



ESCUELA SUPERIOR POLITECNICA DEL LITORAL  
DEPARTAMENTO DE INGENIERIA ELECTRICA

DPTO. INGENIERIA  
ELECTRICA

"DISEÑO Y CONSTRUCCION DE UN MICROCOMPU-  
TADOR CON MICROPROCESADOR MCS-40 PARA LA  
ENSEÑANZA DE MICROPROCESADORES DEL LABORA-  
TORIO DE DIGITALES DE LA ESPOL".

TESIS DE GRADO PREVIA A LA OBTENCION DEL TITULO  
DE INGENIERO EN ELECTRICIDAD

ESPECIALIDAD: ELECTRONICA

P O R

LUIS ALFREDO CELY LASSO

GUAYAQUIL - ECUADOR, AGOSTO 1980



DPTO. INGENIERIA  
ELECTRICA

"DISEÑO Y CONSTRUCCION DE UN MICROCOMPUTADOR  
CON MICROPROCESADOR MCS-40 PARA LA ENSEÑANZA  
DE MICROPROCESADORES DEL LABORATORIO DE  
DIGITALES DE LA ESPOL."

DIRECTOR DE TESIS

ING. PEDRO CARLO PAREDES

AUTOR

LUIS ALFREDO CELY LASSO

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL  
Dpto. de Ingeniería Eléctrica  
BIBLIOTECA

Inv. No. \_\_\_\_\_

## PROLOGO

Esta tesis tiene como objetivo el de presentar una serie de análisis teóricos y experimentales, llevados a efecto sobre el diseño y la construcción de un microcomputador basado en el "Microprocesador Central 4040".

En primera instancia, se hace un estudio analítico de las ventajas que existen en la construcción de un computador al usar un microprocesador, ya que éste consiste en un procesador integrado, es decir, el microprocesador está envasado en un solo chip y en él están contenidos la Unidad Lógica de Aritmética (ALU), el Contador de Programa, Los Registros de Memoria donde se almacenan las direcciones de las instrucciones (STACK), Los registros de Indices, El Acumulador y el Registro de Instrucciones, etc. Aparte de la ventaja

técnica, también existe la ventaja Económica, ya que en un chip se integran una gran cantidad de circuitos electrónicos que en un computador convencional se construirían con un gran número de componentes discretos.

que ellos son la base del funcionamiento del Microcomputador y por lo tanto un buen diseño depende del total conocimiento de los mismos.

Hay ciertos dispositivos de interfase con el Microprocesador 4040 que son programables y que tienen sus propias instrucciones. En este diseño se utilizó el circuito Integrado de Interfase P4269 que se lo programó para un teclado de 16 caracteres y un dispositivo de video de 10 caracteres en el modo de Descarga de Gas.

Todo Microcomputador debe tener un Monitor o Sistema Operativo que hace funcional al equipo desde el momento de encendido del mismo. Este Monitor hace que el Microprocesador reconozca sus periféricos y el Operador pueda trabajar en ellos en la Entrada y Salida de datos.

En el Microcomputador MCS-40 el Monitor se lo cargó en el Circuito Integrado P4702A que es una memoria no volátil y que cuando está conectado al Microprocesador, solamente permite leer información de él.

El Circuito Integrado P4702A, es de la familia de los EPROM'S (ERASABLE PROGRAM READ ONLY MEMORY- Memoria Pro-

gramable de solamente lectura y que puede ser borrada). Este circuito puede ser programado electricamente en un dispositivo propio de grabación de EPROM'S.

El diseño y construcción del dispositivo grabador de EPROM'S fue una parte en el desarrollo de esta Tesis y es un circuito que trabaja totalmente independiente del Microcomputador MCS-40, pero que su construcción fue imprescindible para poder grabar en el EPROM P4702A el Monitor del MCS-40.

El Monitor que activa el teclado y el dispositivo de Video consiste en un conjunto de instrucciones en secuencia que detectan la presión de cualquier tecla y por medio de los Registros A y B del C.I. 4269 permiten ver el caracter presionado en el dispositivo de video.

Los circuitos Integrados que se utilizaron en la construcción del Microcomputador tienen una descripción detallada de sus funciones y características en un capítulo de esta Tesis ya que lo creo necesario para un buen entendimiento de la misma.

Algunos ejemplos y Técnicas de Programación se muestran

en el último capítulo que enseñan como trabaja el Micro  
computador y la manera de programarlo.

UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN ELECTRÓNICA  
CATEDRA DE SISTEMAS DE CONTROL  
Nº DE ELEC.-012

INDICE

	Página
CAPITULO I	
INTRODUCCION	1
1.1 La Economía de usar un microcomputador	1
1.2 Las funciones de un computador	2
1.3 Operación del Sistema MCS-40	12
CAPITULO II	
EL PROCESADOR	17
2.1 Procesador Central 4040	17
2.2 Organización de Programa en la memoria del Sistema MCS-40	26
2.3 Conjunto de Instrucciones	27
2.4 Configuración de la Barra de Datos	30
CAPITULO III	
CONFIGURACION DEL SISTEMA MCS-40	35
3.1 Configuraciones del Sistema	35
3.2 Consideraciones del Sistema	39

CAPITULO IV

Página

COMPONENTES DE APOYO DE LA FAMILIA DEL MCS-40	43
4.1 "4265" Circuito Integrado Programable de propósitos generales de Entrada-Sa lidad.	43
4.2 "4269" Circuito Integrado para teclado y dispositivo de Salida Visual.	47
4.3 "4201" Circuito de Reloj	52
4.4 "4289" Interfase para memoria standard	57
4.5 "2101" (256x4) NMPS RAM	62
4.6 "4702A (256x8) PROM borrable con rayos Ultravioletas	65

CAPITULO V

DISEÑO Y CONSTRUCCION DEL CIRCUITO PARA PROGRAMAR EL PROM 4702A	70
--------------------------------------------------------------------	----

CAPITULO VI

MONITOR PARA CONTROL DE TECLADO Y SALIDA VISUAL	84
----------------------------------------------------	----



CAPITULO VII

Página

PRUEBAS SOBRE EL PROTOTIPO DEL MICROCOMPUTADOR MCS-40 CON SUS RESPECTIVOS PERIFERICOS	104
------------------------------------------------------------------------------------------	-----

CAPITULO VIII

USO DEL SISTEMA DE MICROCOMPUTADOR MCS-40 EN EL LABORATORIO	119
8.1 Técnicas de Programación	119
8.2 Técnicas Programando el 4040	129
8.3 Ejemplos	130

## I.- INTRODUCCION

### 1.1 La Economía de usar un Microcomputador.

El Microcomputador MCS-40 construido en base de la unidad central de Procesamiento 4040 es una nueva extensión en tecnología de computadores, la cual ofrece al usuario excitante posibilidades de crear nuevos productos y servicios.

Hay cinco razones básicas por las que muchos Ingenieros han empezado a usar el MCS-40 (Microcomputer Sistem). Estas son:

- a. Los costos de fabricación de productos pueden ser significativamente reducidos.
- b. Los Productos pueden entregarse al mercado más rápido, siendo suministrados a una compañía con la oportunidad de incrementar las ventas y compartir mercado.
- c. La capacidad de los productos es superada, permitiendo a los fabricantes proveer a los clientes con mejores productos, los cuales pueden frecuentemente comandar un precio más alto en

el mercado.

- d. Los costos y tiempo del desarrollo son reducidos.
- e. La confianza en el producto es incrementada, lo que con  
duce a una correspondiente reducción en costos de servi  
cio y garantía

El Microcomputador simplifica cada fase del desarrollo del productos. El primer paso como en cualquier producto de diseño de programa es identificar las varias funciones que se espera que el sistema desarrolle. Esas funciones son entonces implementadas por secuencias de instrucciones codificadas (programas) en los elementos de memoria.

Los datos y ciertos tipos de programas son almacenados en memorias de acceso aleatorio (RAM Randon Access Memory), en cambio el programa básico será almacenado en circuitos de memoria de lectura solamente ( ROM Read Only Memory).

El microprocesador desarrolla todas las funciones del sistema entregando las instrucciones en memoria, ejecutándolas y luego comunicando los resultados por medio de las vías de Entrada-Salida del Microcomputador.

El Microcomputador que es un solo chip, ejecuta la lógica

almacenada, previamente programada en solamente un elemento que es un ROM, y puede desarrollar las mismas funciones lógicas que antes requerían muchas puertas lógicas.

#### Como la Memoria Reemplaza la Lógica Clásica.

El sistema del microcomputador MCS-40 reemplaza la lógica por secuencia de programas almacenados en memoria en vez de implementar esas secuencias con puertas lógicas y flip-flops. Se puede decir que de 8 a 16 bits de memoria son el equivalente lógico de una sola puerta lógica.

Asumiendo que el tipo de circuito integrado usado actualmente contiene sobre las 10 puertas lógicas, entonces se puede concluir que la lógica puede ser almacenada en memoria con un costo muy efectivo. La Tabla I indica el número de C.I. (circuitos integrados) los cuales son reemplazados por un solo ROM. La tabla fue hecha asumiendo que de 8 a 16 bits de ROM reemplaza una puerta lógica y que un circuito integrado contiene 10 puertas lógicas.

#### Reduciendo Costos de Fabricación.

Si el costo de fabricación de un Sistema Electrónico Digi

tal es dividido por el número de circuitos integrados, uno generalmente encuentra que el sistema cuesta entre dos y seis dólares por circuito integrado. Los costos más altos son generalmente asociados con sistemas fabricados en volúmenes de 10 a 100 unidades anualmente. La tabla II presenta un análisis más detallado de la fuente de estos altos costos.

El precio de venta promedio de un circuito integrado actualmente es aproximadamente de 50¢ de dólar. El costo de inspección y prueba de estos circuitos integrados es aproximadamente 5¢ de dólar. Sin embargo, muchas compañías están ahora comprando circuitos envejecidos y probados, para sus aplicaciones con el fin de incrementar la confiabilidad del sistema, esto suma cerca de 15¢ de dólar, el costo por unidad. Simples tarjetas de circuito impreso pueden costar 25¢ de dólar por instalación de circuito integrado, pero en la mayoría de las aplicaciones para tarjetas de alta calidad, el costo aproximadamente está a 50¢ de dólar. Las tarjetas sofisticadas de multiniveles usadas en muchos circuitos de sistema de alto desarrollo cuestan más de un dólar por instalación del circuito integrado. Cuando el

cliente coloca el circuito integrado en el sócalo y entonces lo conecta por medio de alambre en la tarjeta, el costo por circuito integrado rápidamente alcanza a \$ 2 dólares. Fábricas que tienen equipos automáticos de inserción de circuitos integrados y eficientes máquinas de soldadura por medio de flujo pueden fabricar tarjetas a un costo de 3¢ de dólar por instalación de circuito integrado, aunque el precio promedio es de 5¢ de dólar. La prueba del board le suma 10¢ de dólar al costo del sistema, por lo cual el costo de un conector dividido por el número de circuitos integrados por tarjeta de circuito impreso, frecuentemente excede 5¢ de dólar. En general, resistencias, capacitores barra de potencia, etc. le suma 5¢ de dólar por instalación de circuito integrado. El sistema generalmente promedia uno ó mas alambres por instalación de circuito integrado y puestos con equipos automáticos, el costo pasa los 10¢. de dólar.

Finalmente, el costo de la fuente de potencia y empaquetamiento mecánico le suma otros 20¢ de dólar por instalación de circuito integrado.

Para determinar la economía total del costo del sistema de

fabricación, el usuario debe restar el costo de implementar un sistema equivalente con un microcomputador. En volúmenes moderados, un MCS-40 con 16.384 bits de ROM, un procesador y una cantidad mínima de RAM puede ser vendido bajo los 40 dólares. Este sistema tiene la economía entre 150 y 600 dólares del costo del sistema de fabricación.

#### Complejidad Reducida.

Debido a que el sistema MCS-40 elimina muchos circuitos integrados y por lo tanto las fallas asociadas con estos dispositivos, esto puede significativamente incrementar la confiabilidad en el sistema.

La mayoría de las fallas en un sistema digital ocurre debido a que una interconexión ha fallado. El uso de un típico circuito integrado de 16 pines introducirá aproximadamente 36 interconexiones en un sistema. Hay 16 interconexiones desde el chip al sócalo, 16 del sócalo a la tarjeta impresora, aproximadamente 2 interconexiones de la tarjeta impresora al plano trasero y 2 interconexiones del punto del plano trasero al punto del plano trasero del circuito integrado. Si un ROM elimina cincuenta circuitos integrados,

entonces este elimina aproximadamente 1.800 interconexiones.

### Conclusión.

El microcomputador MCS-40 ha reducido el costo de colocación de dispositivos en un factor de 10-ó más. Por lo tanto esto ha traído para muchos nuevos sistemas beneficios al usar computadores.

Debido a su pequeña medida y a su bajo costo, el sistema MCS-40 puede ser diseñado dentro de muchos dispositivos tales como cajas registradoras, semáforos, instrumentos, etc. donde el uso de un computador antes fue inimaginable.

### 1.2 Las Funciones de un Computador.

Esta sección incluye ciertos conceptos básicos de un computador.

Un típico sistema de un computador consiste de:

- a. Unidad de Procesamiento Central
- b. Memoria
- c. Puertas de Entrada y Salida.

La memoria de Programa sirve primariamente como un lu



gar para almacenar instrucciones, es decir, las partes de los datos codificados que dirigen las actividades de la unidad del procesador central.

Se puede definir un programa como un grupo de instrucciones lógicamente relacionadas que están almacenadas en memoria. El CPU (Central Process Unit - Unidad Central de Procesamiento) "Lee" cada instrucción de memoria en una determinada secuencia lógica y la usa para iniciar las acciones del procesamiento. Si la estructura del programa es coherente y lógica, el procesamiento produce resultados inteligibles y utilizables.

La memoria de datos es usada para almacenar los datos a ser manipulados. El CPU puede tener acceso a cualquier dato almacenado en memoria; pero a menudo la memoria no es lo suficientemente grande para almacenar el banco de datos entero, requerido por una aplicación particular. El problema puede ser resuelto suministrando al computador una o más puertas de entrada. El CPU puede direccionar esas puertas y entrar los datos contenidos allí. La adición de puertas de entradas, habilita al computador a recibir información de equipos externos (tal como una unidad de dis-

co) en altos rangos de velocidad y en grandes volúmenes.

Casi cualquier computador requiere una ó mas puertas de salida para permitir al CPU comunicar el resultado de su procesamiento al mundo exterior. La salida puede ir a una pantalla, a un dispositivo periférico que produce " copias fieles" tal como una impresora, a un periférico de almacenamiento como una unidad de cinta magnética, ó la salida puede constituir señales de control de proceso que dirige las operaciones de otro sistema, tal como una línea automatizada.

Las puertas de entrada y salida direccionables y juntas permiten al procesador ser interactivo con el mundo exterior.

El CPU unifica el sistema, este controla las funciones desarrolladas por los demás componentes. El CPU debe estar activo para entregar instrucciones de memoria, decodificando su contenido binario y ejecutarlos. Debe de estar apto para referenciar la memoria y las puertas de entrada-salida como sea necesario en la ejecución de instrucciones, el CPU estaría apto para reconocer y responder ciertas seña-

les de control externo, tales como requerimientos de interrupción y de parada.

La arquitectura de una típica unidad central de procesamiento consiste de las siguientes unidades funcionales interconectadas:

- a. Registros.
- b. Unidad Lógica de Aritmética (ALU).
- c. Circuito de Control.

Los registros son unidades de almacenamiento temporal dentro del CPU. Algunos registros tales como el Contador de Programa y el registro de Instrucciones tienen usos delicados. Otros registros tales como el Acumulador, son para propósitos de usos más generales.

Todos los procesadores tienen una Unidad lógica de Aritmética, la cual es simplemente referenciada como la ALU. La ALU puede ser considerada como una supermáquina sumadora por sus teclas comandadas automáticamente por las señales de control desarrolladas en el Decodificador de Instrucciones y el Circuito de Control. Así es como esencialmente, el primer computador digital fue concebido.

EL ALU debe contener un sumador el cual es capaz de combinar el contenido de dos registros con la lógica de la Aritmética Binaria.

Esto permite al procesador desarrollar manipulaciones Aritméticas sobre los datos que se obtienen de memoria y de sus otras puertas de entrada.

Usando sólo el sumador básico, un programador capacitado puede escribir rutinas, las cuales restarán, multiplicarán y dividirán, dando la máquina completa capacidad Aritmética.

El ALU contiene bits de banderas, las que registran ciertas condiciones que se alcanzan en el transcurso de las manipulaciones Aritméticas.

Las banderas típicamente incluyen Carry y Cero. Esto hace que sean posible saltos dentro del programa, los que son condicionalmente dependientes de los estatus de una o más banderas.

El circuito de Control es la unidad funcional primaria dentro de un CPU. Usando una entrada de reloj, el circuito

de control mantiene la propia secuencia de los eventos requeridos para cualquier trabajo de procesamiento. Después de que una instrucción es entrada y decodificada, el circuito de control efectúa las señales apropiadas (para ambas unidades, internas y externas del CPU) para iniciar la acción del procesamiento.

El circuito de control estará capacitado para responder a señales externas, tal como un requerimiento de interrupción.

Un requerimiento de interrupción hará que el circuito de control interrumpa temporalmente la ejecución del programa principal, salte a una rutina especial para atender al dispositivo que provocó la interrupción y luego automáticamente retornará al programa principal.

### 1.3 Operación del Sistema MCS-40.

El Microcomputador MCS-40, como todo sistema de computadores, contiene un procesador y una memoria. El procesador es el 4040 que esta contenido en un paquete de circuito integrado. La memoria puede ser ROM, que es la memoria del Programa o RAM que es la memoria de Datos. Además del intercambio de información entre el

procesador y la memoria, el procesador también lo hace con dispositivos externos por medio de las puertas de entrada y salida. Las facilidades del I/O (Input/Output: Entrada/Salida) del sistema MCS-40 son encontradas en dispositivos I/O separados o compartidos con elementos de memoria, permitiendo una reducción en la cantidad de circuitos integrados.

Todas las comunicaciones entre elementos del sistema MCS-40, se hacen por una barra de datos de 4 bits (D0-D3) siendo D3 el bit más significativo.

La barra de datos transfiere información tal como la dirección de un operando y los datos del I/O del procesador a la memoria y a los elementos del I/O. El procesador recibe instrucciones, operandos y datos del I/O de los otros elementos.

Todo el tráfico sobre la barra de datos, esta contenida dentro de un ciclo de instrucción. El ciclo de instrucción esta subdividido en segmentos de tiempos iguales. Cada segmento es equivalente a un periodo del reloj del sistema. La información sobre la barra de datos es alternada

de segmento a segmento. Los primeros tres segmentos de tiempo presentan una dirección de instrucciones a la memoria de 12 bits (tres grupos de 4 bits), el menos significativo es el primer nibble (4bits). Los segmentos cuarto y quinto proveen los 8 bits de la instrucción secuencialmente colocada sobre la barra de datos por Programa de Memoria. El sexto segmento es utilizado para decodificar la instrucción.

Los dos segmentos restantes son usados para la ejecución del programa.

Los Operandos y los Datos de Entrada-Salida pueden ser encontrados en la barra de datos durante este tiempo, dependiendo de la instrucción que se esté ejecutando.

Las instrucciones que requieren dos ciclos proceden como lo expuesto anteriormente.

La ejecución completa requiere 16 periodos de reloj, los cuales son divididos en dos ciclos de 8 segmentos. Durante el primer ciclo la instrucción será entregada. Esta información también puede ser una parte del operando o una dirección indirecta del registro. La segunda instrucción

siempre entregará el operando y desarrollará la ejecución.

Un gráfico de los pulsos de reloj y sincronismo se muestra en la figura 1-1.



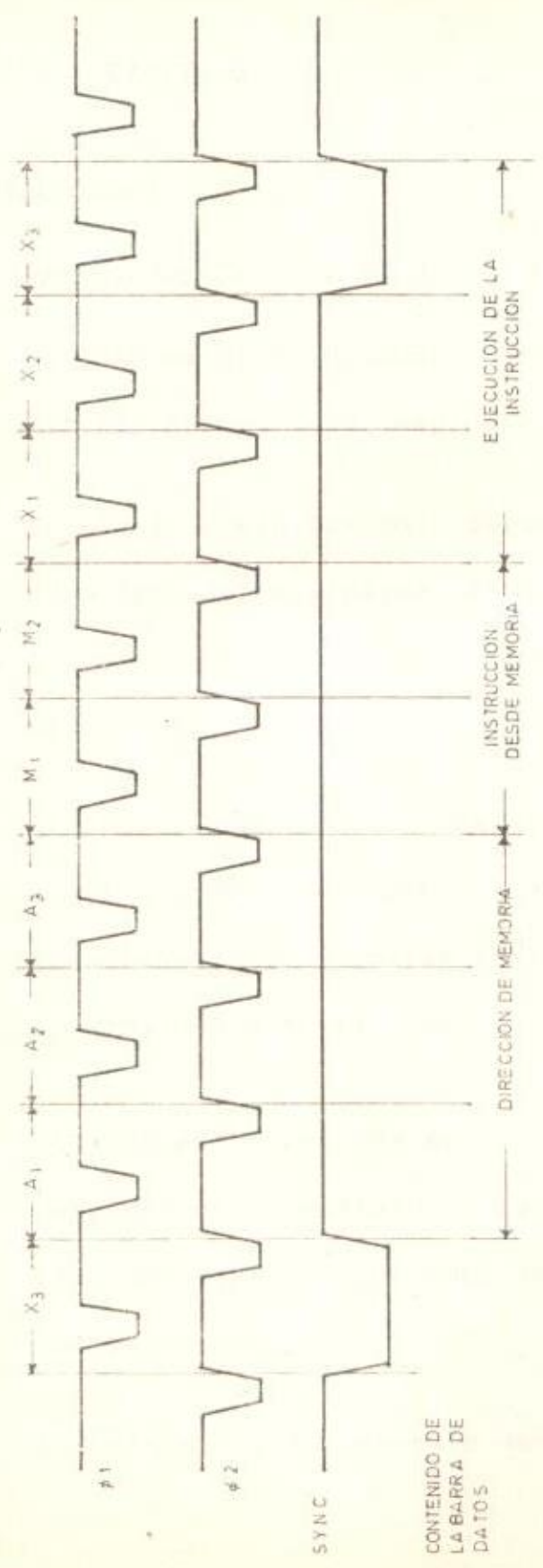


Fig. 1-1  
DIAGRAMA BASICO DE TIEMPO DEL MCS-40

## CAPITULO II

## II. EL PROCESADOR

## 2.1 Procesador Central 4040.

El 4040 es el Procesador Central MOS que esta constituido por un solo chip de 4 bits paralelos.

Esta unidad central de procesamiento esta diseñada para trabajar con otros miembros del sistema MCS-40 para formar un sistema completamente autocontenido.

El CPU se comunica con los otros miembros del conjunto a través de una barra de datos de 4 líneas y con dispositivos periféricos del usuario por medio de RAM, ROM o puertas de Entrada-Salida especiales.

El 4040 es un chip de 40 pines. La configuración de pines esta mostrada en la figura 2-1. Una amplia descripción funcional de cada pin esta dada a continuación:

Los pines 1 a 4 tienen la designación D0-D3 que

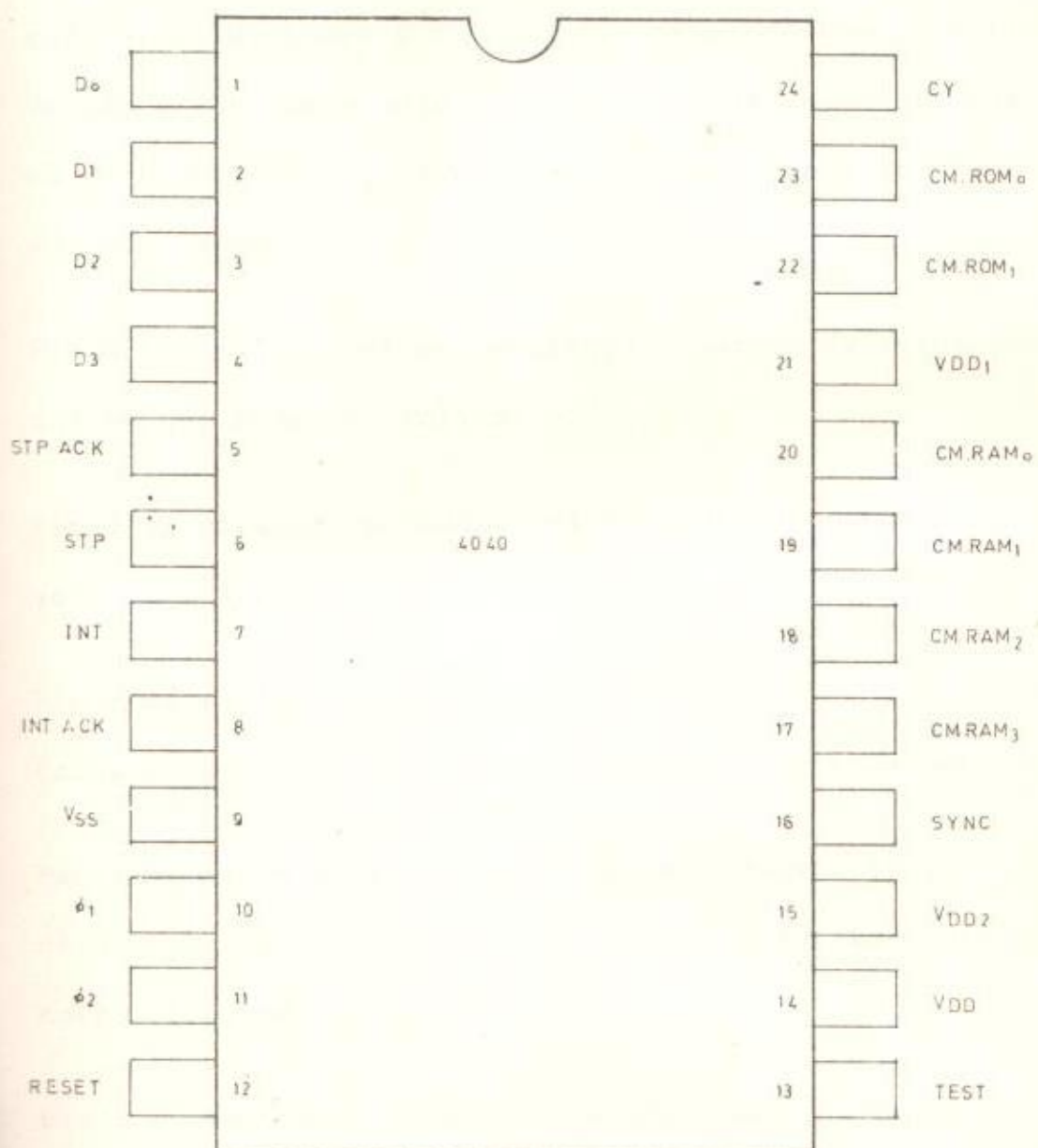


FIG. 2.1

CONFIGURACION DE PINES DEL C.I. 4040

constituye la barra de datos direccional. Toda la comunicación de direcciones y datos entre el procesador y los chips de RAM y ROM son manejados por la vía de esas 4 líneas.  $D_0$  es el bit menos significativo y corresponde a la posición A0 en el Acumulador.

