ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERIA ELECTRICA

"DISENO Y CONSTRUCCION DE UN SISTEMA PROGRAMADOR DE MEMORIAS BORRABLES DE SOLO LECTURA (EPROMS) BASADO - EN UN MICROPROCESADOR 8748".

TESIS DE GRADO

PREVIA A LA OBTENCION DEL TITULO DE:

INGENIERO EN ELECTRICIDAD
ESPECIALIZACION: ELECTRONICA

PRESENTADA POR:

ANGEL RAFAEL LOAYZA ROMERO

GUAYAQUIL - ECUADOR

1.986

AGRADECIMIENTO

AL ING. SERGIO FLORES MACIAS
DIRECTOR DE TESIS, POR SU -
AYUDA Y COLABORACION PARA LA
REALIZACION DE ESTE TRABAJO.

ING. GUSTAVO BERMUDEZ FLORES
SUB-DECANO DE LA FACULTAD DE
INGENIERIA ELECTRICA

ING. SERGIO FLORES MACIAS
DIRECTOR DE TESIS

ING. JUAN DEL POZO LEMOS
MIEMBRO DEL TRIBUNAL

ING. NORMAN CHOOTONG CHING
MIEMBRO DEL TRIBUNAL

DEDICATORIA

- A MIS PADRES
- A MIS HERMANOS
- A MIS SOBRINOS

DECLARACION EXPRESA

"LA RESPONSABILIDAD POR LOS HECHOS, IDEAS Y DOCTRINAS EXPUESTOS EN ESTA TESIS, ME CORRESPONDEN EXCLUSIVAMENTE; Y, EL PATRIMONIO INTELECTUAL DE LA MISMA, A LA ESCUELA SUPERIOR POLITECNICA DEL LITORAL".

(REGLAMENTO DE EXAMENES Y TITULOS PROFESIONALES DE LA ESPOL).

ANGEL RAFAEL LOAYZA ROMERO

R E S U M E N

El presente trabajo consiste del diseño y construcción de un circuito programador de los siguientes Eproms: 2708,2716 (5v), 2732, 2732A y 2758. El diseño está basado en el microprocesador 8748. Las funciones del circuito son las de programar, leer la información grabada en el Eprom, comprobar si la programación del Eprom fue correcta, comprobar el borrado del Eprom y copiar entre dos tipos de Eproms. Se puede programar una o varias direcciones en forma secuencial o en forma aleatoria. El sistema tiene dos modos de operación; el modo independiente y el modo con interface. El primero consiste en manejar el sistema desde el teclado propio del circuito y el segundo consiste en manejar el sistema desde un microcomputador. La interface RS-232 es utilizada para la comunicación del circuito programador con un microcomputador. Esto le permite al usuario coger la información grabada en el Eprom y almacenarla en un archivo o también poder grabar el Eprom desde un archivo.

INDICE GENERAL

PAG.

PAG.

PAG.

PAG.

PAG.

I N T R O D U C C I O N

Los objetivos de este trabajo son los de desarrollar el hardware y el software de un sistema programador de 5 clases de Eproms (2708, 2716 (5v), 2732, 2732A y 2758), y desarrollar el hardware y el software para comunicación del sistema con un microcomputador a través de la interface RS-232. El microprocesador 8748 fue escogido para el diseño, debido a que es recomendable en el desarrollo de prototipos, por poseer una Eprom interna de 1024 palabras - de 8 bits que puede ser modificada una y otra vez por el usuario durante el desarrollo del sistema.

Todo el hardware del sistema es descrito en el capítulo II. En este se describen el Diagrama de Bloques y el direccionamiento de la memoria del sistema (memoria de programa y memoria - de datos) y de los dispositivos periféricos. Una extensión de 4096 palabras de 8 bits de memoria de datos es suficiente para cumplir con las funciones del sistema. La memoria de programa comprende 3070 palabras de 8 bits de las cuales solamente 2048 son utilizadas por los programas.

Los programas cuando el sistema opera en forma independiente es presentado en el Capítulo III. Primero se describe el Diagrama de Flujo general del sistema luego se presentan los Diagramas de Flujo de las rutinas de comando y de las rutinas de servicio.

El Capítulo IV está dedicado a hacer un estudio breve de la interface RS-232 y presentar los diagramas de flujo de las rutinas de comando y de las rutinas de servicio que el sistema utiliza para la comunicación con un microcomputador. Estos programas le permiten al usuario realizar todas las funciones que el sistema presenta cuando opera en forma independiente, además se puede almacenar la información grabada en el Eprom en un archivo o grabar el Eprom desde un archivo.

El manual del usuario es descrito en el Capítulo V. Todos los comandos cuando el sistema opera en forma independiente o con la interface RS-232 son descritos en este capítulo.

El programa que hace posible la comunicación del sistema con un microcomputador es listado en el Apéndice E. Este programa fue corrido en una computadora personal IBM. Se pueden diseñar programas para otros tipos de máquina tomando como referencia el programa presentado en esta tesis.

C A P I T U L O I

ANALISIS DE UN SISTEMA PROGRAMADOR DE EPROMS

1.1. INTRODUCCION

En este capítulo presentaremos en forma breve, la definición de un programador de Eproms, la forma como queda almacenada la información en las memorias Eproms una vez que dicha información ha sido previamente grabada, y por último describiremos como una introducción al trabajo presentado en este escrito un diagrama de bloques general de un programador de Eproms. Pasaremós revista también a la función que cumple el pulso de programación de 25 voltios y la técnica de borrado de las memorias Eproms.

A continuación se verá los pasos que deben seguirse para grabar o leer información de la memoria Eprom, basándonos en el diagrama de bloques expuesto en este Capítulo. El programador de Eproms diseñado en este trabajo tendrá la capacidad de comunicación con un microcomputador

a través de una interface RS-232. La finalidad de esto es recoger la información grabada en el Eprom y almacenarla en un archivo o viceversa.

1.2. QUE ES UN PROGRAMADOR DE EPROMS

Un programador de Eproms se puede definir como un circuito capas de cambiar los unos por ceros de las memorias Eproms, además debe tener la capacidad de leer la información almacenada en las mismas.

Este circuito debe generar los voltajes necesarios para que la programación de la memoria sea correcta. Entre estos voltajes tenemos el pulso de programación de 25V y la señal de selección de escritura de 12V para el Eprom 2708.

1.3. ALMACENAMIENTO DE LA INFORMACION EN LAS MEMORIAS EPROMS

Las memorias Eproms almacenan los bits de datos en celdas formadas de transistores de carga almacenada. Tales transistores son similares a los transistores de efecto de campo de canal positivo - puerta de silicio pero con dos puertas como se muestra en la figura Nº 1.1.

La puerta más baja o "flotante" es completamente circundada

por una capa de aislante de bióxido de silicio y la puerta más alto "control" o "selección" es conectada al circuito externo.

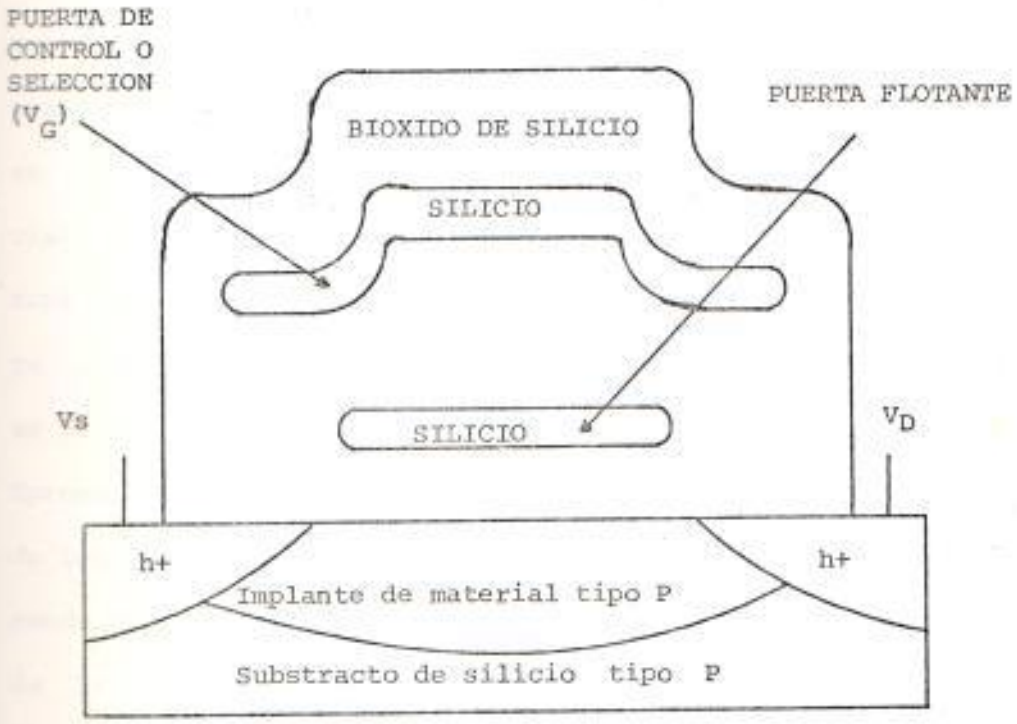


FIGURA Nº 1.1. DIAGRAMA DE UNA CELDA MOSTRANDO LAS DOS PUERTAS LA CAPA DEL AISLANTE Y EL CANAL FUENTE-DRENAJE.

La cantidad de carga eléctrica almacenada en la puerta flotante determina si la celda contiene un uno o un cero . Celdas cargadas son leídas como ceros, celdas descargadas -

son leídas como unos. Cuando la memoria Eprom viene de la fábrica todas las celdas son borradas de carga y son leídas como uno lógico, cada palabra contiene el data hexadecimal FF.

Cuando una celda dada va a ser cambiada de uno a cero, una corriente es pasada a través del canal del transistor de la fuente a la puerta. (Los electrones, se mueven de modo opuesto). Al mismo tiempo un potencial de alto voltaje es aplicado en la puerta de control del transistor, creando un campo eléctrico fuerte cerca de las capas de material semiconductor. (Esta es la función del pulso de programa aplicado a las memorias - Eproms). En presencia de este campo eléctrico fuerte, algunos de los electrones pasan a través del canal fuente-drenaje ganando suficiente energía para abrir un tunel a través de la capa de aislante que normalmente aisla la puerta flotante.

Los electrones acumulados en la puerta flotante hacen que esta tome una carga negativa, lo cual hace que la celda contenga un cero.

Cuando el dato va a ser borrado de la memoria Eprom, esta es expuesta a la luz ultravioleta la cual contie

ne fotones de energía relativamente alta. Los fotones - incidentes excitan los electrones en la puerta flotante a estados de energía suficientemente alta que pueden abrir un tunel de regreso a través de la capa aislante, removiendo la carga de la puerta y retornando la celda al estado lógico uno.

1.4. DIAGRAMA DE BLOQUES GENERAL DE UN PROGRAMADOR DE EPROMS.

La figura Nª 1.2. muestra el diagrama de bloques de un circuito programador de Eproms, que puede ser adaptado a cualquier sistema. Las puertas de salida 1 y 2 suministran las direcciones al Eprom, la puerta de salida número 3 alimenta el dato a ser programado. La puerta de salida número 4 junto con la red resistiva proveen los pulsos de programación para los Eproms. Para leer datos de los Eproms tenemos la puerta de entrada número 1.

El módulo de personalidad es un circuito adicional usado para programar un tipo específico de Eprom.

Cada una de las puertas está conformada por un agarrador 8212. Esta es una puerta de entrada - salida de propósito general. La salida del agarrador es de tres - estados, si el modo de entrada es alto, las salidas - son siempre habilitadas, cuando el dispositivo es selec-

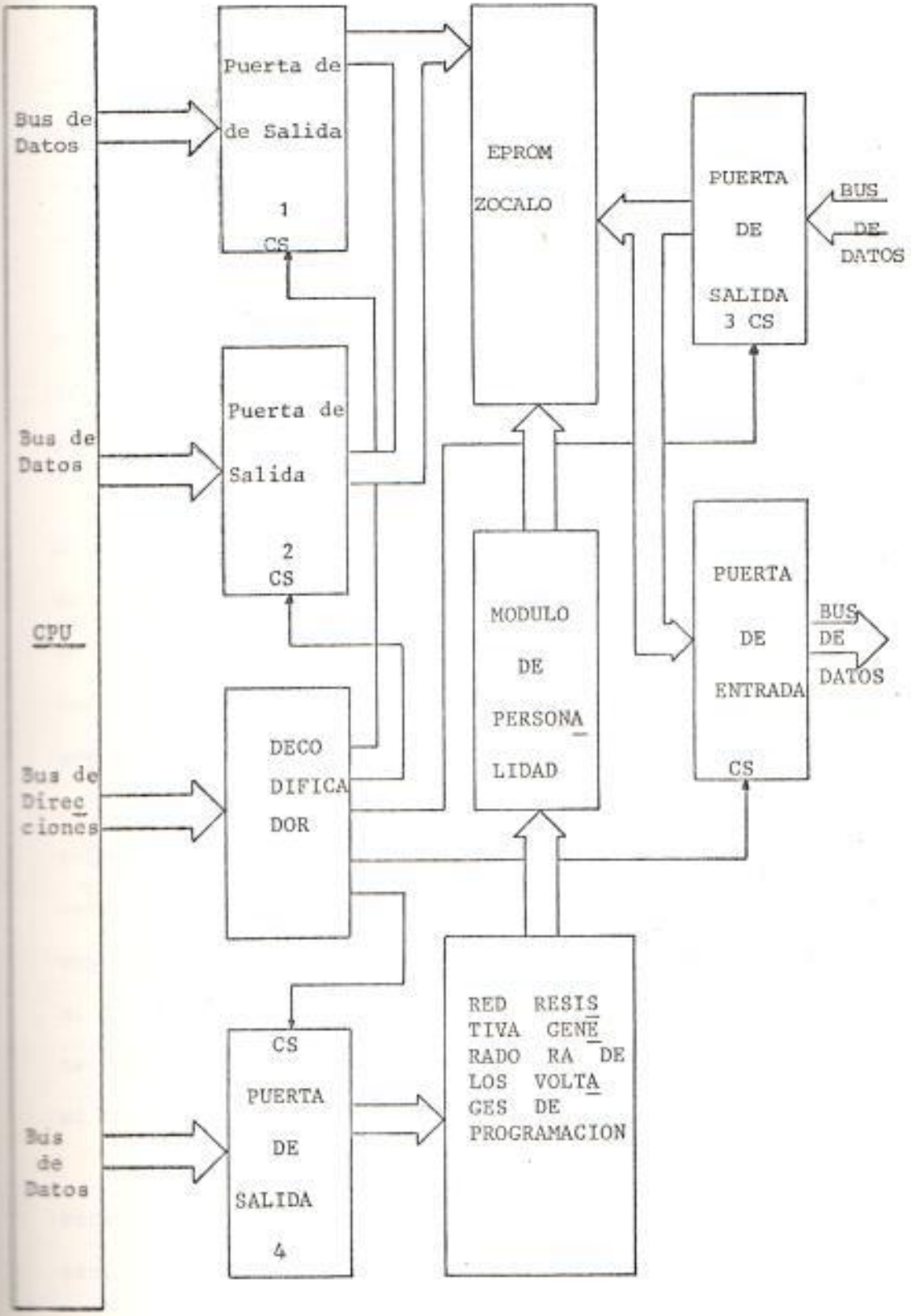


FIG: 1.2. DIAGRAMA DE BLOQUES DE UN PROGRAMADOR DE EPROMS.

cionado por un nivel bajo en DS1 y un nivel alto en DS2, cada vez que el dato está presente en las líneas de entrada, este es agarrado y posteriormente aparece en las líneas de salida . Si el modo de entrada es bajo, las salidas están en estado de alta impedancia hasta que el dispositivo sea seleccionado.

Supongamos que deseamos grabar información en cierto tipo de Eprom, primero escogemos el módulo de personalidad correcto para el Eprom que se va a programar.

El software debe estar diseñado de tal forma que el bus de datos contenga el dato a ser escrito en cada una de las puertas y el bus de direcciones deberá contener la dirección para seleccionar la puerta de tal forma que el dato sea agarrado por la puerta.

Primero los 8 bits de dirección de orden más bajo deberán ser escritos a la puerta número 1, luego los bits de orden más alto deberán ser escritos a la puerta número 2 , el dato a ser programado deberá ser escrito a la puerta número 3, por último la palabra de control adecuada al tipo de Eprom deberá ser escrita a la puerta número 4.

Para el proceso de lectura el procedimiento es similar al anterior, las palabras de dirección de orden más bajo y alto de

berán ser escritas a sus respectivas puertas 1 y 2, luego se envía la palabra de control para leer el Eprom a la puerta número 4, finalmente se lee la puerta de entrada número 1.

C A P I T U L O II

DISEÑO DEL HARDWARE DEL SISTEMA

2.1. INTRODUCCION

En este Capítulo presentaremos las condiciones que debe cumplir el sistema y el diseño del hardware del sistema. Analizaremos cada uno de los circuitos e interfaces del diagrama de bloques. El diseño está basado en el microprocesador 8748 de la firma Intel, la información acerca de este microprocesador será cubierta en el Apéndice B, de este escrito. Definiremos dos tipos de memoria, la memoria de programa y la memoria de datos, la primera es de tipo EPROM y la segunda de tipo RAM . Por último describiremos la forma de direccionamiento de la memoria de programa y de datos del sistema,y el direccionamiento de los dispositivos periféricos que conforman el sistema.

2.2. DEFINICION GENERAL DEL SISTEMA

Diseñar un sistema programador de los siguientes EPROMS:2708,

2716(5V), 2732, 2732A y 2758, utilizando un microprocesador 8748. El sistema debe tener la capacidad de comunicarse con un microcomputador.

El sistema debe verificar si la región donde se va a grabar está completamente borrada, en caso de no estarlo el sistema mostrará en el visualizador la dirección y la información almacenada en esa dirección. La región está indicada por la dirección inicial y final.

Para grabar información en cualquier EPROM, primero debemos colocar el módulo de personalidad adecuado, el paso siguiente es cargar la memoria del sistema con la información que deseamos grabar en el Eprom. La información a grabar puede comenzar en cualquier dirección y finalizar en cualquier dirección.

Una vez que la información deseada ha sido grabada en el Eprom el sistema debe ser capaz de verificar si la grabación fue correcta, si esta es correcta enviar al visualizador un caracter que indique que la grabación fue correcta, en caso de que la grabación fuese incorrecta en el visualizador aparecerá - la dirección y la información almacenada en esa dirección.

El sistema debe tener la capacidad de leer Eproms y copiar entre dos Eproms cualesquiera de los mencionados anteriormente .

Cuando leemos un Eprom la información almacenada en este debe ser pasada a la memoria de datos, luego puede ser leída mediante un comando. Para el caso de copiar entre dos Eproms, primero leemos la información almacenada en el Eprom que se va a copiar, luego colocamos el Eprom en el cual se va a realizar la copia. Finalmente se procede a grabar el Eprom.

2.3. DIAGRAMA DE BLOQUES Y DESCRIPCION DEL MISMO

La figura Nº 2.1., muestra un diagrama de bloques del sistema, en el se muestran los circuitos utilizados para cada tipo de interface. Antes de hacer una descripción del diagrama de bloques, el lector deberá estar familiarizado con las características de funcionamiento del microprocesador 8748, esta información se la podrá encontrar en el Apéndice B.

Haremos referencia a dos tipos de memoria, la memoria de programa y la memoria de datos. La memoria de programa es la memoria donde están almacenadas las instrucciones y las constantes del programa. La memoria de datos es la memoria a la cual el usuario podrá leer, cambiar su contenido o podrá cargar la información que posteriormente será grabada en un Eprom.

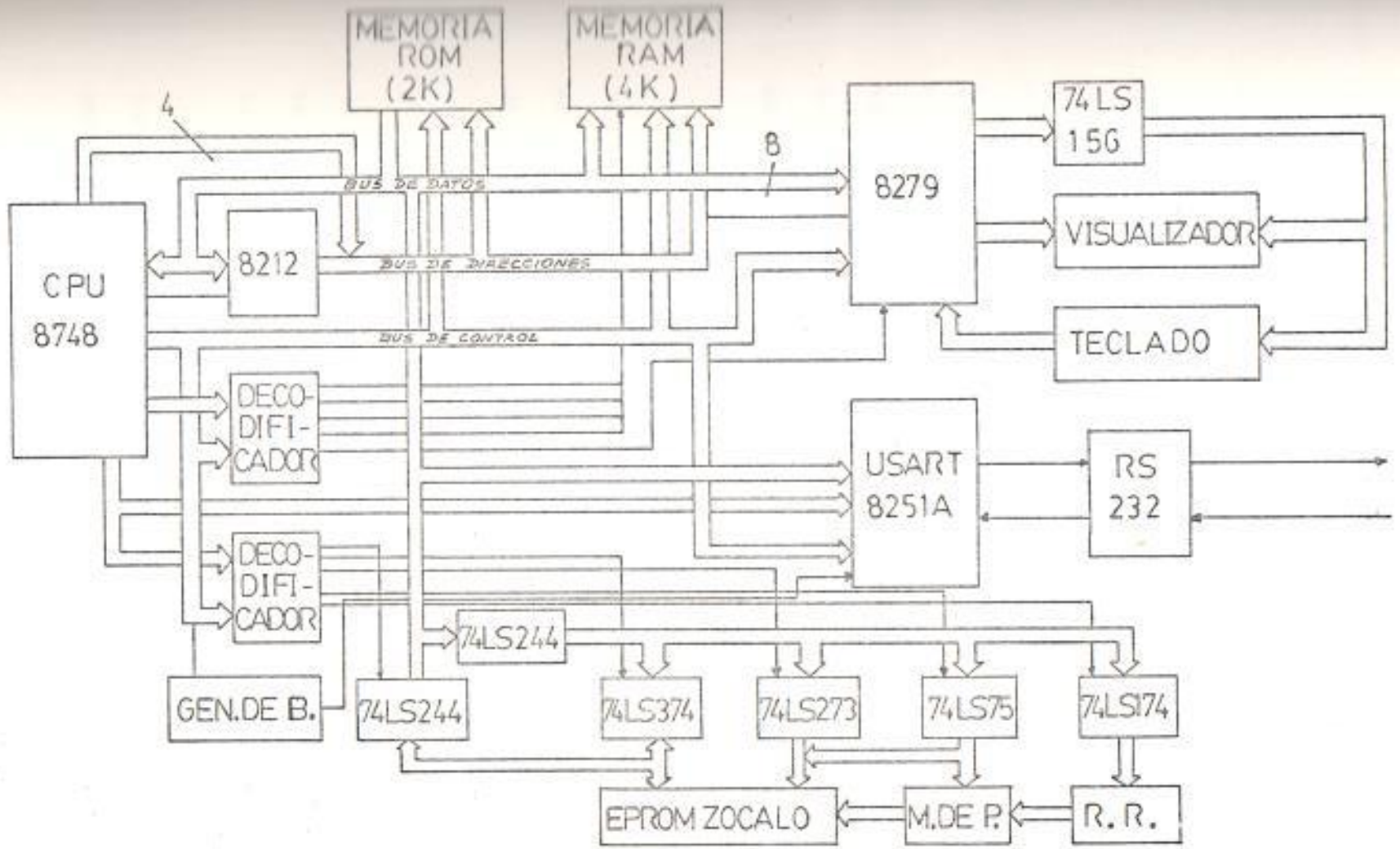


FIG. Nº 2.1 DIAGRAMA DE BLOQUES DEL SISTEMA

El microprocesador 8748 tiene tres puertos, puerto 0 ó bus de datos multiplexado, puertos 1 y 2. Además tiene 1024 palabras de memoria (EPROM) de programa interna y 64 registros de memoria (RAM) interna. La memoria de programa está dividida en secciones de 256 palabras, a cada una de estas secciones se las denomina páginas, por ejemplo la página 0 estará comprendida entre las direcciones 00-FF, la página 1 estará comprendida entre las direcciones 100-1FF., etc. A la memoria de programa o memoria de datos que se encuentre fuera del microprocesador 8748, la llamaremos memoria de programa externa o memoria de datos externa, respectivamente.

Refiriéndonos al diagrama de bloques observamos que el sistema utiliza los dos tipos de memoria externa, de programa y de datos.

El microprocesador 8748 tiene un contador de programa de 12 bits, logrando con esto direccionar directamente 4096 palabras de memoria de programa, 1024 palabras de memoria interna y 3072 palabras de memoria externa.

Para conectar el microprocesador con los dos tipos de memoria externa, es necesario demultiplexar la barra

de datos, esto lo hacemos a través de un agarrador de dirección 8212, el cual separa el bus de dirección del bus de datos, logrando de esta forma 8 líneas de dirección, las 4 líneas restantes están - presentes en los 4 bits de orden más bajo del puerto 2. El sistema utiliza 2048 palabras de memoria de programa externa, la cual está conformada por un - Eprom 2716.

Para direccionar 4096 palabras de memoria de datos - externa, en bancos de 1024 palabras cada uno, necesitamos 10 líneas de dirección, para lograr esta configuración utilizamos las 8 líneas de dirección que resultan de demultiplexar el bus de datos, más las líneas 0 y 1 del puerto 2. La selección de cada uno de los bancos de memoria lo hacemos a través de un decodificador 3/8, el cual decodifica 3 líneas de orden más alto del puerto 2. El decodificador es habilitado solamente cuando se va a leer o escribir la memoria de datos externa.

Cada banco de la memoria de datos externa está conformado por dos memorias RAM estáticas 2114 (1024 4 bits), las cuales emplean una celda para almacenar cada bit de información. Es un dispositivo asincrono que no necesita reloj. Su contenido permanece estable indefinidamen

te, mientras no se suprima la tensión de alimentación.

El circuito de interfase utilizado para la comunicación entre el microprocesador 8748 con un teclado y un visualizador, es el controlador 8279 (Interface a teclado/Visualizador programable).

Para más información acerca de este circuito ver Apéndice C.

Para habilitar el 8279 utilizamos una línea de salida del decodificador de memoria de datos externa.

El teclado está compuesto de 24 teclas, 16 teclas hexadecimales, 5 teclas de comandos, una tecla para inicializar el sistema y dos teclas de información. El visualizador está conformado por 5 visualizadores LED de 7 segmentos, de los cuales 3 conforman el campo de dirección y los otros dos el campo de datos.

La interface utilizada para programar o leer los Eproms (Ver figura Nº 2.1), está conformada por dos circuitos integrados 74LS374 y 74LS273, los cuales contienen 8 celdas tipo D (El 74LS374 tiene salidas de tres estados), el pri

mero se utiliza para enviar el dato que se va a grabar en el Eprom, por el segundo circuito se envían los 8 bits de menor peso de la dirección en la cual se va a almacenar el dato. Los 4 bits de mayor peso de la dirección se envían a través de un 74LS75, el cual contiene 4 celdas tipo D y para enviar la palabra de control la interface utiliza un circuito integrado 74LS174 , el cual contiene 6 celdas tipo D.

Para leer la información almacenada en la memoria Eprom se utiliza un circuíto integrado 74LS244, este es un circuito de entrada/salida, con salidas de tres estados. La interface también está conformada por una red resistiva cuya función es la de generar los voltajes necesarios para la programación del Eprom. Un módulo de personalidad se suma también a la interface, el cual conecta los voltajes de programación de forma adecuada al tipo de Eprom - que se va a grabar.

Para habilitar cada uno de estos circuitos utilizamos - un decodificador de 3/8, el cual decodifica tres líneas - del puerto 1. Un circuito integrado 74LS138 es adecuado para esta función.

El circuito de interface utilizado para la comunicación -

del sistema con un microcomputador es un USART (Receptor/Transmisor - Asíncrono/Sincrono), diseñado para la transmisión de datos en serie en sistemas de microcomputadores, se usa como un dispositivo periférico y se programa por la CPU. Para utilizar cualquier técnica de transmisión - de datos en serie, el USART acepta caracteres de datos de la CPU en formato paralelo y los convierte en datos en serie.Simultáneamente puede recibir datos en serie y convertirlos para la CPU en formato paralelo. La selección del USART está controlada por la puerta 1 del microprocesador 8748 . Para más información acerca de las características de funcionamiento del USART, referirse al Apéndice C.

Además de la conversión paralelo a serie o viceversa, se necesita un circuito adicional para lograr la conversión de nivel ya sea para un lazo de corriente de 20 mA ó para una conexión RS-232. El sistema utiliza la conexión RS-232, la cual está conformada por dos circuitos, los cuales son - los encargados de la conversión de nivel requerida por la interface, RS-232. Más información acerca de esta interface referirse al Capítulo IV.

2.4. DIRECCIONAMIENTO DE LA MEMORIA DEL SISTEMA

2.4.1. Memoria de programa externa

Como habíamos dicho anteriormente la memoria de

programa puede ser expandida más allá de las 1024 palabras de memoria interna residente en el microprocesador 8748, El sistema utiliza una expansión de 2048 palabras de memoria de programa externa.

La figura Nº 2.2., muestra el mapa de memoria de programa del sistema, en el mapa se puede distinguir la memoria de programa interna y la memoria de programa externa. La memoria de programa está dividida en bancos de 2048 palabras cada uno. Cada banco a su vez está dividido en 8 páginas de 256 palabras cada uno. Toda busqueda de la memoria de programa para las direcciones menores que 1024 ocurre internamente sin generar señales externas. Cuando la busqueda es de una dirección mayor que 1024, automáticamente se inicia una busqueda de la memoria de programa externa. Cuando esto ocurre tenemos que el contenido del contador de programa sería sacado en el bus de datos y en los cuatro bits de orden más bajo del puerto 2, luego la señal de habilitación del agarrador de dirección (ALE), indicaría el tiempo en el cual la dirección es válida. A continuación tenemos que la señal de habilitación de la memoria de programa externa, (PSEN) indica que una busqueda de una

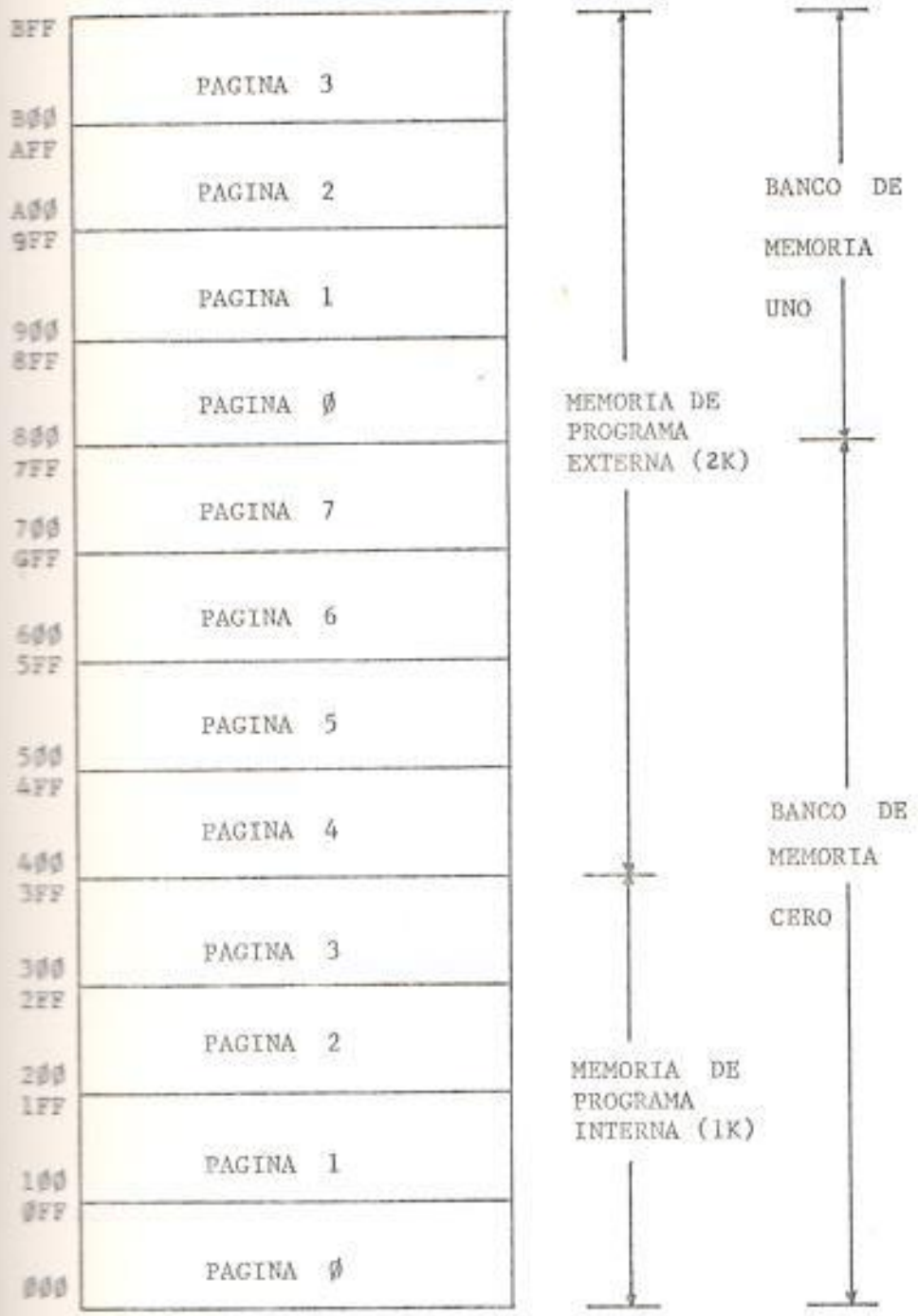


FIG: 2.2. MAPA DE LA MEMORIA DE PROGRAMA DEL SISTEMA.

instrucción externa está en progreso, por último el bus de datos vuelve al modo de entrada y el procesador acepta los 8 bits de información almacenados em esa dirección.

Para programas de 2048 palabras o menos, el - direccionamiento sería en la forma descrita anteriormente. La figura Nº 2.3., muestra los diagramas de tiempo de las señales que se generan en el caso de que una busqueda de una instrucción externa ocurra.

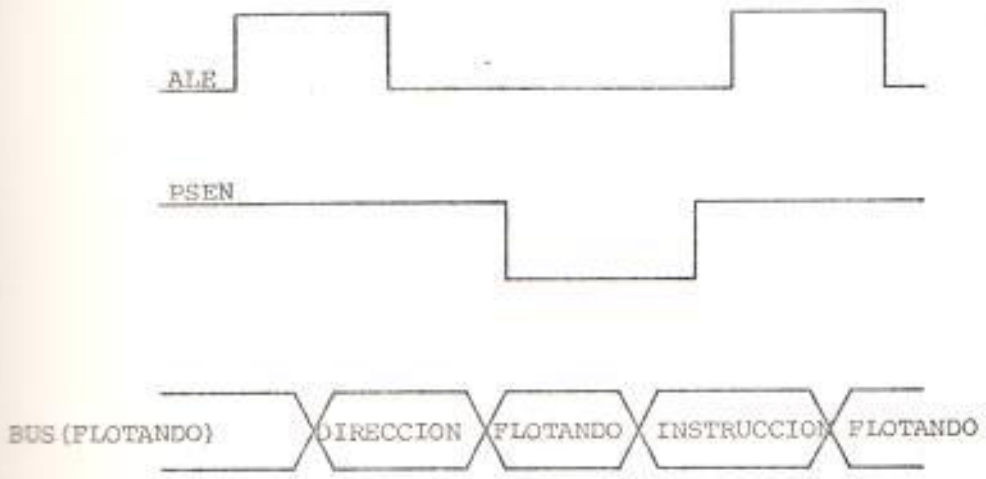


FIGURA Nº- 2.3. BUSQUEDA DE UNA INSTRUCCION DE LA MEMORIA DE PROGRAMA EXTERNA.

Direcciones más allá de 2048 pueden ser alcanza

das ejecutando una instrucción de switcheo de - banco (SEL MB0, SEL MB1) seguida por una instruc ción de salto (JMP o CALL). La característica de switcheo de banco permite saltar de un banco a otro o viceversa. El switcheo de bancos de memoria de 2048 palabras se lo realiza colocando un cero o un uno al bit más significativo del contador de programa (bit 11). El bit 11 no es alterado por el normal incremento del contador de programa pero es cargado con el contenido de una celda espe cial cada vez que una instrucción JMP o CALL es - ejecutada. Esta celda es colocada a uno por la ins trucción SEL MB1 y colocada a cero por la instrucción SEL MB0.

La figura Nº 2.4., muestra el contador de programa - del microprocesaodr 8748. Los bits 10 y 11 del contador de programa se utilizan junto con la señal PSEN para habilitar la memoria de programa externa. La tabla Nº 2.1., muestra la combinación de los bits 10 y 11 para habilitar la memoria de programa externa. Cuando los bits 10 y 11 son cero, el contador de programa direcciona la memoria de programa interna y la señal PSEN no se genera, con la segunda combinación habilitamos las primeras 1024 palabras de me

moria de programa externa, y con la tercera combinación habilitamos las siguientes 1024 palabras - de memoria externa, en estas dos últimas combinaciones la señal PSEN si esta presente. La última combinación no se utiliza en este caso, debido a que solamente se utilizan 2048 palabras de memoria de programa externa.

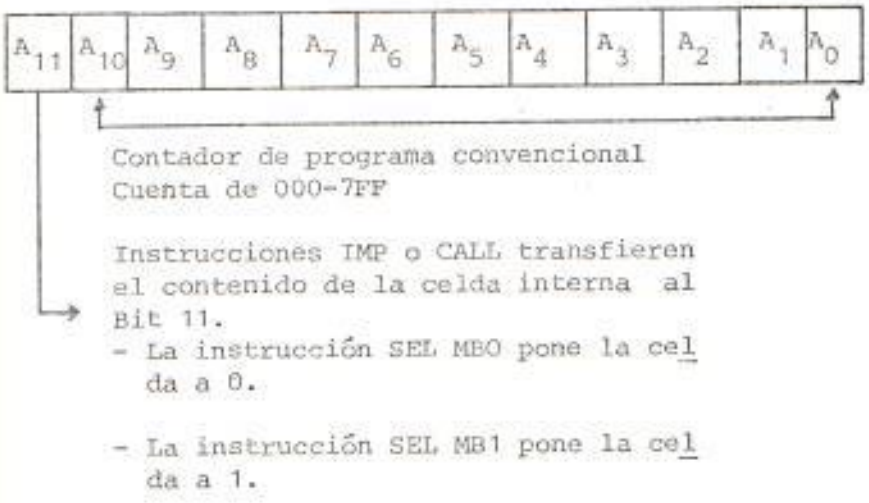


FIGURA Nº 2.4. CONTADOR DE PROGRAMA DEL MICROPROCESADOR 8748.

TABLA 2.1.

COMBINACION DE LOS BITS 10 y 11 PARA HABILITAR LA MEMORIA DE PROGRAMA EXTERNA

A11	A10	HABILITACION
0	0	Memoria de programa interna.
0	1	Primeras 1024 palabras de memoria externa
1	0	Las siguientes 1024 palabras de memoria externa.
1	1	No se utiliza

2.4.2. Memoria de datos externa

Para una mejor comprensión de la memoria de datos externa la dividiremos en bancos de 1024 palabras cada uno, cada banco a su vez estará dividido en 4 páginas de 256 palabras cada página.

El sistema tiene 4096 palabras de memoria de datos externa, sin incluir las 64 palabras de memoria de datos interna. La memoria de datos externa es accesible via las instrucciones MOVX A,#Rj

y MOVX # Rj A. Las cuales transfieren 8 bits de datos entre el acumulador y la localidad de memoria direccionada por los registros Rj (j=0,1). Esto permite - direccionar directamente 256 palabras de memorias de datos externa o lo que es equivalente a una - página de memoria. Para el caso de direccionar más de una página de memoria lo hacemos con líneas de salida extras del microprocesador 8748.

La figura Nº 2.5., muestra el mapa de la memoria de datos interna y externa del sistema. En este Capítulo nos ocuparemos solo de la memoria de datos externa , la información acerca de la memoria de datos interna - referirse al Apéndice B.

