

300617

16  
2-ey



# UNIVERSIDAD LA SALLE

Escuela de Ingeniería Mecánica Eléctrica  
Incorporada a la U.N.A.M.

## “DISEÑO DE UN PROGRAMADOR DE EPROMS COMPUTARIZADO”

### Tesis Profesional

Que para obtener el título de:

INGENIERO MECANICO ELECTRICISTA

P r e s e n t a :

JOSE GRANDA SANCHEZ

México, D. F.

TESIS CON  
FALLA DE ORIGEN

1986



Universidad Nacional  
Autónoma de México



## **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.

# I N D I C E

## CAPITULO I      INTRODUCCION

## CAPITULO II      DESCRIPCION GENERAL DE LAS MEMORIAS "EPROM"

- a) Descripción general
- b) Programación de las memorias
- c) Descripción general del programador

## CAPITULO III      CIRCUITOS DEL PROGRAMADOR

- a) Estructuración del diagrama de bloques
- b) Circuito de reloj
- c) Circuito reiniciador
- d) Diseño de la fuente de voltaje
- e) Memorias
- f) Tablero visualizador y teclado
- g) Diseño de la base conectora de programación de la memoria "EPROM"

## **CAPITULO IV**

### **ELABORACION DE LAS SUBRRUTINAS DE PROGRAMACION**

- a) Descripción general de las subrrutinas
- b) Subrrutina de chequeo o borrado correcto
- c) Subrrutina de programación manual
- d) Subrrutina de programación automática
- e) Subrrutina de copiado de memorias "EPROM"

## CAPITULO I

### INTRODUCCION

Las memorias en una computadora son una de las tres partes mas importantes y básicas , es por esto que el desarrollo de estas se ha llevado a cabo con gran dedicación hasta obtener las ya famosas memorias "EPROM" , las cuales por su facil uso , se han convertido en las memorias - mas deseadas del mercado.

Las memorias solo leibles programables y borrables "EPROM" , son borrables por medio de irradiaciones de rayos ultravioletas atraves de la ventana superior.

Estas memorias han sido largamente usadas en el desarrollo de sistemas basados en micro-computadoras y recientemente sus aplicaciones se han extendido a campos donde corregir erro-

res y la facil modificación de los programas es necesario .

La memoria EPROM goza de una gran integridad es por esto que una memoria grabada , puede conservar su información arriba de diez años .

La misma ventana de cuarzo que deja entrar rayos ultravioletas para el proceso de borrado - permite también al usuario , observar el chip de silicio , esta característica lo diferencia de - los demas circuitos existentes , pues su ventana sobresale del chip lo que la hace facilmente identificable , pero en realidad el gran atractivo del que gozan estas memorias es debido a la combina-ción de sus características , como son su amplio - rango de programación , así como su alta densidad y bajo poder de consumo .

Como podemos apreciar , utilizar una memoria EPROM , se reduce a la simple inserción del componente en una base estrategicamente interconectada con el microcomputador , con la cual , atraves de

circuitos podremos correr los programas pregrabados en estas memorias.

la tarea difícil se encuentra realmente en la forma de insertar estos programas dentro de la memoria EPROM , pues programas largos , representan tareas laboriosas para el programador , pues , el lenguaje que deben de utilizar es assembler, y en una computadora de ocho bits , tenemos que introducir instrucción por instrucción , por esta razon , se debe implementar un instrumento que facilite la tarea de copiar una memoria , y de introducir programas a una memoria EPROM .

Como podemos apreciar el insertar un programa en una memoria EPROM es una tarea rutinaria , por lo que debemos de pensar en el aparato rutinario de mas alta velocidad , este es ; el microcomputador - usamos el microcomputador pues un espacio reducido nos facilita muchas otras tareas , este debera de poseer un intercomunicador con una base donde se conectará la memoria EPROM , para efectuar la lectura y programación de la memoria EPROM .

## CAPITULO II

### DESCRIPCION GENERAL DE LAS MEMORIAS EPROM

#### a) Descripción general.

Para tener una mejor idea de como debe de ser un programador examinemos las memorias EPROM en su mecanismo de operación interno , es decir, a nivel electrón.

Los transistores que se sitúan en su interior son de tecnología mos , excepto por la adición de una compuerta flotante , enterrada en el aislante entre el sustrato y la puerta ordinaria de electrodo como se aprecia en la figura 1 .

Por lo tanto , en un EPROM , el voltaje de la puerta seleccionadora debe ser parejo en serie con el de la puerta flotante.

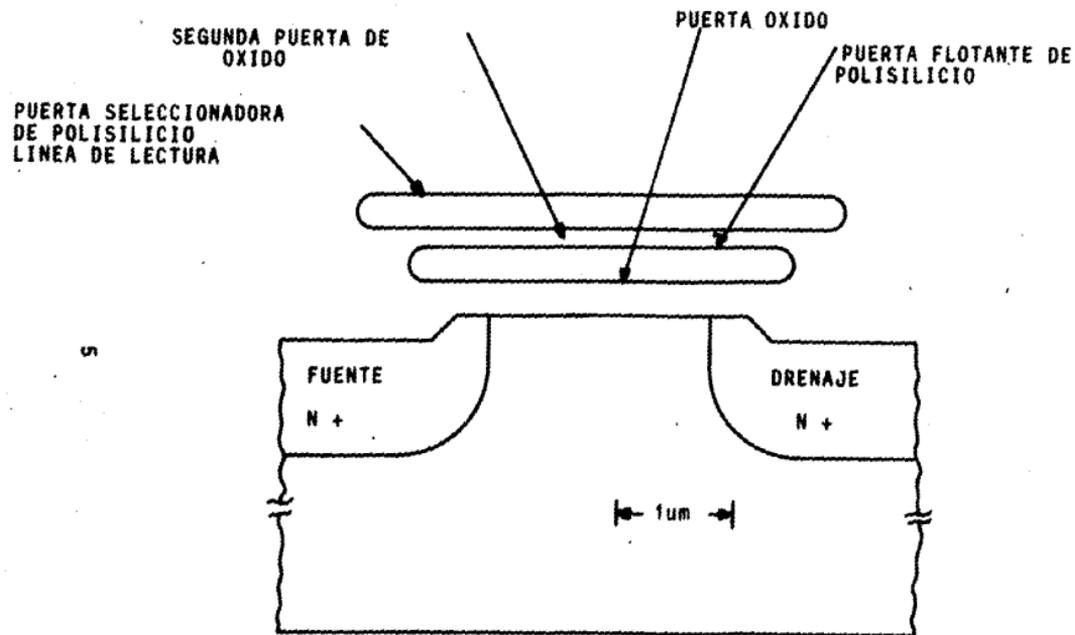
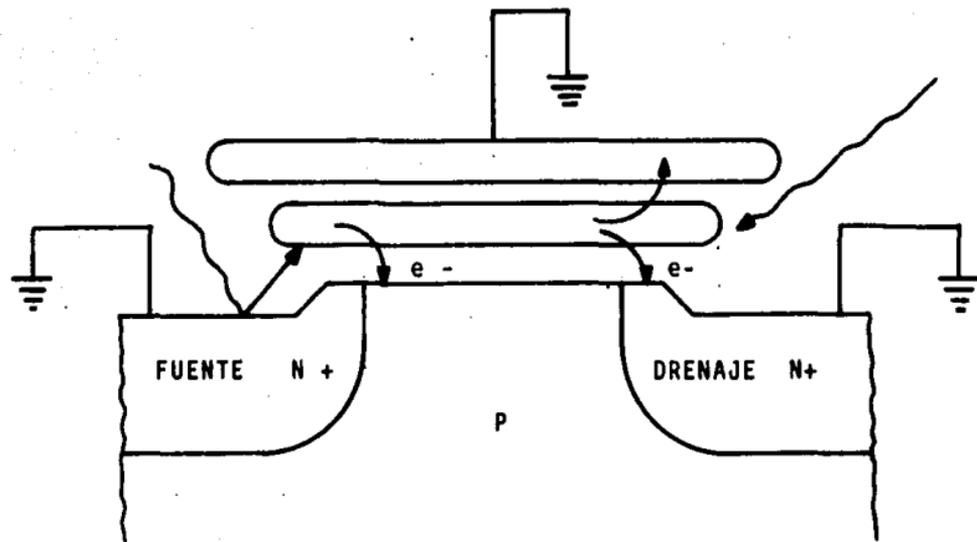


FIGURA 1 . UNA PUERTA FLOTANTE PARA ALMACENAR Y UNA PUERTA SELECCIONADORA DIRECTAMENTE EN UN CANAL MOS FET .

La figura 1 nos muestra también porque pueden ser ejecutadas altas densidades con las celdas pues el almacén de la puerta flotante y la puerta seleccionadora , ambos se encuentran situados inmediatamente arriba del canal de transistores , mas altas densidades se obtienen por la combinación de auto alineación de las dos compuertas , la fuente y la región de drenaje .

Esta disposición vertical que guardan los componentes , admite tener funciones de almacenar y leer dentro del transistor de efecto de campo , situado en la parte inferior de la disposición vertical .

La celda es programada por medio de la inyección a través de la puerta flotante de electrones - calientes , en la figura 2 tenemos graficamente la situación de grabado y borrado.



**FIGURA 2 .** Durante la programación observamos , como los electrones son inyectados dentro de la puerta flotante de silicio .

En la figura 3 podemos apreciar el borrado atraves de la fotoemisión interna sobre la compuerta flotante a la puerta superior y a su vez al sustrato .

La luz ultravioleta da a los electrones suficiente energía como para sobremontar la energía de la barrera entre la puerta flotante y el aislador que lo rodea .

Las cargas de las celdas del almacen de puerta flotante cambia de un valor de umbral de la puerta seleccionadora , por medio de una sumatoria a un valor que se determina por medio de la ecuación :

$$V_t = - Q_{fg} / C$$

Donde C es la capacitancia existente entre la puerta flotante y la puerta seleccionadora ,  $Q_{fg}$  es el cambio en carga en la puerta flotante y  $V_t$  es el incremento que sufre el voltaje de umbral .

La corriente de drenaje frente a el voltaje de

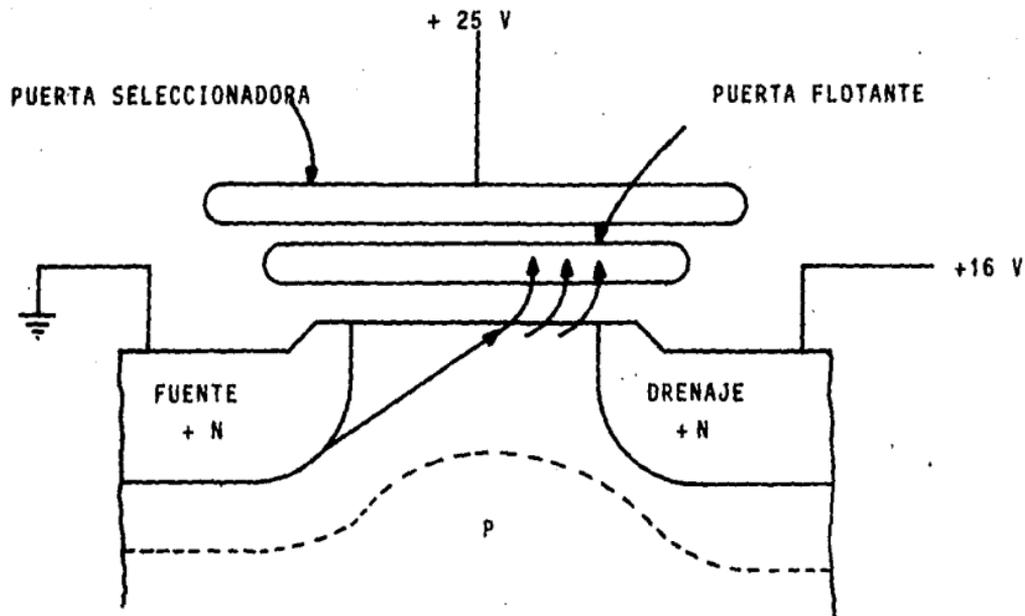


FIGURA 3 . Durante el proceso de borrado observamos , los mismos electrones recibiendo suficiente energía como para penetrar la capa de silicio que los retiene .

la puerta seleccionadora transfiere características para la programación y el borrado , como se aprecia en la figura 4 .