Pin 5 es es Stpa (stop-acknowledge o parada de reconocimiento) en una señal de salida.

Pin 6 es el stop (parada) y es una entrada. Causa al procesador a entrar en el modo de parada.

Pin 7 es el Ini (interrupcion y es una señal de entrada. Causa al procesador a entrar en el modo de interrupción.

Pin 8 es el Inta (reconocimiento de interrupción). Recibe al comando de interrupción y previene interrupciones adicionales al procesador.

Pin 9 es Vss, es la fuente de voltaje más positivo (+5 voltios).

Pines 10 y 11 son  $\phi_1$  y  $\phi_2$  señales de reloj que determinan el tiempo del procesador.

Pin 12 es la señal de Reset (borra y comienza) que a un nivel de "1" ( $V_{dd}, -10V$ ) limpia todas las banderas y los status de los flip-flops forzando al contador de programa a cero.

Pin 13 es la señal de test (prueba). El estado lógico de esta entrada puede ser examinado con la instrucción JCN.

Pin 14 es  $V_{dd}$ , o sea el voltaje más negativo al procesador. Su valor debe ser  $V_{ss} - 15V \pm 5\%$ .

Pin 15 es  $V_{dd2}$  que es la fuente de voltaje para la salida de los registros. Puede ser variado de acuerdo a la condiciones de interfase. Esta línea controla DO-D3, CM-RAM 0-3, CM-ROM0-1 y sincronismo.

Pin 16 es SIN, o sea el sincronismo de salida, es la señal de sincronización generada por el procesador y enviada a los chips ROM y RAM. Indica el comienzo de cada ciclo de instrucción.

Pin 17 - 20, son CM-RAM0- CM-RAM3. Estas salidas son las líneas que seleccionan los chips RAMS del sistema.

Pin 21 es  $V_{dd1}$ , es la fuente de voltaje para el circuito

de tiempo.

Pines 22 - 23 son CM-ROM1- CM-ROMo. Estas líneas seleccionan los chips ROMs del sistema.

Pin 24 es CY. Es la línea del carry para la salida del buffer.

El tiempo básico del circuito esta definido por dos señales de reloj  $\phi 1$  y  $\phi 2$ . El comienzo de cada ciclo de instrucción esta definido por la señal de sincronismo.

Un ciclo de instrucción consiste de las siguientes operaciones:

1. El contenido de 12 bits del contador de programa es enviado a los chips ROMS en tres grupos de 4 bits durante A1, A2, y A3.
2. Los 8 bits de instrucción o datos del ROM direccionado son recibidos por el procesador en M1 y M2 y en este tiempo la instrucción es decodificada.
- 3.- La ejecución de la instrucción ocurre durante X1, X2 y X3. En este tiempo la información de

La instrucción o los datos deben ser enviados a las puertas de salida a los chips RAM'S. Los datos pueden ser recibidos de puertas de entrada o de chips RAM'S o los datos también pueden ser operados dentro del procesador.

La figura 2-2, es un diagrama de bloques del microprocesador 4040 e indica en bloques los mayores circuitos y sus interconexiones.

Los mayores bloques funcionales contenidos en el 4040 son:

1. El Stack (dispositivo de memoria) de dirección de registros y el incrementador de direcciones.
2. El Registro de arreglo de índices.
3. El Acumulador-Sumador de 4 bits.
4. El Registro de Instrucción-Decodificador y control de lógica.
5. Interrupción de Hardware y Control de parada.
6. Circuitos periféricos para controlar el tiempo y comunicación externa.

La figura 2-2 nos muestra un diagrama de bloques de la Uni

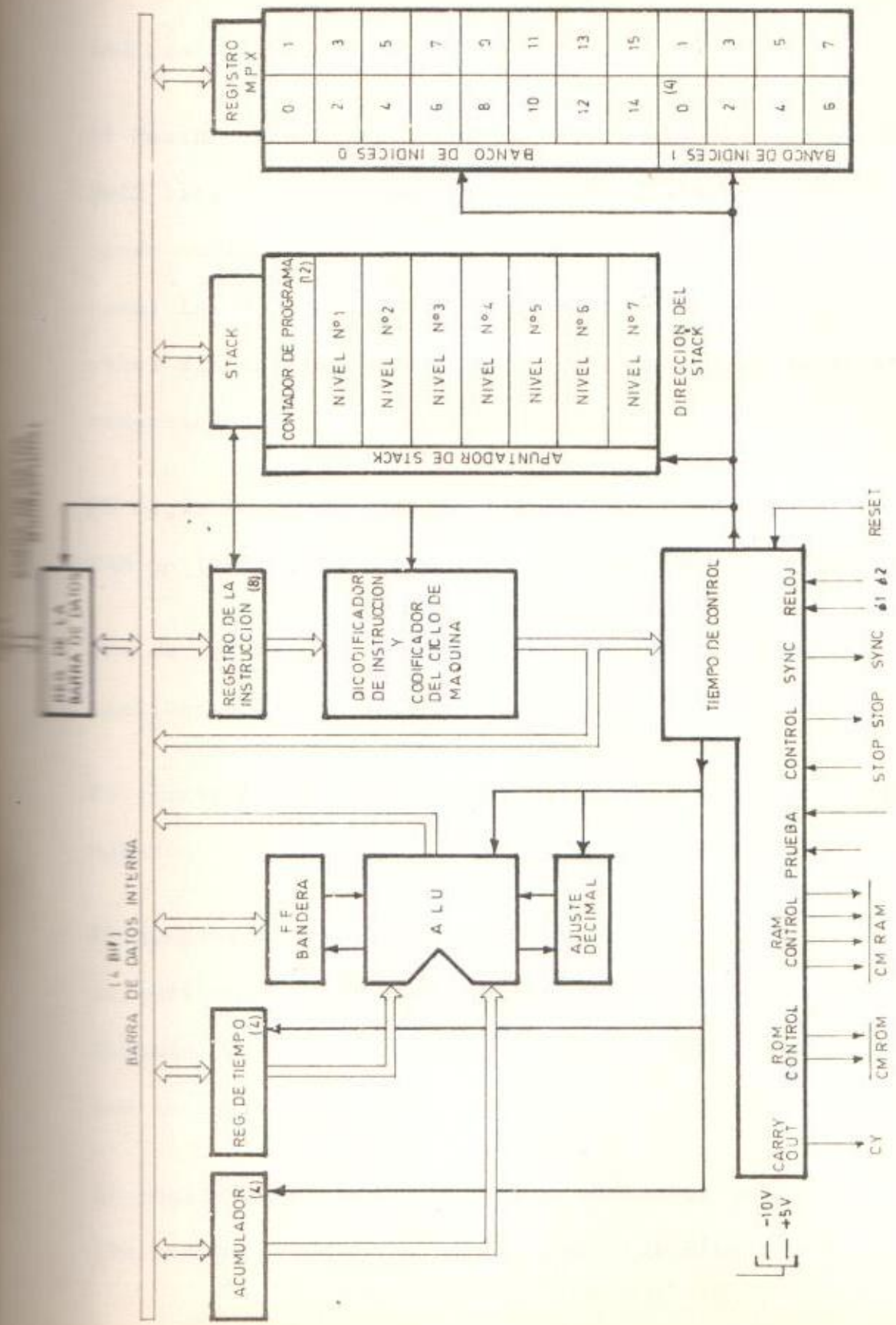


FIG 2.2  
DIAGRAMA DE BLOQUES DEL CPU 4040



dad Central de Procesamiento.

El Registro de Direcciones es un arreglo dinámico de RAM de 8x12 bits que opera como un push-down stack (memoria empujando hacia abajo). Un nivel del stack es usado para almacenar las direcciones de cada instrucción, dejándo siete niveles disponibles para llamar subrutinas y procesar interrupciones.

El registro de arreglo de índices es un arreglo dinámico de RAM de 12x8 bits organizado como tres bancos de 4x8 bits.

Dos de los bancos son seleccionados individualmente por las instrucciones SBO y SBI.

El tercer banco esta siempre disponible para uso de almacenamiento.

El Sumador/Acumulador es donde se efectúan las operaciones Aritméticas. El acumulador se comunica con el registro de comando, con ROMS especiales, con la lógica de condición y con la barra interna.

El registro de comando contiene un código de 3 bits para las líneas de CM-RAM y un bit para las líneas de CM-ROM.

El registro de instrucción es cargado con el contenido de la barra interna en M1 y M2 durante el primer ciclo de instrucción.

Las instrucciones son decodificadas en el decodificador de instrucción y aproximadamente conectados por puertas lógicas con señales de tiempo para proveer las señales de control para los varios bloques funcionales.

La lógica de interrupción detecta y reconoce la presencia de una señal de interrupción y fuerza al procesador para ejecutar una instrucción JMS en la localidad 003.

La lógica de control de parada llega a actuar en una manera similar, detectando y reconociendo la presencia de una señal de parada. El procesador es forzado a ejecutar una instrucción NOP y permanecerá en la condición de parada, hasta que la señal de parada sea removida.

Los Circuitos Periféricos consisten de: Los buffers de entrada-salida de la barra de datos que se comunican entre las vías de datos y la barra interna; Del generador de sincronismo y tiempo; De los buffers de salida, 2CM-ROMS y 4 CM-RAMS y del flip-flop de encendido y limpieza.

Durante un reset, todos los RAMS y flip-flops son limpiados y la barra de datos es establecida a "0" (Vss).

Después del reset, el contador de programa comenzará desde "0" (Vss) y las líneas seleccionadas serán CM-ROMo y CM-RAMo además la lógica de instrucción será desactivada y seleccionará el banco cero del registro de índices.

## 2.2 Organización del Programa en la Memoria del MCS-40.

El programa del sistema MCS-40 está normalmente localizado en un ROM que en este caso es el 4702 A. El interfase entre el procesador y el ROM es otro circuito integrado que es el interfase de memoria estandar 4289.

El programa de memoria puede estar también en un PROM (programa read only memory), es decir un chip que trabaja como un ROM que ya fue definido anteriormente y que es borrable, para luego ser reprogramado.

El programa en la memoria está siempre organizado en un arreglo de memoria de 8 bits de ancho.

El Programa de memoria del MCS-40 está conceptualmente dividido en páginas de 256 bytes. Una página de memo-

ria es por definición un grupo de 256 bytes donde los 4 bits de más alto orden de la dirección de 12 bits es siempre el mismo.

La dirección de 12 bits de los datos o de alguna otra instrucción esta formada combinando los 4 bits de más alto orden del contador de programa (el número corriente de página) con los restantes 8 bits de dirección suministrados por la instrucción.

### 2.3 Conjunto de Instrucciones.

Cuatro grupos de instrucciones pueden ser definidas como sigue:

#### a. Instrucciones de Máquina.

Este grupo de 16 instrucciones estan designados por un código OPR de 0000-1101.

#### b. Grupo 4040.

Este grupo de instrucciones esta designada por un código OPR de 0000 y un código OPA de 0001-1110.

#### c. Grupo de Entrada-Salida.

Esta designado por un código OPR de 1110, este grupo de 16 instrucciones es usado para transferir datos entre el procesador y los chips RAMS o circuitos de Entrada-Salida.

d. Grupo del Acumulador.

Este grupo de 14 instrucciones esta designado por un código de 1111 y opera sólo sobre el acumulador-carry flip-flop, los RAMS especiales y el registro de comando.

En el grupo de instrucciones de máquina tenemos instrucciones de una sola palabra y de dos palabras.

La instrucción de una sola palabra tiene 8 bits de ancho y requiere 8 periodos de reloj (1 ciclo de instrucción) para la ejecución de la misma.

La instrucción de dos palabras requiere 16 bits de ancho y 16 periodos de reloj, (2 ciclos de instrucción) para su ejecución.

Una instrucción de una palabra ocupa una localidad en un ROM (cada localidad puede contener 8 bits) y una instruc-

ción de dos palabras ocupa dos localidades sucesivas en un ROM.

Los 4 bits superiores son llamados OPR y contienen el código de operación. Los 4 bits inferiores son llamados OPR y contienen el codificador.

Para las instrucciones de una sola palabra el código de operación contiene la operación que va a ser desarrollada (suma resta, carga etc.)

El Modificador (OPA) contiene una de las cuatro cosas:

1. Una dirección de registro.
2. Una dirección del par registro.
3. 4 bits de datos.
4. Una instrucción modificadora.

Para una instrucción de dos palabras, la primera palabra es similar a la instrucción de una sola palabra, sin embargo, el modificador (OPA) contiene una de las 4 cosas siguientes:

1. Una dirección de registro.
2. Una dirección del par registro
3. La parte superior de la dirección de otro ROM

#### 4. Una condición de salto.

La segunda palabra contiene la parte media en el OPR y la parte inferior en el OPA de la dirección de otro ROM u 8 bits de datos (los 4 bits superiores en el OPR y los 4 bits inferiores en el OPA).

Los 4 bits superiores de la instrucción (OPR) serán siempre entregados antes que los 4 bits inferiores (OPA de una instrucción durante M1 y M2 respectivamente.

En la tabla 2-1 se ha listado todas las instrucciones del sistema MCS-40, con su respectiva explicación.

#### 2.4. Configuración de la Barra de Datos.

En la Barra de Datos se presentan los bits ya sea de datos o de alguna instrucción en los segmentos de tiempo X2 y X3.

En el tiempo X1 el procesador 4040 saca los contenidos del acumulador para propósitos de depuración del programa.

Durante X2 y X3 ocurre la ejecución de la instrucción y el contenido de la barra de datos depende de la ins

## SUMARIO DE CONJUNTO DE INSTRUCCIONES

MNEMONIC	OPR D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	OPA D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	DESCRIPTION OF OPERATION
NOP	0 0 0 0	0 0 0 0	No operation.
*JCN	0 0 0 1 A <sub>2</sub> A <sub>2</sub> A <sub>2</sub> A <sub>2</sub>	C <sub>1</sub> C <sub>2</sub> C <sub>3</sub> C <sub>4</sub> A <sub>1</sub> A <sub>1</sub> A <sub>1</sub> A <sub>1</sub>	Jump to ROM address A <sub>2</sub> A <sub>2</sub> A <sub>2</sub> A <sub>2</sub> , A <sub>1</sub> A <sub>1</sub> A <sub>1</sub> A <sub>1</sub> (within the same ROM that contains this JCN instruction) if condition C <sub>1</sub> C <sub>2</sub> C <sub>3</sub> C <sub>4</sub> (1) is true, otherwise skip (go to the next instruction in sequence).
*FIM	0 0 1 0 D <sub>2</sub> D <sub>2</sub> D <sub>2</sub> D <sub>2</sub>	R R R 0 D <sub>1</sub> D <sub>1</sub> D <sub>1</sub> D <sub>1</sub>	Fetch immediate (direct) from ROM Data D <sub>2</sub> , D <sub>1</sub> to index register pair location RRR (2).
SRC	0 0 1 0	R R R 1	Send register control. Send the address (contents of index register pair RRR) to ROM and RAM at X <sub>2</sub> and X <sub>3</sub> time in the Instruction Cycle.
FIN	0 0 1 1	R R R 0	Fetch indirect from ROM. Send contents of index register pair location 0 out as an address. Data fetched is placed into register pair location RRR.
JIN	0 0 1 1	R R R 1	Jump indirect. Send contents of register pair RRR out as an address at A <sub>1</sub> and A <sub>2</sub> time in the Instruction Cycle.
*JUN	0 1 0 0 A <sub>2</sub> A <sub>2</sub> A <sub>2</sub> A <sub>2</sub>	A <sub>3</sub> A <sub>3</sub> A <sub>3</sub> A <sub>3</sub> A <sub>1</sub> A <sub>1</sub> A <sub>1</sub> A <sub>1</sub>	Jump unconditional to ROM address A <sub>3</sub> , A <sub>2</sub> , A <sub>1</sub> .
*JMS	0 1 0 1 A <sub>2</sub> A <sub>2</sub> A <sub>2</sub> A <sub>2</sub>	A <sub>3</sub> A <sub>3</sub> A <sub>3</sub> A <sub>3</sub> A <sub>1</sub> A <sub>1</sub> A <sub>1</sub> A <sub>1</sub>	Jump to subroutine ROM address A <sub>3</sub> , A <sub>2</sub> , A <sub>1</sub> , save old address. (Up 1 level in stack.)
INC	0 1 1 0	R R R R	Increment contents of register RRRR. (3)
*ISZ	0 1 1 1 A <sub>2</sub> A <sub>2</sub> A <sub>2</sub> A <sub>2</sub>	R R R R A <sub>1</sub> A <sub>1</sub> A <sub>1</sub> A <sub>1</sub>	Increment contents of register RRRR. Go to ROM address A <sub>2</sub> , A <sub>1</sub> (within the same ROM that contains this ISZ instruction) if result ≠ 0, otherwise skip (go to the next instruction in sequence).
ADD	1 0 0 0	R R R R	Add contents of register RRRR to accumulator with carry.
SUB	1 0 0 1	R R R R	Subtract contents of register RRRR to accumulator with borrow.
LD	1 0 1 0	R R R R	Load contents of register RRRR to accumulator.
XCH	1 0 1 1	R R R R	Exchange contents of index register RRRR and accumulator.
BBL	1 1 0 0	D D D D	Branch back (down 1 level in stack) and load data DDDD to accumulator.
LDM	1 1 0 1	D D D D	Load data DDDD to accumulator.

## GRUPO DE INSTRUCCIONES DEL ACUMULADOR

CLB	1 1 1 1	0 0 0 0	Clear both. (Accumulator and carry)
CLC	1 1 1 1	0 0 0 1	Clear carry.
IAC	1 1 1 1	0 0 1 0	Increment accumulator.
CMC	1 1 1 1	0 0 1 1	Complement carry.
CMA	1 1 1 1	0 1 0 0	Complement accumulator.
RAL	1 1 1 1	0 1 0 1	Rotate left. (Accumulator and carry)
RAR	1 1 1 1	0 1 1 0	Rotate right. (Accumulator and carry)
TCC	1 1 1 1	0 1 1 1	Transmit carry to accumulator and clear carry.
DAC	1 1 1 1	1 0 0 0	Decrement accumulator.
TCS	1 1 1 1	1 0 0 1	Transfer carry subtract and clear carry.
STC	1 1 1 1	1 0 1 0	Set carry.
DAA	1 1 1 1	1 0 1 1	Decimal adjust accumulator.
KBP	1 1 1 1	1 1 0 0	Keyboard process. Converts the contents of the accumulator from a one out of four code to a binary code.
DCL	1 1 1 1	1 1 0 1	Designate command line.



## INSTRUCCIONES DE LOS C.I. 4001, 4002, 4008, 4009 y 4289

MNEMONIC	OPR D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	OPA D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	DESCRIPTION OF OPERATION
WRM	1 1 1 0	0 0 0 0	Write the contents of the accumulator into the previously selected RAM main memory character.
WMP	1 1 1 0	0 0 0 1	Write the contents of the accumulator into the previously selected RAM output port.
WRR	1 1 1 0	0 0 1 0	Write the contents of the accumulator into the previously selected ROM output port. (I/O Lines).
WPM	1 1 1 0	0 0 1 1	Write the contents of the accumulator into the previously selected half byte of read/write program memory (for use with 4008/4009 or 4289).
WR0 <sup>(4)</sup>	1 1 1 0	0 1 0 0	Write the contents of the accumulator into the previously selected RAM status character 0.
WR1 <sup>(4)</sup>	1 1 1 0	0 1 0 1	Write the contents of the accumulator into the previously selected RAM status character 1.
WR2 <sup>(4)</sup>	1 1 1 0	0 1 1 0	Write the contents of the accumulator into the previously selected RAM status character 2.
WR3 <sup>(4)</sup>	1 1 1 0	0 1 1 1	Write the contents of the accumulator into the previously selected RAM status character 3.
SBM	1 1 1 0	1 0 0 0	Subtract the previously selected RAM main memory character from accumulator with borrow.
RDM	1 1 1 0	1 0 0 1	Read the previously selected RAM main memory character into the accumulator.
ROR	1 1 1 0	1 0 1 0	Read the contents of the previously selected ROM input port into the accumulator. (I/O Lines).
ADM	1 1 1 0	1 0 1 1	Add the previously selected RAM main memory character to accumulator with carry.
RD0 <sup>(4)</sup>	1 1 1 0	1 1 0 0	Read the previously selected RAM status character 0 into accumulator.
RD1 <sup>(4)</sup>	1 1 1 0	1 1 0 1	Read the previously selected RAM status character 1 into accumulator.
RD2 <sup>(4)</sup>	1 1 1 0	1 1 1 0	Read the previously selected RAM status character 2 into accumulator.
RD3 <sup>(4)</sup>	1 1 1 0	1 1 1 1	Read the previously selected RAM status character 3 into accumulator.

## INSTRUCCIONES DEL 4040

MNEMONIC	OPR D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	OPA D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	DESCRIPTION OF OPERATION
HLT	0 0 0 0	0 0 0 1	Halt - Inhibit program counter and data buffers.
BBS	0 0 0 0	0 0 1 0	Branch Back from Interrupt and restore the previous SRC. The Program Counter and send register control are restored to their pre-interrupt value.
LCR	0 0 0 0	0 0 1 1	The contents of the COMMAND REGISTER are transferred to the ACCUMULATOR.
OR4	0 0 0 0	0 1 0 0	The 4 bit contents of register #4 are logically "OR ed" with the ACCUM.
OR5	0 0 0 0	0 1 0 1	The 4 bit contents of index register #5 are logically "OR ed" with the ACCUMULATOR.
AN6	0 0 0 0	0 1 1 0	The 4 bit contents of index register #6 are logically "AND ed" with the ACCUMULATOR.
AN7	0 0 0 0	0 1 1 1	The 4 bit contents of index register #7 are logically "AND ed" with the ACCUMULATOR.
DB0	0 0 0 0	1 0 0 0	DESIGNATE ROM BANK 0. CM ROM <sub>0</sub> becomes enabled.
DB1	0 0 0 0	1 0 0 1	DESIGNATE ROM BANK 1. CM ROM <sub>1</sub> becomes enabled.
SB0	0 0 0 0	1 0 1 0	SELECT INDEX REGISTER BANK 0. The index registers 0 - 7.
SB1	0 0 0 0	1 0 1 1	SELECT INDEX REGISTER BANK 1. The index registers 0* - 7*.
EIN	0 0 0 0	1 1 0 0	ENABLE INTERRUPT.
DIN	0 0 0 0	1 1 0 1	DISABLE INTERRUPT.
RPM	0 0 0 0	1 1 1 0	READ PROGRAM MEMORY.

TABLA 2-1

trucción a ser ejecutada.

Un listado del contenido en la barra de datos durante los tiempos X2 y X3 se encuentra en la tabla 2-2.

## CONTENIDO DE LA BARRA DE DATOS

INSTRUCCION	DATA @ X <sub>2</sub>				DATA @ X <sub>3</sub>				COMMENTS
	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
NOP	1	1	1	1	1	1	1	1	
JCN	1	1	1	1	1	1	1	1	
A <sub>2</sub> , A <sub>1</sub>	1	1	1	1	1	1	1	1	
FIN RRR0	(RRR0)				(RRR1)				The content of address par RRR
D <sub>2</sub> , D <sub>1</sub>	1	1	1	1	1	1	1	1	
SAC RRR1	(RRR0)				(RRR1)				
FIN RRR0	(RRR0)				(RRR1)				
2nd cycle	1	1	1	1	1	1	1	1	
JN RRR0	(RRR0)				(RRR1)				
JN A <sub>3</sub>	A <sub>3</sub>				1	1	1	1	
A <sub>2</sub> , A <sub>1</sub>	A <sub>3</sub>				1	1	1	1	
JN A <sub>3</sub>	A <sub>3</sub>				1	1	1	1	
A <sub>2</sub> , A <sub>1</sub>	A <sub>3</sub>				1	1	1	1	
INC RRRR	(RRRR)				(RRRR) + 1				Content of register RRRR; Content +1 of RRRR
ISZ RRRR	(RRRR)				(RRRR) + 1				
A <sub>2</sub> , A <sub>1</sub>	1	1	1	1	1	1	1	1	
ADD RRRR	(RRRR)				1	1	1	1	Content of register RRR
SUB RRRR	(RRRR)				1	1	1	1	
LD RRRR	(RRRR)				1	1	1	1	
RDH RRRR	(RRRR)				(ACC)				Content of register RRRR; the content of ACC
SEL	DDDD				1	1	1	1	Data DDDD
LDM	DDDD				1	1	1	1	Data DDDD
WRW, WR0, WR1, WR2, WR3, WRM, WMP, WRR	(ACC)				1	1	1	1 (CY) For 4004 1	Content of accumulator; Content of CY F/F is present on D <sub>0</sub>
RDW, RD0, RD1, RD2 RD3, ADM, SBM, RDR	(M) or (INPUT)				(M) or (INPUT)				Data fetched from RAM or input
CLR, CLC, IAC, CMC CMA, RAL, PAR, TCC TCS	0	0	0	0	1	1	1	1	
STC, DAC, DCL	1	0	0	1	1	1	1	1	
DAA	0	0	0	0 or 0	1	1	1	1	X <sub>2</sub> depends on ACC content
CBP	0000, 0001, 0010 0011, 0100, 1111				1	1	1	1	X <sub>2</sub> depends on ACC content

TABLA 2-2

## CAPITULO III

## CONFIGURACIONES DEL SISTEMA MCS-40.

## 3.1 Configuraciones del Sistema.

Los siguientes diagramas de bloques ilustran varias configuraciones en el cual los componentes del sistema MCS-40 pueden ser aplicados.