Para habilitar cada banco de memoria, decodificamos las líneas 4,5 y 6 del puerto 2 y para habilitar cada página de memoria de cualquier banco utilizamos - las líneas 0 y 1 del puerto 2. El decodificador de - memoria solo será seleccionado cuando se ejecuten una de las dos instrucciones con las cuales se tiene acceso a la memoria de datos externa.

La tabla Nº 2.2., muestra la combinación de las líneas 4, 5 y 6 del puerto 2, para habilitar cada banco de

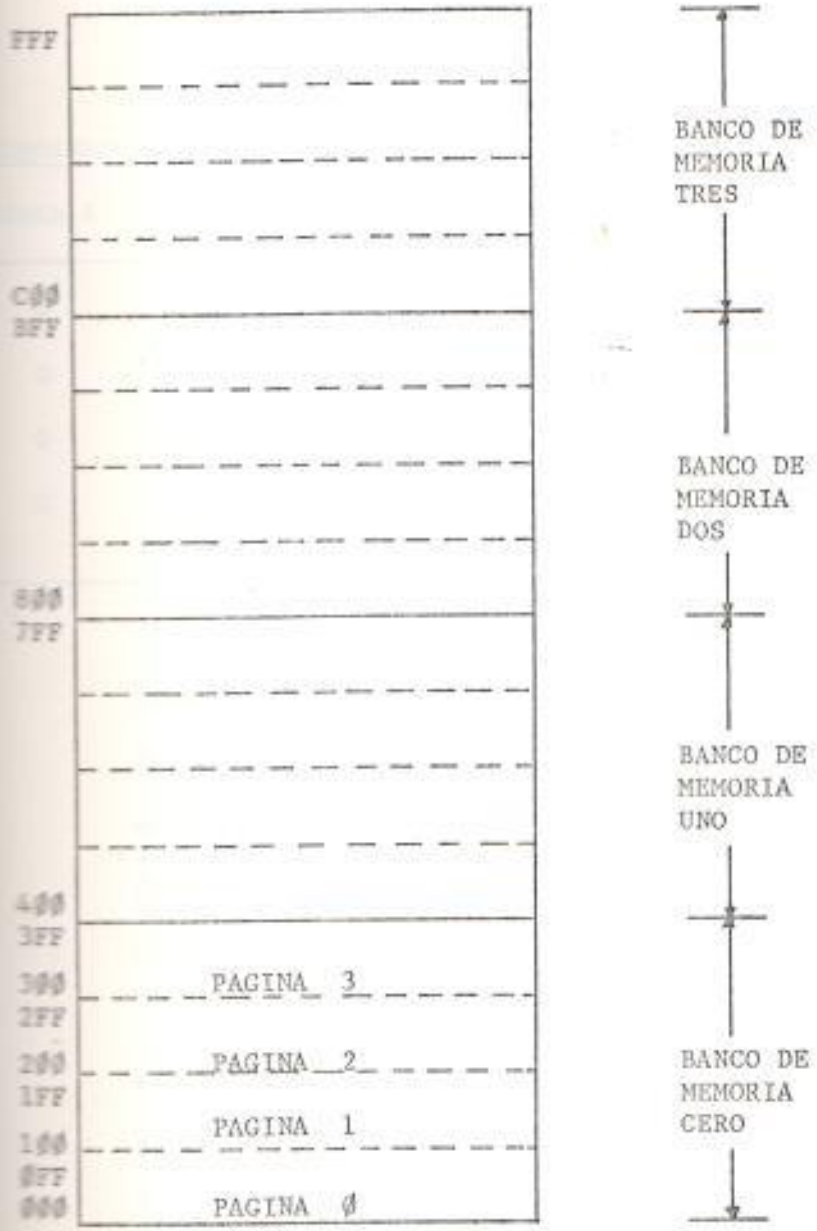


FIG: 2.5. MAPA DE LA MEORIA DE DATOS DEL SISTEMA

memoria de datos externa.

TABLA Nª 2.2.

COMBINACION DE LAS LINEAS 4, 5 Y 6 DEL PUERTO 2

LINEA 6	LINEA 5	LINEA 4	BANCO DE MEMORIA
0	0	0	Banco de memoria 0
0	0	1	Banco de memoria 1
0	1	0	Banco de memoria 2
0	1	1	Banco de memoria 3

Con la combinación 0 direccionamos las posiciones de memoria 000-3FF, con la combinación 1 direccionamos las posiciones 400-7FF, con la combinación 2 direccionamos las posiciones 800-BFF y con la combinación 3 direccionamos las posiciones COO-FFF.

Para cruzar de una página de memoria a otra - dentro de un mismo banco, lo hacemos a través de las líneas 0 y 1 del puerto 2, los cuales equivalen a las líneas de dirección A8 y A9, respectivamente.

La tabla Nª 2.3., muestra la combinación de las líneas 0 y 1 del puerto 2, para direccionar cada una de -

las páginas de memoria del correspondiente banco de memoria de datos.

TABLA Nº2.3.

COMBINACION DE LAS LINEAS 0 y 1 DEL PUERTO 2

LINEA 1	LINEA 0	PAGINA
0	0	Página 0
0	1	Página 1
1	0	Página 2
1	1	Página 3

La tabla Nº 2.4., resume el direccionamiento de cada una de las páginas de memoria en sus respectivos bancos de memoria.

En la siguiente página podemos apreciar la tabla Nº 2.4.

TABLA Nº 2.4.

DIRECCIONAMIENTO DE LAS PAGINAS DE MEMORIA EN SUS RESPECTIVOS BANCOS DE MEMORIA

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Banco	Página
0	0	0	0	0	0	0	0	0	0 (000-0FF)
0	0	0	0	0	0	0	1	0	1 (100-1FF)
0	0	0	0	0	0	1	0	0	2 (200-2FF)
0	0	0	0	0	0	1	1	0	3 (300-3FF)
0	0	0	1	0	0	0	0	1	0 (400-4FF)
0	0	0	1	0	0	0	1	1	1 (500-5FF)
0	0	0	1	0	0	1	0	1	2 (600-6FF)
0	0	0	1	0	0	1	1	1	3 (700-7FF)
0	0	1	0	0	0	0	0	2	0 (800-8FF)
0	0	1	0	0	0	0	1	2	1 (900-9FF)
0	0	1	0	0	0	1	0	2	2 (A00-AFF)
0	0	1	0	0	0	1	1	2	3 (B00-BFF)
0	0	1	1	0	0	0	0	3	0 (C00-CFF)
0	0	1	1	0	0	0	1	3	1 (D00-DFF)
0	0	1	1	0	0	1	0	3	2 (E00-EFF)
0	0	1	1	0	0	1	1	3	3 (F00-FFF)

2.5. DIRECCIONAMIENTO DE LOS DISPOSITIVOS PERIFERICOS

2.5.1. Direccionamiento del circuito de interface a teclado y visualizador (8279)

Este circuito de interface es escrito o leido como una posición de memoria datos externa, para tener acceso a este circuito de interface lo hacemos a través de las instrucciones de lectura o escritura de la memoria de datos externa , (MOV A, Rj y MOV Rj, A, j=0 y 1).

Para direccionar este circuito de interface no es necesario el valor de RO ó R1. Este circuito de interface se habilita a través del decodificador de memoria.

La tabla 2.5., muestra las dos palabras de control que deben ser escritas previamente al puerto 2, para comunicarse con el periférico.

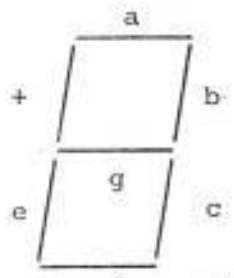
De la tabla observamos que el bit 0 del puerto 2, le indica al circuito de interface cuando son datos o comandos lo que se va a leer o escribir.

La figura N°- 2.6., muestra la configuración de bits para cada uno de los visualizadores, para cada segmento tenemos un bit más un bit para el punto decimal.

TABLA 2.5.

PALABRAS DE CONTROL PARA COMUNICARSE CON EL CIRCUITO DE INTERFACE 8279

PUERTO 2								LEER / ESCRIBIR	COMANDOS O DATOS
7	6	5	4	3	2	1	0		
0	1	0	0	0	0	0	0	Leer	Leer carácter del teclado.
								Escribir	Escribir datos al visualizador.
0	1	0	0	0	0	0	1	Leer	Leer palabra de estados.
								Escribir	Escribir Comandos.



• P.d(punto decimal)

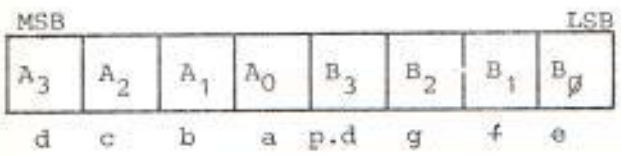


FIGURA Nº-2.6.CONFIGURACION DE BITS PARA CADA VISUALIZADOR

El hardware está configurado de tal forma que escribiendo un cero a cada bit encendería el correspondiente segmento.

EJEMPLO:

Un "2" sería representado como :

0100 1010 = 4A (Hexadecimal)

Existen cinco visualizadores, estos están configurados como muestra la figura Nº 2.7. Tres de estos conforman el campo de dirección y los otros dos el campo de datos.

FIGURA Nº 2.7. CONFIGURACION DEL VISUALIZADOR

2.5.2. Direccionamiento del circuito de interface para la comunicación con un microcomputador (USART)

Este circuito de interface (USART), es una interfa-

ce simple de conectar con un microprocesador 8748. El USART requiere de un reloj de alta velocidad (CLK), una señal de inicialización (RESET), reloj de recepción y transmisión de datos (R x C y T x C) y datos para operar.

El reloj de alta velocidad se lo obtiene conectando la señal CLK del USART al terminal de entrada/salida TO,del microprocesador 8748. Este terminal puede ser usado como entrada y probado bajo control del programa o puede ser usado como salida, en el cual tendremos una señal de reloj de 1/3 la frecuencia del cristal. Hay que tener presente que cuando TO ha sido designado por software para que sea un terminal de salida, la señal de reloj estará presente siempre excepto cuando el sistema es inicializado.

La señal de inicialización del USART (RESET) es obtenido programando el bit 5 del puerto 1. Esto fue necesario hacer para inicializar el USART bajo control del programa por dos razones: la primera es que el microprocesador 8748 no suministra una señal de inicialización a otros dispositivos, la segunda razón es que el USART requiere la presencia de la señal de reloj (CLK) durante el tiempo que la señal de inicialización está presente.

El USART al igual que el circuito anterior, es leido o escrito como una posición de memoria de datos externa. Las instrucciones disponibles para hacer esto son - las mismas con las cuales se tiene acceso a la memoria de datos externa. La primera es MOVX#Rj, a la cual almacena el contenido del acumulador en la posición de la memoria de datos externa direccionada - por el registro j(j = 0 o 1) y la segunda instrucción es MOVX A,#Rj la cual almacena el dato de la posición de memoria de datos externa dentro del acumulador. Sin embargo es posible direccionar el USART sin usar RO y R1. Esta configuración se logra conectando la señal de selección del USART (CS) al bit 4 del puerto 1, similarmente conectamos la señal del USART (C/D) al bit 0 del puerto 1. La secuencia para tener acceso al - USART se logra primero habilitando el periférico - (bit 4=0) colocar el bit 0 del puerto 1 al estado apropiado, y finalmente se utiliza la instrucción MOVX para ejecutar la operación apropiada y luego se deshabilita el USART (bit 4 = 1).

Las dos palabras de control que deben ser escritas previamente para comunicarse con el USART se muestran en la tabla 2.6.

TABLA 2.6.

PALABRAS DE CONTROL PARA COMUNICARSE CON EL USART

PUERTO 1								LEER / ESCRIBIR	FUNCION
7	6	5	4	3	2	1	0		
0	0	0	0	1	1	1	0	Leer	Leer datos del USART.
								Escribir	Escribir datos al USART.
0	0	0	0	1	1	1	1	Leer	Leer estados del USART
								Escribir	Escribir comandos al USART.

2.5.3. Direccionamiento de la interface para leer o escribir un EPROM

Los circuitos de esta interface son leidos o escritos como una posición de memoria de datos externa. El direccionamiento de cada uno de estos circuitos se lo obtiene decodificando las líneas 0, 1 y 2 del puerto 1. El decodificador solo es habilitado cuando se ejecuta una de las instrucciones con las cuales se tiene acceso a la memoria de datos externa.

La tabla 2.7., muestra las palabras de control que deben ser escritas al puerto 1, para direccionar cada -

uno de los circuito de esta interface.

TABLA 2.7.

PALABRAS DE CONTROL PARA DIRECCIONAR CADA UNO DE LOS CIRCUITOS DE LA INTERFACE AL EPROM

PUERTO 1								CIRCUITO HABILITADO
7	6	5	4	3	2	1	0	
0	0	0	1	0	0	0	0	Registro de entrada de datos
0	0	0	1	0	0	0	1	Registro de salida de datos
0	0	0	1	0	0	1	0	Registro de salida de los 8 bits de dirección de menor peso.
0	0	0	1	0	0	1	1	Registro de salida de los 4 bits de dirección de mayor peso.
0	0	0	1	0	1	0	0	Registro de salida de la palabra de control.

C A P I T U L O III

SOFTWARE DEL SISTEMA

3.1. INTRODUCCION

En este capítulo se definirá los registros de la memoria de datos interna reservados por el software del sistema, de modo similar definiremos los símbolos usados en los diagramas de flujo. Posteriormente veremos una descripción general de todo el sofware basándonos en un diagrama de flujo simplificado. Después describiremos cada una de las rutinas con las cuales se da servicio a cada uno de los comandos que se usan en el software. En la última sección de este capítulo se describirá la función de cada una de las rutinas de servicio que conforman el software del sistema.

3.2. REGISTROS DE LA MEMORIA DE DATOS INTERNA RESERVADOS POR EL SOFTWARE DEL SISTEMA

La memoria de datos interna está formada de 64 registros, los cuales están divididos en dos bancos de 8

registros cada uno, una pila de 16 registros y 32 - registros de uso general. Algunos de estos registros de uso general son reservados por el software del sistema.

La tabla N°- 3.1., muestra cada uno de los 64 registros, al lado derecho de cada registro se ilustra la información almacenada en ese registro.

TABLA 3.1.

REGISTROS RESERVADOS POR EL SOFTWARE

REGISTRO		C O N T E N I D O
DECIMAL	HEX	
0	0	Dirección de los caracteres a ser sacados
1	1	Registro de uso general
2	2	Bandera de campo (BNDCP)
3	3	Bandera de punto (BNDPT)
4	4	Bandera de memoria (BNDMM)
5	5	Registro de uso general
6	6	Registro de uso general
7	7	Registro de uso general
8-23	8-17	Pila (8 niveles de rutinas)
24	18	Registro de uso general
25	19	Valor del mensaje a sacar

26	1A	Bandera de lectura/escritura (BNDRW)
27	1B	Direccione. donde se almacena tipo de Eprom y direcciones.
28	1C	Registro de uso general
29	1D	Valor del comando a utilizar
30	1E	Bandera para obtener dato o dirección (BNDDD)
31	1F	Registro de uso general
32	20	Registro imagen del acumulador
33	21	Bandera para sacar dirección y dato al visualizar o al microcomputador. (BNDVT)
34	22	Tipo de Eprom.
35-36	23-24	Dirección inicial del Eprom (DIEPR)
37-38	25-26	Dirección final del Eprom (DFEDR)
39-40	27-28	Dirección final de la RAM (DFRAM)
41-42	29-2A	Dirección inicial de la RAM (DIRAM) o dirección común (ADDCM)
43	2B	Dato común (ADDTC)
44	2C	Palabra de control (ACNTL)
45	2D	Buffer de entrada (BUFIN)
46-48	2E-30	Buffer de salida (ADBFO)
49	31	Byte de dirección de orden más alto (ADBHG)
50-51	32-33	Registro imagen de la dirección inicial de la RAM (DIRIM)
52-53	34-35	Registro imagen de la dirección inicial del Eprom (DIEIM).

A continuación se definirá el contenido de cada uno de los registros asignados:

Dirección de los caracteres a ser sacados (RO).

En este registro se coloca la dirección del mensaje que va a ser mostrado en el visualizador del sistema.

Bandera de campo (BNDCP) (R2).

0 - Campo de datos

1 - Campo de dirección

Bandera de punto (BNDPT) (R3)

0 - No marcar punto

1 - Marcar punto

Bandera de memoria (BNDMM) (R4)

0 - Memoria RAM

1 - Memoria ROM

Valor del mensaje a ser mostrado (R25)

En este registro se coloca un valor de 1 a 5

dependiendo del tipo de mensaje que se de see mostrar en el visualizador

1 - Mensaje EPR

2 - Mensaje DIE

3 - Mensaje DFE

4 - Mensaje DIR

5 - Mensaje DFR

Bandera de lectura/ escritura (BNDRW) (R26)

0 - escritura

1 - lectura

Registro donde se almacena tipo de Eprom y direcciones (R27)

En este registro se almacena temporalmente la dirección donde se va a almacenar posteriormente el tipo de Eprom, DIEPR, DFEPR, DIRAM y DFRAM.

Valor del comando a utilizar (R29)

En este registro se almacenan temporalmente uno de los tres valores que a continuación se - muestran:

0A - valor del comando PROGM

0B - valor del comando READ y COMPG

0C - valor del comando (CPBEP)

Bandera para obtener dato o dirección (BNDDD) (R30)

0 - Dato

1 - dirección

Registro imagen del acumulador (R32)

En este registro se almacena temporalmente la información contenida en el acumulador, en el caso de salto á la rutina de interrupción.

Bandera para mostrar dirección y dato al visualizador o al microcomputador (BNDVT) (R33).

0 - al visualizador

1 - al microcomputador

Tipo de Eprom (EPROM) (R34)

En este registro se almacena el Código del Eprom que se va a leer o grabar.

Dirección inicial del Eprom (DIEPR) (R35-R36)

En este par de registros se almacenan los dos bytes de la dirección inicial del Eprom.

Dirección final del Eprom (DFEPR) (R37-R38)

En este par de registros se almacenan los dos bytes de la dirección final del Eprom.

Dirección final de la RAM (DFRAM) (R39-R40)

En este par de registros se almacenan los dos bytes de la dirección final de la RAM.

Dirección inicial de la RAM (DFRAM) o dirección común . (ADDCM) (R41-R42).

En este par de registros se almacenan los dos bytes de la dirección inicial de la RAM o también se almacenan los dos bytes de la dirección común dependiendo del tipo de comando que se utilice.

Dato común (ADDTC) (R43)

Este registro se lo usa para almacenar tem

poralmente el dato que posteriormente va a ser utilizado en alguna operación.

Palabra de control (ACNTL) (R44)

En este registro se almacena la palabra de control para generar los voltajes necesarios - para grabar o leer un Eprom.

Buffer de entrada (BUFIN) (R45)

En este registro se almacena el carácter leído del teclado.

Buffer de salida (ADBFO) (R46-R48)

Estos registros se los usa para almacenar - los caracteres que van a ser mostrados - en el visualizador.

Byte de dirección de orden más alto (ADBHG) (R49)

En este registro se almacena el Byte de orden más alto de la DIR, después que el byte ha sido cambiado previamente para habilitar la página de memoria de datos externa.

Registro imagen de la DIR (DIRIM) (R50-R51)

En este par de registros se almacenan los dos bytes de la dirección inicial de la RAM, para - el caso de varios lazos de programación, como en el Eprom 2708.

Registro imagen de la dirección inicial del Eprom (DIEIM) - (R52-R53)

En este par de registros se almacenan los dos bytes de la dirección inicial del Eprom, para - el caso de varios lazos de programación, como - en el Eprom 2708.

DIAGRAMA DE FLUJO GENERAL

Antes de entrar a hacer una descripción del Diagrama de Flujo definiremos cada uno de los simbolos utilizados en el diagrama de flujo.

Este símbolo se lo usa para empezar el Diagrama de Flujo.

El rectángulo significa alguna operación que se va a realizar.

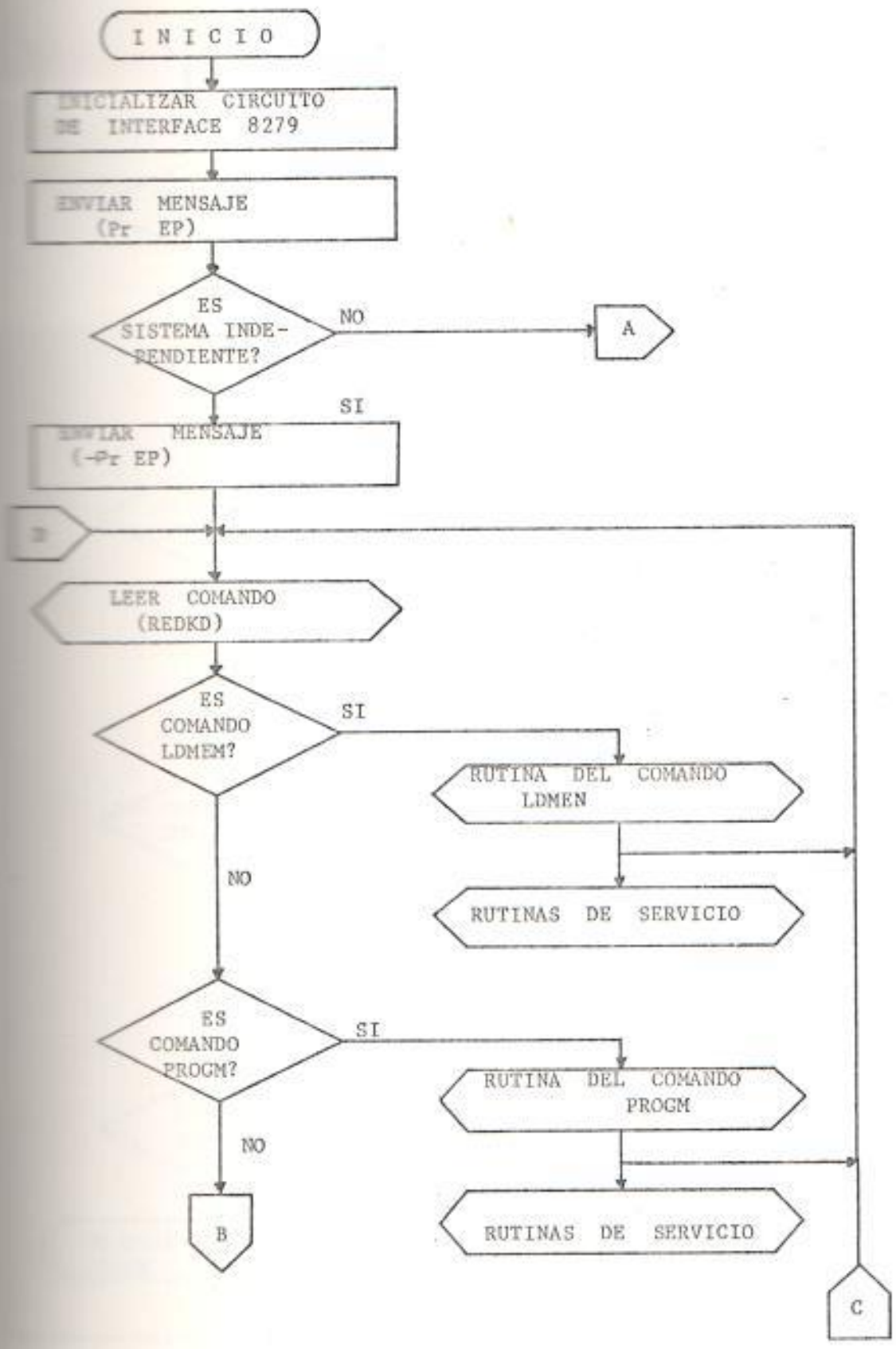
Este símbolo significa llamada a alguna rutina.

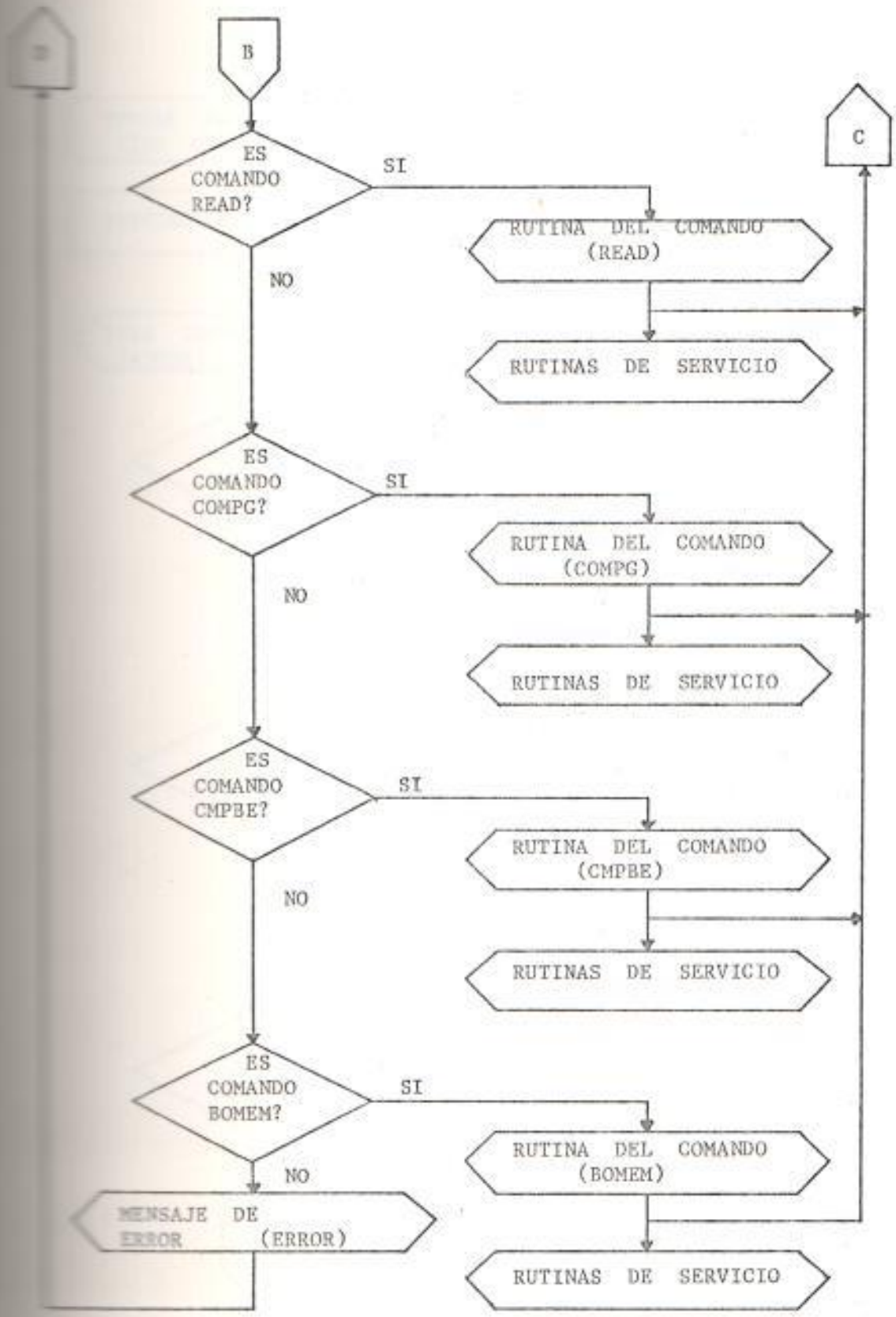
El rombo se lo usa para tomar decisiones.

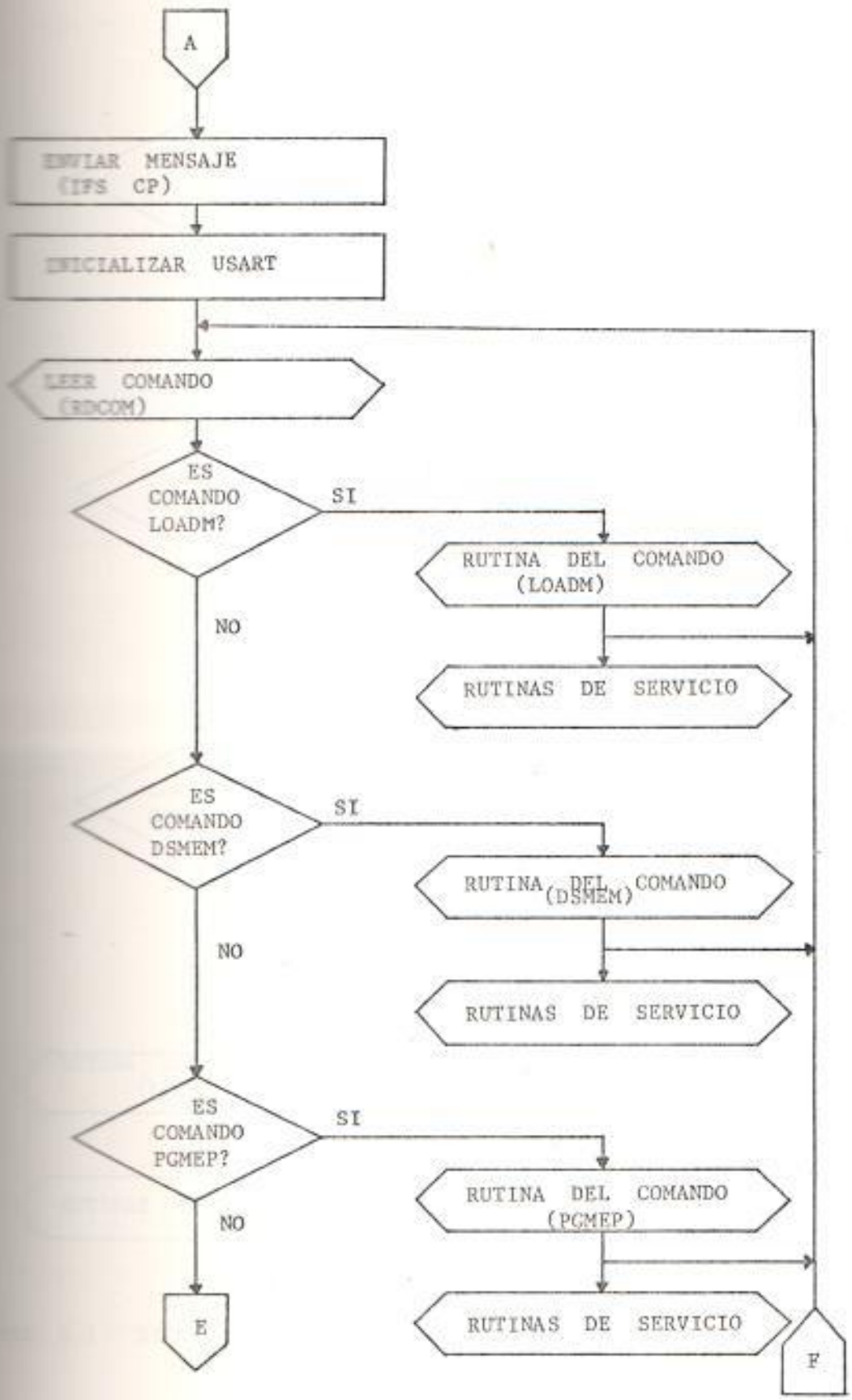
Este símbolo se lo utiliza para enlazar Diagramas de Flujo.

La elipse se la usa para finalizar el Diagrama de Flujo.

Todo el software del sistema se muestra en el Diagrama de Flujo simplificado de la figura Nº 3.1. El software opera de la siguiente manera: primero se inicializa la operación del circuito de interface 8279, luego el software muestra en el visualizador un mensaje de inicialización (Pr EP). Se puede hacer operar el sistema en dos formas: independientemente o con la interface a un microcomputador. Primero explicaremos el software cuando el sistema opera independientemente. Un mensaje es mostrado en el visualizador, el cual consiste de un guión en el primer led visualizador acompañando al primer mensaje que fue mostrado al inicio. (-Pr EP).


INICIO
INICIALIZAR CIRCUITO DE INTERFACE 8279
ENVIAR MENSAJE (Pr EP)
ES SISTEMA INDEPENDIENTE?
NO
A
SI
ENVIAR MENSAJE (→Pr EP)
B
LEER COMANDO (REDKD)
ES COMANDO LDMEM?
SI
RUTINA DEL COMANDO LDMEN
NO
RUTINAS DE SERVICIO
ES COMANDO PROGM?
SI
RUTINA DEL COMANDO PROGM
NO
RUTINAS DE SERVICIO
B
C


B
C
ES COMANDO READ?
SI
NO
RUTINA DEL COMANDO (READ)
RUTINAS DE SERVICIO
ES COMANDO COMPG?
SI
NO
RUTINA DEL COMANDO (COMPG)
RUTINAS DE SERVICIO
ES COMANDO CMPBE?
SI
NO
RUTINA DEL COMANDO (CMPBE)
RUTINAS DE SERVICIO
ES COMANDO BOMEM?
SI
NO
RUTINA DEL COMANDO (BOMEM)
RUTINAS DE SERVICIO
MENSAJE DE ERROR (ERROR)


A
ENVIAR MENSAJE (IFS CP)
INICIALIZAR USART
LEER COMANDO (RDCOM)
ES COMANDO LOADM?
SI
NO
RUTINA DEL COMANDO (LOADM)
RUTINAS DE SERVICIO
ES COMANDO DSMEM?
SI
NO
RUTINA DEL COMANDO (DSMEM)
RUTINAS DE SERVICIO
ES COMANDO PGMEP?
SI
NO
RUTINA DEL COMANDO (PGMEP)
RUTINAS DE SERVICIO
E
F

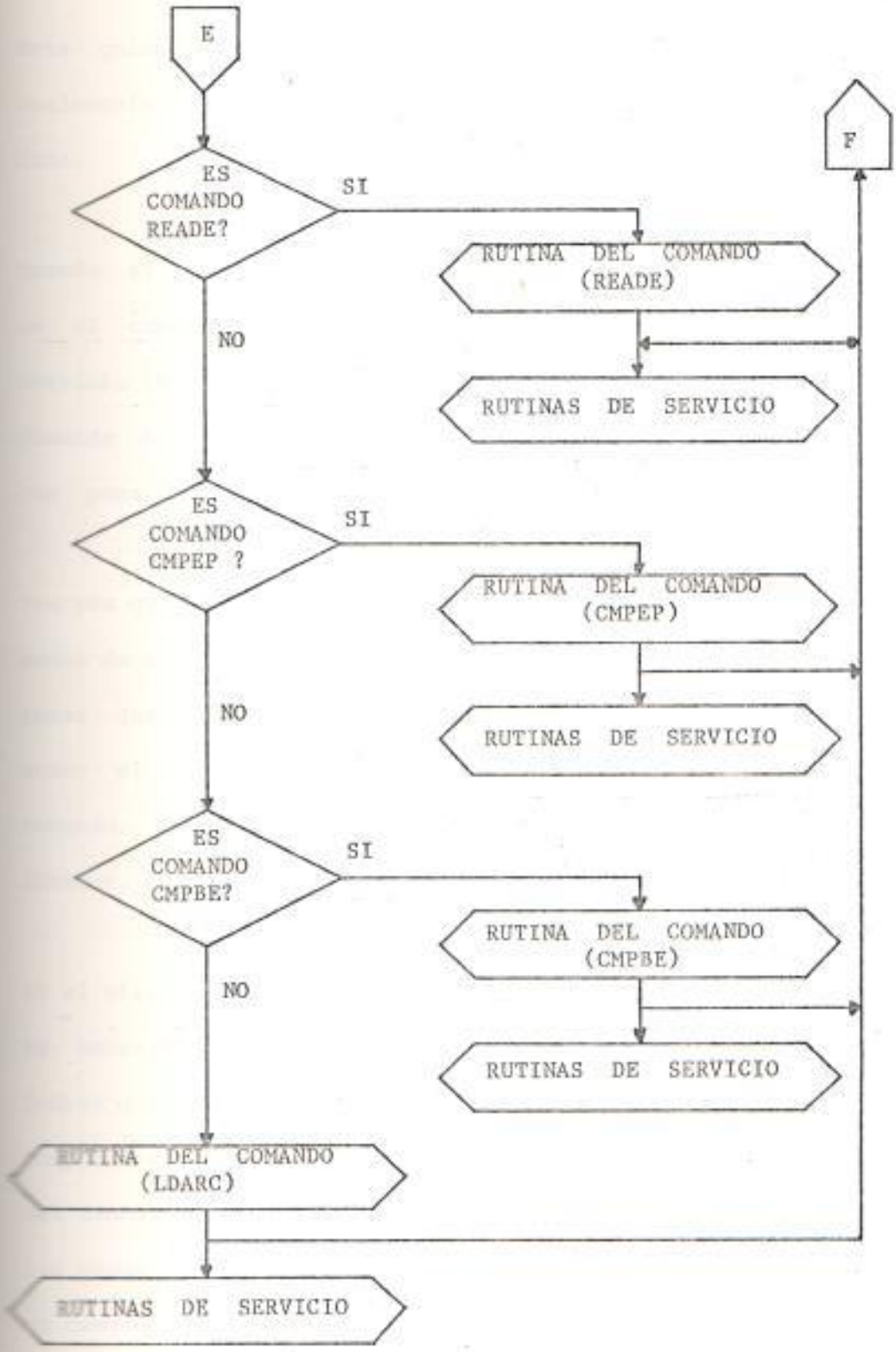


FIG. 3.1. DIAGRAMA DE FLUJO GENERAL DEL SISTEMA.

Este guión indica que el sistema está listo para leer cualesquiera de los seis comandos que el sistema utiliza.

Cuando el usuario entra un comando el software reconoce el comando e inmediatamente salta a la rutina de servicio del correspondiente comando, a su vez cada comando hace uso de un paquete de rutinas de servicio para su operación adecuada.

Una vez que la rutina de comando finaliza o es finalizada por medio de una tecla de información, el software regresa a esperar que otro comando sea entrado. Si por algún error el usuario presiona alguna tecla que no sea un comando, un mensaje de error es mostrado en el visualizador.

Si el sistema va a operar con la interface RS-232, el siguiente mensaje es mostrado en el visualizador (IFS CP) el cual indica que el sistema está listo para conectarse con un microcomputador. El paso siguiente es inicializar la operación del USART, posteriormente se puede utilizar cualesquiera de los comandos que se utilizan para la comunicación. Cuando un comando es entrado este es inmediatamente reconocido por el software y salta a la rutina de servicio de ese comando, a su vez cada rutina de comando hace -

uso de un paquete de rutinas de servicio.

Cuando el comando finaliza o es finalizado por alguna tecla de información, el software regresa a esperar que otro comando sea entrado.

3.4. RUTINA RECONOCEDORA DE COMANDOS

Esta es la rutina de más alto nivel en el software del sistema. Se la utiliza para reconocer el comando entrado por el usuario y obtener la dirección de la rutina de ese comando para posteriormente saltar a la rutina correspondiente. En caso de que la información entrada no corresponda a un comando un mensaje de error es mostrado en el visualizador.

3.5. RUTINA PARA CARGAR LA MEMORIA DE DATOS DEL SISTEMA

Con esta rutina se da servicio al comando LDMEN el cual se utiliza para cargar la memoria de datos externa del sistema con la información que posteriormente será grabada en algún tipo de Eprom.