El voltaje de la puerta seleccionadora durante la lectura descansa entre las dos curvas de la figura 4 , como resultado de esto hay una corriente de drenaje que refleja el estado de la celda poseemos dentro de esto dos estados ; el estado de no conducción reflejando un cero y el estado de conducción o borrado donde se refleja un uno .

Cuando una celda EPROM es programada , la carga negativa en la puerta flotante causa que la fuente de voltaje sea negativa también , esto cambia la celda en posición apagada , aún con un voltaje de lectura aplicado positivo a la puerta seleccionadora directamente .

Si la puerta flotante no está amarrada a una fuente de voltaje , su voltaje es determinado por la carga y su capacitancia a la puerta seleccionado

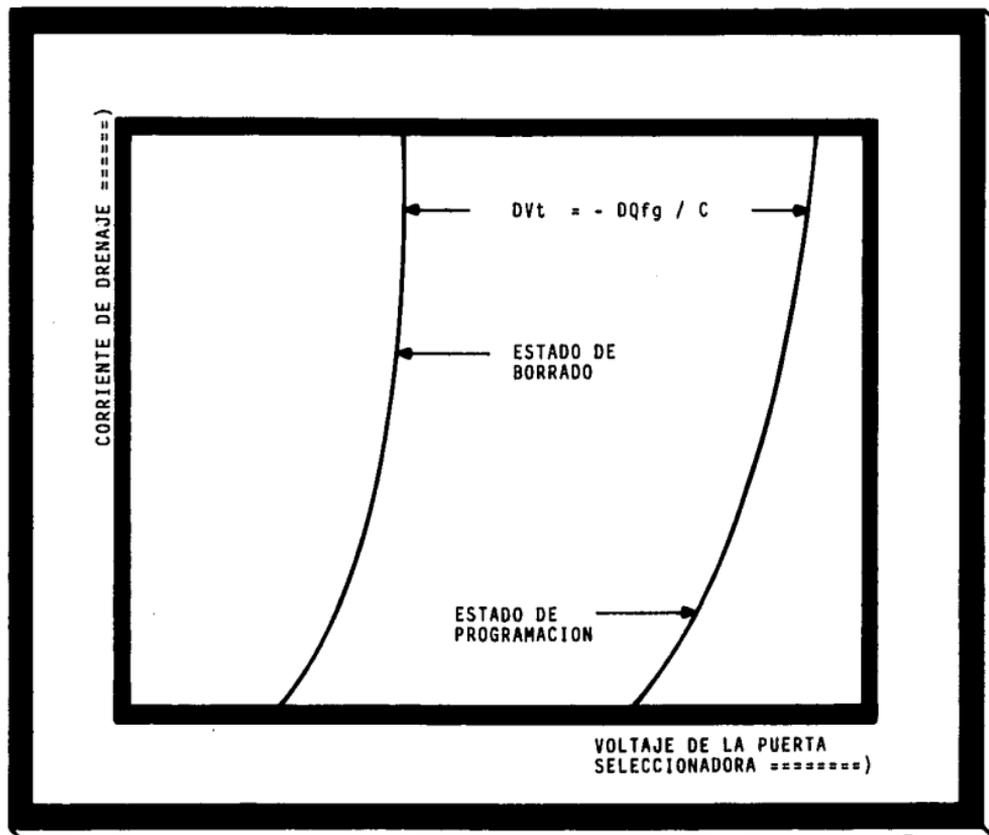


FIGURA 4

ra , el drenaje , el canal y la fuente .

La diferencia entre el voltaje de la puerta flotante y todas las otras areas de la celda , puede ser usado para determinar los campos eléctricos en las diferentes regiones del óxido .

Sin embargo el voltaje aplicado al chip durante la programación es alto , llegando a ser 25 volts a pesar de esto los campos a través de la compuerta óxido son relativamente pequeños , cerca de 2 megavolts en el acceso de programación y menos que 1.5 mega - volts en la lectura .

Durante todas las demas operaciones , aún cuando la compuerta flotante está a toda carga , este campo es menor o igual a 0.7 MV / cm . Esto es típico cuando se opera en tecnología mos.

## b) PROGRAMACION DE LAS MEMORIAS EPROM .

Los electrones calientes toman su energía del voltaje aplicado al drenaje de la celda del EPROM , estos electrones son acelerados a lo largo del canal dentro aún , de los grandes campos que rodean la región de agotamiento del drenaje .

Mientras atraviesan el canal , los electrones van entrando en una región donde el campo eléctrico en el sustrato es cercano a  $1 \times 10^5$  volts / cm ó más grande , en este momento el rango de energía ganada del campo eléctrico no puede ser más tiempo descrita por la temperatura del silicio de aquí el término de electrones calientes .

Una vez que estos electrones ganan suficiente energía ellos pueden sobremontar la barrera , que tiene dimensiones cercanas a los 3.2 electrovolts existentes entre el aislante de óxido de silicio y el sustrato de silicio .

Por causa de la pérdida de energía a la emisión de \*PHONON\* (término conocido como paquete de energía) la temperatura se eleva considerablemente , es por es to que es en este momento cuando se puede obtener - más facil electrones calientes a temperaturas de operación menores .

En complemento con los paquetes de energía emitidos . electrones calientes con energía cercana a los 1.8 electrovolts , deben ceder algo de su energía en otra forma , lo hacen a través del agujero del elec tron , creando un impacto ionizante .

Este fenómeno es observado en los transistores ordinarios , a causa del acceso de la corriente del sustrato a altos voltajes de drenaje .

Sin embargo , en el caso del EPROM , multipli caciones de corrientes significantes producen unas corrientes de sustrato , aún antes de que un voltaje de drenaje sea alcanzado para producir la inyección de los electrones calientes dentro del óxido .

Con los voltajes positivos en el drenaje y el canal , los electrones inyectados dentro del óxido de un canal tipo n , de una memoria EPROM , vuelven al sustrato a menos que un voltaje positivo alto se aplique en la puerta seleccionadora para empujar los electrones por la puerta flotante .

No solo la compuerta flotante debe ser positiva con respecto a la fuente sino que también debe ser positivo respecto al punto del canal donde la inyección de electrones calientes se efectúa .

Cerca de que empiece el proceso de inyección de electrones calientes , la capa de inversión se extiende a casi todo el camino del drenaje , en este momento el campo de óxido es atractivo excepto por una pequeña porción muy cercana al drenaje como se aprecia en la figura 5 .

La corriente empieza a fluir através del óxido al punto donde los electrones están lo más caliente y donde el campo óxido es más favorable .

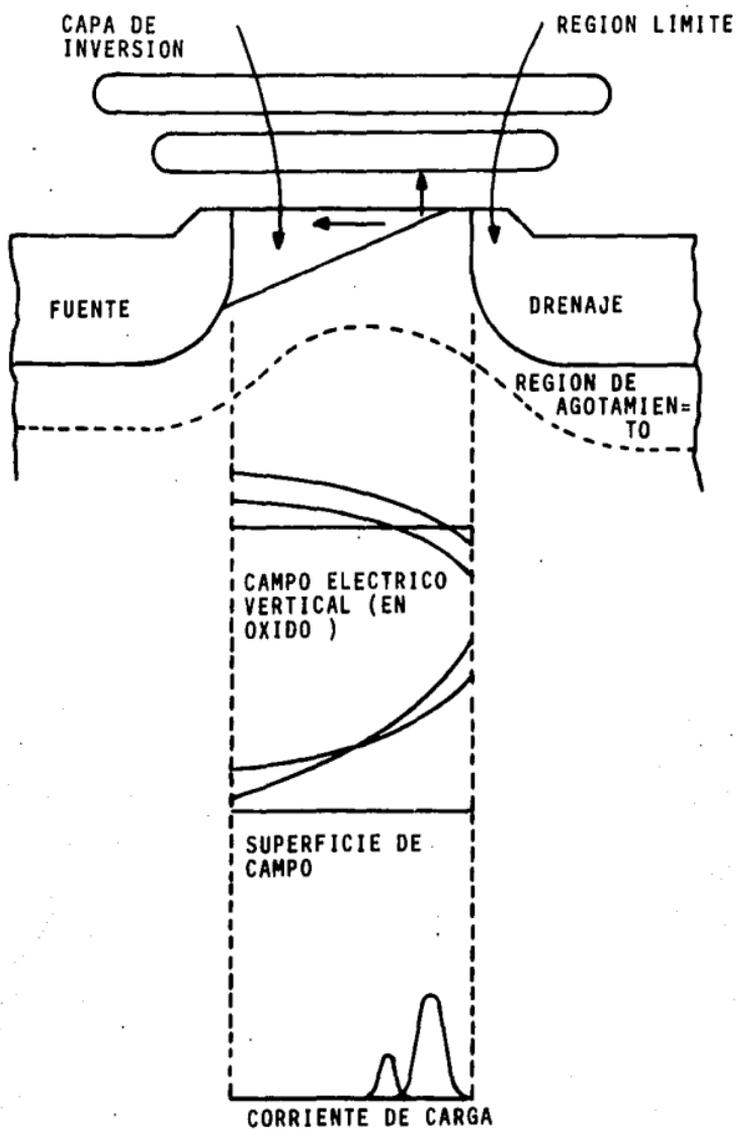


FIGURA 5 . INYECCION CON UNA COMPUERTA POSITIVA FLOTANTE.

Como la puerta flotante se carga , la caída de voltaje de la fuente a la puerta flotante , por la región límite de drenaje se retorna estos a la fuente .

En este momento se acerca la superficie del campo a la intensidad del drenaje , y son producidos mas electrones calientes en el sustrato .

No obstante , como se aprecia en la figura 5 en la región donde , los electrones están lo más caliente , el campo de óxido es por lo menos favorable para la inyección y así la corriente de inyección de electrones empieza a cesar .

De este modo , afortunadamente , el proceso de inyección de electrones es autolimitado . La carga de la compuerta flotante reduce el número de electrones que pueden ser acelerados en la región de alto campo .

Como la puerta flotante se encuentra a toda carga , la corriente del óxido se reduce a casi

cero porque el campo óxido es ahora repulsivo a los electrones inyectados dentro de la región de alto campo .

Desde la reducción del campo eléctrico entre la puerta flotante y el canal , se reduce la corriente del óxido . El voltaje de umbral saturado , sujeta la puerta flotante marcando el voltaje de la puerta seleccionadora durante la programación , en una base de volt por volt .

La puerta flotante se carga al mismo valor relativo de la fuente y el canal.

Un aumento en el voltaje de la puerta seleccionadora , durante la programación , meramente aumenta la carga en la puerta flotante , con subsecuente necesidad de abortar la inyección de electrones calientes .

El voltaje de drenaje en general no tiene mayor efecto en el voltaje final de programación de

la puerta flotante .

No obstante , esto afecta la velocidad a la cual el componente es programado .

### c) DESCRIPCION GENERAL DEL PROGRAMADOR

Como podemos apreciar en la descripción anterior , el programador debe ser exacto y preciso en sus voltajes y tiempos , pues esto es muy importante para que el proceso de programación sea llevado correctamente .

Además necesitamos un almacén donde situar el contenido de inyección a la memoria EPROM , es decir , los datos del programa que queremos depositar en nuestra memoria EPROM , por lo tanto situamos una memoria RAM , en donde guardaremos por medio de direccionamiento todos los componentes de nuestro programa a fijar en nuestra memoria EPROM.

Se necesita además un circuito mediante el cual nuestra memoria EPROM , siga una dirección ascendente , es decir comience por la dirección

0000 y termine en la dirección más alta de la capacidad de nuestra memoria EPROM , este circuito contador debe ser accionado por señales de reloj mandadas por la microcomputadora , es decir por los canales de direccionamiento , mandamos la dirección de inicio , en este caso en particular usaremos la dirección 4000 hexadecimal como si fuera la dirección 0000 hexadecimal de nuestra memoria EPROM .

Tenemos la necesidad de dos circuitos más ; uno que habilite la entrada del voltaje de programación es decir , un voltaje de +25 volts , y otro que mande el pulso de programación que se necesita , es decir , el pulso de 50 mseg. de programación .

