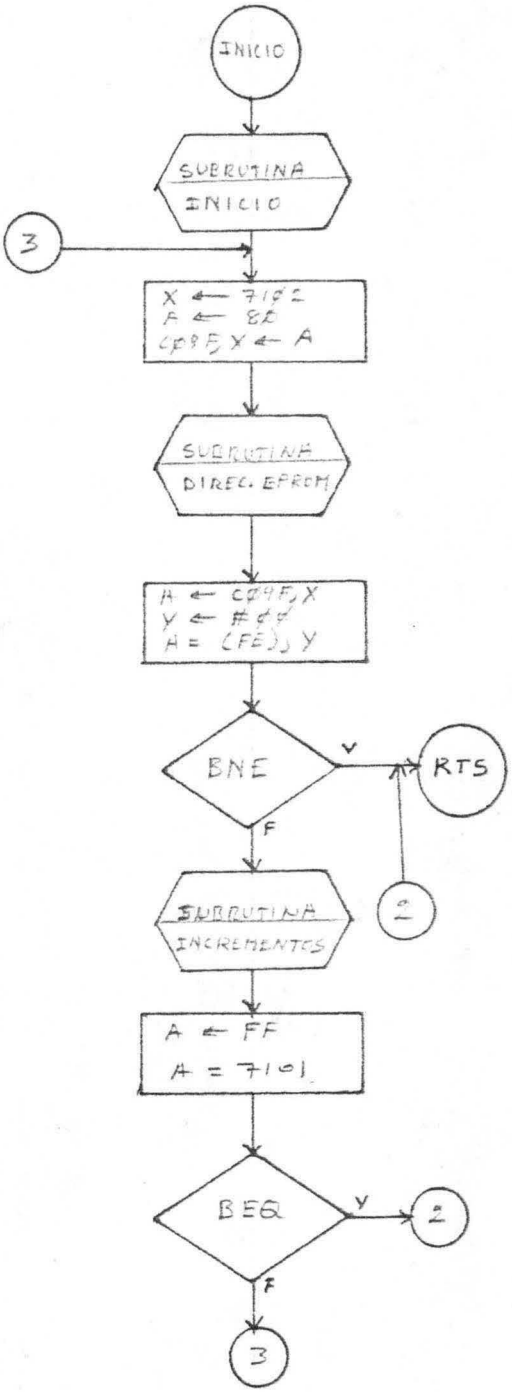
f) SACAR UN LISTADO DEL ESPACIO DE TRABAJO:



g) COMPARAR ESPACIO DE TRABAJO CON EPROM:

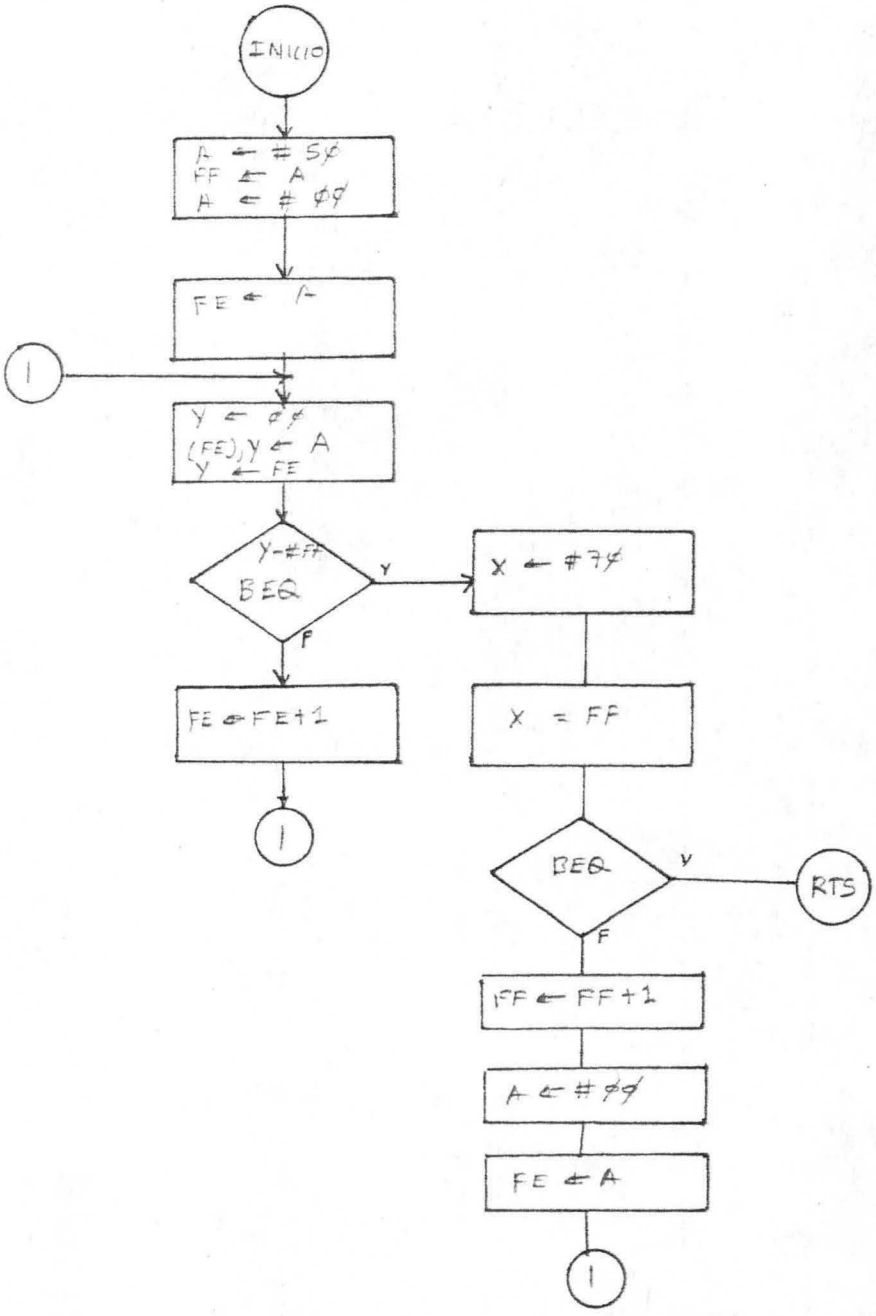


Rutina Ensamblador:



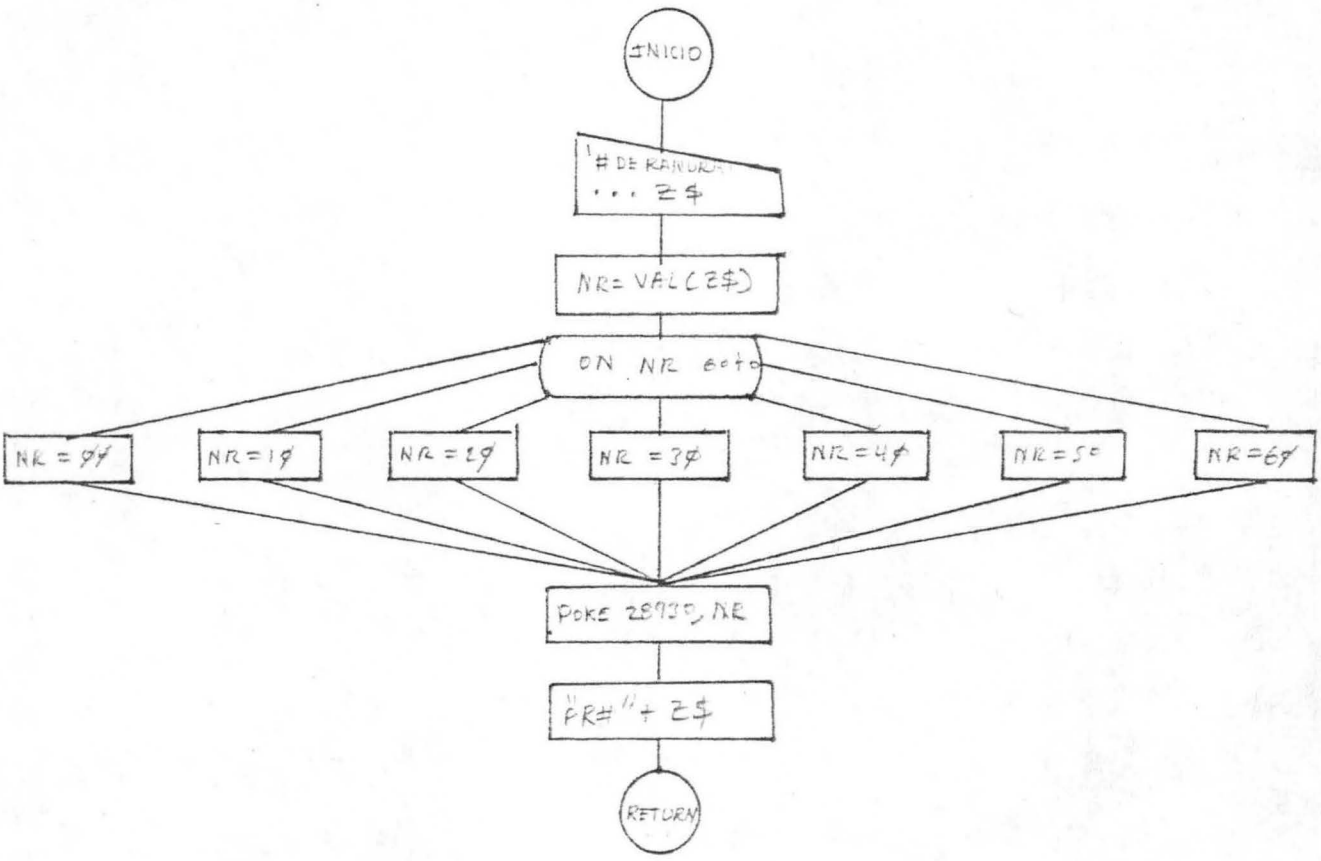
h) DESPEJAR AREA DE TRABAJO:

Rutina Ensamblador:

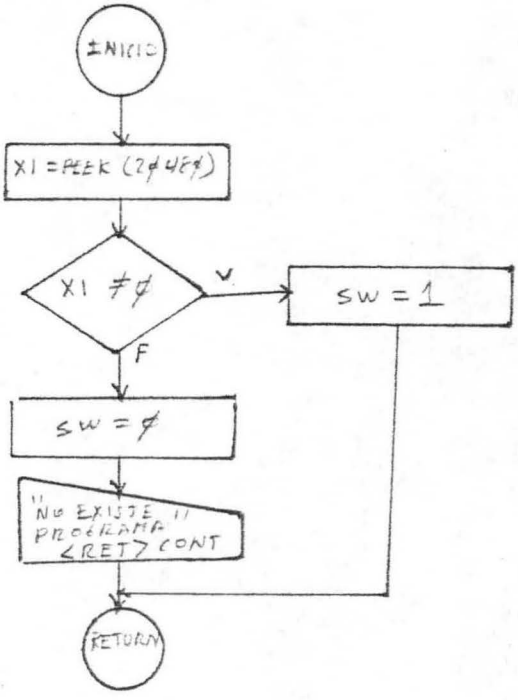


RUTINAS AUXILIARES EN BASIC:

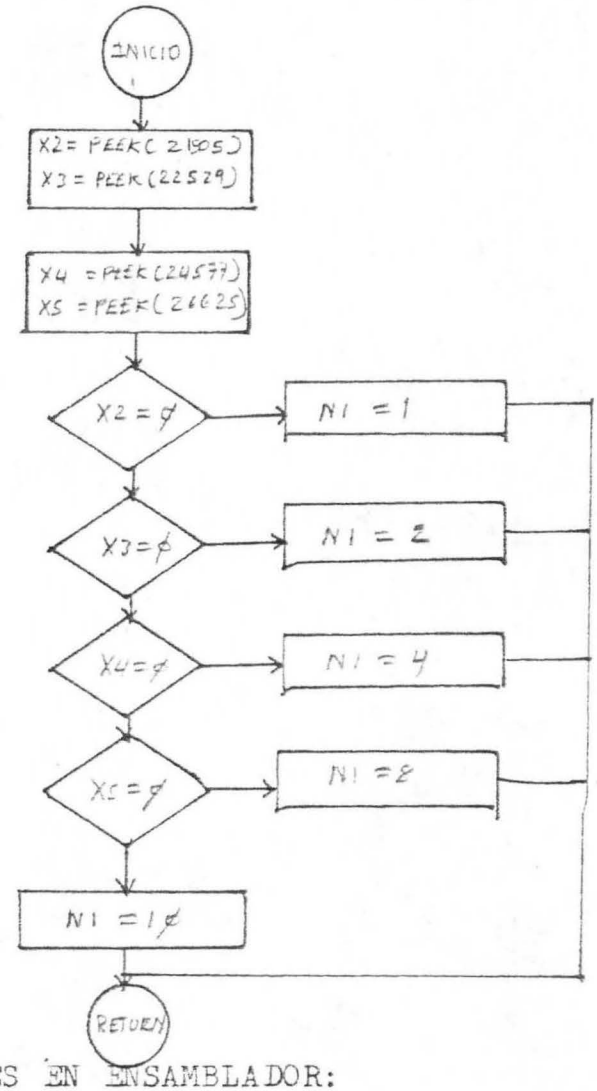
1.- RUTINA SELECCION DE RANURA



2.- RUTINA QUE CHECA SI EXISTE PROGRAMA EN EL AREA DE TRAB.

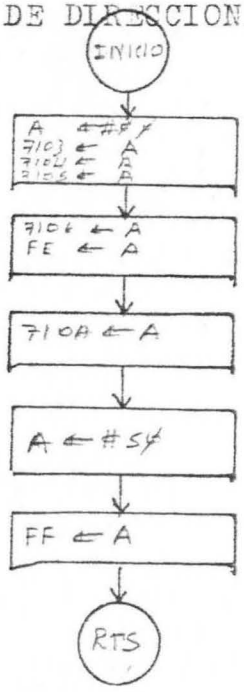


3.- RUTINA CALCULO DEL VOLUMEN DEL PROGRAMA.

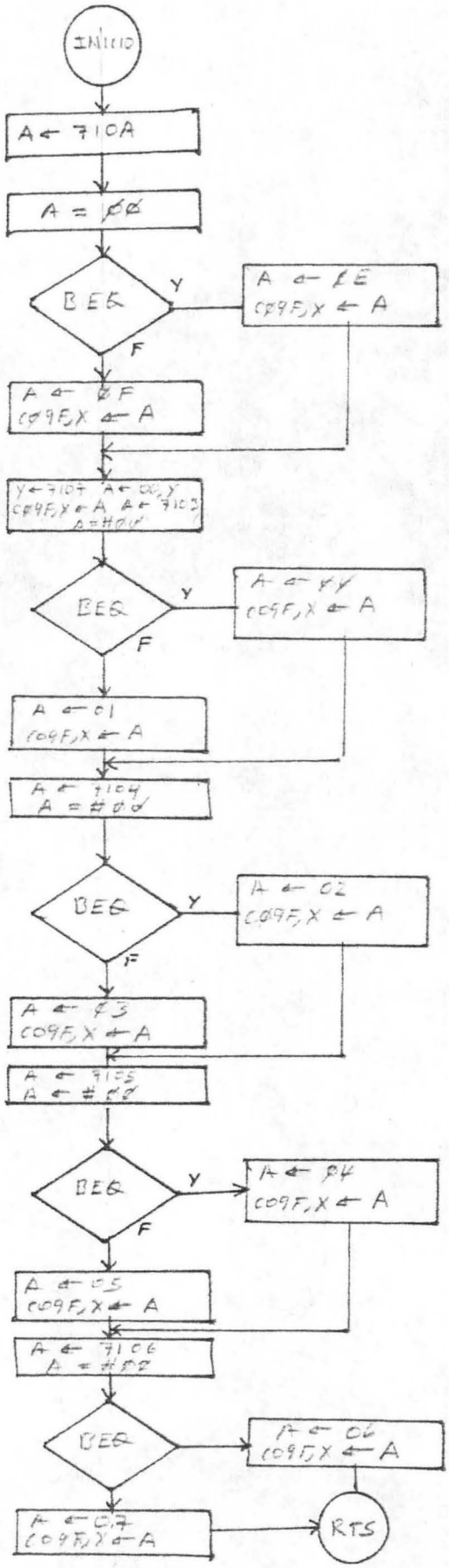


RUTINAS AUXILIARES EN ENSAMBLADOR:

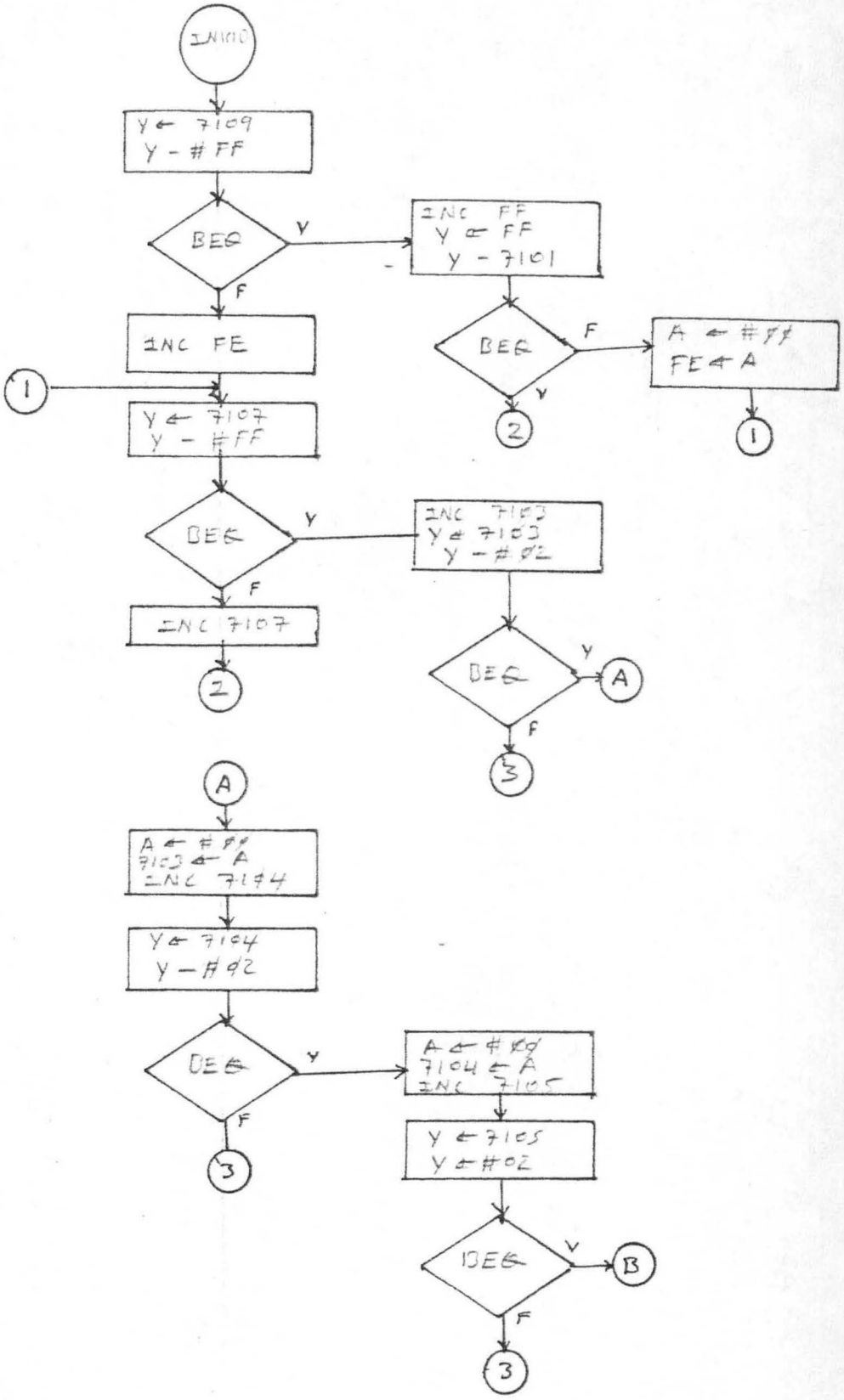
1.- RUTINA INICO DE DIRECCIONES AUXILIARES EN 00