Todos los diagramas de bloques pueden ser expandidos para acomodarse de acuerdo a una aplicación específica.

## a. Controlador Básico de Proceso.

El controlador básico de proceso tendrá algunos instrucciones específicas programadas en el ROM 4308.

La figura 3-1 muestra el Controlador Básico de Proceso.

El convertidor de nivel decodificará la señal de entrada proporcionada por los sensores de proceso. La salida del convertidor de nivel, será la dirección del ROM donde habrá un conjunto de instrucciones que serán ejecutadas por el CPU 4040 y la salida del ROM será la que establezca el control de secuencia necesitado.

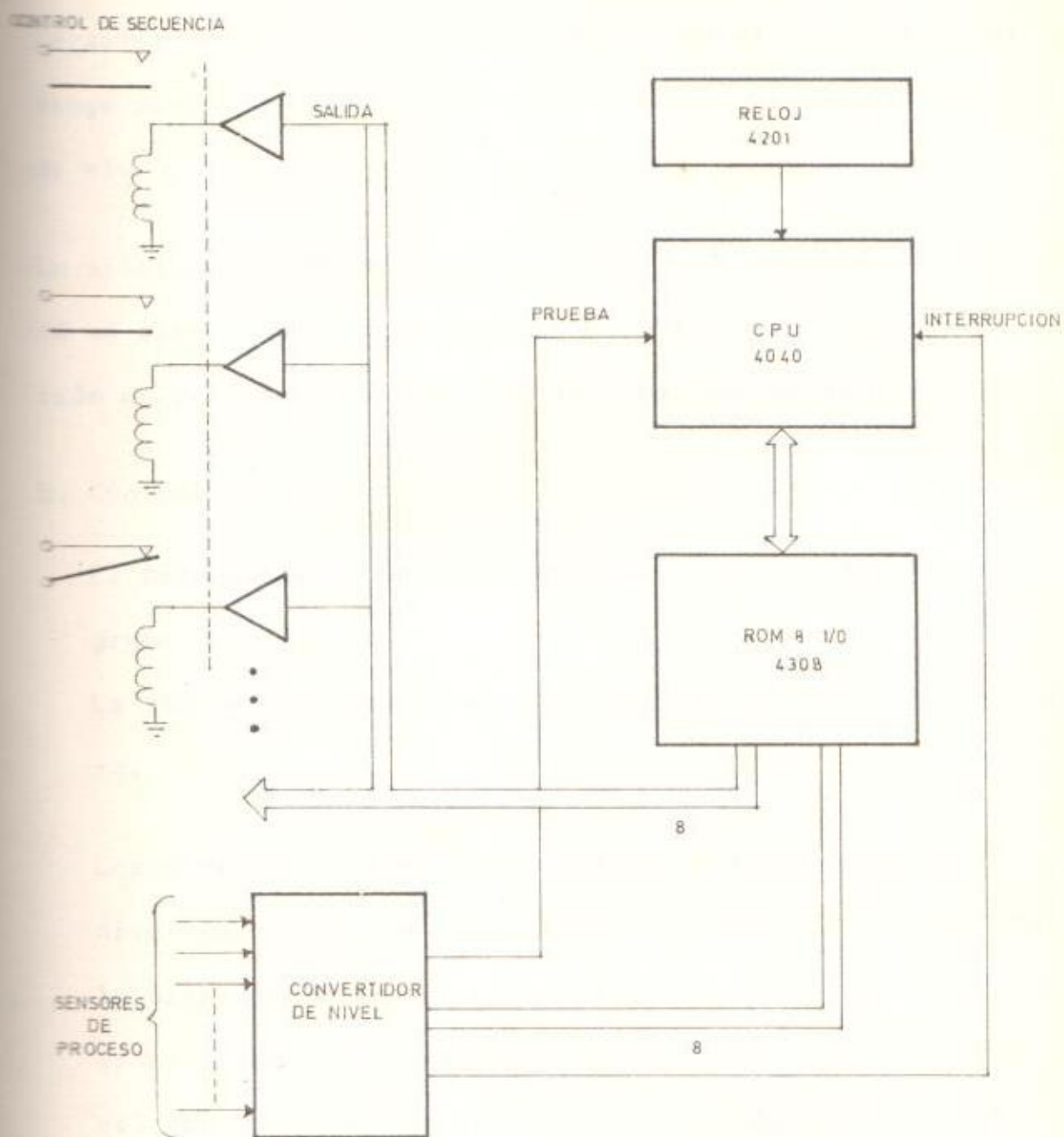


FIG. 3.1

CONTROLADOR BASICO DE PROCESO

La señal de interrupción hará que el proceso se detenga de tenga debido a alguna falla detectada por el convertidor de nivel.

La señal de prueba que trabaja como un interruptor provoc rá un alto o bajo lógico en el CPU y que según esta condición de entrada el CPU tomará una decisión pertinente.

#### b. Control de Semáforo.

El dispositivo 4308 es la memoria donde estará el programa del sistema.

La figura 3-2 muestra el gráfico del Control de Semáforo.

Los sensores de las calles activaron el dispositivo del nivel de desvío que direccionará el ROM. De acuerdo a la dirección del ROM, el procesador central ejecutará el programa y por las puertas de entrada-salida, tanto del ROM 4308 como de los dispositivos de comunicación serial el 4265 y 8251 enviará los datos de salida al semáforo y una salida serial (puede ser un terminal de video) respectivamente.

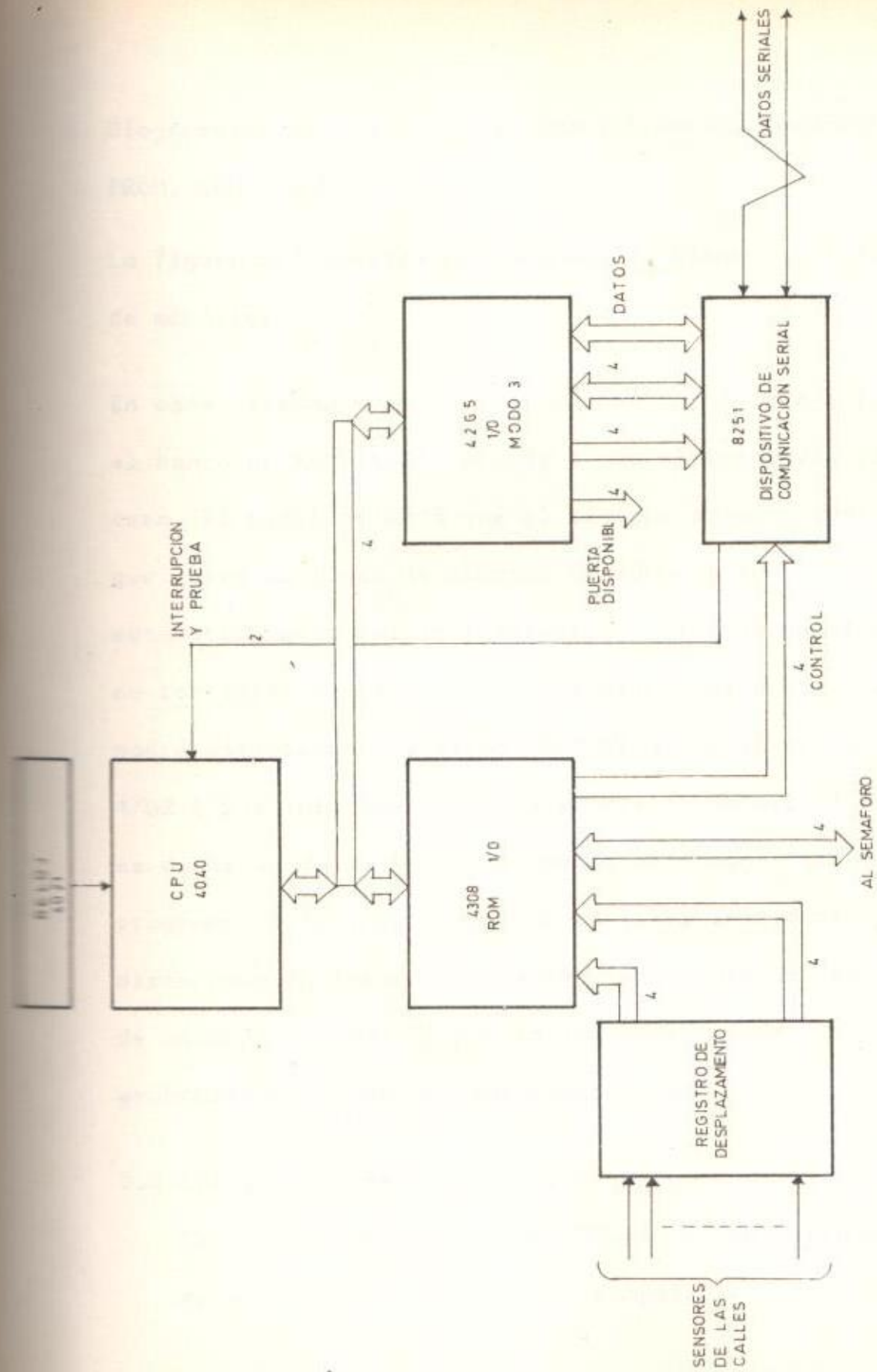


FIG. 3 - 2  
CONTROL DE SEMAFORO

- c. Diagrama de un sistema indicando el uso de memorias PROM, ROM y RAM.

La figura 3-3 muestra un diagrama de bloques del sistema de memorias.

En este sistema de memorias, el monitor lo puede tener el banco de ROMS 4308, debido a que al iniciarse el sistema, el banco de ROMS que el sistema primero lee, es el que lleva la línea de comando CM-ROM0 porque el sistema automáticamente así se inicializa. De acuerdo al programa contenido en este banco el sistema, es decir, el CPU podrá direccionar la línea CM-ROM1 que activará el PROM 4702 ( por intermedio del interfase de memoria 4289) que es donde se encuentra un programa de diagnóstico. Este programa de diagnóstico podrá contener instrucciones que direccionarán las memorias RAM (por medio de las líneas de comando CM-RAM0-2) y a la vez podrán hacer pruebas de grabación y lectura en las memorias RAM.

### 3.2 Consideraciones del Sistema.

Todos los componentes del MCS-40 estan diseñados para operar como una familia compatible. Sin embargo,



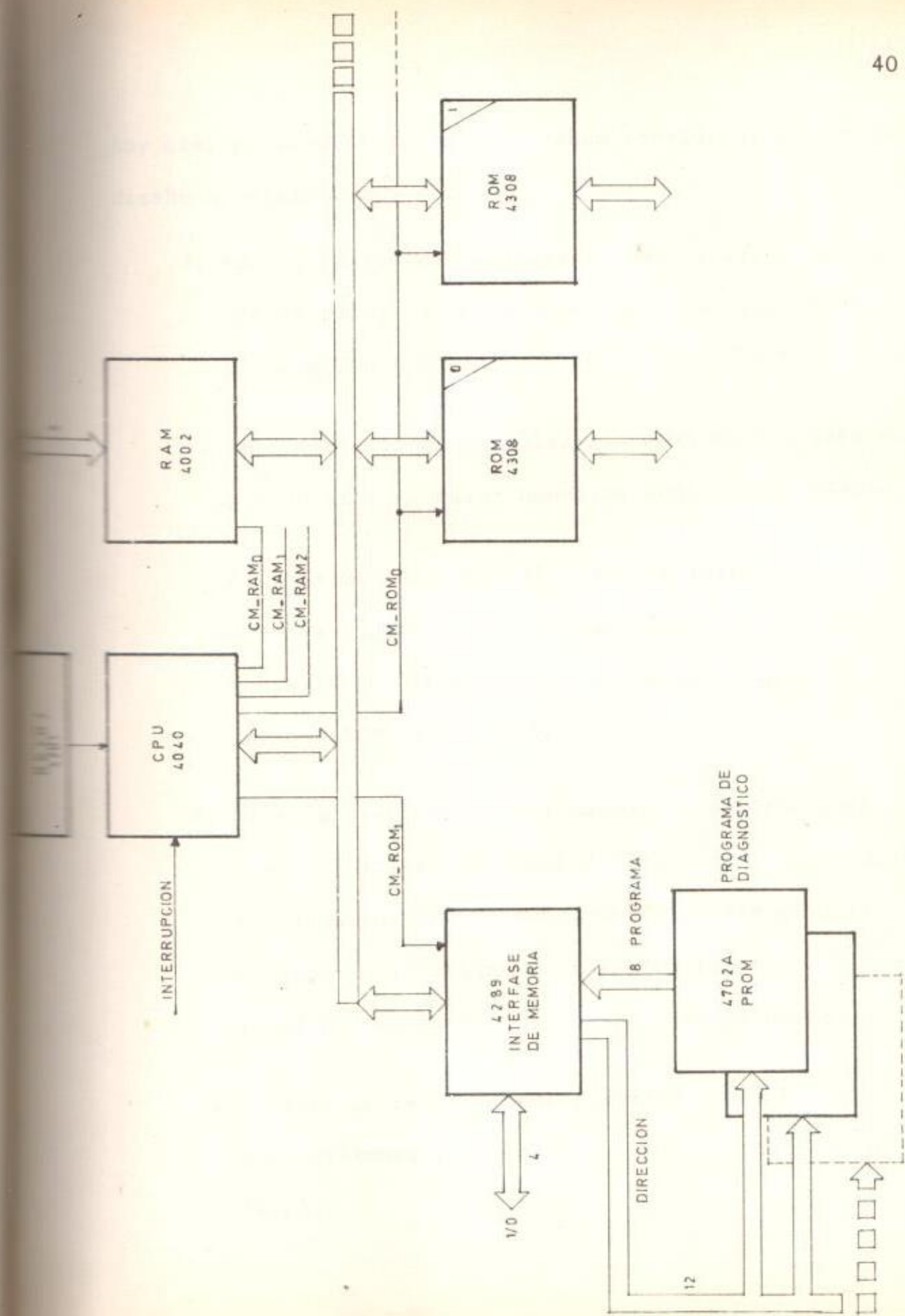


FIG. 3.3  
DIAGRAMA DE UN SISTEMA INDICANDO EL USO DE MEMORIAS : PROM, ROM y RAM

hay ciertas condiciones que se deben considerar cuando se diseña un sistema MCS-40.

1. Para garantía de la operación del sistema, el ciclo de reloj debe estar entre 1.35useg (740KHZ) y 2useg(500 KHZ).

2. Para operaciones normales los ROMS 4001 ó 4308 no pueden usar el mismo banco de ROMS con el CI4289.

Esto es debido a que el C.I.4289 responderá a todas las instrucciones del programa dedicados a él y provocaría un conflicto en la barra de datos con los ROMS 4001 y 4308.

3. Si el sistema tiene dos bancos de memoria y el sistema utiliza la señal de interrupción del 4040 la localidad 003 de ambas memorias dirigirá el programa a la rutina de interrupción, en la localidad 003 empieza el proceso de interrupción.

4. Durante un reset, todos los RAMS y los Flip-Flops son resetados y la barra de datos es seteado a "0" (Vss).

Después del reseteo, el control de programa empezará desde la dirección "0" y las líneas CM-RAMO/CM-ROMO son seleccionadas.

Para limpiar completamente todos los registros y las localidades del RAM en el CPU, la señal de reset debe de ser aplicada por lo menos 12 ciclos completos de instrucción, es decir pulsos de reloj para permitir al contador del registro de índices, hacer la búsqueda en todas las localidades de la memoria.

5. Una puerta de entrada-salida asociada con los 4289 ó 4009 puede tener líneas con capacidad de entrada y salida, en cambio en los 4001 ó 4308 las puertas de entrada -salida deben tener una sola función ya sea entrada ó salida.

## CAPITULO IV

## COMPONENTES DE APOYO DE LA FAMILIA DEL MCS-40

4.1 "4265 Circuito Integrado Programable de Propósitos Ge  
nerales de Entrada-Salida.

El 4265 es un dispositivo de Entrada-Salida de propósitos generales designado como interfase con la familia del Microcomputador MCS-40.

Este dispositivo provee 4 puertas de Entrada-Salida de 4 bits programables que puede ser configurado para permitir uno de los catorce modos de operación únicos para ponerlos de interfase con la memoria de datos o una variedad de dispositivos periféricos del usuario.

Un solo sistema MCS-40 puede acomodar hasta cuatro CI 4265 (uno por cada línea CM-RAM) sin lógica externa o hasta ocho CI 4265 con un decodificador externo.

El 4265 reside sobre la barra de datos del MCS-40 y usa el mismo procedimiento de instalación como el dispositivo RAM 4002. La selección ocurre solamente

cuando el código propio de la instrucción SRC y la señal CM están presentes simultáneamente.

El 4265 es extremadamente flexible para sistemas de Entrada-Salida y es capaz de manejar 4 u 8 bits de datos de Entrada o salida.

La figura 4-1 nos muestra el circuito integrado 4265 con su configuración de pines.

El Pin 1 es designado para la fuente de voltaje mas positiva ( $V_{ss} = +5V$ ).

Los pines 2 a 5 son designados para D0-D3 que consisten en la barra de datos bidireccional. Todas las direcciones, instrucciones y comunicaciones de datos entre el procesador y las puertas de entrada-salida son transferidos por esta puerta.

El Pin 6 es para la señal de reset.

Un voltaje negativo ( $V_{dd}$ ) aplicado a este pin limpia todos los elementos de almacenamiento y coloca el 4265 en el modo de reset, deseleccionando el dispositivo.

El pin 7 es el CM, que es un comando de entrada manejado

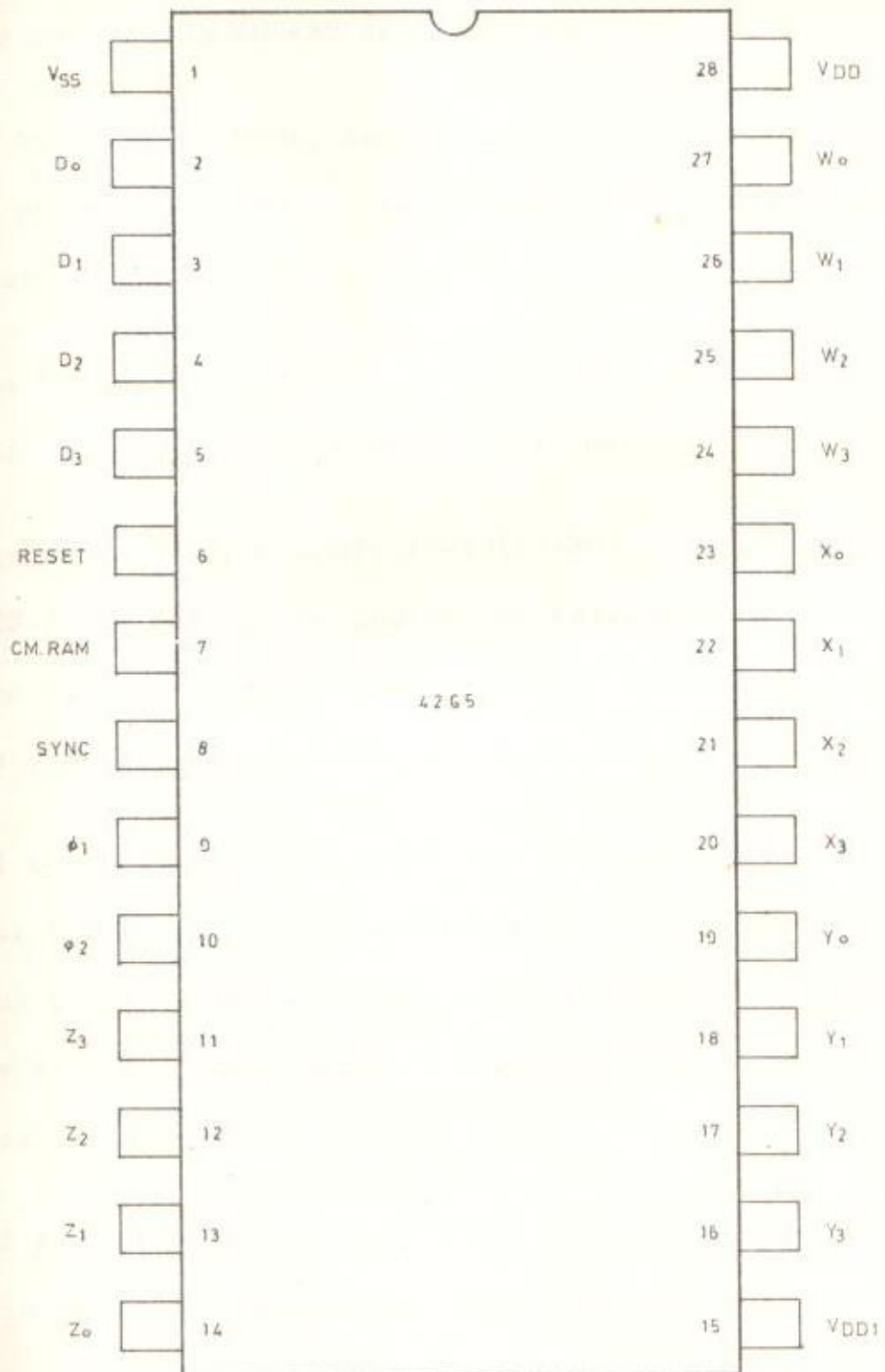


FIG. 4.1

CONFIGURACION DE PINES DEL C.I. 4265

por una salida CM-RAM del Procesador.

El Pin 8 es el SYNC, que es la señal de sincronismo generada por el procesador. Esta señal indica el comienzo de una instrucción.

Los Pines 9-10 están designados con  $\phi 1$ - $\phi 2$  que son las señales de reloj que determinan el tiempo.

Los pines 24-27(W0-W3): 20-23(X3-X0); 16-19(Y3-Y0); 11-14 (Z3-Z0). Son cuatro puertas de Entrada-Salida programables que tienen diferentes designaciones funcionales, dependiendo del modo de operación del 4265.

Si sobre la barra de datos hay lógica negativa "1", es decir con un voltaje  $V_{dd}(-10V)$  entonces en la salida de estas puertas, habrá una lógica positiva "1", es decir voltaje  $V_{ss}(+5V)$  para hacerlo compatible con circuitos periféricos TTL.

El Pin 15 está designado a  $V_{dd1}$  que es la fuente de potencia para las puertas de Entrada-Salida.

El Pin 28 está designado a  $V_{dd}$  que es la fuente de potencia principal del 4265. Su valor debe ser  $V_{ss}-15\pm 5\%$ .

#### 4.2 "4269" Circuito Integrado Programable para Teclado y Dispositivo de Salida Visual.

El 4269 tiene dos secciones separadas y distintas. la sección de Teclado y la sección de Video.

La sección de Teclado puede programarse de interfase, tanto a un simple conjunto de switchs como a un panel completo de teclado estilo máquina de escribir.

La sección de video puede ponerse de interfase desde un arreglo de diodos luminosos (LED) que sirven como indicadores, hasta un video alfanumérico de descarga de gas.

El 4269 que es un circuito programable de Teclado-Video, libera al procesador central (CPU 4040) de una continua búsqueda de teclado para saber que tecla se ha presionado y de un refresco continuo al dispositivo de video por control de software, es decir, el CPU no necesita tener un programa que continuamente este haciendo búsquedas para los dispositivos de teclado y video, sino que el 4269 lo hace de acuerdo al modo en



que se lo ha programado.

El 4269 expande enormemente la capacidad del CPU 4040.

El 4269 puede rastrear un teclado de matriz 8x8 o un teclado de 2x8x8 con la ayuda de las teclas de entrada shift y control.

La parte de Video puede refrescar continuamente un solo video alfanumérico de 16x8; Un solo video alfanumérico de 8x8; Un video digital de 16x4 dígitos; un solo video de 32x4 dígitos; un 16x6, 18x6 ó 20x6 video alfanumérico de gas descarga.

La figura 4-2 muestra el 4269 con su configuración de Pines

El Pin 1 esta designado a Vss que es la fuente de voltaje más positivo (+5V).

El Pin 2 esta designado al reset que es una señal de entrada. Un nivel bajo (Vdd) aplicado a esta entrada resetea el PKD ó 4269.