Del Diagrama de Flujo (figura N°- 3.3), podemos observar que primero se blanquea el visualizador, luego se obtiene la dirección inicial a partir de la cual se comienza

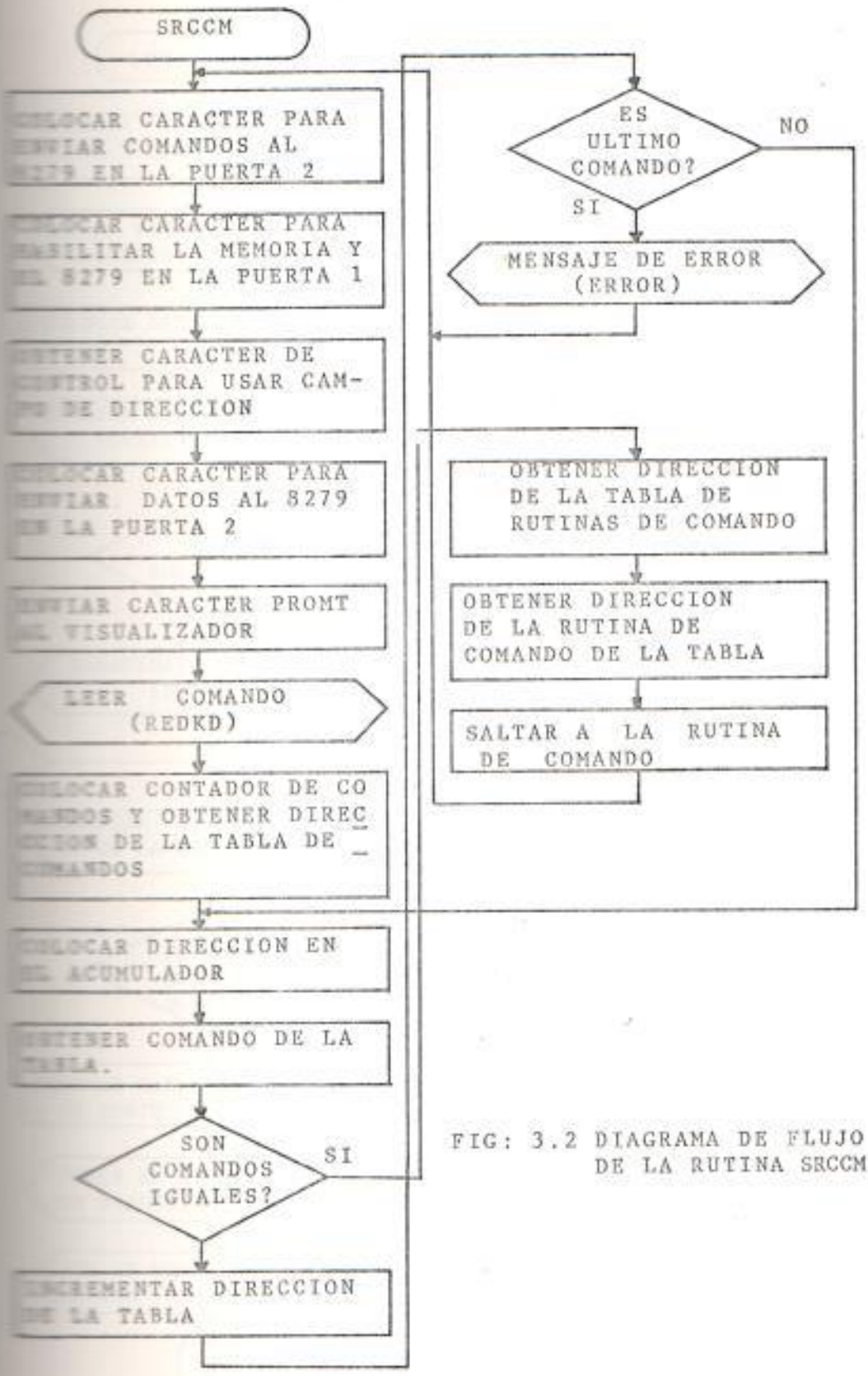
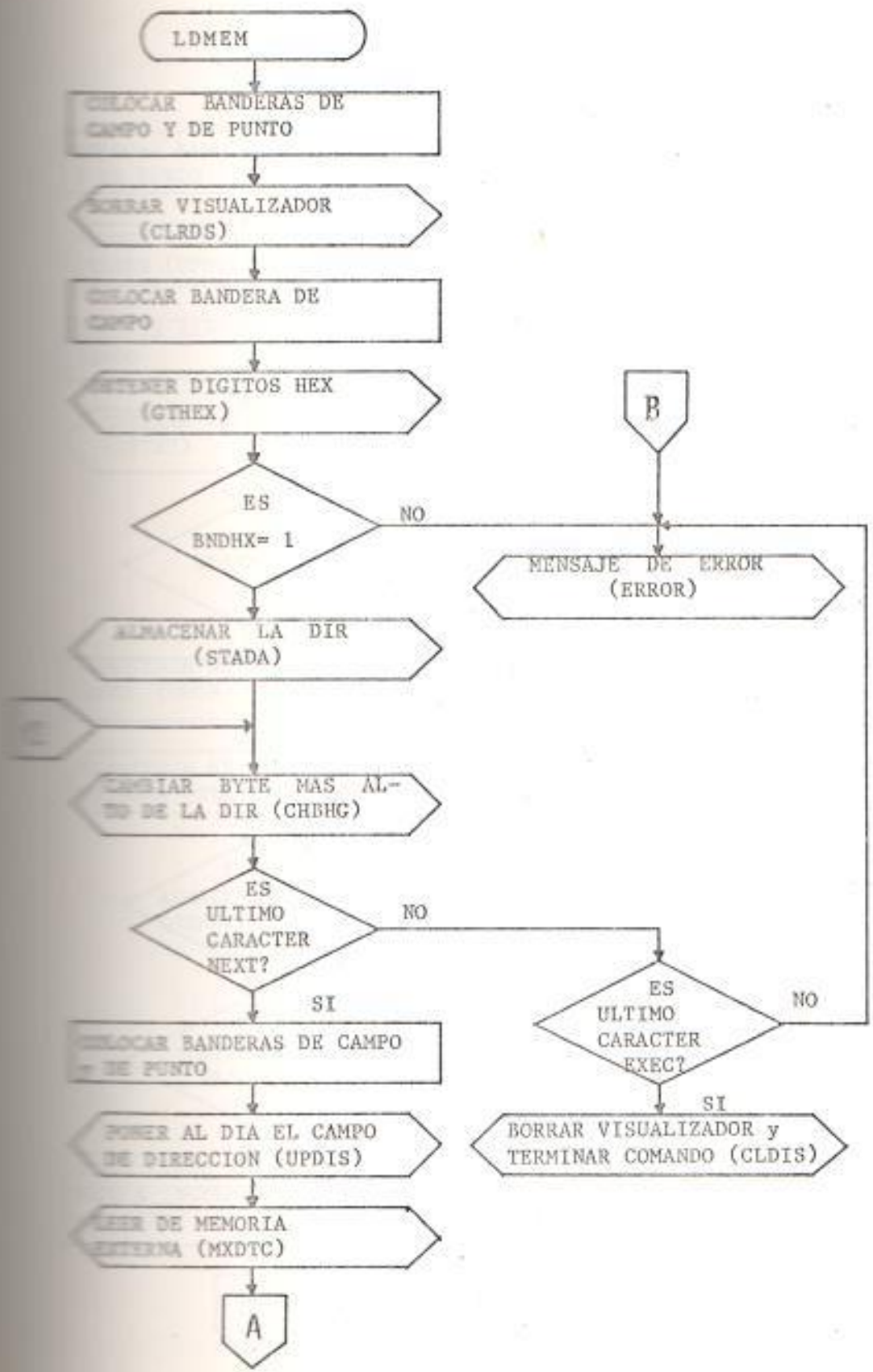


FIG: 3.2 DIAGRAMA DE FLUJO DE LA RUTINA SRCCM


LDMEM
COLOCAR BANDERAS DE CAMPO Y DE PUNTO
BORRAR VISUALIZADOR (CLRDS)
COLOCAR BANDERA DE CAMPO
OBTENER DIGITOS HEX (GTHEX)
B
ES BNDHX= 1
NO
MENSAJE DE ERROR (ERROR)
ALMACENAR LA DIR (STADA)
CAMBIAR BYTE MAS AL- TO DE LA DIR (CHBHG)
ES ULTIMO CARACTER NEXT?
NO
ES ULTIMO CARACTER EXEC?
NO
SI
SI
COLOCAR BANDERAS DE CAMPO Y DE PUNTO
BORRAR VISUALIZADOR y TERMINAR COMANDO (CLDIS)
PONER AL DIA EL CAMPO DE DIRECCION (UPDIS)
LEER DE MEMORIA EXTERNA (MXDTC)
A

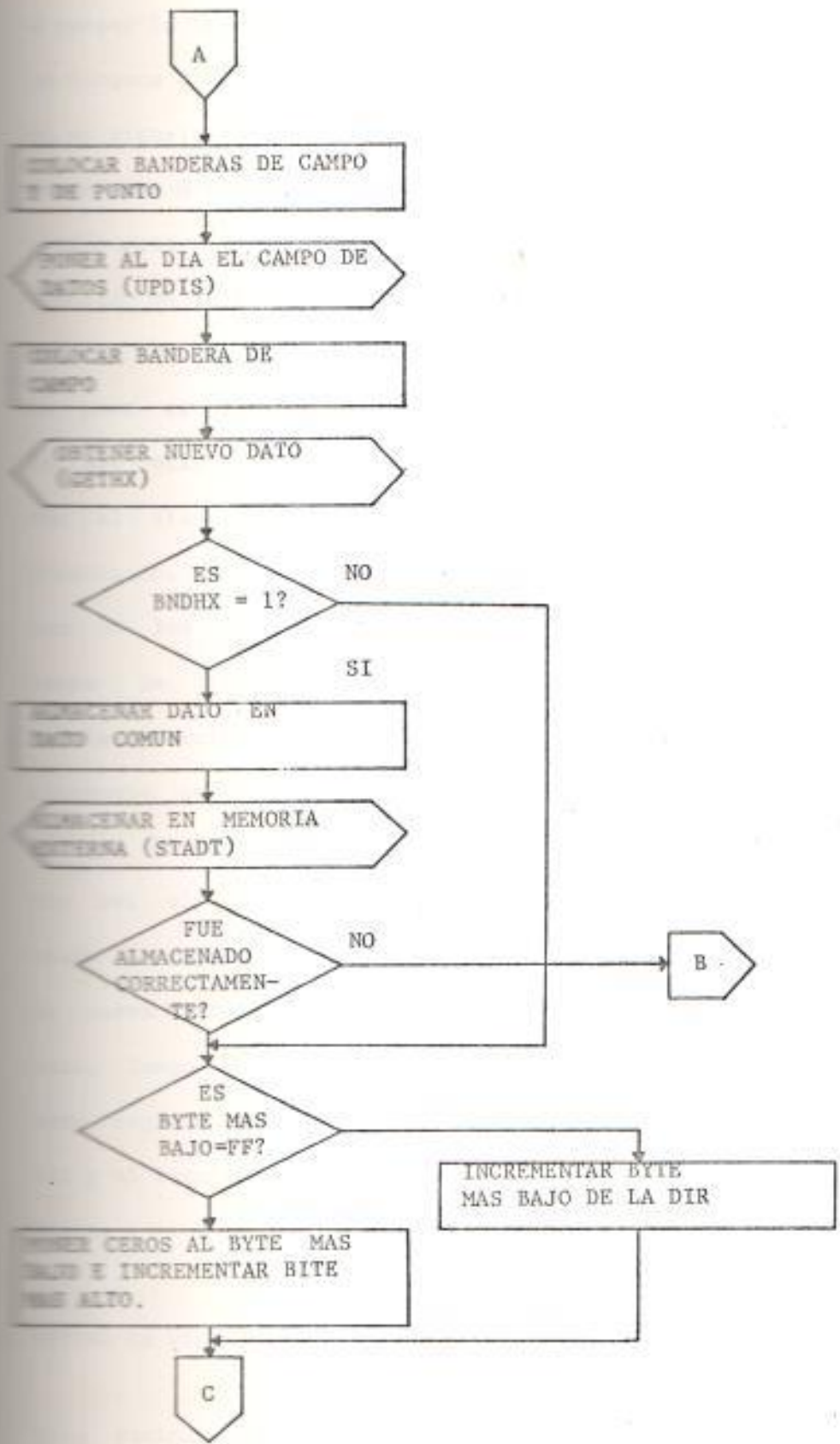


FIG. 3.3. DIAGRAMA DE FLUJO DE LA RUTINA LDMEM.

a cargar la información deseada. En caso de que no se obten- ga ninguna dirección inicial un mensaje de error es mostrado en el visualizador y el comando será finalizado. Posterior mente esta dirección inicial será almacenada en alguna po sición determinada y su byte de orden más alto será - cambiado en la forma correcta para habilitar la página correspondiente de la memoria de datos externa. El siguien te paso es leer un carácter del teclado, el cual pue de ser NEXT o EXEC en caso de no ser ninguno de los dos el visualizador será borrado y el comando será finalizado, por el contrario si el caracter leído fue uno de los dos descritos anteriormente, entonces el campo de dirección del visualizador es puesto al día con la dirección actual existente en ese momento. Poste- riormente se procede a leer la información almacenada en esa dirección y se pone al día el campo de da tos del visualizador. Esta información puede ser cam biada o dejada tal como está, si esta es cambiada - la nueva información será almacenada en la dirección ac tual, luego la dirección será incrementada y el soft ware regresa a cambiar el byte de orden más alto de la DPH y el lazo se repetirá en la misma forma descrita anterior mente.

RUTINA DE VERIFICACION DE BORRADO DEL EPROM

Esta rutina sirve para dar servicio al comando CPBEP,

el cual se utiliza para verificar si el Eprom está completamente borrado. El software para verificar si un Eprom está borrado se muestra en la figura N°- 3.4. Primero tenemos que colocar el valor del comando correspondiente y dar el valor adecuado a la bandera de lectura/escritura, seguidamente se obtienen los datos de entrada tales como: tipo de Eprom que se va a verificar, dirección inicial del Eprom y dirección final del Eprom.

Una vez que se obtienen todos los datos de entrada, se procede a leer una tecla de información (EXEC), para continuar con la rutina que se encarga de verificar si el Eprom está borrado. Si la tecla de información que se lee no es la correcta un mensaje de error es mostrado en el visualizador y automáticamente se dará por finalizado el comando.

Igual caso sucede cuando todas las direcciones comprendidas entre la dirección inicial y final han sido verificadas. La rutina de verificación del borrado del Eprom será descrita en la última sección de este Capítulo.

RUTINA PARA GRABAR INFORMACION

Esta rutina se usa para dar servicio al comando PROGM, que sirve para grabar la información almacenada en la

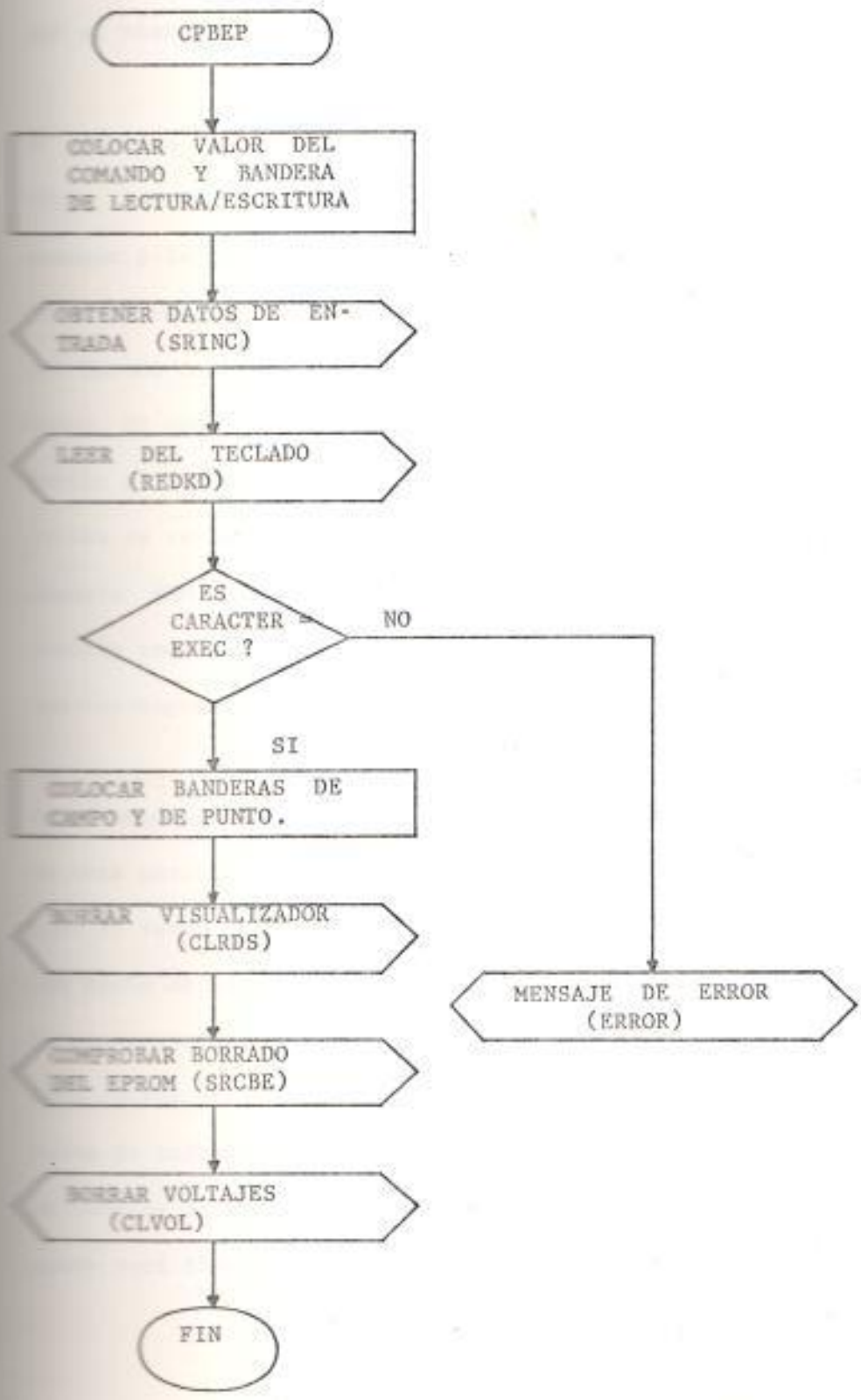


FIG. 3.4. DIAGRAMA DE FLUJO DE LA RUTINA CPBEP

memoria de datos externa dentro de cualquiera de los Eproms que el sistema graba.

El software de esta rutina se muestra en el Diagrama de - flujo de la figura N°- 3.5., primero colocamos el valor del comando y la bandera de lectura/escritura, luego obtenemos los datos de entrada tales como: tipo de eprom , dirección inicial del Eprom, dirección inicial y final de la memoria de datos ex terna. El paso inmediato es blanquear el visualizador y pre guntar si el Eprom que se va a grabar es el 2708, si la res puesta es verdadera se almacena la dirección inicial de la memoria RAM y la dirección inicial del Eprom en el par de re gistros imagen correspondientes, y colocamos el número de la zas correspondientes al Eprom 2708.

Ahora tomemos el caso contrario cuando la respuesta es falsa , en este caso se coloca el número de lazos a un valor de uno , para el resto de Eproms. El paso inmediato es leer del teclado una tecla de información (EXEC), para continuar con la rutina de programación, la cual se encarga de la grabación del Eprom. Esta rutina será descrita más tarde en este Capítulo. Si la tecla de información leída anteriormente no es la correcta - un mensaje de error será mostrado en el visualizador y el co mando será finalizado.

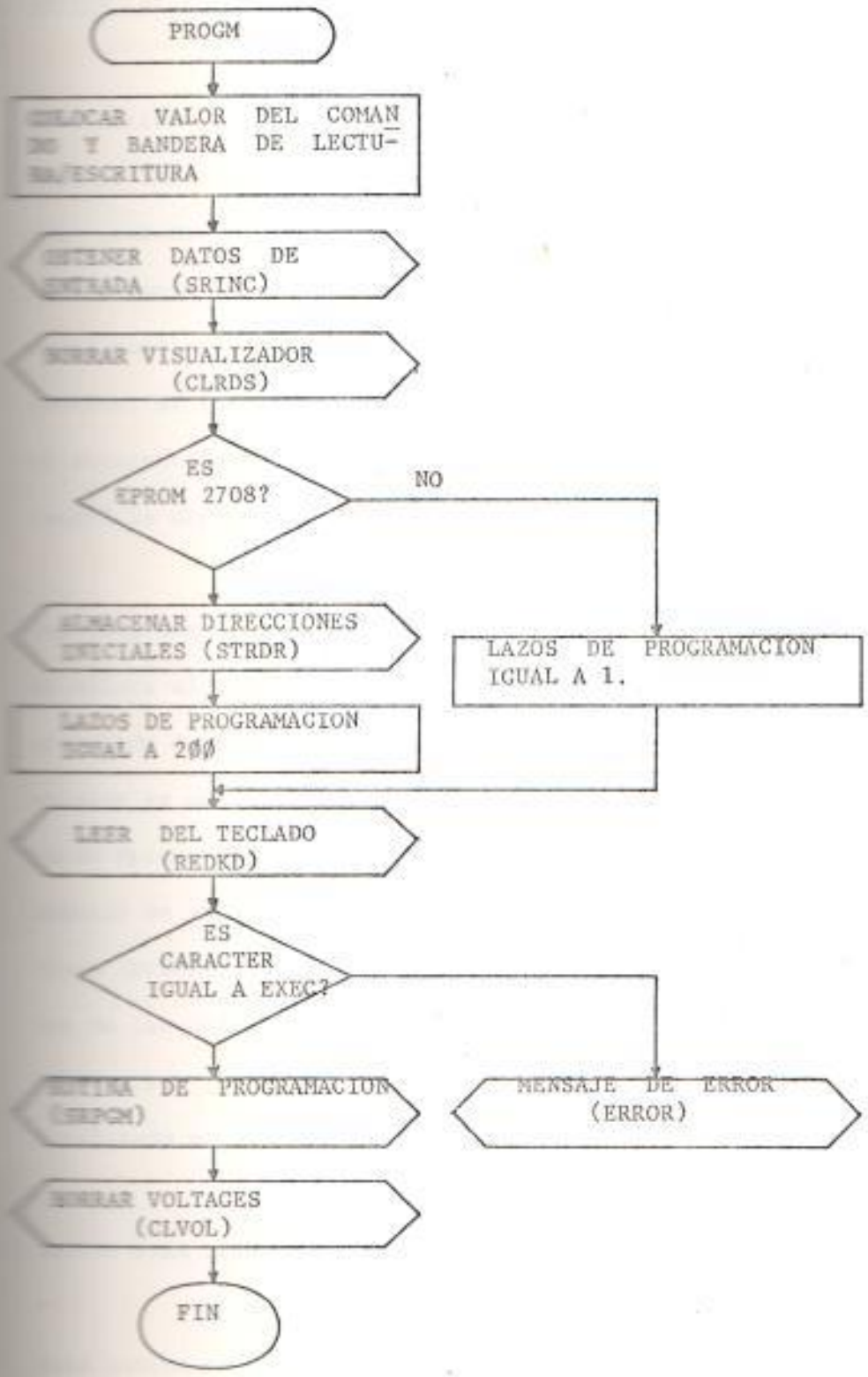


FIG. 3.5. DIAGRAMA DE FLUJO DE LA RUTINA PROGM.

RUTINA PARA COMPROBAR PROGRAMACION DEL EPROM

Esta rutina sirve para dar servicio al comando COMPG el cual se usa para comprobar si cada uno de los caracteres grabados en el Eprom es el correcto, para hacer esto primero lee el caracter grabado y lo compara con el correspondiente caracter almacenado en la memoria de datos externa, si los dos caracteres son iguales se pasa a comparar el siguiente caracter, caso contrario en el visualizador aparecerá la dirección y el caracter grabado en esa dirección.

Del diagrama de flujo (figura N°- 3.6), observamos que primero se coloca el valor del comando correspondiente y el valor apropiado de la bandera de lectura/escritura, el paso siguiente es obtener los datos de entrada tales como: tipo de Eprom, dirección inicial y final del Eprom y dirección inicial de la RAM. Posteriormente leemos un caracter del teclado (EXEC), si este es el correctp se pasa a comprobar cada una de las direcciones del Eprom, en caso contrario un mensaje de error será mostrado en el visualizador y se dará por finalizado el comando.

RUTINA PARA LEER LA INFORMACION ALMACENADA EN EL EPROM

Esta rutina se usa para dar servicio al comando READ la cual

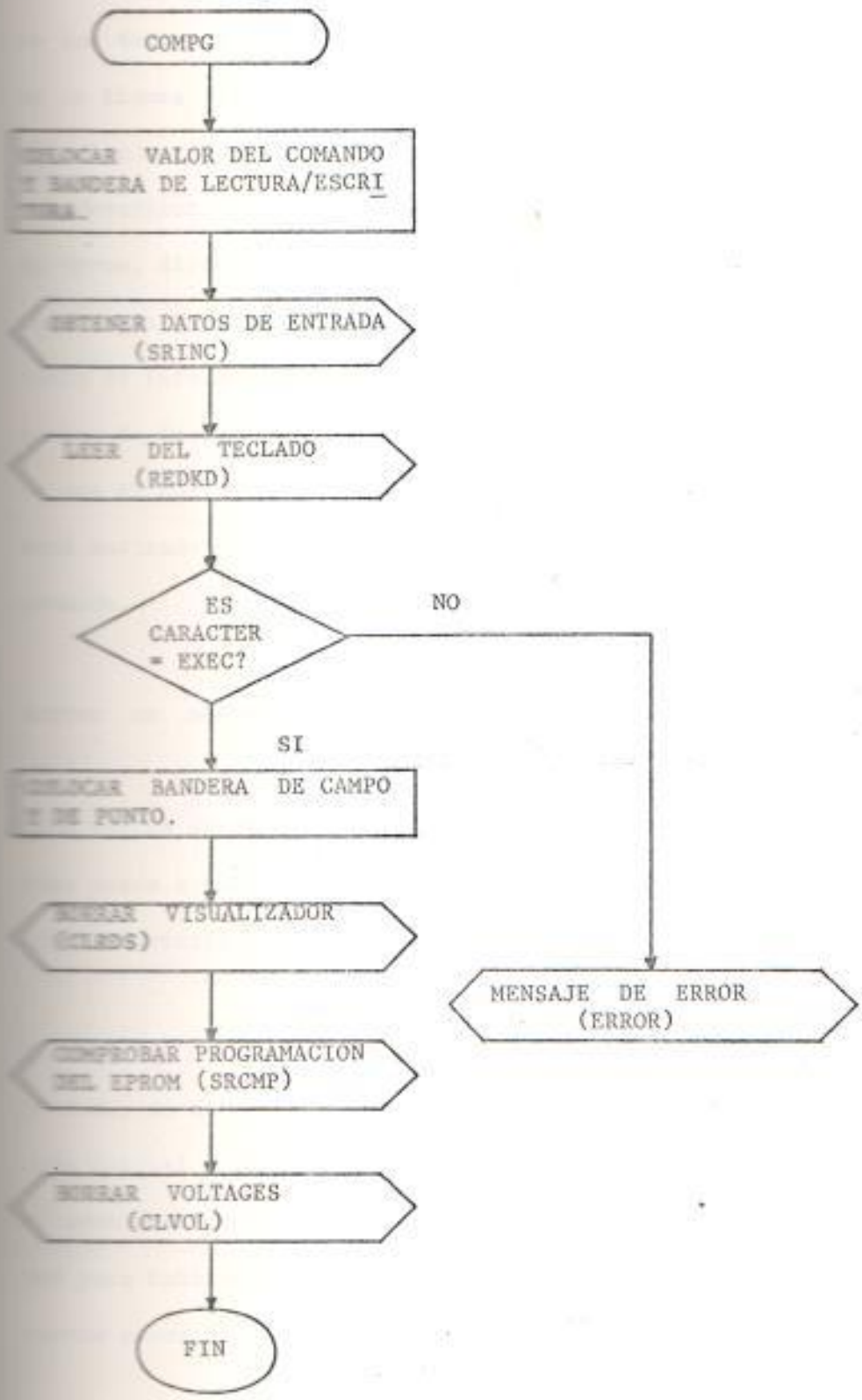


FIG. 3.6. DIAGRAMA DE FLUJO DE LA RUTINA COMPG.

la información almacenada en el Eprom dentro de la memoria de los datos externa. El software de esta rutina se muestra en la figura 3.7., de este observamos que primero se coloca - el valor correspondiente del comando y de la bandera de lectura/escritura, luego los datos de entrada tales como; tipo - de Eprom, dirección inicial y final del Eprom y dirección inicial de la Ram, son obtenidos. El paso siguiente es leer una tecla de información (EXEC), para más tarde continuar con la rutina de lectura del Eprom, en caso de que el caracter leído sea el correcto, si no es el correcto un mensaje de error será mostrado en el visualizador y se dará por finalizado el comando.

RUTINA DE BORRADO DE LA MEMORIA DE DATOS EXTERNA

Esta rutina sirve para dar servicio al comando BOMEM el cual pone ceros a todas las posiciones de memoria comprendidas en tre la dirección inicial y final.

El diagrama de flujo de esta rutina se muestra en la figura - No 3.8., de este observamos que primero se obtienen la dirección inicial y final de la RAM, el paso siguiente es cambiar el byte del orden más alto de la dirección inicial de la RAM para habilitar la página correcta y por último se van poniendo ceros en cada una de las posiciones de memoria hasta

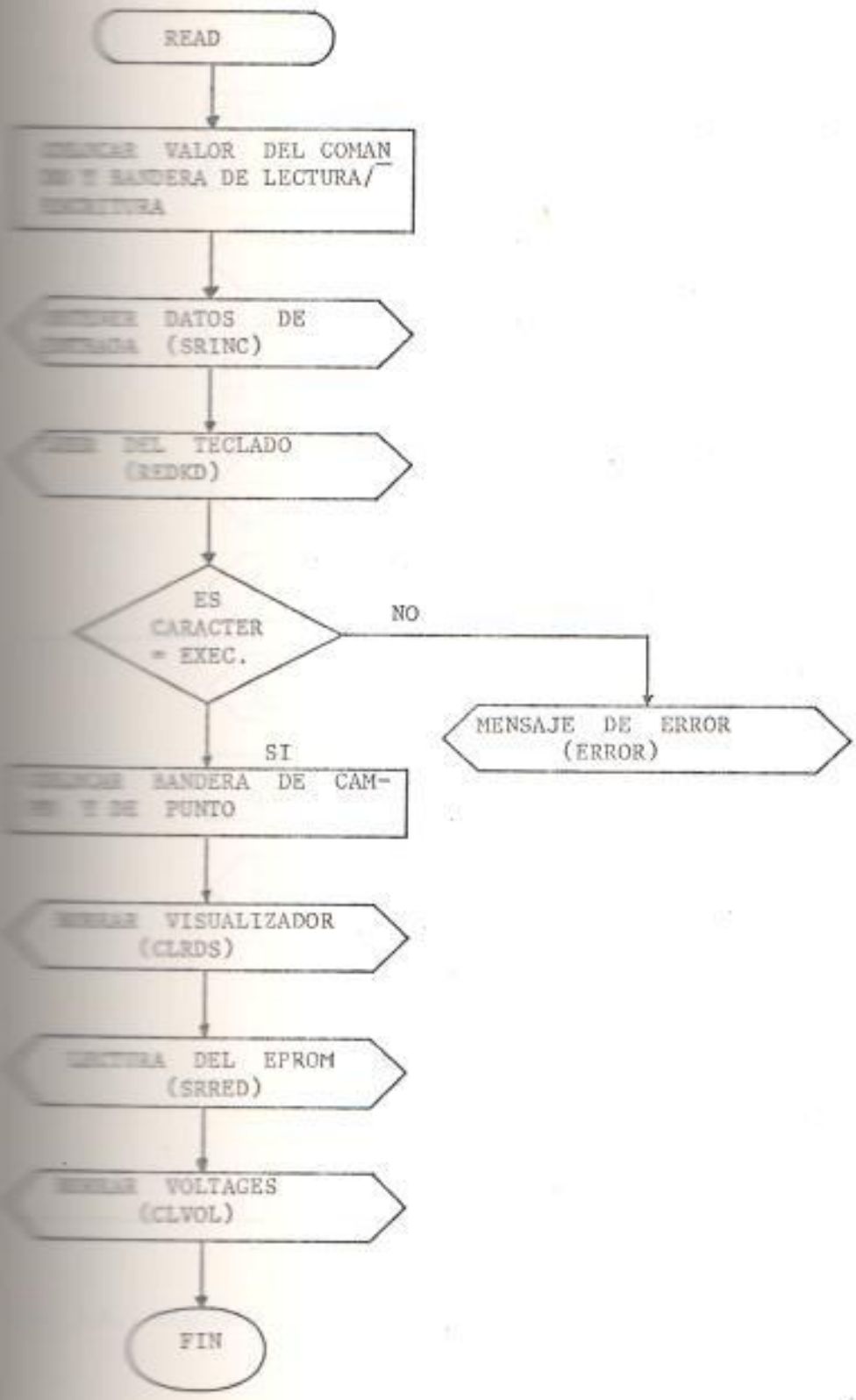


DIAGRAMA DE FLUJO DE LA RUTINA READ

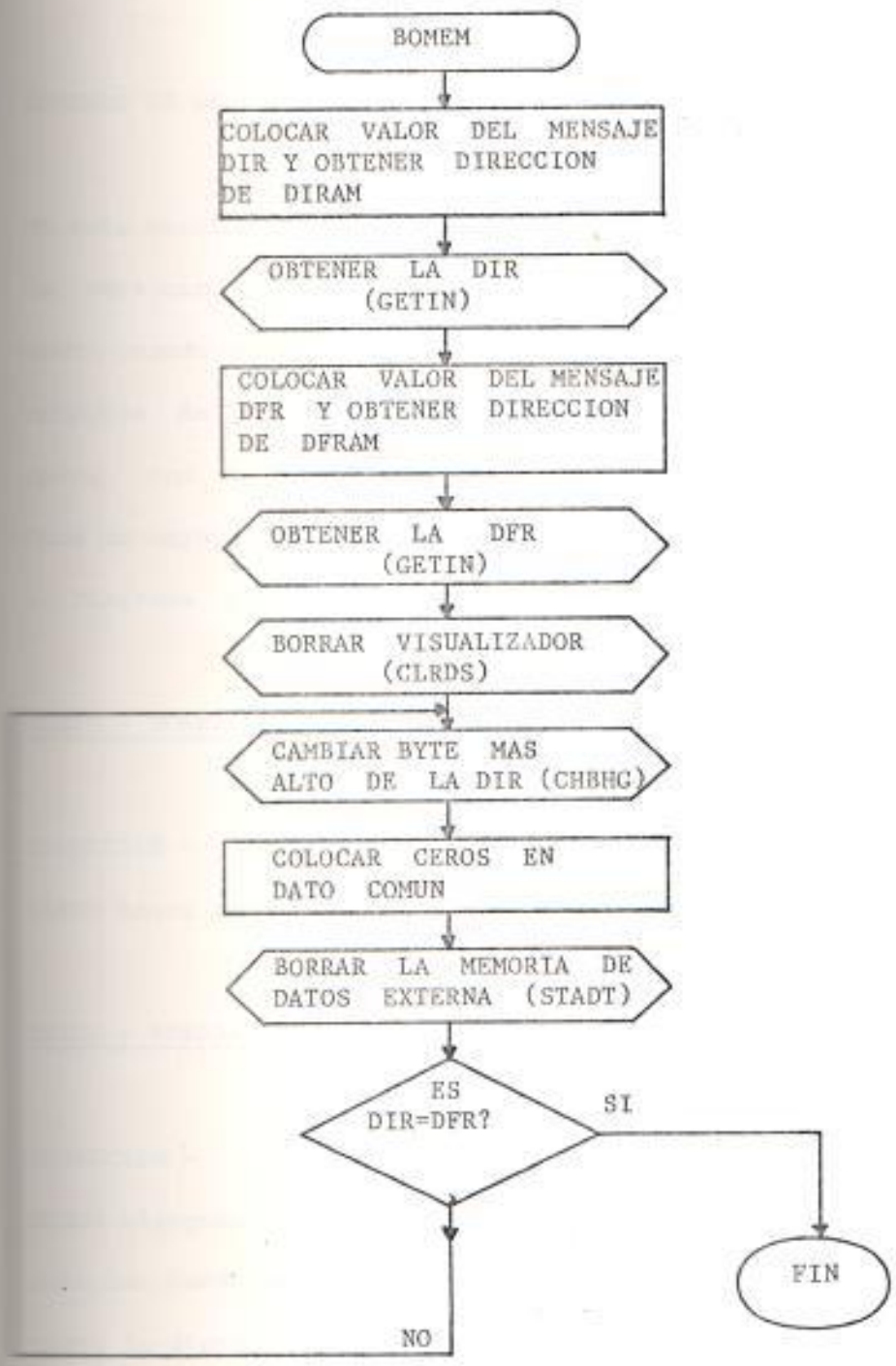


FIG. 3.8. DIAGRAMA DE FLUJO DE LA RUTINA BOMEN

que la dirección inicial sea igual a la dirección final.

RUTINAS DE SERVICIO DESCRIPCION DE CADA UNA

En esta sección describiremos cada una de las rutinas de servicio que se usan en el monitor del sistema solamente cuando este opera en forma independiente. La descripción de las rutinas de servicio cuando el sistema opera con la interface RS-232, se verá más adelante. Para un mayor entendimiento de estas rutinas presentaremos el Diagrama de Flujo de cada una.

CLEPO - BORRAR BUFFER DE SALIDA (Figura N°- 3.9)

DIRECCION -

CLEPO borra cada una de las posiciones del buffer de salida.

CLEDS - BORRAR VISUALIZADOR (Figura N°- 3.10)

DIRECCION -

CLEDS blanquea el visualizador, para hacer esto primero se coloca las banderas para blanquear campo de dirección y se obtiene la dirección de los caracteres BLANK, luego los caracteres son enviados al visualizador por medio de la rutina OUT DSL. Para blanquear el campo de datos el procedimiento es igual.

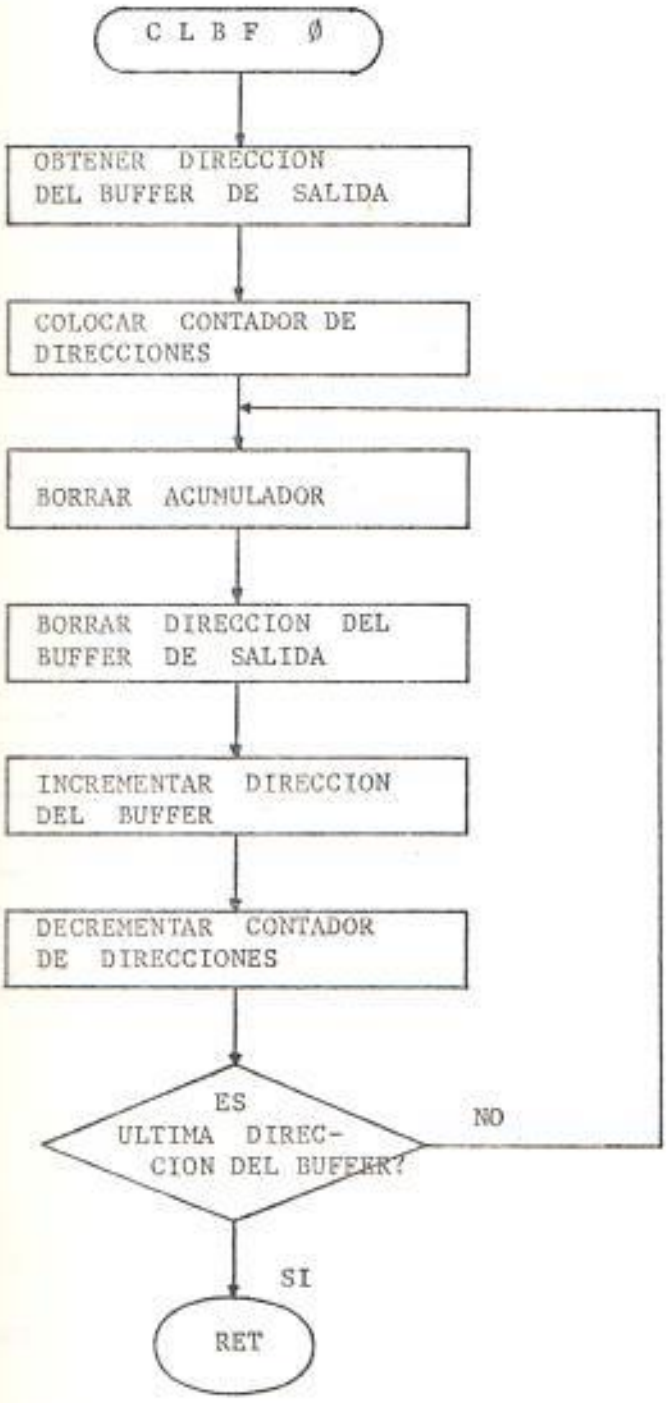


FIG. 3.9. DIAGRAMA DE FLUJO DE LA RUTINA CLBFØ

FIG. 3.18. DIAGRAMA DE FLUJO DE LA RUTINA CLRDS.

