Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

ESTACION PARA DESARROLLO DE HARDWARE

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A :

WENCESLAO GRANDE OLGUIN

BAJO LA DIRECCION DEL

M. en C. JUAN ANTONIO NAVARRO MARTINEZ

MEXICO, D, F. ABRIL DE 1985

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

ESTACION PARA DESARROLLO DE HARDWARE
(E D H)

Abril de 1985.

Wenceslao Grande Olguin.

ESTACION PARA DESARROLLO DE HARDWARE
(E D H)

I. Importancia de Disenar y Construir una EDH.

I.1. Introduccion. El Sistema EDH (Estacion para Desarrollo de Hardware) tiene el proposito de ser una herramienta en el diseno y prueba de Modulos para Sistemas de Computo basados en el Microprocesador Z80. El Sistema EDH es un computador con una arquitectura en base a un Sistema de Bus, lo que permite que en cualquiera de sus Conectores de Bus pueda ser conectado cualquier Modulo. La compatibilidad con tarjetas comerciales, permite un rapido y economico desarrollo del Modulo.

El Sistema de Bus permite la conexion de Modulos Bajo Prueba, lo cual es el objetivo fundamental del Sistema EDH, sin embargo, el Sistema EDH puede funcionar completamente solo con sus Modulos Basicos. El Sistema de Bus esta integrado fundamentalmente de las senales del Microprocesador Z80, en un Bus EDH y en un Bus Emulador, ademas de posiciones libres en un Bus Libre.

La eleccion del Microprocesador Z80 es consecuencia de ser, a la fecha, el Microprocesador mas usado en el mercado. El Z80 tiene un conjunto de 158 instrucciones que permiten cubrir practicamente todas las necesidades actuales en cuanto a instrucciones, ademas de ser compatible con los Microprocesadores 8080 y 8085, dos Microprocesadores muy comunes en el mercado.

Los Modulos Basicos del Sistema EDH son:

a) Sistema de Bus.
b) Procesador.
c) Memoria.
d) Teclado/Unidad Visual.

El Modulo del Procesador contiene un Z80 y buffers de entrada y de salida para cumplir los requerimientos electricos del Bus.

El Modulo de Memoria contiene memorias de solo lectura o de lectura /escritura en grupos de 16 Kbytes. Este Modulo contiene conmutadores para programar varias opciones como su posicion dentro del rango de direccionamiento y su habilitacion. Es posible tener Modulos con memorias de solo lectura o de lectura/escritura o una combinacion de ellas, en Sub-Modulos de 2 Kbytes.

El Modulo de Teclado/Unidad Visual permite la comunicacion primaria hombre-maquina. Esta comunicacion se realiza a traves de un teclado y varias unidades visuales de siete segmentos.

*

I.2. Usos para el Sistema EDH. El Sistema EDH tiene el objetivo de cubrir las necesidades fundamentales de los siguientes usuarios:

a) El Estudiante.
b) El Profesor.
c) El Ingeniero.

El Estudiante de Electronica que desea experimentar con Sistemas de Computo generalmente tiene conocimientos sobre Sistemas Digitales - Combinatorios y Secuenciales, conoce Lenguajes de Programacion de - Alto Nivel, Arquitectura de Computadoras y ha manejado Microcomputadoras de baja capacidad. Este estudiante desea tener facilidades para poder experimentar disenos propios mas elaborados, facilidades que le permitan verificar las funciones de su modulo y evaluar su - comportamiento en un medio y en un tiempo real.
La Arquitectura del Sistema EDH le permite al estudiante satisfacer sus necesidades de Mecamatica (Hardware) y sus necesidades de Programatica de Soporte (Software de Soporte). En la parte de aplicaciones se muestra un ejemplo con el Modulo del Emulador. El Sistema EDH se ha desarrollado pensando en las necesidades futuras, como la necesidad de Interfaces IEEE-488 para comunicacion con diferentes Sistemas de Computo, Interfaces RS-232-C para la comunicacion - con Equipos Perifericos e Interfaces Analogico/Digital Digital/Analogico para el procesamiento de senales analogicas.