Dentro del programa de programación deberemos también tener un programa de copiado , mediante el cual podremos copiar en unos cuantos segundos toda la información de una memoria ROM , PROM , EPROM ó EEPROM.

### CAPITULO III

#### CIRCUITOS DEL PROGRAMADOR

##### a) ESTRUCTURACION DEL DIAGRAMA DE BLOQUES.

Antes de empezar con nuestros circuitos debemos de estructurar un diagrama a bloques de nuestro programador , ya que se incluye una micro computadora que lo controla .

Este diagrama será muy parecido al de una microcomputadora , en primer término , pero en realidad , tendrá variaciones pues será un programador de memorias EPROM .

He decidido usar un Z80A en el diseño , pues este microprocesador cumple con las necesidades de nuestro circuito y además tiene la velocidad adecuada

da .

Por lo que nuestros diagramas se centran en el microprocesador Z80A , las fuentes básicas con las que este microprocesador es alimentado , son ; señal de reloj , fuente de voltaje y circuito reiniciador de dirección .

El Z80A es el que controla a su vez a todos los demás integrantes del circuito total , contenemos además dos diferentes tipos de memoria ; la memoria RAM y la memoria ROM , donde almacenamos todo lo relacionado al microcomputador .

Contenemos además un puerto programable 8279 mediante el cual manejaremos nuestro tablero visualizador , así como el teclado de nuestro programador .

Poseemos decodificadores que nos darán señales de habilitación para los puertos y las memorias , y diferentes memorias , así como un decodificador mediante el cual mandaremos señales del control al programador .

EN la figura 6 , se muestra el diagrama a bloques del programador computarizado entero . En él podemos apreciar cada una de las partes integrantes de nuestro circuito , podemos ver las memorias y el microprocesador unidos por los canales respectivos.

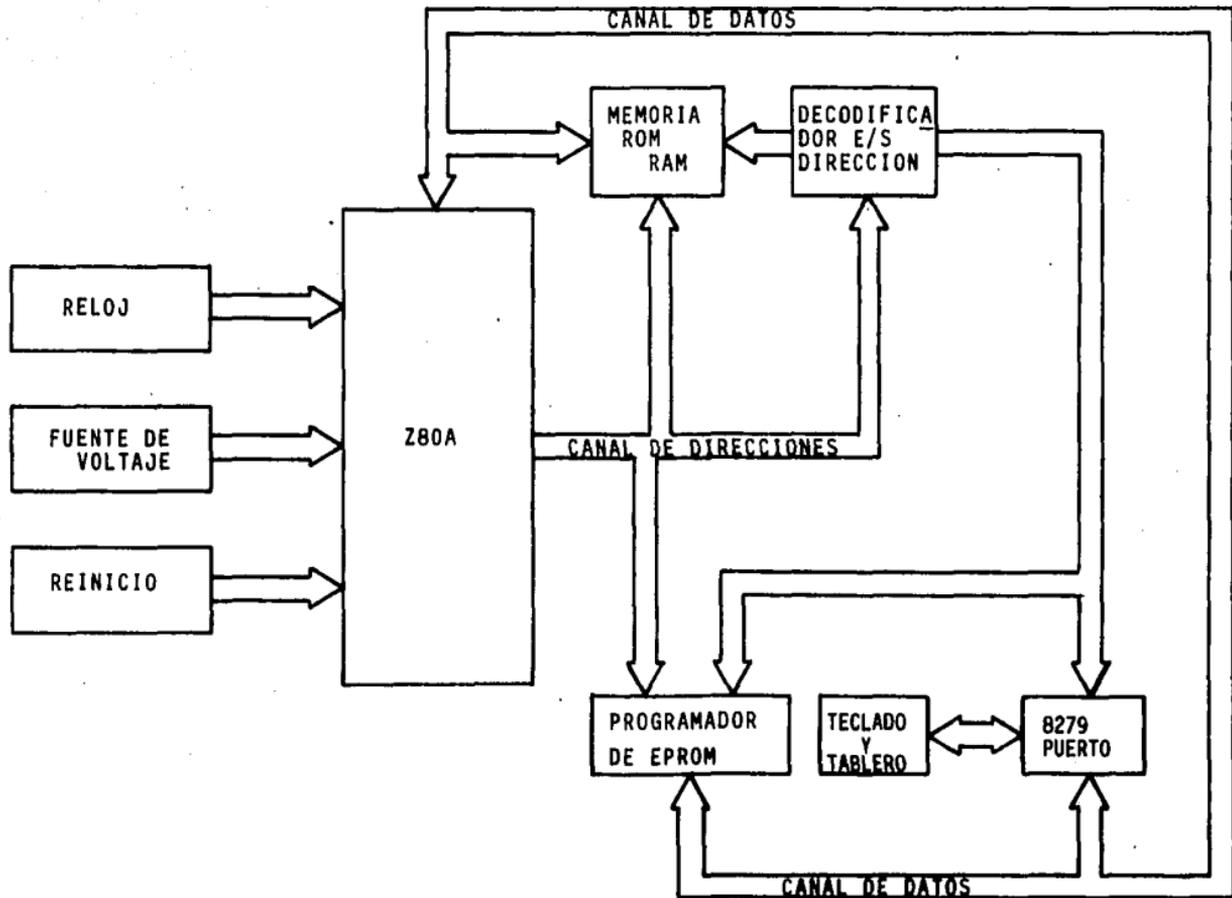


FIGURA 6. DIAGRAMA A BLOQUES DEL PROGRAMADOR COMPUTARIZADO.

FUNCIONES DE LAS PATITAS  
DEL Z80A

En la figura A se encuentra un diagrama de la disposición de las patitas del integrado microprocesador Z80A , el cual utilizaremos para la computarización de nuestro programador .

A continuación describiremos las funciones de cada una de las patitas del z80A ;

SIMBOLO	CARACTERISTICAS
A <sub>0</sub> -A <sub>15</sub>	CANAL DE DIRECCIONES Salidas tri-estado , activadas en uno lógico .
D <sub>0</sub> -D <sub>7</sub>	CANAL DE DATOS Triestado salida y entrada , activada con uno lógico .

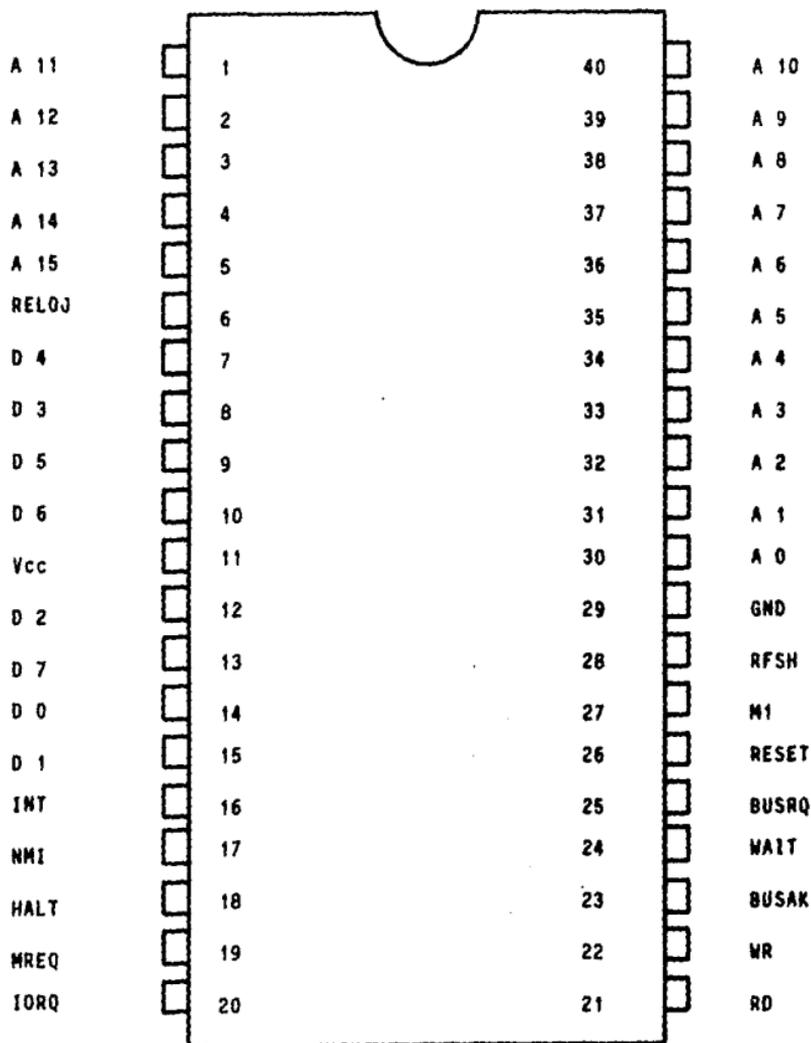


FIGURA A . CONFIGURACION DE LAS PATITAS DEL 280A

**MT**

#### **CICLO DE MAQUINA 1**

Salida , activa a nivel bajo .  
M1 indica que el ciclo de máquina en curso es el ciclo de búsqueda del código de operación de una ejecución de instrucción . Ha de tenerse presente que durante la ejecución de códigos de operación de dos octetos , M1 es generada cuando se busca cada octeto de código de operación. estos códigos de operación de dos octetos comienzan siempre con CBH,DDH,EDH o FDH. M1 se presenta también con IORQ para indicar un ciclo de acuse de recibo de interrupción.

**MREQ**

#### **PETICION DE MEMORIA**

Salida triestado a nivel bajo .la señal de petición de memoria indica que el canal de direcciones retiene una dirección válida para

una operación de lectura de memoria o escritura de memoria .

## IORQ

### PETICION DE ENTRADA/SALIDA

Salida triestado , activa a nivel bajo . Esta señal indica que la mitad inferior del canal de direcciones retiene una dirección de e/s válida para una operación de lectura o de escritura de e/s . Una señal IORQ se genera también con una señal M1 , cuando se acusa recibo de una interrupción , para indicar que un vector de respuesta de interrupción puede colocarse en el canal de datos . Operaciones de acuse de recibo de interrupción pueden producirse durante el tiempo de M1, aunque están prohibidas las operaciones de e/s .

RD

#### LECTURA DE MEMORIA

Salida triestado activa a nivel bajo , RD indica que el procesador central desea la lectura de memoria o de un dispositivo de e/s. La memoria o dispositivo de e/s direccionado , debe utilizar esta señal para controlar los datos en el canal de datos del procesador central.

WR

#### ESCRITURA DE MEMORIA

Salida triestado , activa a nivel bajo . WR indica que el canal de datos del procesador central retiene datos válidos a almacenarse en la memoria o dispositivo de e/s objeto de direccionamiento .

RFSH

#### REGENERACION

Salida , activa a nivel bajo , RFSH indica que los siete bits inferiores

del canal de direcciones obtienes una dirección de regeneración para memorias dinámicas y la señal MREQ en curso debe emplearse para realizar una lectura de regeneración para todas las memorias dinámicas .

## HALT

### ESTADO DE PARADA

Salida , activa a nivel bajo . HALT indica que el procesador central a ejecutado una instrucción de alto y está esperando una interrupción no enmascarable o enmascarable (con la máscara válida ) antes de que pueda reanudarse la operación. Mientras se está en esta condición el procesador central ejecuta NOPs (sin operación ) para mantener la actividad de regeneración de memoria .

## WAIT

## ESPERA

Entrada , activada a nivel bajo .  
espera indica al procesador central  
en los dispositivos de e/s o memo-  
ria , objetom de direccionamiento  
no están preparados para una trans-  
ferencia de datos . E l procesador  
central continúa introduciendo es-  
tados de espera mientras que WAIT  
es activa ; esta señal permite que  
la memoria de dispositivos de e/s  
se sincronice con el procesador cen-  
tral .

## INT

## INTERRUPCION

Entrada activa a nivel bajo.  
La señal de petición de interrup-  
ción se genera por dispositivos  
de e/s . Una petición se aceptará  
al final de la situación en curso si  
está activado el flip-flop de vali-  
dación de interrupción bajo control

de los programas internos , y si no es activa la señal BUSRQ . Cuando el procesador central acepta la interrupción una señal de acuse de recibo IORQ durante el tiempo M1 se emite al principio del siguiente ciclo de interrupción . El procesador central puede responder a una interrupción en los tres modos diferentes.