El Pin 3 esta designado a SYNC. La señal de entrada de sincronización esta dada por la salida SYNC (sincroniza-

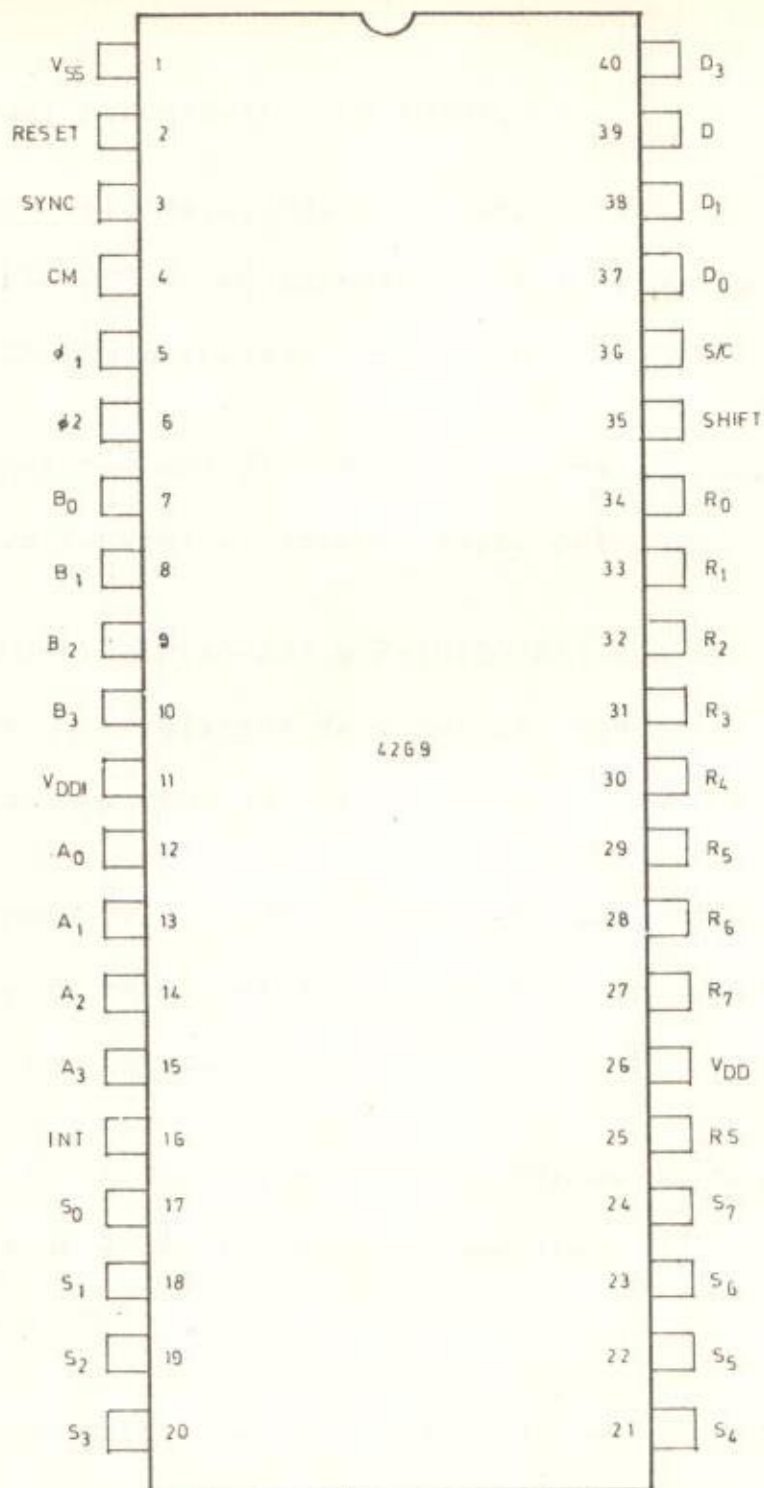


FIG. 4-2  
CONFIGURACION DE PINES DEL C.I. 4269

ción) del procesador (CPU 4040).

El Pin 4 esta desigando a CM que es la línea de comando que entra al 4269 seleccionandolo. Esta línea de comando es la señal CM-RAM originada por el CPU.

Los pines 5-6 son  $\phi 1$  y  $\phi 2$  que son las señales de reloj usadas para generar el tiempo básico del chip.

Los pines 12-15(A0-A3) y 7-10(B0-B3) son dos puertas de salida de los registros de video de 16x4 bits y que estan recirculando y sincronizados por las líneas de búsqueda S.

En el modo de video de descarga de gas, A3, es la señal de reset y A2 es el reloj al video de descarga de gas. Estas líneas son activas a Vss.

El Pin 11 es Vdd1, que es la fuente de voltaje para los registros de video A y B, asi como tambien para la señal de interrupción.

El Pin 16 esta asignado a INT, esta es la salida usada para indicar cuando un caracter del teclado ha entrado al buffer

Los Pines 17-24 estan asignados a 50-67 que son las líneas

de búsqueda usadas para detectar algún carácter del teclado y del dispositivo de video.

El Pin 25 esta asignado a RJ. Esta señal es usada cuando se desea tener en un dispositivo de video más de 8 digitos con un máximo de 16. Es activo Vss para los ultimos 8 digitos.

El Pin 26 es Vdd que la fuente de voltaje principal y su valor es  $V_{ss}-15V \pm 5\%$ .

Los pines 27-34 estan asignados a R0-R7 que son líneas de entrada y llamadas de retorno, ya que por ellas ingresa el caracter detectado en el teclado por las líneas S.

El Pin 25 esta designado a shift, este pin es usado cuando se está trabajando con un teclado que tenga la tecla shift

El Pin 36 esta asignado a S/C. Esta entrada se usa cuando se necesita un teclado externo.

Los pines 37-40 son D0-D3 que forman la barra de datos bidireccionales y comunicaciones de datos entre el CPU y el PKD(4269) son transferidos por estos 4 pines.

#### 4.3 "4201" CIRCUITO DE RELOJ.

El 4201 es un Circuito Integrado diseñado para cumplir con los requerimientos de la familia del Microcomputador MCS-40.

El 4201 contiene un oscilador de cristal controlado (cristal externo) que es el circuito de generación del reloj. Este circuito es compatible con la tecnología MCS y TTL.

El 4201 también desarrolla la función inicializado al momento de encender el equipo, que es el requerido por los componentes de la familia del MCS-40.

El 4201 provee la lógica necesaria para implementar la función de un solo paso de la Unidad Central de Procesamiento 4040.

La figura 4-31 nos muestra el circuito integrado 4201 con su configuración de Pines.

En la Figura 4-32 tenemos un diagrama de bloques del 4201.

El Pin 1 es GND, que es el circuito de tierra.

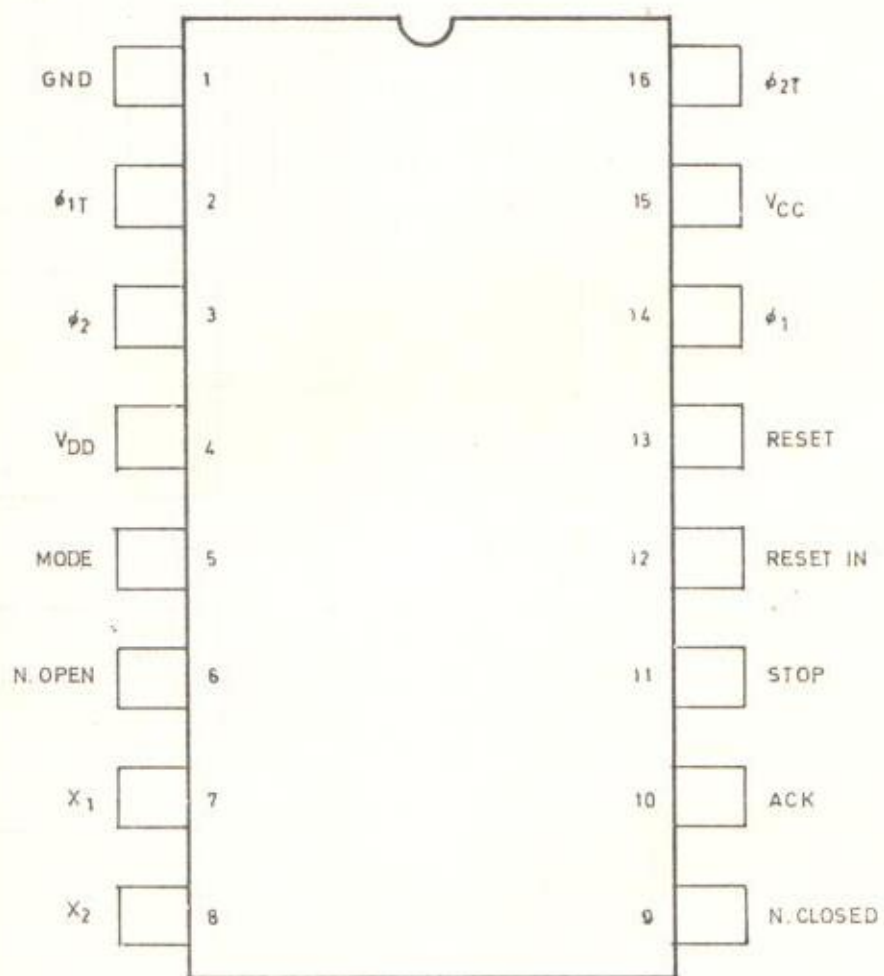


FIG- 4.31

CONFIGURACION DE PINES DEL C.I 4201

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL  
Dpto. de Ingeniería Eléctrica  
BIBLIOTECA  
Inv. No. ELEC.-012

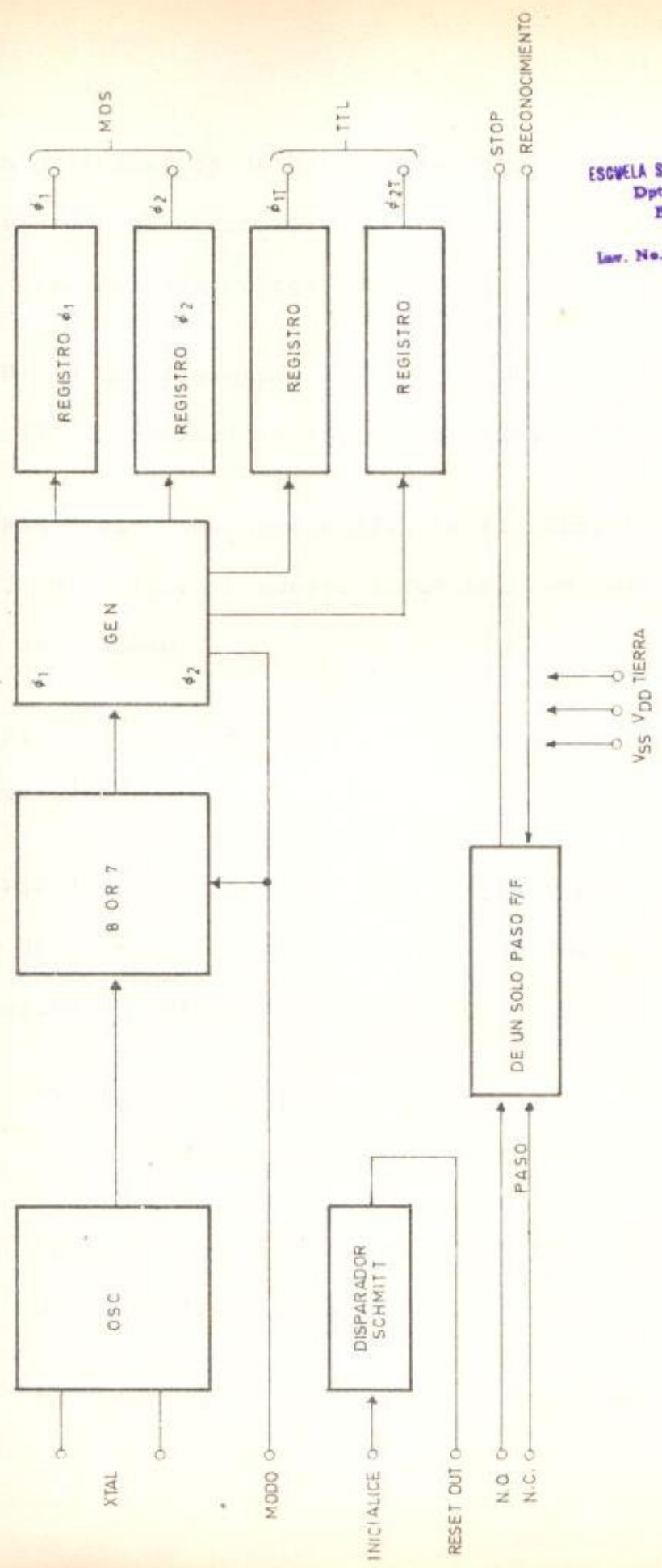


FIG 4.32  
DIAGRAMA DE BLOQUES DEL CI. 4201

para aplicaciones de baja potencia este Pin se puede dejar flotante y será operativo cuando se trabaje con tecnología MOS pero no operará con tecnología TTL.

El Pin 2 esta asignado a  $\emptyset 1T$ . Es el reloj fase 1 del nivel TTL se define con lógica positiva.

El Pin 3 esta asignado a  $\emptyset 2$ . Es el reloj fase 2 del nivel MOS. Directamente maneja todos los componentes de la familia del MCS-40.

El Pin 4 es Vdd, que es la fuente de voltaje principal  $V_{dd} = V_{ss} - 15 \pm 5\%$ .

El Pin 5 esta asignado a MODE. A este pin llega la señal que determina el Modo de Control. Estos modos son como se describe a continuación:

En modo 1, el Pin 5 será conectado a Vss y el contador dividirá para 7.

En modo 2, el Pin 5 será conectado a Vdd y el Contador dividirá para 8.

El pin 6 es N. Open. Es la entrada del circuito de



un solo paso que está conectado al contacto del switch (SPDT) normalmente abierto.

El Pin 7 es X1 que es la conexión externa del cristal. Este Pin también puede ser conectado a una fuente de frecuencia externa, pues en este caso el Pin 2 se dejaría desconectado.

El Pin 8 es X2, que es la conexión externa del cristal.

El Pin 9 está asignado a N. closed, que es la entrada del circuito de un solo Paso, y está conectado al contacto del switch (SPDT) normalmente cerrado.

El Pin 10 es ACK. es la entrada de reconocimiento del circuito de un solo Paso que está normalmente conectada a la salida del reconocimiento de parada del 4040.

El Pin 11 es STOP Es la salida de Parada del circuito de un solo Paso y está normalmente conectado a la entrada de parada del 4040.

El Pin 12 es reset in. Es la entrada de inicialización que está conectada a la red RC para proveer el tiempo de inicialización.

El Pin 13 es reset. Es la señal de salida de inicialización, la que conecta directamente a todas las entradas de los componentes de la familia del MCS-40.

El Pin 14 es  $\phi 1$ . Esta salida del Pin 14 es la fase del reloj para nivel MOS. Esta señal de reloj maneja directamente todos los componentes de la familia del MCS-40.

El Pin 15 es Vss, que es la fuente de voltaje más positivo

El Pin 16 es  $\phi 2T$ . Es la fase 2 del reloj para nivel TTL.

#### 4.4 "4289" INTERFASE PARA MEMORIA STANDARD.

El circuito integrado 4289 es un interfase para memoria standard y también es interfase para Puertas de Entrada-Salida.

EL CI 4289 habilita al CPU 4040 para utilizar componentes de memoria standard como Memoria para Datos de Programas. Notablemente, los PROMS (memoria programable y de solamente lectura), RAMS, y ROMS pueden ser colocados en un arreglo de memoria para facilitar el desarrollo de un sistema.

El 4289 también contiene una puerta de Entrada-Salida

bidireccional de 4 bits.

Este dispositivo direcciona directamente 4k de memoria programable. La dirección es obtenida secuencialmente durante los estados de A1-A3 de un ciclo de instrucción. La instrucción de 8 bits es presentada al CPU durante los estados de M1 y M2 del ciclo de instrucción por medio de la barra de datos de 4 bits.

El CI 4289 almacena la instrucción SRC operando como una dirección de Entrada-Salida y responde a las instrucciones de el ROM (WRR y RDR) leyendo o escribiendo datos hacia o del procesador y la barra de Entrada-Salida del 4289.

La figura 4-41 nos muestra el CI 4289 con su configuración de PINES.

Los Pines 1-A son 16-20, que consisten en la barra de la  
 las direcciones. Todas las direcciones, instrucciones,  
 memoria, configuración de datos entre el Registrador y el Memori-  
 zador, Memoria a los puertos de Entrada-Salida con el  
 Pines 21-24, 25-28, 29-32.

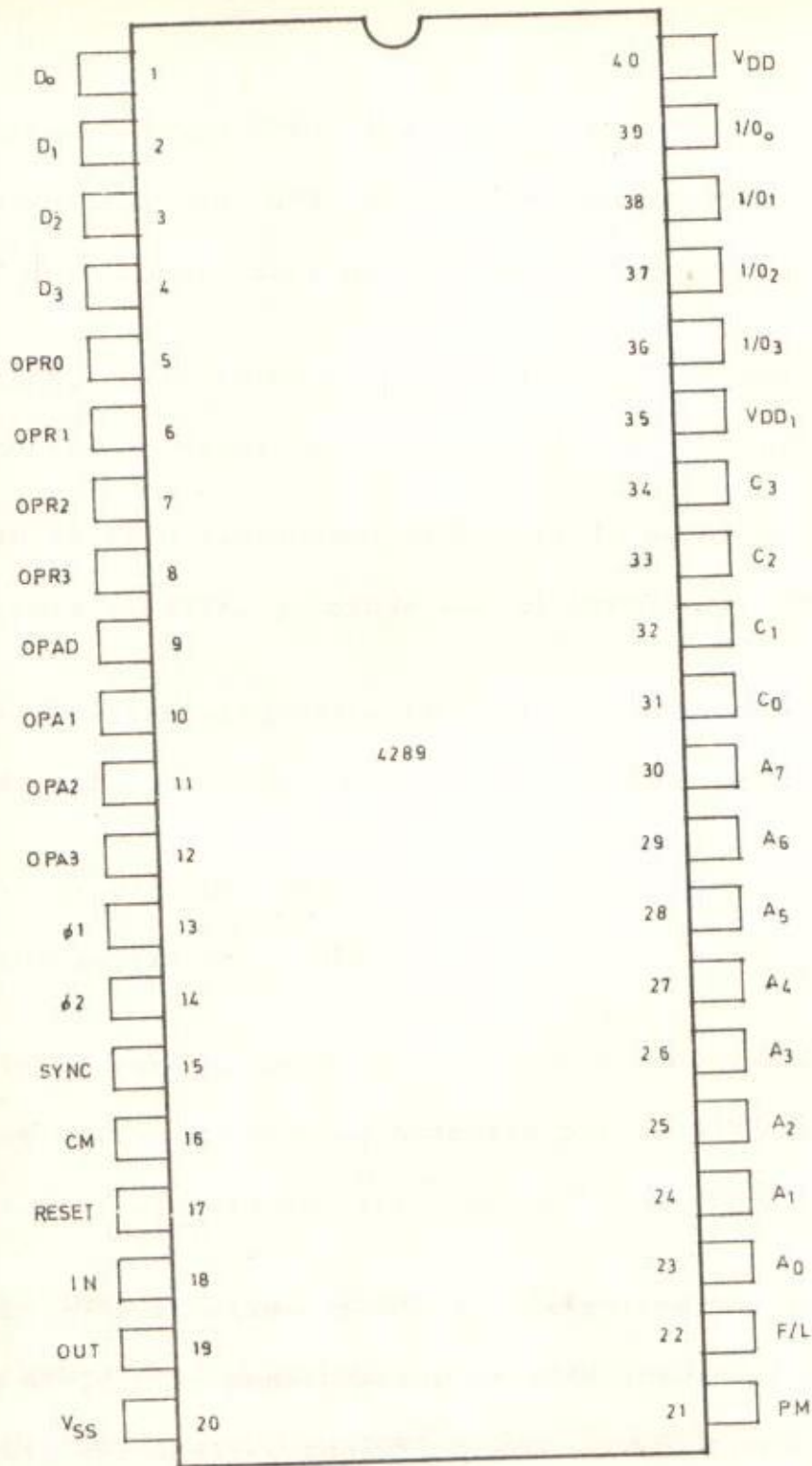


FIG. 4.41

CONFIGURACION DE PINES DEL C.I. 4289

Los Pines 5-8 son OPRO-OPR3. Los 4 bits de orden más alto de la instrucción (OPR) o datos del programa de memoria (RPM) son transferidos al 4289 sobre estos pines.

Los Pines 13-14 están asignados a  $\phi 1$  y  $\phi 2$ . Son las señales de reloj usadas para generar el tiempo básico del chip.

El Pin 15 está asignado a SYNC. Es la señal de sincronización para el 4289, generada por el Procesador 4040.

El Pin 16 está asignado a CM. Es la entrada de comando manejada por la salida CM-ROM del Procesador 4040.

El Pin 17 está asignado a reset. Es la señal de entrada que inicializa el CI 4289.

El Pin 18 está asignado a In. Esta es una señal de salida que se activa a Vdd y es generada por el 4289 cuando el procesador ejecuta una instrucción RDR o RPM.

El pin 19 está asignado a Out. Esta señal de salida se activa a Vdd y es generada por el 4289 cuando el procesador ejecuta una instrucción WRR ó WPM.

El Pin 20 es Vss. Es la fuente de voltaje más positivo.

El Pin 21 esta asignado a PM, es una señal de salida, generada por el 4289, cuando el procesador ejecuta una instrucción RPM o WPM.

El Pin 22 esta asignado a F/L. Es una señal de salida generada por el 4289, para indicar cual mitad de la memoria de programa va a ser operada. Si la línea tiene Vdd nos indica que opera los OPR y si la línea tiene Vss opera los OPA.

Los Pines 23-30 estan asignados a A0-A7.

Son las direcciones de los registros de salida, decodificadas por el 4289, que son suministradas por el procesador en A1 y A2.

Los Pines 31-34 son C0-C3. Estas líneas seleccionan el chip de memoria.

El Pin 35, es Vdd1, que es la fuente de poder para las direcciones y para la selección del chip de memoria.

Los pines 36-39 son I/00-I/03. Estas líneas son para los datos bidireccionales de las puertas de Entrada-Salida.

El Pin 40 es Vdd, es decir, el Pin para la fuente de volta  
je principal. Su valor debe ser  $V_{ss}-15^{+5}\%$ .

#### 4.5 "2101" (256x4)N MPS RAM.

El circuito Integrado 2101 es una memoria de acceso a  
leatorio de 256 palabras. Utiliza un circuito esta-  
ble DC, por lo tanto no necesita reloj ni refresco pa  
ra operar. Cuando los datos son leídos, no son des-  
truidos y tienen la misma polaridad que los datos de  
entrada.

El 2101 ha sido diseñado para aplicaciones de memoria  
donde un alto desarrollo, bajo costo, gran almacenaje  
de bits y simple interfase son importantes objetivos  
del diseño.

Es directamente compatible con TTL en todos los aspec  
tos, en entradas y salidas de datos. asi como también  
con una sola fuente de voltaje de +5 volts.

Se puede usar dos chips, ya que permiten una fácil se  
lección de un paquete individual cuando las salidas  
son atadas por medio de una puerta lógica "0".

Ha sido previsto un desactivo permitiendo así que los datos de entrada-salida puedan ser atados por sistemas de Entrada-Salida común. La salida de desactivo es entonces usada para eliminar cualquier lógica bidireccional.

El Intel 2101 es fabricado con tecnología Lógica de Canal-N en Silicón. Esta tecnología permite el diseño y la producción de alto desarrollo, fácil para usar circuitos MOS y proveer una densidad funcional más alta sobre un Chip monolítico que cualquier tecnología convencional MOS o Tecnología Lógica de Canal-P en Silicón.

La Tecnología Lógica de Silicón, también provee excelente protección contra la contaminación.

Esto permite el uso de un empaquetamiento plástico de bajo costo.

La figura 4-51 muestra la configuración de Pines del CI 2101.

Los Pines 1-7 y 21 son A0-A7 y están asignados a las entradas de las direcciones.

Los Pines 9, 11, 13 y 15 son DI1-DI4 que están asignados



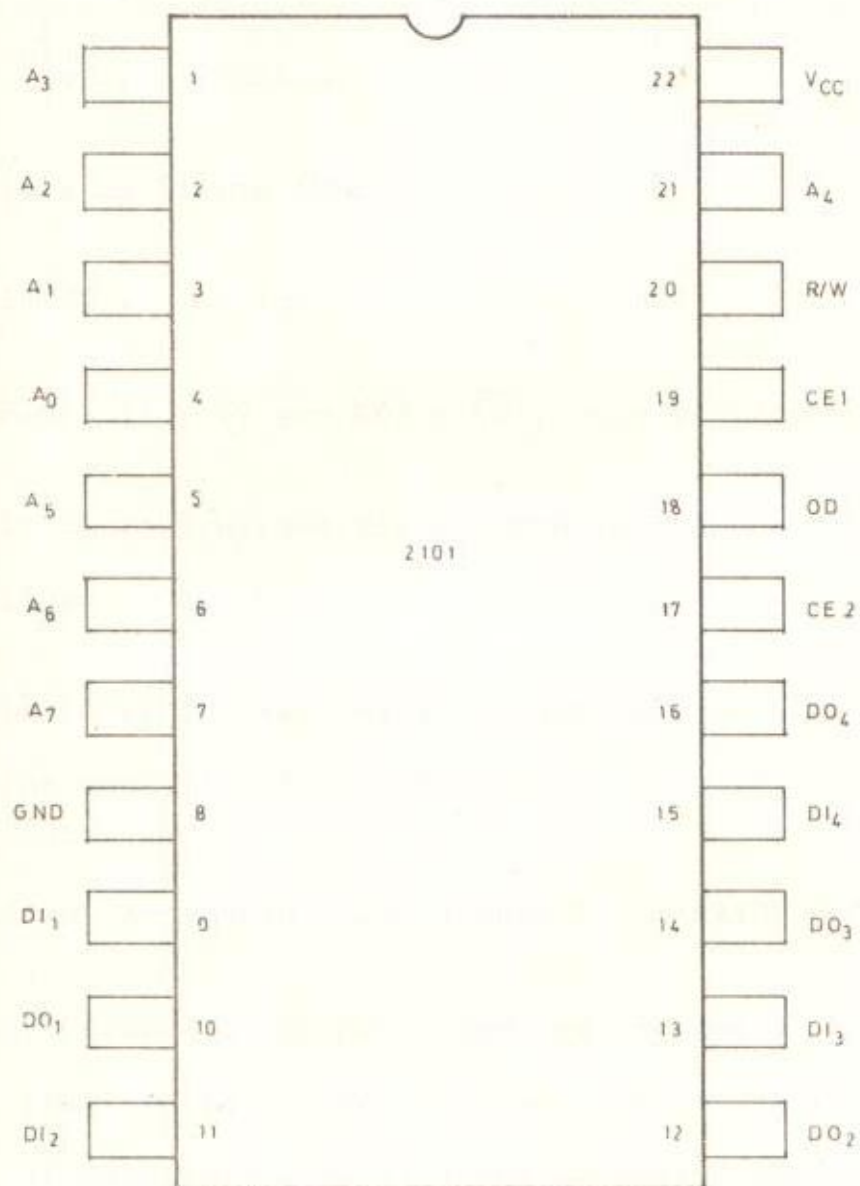


FIG. 4-51

CONFIGURACION DE PINES DEL C.I. 2101

a los datos de entrada.

