



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
"ARAGÓN"

DISEÑO, CONSTRUCCIÓN Y PRUEBAS DE UN  
PROTOTIPO LECTOR DE EPROMS CON  
SALIDA POR PUERTO RS - 232

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A :

LUIS MANUEL ELIAS HIGAREDA

TESIS CON  
FALLA DE ORIGEN

San Juan de Aragón Edo. de Méx.

1988



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

## CONTENIDO

INTRODUCCION . . . . .	i
------------------------	---

### CAPITULO I

#### METODOLOGIA DE DISEÑO

1.1	Introducción . . . . .	2
1.2	Metodología de diseño . . . . .	2
1.3	Definición del sistema . . . . .	3
1.3.1	Análisis del sistema . . . . .	3
1.3.2	Localización de funciones . . . . .	5
1.3.3	Especificación del equipo . . . . .	6
1.4	Planeación del hardware/software . . . . .	7
1.5	Análisis de los requisitos del Hardware/Software. . . . .	8
1.6	Implementación del Hardware . . . . .	9
1.7	Diseño del Software . . . . .	10
1.8	Integración de Hardware/Software . . . . .	11
1.9	Pruebas de Operación y Funcionamiento . . . . .	12

## CAPITULO II

### DISEÑO DETALLADO

2.1	Introducción . . . . .	14
2.2	Descripción del diagrama electrónico . . . . .	15
2.2.1	Unidad de control . . . . .	16
2.2.2	Unidad de lectura . . . . .	21
2.2.3	Unidad de comunicación . . . . .	24
2.2.3.1	Protocolo de comunicación . . . . .	28
2.2.3.2	Acondicionamiento de señales . . . . .	30
2.2.4	Fuente de alimentación . . . . .	31
2.2.5	Diagrama Eléctrico del Lector de Memorias. . . . .	34
2.3	Lista de partes electrónicas . . . . .	36
2.3.1	Lista de proveedores . . . . .	37
2.3.2	Lista de catálogos . . . . .	38
2.4	Diseño mecánico . . . . .	49
2.4.1	Diagramas de partes mecánicas . . . . .	50
2.4.2	Lista de partes mecánicas . . . . .	58
2.4.2.1	Lista de proveedores . . . . .	59
2.4.2.2	Lista de catálogos . . . . .	60

### CAPITULO III

#### SOFTWARE

#### PROGRAMACION DEL MICROPROCESADOR

3.1	Introducción . . . . .	64
3.2	Descripción del programa . . . . .	65
3.2.1	Comandos seleccionados . . . . .	70
3.3	Diagramas de flujo y codificación . . . . .	71

### CAPITULO IV

#### MANUAL DE REPRODUCCION

#### Y MANTENIMIENTO

4.1	Introducción . . . . .	113
4.2	Diagrama de ensamble . . . . .	114
4.3	Pruebas iniciales . . . . .	116
4.4	Mantenimiento . . . . .	126

### CAPITULO V

#### MANUAL DEL USUARIO

5.1	Introducción . . . . .	128
5.2	Instalación . . . . .	129
5.3	Operación . . . . .	131
5.3.1	Comandos utilizados. . . . .	132

5.3.2	Procesamiento de la información . . . . .	134
5.4	Mantenimiento . . . . .	139
5.5	Especificaciones . . . . .	140
CONCLUSIONES . . . . .		142
BIBLIOGRAFIA . . . . .		145

## ANEXO I

### TEORIA DEL MICROPROCESADOR

1.1	Computadoras Digitales . . . . .	150
1.2	Estructura de una computadora digital . . . . .	152
1.2.1	Unidad de Procesamiento Central CPU . . . . .	153
1.2.2	Unidad de Memoria . . . . .	156
1.2.3	Unidad de Entrada-Salida . . . . .	158
1.3	El Microprocesador . . . . .	160
1.3.1	Clases de microprocesadores . . . . .	162
1.4	Estructura del microprocesador. . . . .	163
1.4.1	Unidad Aritmética y Lógica . . . . .	166
1.4.2	Unidad de Control . . . . .	168
1.4.3	Registros . . . . .	168
1.5	Funcionamiento de la CPU . . . . .	174
1.5.1	Reloj de una CPU . . . . .	174
1.5.2	Ciclos de instrucción . . . . .	175
1.6	Implementación de las Instrucciones . . . . .	180
1.6.1	Instrucciones de un ciclo . . . . .	180

1.6.2	Instrucciones con referencia a memoria . . . . .	180
1.6.3	Instrucciones sin referencia a memoria . . . . .	180
1.6.4	Instrucciones de salto . . . . .	181
1.7	Elementos de Diseño . . . . .	182
1.8	Selección del Microprocesador . . . . .	182
1.9	Dispositivos de Memoria . . . . .	184
1.9.1	Tipos de Memorias . . . . .	187
1.9.2	Memorias utilizadas con microprocesadores . . .	192
1.9.3	Memorias RAM estáticas y dinámicas . . . . .	197
1.10	Interfaces de Entrada/Salida . . . . .	200
1.10.1	Operaciones de una interfaz . . . . .	201
1.10.2	Interfaz en un integrado. . . . .	204
1.10.3	Tipos de transferencias de E/S . . . . .	205
1.10.4	Transferencia de E/S en paralelo. . . . .	212
1.10.5	Transferencia de E/S en serie . . . . .	212
1.10.6	Protocolo de Entrada/Salida serie . . . . .	218
1.10.7	Requisitos de un dispositivo interfaz de comunicaciones . . . . .	221
1.11	Programación del microprocesador . . . . .	228
1.11.1	Lenguaje de máquina . . . . .	228
1.11.2	Lenguaje ensamblador. . . . .	230
1.11.3	El ensamblador . . . . .	232
1.11.4	Lenguajes de alto nivel . . . . .	233
1.11.5	El compilador . . . . .	234

**ANEXO II**

**Microprocesador 8748 . . . . . 236**

**ANEXO III**

**Norma EIA-RS-232C . . . . . 248**

**ANEXO IV**

**Fotografías. . . . . 254**



## **INTRODUCCION**

## INTRODUCCION

En las últimas décadas la industria electrónica ha sido objeto de cambios significativos, el uso de dispositivos semiconductores aumento la confiabilidad en los equipos, reduciendo el tiempo medio entre fallas, el uso de transistores y después de circuitos integrados hizo posible reducir el costo de fabricación de equipos electrónicos. En los últimos años uno de los circuitos integrados de más popularidad ha sido el microprocesador, circuito utilizado en el diseño de diversos equipos.

El uso de microprocesadores se ha incrementado grandemente, teniendo aplicaciones de estos en equipos diversos, el bajo costo de estos circuitos y su capacidad de procesamiento es lo que los ha hecho populares en el diseño de nuevos equipos, principalmente en aquellos en los que se requiere programar las operaciones.

El uso de estos en el diseño de equipos para adquisición de datos, es común ya que en este tipo de equipos se programa el tipo de variables a medir, el resultado de estas mediciones es almacenado en dispositivos de memoria; cintas magnéticas o memorias de estado sólido. La recuperación de la información almacenada requiere el empleo de equipos para "leer" esta

información.

Los sistemas de adquisición de datos tradicionales utilizan como medio de registro, graficadores los cuales registran sobre papel previamente impreso con las anotaciones necesarias, el valor de la variable a lo largo del tiempo. La recuperación de información debe hacerse en forma visual, lo que representa una tarea de muchas horas-hombre, resultando esta muy cansada y con un alto grado de fallas en las realización de las lecturas. Cuando se cuenta con este tipo de información la creación de archivos, necesarios para el procesamiento en una computadora, involucra también una gran cantidad de horas-hombre, comparadas con el tiempo necesario para su procesamiento.

La solución a este tipo de problemas se realiza empleando equipos lectores de cinta magnética, cuando este es el medio de almacenamiento utilizado por el equipo de adquisición de datos, en este tipo de equipo se tiene el inconveniente por parte del usuario de depender de toda la línea de productos que ofrece el mismo fabricante, ya que el formato con el cual son registrados los datos pocas veces es dado a conocer al usuario, además del alto costo de este tipo de equipo, empleado en la lectura de información almacenada en cintas magnéticas.

Si el medio de registro de la información es en memorias de estado sólido, la recuperación de esta requiere la utilización de un lector de memorias, en este trabajo se describe el diseño de un equipo lector de memorias, empleado con el sistema ANEMODATA diseñado en el Instituto de Investigaciones Eléctricas, este sistema de adquisición de datos adquiere datos de velocidad y dirección del viento en intervalos de 15 minutos, almacenando estos en memorias de estado sólido.

La descripción del lector de memorias diseñado esta dividida en cinco capítulos, además de varios anexos en los que se tiene información adicional sobre los componentes del equipo.

En el Capítulo I se presenta la metodología de diseño utilizada en el diseño de este equipo, describiendo en forma breve cada etapa involucrada.

En el Capítulo II se realiza una descripción de los elementos de hardware empleados, así como la forma de conexión de estos. Son descritas las funciones realizadas considerando la parte de la que forman parte.

El Capítulo III contiene la información referente al software desarrollado para la operación del equipo, son descritos los comandos utilizados en el protocolo de comunicación, así como la función realizada por estos, y la

forma de selección, se incluyen los diagramas de flujo de cada subrutina y su listado correspondiente en lenguaje ensamblador.

El Capítulo IV contiene la información necesaria para el ensamble de una unidad del lector de memorias, incluye además las pruebas de operación a las que debe ser sometido el equipo para asegurar una adecuada operación del equipo; se mencionan posibles fallas del equipo, así como la operación a realizar para la corrección de la misma. En cuanto al mantenimiento se indican recomendaciones durante el empleo de este equipo.

En el Capítulo V son descritos los pasos a seguir durante la instalación y operación del equipo, así como recomendaciones necesarias cuando se tengan fallas en el equipo, se incluye además la especificación del equipo.

El Anexo I contiene los conceptos teóricos relacionados al funcionamiento del microprocesador y los elementos empleados en un sistema basado en un microprocesador.

El Anexo II contiene la descripción de las características del microprocesador 8748, empleado en el diseño del lector de memorias LM-1-IIE.

El Anexo III describe las características, mecánicas y eléctricas de la norma EIA-RS-232C.

**En el Anexo IV se muestran fotografías del prototipo de  
Lector de Memorias desarrollado.**

**CAPITULO I**

**METODOLOGIA DE DISEÑO**

---

## CAPITULO I

### METODOLOGIA DE DISEÑO

#### 1.1 INTRODUCCION

En el presente capitulo se presenta una descripción de la metodología empleada durante el desarrollo del equipo lector de memorias objeto de este trabajo, se incluye la información referente a cada etapa considerada en el desarrollo de un sistema tanto en el aspecto de hardware como software.

#### 1.2 METODOLOGIA DE DISEÑO

Durante el desarrollo de cualquier equipo o sistema es necesario establecer metas a alcanzar y orden para alcanzar la culminación del proyecto, estas pueden agruparse en dos fases: planeación y diseño. Cada una de estas involucra una o mas etapas, las cuales son aplicables durante el diseño de un equipo electrónico cualquiera, en cuanto al lector de memorias descrito en este trabajo, la aplicación de cada una de las tareas involucradas es descrita considerando el medio en el que fué realizado.



Las etapas a considerar dentro de cada fase durante el desarrollo de un nuevo equipo son las siguientes:

#### **FASE DE PLANEACION**

1. Definición del sistema.
2. Planeación del hardware/software.
3. Análisis de los requisitos del hardware/software.

#### **FASE DE DISEÑO**

1. Implementación del hardware.
2. Diseño del software.
3. Integración del hardware y software.
4. Pruebas de operación y funcionamiento.

### **1.3 DEFINICION DEL SISTEMA**

Para definir completamente un sistema se requiere de: Análisis del sistema, Localización de las funciones de hardware y software y una especificación del sistema.

#### **1.3.1 ANALISIS DEL SISTEMA**

Esta es la primer tarea involucrada en la definición de un sistema, para lo cual es necesario establecer los objetivos del equipo a desarrollar, factibilidad y un análisis costo-beneficio de los mismos.

El objetivo a alcanzar consiste en el desarrollo de un equipo basado en un microprocesador, por medio del cual recuperar la información almacenada en dispositivos de memoria de estado sólido y transferir esta a una microcomputadora, a través de un puerto de comunicación serie. Los tipos de memorias que se desea leer son memorias EPROM 2716 y 2532.

La estimación de factibilidad se realiza tomando en cuenta los elementos de que se dispone para la realización de este proyecto, los que incluyen, información técnica de elementos a utilizar, así como equipo de soporte. Para el diseño del software se empleó el sistema de desarrollo HP-64000, donde se realizó el ensamble del programa desarrollado en lenguaje ensamblador y se programaron las instrucciones en la memoria de programación, la información para el uso de este sistema es la proporcionada por el fabricante en sus manuales. Para los elementos de hardware se utilizó la información proporcionada por los fabricantes en sus manuales.

Para la realización de un análisis costo-beneficio es necesario una estimación del costo del proyecto, donde se contemple el costo de los elementos utilizados, así como el derivado por la utilización de equipos de soporte, incluyendo gastos de personal. El costo de los circuitos a utilizar es estimado empleando listas de precios proporcionadas por los fabricantes o sus distribuidores, mientras que el costo de

utilización de equipo de soporte, se estima considerando el tiempo de utilización del mismo, así como el precio estimado por hora de empleo del mismo, costo determinado o establecido por los propietarios del equipo. De igual forma se estima el costo de utilización de equipo de utilería.

El beneficio inmediato que se obtendrá con este equipo, es la utilización del mismo en el sistema ANEMODATA, desarrollado en el Instituto de Investigaciones Eléctricas y eliminar la importación de equipo similar de costo mayor.

### 1.3.2 LOCALIZACION DE FUNCIONES

Esta tarea tiene por objeto designar que funciones serán ejecutadas por elementos de "hardware" y cuales en base a "software".

- Las funciones por realizar en base a las características del hardware son: conversión paralelo-serie y serie-paralelo de señales que serán intercambiadas por el lector de memorias y la microcomputadora, así como el acondicionamiento de señales en cuanto a niveles de voltaje se refiere, ya que para la transmisión de datos es necesario que la señal cuente con niveles adecuados, lo mismo en el caso de las que recibirá el lector de memorias, para de esta forma estar dentro de los estándares de la norma RS-232C.

- Las funciones realizadas en base a "software" comprenden funciones como, la selección del formato de comunicación, la lectura de memorias, selección del tipo de memoria a leer, así como el protocolo de comunicación con el que se selecciona la función por realizar.

### 1.3.3 ESPECIFICACION DEL EQUIPO

Durante esta tarea se incluye la definición de objetivos, así como criterios que sirvan de base al desarrollo del equipo, especificando en forma clara metas a alcanzar en cada etapa.

El objetivo de este trabajo consiste en desarrollar un equipo basado en un microprocesador por medio del cual recuperar la información almacenada en dispositivos de memoria de estado sólido y transmitir esta a una microcomputadora a través de un puerto de comunicación serie.

Las funciones a realizar por el equipo serán: Despliegado de un texto de presentación, despliegado del menú de memorias que pueden ser leídas (EPROM 2716 y 2532) y lectura de memorias, en bloques de 32 bytes. El inicio de estas operaciones se controlará por medio de comandos seleccionados en el protocolo de comunicación, los cuales son enviados por la microcomputadora, la descripción de estos, se realiza en el

capítulo tres de este documento.

La cantidad de memoria a utilizar, como memoria de programación del microprocesador se estima de acuerdo a las funciones que ejecutará el equipo considerando además la capacidad necesaria para modificaciones posteriores o ampliación de las funciones originales.

#### 1.4 PLANEACION DEL HARDWARE/SOFTWARE

La planeación del hardware comprende la elección de circuitos que serán utilizados, considerando sus características en cuanto al tipo de salida y consumo de energía, además de disponibilidad de estos en el mercado nacional.

Durante la elección del microprocesador, se toman en cuenta factores como: costo, consumo de energía, capacidad de procesamiento, flexibilidad, velocidad y otros, de acuerdo a características requeridas en el producto final.

Los equipos periféricos están en función de las operaciones que serán ejecutadas, el lector de memorias emplea una interfaz de comunicación, que realiza la conversión serie-paralelo y paralelo-serie de señales intercambiadas con la microcomputadora.

En lo referente al software, se hace una estimación del costo del mismo, para lo cual se realiza una evaluación de recursos disponibles, estableciendo fecha probable de terminación.

En cuanto a herramientas de que se disponga estas son agrupadas de acuerdo a la parte del diseño en la que serán utilizadas, hardware o software; se incluyen herramientas de soporte y utilería.

La estimación del costo del hardware se hace considerando el costo propio de los circuitos que serán utilizado. En cuanto al software se realiza considerando factores humano, técnico, y ambiental, que son los que determinan el costo final.

#### 1.5 ANALISIS DE LOS REQUISITOS DE HARDWARE/SOFTWARE

La razón de esta etapa es cubrir huecos en la especificación del equipo, así como determinar limitaciones del diseño, para lo cual es necesario una comunicación adecuada entre la persona que desarrolla el proyecto y la que lo solicita.

Las funciones a ejecutar por elementos de hardware, requieren la utilización de una interfaz de comunicación, con la cual establecer el protocolo de comunicación con la microcomputadora, además de circuitos para el

acondicionamiento de las señales intercambiadas. La ejecución de estas operaciones se realiza con el empleo de una interfaz de comunicación y de circuitos que trasladan de nivel las señales intercambiadas respectivamente. La descripción de los circuitos con los que se realizan estas operaciones se describe en el capítulo dos de este documento.

En cuanto al software, las rutinas a realizar están de acuerdo a las operaciones que debe realizar el equipo, cada operación puede requerir la utilización de una o más subrutinas, estas son descritas en el capítulo tres.

#### 1.6 IMPLEMENTACION DEL HARDWARE

Durante esta etapa se describe la forma de implementación del equipo, en cuanto al hardware se refiere, considerando la razón por la que se ha utilizado un circuito en particular y que función realiza. Una vez que se tiene la estructura física se procede a diseñar el software necesario para la operación del mismo.

La implementación del hardware utilizado se realizo en partes, esto con el fin de realizar la integración del software desarrollado para cada etapa en particular con más facilidad. En el capítulo dos se realiza una descripción de los elementos empleados en la ejecución de las operaciones necesarias para el funcionamiento del equipo, así como la

manera en que se encuentran conectados.

## 1.7 DISEÑO DEL SOFTWARE

Durante esta etapa se describen las operaciones realizadas en base al software. El programa consta de varias subrutinas las cuales realizan una operación determinada, la descripción de las mismas se presenta en el capítulo tres de este documento.

La operación a realizar por el equipo lector de memorias, es seleccionada por comandos enviados por la microcomputadora. Estos se seleccionaron de acuerdo al protocolo de comunicación empleado, las funciones inicializadas por estos son las siguientes: El desplegado del texto de presentación del equipo, con la subrutina TEXTO; El menú de las memorias es desplegado por la subrutina MENU; mientras que la lectura de memorias 2716 es realizada por la subrutina MENIA, y de memorias 2532 por la subrutina DIREC. Las subrutinas de lectura de memorias hacen uso de otras con las cuales se controlan las operaciones ejecutadas durante la lectura de un dispositivo de memoria, se emplean subrutinas para el direccionamiento de las localidades de memoria, conteo de los datos recuperados, transmisión de datos y habilitación de recepción de comandos, esto cuando son recuperados 32 bytes, el comando a recibir indica la realización de la lectura de 32 bytes más de datos. La subrutina utilizada para direccionar



las localidades de memoria registra el número de localidades direccionadas y determina cuando se ha direccionado la memoria en su totalidad.

#### 1.8 INTEGRACION DE HARDWARE Y SOFTWARE

En esta etapa se describe el proceso mediante el cual se integran los elementos de hardware con el software desarrollado. El proceso de integración se realizó por partes, cada una de estas incluye una parte del hardware implementado y sus correspondientes subrutinas, con las que se controlan las operaciones efectuadas. La verificación de cada etapa realizada involucra pruebas de funcionamiento, en las cuales se comprueba que el software ejecute las tareas programadas.

La integración final del hardware y software se hizo agrupando todas las subrutinas desarrolladas, en un solo programa, las pruebas realizadas para verificar la integración de estos elementos se hizo efectuando pruebas de operación y funcionamiento, como se describe en el siguiente inciso.

## 1.10 PRUEBAS DE OPERACION Y FUNCIONAMIENTO

En esta etapa se establecen las pruebas a las que será sometido el equipo una vez que ha quedado terminado para determinar si funciona adecuadamente. En las pruebas de operación, se debe verificar que el lector de memorias "conteste" únicamente cuando reciba uno de los comandos seleccionados para iniciar sus operaciones.

Para verificar la operación del lector durante la integración del hardware implementado y las subrutinas desarrolladas para esta parte de hardware, deben realizarse pruebas con las que se garantice un funcionamiento adecuado de la parte que ha sido integrada.

El funcionamiento del equipo puede ser probado, comparando el resultado de lectura de memorias realizado con otro equipo, o bien empleando memorias con información conocida de antemano, y así determinar si "lee" correctamente la información almacenada en los dispositivos de memoria.

## CAPITULO II

### DISEÑO DETALLADO

## CAPITULO II

### DISEÑO DETALLADO

#### 2.1 INTRODUCCION

En el presente capítulo se presenta una descripción de la fase de diseño referente al hardware del equipo lector de memorias LM-1-IIE. Esta se hace considerando en forma separada las etapas de diseño electrónico y mecánico del equipo.

La parte referente al diseño electrónico se describe mediante el diagrama del equipo, el cual es dividido en tres secciones principales, por medio de las cuales son efectuadas las operaciones necesarias para la lectura de los dispositivos de memoria.

En la parte correspondiente al diseño mecánico, se detalla la forma física dada al equipo.

Por último se presenta la lista de partes necesarias para el ensamble de una unidad.

## 2.2 DESCRIPCION DEL DIAGRAMA ELECTRONICO

El lector de memorias LM-1-IIE está constituido por tres secciones: unidad de control, unidad de lectura y la unidad de comunicación, por medio de las cuales son efectuadas las operaciones del equipo.

En la figura 2.1 se muestra el diagrama a bloques de las secciones que componen este equipo.

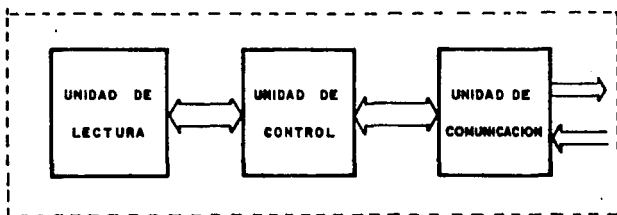


Figura 2.1. Diagrama a bloques del equipo  
Lector de Memorias.

El funcionamiento del LM-1-IIE en forma general es el siguiente: Una vez que el equipo ha sido conectado a la computadora personal y este ha sido energizado, se procede a insertar la memoria a leer en el conector con que cuenta el equipo.

Cuando la memoria ha sido conectada, el microprocesador del LM-1-III queda en espera de el comando que le indique iniciar la lectura del dispositivo de memoria.

Al recibir este comando se inicia la lectura obteniendo la información de la memoria, la cual es transmitida hacia la computadora personal a través de la unidad de comunicación, para que sea procesada. Una vez que la información ha sido procesada, el usuario puede realizar una adecuada interpretación de la información adquirida.

#### 2.2.1 UNIDAD DE CONTROL

Es en esta sección donde se controlan todas las señales y se realizan las operaciones necesarias para el funcionamiento del equipo. Para llevar a cabo este control se tiene grabado en la memoria externa (A7) el programa\* bajo el cual se realizan las acciones necesarias para la correcta operación del equipo.

La unidad está compuesta por circuitos que efectúan el control de las operaciones realizadas por el equipo. Estos circuitos son el microprocesador 8748 (A4), la memoria de

---

\* La descripción de este programa se realiza en el capítulo tres de este documento

programación. (A7), (la cual es una memoria MCM2716 externa al circuito que contiene al microprocesador) y los circuitos empleados para direccionar a esta memoria A5 y A6 (circuitos MC14042). En la figura 2.2 se muestra el diagrama eléctrico de esta unidad.

El microprocesador empleado en el diseño de este equipo, cuenta dentro del mismo circuito integrado con una memoria EPROM de capacidad igual a 1 Kbyte, está puede ser empleada para almacenar el programa bajo el cual se realizan las operaciones del equipo, sin embargo, esta no fué utilizada.

La razón por la cual esta memoria no fué usada, es la de contar con una memoria de mayor capacidad, en caso de que las funciones originales del equipo sean modificadas y estas requieran de una memoria de mayor capacidad.

Las señales de los comandos\*\* con los que se inician las operaciones llegan al microprocesador a través de la unidad de comunicación, la cual también es utilizada para la transmisión de datos recuperados de las memorias.

La velocidad a la cual son efectuadas las operaciones del microprocesador, está determinada por la referencia de

---

\*\* La descripción de los comandos seleccionados se muestra en el capítulo tres de este documento.

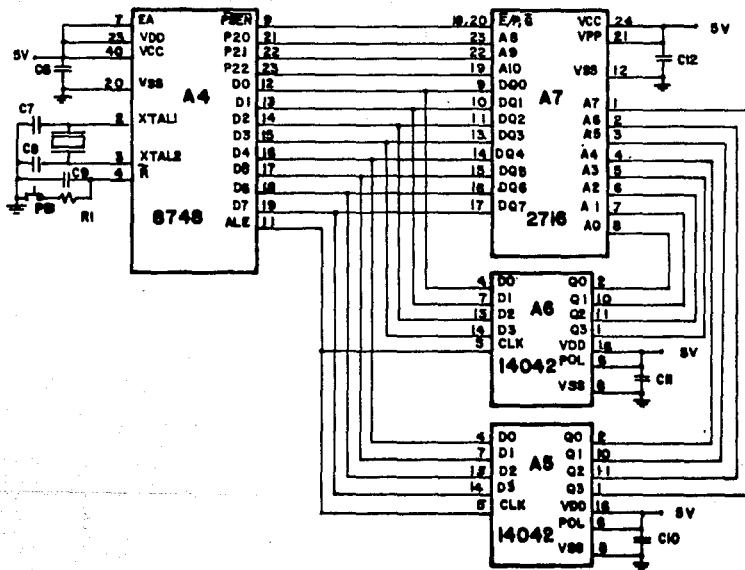


FIGURA 2.2 UNIDAD DE CONTROL



frecuencia de que dispone, en este caso se obtiene de un cristal de cuarzo que oscila a una frecuencia de 4.608 Mhz con lo que cada ciclo tiene una duración de 3.25 uS.

La frecuencia de oscilación del cristal de cuarzo, fué seleccionada de tal forma, que a partir de ésta fuesé posible obtener en las salidas ALE y T0 del microprocesador, señales con las cuales proporcionar las referencias de frecuencia para la interfaz de comunicación.

En la salida ALE se obtiene una señal con una frecuencia de 307.2 Khz., esta referencia de frecuencia determina, la velocidad a la cual se realiza el intercambio de señales entre el lector de memorias y la microcomputadora. Para obtener la frecuencia que determina la velocidad de intercambio de señales, esta señal pasa a través de un divisor (SN74161), en cuya salida Qd se obtiene una señal con frecuencia de 19.2 Khz, que es la que realmente determina la velocidad de intercambio entre el lector de memorias y la microcomputadora.

En la salida T0 del microprocesador se obtiene una señal con una frecuencia de 1.536 MHz., esta se utiliza para proporcionar la referencia de frecuencia propia de la interfaz de comunicación. En base a esta referencia son efectuadas las operaciones de la interfaz de comunicación (P8251A).

Puesto que en este equipo el microprocesador funciona con una memoria externa, la búsqueda de instrucciones en la misma es controlada por medio de las señales ALE y PSEN del microprocesador. En la figura 2.3 se muestra el diagrama de tiempo de las señales empleadas en la búsqueda de instrucciones en la memoria externa.

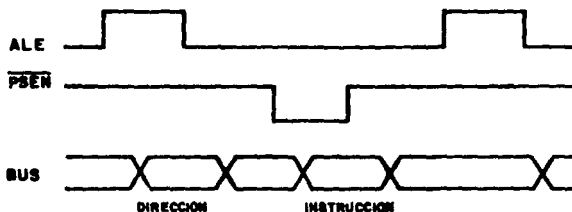


Figura 2.3. Ciclo de búsqueda de instrucciones en la memoria externa.

El microprocesador seleccionado en el diseño de este equipo es el 8748 de INTEL, esto debido a su gran versatilidad y relativa sencillez de operación.

El tipo de memoria utilizada como memoria de programación del microprocesador, es externa al circuito que contiene al microprocesador, tiene una capacidad de 2Kbytes (Memoria EPROM MCM2716), por sus características el contenido de la misma puede ser borrado por medio de luz ultravioleta.

### 2.2.2 UNIDAD DE LECTURA

La recuperación de información almacenada en los dispositivos de memoria, se realiza a través de esta unidad. La cual está constituida por los circuitos A1 (MC14599), A2 (MC14599), A3 (MC14051) y los conectores señalados como S1 y K en el diagrama. En la figura 2.4 se muestra el diagrama eléctrico de esta unidad.

Para leer una memoria, es necesario colocarla en la base (S1) con que cuenta el equipo o bien que el cartucho que contiene la memoria sea insertado en el conector K.

Para iniciar la lectura, el microprocesador debe recibir de la computadora personal el comando que le indique iniciar la lectura del dispositivo de memoria.

Una vez que el lector de memorias ha recibido el comando adecuado, la primera función efectuada es el direccionamiento de la localidad de memoria a leer, lo cual es efectuado a través de los circuitos A1 y A2. Después de colocar la dirección, se realiza la lectura del dato almacenado en la localidad direccionada a través del circuito A3.

El dato obtenido es transferido hacia el microprocesador, donde se llevan a cabo las operaciones necesarias para transmitirlo a través de la unidad de comunicación, hacia la

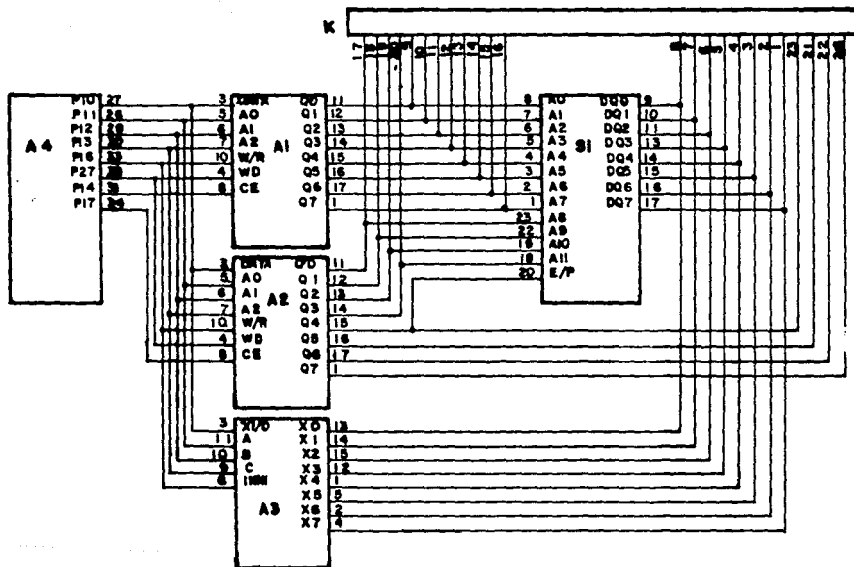


FIGURA 2.4 UNIDAD DE LECTURA

computadora personal.

La lectura de los datos de la memoria se realiza en bloques de 32 bytes, por lo que para leer mas datos se requiere que el microprocesador reciba el comando que le indique efectuar la lectura de otro bloque de 32 bytes de datos. Este proceso se repite hasta obtener la totalidad de datos almacenados en la memoria.

En la lectura de las memorias, es necesario tener en cuenta la secuencia en la que se aplican las señales, que habilitan o deshabilitan la lectura de datos de la memoria. La secuencia en que son aplicadas las señales necesarias para la lectura, se muestra en la figura 2.5.

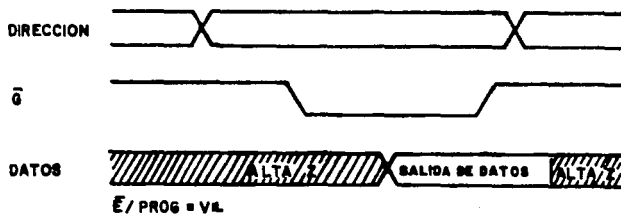


Fig. 2.5 Diagrama de tiempo para la lectura de datos de la memoria.

### 2.2.3 UNIDAD DE COMUNICACION

La función de esta unidad es, servir como medio de enlace entre el lector de memorias LM-1-IIE y la computadora personal.

Esta unidad como se muestra en la figura 2.6, está formada por la interfaz de comunicación, una sección de acondicionamiento de señales y un puerto de entrada-salida.

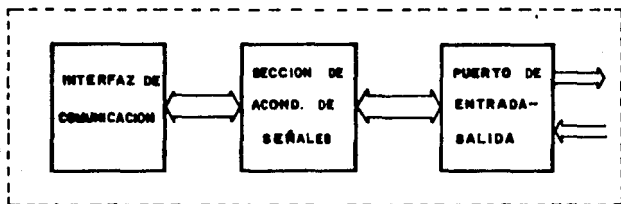


Figura 2.6 Diagrama a bloques de la Unidad de Comunicación.

Está constituida por los circuitos integrados P8251A (A8), MC1488 (A10), MC1489 (A11), SN74161 (A9) y MC14049 (A12). El diagrama eléctrico de esta unidad se muestra en la figura 2.7.