#### NMI

#### INTERRUPCION NO ENMASCARABLE

Entrada , disparada por flanco negativo . La línea de petición de interrupción no enmascarable tiene una prioridad mas alta que INT , y siempre se identifica al final de la instrucción en curso, prescindiendo del estado del flip-flop de validación de interrupción . NMI obliga al procesador central a reiniciarse en la posición 0066<sub>16</sub> . El contador de programa se conserva automáticamente

en la pila externa de modo que el usuario pueda volver al programa que fue interrumpido . Ha de tenerse en cuenta que los ciclos de espera ( WAIT ) continuos pueden impedir que finalice la instrucción en curso y que una señal BUSRQ anular una señal NMI .

## b) CIRCUITO DE RELOJ.

El Z80A cuenta con una entrada de reloj , mediante la cual se obtiene la sincronia y velocidad de programas e instrucciones , en las hojas de especificaciones de nuestro Z80A , se lee que la velocidad de nuestro reloj debe ser de 4 MHz .

En nuestro caso como ya hemos visto , la velocidad de proceso es suficiente , como para grabar un EPROM , pero necesitamos que esta señal cuadrada sea completamente exacta o lo más cercano , para que la coordinación y velocidades así como los tiempos de grabación sean exactos , por lo que , la forma más segura de lograrlo es diseñando un circuito de reloj , que esté controlado por un cristal de cuarzo , el cual es lo bastante exacto para nuestras metas .

Ademas el cristal de cuarzo es lo suficiente mente estable como para no variar los tiempos de programación o lectura .

Ahora simplemente necesitamos un circuito que active el cristal de cuarzo , es decir , lo inicialice y mantenga oscilando a la frecuencia a la cual ha sido diseñado, existen en el mercado un sin número de circuitos, pero necesitamos uno sencillo y que sea confiable.

De los circuitos activadores de cristal de cuarzo uno de los más sencillos es el de activación por medio de dos compuertas nand como si fuesen inversores.

Se utiliza despues del circuito activador de cristal de cuarzo, un inversor para darle la componente de corriente directa que necesita , y a su vez le siva como amplificador de linea además necesitamos insertarle una resistencia de levantamiento para cerciorarnos de que el voltaje de +5 volts estará presente en la parte alta del pulso de reloj.

Esta resistencia bastará con que tenga un valor de 330 ohms.

El cristal de cuarzo se vende en las tiendas con un valor especial para el Z80A , es decir , la frecuencia especificada en sus características es de 4 MHz , con el circuito que implementamos en la figura 7 tenemos que la oscilación será de 4MHz , aún cuando nosotros cambiáramos el cristal a otro valor , este circuito lo activaría a oscilar a la frecuencia que se especifica en el cristal que se inserte .

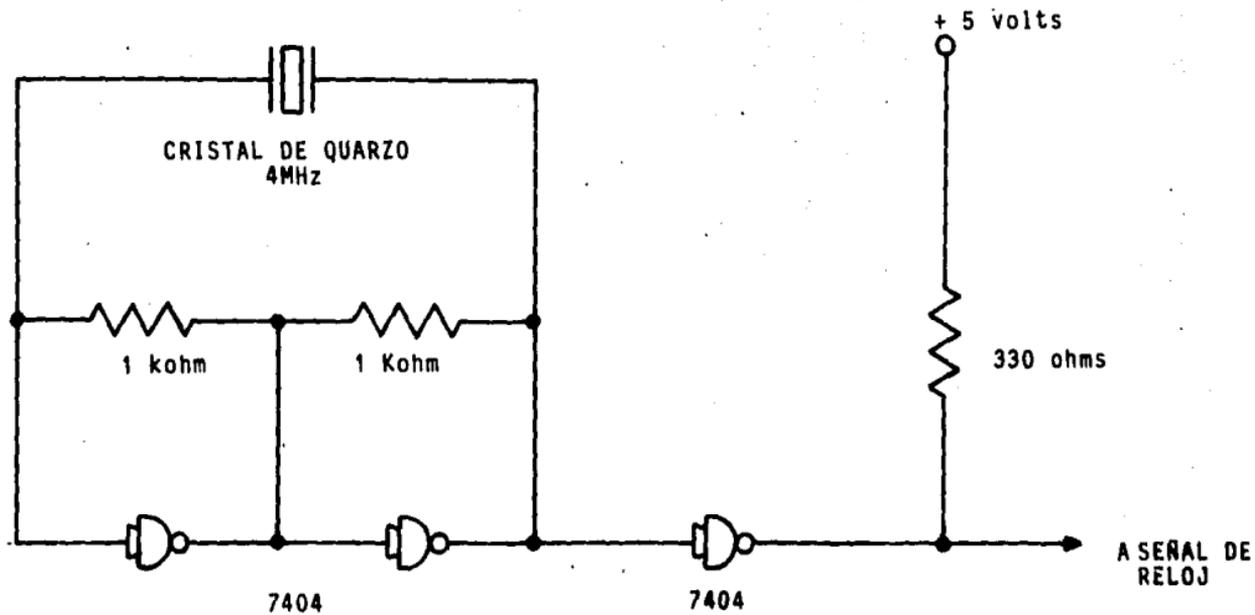


FIGURA 7 . CIRCUITO ACTIVADOR DE CRISTAL DE QUARZO .

### C) CICUITO REINICIADOR.

Esta función aunque aparentemente no tiene mucha importancia, es uno de los controles mas importantes y necesarios , pues es la única forma de inicializar el contador del programa del programador, es decir empieza ejecutando las instrucciones grabadas en la memoria rom dirección 0000 .

Cuando un programa se ejecuta en forma incorrecta esta tecla es la única forma de terminar el loop , pues inicializa el contador.

Puesto que la activación de esta función es por medio de la transición de un estado lógico de uno a el estado lógico 0; utilizaremos un circuito mediante el cual mantengamos activado el canal de "RESET" reinicio , en el z80 es una

entrada directa que precisamente se titula "RESET" que en español significa reinicio , en esta entrada debemos de mantener el nivel antes descrito , por mesta razón utilizaremos una resistencia de - 2k2 ohm interconectada al nivel positivo de +5 v. con lo que obtendremos un nivel constante de +5 volts en la entrada de reinicio , cuando queramos hacer la transición tan solo tendremos que activar una tecla que interconecte el extremo de la resistencia de 2k2 ohm con la terminal tierra, esto nos proporcionará un estado lógico de cero.

Para asegurarnos de que la transición sea llevada acabo amplificaremos la señal a través de dos compuertas lógicas, con las cuales obtendremos niveles lógicos con más seguridad.

Las dos compuertas lógicas son inversores en serie que dan una respuesta igual pero asegurando un nivel lógico , el circuito entero se encuentra en la figura 8.

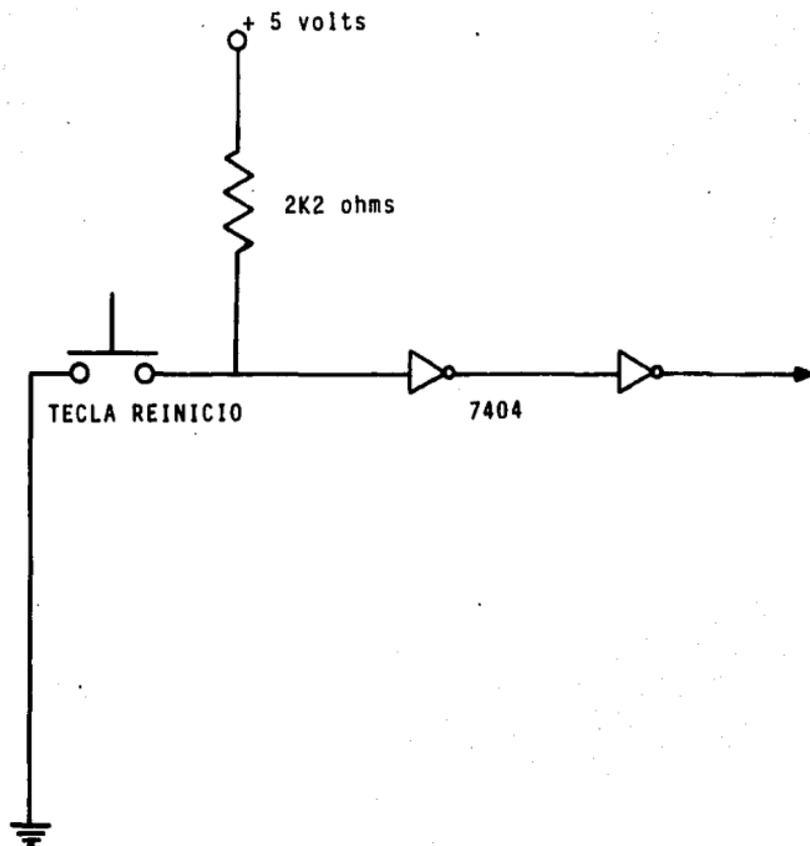


FIGURA 8 . CIRCUITO REINICIADOR DE DIRECCION

#### d) DISEÑO DE LA FUENTE DE VOLTAJE .

El programador requiere de dos fuentes de voltaje ; + 5 volts y + 25 volts , siendo éstos los únicos necesarios para suministrar la energía necesaria de operación al programador completo.

La fuente de voltaje es un elemento básico pues de ella depende el funcionamiento adecuado del programador.

Las exigencias de este pequeño programador en su alimentación , son muy inferiores a las que tienen las grandes computadoras ; podemos sacar provecho de los métodos de diseño convencionales e incorporar al mismo tiempo , los más recientes adelantos en la tecnología de los reguladores .

Me refiero a la implementación de un circuito integrado regulador de voltaje en la fuente .

127 volts

TRANSFORMADOR DE 127  
a 5 y 25 volts .

RECTIFICADORES  
Y FILTROS

REGULADORES

PROTECCION  
SOBRETENSION

TIERRA

+ 5 volts

+ 25 volts

FIGURA 9 . DIAGRAMA A BLOQUES DE FUENTE DE VOLTAJE .

Como podemos apreciar en la figura 9 la etapa del diagrama de bloques que inicia , es la del transformador ; aquí es donde se reduce el voltaje de línea de 127 volts ac. a un voltaje alterno apropiado para que la etapa de rectificación y regulación funcionen adecuadamente .

La segunda etapa es la nombrada ; rectificador y filtraje ; en donde se convierte el voltaje alterno reducido en un voltaje directo , es decir , de corriente directa, así como una etapa de filtrado que reduce el rizo producido por el voltaje que logra atravesar el rectificador .

La tercera etapa nombrada reguladores , es donde regularemos que el voltaje se mantenga en un nivel deseado y no exceda este nivel , ni arriba ni abajo de los niveles marcados .

La última etapa de protección contra tensiones altas , nos protege de un exceso de voltaje que

podría llegar a dañar a nuestro programador .

La entrada en el primario del transformador es una onda senoidal alterna de 127 volts , la salida del secundario del transformador es una onda senoidal , parecida a la de entrada pero su nivel es diferente en amplitud .

En la siguiente etapa se utilizan puentes rectificadores de onda completa , con diodos rectificadores de silicio , presentan una caída de 1 volt , a través de cada diodo .

Esta caída de voltaje se hace considerable en diseños de puentes de onda completa .

Esta caída por ser dos diodos en serie , se acumulan dos volts , por lo que es importante considerarlo en el cálculo .

El regulador de tensión requiere un nivel mínimo para mantener una tensión de salida cons-

tante , si el voltaje cae por debajo de este punto, la estabilidad a la salida queda muy baja .

Por lo que un condensador de salida se utiliza para suavizar las protuberancias de la onda senoidal rectificada .

Cuando los diodos conducen , el condensador almacena la carga suficiente para mantener la tensión mínima requerida hasta el siguiente ciclo de carga .

La entrada en el transformador es de 60 Hz pero como es rectificación completa , se producen ciclos de 120 Hz , el condensador se carga en un ciclo de 8.3 ms , la descarga del condensador es lo que conocemos como rizo .