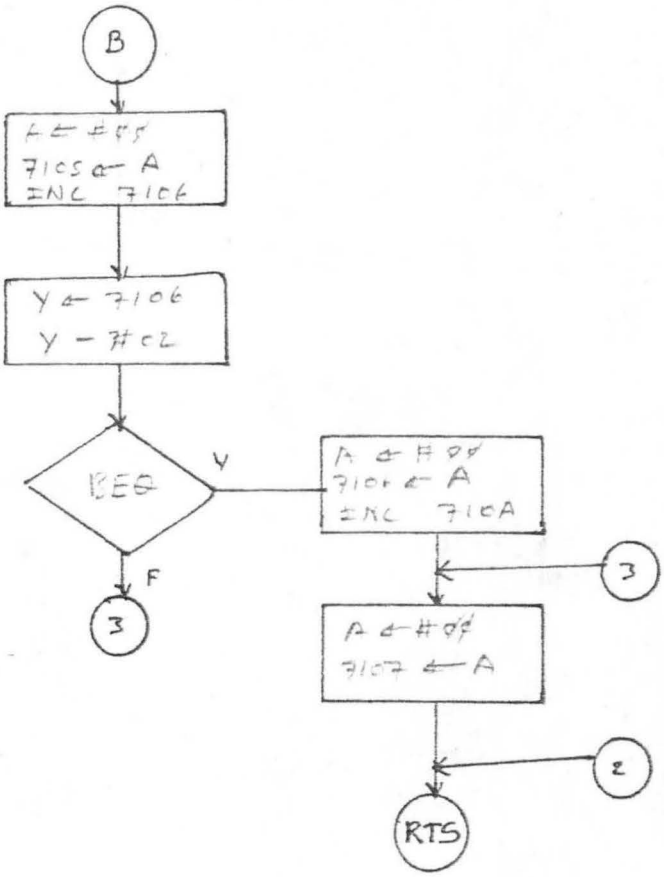


2.- RUTINA DE DIRECCIONAMIENTO DE EPROM:



3.- RUTINA INCREMENTO DE DIRECCIONES DE MEMORIA Y EPROM:




B
A ← #$5
710$ ← A
INC 7106
Y ← 7106
Y - #$2
BEQ
Y
A ← #$0
7104 ← A
INC 710A
F
3
3
A ← #$0
7107 ← A
2
RTS

========== C A P I T U L O V I

PROCEDIMIENTO DE OPERACION

El objetivo principal que persigue este capítulo, es guiar al usuario en la forma de operación del sistema.

Es aconsejable que el usuario lea este manual antes de empezar a manipular el sistema.

6.1 PREPARACION Y ENCENDIDO:

Para el encendido del microcomputador y el programador de Eprom's tome en cuenta los siguiente pasos:

1.- Compruebe que la micro este conectada al socket

de la corriente.

2.- Cheque que el monitor este conectado ala micro al igual que la impresora y los drives.

3.- Enchufe la tarjeta interfaz para el programador de Eprom's en las ranuras (1-7).

4.- Conecte el programamdor de Eprom's al socket de corriente.

5.- Coloque el diskette Sistema Software en el drive 1 y su diskette en el drive 2.

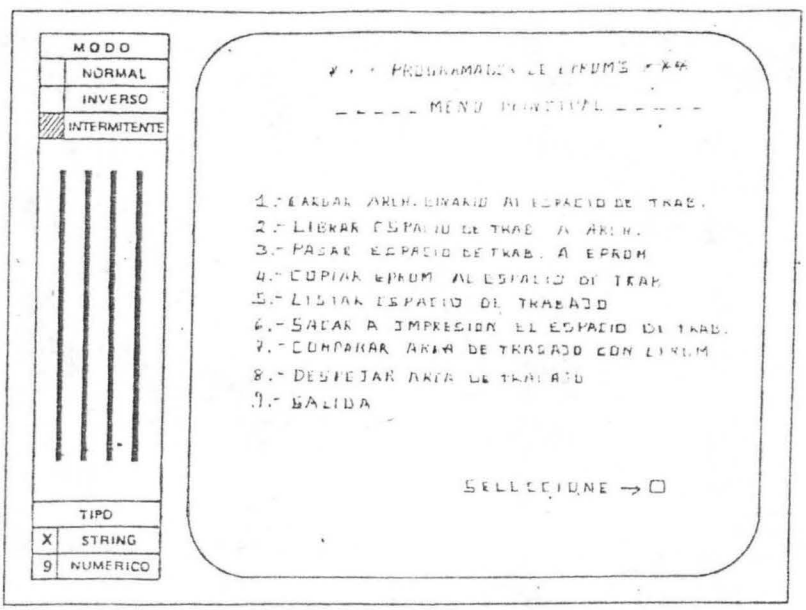
6.- Encienda el microcomputador y luego la pantalla.

7.- Encienda el Proramador de Eprom's.

Una vez que haya ejecutado estos pasos, el Sistema inicializara automaticamente y mostrara en la pantalla el Menu Principal.

=== MENU PRINCIPAL ====

PROGRAMA PROGRAMADOR DE EPROM'S PANTALLA NUM. 1



DESCRIPCION

Si existe algún error en el encendido, avise al encargado del soporte técnico.

En la figura 6.1 se observa que existen 9 opciones, 7 de los cuales no es conveniente discutirlos.

Para facilitar en entendimiento del sistema se tomo como ejemplo las dos opciones mas importantes:

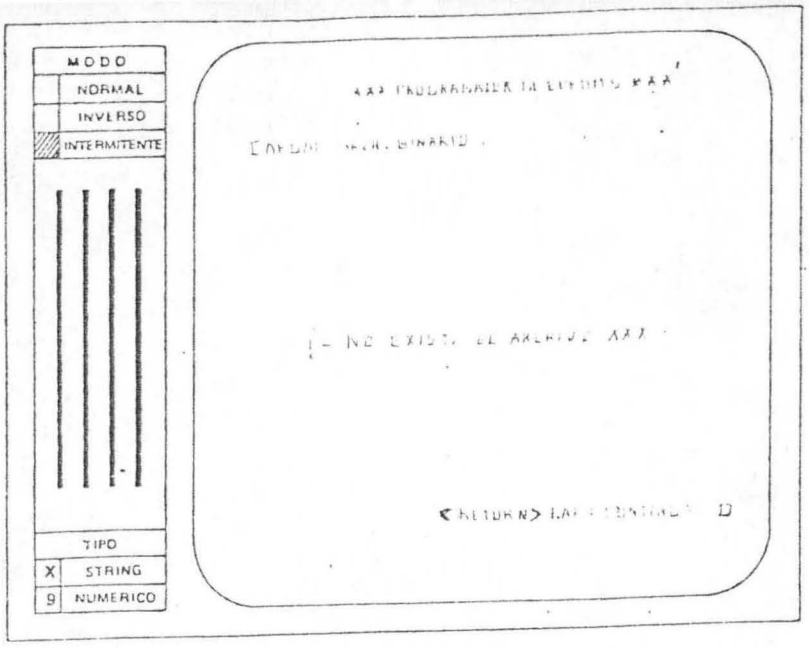
6.2 ALAMACENAMIENTO DE INFORMACION A EPROM'S

Elija la opcion 3 para poder programar un eprom.

Antes de llevar a cabo esta opción, supuestamente tuvo que haber realizado los dos siguiente puntos:

1.- Su programa Ensamblador tiene que estar ensamblado en código binario.
2.- Este programa ya ensamblado tiene que estar en el área de trabajo.