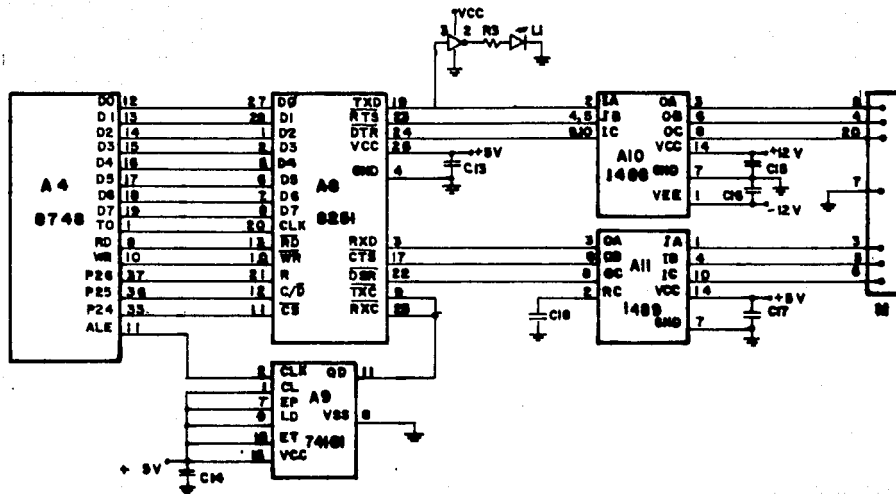


FIGURA 2.7 UNIDAD DE COMUNICACION

La interfaz de comunicación empleada es el circuito 8251 de INTEL, seleccionada por su gran versatilidad en cuanto a los formatos de comunicación serie que puede manejar. Esta es una interfaz universal de transmisión-recepción síncrona-asíncrona (USART), la cual está diseñada para comunicación entre microcomputadoras, la forma de operación de esta interfaz es programada mediante el microprocesador.

En la programación de esta interfaz, se selecciona la forma de comunicación asíncrona, empleada para el intercambio de señales entre el lector de memorias y la microcomputadora; así como el protocolo de comunicación con el que serán manejadas las señales (Comandos y datos), tanto en la recepción como en la transmisión.

Al seleccionar la comunicación asíncrona, se elige también el factor que determina la razón de Bauds. El factor seleccionado para este equipo es 16. Este determina el valor de las referencias de frecuencia necesarias, para efectuar el intercambio de señales (datos y comandos), a una velocidad de 1200 Bauds.

Para obtener esta velocidad, es necesario que la referencia de frecuencia utilizada, sea 16 veces el valor de la velocidad; esto es 19.2 KHz. Esta referencia se proporciona a la interfaz por sus terminales RxC y TxC, mediante la terminal Qd del circuito A9 (SN74161), la cual a



su vez es una señal que se adquiere de la terminal ALE del microprocesador, esta última tiene una frecuencia de 307.2 KHz.

En cuanto a la referencia propia de la interfaz de comunicación, se requiere que sea como mínimo 4.5 veces mayor que la frecuencia del reloj de entrada, además de que el período de esta señal debe estar dentro del rango de 0.42 uS a 1.35 uS. Esta referencia se obtiene de la terminal T0 del microprocesador, ya que esta cumple con los requisitos para la referencia de frecuencia propia de la interfaz. La frecuencia que se obtiene en T0 es de 1.536 MHz.

Esta interfaz recibe comandos en forma serie de la computadora personal y los convierte a un formato paralelo para el microprocesador. Acepta también datos en paralelo del microprocesador los cuales convierte a un formato serie para ser transmitidos hacia la computadora personal, indica también al microprocesador cuando ha recibido o transmitido un carácter.

El circuito A12 es empleado para obtener un indicador que permite conocer cuando se está efectuando la lectura de un dispositivo de memoria. Lo que permite al operador conocer el momento en que puede retirar la memoria del socket, esto cuando el indicador se encuentra apagado.

### 2.2.3.1 PROTOCOLO DE COMUNICACION

El protocolo o formato de comunicación es el marco empleado en el intercambio de señales, entre el lector de memorias y la microcomputadora.

Puesto que el intercambio de información tiene lugar en código ASCII. En la comunicación serie asíncrona, cada carácter es precedido por un bit de arranque, con el cual se indica el inicio y terminado con uno o dos bits de parada, por medio de los cuales se indica el fin.

Dentro de este marco es posible utilizar un bit para detección de errores, este bit se conoce como bit de paridad, no forma parte del carácter, pero puede ser utilizado dentro del marco, en caso de ser utilizado puede ser par, impar o no ser empleado.

Cuando se selecciona paridad par, el bit de paridad será cero si el número de bits del carácter que tienen el nivel lógico uno es par, y será uno si este número es impar.

Si en la transferencia de un carácter especificado con paridad par, la suma total de "unos" es impar, entonces ha ocurrido un error en la transmisión.

En el protocolo de comunicación empleado por el lector de memorias LM-1-IIE son establecidas las siguientes características:

- Comunicación serie asíncrona.
- Longitud de la palabra de 8 bits.
- Un bit de parada.
- No se emplea el bit de paridad.
- Razón de bauds de 16.

En la siguiente figura se muestra el formato de comunicación empleado por el equipo lector de memorias.

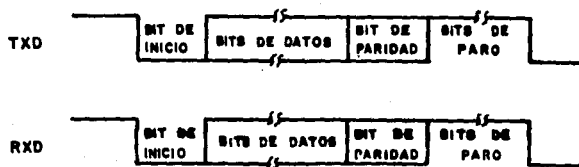


Figura 2.8 Protocolo de comunicación.

### 2.2.3.2 ACONDICIONAMIENTO DE SEÑALES

La sección de acondicionamiento de señales, está formada por los circuitos A10 (MC1488) y A11 (MC1489).

En esta se realiza el traslado de los niveles de voltaje, de señales que recibe el lector de memorias, así como de las que este transmite.

Las señales recibidas por el lector de memorias tienen niveles de voltaje de acuerdo a la norma RS-232C, por lo que, para que puedan ser interpretadas por los circuitos del lector, se requiere que el nivel de estas se traslade a un nivel TTL (0 a +5 volts), y de esta forma sean interpretadas.

En forma parecida, las señales que deben transmitirse, tienen que trasladarse a un nivel de voltaje compatible con el de la norma RS-232C, esto es a un nivel de +10 volts, para de esta forma puedan ser transmitidas e interpretadas en el extremo receptor, por la microcomputadora.

El puerto de entrada-salida empleado como medio de comunicación entre el lector de memorias y la computadora personal es un puerto serie RS-232C.

#### 2.2.4 FUENTE DE ALIMENTACION

La fuente de alimentación empleada en el equipo lector de memorias proporciona los voltajes requeridos por los diferentes circuitos utilizados. Los diferentes voltajes que se obtienen de esta fuente son +5, +12 y -12 volts.

Para obtener estos voltajes se emplea un transformador con dos devanados en el secundario, esto debido a que practicamente se tienen dos fuentes, una de +5 volts y otra complementaria en la cual se obtienen voltajes de +12 y -12 volts. Como elemento rectificador en cada fuente se emplea un puente de diodos. El voltaje de cada devanado secundario esta en función del voltaje de salida regulado que se desea obtener; mientras que la corriente que debe soportar esta en función de la corriente consumida a plena carga por el circuito.

Para determinar el voltaje de los devanados secundarios del transformador se hace uso de la ecuación (1) mostrada a continuación:

$$V_{AC} = \frac{(V_{OUT} + V_{REC} + V_{RECT} + V_{RIZO}) V_{NOM}}{0.92 V_{lowline} \sqrt{2}}$$

Donde:

0.92 = eficiencia típica del rectificador.

Vout = voltaje de salida en DC.

Vreg = diferencia de voltajes de salida y entrada al regulador (3 volts.).

Vrect = voltaje en cada diodo del puente rectificador (1.25 V.).

Vrizo = voltaje de rizo 10% VDC

Vca = voltaje en el secundario del transformador.

Vnom = voltaje de línea (125 Volts ca).

Vlowline = voltaje mínimo de línea (100 Volts).

Para determinar la corriente que circulará por cada uno de los devanados se emplea la relación.

$$I_{rms} = 1.8 \times I_{plena\ carga}. \quad (2)$$

El primer devanado, utilizado para obtener el voltaje de +5 volts y proporcionar una corriente de 400mA, queda especificado de acuerdo con las ecuaciones 1 y 2 como un

devanado a 10 volts y 720 mA, aunque en la especificación de la corriente se considerará una tolerancia del 10% de ahí que la especificación final para este devanado sea 10 v y 800 mA.

El segundo devanado, utilizado para obtener voltajes de +12 y -12 volts (fuente complementaria) debe proporcionar una corriente de 150 mA. Por lo que de acuerdo con la ecuación 1 el voltaje del mismo es de 16.75 volts, pero como en esta fuente se emplea un tap central, el voltaje total del devanado es de 32 volts, mientras que la corriente que puede circular a través del mismo de acuerdo a la ecuación 2 es de 270 mA. Considerando una tolerancia del 10% en el caso de la corriente, este devanado queda especificado como un devanado a 32 volts y 300 mA con tap central.

El valor de los capacitores utilizados como filtro a la salida del rectificador, se calcula utilizando la siguiente ecuación.

$$C = \frac{I_L \times 6 \times 10^{-3}}{\Delta V} \quad (3)$$

Donde:

C = capacitancia en farads.

I<sub>L</sub> = corriente a plena carga.

**AV = voltaje de rizo pico a pico. (3 volts)**

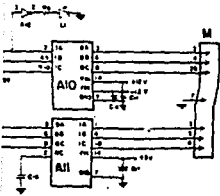
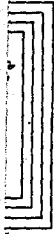
En el caso de la fuente de +5 volts se obtiene de la ecuación 3 un capacitor de valor de 800 uf, el valor comercial más cercano es de 1000 uf/20v.

Para la fuente complementaria considerando el mismo voltaje de rizo, se obtiene de la ecuación 3 que se requiere de un capacitor de 300 uf, empleando el valor comercial más cercano de 470 uf/25v.

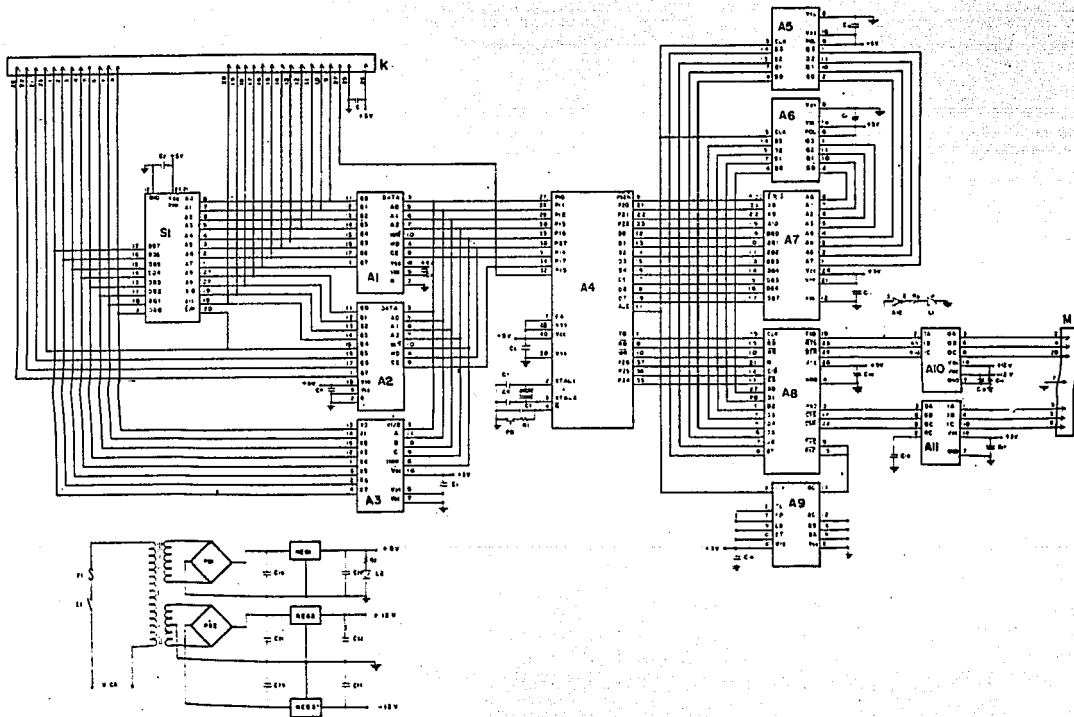
Los capacitores empleados a la salida del regulador de voltaje, funcionan como estabilizadores y el valor de los mismos es de 10 uf.

#### **2.2.5 DIAGRAMA ELECTRICO DEL LECTOR DE MEMORIAS**





INSTITUTO DE INVESTIGACIONES ELECTRICAS		
REGISTRADO DE PATENTE N.º 111/1957		
NO. VOUCHER P.L.S. -	DIAGRAMA	
NO. TABLA DE IDENTIFICACION	ELECTRONICO	
NO. TABLA DE NOMBRES	LECTOR DE	
	MEMORIAS	
	LM-1-11E	



INSTITUTO DE INVESTIGACIONES	
ELECTRONICA Y ELECTRICIDAD	
TIPO DE PLANO =	DIAGRAMA
NO. DE C. A. PARALELA =	ELECTRONICO
NO. DE C. A. EN SERIE =	LECTOR DE
	MEMORIAS
	LM-1-11E

### 2.3 LISTA DE PARTES ELECTRONICAS

A continuación se presenta la información referente a las partes que forman el equipo.

Se incluyen los datos necesarios para su localización en el plano, así como para su adquisición.

La información contenida en las Tablas 2.1 a 2.9 descritas a continuación es la siguiente:

- A - Número de parte.
- B - Descripción (Español - Inglés).
- C - Localización en el plano.
- D - Cantidad.
- E - Número de parte comercial.
- F - Fabricante.
- G - Precio unitario.
- H - Precio total.
- I - Tipo de moneda (P - Moneda Nacional. D- Dolares).
- J - Mes y año de cotización.
- K - Proveedor de acuerdo a la lista anexa.
- L - Catálogo de acuerdo a la lista anexa.
- M - Página de catálogo.

### 2.3.1 LISTA DE PROVEEDORES

1. KLEIGO SUPPLY, INC.  
2418 Treadslow Lane  
Houston, Texas 77067  
Phone (713) 893-4681  
Telex 790-550
2. PROVEEDORA ELECTRONICA, S.A.  
Prolg. Moctezuma Ote. No. 24  
Col. Romero de Terreros  
04310 México, D. F.
3. DISTRIBUIDORA SAGO, S. A.  
Rep. de El Salvador 20-F  
06000 México, D. F.  
Tel: 510-8846, 518-5733
4. TRANSFORMADORES AVILA  
Meave No. 28-3  
México, D. F.  
Tel.: 510-4793
5. CRICUSA  
Kansas No. 18  
Col. Napoles  
México, D. F.
6. Los dispositivos marcados como proveedor "A" son de uso general y pueden adquirirse en tiendas de refacciones electrónicas.

### 2.3.2 LISTA DE CATALOGOS

1.       **MOTOROLA CMOS DATA BOOK**  
CMOS Marketing  
Motorola Semiconductor Products Inc.  
3501 Ed Bluestein Blvd.  
Austin, Texas 78721  
(512) 928-6000
  
3.       **VOLTAGE REGULATOR HANDBOOK**  
**NATIONAL SEMICONDUCTOR CORP.**  
2900 Semiconductor Drive  
Santa Clara, California 95051  
(408) 737-5000  
TWX (910) 339-9240
  
6.       **MOTOROLA MEMORY DATA MANUAL 1980**  
Motorola Semiconductor Products Inc.  
3501 Ed Bluestein Blvd.  
Austin, Texas 78721  
(512) 928-6000
  
7.       **THE TTL DATA BOOK**  
For Design Engineers  
Second Edition 1976  
Texas Instruments Inc.  
Post Office Box 5012  
Dallas, Texas 75222
  
12.      **MOTOROLA SEMICONDUCTOR**  
**MASTER SELECTION GUIDE AND CATALOG**  
Motorola Semiconductor Products Inc.  
3501 Ed Bluestein Blvd.  
Austin, Texas 78721  
(512) 928-6000
  
14.      **NEWARK ELECTRONICS CATALOG**  
4801 N. Ravenswood  
Chicago, Il 60640  
(312) 784-5100

- 19A. BERG ELECTRONICS CATALOG  
New Cumberland, PA 17070  
(717) 938-6711
- 19Q. AMP INC.  
AMPLIMITE CATALOG  
Harrisburg Pennsylvania 17105  
(717) 564-0100  
TWX 510-657-4110
22. INTEL COMPONENT DATA CATALOG  
INTEL CORPORATION  
Literature Department SV3-3  
3065 Bowers Avenue  
Santa Clara, CA 95051

TABLE 2.1

LISTA DE PARTES ELECTRONICAS      IA-1-11E

A	B	C	D	E	F	G	H	I	J	K	L	M
CELDA	<p>CIERCALOS INTEGRADOS INTEGRATED CIRCUITS</p> <hr/> <p>BIENOSCOMPUTADOR CON EPROM INTEGRADA. BIENOS CHIP EPROM BIENOS COMPUTER.</p>	A4	1	P07408	INTEL	20.50	20.50	D	DIC/85	1	22	6-53
CELDA	<p>BIENOS A EPROM DE 20480 BITS. BIENOS 8 BIT EPROM BIENOS PROM.</p>	A7	1	HC2716	NOTOROLA	3.94	3.94	D	DIC/85	1	6	2-125
CELDA	<p>LATCH CUADRUPLE. QUAD LATCH.</p>	A9, A6	2	HC140423	NOTOROLA	685	1370	F	002/86	3	1	7-116
CELDA	<p>LATCH DIRECCIONABLE DE 8 BITS. 8-BIT ADDRESSABLE LATCHES.</p>	A1, A2	2	HC145998	NOTOROLA	2.65	5.30	D	DIC/85	1	1	7-169
CELDA	<p>MULTIPLEXOR/MULTI- PLEXOR ANALOGICO. MUL OF MULTIPLEXER/ DEMULTIPLEXERS.</p>	A2	1	HC140513	NOTOROLA	0.85	0.85	D	DIC/85	1	1	7-133

TABLE 2.2

LISTA DE PARTES ELECTRONICAS LM-1-11R

A	B	C	D	E	F	G	H	I	J	K	L	M
CEL06A	INVERSOR DE COMUNICACION PROGRAMABLE. PROGRAMMABLE COMMUNICATIONS INVERTER.	A0	1	P0231A	INTEL	7.50	7.50	9	BIC/68	1	22	9-168
CEL07A	CONTADOR SINCRONO DE 4 BITS. SYNCHRONOUS 4-BIT COUNTERS.	A9	1	8N74161	TIERRA INSTRUMENT	565	565	9	A02/66	1	7	7-190
CEL08A	INVERSOR HEXAUPLE CON BUFFER. HEX INVERTER BUFFER.	A12	1	HC140490B	ROTOROLA	0.50	0.50	9	ENE/66	1	1	7-189
CEL09A	PRIMER DE LINEA COMPLETO (MS-118C). FIRST IN LINE SWITCH (MS-118C).	A10	1	HC1409	ROTOROLA	1.05	1.05	9	ENE/66	1	12	101
CEL10A	RECEPCION DE LINEA COMPLETO (MS-118C). FIRST IN LINE RECEIVER.	A11	1	HC1409	ROTOROLA	1.05	1.05	9	ENE/66	1	12	101
CEL11A	REGULADOR DE VOLTAJE POSITIVO +5 VOLTS. POSITIVE VOLTAGE REGULATOR.	REG1	1	L07805	NATIONAL ELECTRONIC	395	395	7	OCT/66	3	3	10-67



TABLE 2.3

LISTA DE PARTES ELECTRONICAS												
LA-1-11E												
A	B	C	D	E	F	G	H	I	J	K	L	M
CEL12A	REGULADOR DE VOLTAJE POSITIVO -15 OHMS. 2-TERMINAL POSITIVE REGULATOR.	REG3	1	LN7912	NATIONAL SEMICON- DUCTOR	395	395	9	OCT/66	3	3	10- 180
CEL12B	REGULADOR DE VOLTAJE NEGATIVO -15 OHMS. 2-TERMINAL NEGATIVE REGULATOR.	REG3	1	LN7912	NATIONAL SEMICON- DUCTOR	395	395	9	OCT/66	3	3	10- 180
CEL14A	PUNTE RECTIFICADOR DE DIODOS. BRIDGE DIODE RECTIFIER.	FD1, FD3	2	ND4100A	ROTORAL	2.10	4.20	3	ENE/66	1	14	39
	DIODOS EMISORES DE LUM. LIGHT EMITING DIODE.											
CEL15A	DIODO EMISOR DE LUM. LIGHT EMITING DIODE.	L1,L2	2	SEL1120R	GMCP	0.39	0.56	3	DIC/66	1	-	-
	CAPACITORES. CAPACITORS.											
CEL16A	ELECTROLITICO 1000 W/250. ELECTROLYTIC 1000 W/250.	C19	1	SD1000W16 PI	SPRAGUE	1.37	1.37	3	ENE/66	1	14	213



TABLE 2.3

LISTA DE PARTES ELECTRONICAS														LA-1-112	
A	B	C	D	E	F	G	H	I	J	K	L	M	N		
	RESISTENCIAS FIJAS. FIXED RESISTORS.														
CEL238	270 OHMS 1/4 W 5%. -----	R2	1	SEGUN MARCA	INDIFEREN- TE	0.06	0.06	D	DIC/85	A	-	-			
CEL240	470 OHMS 1/4 W 5%. -----	R3	1	SEGUN MARCA	INDIFEREN- TE	0.06	0.06	D	DIC/85	A	-	-			
CEL250	1 KOHM 1/4 W. 5%. -----	R1	1	SEGUN MARCA	INDIFEREN- TE	0.06	0.06	D	DIC/85	A	-	-			
	CONEXIONES. CONNECTORS.														
CEL260	HAGO DE 25 POSICIO- NES (25-252C). DE POSICIONES DOBLE CONEXION (25-252C).	H	1	206104-2	AMP	5.00	5.00	D	ENE/86	1	190	16			
CEL270	CONEXION DOBLE DE 25 POSICIONES EXHIBITE SIMILAR CON CONEXION EN PINE.	H	1	63566	BERO	3.25	3.25	D	DIC/85	1	190	13			

TABLE 2.6

LISTA DE PARTES ELECTRONICAS		LM-1-11E										
A	B	C	D	E	F	G	H	I	J	K	L	M
	INTERRUPTORES. SWITCHES.											
CEL298	INTERRUPTOR SPST. SWITCH TOGGLE SPST.	11	1	SP4-112- 0101-011	DIMLIGHT	2.16	2.16	D	DIC/88	1	14	800
CEL299	POSIMUTUON N. O. SPST. POSIMUTUON NO SPST.	70	1	FD4-126-00	CUTLER RAMBER	3.40	3.40	D	ENE/86	1	14	802
	QUICROS.											
CEL300	CRISTAL DE CUARZO DE 4.600 MHz. 4.600 MHz CRYSTAL.	11	1	-	CRICUSA	6500	6500	F	SEPT/86		-	-
CEL318	IMPRESA DE CIRCUITO IMPR. DE CIRCUIT PRINT BOARD.											
CEL328	CABLE DE ALIMENTACION CAB. CLAVIUM.(6") POWER SUPPLY CABLE.	-	1	171062	BELDEN CP9	0.00	0.00	D	ENE/86	1	14	643

TABLE 2.7

LISTA DE PARTES ELECTRONICAS														LM-1-11E	
A	B	C	D	E	F	G	H	I	J	K	L	M	N		
CEL33A	PORTAFUSIBLE. PANEL FUSE HOLDER.	F1	1	342001	LITTELFUSE	1.41	1.41	D	DIC/85	1	14	425			
CEL34A	FUSIBLE DE 0.5 A. 0.5 A FUSE.	F1	1	313.500	LITTELFUSE	0.75	0.75	P	ENE/86	1	14	422			
CEL35A	TRANSFORMADOR 120/10-32 V. 120/10-32 V TRANSFORMER.	T1	1	-	TRANSFORMADORES AVILA	5000	5000	P	DIC/86	4	-	-			
CEL36A	DISIPADOR PARA REGULADOR DE VOLTAJE. HEAT SINK FOR VOLTAGE REGULATOR.	REG1	1	M2-800	EGG MANE- FIELD ENGINEE- RING	0.24	0.24	D	ENE/86	1	14	141			
	PARTE PARA CIRCUITOS INTEGRADOS. INTEGRATED CIRCUITS DIP SOCKETS.														
CEL37A	BASE DE 40 PINES. DIP SOCKET 40 PINES.	94	1	240-40299	AVOANT	0.53	0.53	D	DIC/85	1	14	550			

TABLE 2.0

LISTA DE PARTES ELECTRONICAS												LA-1-11E	
A	B	C	D	E	F	G	H	I	J	K	L	M	
CEL300	BASE DE 20 PINES. 20 PINES DIP SOCKET.	60	1	220-00250	AGOST	0.20	0.20	D	ENE/06	1	14	550	
CEL390	BASE DE 24 PINES. 24 PINES DIP SOCKET.	67	1	224-00250	AGOST	0.22	0.22	D	ENE/06	1	14	550	
CEL400	BASE DE 10 PINES. 10 PINES DIP SOCKET.	A1, A2	2	210-00250	AGOST	0.24	0.40	D	DIC/05	1	14	550	
CEL410	BASE DE 16 PINES. 16 PINES DIP SOCKET.	A2, A3 A4, A5 A11	3	216-00250	AGOST	0.21	1.05	D	DIC/05	1	14	550	
CEL420	BASE DE 14 PINES. 14 PINES DIP SOCKET.	A10 A11	2	214-00250	AGOST	0.19	0.20	D	DIC/05	1	14	550	
CEL430	BASE WIDE HEAP DE 24 PINES. 24 PINES WIDE HEAP DIP SOCKET.	61	1	CR13004	TIENE INSTRUMENT	1.00	1.00	D	DIC/05	1	14	540	

TABLE 2.9

LISTA DE PARTES ELECTRONICAS		LM-1-11B											
A	B	C	D	E	F	G	H	I	J	K	L	M	
	SOCKETS												
CEL444	SOCKET DE 24 PINES. 24 PINES DIP SOCKET.	01	1	224-3344	TEXTOL						1		

## 2.4 DISEÑO MECANICO

En esta sección se describen las características mecánicas proporcionadas al equipo lector de memorias LM-1-IIE.

En el diseño de la envolvente mecánica se consideran los factores ambientales que influyen en la operación del equipo, por lo que la envolvente mecánica además de servir como soporte a la tableta de circuito impreso, protege a ésta de agentes externos que pueden reducir la confiabilidad en el equipo.

Para evitar al máximo estos problemas se cuenta con una cubierta de acrílico, la cual brinda protección contra el polvo principalmente a los conectores en los que deben insertarse las memorias. Esta cubierta mejora la presentación externa del equipo.

Las partes necesarias para el ensamble del equipo, son mínimas debido a que en la tableta de circuito impreso se encuentra la mayor parte de los componentes, quedando fuera de esta solamente el interruptor, el portafusible y el transformador.

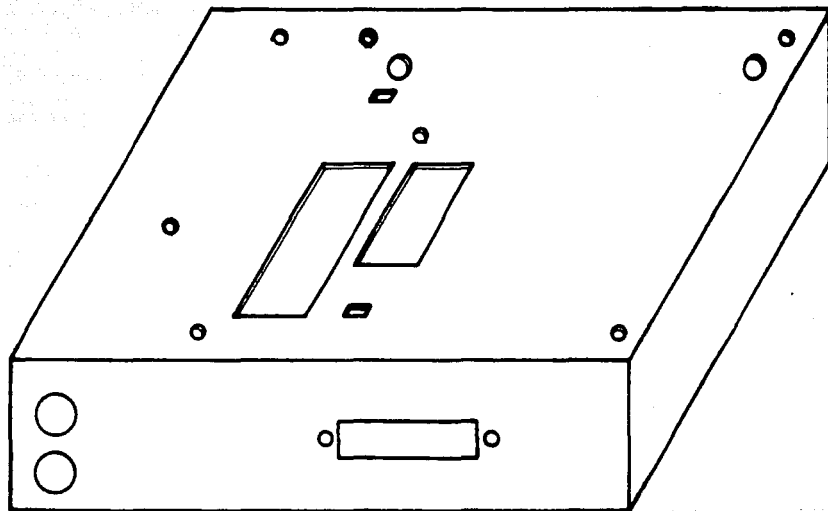


#### 2.4.1 DIAGRAMAS DE PARTES MECANICAS

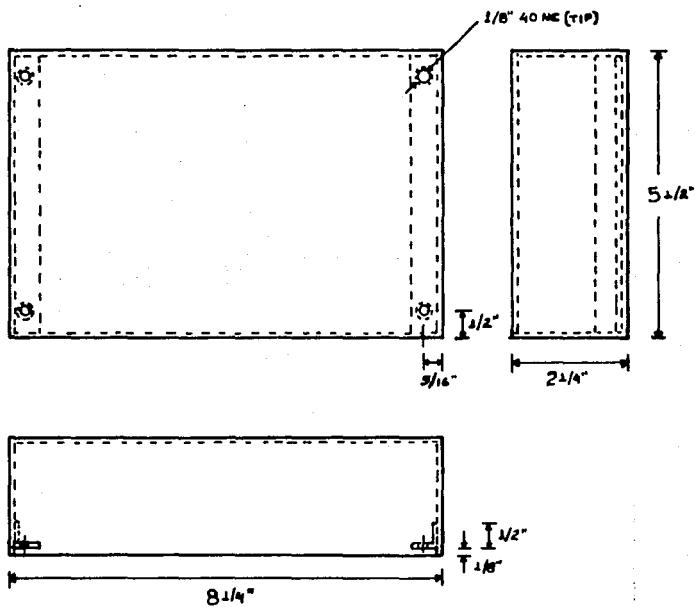
A continuación se muestran los diagramas de las partes mecánicas utilizadas en el ensamble del equipo.

El número de partes mecánicas necesarias para el ensamble del equipo, fué reducido al mínimo, con el propósito de que este sea realizado con la mayor facilidad.

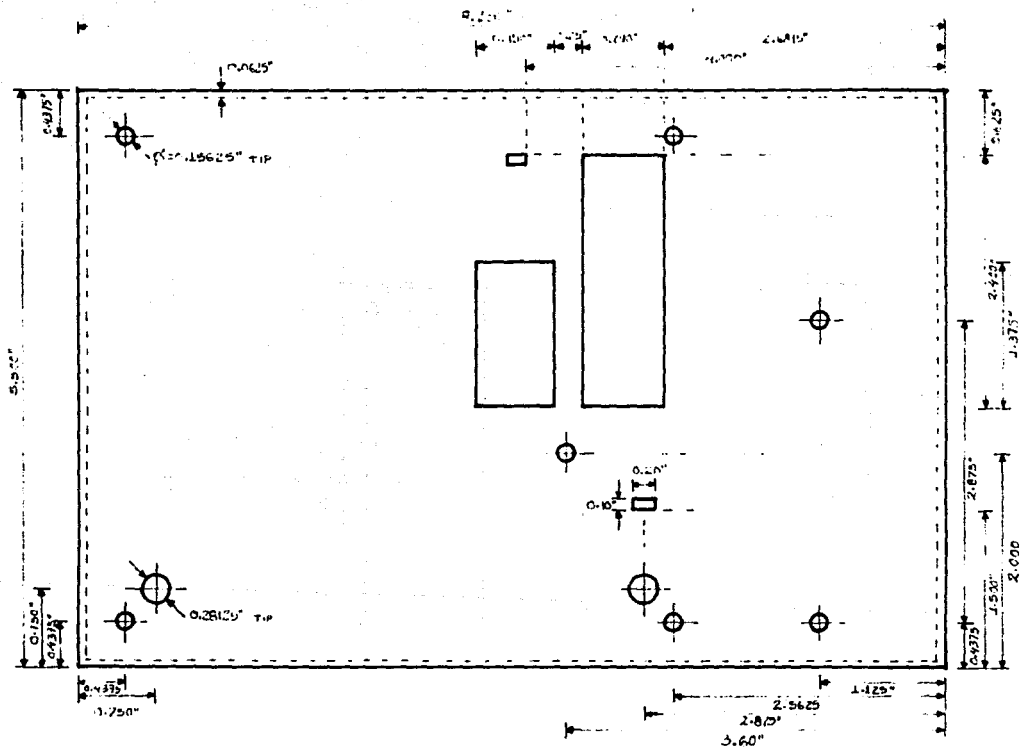
No se incluye el "lay out" del circuito debido a que este quedo implementado con "wire-wrap".