En la figura 10 se muestra la característica de carga-descarga del capacitor , y en la figura 11 el voltaje de rizo , con una referencia al voltaje alimentación .

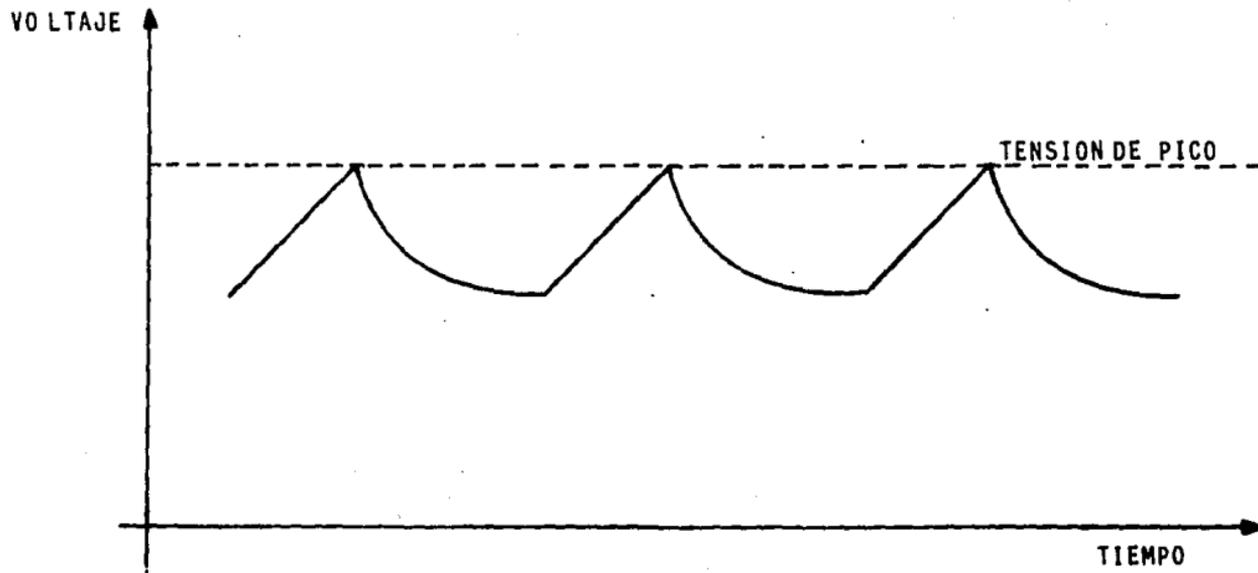


FIGURA 10 . CARACTERISTICA CARGA DESCARGA .

Se necesitan dos fuentes ; una de + 5 volts y la otra de + 25 volts , nos referiremos primeramente a la más importante que es la de + 5 volts pues suministra de voltaje a todos los componentes del programador , el mínimo voltaje requerido para -- hacer funcionar el regulador es el nivel de 8.5 volts por lo que suministraremos un voltaje de 12 volts con lo que tendremos un margen bastante seguro de suministro .

Para calcular el factor de rizado tenemos que considerar dos cosas ; el voltaje de pico que es de 25 % mayor que el voltaje de alimentación  $V_c$  o voltaje de entrada , que también es considerado para el factor de rizado , analicemos la ecuación:

#### FACTOR DE RIZADO

$$Y_f = ( V_{pico} - V_c ) / V_c$$

$$Y_f = ( 15 - 12 ) / 12 =$$

$$Y_f = 25 \%$$

## CALCULO DEL CONDENSADOR

El condensador debe soportar por lo menos una tensión de 12 volts a partir de un voltaje de pico de entrada de 15 volts .

$$V_{\text{pico}} = 15 \text{ volts}$$

$$V_{\text{rizado}} = 15 - 12$$

$$V_{\text{rizado}} = 3$$

Para obtener el valor del capacitor tenemos que considerar la ecuación ;

$$C = dt/dv \times I$$

Donde c es el valor del condensador en fara-

## CALCULO DEL CONDENSADOR

El condensador debe soportar por lo menos una tensión de 12 volts a partir de un voltaje de pico de entrada de 15 volts .

$$V_{pico} = 15 \text{ volts}$$

$$V_{rizado} = 15 - 12$$

$$V_{rizado} = 3$$

Para obtener el valor del capacitor tenemos que considerar la ecuación ;

$$C = dt/dv \times I$$

Donde c es el valor del condensador en fara-

dios , I la corriente máxima del regulador , cuyo valor en nuestro proyecto , como un máximo establecemos de 3 amperes , dt el tiempo de carga del capacitor ; cuyo valor es de 8.3 ms (120 Hz) y por último dv que es la tensión de rizado admisible cuyo valor es de 3 volts.

Por lo que aplicando la fórmula nos queda ;

$$C = 3 \times 8.3 \times 10^{-3} / 3$$

$$C = 8.3 \text{ mf}$$

Como el valor comercial es de 20 000 uf , lo tomamos y esto nos reduce el voltaje de rizado a 0.925 volts , lo que nos dá más margen de seguridad que permite en realidad seguridad en la programación pues protege de variaciones de voltaje que puedan producir errores en la realización de los programas.

Como el transformador aumenta su valor nominal del marcado , pues cambia el voltaje de entrada y a su vez el de salida , por lo que el voltaje del capacitor le damos un margen de 20 volts .

Para los reguladores se utiliza simplemente un 7805 , para regular + 5 volts , este circuito integrado nos regula el voltaje a través de sus interconexiones internas, proporcionando una seguridad casi total en la regulación .

Para la regulación de los 25 volts utilizamos un regulador de 24 volts , 7824 , pues el voltaje de programación es lo que se suministra con esta fuente y puede establecerse en 24 volts que son suficientes para llevar a cabo una programación bien realizada .

El circuito de protección por sobre voltaje consiste en un dispositivo que al exceder un voltaje

al previsto de pico , desconecta por medio de un triac , mandando toda la corriente a tierra.

## e) MEMORIAS .

Este programador tiene la necesidad de hacer uso de dos tipos de memorias que comunmente se utilizan en el diseño de computadoras , estas son ; las memorias RAM y ROM .

Puesto que tenemos un programador que usa un puerto programable para el uso de teclado y tablero visualizador , se debe de tener en consecuencia un programa para manejar el puerto 8279 , este programa se carga en una memoria EPROM , que como ya sabemos es programable, leible y borrable .

El programa que maneja a nuestro puerto , se graba en un EPROM , que constituye una parte de la memoria ROM , en la segunda parte , es decir ; en el segundo EPROM de memoria ROM , tendremos el programa que dirige nuestro programador.

El otro tipo de memoria es la memoria RAM

donde se cargarán los programas que deseemos grabar en las memorias EPROM , así como programas para dirigir la computadora a otras aplicaciones, sobre el mismo campo .

La capacidad de la memoria RAM, debe ser lo suficientemente grande para poder grabar integralmente el programa en la memoria EPROM .

Como mínimo deberemos de tener una memoria de 8K x 8 , para cargar el programa que puede ser introducido en una memoria EPROM de esa capacidad , en este caso por la necesidad que pueda ofrecerse expandemos la memoria a una capacidad mas alta es decir , la aumentamos a 10 K x 8 .

Para el uso de las memorias se deberá acoplar un decodificador , mediante el cual daremos una habilitación de la memoria que se desee usar en un momento dado , es decir , a través del decodificador damos un pulso negativo que habilita el acceso y salida de información a la memoria seleccionada

por lo que colocamos nuestras memorias a diferentes alturas en nuestras direcciones de memoria , tratando de cubrir la secuencia total , de un determinado número de capacidad en una de nuestras memorias, es decir, tenemos en nuestro caso que acoplar este decodificador para que cada dos mil por ocho instrucciones de dirección de memoria se habilite la siguiente memoria , puesto que tanto la capacidad de nuestra memoria RAM como la capacidad de cada una de nuestras memorias ROM , tienen una capacidad de 2K x 8 .

## f) TABLERO VISUALIZADOR Y TECLADO .

Estas partes básicas del programador, son de suma importancia , pues el teclado por ejemplo , no tan solo es usado para cargar los programas a grabar en la memoria EPROM , es decir los que se graban anteriormente en la memoria RAM , sino que se usa para las múltiples funciones que se desarrollan con una microcomputadora , en conclusión el teclado no solo es necesario , sino que es indispensable para poder tener un control sobre el programador , digamos que el teclado es el medio mediante el cual nosotros , personas nos comunicamos con el microprogramador .

También el tablero visualizador es de suma importancia e imprescindible , puesto que sin él el micro programador no podría comunicarse con nosotros , entonces ; nosotros nos comunicamos con el micro programador por medio del teclado

y el microprogramador se comunica con nosotros por medio del tablero visualizador .

Para manejar estas partes esenciales de nuestro micro programador utilizaremos un puerto 8279 mediante el cual podremos manejar todas las funciones de nuestro teclado y tablero visualizador.

En la figura 12 se muestra un diagrama esquemático donde se interrelacionan los componentes de la sección que estamos describiendo .

Como podemos apreciar: a través del 8279 nos intercomunicamos con el microprocesador 280A , digamos que un puerto es el enlace decodificador.

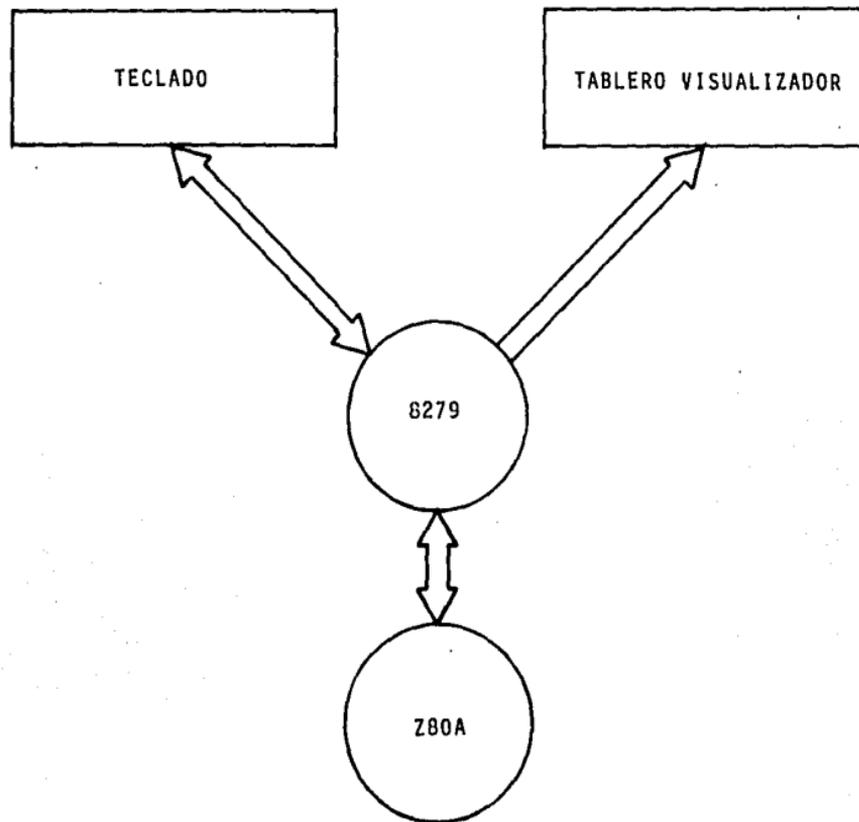


FIGURA 12 . CONTROLADOR DE TABLERO VISUALIZADOR Y TECLADO.

DESCRIPCION DEL PUERTO PROGRAMABLE  
DE TECLADO Y TABLERO VISUALIZADOR  
8279

El puerto programable de Intel 8279 es de pro  
posito general para tableros visualizadores y tecla  
do, puerto de entrada y salida programable interfa  
se , diseñado para usarse con micro computadores de  
intel o similares , como es el caso de nuestro Z80A  
que lo hacemos compatible con este puerto .

La porción del teclado proporciona un barrido  
de interfase de matriz de 64 , también puede usar-  
se como interfase para un arreglo de sensores o es-  
trobo interfase de teclado.