Los pines 10, 12, 14 y 16 son D01-D04 que estan asignados a los datos de salida.

El Pin 8 es Tierra (GND).

El Pin 22 es Vss (5V).

Los Pines 17 y 19 son CE2 y  $\overline{CE1}$ , que activan el Chip.

El Pin 20 es R/W, que sirven para seleccionar lectura o escritura en el chip.

El Pin 18 es OD, esta es la salida de desactivo, es decir elimina cualquier lógica bidireccional.

#### 4.6 "4702 A" (256x8) PROM BORRABLE CON RAYOS ULTRAVIOLETAS

El 4702 A es un ROM ( Memoria unicamente de lectura) programable de 256 palabras de 8 bits y es ideal para desarrollos de sistemas de Microcomputadoras.

El 4702 A es empaquetado en un envase de 24 pines y tiene una ventanita transparente de cuarzo.

La ventanita de cuarzo transparente permite al usuario exponer el chip a la luz ultravioleta para borrar el patrón de bits. Se puede escribir entonces un nuevo patrón dentro del dispositivo.

Este procedimiento puede ser repetido tanta veces como sea requerido.

El Circuito 4702A es enteramente estático y por lo tanto no necesita pulsos de reloj.

El 4702 A es fabricado con tecnología Lógica de Silicón.

En la figura 4-61 tenemos un diagrama de bloques explicativo de la estructura interna del PROM 4702 A.

En la figura 4-62 se muestra el circuito Integrado 4702 A con la configuración de Pines.

Los Pines 1-3 y 17-21 son A0-A7, esto es la entrada de las direcciones al PROM.

Los Pines 4-1 son D01-D07, que son los datos de salida del PROM. Estos Pines son las entradas de las instrucciones cuando se esta programando el PROM.

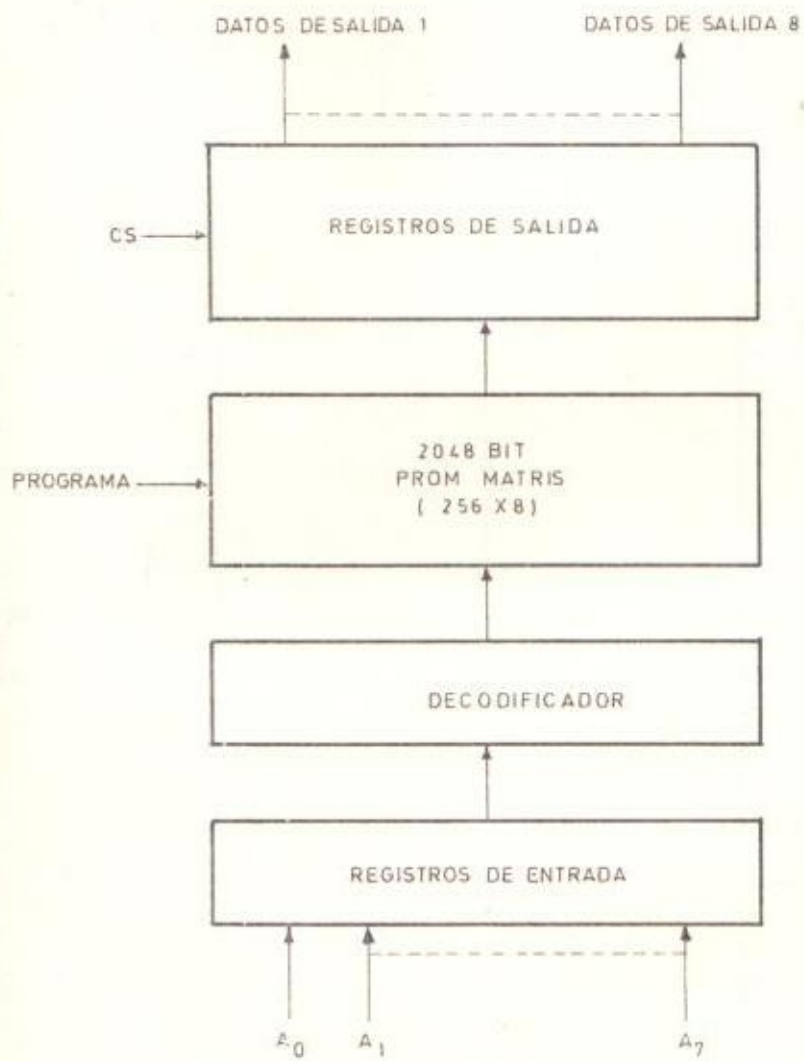


FIG 4-61  
DIAGRAMA DE BLOQUES DEL C.I. 4702 A

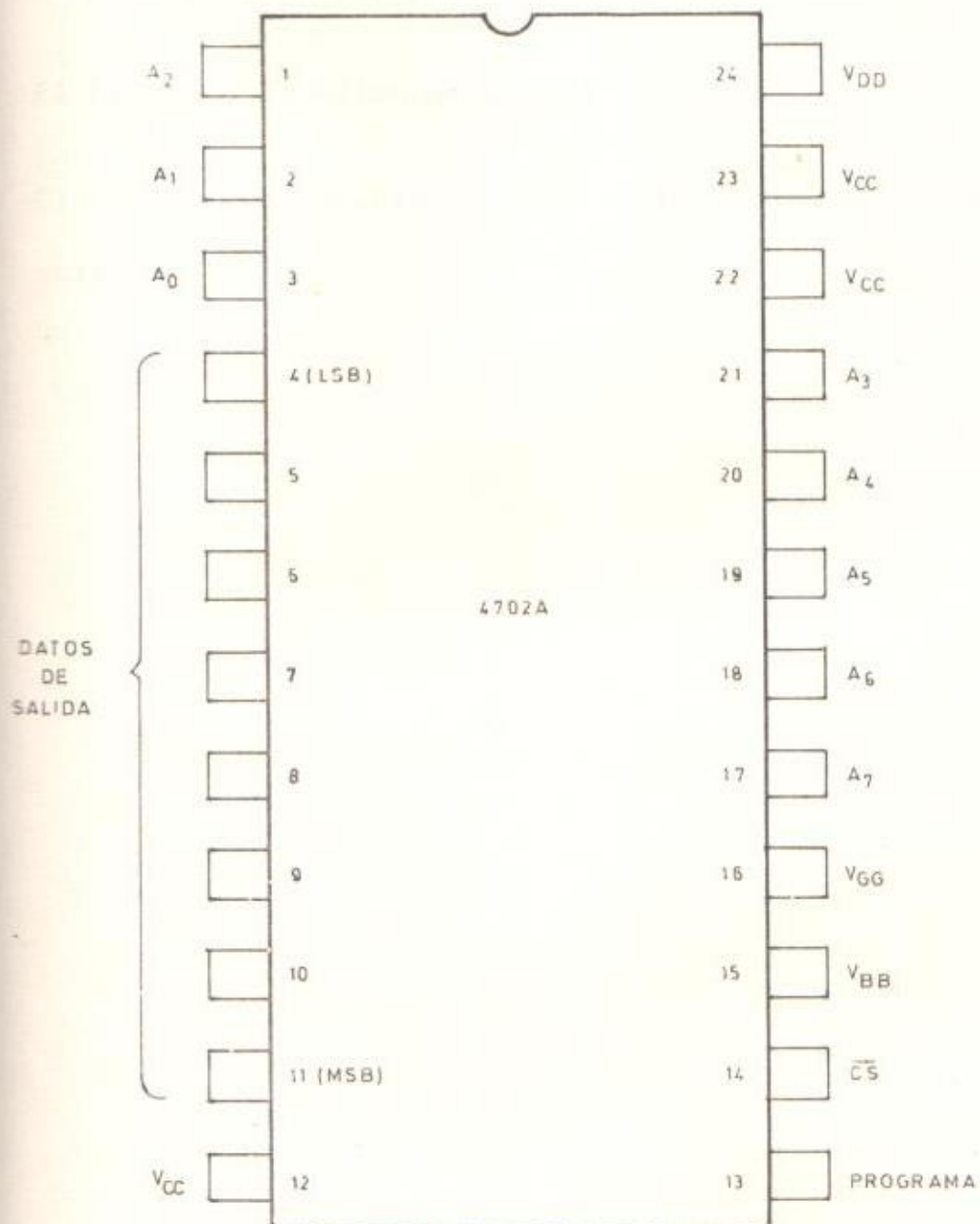


FIG. 4-62

CONFIGURACION DE PINES DEL C.I. 4702 A

Los Pines 12, 13, 15, 22, y 23, estan asignados a  $V_{cc}(+5V)$

El Pin 16 y 24 estan asignados a  $V_{dd}(-10V)$ .

El Pin 14 esta asignado a  $\overline{CS}$ . Esta Línea selecciona el chip.

## CAPITULO V

## 5. DISEÑO Y CONSTRUCCION DEL CIRCUITO PARA PROGRAMAR EL PROM 4702 A.

El programador que se diseñó y construyó es una máquina de grabación y se la hizo específicamente para la familia de los EPROM'S (Memoria solamente de lectura que es Programable y Borrable) 1702A, 4702A y 8702A.

En el Laboratorio de Digitales de la ESPOL, se lo utilizó para programar el EPROM 4702A, que luego se lo utilizó en el sistema del microcomputador MCS-40.

La más grande utilidad de los EPROM'S esta en el campo del Microcomputador. Por ejemplo, hay una cantidad de diferentes Monitores (Instrucciones de un Programa grabados en un EPROM) disponibles para cada microprocesador. Aunque los monitores comparten algunas instrucciones comunes, algunos tienen más o menos dispositivos que otros y por tanto la cantidad y secuencia de instrucciones de los monitores, son diferente.

Para el Microcomputador es muy necesario el monitor, ya

que sin él no podría operarse al encenderlo.

Uno podría cargar los datos del monitor de una máquina de cassette de cinta magnética, pero este es un método engorroso. Para mejor conveniencia y velocidad, se debe tener un programa monitor en un ROM ya que al encender el equipo uno está listo para operarlo y además cuando hay pérdidas momentáneas de energía, no se pierde el programa monitor.

La ventaja de un EPROM sobre un ROM es que el primero puede ser borrado y reprogramado muy fácilmente.

El Patrón de Bits del 4702 A puede ser borrado exponiendo la ventana de cuarzo del chip a la luz ultravioleta por es pacio de unos 20 minutos.

Los fotones de luz ultravioleta desplazan los electrones que fueron inducidos en el silicón y forman el equivalente de "unos" en el patrón de bits.

Usando el Programador EPROM, la memoria puede ser regrabada electrónicamente. Una vez que se ha programado, los da tos se mantendrán en la memoria aunque se apague el microcomputador.



Los datos se pueden perder cuando el EPROM se queme o cuando se lo haya expuesto a la luz ultravioleta.

El EPROM 4702A es completamente estático, es fácil instalarlo por medio de un circuito de interfase al procesador, no necesita pulsos de reloj, y sus entradas y salidas son compatibles con la lógica TTL. Requiere una fuente de poder de -10V y una de +5V.

El Programador que se hizo en el laboratorio de la ESPOL consiste en un dispositivo que utiliza switches para la selección de direcciones y de datos.

El diagrama I, nos muestra el circuito que provee todos los pulsos de tiempo que necesita el programador.

El Reloj de 33.3 KHZ es generado por ICI que es enviado a un registro de desvío de 8 bits, el IC7 y a un contador sincrónico en cadena que consiste de IC2, IC3, IC4, IC5, e IC6. Los circuitos integrados IC4, IC5 e IC6, conjuntamente con IC8D, forman un Contador-Divisor de 430, cuya salida de Carry activa un Contador-Divisor de 32, formado por IC2 e IC3.



El condensador C5 y la Resistencia R12, proveen la Potencia de inicialización de la cadena.

La acción del circuito comienza con el comando de programación (PROG), que es un pulso negativo usado asincrónicamente para limpiar IC2 e IC3 necesitando que el ancho del pulso de Comando de Programación sea de por lo menos 5 milisegundos. La salida Qb de IC2 (PIN 13) es invertido por IC8C para controlar la operación del Contador-Divisor de 430. La Salida de este divisor es tomado de IC4 (Pin 11) y es 7752 Hz(33.3Khz/430), siendo el ciclo de 2.58ms positivo y de 10.32ms negativo. Esta señal es aplicada a la entrada serial de IC7 que causa a sus 8 salidas a ponerse en un nivel lógico alto, siguiendo una secuencia de 30useg de intervalo (el periodo de reloj). La salida Qb de IC1 empieza el ciclo de programación encendiendo la fuente de +47 voltios a través de Q2 de la fuente de Poder mostrada en el diagrama II.

Esta sección del circuito establece las líneas de direcciones y datos a sus propios niveles. La dirección es completa en este tiempo.

Teinta microsegundos más tarde, los voltajes Vdd y Vgg, se

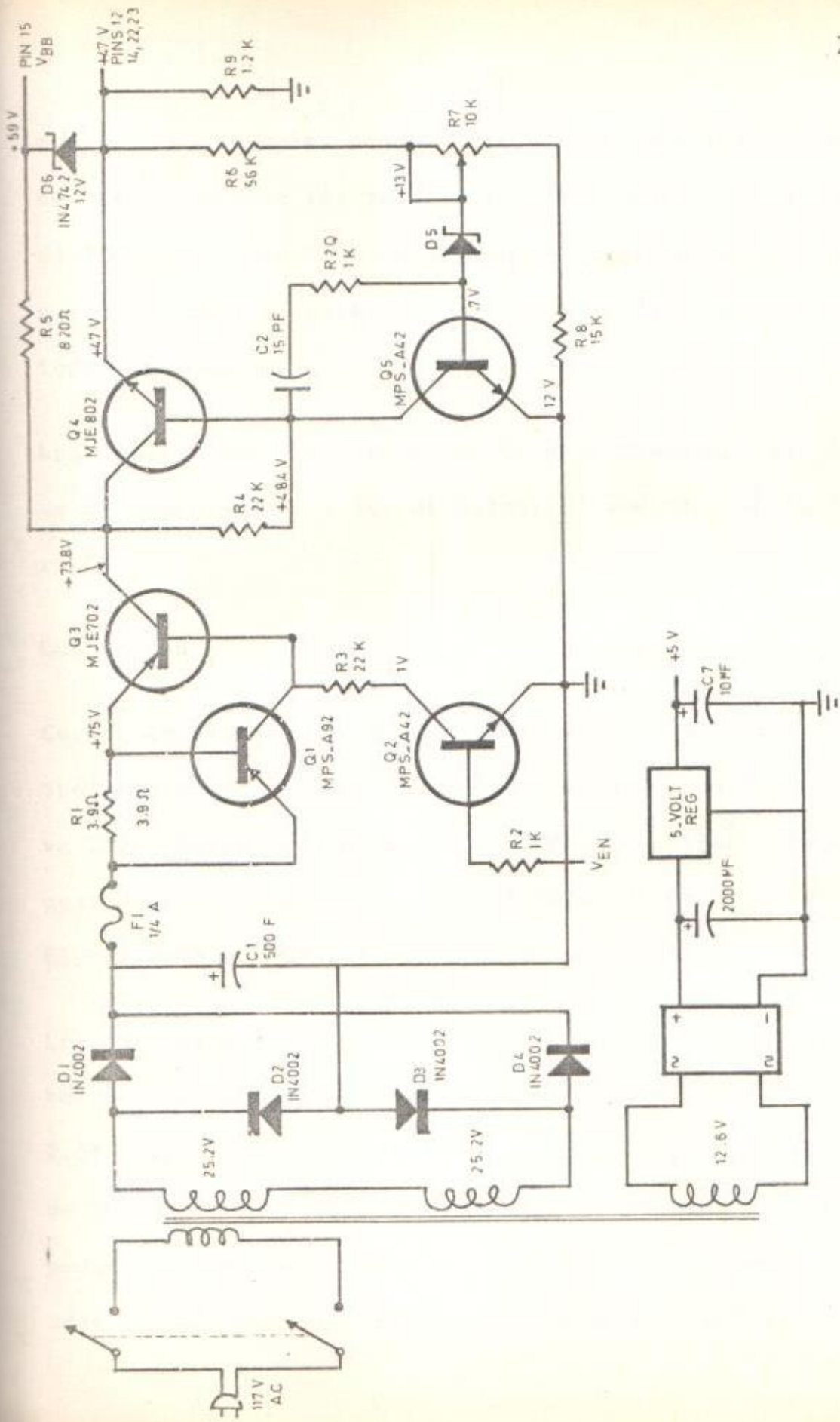


DIAGRAMA II FUENTE DE VOLTAJE

mueven a sus niveles negativos controlados por la salida C de ICA que maneja los transistores Q7 y Q8. La salida D de IC7 (T/C verdadera o complemento) continúa al próximo pulso de reloj e invierte las líneas de dirección a su estado verdadero.

Las señales Vdd, Vgg, el pulso de Programación, las líneas de las direcciones y las de datos, se muestran en la figura 5-1.

Las salidas E, F y G de IC7 no son usadas.

Cuando la salida H va a un alto lógico (esta conectada conjuntamente con la salida A ó una puerta "y" de IC8A) activa IC8A, luego pasa a través de IC8B que invierte la señal para manejar Q6 generando así el Pulso de Programa en 501. El circuito de 501 esta en el diagrama III.

Las direcciones que mencioné anteriormente son estables hasta la terminación de el Pulso de Programa que salió de 2.58 mseg. en el Laboratorio. A este tiempo la salida A de IC7 se va a un bajo lógico, finalizando así el pulso de programa. Entonces la salida B desactiva la fuente de voltaje de +47 voltios. El registro de desvío IC7 es comple-

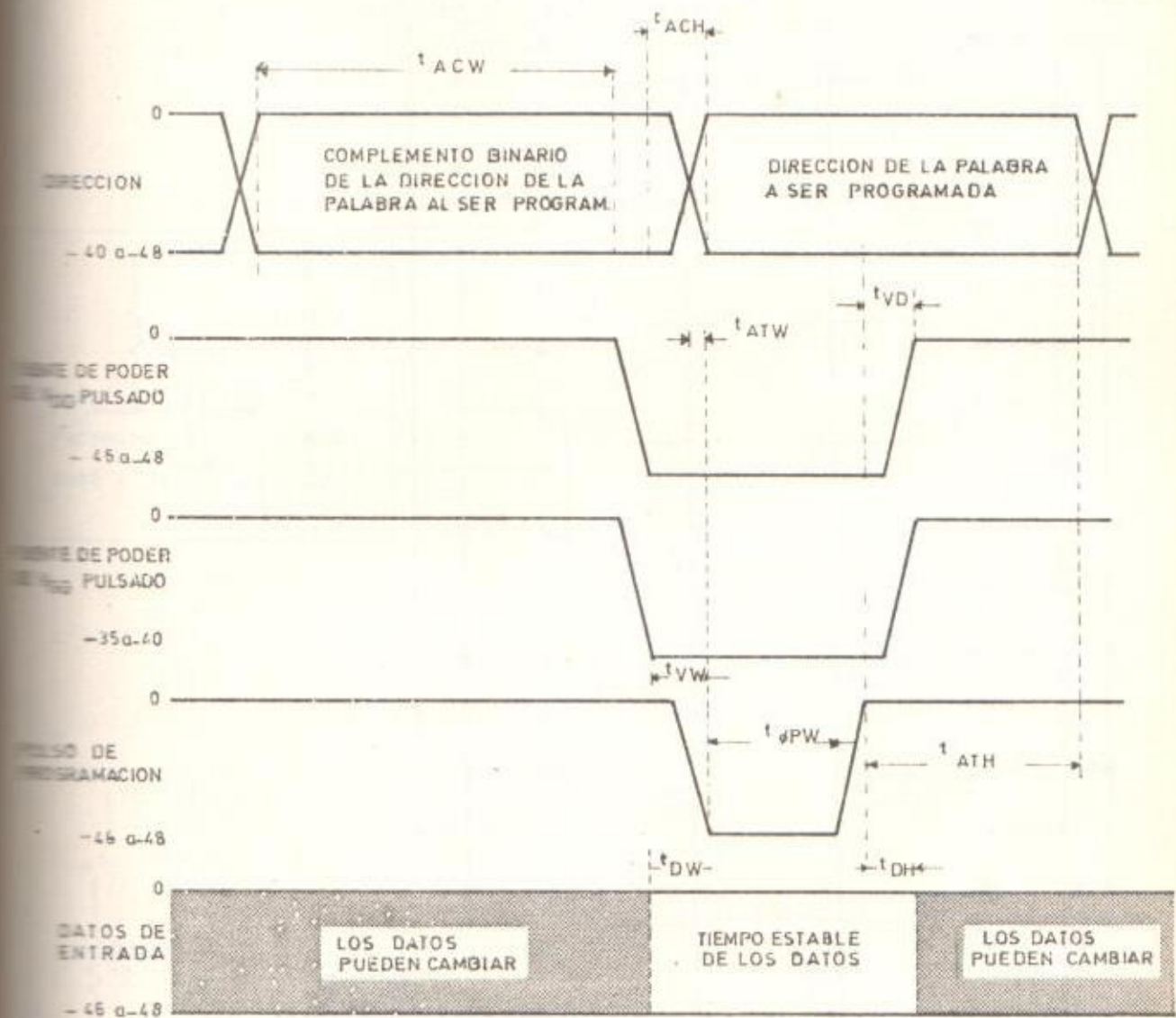


FIG. 5-1

PULSOS DE PROGRAMACION, DIRECCION Y DE DATOS

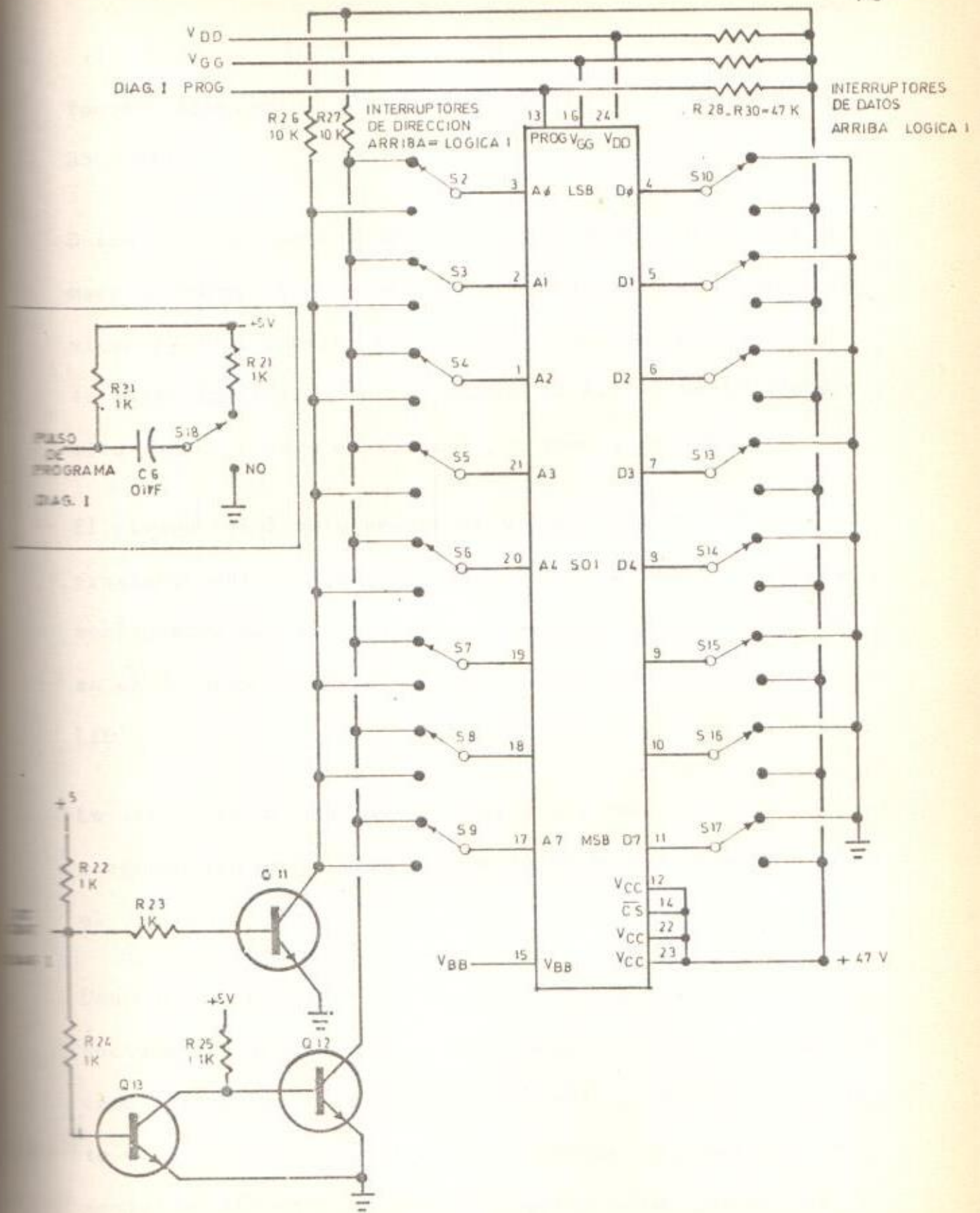


DIAGRAMA III CIRCUITOS DEL 4702A

tamente limpiado en 6 pulsos más de reloj de la señal de 33.3 Khz.