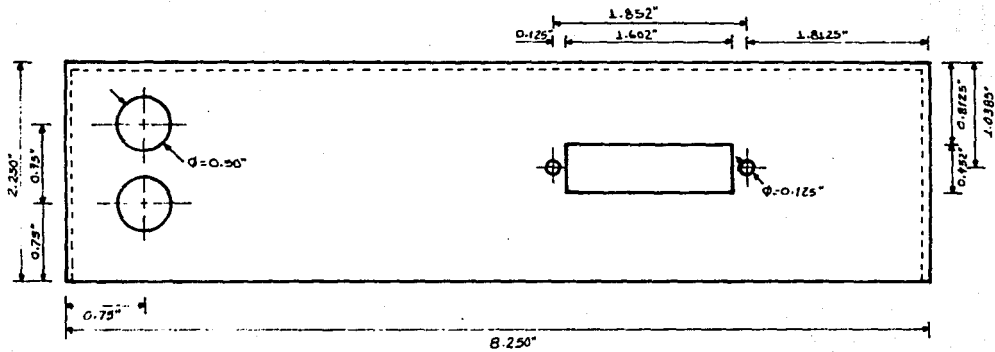
ENVOLVENTE MECANICA



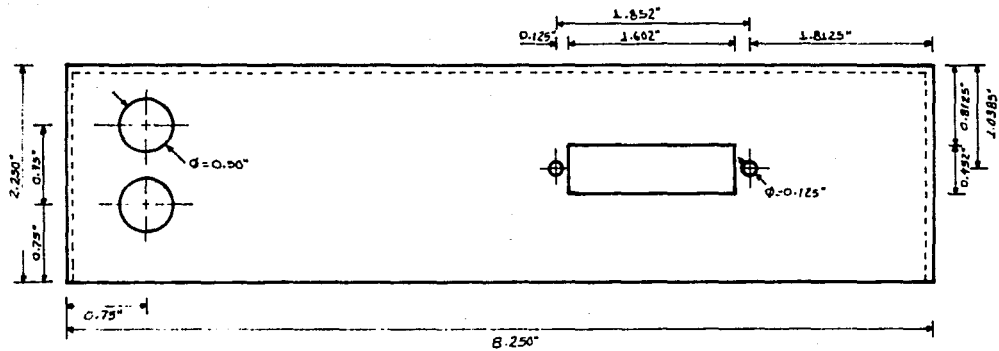
ENVOLVENTE MECANICA (DIMENSIONES)



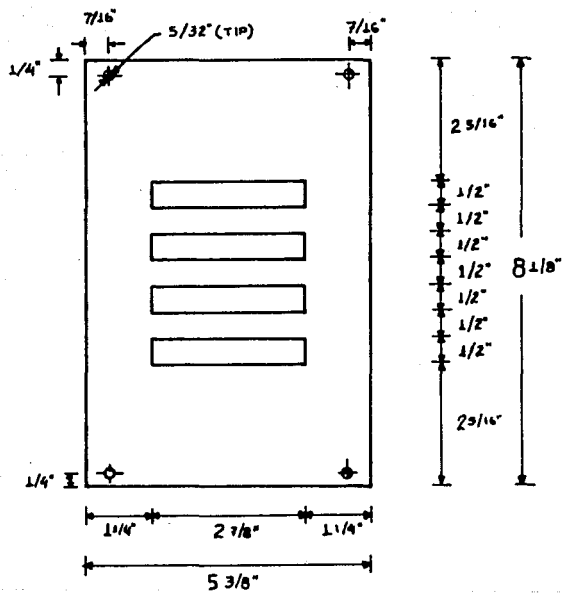
VISTA SUPERIOR DE LA ENVOLVENTE MECANICA



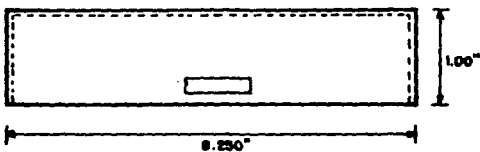
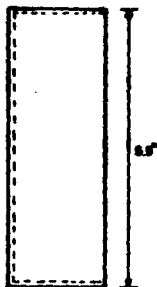
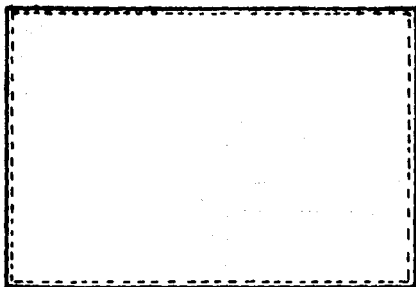
VISTA POSTERIOR DE LA ENVOLVENTE MECANICA



VISTA POSTERIOR DE LA ENVOLVENTE MECANICA

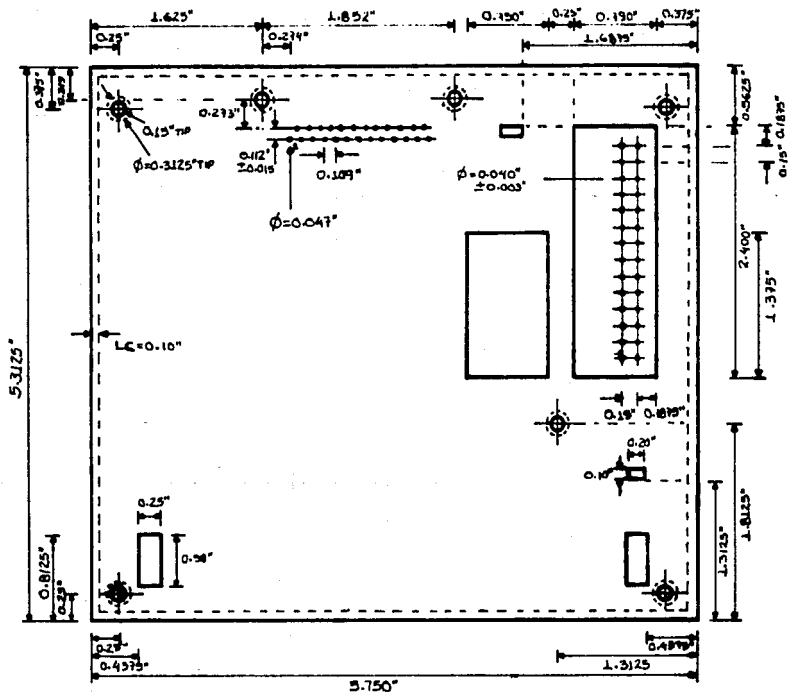


TAPA DE LA ENVOLVENTE MECANICA



CUBIERTA DE ACRILICO





DIMENSIONES DE LA TARJETA DE CIRCUITO IMPRESO

#### 2.4.2 LISTA DE PARTES MECANICAS

En las tablas 2.10 y 2.11 mostradas a continuación se presenta la información referente a las partes mecánicas del equipo.

Se incluyen los datos necesarios tanto para su localización en el plano, como para su adquisición o fabricación.

La información contenida en las listas descritas a continuación es:

- A - Número de parte.
- B - Descripción (Español - Inglés).
- C - Localización en el plano.
- D - Cantidad.
- E - Número de parte comercial.
- F - Fabricante.
- G - Precio unitario.
- H - Precio total.
- I - Tipo de moneda (P - Moneda Nacional. D- Dolares).
- J - Mes y año de cotización.
- K - Proveedor de acuerdo a la lista anexa.
- L - Catálogo de acuerdo a la lista anexa.
- M - Página de catálogo.

#### **2.4.2.1 LISTA DE PROVEEDORES**

- 1. KLEIGO SUPPLY, INC.  
2418 Treadslow Lane  
Houston, Texas 77067  
Phone (713) 893-4681  
Telex 790-550**

#### 2.4.2.2 LISTA DE CATALOGOS

14. NEWARK ELECTRONICS CATALOG  
4801 N. Ravenswood  
Chicago, Il 60640  
(312) 784-5100

TABLE 2.10

LISTA DE PARTES MECANICAS		LM-1-11E										
A	B	C	D	E	F	G	H	I	J	K	L	M
CUBIERTA	CAJA METALICA CON TAPA. METALIC BOX WITH COVER.		1	-	III	10000	10000	P	DIC/66	-	-	-
CUBIERTA	BIANCA. NINGO.		2									
CUBIERTA	CUBIERTA DE ACRILICO. ACRYLIC COVER.		1	-								
CUBIERTA	SEPARADORES METALICO- ALUMINIO 1/8"x1/4". SPACERS MET ALUMINUM 1/8"x1/4".		5	HTA-4-750	AMERLON BEAMS	0.16	0.10	D	ENE/66	1	14	500
CUBIERTA	TORNILLOS 1/8"x1/4". 1/8"x1/4" SCREWS.		10	HTA-4-750 HT-50	AMERLON BEAMS	0.04	0.40	D	ENE/66	1	14	500
CUBIERTA	TORNILLOS 1/8"x1/2". 1/8"x1/2" SCREWS.		9	HTA-4-750 HT-50	AMERLON BEAMS	0.05	0.45	D	ENE/66	1	14	500

TABLE 2.11

LISTA DE PARTES RECONOCIDAS													LN-1-11E	
A	B	C	D	E	F	G	H	I	J	K	L	M		
CHL079	ARRODELA METALICA FLANG. FLAT METAL WASHER.		19	NY-04-010 -04	ALUMIN BRASS	0.00	0.00	D	ENE/06	1	14	593		
CHL084	TORNILLO METALICO MEXMETAL. NUT METALIC NUT.		19	NY-0400- 04-10	ALUMIN BRASS	0.03	0.57	D	ENE/06	1	14	593		
CHL090	REPRESOR DE CABLE. CORD STRAIN RELIEF.		1	SKR-10	NYLON BRASS	0.00	0.00	D	DIC/05	1	14	604		

## **CAPITULO III**

### **SOFTWARE**

#### **PROGRAMACION DEL MICROPROCESADOR**

## CAPITULO III

### SOFTWARE

#### PROGRAMACION DEL MICROPROCESADOR

##### 3.1 INTRODUCCION

En el presente capítulo se presenta la descripción, de la parte de diseño del equipo lector de memorias, relacionada a la programación del microprocesador (software).

El lector de memorias LM-1-IIE funciona bajo un programa monitor denominado LEPROM1, el cual está contenido en una memoria de solo lectura, de capacidad igual a 2 Kbytes (EPROM 2716).

El programa monitor está compuesto por un programa principal y 15 subrutinas, realizadas en forma estructurada, para permitir la adición en forma fácil de nuevas subrutinas en futuras aplicaciones y dotar al lector de memorias con un programa eficiente y confiable.



### 3.2 DESCRIPCION DEL PROGRAMA

El desarrollo del software del equipo lector de memorias, comprende todas las actividades realizadas para obtener un programa de aplicación, mediante el cual sean ejecutadas las operaciones del equipo. A través de este programa, el microprocesador adquiere las instrucciones adecuadas, para realizar la lectura de los dispositivos de memoria tipos 2716 y 2532, y transmitir los datos obtenidos de estos a una microcomputadora.

Para realizar las operaciones involucradas en la lectura de memorias, se requiere que las instrucciones sean adquiridas con una secuencia determinada, de acuerdo a la operación a realizar. Como son el direccionamiento de la localidad de memoria, la lectura de la misma y transmisión del dato obtenido.

Las funciones realizadas por el equipo lector de memorias basicamente son las siguientes: desplegado del texto de presentación del equipo, desplegado del menú de las memorias que pueden ser leídas, lectura de memorias 2716 y lectura de memorias 2532. Cada una de estas es iniciada por un comando seleccionado en el programa bajo el cual funciona el equipo, en la figura 3.1 se muestra en forma esquemática las funciones realizadas, así como el comando por el cual son iniciadas

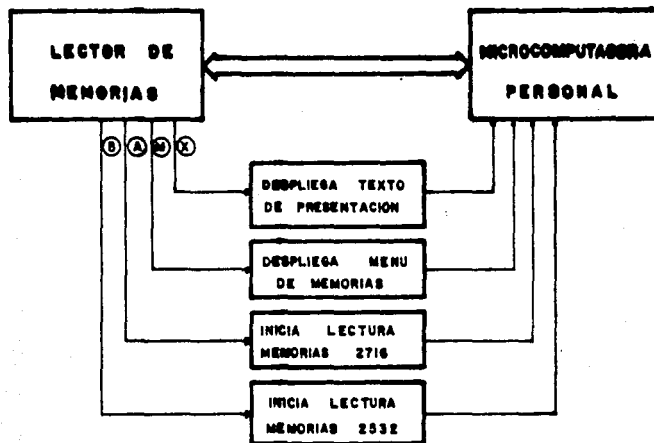


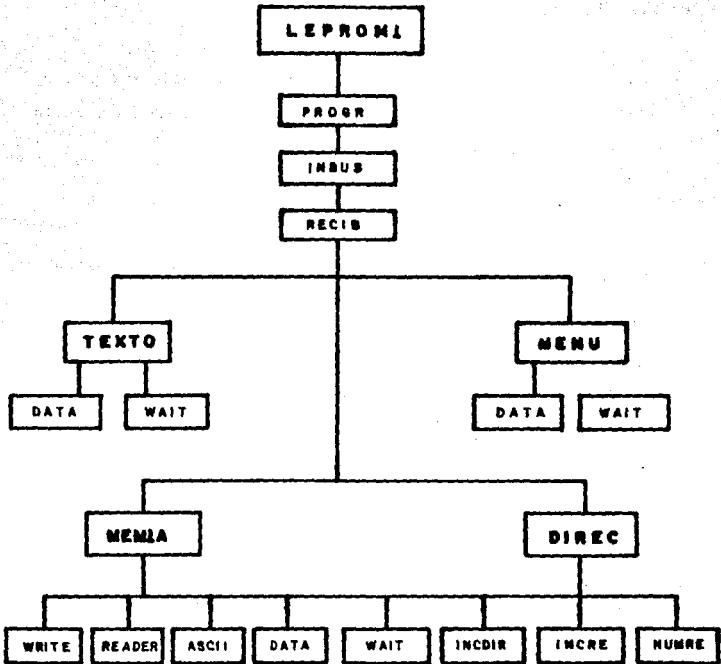
FIGURA 3.1. FUNCIONES REALIZADAS POR EL LM-1-11E.

estas. En esta figura se muestra a una microcomputadora, el lector de memorias y el canal de comunicación empleado para establecer el intercambio de información entre estos, en esta figura no se incluye el comando "C" dado que este se considera como un comando auxiliar en la lectura de los dispositivos de memoria; ya que este indica al lector realizar la lectura de un segundo bloque de datos, (cada bloque consta de 32 bytes de información).

El programa monitor está desarrollado en forma estructurada, lo cual permite al usuario mayores facilidades al realizar la codificación, correcciones y modificaciones que se hagan necesarias. En la figura 3.2 se muestra en forma esquemática la estructura jerárquica del programa, en éste se observa en forma gráfica la relación de cada subrutina con las operaciones realizadas por el equipo lector de memorias.

A continuación se presenta una descripción de las operaciones realizadas por el programa bajo el cual funciona el equipo lector de memorias, esto en forma general, ya que se consideran las principales funciones realizadas sin mencionar las subrutinas involucradas en la ejecución de estas.

A continuación se hace una descripción del diagrama de flujo del programa principal, de este programa, el cual se muestra en la figura 3.3:



**FIGURA 3.2 ESTRUCTURA JERARQUICA DEL PROGRAMA.**

Al aplicar energía eléctrica al equipo, el microprocesador realiza las siguientes operaciones:

- La lectura de memorias es deshabilitada con el propósito, de realizar las operaciones necesarias antes de iniciar la lectura de las memorias.
- Es habilitada la interfaz de comunicación, la cual se programa con el formato y forma de comunicación seleccionados para la transmisión y recepción de señales.
- Se realiza una lectura inválida del bus del microprocesador, para evitar que la interfaz de comunicación contenga datos extraños antes de recibir los comandos.
- Inicializa los registros R0 y R7, empleados para direccionar la memoria a leer.
- Habilita la recepción de comandos, con los cuales son inicializadas las operaciones del equipo lector de memorias.

### 3.2.1 COMANDOS SELECCIONADOS

Quando se recibe un comando, se verifica que corresponda a uno de los que han sido programados para ser reconocidos por el microprocesador e iniciar las operaciones del lector de memorias, si el comando recibido no corresponde a ninguno de los comandos programados, este es desechado y el microprocesador queda en espera de un nuevo comando.

Si el comando recibido es reconocido por el microprocesador, efectuará la operación correspondiente y al término de la misma, el equipo queda en espera de un nuevo comando.

Los comandos que reconoce el microprocesador del LM-1-IIE, así como la función inicializada por cada uno se describen a continuación:

- COMANDO "X"

Inicia el despliegado del texto de presentación del equipo.

INSTITUTO DE INVESTIGACIONES ELECTRICAS

LECTOR DE MEMORIAS

- COMANDO "M"

Inicia el desplegado del menú de las memorias que pueden ser leídas con el equipo.

2716=A

2532=B

- COMANDO "A"

Inicia la lectura de las memorias tipo 2716.

- COMANDO "B"

Inicia la lectura de las memorias tipo 2532.

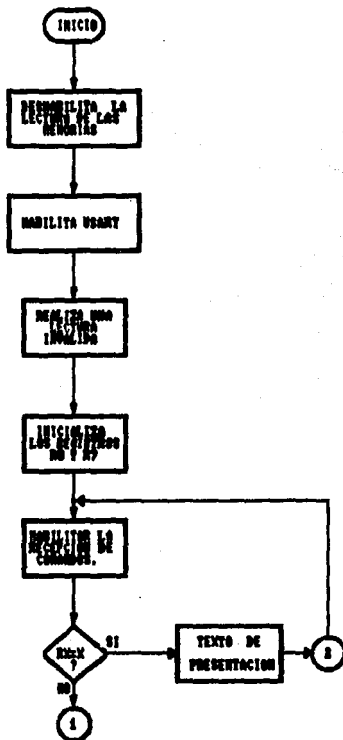
- COMANDO "C"

Indica al equipo que debe continuar con la lectura de las memorias la cual efectua por bloques, (32 bytes/bloque).

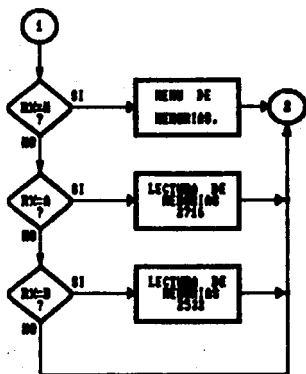
### 3.3 DIAGRAMAS DE FLUJO Y CODIFICACION

A continuación se muestran los diagramas de flujo correspondientes a cada subrutina, acompañados de los listados correspondientes, codificados en lenguaje ensamblador. En el anexo II se presenta una descripción de la hoja de instrucciones del microprocesador utilizado en el diseño del Lector de memorias.

PROGRAMA PRINCIPAL







"8048"

TITLE "ELI20"

NAME "LECTOR"

; PROGRAMA PRINCIPAL

;EN ESTE PROGRAMA SE SELECCIONA LA FUNCION QUE DEBE SER  
;REALIZADA POR EL LECTOR DE MEMORIAS.

;LAS FUNCIONES QUE SE PUEDEN SELECCIONAR SON: DESPLEGADO

;DEL TEXTO DE PRESENTACION, MENU DE LAS MEMORIAS QUE SE

;PUEDEN LEER, E INICIAR LA LECTURA DE LA MEMORIA

;SELECCIONADA. PARA INICIAR ESTAS OPERACIONES EL LECTOR

;DEBE RECIBIR EL COMANDO QUE LE INDIQUE EL INICIO DE UNA

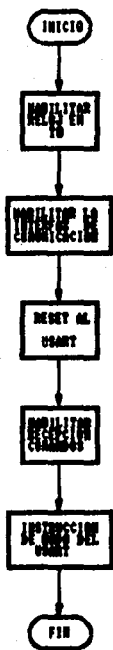
;DE LAS POSIBLES OPERACIONES, A LO QUE DEBE RESPONDER CON

;EL INICIO DE LA OPERACION SELECCIONADA.

EXT           PROGR,TEXTO,RECIB,MENU,DIREC,WRITE,  
              MEMIA,INBUS

LECTOR	PROG		
	MOV	A, 0C9H	DESHABILITA LECT. DE MEMORIA
	OUTL	P1,A	
	CALL	WRITE	
	CALL	PROGR	HABILITA USART
	ORL	P2, 0A0H	P25=1 CONTROL
	MOV	A, 016H	CARGA COMANDOS RECEPCION
	OUTL	BUS,A	TRANSFIERE COMANDOS
	CALL	INBUS	REALIZA LECTURA INVALIDA
ESP	MOV	R0, 000H	LIMPIA R0 Y R7
	MOV	R7, 000H	
	CALL	RECIB	RECEPCION DE COMANDOS
	MOV	R2,A	
	XRL	A, 058H	XRL A Y X=58H
	JNZ	COMP	A DIF X
	CALL	TEXTO	COMANDO RECIBIDO FUE "X"
	JMP	ESP	DESP. TEXTO DE PRESENTACION
COMP	MOV	A,R2	
	XRL	A, 04DH	COMANDO RECIBIDO "M" MENU
	JNZ	LEM1	MEMORIAS QUE SE PUEDEN LEER
	CALL	MENU	
	JMP	ESP	
LEM1	MOV	A,R2	
	XRL	A, 041H	A=41H (2716)
	JNZ	MEM	
	CALL	MEMIA	INIC. LECT. DE MEMORIA 2716
	JMP	ESP	
MEM	MOV	A,R2	
	XRL	A, 042H	B=42H (2532)
	JNZ	ESP	
	CALL	DIREC	INIC. LECT. DE MEMORIA 2532
	JMP	ESP	

# SUBROUTINA PROCES

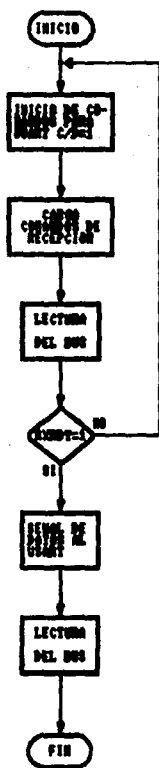


"8048"  
TITLE "ELI14"  
NAME "PROGR"

;SUBROUTINA PROGR  
;ESTA SUBROUTINA HABILITA LA INTERFAZ DE COMUNICACION.  
;AQUI SE SELECCIONA EL FORMATO CON EL CUAL VAN A SER  
;TRANSMITIDOS LOS DATOS POR EL PUERTO DE SALIDA SERIE,  
;ESTO SE REALIZA CON LA INSTRUCCION DE MODO. LA  
;HABILITACION DE LA INTERFAZ Y SU PROGRAMACION SE  
;REALIZAN A TRAVES DE SEÑALES ENVIADAS A TRAVES DEL  
;PUERTO 2 DEL MICROPROCESADOR.

	GLB	PROGR	
PROGR	ENT0	CLK	HABILITA RELOJ EN T0
	ANL	P2, 087H	HABILITA USART P24=0
	ORL	P2, 0C0H	RESET DEL USART P26=1
	ANL	P2, 087H	LIMPIA PUERTO 2
	ORL	P2, 0A0H	COLOCA P25=1 CONTROL
	MOV	A, 04EH	CARGA INST DE MODO EN A.
	OUTL	BUS,A	TRANSFIERE INST. DE MODO
	RET		

# SUBROUTINA RECIB

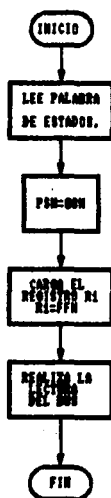


```

"8048"
TITLE "ELI19"
NAME "RECIB"
;SUBROUTINA RECIB
;ESTA SUBROUTINA PRUEBA LAS CONDICIONES PARA RECIBIR UN DATO
;(COMANDO), CON EL CUAL INICIAR LAS OPERACIONES DEL EQUIPO
;SI ESTAS SE CUMPLEN, SE REALIZA LA RECEPCION DEL COMANDO
;ADEMAS SE INDICA EL FORMATO CON EL CUAL DEBEN SER
;RECIBIDOS LOS DATOS.
      EXT      INBUS
      GLB      RECIB
      PROG
RECIB  ORL      P2, 0A0H          P25=1 CONTROL
      MOV      A, 016H          COMANDOS DE RECEPCION
      OUTL     BUS,A           TRANSFIERE COMANDOS
      CALL     INBUS
      ANL      A, 002H          PRUEBA ESTADO DE RXRDY
      JZ       RECIB           RXRDY=0
      ANL      P2, 087H        P25=0 DATOS
      CALL     INBUS
      RET

```

**SUBROUTINA INBUS**



**ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**

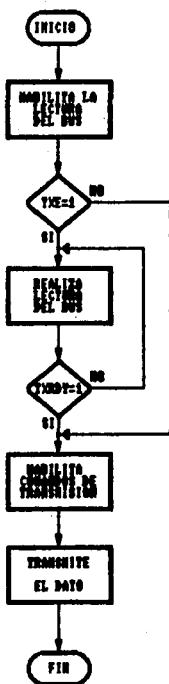
"8048"  
TITLE "ELI31"  
NAME "INBUS"

;SUBROUTINA INBUS  
;ESTA SUBROUTINA REALIZA LAS OPERACIONES NECESARIAS PARA  
;REALIZAR LA LECTURA DEL BUS DEL MICROPROCESADOR.

	GLB	INBUS
	PROG	
INBUS	MOV	A,PSW
	ORL	A,008H
	MOV	PSW,A
	MOV	R1,0FFH
	MOVX	A,R1
	INS	A,BUS
	RET	



SUBROUTINA DATA

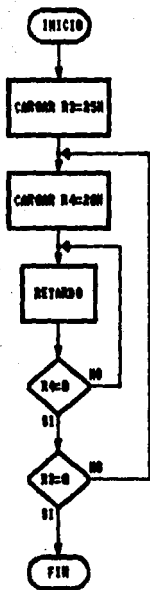


"8048"  
 TITLE "ELI17"  
 NAME "DATA"

;SUBROUTINA DATA  
 ;ESTA SUBROUTINA REALIZA LA PRUEBA DE LAS CONDICIONES  
 ;NECESARIAS PARA EFECTUAR LA TRANSMISION DE UN DATO,  
 ;CUANDO ESTAS SON ADECUADAS REALIZA LA TRANSMISION.  
 ;LA PRUEBA DE LAS CONDICIONES SE REALIZA POR MEDIO DE  
 ;LECTURA DE BANDERAS DE ESTADOS DE LA INTERFAZ DE  
 ;COMUNICACION.  
 ;EL DATO A TRASMITIR ES EL ALMACENADO EN EL REGISTRO R5

	EXT	INBUS	
	GLB	DATA	
	PROG		
DATA	ORL	P2, 0A0H	P25=1 CONTROL
	CALL	INBUS	
	ANL	A, 004H	PRUEBA ESTADO DE TXE
	JZ	SUB	
LIST	ORL	P2, 0AH	P25=1 CONTROL
	CALL	INBUS	
	ANL	A, 001H	PRUEBA ESTADO DE TXRDY
	JZ	LIST	TXRDY=0
SUB	ORL	P2, 0A0H	P25=1 CONTROL
	MOV	A, 013H	INST. DE COMANDOS EN A.
	OUTL	BUS,A	TRANSFIERE INST DE COMANDOS
	ANL	P2, 007H	P25=0 DATOS
	MOV	A,R5	A=R5
	OUTL	BUS,A	TRANSMITE DATO
	RET		

SUBSTRING UNIT



"8048"  
TITLE "ELI16"  
NAME "WAIT"

; SUBROUTINA WAIT  
;ESTA SUBROUTINA OCASIONA UN RETARDO, EL CUAL ES EMPLEADO  
;PARA LA TRANSMISION ENTRE CADA UNO DE LOS DATOS. LA RAZON  
;DE EMPLEAR ESTE ES PARA NO SATURAR EL BUFFER DE LA  
;MICROCOMPUTADORA AL MOMENTO DE ALMACENAR LOS DATOS EN  
;UN DISKETTE.

	GLB	WAIT	
	PROG		
WAIT	MOV	R3, 025H	R3=25H
TRES	MOV	R4, 020H	R4=20H
FALT	NOP		
	DJNZ	R4, FALT	DECREMENTA R4
	DJNZ	R3, TRES	DECREMENTA R3
	RET		

### SUBRUTINA TEXTO



```

"8048"
TITLE      "ELI21"
NAME      "TEXTO"
;SUBROUTINA TEXTO
;ESTA SUBROUTINA DESPLIEGA EL TEXTO DE PRESENTACION DE EL
;EQUIPO, CUANDO RECIBE UNA "X" COMO COMANDO DE INICIO.
;ESTA SUBROUTINA HACE USO DE LAS SUBROUTINAS DE TRANSMISION
;DE DATOS Y DEL RETARDO EMPLEADO DURANTE LA TRANSMISION DE
;LOS MISMOS.
GLB      TEXTO
EXT      DATA, WAIT
PROG
TEXT0    MOV      R5, 00AH      AVANCE DE LINEA
          CALL     DATA
          CALL     WAIT
          MOV      R5, 00DH      RET
          CALL     DATA
          CALL     WAIT
          MOV      R6, 014H
          MOV      R5, 020H      ESP
REP      CALL     DATA
          CALL     WAIT
          DJNZ     R6, REP
          MOV      R5, 049H      I
          CALL     DATA
          CALL     WAIT
          MOV      R5, 04EH      N
          CALL     DATA
          CALL     WAIT
          MOV      R5, 053H      S
          CALL     DATA
          CALL     WAIT
          MOV      R5, 054H      T
          CALL     DATA
          CALL     WAIT
          MOV      R5, 049H      I
          CALL     DATA
          CALL     WAIT
          MOV      R5, 054H      T
          CALL     DATA
          CALL     WAIT
          MOV      R5, 055H      U
          CALL     DATA
          CALL     WAIT
          MOV      R5, 054H      T
          CALL     DATA
          CALL     WAIT
          MOV      R5, 04FH      O
          CALL     DATA
          CALL     WAIT
          MOV      R5, 020H      ESP

```

CALL	DATA	
CALL	WAIT	
MOV	R5, 044H	D
CALL	DATA	
CALL	WAIT	
MOV	R5, 045H	E
CALL	DATA	
CALL	WAIT	
MOV	R5, 020H	ESP
CALL	DATA	
CALL	WAIT	
MOV	R5, 049H	I
CALL	DATA	
CALL	WAIT	
MOV	R5, 04EH	N
CALL	DATA	
CALL	WAIT	
MOV	R5, 056H	V
CALL	DATA	
CALL	WAIT	
MOV	R5, 045H	E
CALL	DATA	
CALL	WAIT	
MOV	R5, 053H	S
CALL	DATA	
CALL	WAIT	
MOV	R5, 054H	T
CALL	DATA	
CALL	WAIT	
MOV	R5, 049H	I
CALL	DATA	
CALL	WAIT	
MOV	R5, 047H	G
CALL	DATA	
CALL	WAIT	
MOV	R5, 041H	A
CALL	DATA	
CALL	WAIT	
MOV	R5, 043H	C
CALL	DATA	
CALL	WAIT	
MOV	R5, 049H	I
CALL	DATA	
CALL	WAIT	
MOV	R5, 04FH	O
CALL	DATA	
CALL	WAIT	
MOV	R5, 04EH	N
CALL	DATA	
CALL	WAIT	
MOV	R5, 045H	E
CALL	DATA	

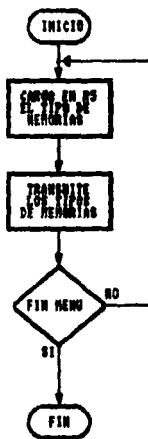
CALL	WAIT	
MOV	R5, 053H	S
CALL	DATA	
CALL	WAIT	
MOV	R5, 020H	ESP
CALL	DATA	
CALL	WAIT	
MOV	R5, 045H	E
CALL	DATA	
CALL	WAIT	
MOV	R5, 04CH	L
CALL	DATA	
CALL	WAIT	
MOV	R5, 045H	E
CALL	DATA	
CALL	WAIT	
MOV	R5, 043H	C
CALL	DATA	
CALL	WAIT	
MOV	R5, 054H	T
CALL	DATA	
CALL	WAIT	
MOV	R5, 052H	R
CALL	DATA	
CALL	WAIT	
MOV	R5, 049H	I
CALL	DATA	
CALL	WAIT	
MOV	R5, 043H	C
CALL	DATA	
CALL	WAIT	
MOV	R5, 041H	A
CALL	DATA	
CALL	WAIT	
MOV	R5, 053H	S
CALL	DATA	
CALL	WAIT	
MOV	R5, 00AH	AVANCE DE LINEA
CALL	DATA	
CALL	WAIT	
MOV	R5, 00DH	RET
CALL	DATA	
CALL	WAIT	
MOV	R6, 01EH	
MOV	R5, 020H	ESP
CALL	DATA	
CALL	WAIT	
DJNZ	R6, REPl	
MOV	R5, 04CH	L
CALL	DATA	
CALL	WAIT	
MOV	R5, 045H	E



CALL	DATA	
CALL	WAIT	
MOV	R5, 043H	C
CALL	DATA	
CALL	WAIT	
MOV	R5, 054H	T
CALL	DATA	
CALL	WAIT	
MOV	R5, 04FH	O
CALL	DATA	
CALL	WAIT	
MOV	R5, 052H	R
CALL	DATA	
CALL	WAIT	
MOV	R5, 020H	ESP
CALL	DATA	
CALL	WAIT	
MOV	R5, 044H	D
CALL	DATA	
CALL	WAIT	
MOV	R5, 045H	E
CALL	DATA	
CALL	WAIT	
MOV	R5, 020H	ESP
CALL	DATA	
CALL	WAIT	
MOV	R5, 04DH	M
CALL	DATA	
CALL	WAIT	
MOV	R5, 045H	E
CALL	DATA	
CALL	WAIT	
MOV	R5, 04DH	M
CALL	DATA	
CALL	WAIT	
MOV	R5, 04FH	O
CALL	DATA	
CALL	WAIT	
MOV	R5, 052H	R
CALL	DATA	
CALL	WAIT	
MOV	R5, 049H	I
CALL	DATA	
CALL	WAIT	
MOV	R5, 041H	A
CALL	DATA	
CALL	WAIT	
MOV	R5, 053H	S
CALL	DATA	
CALL	WAIT	
MOV	R5, 00AH	AVANCE DE LINEA
CALL	DATA	

CALL	WAIT	
MOV	R5, 00AH	AVANCE DE LINEA
CALL	DATA	
CALL	WAIT	
MOV	R5, 00DH	RET
CALL	DATA	
CALL	WAIT	
RET		

SUBROUTINA MENU



```

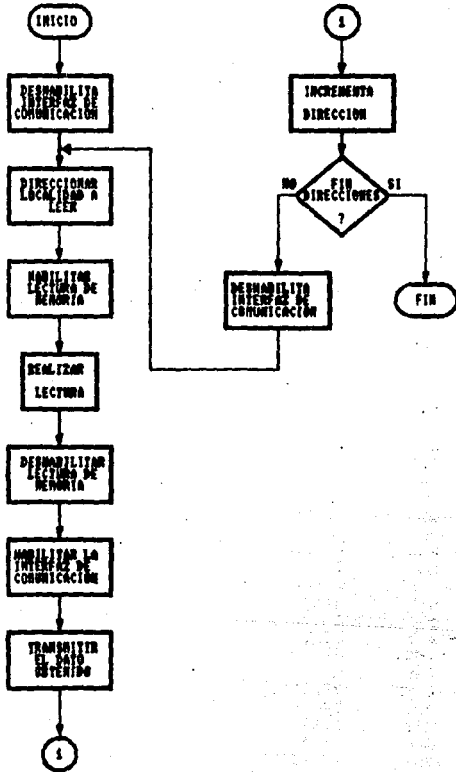
"8048"
TITLE "ELI22"
NAME "MENU"
;SUBROUTINA MENU
;CON ESTA SUBROUTINA SE REALIZA UN DESPLEGADO DEL MENU DE
;MEMORIAS QUE SE PUEDEN LEER, ASI COMO DEL COMANDO QUE CADA
;UNA REQUIERE PARA INICIAR LA LECTURA DE LA MEMORIA
;SELECCIONADA.
;ESTA ES INICIADA CUANDO EL LECTOR RECIBE EL COMANDO "M"