Si más de ocho caracteres son tecleados el  
estado de sobre entrada se acciona , y colocan la  
salida de interrupción a la cpu.

La porción del tablero visualizador , produce

un barrido para led , incandescente y otras tecnologías de tableros visualizadores , ambos numéricos y alfanuméricos , también pueden ser usados tableros visualizadores de segmentos como simples indicadores , el tablero visualizador posee un arreglo de 16 x 8 de memoria RAM , el cual puede también organizarse en dos de 16 x 4 , la RAM puede ser cargada e interrogada por la CPU .

En la figura 13 apreciamos la disposición de las patitas del puerto 8279 .

#### DESCRIPCION FUNCIONAL DE LAS TERMINALES DEL 8279

##### Símbolo

DB0 - DB7 BUS DE DATOS BIDIRECCIONAL

Todos los datos y comandos son transmitidos por estas líneas .

- CLK**            **RELOJ**  
Entrada de reloj del sistema usado para generar los tiempos internos .
- RESET**        **REINICIO**  
Una señal alta en esta patita reinicia el 8279 , despues de haber sido reiniciado el 8279 se situa en el siguiente modo;  
1) 16-8 bit caracteres tablero visualizador..  
2) barrido codificado de teclado .
- CS**            **SELECTOR DE INTEGRADO**  
Un pulso bajo en esta patita y se habilita el modo de entrada-salida ..
- Ao**            **AMPLIFICADOR DE DIRECCION**  
Un uno lógico en esta patita indica que las señales entrada o salida son interpretadas como comandos o estados . Un cero lógico indica que son datos .

RD , WR      ENTRADA/SALIDA LEER ESCRIBIR .  
Estas señales habilitan los amplificadores de datos , para mandar datos al bus externo o recibir datos del bus externo .

IRQ            PETICION DE INTERRUPCION  
En modo de teclado , la línea de interrupción es alta cuando hay un dato en FIFO/sensor RAM . la interrupción va al estado cero lógico cuando con cada FIFO/sensor RAM lee y regresa a uno lógico si todavía hay información en la RAM . En modo de sensor , la interrupción se convierte en uno lógico cuando cualquier carga se detecta en el sensor.

Vss, Vcc      TIERRA Y VOLTAJE .

SLo-SL<sub>3</sub>      LINEAS DE BARRIDO  
Son usadas para realizar el barrido de teclado, o la matriz de sensores y el

tablero visualizador.

RLO-RL<sub>7</sub>

LINEA DE REGRESO

Estas líneas de regreso son con las cuales están conectadas a las líneas de barrido a través del teclado o sensores, están diseñados para mantenerse en estado lógico uno, a menos que se sitúe un contacto en posición de cero lógico.

SHIFT

CAMBIANTE

La entrada de estado de shift es guardada a lo largo que con la posición de tecla prendida constante, tiene un activador interno que la mantiene en uno lógico a menos que sea introducido un cero lógico.

CNTLS/STB

MODO DE ENTRADA ESTROBO/CONTROL

Para modo de teclado esta línea es usada como un control de entrada y almace-

naje como estado sobre una tecla cerrada.  
La linea también es una linea de estro-  
bo que entra el dato dentro del FIFO en  
el modo de entrada de estrobo .

OUT A<sub>0</sub>-OUT A<sub>3</sub> SALIDAS

OUT B<sub>0</sub>-OUT B<sub>3</sub>

estos dos puertos son salidas para el  
tablero visualizador de 16 x 4 , estos  
dos puertos pueden ser considerados co  
mo uno de 8 bits .

**BD**

TECLADO VISUALIZADOR EN APAGADO

Esta salida se usa para borrar el table-  
ro visualizador durante un swicheo de  
un dígito o un comando de borrado .

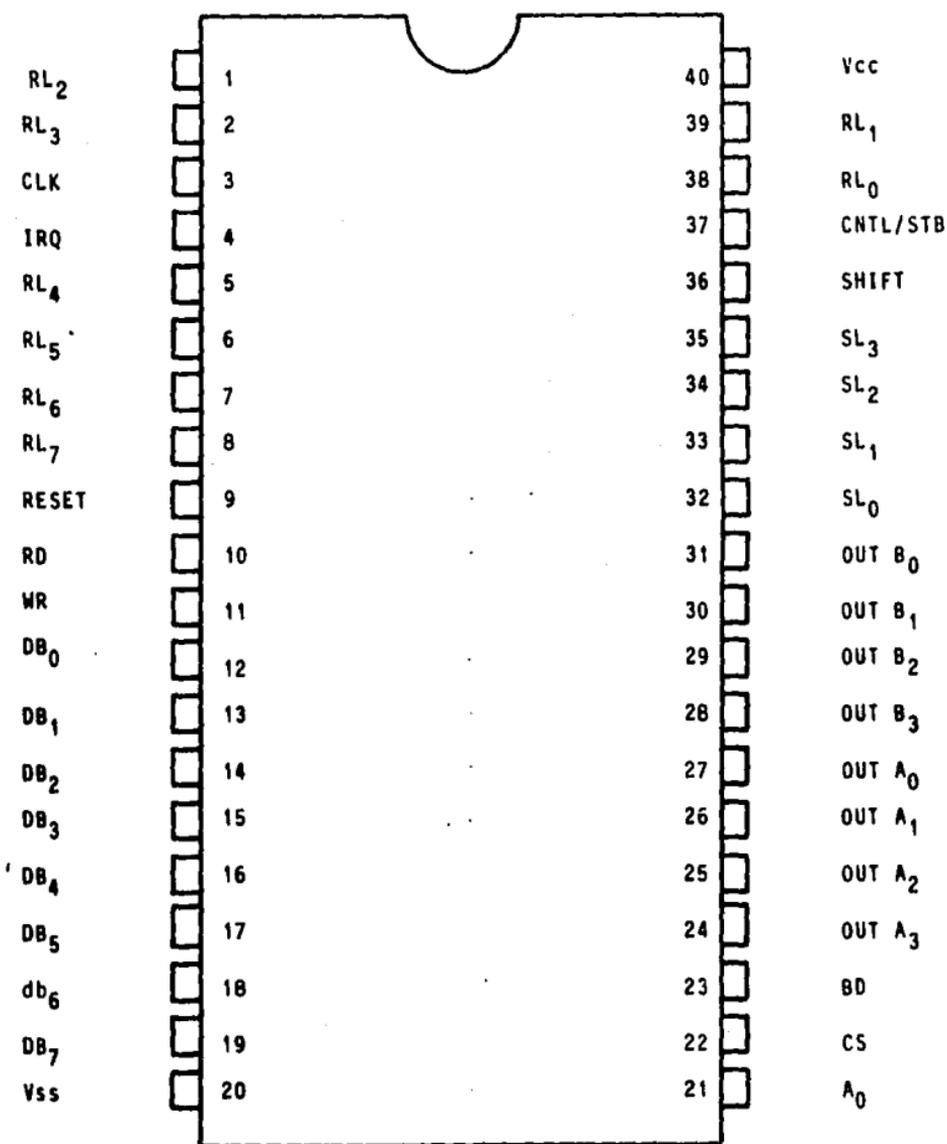


FIGURA 13 . CONFIGURACION DE LAS PATITAS DEL PUERTO

## g )DISEÑO DE LA BASE CONECTORA DE PROGRAMACION DE MEMORIA EPROM .

Para una elaboración correcta de la base debemos primero analizar el EPROM , en este caso nos dedicaremos al análisis del 2716 , pues por su capacidad se ha vuelto de los más populares del mercado .

En la tabla de la figura 16 encontramos los modos tabulados de operación mediante los cuales , debemos de manejar la memoria 2716 .

En ella se dan los valores de voltajes , a los cuales deben fijarse en cada una de las señales correspondientes tabuladas .

Por el momento concentraremos nuestra atención en el modo de programación .

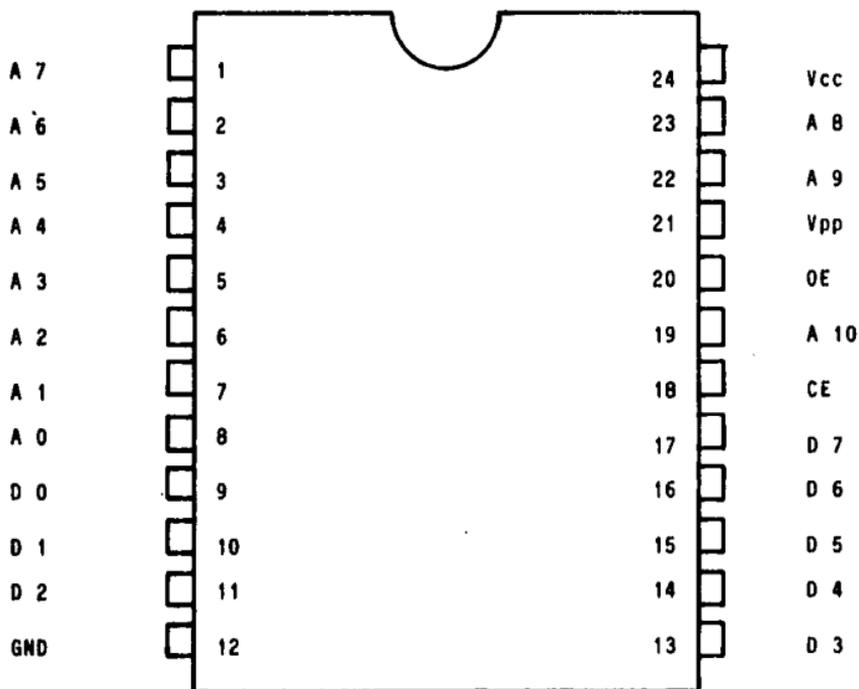


FIGURA 14 . CONFIGURACION DE LAS PATITAS DE LA MEMORIA EPROM 2716 .

TIERRA  
Vpp  
Vcc  
OE  
CE/PGM

ENTRADAS DIRECCION

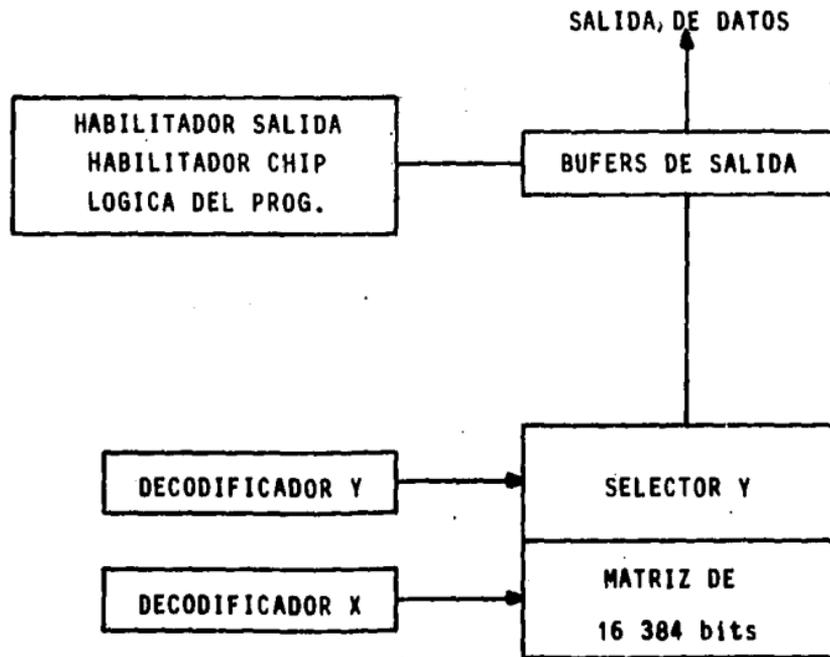


FIGURA 15 . DIAGRAMA A BLOQUES DE LA MEMORIA EPROM 2716.

MODO	CE/PGM	SENALES		
		OE	Vpp	Vcc
LECTURA	0	0	5	5
RESERVA	0	-	5	5
PROGRAMACION	PULSOS	5	25	5
VERIFICACION	0	0	25	5
INHIBICION	0	5	25	5

FIGURA 16 . TABLA DE MODOS DE OPERACION DE LA MEMORIA EPROM 2716 .