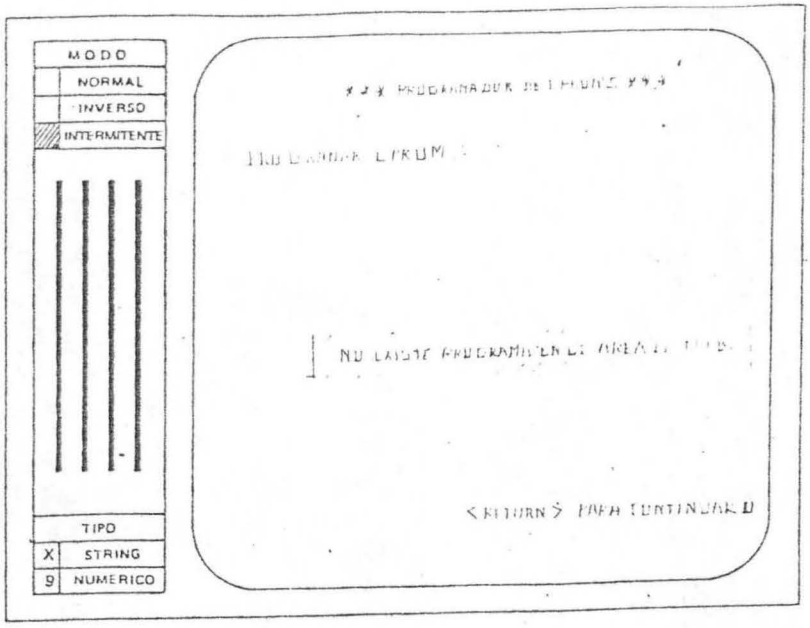
El sistema no puede cargar al área de trabajo un programa ensamblador que no este en codigo binario. Al momento de cargar un programa en ese estado la pantalla mostrara el siguiente mensaje:



- FIGURA 6.3 -

O en otro caso de que se vaya directamente a la opción sin haber cargado previamente el programa aparecerá:

PROGRAMA _____ PANTALLA NUM. ____

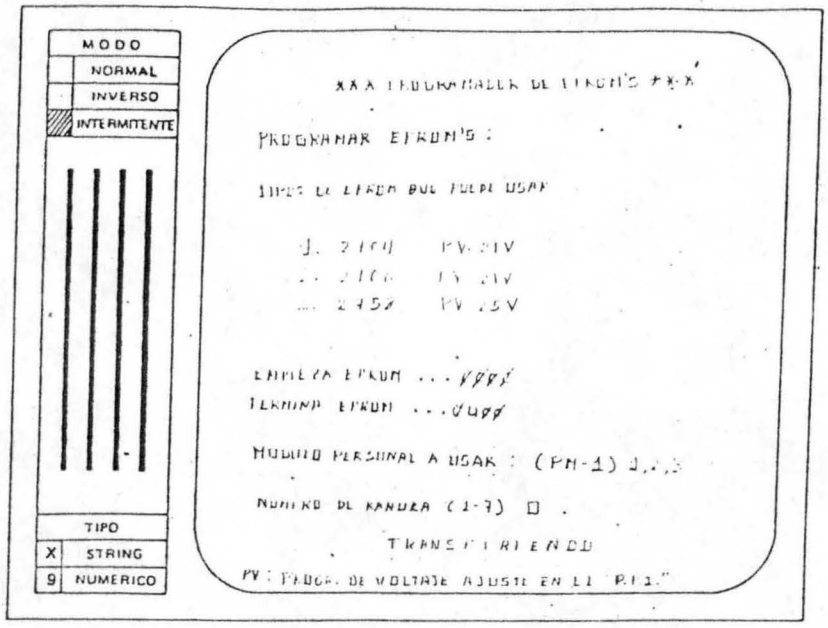


- FIGURA 6.3 -

Asi se considera dichos puntos, continuara la ejecucion. El sistema calculará el volumén de su programa en el area de

trabajo y aparecera en la pantalla los tipos y capacidades de Eprom s que puede usar para dicha capacidad.

PROGRAMA PROGRAMADOR DE EPROM PANTALLA NUM. 4



DESCRIPCION

- FIGURA 6.4 -

Como se observa en la figura 6.4, tiene que realizar los siguientes pasos.

1.- Conecte cualquiera de los Chips permitidos en el cocket de Eprom's.

2.- Conecte el modulo personal que se requiere para ese tipo de Eprom.

3.- Ajuste en el programador de Eprom's el voltaje necesario para la programacion de ese tipo de Chips.

4.- Elija el tipo de ranura a usar (supuestamente el numero de ranura a elejir tiene que ser la que este conectada la tarjeta periferico.

5.- Pulse return.

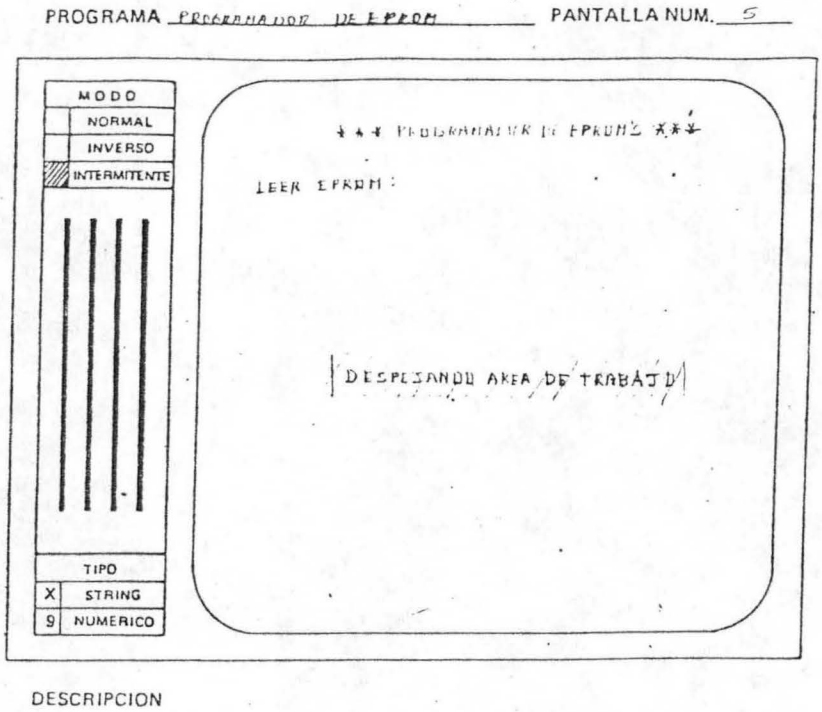
Luego de pulsar Return empieza a transferir el programa.

IMPORTANTE: Fíjese bién el tipo de Eprom y el módulo personal a conectar. también asegurese que el voltaje de programación sea el correcto.

Si alguno de estos pasos no se lleva acabo correctamente, la programación sera errónea.

6.3 LECTURA DE INFORMACION DE EPROM'S:

Para leer un Eprom elija la opción 4. No se alarme si aparece el mensaje:

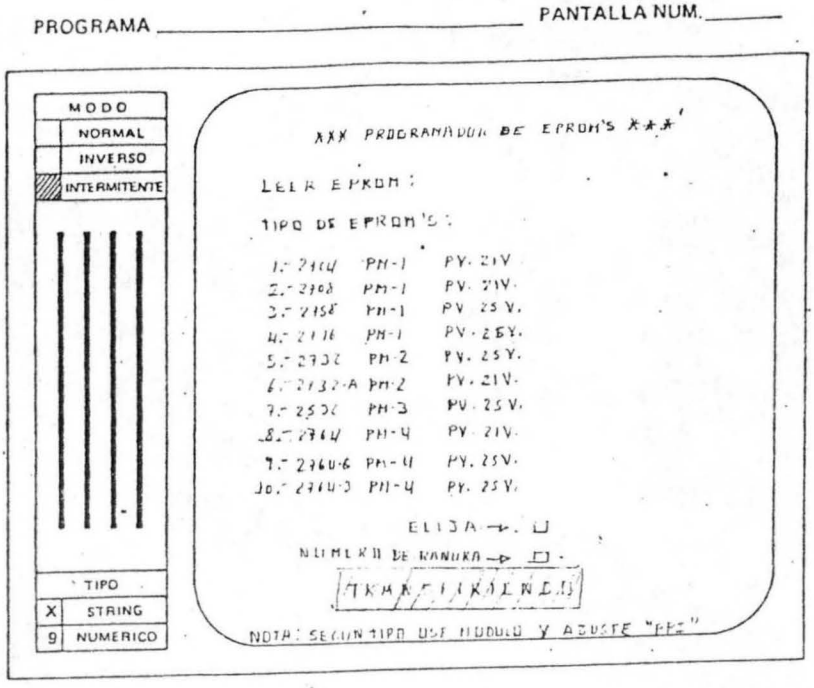


DESCRIPCION

- FIGURA 6.5 -

Antes de comenzar a leer información de un Eprom, despejará el área de trabajo (Coloca en ØØ).

Luego aparece en la pantalla los tipos y capacidades de Eprom's que se puede programar (fig.6.6). De acuerdo al tipo de Eprom que se quiera leer elija su opcion.



- FIGURA 6.6 -

Tambien se puede observar que seguido del tipo de Eprom esta el modulo personal a usar. Antes de continuar haga los siguientes ajustes:

1.- Conecte el Chips que va a leer.

2.- Conecte el módulo personal.

3.- Elija la ranura (1-7). (Obviamente elija la ranura en la cual esta conectada el periferico.

4.- Pulse Return.

Seguidamente empezará a trasferir el programa.

IMPORTANTE: Fíjese bién el modulo personal a conectar y la tarjeta este conectada en la ranura seleccionada.

6.4 BORRAR INFORMACION DE EPROM:

Como se mensionó al principio de nuestro estudio, estas memorias tiene la facilidad de borrarse.

Osea un Eprom programado, puede perder la informacion que se encuentre grabada en ella.

Esta operación no lo hace el sistema por que es una tarea independiente.

La forma de borrar la informacion de ■ Eprom es aplicando una luz ultravioleta al Chips unos segundos. una ves aplicado este rayo de luz, el Eprom quedará nuevamente en el estado uno (estado inicial) y puede volverse a programar.

C O N C L U S I O N E S

1.- La utilización de este equipo periferico facilitara el desarrollo de nuevas aplicaciones de software.

2.- Este instrumento puede contribuir a resover de una forma mas eficaz algunos problemas que se presenten en la aplicación de los sistemas.

3.- Con esta herramienta y mediante la operacion de "programacion de eprom's" permite sacar un mejor provecho de nuestro equipo computacional.