[illegible] - MENSAJE DE ERROR Y TERMINAR COMANDO. (Figura Nº 3.11)

[illegible]CCION -

[illegible] envia al campo de dirección del visualizador un mensaje de error y blanquea el campo de datos. Primero coloca- [illegible] banderas para enviar mensaje de error y obtenemos di[illegible]ción del mensaje, ERR luego el mensaje es mostrado por [illegible] de la rutina OUTDS. Para blanquear el campo de datos [illegible] procedimiento es igual que el caso de la rutina ante[illegible].

[illegible] - BORRAR VISUALIZADOR Y TERMINAR COMANDO (FIG. 3.12)

[illegible]CCION

[illegible] blanquea el visualizador y regresa a la rutina recono[illegible]dora de comandos.

[illegible] - CAMBIAR BYTE DE ORDEN MAS ALTO DE DIR.(FIG.3.13)

[illegible]CCION

[illegible] cambia el byte de orden más alto de la dirección [illegible]cial de la memoria de datos externa (RAM), de modo que se [illegible] habilitar la página correspondiente al valor de la [illegible]. Una vez que el byte ha sido cambiado este es almacenado [illegible] la posición ADBHG.

[illegible] - LEER DE MEMORIA DE DATOS EXTERNA (FIG.3.14)

FIG. 3.11. DIAGRAMA DE FLUJO DE LA RUTINA ERROR

FIG. 3.12. DIAGRAMA DE FLUJO DE LA RUTINA CLDIS

FIG. 3.13. DIAGRAMA DE FLUJO DE LA RUTINA CHBHG

[illegible]CCION

[illegible] lee el caracter almacenado en la memoria de datos exter[illegible]. Para hacer esto primero colocamos el byte de orden - [illegible] alto de la DIR después que este ha sido cambiado previamen[illegible], en la puerta 2. El byte de orden más bajo es colocado [illegible] el registro RO, el paso siguiente es leer de memoria y [illegible]macenar el caracter leído en la posición de dato común.

[illegible] - ALMACENAR DATO EN LA MEMORIA DE DATOS EXTERNA (FIG.3.15)

[illegible]CCION

[illegible] Almacena el dato que se encuentra en la posición de da-[illegible]n común en la memoria de datos externa. Primero se coloca - [illegible] la puerta 2 el byte de orden más alto de la DIR después - [illegible] este ha sido cambiado previamente. El byte de orden más [illegible] se lo coloca en el registro 0,y se almacena el dato.

[illegible] - LEER DEL TECLADO (FIG. 3.16)

[illegible]CCION

[illegible] espera hasta que un caracter sea entrado por medio - [illegible] teclado y retorna el caracter entrado como una salida via [illegible] acumulador.

[illegible] - RUTINA DE INTERRUPCION (FIG. 3.17)

FIG. 3.34. DIAGRAMA DE FLUJO DE LA RUTINA MXDTC.

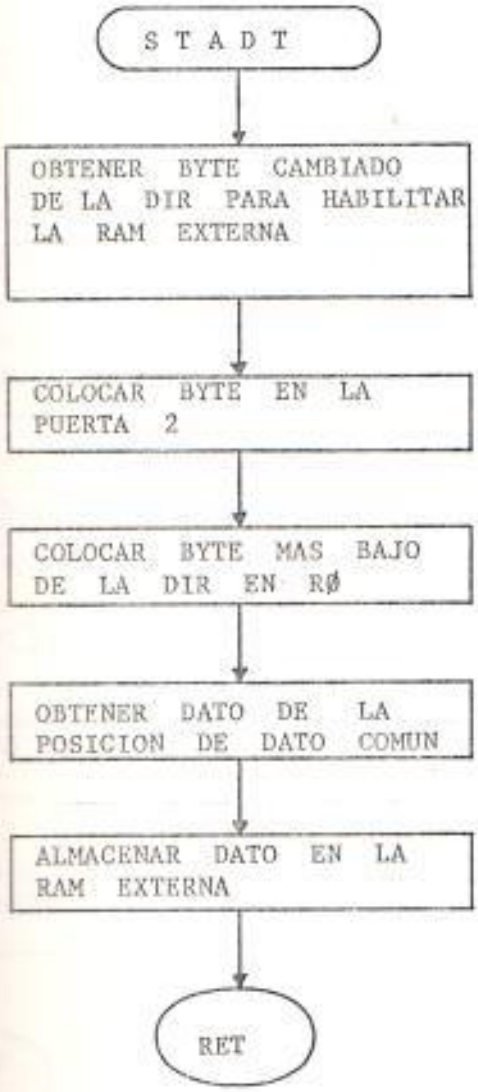


FIG. 3.15. DIAGRAMA DE FLUJO DE LA RUTINA STADT

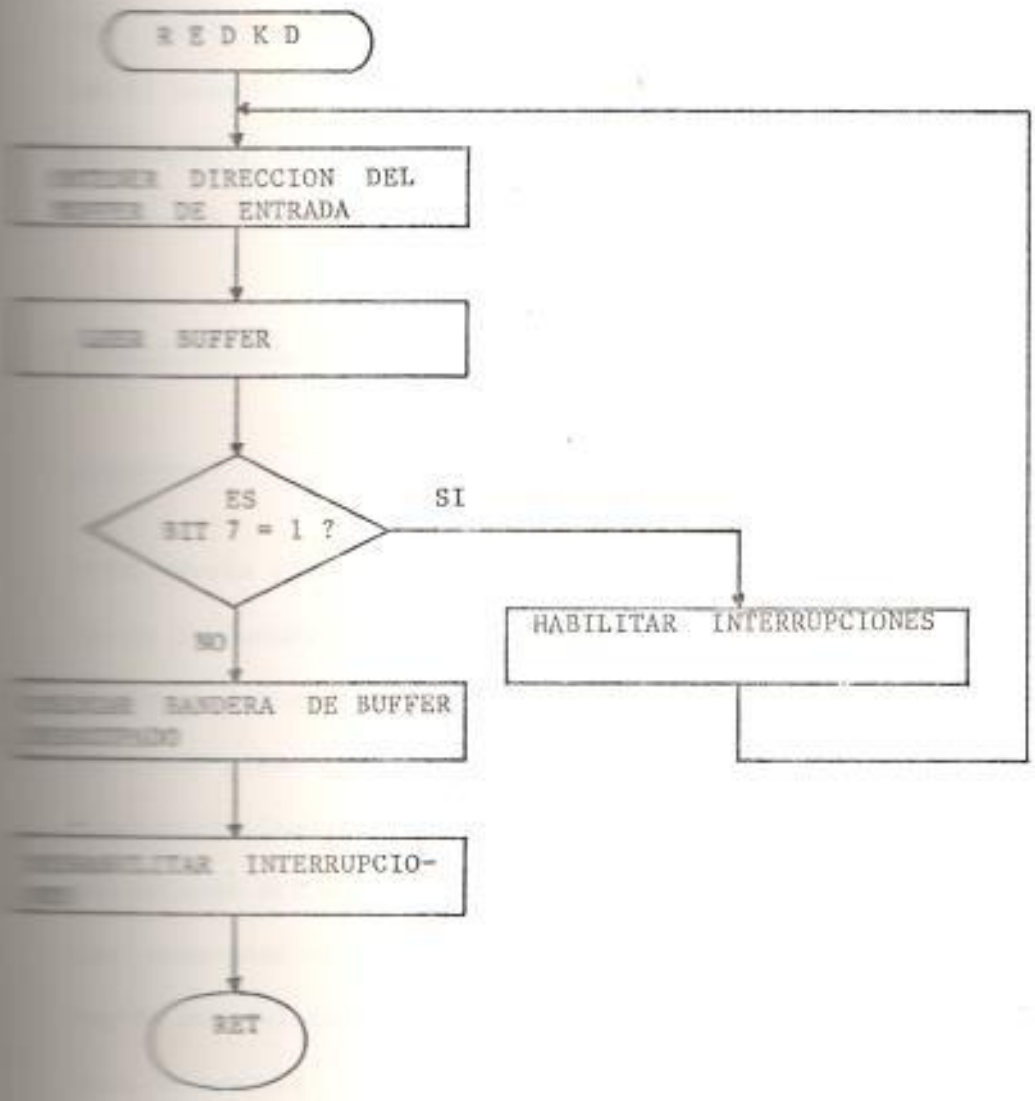


DIAGRAMA DE FLUJO DE LA RUTINA REDKD

[illegible]ECCION

[illegible]INT es entrada por un vector de interrupción cuando la [illegible]tina de lectura del teclado está esperando por un caracter. Cuando se ha presionado una tecla (Excepto RESET) RTINT [illegible] el caracter del circuito de interface a teclado/visualizador y lo almacena en el buffer de entrada, después [illegible]ansfiere el control a la rutina de lectura del teclado.

[illegible]TDS - ENVIAR CARACTERES AL VISUALIZADOR (FIG.3.18)

[illegible]ECCION

[illegible]TDS Envía caracteres al visualizador, dependiendo del valor de la bandera de campo, los caracteres pueden ser enviados al campo de dirección o al campo de datos. Para hacer esto primero se habilita el circuito de interface a teclado/visualizador, luego se prueba la bandera de campo y dependiendo del valor de esta obtenemos el caracter de control para usar el campo apropiado. Posteriormente el caracter de control es enviado al circuito de interface, la bandera de memoria es probada para averiguar si los caracteres a ser enviados van a ser obtenidos de la memoria RAM o de la memoria ROM. Posteriormente se obtiene la dirección de la tabla de de caracteres del visualizador, obtenemos el caracter de la tabla y probamos si es el último caracter, si es así se coloca punto al caracter dependiendo del valor -

FIG. 3.17. DIAGRAMA DE FLUJO DE LA RUTINA RTINT.

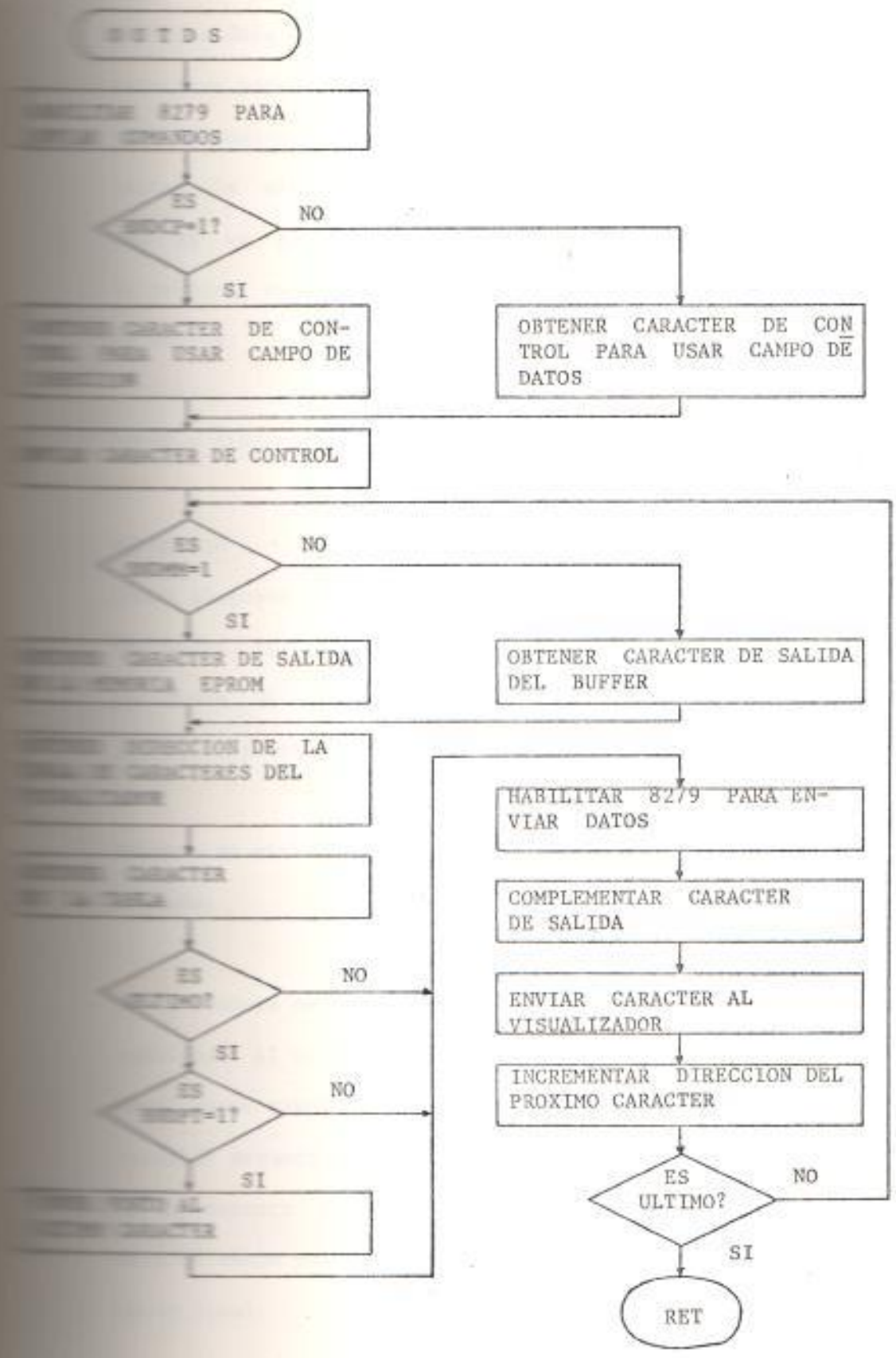


DIAGRAMA DE FLUJO DE LA RUTINA OUTDS.

de la bandera de punto, luego habilitamos el circuito de - interface para enviar datos, se complementa el caracter y este es enviado al visualizador. El paso último es incrementar la dirección del próximo caracter y probar para ver si es el último, si no lo es procedemos a obtener - el próximo caracter de salida, caso contrario retornar de la rutina.

RUTINA - ALMACENAR EN BUFFER DE SALIDA (FIG. 3.19)

DESCRIPCION

RUTINA almacena datos en el buffer de salida. Dependiendo - del valor del argumento de entrada, esta rutina almace na datos en el buffer para luego ser enviados al cam po de dirección o al campo de datos.

Primero se obtiene dirección del buffer de salida posición segunda.

Si la bandera de campo es igual a uno (campo de dirección), cambiamos el dato de la segunda posición a la posición primera. Luego cambiamos el dato de la tercera posición a la posición segunda y por último el nuevo dato es almacenado - en la tercera posición. Para el caso cuando la bandera de campo es igual a cero (campo de datos) el procedimien to es igual.

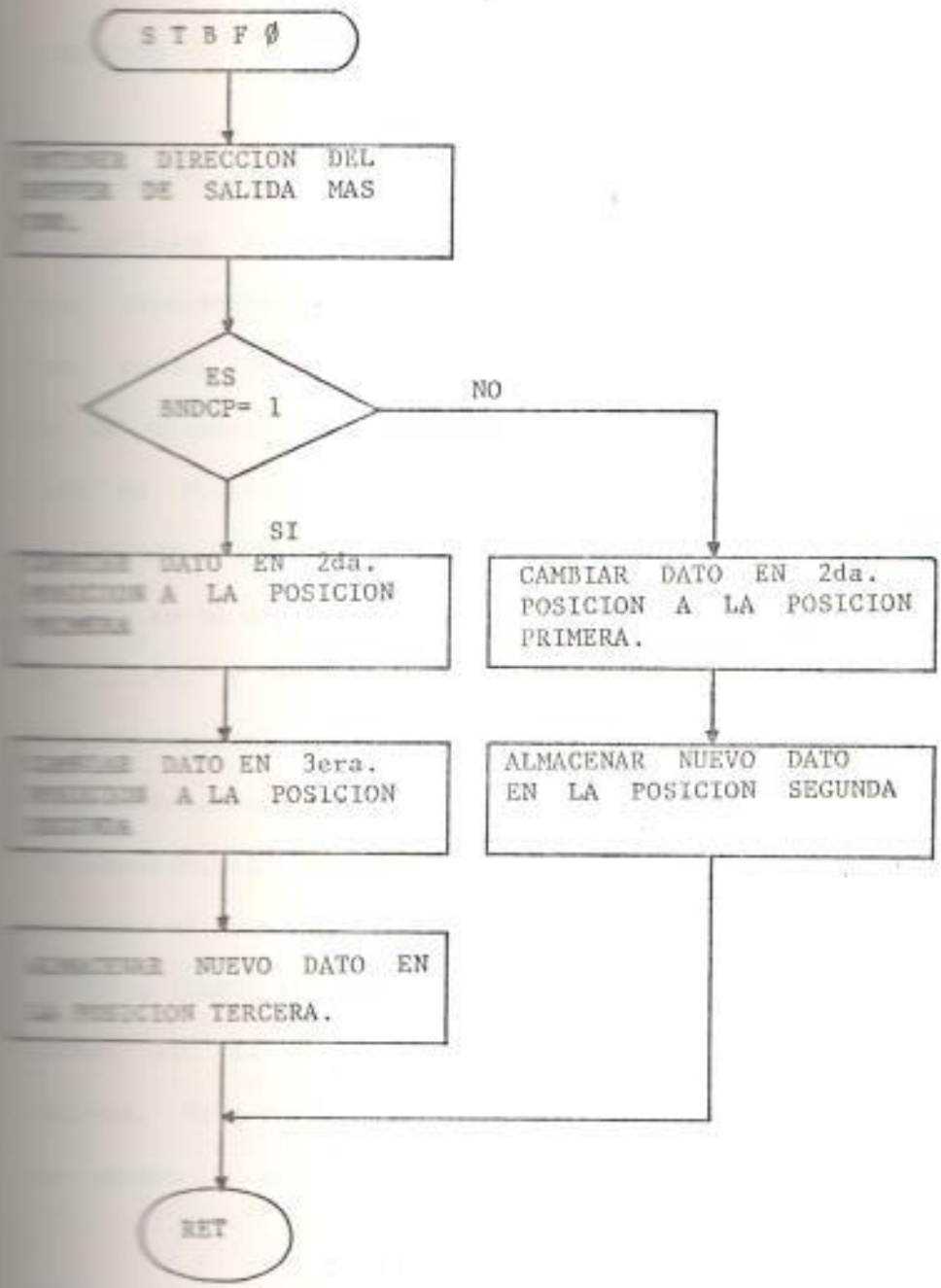


DIAGRAMA DE FLUJO DE LA RUTINA STBFØ

[illegible] - OBTENER DIGITOS HEXADECIMALES (FIG. 3.20)

[illegible]

[illegible] acepta digitos hexadecimales del teclado los cuales [illegible] enviados al visualizador tal como ellos son recibi[illegible]. dependiendo del valor de la bandera de campo es[illegible] pueden ser enviados al campo de datos o al cam[illegible]po de dirección. El primer paso es inicializar la ban[illegible]dera de digito hexadecimal, luego borramos el buffer [illegible]de salida y procedemos a leer caracter del teclado,- [illegible]si este es un digito hexadecimal lo almacenamos en el buffer [illegible]de salida, y colocamos bandera de digito hexadecimal . [illegible]El paso siguiente es enviar digito al visualizador. Los [illegible]digitos hexadecimales son terminados por un caracter de [illegible]información, el cual es retornado como una función de sa[illegible]lida via el acumulador. Si el caracter de información es [illegible] o (EXEC) los digitos obtenidos son considerados vá[illegible]lidos, por el contrario los digitos son considerados in[illegible]válidos. Esta rutina retorna una bandera indicando si - [illegible]se obtuvo o no digitos hexadecimales.

[illegible] - ALMACENAR DATO O DIRECCION (Fig. 3.21)

[illegible] -

[illegible] almacena caracteres que pueden ser datos o direc[illegible]ciones dependiendo del valor de su argumento de entrada.

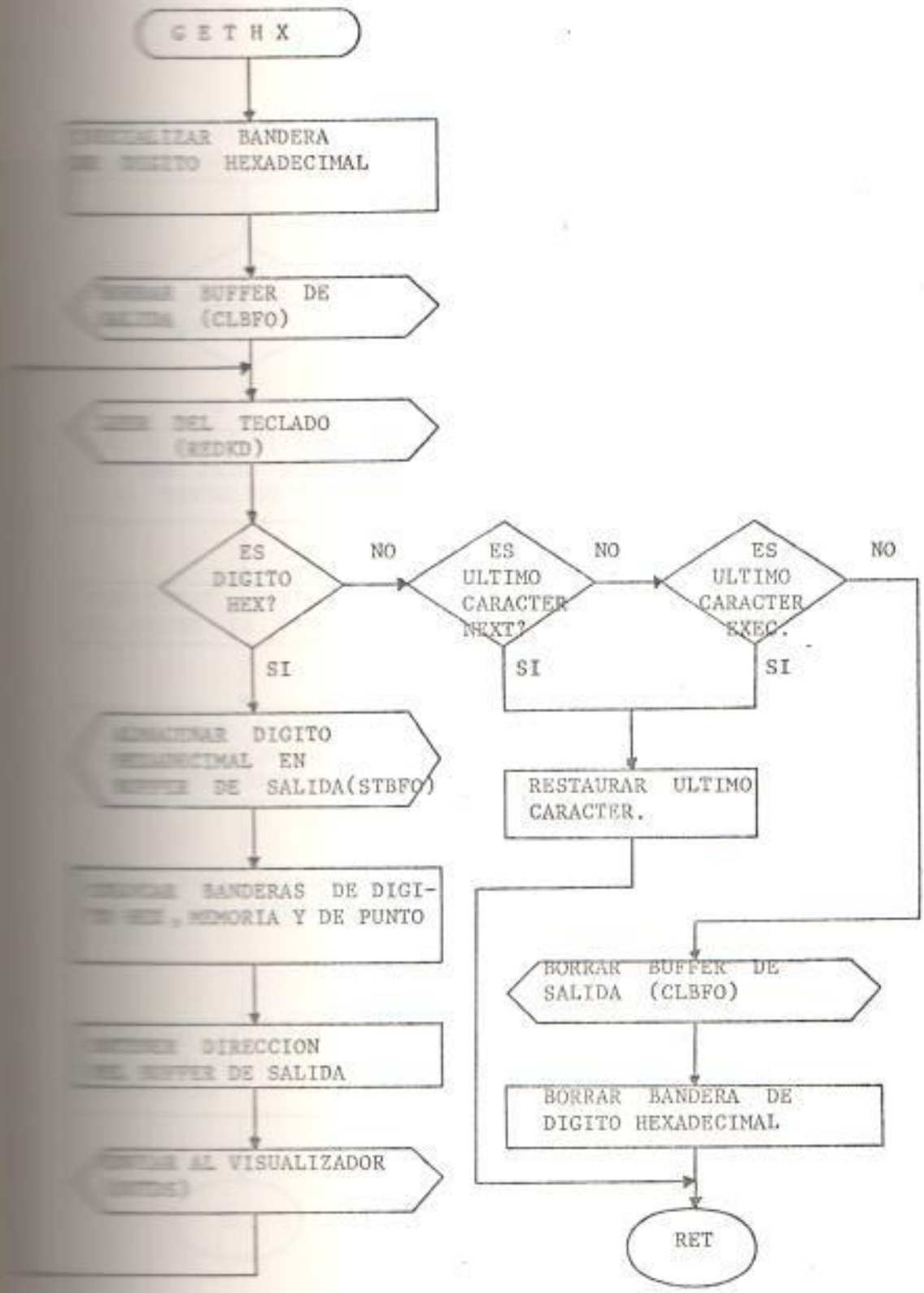


DIAGRAMA DE FLUJO DE LA RUTINA GETHX.

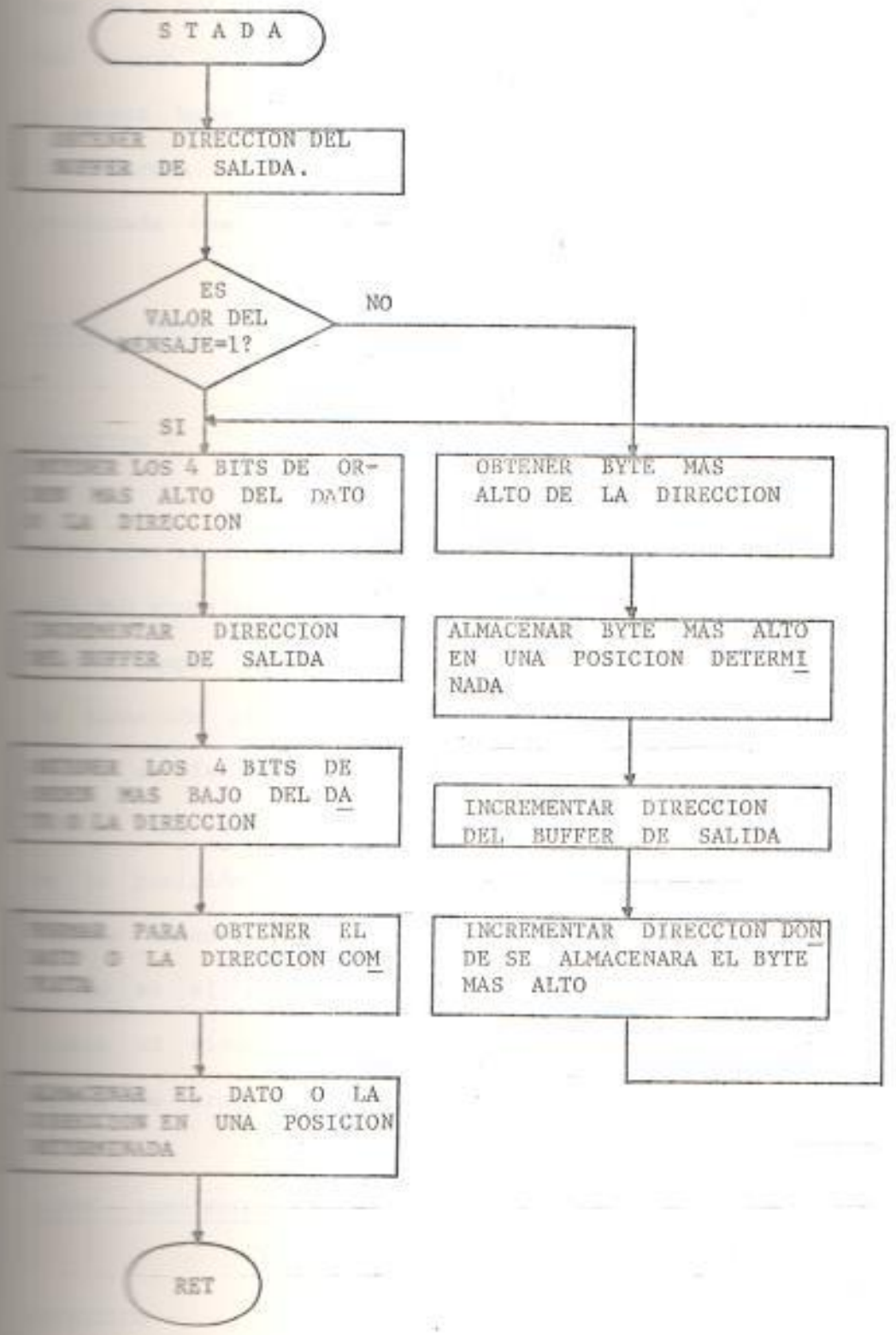


DIAGRAMA DE FLUJO DE LA RUTINA STADA .

[illegible] del buffer de salida los digitos hexadecimales y [illegible] agrupa en un sólo byte si se trata de un dato y en dos bytes si se trata de una dirección, luego - [illegible] almacena en una posición determinada la cual es es[illegible] especificada como una entrada de la rutina.

[illegible] - PONER AL DIA VISUALIZADOR (Fig. 3.22)

[illegible] -

[illegible] pone al día el campo de datos o el campo de direc[illegible] del visualizador dependiendo del valor de la ban[illegible] de campo. El campo de dirección es puesto al día con [illegible] dirección presente en la dirección común y el dato es [illegible] al día con el dato presente en la posición de dato [illegible]. Para hacer esto, esta rutina obtiene los bytes [illegible] la posición de dirección común y los separa en [illegible] hexadecimales, luego los almacena en el buffer de [illegible] en el orden correspondiente y después son en[illegible] al visualizador. El procedimiento es similar - [illegible] se trata de un dato.

[illegible] - LEER EPROM. (Fig. 3.23)

[illegible] -

[illegible] se usa para leer el dato del Eprom y luego almace-

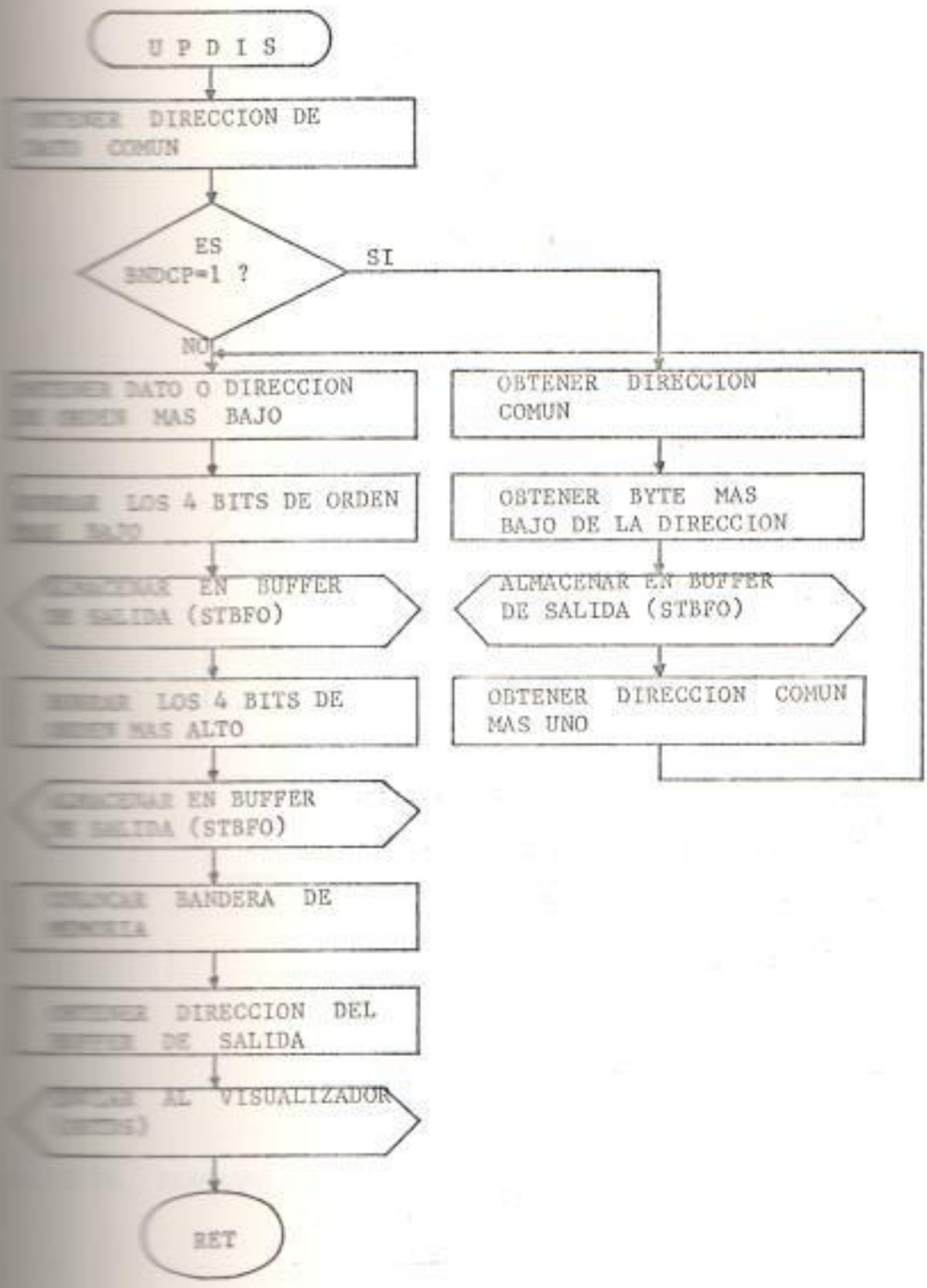


DIAGRAMA DE FLUJO DE LA RUTINA UPDIS.

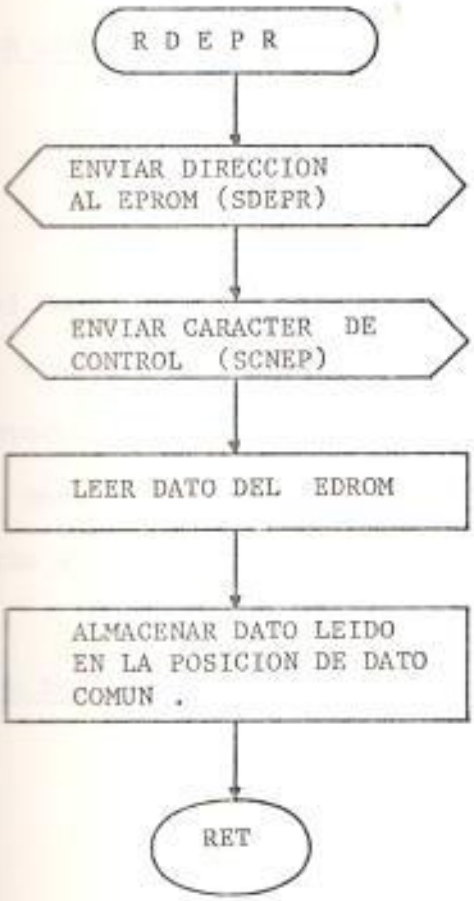


DIAGRAMA DE FLUJO DE LA RUTINA RDEPR.

[illegible] en la posición de dato común. Primero se envía la Di[illegible] al Eprom, y el caracter de control. El siguiente [illegible] es leer el Eprom y almacenar el carácter leído - [illegible] la posición de dato común.

[illegible] - ENVIAR CARACTER DE CONTROL . (Fig. 3.24)

[illegible] -

[illegible] envia el caracter de control para leer o escribir - [illegible] al registro de control. El primer paso es co[illegible] el caracter para habilitar el registro de control [illegible] la puerta uno, el siguiente es obtener caracter de con[illegible] y enviarlo al registro de control.

[illegible] - ENVIAR DIRECCION AL EPROM. (Fig. 3.25)

[illegible] -

[illegible] envia los dos bytes de la DIE a sus respectivos re[illegible] de dirección. Primero colocamos el carácter para [illegible] el registro de dirección del byte de orden más ba[illegible]. Luego obtenemos el byte de orden más bajo de la DIE y [illegible] al registro. El paso siguiente es colocar el ca[illegible] para habilitar el registro de dirección del byte de or[illegible] más alto, luego obtenemos el byte de orden más alto

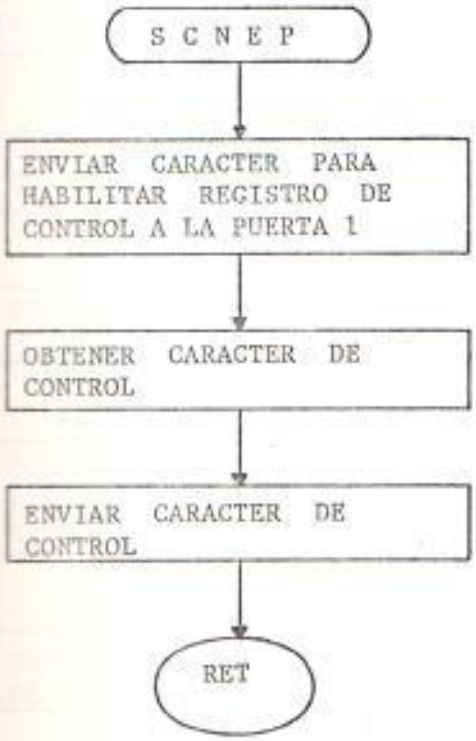


FIG. 3.24: DIAGRAMA DE FLUJO DE LA RUTINA SCNEP.

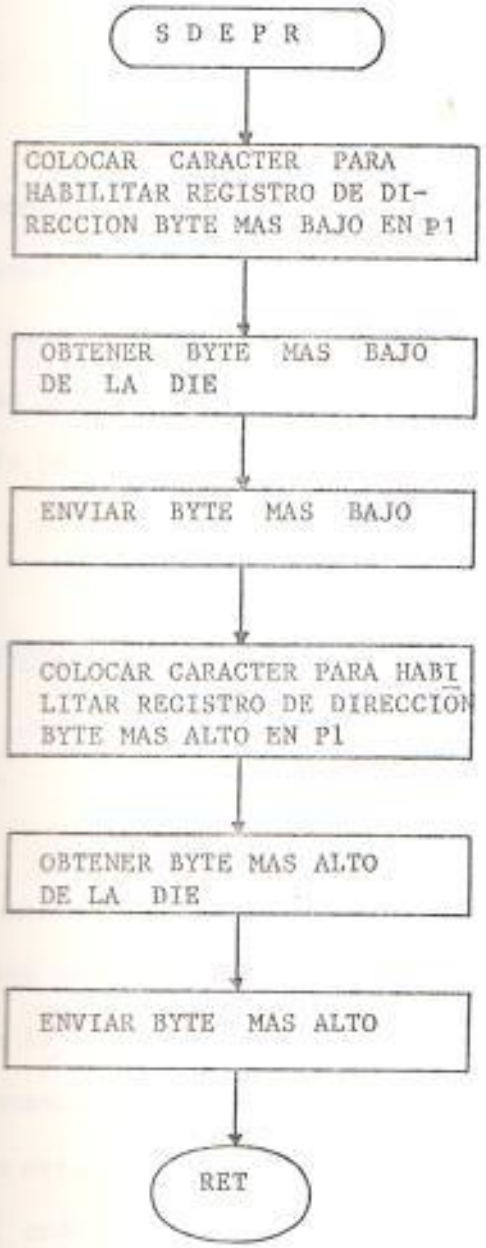


FIG. 3.25. DIAGRAMA DE FLUJO DE LA RUTINA SDEPR

de la DIE y lo enviamos al registro respectivo.

[illegible] - INCREMENTAR DIRECCIONES DEL EPROM. (Fig.3.26)

[illegible] -

[illegible] incrementa la dirección inicial del Eprom. El primer pa[illegible] obtener el byte de orden más bajo de la DIE y averi[illegible] si es igual a FF, si no es igual incrementamos el byte [illegible] más bajo, en caso contrario ponemos a cero el byte [illegible] más bajo e incrementamos el byte de orden más [illegible].

[illegible] - INCREMENTAR DIRECCIONES DE LA RAM. (Fig. 3.27)

[illegible] -

[illegible] incrementa la dirección inicial de la RAM. Al igual - [illegible] la rutina anterior, primero se obtiene el byte de orden [illegible] de la DIR y averiguamos si es igual a FF, si no lo [illegible] incrementamos el byte de orden más bajo. En caso contra[illegible] ponemos ceros al byte de orden más bajo e incrementamos [illegible] de orden más alto.