En un principio cuando la memoria esta recién borrada o la memoria es nueva , todos los bits de la memoria se encuentran en estado uno lógico , la programación se realiza grabando ceros lógicos en los bits deseados , aunque solo se graban ceros ambos estados pueden estar presentes en el momento de grabación en cada bloque de datos.

La única forma de cambiar un cero lógico grabado a un uno lógico es por medio del proceso de borrado a través de la irradiación sobre la ventana del chip de rayos ultravioletas por un periodo de tiempo determinado .

La 2716 está en el modo de programación cuando la fuente de alimentación conectada a Vpp suministra 25 volts y la terminal OE toma el valor de 5 volts además debe de existir una pulsación de 50 ms de tiempo con una amplitud de 5 volts , la información a grabar se presenta en grupos de 8 bits en paralelo , en las señales del canal de datos .

Cuando la dirección y los datos se estabilizan se aplica un voltaje TTL de programación de una duración de 50 ms , activa de nivel alto , a la entrada CE/PGM .

Hay que aplicar un impulso de programación a cada una de las direcciones de las posiciones que se van a programar .

Se puede programar cualquier dirección en cualquier momento incluso individualmente , secuencialmente o aleatoriamente .

El impulso de programación deberá tener una anchura máxima de 55 ms . La 2716 no debe ser programada aplicando una señal de corriente continua a la terminal CE/ PGM .

La elaboración de un generador de impulsos de 50 ms por medio del cual la memoria graba el dato en la dirección previamente establecida en el canal de datos .

Como hemos podido apreciar se necesita una sincronización metódica para la grabación correcta de nuestra memoria , por lo que debemos de poseer de un elemento que controle las funciones en nuestra base de programación, después de analizar el mercado , he observado que un puerto en paralelo es lo más apropiado para realizar esta tarea , como cuando necesitábamos un puerto para manejar el teclado y el tablero visualizador -- ahora necesitamos uno que nos proporcione comunicación con nuestro "EPROM" .

Existe un integrado que posee una característica muy importante para nosotros , es el 8255 de Intel , este posee 24 terminales programables entrada/salida , estos se dividen en 3 puertos que se nombran A,B y C , cuya aplicación es perfecta para usarlo como temporizador .

De esta forma podremos usar los puertos por separado y mantener los datos el tiempo necesario

para poder dar el impulso de programación.

Es muy importante que tengamos facilidad de realizar una secuencia exacta de cada una de las entrada/salida que tengamos.

Por supuesto como primer punto debemos de tener la facilidad de llevar al interior de la memoria "EPROM" a través del puerto el canal de datos por esta razón el puerto nombrado como A lo destinaremos a conectarlo con el canal de datos y este puerto tiene exactamente las terminales que necesitamos , es decir , desde el dato cero hasta el dato siete ; suman en total ocho terminales las cuales se acoplan perfectamente.

En nuestra entrada del puerto tenemos una entrada directa del canal de datos proveniente del Z80A la cual conectaremos al 8255 por medio de un amplificador bidireccional de tecnología TTL cuyo numero identificador será 74 245 , este nos proporcionará mas confiabilidad en el

sistema , pues evita errores causados por caídas de corrientes en la línea, pues necesita niveles TTL para activarse.

Tenemos otras señales de entrada que deben interconectarse con el Z80A , estas señales son señales de control las cuales van conectadas unas directamente y otras a través del decodificador habilitador , esto nos dá la facilidad de controlar nuestro puerto a nuestra voluntad a través de programas grabados en nuestra memoria RAM .

En el puerto b conectaremos nuestras señales de direccionamiento de la memoria "EPROM" , de esta forma podremos direccionar la memoria el tiempo que sea necesario para la programación , pues con el puerto en paralelo podemos mantener nuestra señal el tiempo que deseemos , a través de nuestro programa.

Como podemos apreciar la conexión de nuestra base de programación será definitiva y a través de

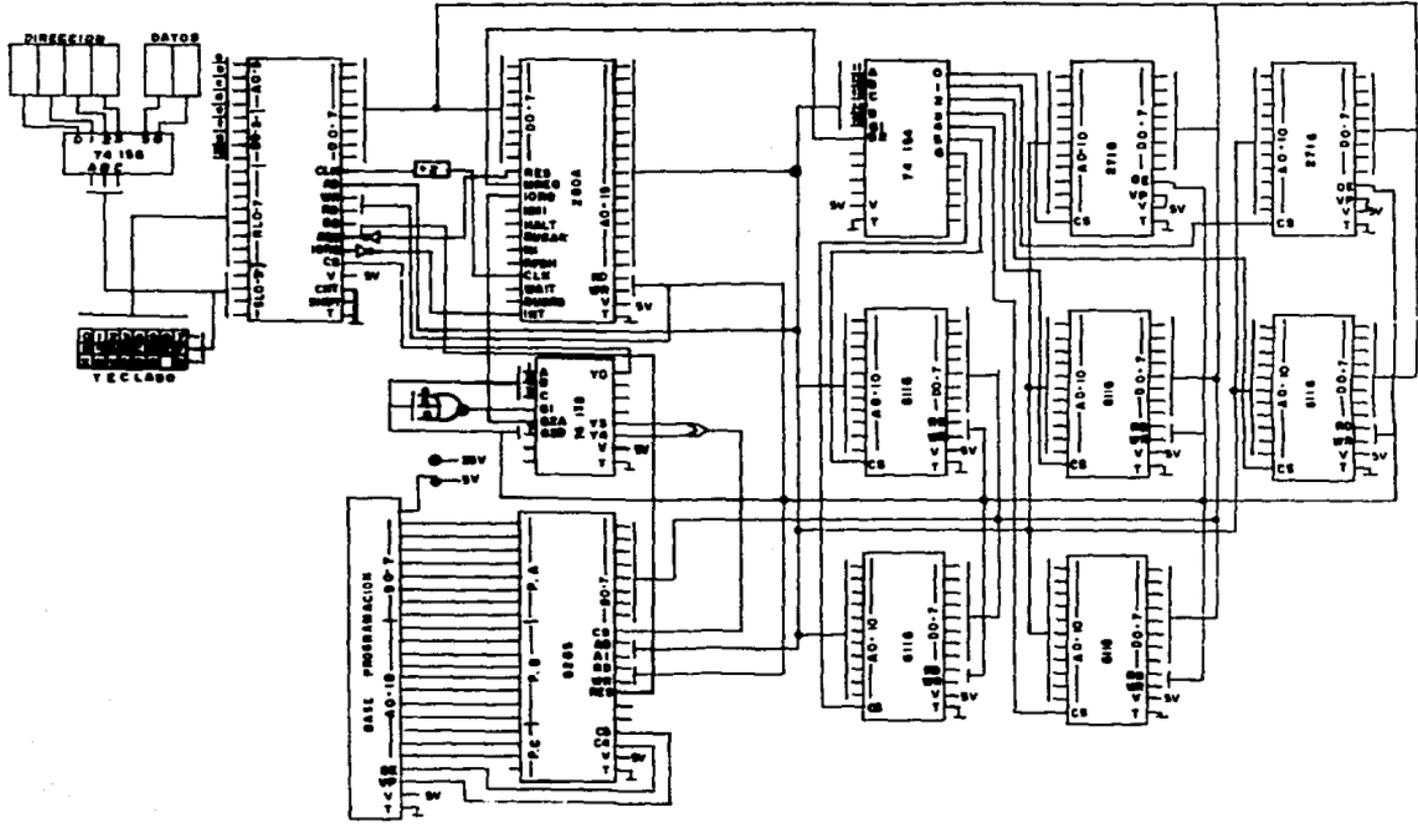
nuestro programa de grabación podremos efectuar las funciones deseadas , desde lectura , verificación , grabación , etc.

Incluso otras funciones pueden llevarse a cabo con este puerto pues recordemos que con solo cambiar el programa podremos grabar cualquier tipo de memoria .

En nuestro puerto C daremos las señales de control a nuestro "EPROM" , es decir , en este puerto las funciones del impulso de grabación y todos los requisitos que hemos denotado para llevar a cabo la programación , lectura , verificación y otras , seran manejadas atraves de este puerto.

Este puerto es programable , es decir con una señal de control se le ordena que puerto es de entrada y que puerto es de salida , con esto dominamos perfectamente las señales del canal de datos , pues recordemos que es una señal bidirec-

cional y a través de este mecanismo podemos efectuar a través del puerto A lectura y programación de los datos.



## CAPITULO IV

### ELABORACION DE LAS SUBRRUTINAS DE PROGRAMACION

#### a) DESCRIPCION GENERAL DE LAS SUBRRUTINAS

Estas subrrutinas son la base esencial del programador , pues son los programas que manejan la programación de la memoria EPROM , pero no solo manejan la programación sino que tambien verifican el borrado correcto de la memoria así como el copiado de una memoria .

b) SUBRRUTINA DE CHEQUEO DE BORRADO CORRECTO .

Esta subrrutina checa contra un número hexadecimal FF cada una de las direcciones de una memoria EPROM , esto es debido a que cuando una memoria EPROM se encuentra en borrado correcto esta contiene en todas sus localidades unos lógicos , es decir , todas sus direcciones contienen números hexadecimales de FF .

Al efectuar el chequeo de todas y cada una de las localidades , indica en su tablero visualizador si la memoria se encuentra en estado correcto para ser programada o no .

La subrrutina toma un dato de la primera localidad de la memoria EPROM y lo carga en el acumulador del microprocesador , despues carga el dato FF en un registro , posteriormente compara el acumulador con el registro y en caso de

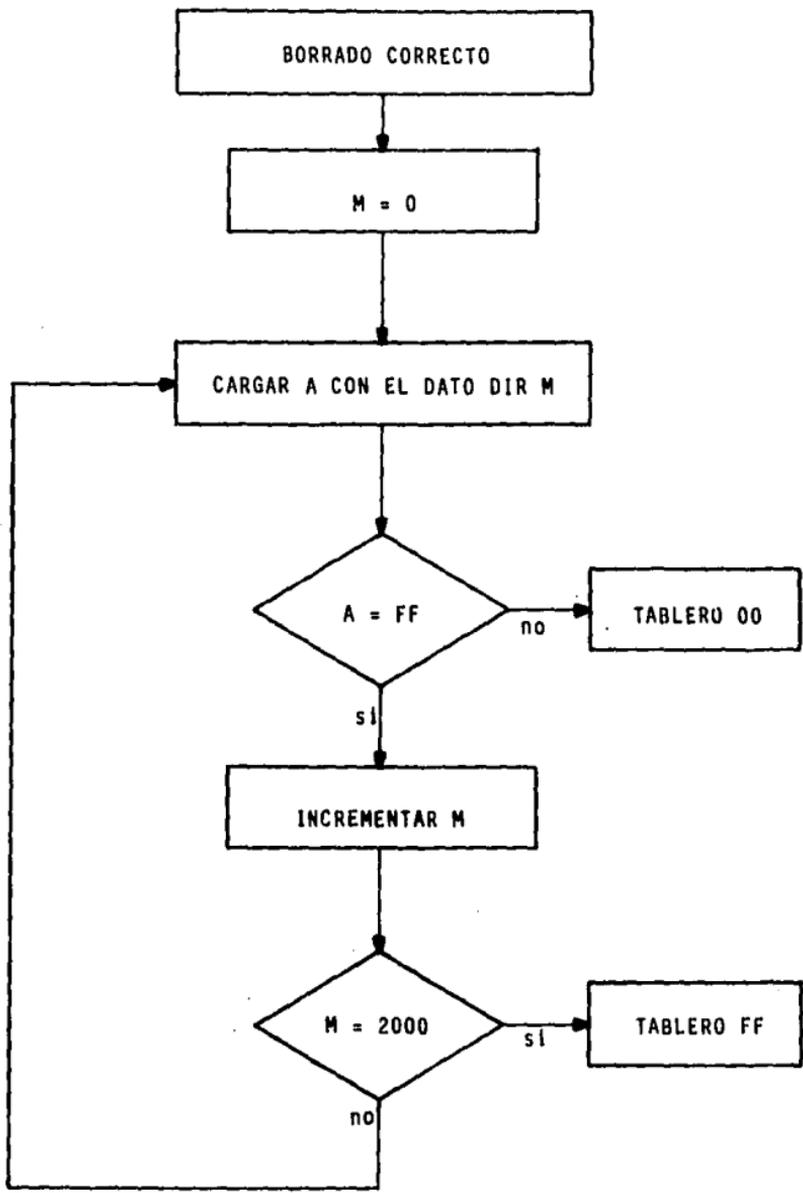
igualdad continúa con la siguiente localidad , en el caso de que no sea igual automáticamente indica en el tablero visualizador con los números 00 que la memoria no está bien borrada .