```

	GLB	MENU	
	EXT	DATA, WAIT	
	PROG		
MENU	MOV	R5, 00AH	AVANCE DE LINEA
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 00DH	RET
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 032H	2
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 037H	7
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 031H	1
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 036H	6
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 03DH	-
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 041H	A
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 00AH	AVANCE DE LINEA
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 00DH	RET
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 032H	2
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 035H	5
	CALL	DATA	
	CALL	WAIT	
	MOV	R5, 033H	3
	CALL	DATA	
	CALL	WAIT	

MOV	R5, 032H	2
CALL	DATA	
CALL	WAIT	
MOV	R5, 03DH	-
CALL	DATA	
CALL	WAIT	
MOV	R5, 042H	B
CALL	DATA	
CALL	WAIT	
MOV	R5, 00AH	AVANCE DE LINEA
CALL	DATA	
CALL	WAIT	
MOV	R5, 00DH	RET
CALL	DATA	
CALL	WAIT	
MOV	R5, 03FH	?
CALL	DATA	
CALL	WAIT	
MOV	R5, 00AH	AVANCE DE LINEA
CALL	DATA	
CALL	WAIT	
MOV	R5, 00DH	RETR
CALL	DATA	
CALL	WAIT	
RET		

SUBROUTINA MEMIA



```

"8048"
TITLE "ELI28"
NAME "MEMIA"
;
SUBROUTINA MEMIA
;SE INICIA CUANDO EL COMANDO RECIBIDO ES "A"
;ESTA SUBROUTINA REALIZA LA LECTURA DE MEMORIAS TIPO 2716, LOS
;DATOS QUE SE OBTIENEN SE TRANSMITEN EN BLOQUES DE 32 BYTES
;ESTA SUBROUTINA UTILIZA OTRAS CON LAS CUALES SE REGISTRA EL
;NUMERO DE LOCALIDADES DIRECCIONADAS, INCREMENTA EL NUMERO DE
;LA LOCALIDAD A LEER, ADQUIERE EL DATO DE LA LOCALIDAD
;DIRECCIONADA, LO CONVIERTE A UN FORMATO EN CODIGO ASCII Y
;TRANSMITE ESTE HACIA LA MICROCOMPUTADORA.
      GLS      MEMIA
      EXT      WRITE, WAIT, INCDIR, READER, ASCII, DATA
            INCR, NUMRE
BITS     EQU      008H
DIR      EQU      000H
SELA1    EQU      050H
SELA2    EQU      0C0H
BIT3     EQU      003H
            PROG
MEMIA    ORL      R2, 090H      DESHABILITA USART
         SEL      R1
         MOV      R0, 010H     INICIALIZA R0 Y R2 COMO
         MOV      R2, 020H     CONTADORES
         SEL      R0
LIMPIA   MOV      A, R0        CARGA R0 EN R6 PARA SALVARLO
         MOV      R6, A        AL FINAL DEL PROGRAMA
         CLR      F0          LIMPIA BANDERA F0
         MOV      R3, BITS     LAS SIGUIENTES INSTRUCCIONES
         MOV      R2, DIR      COLOCAN LA DIRECCION DE LA
         MOV      A, R0        LOCALIDAD QUE SE VA A LEER
         ANL      A, 001H
         ORL      A, R2
         JPO     SELA2
         ORL      A, SELA1
         JMP     SELA1
SELA2    ORL      A, SELA2
SELA1    OUTL    P1, A
         CALL   WRITE
         MOV      A, R0
         RR      A
         MOV      R0, A
         INC     R2
         INC     R2
         DJNZ   R3, BIT31
         JPO     FUERA
         MOV      A, R7
         MOV      R0, A
         MOV      R3, BIT3
         CPL     F0

```

FUERA      JMP            BIT31  
           MOV            A, 0C6H  
           OUTL          P1,A  
           CALL          WRITE  
           MOV            A, 0CAH  
           OUTL          P1,A  
           CALL          WRITE  
           CALL          READER  
           MOV            A, 0CBH  
           OUTL          P1,A  
           CALL          WRITE  
           MOV            A,R6  
           MOV            R0,A  
           ANL            P2, 0EFH  
           CALL          ASCII  
           CALL          WAIT  
           MOV            R5, 020H  
           CALL          DATA  
           CALL          WAIT  
           CALL          INCDIR  
           MOV            A,R7  
           XRL            A, 008H  
           JZ             META  
           CALL          INCRE  
           CALL          NUMRE  
           ORL            P2, 090H  
           JMP            LIMPIA  
 META      RET

HAB. LECT. DE MEMORIA E/P

G=0

LEE LOCALIDAD DIRECCIONADA  
 DESHABILITA LECTURA G=1

RECUPERA RO ORIGINAL

HABILITA USART

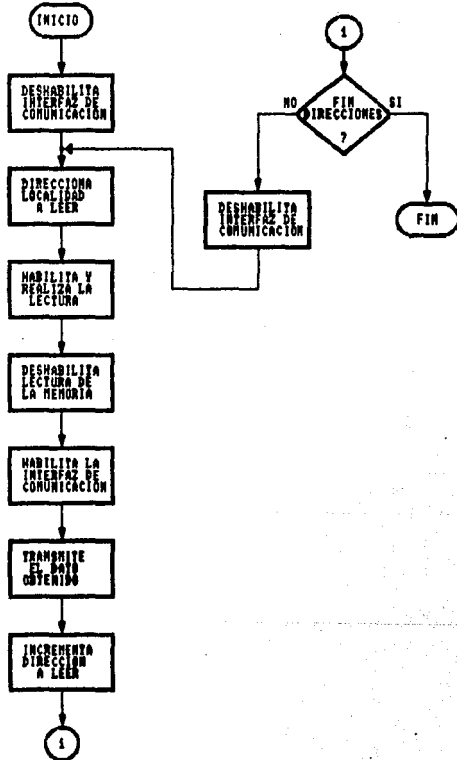
ESP

PREGUNTA FIN DIRECCIONES  
 SI FIN TERMINA LECTURA

DESHABILITA USART



SUBROUTINA DIREC



```

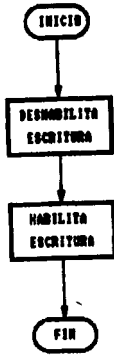
"8048"
TITLE "ELI25"
NAME "DIREC"
;
SUBROUTINA DIREC
;SE INICIA CUANDO SE RECIBE EL COMANDO "B"
;ESTA SUBROUTINA REALIZA LA LECTURA DE MEMORIAS TIPO 2532
;LOS DATOS ADQUIRIDOS SON TRANSMITIDOS EN BLOQUES DE 32 BYTES
;EL CONTROL DE LAS LOCALIDADES A LEER SE REALIZA POR MEDIO
;DE LA SUBROUTINA INCDIR, SON UTILIZADAS OTRAS CON LAS CUALES
;SE CONTROLA EL FORMATO DE TRANSMISION DE LOS DATOS RECUPERA-
;DOS, Y LA CONVERSION DE LOS MISMOS A FORMATO ASCII PARA SER
;TRANSMITIDOS HACIA LA MICROCOMPUTADORA.

```

	GLB	DIREC	
	EXT	WRITE, WAIT, INCDIR, READER, ASCII, DATA,	
		INCRE, NUMRE	
BIT8	EQU	008H	
DIR	EQU	000H	
SELA1	EQU	050H	
SELA2	EQU	0C0H	
BIT4	EQU	004H	
	PROG		
DIREC	ORL	P2, 090H	DESHABILITA USART
	SEL	RB1	
	MOV	R0, 010H	INICIALIZA R0 Y R2 COMO
	MOV	R2, 020H	CONTADORES
	SEL	RB0	
LIMPIA	MOV	A, R0	SALVA R0 ORIGINAL
	MOV	R6, A	CARGANDOLO EN R6
	CLR	F0	LIMPIA BANDERA F0
	MOV	R3, BIT8	LAS SIGUIENTES INSTRUCCIONES
BIT31	MOV	R2, DIR	COLOCAN LA DIRECCION DE LA
BIT81	MOV	A, R0	LOCALIDAD QUE SE VA A LEER
	ANL	A, 001H	
	ORL	A, R2	
	JF0	SELA2	
	ORL	A, SELA1	
	JMP	SELA1	
SELA2	ORL	A, SELA2	
SELA1	OUTL	P1, A	
	CALL	WRITE	
	MOV	A, R0	
	RR	A	
	MOV	R0, A	
	INC	R2	
	INC	R2	
	DJNZ	R3, BIT81	
	JF0	FUERA	
	MOV	A, R7	
	MOV	R0, A	
	MOV	R3, BIT4	

	CPL	FO	
	JMP	BIT31	
FUERA	MOV	A, 0CAH	PERMITE LECTURA DE MEMORIA
	OUTL	P1,A	
	CALL	WRITE	LEE LOCALIDAD DIRECCIONADA
	CALL	READER	INHIBE LECTURA DE MEMORIA
	MOV	A, 0CBH	
	OUTL	P1,A	
	CALL	WRITE	RECUPERA R0 ORIGINAL
	MOV	A,R6	
	MOV	R0,A	HABILITA USART
	ANL	P2, 0EFH	
	CALL	ASCII	ESP
	CALL	WAIT	
	MOV	R5, 020H	
	CALL	DATA	
	CALL	WAIT	
	CALL	INCDIR	
	MOV	A,R7	
	XRL	A, 010H	PREGUNTA POR ULTIMA DIRECC. DE MEMORIA
	JZ	META	
	CALL	INCRE	
	CALL	NUMRE	
	ORL	P2, 090H	DESHABILITA USART
	JMP	LIMPIA	
META	RET		

# SUBROUTINA MBIYE



"8048"  
TITLE "ELI24"  
NAME "WRITE"

;SUBROUTINA WRITE  
;ESTA SUBROUTINA HABILITA LA ESCRITURA EN CADA UNA DE LAS  
;LINEAS EMPLEADAS PARA DIRECCIONAR CUALQUIER LOCALIDAD A LEER  
;LAS SEÑALES SON TRANSMITIDAS A TRAVES DEL PUERTO 2.

	GLB	WRITE		
WRITE	PROG		WD=0	ESC, DESHABILITADA
	ANL	P2, 07FH	WD=1	ESC. HABILITADA
	ORL	P2, 080H		
	RET			

SUBROUTINA READEN



"8048"

TITLE "ELI27"

NAME "READER"

;SUBROUTINA READER

;ESTA SUBROUTINA REALIZA LA LECTURA DE LA LOCALIDAD

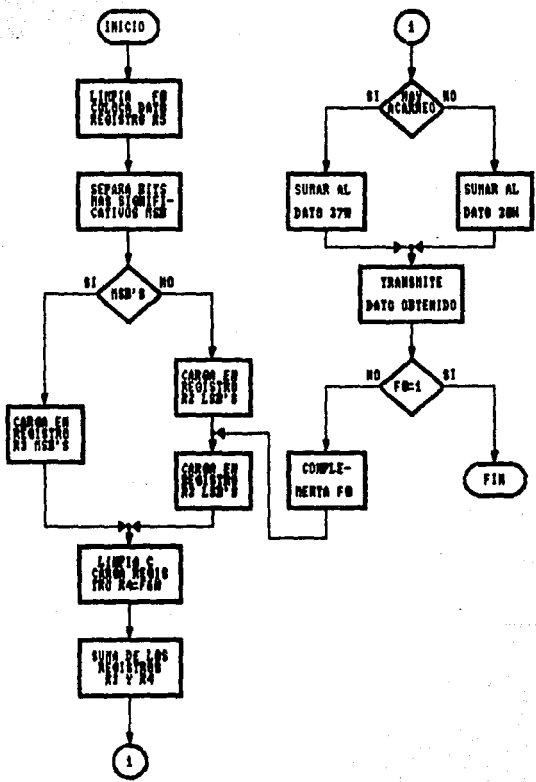
;DIRECCIONADA, BIT POR BIT Y EL DATO OBTENIDO LO ALMACENA EN

;EL REGISTRO R5, DE DONDE SE OBTIENE PARA SER COVERTIDO A

;CODIGO ASCII Y TRANSMITIRLO A LA MICROCOMPUTADORA.

	GLB	READER	
	PROG		
READER	MOV	R4, 008H	INIC. R4 COMO CONTADOR DE 8
	MOV	R5, 000H	
	MOV	R2, 041H	SELEC. A3 PARA LECTURA DE
CONT1	MOV	A, R2	LOCALIDAD DIRECCIONADA
	OUTL	P1, A	
	ANL	P1, 00FH	HABILITA LECTURA
	NOP		
	IN	A, P1	LEE BIT DIRECCIONADO
	ANL	A, 001H	
	ADD	A, R5	
	RR	A	
	MOV	R5, A	CARGA DATO LEIDO EN R5
	INC	R2	
	INC	R2	
	DJNZ	R4, CONT1	
SALE	RET		

SUBROUTINA ABC11





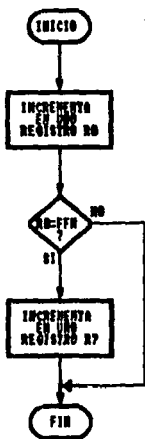
"8048"  
 TITLE "ELI32"  
 NAME "ASCII"

;SUBROUTINA ASCII

;ESTA SUBROUTINA REALIZA LA CONVERSION DE LOS DATOS OBTENIDOS  
 ;EN BINARIO ALMACENADOS EN EL REGISTRO R5 A CARACTERES  
 ;HEXADECIMALES (ASCII) Y REALIZA LA TRANSMISION DE LOS MISMOS.  
 ;LA CONVERSION SE REALIZA DIVIDIENDO EL BYTE EN NIBLES, LOS  
 ;CUALES SON CONVERTIDOS UNO A LA VEZ EN EL EQUIVALENTE ASCII.

	EXT	DATA, WAIT	
	GLB	ASCII	
	PROG		
ASCII	CLR	F0	LIMPIA BANDERA F0
	MOV	A, R5	
	MOV	R2, A	
	ANL	A, 0F0H	DIVIDE BYTE
	SWAP	A	
	MOV	R3, A	PARTE MAS SIGNIF. EN R3
	MOV	A, R2	
	ANL	A, 00FH	
	MOV	R2, A	PARTE MENOS SIGNIF, EN R2
	JMP	MSBIT	
LSBIT	MOV	A, R2	
	MOV	R3, A	
MSBIT	CLR	C	LIMPIA BANDERA DE ACARRREO
	MOV	R4, 0F6H	RESTA 10
	MOV	A, R4	
	ADD	A, R3	
	JNC	TRANS1	
	MOV	A, 037H	DATOS ABCDEF
	JMP	LINE	
TRANS1	MOV	A, 030H	DATOS 0123456789
LINE	ADD	A, R3	
	MOV	R5, A	
	CALL	DATA	TRANSMITE DATO
	CALL	WAIT	
	JF0	FIN	
	CPL	F0	
	JMP	LSBIT	
FIN	RET		

### SUBROUTINA INCDIR



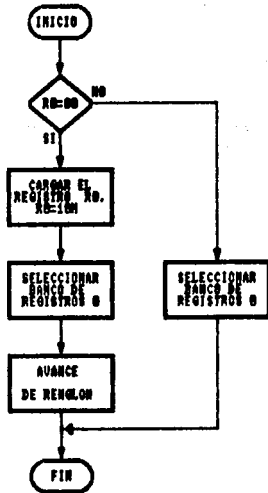
"8048"  
TITLE "ELI26"  
NAME "INCDIR"

;SUBROUTINA INCDIR

;ESTA SUBROUTINA INCREMENTA LA DIRECCION DE LA LOCALIDAD DE  
;MEMORIA QUE SE VA A LEER, PARA LO CUAL UTILIZA UN ARREGLO  
;DE DOS REGISTROS EN LOS QUE SE CONTROLA LA DIRECCION DE LA  
;LOCALIDAD A LEER. LOS REGISTROS EMPLEADOS SON R0 Y R7.

	GLB	INCDIR	
	PROG		
INCDIR	CLR	C	LIMPIA BANDERA DE ACARREO
	MOV	A,R0	INCREMENTAMOS LA PARTE MENOS
	ADD	A,001H	SIGNIFICATIVA ARREGLO (R0)
	JNC	ADIOS	SI NO HAY ACARREO SALE
	INC	R7	SI HAY ACARREO INC R7
ADIOS	MOV	R0,A	
	RET		

**SUBROUTINA INCRE**

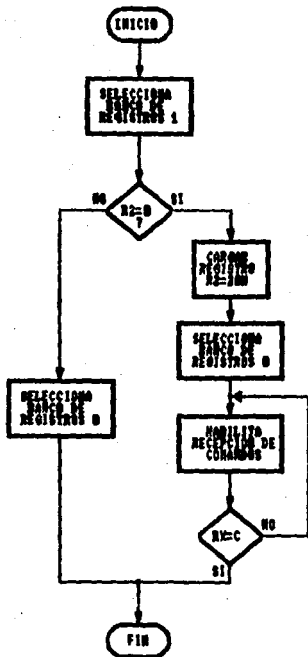


```

"8048"
TITLE "ELI33"
NAME "INCRE"
;SUBROUTINA INCRE
;LOS DATOS ADQUIRIDOS SON TRANSMITIDOS EN BLOQUES DE 32 BYTES
;CONTENIDOS EN DOS RENGLONES, ESTA SUBROUTINA PERMITE "CONTAR"
;EL NUMERO DE CARACTERES ESCRITOS EN CADA RENGLON, EL CUAL
;DEBE SER DE 16 BYTES POR RENGLON. PARA ESTE CONTROL SE EMPLEA
;UN REGISTRO COMO CONTADOR DE BYTES TRANSMITIDOS EL CUAL
;DETECTA CUANDO SE HAN TRANFERIDO LOS BYTES CORRESPONDIENTES
      EXT      DATA, WAIT
      GLB      INCRE
      PROG
INCRE  SEL      RB1
      DJNZ     R0, NEXT      EMPLEA R0 COMO CONTADOR
      MOV      R0, 010H     POR CADA RENGLON
      SEL      R0
      MOV      R5, 00AH     AVANCE DE LINEA
      CALL     DATA
      CALL     WAIT
      MOV      R5, 00DH     CARRY RETURN
      CALL     DATA
      CALL     WAIT
      JMP      NEXT1
NEXT   SEL      R0
NEXT1  RET

```

SUBROUTINA NUMER



"8048"

TITLE "ELI34"

NAME "NUMRE"

;SUBROUTINA NUMRE

;ESTA SUBROUTINA REALIZA UN CONTEO DEL NUMERO DE RENGLONES

;QUE DEBEN SER TRANSMITIDOS EN CADA BLOQUE DE DATOS OBTENIDO.

;CUANDO SE ALCANZA EL NUMERO DE RENGLONES QUE DEBEN SER

;TRANSMITIDOS EN CADA BLOQUE, DOS, SE ESPERA POR LA SENAL DE

;CONTINUACION LA CUAL ES EL COMANDO "C".

	EXT	RECIB	
	GLB	NUMRE	
	PROG		
NUMRE	SEL	RB1	
	DJNZ	R2, NEXT2	
	MOV	R2, 020H	REINICIALIZA EL CONTADOR
	SEL	RB0	
ESPE1	CALL	RECIB	
	XRL	A, 043H	CARACTER RECIBIDO =C ?
	JNZ	ESPE1	
	JMP	RENG	
NEXT2	SEL	RB0	
RENG	RET		

**CAPITULO IV**

**MANUAL DE REPRODUCCION**

**Y MANTENIMIENTO**



## CAPITULO IV

### MANUAL DE REPRODUCCION Y MANTENIMIENTO

#### 4.1 INTRODUCCION

En el presente capitulo se presenta la información necesaria, para realizar el ensamble de una unidad del lector de memorias.

Se incluye además la información para realizar las pruebas iniciales de operación del equipo LM-1-IE. En estas se verifica que las señales en los diferentes puntos de prueba, correspondan a los indicados.

Se explica también el proceso, mediante el cual se prueba la comunicación entre el equipo lector de memorias y la microcomputadora.

#### 4.2 DIAGRAMA DE ENSAMBLE

A continuación se describe la forma de realizar el ensamble de las partes que constituyen el lector de memorias.

Para el ensamble de los componentes electrónicos, se debe tomar como guía la tableta de circuito impreso, en la cual se indica la posición de cada elemento, así como el número con el que se le identifica, tanto en el diagrama electrónico como en la lista de partes.

Los elementos que no van ensamblados en la tarjeta de circuito impreso, deben conectarse como se señala en el diagrama eléctrico.

En cuanto al ensamble de las partes mecánicas, este se realiza teniendo como guía el diagrama explosivo del equipo, donde se muestra la posición de cada componente. Este diagrama se muestra en la figura 4.1.

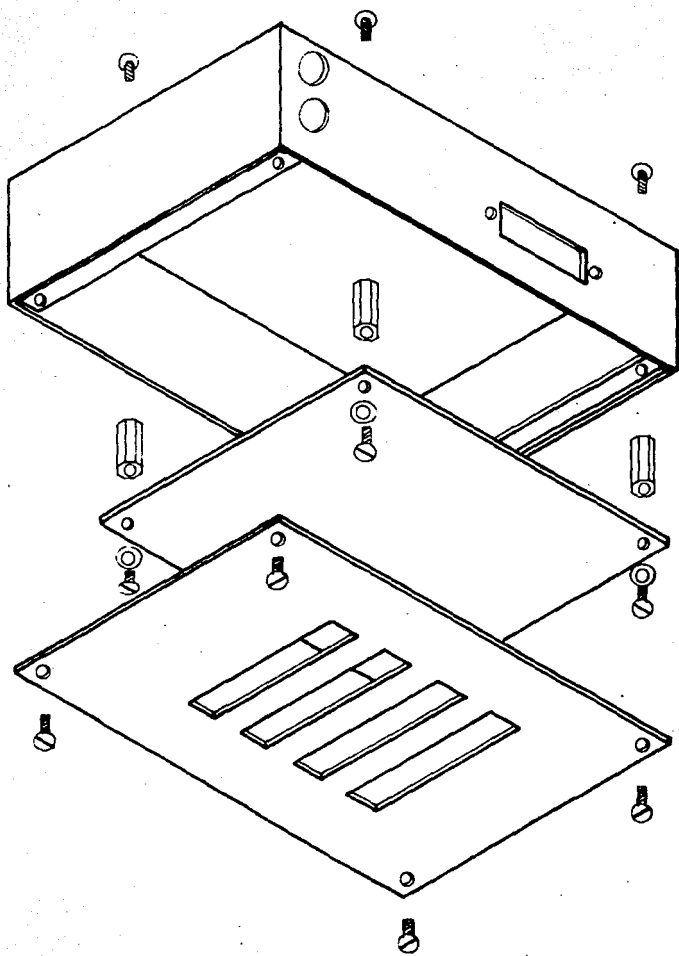


FIG. 4.1 ENSAMBLE MECANICO

### 4.3 PRUEBAS INICIALES

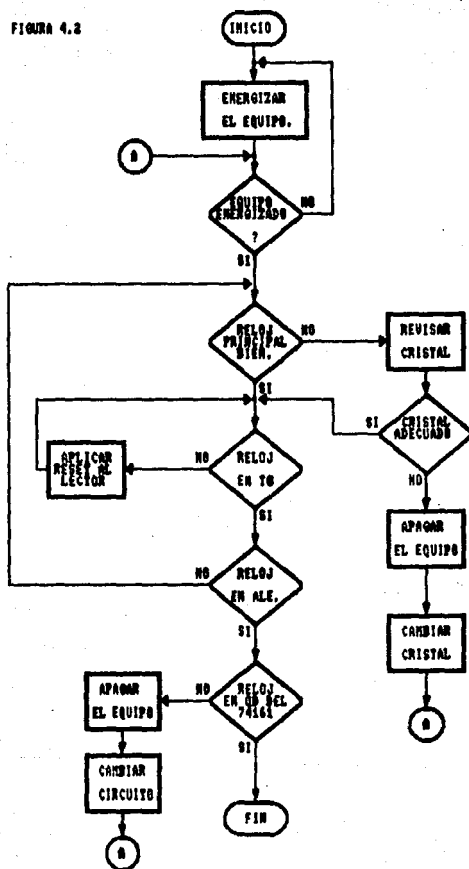
Una vez que el ensamble de los componentes electrónicos ha quedado concluido, se procede a realizar las pruebas por medio de las cuales se determina el adecuado funcionamiento del equipo LM-1-IIE.

Una primer prueba consiste en comprobar que los elementos empleados correspondan a los indicados en el diagrama electrónico, lo cual puede corroborarse por medio de la lista de partes.

Después de comprobar que los elementos empleados son los adecuados, se procede a energizar el equipo, y de esta forma comprobar la existencia de señales adecuadas, en los diferentes puntos de prueba.

El orden o secuencia, con la que deben realizarse las mediciones se indica en el diagrama de flujo de la figura 4.2. En este se indican algunas posibles fallas y operaciones que deben ejecutarse para su solución.

FIGURA 4.2



Para realizar la medición de las diferentes señales de reloj, así como de voltajes en los puntos de prueba indicados, se emplean un multímetro, un osciloscopio y un contador. Este último cuando se requiere mayor precisión en las lecturas de frecuencia de las señales.

A continuación se describen las operaciones que se llevan a cabo, para probar el funcionamiento del lector de memorias.

- Cuando el equipo se encuentra energizado, se verifica que los voltajes de polarización, indicados en el diagrama electrónico para los diferentes circuitos sean los adecuados, en caso negativo, se procede a revisar los niveles de salida de la fuente de alimentación, y posibles falsos contactos.
- Una vez que los voltajes de polarización son los adecuados, se debe revisar la frecuencia proporcionada por el cristal de cuarzo. Esta debe ser de 4.608 MHz y la lectura de la misma, debe hacerse en la terminal 3 (XTAL2) del microprocesador.

En caso de que la frecuencia de oscilación del cristal, no sea la requerida por el equipo, este debe ser cambiado y probar nuevamente la señal adquirida.

- En la salida T0 (terminal 1) del microprocesador, se debe tener una señal cuadrada de frecuencia igual a 1.536 MHz. Cuando esta señal no aparece, se debe aplicar un "reset" al equipo y probar nuevamente.

Si después de realizada esta operación, no se tiene señal de salida, se debe revisar que la memoria haya sido programada adecuadamente, ya que esta señal se habilita por medio de programación.

- La señal en la terminal 11 (ALE) del microprocesador, debe tener un valor igual a una decimoquinta parte de la frecuencia proporcionada por el cristal de cuarzo (4.608 Mhz/15), es decir 307.2 KHz.
- La frecuencia de salida en la terminal Qd del circuito SN74161, debe ser de 19.2 KHz. Esta es la referencia que determina, la velocidad de intercambio de señales entre el lector de memorias y la microcomputadora. Si en esta terminal no se tiene señal alguna, se debe cambiar el circuito y realizar la medición nuevamente.

Una vez que las señales de reloj han sido medidas y estas son adecuadas, se procede a realizar pruebas de transmisión y recepción de señales. Para lo cual se requiere de una terminal de video, está puede ser de cualquier marca (Cromenco, Digital, etc.). Antes de iniciar éstas, el equipo

debe apagarse para ser conectado a la terminal de video (CRT).

En el diagrama de flujo que se muestra en las figuras 4.3 a,b y c, se indica la secuencia para realizar estas pruebas.

- Para que el intercambio de señales pueda ser realizado, es necesario seleccionar en la terminal de video el formato de comunicación empleado por el lector de memorias, esto es:
  - Comunicación serie asíncrona.
  - Velocidad de 1200 Bauds.
  - Longitud de la palabra de 8 bits.
  - Un bit de parada.
  - No paridad.
  
- Después de que tanto el lector de memorias como la terminal de video han sido energizados, se procede a realizar las pruebas de transmisión y recepción de señales, tanto en el lector de memorias como en la terminal de video.



FIGURA 4.3 A

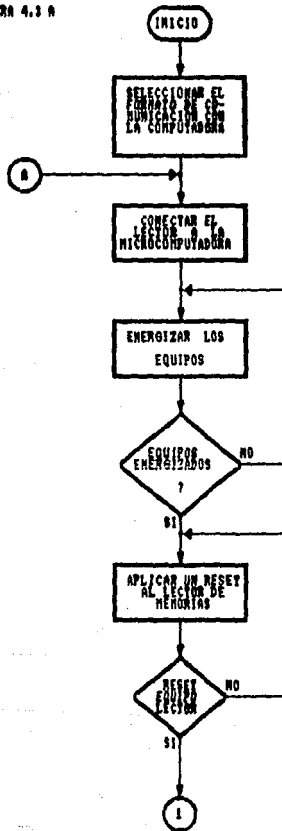


FIGURA 4.3 B

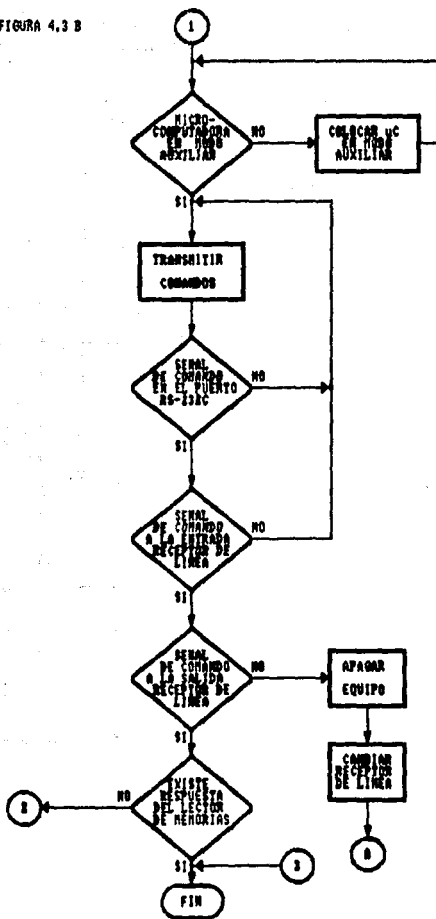
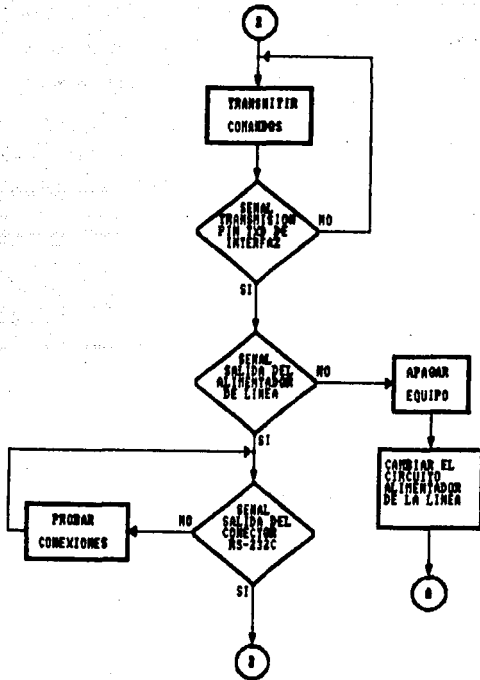


FIGURA 4.3 C



Para verificar que el lector de memorias recibe las señales (comandos) de la terminal de video, se conecta la punta de prueba del osciloscopio en la terminal 2 (RxD) del conector RS-232C, de esta forma cuando en el teclado de la terminal de video se presiona la tecla "X" o alguna otra, debe aparecer en la pantalla del osciloscopio la señal enviada, si esta fué la "X", el lector debe responder con el desplegado del texto de presentación del equipo.

En caso de que se reciba la señal en el lector de memorias, pero no exista respuesta de este, es conveniente aplicar un RESET al lector de memorias e intentarlo nuevamente.

- Si en este momento se comprueba que el lector recibe la señal y no existe respuesta\* se debe verificar que la señal llegue a la interfaz de comunicación, para lo cual debe probarse la existencia de la señal tanto a la entrada como a la salida del circuito receptor de línea (MC1489),

---

\* Existirá respuesta siempre y cuando el comando recibido por el lector, sea uno de los seleccionados, para iniciar cada una de las operaciones.

en caso de que la señal exista a la entrada y no en la salida, se debe cambiar este circuito y probar nuevamente.

- Cuando se ha comprobado que la señal llega a la interfaz de comunicación y el equipo aún no responde, se debe comprobar que exista la señal de salida en la terminal 19 (TxD) de la interfaz de comunicación (P8251A).

Si esta señal aparece pero no se tiene respuesta en la terminal de video se procede a probar los niveles de las señales tanto a la entrada como a la salida del circuito alimentador de línea (driver MC1488), la señal de salida de este circuito debe tener un nivel de  $\pm 10$  volts.

En caso de que no exista salida en este circuito, este debe ser cambiado y probar nuevamente.

- Como prueba final se debe realizar, la lectura de dispositivos de memoria que contengan información conocida de antemano, para asegurar que la información obtenida sea realmente la que contiene la memoria.

Para esta prueba deben ser empleadas una memoria 2716 y una 2532, ya que estos son los tipos de memoria que pueden ser leídos.