[illegible] - RUTINA DE LECTURA DEL EPROM. (Figura N°- 3.28)

[illegible] -

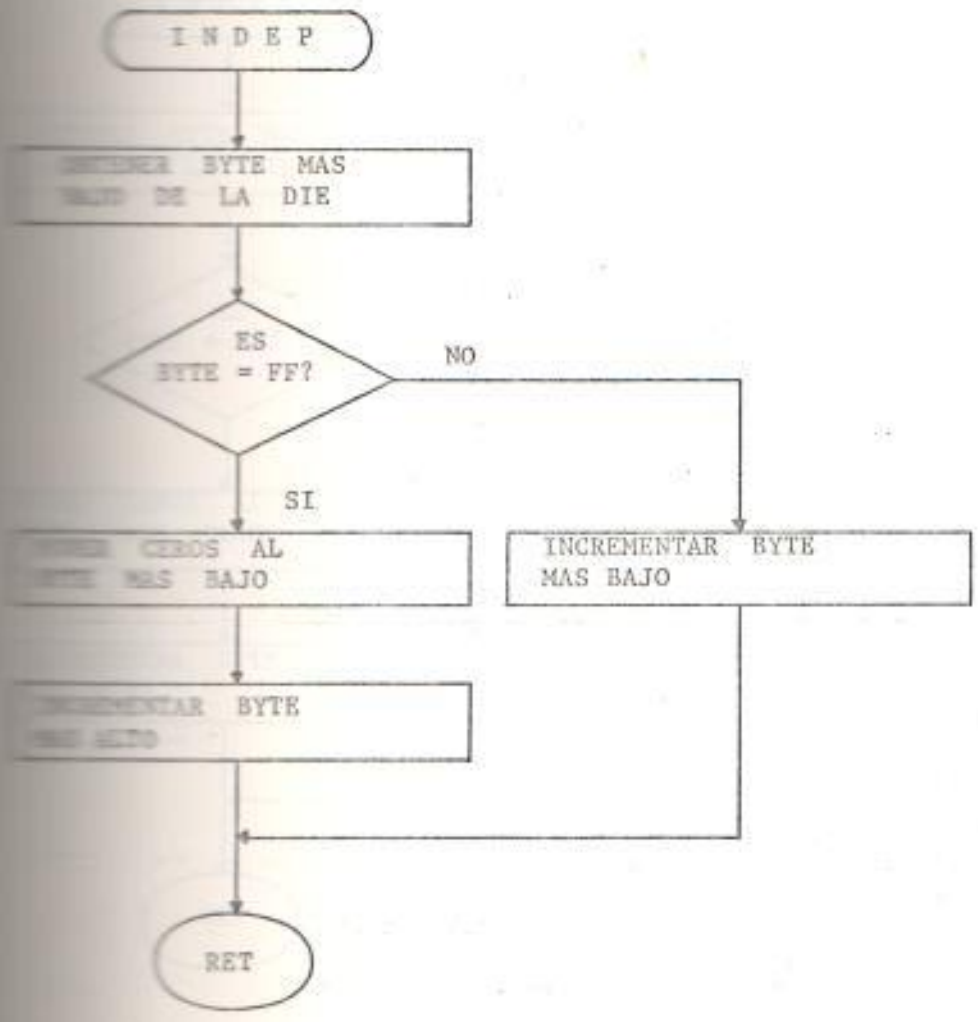


FIG. 3.26. DIAGRAMA DE FLUJO DE LA RUTINA INDEP

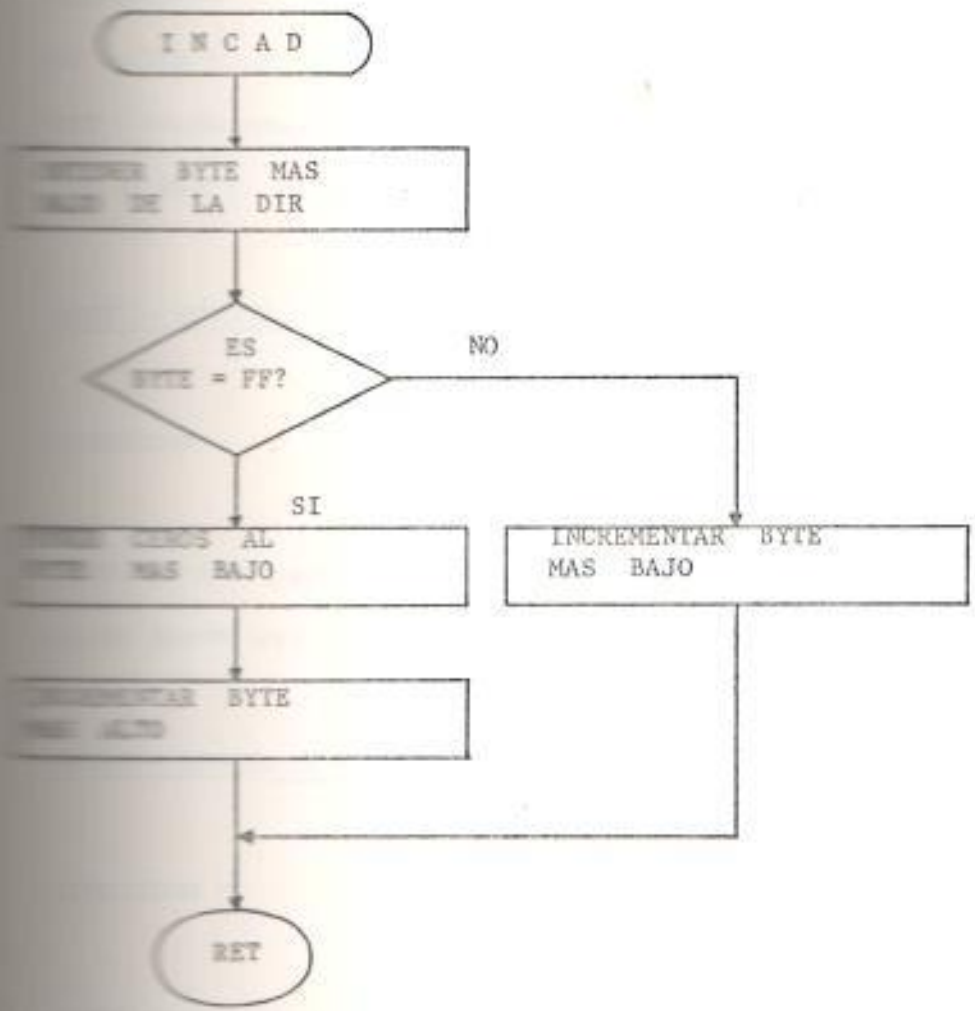


DIAGRAMA DE FLUJO DE LA RUTINA INCAD.

se utiliza para leer el Eprom. El primer paso es inhibir lectura del Eprom, luego se procede a cambiar el byte de más alto de la DIR, posteriormente leemos el Eprom y almacenamos el caracter leído en la memoria de datos externa. Si la DIE y la DFE no son iguales pasamos a incrementar la y regresamos a leer el proximo caracter del Eprom, hasta que la DIE y la DFE sean iguales.

- RUTINA INICIAL PARA OBTENER DATOS DE ENTRADA (Fig.3.29)

se usa para obtener los siguientes datos de entrada;tipo de Eprom que se va a leer o programar, DIE, DFE, DIR y DFR.

- MENSAJES A MOSTRAR EN EL VISUALIZADOR (Fig.3.30)

envia al campo de dirección del visualizador los siguientes mensajes: Tipo de Eprom (EPR), DIE, DFE, DIR y DFR. El campo de datos en blanqueado.

- RECONOCER TIPO DE EPROM. (Figura N°- 3.31)

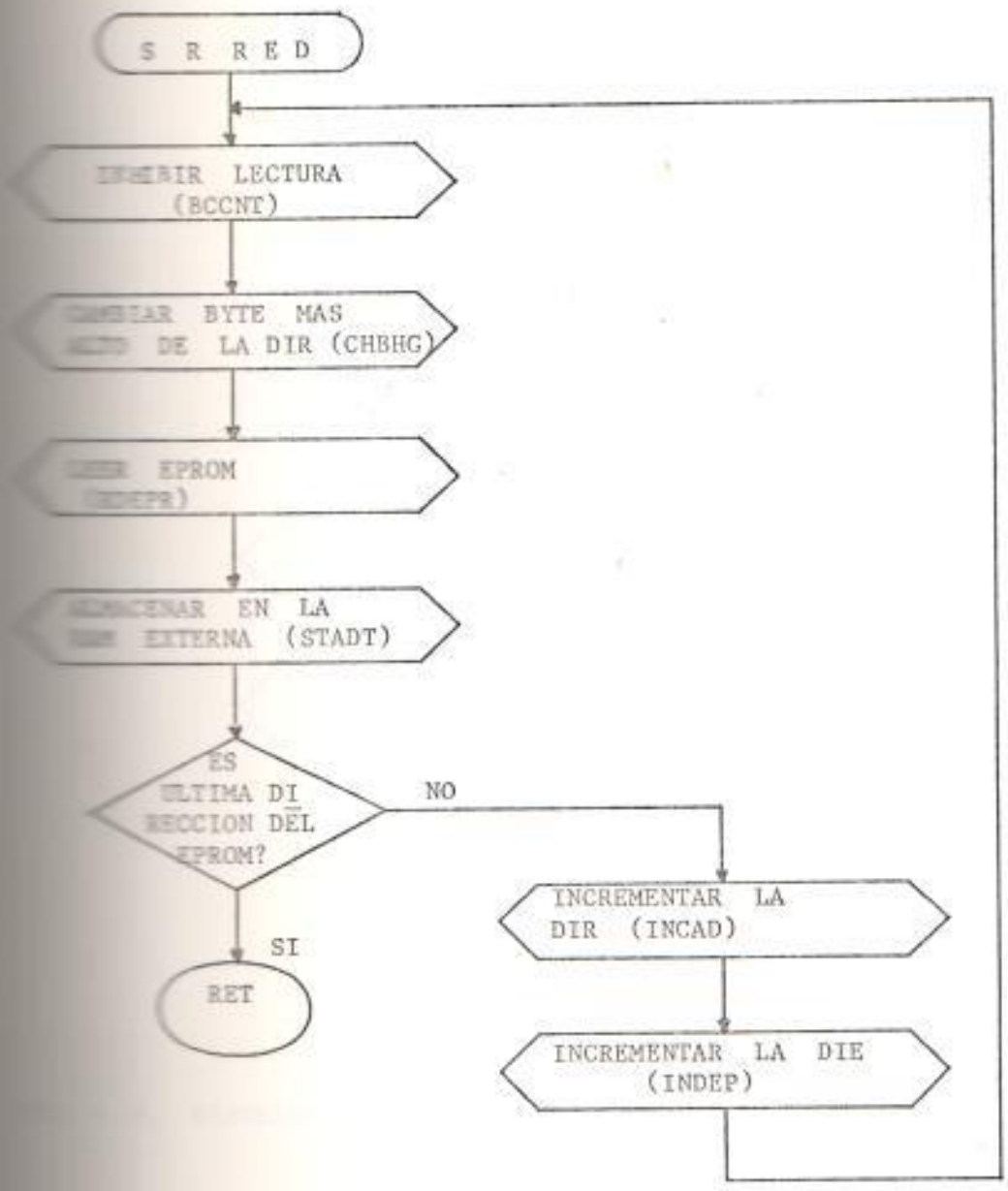
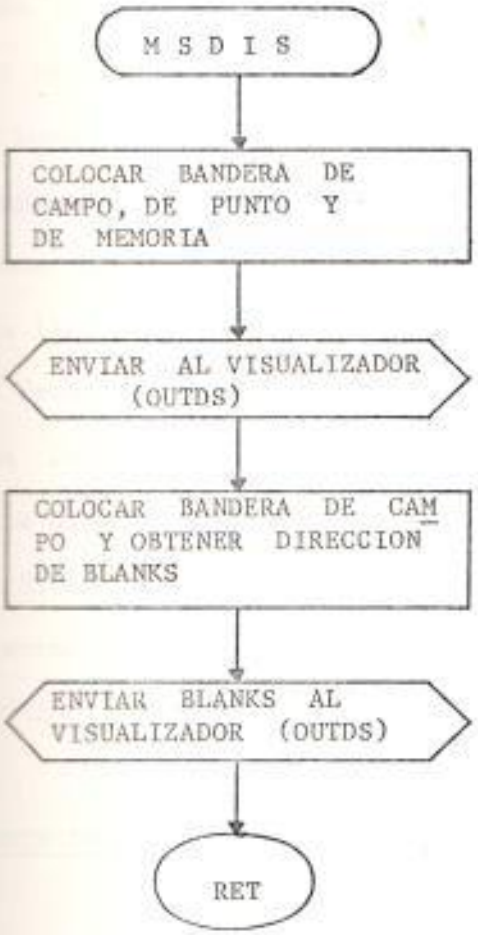


DIAGRAMA DE FLUJO DE LA RUTINA SRRED.

DIAGRAMA DE FLUJO DE LA RUTINA SRINC



[illegible] DIAGRAMA DE FLUJO DE LA RUTINA MSDIS

[illegible] reconoce el tipo de Eprom y dependiendo del valor de - [illegible] argumento de entrada obtiene el carácter de control para leer o escribir el Eprom. El primer paso es colocar el número de Eproms en un registro, obtener la dirección donde se almacena el tipo de Eproms leído desde el teclado y obtener la dirección de la tabla - de Eproms. Luego se procede a obtener el Eprom de la tabla y compararlo con el tipo de Eprom leído anteriormente, si los dos son iguales se obtiene el caracter de control para leer o escribir dependiendo del valor de la bandera de lectura / escritura. Si el tipo de Eprom - leído no es igual al Eprom obtenido de la tabla se incrementa la dirección de la tabla siempre y cuando - no sea el último Eprom, si es el último entonces un mensaje de error es mostrado en el visualizador indicando que el tipo de Eprom entrado no es el correcto.

[illegible] - OBTENER DATOS DE ENTRADA (Fig. 3.32)

[illegible] -

[illegible] se usa para enviar mensajes al visualizador y obtener el dato o la dirección en respuesta al mensaje enviado. Dependiendo del valor de su argumento de entrada, esta rutina selecciona el mensaje a ser enviado y luego espera que un dato o una dirección sea entra

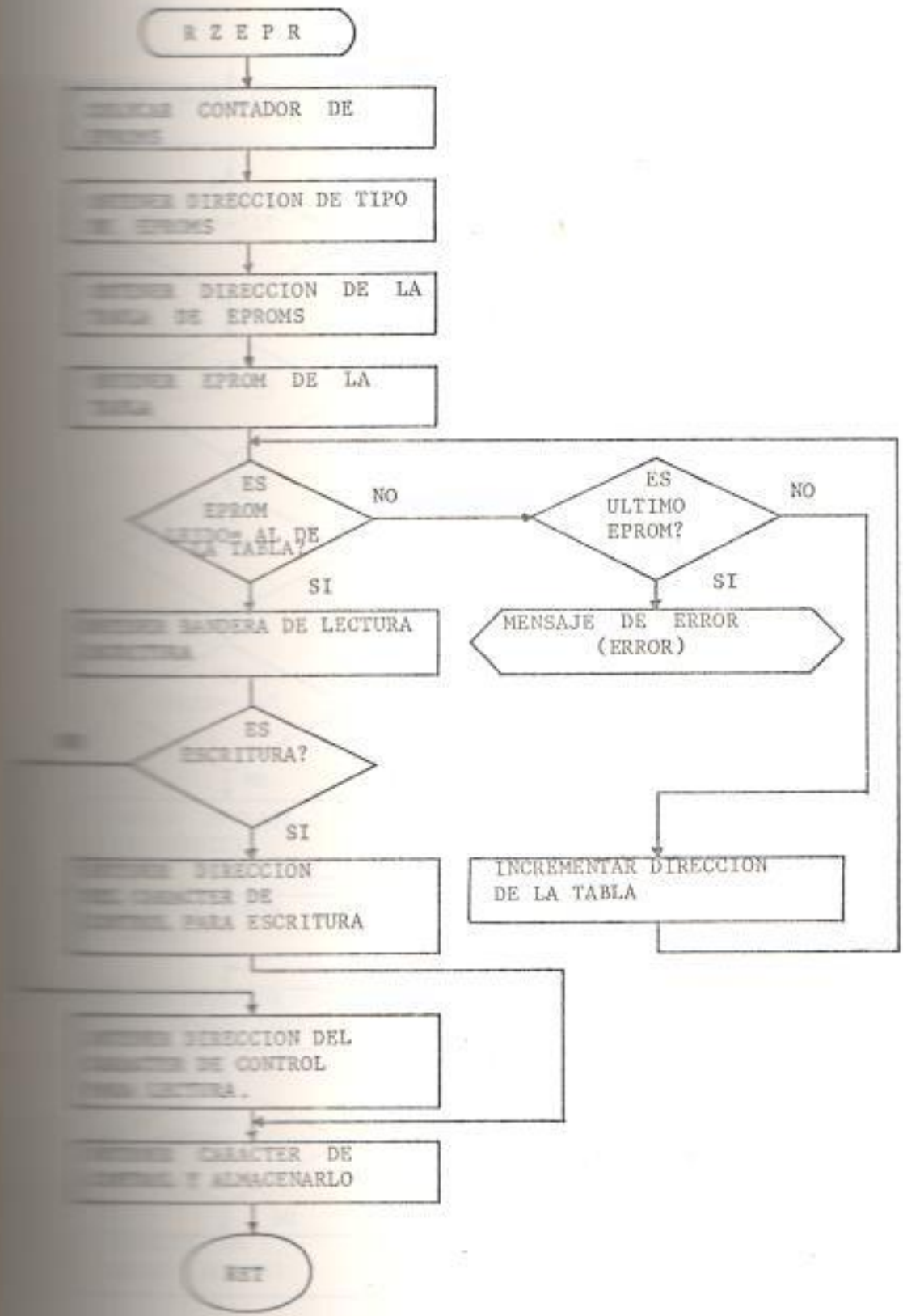


DIAGRAMA DE FLUJO DE LA RUTINA RZEPR.

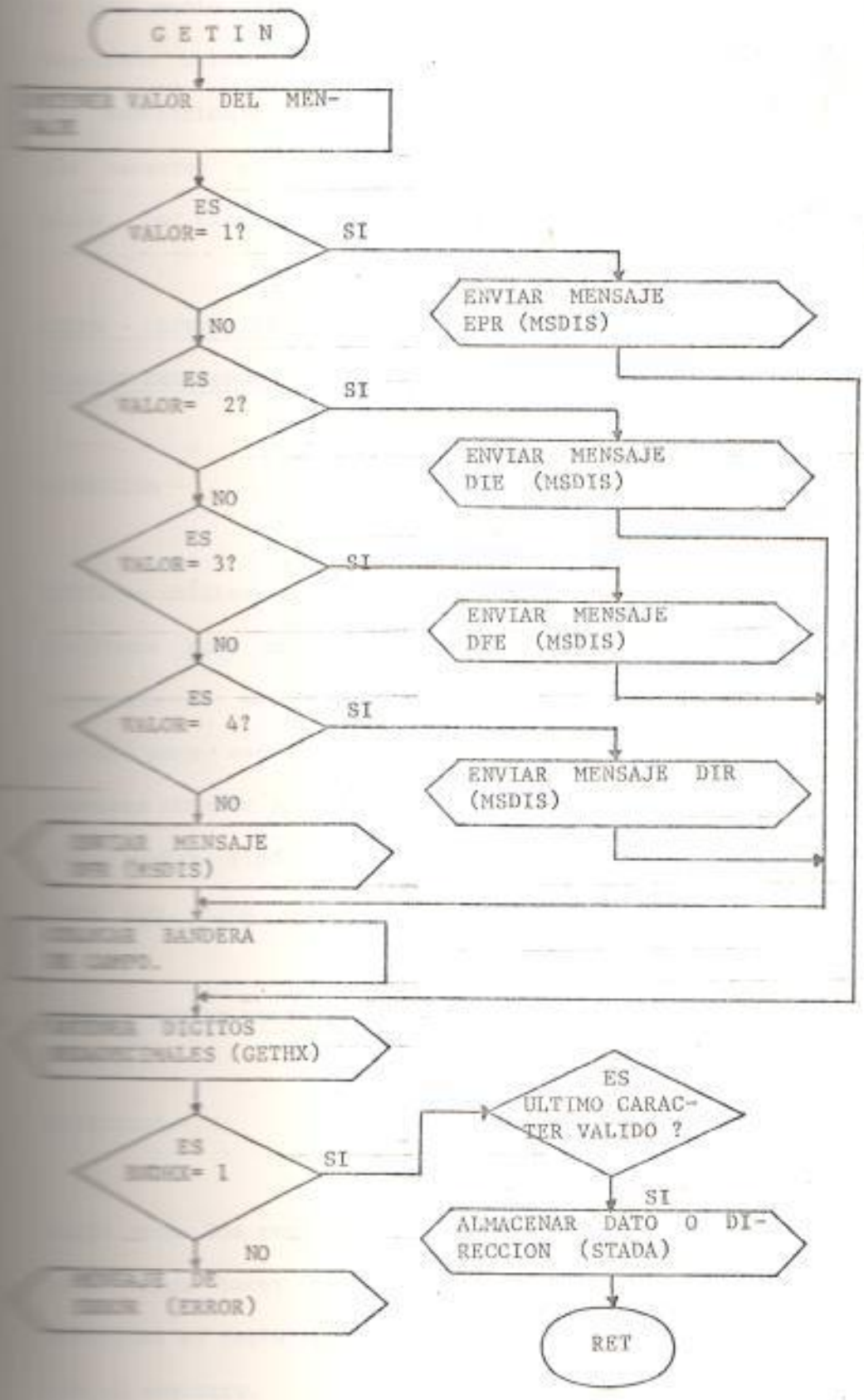


DIAGRAMA DE FLUJO DE LA RUTINA GETIN

[illegible] por medio del teclado, posteriormente el dato o la [illegible] obtenida es almacenada en una posición de memo[illegible] especificada como una entrada de la rutina. Si nin[illegible] caracter es obtenido en respuesta al mensaje en[illegible], un mensaje de error es mostrado.

[illegible] - OBTENER TIPO DE EPROM. Y DIRECCIONES INICIALES Y [illegible] DEL EPROM Y LA MEMORIA RAM. (Fig.3.33)

[illegible] -

[illegible] se utiliza para obtener los datos de entrada es[illegible] para cada comando. Dependiendo del valor del [illegible] de entrada esta rutina obtiene los datos nece[illegible] para ese comando. Por ejemplo supongamos que [illegible] obtener los datos de entrada para el comando PROGM, [illegible] serían: Tipo de Eprom a escribir (EPR), DIE, DIR y DFR. [illegible] los otros comandos el caso es similar.

[illegible] - BORRAR VOLTAJES DE PROGRAMACION (Fig. 3.34)

[illegible] -

[illegible] borra los registros de dirección y el registro de con[illegible]. Para hacer esto primero se coloca el caracter para [illegible] el registro correspondiente y luego se envian ce[illegible] al registro.

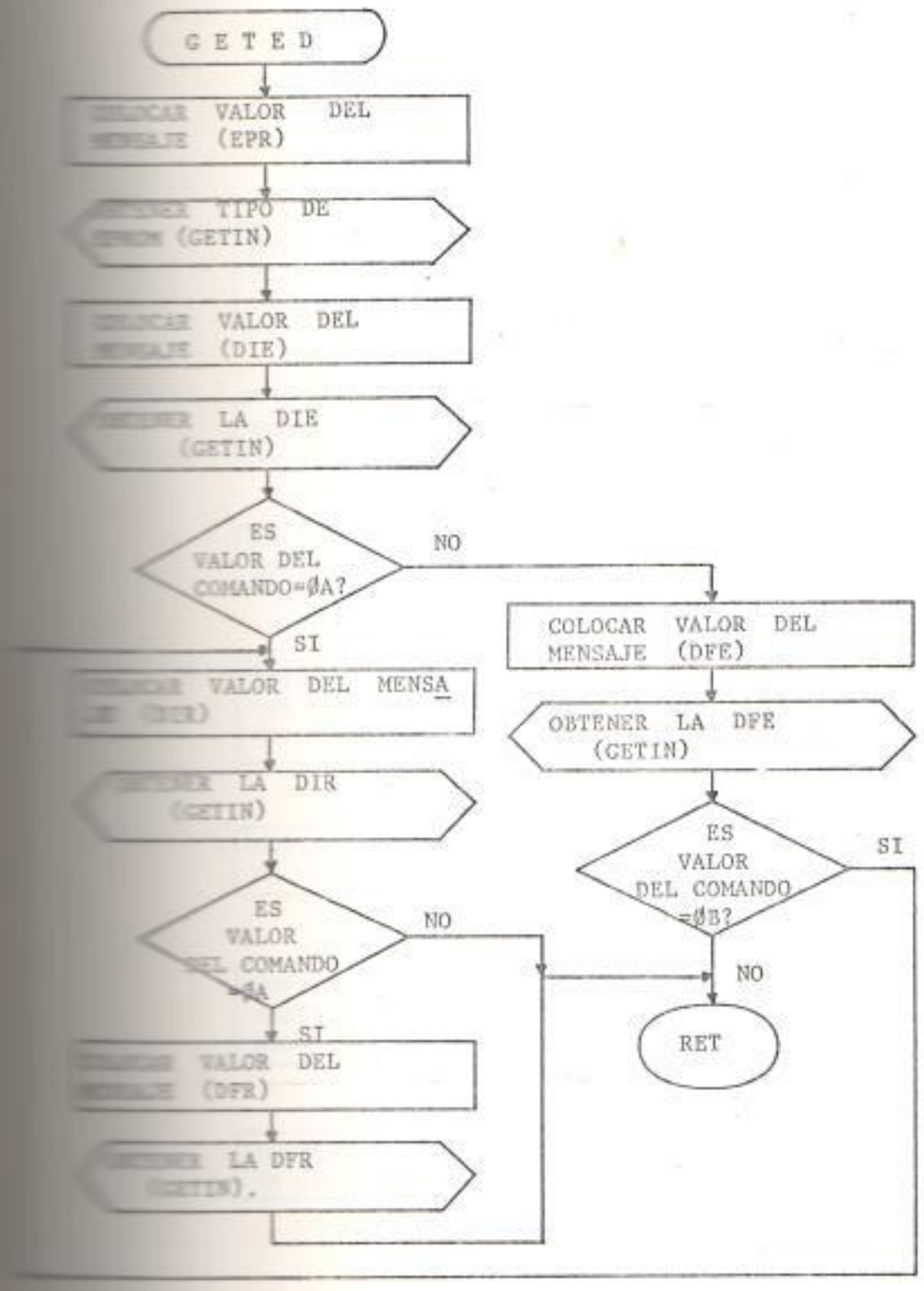


DIAGRAMA DE FLUJO DE LA RUTINA GETED.

DIAGRAMA DE FLUJO DE LA RUTINA CLVOL.

[illegible] - RUTINA DE COMPROBACION DE LA PROGRAMACION (Fig.3.35)

[illegible] se utiliza para comprobar la programación correcta del Eprom. En caso de que la información grabada no sea igual a la información almacenada en la memoria RAM, esta rutina mostrará en el visualizador la dirección y el dato grabado en esa dirección. Para hacer esto primero inhibimos la lectura del Eprom y cambiamos el byte de orden más alto de [illegible], luego leemos la información almacenada en la memoria RAM y la información grabada en el Eprom. El paso siguiente es comparar si los dos caracteres leídos son iguales, si lo son se pasa a comparar la siguiente dirección, caso contrario en el visualizador se mostrará la dirección y el dato almacenado en esa dirección. Si la DIE no es igual a la DFE, se incrementa la DIE y la DIR, y regresamos a comparar la siguiente dirección.

[illegible] - MOSTRAR DATO Y DIRECCION (Fig. 3.36)

[illegible] -

[illegible] muestra en el visualizador la dirección y la información almacenada en esa dirección del Eprom mal programado o del Eprom mal borrado. Para lograr hacer esto primero la DIE es separada en digitos hexadecimales, los cuales después son enviados al buffer de salida en el orden correspondien

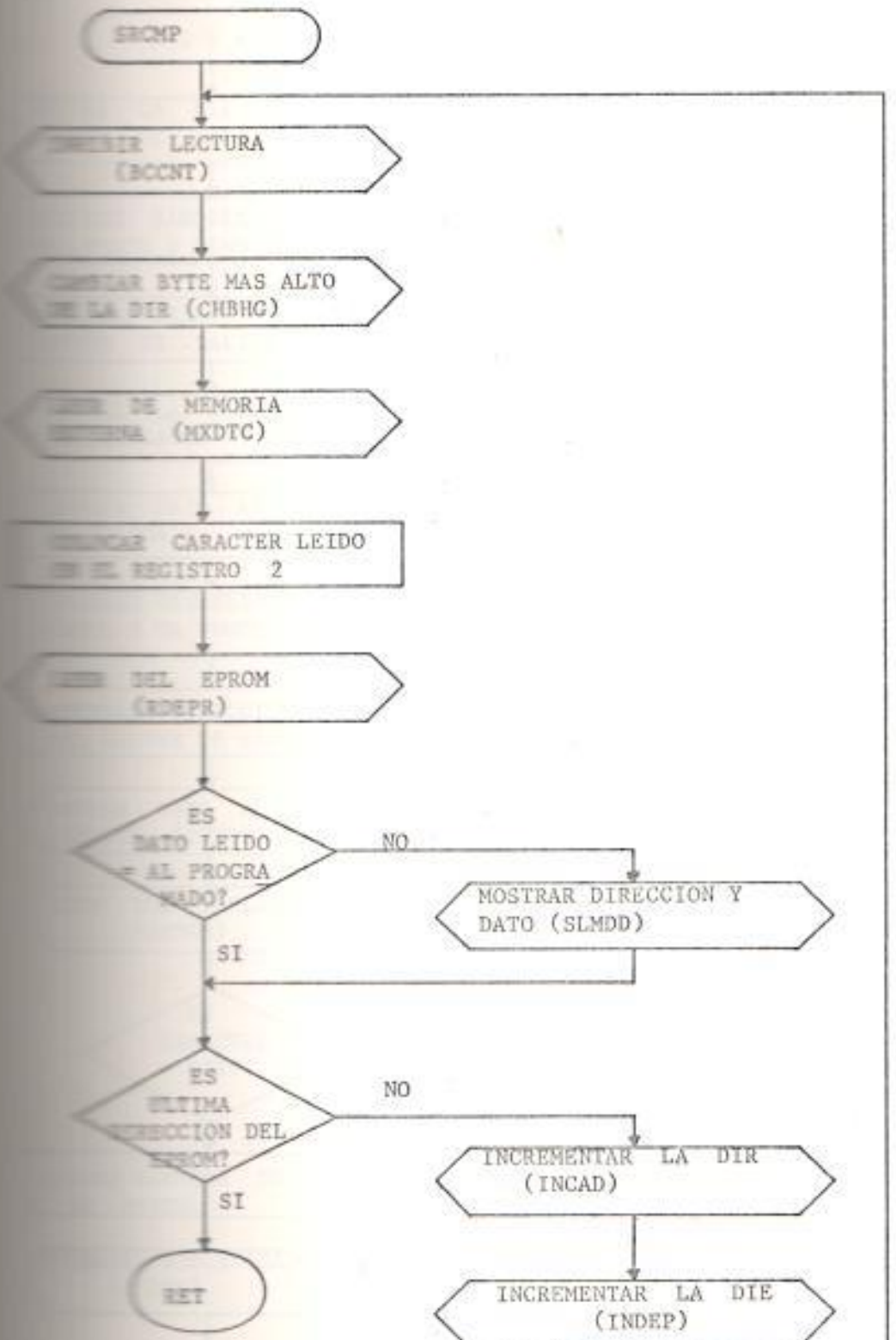


DIAGRAMA DE FLUJO DE LA RUTINA SRCMP.

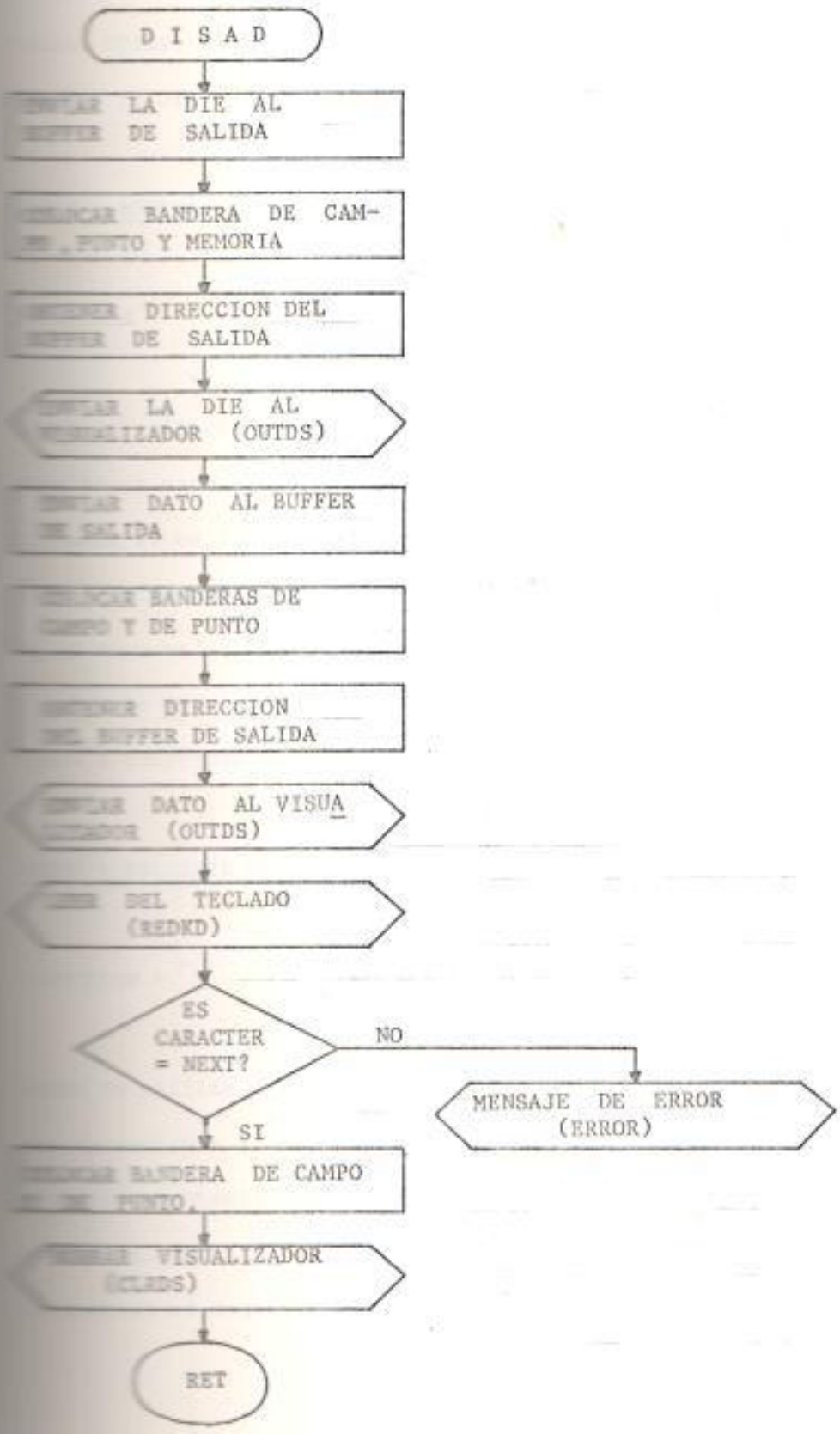


FIG. 3.36. DIAGRAMA DE FLUJO DE LA RUTINA DISAD.

[illegible] para luego ser enviados al visualizador por medio de la [illegible] OUTDS. En igual forma se procede con el dato almacenado en la DIE.

[illegible] - RUTINA DE COMPROBACION DEL BORRADO DEL EPROM(Fig.3.37)

[illegible] -

[illegible] se utiliza para comprobar el borrado correcto del Eprom. Para hacer esto leemos el caracter almacenado en el Eprom y averiguamos si es igual a FF, si no lo es en el visualizador aparecerá la dirección y la información almacenada en esa dirección. Si el caso es el contrario entonces se averigua si la DIE es igual a la DPE, si no son iguales se incrementa la DIE y regresamos a comprobar la siguiente dirección.

[illegible] - ESCRIBIR DATOS AL EPROM - (Fig. 3.38)

[illegible] -

[illegible] Se utiliza para enviar datos al Eprom. Primero se envían las direcciones a los registros de dirección, el dato a ser escrito es obtenido de la posición de dato común y luego es enviado al registro de datos. Por último se envia el caracter de control al registro de control.

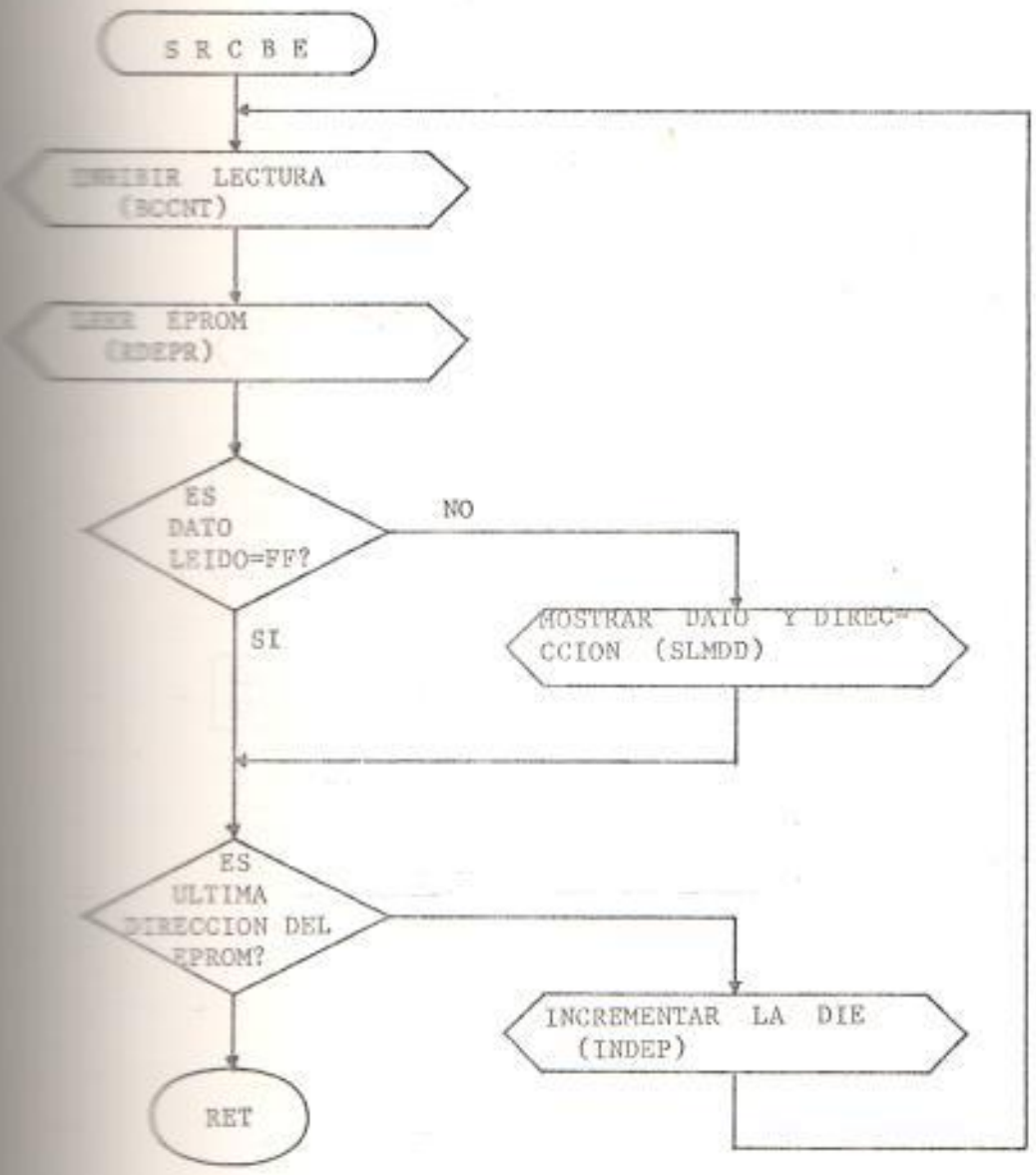


FIG. 3.37. DIAGRAMA DE FLUJO DE LA RUTINA INDEP.

DIAGRAMA DE FLUJO DE LA RUTINA WREPR

[illegible] - INHIBIR LECTURA O PROGRAMACION DEL EPROM(Fig.3.39)

[illegible]CCION -

[illegible] inhibe la lectura o la programación del Eprom. Esta ru[illegible] coge el Eprom obtenido por medio del teclado y lo com[illegible] con el Eprom obtenido de la tabla, si los dos son igua[illegible], se obtiene el caracter de control para inhibir lectura [illegible] programación del Eprom, dependiendo del valor que tenga - [illegible] bandera de lectura/escritura. Una vez obtenido el carácter [illegible] control éste es enviado al registro de control.