Cuando la memoria se encuentra bien borrada se indica en el tablero visualizador el número hexadecimal FF que recuerda la comparación efectuada .

En la figura 18 se aprecia el diagrama de flujo del programa verificador de borrado correcto .

Con este programa , absolutamente necesario , tendremos la certeza de que cuando grabamos un programa , este será correctamente grabado , es decir , que no habrá errores por una localidad mal borrada .

FIGURA 18 . DIAGRAMA DE FLUJO DEL PROGRAMA DE CHEQUEO DE BORRADO CORRECTO .



c ) SUBRRUTINA DE PROGRAMACION MANUAL .

Mediante este programa se introducen datos a una memoria EPROM en forma permanente , hasta la proyección de rayos ultravioletas sobre el circuito integrado .

Con este programa podemos insertar cualquier instrucción en cualquier localidad y tan solo una instrucción puede ser insertada en una sola localidad .

Para el caso de insertar una sola instrucción en direcciones intercaladas este programa es exelente , pero cuando se quiere insertar varias intrucciones , se tiene que utilizar un método un poco más ciclico , es decir , que tome los datos de la memoria RAM y los vaya grabando .

El diagrama de flujo de esta forma de programación se encuentra en la figura 19 .

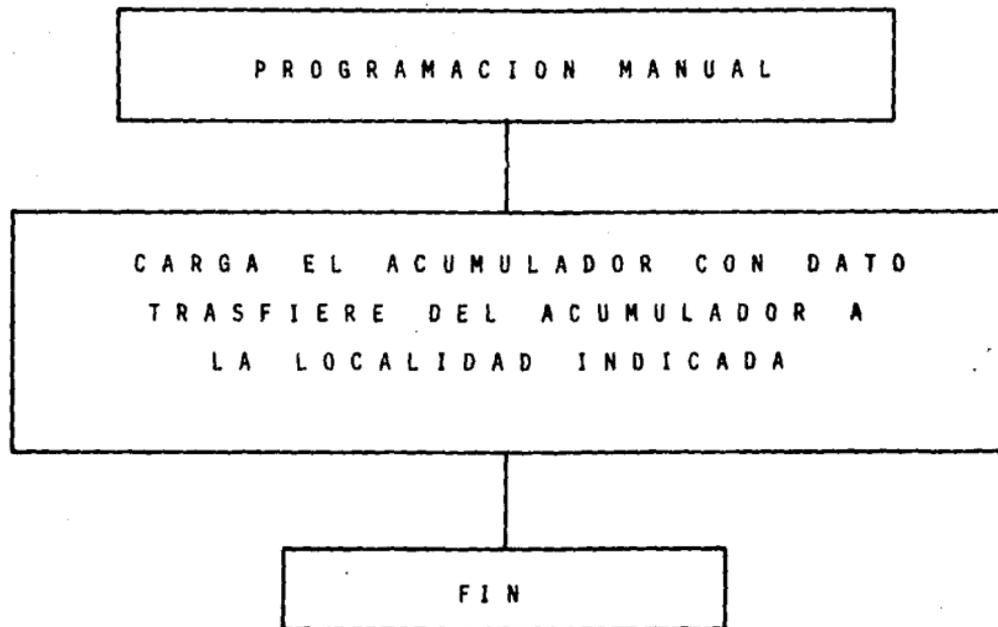


FIGURA 19 . DIAGRAMA DE FLUJO DE PROGRAMACION MANUAL .

#### d ) SUBRRUTINA DE PROGRAMACION AUTOMATICA .

Esta subrutina es la más valiosa del programador , puesto que elabora el proceso más largo en forma totalmente automática , en tan solo unos segundos .

Para simplificar el proceso de grabado , se debe introducir el número hexadecimal desde el cual se quiera empezar a grabar , después el número hexadecimal de instrucciones que se quieran grabar y por último se debe de grabar el programa a introducir en la memoria RAM empezando de la dirección 1000 en adelante .

De esta forma , tan solo con oprimir la tecla de correr programa y dando la dirección de subrutina adecuado , automáticamente se grabará en unos segundos las instrucciones en la memoria EPROM , el diagrama de flujo se aprecia en la figura 20 .

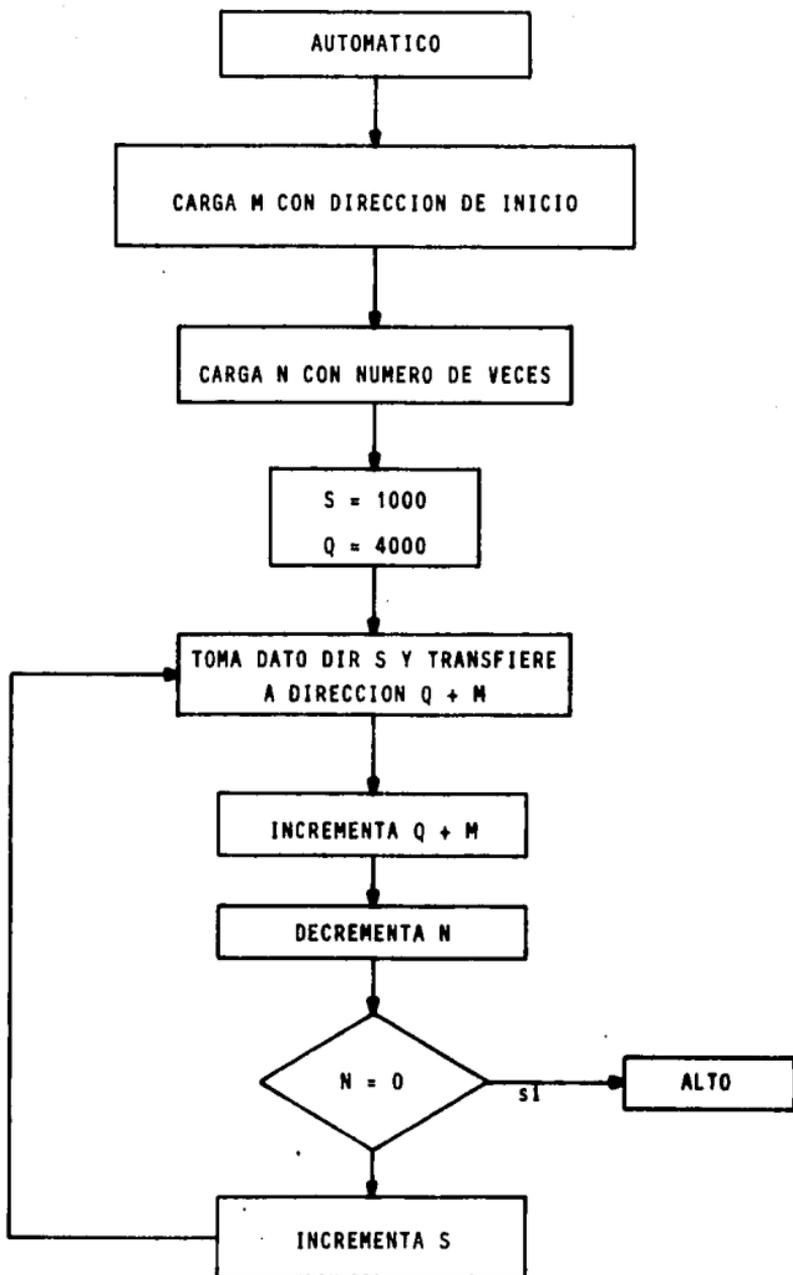


FIGURA 20 . DIAGRAMA DE FLUJO DE PROGRAMADOR AUTOMATICO .

e ) SUBRRUTINA DE COPIADO DE MEMORIAS EPROM .

Esta subrutina es muy importante pues facilita la tarea del programador , pues cuando tecleamos aunque sea copiando un programa de 2000 H instrucciones , es una labor bastante tardada y dificultosa en cambio , de esta forma tendremos la opción de tener una memoria ya grabada con un programa específico y copiarlo simplemente en el rango de memoria RAM destinado para la programación automática .

Simplemente se tiene que elaborar un programa sistemático que tome un dato de una localidad inicial lo transfiera a la localidad inicial de nuestra memoria RAM y vaya incrementando hasta llegar al final de la memoria EPROM a copiar .

De esta forma la subrutina de programación automática grabará sobre una memoria EPROM colocada

en la base de programación .

En la figura 2f podemos apreciar el diagrama de flujo de la subrutina de copiado de memorias EPROM .

Como se aprecia es corta pero cíclica , es por esto que es una labor sencilla pero laboriosa y el uso del microprocesador , ahorra bastante tiempo .

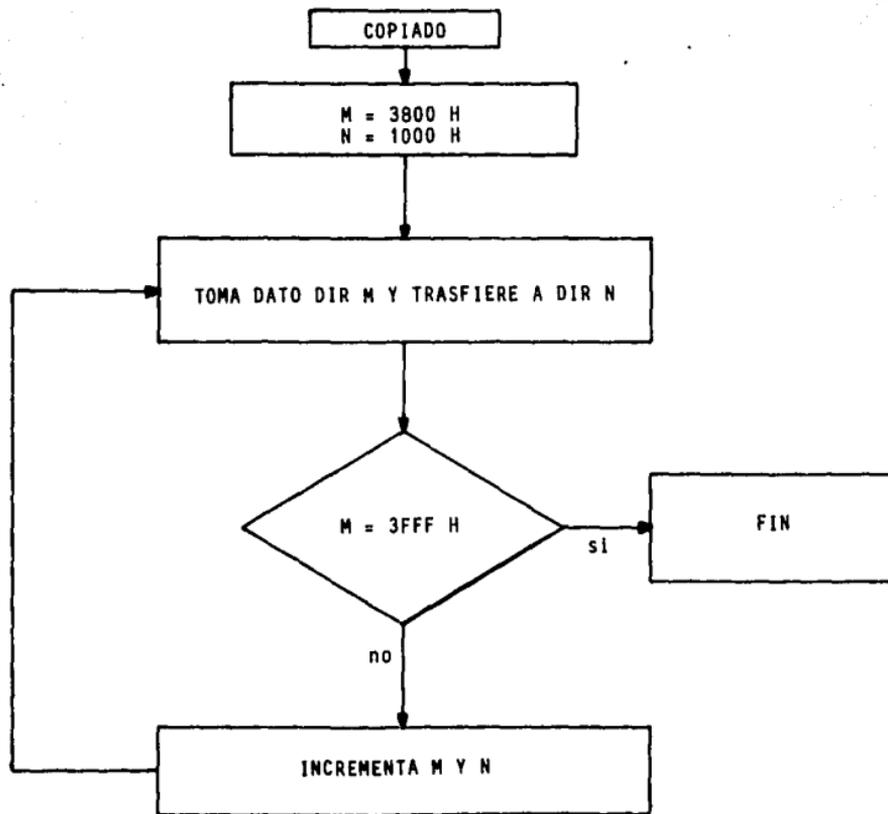


FIGURA 21 . DIAGRAMA DE FLUJO DE COPIADO DE MEMORIA EPROM .

## BIBLIOGRAFIA

UNDERSTANDING MICROCOMPUTERS & SMALL COMPUTER SYSTEMS  
NAT WADSWORTH  
DA CAPO PAPERBACK, 1977.

MICROSYSTEM COMPONENTS HANDBOOK  
MICROPROCESSORS AND PERIPHERALS  
VOLUME II, 1985  
INTEL CORP.

UNDERSTANDING MICROPROCESORS  
DON L. CANNON, PH.D  
RADIO SHACK , 1979

CONSTRUYA UNA MICROCOMPUTADORA BASADA EN EL Z80  
Gula de diseño y funcionamiento  
STEVE CIARCIA  
Mc GRAW HILL , 1984