El Profesor de Electronica generalmente requiere de un medio economico y eficiente de preparar Practicas de Laboratorio a fin de que sus alumnos las desarrollen durante su Curso de Electronica. Debido al mayor grado de dificultad de las Practicas de este tipo, tambien requiere facilidades para desarrollar sus Practicas de manera confiable.
El Sistema EDH esta disenado para presentar una alternativa economica para el desarrollo de Practicas de Laboratorio, debido al tipo de tarjetas que usa, las cuales son facilmente adquiribles en el mercado nacional. Las tarjetas pueden adquirirse para conectar por soldadura o por alambrado (wire wrap) lo que permite disminuir el - tiempo de desarrollo y prueba de los modulos.

El Ingeniero en Electronica requiere de una herramienta confiable - para evaluar si sus disenos pueden resolver sus problemas en la industria.
El diseno del Sistema EDH le brinda al Ingeniero en Electronica toda la versatilidad que ofrece el procesador Z80 a traves de su Sistema de Bus. La potencia de los comandos integrados en el Monitor - del Sistema permiten de una manera sencilla y rapida probar los Modulos que disene el Ingeniero. Es factible probar Modulos de Memoria ROM, Modulos de Memoria RWM, Modulos de Interface a Modems, Modulos de Interface a Impresores, Modulos de Interface a Terminales y en general cualquier Modulo de Interface con el procesador Z80. -

*

I.3. Resumen. El Sistema EDH es un Sistema desarrollado para facilitar - el desarrollo y la prueba de modulos para Sistemas de Computo; es - util al Estudiante, al Profesor y al Ingeniero en Electronica a --- quienes ofrece una alternativa economica, versatil y confiable para sus necesidades.

El marcado incremento en el uso de Sistemas de Computo y los dife-- rentes requerimientos en la Industria Nacional, hacen del Sistema - EDH, una herramienta util y necesaria.

II. Definicion General de la EDH.

II.1. Generalidades. El Sistema EDH esta construido modularmente tomando como base un Sistema de Bus. Este Sistema de Bus esta definido por medio de una tarjeta madre sobre la cual existen un numero de co-- nectores donde van colocados los modulos.

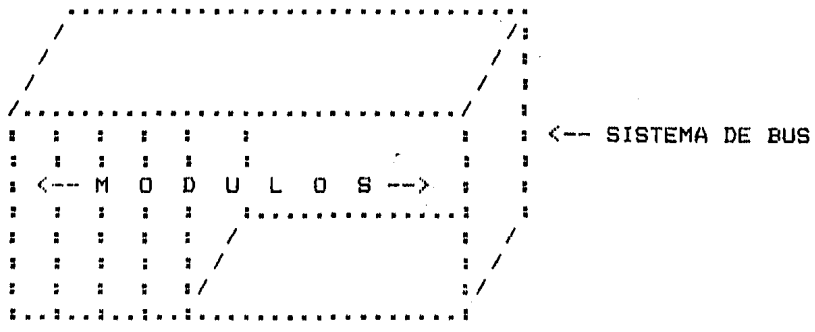


Fig. 1. Sistema EDH.

El Sistema EDH soporta conectores de 100 contactos, 50 por cada - lado, para tarjetas de 5.3"x10"x.062" (ancho, alto y grosor) con - .125" de centro a centro de contactos. Estas tarjetas tienen las - mismas dimensiones que las que se usan en el Bus S-100.
El Modulo de Teclado/Unidad Visual tiene un cable que le permite - conectar al Teclado y a las Unidades Visuales.
El Modulo del Procesador tiene un cable con 40 terminales. Cada terminal equivale a una terminal del Microprocesador Z80. Este as pecto del Sistema EDH es de especial importancia, ya que facilita probar Modulos sin alterar sus conexiones de diseno.