4.- Mediante el uso de los Lenguajes Basic y Ensamblador, hacen de esta un sistema sencillo, flexible y muy rápido.

5.- La gran capacidad que tienen los Eprom's nos permiten realizar una serie de tareas que de otro modo serían imposibles.

6.- Este trabajo proporcionará a los alumnos de computación a comprender la aplicacion de los principios aprendidos , en la practica que incluyen construcción de aparatos de aspecto profecional.

7.- Considero que la electronica ligado con la computacion es un campo que tiene muchas promesas para el futuro.

A P E N D I C E (A)

TERMINOLOGIA USADA (Microcomputadora)

Hardware: Es la parte que incluye el diseño y la realización del computador, es decir la parte electronica del mismo.

Software: Una serie de programas que permite que un computador realize una tarea determinada.

Bit: Es un digito binario, ya sea un 0 o 1.

Byte: Es un conjunto de ocho digitos binarios u ocho bit's.

Localidad de Memoria: Se compone de 4 digitos hexadecimales e indica la posición donde se encuentra un contenido de memoria.

Contenido de Memoria: Se compone de 2 digitos hexadecimales que contiene una instrucción, dato u operando.

Bus de Datos: Es un conjunto de conductores usados para acarrear

datos del / al CPU.

Bus de Direcciones: Son las lineas que direccionan las memorias

(16 bit's).

Bus de Control: Son lineas que se usan para sincronizar

la operacion del CPU con los circuitos externos

(de 9 a 12 bit's).

Alto Byte: Son 8 bit's de mayor peso en una palabra de 16 bit's (2A2f).

Bajo Byte: Son 8 bit's de menor peso en una palabra de 16 bit's (2a2F).

Lenguaje Asembler: Son una serie de instrucciones Mnemonicos.

Lenguaje Maquina: Son los codigos de operación en hexadecimal que

identifica el microprocesador para ejecutar

instrucciones.

Stack: Es un apilamiento o almacenamiento de datos en

localidades de memorias continuas.

Registros: Son contenidos con los cuales podemos hacer funciones especiales como:

- Almacenamiento de dato (Acumulador A).
- Contador de programa (PC)
- Almacenamineto de estados.
- Almacenamiento del estado mas alto del stack (SP).
- Almacenamiento de datos y como contador (reg. X, Y).

TERMINOLGOGIA USADA (Electrónica)

Corriente : Es el movimiento progresivo de electrones libres, a lo largo de alambre o conductor.

Resitencia: Cualquier efecto de oposicion, que dificulta el movimiento libre atravez de los alambres.

Corriente Continua: En ella no varia la intensidad de la corriente o tensión.

Corriente Alterna: El flujo de electrones se invierte alternandose ritmicamente. La amplitud cambia regularmente.

Diodo : Dispositivo rectificador fundado en el uso de semiconductores.

Transformador De Voltaje : Aparato para elevar o reducir las tensiones eléctricas o para hacer variar algunas de las caracteristicas de la corriente alterna.

Transistor: Dispositivo fundado en el uso de semiconductores que actua en los circuitos como lámparas, detectores, amplificadores u osciladores.

Relevador : Dispositivo que se interpone en ciertos organos de mando con el objeto de que una impulsión electrica breve o de escasa intensidad permita gobernar un aparato, regular una corriente o alguna acción en lo que requiera el relevador.

Capacitor : Dispositivo que acumula cargas lectricas de signos opuestos.