[illegible] - SELECCIONAR DIRECCION Y DATO PARA SER ENVIADOS AL VI-[illegible]LIZADOR O AL MICROCOMPUTADOR (Fig. 3.40).

[illegible]CCION --

[illegible] selecciona la rutina apropiada para enviar la dirección [illegible] el dato del Eprom mal programado o del Eprom mal borrado,al [illegible]alizador o al microcomputador.

[illegible] - ALMACENAR DIRECCION INICIAL DE LA RAM Y DIRECCION INI-[illegible]AL DEL EPROM (Fig. 3.41).

[illegible]CCION -

[illegible] almacena los dos bytes de la DIE y la DIR en

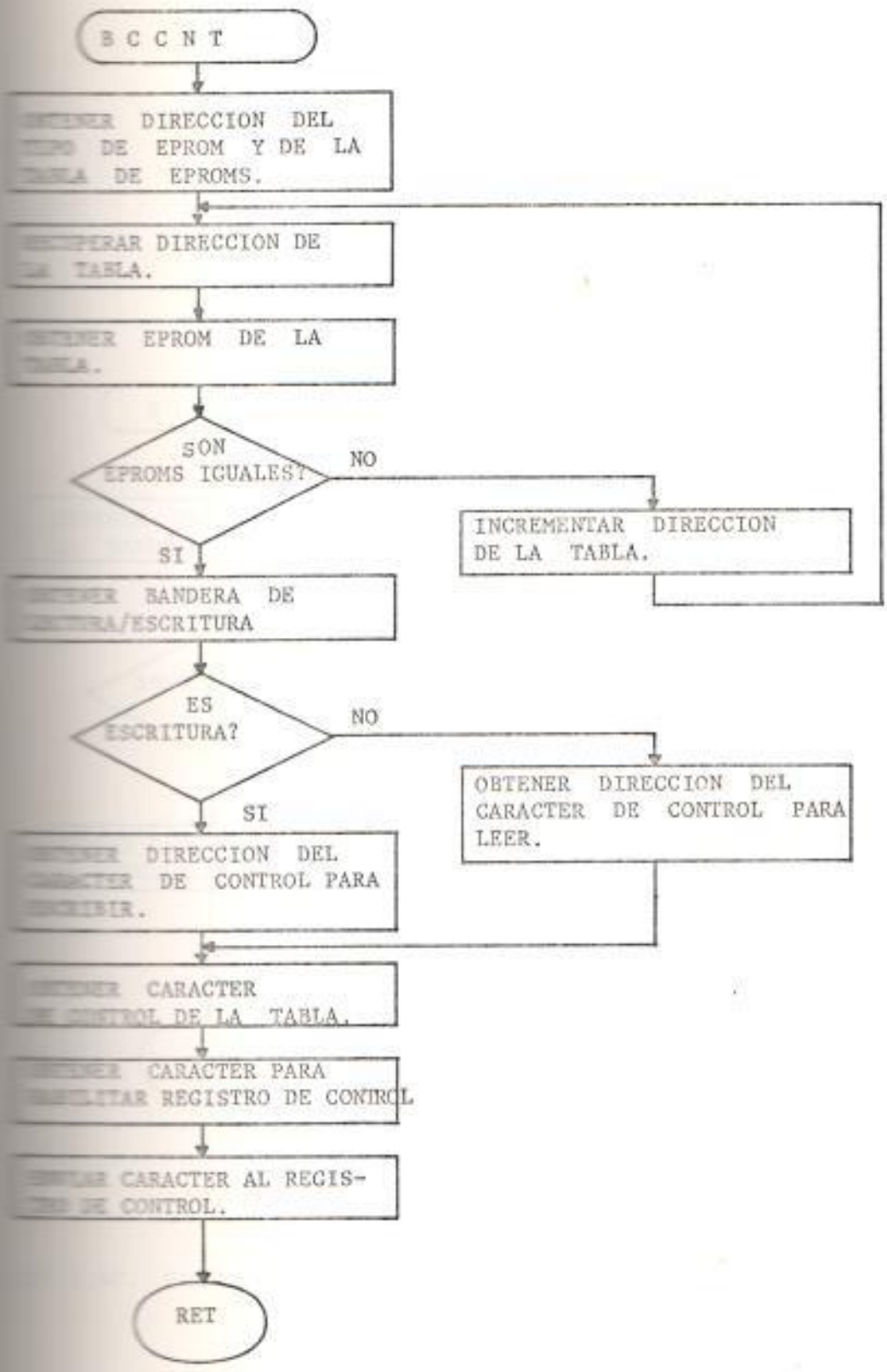


FIG. 3.39. DIAGRAMA DE FLUJO DE LA RUTINA BCCNT.

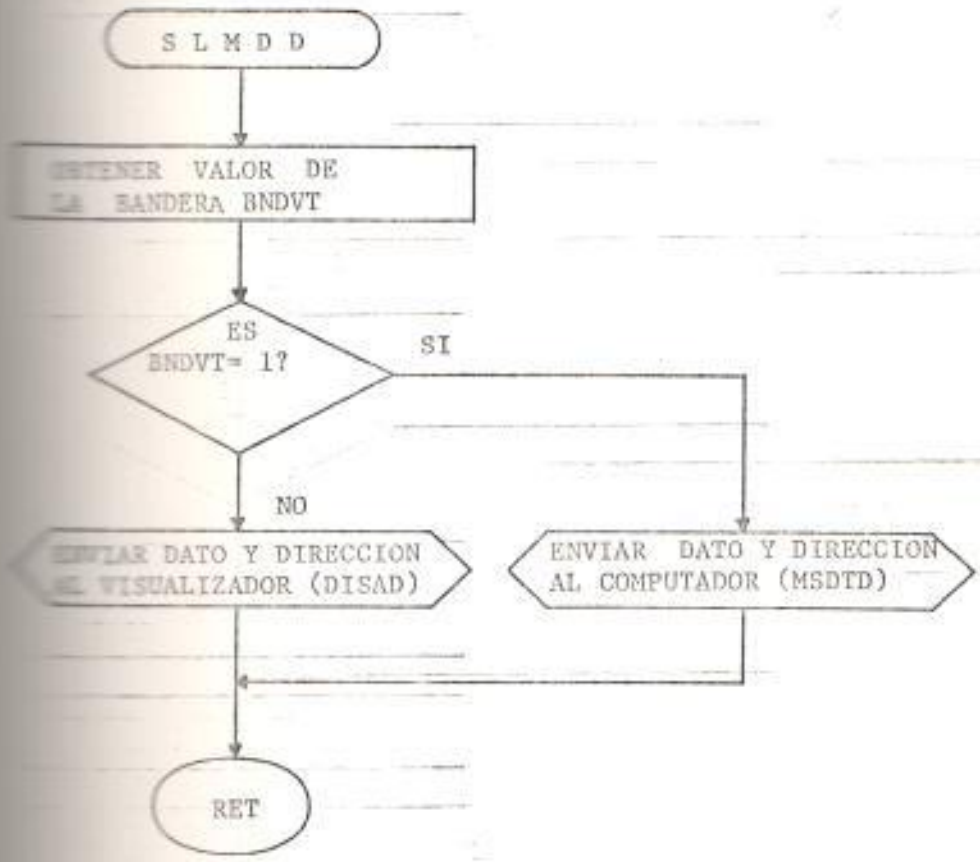


FIG. 3.40. DIAGRAMA DE FLUJO DE LA RUTINA SLMDD.

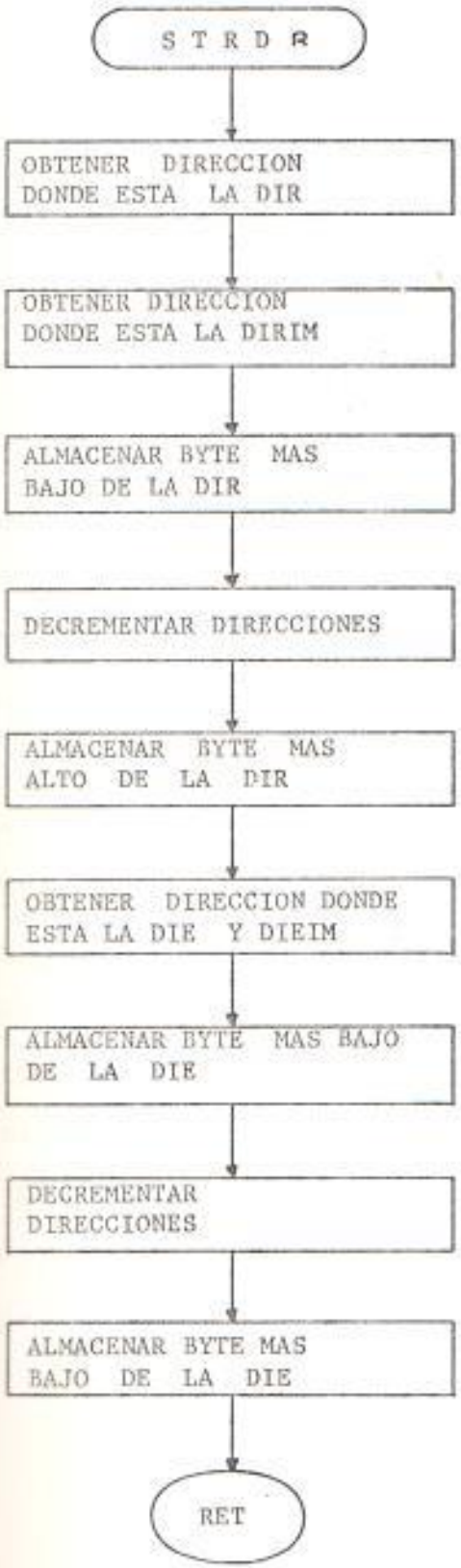


DIAGRAMA DE FLUJO DE LA RUTINA STRDR

dos pares de registros imagen para el caso de programar el Eprom 2708.

EPROM - RUTINA DE PROGRAMACION DEL EPROM. (Figura 3.42)

DESCRIPCION -

EPROM se utiliza para grabar datos en el Eprom. Primero se inicia la programación del Eprom, cambiamos el byte de orden más alto de la DIR y leemos el dato a grabar de la memoria de datos externa. Luego se envían las direcciones, el dato y el carácter de control al Eprom, el paso siguiente es averiguar si se trata del Eprom 2708, si este es el caso entonces colocamos el lazo de retardo para el EPROM 2708, en caso contrario colocamos el lazo de retardo para el resto de los Eproms. Si la DIR y la DFR no son iguales, incrementamos la DIR y la DIE y retornamos a grabar un nuevo dato, en caso de que sean iguales se averigua si es el último lazo,si lo es retornamos de esta rutina, en caso contrario colocamos a la DIR y a la DIE los valores que tenían al inicio antes de grabar el primer dato, y procedemos a grabar el siguiente lazo.

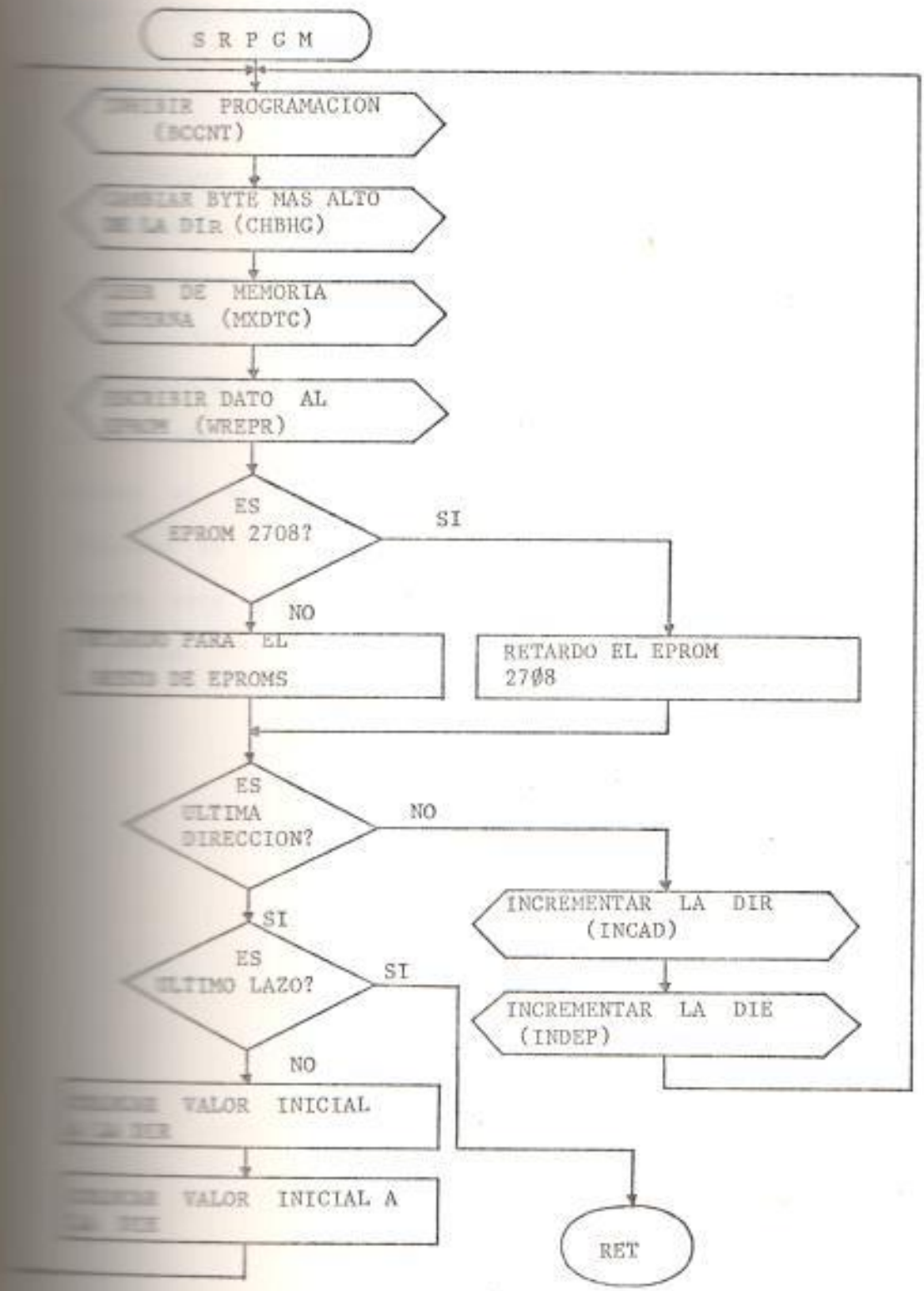


DIAGRAMA DE FLUJO DE LA RUTINA SRPGM

- CHEQUEAR DIRECCIONES INICIALES Y FINALES. (FIG.3.43)

-

chequea si la dirección final es mayor o igual a
inicial. Si la dirección inicial es mayor
dirección final un mensaje de Error es mostrado
visualizador. La comparación se la realiza sacando el
a dos de la dirección inicial y luego sumarla a
final, esta operación coloca la bandera de
valor correspondiente según el caso. El paso
es chequear el valor de la bandera de acarreo,
este mostrar un mensaje de error o retornar de

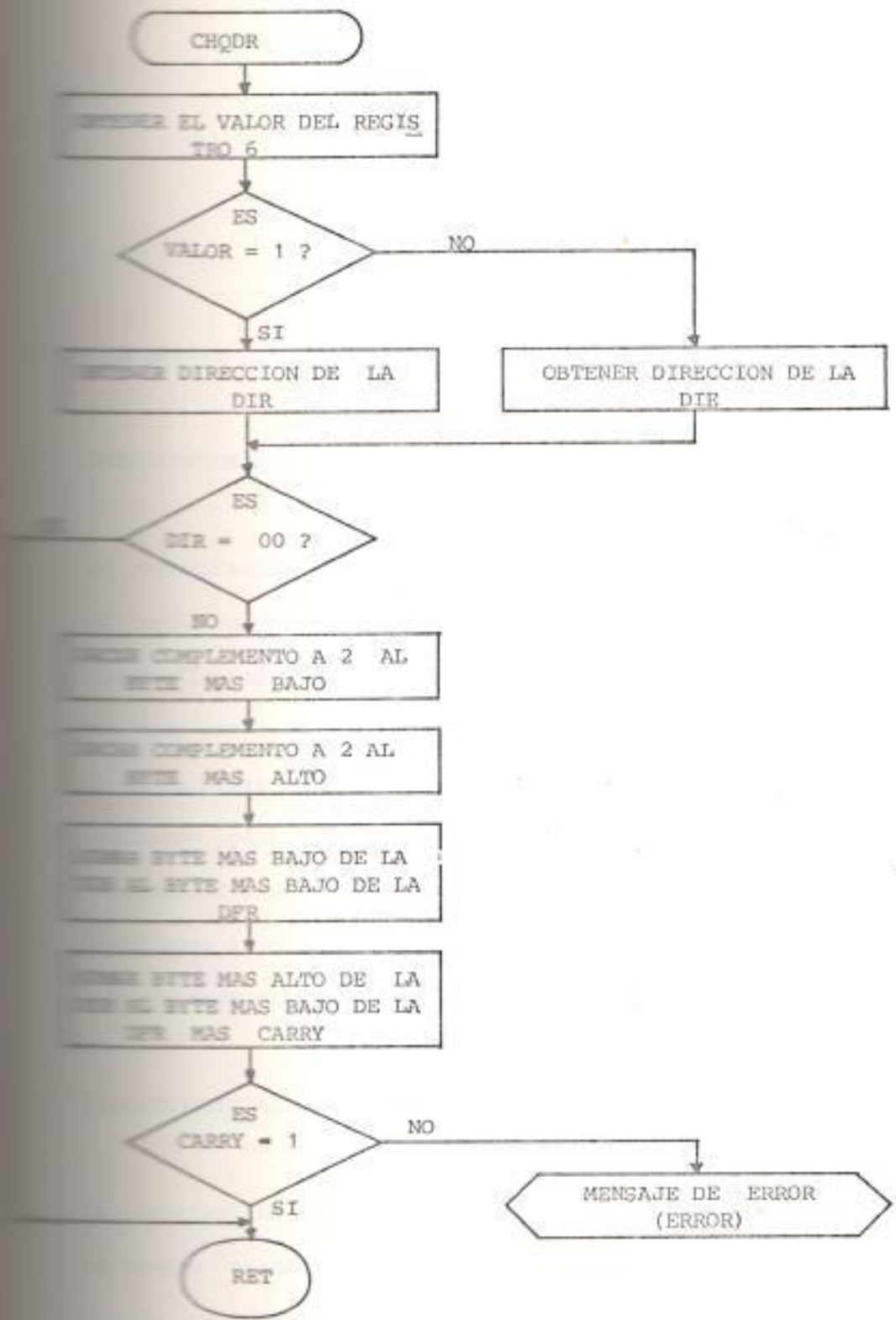


FIGURA Nº-3.43. DIAGRAMA DE FLUJO DE LA RUTINA CHQDR

C A P I T U L O IV

[illegible] RS-232

[illegible]INTRODUCCION

La comunicación del programador con un microcomputador es hecha a través de un puerto RS-232. Las características físicas y eléctricas, así como también la descripción de cada una de las señales de interface serán estudiadas en forma breve en la próxima sección. El 8251 (USART) es utilizado para la transferencia de información entre el programador y el microcomputador. El software que permite establecer la comunicación será presentado en la última sección de este capítulo.

[illegible]CARACTERISTICAS GENERALES DE LA INTERFACE

La interface RS-232 estandar define la comunicación entre un equipo terminal de datos (ETD) y un equipo de comunicación de datos (ECD), empleando intercambio de datos

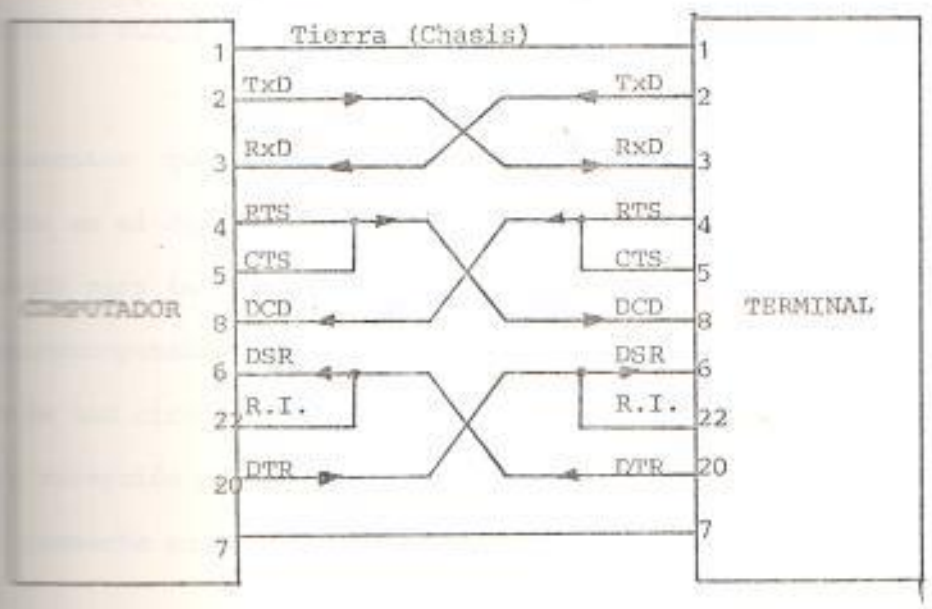
TABLA Nª 4.1.

SEÑALES DE LA INTERFACE RS-232

	I	DESCRIPCION
		Tierra
		Transmisión de datos (TxD)
		Recepción de datos (RxD)
		Solicitud para enviar (RST)
		Listo para enviar (CTS)
		Listo el dato (DSR)
		Tierra común (GND)
		Detector de la señal recibida en la línea (DCD)
		Reservada para prueba de datos
		Reservada para prueba de datos
		No asignada
		Línea secundaria de la señal DCD
		Línea secundaria de la señal CTS
		Línea secundaria de transmisión de datos
		Señal de transmisión del elemento de tiempo (Fuente ECD)
		Línea secundaria de recepción de datos
		Señal de recepción del elemento de tiempo (Fuente ECD)
		No asignada
		Línea secundaria de la señal RTS
		Listo el terminal de datos (DTR)
		Detector de la calidad de la señal
		Indicador Ring
		Selector de la razón de la señal de datos(Fuente ETD/ECD)
		Señal de transmisión del elemento de tiempo (Fuente ETD)
		No asignada

[illegible] en serie. La interface completa consiste de 25 - [illegible] de datos, muchas de estas líneas son bien especifica[illegible] y unas pocas son indefinidas. Muchos terminales de com[illegible] solamente requieren de 3 a 5 de estas líneas para [illegible]. La tabla Nª 4.1., describe cada una de las 25 [illegible].

La conexión física entre un ETD y un ECD es hecha a [illegible] de un conector de 25 terminales. El conector ma[illegible] es siempre asociado con el ETD y la hembra es siempre - [illegible] con el ECD. La figura Nª 4.1., muestra la conexión [illegible] un computador y un terminal, unas pocas líneas de[illegible] ser trocadas si ningún moden u otro equipo de co[illegible] de datos es usado. Esta conexión es llama[illegible] de un cable de moden nulo.



[illegible] Nª 4.1. SEÑALES USADAS ENTRE UN COMPUTADOR Y UN TERMINAL

Las especificaciones eléctricas de la interface RS-232 son las siguientes:

1. Los drivers deben ser capaces de resistir circuítos abiertos y cortocircuítos entre cualesquiera de los terminales de la interface.

2. Los voltajes bajo -3V serían llamados potenciales mark (lógico 1) y los voltajes arriba de +3V son llamados voltajes space (lógico 0).

3. La razón de transmisión máxima es de 20.000 bits por segundo.

4. La impedancia de carga del lado del terminador de la interface debe estar entre 3000 y 7000 ohmios y no más de 2500 pf.

DIAGRAMA DE BLOQUES DE LA INTERFACE

Los elementos que conforman la interface se muestran conectados en el diagrama de bloques de la figura Nª 4.2. Un USART es usado para la transferencia de datos entre el programador y el microcomputador, la conversión de lógica TTL a DTL está a cargo de los circuítos 1488 y 1489. El reloj para la transmisión y recepción de datos se lo obtiene dividiendo la señal de reloj presente en el terminal TO del microprocesador 8748,para

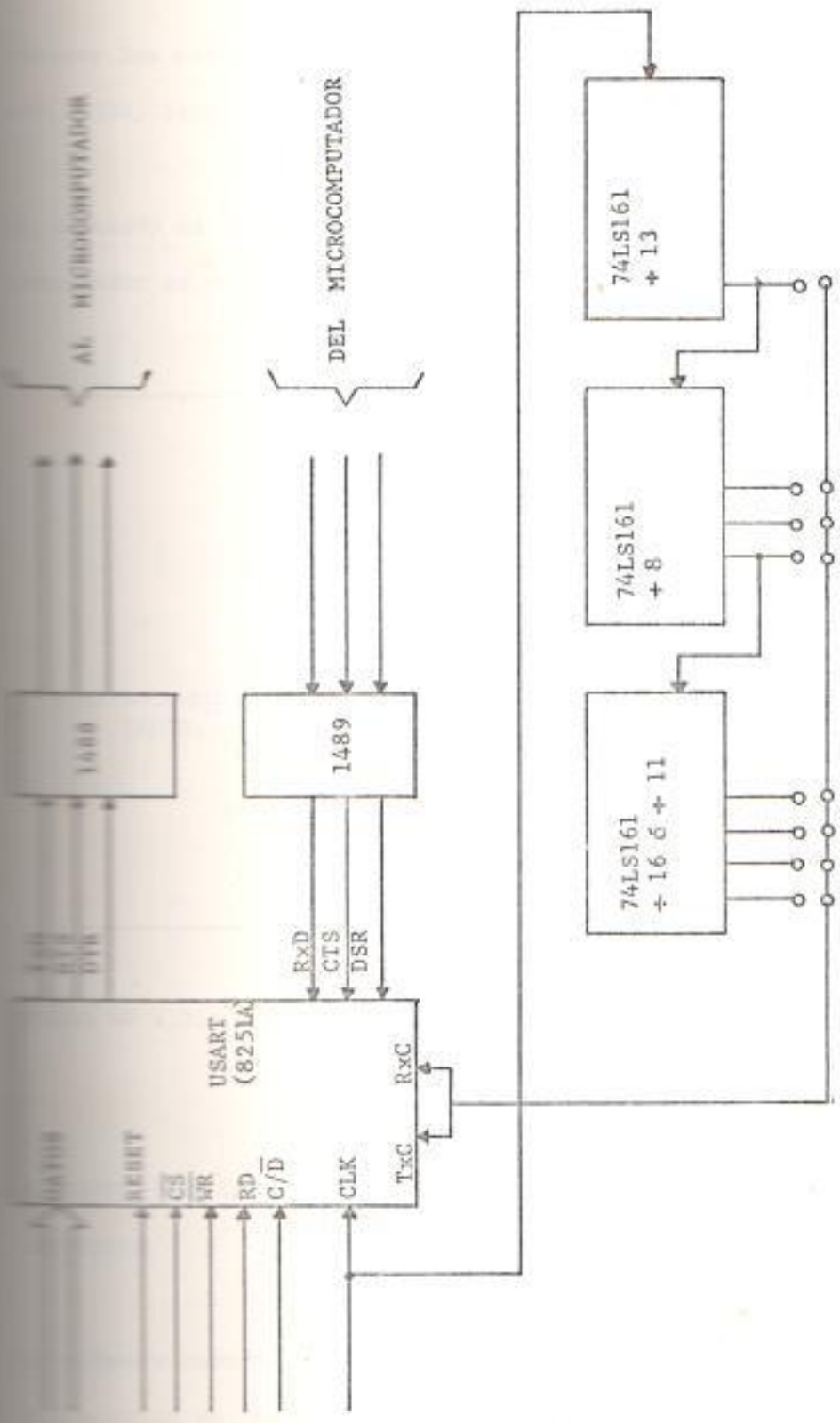


DIAGRAMA DE BLOQUES DE LA INTERFACE.

las siguientes razones de transmisión: 110, 150, 300, 1200, 2400, 4800 y 9600.

El cableado de las señales entre el programador y el microcomputador se muestra en la figura Nº 4.3.

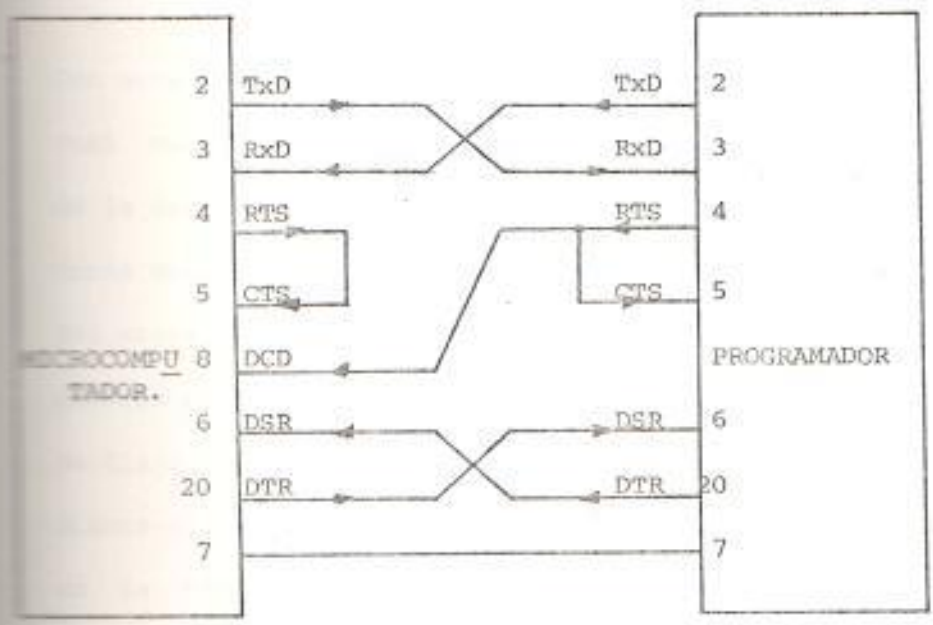


Figura Nº 4.3. CONEXION DE LAS SEÑALES ENTRE EL PROGRAMADOR Y EL COMPUTADOR.

PARA LA COMUNICACION ENTRE EL PROGRAMADOR Y EL MICRO COMPUTADOR

El software cuando el sistema opera con la interface RS-232 descrito en forma general en el Capítulo III. En esta

sección se describirá las rutinas de comando y las rutinas de servicio que el software utiliza para la comunicación - con un microcomputador.

4.4.1. Rutina para mostrar la información almacenada en la memoria de datos externa

Con esta rutina se da servicio al comando DSMEM, el - cual muestra en el monitor del computador toda o parte de la información almacenada en la memoria de datos externa del programador. La DIR y la DFR son obtenidas - del microcomputador y almacenadas en las posiciones de memoria respectivas, tal como lo describe el diagrama - de flujo de la figura Nº 4.4. Si el caracter correspondiente al código hexadecimal de la letra q fue obtenida en la DIR o la DFR, el comando es inmediatamente finalizado. El byte más bajo de la DIR es enviado al microcomputador seguido por el byte más alto, posteriormente se lee de la memoria externa y el dato leído es enviado al microcomputador. Si la DIR no es igual a la DFR, se incrementa la DIR y se continua enviando la información leída de la memoria externa. Cada 16 bytes de datos se envía la - nueva DIR y el lazo se repite hasta que la DIR sea igual a la DFR.

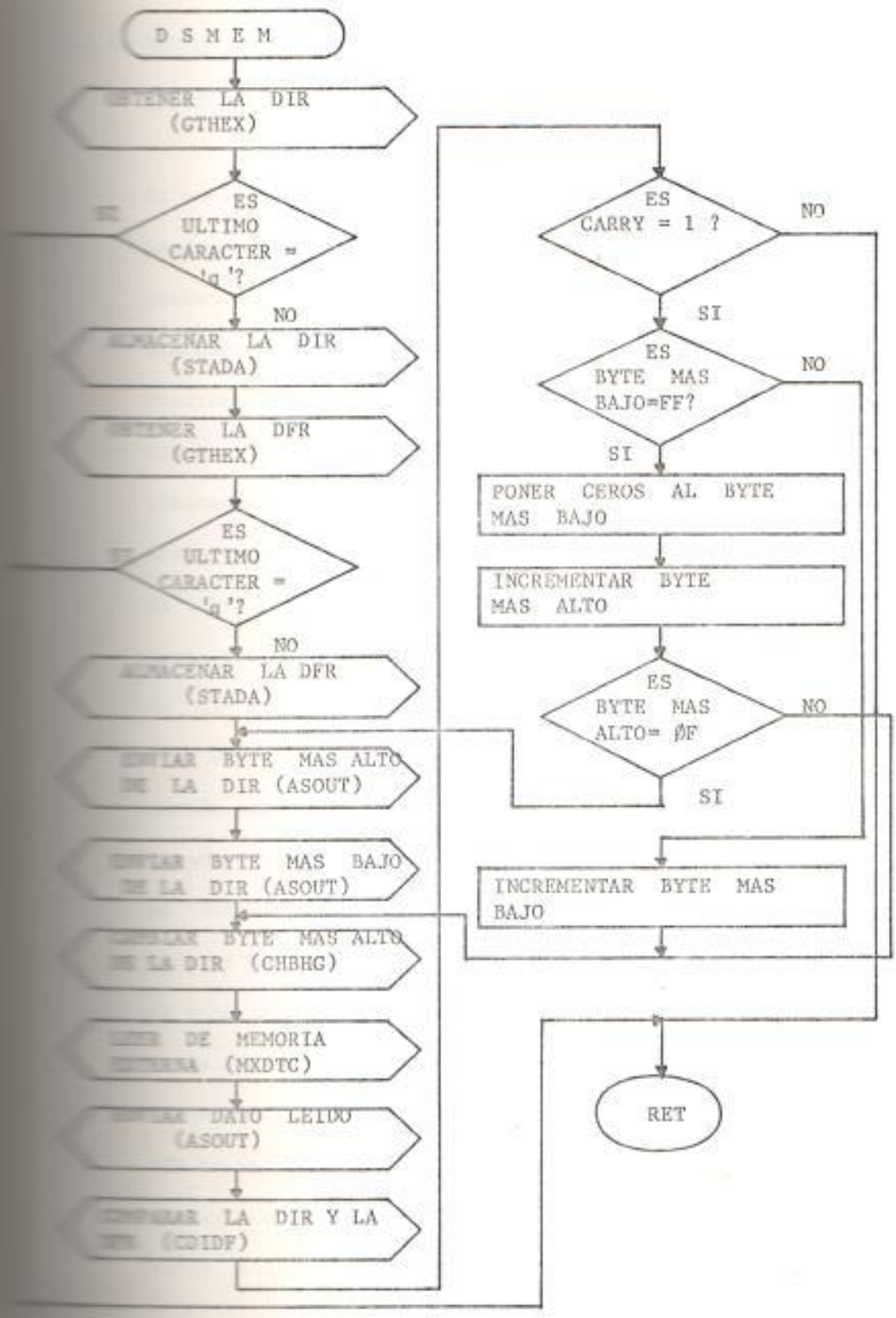


FIG. 4.4. DIAGRAMA DE FLUJO DE LA RUTINA DSMEM

4.4.2. Rutina para cargar la memoria de datos externa mediante el teclado del microcomputador

El comando LOADM es implementado mediante esta rutina. La función de este comando es cargar cada una de las posiciones de la memoria de datos externa del programador mediante el teclado del microcomputador. El diagrama de flujo de esta rutina se muestra en la figura N°- 4.5. El primer paso es obtener la DIR y almacenarla en la posición de memoria repsectiva, siempre que el caracter hexadecimal correspondiente a la letra q no haya sido obtenido. Si este no es el caso entonces el byte más alto de la DIR es enviado al microcomputador seguido por el byte más bajo y un espacio en blanco. Posteriormente se lee la información almacenada en la DIR y se la envía al microcomputador seguida por un guión. La nueva información es obtenida del microcomputador y almacenada en la posición de dato común. Posteriormente esta es almacenada en la memoria de datos externa. Si la información existente en la DIR no va ha ser modificada, un espacio en blanco es necesario para continuar con la próxima dirección. La DIR es incrementada y el lazo se repite de igual forma. Con el código hexadecimal de la letra q se puede salir del lazo y finalizar el comando.

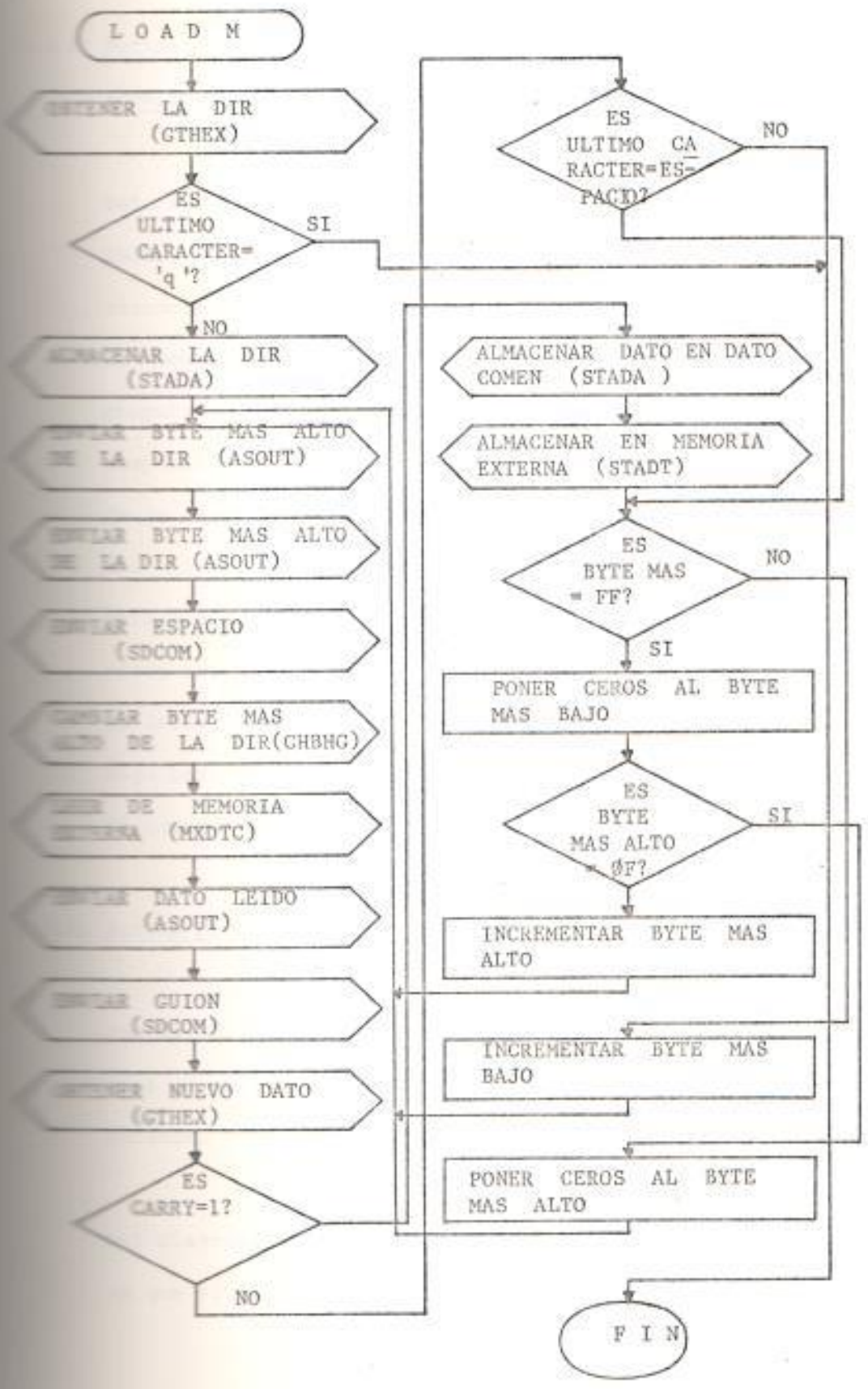


DIAGRAMA DE FLUJO DE LA RUTINA LOADM.