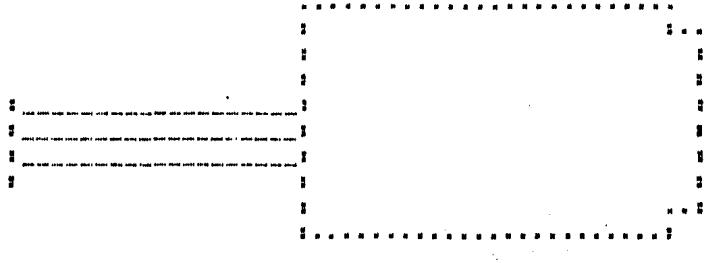


Fig. 2. Modulo con Cable.

II.2. Operacion del Sistema EDH. Para operar el Sistema EDH se requiere inicialmente del Modulo de Teclado/Unidad Visual. Una vez inicializado el Sistema, es posible la comunicacion hombre-maquina a través de un Modulo disenado con ese proposito y la programatica necesaria.

La parte de entrada es un teclado hexadecimal con teclas adicionales que permiten conocer y alterar el contenido de la memoria y de los registros; permiten tambien emitir y recibir informacion de los puertos; y otras que permiten manejar la ejecucion de los programas. Existen ademas ciertas teclas de funcion que son programables.

La parte de salida esta integrada por un conjunto de seis unidades visuales de siete segmentos; en las cuatro unidades de la izquierda, generalmente se muestra la direccion de memoria y, en las dos unidades de la derecha, el dato.

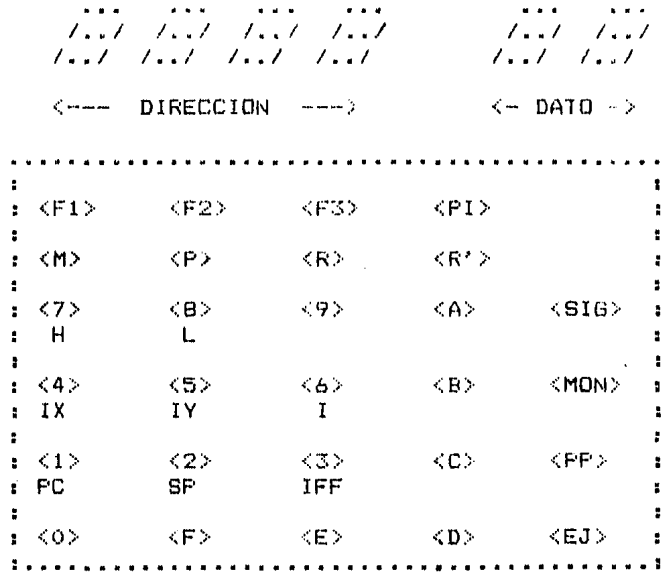


Fig. 3. Unidad de Teclado/Unidad Visual.

La Unidad de Teclado/Unidad Visual esta conectada al Modulo de Teclado/Unidad Visual por medio de un cable. Observese que algunas teclas tienen doble uso.

El significado de las teclas es el siguiente:

F1, F2, F3:	Teclas de Funcion Programables.
PI:	Punto de Interrupcion en el Programa.
M:	Muestra y permite cambio de datos en Memoria.
R:	Muestra y permite cambio de datos en Registros.
R':	Similar a R pero con Registros Alternos.
7 o H:	Dato 7H o Registro H.
8 o L:	Dato 8H o Registro L.
9:	Dato 9H.
A:	Dato AH o Registro A.
SIG:	Siguiente Direccion de Memoria o Puerto.
4 o IX:	Dato 4H o Registro IX.
5 o IY:	Dato 5H o Registro IY.
6 o I:	Dato 6H o Registro I.
B:	Dato BH.
MON:	Regresa al Monitor guardando datos de Registros.
1 o PC:	Dato 1H o Registro PC.
2 o SP:	Dato 2H o Registro SP.
3 o IFF:	Dato 3H o Flip Flop de Interrupcion IFF.
C:	Dato CH.
PP:	Tecla para ejecutar programas Paso a Paso.
0:	Dato 0H.
F:	Dato FH.
E:	Dato EH.
D:	Dato DH.
EJ:	Tecla para ordenar la Ejecucion de Programas.