A P E N D I C E (B)

```
?CHR$(9)"80N"


LIST

   REM  *RUTINA DE VALIDACION*
0  GOTO 160
5  PRINT  CHR$ (7):: GOTO 40
0  HTAB Z6%: VTAB Z7%
0 Z1% = 0:Z2% = 0: FOR Z5 = 1 TO Z0%: PRINT "-":: NEXT : FOR Z5 = 1 TO Z0%: P
1  CHR$ (8):: NEXT :Z$ = ""
0  GET Z2$:Z3% =  ASC (Z2$): IF Z3% = 13 THEN  PRINT  SPC( Z0% - Z1%): RETURN
0  IF Z3% = 8 THEN  GOTO 130
0  IF Z1% = Z0% THEN 15
0  ON Z4% GOTO 100,90
0  ON (Z3% < 48 OR Z3% > 57) AND (Z3% <  > 45 OR Z1% <  > 0) GOTO 15: GOTO 12
0  ON (Z3% < 32 OR Z3% > 95) GOTO 15: GOTO 120
00  ON (Z3% < 48 OR Z3% > 57) AND (Z3% <  > 45 OR Z1% <  > 0) AND (Z3% <  > 4
R Z2%) GOTO 15
10  IF Z2$ = "." THEN Z2% = 1
20 Z1% = Z1% + 1: PRINT Z2$:: Z$ = Z$ + Z2$: GOTO 40
30  IF Z1% = 0 THEN  GOTO 15
40  PRINT  CHR$ (8);"-"; CHR$ (8):: Z1% = Z1$ - 1: IF  RIGHT$ (Z$,1) = "." THE
2% - 0
50  IF Z1% = 0 THEN Z$ = "": GOTO 40
55 Z$ = LEFT$(Z$,Z1%): GOTO 40
60  REM  FIN DE RUTINA
00  REM  MENU PRINCIPAL EN PANTALLA
10  GOSUB 3000: VTAB 5: HTAB 7: PRINT "------ MENU PRINCIPAL ------"
20  VTAB 7: HTAB 4: PRINT "1.- CARGAR ARCH.BINARIO AL ESPACIO": HTAB 8: PRINT
E TRABAJO"
30  HTAB 4: PRINT "2.- LIBRAR ESPACIO DE TRAB. A ARCH."
40  HTAB 4: PRINT "3.- PASAR ESPACIO DE TRAB. A EPROM."
50  HTAB 4: PRINT "4.- COPIAR EPROM AL ESPACIO DE": HTAB 8: PRINT "TRABAJO"
60  HTAB 4: PRINT "5.- LISTAR ESPACIO DE TRABAJO"
70  HTAB 4: PRINT "6.- SACAR A IMPRESION EL ESPACIO DE": HTAB 8: PRINT "TRABA

80  HTAB 4: PRINT "7.- COMPARAR ESPACIO DE TRABAJO CON": HTAB 8: PRINT "EPROM

10  HTAB 4: PRINT "8.- DESPEJAR AREA DE TRABAJO"
20  HTAB 4: PRINT "9.- SALIDA"
30  VTAB 22: HTAB 15: PRINT "SELECCIONE ->":Z0% = 1:Z4% = 0:Z6% = 28:Z7% = 22
OSUB 20
40 A1 =  VAL (Z$)
50  IF A1 < 1 OR A1 > 9 THEN 330
60  ON A1 GOTO 400,600,800,1000,1200,1400,1600,1800,2000
00  REM  RUTINA QUE CARGA EL ARCH.BINARIO AL ESPACIO DE TRABAJO
10  GOSUB 1820: GOSUB 3000: VTAB 8: HTAB 5: PRINT "NOMBRE DEL ARCHIVO :"
20 Z0% = 15:Z4% = 2:Z6% = 5:Z7% = 11: GOSUB 20:NOM$ = Z$
30  VTAB 16: HTAB 7: FLASH : PRINT "-- CARGANDO ARCHIVO --": NORMAL
```

```
440  PRINT D$:"BLOAD"NOM$",A$5000"
450  GOTO 200
500  REM  RUTINA LIBRAR ESPACIO DE TRABAJO
610  GOSUB 2200
620  IF SW = 1 THEN 640
630  VTAB 18: HTAB 15: INPUT "<RETURN PARA CONTINUAR> ":Z$: GOTO 200
640  GOSUB 2500
650  IF N1 > 8 THEN 730
660  GOSUB 3000: VTAB 8: HTAB 5: PRINT "NOMBRE DEL ARCHIVO A LIBRAR"
670 Z0% = 15:Z4% = 2:Z6% = 5:Z7% = 11: GOSUB 20:NOM$ = Z$
680  VTAB 16: HTAB 7: FLASH : PRINT "-- LIBRANDO ARCHIVO --": NORMAL
690  IF N1 = 1 THEN  PRINT D$:"BSAVE "NOM$",A$5000,L1024": GOTO 740
700  IF N1 = 2 THEN  PRINT D$:"BSAVE "NOM$",A$5000,L2048": GOTO 740
710  IF N1 = 4 THEN  PRINT D$:"BSAVE "NOM$",A$5000,L4096": GOTO 740
720  PRINT D$"BSAVE "NOM$",A$5000,L8192": GOTO 740
730  GOSUB 3000: VTAB 12: HTAB 7: FLASH : PRINT "NO MANEJA EL SISTEMA > A 8K":
ORMAL : GOTO 630
740  GOTO 200
800  REM  RUTINA QUE QUEMA EL CHIP
810  GOSUB 2200
820  IF SW = 1 THEN 840
830  VTAB 18: HTAB 15: INPUT "<RETURN> PARA CONTINUAR ":Z$: GOTO 200
840  GOSUB 2500
850  IF N1 < 1 OR N1 > 8 THEN 870
860  GOSUB 3000: VTAB 6: HTAB 5: PRINT "TIPOS DE EPROM QUE PUEDE USAR:": GOTO
)
870  GOSUB 3000: VTAB 12: HTAB 7: FLASH : PRINT "NO MANEJA EL SISTEMA > 8K": N
AL : GOTO 830
880  IF N1 = 1 THEN  GOSUB 2700:XR = 84: GOTO 920
890  IF N1 = 2 THEN  GOSUB 2800:XR = 88: GOTO 920
900  IF N1 = 4 THEN  GOSUB 2850:XR = 96: GOTO 920
910  GOSUB 2900:XR = 112
920  VTAB 23: PRINT "VP:PROGR.DE VOLTAJE AJUSTE EN EL "PPI"."
925  POKE 28929,XR
927  GOSUB 930
928  CALL 12800
929  PRINT D$"PR#0": GOTO 200
930  VTAB 19: HTAB 5: PRINT "NUMERO DE RANURA <1-7>":Z0% = 1:Z4% = 0:Z6% = 30:
 = 19: GOSUB 20:NR =  VAL (Z$):RA$ = "PR#" + Z$
940  VTAB 22: HTAB 5: INPUT "TODO LISTO...<RET> PARA CONT.":Z$
950  VTAB 22: HTAB 5: FLASH : PRINT "T R A N S F I R I E N D O ": NORMAL
970  ON NR GOTO 972,974,976,978,980,982,984
972 NR = 0: GOTO 986
974 NR = 10: GOTO 986
976 NR = 20: GOTO 986
978 NR = 30: GOTO 986
980 NR = 40: GOTO 986
982 NR = 50: GOTO 986
984 NR = 60
986  POKE 28930,NR
988  PRINT D$""RA$"": RETURN
1000  REM  RUTINA QUE LEE EPROM
1010  GOSUB 1820: GOSUB 3000: VTAB 4: HTAB 8: INVERSE : PRINT "  L E E R    E
 D M  ": NORMAL
1015  VTAB 6: HTAB 5: PRINT "TIPO DE EPROM'S :"
1020  HTAB 5: PRINT "1.- 2704   PM-1  PV.21V.": HTAB 5: PRINT "2.- 2708   PM-1
```

```
V.21V."
030  HTAB 5: PRINT "3.- 2758   PM-1  PV.25V.": HTAB 5: PRINT "4.- 2716   PM-1
V.25V."
040  HTAB 5: PRINT "5.- 2732   PM-2  PV.25V.": HTAB 5: PRINT "6.- 2732-A PM-2
V.21V."
050  HTAB 5: PRINT "7.- 2532   PM-3  PV.25V.": HTAB 5: PRINT "8.- 2764   PM-4
V.21V."
060  HTAB 5: PRINT "9.- 2764-Q PM-4  PV.25V.": HTAB 4: PRINT "10.- 2764-3 PM-
PV.25V."
065  VTAB 24: FLASH : PRINT "NOTA:": NORMAL : PRINT "SEGUN TIPO USE MODULO Y
AJUSTE PPI"
070  VTAB 17: HTAB 20: PRINT "ELIJA ->":Z0% = 2:Z4% = 0:Z6% = 30:Z7% = 17: GO
: 20
085 CA =  VAL (Z$)
090  IF CA < 1 OR CA > 11 THEN 1070
095  ON CA GOTO 1122,1122,1122,1124,1126,1126,1126,1128,1128,1128
100  GOSUB 930
110  CALL 12865
120  PRINT D$:"PR#0": GOTO 200
122  POKE 28929,84: GOTO 1100
124  POKE 28929,88: GOTO 1100
126  POKE 28929,96: GOTO 1100
128  POKE 28929,112: GOTO 1100
200  REM  RUTINA QUE LISTA ESPACIO DE TRABAJO
210  GOSUB 2200
220  IF SW = 0 THEN  VTAB 18: HTAB 15: INPUT "<RETURN> PARA CONTINUAR ":Z$: G
 200
230  GOSUB 3000: VTAB 6: HTAB 5: INVERSE : PRINT "LISTA AREA DE TRABAJO:": NO
L : GOSUB 1250
235  CALL  - 151
240  GOTO 200
250  SPEED= 100: VTAB 8: HTAB 5: PRINT "PARA LISTAR SU PROG. PASARA AL MONI-"
TAB 10: HTAB 5: PRINT "TOR (*) Y DARA LA SIGUIENTE INSTRUC-"
253  VTAB 12: HTAB 5: PRINT "CION <5000L> Y VERA SU PROG. LUEGO <L> HASTA"
255  VTAB 14: HTAB 5: PRINT "TERMINAR DE VER SU PROG.": HTAB 5: PRINT "CUANDO
RMINES TECLEA <3DOG>"
260  VTAB 18: HTAB 5: INPUT "LISTO...<RET> PARA CONTINUAR ":Z$
270  SPEED= 255: RETURN
400  REM  RUTINA QUE SACA LISTADO
410  GOSUB 2200
420  IF SW = 0 THEN  VTAB 18: HTAB 15: INPUT "<RETURN> PARA CONTINUAR ":Z$: G
 200
430  GOSUB 3000: VTAB 6: HTAB 5: INVERSE : PRINT "SACAR LISTADO DEL PROGRAMA
 NORMAL : GOSUB 1250
440  PRINT D$:"PR#1"
450  CALL  - 151
460  PRINT D$:"PR#0": GOTO 200
600  REM  RUTINA QUE COMPARA A.DE TRAB.CON EPROM
610  GOSUB 2200
620  IF SW = 1 THEN 1640
630  VTAB 18: HTAB 15: INPUT "<RETURN> PARA CONTINUAR ":Z$: GOTO 200
640  GOSUB 3000: VTAB 6: HTAB 4: INVERSE : PRINT "COMPARA A.DE TRABAJO CON EP
": NORMAL
650  VTAB 12: HTAB 5: SPEED= 100: PRINT "SUPUESTAMENTE ACABA DE COPIAR EPROM"
```

```
660  VTAB 14: HTAB 5: PRINT "O PASAR ESPACIO DE TRABAJO A EPROM.": SPEED= 255
670  VTAB 18: HTAB 10: INPUT "<S> SALIR  <RETURN> CONTINUAR ";Z$
680  IF Z$ = "S" THEN 200
690  IF Z$ <  > "" THEN 1670
700  VTAB 22: HTAB 10: FLASH : PRINT " C O M P A R A N D O ": PRINT D$;""RA$"
710  CALL 12900
720  PRINT D$;"PR#0":AU =  PEEK (28929):AB =  PEEK (255)
730  IF AU > AB THEN 1750
740  VTAB 24: HTAB 7: INPUT " P E R F E C T O  <RET> CONT. ";Z$: GOTO 200
750  VTAB 24: HTAB 5: FLASH : PRINT " N O  E S  I G U A L":: NORMAL : INPUT "
ET> CONT. ";Z$: GOTO 200
800  REM  RUTINA QUE DESPEJA EL AREA DE TRABAJO
810  GOSUB 1820: GOTO 200
820  GOSUB 3000: VTAB 12: HTAB 5: PRINT "****":: FLASH : PRINT "MOMENTO DESPE
DO AREA":: NORMAL : PRINT "****"
830  VTAB 14: HTAB 17: FLASH : PRINT "DE TRABAJO": NORMAL
840  CALL 13191
850  RETURN
2000  REM  RUTINA DE SALIDA
2010  GOSUB 3000: VTAB 5: HTAB 7: PRINT "------   SALIDA   ------"
2020  VTAB 7: HTAB 5: INVERSE : PRINT "N O T A :": NORMAL
2030  VTAB 10: HTAB 5: SPEED= 100: PRINT "NO SE OLVIDE APAGAR LA MICROCOM-"
2040  VTAB 12: HTAB 5: PRINT "PUTADORA Y EL PROGRAMADOR PPI.": SPEED= 255: VTA
7: HTAB 15: PRINT "..G R A C I A S!"
2050  END
2200  REM  RUTINA CHECA SI HAY PROG.EN A.DE TRAB.
2210 X1 =  PEEK (20480)
2220  IF X1 <  > 0 THEN 2240
2230 SW = 0: GOSUB 3000: VTAB 11: HTAB 14: FLASH : PRINT "NO EXISTE PROG.": NO