4.4.3. Rutina para cargar la memoria de datos externa desde un archivo

Esta rutina se utiliza para implementar el comando LDARC, el cual lee la información enviada por el microcomputador desde un archivo y la almacena en la memoria externa. La DIR es obtenida del microcomputador y es almacenada en la posición de memoria respectiva tal como lo describe el diagrama de flujo de la figura N°- 4.6. Luego la información es leída del microcomputador y almacenada en la posición de dato común, posteriormente esta es almacenada en la memoria externa. La DIR es incrementada y un nuevo dato es leído, hasta que un caracter correspondiente al código hexadecimal de la letra q sea obtenido para que el comando sea finalizado.

4.4.4. Rutina para grabar información

Con esta rutina se da servicio al comando PGMEP, el cual se utiliza para grabar el Eprom desde un microcomputador.

Del diagrama de flujo de la figura N°- 4.7., se observa que el primer paso es colocar el valor del comando -

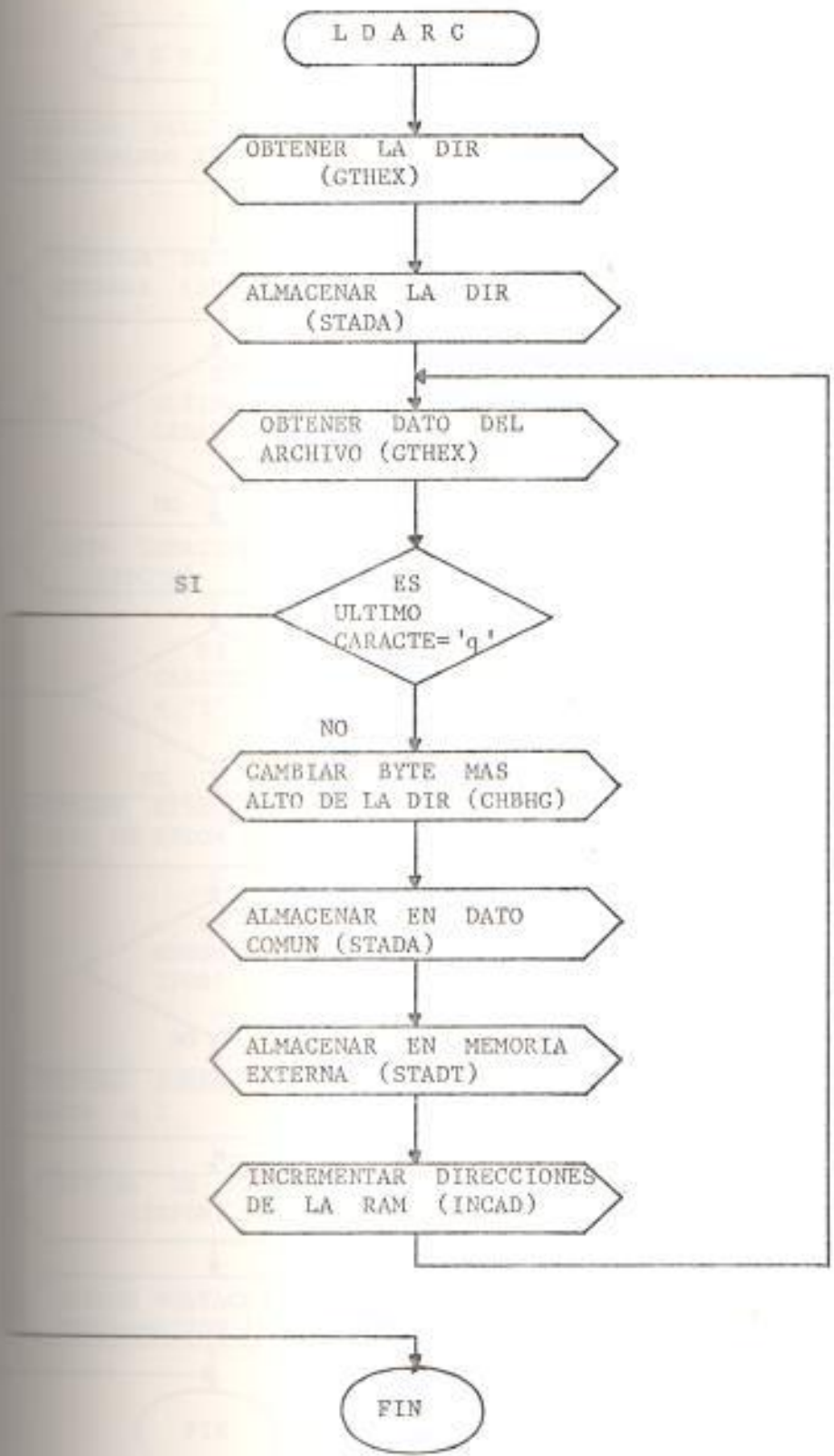
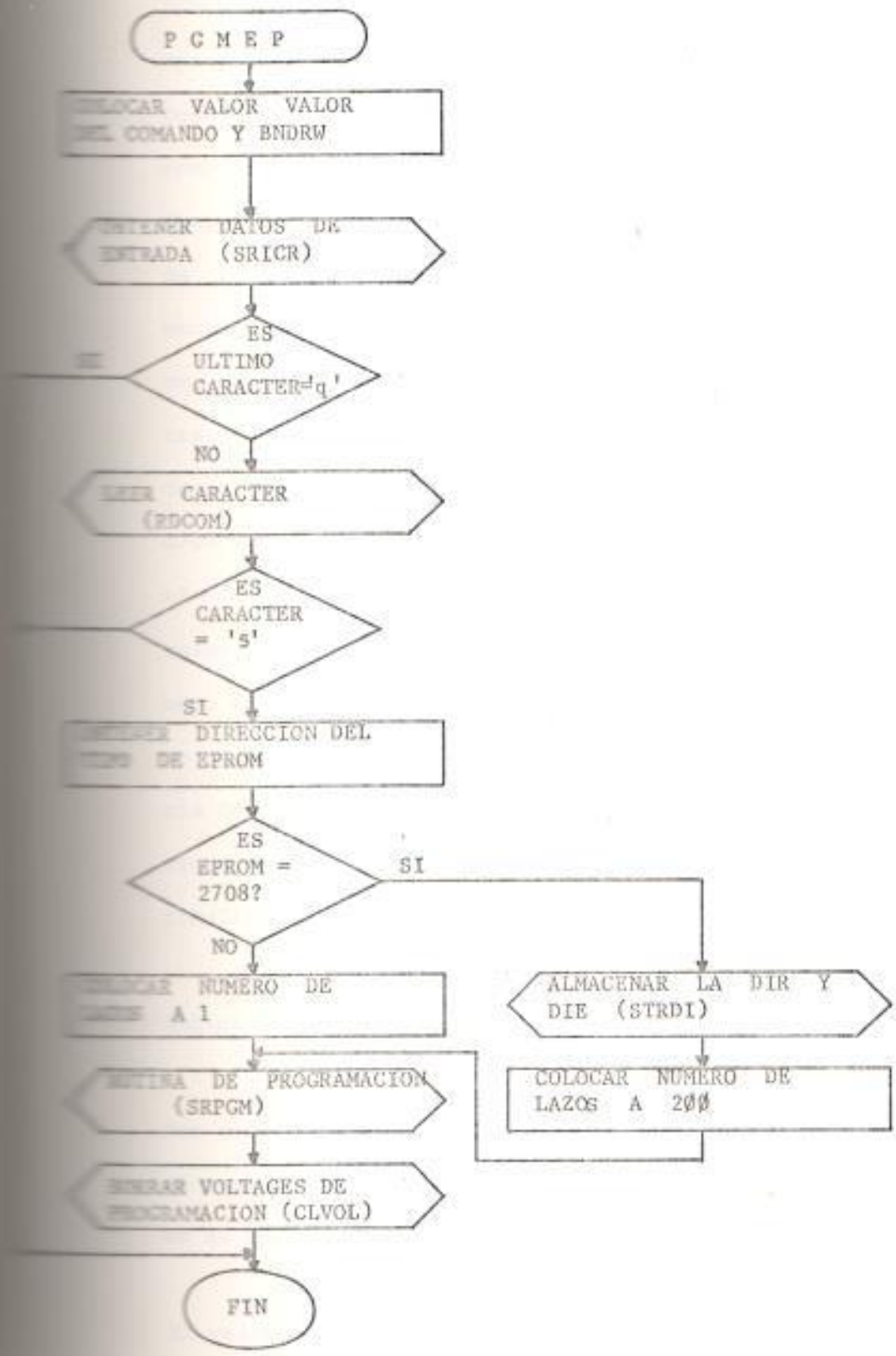


DIAGRAMA DE FLUJO DE LA RUTINA LDARC



4.7. DIAGRAMA DE FLUJO DE LA RUTINA PGMEP

PGMEP y dar a la bandera de lectura/escritura el valor apropiado, posteriormente los datos de entrada correspondiente al comando PGMEP son obtenidos del microcomputador. Si un caracter equivalente al código hexadecimal de la letra q es obtenido en cualesquiera de los datos de entrada, el comando es inmediatamente finalizado. Si este no es el caso entonces un caracter equivalente al código hexadecimal de la letra y es necesario para continuar con el comando, en caso contrario el comando es finalizado. Si el Eprom 2708 va a ser grabado se almacena la DIR y la DIE y el número de lazos de programación es puesto a un valor de 200. Para el resto de los Eproms, el número de lazos es puesto a un valor de uno. El siguiente paso es grabar una o varias direcciones de la memoria Eprom. Una vez hecho esto los voltajes de programación son borrados del Eprom y el comando es finalizado.

Rutina para leer la información almacenada en el Eprom

Por medio de esta rutina se implementa el comando READE, el cual lee toda o parte de la información almacenada en el Eprom y la almacena en la memoria de datos externa.

El valor del comando READE y el valor adecuado de

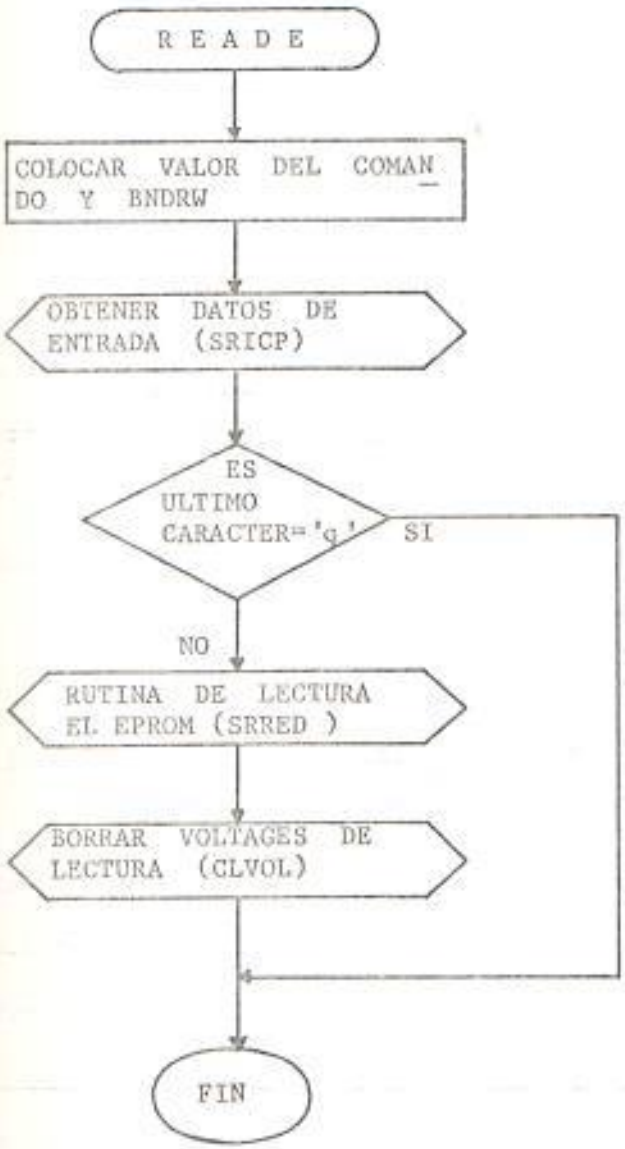


DIAGRAMA DE FLUJO DE LA RUTINA READE.

la bandera de lectura/escritura son colocados primero, tal como lo describe el diagrama de flujo de la figura N°- 4.8. Luego los datos de entrada correspondientes al comando READE son obtenidos del microcomputador. Si un caracter equivalente al código hexadecimal de la letra q es obtenido es cualesquiera de los datos de entrada el comando es inmediatamente finalizado. Si esto no sucede entonces se procede a leer la información grabada en el Eprom y luego se la almacena en la memoria externa, una vez leida toda la información deseada los voltajes de lectura - son borrados del Erpom y el comando es finaliza do.

Rutina para comprobar la programación correcta del EPROM

Con esta rutina se dá servicio al comando CMPPG, el - cual se utiliza para comprobar si la programación del Eprom fue correcta. El diagrama de flujo de esta - rutina se muestra en la figura N°- 4.9. El primer paso es colocar el valor del comando CMPPG y dar el valor adecuado a la bandera de lectura/escritura, posteriormente los datos de entrada correspondientes al comando CMPPG son obtenidos del microcomputador. Si un caracter equivalente al código hexadecimal de la letra q fue obtenido en cualesquiera de los da

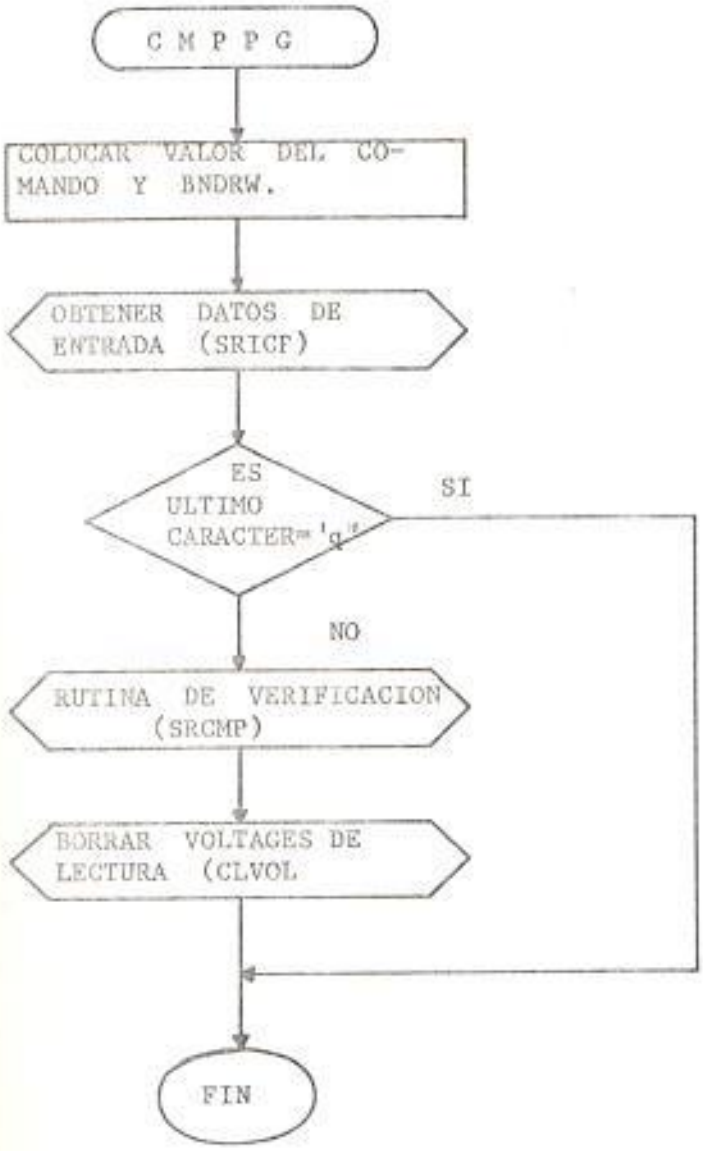
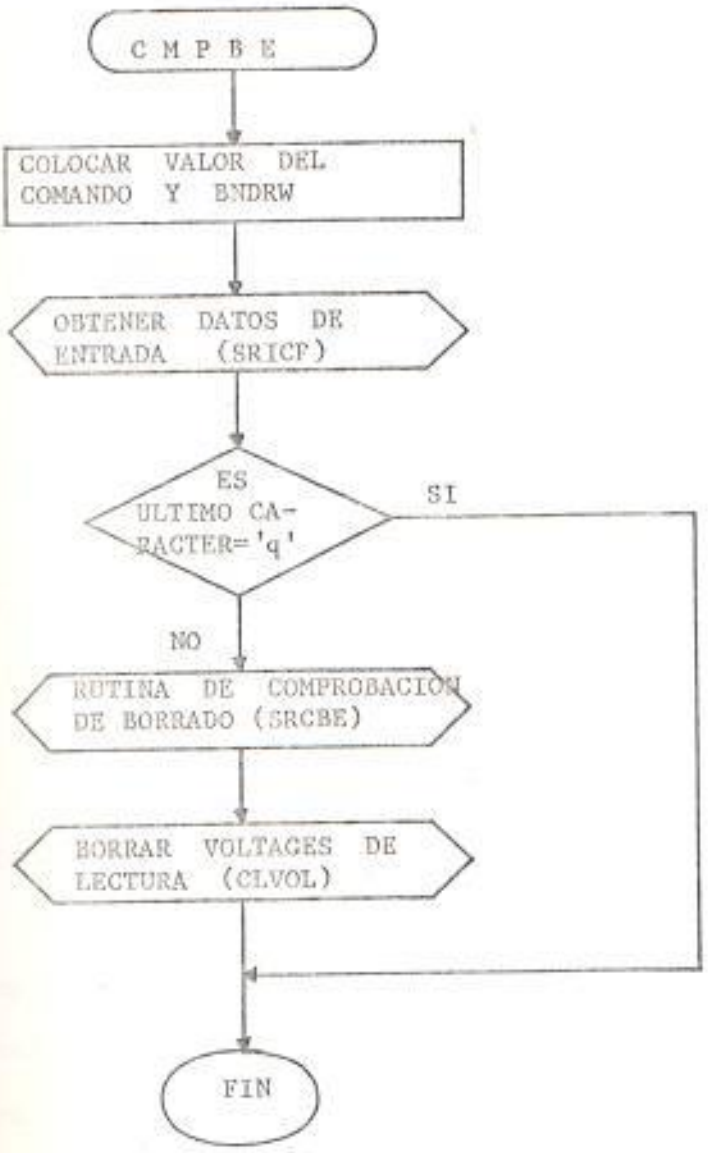


FIG. 4.9. DIAGRAMA DE FLUJO DE LA RUTINA CMPPG

tos de entrada, el comando es inmediatamente finaliza do. Si este no es el caso entonces se procede a comprobar cada una de las direcciones deseadas de la memoria Eprom. Una vez hecho se borra del Eprom los voltajes de lectura y el comando es fina lizado.

4.4.7. Rutina para comprobar el borrado del Eprom

Esta rutina se utiliza para implementar el coman do CMPBE, el cual comprueba si cada una de las direcciones de memoria del Eprom contienen el ca racter hexadecimal FF. La figura N°- 4.10 , muestra el diagrama de flujo de esta rutina. Primero se co loca el valor del comando CMPBE y se da a la bandera de lectura/escritura el valor adecuado. Pos teriormente los datos de entrada correspondientes - al comando CMPBE son obtenidos del microcomputador. Si el caracter hexadecimal de la letra q fue ob tenido en cualesquiera de los datos de entrada, el comando es inmediatamente finalizado. Si este no - es el caso entonces se procede a comprobar una o va rias direcciones de la memoria Eprom. Una vez rea lizado todo esto los voltajes de lectura son bo rrados del Eprom y el comando es finalizado.



4.18. DIAGRAMA DE FLUJO DE LA RUTINA CMPBE

4.4.8. Rutinas de Servicio

Para una mejor comprensión de las rutinas de servicio presentaremos los diagramas de flujo de cada una.

RDCOM - LEER UN CARACTER DEL MICROCOMPUTADOR (FIG.4.11)

DIRECCION -

RDCOM lee un caracter del microcomputador. Primero se coloca el caracter para enviar comandos al USART en la puerta 1 y se activa la señal DTR escribiendo el comando adecuado al USART. Luego esta rutina espera que un caracter sea leído del microcomputador y lo retorna como una salida vía al acumulador.

SDCOM - ENVIAR CARACTERES AL MICROCOMPUTADOR (FIG.4.12)

DIRECCION -

SDCOM envia el caracter ascii de entrada al microcomputador. Primero se averigua si el miccrocomputador está listo para recibir datos, luego se activan las señales RTS y DTR. Si el buffer de transmisión del USART está desocupado, se escribe el caracter al USART para que después este sea enviado al microcomputador. El retardo de tiempo es utilizado para sincronizar la transferencia de información.

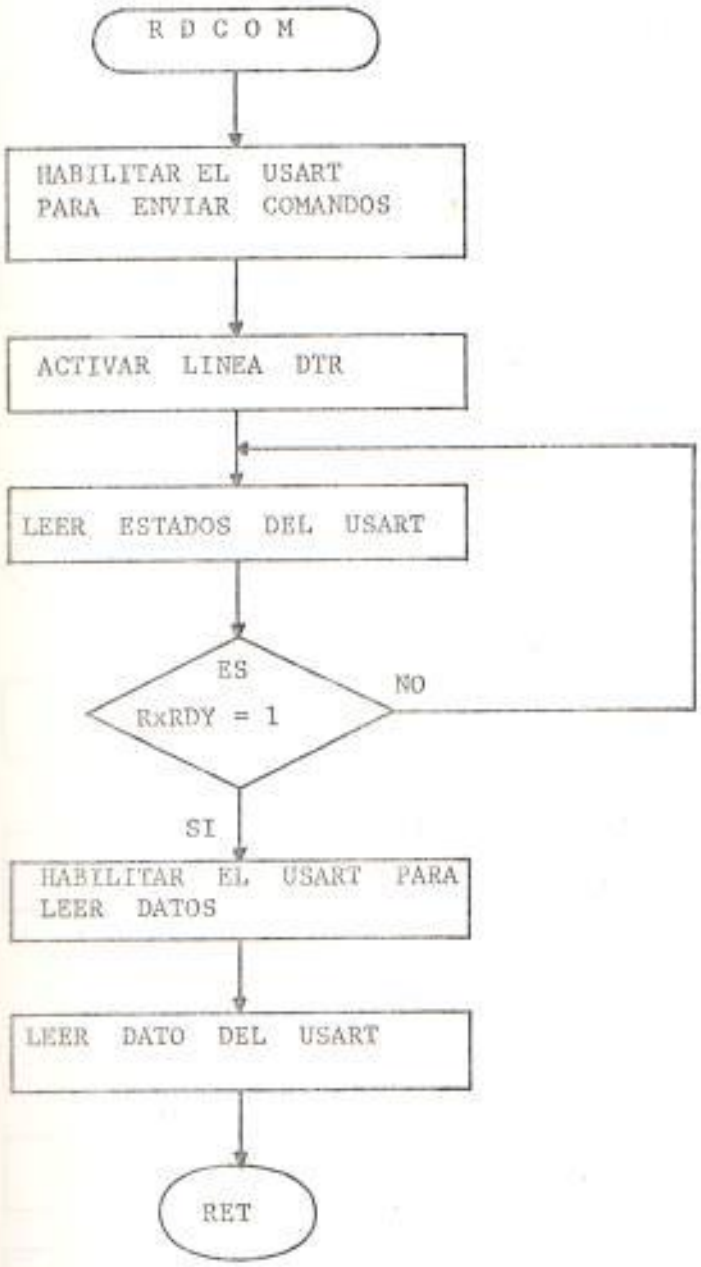


DIAGRAMA DE FLUJO DE LA RUTINA RDCOM

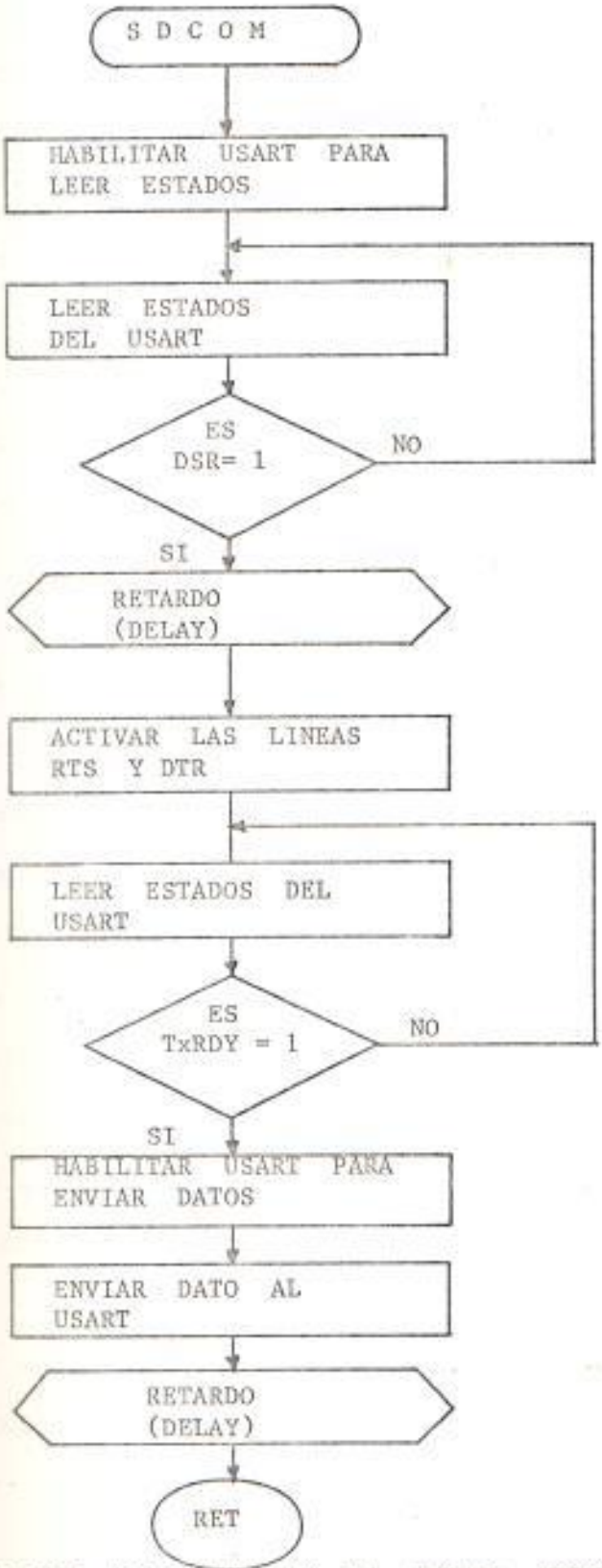


FIG. 4.12. DIAGRAMA DE FLUJO DE LA RUTINA SDCOM

HXASC - CONVERTIR HEXADECIMAL A ASCII (FIG.4.13)

DIRECCION -

HXASC convierte el caracter hexadecimal de entrada a su correspondiente valor en ascii. Para lograr esto primero el caracter hexadecimal es sumado a la dirección de la tabla de caracteres ascii, luego el caracter ascii correspondiente es obtenido de la tabla y almacenado en el registro 2.

ASCHX - CONVERTIR ASCII A HEXADECIMAL (FIG. 4.14)

DIRECCION -

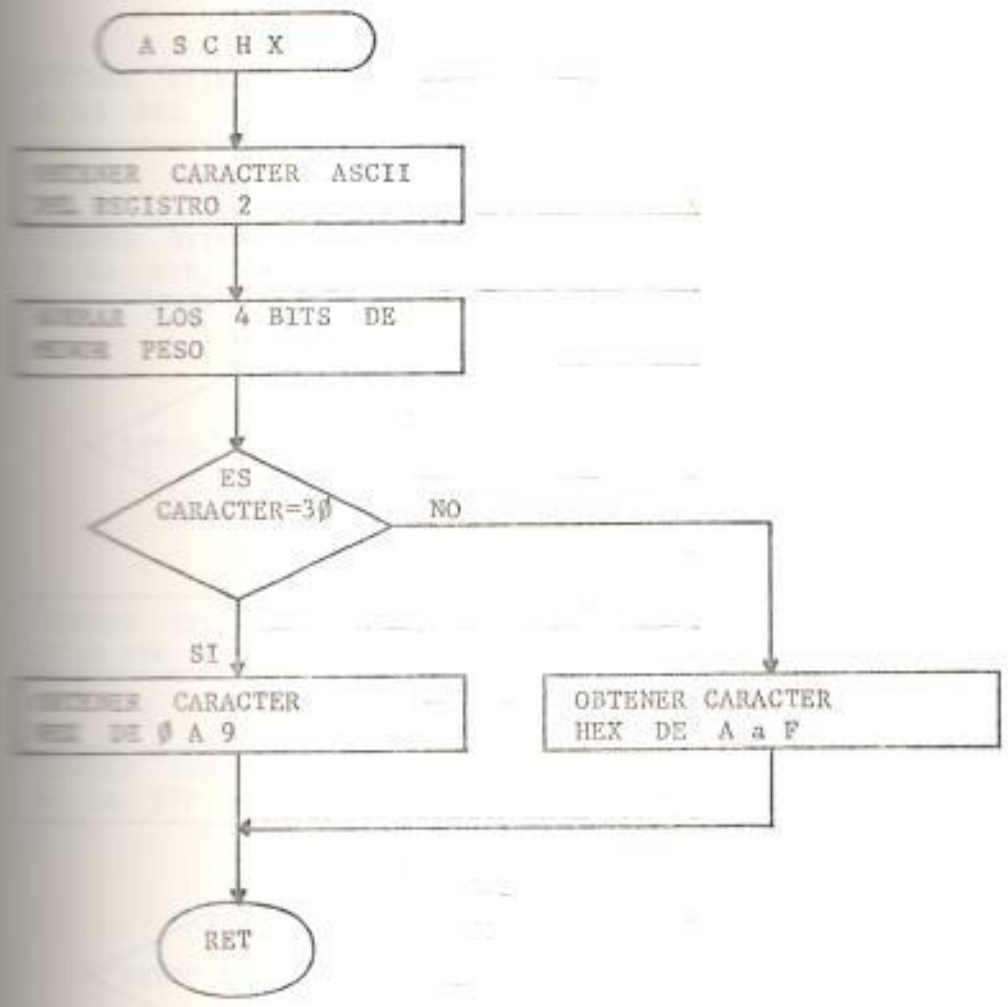
ASCHX convierte el caracter ascii de entrada a su correspondiente valor en hexadecimal. Primero se borran los 4 bits de menor peso y se averigua si el caracter ascii corresponde a los números hexadecimales de 0 a 9, si esto es correcto se filtran los 4 bits de menor peso para obtener el caracter hexadecimal de 0 a 9. En caso contrario tenemos que filtrar los 4 bits de menor peso y sumar 9 para obtener el caracter hexadecimal de A a F.

CDIDF - COMPARAR LA DIR Y LA DFR (FIG. 4.15)

DIRECCION -

CDIDF compara los dos bytes de la DIR con los

FIG. 4.13. DIAGRAMA DE FLUJO DE LA RUTINA HXASC.



4.14. DIAGRAMA DE FLUJO DE LA RUTINA ASCHX.

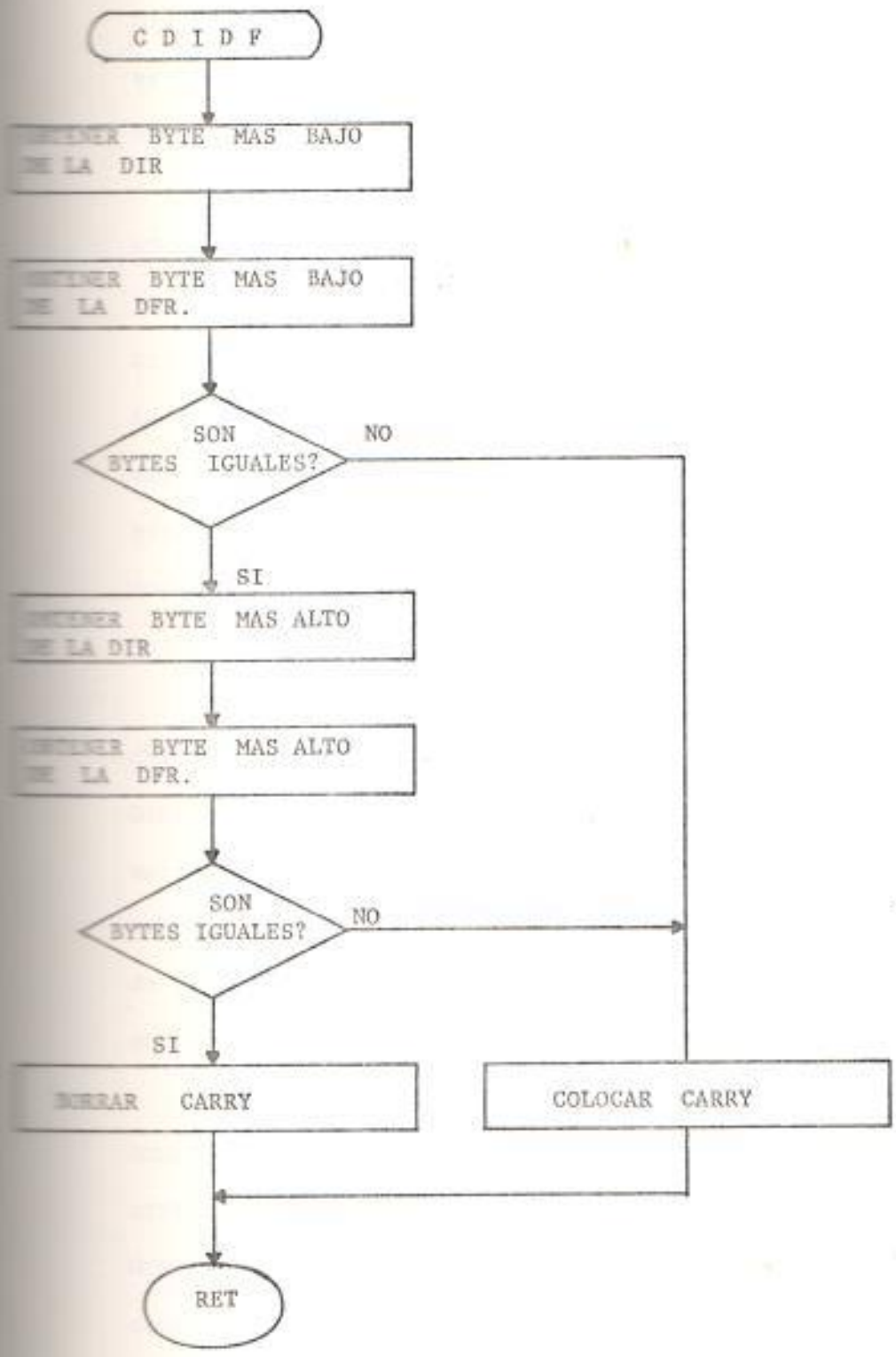


FIG. 4.15. DIAGRAMA DE FLUJO DE LA RUTINA CDIDF.

dos bytes correspondientes de la DFR. Si la DIR no - es igual a la DFR la bandera de acarreo es colocada a uno, en caso contrario esta es colocada a cero.

ASOUT - CONVERTIR A ASCII Y ENVIAR AL MICROCOMPUTADOR (FIG. 4.16)

DIRECCION -

ASOUT convierte el byte de entrada en dos caracteres ascii y los envía al microcomputador. Primero los 4 bits de mayor peso son convertidos al caracter ascii correspondiente, luego este es enviado al microcomputador. Igual cosa sucede con los bits de menor peso.

CHETR - CHEQUEAR CARACTER DE TERMINACION (FIG. 4.17)

DIRECCION - CHETR chequea si el caracter ascii de - entrada corresponde al código hexadecimal de la letra q o es un espacio en blanco. Si es cualquiera de los dos la bandera de acarreo es colocada a uno , en caso contrario esta es puesta a cero.

GTHEX - OBTENER CARACTERES HEXADECIMALES (FIG. 4.18)

DIRECCION -

GTHEX lee caracteres ascii del microcomputador,los convierte a caracteres hexadecimales y los almace_ na en el buffer de salida. Primero la bandera FO,

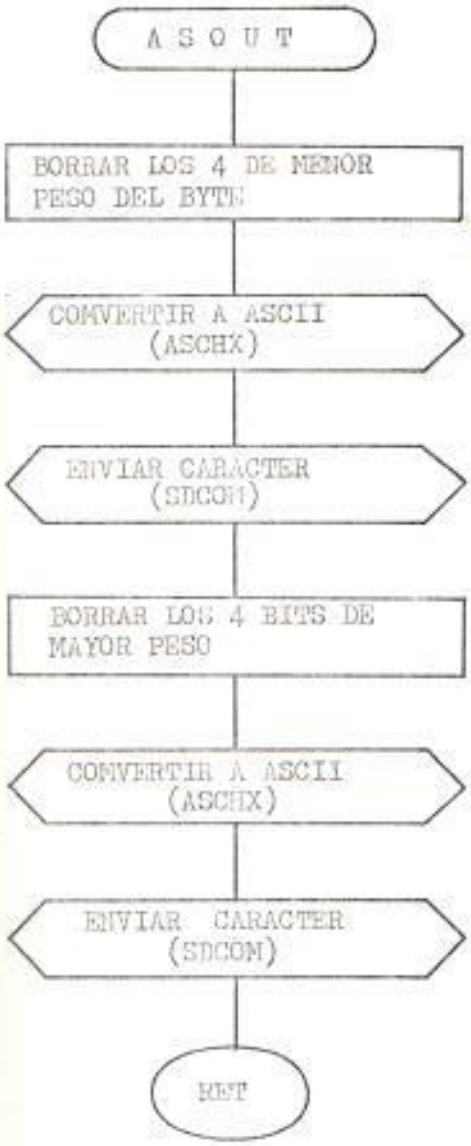


FIG. 3.16 DIAGRAMA DE FLUJO DE LA RUTINA ASOUT

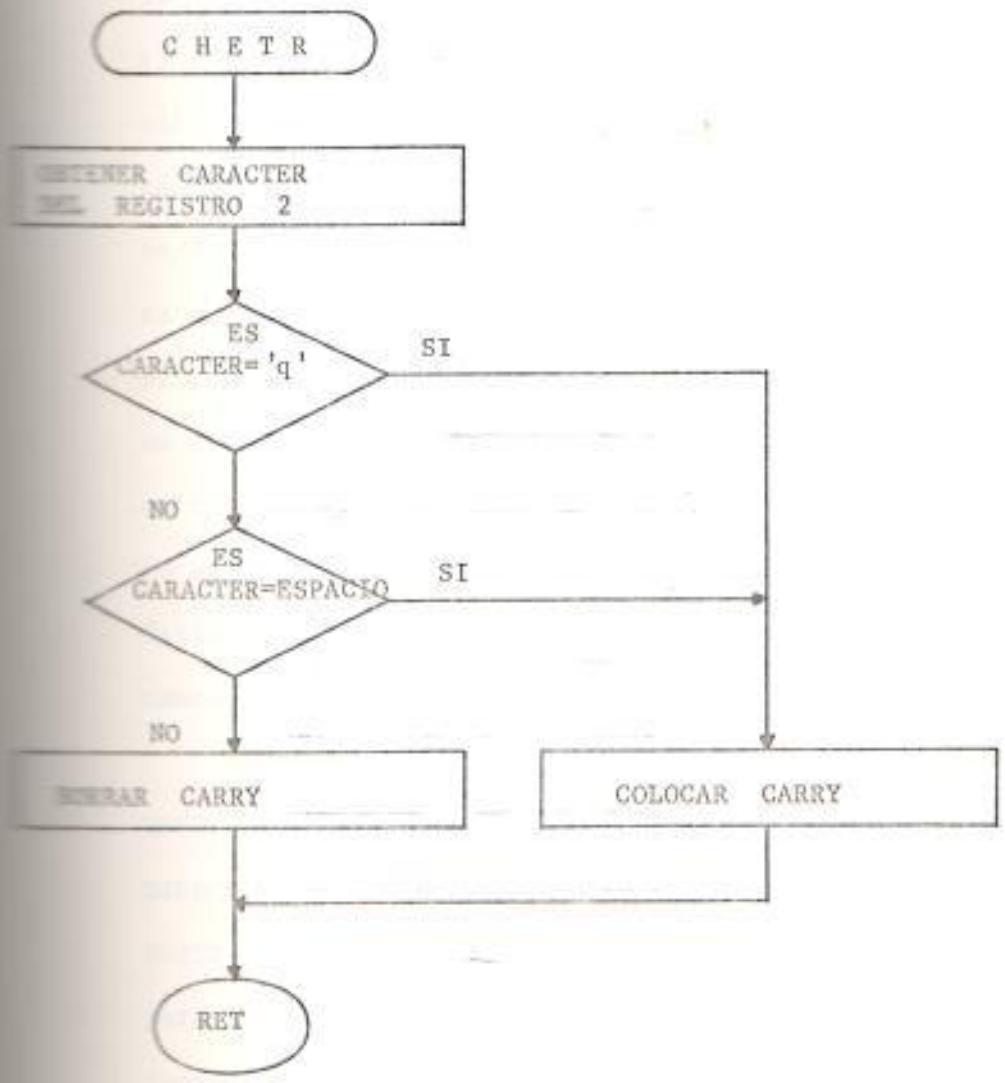


FIG. 4.17. DIAGRAMA DE FLUJO DE LA RUTINA CHETR.