Los comandos con los cuales debe iniciarse la lectura de las memorias son: "A" para memorias 2716, y "B" para memorias 2532. La descripción de estos comandos se realiza en el capítulo tres de este documento.

#### 4.4 MANTENIMIENTO

El lector de memorias es un equipo diseñado a base de componentes electrónicos de estado sólido, sin partes mecánicas en movimiento. Esto proporciona al usuario una gran confiabilidad respecto a la continuidad de operación del equipo.

Este puede operar en forma continua durante años y sus fallas solamente se deberán, a fallas propias de los componentes que utiliza. Sin embargo, este debe protegerse de agentes externos, ya que descuidos al respecto pueden ocasionar fallas que requieran de atención especializada.

Cuando sea necesario realizar una revisión de la operación del equipo, esta debe realizarse como se menciona, en la parte correspondiente a pruebas iniciales de operación.

## CAPITULO V

### MANUAL DEL USUARIO

## CAPITULO V

### MANUAL DEL USUARIO

#### 5.1 INTRODUCCION

En el presente capítulo, se describe el procedimiento a seguir durante la instalación y operación del equipo lector de memorias LM-1-IIE.

En la parte referente a la instalación, se describen las condiciones con las que se debe cumplir, así como el formato empleado para el intercambio de señales entre el lector de memorias y la microcomputadora.

Dado que las operaciones realizadas por el equipo deben ser inicializadas a través de un comando, la descripción de estos y las operaciones ejecutadas son descritas en la sección de operación del equipo.



## 5.2 INSTALACION

El lector de memorias LM-1-IIE, es un equipo fácil de instalar, sin embargo, es conveniente que el usuario conozca las condiciones con las que se asegure una correcta instalación.

Antes de energizar el equipo, se debe revisar que el cable RS-232C empleado para conectar el lector de memorias y la microcomputadora, este conectado como se muestra en la figura 5.1.

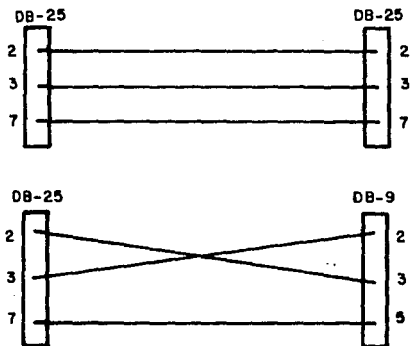


Figura 5.1 Conexiones del cable RS-232C.

Como se observa en esta figura se tienen dos posibles conexiones del cable RS-232C, estas corresponden a las microcomputadoras Olivetti M24 y Televideo AT respectivamente. Los conectores indicados en esta figura son conectores hembra. La configuración del cable puede cambiar dependiendo de la microcomputadora que sea utilizada.

- El primer paso en la instalación, es conectar el equipo a la línea de 120 V, c.a.
- A continuación se debe seleccionar en la microcomputadora el formato de comunicación empleado por el lector de memorias, para realizar el intercambio de señales.

La configuración a seleccionar es la siguiente:

- Comunicación asíncrona.
- Velocidad 1200 Bauds.
- Longitud de la palabra 8 bits.
- Un bit de parada.
- No se emplea bit de paridad.

Una vez que estas condiciones han sido establecidas, se procede a energizar tanto el lector como la microcomputadora e iniciar la lectura de las memorias.

### 5.3 OPERACION

En esta sección, se describe el proceso que se debe seguir durante el funcionamiento del equipo.

- Al encender el equipo, se indicará por medio del LED de encendido. En este momento el equipo queda en espera de los comandos que le indiquen el inicio de una operación.
- Al colocar una memoria en la base S1, se debe tener cuidado de colocar esta de forma que el pin 1 de la memoria coincida con el pin 1 señalado en la base. Ver figura 5.2.
- Cuando una memoria ha sido leída total o parcialmente puede ser retirada de la base, siempre y cuando el LED que indica transmisión (TX), se encuentre apagado, ya que en caso contrario se pueden provocar daños permanentes a la memoria.

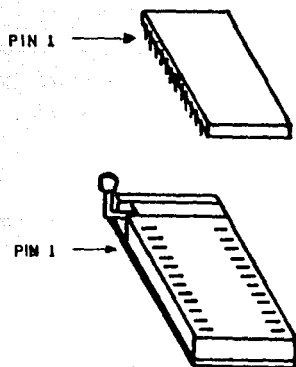


Figura 5.2 Forma de colocar las memorias.

### 5.3.1 COMANDOS UTILIZADOS

Cuando el microprocesador del LM-1-IIE recibe un comando, verifica que corresponda a uno de los seleccionados para iniciar las operaciones. Si este corresponde a uno de estos, se inicia la operación correspondiente, en caso de no ser uno de los seleccionados es desechado y el microprocesador queda en espera de un nuevo comando.

A continuación se muestran los comandos que reconoce el microprocesador del LM-1-IIE, así como la operación iniciada por cada uno de estos:

- COMANDO "X"

Inicia el desplegado del texto de presentación del equipo.

INSTITUTO DE INVESTIGACIONES ELECTRICAS

LECTOR DE MEMORIAS

- COMANDO "M"

Inicia el desplegado del menú de las memorias que pueden ser leídas con el equipo.

2716=A

2532=B

- COMANDO "A"

Inicia la lectura de memorias tipo 2716.

- COMANDO "B"

Inicia la lectura de memorias tipo 2532.

- COMANDO "C"

Indica al equipo que debe continuar con la lectura de las memorias, la cual efectua por bloques, (32 bytes/bloque).

Los comandos reconocidos por el microprocesador del LM-1-IIE pueden ser seleccionados en dos etapas. En la primera es posible seleccionar los comandos "X", "M", "A" ó "B"; mientras que el comando "C" puede ser seleccionado en la

segunda.

Al seleccionar los comandos A ó B, se inicia la lectura de los primeros 32 bytes de la memoria correspondiente, para obtener la información de los bytes restantes es necesario utilizar el comando "C", el cual indica al microprocesador realizar la lectura del siguiente bloque de 32 bytes, mismo que se emplea para obtener toda la información almacenada en el dispositivo de memoria en bloques de 32 bytes. Una vez terminada la lectura de un dispositivo, se pueden seleccionar nuevamente los comandos X, M, A ó B.

### 5.3.2 PROCESAMIENTO DE LA INFORMACION

La información recuperada por medio del equipo lector de memorias LM-1-IIE, de los dispositivos de memoria se obtiene en forma hexadecimal, por lo que es necesario que está sea procesada para realizar una adecuada interpretación de la misma. Para el procesamiento de la información el usuario puede desarrollar programas que le permitan determinar los parametros de interés, de acuerdo al tipo de información que se tenga.

En el caso de la información que se obtiene con el equipo anemométrico ANEMODATA-1-IIE, la recuperación de esta y el procesamiento de la misma puede llevarse a cabo por programas realizados por el usuario o bien utilizando el programa

"ALE.BAS", el cual en forma tutorial permite realizar el procesamiento de los datos de velocidad y dirección del viento que se adquieren con el equipo ANEMODATA.

En la figura 5.3 se muestra el ciclo de recuperación de la información almacenada en los dispositivos de memoria por el equipo ANEMODATA.

A continuación se muestran los archivos creados por el programa ALE.BAS para realizar el procesamiento de la información adquirida por el equipo anemométrico ANEMODATA-1-11E.

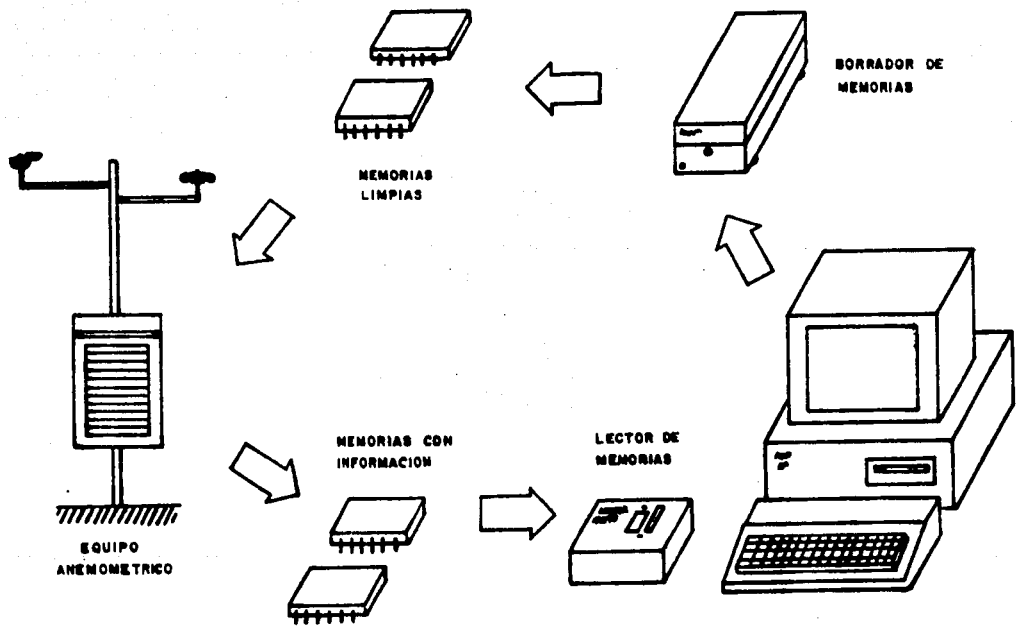


FIGURA 5.3 RECUPERACION DE LA INFORMACION.



LA RICA  
6-MAY-87

12-45

00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
05 13 03 1B 05 14 05 14 01 1D 01 2E 01 1D 05 1B  
05 1B 05 16 05 19 05 16 05 18 05 1C 05 1F 05 16  
05 16 05 17 05 17 05 13 01 12 01 0F 01 0F 03 0E  
05 10 01 12 01 0E 01 15 05 12 05 0F 03 11 05 0D  
05 0F 05 0E 01 0D 05 0F 03 11 01 1C 03 1C 03 0F  
03 12 04 01 06 0B 02 0C 02 05 02 02 04 00 04 0D  
02 13 02 11 02 0C 02 10 02 07 01 11 01 12 01 0B  
01 0C 03 0B 01 06 01 05 01 04 01 04 04 08 04 09  
01 07 01 04 01 09 05 01 05 02 05 02 05 03 01 0D  
01 0A 05 0F 03 0F 03 0F 03 10 03 12 04 14 04 12  
03 12 03 0E 03 13 03 13 03 0F 04 11 04 0F 02 0F  
05 0F 03 13 03 0E 03 10 05 10 03 17 05 11 05 11  
01 16 05 14 05 14 05 15 01 18 05 20 05 1C 05 1F  
05 19 05 13 05 1A 05 17 05 1A 05 18 01 12 05 16  
05 19 05 16 05 1A 05 1D 05 20 05 1E 05 17 05 11  
05 0C 05 07 03 14 03 18 03 21 01 11 05 06 03 11  
05 10 05 14 05 15 01 17 01 0F 01 17 01 0C 01 0C  
01 07 01 08 01 09 01 16 01 17 05 16 05 0D 01 0B  
05 06 01 02 01 04 01 09 01 0B 03 15 03 10 03 09  
01 0A 01 0C 01 10 03 08 03 09 04 0B 03 0B 04 0C  
04 05 04 04 03 01 03 01 03 00 03 0D 03 06 03 01  
03 02 04 29 04 15 04 16 04 26 04 36 04 1E 04 22  
04 22 02 0E 02 0F 02 16 02 13 02 13 02 1C 06 14  
04 0F 05 10 05 15 03 13 03 0C 03 10 01 17 01 18  
05 13 05 14 05 1C 05 19 05 1C 01 21 05 18 05 1E  
05 20 05 1C 05 1B 05 1C 05 1E 05 1E 05 1B 05 19  
05 18 05 19 05 16 05 15 05 14 05 16 01 17 01 0E  
05 14 05 12 01 12 05 0F 01 12 05 19 01 20 01 17  
01 1C 01 28 01 17 01 16 01 11 01 0E 04 0C 04 02  
04 03 04 04 02 0C 02 12 02 04 02 02 02 13 06 14  
06 13 06 0F 02 0B 04 01 06 14 06 17 06 1E 08 11  
08 09 06 08 07 0F 03 05 04 03 04 0E 06 2F 07 15  
03 0E 03 01 03 00 02 08 04 02 04 16 02 08 03 03  
03 0C 03 0E 04 0D 04 21 04 13 04 0A 04 16 04 10  
04 14 04 0A 03 0B 04 0F 04 0D 04 08 04 12 02 16  
02 11 02 11 06 19 02 15 02 13 02 12 04 0E 02 10  
06 17 06 16 04 08 04 0C 03 10 03 0F 05 12 03 0F  
05 15 01 16 05 12 05 14 05 12 05 11 05 15 05 15  
05 13 05 12 05 10 05 10 05 15 03 0F 01 0F 03 14  
03 08 03 06 02 1B 04 20 04 17 02 16 02 06 07 10  
05 0C 07 15 03 03 04 02 06 1D 06 1B 06 16 01 06  
03 22 03 1D 03 08 06 0E 06 03 02 0E 02 15 02 15  
06 10 04 20 04 24 02 24 04 20 04 1D 02 02 02 1B  
02 20 02 23 06 28 02 21 04 1F 02 18 02 29 02 21  
02 1B 02 19 06 16 02 10 02 13 02 13 02 15 02 11  
04 12 02 10 02 0D 02 0A 04 07 02 0D 02 0E 02 14  
02 1B 06 1E 06 1B 06 1F 06 22 06 1F 02 22 02 1A  
06 1C 02 19 02 1F 06 20 02 1C 06 1E 06 22 06 16  
06 14 06 21 02 1B 02 19 04 0C 06 1B 02 18 04 17  
02 1C 04 1A 04 1B 04 19 02 13 07 04 08 0C 08 14

LA RICA  
6-MAY-87  
12-45

SO, 0.54, SO, 0.22, S, 0.00, S, 1.40, SO, 2.05, SO, 1.84, SO, 1.30, SO, 1.73  
SO, 0.76, E, 1.84, E, 1.95, E, 1.19, E, 1.30, SE, 1.19, E, 0.65, E, 0.54  
E, 0.43, E, 0.43, S, 0.86, S, 0.97, E, 0.76, E, 0.43, E, 0.97, NE, 0.11  
NE, 0.22, NE, 0.22, NE, 0.32, E, 1.40, E, 1.08, NE, 1.62, SE, 1.62, SE, 1.62  
SE, 1.73, SE, 1.95, S, 2.16, S, 1.95, SE, 1.95, SE, 1.51, SE, 2.05, SE, 2.05  
SE, 1.62, S, 1.84, S, 1.62, SO, 1.62, NE, 1.62, SE, 2.05, SE, 1.51, SE, 1.73  
NE, 1.73, SE, 2.49, NE, 1.84, NE, 1.84, E, 2.38, NE, 2.43, NE, 2.16, NE, 2.27  
E, 2.59, NE, 3.46, NE, 3.03, NE, 3.35, NE, 2.70, NE, 2.05, NE, 2.81, NE, 2.49  
NE, 2.81, NE, 2.59, E, 1.95, NE, 2.38, NE, 2.70, NE, 2.38, NE, 2.81, NE, 3.13  
NE, 3.46, NE, 3.24, NE, 2.49, NE, 1.84, NE, 1.30, NE, 0.76, SE, 2.16, SE, 2.59  
SE, 3.57, E, 1.84, NE, 0.65, SE, 1.84, NE, 1.73, NE, 2.16, NE, 2.27, E, 2.49  
E, 1.62, E, 2.49, E, 1.30, E, 1.30, E, 0.76, E, 0.86, E, 0.97, E, 2.38  
FINDIA, 100  
E, 2.49, NE, 2.38, NE, 1.40, E, 1.19, NE, 0.65, E, 0.22, E, 0.43, E, 0.97  
E, 1.19, SE, 2.27, SE, 1.73, SE, 0.97, E, 1.08, E, 1.30, E, 1.73, SE, 0.86  
SE, 0.97, S, 1.19, SE, 1.19, S, 1.30, S, 0.54, S, 0.43, SE, 0.11, SE, 0.11  
SE, 0.00, SE, 1.40, SE, 0.65, SE, 0.11, SE, 0.22, S, 4.43, S, 2.27, S, 2.38  
S, 4.11, S, 5.84, S, 3.24, S, 3.67, S, 3.67, SO, 1.51, SO, 1.62, SO, 2.38  
SO, 2.05, SO, 2.05, SO, 3.03, O, 2.16, S, 1.62, NE, 1.73, NE, 2.27, SE, 2.05  
SE, 1.30, SE, 1.73, E, 2.49, E, 2.59, NE, 2.05, NE, 2.16, NE, 3.03, NE, 2.70  
NE, 3.03, E, 3.57, NE, 2.59, NE, 3.24, NE, 3.46, NE, 3.03, NE, 2.92, NE, 3.03  
NE, 3.24, NE, 3.24, NE, 2.92, NE, 2.70, NE, 2.59, NE, 2.70, NE, 2.38, NE, 2.27  
NE, 2.16, NE, 2.38, E, 2.49, E, 1.51, NE, 2.16, NE, 1.95, E, 1.95, NE, 1.62  
E, 1.95, NE, 2.70, E, 3.46, E, 2.49, E, 3.03, E, 4.32, E, 2.49, E, 2.38  
E, 1.84, E, 1.51, S, 1.30, S, 0.22, S, 0.32, S, 0.43, SO, 1.30, SO, 1.95  
FINDIA, 100  
SO, 0.43, SO, 0.22, SO, 2.05, O, 2.16, O, 2.05, O, 1.62, SO, 0.86, S, 0.11  
O, 2.16, O, 2.49, O, 3.24, NO, 1.84, NO, 0.97, NO, 0.86, N, 1.62, SE, 0.54  
S, 0.32, S, 1.51, O, 5.08, N, 2.27, SE, 1.51, SE, 0.11, SE, 0.00, SO, 0.86  
S, 0.22, S, 2.38, SO, 0.86, SE, 0.32, SE, 1.30, SE, 1.51, S, 1.40, S, 3.57  
S, 2.05, S, 1.08, S, 2.38, S, 1.73, S, 2.16, S, 1.08, SE, 1.19, S, 1.62  
S, 1.40, S, 0.86, S, 1.95, SO, 2.38, SO, 1.84, SO, 1.84, O, 2.70, SO, 2.27  
SO, 2.05, SO, 1.95, S, 1.51, SO, 1.73, O, 2.49, O, 2.38, S, 0.86, S, 1.30  
SE, 1.73, SE, 1.62, NE, 1.95, SE, 1.62, NE, 2.27, E, 2.38, NE, 1.95, NE, 2.16  
NE, 1.95, NE, 1.84, NE, 2.27, NE, 2.27, NE, 2.05, NE, 1.95, NE, 1.73, NE, 1.73  
NE, 2.27, SE, 1.62, E, 1.62, SE, 2.16, SE, 0.86, SE, 0.65, SO, 2.92, S, 3.46  
S, 2.49, SO, 2.38, SO, 0.65, N, 1.73, NE, 1.30, N, 2.27, SE, 0.32, S, 0.22  
O, 3.13, O, 2.92, O, 2.38, E, 2.38, SE, 3.67, SE, 3.13, SE, 0.86, O, 1.51  
FINDIA, 100  
O, 0.32, SO, 1.51, SO, 2.27, SO, 0.54, O, 1.73, S, 3.46, S, 3.89, SO, 3.89  
S, 3.46, S, 3.13, SO, 0.22, SO, 2.92, SO, 3.46, SO, 3.78, O, 4.32, SO, 3.57  
S, 3.35, SO, 2.59, SO, 4.43, SO, 3.57, SO, 2.92, SO, 2.70, O, 2.38, SO, 1.73  
SO, 2.05, SO, 2.05, SO, 2.27, SO, 1.84, S, 1.95, SO, 1.73, SO, 1.40, SO, 1.08  
S, 0.76, SO, 1.40, SO, 1.51, SO, 2.16, SO, 2.92, O, 3.24, O, 2.92, O, 3.35  
O, 3.67, O, 3.35, SO, 3.67, SO, 2.81, O, 3.03, SO, 2.70, SO, 3.35, O, 3.46  
SO, 3.03, O, 3.24, O, 3.67, O, 2.38, O, 2.16, O, 3.57, SO, 2.92, SO, 2.70  
S, 1.30, O, 2.92, SO, 2.59, S, 2.49, SO, 3.03, S, 2.81, S, 2.92, S, 2.70  
SO, 2.05, N, 0.43, NO, 1.30, NO, 2.16, NO, 2.49, SO, 4.76, S, 4.32, S, 5.40  
S, 4.43, S, 6.27, S, 4.54, SO, 3.46, SE, 5.30, SE, 5.73, SE, 7.89, SE, 6.59  
S, 3.24, S, 3.35, S, 4.00, SO, 2.81, S, 2.27, SO, 2.59, O, 1.40, S, 0.43  
S, 0.54, S, 1.08, S, 0.32, S, 0.86, S, 0.54, S, 0.32, S, 0.22, S, 0.86  
FINDIA, 100  
SO, 1.85, SO, 1.73, SO, 1.73, S, 2.16, S, 2.38, S, 2.81, SO, 2.16, S, 2.1

#### 5.4 MANTENIMIENTO

El lector de memorias es un equipo diseñado a base de componentes electrónicos de estado sólido, lo que proporciona confiabilidad respecto a su funcionamiento. Este equipo puede ser operado durante largos períodos de tiempo y sus fallas serán únicamente debido a fallas propias de los componentes con que esta construido.

Sin embargo es conveniente tomar las precauciones necesarias durante la utilización del mismo, ya que descuidos al respecto pueden ocasionar fallas que requieran de atención especializada.

Durante la manipulación del equipo, se debe tener cuidado de no dañar este, ya que golpes fuertes, pueden ser causa de errores en su funcionamiento.

En cuanto al almacenamiento del equipo, este debe ser colocado en un sitio, en el cual este protegido contra agentes externos (líquidos, gases corrosivos, etc.), los cuales pueden afectar el funcionamiento del equipo.

Cuando sea necesario proporcionar mantenimiento al equipo, este debe realizarse por personal capacitado siguiendo las indicaciones descritas en el capítulo cuatro, en la parte correspondiente a pruebas iniciales de operación.

## **5.5 ESPECIFICACIONES**

### **CONSUMO DE ENERGIA**

120 V C.A. 70 mA C.A.

### **FORMATO DE COMUNICACION**

- Comunicación serie asincrónica.
- Velocidad 1200 Bauds.
- Longitud de la palabra de 8 bits.
- Un bit de paro.
- No se emplea paridad.

### **PUERTO DE SALIDA**

Puerto RS-232C (macho).

### **LECTURA DE MEMORIAS**

Se puede realizar la lectura de las memorias que a continuación se indican: 2716 y 2532. Además de estas es posible realizar la lectura de memorias compatibles pin por pin con los tipos indicados anteriormente.

## CONCLUSIONES

## CONCLUSIONES

En el presente trabajo se describieron las etapas involucradas en el desarrollo del equipo lector de memorias LM-1-III, diseñado para ser empleado en el sistema ANEMODATA desarrollado en el Instituto de Investigaciones Eléctricas.

Este trabajo incluye las etapas correspondientes a consideraciones teóricas, diseño conceptual y desarrollo del equipo. Esta última dividida en dos partes, hardware y software, de igual forma se describen los procesos mediante los cuales realizar el ensamble de una unidad y los requisitos para una instalación y operación adecuada del equipo. Cada etapa fué descrita en el capítulo correspondiente.

Para modificar las funciones originales del equipo, en caso de ser necesario, debe modificarse el programa, esto para realizar lectura de tipos diferentes de memorias a los considerados en el diseño original.

Actualmente el equipo lector de memorias es utilizado con el sistema de adquisición de datos denominado ANEMODATA, aunque su empleo no esta restringido a este equipo ya que puede ser utilizado por entidades que requieran recuperar información almacenada en dispositivos de memoria de estado sólido, de los tipos que pueden ser leídos con este equipo.

Otra posible modificación de las funciones del equipo, es utilizar este para programar dispositivos de memoria además de "leer" el contenido de los mismos. Para esto sería necesario modificar el programa y agregar algunos elementos de hardware, con los cuales obtener los niveles de voltaje necesarios para la programación de memorias.

## BIBLIOGRAFIA



## BIBLIOGRAFIA

1. Roger S. Pressman  
SOFTWARE ENGINEERING: A Practitioners Approach  
Ed. Mc Graw Hill.
2. Will D. Gillett, Seymour V. Pollack  
AN INTRODUCTION TO ENGINEERED SOFTWARE  
Ed. HRW (Holt, Rinehart and Winston)  
1982 CBS College Publishing.
3. Michael M. Cirovic  
HANDBOOK OF SEMICONDUCTOR MEMORIES  
Ed. Reston Publishing Company Inc. 1981.
4. Robert L. Glass  
SOFTWARE RELIABILITY GUIDEBOOK  
Ed. Prentice Hall, Inc. 1979.
5. Walter H. Buchsbaum, Gina Weissenberg  
MICROPROCESSOR AND MICROCOMPUTER DATA DIGEST  
Ed. Reston Publishing Company Inc.  
a Prentice Hall company. 1983.
6. Garcia Narcia Octavio  
MICROPROCESADOR 8080/8085 E INTERFACES  
Ed. Bioediciones.
7. Varios Autores  
MICROPROCESADORES Y MICROCOMPUTADORES  
Mundo Electrónico  
Ed. Marcombo-Boixereau Editores  
Representaciones y Servicios de Ingeniería S.A. 1978.
8. R.E. H. Bywater  
HARDWARE/SOFTWARE DESIGN OF DIGITAL SYSTEMS  
Prentice Hall International 1981.

9. Mandado Enrique  
SISTEMAS ELECTRONICOS DIGITALES  
Editorial Marcombo 1977.
10. D. E. Heffer, G. A. King and D. Keith  
BASIC PRINCIPLES AND PRACTICE OF MICROPROCESSORS  
Edward Arnold (Publishers) LTD.  
London, 1981.
11. Granino A. Korn  
MICROPROCESSORS AND SMALL DIGITAL COMPUTERS  
SYSTEMS FOR ENGINEERS AND SCIENTISTS  
Ed. Mc Graw Hill Book Co. 1977.
12. Garcia Narcia Octavio  
MICROPROCESADOR Z80 E INTERFACES  
Ed. Bioediciones.
13. Thomas C. Bartee  
FUNDAMENTOS DE COMPUTADORAS DIGITALES  
Ed. Mc Graw Hill  
Quinta edición, 1984.
14. Adi J. Kambata  
MICROPROCESSORS/MICROCOMPUTERS  
Architecture, Software and Systems  
John Wiley and Sons 1982.

#### ARTICULOS TECNICOS

15. Victor R. Basili  
STRUCTURED PROGRAMMING  
Auerbach Publishers Inc. 1978.
16. E. W. Dijkstra  
STRUCTURED PROGRAMMING  
Software Engineering: Concepts and techniques  
Mason/Charter Publishers Inc. 1976.
17. James W. Hunt  
(Bell Laboratories)  
PROGRAMMING LANGUAGES  
Computer, april 1982, pages 70-87

IEEE Inc. 1982.

18. W. P. Stevens, G. J. Myers and L. L. Constantine  
STRUCTURED DESIGN  
IBM Systems Journal, vol. 13, number 12, pages 115-139  
1974.
19. Leonard L. Tripp  
TOP-DOWN, BOTTOM-UP APPROACH TO  
SOFTWARE ENGINEERING STANDARDS  
Boeing Computer Services Co.
20. Harlan D. Mills  
SOFTWARE ENGINEERING  
A Mathematical basis is needed for the practical  
control of computers in complex applications.  
Science, march 1977, pages 1199-1204  
American Association for the Advancement of  
Science, 1977.
21. F. Terry Baker  
STRUCTURED PROGRAMMING IN A PRODUCTION  
PROGRAMMING ENVIRONMENT  
IEEE Transactions on Software Engineering  
Vol. SE-1, number 2, pages 193-204, june 1975  
IEEE Inc. 1975.

#### MANUALES

22. Intel Corporation  
MCS-48 MICROCOMPUTER USERS MANUAL
23. Motorola Semiconductor Products Inc.  
MOTOROLA CMOS DATA BOOK  
3501 Ed Bluestein Blvd  
Austin, Texas
24. Motorola Semiconductor Products  
MOTOROLA MEMORY DATA MANUAL  
3501 Ed Bluestein Blvd  
Austin, Texas

25. National Semiconductor Corp.  
VOLTAJE REGULATOR HANDBOOK 1980  
2900 Semiconductor Drive  
Santa Clara, CA.
  
26. Texas Instruments Inc.  
THE TTL DATA BOOK  
for Design Engineers  
Second Edition 1976
  
27. EIA Communications Standard Manual  
EIA
  
28. NORMAS DEL CCITT

**A N E X O I**

**TEORIA DEL MICROPROCESADOR**

## ANEXO I

### TEORIA DEL MICROPROCESADOR

#### 1.1 COMPUTADORAS DIGITALES

Una computadora es un sistema o máquina compuesto por diferentes elementos relacionados entre si, para realizar un proceso determinado el cual ejecuta una secuencia de operaciones sobre datos determinados.

El proposito de cada proceso puede ser la solución de problemas matemáticos, el control de ciertas funciones de otros dispositivos, o bién una combinación de los dos.

Existen dos tipos básicos de computadoras: Digital y Analógica. Una computadora analógica, es aquella que trabaja con señales eléctricas continuas, mientras que las computadoras digitales trabajan con señales eléctricas discretas. Existe un tercer tipo de computadoras, estas son las computadoras híbridas las cuales manejan tanto señales continuas como señales discretas. En este trabajo se hace referencia a las digitales.

En las computadoras digitales, las cantidades numéricas están representadas por niveles de voltaje, o pulsos eléctricos. La presencia o ausencia de un pulso de los niveles adecuados, define un dígito binario o bit. Un grupo de bits, es llamado una palabra, la cual consiste de dígitos binarios o bits, expresados como unos y ceros y estos por pulsos eléctricos definidos apropiadamente. Una palabra puede representar a una instrucción o a un dato.

La primera computadora digital fue la máquina llamada ENIAC (Electronic Numerical Integrator and Calculator), la cual fue diseñada y construida en la Universidad de Pennsylvania en 1943. Esta utilizaba cerca de 18000 válvulas, y ocupaba una habitación de aproximadamente 20 x 10 m, con un peso de más de 30 ton. Desafortunadamente uno de los mayores problemas a los que se enfrentó fue la poca confiabilidad hasta entonces desarrollada en los componentes involucrados en su diseño, lo cual provocaba que el tiempo medio entre fallas fuese de solo minutos.

Con la invención del transistor en 1948 por William Shockley (transistor de germanio), se inició a principios de los 60s una nueva generación de computadoras, las cuales fueron fabricadas empleando la nueva tecnología de "estado sólido".

A mediados de 1960s un mayor avance tecnológico tuvo lugar dentro de la industria de semiconductores, donde empleando una avanzada técnica de fabricación de transistores, fue posible producir en una sola oblea de silicio varios transistores interconectados, con lo cual se obtuvo el circuito integrado (CI); Este avance en la fabricación de semiconductores dió origen a una nueva generación de computadoras, en las que se aumentó la confiabilidad en el equipo, se logro un tamaño reducido y un sustancial ahorro en su costo de fabricación y en la energía requerida para su funcionamiento.

## 1.2 ESTRUCTURA DE UNA COMPUTADORA DIGITAL

Una computadora digital esta formada por cuatro partes principales que son: La Unidad de Procesamiento Central (CPU), La Memoria, La Unidad de Entrada y la Unidad de Salida. En la figura A1.1 se muestra un diagrama de bloques con sus elementos esenciales. La diferencia principal entre los diferentes tamaños de computadoras es el número de teclados, pantallas de TV, discos, memoria y otros periféricos conectados a la CPU (Unidad de Procesamiento Central). Esto por supuesto, requiere de una CPU de diferente tamaño y velocidad, pero las funciones esenciales de cada uno de los elementos mostrados en la figura A1.1 son siempre las mismas.



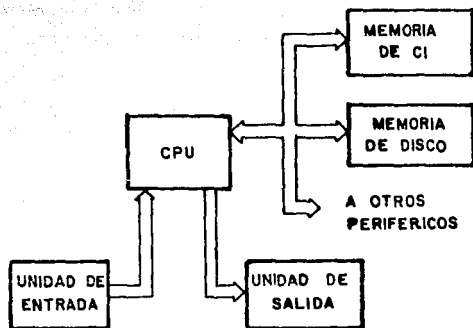


Figura A1.1 Diagrama a bloques de una Computadora.

### 1.2.1 UNIDAD DE PROCESAMIENTO CENTRAL (CPU)

Una de las principales partes de una computadora es la Unidad de Procesamiento Central (CPU), ya que es aquí donde se realizan todas las operaciones aritméticas y lógicas, y de donde emanan la mayoría de señales de control. La CPU actúa como el centro de ordenes y se considera el cerebro del sistema; Recibe instrucciones del operador humano a través de un dispositivo de entrada y muestra información al operador a través de un dispositivo de salida. La información que entra y sale de la CPU es generalmente dividida en dos categorías: El programa y los datos.

El grupo o conjunto de instrucciones presentados a la computadora se conoce como "programa", y al proceso o procedimiento con el cual son ejecutados los cálculos matemáticos y operaciones lógicas, se le llama "algoritmo".

El programa, consiste de dos partes principales: El grupo de instrucciones que provocan que una tarea sea ejecutada automáticamente, y los datos sobre los que las instrucciones deben operar.

Cada instrucción tiene un código binario único, el cual es reconocido por la computadora, la que a su vez realiza la operación indicada. De aquí que uno de los aspectos mas importantes de la operación de una computadora, es la capacidad de distinguir entre una palabra binaria que representa a una instrucción y aquella que representa a un dato.