L : VTAB 13: HTAB 11: PRINT "EN EL AREA DE TRABAJO": GOTO 2250
2240 SW = 1
2250  RETURN
2500  REM     RUTINA CALCULA CAP.DE EPROM
2510 X2 =  PEEK (21505):X3 =  PEEK (22529):X4 =  PEEK (24577):X5 =  PEEK (2662
N1 = 0
2520  IF X2 = 0 THEN N1 = 1: GOTO 2570
2530  IF X3 = 0 THEN N1 = 2: GOTO 2570
2540  IF X4 = 0 THEN N1 = 4: GOTO 2570
2550  IF X5 = 0 THEN N1 = 8: GOTO 2570
2560 N1 = 10
2570  RETURN
2700  REM  RUTINA LETRERO TIPO EPROM
2710  VTAB 8: HTAB 5: PRINT "1.- 2704   PV.21V": HTAB 5: PRINT "2.- 2708   PV.
."
2720  HTAB 5: PRINT "3.- 2758   PV.25V.": GOSUB 2740: VTAB 14: HTAB 21: PRINT
00"
2730  VTAB 16: HTAB 26: PRINT "(PM-1) 1,2,3": RETURN
2740  VTAB 13: HTAB 5: PRINT "EMPIEZA EPROM..0000": HTAB 5: PRINT "TERMINA EPR
."
2750  VTAB 16: HTAB 5: INVERSE : PRINT "MODULO PERS.A USAR: ": NORMAL : RETURN
2800  VTAB 8: HTAB 5: PRINT "1.- 2716   PV.25V.": HTAB 5: PRINT "2.- 2716-1 PV
V."
2810  GOSUB 2740: VTAB 14: HTAB 21: PRINT "0800": VTAB 16: HTAB 26: PRINT "(PM
```

```
 1,2": RETURN
2850  VTAB 8: HTAB 5: PRINT "1.- 2532   PV.25V.": HTAB 5: PRINT "2.- 2732   PV
5V.": HTAB 5: PRINT "3.- 2732-A PV.21V."
2860  GOSUB 2740: VTAB 14: HTAB 21: PRINT "OFFF": VTAB 16: HTAB 26: PRINT "(PM
 3,4": HTAB 26: PRINT "(PM-3) 1": RETURN
2900  VTAB 8: HTAB 5: PRINT "1.- 2764   PV.21V.": HTAB 5: PRINT "2.- 2764-3 PV
5V." HTAB 5: PRINT "3.- 2764-Q PV.25V."
2910  GOSUB 2740: VTAB 14: HTAB 21: PRINT "2000": VTAB 16: HTAB 26: PRINT "(PM
 1,2,3.": RETURN
3000  REM  RUTINA DE LETRERO
3010  HOME : VTAB 3: HTAB 4: PRINT "****":: INVERSE : PRINT "  PROGRAMADOR DE
COMPS  ":: NORMAL : PRINT "****": RETURN
```

```
SOURCE FILE: MEJOR
------ NEXT OBJECT FILE NAME IS MEJOR-1
3200:              1          ORG  $3200
3200:20 89 32      2          JSR  INICIO     ; INICIA PROGRAMA
3203:AE 02 71      3 EMPZA    LDX  $7102      :ACTIVA MODO PROG
3206:A5 82         4          LDA  $82
3208:9D 9F C0      5          STA  $C09F,X
320B:20 A4 32      6          JSR  DIREPR     :DIRECCIONAMIENTO DE EPROM
320E:A5 0C         7          LDA  $0C        :VOLTAJE DE PROG.ON
3210:9D 9F C0      8          STA  $C09F,X
3213:A5 09         9          LDA  $09
3215:9D 9F C0     10          STA  $C09F,X
3218:A0 00        11          LDY  #$00       :DIRECCIONAMIENTO DE MEMORIA
321A:B1 FE        12          LDA  ($FE),Y
321C:9D 9D C0     13          STA  $C09D,X
321F:A5 0B        14          LDA  $0B        :VOLTAJE DE PROG.OFF
3221:9D 9F C0     15          STA  $C09F,X
3224:A5 0A        16          LDA  $0A
3226:9D 9F C0     17          STA  $C09F,X
3229:A5 08        18          LDA  $08
322B:9D 9F C0     19          STA  $C09F,X
322E:A5 0D        20          LDA  $0D
3230:9D 9F C0     21          STA  $C09F,X
3233:20 12 33     22          JSR  INCRE      :INCREMENTO DE DIRECCIONES
3236:A5 FF        23          LDA  $FF
3238:CD 01 71     24          CMP  $7101
323B:F0 03        25          BEQ  FINPRO
323D:4C 03 32     26          JMP  EMPZA
3240:60           27 FINPRO   RTS
3241:20 89 32     28          JSR  INICIO     :PROG.LECTURA DE EPROM
3244:AE 02 71     29 EMLEC    LDX  $7102      :ACTIVA LEC.
3247:A5 80        30          LDA  $80
3249:9D 9F C0     31          STA  $C09F,X
324C:20 A4 32     32          JSR  DIREPR     :DIRECCIONAMIENTO DE EPROM
324F:BD 9F C0     33          LDA  $C09F,X
3252:A0 00        34          LDY  #$00
3254:91 FE        35          STA  ($FE),Y
3256:20 12 33     36          JSR  INCRE      :INCREMENTOS
3259:A5 FF        37          LDA  $FF
325B:CD 01 71     38          CMP  $7101
325E:F0 03        39          BEQ  FINLEC
3260:4C 44 32     40          JMP  EMLEC
3263:60           41 FINLEC   RTS
3264:20 89 32     42          JSR  INICIO     :PROG.DE COMPARA.
3267:AE 02 71     43 EMCOM    LDX  $7102      :ACTIVA MODO LEC.
326A:A5 80        44          LDA  $80
326C:9D 9F C0     45          STA  $C09F,X
326F:20 A4 32     46          JSR  DIREPR     :DIREC.EPROM
3272:BD 9F C0     47          LDA  $C09F,X    :LEE DATO
3275:A0 00        48          LDY  #$00
3277:D1 FE        49          CMP  ($FE),Y
3279:D0 0D        50          BNE  FINCOM
327B:20 12 33     51          JSR  INCRE      :INCREMENTOS
327E:A5 FF        52          LDA  $FF
3280:CD 01 71     53          CMP  $7101
```

```
3283:F0 03      54          BEQ  FINCOM
3285:4C 67 32   55          JMP  EMCOM
3288:60         56 FINCOM   RTS
3289:A9 00      57 INICIO   LDA  #$00      ;INICIO DE DIRECCIONES AUXILIARES EN
328B:8D 03 71   58          STA  $7103
328E:8D 04 71   59          STA  $7104
3291:8D 05 71   60          STA  $7105
3294:8D 06 71   61          STA  $7106
3297:8D 07 71   62          STA  $7107
329A:85 FE      63          STA  $FE
329C:8D 0A 71   64          STA  $710A
329F:A9 50      65          LDA  #$50
32A1:85 FF      66          STA  $FF
32A3:60         67          RTS
32A4:AD 0A 71   68 DIREPR   LDA  $710A      ;DIRECCIONAMIENTO DE EPROM
32A7:C9 00      69          CMP  #$00
32A9:F0 08      70          BEQ  PG8K
32AB:A5 0F      71          LDA  $0F        ;SW PROGRAMACION = A 8K
32AD:9D 9F C0   72          STA  $C09F,X
32B0:4C B8 32   73          JMP  CONT
32B3:A5 0E      74 PG8K     LDA  $0E        ;SW PROGRAMACION < A 8K
32B5:9D 9F C0   75          STA  $C09F,X
32B8:AC 07 71   76 CONT     LDY  $7107      ;DIRECCIONAMIENTO DE 2 BYTES BAJOS
32BB:B9 00 00   77          LDA  $00,Y
32BE:9D 9C C0   78          STA  $C09C,X
32C1:AD 03 71   79          LDA  $7103      ;DIRECCIONAMIENTO DE BIT'S ALTOS
32C4:C9 00      80          CMP  #$00
32C6:F0 08      81          BEQ  RESET1
32C8:A5 01      82          LDA  $01        ;SET 1* BIT ALTO (A8)
32CA:9D 9F C0   83          STA  $C09F,X
32CD:4C D5 32   84          JMP  CONT1
32D0:A5 00      85 RESET1   LDA  $00        ;RESET 1* BIT ALTO
32D2:9D 9F C0   86          STA  $C09F,X
32D5:AD 04 71   87 CONT1    LDA  $7104      ;2BIT ALTO
32D8:C9 00      88          CMP  #$00
32DA:F0 08      89          BEQ  RESET2
32DC:A5 03      90          LDA  $03        ;SET 2* BIT ALTO (A9)
32DE:9D 9F C0   91          STA  $C09F,X
32E1:4C E9 32   92          JMP  CONT2
32E4:A5 02      93 RESET2   LDA  $02        ;RESET 2* BIT ALTO
32E6:9D 9F C0   94          STA  $C09F,X
32E9:AD 05 71   95 CONT2    LDA  $7105      ;3 BIT.ALT
32EC:C9 00      96          CMP  #$00
32EE:F0 08      97          BEQ  RESET3
32F0:A5 05      98          LDA  $05        ;SET 3* BIT ALTO (A10)
32F2:9D 9F C0   99          STA  $C09F,X
32F5:4C FD 32  100          JMP  CONT3
32F8:A5 04     101 RESET3   LDA  $04        ;RESET 3* BIT ALTO
32FA:9D 9F C0  102          STA  $C09F,X
32FD:AD 06 71  103 CONT3    LDA  $7106      ;4BIT ALTO
3300:C9 00     104          CMP  #$00
3302:F0 08     105          BEQ  RESET4
3304:A5 07     106          LDA  $07        ;SET 4* BIT ALTO (A11)
```

```
3306:9D 9F C0   107          STA  $C09F,X
3309:4C 11 33   108          JMP  FINDI
330C:A5 06      109 RESET4   LDA  $06       ;RESET 4* BIT ALTO
330E:9D 9F C0   110          STA  $C09F,X
3311:60         111 FINDI    RTS
3312:A4 FE      112 INCRE    LDY  $FE       ;INCREMENTO DE DIREC.DE MEM.
3314:C0 FF      113          CPY  #$FF
3316:F0 0F      114          BEQ  INMEM
3318:E6 FE      115          INC  $FE       ;INCREMENTO DE 2 BYTES BAJOS
331A:AC 07 71   116 CONT4    LDY  $7107     ;INCREMENTO DE DIREC. DE EPROM
331D:C0 FF      117          CPY  #$FF
331F:F0 16      118          BEQ  INEPR
3321:EE 07 71   119          INC  $7107
3324:4C 87 33   120          JMP  FININ
3327:E6 FF      121 INMEM    INC  $FF       ;INCREMENTA 2 BYTES ALTOS DE MEM.
3329:A4 FF      122          LDY  $FF
332B:CC 07 71   123          CPY  $7107
332E:F0 57      124          BEQ  FININ
3330:A9 00      125          LDA  #$00
3332:85 FE      126          STA  $FE
3334:4C 1A 33   127          JMP  CONT4