Durante los últimos 30 microsegundos de el pulso de 2.58 mseg un carry es generado por IC4, causando al Contador-Divisor de 32 a avanzar (IC2, IC3). Esta secuencia se repite hasta la iteración 32, cuando el Pin 13 de IC2 va a un alto lógico y para el contador, a través de IC8C.

El tiempo total para programar una palabra de 8 bits es aproximadamente 413 milisegundos. Este periodo puede ser monitoreado por el indicador de estado opcional mostrado en el diagrama III y que esta formado por Q10, R18, R19 y LED1.

La selección de las Direcciones y los Datos, durante la programación se la hizo a base de 16 switches mostrados en el diagrama III.

Una Lógica "C" sobre las líneas de las direcciones esta conectada por un interruptor al conector de Q11. Usando Vcc como referencia, resultará a un nivel de -47 voltios durante el pulso de programa cuando la señal T/C (verdad-complemento) de IC7 esté en un alto lógico, seleccionando asi la



dirección verdadera. Colocando el switch de dirección en la posición 1, conecta esa línea al complemento de la señal presente en el colector de Q12, resultando una lógica 1.

Los datos de entrada se conectan a una línea de salida a tierra a través del switch y de la misma manera se conectan a -47 voltios.

El Pulso de Comando de Programación se lo genera presionando el switch S18 mostrado en el diagrama III.

La fuente de voltaje mostrada en el diagrama II libera aproximadamente + 75 voltios a un limitador de corriente que consiste de Q1, Q2, Q3, R1, R2 y R3. Los elementos, Q4, Q5 R5, R6, R7 y R8 regulan la salida de +75 voltios a + 47 voltios. El diodo D6 y la resistencia R5 proveen al voltaje de polarización Vbb. La resistencia R9 asegura una carga mínima al regulador y provee una vía de corriente al diodo zener D6. El condensador C2 y la resistencia R 20 evitan que oscile el circuito de alta ganancia de Q5.

La figura 5-2 nos muestra el negativo de la fotografía de el circuito impreso del Programador de EPROM'S.

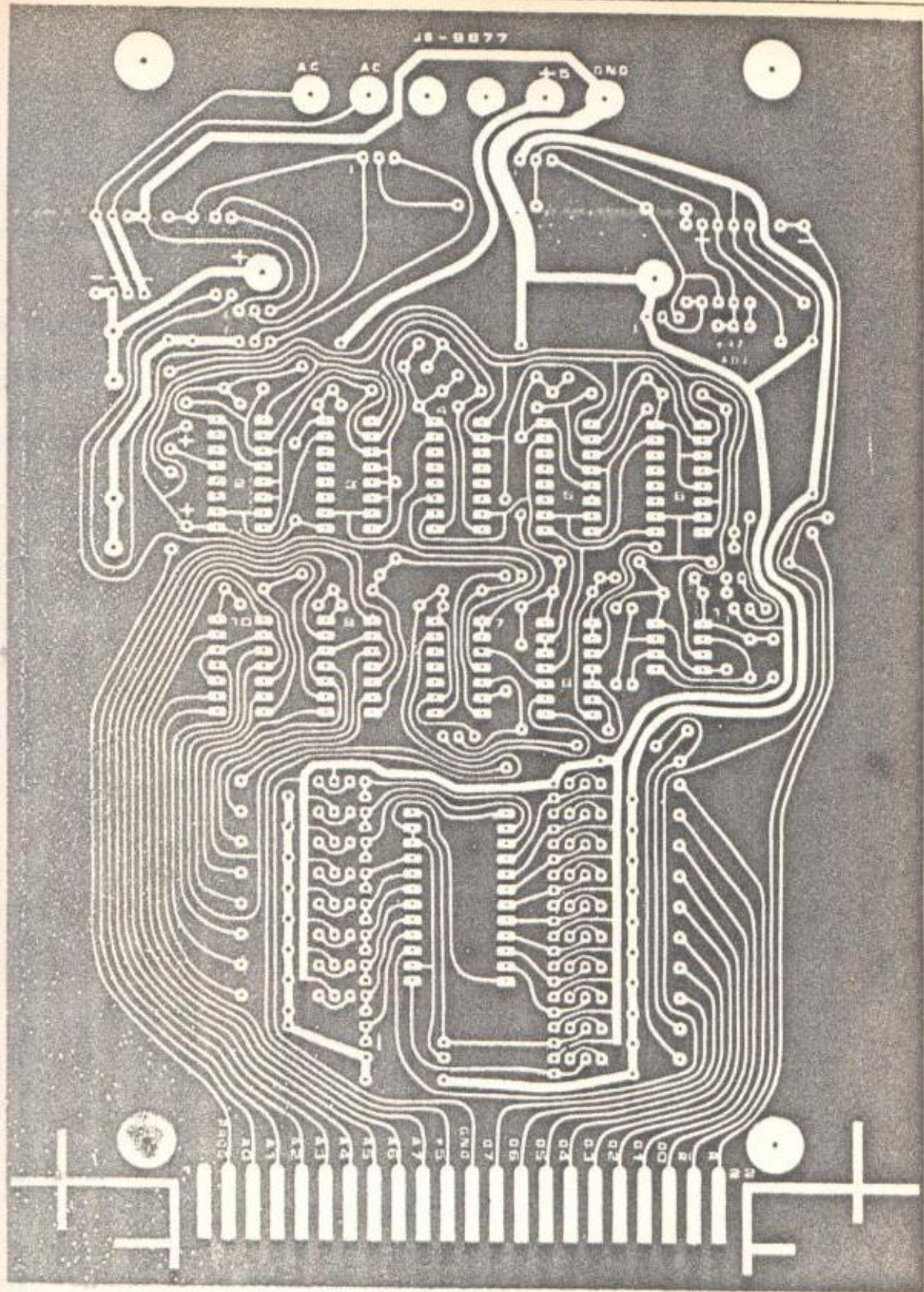


FIGURA 5-2 NEGATIVO DEL CIRCUITO IMPRESO DEL PROGRAMADOR DE EPROM'S

En la tarjeta de circuito impreso se instaló los circuitos que muestran los diagramas I y II.

Los circuitos de los diagramas II y III se los montó en un Proto-Board y luego se conectó por medio de alambres la tarjeta del circuito impreso con los circuitos instalados en el Proto-Board como muestra la fotografía en la figura 5-3.

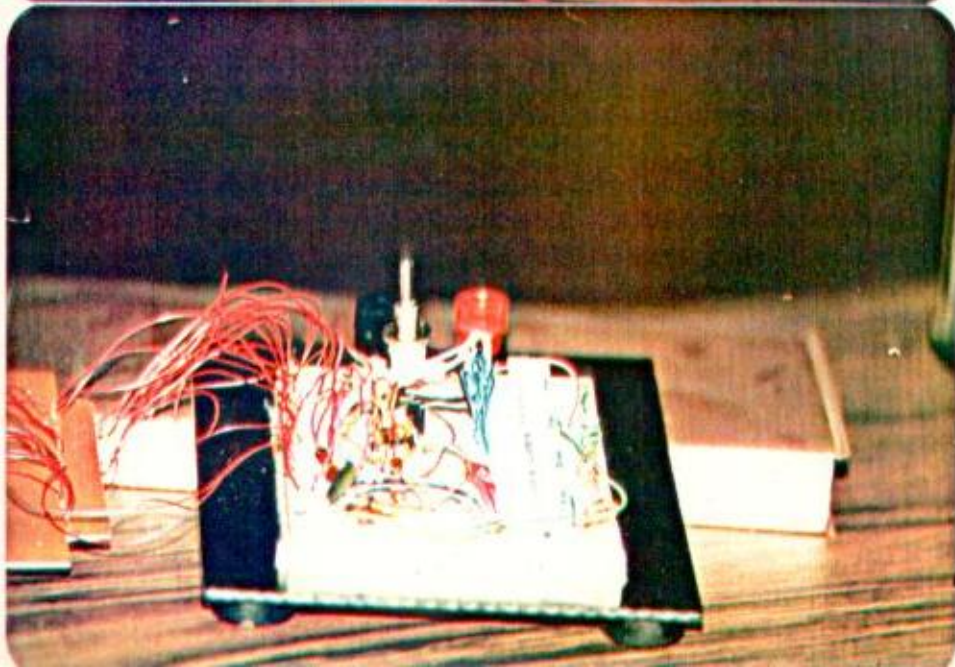
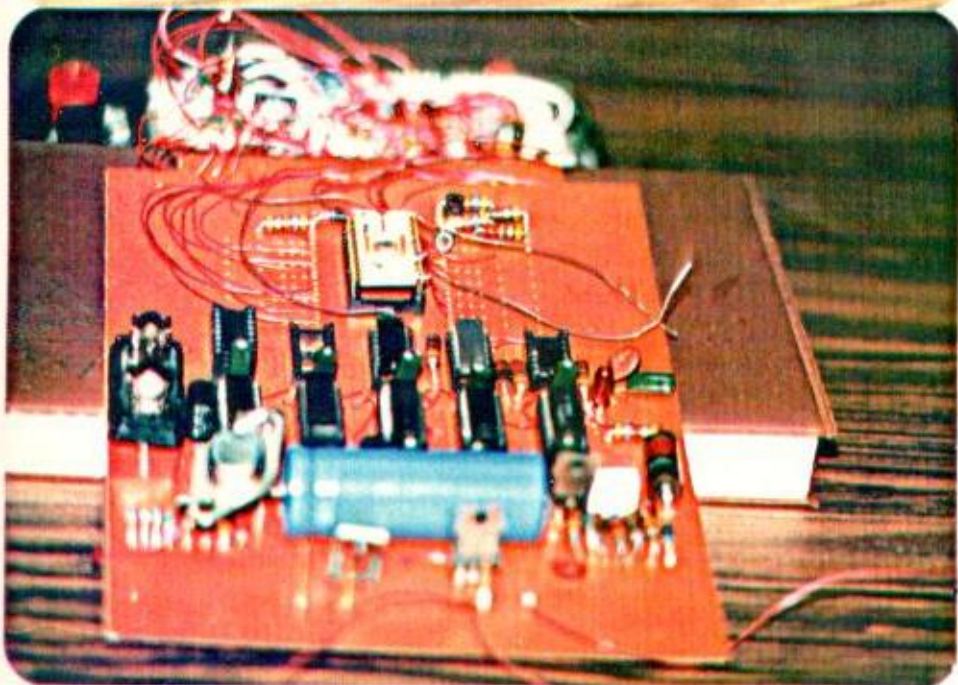
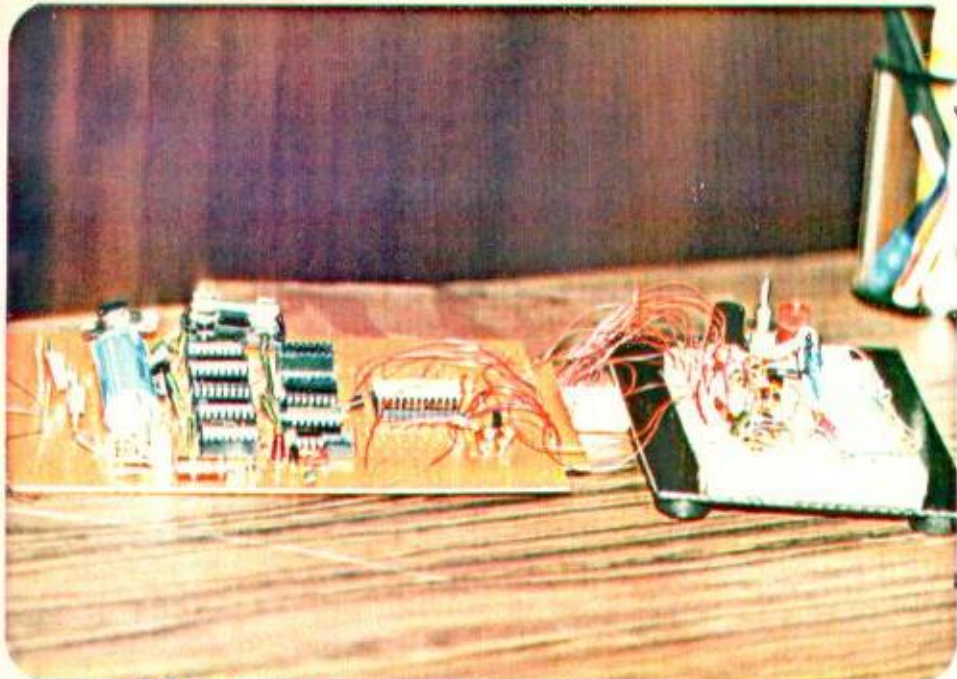


FIGURA 5-3 PROGRAMADOR DEL EPROM 4702A

## CAPITULO VI

## 6. MONITOR PARA CONTROL DE TECLADO Y SALIDA VISUAL.

El monitor es un conjunto de instrucciones en secuencia (un programa) que estan almacenados en una memoria.

Este programa existente en el memoria debe ejecutarse cada vez que se enciende el computador y activar el Procesador Central y sus Periféricos, es decir, ponerlos listos para cualquier entrada o salida de datos que el operador quiera efectuar.

Cada instrucción del monitor manipula datos en alguna forma. Los datos pueden ser los contenidos de el Contador de Programa, el cual indica donde se va a encontrar la proxima instrucción en la memoria, el Acumulador, el Flip-Flop de carry, los contenidos de un RAM ó ROM, o las señales en una Puerta.

La memoria en la que se encuentra el monitor debe ser no volátil a la falta de energía, ya que es imprescindible que al momento de encender el computador, este se

ponga listo para poder empezar a trabajar en él.

Esta memoria no volátil debe de ser un ROM ó un EPROM (como es el caso en esta Tesis que se utiliza un EPROM 4702) ya que una vez grabados no se borran a la falta de voltaje.

El EPROM (Memoria de solamente Lectura que es Borrable y Programable) 4702A tiene la bondad de poder ser borrado cuando se quiere cambiar el programa o los datos grabados en él. Un exceso de voltaje puede quemar el EPROM.

Con una lámpara de rayos ultravioletas y exponiendo el 4702A a una intensidad por tiempo de exposición de 6 voltios-seg/cm<sup>2</sup>, se logra borrarlo. En el Laboratorio de Comunicación de la ESPOL se borró un 4702A exponiendolo a la luz ultravioleta por un tiempo de 20 minutos. Luego se reprogramó en él.

El monitor que se ha hecho en este trabajo consiste en un programa básico demostrativo que al encender el Microcomputador MCS-40 lo pone listo y entonces se puede trabajar en él.

El monitor esta constantemente haciendo una búsqueda por

intermedio de las líneas sensoras (S0-S1) y las líneas de retorno (R0-R7) del circuito integrado de interfase 4269 al teclado, es decir, el monitor hace que el CPU esté atento a la presión de cualquier tecla. Una vez que se ha detectado una tecla, el caracter correspondiente a la misma, es sacado a un dispositivo de video por intermedio de las líneas A0-A3 y B0-B3 del C.I. 4269 de Interfase.

Cuando dos ó mas teclas se han presionado dentro del tiempo completo ( aproximadamente 11 milisegundo) de búsqueda de una sola tecla, se dice que es una presión simultánea de teclas. Entonces el PKD (4269) continúa la búsqueda del teclado y espera hasta que una sola tecla permanezca presionada.

Cuando una sola tecla ha sido presionada un contador interno empieza a contar. El código de la tecla es también almacenado internamente en un registro temporal del PKD(4269) con un código que lo dan las líneas de búsqueda y de retorno. El PKD hace 4 búsquedas completas adicionales a todas la teclas. Si no encuentra otra tecla presionada durante las 4 búsquedas completas y si todavía se encuentra presionada la tecla original al final de la cuarta búsqueda, el

El Código de la tecla es almacenado en el FIFO (registro secuencial, primero que entra primero que sale), (memoria del 4269 destinada al teclado). La figura 6-1 nos muestra la estructura del FIFO.

En el FIFO puede entrar hasta 8 caracteres. Nuestro monitor ha sido programado para utilizar 6 posiciones del FIFO. Una vez que han entrado 6 caracteres y estos han sido almacenados en los registros de video por instrucciones del monitor, el FIFO es limpiado y nuevamente estará activo para seguir almacenando más caracteres, es decir, los caracteres pueden ser almacenados consecutivamente en el FIFO, ya que éste una vez que se ha llenado se limpia. No hay pérdidas de caracteres ya que a medida que entra un caracter es enviado a registros de video.

Cuando un caracter ha entrado al FIFO, la línea de interrupción es activada a la lógica "1"(VDD1).

Si a la cuarta búsqueda completa de teclado, la tecla original no se ha mantenido presionada, es ignorada como si nunca antes hubiera sido presionada. El tiempo de las 4 búsquedas es aproximadamente de 11 mseg. El PKD mantiene



F I F O  
=====

FIRST INPUT - FIRST OUTPUT  
PRIMERO ENTRA - PRIMERO SALE

POSICION 0	PRIMER CARACTER
POSICION 1	SEGUNDO CARACTER
POSICION 2	TERCER CARACTER
POSICION 3	CUARTO CARACTER
POSICION 4	QUINTO CARACTER
POSICION 5	SEXTO CARACTER
POSICION 6	SEPTIMO CARACTER
POSICION 7	OCTAVO CARACTER

FIG. 6-1. FIFO

así una función de depurado para el teclado.

El monitor de este Microcomputador MCS-40 esta grabado en el EPROM 4702A que consiste en una Página de 256 bytes, en cada byte es de 8 bits.

Las direcciones para cada posición de memoria del 4702A contienen 12 bits. Los 4 bits de orden más alto son usados cuando el Microcomputador esté trabajando con más de un 4702A. En este caso los 4 bits de orden más alto siempre serán los mismos porque se utiliza sólo un 4702A. Los 4 bits de orden más alto se los llama PH, los 4 bits de orden medio se los llama PM. Los 4 bits de orden más bajo se los llama PL.

A continuación presento el conjunto de instrucciones que forman el monitor que van a controlar la entrada de caracteres del teclado y su salida visual en el microcomputador MCS-40.

DIRECCION BINARIA EN EL 4702A			INSTRUCCIONES		
PH	PM	PL			
0000	0000	0000	FIM	0010	0000
0000	0000	0001		0100	0000
0000	0000	0010	SRC	0010	0001
0000	0000	0011	LDM	1101	1101
0000	0000	0100	WRD	1110	0100
0000	0000	0101	WRZ	1110	0110
0000	0000	0110	RD1	1110	1101
0000	0000	0111	WMP	1110	0001
0000	0000	1000	RDM	1110	1001
0000	0000	1001	RD2	1110	1110
0000	0000	1010	WRM	1110	0000
0000	0000	1011	RDO	1110	1100
0000	0000	1100	FIM	0010	0010
0000	0000	1101		1000	0000
0000	0000	1110	ADD	1000	0100
0000	0000	1111	ICN	0001	0010
0000	0001	0000		0000	0010
0000	0001	0001	JUN	0100	0000
0000	0001	0010		0000	0110

A continuación se explicará el significado de cada una de las instrucciones y el trabajo de cada una de ellas, pero antes quiero hacer una explicación breve de lo que son los registros de video A y B, así como también los Registros de Índices y los Registros de Índices Dobles.

Los registros A y los Registros B son exactamente iguales. Estos son los registros de video, es decir, aquí son almacenados los caracteres que se los va a ver en el dispositivo de video.

Estos registros son de  $16 \times 4$  bits. La figura 6-2 nos muestra la configuración de estos registros.

Los registros de Índices y los Registros de Índices Dobles tienen la misma localidad en la memoria y se los identifica de acuerdo a la instrucción que se utilice.

Los registros de Índices son 16, del 0 al 15 y cada uno de 4 bits.

Los registros de Índices Dobles son 8 del 0 al 7 y cada uno de 8 bits.

La figura 6-3 nos muestra un gráfico de la configuración de estos registros.

REGISTROS DE VIDEO A Y B

REGISTRO A		REGISTRO B	
4 bits	A0	B0	4 bits
	A1	B1	
	A2	B2	
	A3	B3	
	A4	B4	
	A5	B5	
	A6	B6	
	A7	B7	
	A8	B8	
	A9	B9	
	A10	B10	
	A11	B11	
	A12	B12	
	A13	B13	
	A14	B14	
	A15	B15	

FIG. 6-2 REGISTRO DE VIDEO DE 16x4 Bits

REGISTROS DE INDICES  
SOLAMENTE

Número del Registro	14	15
	12	13
	10	11
	8	9
	6	7
	4	5
	2	3
	0	1

(4 bits)

REGISTROS DE INDICES  
PARES

Número del Registro Par	14
	12
	10
	8
	6
	4
	2
	0

(8 bits)

FIG. 6-3 REGISTROS DE INDICES SOLAMENTE (4bits)  
REGISTROS DE INDICES PARES ( 8 Bits).

Una explicación más exhaustiva de estos registros la haré a continuación en el desarrollo de la instrucciones del Monitor.

La primera instrucción que se utiliza en este monitor es FIM, que consta de 2 palabras (cada palabra es un byte). La primera palabra consta de dos nibles (Un nible son 4 bits).

NEMONICO: FIM (Obtención inmediata de datos del EPROM) 1ra. palabra OPR OPA: 0010 RRR0.

2da. Palabra OPR OPA: D2 D2 D2 D2 D1 D1 D1 D1.

El OPR o Código de Operación de la primera palabra identifica la instrucción como FIM y el OPA ó modificador de la primera palabra define el Registro de Indices Par. donde va a ser almacenado la segunda palabra que son los datos.

El RRR del OPR de la primera palabra puede ser de 0 a 7, apuntando así el Registro de Índice Doble, que se va a utilizar. Como cada Registro de Índice Doble tiene 8 bits, entonces D1D1D1D1 serán almacenados en los 4 bits inferiores y D2D2D2D2 serán almacenados en los 4 bits superiores.

En nuestro monitor la instrucción FIM ha definido que en el registro de Indices Doble cero se almacenen los datos 0100

0000.

La siguiente instrucción es SRC que consta de una sola palabra.

NEMONICO: SRC.

OPR OPA : 0010 RRR1

RRR define el Registro de Indice Par.

En nuestro monitor SRC es: 0100 0001.

Esta instrucción toma los datos del Registro de Indice Doble cero (anteriormente almacenados) 0100 0000 y que van a definir lo siguiente:

Los dos bits de orden superior (01) definen del CI 4269, que se lo asigna por medio de la línea CM-RAM.

Los dos bits siguientes (00) definen el registro de video A, donde se a empezar a almacenar los datos de video.

Los cuatro bits de orden inferior (0000) indican que el contador de los registros de video apunta a la posición cero (en este caso es la posición cero del registro A).

La siguiente instrucción es LDM, que es de una sola palabra NEMONICO: LDM



OPR OPA: 1101 DDDD

Los bits DDDD son datos que seran almacenados en el acumulador. En nuestro monitor tenemos la instrucción LDM así:

1101 1101

Los dos bits de orden superior (11) del modificador nos van a definir el 4269 para el video en Modo de Descarga de Gas de 16 caracteres y los dos bits de orden inferior del modificador (01) nos definen el 4269 para el teclado en el Modo de búsqueda de Teclado.

La siguiente instrucción es WRO que es de una sola palabra

NEMONICO: WRO

OPR OPA: 1110 0100

Esta instrucción toma los datos del Acumulador y los graba en la memoria del 4269, estableciendo así los modos de Entrada-Salida, es decir, los modos de Video y Teclado.

La próxima instrucción es WR2 que es de una sola palabra.

NEMONICO: WR2

OPR OPA: 1110 0110

Esta instrucción hace una limpieza total del FIFO y deshabilita

lita la línea INT(Interrupción) que se pondrá activa en el momento que se detecte la presión de una tecla en el teclado.

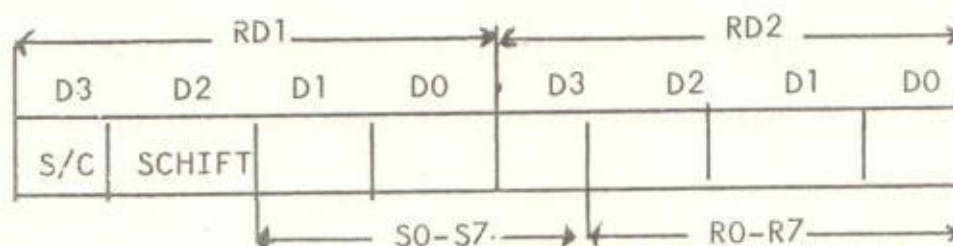
La siguiente instrucción es RD1, que consta de una sola palabra.

NEMONICO: RD1

OPR OPA: 1110 1101

Esta instrucción espera que una tecla haya sido presionada y almacenada en el FIFO, entonces lee el primer nibble del FIFO y lo graba en el acumulador.

En el FIFO la intervención de un caracter está almacenada de la siguiente forma:



S0-7 y RD-7 son los bits que identifican el caracter.