En el diagrama a bloques de la figura A1.2 se muestran los elementos esenciales que comprenden una CPU. La configuración de estos elementos varía entre diferentes CPUs. En algunos de ellos ciertas funciones están expandidas, duplicadas, y triplicadas pero todos tienen al menos una de cada una de las funciones mostradas en la figura A1.2. La información entra desde el exterior vía los puertos de Entrada-Salida (E/S ó I/O).

Las líneas punteadas representan a las líneas de control, son empleadas por la sección de control para determinar la función y sincronización de otros elementos.

La Unidad Aritmética Lógica (ALU) es el bloque que realiza las funciones lógicas y aritméticas sobre los datos.

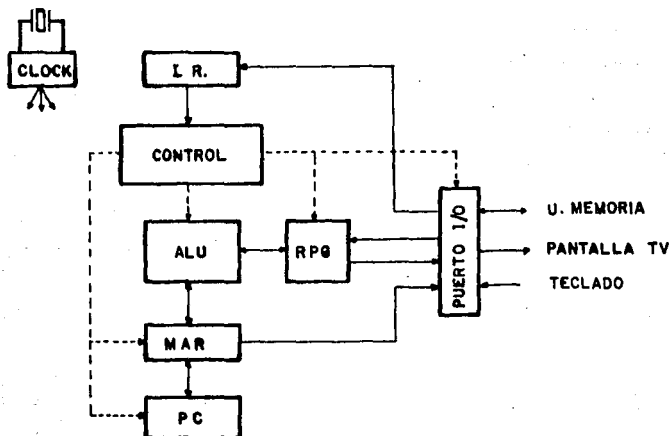


Figura A1.2 Diagrama a bloques de la CPU.

Uno de los elementos básicos de cualquier CPU es el contador de programa (PC), el cual es actualizado para cada instrucción y sirve como almacenamiento temporal para el número de la instrucción. Cuando en el programa se indica un salto a una instrucción completamente diferente, el PC es colocado con el número de la nueva instrucción.

## 1.2.2 UNIDAD DE MEMORIA

En una computadora, la unidad de memoria es usada principalmente para dos propósitos: Almacenar datos y comandos o instrucciones.

La porción de la memoria que almacena a los operandos, es llamada la "memoria de datos". Esta almacena los datos que serán utilizados por la computadora durante la ejecución del programa. Esta es generalmente una memoria de acceso aleatorio (RAM). En ciertas aplicaciones la memoria interna de datos de la computadora no es suficiente para retener todos los datos requeridos, por lo que la computadora debe adquirir estos de fuentes auxiliares, como son: cintas magnéticas, memorias de burbuja o discos magnéticos.

La parte de la memoria que almacena los comandos o instrucciones se conoce como "memoria de programa". Cada instrucción contenida en esta memoria, es transferida a la computadora en una secuencia preestablecida. La computadora decodifica cada instrucción e inicia el proceso específico indicado.

Normalmente las instrucciones son cargadas dentro de la memoria de programa, antes de la ejecución del programa. Por esta razón la memoria de programa, es algunas veces llamada memoria de solo lectura (ROM-Read Only Memory).

A la memoria interna de una computadora, se le conoce como memoria principal. Esta incluye la memoria de datos y la de programa. En la figura A1.1 se muestran separadas, una memoria de circuito integrado y un disco, el cual también sirve como memoria. La principal diferencia entre ellos es que, mientras la memoria de CI opera mas rápido que el disco, su contenido se pierde cuando la fuente de poder es desconectada, mientras la información magnética en el disco permanece disponible hasta que sea borrada en forma deliberada.

En la mayoría de sistemas la memoria de CI es usada como un almacén temporal y la información del disco es cargada en la memoria de CI para que la CPU trabaje con los datos a relativamente alta velocidad.

### 1.2.3 UNIDAD DE ENTRADA-SALIDA

Esta unidad sirve a la computadora como el medio para comunicarse con dispositivos periféricos de entrada/salida (I/O).

El puerto de entrada permite a la CPU adquirir datos de dispositivos externos o bancos de memoria. El puerto de salida habilita a la CPU comunicar el resultado de sus cálculos o procesos lógicos a través de periféricos. Puede también ser empleado para transmitir señales de control de

procesos, que pueden iniciar la operación de sistemas externos.

Dispositivos de entrada comunes son: lectora de tarjetas, cinta de papel perforada, teclado, cinta y disco magnéticos. Los dispositivos de salida proporcionan los resultados de las operaciones realizadas a través de teletipos, tubos de rayos catódicos (CRT), cinta perforada, impresora de líneas, cintas magnéticas, etc.

Otros periféricos que generalmente forman parte de un sistema de computo, son discos adicionales, cintas magnéticas, cintas de papel perforado y por supuesto impresoras las cuales cambian la información codificada por la computadora en alfanuméricos familiares al operador humano.

Los sistemas de computo son generalmente diseñados para un cierto número de teclados y pantallas de TV (llamados terminales de tubos de rayos catódicos CRT), un complemento específico de controladores (drives) de discos, memorias de CI, equipos de cinta y otros periféricos. Cualquiera que sea la configuración, la CPU es el conductor de esta orquesta digital y por sí sola determina cuando cada uno de los bloques funcionales en el sistema debe ejecutar la tarea asignada.

La transferencia de información de una unidad a otra de la computadora se realiza a través de canales de transferencia de datos (buses); básicamente son dos los tipos de estructura

del bus: Unidireccionales, como los empleados desde el teclado a la CPU y de la CPU a la pantalla de TV; Bidireccionales, como el empleado para conectar la CPU a la memoria de CI, el disco y otros periféricos.

### 1.3 EL MICROPROCESADOR

En una minicomputadora la CPU consiste de varios circuitos integrados (CI), conectados entre si y ensamblados en una sola tarjeta de circuito impreso. Sin embargo en 1969 la compañía INTEL CORP. inició el diseño de un circuito integrado de alta escala de integración (LSI), a solicitud de un fabricante de calculadoras. El producto resultante fué un circuito integrado (CHIP), que ejecutaba todas las operaciones aritméticas y de control de acuerdo con un programa que se le había almacenado. Tal CI tenía características de operación similares a la CPU de un minicomputador por lo que se le dió el nombre de microprocesador.

El primer microprocesador fue conocido con el número 4004 (INTEL), este tiene una longitud de palabras de datos e instrucciones de 4-bits. Este dispositivo fue utilizado para aplicaciones distintas a las de una calculadora, con lo que apareció la primera generación de sistemas basados en un microprocesador.

Poco tiempo después el 4004 fue seguido por un microprocesador de 8 bits el 8008. Este es un dispositivo más general el cual cuenta con un grupo de instrucciones similar a las de una minicomputadora. Un año después apareció el 8080: un dispositivo de 8 bits que opera 20 veces más rápido que el 8008.

En 1974 Motorola introdujo al mercado el microprocesador de 8 bits 6800, el cual junto con el 8080 fueron los líderes en el mercado a pesar de la entrada al mercado de otros fabricantes, compañías tales como: Fairchild, Texas, Signetics, RCA, National Semiconductor, Zilog y Mostek, los cuales han aportado diferentes dispositivos desde la aparición en 1972 del 4004.

Los principales aportes realizados han sido más facilidades disponibles dentro de un solo CI; operación más rápida; empleo de palabras con una longitud de 16 y 32 bits o bien han ampliado la capacidad aritmética del equipo.

Por medio de la conexión de un microprocesador con otros circuitos integrados y componentes adicionales es posible producir una microcomputadora. Dispositivo que opera en forma similar a una minicomputadora, aunque considerablemente más barato, pero de operación más lenta. Sin embargo, estas diferencias están disminuyendo, y las últimas generaciones de microcomputadores son tan rápidas y tienen tantas habilidades como las minicomputadoras de hace solo unos años.



### 1.3.1 CLASES DE MICROPROCESADORES

El término microprocesador es generalmente aplicado a cualquier clase de circuito integrado, que realiza las funciones de una unidad de procesamiento central, el centro de acción de cualquier computador. Desafortunadamente en muchos casos las palabras microcomputador y microprocesador son empleadas como sinónimos, resultando que la palabra microprocesador es usada para describir unicamente a la CPU (uso correcto), o al sistema basado en un microprocesador o microcomputadora.

La vasta mayoría de microprocesadores, son cualquier unidad de procesamiento central, o completos microcomputadores. Variaciones de estos son los microprocesadores de capas, algunas veces llamados procesadores de capas (bit slices) y el microcontrolador.

El microprocesador de capa como el nombre lo implica, realiza solamente algunas de las funciones del microprocesador, generalmente se requiere de varios microprocesadores de este tipo unidos con circuitos auxiliares para hacer una simple CPU. Arreglos de este tipo son útiles para aplicaciones especiales donde alta velocidad de ejecución de alguno de los elementos de la CPU es mas importante que el menor tamaño posible, cuando todas las funciones son

contenidas en un solo chip.

Los microcontroladores son, generalmente microcomputadores de propósito especial, los cuales son programados para ejecutar tareas específicas de control. Un microcontrolador está generalmente definido como un circuito integrado el cual contiene una CPU, una memoria de solo lectura (ROM), una memoria de acceso aleatorio (RAM), y todo el bus de interconexión y facilidades de puerto que hacen una pequeña computadora de capacidad limitada. Hay algunos microcontroladores que pueden ser usados como una CPU o como una microcomputadora. Estos dispositivos generalmente tienen una ROM y algunas veces también una RAM y pueden ser programados para funcionar como una microcomputadora contenida en un solo chip, o como una muy flexible y poderosa unidad de procesamiento central.

#### 1.4 ESTRUCTURA DEL MICROPROCESADOR

La disminución de la complejidad de la Unidad Central de Procesamiento ha permitido la realización en un solo circuito integrado de la CPU. Para reducir al mínimo el número de terminales del microprocesador, las transferencias de información con los periféricos se realiza a través de la misma conexión de las de la memoria. En la figura A1.3 se muestra un diagrama a bloques de los elementos que componen a la CPU.

El microprocesador envía señales de control a la memoria y a los periféricos para realizar transferencias con cualquiera de ellos. Pero además, los periféricos pueden decidir la iniciación de las transferencias y por ello envían señales de control al microprocesador.

A continuación se realiza una descripción de los bloques básicos que componen la CPU y sus relaciones funcionales. La CPU está formada por tres secciones principales que son:

1. Unidad Aritmética y Lógica, la cual cuenta con diferentes banderas de estados.
2. La Unidad de Control.
3. Registros de propósito especial y general.

### 1.4.3 REGISTROS

Los elementos restantes de la CPU son el bus de datos interno y los registros. El bus de datos interno proporciona el camino para la transferencia de la información binaria entre las diferentes partes de la CPU. Este bus está formado por varias líneas, donde en cada línea se transmite un bit. Este es un bus paralelo bidireccional.

La CPU cuenta además con uno o varios registros en los cuales se almacena la información durante el procesamiento de las instrucciones y en donde se recibe la información que se obtiene de la memoria. Estos se encuentran agrupados en dos categorías que son:

1. Registros de Propósito General.
2. Registros de Propósito Especial.

#### 1.4.3.1 EL ACUMULADOR

Entre los registros de propósito general se encuentra el acumulador y otros registros que se utilizan para almacenar información o para proporcionar un operando en las instrucciones aritméticas y lógicas.

- Paridad.
- Signo.
- Sobreflujo.
- Acarreo intermedio (del bit 3 a 4).

La serie de instrucciones de la CPU debe tener instrucciones (condicionales) que le permitan conocer los valores de las banderas en cualquier momento y en base al resultado realizar una lógica u otra de procesamiento.

#### 1.4.2 LA UNIDAD DE CONTROL

La unidad de control es la responsable de ordenar y sincronizar las acciones que se deben desarrollar en la CPU. Recibe las instrucciones codificadas en binario desde la memoria y una vez decodificado el código, en el "Decodificador de Instrucciones" decide cuando, como y que operaciones efectuar para la ejecución de las instrucciones. Conoce cuando termina la ejecución de una instrucción e indica cual es la instrucción que se debe ejecutar a continuación. Esta unidad tiene el control de la ALU, los registros y el registro de banderas de estado.

- Paridad.
- Signó.
- Sobreflujo.
- Acarreo intermedio (del bit 3 a 4).

La serie de instrucciones de la CPU debe tener instrucciones (condicionales) que le permitan conocer los valores de las banderas en cualquier momento y en base al resultado realizar una lógica u otra de procesamiento.

#### 1.4.2 LA UNIDAD DE CONTROL

La unidad de control es la responsable de ordenar y sincronizar las acciones que se deben desarrollar en la CPU. Recibe las instrucciones codificadas en binario desde la memoria y una vez decodificado el código, en el "Decodificador de Instrucciones" decide cuando, como y que operaciones efectuar para la ejecución de las instrucciones. Conoce cuando termina la ejecución de una instrucción e indica cual es la instrucción que se debe ejecutar a continuación. Esta unidad tiene el control de la ALU, los registros y el registro de banderas de estado.

### 1.4.3 REGISTROS

Los elementos restantes de la CPU son el bus de datos interno y los registros. El bus de datos interno proporciona el camino para la transferencia de la información binaria entre las diferentes partes de la CPU. Este bus esta formado por varias líneas, donde en cada línea se transmite un bit. Este es un bus paralelo bidireccional.

La CPU cuenta además con uno o varios registros en los cuales se almacena la información durante el procesamiento de las instrucciones y en donde se recibe la información que se obtiene de la memoria. Estos se encuentran agrupados en dos categorías que son:

1. Registros de Propósito General.
2. Registros de Propósito Especial.

#### 1.4.3.1 EL ACUMULADOR

Entre los registros de propósito general se encuentra el acumulador y otros registros que se utilizan para almacenar información o para proporcionar un operando en las instrucciones aritméticas y lógicas.

El acumulador es el registro principal de la CPU, ya que está operando con el contenido de este registro. En las operaciones que realiza la ALU el acumulador casi siempre contiene uno de los operandos. La ALU obtiene el segundo, si se requiere, de otros registros, de propósito general o de la memoria. Generalmente los resultados de las operaciones efectuadas por la ALU se almacenan en el acumulador, del cual son transferidos hacia otros registros o a algún dispositivo de salida.

Los registros de propósito especial realizan una función especial dentro de la CPU, estos son utilizados en algunos casos como contadores, entre estos se encuentran el contador del programa (PC), el contador de datos (DC) y el registro de instrucción (IR).

#### 1.4.3.2 CONTADOR DEL PROGRAMA (PC)

Este es un registro que contiene la dirección de la localidad de memoria de donde la CPU obtiene el código de la instrucción que se debe ejecutar a continuación. Cada vez que la CPU realiza un acceso a memoria para obtener un código de instrucción (FETCH) o un dato (Leer memoria) el contenido del PC se incrementa en uno.



El PC toma valores secuenciales, la única excepción es cuando la CPU ejecuta instrucciones de salto (JUMP). El número de bits del PC limita el número máximo de direcciones que puede direccionar la CPU. Un PC con 16 líneas puede direccionar 64,536 localidades de memoria (64K).

Cuando una instrucción de salto es insertada en el programa principal, la secuencia normal del programa es suspendida. En esta situación la instrucción de salto dirige el PC a alguna dirección dentro del programa principal. La instrucción de salto contiene la dirección de la instrucción en la cual el programa debe continuar. Cuando se requiere ejecutar una subrutina es necesario salvar la dirección de la próxima instrucción que debe ejecutarse después de terminada la subrutina en algún registro para que sea recuperada después de terminada la subrutina. El área o grupo de registros en los que se salva esta dirección es conocida como stack.

#### 1.4.3.3 EL CONTADOR DE DATOS (DC)

EL contador de programa permite acceder los códigos de las instrucciones y datos que se encuentran a continuación de la instrucción que se está ejecutando. La CPU requiere de datos que no se encuentran a continuación de los códigos de instrucción, en esta situación no es conveniente cargar el PC con la dirección del dato ya que se perdería la secuencia del programa.

Para evitar este problema la CPU cuenta con uno o varios registros que le permiten direccionar a la memoria para obtener datos sin interferir con el funcionamiento del PC. Estos registros se conocen como "Contadores de Datos" o apuntadores de datos. El número de bits que componen el DC es igual al número de bits del PC, para que se pueda direccionar a toda la memoria. Uno de estos es el contador de direcciones, el cual es un dispositivo de almacenamiento temporal utilizado para retener la dirección de la localidad a ser accesada durante la lectura o escritura de un dato en la memoria de datos.

#### 1.4.3.4 REGISTRO DE INSTRUCCION (IR)

Este registro se utiliza para recibir el código, de la instrucción que la CPU obtiene durante el ciclo de búsqueda de la instrucción (FETCH). El cual se obtiene de la localidad de memoria direccionada por el PC.

La CPU cuenta con un solo registro de instrucción. Este alimenta al decodificador de instrucciones, el cual decodifica el código y alimenta a la unidad de control para que realice las operaciones necesarias para la ejecución de la instrucción.

#### 1.4.3.5 REGISTRO DE ESTADOS

Este es un registro utilizado por la ALU para indicar ciertas condiciones resultado de las operaciones realizadas; mismas que pueden requerir de un manejo especial. En un programa estas condiciones reflejan puntos de decisión, que pueden resultar en saltos dentro del programa o a una subrutina. Cada una de estas condiciones es almacenada en un flip-flop conocido como bandera de estado, cada una de estas banderas forman parte del registro denominado "registro de estados".

La gran mayoría de microprocesadores contienen alguna de las siguientes banderas: Signo, Sobreflujo, Acarreo, Cero y Paridad.

La bandera de signo es empleada para indicar el signo del operando, este siempre ocupa el bit mas significativo de la palabra.

La bandera de sobreflujo es utilizada para indicar que en el resultado de la suma de dos operandos se obtuvo un acarreo lo cual ocasiona que la cantidad resultante sea un bit mayor que el espacio de bits permitidos en la CPU.

La bandera de acarreo es utilizada para indicar la existencia de un acarreo en el resultado de operaciones.

La bandera de cero es utilizada para indicar que el resultado de ciertas operaciones es cero. Cuando esto sucede esta bandera es colocada en el estado uno.

La bandera de paridad es utilizada durante operaciones de transferencia de datos, sirve para indicar cuando ha ocurrido un error en la transmisión.

## **1.5 FUNCIONAMIENTO DE LA CPU**

A continuación se describe la forma en que son ejecutadas cada una de las operaciones por la unidad de procesamiento central (CPU).

### **1.5.1 RELOJ DE UNA CPU**

Todas las operaciones realizadas por la CPU están sincronizadas con y por una señal de reloj, la cual determina la velocidad con la que un microprocesador realiza sus operaciones. Muchas CPUs contienen el circuito para la generación de la señal de reloj dentro del mismo circuito integrado. En estos casos un cristal externo u otro elemento resonante es necesario para determinar la frecuencia de este reloj (oscilador).

Cuando el cristal es externo al circuito integrado que tiene la lógica de la CPU, la lógica que genera las señales de reloj, que denominaremos  $\emptyset$ , puede o no estar implementado en el mismo circuito integrado de la CPU. Sin embargo, dependiendo del diseño de la CPU, las señales pueden ser simples como se ilustra en la figura A1.4 o una combinación mas compleja como se ilustra en la figura A1.5, en la que se ilustra una combinación de dos señales de reloj, las que se identifican con los símbolos  $\emptyset 1$  y  $\emptyset 2$ .



Figura A1.4 Señal de reloj simple.

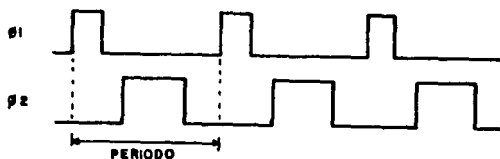


Figura A1.5 Señales de reloj combinadas.

### 1.5.2 CICLOS DE INSTRUCCION

La CPU es un dispositivo dinámico ya que sus elementos de almacenamiento y circuitos de lógica interna requieren señales de tiempo (reloj). El ciclo de instrucción se define como el tiempo requerido para realizar la búsqueda de una instrucción en la memoria del programa, decodificarla y ejecutarla. Un ciclo de instrucción esta formado por uno o más ciclos de máquina.

Todas las operaciones en las computadoras se realizan paso a paso. En cada uno de los pasos se realiza una función específica y se conocen como "ciclos de máquina". Una instrucción puede requerir de uno o más ciclos de máquina dependiendo de la complejidad de la función implícita.

Los ciclos de máquina a su vez están formados por pequeños pasos de procesamiento, conocidos como estados (T1, T2, etc). Cada estado realiza una función específica de procesamiento y son generalmente los mismos en todos los microprocesadores. El estado utiliza un tiempo igual a un periodo de reloj del microprocesador. Cada ciclo de máquina estará formado por tres, cuatro, cinco o seis estados dependiendo de la instrucción y del microprocesador.

Existen tres ciclos de máquina básicos en los microprocesadores: Fetch (Busqueda), Leer Memoria y Escribir en Memoria.

La ejecución de las instrucciones de cualquier microprocesador se puede dividir en dos partes: Busqueda y Ejecución. El ciclo de busqueda siempre es uno de los ciclos de máquina de una instrucción y siempre es el primero de los ciclos que constituyen a una instrucción. La parte de ejecución estará constituida por uno o más ciclos de máquina.

#### 1.5.2.1 EL CICLO FETCH (BUSQUEDA)

El ciclo fetch se utiliza para leer de la memoria el código de la próxima instrucción y depositarlo en el registro de instrucción. La dirección de la localidad de memoria de donde se obtiene el código es proporcionada por el contador del programa. Algunas instrucciones de un byte requieren únicamente del ciclo fetch para obtener el código de la instrucción y ejecutarse.

Los pasos que se realizan durante un ciclo fetch son:

1. El contenido de la localidad de memoria especificada por el contador del programa se lee en el registro de instrucción, obtiene el código de la instrucción.

2. El contenido del contador del programa se incrementa en uno.
3. El código presente en el registro de instrucción pasa al decodificador de instrucciones.
4. Una vez que el código de la instrucción se encuentra en el registro de instrucción y se decodifica, se inicia una secuencia de eventos controlados por la unidad de control. Esta secuencia de eventos constituyen la parte de ejecución de la instrucción. Si se obtuvo una instrucción de un ciclo, las operaciones de la instrucción se ejecutan durante la última parte del ciclo fetch.

#### 1.5.2.2 CICLO DE LEER MEMORIA

El ciclo de leer memoria, lee el dato contenido en la localidad de memoria cuya dirección es proporcionada por el contador del programa o un contador de datos. El dato se puede cargar en cualquiera de los registros de la CPU, según se especifique en el código de la instrucción. Entre estos registros se encuentran el acumulador, el contador del programa y otros de propósito especial que tenga la CPU. La única excepción es el registro de instrucción el cual como se mencionó se carga con un código de instrucción durante el ciclo fetch. Este ciclo se utiliza para obtener datos o



direcciones.

Los pasos que se realizan durante el ciclo de leer memoria son:

1. El contenido de la localidad de memoria especificada por el contador del programa o un contador de datos se lee en el registro de la CPU especificado en el código de la instrucción que se obtuvo en el ciclo fetch.
2. Si la dirección la proporcionó el contador del programa el contenido de este se incrementa en uno. Si es una instrucción para leer dos datos utilizando el contador de datos, éste se incrementa en uno.

#### 1.5.2.3 CICLO DE ESCRIBIR EN MEMORIA

El ciclo de escribir en memoria escribe el dato contenido en uno de los registros de la CPU en la localidad de memoria apuntada por un contador o apuntador de datos. El dato se proporciona por cualquiera de los registros de propósito general o el acumulador, el cual se especifica en el código de instrucción.

Los pasos que se realizan durante el ciclo de escribir en memoria son:

1. El contenido del registro se carga en la localidad de memoria especificada por un apuntador de datos. EL registro se especifica en el código de la instrucción que se obtuvo en el ciclo fetch.
2. Si es una instrucción para escribir un dato de dos bytes, se incrementa en uno el contenido del contador de datos.

## 1.6 IMPLEMENTACION DE LAS INSTRUCCIONES

### 1.6.1 INSTRUCCIONES DE UN CICLO

Algunas instrucciones de un ciclo se obtienen de la memoria, se cargan en el registro de instrucción y se ejecutan en un solo ciclo de máquina, el ciclo fetch. Generalmente son instrucciones de un ciclo aquellas que operan sobre un operando y éste se encuentra en uno de los registros de la CPU, y aquellas en las que dos operandos se encuentran en los registros de la CPU. Por ejemplo la instrucción de incrementar en uno el contenido de un registro y la de sumar dos datos contenidos en dos registros de la CPU.

### 1.6.2 INSTRUCCIONES CON REFERENCIA A MEMORIA

Las instrucciones que hacen referencia a memoria son utilizadas para acceder una localidad de memoria en la cual se tenga almacenado un operando o bien para escribir un operando en una localidad de memoria. El microprocesador deberá acceder al menos dos veces a la memoria, dos ciclos de máquina. La primera vez para obtener el código de la instrucción, ciclo fetch y el segundo para obtener el operando, ciclo de leer memoria.

### 1.6.3 INSTRUCCIONES SIN REFERENCIA A MEMORIA

Estas son instrucciones en las cuales la CPU no hace referencia alguna a la memoria de datos, lo cual significa que durante la ejecución de las instrucciones no lee o escribe datos en la memoria. Estas instrucciones involucran la manipulación de datos internamente en la CPU.

Entre estas instrucciones se tienen las utilizadas en la transferencia entre registros, así como instrucciones que no involucran transferencia entre registros pero indican que la operación debe ser realizada sobre el contenido de algún registro y las instrucciones de transferencia entre registros que emplean más de un byte.

#### 1.6.4 INSTRUCCIONES DE SALTO

Todos los microprocesadores tiene instrucciones de salto condicional e incondicional. Por medio de estas instrucciones el microprocesador ejecuta rutinas u operaciones que son un complemento de las operaciones principales.

En los microprocesadores 8080/8085 y Z80 la instrucción de salto consiste de tres bytes. El primer byte contiene el código, el segundo los 8 bits menos significativos de la dirección a donde va a saltar y el byte tres contiene los 8 bits mas significativos de la misma dirección. En este caso el microprocesador deberá acceder tres veces a la memoria un ciclo fetch y dos de leer memoria.

## 1.7 ELEMENTOS DE DISEÑO

Durante el desarrollo de cualquier producto basado en un microprocesador se deben tomar en cuenta todos los elementos que pueden ser utilizados durante el diseño.

Todo producto diseñado en base a un microprocesador requiere básicamente de los siguientes dispositivos de hardware: El microprocesador, unidad de memoria, interfaces de entrada/salida y circuitos auxiliares para interconectar cada una de las unidades. Se requiere además de un programa (software) por medio del cual se proporcionen al microprocesador las instrucciones que le indiquen que operación realizar y la secuencia en que debe realizarse.

## 1.8 SELECCION DEL MICROPROCESADOR

El primer paso consiste en establecer que circuitos deben ser utilizados para lo cual debe hacerse una evaluación de las limitaciones bajo las cuales debe hacerse el diseño. En la selección del microprocesador a utilizar deben tomarse en cuenta ciertos factores como son:

- Costo.

- Capacidad de procesamiento.
- Flexibilidad.
- Facilidad de uso.
- Soporte del fabricante.
- Existencia de un segundo proveedor.
- Consumo de energía.
- Rango de hardware complementario.
- Limitaciones ambientales especiales.

Dependiendo de la aplicación en particular, se consideran de mayor o menor importancia ciertos factores, ya que en algunas aplicaciones puede ser más importante un bajo consumo de energía que una mayor velocidad de procesamiento. Las características del microprocesador empleado en el diseño del lector de memorias son descritas en el ANEXO II.

## 1.9 DISPOSITIVOS DE MEMORIA

Los dispositivos de memoria son fundamentalmente dispositivos lógicos con la capacidad de almacenar una gran cantidad de dígitos binarios o bits (unos y ceros). Sin importar el material o formato de la memoria, el sistema numérico binario es el utilizado para almacenar información en los dispositivos de memoria. Cada localidad de memoria (un bit de información) tiene uno de dos posibles valores, a los que se hace referencia como uno y cero, alto y bajo, cierto y falso o cualesquier dos nombres mutuamente excluyentes.

Debido a que un solo bit almacenado en memoria contiene una cantidad limitada de información, para almacenar información en dispositivos de memoria generalmente se hace una agrupación de bits, a la que se le conoce como palabra. La longitud de palabra en memorias utilizadas en microprocesadores es la misma que la longitud de palabra que se maneja en el microprocesador. Longitudes de palabra común son 4, 8, 12 y 16 bits.

A la palabra de 8 bits se le conoce como BYTE y a las palabras de 4 bits con el de NIBBLE. Las palabras con más de 8 bits también hacen referencia al byte.

En una palabra generalmente se asigna a cada bit una posición determinada, el bit menos significativo ocupa la posición mas a la derecha, mientras que la posición más a la izquierda es ocupada por el bit más significativo de la palabra.

Cuando se requiere almacenar una serie de palabras en un espacio de memoria se tiene lo que se conoce como banco de memoria. Otro concepto involucrado en los dispositivos de memoria es la dirección. Desde el punto de vista del hardware un chip de memoria cuenta con  $n$  líneas de dirección, las cuales cuando son aplicadas, especifican exactamente que localidad en la memoria (banco) debe ser escrita o leída. La dirección es una de las entradas de la memoria; el contenido de la localidad de memoria es la salida si se esta leyendo o una entrada si se esta escribiendo.

La memoria es la parte en donde se almacenan las instrucciones (codificadas en binario) y los datos de un programa.

El desarrollo de las memorias de semiconductores, con la habilidad de almacenar varios miles de bits dentro de un circuito integrado, ha modificado la tecnología de construcción de las memorias de las computadoras. Entre las características que proporcionan las memorias de semiconductores se pueden mencionar las siguientes:



- Las memorias de semiconductores han hecho posible la construcción de memorias más rápidas, más compactas y de bajo consumo de potencia.
- Han causado un cambio completo en la arquitectura de las computadoras.
- Su versatilidad, y compatibilidad con otros circuitos integrados los han hecho útiles para nuevos propósitos, completamente fuera de la tecnología de las computadoras. Se utilizan para almacenar palabras de control en máquinas de control numérico.

Las microcomputadoras usan memorias de semiconductores de alta escala de integración (LSI) como memoria propia y consisten básicamente de arreglos de "compuertas" que pueden estar o no conduciendo.

Existen dos operaciones básicas que son realizadas en una memoria, estas son: lectura y escritura. Escribir un bit consiste en almacenar un 1 ó 0 en una localidad específica. Leer una localidad es obtener el bit 1 ó 0 que se almacenó con una función de escribir.

Las memorias utilizadas en microcomputadoras son conocidas también como chips de memoria. Estas son fabricadas con tecnología bipolar y MOS/CMOS.

### 1.9.1 TIPOS DE MEMORIAS

#### 1.9.1.1 MEMORIAS DE DIRECCIONAMIENTO SECUENCIAL

Estas son memorias en las que es necesario pasar a través de todas las direcciones anteriores antes de direccionar la localidad deseada. Generalmente son no volátiles y de lectura no destructiva, el contenido de estas puede leerse sin ocasionar cambios en el mismo, la escritura en las mismas se realiza en forma secuencial. Las cintas magnéticas son ejemplo de este tipo.

#### 1.9.1.2 MEMORIAS DE ACCESO ALEATORIO (RAM)

Memorias en las que cualquier byte puede ser direccionado en forma aleatoria, sin tener que pasar por las direcciones anteriores a este. Las RAMs pueden ser de lectura destructiva y no volátiles tal como las de núcleos magnéticos o bien de lectura no destructiva y volátiles como algunas memorias de semiconductores. Dentro de las memorias de acceso aleatorio se tienen memorias en las que el contenido de las mismas puede únicamente ser leído, estas son las memorias de solo lectura.

#### 1.9.1.3 MEMORIAS DE SOLO LECTURA (ROM)

Memorias diseñadas para que la información de la localidad de memoria direccionada pueda únicamente ser leída, durante la operación normal del sistema. La información es

escrita en las ROM durante el proceso de fabricación de la misma. Estas son memorias no volátiles, de lectura no destructiva y de acceso aleatorio. Existen variaciones de este tipo de memoria, en las cuales es posible escribir el contenido deseado empleando procesos especiales de programación.

A continuación se describen las características de un grupo de memorias de semiconductores de propósito especial. No existe nada realmente especial en estos tipos de memorias, simplemente no caen dentro de la clasificación dada anteriormente.

#### **1.9.1.4 MEMORIAS BORRABLES ELECTRICAMENTE**

Este es un tipo de memoria que ofrece las características básicas de una memoria ROM, además de ciertas características aunque limitadas de las memorias RAM. En este tipo de memorias se tiene la opción de modificar o alterar una localidad determinada sin tener que alterar el contenido de otras localidades. Este tipo de memorias es conocido como memorias EAROMS (Electrically Alterable ROMs), EEROMS (Electrically Erasable ROMs), o EEPROMS (Electrically Erasable Programmable Read Only Memory).

Las diferencias principales de las memorias EAROM y RAM es que en las memorias RAM, tanto el ciclo de lectura como el de escritura tienen aproximadamente igual duración, mientras que en las EAROM el ciclo de escritura ocupa un período de tiempo más grande.

En cuanto a las memorias ROM, las EAROM no requieren de luz ultravioleta para ser borradas, ya que son reprogramadas directamente sin tener que removerlas de la tarjeta. Estas pueden ser reprogramadas en milisegundos, a diferencia de las EPROM que requieren de minutos para ser borradas y reprogramadas.

El procedimiento por medio del cual se obtiene una memoria eléctricamente reprogramable es utilizando la celda flotax (Floating gate tunnel oxide) FET mostrada en la figura A1.2. Para cargar la compuerta flotante, la compuerta de programación se hace positiva y el drain es aterrizado. El túnel de electrones pasa a través de la delgada capa de óxido desde el drain a la compuerta flotante donde son atrapados. El dispositivo flotax en una compuerta cargada alcanza un voltaje de disparo que bajo operación normal permanece desconectado.