3337:EE 03 71   128 INEPR    INC  $7103     ;ENCREMENTA 1* BIT ALTO (A8)
333A:AC 03 71   129          LDY  $7103
333D:C0 02      130          CPY  #$02
333F:F0 03      131          BEQ  CONT5
3341:4C 82 33   132          JMP  SIGA
3344:A9 00      133 CONT5    LDA  #$00
3346:8D 03 71   134          STA  $7103
3349:EE 04 71   135          INC  $7104     ;INCREMENTA 2* BIT ALTO (A9)
334C:AC 04 71   136          LDY  $7104
334F:C0 02      137          CPY  #$02
3351:F0 03      138          BEQ  CONT6
3353:4C 82 33   139          JMP  SIGA
3356:A9 00      140 CONT6    LDA  #$00
3358:8D 04 71   141          STA  $7104
335B:EE 05 71   142          INC  $7105     ;INCREMENTO 3* BIT ALTO (A10)
335E:AC 05 71   143          LDY  $7105
3361:C0 02      144          CPY  #$02
3363:F0 03      145          BEQ  CONT7
3365:4C 82 33   146          JMP  SIGA
3368:A9 00      147 CONT7    LDA  #$00
336A:8D 05 71   148          STA  $7105
336D:EE 06 71   149          INC  $7106     ;INCREMENTO 4* BIT ALTO (A11)
3370:AC 06 71   150          LDY  $7106
3373:C0 02      151          CPY  #$02
3375:F0 03      152          BEQ  CONT8
3377:4C 82 33   153          JMP  SIGA
337A:A9 00      154 CONT8    LDA  #$00
337C:8D 06 71   155          STA  $7106
337F:EE 0A 71   156          INC  $710A     ;INCREMENTO PARA 8K DE MEM.
3382:A9 00      157 SIGA     LDA  #$00
3384:8D 07 71   158          STA  $7107
3387:60         159 FININ    RTS
3388:A9 50      160          LDA  #$50      ;BORRA AREA DE TRAB.
```

```
338A:85 FF      161           STA  $FF
338C:A9 00      162           LDA  #$00
338E:85 FE      163           STA  $FE
3390:A0 00      164 VUELVE    LDY  #$00
3392:91 FE      165           STA  ($FE),Y
3394:A4 FE      166           LDY  $FE
3396:C0 FF      167           CPY  #$FF
3398:F0 05      168           BEQ  MEMIN
339A:E6 FE      169           INC  $FE
339C:4C 90 33   170           JMP  VUELVE
339F:A2 70      171 MEMIN     LDX  #$70
33A1:E4 FF      172           CPX  $FF
33A3:F0 09      173           BEQ  FINBO
33A5:E6 FF      174           INC  $FF
33A7:A9 00      175           LDA  #$00
33A9:85 FE      176           STA  $FE
33AB:4C 90 33   177           JMP  VUELVE
33AE:60         178 FINBO     RTS

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

A P E N D I C E (C)

- SET DE INSTRUCCIONES DE 6502

El microprocesador 6502 cuenta con 56 instrucciones para el procesamiento de datos.

INSTRUCCIONES ARITMETICAS:

MNEMONICO	OPERACION	FUNCION
ADC	A+M+C->A	Suma la memoria al acumulador con el carry.
SBC	A-M-C->A	Resta el contenido de memoria del contenido del acumulador.

INSTRUCCIONES LOGICAS

AND	A*M->A	Efectua el AND con el contenido del acumulador y la localidad de mem.
EOR	A+M->A	Efectua el EXOR al contenido del acumulador y la localidad de mem.

ORA A+M->A Ejecuta OR al contenido de un acumulador y la localidad de mem.

INSTRUCCIONES DE BRINCOS Y RAMIFICACIONES

BCC Brinco si C=0 brinca si el Carry = 0.

BCS Brinco si C-1 Brinca si el Carry= 1.

BEQ Brinco si Z=1 Brinco si el resultado=0.

BNE Brinco si Z=0 Brinco si el resultado es <> 0.

BMI Brinco si N=1 Brinco si el resultado es negativo

BPL Brinco si N=0 Brinco si el resultado no es negativo

BRK ----- Brinco a la fuerza.

BVC Brinco si V=0 Brinco si no hay sobreflujo.

BVS Brinco si V=1 Brinco si hay sobre flujo.

JMP ----- Salto a una nueva localización.

NOP ----- No operación.

RTS ----- Retorno de un prog. de subrutina.

RTI ----- Retorno de una interrupción.

INSTRUCCIONES PARA CORRIMIENTO Y ROTACION DE DATOS

ASL Corrimiento a la izquierda el contenido de una localidad de mem.

LSR Corrrimiento a la derecha el contenido de una localidad de mem.

ROL Rotacion a la izquierda de contenido de una localidad de mem.

ROR Rotacion a la derecha el contenido de una localidad de mem.

INSTRUCCIONES DE PRUEVA DE DATOS:

BIT A.M Prueba de bit del contenido de un acumulador que se efectua un AND con el valor de una localidad de mem.

CMP A-M Compara el valor del acumulador con el valor de una localidad de mem.

INTRUCCIONES PARA REGISTRO CODIGO DE CONDICION:

CLC	0->C	Borra el contenido del carry.
CLD	0->M.	Borra el modo bit decimal.
CLI	0->I	Borra el contenido del bit(I) del reg. codigo de condicion.
CLV	0->V	Borra el contenido del bit(V) del reg. codigo de condicion.
CEC	1->C	Establece el bit (C) del reg. codigo de condicion.
SED	1->M.	Establece el bit (m.) del reg. codigo de condicion.
SEI	1->I	establece el bit (I) del reg. codigo de condicion.

INSTRUCCIONES DEL REGISTRO INDICE Y STACK POINTER:

CPX	A-X	Compara contenido del indice X con el contenido de mem.
CPY	A-Y	Compara contenido del indice Y con el contenido de mem.
DEX	X-1->X	Decrementa el contenido del reg. indice (X) en uno.
DEY	Y-1->Y	Decrementa el contenido del reg. indice (Y) en uno.
INX	X+1->X	Incrementa el contenido de X.
INY	Y+1->Y	Incrementa el contenido de Y.
LDX	M->X	Carga valor de mem. a X.

LDY	M->Y	Carga valor de mem. a Y.
STX	X->M	Carga valor de X a mem.
STY	Y->M	Carga valor de Y a mem.

INSTRUCCIONES PARA ALTERAR DATOS:

DEC	M-1->M	Decrementa valor de mem.
INC	M+1->M	Incrementa valor de mem.

INSTRUCCIONES DE MANEJO DE DATOS

LDA	M->A	Carga valor de mem.a A.
PHA	A->Msp	Carga valor de A a mem.
PHP	P->Msp	Carga valor de P a mem.
PLA	Msp->A	Carga valor de mem.con Sp a A.
PLP	Msp->P	Carga valor de mem.con Sp a P.
STA	A->M	Carga valor de A a Mem.
TAX	A->X	Transfiere valor de A a X.
TAY	A->Y	Transfiere valor de A a Y.
TSX	SP->X	Transfiere valor de Sp a X.
TXA	X->A	Transfiere valor de X a A.
TXS	X->SP	Transfiere valor de X a Sp.
TYA	Y->A	Transfiere valor de Y a A.

A P E N D I C E (D)

CARACTERISTICAS DEL FABRICANTE:

EPROM 2716 :

- Tiempo de acceso :
  . 2716 350 ns. Max
  . 2716-1 350 ns. Max
  . 2716-2 390 ns. Max
  . 2716 450 ns. Max
  . 2716-5 450 ns. Max
- Pin compatible to intel 2732a
- Programacion simple requerida
- solo localizacion de prog.
- Programacion con 50 ns.de pulso
- Entrada y salida compatible con TTL durante lectura y prog.
- Solo + 5v. de potencia
- baja potencia de disipasion
  - potencia activar 525 Max.
  - potencia de resistencia 132 mW. Max.

EPROM 2732-A :

- 200 ns. Tiempo de acceso
- dos lineas de control
- industria JEDEC
- baja resistencia de 35 mA Max.

- Pin compatible con 2764 . +- 10% Vcc. de tolerancia aprovechable

EPORM 27c4 :

- 200 ns. Tiempo de acceso . pin compatible con el 3732-A

- Dos lineas de control . Industria JEDEC

. +- 10% Vcc. tolerancia aprovechable

TEMPERATURA EN LAS OPERACIONES:

- 0° C- 70°C

En todos los Eprom's.

A P E N D I C E (E)

APLICACIONES DE EPORM S

Los Eprom's como ya dijimos anteriormente tienen una gran variedad de aplicaciones. Mensionaremos algunas de ellas:

EN COMPUTACION:

- control de microprocesadores
- control de circuitos
- compiladores
- traductores
- funciones y tablas

EN OTROS ARTEFACTOS:

- televisores
- radios
- video caseteras
- etc.

En estos artefactos la funcion del eprom mas que todo es para el control de ellos en la forma de operacion.

También mediante estos Chip's la minimización de circuitos en los artefactos.

B I B L I O G R A F I A

Poole L..McNiff M. APPLE II USER'S GUIDE

Mc.Gran Hill

U.S.A 1982

Mano Morris M. COMPUTER SISTEM ARCHITECTURE.

Prentice Hall Inc.

U.S.A 1978

Howord H. Gerrish FUNDAMENTOS DE ELECTRONICA

Limusa.

Mexico 1981

Mingot Tomas de Galeana LAROUSE DE CIENCIAS TECNICAS

Larouse S.A.

Mexico 1983.

NAFINSA FONEP LAS MICROCOMPUTADORAS

Nafinsa.

Mexico 1983.

C.E.C. U.N.L. INTRODUCCION MICROCPROCESADORES

U.N.L. Monterrey Mexico.

(Septiempbre, 1983)pp.5-255

De Jone Marvin "HMS3264 EPROM PROGRAMER"
En Byte Publication U.S.A.
(Junio, 1983),pp.288-294.

Doolittle J. Tkalcevic "PROGRAMMING EPROM'S WITH
A SMALL COMPUTER"
En Popular Electronics U.S.A.
(Julio,1982),pp.61-68.
(Agosto, 1982),pp.67-70.