II.3. Monitor del Sistema EDH. Un programa que maneja en forma versatil todos los recursos de un Sistema de Computo y que ofrece varias facilidades adicionales al usuario, se conoce como Sistema Operativo. El Sistema EDH no esta provisto de un Sistema Operativo. El Sistema EDH contiene un Monitor, el cual es un programa menos complejo, pero que siendo un programa que ocupa poca memoria, maneja los recursos del Sistema EDH en una forma satisfactoria para sus requerimientos. El Monitor del Sistema EDH se encuentra almacenado en una memoria EPROM (Electrically Programmable Read Only Memory), de 2048 bytes, en el Modulo de Memoria.
El Monitor usa un teclado hexadecimal como entrada y un conjunto de seis unidades visuales de siete segmentos como salida.
El Monitor del Sistema EDH soporta 13 comandos, 3 de los cuales son programables. Los comandos que soporta son: ARR, MON, M, P, SIG, R, R', PI, PP, EJ, F1, F2 y F3.
Los comandos del Monitor estan asignados a teclas especificas. Las teclas se encuentran en la Unidad de Teclado/Unidad Visual, exceptuando a la tecla <ARR> que se encuentra en el Modulo del Procesador. Para definir la manera de utilizar los comandos del Sistema EDH, se hara uso de la siguiente simbologia:

MON	Tecla MON.
--->	Flujo de Operacion.
dddd	Direccion Hexadecimal de la Memoria.
aa	Dato Hexadecimal actual.
nn	Dato Hexadecimal nuevo.
pp	Direccion Hexadecimal del Puerto.
r8	Alguno de los registros A, B, C, D, E, F, H, L, I, IFF.
r8'	Alguno de los registros A', B', C', D', E', F', H', L'.
r16	Alguno de los registros PC, IX, IY.
rSP	Registro SP.
rPC	Registro PC.

a) Comando ARR. Forza al Z80 a reiniciar y empezar la ejecucion del programa en 0000 H. Inicializa al registro SP en 23C0 H y a los datos variables que usa el Monitor. Envia a la Unidad de E/S el indicador "-" y espera a que se oprima alguna tecla. Las interrupciones se deshabilitan y se selecciona el Modo 0 de Interrupcion.
El comando ARR es el ultimo recurso para recuperar el control del Sistema EDH, y solo se recomienda usarlo en ultima instancia debido a las inicializaciones que realiza.

--->ARR--->

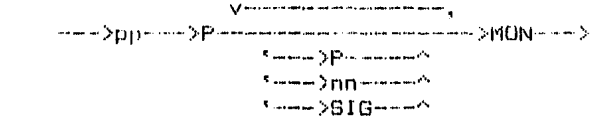
b) Comando MON. Forza al Z80 a regresar al Monitor esperando se oprima alguna tecla. A diferencia de ARR, MON conserva el contenido de los Registros. La tecla MON se puede usar para cancelar un comando o un dato; se puede usar tambien para producir una interrupcion no mascarable al Z80 cuando el procesador se encuentre ejecutando un programa del usuario y se requiera volver a tener el control del Sistema. Si el comando MON tuvo exito, se mostrara el indicador "-" en la Unidad de E/S.

--->MON--->

c) Comando M. El comando M muestra y permite el cambio del contenido de la memoria en la direccion dddd H. Los digitos dddd H asi como el dato actual aa H seran mostrados en la Unidad de E/S. Si se oprime M sin antes oprimir los 4 digitos nnnn H de la direccion, no se mostrara ningun dato y el monitor esperara a que se opriman. Si se oprime la tecla SIG la direccion de memoria se incrementara en 1 y se mostrara el dato correspondiente. Al mostrarse los 6 digitos (direccion y datos) se puede alterar el contenido de la memoria tecleando 2 digitos nn H que

seran el nuevo contenido. Si el cambio tuvo exito se mostraran los 2 nuevos digitos en la Unidad de E/S. Si se oprime repetidamente M, el Monitor volvera a leer y a mostrar el contenido de la direccion que se este mostrando en la Unidad de E/S. Para terminar con este comando se debe teclear MON.

d) Comando P. El comando P muestra y permite el cambio del contenido del puerto cuya direccion es pp H. Los digitos pp H asi como el dato actual aa H del puerto, seran mostrados en la Unidad de E/S. Si se oprime P antes de oprimir los 2 digitos pp H, no se mostrara ningun dato y el Monitor esperara a que se opriman. Si ahora se oprime SIG, la direccion del puerto se incrementara en 1 y se mostrara el dato correspondiente. Al mostrarse los 4 digitos (direccion y datos), se puede alterar el contenido del puerto tecleando 2 digitos nn H como nuevo contenido. Si el cambio tuvo exito se mostraran los 2 nuevos digitos en la Unidad de E/S. Si se oprime nuevamente P, se mostrara el contenido actualizado del puerto.