Para nuestro monitor la matriz de los caracteres es la siguiente:

		R0	R1	R2	R3	R4	R5	R6	R7
		000	001	010	011	100	101	110	111
S0	000	0	1	2	3	4	5	6	7
S1	001	8	9	A	B	C	D	E	F

Las teclas S/C y Shift no se las utilizó

La siguiente instrucción es WMP que es de una sola palabra.

NEMONICO: WMP

OPR OPA: 1110 0001

Esta instrucción toma los datos ya grabados en el acumulador y los envía al Registro de Video a posición cero.

La siguiente instrucción es RDM.

NEMONICO: RDM

OPR OPA: 1110 1001

Esta instrucción incrementa el contador del Registro de Video. Ahora apuntará al Registro B posición cero. Al incrementar el contador existe un cambio alterno de A0 a B0 a A1 a B1 a A2 etc.

La siguiente instrucción es RD2 que es de una sola palabra.

NEMONICO: RD2

OPR OPA 1110 1110

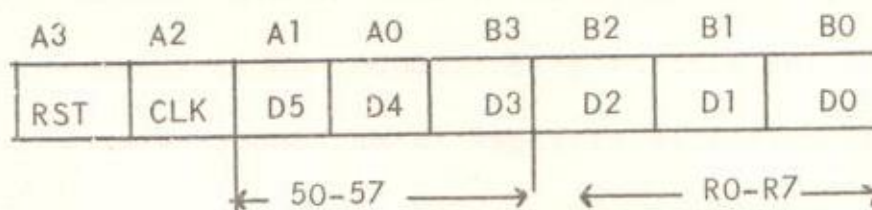
Esta instrucción lee el segundo nibble de FIFO y lo almacena en el Acumulador, además incrementa al contador del FIFO.

La siguiente instrucción es WRM, que es de una sola palabra NEMONICO: WRM

OPR OPA: 1110 0000

Esta instrucción toma los datos del acumulador y los envía al Registro de Video B, posición cero. Además genera los pulsos de reset (RST) y reloj (CLK), que son las salidas A3 y A2 del CI 4269 y que van a actuar en el CI 4003 (Registro de desvío) del circuito de Video. En este momento en que podemos ver en el dispositivo de video, el caracter que hemos presionado en el teclado.

La salida en los registros de video es la siguiente:



Además esta instrucción incrementa el contador de registros de video, es decir, ahora apuntará al registro A posición U no.

La siguiente instrucción es RDO, que es de una sola palabra

NEMONICO: RDO

OPR OPA: 1110 1100

El FIFO tiene un contador que lleva la cuenta de caracteres que han sido almacenados en él, entonces a este contador se lo llama el Estatus del FIFO.

La instrucción RDO toma el Estatus del FIFO y lo almacena en el acumulador. Así podemos determinar la cantidad de ca racteres que van siendo digitados.

La próxima instrucción que tenemos en nuestro Monitor es FIM, que ya la usamos antes y es de dos palabras.

NEMONICO: FIM

OPR OPA: 0010 0010

OPR OPA: 1000 0000

Con esta instrucción almacenamos en el Registro de Indices doble # 2 los datos (1000 0000). Esto también equivale a almacenar los datos (1000) en el Registro de Indices #4.

Este número 8 lo vamos a utilizar para limitar los caracteres en el dispositivo de video que aunque puede contener 10 caracteres, sólo podemos utilizar 8 debido a que la capacidad del FIFO es de 8 dígitos.

La siguiente instrucción es ADD, que es de una sola palabra

NEMONICO: ADD

OPR OPA: 1000 RRRR (0100)

El contenido de 4 bits del Registro de Indices # 4 es sumado al contenido del acumulador.

El resultado es almacenado en el acumulador. Si la suma es más grande que F(hexadécimal), entonces se establece el carry a "1" lógico, de otra manera el carry es establecido a "0" lógico. El contenido del Registro de Indices no es afectado.

La siguiente instrucción es JCN, que es de dos palabras.

NEMONICO: JCN

1era. palabra OPR OPA: 0001 C1 C2 C3 C4.

2da. palabra OPR OPA: A2 A2 A2 A2 A1 A1 A1 A1.

C1 C2 C3 C4 establece una condición de salto dependiendo si es verdad o no. En nuestro caso C3 es el Carry, es decir, si C3 es "1", se hará un salto a la dirección determinada por A2 A2 A2 A2 A1 A1 A1 A1, de lo contrario se continuará con la próxima instrucción.

En nuestro monitor ICN esta definida así:

Instr. No. ELEC. - 012

JCN: 0001 0010

0000 0010

Si la suma que acabamos de hacer excede de F (hexadécimal), quiere decir que ya hemos pasado la posición # 8 del teclado y entonces el programa irá a la dirección # 2 (0000 0010). del EPROM donde inicializarán el FIFO y los registros de video nuevamente.

Si la suma no excede de F, el programa continuará con la próxima instrucción.

La última instrucción del monitor es JUN, que es de dos palabras.

NEMONICO: JUN (Salto incondicional)

1era. palabra OPR OPA: 0101 A3 A3 A3 A3.

2da. palabra OPR OPA: A2 A2 A2 A2 A1 A1 A1 A1

En nuestro monitor A3 A3 A3 A3 es 0000 debido a que el salto incondicional es hecho en la misma página de 256 bytes, es decir dentro del mismo EPROM 4702A.

La segunda palabra A2 A2 A2 A2 A1 A1 A1 A1, la tenemos definida en 0000-0110 que es la posición 7 del EPROM, donde

queremos que el programa continúe detectando las teclas que van a seguir siendo presionadas.



## CAPITULO VII

VII.- PRUEBAS SOBRE EL PROTOTIPO DEL MICROCOMPUTADOR  
MECS-40 CON SUS RESPECTIVOS PERIFERICOS.

Una de las primeras pruebas que se hizo en el Laboratorio de la ESPOL, fue probar el circuito que programa la memoria de solamente lectura del sistema del Microcomputador MCS-40.

La memoria de solamente lectura del MCS-40 es el Circuito Integrado 4702A llamado EPROM ya se habló de él en el Capítulo IV en Componentes de apoyo de la Familia del MCS-40.

En el circuito Programador de EPROM'S se montó el CI 4702A, en el que se grabó cualquier tipo de información y generalmente en las últimas direcciones de la memoria. La grabación se la hizo en las últimas posiciones de la memoria por precaución, ya que en caso de que se dañara alguna posición del circuito 4702A, en la primera prueba de experimentación se salvarían las primeras posiciones de la memoria, que es donde

va grabado el monitor del MCS-40. En la primera prueba se ocupó las cinco últimas posiciones, es decir, desde la posición 251 hasta la posición 255 del 4702A y se grabó el siguiente patrón de datos.

1110 0000 en cada una de las direcciones 251 (1111 1011); 252 (1111 1100); 253(1111 1101); 254(1111 1110) y 255(1111 1111).

En la figura 5-3, se muestra cuatro bancos de interruptores que son los que dan las direcciones y los datos al CI 4702A.

Cada banco contiene 8 interruptores, debido a que cada posición de memoria del EPROM es de 8 bits y también las direcciones en la memoria son de 8 bits(256 posiciones de memoria).

Dos bancos de interruptores son para los datos o las instrucciones del monitor y los otros dos bancos de interruptores son para las direcciones de la memoria.

Los dos bancos de interruptores de la izquierda del gráfico 5-3 son para los datos o instrucciones y los dos bancos

de interruptores de la derecha son para las direcciones. Contando de izquierda a derecha en el gráfico 5-3 los bancos de interruptores de uno a cuatro, se tiene que el banco #1 es para los "1'S" lógicos de los datos o instrucciones, el banco # 2 es para los "0'S" lógicos de los datos o instrucciones, el banco #3 es para los "1'S" lógicos de las direcciones de la memoria y el banco #4 es para los "0'S" lógicos de las direcciones de la memoria.

Después de que se hizo la grabación del patrón de bits, 1110 0000, se efectuó la prueba del mismo para ver si la grabación fue correcta, entonces se polarizó el CI 4702A con las fuentes de voltaje correspondiente al Modo de Lectura del EPROM. A continuación detallo una pequeña tabla que muestra los voltajes de polarización en los diferentes pines del CI 4702A que son necesarios en los modos de Programación y lectura.

.... /

MODO \ PIN	12 (Vcc)	13 (Programa)	14 $\overline{CS}$	15 (Vbb)	16 (Vgg)	22 (Vcc)	23 (Vcc)
LECTURA	Vcc	Vcc	GND	Vcc	Vgg	Vcc	Vcc
PROGRA- MACION	GND	Pulso de Programa	GND	Vbb	Vgg	GND	GND

Luego de polarizar el Circuito Integrado 4702 en el Modo de lectura se determinó que los "1'S" lógicos que se grabaron, salieron a -5 voltios y los "0'S" lógicos a +5 voltios. Por lo tanto estan de acuerdo con la tecnología MCS con que trabaja el Sistema MCS-40.

El Conjunto de Instrucciones en secuencia o Monitor que se explicó en el capítulo anterior, se grabó en el EPROM 4702A

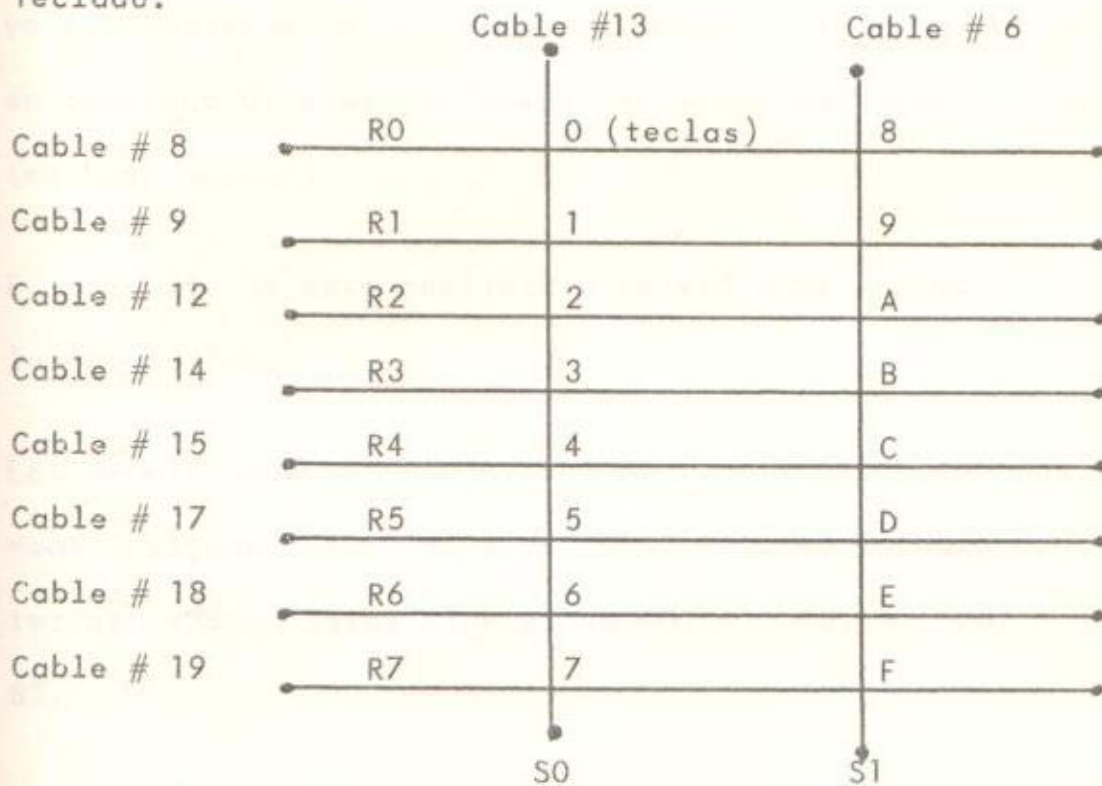
El siguiente periférico que se probó fue la unidad de teclado que va conectada al circuito integrado de interfase 4269

El Microcomputador MCS-40 en este trabajo usa un teclado de 16 teclas, de 0 a F (hexadecimal).

Se tomó un teclado cualquiera y se interconectó las teclas de acuerdo a la matriz que necesita el dispositivo de inter

fase 4269.

A continuación se muestra la matriz del teclado indicando la líneas de Búsqueda "S" y las líneas de Retorno "R" requeridas por el sistema MCS-40 en el Modo de Búsqueda de Teclado.



Las líneas R son las líneas de Retorno.

Las Líneas S son las líneas de Búsqueda.

Los números del 0 al F (hexadécimal) son las teclas.

Los cables especificados con su número respectivo indican la posición en el manejo de cables que van del teclado a la tarjeta impresa del MCS-40.

La figura 7-1 muestra una fotografía del teclado.

El próximo periférico que se probó en el Laboratorio fue el de Video.

En el Periférico de Video se pueden ver hasta 10 caracteres ya que contiene 10 dispositivos de LED'S de 7 segmentos y en cada uno de ellos se puede ver desde la tecla 0 a la F (en hexadecimal).

El circuito de este periférico de video se lo puede ver en la figura 7-2.

Las señales que van a accionar este periférico son las líneas designadas para el video del circuito integrado de interfase 4269. Estas líneas son A0, A1, A2, A3, B0, B1, B2, B3.

Las señales B0, B1, B2, B3, A0, y A1 son los que forman el caracter que va a ser visto. La señal A2 es generada por el circuito integrado 4269 y es la señal de reloj que activará el Registro de Desplazamiento formado por el circuito integrado 4003.

La señal A3 es también generada por el circuito integrado

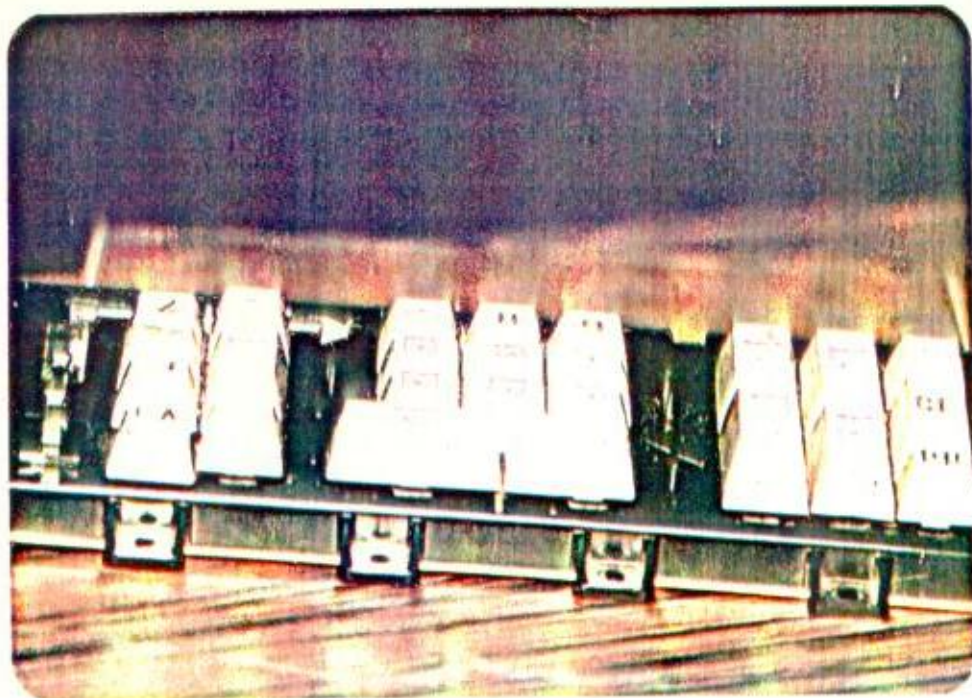


FIGURA 7-1

TECLADO DEL MCS-40

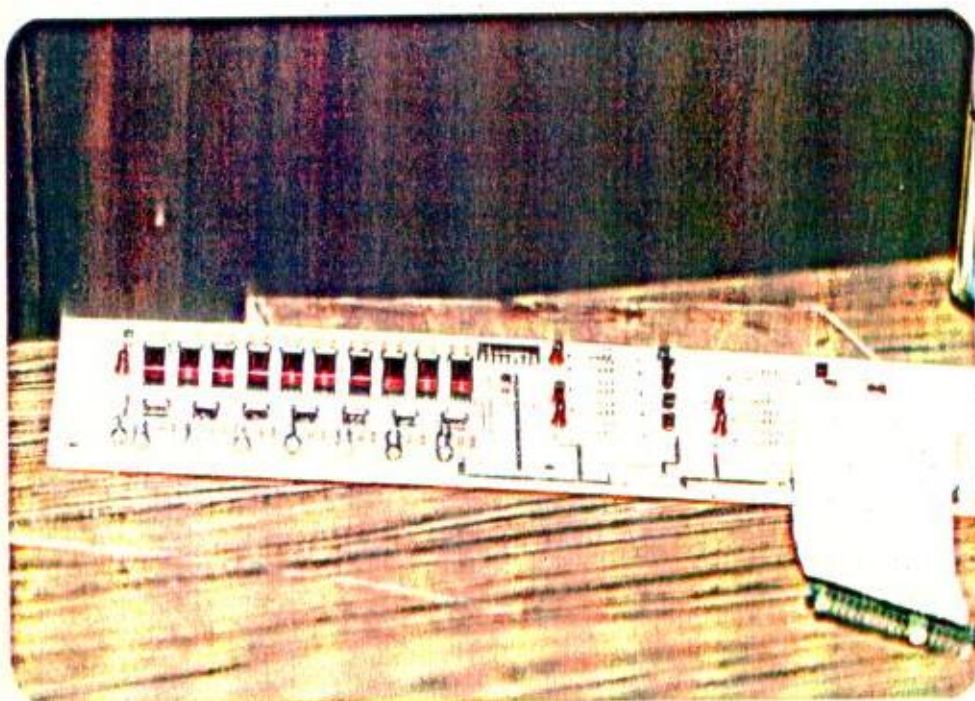


FIGURA 7-4

DISPOSITIVO DE VIDEO

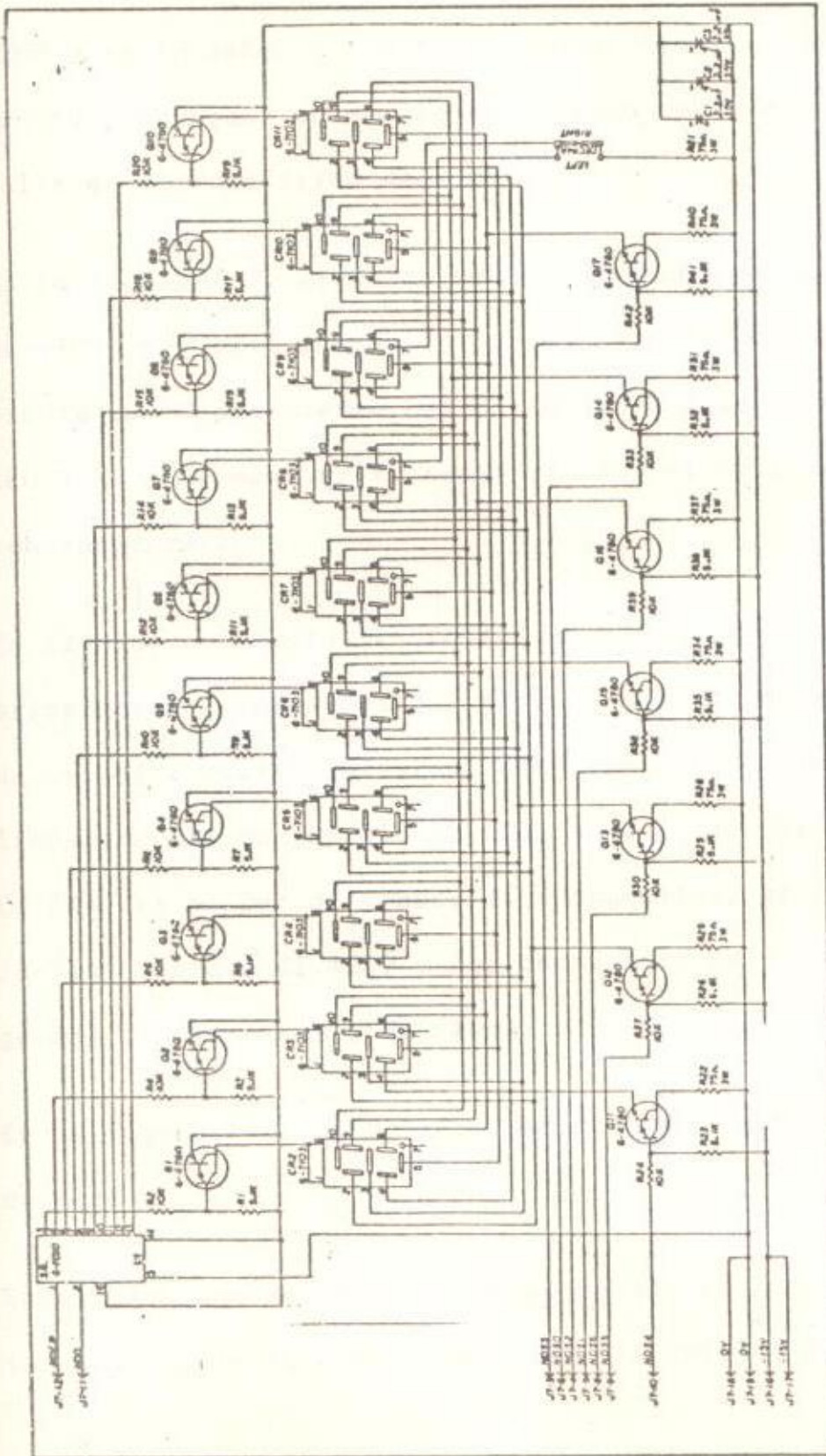


FIG 7.2  
PERIFERICO DE VIDEO



4269 y es la señal que actúa sobre el Registro de Desplazamiento para hacer un avance al próximo carácter que va a salir en el Periférico de Video.

En la figura 7-2, se puede ver el diagrama del periférico de video que está compuesto por el Registro de Desplazamiento de 10 caracteres, el CI 4003; 10 dispositivos de LED'S de 7 segmentos; 17 transistores; 42 resistencias y 3 condensadores.

En el diagrama de la figura 7-3, donde se muestran los principales componentes del microcomputador MCS-40, se puede ver el circuito integrado 7448 que está conectado a las líneas de salida B0, B1, B2, B3, A0 y A1 del CI 4269. Este CI 7448 es el Decodificador de 7 segmentos, es decir, toma las señales del CI 4269 y las decodifica para luego entregarlas al dispositivo de video.

En la figura 7-4, se muestra una fotografía del periférico de video.

En el diagrama del MCS-40 que se mostró en la figura 7-3 están los siguientes componentes P4201A, D4040, 4289, B4702A



P4265, P211A, 4269, y 7448. A continuación haré una explicación del trabajo del Microcomputador MCS-40 con cada uno de sus componentes.

Una vez de que se haya cargado al Monitor en el CI 4702A con el Programador de EPROM'S y se lo haya conectado al MCS-40, el Microcomputador estará listo para empezar a trabajar una vez que se le haya aplicado la energía correspondiente.

Las fuentes de Voltaje que necesita el MCS-40 son +5V, -10V y Tierra.

El circuito integrado P4201A es el circuito de Reloj del MCS-40. A este dispositivo están conectados dos interruptores.

El interruptor SW1 hace una iniciación al MCS-40 con la función Reset. El interruptor SW2 es para ejecutar un programa Paso por Paso, es decir, con la ejecución de cada instrucción, hará una parada y continuará a la ejecución de la próxima instrucción cuando se active nuevamente el interruptor.

El CI D4040 es el Microprocesador central, donde se ejecutaran todas las instrucciones. El CI P4289 es el circuito de

interfase entre el Microprocesador 4040 y la memoria EPROM 4702A.

El CI B4702 es la memoria que solamente puede ser leída cuando esté trabajando con el MCS-40. Esta memoria puede ser escrita o grabada con el Programador de EPROMS y puede ser borrada con una aplicación de Rayos Ultravioletas. En esta memoria se encuentra el Monitor y su capacidad es de 256 bytes.

El CI P4265 es el circuito de interfase entre el microprocesador 4040 y la memoria RAM.

El CI P2111A es la memoria RAM del Microprocesador MCS-40.

Esta memoria es diferente a la memoria de EPROM'S ya que en ella se pueden escribir o grabar con el sistema MCS-40 pero en cambio así falta de energía la información se pierde, ya que es una memoria volátil. En esta memoria se graban los programas y los datos que van a ser ejecutados.

El CI 4269 es el circuito de interfase entre el Microprocesador 4040 y los Periféricos de Video y de Teclado.

Tiene sus propios registros y memoria, además es programado según los periféricos que se utilice. En este trabajo

se lo programó para el periférico de Video en el MODO de DESCARGA DE GAS y para el periférico de teclado en el MODO de BUSQUEDA DE TECLADO.

El Monitor debe estar bien grabado en el 4702A ya que de las instrucciones contenidas en el depende el correcto funcionamiento del MCS-40.

En este caso se grabó el Monitor que ya se explicó en el capítulo II y que comprende un conjunto de instrucciones en secuencia que al ejecutarse ponen activos a los periféricos del MCS-40, como el Teclado y el Dispositivo de Video.

Al iniciarse la ejecución del programa el Microprocesador 4040 pone activo al 4269 que está en espera de que se presione una tecla de las 16 teclas que contiene el Teclado y una vez que ésta haya sido detectada como válida se almacenará en el FIFO del 4269, luego otra instrucción lee el carácter de la tecla almacenada en el FIFO y lo guarda en el Acumulador. Una siguiente instrucción toma los datos del Acumulador y los almacena en los registros de Video 4269, los cuales son sacados por una siguiente instrucción al Periférico de Video. Las señales A2 y A3 colocan al carácter

en el dispositivo de LED'S de 7 segmentos correspondiente.