BNDHX y el buffer de salida son borrados. Luego un caracter ascii es leído y chequeado para ver si se trata de un caracter de terminación. Posteriormente este es convertido a caracter hexadecimal y la bandera de digito hexadecimal es puesta a uno. Dependiendo del valor de la bandera de dato/dirección (BNDDD), colocamos el valor de la bandera de campo y almacenamos el caracter hexadecimal en el buffer de salida. Este lazo se repite hasta que se lea un caracter de terminación, luego averiguamos si se trata del caracter hexadecimal de la letra q,si esto es correcto complementamos la bandera restauramos - último caracter, colocamos la bandera de acarreo retornamos.

SRICP - RUTINA INICIAL PARA LOS COMANDOS (FIG. 4.19)

DIRECCION -

SRICP se usa para obtener los datos de entrada para los comandos y reconocer el código del Eprom ingresado. Primero se almacena valor del comando en una posición de memoria determinada, luego se obtienen los datos de entrada del correspondiente comando que está siendo usado. Si el código hexadecimal de la letra que fue obtenido en cualesquiera de los datos de - entrada inmediatamente retornamos de esta rutina .

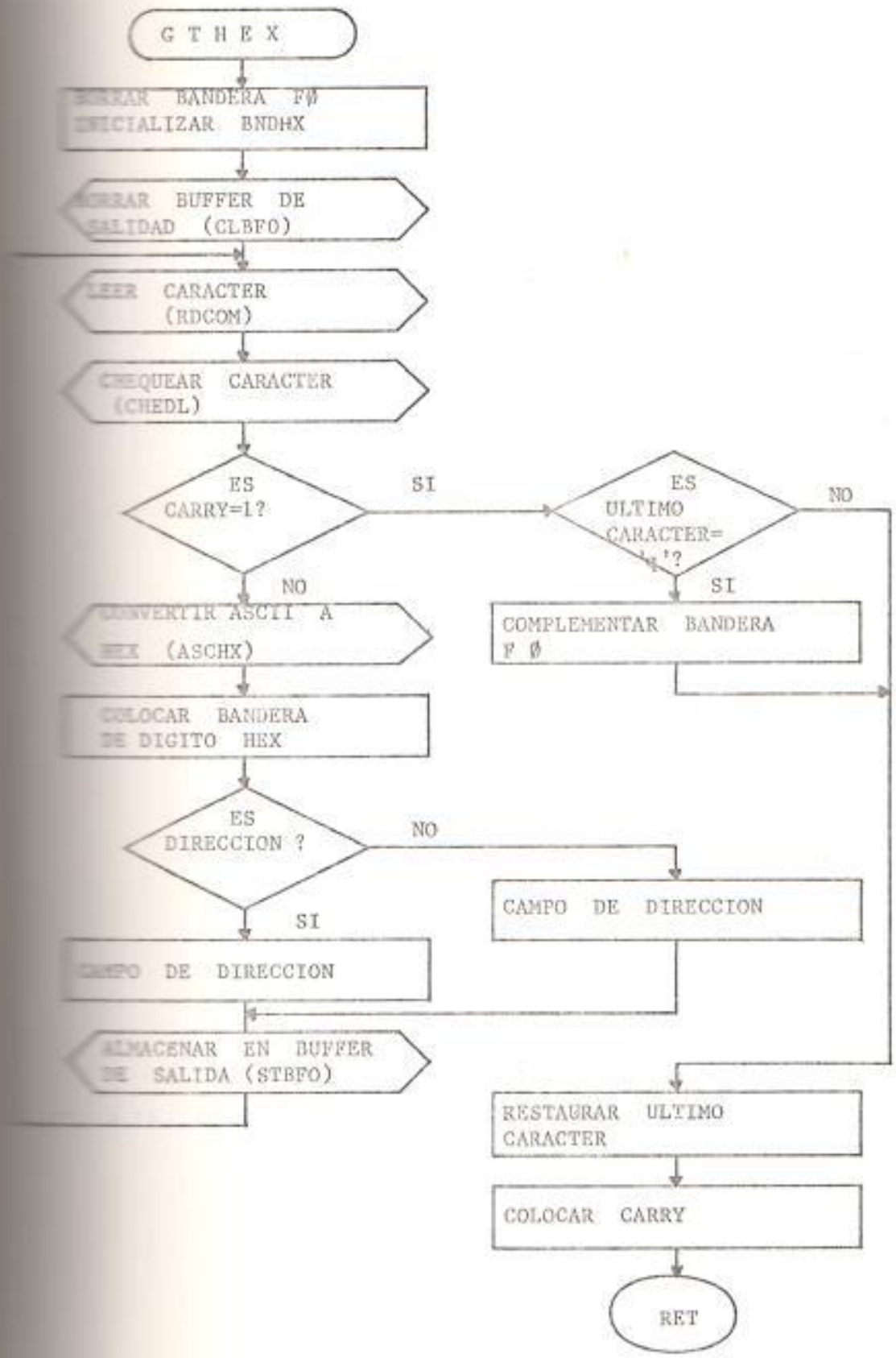
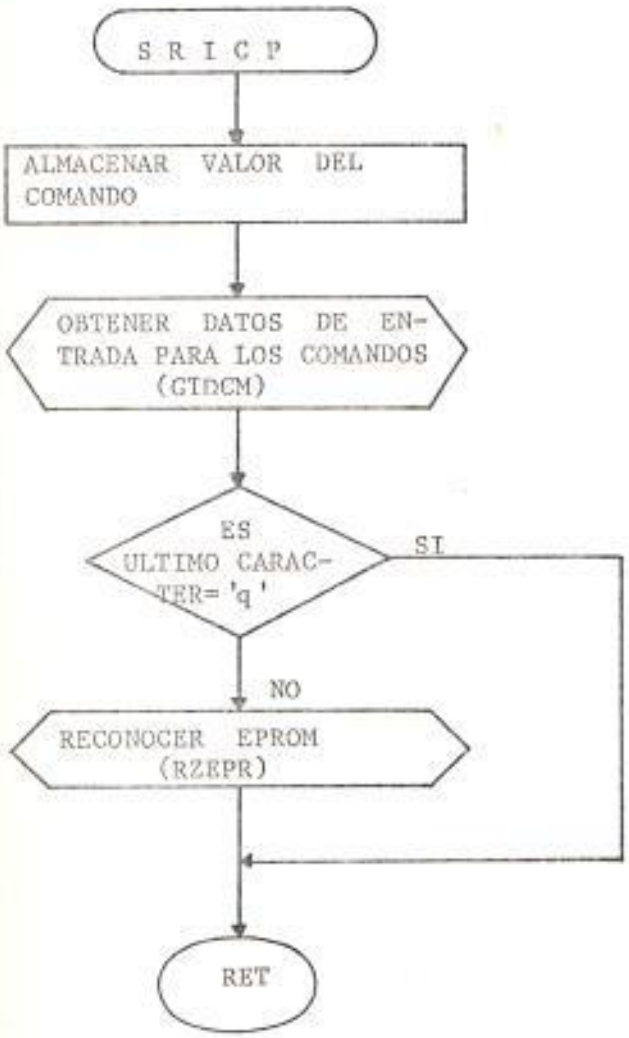


FIG. 4.18. DIAGRAMA DE FLUJO DE LA RUTINA GTHEX.



4.19. DIAGRAMA DE FLUJO DE LA RUTINA SRICP.

Si el caso es el contrario pasamos a reconocer el Eprom y obtener el caracter de control para leer o escribir el Eprom.

GTEPD - OBTENER CODIGO DEL EPROM Y DIRECCIONES (FIG.4.20)

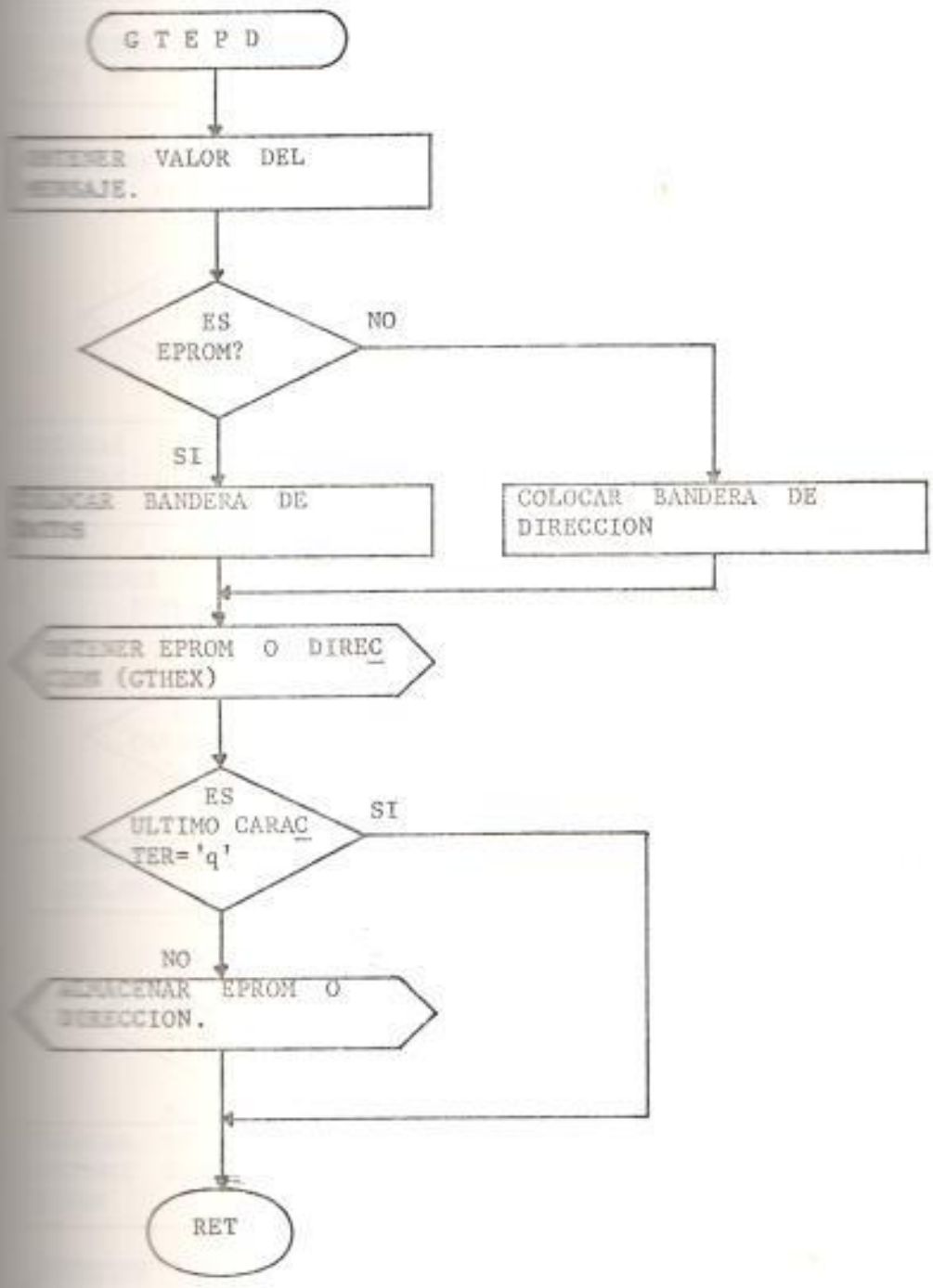
DIRECCION -

GTEPD obtiene el código del Eprom que se va a leer o grabar las direcciones iniciales y finales del Eprom y la memoria de datos externa. Dependiendo del valor del mensaje primero se averigua si se va a obtener un dato (Código del Eprom) o una dirección, y se da el valor apropiado a la bandera de dato/dirección. Luego se obtiene el dato o la dirección. Si el código hexadecimal de la letra q fue obtenido en el dato o la dirección, retornamos de la rutina, Si este no es el caso el dato la dirección es almacenada en la posición de memoria respectiva.

GTDCM - OBTENER LOS DATOS DE ENTRADA PARA LOS COMANDOS - (FIG. 4.21)

DIRECCION -

GTDCM obtiene los datos de entrada específicos para cada comando dependiendo del valor de su argumento de entrada.



4.20. DIAGRAMA DE FLUJO DE LA RUTINA GTEPD

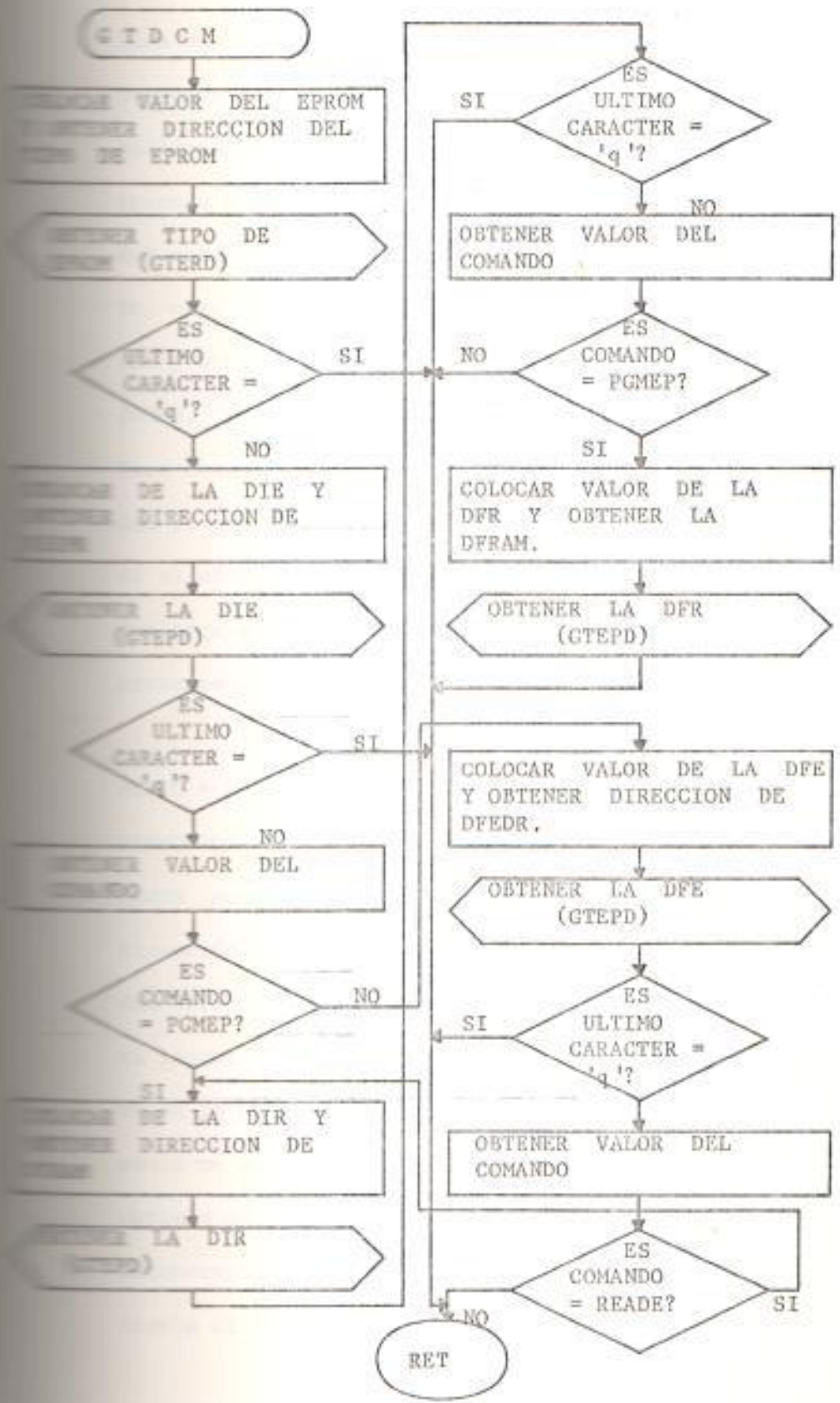


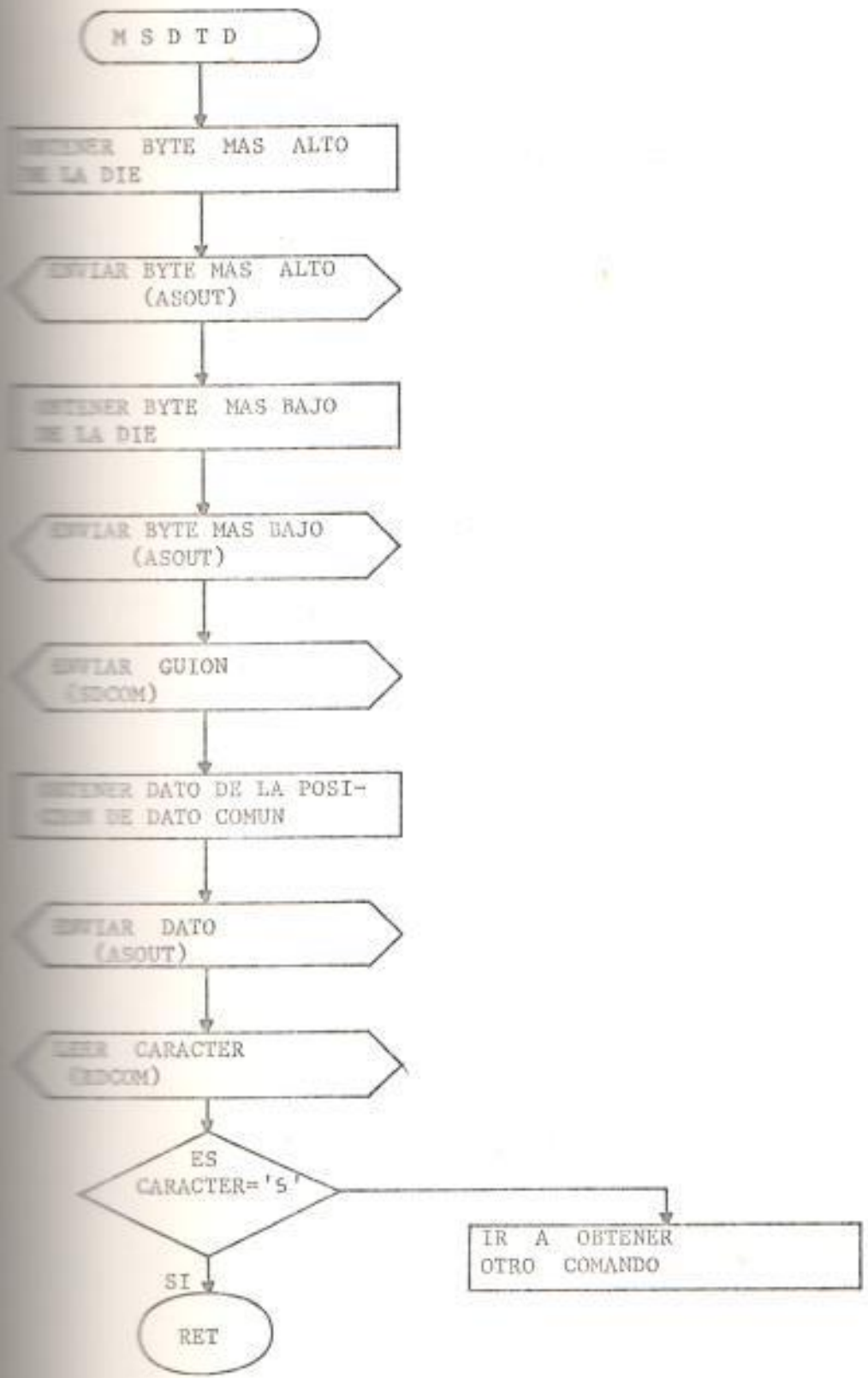
DIAGRAMA DE FLUJO DE LA RUTINA GTDCM.

Para explicar el diagrama de flujo de esta rutina suponga-
mos que se desea obtener los datos de entrada del comando PGMEP. Primero se obtiene el código del Eprom a grabar y la DIE. Luego se obtiene el valor del comando y se averigua si se trata del comando PGMEP, como este es el caso entonces se obtiene la DIR y otra vez se averigua si se trata del comando PGMEP. Por último la DFR es obtenida y retornamos. Para obtener los datos de entrada de los otros comandos el procedimiento es similar al del comando PGMEP. Si el código hexadecimal de la letra q es obtenido en cualesquiera de los datos de entrada, inmediatamente retornamos de la rutina.

MSDTD - MOSTRAR EL DATO Y LA DIRECCION (FIG.4.22)

DIRECCION -

MSDTD muestra en el monitor del microcomputador la dirección y la información almacenada en esa dirección del Eprom que ha sido mal programado o mal borrado. Para lograr esto lo que se hace es enviar el byte más alto de la DIE seguido por el byte más bajo. Después se envía un guión, el dato almacenado en la DIE se lo obtiene de la posición de dato común y se lo envía al microcomputador. Luego se lee un caracter correspondiente al código hexadecimal de la letra y para retor-



4.22. DIAGRAMA DE FLUJO DE LA RUTINA MSDTD.

nar de la rutina. Si el caracter leído es diferente un mensaje de error es enviado al microcomputador.

C A P I T U L O V

DE UTILIZACION DEL PROGRAMADOR

INTRODUCCION

En este capítulo se presentará el manual del usuario del programador. Antes de entrar a operar el programador el usuario debe conocer cuales deben ser las precauciones que se deben tomar para el manejo correcto del programador, las cuales son presentadas en la próxima sección. El resto del Capítulo será dedicado a estudiar el modo de operación de los comandos - cuando el programador opera en forma independiente y cuando opera con la interface RS-232.

PRECAUSIONES QUE SE DEBEN TOMAR

a. Antes de programar o leer un Eprom inserte el módulo de personalidad respectivo y luego inserte el Eprom.

b. Después de programar o leer un Eprom remueva primero el Eprom luego el módulo de personalidad.

c. Asegurese que el módulo de personalidad corresponda al Eprom que va a leer o programar.

d. Asegurese que el Código del Eprom ingresado en - cualquiera de los comandos corresponda al Eprom que va a leer o programar.

SISTEMA INDEPENDIENTE

Cuando la energía ha sido suministrada al programador el visualizador mostrará el siguiente mensaje:

Para poner al programador en modo de operación independiente presionamos la tecla IND, después de esto, un guión será mostrado en el primer led del visualizador.

Una vez seleccionado el modo de operación, el usuario puede entrar cualesquiera de los 6 comandos que existen cuando el sistema opera en forma independiente. A continuación explicaremos cada uno de ellos:

5.3.1. Comando para cargar y leer la memoria de datos externa

Para seleccionar este comando presionamos la tecla LDM, un punto será mostrado al lado derecho del tercer led del campo de dirección.

El punto significa que la dirección inicial de la memoria de datos externa a partir de la cual se comenzará a cargar o leer la memoria puede ser ingresada.

La dirección es terminada por medio de la tecla NEXT, después de presionar esta tecla el visualizador mostrará lo siguiente:

DDD = Dirección inicial

XX = Datos desconocidos

El campo de dirección mostrará la dirección ingresada y el campo de datos la información almacenada en esa dirección. El punto al lado derecho

del segundo led del campo de datos indica que un nuevo dato puede ser ingresado. Si un dato es ingresado, presionamos la tecla NEXT para continuar con la próxima dirección. Para leer la memoria - basta con presionar la tecla NEXT sucesivamente - hasta alcanzar la última dirección deseada. Para salir de este comando presionamos la tecla EXEC. Despues de esto el visualizador mostrará un guión en el primer led del campo de dirección.

Esto indica que un nuevo comando puede ser ingresado.

EJEMPLO 1:

Supongamos que deseamos leer la memoria a partir de la dirección 120.

TECLADO	VISUALIZADOR
LDM	.
120	120.
NEXT	120 XX.
NEXT	121 XX.
NEXT	122 XX.
NEXT	123 XX.
..	. .

EJEMPLO 2:

Supongamos que deseamos cargar la memoria a partir de la dirección 100.

TECLADO	VISUALIZADOR
LDM	.
100	100.
NEXT	100 XX.
34	100 34.
NEXT	101 XX.
40	101 40.
NEXT	102 XX.
AB	102 AB.
NEXT	103 XX.
..	. .
..	. .
NEXT	nnn XX.
CO	nnn CO.
NEXT	nnn+1 XX.
EXEC	-

5.3.2. Comando para programar el Eprom

Para ingresar a este comando presionamos la tecla PGM. El formato de este comando es el siguiente:

(PGM)

(Código del Eprom) (NEXT)

(Dirección inicial del Eprom (DIE)) (NEXT)

(Dirección inicial de la RAM (DIR)) (NEXT)

(Dirección final de la RAM (DFR)) (NEXT)

(EXEC)

Los códigos de los Eproms son los siguientes:

CODIGO	EPROM
08	2708
16	2716
32	2732
3A	2732A
58	2758

(DIE) Es la dirección a partir de la cual se comienza a grabar el Eprom.

(DIR) Es la dirección a partir de la cual comienza la información que será grabada en el Eprom.

(DFR) Es la dirección que contiene el último byte a grabarse.

Este comando programa todo o parte de la memoria Eprom.

EJEMPLO

Supongamos que se desea programar el Eprom 2732A desde la dirección 130. La información a grabarse esta contenida en la memoria RAM a partir de la dirección 000 hasta la dirección 1A0.

Secuencia de pasos que deben seguirse según el formato:

TECLADO	VISUALIZADOR
PGM	EPr
3A	EPr 3A
NEXT	dIE
130	130
NEXT	dir
000	000
NEXT	dPr
1AO	1AO
NEXT	1AO
EXEC	

Después de esto el visualizador será blanqueado - para indicar que el Eprom está siendo programado. Una vez programadas todas las direcciones deseadas el visualizador mostrará un guión en el primer led -

del campo de dirección.

5.3.3. Comando para leer la información almacenada en el Eprom

Para ingresar a este comando presionamos la tecla RD.

El formato de este comando es el siguiente:

(RD)
(Código del Eprom) (NEXT)
(Dirección inicial del Eprom (DIE))(NEXT)
(Dirección final del Eprom (DFE))(NEXT)
(Dirección inicial de la RAM (DIR))(NEXT)
(EXEC)

(DIE) es la dirección a partir de la cual se comienza a leer el Eprom
(DFE) es la última dirección del Eprom que será leída.
(DIR) es la dirección a partir de la cual se comenzará a almacenar la información leída del Eprom.

Este comando lee una o varias direcciones de la memoria Eprom.

EJEMPLO:

Supongamos que se desea leer el Eprom 2708 a par

tir de la dirección 000 hasta la dirección 2FF. La información leída se almacenará en la memoria RAM a partir de la dirección 000.

Secuencia de pasos que deben seguirse según el formato:

TECLADO	VISUALIZADOR
RD	EPr
08	EPr 08
NEXT	dIE
000	000
NEXT	dFE
2FF	2FF
NEXT	dIr
000	000
NEXT	000
EXEC	-

Después de haber leído todas las direcciones deseadas del Eprom, se puede utilizar el Comando LDM para visualizar la información leída de la memoria Eprom.

5.3.4. Comando para comprobar la programación del Eprom

Para ingresar a este comando presionamos la tecla CMP. El formato de este comando es el siguiente:

(CMP)
(Código del Eprom) (NEXT)
(Dirección inicial del Eprom (DIE)) (NEXT)
(Dirección final del Eprom (DFE)) (NEXT)
(Dirección inicial de la RAM (DIR)) (NEXT)
(EXEC)

(DIE) es la dirección a partir de la cual se comienza a verificar la programación del Eprom.

(DFE) es la dirección que contiene el primer dato que fue grabado en el Eprom.

Este comando verifica si una o varias direcciones de la memoria Eprom fueron correctamente grabadas. Si una o varias direcciones del Eprom no fueron grabadas correctamente, en el visualizador se mostrará la dirección y la información almacenada en esa dirección , las próximas direcciones pueden ser visualizadas presionando la tecla NEXT.

EJEMPLO 1

El Eprom 2732 ha sido grabado desde la dirección 000 hasta la dirección 7FF, con la información almacenada en la memoria RAM a partir de la dirección 000. Se desea chequear si el Eprom fue correctamente grabado.

La secuencia de pasos que deben seguirse según el formato son:

TECLADO	VISUALIZADOR
CMP	EPr
32	EPr 32
NEXT	dIE
000	000
NEXT	dFE
7FF	7FF
NEXT	dIr
000	000
NEXT	000
EXEC	-

Si todas las direcciones fueron correctamente grabadas un guión será mostrado en el primer led del campo de dirección.

EJEMPLO 2:

Supongamos que las direcciones 178, 230 y 700 del Eprom del Ejemplo 1 fueron mal grabadas.

TECLADO	VISUALIZADOR
CMP	EPr
32	EPr 32
NEXT	dIE
OOO	OOO
NEXT	dFE
7FF	7FF
NEXT	dIr
OOO	OOO
NEXT	OOO
EXEC	178 XX
NEXT	230 XX
NEXT	700 XX
NEXT	-

3.3.5. Comando para comprobar el borrado del Eprom

Para ingresar a este comando presionamos la tecla CB. El formato de este comando es el siguiente:

(CB)

(Código inicial del Eprom (DIE)) (NEXT)

(Dirección inicial del Eprom (DIE)) (NEXT)

(Dirección final del Eprom (DFE)) (NEXT)

(EXEC)

(DIE) es la dirección a partir de la cual se - comienza a verificar el borrado del Eprom

(DFE) es la última dirección a verificarse

Este comando verifica una o varias de las direcciones del Eprom. Si una o varias direcciones - no fueron borradas completamente,en el visualizador se mostrará la dirección y la información almacenada en esa dirección.

5.3.6. Comando para borrar la memoria de datos externa

Para ingresar este comando presionamos la tecla C.

El formato de este comando es el siguiente:

(C)

(Dirección inicial de la RAM (DIR))(NEXT)

(Dirección final de la RAM (DFR)) (NEXT)

(DIR) es la primera dirección a ser borrada

(DFR) es la última dirección a ser borrada

Este comando borra una o varias direcciones de la memoria de datos externa.

SISTEMA CON LA INTERFACE RS-232

Antes de iniciar la comunicación entre el programador y el microcomputador debe chequearse la conexión entre el puerto RS-232 del microcomputador y el puerto del programador. La velocidad de transmisión es seleccionada en el programador mediante los microswitches que están montados sobre la tarjeta, ver diagrama de posicionamiento de los elementos en el Apéndice A. Cada microswitche es está asociado con una determinada velocidad.

0= 110 baudios

1= 150 baudios

2= 300 baudios

3= 600 baudios

4=1200 baudios

5=2400 baudios

6=4800 baudios

7=9600 baudios

Una vez seleccionada la velocidad podemos suministrar energía al programador. Para poner al programador a operar con la interface RS-232 presionamos la tecla - EPS, el mensaje mostrado en el visualizador es el siguiente:

Esto indica que el programador está listo para la comunicación con el microcomputador.

Antes de comenzar a correr el programa que establece la comunicación, debe inicializarse el puerto RS-232 del microcomputador mediante el comando MODE del sistema operativo. El - formato de este comando es el siguiente:

MODE COM1: Velocidad, paridad, bits de datos, bits Stop

El único parámetro que puede ser cambiado es la velocidad , los demás tienen en este caso su valor fijo determinado por el programador. De esta forma el comando MODE se resume - al siguiente formato:

MODE COM1: Velocidad, N,7,2

Después de haber inicializado el puerto de comunicación del

microcomputador , estamos listos para correr el programa.

El programa contiene los siguientes comandos:

M - mostrar memoria

C - cargar memoria

P - programar Eprom

R - Leer Eprom

V - Verificar programación

B - comprobar borrado del Eprom

G - grabar el Eprom desde un archivo

A - almacenar la información grabada en el Eprom en un archivo.

F - leer archivo

I - listar los códigos de los Eproms

L - lustar comandos

Presione la tecla ENTER para salir de un comando

5.4.1. Comando para mostrar la información almacenada en la memoria de datos externa

El formato de este comando es el siguiente:

(M) (ENTER)

(Dirección inicial de la RAM(DIR)) (ENTER)

(Dirección final de la RAM (DFR)) (ENTER)

X = cualquier número de 0 a F

DIR = XX0→último número debe ser siempre cero

Este comando muestra en el monitor del computador, toda o parte de la información almacenada en la memoria de - datos externa.

5.4.2. Comando para cargar la memoria de datos externa

El formato de este comando es el siguiente:

(C) (ENTER)

(Dirección inicial de la RAM (DIR)) (ENTER)

Mediante este comando se puede cargar una o varias direcciones de la memoria de datos externa.

5.4.3. Comando para programar la memoria Eprom

El formato de este comando es el siguiente:

(P) (ENTER)

(Código del Eprom) (ENTER)

(Dirección inicial del Eprom (DIE)) (ENTER)

(Dirección inicial de la RAM(DIR)) (ENTER)

(Dirección final de la RAM (DFR)) (ENTER)

Este comando graba en el Eprom la información - almacenada en la memoria de datos externa. Se pue de grabar una o varias direcciones.

5.4.4. Comando para leer la memoria Eprom

El formato de este comando es el siguiente:

(R) (ENTER)
(Código del Eprom) (ENTER)
(Dirección inicial del Eprom (DIE)) (ENTER)
(Dirección final del Eprom (DFE)) (ENTER)
(Dirección inicial de la RAM (DIR)) (ENTER)

Este comando lee la información grabada en el Eprom y la almacena en la memoria de datos externa.

5.4.5. Comando para verificar programación

El formato de este comando es el siguiente:

(V) (ENTER)
(Código del Eprom) (ENTER)
(Dirección inicial del Eprom (DIE)) (ENTER)
(Dirección final del Eprom (DFE)) (ENTER)
(Dirección inicial de la RAM (DIR)) (ENTER)

Este comando verifica si la información grabada en el Eprom fue correcta. En caso de que una o varias direcciones no fueron correctamente grabadas, la dirección y la información almacenada en esa dirección serán mostradas en el monitor del microcomputador.

5.4.6. Comando para comprobar el borrado del Eprom

El formato de este comando es el siguiente:

(B) (ENTER)
(Código del Eprom) (ENTER)
(Dirección inicial del Eprom (DIE)) (ENTER)
(Dirección final del Eprom (DFE)) (ENTER)

Este comando comprueba si el Eprom está completamente borrado. Las direcciones que no fueron borradas completamente son mostradas en el monitor del computador junto con la información almacenada en esas direcciones.

5.4.7. Comando para grabar el Eprom desde un archivo

El formato de este comando es el siguiente:

(G) (ENTER)

(Nombre del archivo) (ENTER)

(Entre la unidad de disco donde se encuentra el archivo)

(ENTER)

(Código del Eprom) (ENTER)

(Dirección inicial del Eprom (DIE)) (ENTER)

(Dirección final del Eprom (DFE)) (ENTER)

Este comando graba el Eprom con la información almacenada en un archivo. Primero la información es cargada a la memoria RAM del programador y luego es grabada en el Eprom.

5.4.8. Comando para almacenar la información grabada en el Eprom en un archivo

El formato de este comando es el siguiente:

(A) (ENTER)

(Código del Eprom) (ENTER)

(Dirección inicial del Eprom (DIE)) (ENTER)

(Dirección final del Eprom (DFE)) (ENTER)

(NOmbre del archivo) (ENTER)

(Entre la unidad de disco donde se va a almacenar el archivo)

(ENTER).

Este comando crea un archivo con la información leída - desde la memoria Eprom. Primero la información leída - del Eprom es almacenada en la memoria RAM del programador y luego es transferida al archivo.

5.4.9. Comando para leer un archivo

El formato de este comando es el siguiente:

(F) (ENTER)
(Nombre del archivo)(ENTER)
(Entre la unidad de disco donde se encuentra el archivo)
(ENTER)

Este comando muestra en el monitor del microcomputador - la información almacenada en el archivo deseado. Se puede leer todo el archivo o por bloques. En el segundo caso se entra la dirección inicial y la dirección final . También se puede editar cualquier dirección del archivo y volver a almacenar el nuevo archivo si se desea. En el caso de que se desee leer el archivo por bloques tenemos la siguiente observación para entrar el valor de la dirección inicial.

Dirección inicial = XX0
- → debe ser siempre cero
- → cualquier valor de 0 a F.

5.4.10. Comando para mostrar los códigos de los Eproms

El formato de este comando es el siguiente:

(E) (ENTER)

Este comando se usa para mostrar los códigos de los - Eproms en el monitor del microcomputador.

5.4.11. Comando para mostrar la lista de comandos

El formato de este comando es el siguiente:

(L) (ENTER)

Este comando muestra la lista de todos los comandos utilizados por este programa.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. El diseño y construcción de cualquier sistema presenta muchos problemas de tipo práctico que muy pocas veces se presentan en la teoría.

2. Después de hacer varias pruebas de programación con los Eproms 2708, 2716 y 2732, se comprobó que el sistema cumple con los objetivos para el cual fue construído.

3. La comunicación del sistema con un microcomputador fue probada con una computadora personal IBM. El resultado de esto fue satisfactorio cumpliendo con los dos objetivos principales por los cuales fue establecida la comunicación. Primero almacenar la información grabada en el Eprom en un archivo y segundo grabar el Eprom desde un archivo.

RECOMENDACIONES

1. Equipar los laboratorios con el equipo necesario para incentivar al estudiante a realizar trabajos que consistan en di

ñar y construir cualquier circuito.

2. Presentar facilidades al estudiante para conseguir los elemen tos necesarios para la construcción de los trabajos de Tesis.

3. Se puede diseñar un programador de memorias de mayor capa cidad de programación basándose en el trabajo presentado en esta Tesis.

4. El programa de comunicación entre el programador y un microcom putador fue diseñado para trabajar con la computadora personal IBM. Se puede diseñar un programa similar para conectar - el programador con otra microcomputadora.

BIBLIOGRAFIA

1. JOSE ANGULO USATEGUI, "Microprocesadores curso sobre aplicaciones en sistemas industriales", Paraninfo, 1.982.
2. PETER GROGONO, "Programación Pascal", Fondo Educativo Interamericano, Edición revisada, 1.984.
3. JOHN E. McNAMARA, " Technical aspects of data comunication",1977.
4. APLICATION TECHNIQUES FOR THE MCS-48 Family, Intel.
5. MCS-48 User's Manual, Intel.
6. MCS-48 and UPI-41 Assembly Language Manual, Intel.
7. MCS-80/85 Family, User's Manual, Octubre 1979, Intel.
8. SDK-85 System Design kit User's Manual, Intel 1977.
9. COMPONENT DATA CATALOG, Intel, 1982.
10. PROGRAMADOR DE EPROMS , Byte Publications Inc., Abril, 1980.
11. EIA STANDARD RS-232C, 1982 Datapro Research Corporation Delran;NJ 08075 USA, mayo de 1982.
12. THE TTL DATA BOOK, Texas Instruments, segunda edición, 1981.