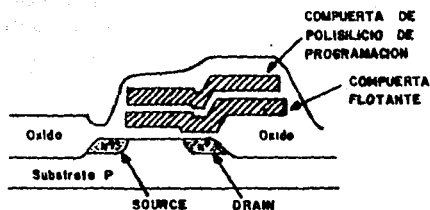


Figura A1.6 Celda Flotox

A continuación se muestra la celda básica de una EEPROM utilizando el dispositivo flotox. Para programar un arreglo de tales dispositivos, son primeramente borrados. Esto carga todas las compuertas flotantes con unos, después de lo cual son programadas las localidades deseadas.

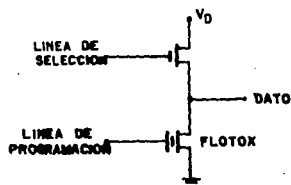


Figura A1.7 Celda EEPROM

#### 1.9.1.5 MEMORIAS LIFO (LAST IN FIRST OUT)

Este es un registro de propósito especial que es utilizado para el almacenamiento de la información de una dirección determinada durante la ejecución de un programa en un sistema de microprocesador.

Este tipo de memoria es utilizado en registros como el apuntador del stack de una unidad de procesamiento central (CPU). En este tipo de memorias el primer dato que es accesado es el último que puede ser recuperado.

#### 1.9.1.6 MEMORIAS FIFO (FIRST IN FIRST OUT)

El registro de primero en entrar-primero en salir es un tipo de memoria en el que la lectura y escritura puede realizarse en forma simultanea. En este sentido es una memoria de lectura/escritura, pero no requiere de una dirección de entrada ya sea para la lectura o la escritura. Cuando un dato de entrada es sensado en el FIFO, un contador interno, cuyas salidas sirven como la dirección escrita es incrementado. Si el contador se inicia en cero, la primer palabra de entrada es colocada en la localidad cero, y asi sucesivamente. Un contador separado es usado para las direcciones de lectura, si este se inicia en cero, la primer palabra de datos es leida de la localidad cero.

### **1.9.1.7 ARREGLOS LOGICOS DE CAMPO PROGRAMABLE (FPLA)**

En realidad un arreglo lógico programable (PLA) no es una memoria del todo, ya que como su nombre lo indica, es un arreglo lógico de compuertas AND para dar el producto de los terminos, y compuertas OR para proveer la salida. Este es especificado como una tabla de programa digamos 16x48x8; donde 16 indican las variables de entrada, 48 el número posible de términos producto y 8 las líneas de salida.

Al igual que una ROM, un PLA puede ser obtenido programado de fabrica, o bien una versión de campo programable (FPLA). La aplicación de estos se tiene en las mismas áreas de las memorias ROM. Por ejemplo si consideramos que las entradas de un PLA corresponden a las direcciones de una ROM, un PLA de 16 entradas puede tomar el lugar de una ROM de 64k. Obviamente la ROM de 64k tiene 4k de palabras de memoria, mientras que el PLA tiene unicamente el equivalente a 48 palabras de memoria.

### **1.9.2 MEMORIAS UTILIZADAS CON MICROPROCESADORES**

Básicamente son dos los tipos de memorias utilizados, estas son las memorias de acceso aleatorio (RAM) y las memorias de solo lectura (ROM).

No existe una buena razón por la cual a las memorias RAM, y no también a las ROM se les llame como memorias de acceso aleatorio. Ya que tanto las memorias ROM como RAM son de acceso aleatorio. Sin embargo, la terminología común refiere a las memorias ROM como de solo lectura y a las RAM como de acceso aleatorio.

#### 1.9.2.1 MEMORIA DE SOLO LECTURA (ROM)

Una memoria ROM (Read Only Memory), es aquella cuyo contenido no puede ser alterado durante la operación normal, ya que esta es una memoria permanente, es decir, no volátil la cual puede unicamente ser leída. El contenido de este tipo de memoria es programado durante el proceso de fabricación de la misma.

La capacidad de cualquier memoria, una ROM especificamente esta dada en términos del número de palabras o bits que puede almacenar. Por ejemplo una memoria ROM especificada como de 4K (1Kx4), tiene 1024 palabras de 4 bits. Esta especificación también nos dice el número de líneas de dirección que esta requiere, así como el número de líneas de salida.

Internamente la ROM esta constituida por un arreglo cuadrado o casi cuadrado. La memoria de 1Kx4 tiene un arreglo de 64 columnas y 64 renglones (64 x 64 =4096). Esta memoria



cuenta con 10 líneas de dirección de las cuales 4 son utilizadas para direccionar las columnas y 6 para direccionar a los renglones. Ver figura A1.8.

Existen memorias cuya longitud de palabra no puede ser obtenida en un solo circuito integrado, por lo que se emplean varios chips conectados de forma que se pueda obtener la capacidad de la memoria deseada. Este procedimiento puede ser utilizado para obtener una memoria de cualquier tamaño.

Existen dos variaciones de las memorias ROM que permiten mas versatilidad al sistema de microcomputador. Estas son las memorias PROM y EPROM.

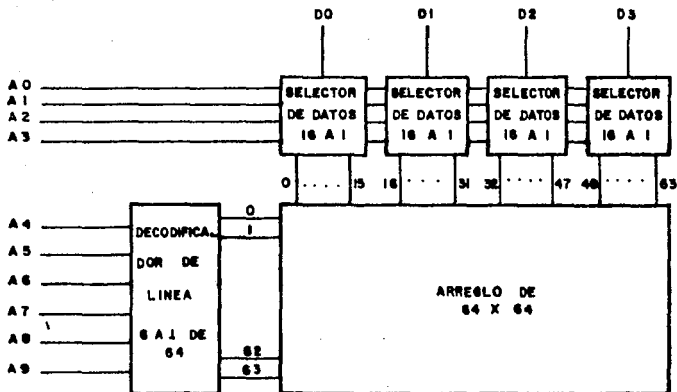


Figura A1.8 Organización interna de una ROM de 1Kx4

### 1.9.2.2 MEMORIAS DE SOLO LECTURA DE CAMPO PROGRAMABLE (PROM)

En muchas aplicaciones el volumen de producción de un equipo no garantiza el costo de fabricación de una máscara para obtener una memoria ROM. En tales casos se emplea una memoria ROM que puede ser programada por el usuario una sola vez. Esta es una memoria PROM (PROGRAMMABLE READ ONLY MEMORY). Durante la programación de este tipo de memorias los eslabones fusibles de cada localidad de memoria son quemados (para obtener circuitos abiertos) por la aplicación de voltajes específicos durante un tiempo determinado. El arreglo de la memoria es construido utilizando una matriz de fusibles, ocasionando que todas las salidas sean bajas (o todas altas). Dependiendo del circuito de PROM, el fabricante especifica los voltajes exactos y la duración de los mismos que deben ser aplicados para la programación de la memoria.

### 1.9.2.3 MEMORIAS DE CAMPO BORRABLE (EPROM)

Una EPROM (ERASABLE PROGRAMMABLE READ ONLY MEMORY), es una memoria que ofrece al usuario la opción de ser programada como una PROM, pero tiene la ventaja de que su contenido puede ser borrado y reprogramado fácilmente.

Este tipo de memoria es la mas utilizada para el desarrollo de prototipos o bien en aplicaciones donde se requiere realizar cambios con bastante frecuencia. Este tipo de memorias permite depurar los programas sin necesidad de utilizar varias memorias.

La celda básica de una memoria EPROM es un dispositivo de compuerta de polisilicio doble (algunas veces llamado silicio apilado). Ver figura A1.9.

Este dispositivo es similar a la compuerta FET de silicio, con la adición de la compuerta de silicio flotante la cual esta electricamente aislada por el oxido que la rodea.

Para programar una localidad dada, la compuerta FET de silicio apilado es habilitada con la aplicación de un alto voltaje (25 volts) a su compuerta.

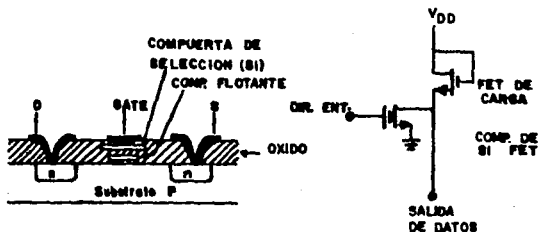


Figura A1.9 Celda de EPROM

Para borrar la información original el chip es removido de la tarjeta del circuito y expuesto a una luz ultravioleta (UV) de alta intensidad, durante un periodo de tiempo recomendado por el fabricante (15 a 20 minutos). Esta puede ser borrada y reescrita un número indefinido de veces, siempre y cuando el chip no sea sobreexpuesto a la luz ultravioleta por largos periodos de tiempo.

Para permitir la exposición del chip a los rayos ultravioleta, este viene encapsulado con una ventana de vidrio en la parte superior.

Las memorias del tipo ROM y principalmente las EPROM son ampliamente utilizadas para almacenar tablas de funciones, conversión de códigos, microprogramación, generadores de caracteres y síntesis de lógica aleatoria.

### **1.9.3 MEMORIAS RAM ESTATICAS Y DINAMICAS**

Al igual que en las memorias ROM, el tamaño de la memoria es especificada por el número total de bits que contiene (4Kx4 = 4096 bits).

Las memorias RAM son construidas utilizando las tecnologías bipolar y MOS. En las memorias RAM fabricadas con tecnología bipolar, se encuentran tipos TTL, ECL, IIL. Estas se caracterizan por un consumo de energía mayor así como una

operación más rápida comparada con el de las memorias fabricadas con tecnología MOS.

Las memorias más comunes fabricadas con el proceso MOS son las de los tipos NMOS, CMOS y CCD (Charge Coupled Devices).

Las memorias RAM pueden tener una estructura de flip-flop, a estas se les conoce como memorias estáticas, o bien tener un tipo de estructura capacitiva, en tal caso se conocen como memorias dinámicas. Las memorias estáticas conservan la información tanto tiempo como la energía esta presente, mientras que a las dinámicas se les debe refrescar cada pocos milisegundos para regenerar la carga almacenada en cada localidad.

El elemento básico de almacenamiento de las memorias estáticas NMOS, es el flip-flop mostrado en la siguiente figura A1.10.

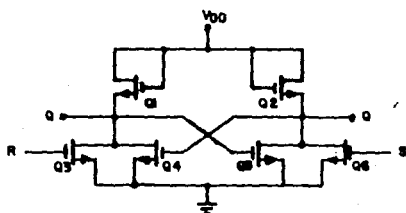


Figura A1.10 Flip-flop RS NMOS

Las memorias RAM dinámicas difieren de las memorias estáticas en dos aspectos principalmente: primero ofrecen una mayor densidad de fabricación, por lo que están disponibles en tamaños mas grandes, el segundo aspecto es que requieren de un secuenciamiento externo y lógica externa para asegurar una operación adecuada. En contraste con la estructura del FF utilizado en las memorias estáticas, el utilizado en las memorias dinámicas es significativamente mas simple. Un solo capacitor almacena la carga correspondiente a un uno, o su descarga correspondiente a un cero. Una de las áreas donde es más ampliamente utilizada es en los sistemas basados en microprocesador.

## 1.10 INTERFACES DE ENTRADA Y SALIDA

Ninguna computadora sin importar su tamaño, velocidad, capacidad de procesamiento, u otras características más sofisticadas es muy útil a menos que pueda comunicarse con el mundo exterior (equipos periféricos del sistema). Esta comunicación involucra datos que llegan a la computadora desde los periféricos y la transferencia de los datos procesados (o señales de control) de la computadora a los periféricos.

Interfazar se define como la unión de miembros de un grupo de manera que puedan funcionar en forma compatible y coordinada. Interfazar con una computadora se define como la sincronización de la transmisión de datos digitales entre la computadora y los dispositivos externos, incluyendo dispositivos periféricos de Entrada/Salida (E/S) y memorias.

Los circuitos conocidos como interfaces, son los que permiten el control de la comunicación con los dispositivos periféricos de entrada y salida. Los sistemas de computación deben contar, al menos, con un dispositivo periférico de entrada que le permita al usuario enviar información a la CPU y con un dispositivo de salida para que la CPU pueda mostrar los resultados de los procesamientos ordenados.

Tradicionalmente cuando se habla de dispositivos periféricos de entrada y de salida se hace referencia a lectoras de tarjetas, unidades de cinta magnética, teletipos, CRT, etc.. Sin embargo, circuitos integrados tales como los registros de corrimiento, contadores o memorias, también pueden considerarse como dispositivos de entrada y de salida.

#### **1.10.1 OPERACIONES DE UNA INTERFAZ**

Normalmente los dispositivos de E/S son más lentos que las CPUS. De aquí que una computadora tiene que esperar a que el dispositivo se encuentre listo para recibir o transmitir un dato nuevo. Todo dispositivo periférico debe ser conectado a los canales de transferencia (buses) de datos, control y dirección de la CPU, a través de una interfaz, la cual debe realizar las siguientes operaciones:

1. Decodificar el código de selección del dispositivo que envía la CPU y responder solo si el código es idéntico al de él.
2. Decodificar los códigos de los comandos que recibe de la CPU y generar las señales de control para efectuar las operaciones ordenadas.



3. Enviar a la CPU la información que describa el estado del dispositivo periférico.
4. Efectuar la transferencia de datos entre la CPU y el dispositivo periférico.

#### 1.10.1.1 SELECCION DEL DISPOSITIVO

La selección del dispositivo de E/S se realiza por medio de un pulso de selección de dispositivo, el cual es un pulso de sincronización generado por la interfaz para sincronizar la transferencia de datos entre la CPU y un dispositivo periférico de entrada o salida específico. El término de selección de dispositivo se asocia con el de selección de integrado o habilitar integrado que se utiliza en los circuitos de memoria. Cada interfaz debe tener un selector de código de selección que le permita generar el pulso de selección de dispositivo cada vez que la CPU envía por el bus de dirección su código de selección. Al generarse el pulso de selección de dispositivo la interfaz queda habilitada para recibir las señales de control de la CPU para el periférico.

### 1.10.1.2 DECODIFICADOR DE COMANDOS Y DE CONTROL

La interfaz debe tener un circuito para decodificar las señales de control y posteriormente indicar al dispositivo periférico el comando que debe ejecutar. El circuito decodificador depende de las funciones que puede ejecutar el periférico. Algunos periféricos realizan funciones sencillas de entrada y salida por lo que requieren decodificadores uno a uno que permitan únicamente el paso de las señales de control. Otros periféricos requieren de más información para realizar adecuadamente sus funciones por lo que las señales de control de la CPU no son suficientes. Algunos periféricos necesitan información adicional, por ejemplo: velocidad de transmisión, cantidad de bits, número de bits de parada, tipo de paridad de transmisión, etc. Para resolver este problema se utilizan las palabras de control, las cuales tienen codificados los parámetros que se requieren. La interfaz del dispositivo debe tener un registro para recibir la palabra de control y alimentar al decodificador. Las salidas del decodificador controlan las diferentes partes del dispositivo periférico.

Los estados en que se encuentra un periférico se registran en registros de un bit conocidos como "banderas de estados". Entre los estados más comunes se pueden mencionar los siguientes: Receptor listo, transmisor listo, error de paridad, error del formato del dato, etc.

Antes de transmitir o recibir un dato la CPU debe probar el estado de la bandera correspondiente a la función a realizar. Para efectuar esta prueba se puede leer la palabra de estado del periférico en el acumulador y posteriormente checar el valor de la bandera de interés.

### **1.10.1.3 REGISTRO DE DATOS**

La interfaz de un periférico debe contar con un registro de datos que le permita recibir el dato de salida de la CPU para después enviarlo al periférico de salida y de uno que le permita recibir un dato de entrada desde el periférico de entrada el cual posteriormente queda disponible a la CPU. La transferencia de los datos se realiza entre el acumulador y los registros de datos.

### **1.10.2 INTERFAZ EN UN INTEGRADO**

Los integrados que realizan las funciones de interfaz tienen dos entradas para seleccionar la interfaz e indicarle si se tiene una palabra de control o un dato. Estas entradas toman los nombres de CS (Selección del integrado) y C/D (control/dato). La entrada CS permite seleccionar el integrado y la entrada C/D permite indicarle si es una palabra de control, o un dato el que está transmitiendo. La complejidad de las interfaces, depende de varios factores, como son: versatilidad (la que se quiera alcanzar con el

dispositivo), velocidad de operación y complejidad de la función que desarrolla el dispositivo de E/S. A continuación se muestra el diagrama lógico general de una interfaz.

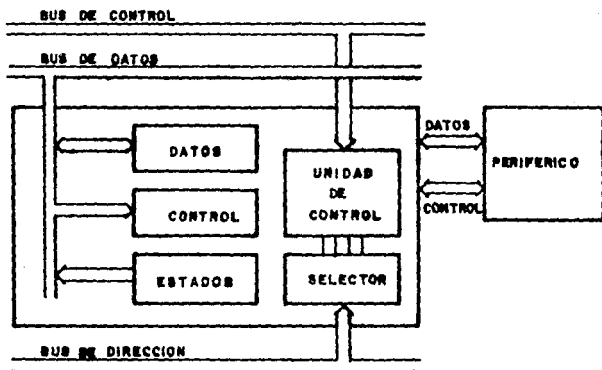


Figura A1.12 Diagrama lógico general de una interfaz.

### 1.10.3 TIPOS DE TRANSFERENCIAS DE E/S

Las principales aplicaciones de las microcomputadoras son el control de procesos, recopilación de datos (data logging) y mediciones. Para estas aplicaciones es importante que la microcomputadora sea capaz de comunicarse con los dispositivos de medición y control. El intercambio de información entre los dispositivos periféricos y la microcomputadora es controlada ya sea por el programa de la microcomputadora o por

elementos especiales del periférico.

Las transferencias de E/S de datos controlados por el programa se llaman "transferencias programadas de información" estas son iniciadas por la CPU. Las transferencias controladas por el periférico se efectúan sin la intervención del programa por medio de canales especiales de información que roban intervalos de tiempo al procesador central, cada vez que es necesario, estas transferencias son iniciadas por el dispositivo periférico.

Las transferencias programadas de datos se ordenan con las instrucciones de entrada/salida, las cuales permiten ordenar la ejecución de las siguientes tareas:

1. Mandar el comando al dispositivo.
2. Recibir información que describa el estado del periférico.
3. Leer datos de un periférico.
4. Enviar datos a un periférico.

Estas funciones se pueden efectuar en una de tres formas de transferencia programada: Incondicional, Condicional y por Interrupción de programa.

### 1.10.3.1 TRANSFERENCIA INCONDICIONAL

En este tipo de transferencia el dispositivo periférico siempre debe estar listo para aceptar datos de la CPU; también cuando un dispositivo de entrada, tenga datos para que el microprocesador los pueda leer en cualquier momento. En tales casos no existe intercambio alguno de señales de control entre la CPU y los dispositivos de E/S. Este tipo de transferencias se utiliza para procesos cuyos tiempos son conocidos y fijos.

### 1.10.3.2 TRANSFERENCIA CONDICIONAL

Las transferencias de los datos se realiza, bajo control del programa, únicamente si el periférico se encuentra listo para la transferencia. Normalmente este procedimiento se realiza en tres pasos. En el primero la CPU debe leer la información del estado del dispositivo; en el segundo se prueba este estado para ver si el dispositivo esta listo y finalmente se realiza la transferencia de datos cuando el dispositivo esta listo. Como se ilustra esquematicamente en la figura A1.13a. Las interfaces para este tipo de transferencia cuentan entre otras con las banderas de "listo para recibir" y "listo para transmitir" para indicar a la microcomputadora el estado en que se encuentra el periférico.

Esta lógica de preguntar si el periférico se encuentra listo para realizar la transferencia se utiliza con el fin de asegurar que el periférico de salida reciba el dato que le envía la microcomputadora y que el periférico de entrada tenga un dato cuando se lea su buffer de datos.

### 1.10.3.3 TRANSFERENCIA POR INTERRUPCION DE PROGRAMA

Este tipo de transferencia hace un uso más eficiente del tiempo de la microcomputadora. La transferencia se efectúa bajo el control del programa, pero la computadora no tiene que sensar repetidamente si el periférico se encuentra listo para la transferencia. La computadora puede estar ejecutando un programa "principal" y cuando un periférico está listo para recibir o transmitir un dato, solicita una interrupción a la computadora para que deje de ejecutar el programa principal y ejecute un programa especial al que comunmente se le llama rutina de servicio a interrupción. Es en este programa, en el que se ejecutan las instrucciones que logran la transferencia de o hacia el dispositivo que interrumpe. Cuando termina la ejecución de este programa, la CPU regresa al programa que estaba realizando en el momento en que ocurrió la interrupción. Esta operación se ilustra en forma esquemática en la Figura A1.13b.

La característica principal de este tipo de transferencia es que la comunicación entre la computadora y el periférico se realiza en base a una solicitud del periférico. Esto evita los tiempos de espera que desperdicia la computadora cuando se utiliza transferencia condicional mientras el periférico pasa al estado de listo.

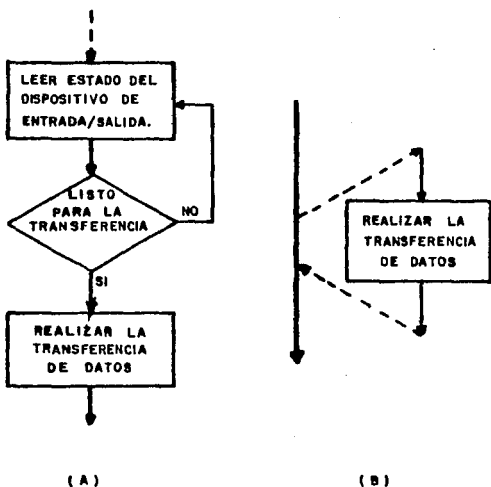


Figura A1.13 Transferencia de E/S programada.



#### 1.10.3.4 ACCESO DIRECTO A MEMORIA (DMA)

Hay ocasiones en que se deben realizar transferencias de datos entre la memoria y dispositivos externos. Un dispositivo de entrada puede tener información que debe colocarse en RAM, o bien que se deseé guardar información en cinta o disco flexible.

Existen dos métodos básicos para realizar las transferencias de bloques de datos desde o hacia la RAM. El método más obvio es realizar un programa para la transferencia, en este método la información pasa primero por la CPU y luego al dispositivo periférico. Esto sucede tanto con un dispositivo de entrada como con uno de salida. La transferencia de una sola palabra de dato por medio de un programa toma entre 5 y 10 microsegundos, la mayor parte de este tiempo la consume la CPU al efectuar la búsqueda y ejecución de las instrucciones y el paso de los datos por la CPU. Cuando los datos se transfieren directamente entre el dispositivo periférico y la memoria, la transferencia de un dato tomaría solamente un microsegundo (tiempo típico).

A este último método se le conoce como acceso directo a memoria (DMA Direct Memory Access) y es la alternativa a una transferencia por programa. Este método permite incrementar en forma substancial la velocidad de transferencia de datos en

comparación con las transferencias por programa.

La realización de transferencias por DMA involucra el empleo de un circuito especializado que sea capaz de acceder memoria para realizar operaciones de lectura y escritura, mismas que deben realizarse sin que la CPU intervenga en ello. Un circuito de este tipo debe poder efectuar lo siguiente:

1. Responder a una petición de DMA de un dispositivo periférico para generar una señal de control que provoque que el microprocesador pase a un estado de paro "HOLD STATE".
2. Una vez que el microprocesador reconoce que se encuentra en el estado de paro "HOLD ACKNOWLEDGE", la interfaz del circuito para DMA toma el mando de los buses de control y direcciones, para entonces generar las señales que permitan realizar la transferencia entre memoria y el dispositivo periférico.
3. Cuando se termina la transferencia el circuito para DMA debe remover al microprocesador del estado de paro. De esta forma el microprocesador obtiene el control de los buses del sistema y puede continuar ejecutando su programa.

#### 1.10.4 TRANSFERENCIAS DE E/S EN PARALELO

Este tipo de transferencia es utilizado cuando se requiere que la entrada al sistema sea en paralelo o que la salida hacia un periférico se realice con el mismo formato. Además de las líneas para la transferencia se cuenta con líneas para las señales de control empleadas para el intercambio entre el dispositivo de E/S y el periférico.

Se emplea cuando se requiere transferir grandes cantidades de datos entre la memoria de datos y varios dispositivos de E/S (I/O), o bien con dispositivos auxiliares de memoria. En tales casos la forma de acceso directo a memoria (DMA) es una forma atractiva para transferir grandes cantidades o bloques de datos.

#### 1.10.5 TRANSFERENCIAS DE E/S EN SERIE

Algunos dispositivos de E/S tales como el teletipo, el CRT transmiten y reciben datos en forma serie. En tal situación si una CPU va a transmitir o a recibir datos en serie deberá tener una interfaz con lógica capaz de convertir los datos serie a datos paralelos y los datos paralelos a datos serie.

La característica de un flujo de datos serie es que los datos son transmitidos y recibidos como una sola señal a través de terminales de los dispositivos. Un dispositivo de recepción interpreta a una señal serie como cualquier otra señal digital, en la siguiente figura se ilustra la representación de datos serie en la cual es fácil identificar los bits de un dato serie e interpretarlos dentro de las líneas verticales punteadas, el dispositivo receptor serie requerirá de evidencia más tangible para identificar los extremos de los bits de un dato serie.

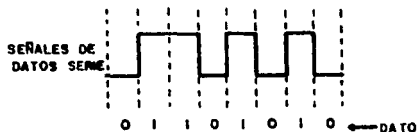


Figura A1.14 Representación de datos serie.

Para identificar el instante en el cual el dispositivo receptor debe interpretar el nivel de la señal como dato, se utiliza una señal de reloj como se ilustra en la figura A1.15.

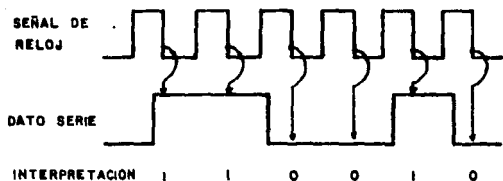


Figura A1.15 Señal de reloj para interpretar los datos.

En la figura anterior se muestra el borde de bajada de la señal de reloj como el instante en el cual la señal de datos serie se debe muestrear. También se puede utilizar el borde de subida de la señal de reloj.

La señal de datos serie se debe generar en el dispositivo transmisor antes de que pueda ser interpretada por el dispositivo receptor. Si el dispositivo receptor usa un reloj para interpretar las señales de datos serie entonces el dispositivo transmisor debe usar un reloj con la misma frecuencia para generar las señales de datos serie. En este caso se puede usar una sola señal de reloj para transmitir y recibir, siempre y cuando se transmita con el borde de bajada del pulso de reloj y se reciba con el borde de subida del próximo pulso de reloj.

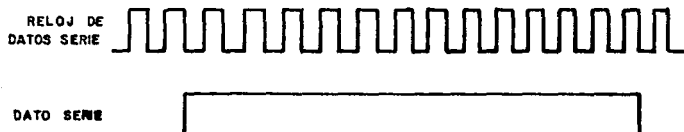
#### 1.10.5.1 RAZON DE BAUDS

El intervalo de tiempo en el cual la señal de datos serie representa un solo dígito binario relaciona directamente la velocidad a la cual los datos son transmitidos. Supongase que se están transmitiendo 110 dígitos por segundo; cada dígito serie durará entonces:  $1/110=9091$  microsegundos.

Sin embargo, la duración de un dígito en un flujo de datos serie no es la forma en la cual se mide la transferencia de los datos serie; en su lugar, se miden "bits por segundo" y se refiere a este número como la RAZON DE BAUDS. Cuando se transmiten 110 dígitos por segundo, esto equivale a una razón de bauds de 110.

Los sistemas de microprocesadores requieren de una señal de reloj para dar los tiempos para la ejecución de las instrucciones dentro de la CPU y de una señal de reloj para determinar la velocidad de transferencia de los datos serie.

El reloj serie de E/S no necesariamente tiene que pulsar a la misma velocidad exacta a la de transmisión de los bits aunque frecuentemente lo hace. Es bastante común que la velocidad del reloj sea 16 veces la velocidad de bauds, siendo también frecuente una velocidad de reloj de 64 veces la velocidad de bauds. Ver figura A1.16.



**Fig. A1.16 Reloj de 16 veces la velocidad de bauds.**

La razón de tener relojes de  $\times 16$  y  $\times 64$  es para llegar tan cerca como sea posible al centro del intervalo de tiempo de cada dígito cuando se muestrean señales de datos serie. El hecho de que los datos serie necesitan de una señal de reloj no significa que los dispositivos de E/S serie requieren de 2 líneas de señal.

Cuando se usan señales de reloj de  $\times 16$  y  $\times 64$ , el reloj de recepción puede estar uno o dos pulsos fuera de fase con el reloj de transmisión y no habrá ningún error de interpretación. Simplemente el punto de muestreo estará corrido del centro pero no existirá error en la lectura. Es posible también definir una secuencia de bits especiales de sincronización y fijar las reglas para indicar que cada flujo de datos serie debe ser precedido de este patrón de sincronización. Este conjunto de reglas se conoce como el "PROTOCOLO DE COMUNICACION".

Cada enlace de datos serie de E/S debe tener un protocolo de comunicación, puesto que los datos serie deben ser completamente autodefinidos.

#### 1.10.5.2 DETECCION DE ERROR

Cuando se transmiten datos serie sobre líneas telefónicas o directamente entre un dispositivo receptor y un transmisor se debe checar por errores de transmisión. Algunas veces durante la transmisión se cuelan señales espúreas de datos en las líneas de datos serie, por lo que el dispositivo receptor debe tener algún medio para determinar cuando existen errores dentro de los datos.

En un nivel primitivo el bit de paridad hace este trabajo, este bit se pone o se limpia para asegurar que el número total de bits 1 en la unidad de datos sea par o impar. Entonces si se recibe un número con la cantidad de bits 1 diferente a la especificada, indicará que existe un error.

Otra técnica empleada para checar error en la transmisión es agregar un carácter de redundancia cíclica a continuación de los segmentos de flujo de datos. El carácter de redundancia cíclica es un número que se crea al dividir el segmento de datos transmitidos por un polinomio fijo. El receptor recibe el flujo de datos y el carácter de redundancia cíclica, entonces, el receptor divide el flujo de datos



recibidos por el polinomio fijo para generar otra vez el caracter de redundancia cíclica. Si el resultado no es igual al caracter de redundancia cíclica que acompaña al dato, entonces existe un error.

El caracter de redundancia cíclica es un método simple de detectar errores en las transmisiones, existen métodos para detectar errores y determinar efectivamente cual es el error de tal manera que pueda corregirse.

#### **1.10.6 PROTOCOLO DE ENTRADA/SALIDA SERIE**

##### **1.10.6.1 TRANSFERENCIA SINCRONA DE DATOS EN SERIE**

En forma general el protocolo de comunicación de datos serie se puede dividir en categorías síncrona y asíncrona.

La característica principal de la transferencia de datos serie síncrona es que los datos se conforman exactamente a una señal de reloj. Habiendo establecido una vez la velocidad de bauds de transferencia, el dispositivo transmisor debe transmitir un bit de datos en cada pulso de reloj; por lo tanto el dispositivo receptor debe conocer inicialmente como interpretar la señal de dato serie.

En el protocolo síncrono se debe definir la longitud de las unidades individuales de datos y se debe proporcionar al dispositivo receptor con alguna forma de sincronizar los extremos de la unidad de datos. El carácter SYNC se usa para este propósito. Este carácter es una palabra fija previamente definida. Cada flujo de datos comienza con uno o dos caracteres SYNC.

La unidad de datos de un flujo de datos serie síncrono, generalmente consiste de bits de datos sin paridad; pero puede tener un bit de paridad.

Un dispositivo receptor entra en modo "hunt" mientras espera a que comiencen a llegar los datos serie síncronos. Durante esta espera el receptor lee continuamente los datos serie que recibe tratando de encontrar en el flujo serie de datos el patrón estándar SYNC. Si su protocolo llama por un solo carácter SYNC, entonces el dispositivo receptor comenzará a interpretar los datos en cuanto encuentre un carácter SYNC. El protocolo también puede esperar por dos caracteres SYNC iniciales, en cuyo caso el dispositivo receptor no comenzará a interpretar para la CPU los datos que recibe hasta que haya encontrado 2 caracteres SYNC secuenciales.

La transmisión de datos síncrona requiere que el dispositivo transmisor mande los datos continuamente. Si el dispositivo transmisor no tiene datos listos para mandar debe meter relleno con caracteres SYNC hasta que el próximo

carácter real este listo para transmitir.

#### 1.10.6.2 TRANSFERENCIA ASINCRONA DE DATOS EN SERIE

En la transferencia de datos serie en forma asíncrona, el dispositivo transmisor enviará una señal conocida como "marca" mientras no tenga un dato para transmitir. Para indicar que comenzará a transmitir un dato válido el transmisor envía un bit 0 (cero), el cual se conoce como señal o bit de arranque. Después del bit de arranque el transmisor envía un dato compuesto de una cantidad predefinida de bits. Para indicar que terminó la transmisión de un dato el transmisor envía una señal conocida como "señal de parada". Esta señal de parada puede consistir de uno, uno y medio o dos bits con nivel alto. Además de estas señales puede ser empleado un bit de paridad si se requiere detectar cuando existe un error en la transmisión.

Cada dato en la transmisión asíncrona tiene el siguiente formato: Señal de arranque, Dato, Bit de paridad (opcional) y señal de parada.

La transmisión de información se realiza en base a datos de 8 bits, pero pueden ser de 5, 6, ó 7 bits.

Durante la transferencia de datos asíncronos el dispositivo receptor tiene un medio adicional de checar error por transmisión. El primer dígito binario de cada unidad de datos debe ser un cero representando el bit de arranque y los últimos dos dígitos binarios de la unidad de datos pueden ser uno representando los bits de parada. Si el dispositivo receptor no detecta los bits adecuados de arranque y de parada para cualquier unidad de datos, entonces, reportará un error de formato.