Una Subrutina de Suma entre el Acumulador y el Registro de Indices determinan el número de veces que las teclas han sido detectadas, (en este caso es 8) y entonces inicializan el FIFO y los Registros de Video, no dejando así que se sobrecarguen el FIFO y los Registros de Video, manteniéndolos en constante espera de la entrada de un nuevo caracter.

La cantidad de dígitos que se pueden entrar en secuencia, son diez, ya que están limitados por el Periférico de Video que tiene capacidad para diez dígitos. Una vez que se hayan sido visto los diez caracteres detectados por el CI 4269, entonces se puede dar un RESET O INICIALIZACION con el interruptor SWL y el MCS-40 estará activo nuevamente para detectar diez caracteres más.

En la figura 7-5 se muestra una fotografía del microcomputador MCS-40.

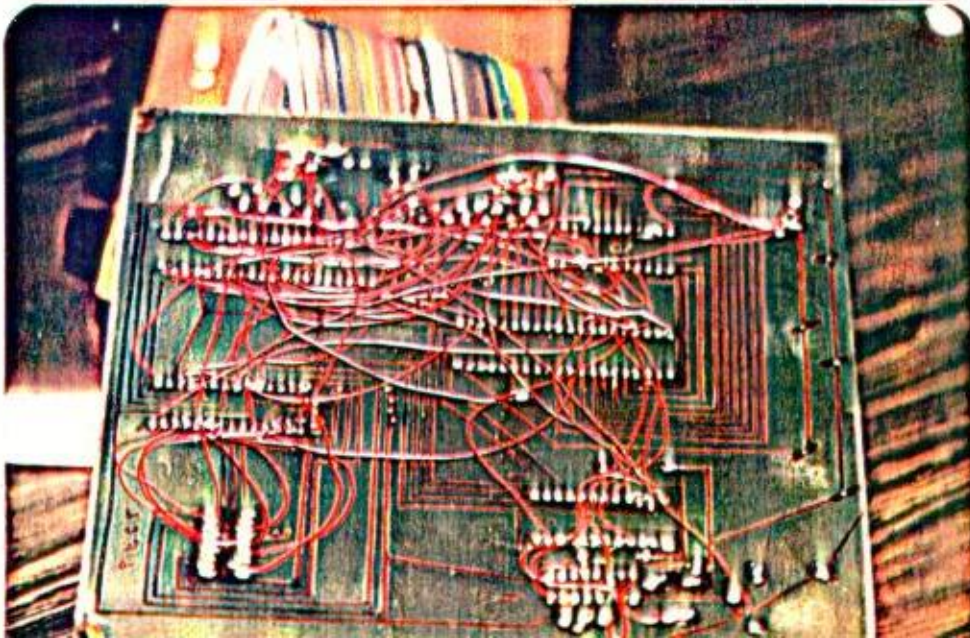
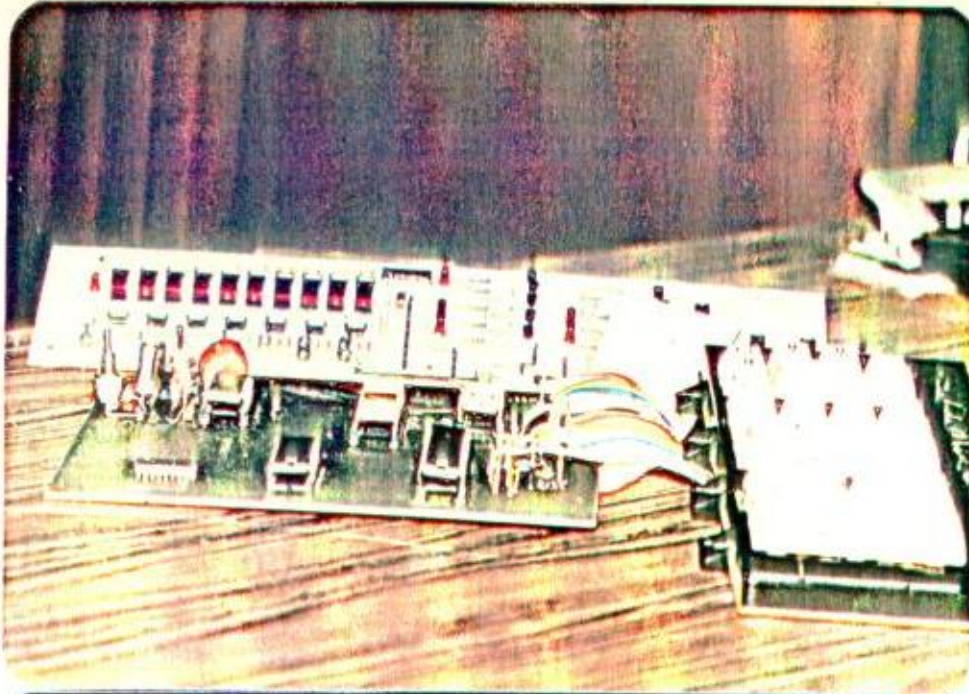


FIGURA 7-5 MICROCOMPUTADOR MCS-40

## CAPITULO VIII

## 8. USO DEL SISTEMA DEL MICROCOMPUTADOR MCS-40 EN EL LABORATORIO.

## 8.1. Técnicas de Programación.

Esta sección incluye algunas técnicas, las cuales pueden ayudar al Programador del Microcomputador MCS-40.

A continuación presento algunas técnicas para programar; Cómo usar más de una Página de Memoria; El uso de Subrutina, y Operaciones Lógicas.

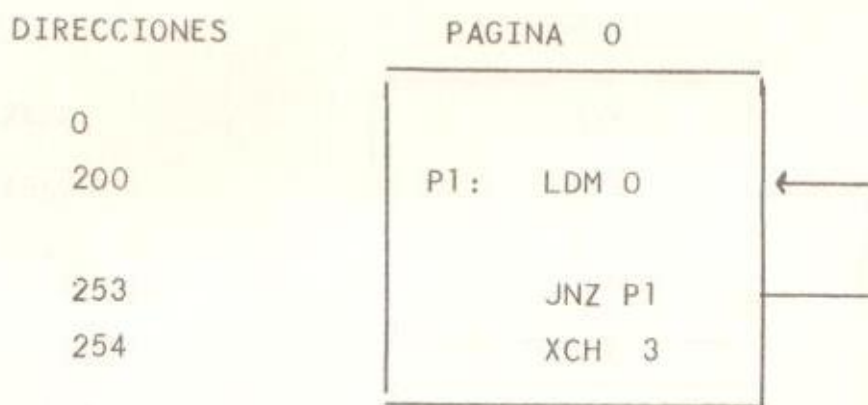
Cómo Programar Usando más de Una Página de Memoria. En los capítulos anteriores he hablado de que los Programas pueden ser mantenidos en ROM, PROM o RAM pues bien, ellos ahora pueden ser divididos en Páginas. Cada Página consiste de 256 localidades de 8 bits. Las direcciones de 0 a 255 están contenidas en la primera página, las direcciones de 256 a 511 están contenidas en la segunda Página y así sucesivamente.



Al programar no se puede pasar nunca de una página a otra, excepto cuando se usen las instrucciones JUN o JMS.

El siguiente ejemplo aclarará lo que se dijo en el párrafo anterior.

Supongamos que un programa en memoria aparece de la siguiente manera:



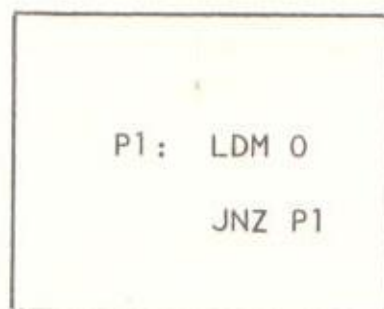
Si en este programa el acumulador no es cero cuando la instrucción JNZ es ejecutada, el Control de Programa será transferido a la localidad 200, que efectivamente fue lo que se intentó.

Pero supongamos ahora que un error se descubrió en el Programa y que se quiere insertar una instrucción entre las localidades 200 y 253, entonces el Programa aparecerá como sigue:

DIRECCION DECIMAL

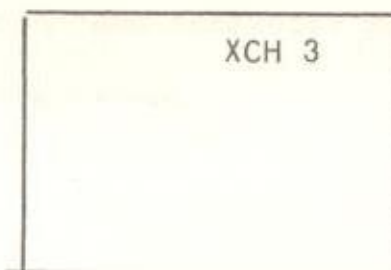
PAGINA 0

0  
200  
254



PAGINA 1

256  
456  
511



Ahora que la instrucción JNZ está ubicada en las dos últimas localidades de la página funciona diferente. Entonces si el acumulador no es cero cuando la instrucción JNZ es ejecutada el control del programa será erróneamente transferido a la localidad 456, causando por lo tanto resultados inválidos.

El control del Programa puede ser transferido correctamente de una página a otra solamente por las instrucciones JUN y JMS debido a que usan 12 bits para direccionar cualquier localidad sobre cualquier página.

## SUBRUTINAS.

Frecuentemente, un grupo de instrucciones debe ser repetido muchas veces en un programa.

El grupo puede ser escrito "n" veces, si éste es necesitado en "n" diferentes puntos en un programa, pero hay una mejor economía usando Subrutinas.

Una subrutina es decodificada como cualquier otro grupo de instrucciones del Lenguaje Assembly.

El Programador referencia el nombre de la subrutina escribiendo en el campo del operando de la instrucción JMS. cuando la instrucción JMS es ejecutada, la dirección de la próxima instrucción secuencial es escrita en un STACK (Memoria de las Direcciones) y la ejecución del Programa procede con la primera instrucción de la Subrutina.

Cuando la subrutina ha terminado su trabajo una instrucción BBL es ejecutada, la misma que carga un valor en el acumulador y lee una dirección de el STACK, produciendo que la ejecución del programa continúe con la instrucción que sigue a la instrucción JMS en el Programa.

Así, una subrutina puede ser llamada desde cualquier punto de la memoria.

Como las direcciones del STACK y la instrucción JMS usan 12 bits, los programas y Subrutinas pueden estar localizados en cualquier parte de la memoria del programa incluso estando en diferente página de memoria.

#### OPERACIONES LOGICAS.

##### La Lógica "Y"

La función "Y" de dos bits, está dada por la siguiente tabla de verdades:

	0	1
0	0	0
1	0	1

Cualquier bit que trabaje por intermedio de la función "Y" con un cero produce un cero y cualquier bit que trabaje por intermedio de la función "Y" con uno, permanece igual.

La siguiente subrutina produce la función "Y", bit por bit, de las dos cantidades mantenidas en los Registros de

Indices 0 y 1. El resultado es colocado en el Registro ce ro y el Registro 1 es establecido a cero. Los registros de indices 2 y 3 también se los usa en este programa.

A continuación presento un desarrollo de la subrutina usando la Operación Lógica "Y".

En el Registro 0 se tiene 1110 y en el Registro 1 se tiene 0011, si estas dos cantidades se las trabaja con una operación lógica "Y" el resultado será 0010, esto es:

	1110	REGISTRO 0
"Y"	<u>0011</u>	REGISTRO 1
	0010	REGISTRO 1

La Subrutina produce la operación "Y" de dos bits colocando los bits en la posición izquierda del acumulador y del Registro 2 respectivamente y encerrando los tres bits de la derecha del acumulador y del Registro 2. El registro 2 es entonces sumado al acumulador y el Registro del Carry es igual a la operación lógica "Y" de los dos bits.

El Programa de la Subrutina lo muestro a continuación, per antes quiero explicar la nomenclatura que utilizo en el Programa.

R0, R1, R2, R3, son los registros 0, 1, 2 y 3 respectivamente.

AC es el Acumulador.

NOMBRES	CODIGOS	OPERANDOS	REGISTROS
"Y"	: FIM	2,11 ;	R2=0 R3=11
L1	: LDM	0 ;	AC=0
	XCH	0 ;	AC= Los datos de R0; R0=0
	RAL		; 1er "Y" bit a Carry
	XCH	0 ;	Datos salvados en R2 AC=0
	ISZ	3,L2 ;	Se termina SI R3=0
	JUN	L3	
L2	: RAR		; El bit de R0 está solo en el AC.
	XCH	2 ;	Salva el 1er "Y" bit en el R2.
	XCH	1 ;	Obtiene el Bit del R1
	RAL	1 ;	Salva los datos desplazados en R1
	RAR		; 2do. "Y" bit a AC.
	ADD	2 ;	La suma da "Y" de los dos bits en Carry
	JUN	L1	
L3	BBL	0 ;	Regresa al programa principal.

Seguidamente presento el programa desarrollado paso por paso

hasta llegar a tener en el registro cero el valor deseado.

NOMBRES	CODIGOS	OPERANDOS	REGISTROS
"Y" :	FIM	2,11	; R2=0000, R3= 1011
L1 :	LDM	0	: AC=0
	XCH	0	; AC=1110 R0=0
	RAL		; RC=110C0 CY=1
			(C0 es el carry anterior y no se conoce su valor)
	XCH	0	; AC=0 R0=110C0
	1SZ	3,L2	; A3=12
L2 :	RAR		; AC=1000 CY=0
	XCH	2	; AC= 0 R2=1000
	XCH	1	; AC=0011 R1=0000
	RAL		; AC=0110 CY=0
	XCH	1	; AC=0 R1=0110
	RAR		; AC=0000 CY=0
	ADD	2	; AC=1000 CY=0
	JUN	L1	
L1 :	LDM	0	; AC=0
	XCH	0	; R0=0 AC=110C0
	RAL		; AC= 10C00 CY=1
	XCH	0	; AC=0 R0= 10C00
	1SZ	3,L2	
L2 :	RAR		; AC=1000 CY=0
	XCH	2	; AC=1000 R2=1000
	XCH	1	; AC=0110 R1=1000
	RAL		; AC=1100 CY=0
	XCH	1	; AC=1000 R1=1100
	RAR		; AC=0100 CY=0

NOMBRES	ADD CODIGOS	2 OPERANDOS	;	AC= 1100	CY=0	REGISTROS
L1 :	LDM	0	;	AC= 0		
	XCH	0	;	AC= 10Co0	R0=0	
	RAL		;	AC= 0Co00	CY=1	
	XCH	0	;	AC= 0	R0=0Co00	
	1SZ	3,L2	;	R3=14		
L2 :	RAR		;	AC=1000	CY=0	
	XCH	2	;	R2=1000	AC=1000	
	XCH	1	;	R1=1000	AC=1100	
	RAL		;	AC=1000	CY=1	
	XCH	1	;	AC=1000	R1=1000	
	RAR		;	AL=1100	CY=0	
	ADD	2	;	AC=0100	CY=1	
	JUN	L2				
L1 :	LDM	0	;	AC=0		
	XCH	0	;	R0=0	AC=0Co00	
	RAL		;	AC=C0001	CY=0	
	XCH	0	;	AC=0	R0=C0001	
	1SZ	3,L2	;	R3=15		
L2 :	RAR		;	AC=0	CY=0	
	XCH	2	;	R2=0	AC=1000	
	XCH	1		AC=1000	R1=1000	
	RAL		;	AC=0000	CY=1	
	XCH	1	;	R1=0000	AC=1000	
	RAR		;	AC=1100	CY=0	
	ADD	2	;	AC=1100	CY=0	
	JUN	L1				



NOMBRES	CODIGOS	OPERANDOS	REGISTROS
L1	:	LDM	0 ; AC= 0000
		XCH	0 ; AC= Co001 R0=0000
		RAL	; AC= 0010 CY=C0
		XCH	0 ; R0= 0010 AC=0000
		1SZ	3,L2 ; R3= 0000
		JUN	L3
L3		BBL	0 ; Regresa al Programa Principal.

Por lo tanto en el Registro 0 se obtuvo el registro esperado 0010 de la Operación Lógica "Y".

Un procedimiento parecido se utiliza en las operaciones lógicas "0" y "X0".

## 8.2 TECNICAS PROGRAMANDO DEL 4040.

Un programa consiste en escribir instrucciones en secuencia para un computador.

Se puede decir que un programador está listo para hacer un programa cuando conozca la acción de cada una de las instrucciones máquina. ( El conjunto de instrucciones del Microcomputador MCS-40 se describieron en el capítulo II).

Cada una de las instrucciones máquina manipulan los datos en alguna forma. Los datos pueden ser los contenidos de el Contador de Programa, el cual indica donde la próxima instrucción va a ser encontrada, los contenidos de uno de los Registros del CPU, el Acumulador, el Flip-Flop de Carry, los contenidos de los RAM'S, o ROM'S, o las señales en una Puerta.

La programación es más fácil cuando se la aprende usando ejemplos, es por esto que a continuación presento algunos ejemplos que ilustran tecnicas como el uso de las Puertas de Entrada-Salida, lazos básicos de programa y el uso de subrutinas. Recomiendo que cuando se

revise estos ejemplos se refieran frecuentemente a las definiciones de las instrucciones.

### 8.3 EJEMPLOS.

#### Ejemplo No. 1

El CPU 4040 tiene un pin de prueba y se puede chequear el estado de este pin cuando está conectado a un interruptor, es decir, se prueba si tiene un estado lógico "1" o un estado lógico "0". Una instrucción con condición de Salto (JCN) puede usarse para desarrollar esta prueba. La dirección a la que va a hacer el salto debe estar en la misma página.

La instrucción de dos palabras JCN PRUEBA y 16 están almacenados en las localidades 2 y 3 de la memoria ROM

		<u>OPR</u>	<u>OPA</u>
Localidad	No. 2	0001	0001
		(JCN)	(Salte si la señal de prueba es igual a la lógica "0".
Localidad	No. 3	0001	0000
			Salte a la localidad de la memoria ROM No. 16.)

Si el interruptor está conectado al lógico "0" (vss) al Pin de brueba del CPU, cuando esta instrucción es ejecutada, el Contador de Programa en el CPU, saltará a la localidad 16 (Esto es, la próxima instrucción que se ejecutará será la que esta en la localidad 16 de la memoria de ROM).

Si el interruptor está conectado al lógico "1" (Vdd), el Contador de Programa no saltará, pero en cambio será incrementado por 1 y la instrucción que está en la localidad No. 4 de la memoria de ROM será la próxima a ejecutarse. Asi el estado del interruptor puede ser probado con una instrucción.

#### Ejemplo No. 2.

En este ejemplo genero 10 pulsos que sirven para manejar la línea de reloj del registro de Desplazamiento 4003. Utilizo la puerta de salida de un RAM para poder sacar la señal de reloj.

A continuación detallo el Programa:

No. de la Instrucción	Nombre de dirección	Nemónico	Opa
(1)		LDM	8
(2)		XCH	15
(3)		LDM	6
(4)		XCH	14
(5)		FIM	0
		12,	0
(6)		SRC	0
(7)		LDM	0
(8)		ADD	15
(9)		WMP	
(10)		ADD	15
(11)		WNP	
(12)		1SZ	14, LAZO

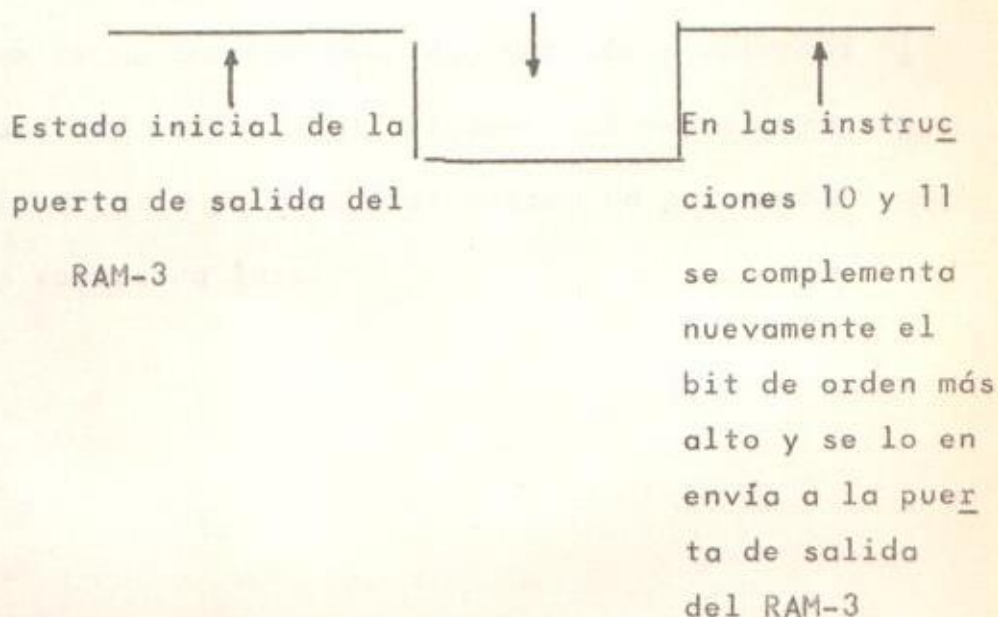
Una explicación del programa paso por paso hago a continuación.

- a. En las instrucciones No. 1 y No. 2, se carga el número 8 (1000) dentro del Registro de Indices número 15 (1111).

- b. Las instrucciones No.3 y No. 4 cargan el número 6(0110) dentro del Registro de Indices número 14 (1110).
- c. La instrucción No. 5 entrega la dirección de el RAM deseado y lo almacena en el Registro par de Indices cero.
- d. La instrucción No. 6 envía la dirección almacenada al banco de RAM'S y selecciona la puerta de salida del RAM deseada.
- e. La instrucción No. 7 inicializa el acumulador a cero.
- f. La instrucción No. 8 suma los contenidos del Registro 15 al acumulador.
- g. La instrucción No. 9 escribe el contenido del acumulador dentro de la Puerta de Salida del RAM-3.
- h. La instrucción No. 10 suma el contenido del registro 15 al acumulador.
- i. La instrucción No. 11 escribe el contenido del acumulador dentro de la puerta de salida del RAM-3.

Las instrucciones 8,9,10 y 11 generan un pulso de reloj de la siguiente manera:

En las instrucciones 8 y 9 se complementa el bit de orden más alto y se envía a la puerta de salida de RAM-3.



j. En la instrucción No. 12, el contenido del registro 14 es incrementado por 1(0001).

El número 7 (1110) está ahora almacenado en el Registro 14. Como éste resultado no es igual a cero, el control del programa salta a la dirección especificada en la segunda palabra de esta instrucción.

En este caso la dirección almacenada en la segunda palabra

es la dirección de la instrucción No. 8. Entonces el programa ejecuta las próximas 4 instrucciones en secuencia y genera el segundo pulso de reloj.

Esta secuencia es repetida 10 veces, generando así los 10 pulsos de reloj: Después de la décima vez el contenido del Registro 14 es incrementado una vez más y entonces alcanza el valor de cero (0000) obligando al control de programa continuar a la próxima instrucción en secuencia y salir de la subrutina lazo.



## CONCLUSION

El sistema MCS-40 es un microcomputador diseñado específicamente para soportar el desarrollo y la implementación de la Unidad Central de Procesamiento "CPU 4040".

EL MCS-40 consiste de 4 módulos:

- a. Unidad de Procesamiento Central.
- b. Memoria RAM.
- c. Memoria de Control (EPROM).
- d. Programador del EPROM.

Además de estos módulos, el sistema MCS-40 contiene la fuente de Poder, los Conectores de Entrada-Salida, el Teclado y el dispositivo de Video.

El corazón del sistema es el módulo de Procesamiento Central 4040 y está formado por un solo chip de 4 bits de transferencia de datos.

El MCS-40 es un sistema completo, conteniendo el sistema de reloj, 2K bits de memoria PROM y 1K bits de memoria RAM, compuesta por la memoria estática de acceso aleatorio 2111.

El MCS-40 contiene el circuito requerido para el interfase entre el Procesador Central y el módulo de la memoria de datos RAM.

El programador de EPROM'S se lo puede utilizar para programar otros PROM'S diferentes al P4702A, que es el utilizado en este trabajo, como el 1702A, 2702A y 8702A.

El microprocesador 8080 que es de mayor capacidad que el 4040 también puede utilizar este tipo de PROM'S.

Recomiendo que para entender el funcionamiento del Monitor o de cualquier programa aplicable al microcomputador MCS-40 se lea detenidamente las instrucciones listadas en las tablas establecidas en esta Tesis. Así también, si se desea modificar o incrementar el monitor o desarrollar algún programa cualquiera, se lea los ejemplos y las técnicas de programación.

El Microcomputador MCS-40, se diseñó con una configuración básica, es decir, con un teclado de 16 caracteres, con un dispositivo de video de 10 caracteres, con la memoria de datos RAM de 1K bits y la memoria de programa PROM con 2K bits. Entonces su capacidad puede ser incrementa-

da a un teclado de 128 caracteres; A un dispositivo de video de 16 caracteres; A una memoria de datos RAM de 16 bancos de 1K x 4 bits en que cada banco contiene cuatro circuitos integrados RAM 2111 ( aunque su máxima capacidad puede ser de 48K x 4 bits); la memoria de programa PROM puede ser incrementada hasta 4K bytes.

En este Microcomputador se puede desarrollar un sistema de Monitor que provea la capacidad de poder leer y modificar los contenidos de memoria, leer y perforar cintas, asignar dinámicamente sistemas periféricos, programar y verificar PROM'S y desarrollar otras funciones que reduzcan significativamente el tiempo de procesamiento y trabajo.

## BIBLIOGRAFIA

- 1.- WILLIAM BARDEN JR.: How To Program Microcomputer.
- 2.- DAVID G. LARSEN: The Bugbook V
- 3.- DAVID G. LARSEN: The Bugbook VI
- 4.- SILVANIA ECG.: Technical Manual Linear Modules and Integrated Circuits.
- 5.- INTEL:4040: User's Manual.
- 6.- NATIONAL SEMICONDUCTOR: TTL Databook.
- 7.- GENERAL ELECTRIC: Ge Replacement Semiconductor Guide.
- 8.- POPULAR ELECTRONICS.: Revista de Febrero de 1978
- 9.- POPULAR ELECTRONICS.: Revista de Marzo de 1978