#### 1.10.7 REQUISITOS DE UN DISPOSITIVO INTERFAZ DE COMUNICACIONES

La interfaz de un dispositivo de entrada/salida de comunicación serie puede considerarse formada por tres interfaces: una para comunicarse con la CPU de la microcomputadora, una para la E/S serie externa síncrona y otra para la E/S externa asíncrona. Cada interfaz tendrá líneas de datos y señales de control. Para la interfaz de E/S serie las señales se pueden agrupar en controles generales y controles del modem. Los controles generales se aplican a cualquier lógica externa, mientras que los controles del Modem llenan las necesidades específicas de los estándar de la industria de Modems. La interfaz de comunicaciones 8251 empleada para interfazar el lector de memorias y la microcomputadora cuenta con las terminales adecuadas para realizar el intercambio de las señales en forma síncrona y

asíncrona, así como para el manejo de las señales de control empleadas cuando se utiliza un modem.

El dispositivo de entrada/salida para comunicación serie debe cumplir con algunas funciones lógicas como son:

1. Bus de datos.
2. Alimentación y reloj.
3. Selección de integrado.
4. Control de transferencia.
5. Terminales para recepción y transmisión.
6. Señales de control de E/S serie.

#### 1.10.7.1 BUS DE DATOS, ALIMENTACION Y RELOJ

El dispositivo de interfaz debe contar con terminales para conectarse al bus de datos del sistema para la entrada y salida de datos en paralelo. La transferencia se realiza por medio de un buffer de datos, como se ilustra en la figura A1.17.

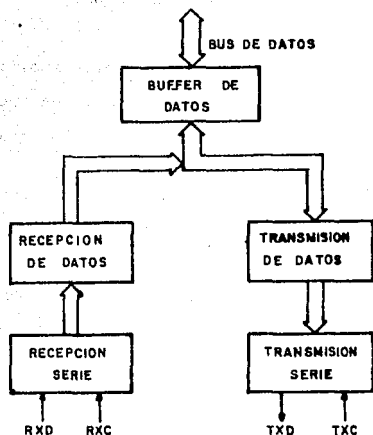


Figura A1.17 Registros de una interfaz serie.

El dispositivo necesita de terminales en cantidad suficiente para alimentación, también una entrada para señales de reloj, estas señales de reloj no son para la transmisión y recepción de datos, sino únicamente para generar los tiempos internos del dispositivo.

### 1.10.7.2 SELECCION Y CONTROL DE TRANSFERENCIA

Durante la transferencia de datos el dispositivo interfaz se considera formado de dos puertos: un puerto para la transferencia de datos y otro para la transferencia de palabras de control y de estado.

Para lograr la selección adecuada para la transferencia de los datos se utilizan generalmente dos entradas: CS la cual permite seleccionar al dispositivo y la señal C/D que permite indicar el tipo de dato durante la transferencia.

### 1.10.7.3 TERMINALES DE TRANSMISION Y RECEPCION

A las terminales de transmisión y recepción se les da los nombres de TxD (Transmit data) y RxD (Receive data) respectivamente. Las velocidades de transmisión y recepción se proporcionan en dos entradas conocidas como TxC (Transmit Clock) y RxC (Receive clock). Estas señales se pueden alimentar de la señal de reloj del sistema pasándolas por circuitos divisores o tener su propia fuente de reloj.

Normalmente se utiliza la misma velocidad para transmitir que para recibir por lo que ambas señales se alimentan de la misma fuente de reloj. Durante la recepción leerán el nivel de la línea RxD en el buffer de recepción serie. Cuando el

buffer de recepción serie contenga la cantidad de bits especificados para formar un dato, su contenido se envía al buffer de recepción de datos. Una vez hecho esto, se comienza a cargar otra vez el buffer de recepción serie.

Durante la transmisión, una vez que la microcomputadora ha cargado un dato en el buffer de transmisión de datos éste pasa al buffer de transmisión serie. Los bits del buffer de transmisión serie se envían en orden ascendente comenzando con el bit 0. Tan pronto como el contenido del buffer de transmisión de datos se deposita en el buffer de transmisión serie, se encuentra listo para recibir otro dato de salida el cual enviará al buffer de transmisión serie en el momento en que éste termine de enviar el último dato serie. Ver figura A1.17.

#### **1.10.7.4 SEÑALES DE CONTROL DE E/S EN SERIE**

El buffer del bus de datos no se puede usar simultáneamente para recibir bytes de datos ensamblados del registro de recepción serie y para transmitir bytes de datos para desensamblar en el registro de transmisión. A continuación se describen las señales de control que deben estar presentes para coordinar la transmisión y recepción de los datos serie.



La lógica de transmisión necesita de dos señales de control; una para indicar que el buffer de transmisión serie esta vacío y otra para indicar que el buffer de transmisión de datos está listo para recibir otro byte de datos. Estas dos señales se llaman TE (Transmit empty) y TRDY (Transmit ready). Cuando los datos serie se estan transmitiendo, en modo asíncrono, TE tendrá nivel bajo mientras la salida TxD esta transmitiendo el dato del buffer de transmisión serie; sin embargo, TRDY será bajo para indicar que el buffer de transmisión de datos se encuentra listo para recibir otro byte de datos, aún cuando un dato está actualmente transmitiendose.

La lógica de recepción usa únicamente la señal RRDY (Receiver ready). Esta señal dice a la CPU que se ha cargado un byte de datos en el buffer de datos y puede leerse.

#### 1.10.7.5 SEÑALES DE CONTROL DEL MODEM

A continuación se describen las señales de control del modem, con que debe contar una interfaz de comunicación; estas señales son empleadas cuando se hace uso de un modem para realizar el intercambio de señales.

1. DSR (Data set ready). El modem envía esta entrada a nivel bajo cuando está listo para recibir datos y a nivel alto en cualquier otro momento. Esto permite probar el estado de la lógica externa antes de intentar comunicarse con

ella.

2. DTR (Data Terminal Ready). Esta señal de control es la equivalente en los periféricos de E/S serie de la señal DSR y lo envía la interfaz E/S indicándole a la lógica externa que está listo para comunicarse. Bajo control de programa se puede poner esta señal en alto para inhibir todas las operaciones de E/S o ponerla en bajo para iniciar las operaciones de E/S.
3. RTS (Request To Send). En duplex completo esta salida la utiliza el periférico para indicar que está en el modo de transmisión y en medio duplex para controlar la dirección de la transmisión de los datos del periférico.
4. CTS (Clear To Send). Esta señal la utiliza el modem para indicar que está listo para transmitir. En duplex completo CTS alimenta a la salida RTS del modem; en un enlace medio duplex, el modem que recibe la señal RTS envía de regreso la señal CTS.

## 1.11 PROGRAMACION DEL MICROPROCESADOR

El elemento básico del software de un sistema que incluye un microprocesador, es el lenguaje de programación. Un microprocesador realiza las acciones que le especifica su programa. El programa está formado por una secuencia de instrucciones. Una instrucción es un conjunto de bits que tienen un significado para la unidad de control del microprocesador, este puede ser el desencadenamiento de microprogramas o la ejecución de acciones sobre registros o puertos a través de un circuito cableado.

El programa bajo el cual son efectuadas cada una de las operaciones del microprocesador, puede ser desarrollado en lenguajes de alto nivel como FORTRAN, BASIC, COBOL, o bien en lenguaje ensamblador o lenguaje de máquina.

### 1.11.1 LENGUAJE DE MAQUINA

El conjunto de instrucciones válidas para un microprocesador es lo que se denomina su lenguaje de máquina. Programar en lenguaje de máquina supone por lo tanto escribir secuencias de números en binario (instrucciones), que son directamente decodificables por los circuitos de la unidad de control o interpretables por los microprogramas de la memoria de control.

Las ventajas de programar en lenguaje de máquina son casi inexistentes. Sin embargo, para microprocesadores no provistos de algún otro tipo de lenguaje o en el caso de programas muy cortos (menos de 100 instrucciones) puede ser obligado o menos costoso el programar en lenguaje de máquina.

Al programar directamente en lenguaje de máquina se tienen las siguientes dificultades.

1. Los códigos de operación son difíciles de recordar en binario; la codificación se hace pues lenta y penosa por los números en binario.
2. Las direcciones de los operandos de las instrucciones son también difíciles de recordar.
3. El problema de cargar el programa en memoria. Si no se va a hacer en ROM directamente, habrá que trasladar una página llena de unos y ceros a algún medio físico. Sea cual sea éste, el paso será lento y sujeto a numerosos errores.
4. Es difícil seguir las ejecuciones de prueba, a través de direcciones en binario.
5. El lenguaje de máquina es el que produce el mayor grado de incompatibilidad entre programas. Un programa escrito en lenguaje de máquina solo puede ser trasladado a otro microprocesador igual al primero.

6. Un programa en lenguaje de máquina está densamente codificado por lo que es casi imposible de entender.

La programación en lenguaje de máquina puede sistematizarse y mejorarse con una metodología adecuada, el escribir previamente el programa en un lenguaje simbólico constituye una gran ayuda. El empleo de lenguaje ensamblador ofrece esta ventaja.

#### 1.11.2 LENGUAJE ENSAMBLADOR

Se llama ensamblador a un lenguaje simbólico en que se pueden escribir programas para un microprocesador. El ensamblador proporciona tres grandes ayudas al programador. Le permite utilizar símbolos (nemónicos) para designar direcciones y especificar datos (constantes) en otras formas que binario puro.

En una instrucción de lenguaje ensamblador el código de operación no aparece como un campo de unos y ceros en la instrucción sino que viene indicado por un nombre que recuerda el significado de la operación. Esto hace que sean fáciles de recordar, además de que la velocidad a la que se escriben las instrucciones aumenta considerablemente y la posibilidad de error se reduce. Cada ensamblador tiene una lista prefijada de símbolos para las instrucciones. Todos los fabricantes de microprocesadores proporcionan una lista de las instrucciones

que pueden ser utilizadas durante el desarrollo de un programa, este conjunto de instrucciones forman lo que se conoce como la hoja de instrucciones.

Cuando se utiliza lenguaje ensamblador se puede asignar un nombre a una dirección utilizando este nombre como etiqueta. El programa ensamblador hace equivalentes los nombres con las direcciones. El nombre es libremente inventado por el programador y solo esta limitado en longitud. Con lo cual es posible denominar las direcciones por un nombre relacionado con el significado de su contenido. La legibilidad del programa aumenta considerablemente y el ensamblador pasa a manejar automáticamente todas las direcciones relativas.

La tercera ayuda es la especificación de constantes, es decir proporciona la posibilidad de introducir directamente datos de distintos tipos como decimal, octal, hexadecimal, carácter y punto flotante, además de proporcionar la posibilidad de algún tipo de aritmética sobre las direcciones.

Además de las ayudas directas a la codificación, la utilización de un ensamblador permite la incorporación de comentarios al texto mismo del programa. Pueden ser frases cortas explicativas o la versión en algún lenguaje de alto nivel.

### 1.11.3 EL ENSAMBLADOR

La programación en lenguaje ensamblador es mucho más fácil que la programación en lenguaje de máquina. Sin embargo, el programador requiere de una herramienta por medio de la cual convertir el programa escrito en lenguaje ensamblador a un programa en lenguaje de máquina.

Los ensambladores son programas diseñados para trasladar programas desarrollados en lenguaje ensamblador, a programas en lenguaje de máquina.

La principal función del ensamblador es convertir el programa en lenguaje ensamblador (programa fuente) en un programa en lenguaje de máquina (programa objeto), que puede ser cargado en la memoria de programa de la microcomputadora. Este es un programa encargado de convertir (ensamblar) los programas escritos en lenguaje simbólico en programas objeto en lenguaje de máquina.

Cada programa ensamblador está desarrollado para un microprocesador en particular, por lo que este debe ser utilizado únicamente para trasladar programas para este microprocesador en particular.

La operación de llevar el programa ya traducido a memoria para ser ejecutado o probado es una de las características de este programa. El ensamblador deja el programa traducido directamente en memoria o produce una salida sobre algún medio físico. Paralelamente produce un listado del programa con los comentarios, la traducción de las instrucciones y una tabla de equivalencias entre nombres y direcciones del programa.

Con la ayuda de un buen programa ensamblador, un programador puede realizar programas con un aprovechamiento casi óptimo de memoria y de la potencia del microprocesador.

#### 1.11.4 LENGUAJES DE ALTO NIVEL

La utilización de un lenguaje de alto nivel reduce los costos de programación, incrementa la confiabilidad del software producido y simplifica el mantenimiento y documentación de los programas si se compara con la utilización de lenguajes de bajo nivel (máquina o ensamblador). La utilización de lenguajes de alto nivel supone la utilización de volúmenes de memoria que son desde un 10 a un 100% mayores que los que necesitaría un programa equivalente en ensamblador.

Cuando se programa en un lenguaje de alto nivel, pasamos a manejar no ya nuestro microprocesador, con su estructura de registros y puertos sino un procesador de estructura distinta,



concebido no para ser realizado físicamente, sino para adaptarse a la solución de los problemas planteados.

Las instrucciones contienen directamente expresiones aritméticas y lógicas y los datos pueden estructurarse. El uso de lenguajes de alto nivel requiere de programas traductores bastante complejos, llamados "compiladores", para generar programas en código de máquina a partir de sentencias en lenguaje de alto nivel.

#### 1.11.5 EL COMPILADOR

Los compiladores son programas que convierten programas escritos en lenguajes de alto nivel a programas en lenguaje de máquina. A diferencia del ensamblador, el compilador genera varias instrucciones en lenguaje de máquina por cada enunciado fuente. El compilador puede ser fácilmente modificado y hacer que interface con diferentes microcomputadoras.

Además, por medio de un compilador, el usuario es capaz de formular problemas eficientemente, sin un conocimiento preciso de la arquitectura de la computadora. El principal defecto de los compiladores es que son complejos y grandes comparados con los ensambladores. Consecuentemente, requieren mayor espacio en la memoria, lo cual no siempre esta disponible en pequeñas computadoras (microcomputadoras).

**A N E X O   I I**

**MICROPROCESADOR 8748**

## ANEXO II

### MICROPROCESADOR 8748

El microprocesador 8748 forma parte de la familia MCS-48, en la cual además de la versión con EPROM se tiene una versión con ROM (8048) y una sin memoria ROM dentro del chip (8035).

Esta formado por los siguientes bloques:

- Unidad Aritmética y Lógica (ALU)
- Acumulador.
- Bandera de acarreo.
- Decodificador de instrucciones.

#### UNIDAD ARITMETICA Y LOGICA

La ALU acepta palabras de datos de 8-bits de una o dos fuentes y genera un resultado de 8 bits bajo el control del decodificador de instrucción. La ALU puede ejecutar las siguientes funciones:

- Suma con o sin acarreo.
- And, Or, Or exclusiva.
- Incrementos/Decrementos.
- Complemento de bits.
- Rotación a la izquierda o derecha.
- Separar en nibbles.

- Ajuste decimal en BCD.

Si en la operación ejecutada por la ALU resulta un valor representado por más de 8 bits, la bandera de acarreo es colocada en la palabra de estados del programa.

#### MEMORIA DE PROGRAMA

La memoria residente de programa con que cuenta, consiste de 1024 palabras de 8 bits, las cuales son direccionadas por el contador del programa. En el 8748 esta es una memoria EPROM, mientras que la versión ROM se tiene en el 8048, así como la versión en la que no se tiene memoria alguna se tiene en el 8035 el cual debe ser empleado con una memoria externa.

#### MEMORIA DE DATOS

Cuenta con una memoria residente de datos organizada en 64 palabras de 8 bits; localidades que son direccionadas indirectamente a través de cualquiera de los dos registros apuntadores de la RAM, los cuales se encuentran en las direcciones 0 y 1 del arreglo de registros. Las primeras 8 localidades de este arreglo son designadas como registros de trabajo y son directamente direccionables por varias instrucciones. Los registros de trabajo son usados en forma

eficiente por medio de la instrucción DJNZ ya que esta permite al usuario decrementar y probar el registro en una sola instrucción.

Cuando se ejecuta la instrucción de conmutar el banco de registros (SEL RB) las localidades 24-31 de la memoria RAM son designadas como los registros de trabajo en lugar de las localidades 0-7 y son entonces directamente direccionables. Este segundo banco de registros de trabajo puede ser usado como una extensión del primer banco o reservado para su uso durante el servicio a subrutinas, permitiendo que los registros del banco 0 usados en el programa principal sean "salvados" por una selección del banco de registros.

#### LÍNEAS DE ENTRADA/SALIDA

Se tienen 27 líneas de entrada/salida, estas se encuentran agrupadas como tres puertos de 8 líneas cada uno, cada línea puede ser usada como línea de entrada o de salida o bidireccional y 3 líneas de entrada de "prueba", las cuales pueden alterar la secuencia del programa cuando es probada por una instrucción de salto condicional.

Los dos primeros grupos se conocen como puerto 1 y puerto 2. Estos son de 8 bits de ancho y tienen características idénticas. Los datos escritos en estos puertos permanecen sin cambio alguno hasta que son escritos nuevamente. Como líneas

de entrada están presentes hasta que son leídas por una instrucción de lectura. Las entradas son completamente compatibles con TTL y las salidas pueden manejar una carga estandar TTL.

Las líneas de los puertos 1 y 2 son llamadas quasibidireccionales debido a su estructura especial de salida, la cual permite que cada línea pueda ser usada como entrada, salida o ambas.

El tercer grupo se conoce como BUS, este es un puerto de 8 bits el cual tiene asociado indicadores de entrada y salida, si la condición bidireccional no es necesaria este puede ser usado como un puerto de entrada o de salida. Como puerto de salida el dato escrito es retenido usando la instrucción OUTL y accesado por medio de la instrucción INS.

#### LINEAS DE PRUEBA E INT

Se tienen tres terminales que sirven como entradas las cuales son revisadas por medio de la instrucción de salto condicional, estas son INT, T0 y T1. Estas terminales permiten entradas que ocasionan bifurcaciones sin la necesidad de cargar un puerto de entrada al acumulador.

## **PALABRA DE ESTADOS DEL PROGRAMA**

Esta formada por una colección de 8 flip-flops a través de los cuales la máquina puede ser leída o escrita como un todo, están designados como se indica a continuación:

- bit 0-2 bits del apuntador del stack.
- bit 3 No es usado.
- bit 4 Selección del banco de trabajo.
- bit 5 Bit de bandera F0.
- bit 6 Bit de acarreo auxiliar (AC).
- bit 7 Bit de acarreo (C).

## **TIMER/COUNTER**

Contiene un contador, el cual permite al usuario realizar el conteo de eventos externos y la generación precisa de retardos de tiempo sin colocar una carga al procesador para estas funciones.

## **RELOJ Y CIRCUITOS DE TEMPORIZACION**

La generación de las señales de temporización esta completamente contenida en el integrado a excepción de la referencia de frecuencia la cual puede ser un cristal de cuarzo (XTAL), un circuito RC o una fuente externa de reloj. El oscilador con que cuenta es un circuito serie de alta

ganancia el cual funciona dentro de un intervalo de 1 a 6 MHz. La salida del oscilador es dividida por tres en el contador de estados para crear un reloj que define el ciclo de estado de la máquina (CLK). La señal CLK es dividida por 5 en el contador de ciclos para proveer el reloj que define el ciclo de la máquina que consiste de 5 estados de máquina. A este reloj se le llama ALE.

#### HOJA DE INSTRUCCIONES

A continuación se describe la hoja de instrucciones utilizadas en la programación del microprocesador, se incluyen las instrucciones empleadas en el programa desarrollado.

En la descripción de estas instrucciones se hace referencia al acumulador (A), BUS (BUS), Banderas de estado F0 y F1, Palabra de estados del programa (PSW), registros Rr (donde  $r=0,1,2,3,4,5,6,7$ ), puertos Pp ( $p=1,2$ ), etc.

ADD A Rr	Suma del contenido del registro r al acumulador.
ADD A, Rr	Suma contenido del registro r de la memoria de datos al acumulador.
ADD A, data	Suma el dato inmediato al acumulador.
ADDC A,Rr	Suma el contenido de la bandera de acarreo y



del registro r al acumulador.

**ADDC A,Rr** Suma el acarreo y el contenido del registro de la memoria de datos al acumulador.

**ADDC A, data** Suma el acarreo y el dato inmediato al acumulador.

**ANL A,Rr** AND lógica entre el contenido del acumulador con el registro r.

**ANL A, data** AND lógico entre el acumulador y el valor inmediato.

**ANL BUS, data** AND lógico entre el BUS y el dato inmediato.

**ANL Pp, data** AND lógico entre el puerto 1 ó 2 y el dato inmediato.

**CALL address** Llamada a una subrutina.

**CLR A** Limpiar el acumulador.

**CLR C** Limpiar el bit de acarreo.

**CLR F0 (F1)** Limpiar la bandera F0 (F1).

**CPL A** Complementa contenido del acumulador.

**CPL C** Complementa el bit de acarreo.

**CPL F0 (F1)** Complementa F0 (F1).

DEC A            Decrementa contenido del acumulador en uno.

DEC Rr           Decrementa el contenido del registro r.

DJNZ Rr, address Decrementa el contenido del registro Rr y prueba si este es igual a cero.

ENTO CLK        Habilita salida de reloj en la terminal T0 del microprocesador.

IN A,Pp         Transfiere el contenido del puerto Pp al acumulador.

INC A            Incrementa el contenido del acumulador en uno.

INC Rr           Incrementa el contenido del registro Rr en uno.

INC Rr           Incrementa el contenido de la localidad de memoria de datos direccionada por el registro Rr.

INS A,BUS       Transfiere el dato del BUS al acumulador.

JC address      Salta a la dirección indicada, si la bandera de acarreo esta puesta.

JFO address     Salta a la localidad indicada si la bandera F0 esta colocada.

JMP address     Salto a la localidad indicada, dentro del

bloque de 2K.

JNC address	Salto a la localidad indicada dentro de la página si la bandera de acarreo no esta colocada.
JNZ address	Salto a la localidad indicada si el contenido del acumulador no es cero.
JT0 address	Salto a la localidad indicada si T0 es alto.
JZ address	Salto a la localidad indicada si el contenido del acumulador es cero.
MOV A, data	Carga el dato inmediato al acumulador.
MOV A,PSW	Carga el contenido del registro de la palabra de estados (PSW) al acumulador.
MOV A,Rr	Carga el contenido del registro Rr al acumulador.
MOV Rr, data	Carga el dato inmediato al registro Rr.
MOV Rr,A	Carga el acumulador con el contenido de la localidad de memoria direccionada por el registro Rr.
ORL BUS, data	OR lógico entre el dato del BUS y el valor del dato inmediato.
ORL Pp, data	OR lógico entre el contenido del puerto 1 .

ó 2 y el dato inmediato.

OUTL BUS,A      Transfiere el contenido del acumulador al BUS.

OUTL Pp,A      Transfiere el contenido del acumulador al  
puerto 1 ó 2.

RET              Fin de una subrutina sin restablecimiento  
de PSW.

RETR             Fin de una subrutina restableciendo el estado  
de PSW.

RL A             Rotar hacia la izquierda el contenido del  
acumulador.

RLC A            Rotar hacia la izquierda el contenido del  
acumulador con acarreo.

RR A             Rotar hacia la derecha el contenido del  
acumulador.

RRC A            Rotar hacia la derecha el contenido del  
acumulador con acarreo.

SEL MBO         Selecciona el banco de memoria 0.

SEL RBO         Selecciona el banco de registros 0.

SWAP A          Divide el contenido del acumulador en nibbles.

XRL A,Rr        OR exclusiva entre el contenido del acumulador

y el contenido del registro Rr.

XRL A, data OR EXC entre el contenido del acumulador y el dato inmediato.

**A N E X O   I I I**

**NORMA EIA-RS-232C**

---

## ANEXO III

### NORMA EIA-RS-232C

La presente recomendación se aplica a los circuitos de interconexión, llamados circuitos de enlace del interfaz entre el equipo terminal de datos (ETD) y el equipo de terminación del circuito de datos (ETCD), para la transferencia de datos binarios, señales de control y de temporización y señales analógicas, según proceda.

El ETCD puede comprender convertidores de señales, generadores de temporización, regeneradores de impulsos y dispositivos de control, junto con equipo con otras funciones como protección contra errores o llamada y respuesta automáticas. Algunos de estos equipos pueden ser equipos intermedios separados, o situados en el ETD.

La presente recomendación se aplica a todos los circuitos de enlace para velocidades binarias inferiores a 20000 bits/seg. los cuales son empleados en:

1. Las comunicaciones de datos síncronas y asíncronas.
2. La transmisión de datos por líneas arrendadas, con explotación a dos o a cuatro hilos, punto a punto o multipunto.
3. La transmisión de datos por la red con conmutación, a dos o a cuatro hilos.
4. Los cables cortos de interconexión entre el ETD y el ETCD.

El interfaz entre el ETD y el ETCD se halla en un conector, que es el punto de enlace entre esas dos clases de equipos. Se pueden utilizar conectores separados para los circuitos de enlace asociados con el equipo conversor de señales u otro similar y con el equipo de llamada automática (ELLA).

#### CIRCUITO DE ENLACE EQUIVALENTE

En la siguiente figura se representa el circuito de enlace equivalente aplicable a todos los circuitos de enlace, sin hacer caso de la categoría (datos, temporización o control) a la cual pertenecen.



En este circuito equivalente no influye para nada que el generador se encuentre en el equipo de terminación del circuito de datos y la carga en el equipo terminal de datos, o inversamente.

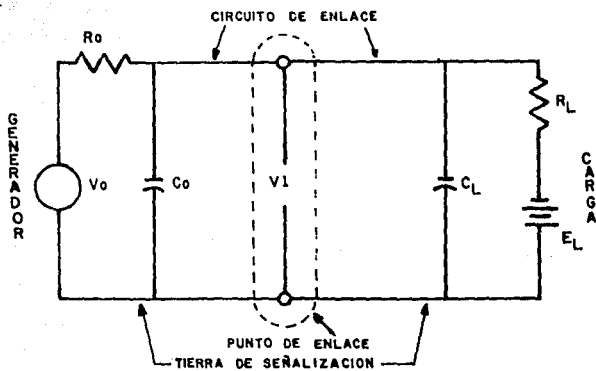


Figura A3.1 Circuito de Enlace Equivalente.

En la figura anterior se tiene que:

- $V_o$  tensión del generador en circuito abierto.
- $R_o$  resistencia efectiva total en corriente continua, asociada al generador, medida en el punto de enlace.
- $C_o$  capacidad efectiva total asociada al generador, medida en el punto de enlace.
- $V_i$  tensión en el punto de enlace con relación a la tierra de señalización o retorno común.
- $C_L$  capacidad efectiva total asociada a la carga, medida en el punto de enlace.

- RL resistencia efectiva total en corriente continua, asociada a la carga, medida en el punto de enlace.
- EL tensión de carga en circuito abierto (tensión de polarización).

La impedancia asociada al generador (carga) comprende toda la impedancia del cable del lado del generador (carga) del punto de enlace.

#### GENERADOR

El generador de un circuito de enlace ha de poder resistir las condiciones de circuito abierto y de corto circuito, entre él y cualquier otro circuito de enlace (generadores y cargas inclusive), sin que el mismo o el equipo asociado sufra daños importantes.

La tensión del generador en circuito abierto ( $V_o$ ) en cualquier circuito de enlace no debe exceder de 25 volts. La combinación de  $V_o$  y  $R_o$  se elegirá de forma que, de producirse un cortocircuito entre dos circuitos de enlace cualesquiera, la corriente resultante no exceda de medio ampere en ningún caso.

Además cuando la tensión de carga en circuito abierto (EL) sea cero, la tensión (V1) en el punto de enlace debe ser igual como mínimo a 5 volts y como máximo a 15 volts (con polaridad positiva o negativa) para cualquier resistencia de carga (RL) de valor comprendido entre 3000 y 7000 ohms.

No se especifica la capacidad en paralelo efectiva (Co) con un circuito de enlace del lado del generador; no obstante, el generador ha de poder soportar todas las capacidades de su lado (Co), más una capacidad de carga (CL) de 2500 picofarads.

En los circuitos de enlace para transmisión de datos, se considerará que el estado binario de la señal es UNO cuando la tensión (V1) en el circuito, medida en el punto de enlace, sea más negativa que -3 volts. Se considerará que la condición binaria de la señal es CERO cuando la tensión (V1) sea más positiva que +3 volts.

En el caso de los circuitos de enlace de control y de base de tiempo, se considerará que están en estado CERRADO cuando la tensión (V1) sea en ellos más positiva que +3 volts, y en estado ABIERTO cuando la tensión (V1) sea más negativa que -3 volts.

La región comprendida entre +3 volts y -3 volts se denomina región de transición. Cuando la tensión (V1) está en la región de transición, no está definido con precisión el estado de la señal o del circuito.

## **CARGA**

La impedancia del lado de la carga ( $R_L$  y  $C_L$ ) de un circuito de enlace debe tener una resistencia en corriente continua ( $R_L$ ) de valor no menor a 3000 ohms, medida con un voltaje aplicado no mayor de 25 volts, y un valor no mayor de 7000 ohms medido con un voltaje aplicado de 3 a 25 volts en magnitud.

La tensión de carga en circuito abierto ( $E_L$ ) no debe exceder de 2 volts. Mientras la capacidad efectiva en paralelo con la carga ( $C_L$ ), medida en el punto de enlace, no debe ser superior a 2500 picofarads.

## **CARACTERISTICAS MECANICAS DE LA INTERFAZ**

La interfaz entre el equipo terminal de datos y el equipo de terminación del circuito de datos está localizada en un conector el cual señala el punto de interfaz de los dos equipos. Este conector cuenta con 25 pines cada uno de los cuales tiene una función determinada como se muestra en la figura A3.2.

NUMERO PIN	CIRCUITO	DESCRIPCION
1	AA	PROTECTIVE GROUND
2	BA	TRANSMITTED DATA
3	BB	RECEIVED DATA
4	CA	REQUEST TO SEND
5	CB	CLEAR TO SEND
6	CC	DATA SET READY
7	AB	SIGNAL GROUND (COMMON RETURN)
8	CF	RECEIVED LINE SIGNAL DETECTOR
9	--	RESERVED FOR DATA SET TESTING
10	--	RESERVED FOR DATA SET TESTING
11	--	NO ASIGNADO
12	UCF	SEC. RECD. LINE SIGNAL DETECTOR
13	UCB	SEC. CLEAR TO SEND
14	UBA	SECONDARY TRANSMITTED DATA
15	UB	TRANSMISSION SIGNAL ELEMENT TIMING
16	UBD	SECONDARY RECEIVED DATA
17	UB	RECEIVER SIGNAL ELEMENT TIMING
18	--	NO ASIGNADO
19	UCB	SECONDARY REQUEST TO SEND
20	CB	DATA TERMINAL READY
21	CB	SIGNAL QUALITY DETECTOR
22	CE	RING INDICATOR
23	CB/CI	DATA SIGNAL RATE SELECTOR
24	BA	TRANSMIT SIGNAL ELEMENT TIMING
25	--	NO ASIGNADO

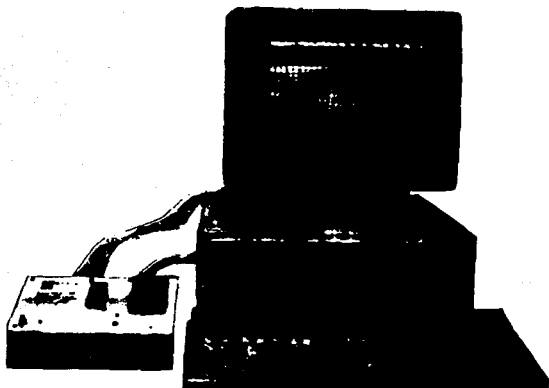
FIGURA 3.3 ASIGNACION DE PINES  
DEL CONECTOR DE LA INTERFAS.

**A N E X O    I V**

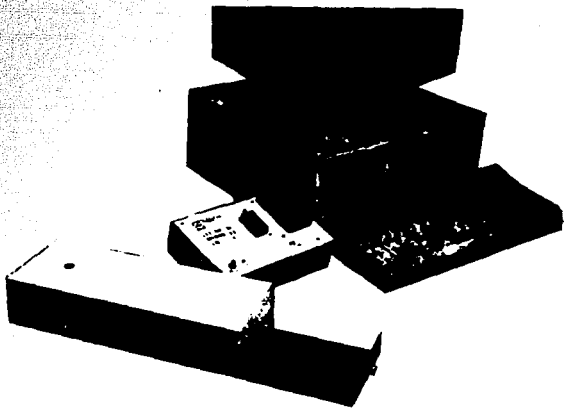
**FOTOGRAFIAS**



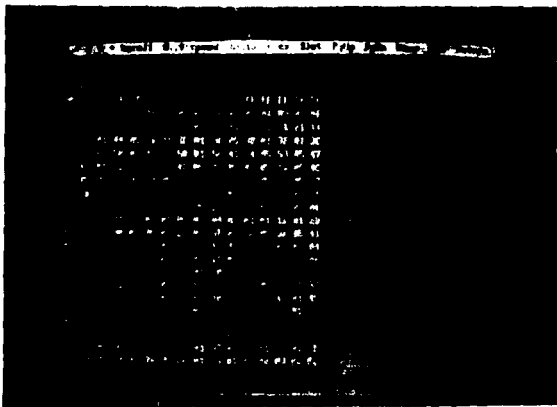
LECTOR DE MEMORIAS



MICROCOMPUTADORA Y LECTOR DE MEMORIAS



MICROCOMPUTADORA, LECTOR Y BORRADOR DE MEMORIAS



DATOS OBTENIDOS DE UNA MEMORIA



Esta Tesis fué elaborada en su  
totalidad en Los Talleres de -  
Impresos Moya, Rep. de Cuba -  
No. 99, Despacho 24.  
México 1, D.F.  
Sr. Roberto Moya Ahumada.