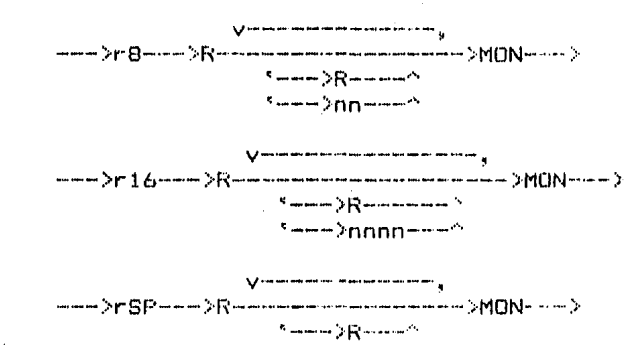
e) Comando SIG. El comando SIG incrementa la direccion de memoria o del puerto en 1, y muestra el contenido correspondiente. Se usa con los comandos M y P.

Veanse los diagramas de sintaxis de M y P.

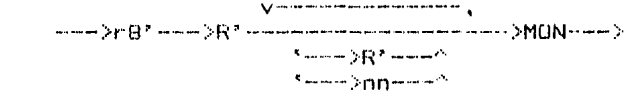
f) Comando R. El comando R muestra y permite el cambio del contenido de los registros r8 y r16, ademas muestra el contenido del registro rSP. Despues de oprimir la tecla correspondiente al registro deseado se oprime la tecla R y se mostrara el contenido del registro seleccionado aa H en el caso de r8 y aaaa H en en el caso de r16 y rSP. Para cambiar el contenido de un registro r8, se teclea nn H, si se desea cambiar el contenido de un registro r16, se teclea nnnn H. Al registro rSP, el Monitor no

permite que se le cambie su contenido. Si el cambio del contenido de los registros r8 y r16 tuvo exito, se mostrara el nuevo contenido en la Unidad de E/S. Para terminar con este comando se debe teclear MON.

Nota: El registro IFF tiene dos valores significativos, 00 H significa que las interrupciones estan deshabilitadas y 04 H significa que estan habilitadas.



g) Comando R'. El Comando R' muestra y permite el cambio del contenido de los registros r8'. Despues de oprimir la tecla correspondiente al registro deseado se oprime la tecla R' y se mostrara el contenido del registro seleccionado aa H. Para cambiar el contenido del registro r8', se teclea nn H. Si el cambio tuvo exito, se mostrara el nuevo contenido en la Unidad de E/S. Para terminar con este comando se debe teclear MON.

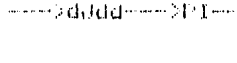


h) Comando PI. El comando PI permite introducir hasta 5 Puntos de Interrupcion en el programa del usuario. Para realizar estas interrupciones el Monitor intercambia la Parte Operativa de las instrucciones con la instruccion RST8 (CF H) y guarda tales Partes Operativas en una Tabla de Puntos de Interrupcion y Partes Operativas BPTAB. Cuando una interrupcion se introduce con el comando PI, este se coloca en la tabla y la instruccion RST8 se coloca en el programa del usuario en la direccion correspondiente. Cuando el procesador encuentra la instruccion RST8 al ejecutar el programa del usuario, realiza una interrupcion,

guardando el contenido de los registros y colocando todas las Partes Operativas que se guardaron en la tabla BPTAB en el programa del usuario, de tal manera que la colocacion de instrucciones RST8 es transparente al usuario.
Para introducir Puntos de Interrupcion se oprimen los digitos de la direccion de memoria dddd H y posteriormente la tecla PI. La Unidad de E/S mostrara la direccion dddd H despues de que se haya oprimido la tecla PI, si la operacion tuvo exito. Para introducir otro Punto de Interrupcion se debe oprimir primero la tecla MON.
Para proseguir la ejecucion de un programa que se ha detenido en un Punto de Interrupcion, se debe usar el comando EJ.
Los Puntos de Interrupcion se pueden cancelar en la siguiente forma:

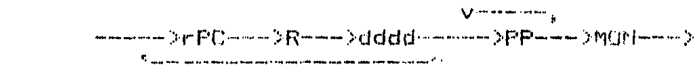
1: Oprimiendo la tecla PI antes de oprimir los 4 digitos dddd H cancelara todos los Puntos de Interrupcion.
2: El uso del comando PP cancelara todos los Puntos de Interrupcion.
3: El uso del comando ARR cancelara todos los Puntos de Interrupcion.

Los Puntos de Interrupcion se pueden verificar analizando la tabla BPTAB (23E4 H). La cantidad de Puntos de Interrupcion activos se encuentra en BFLG (23F4 H).

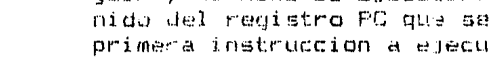


i) Comando PP. El comando PP permite la ejecucion de un programa Paso a Paso es decir, una instruccion a la vez. Despues de ejecutar una instruccion, la ejecucion del programa se detiene, permitiendo analizar a la memoria, a los registros, a los puertos o permitiendo ejecutar algun comando. Debido a que no se modifica el programa, el comando PP se puede usar en programas que esten almacenados en RWM, o en algun tipo de ROM. Un canal del Z80-CTC (Counter Timer Circuit) produce una interrupcion no mascarable (NMI) al Z80 al principio de cada instruccion, obligando al procesador, por lo tanto, a detenerse en cada instruccion. El contenido de los registros se guarda despues de que la instruccion se ha ejecutado.
La instruccion que se ejecutara al oprimir la tecla PP, sera la que corresponda al contenido que se tenga guardado en el Mapa de Registros para el registro PC, el cual puede ser examinado y modificado usando el comando R, o la instruccion que corresponda a la direccion dddd H tecleada.

Cuando se esta ejecutando el comando PP, la direccion de la siguiente instruccion a ser ejecutada y el contenido del registro A (Acumulador), se mostraran en la parte visual de la Unidad de E/S. Oprimiendo repetidamente la tecla PP, es posible ir Paso a Paso a traves de un programa examinando lo que sucede despues de la ejecucion de cada instruccion.
El uso del comando PP, cancelara cualquier Punto de Interrupcion.



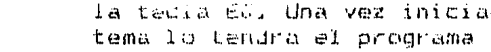
j) Comando EJ. El comando EJ ordena al Z80 empezar la ejecucion de un programa. Se puede proceder de dos Modos: el Modo de Proseguir y el Modo de Ejecutar. El Modo de Proseguir usa el contenido del registro PC que se ha guardado como la direccion de la primera instruccion a ejecutar. Para usar este Modo solo se debe oprimir la tecla EJ. El Modo de Ejecucion requiere que se indique la direccion de la primera instruccion a ejecutar. Para usar este Modo se introducen 4 digitos dddd H y luego se oprime la tecla EJ. Una vez iniciada la ejecucion, el control del sistema lo tendra el programa del usuario, hasta que se encuentre un Punto de Interrupcion, devuelve el control al Monitor o se oprima la tecla MON (o excepcionalmente la tecla ABR).



k) Comandos F1, F2, F3. Los comandos programables F1, F2 y F3 permiten iniciar la ejecucion de un programa que se encuentre a partir de las siguientes direcciones:

Comando	Direccion de la Memoria
F1	0800 H
F2	1000 H
F3	1800 H

Estos comandos pueden iniciar la ejecucion de programas que se encuentren en otras direcciones, alterando las instrucciones del Monitor que se localizan en las direcciones 022D H, 022A H y 0227 H para F1, F2 y F3 respectivamente.



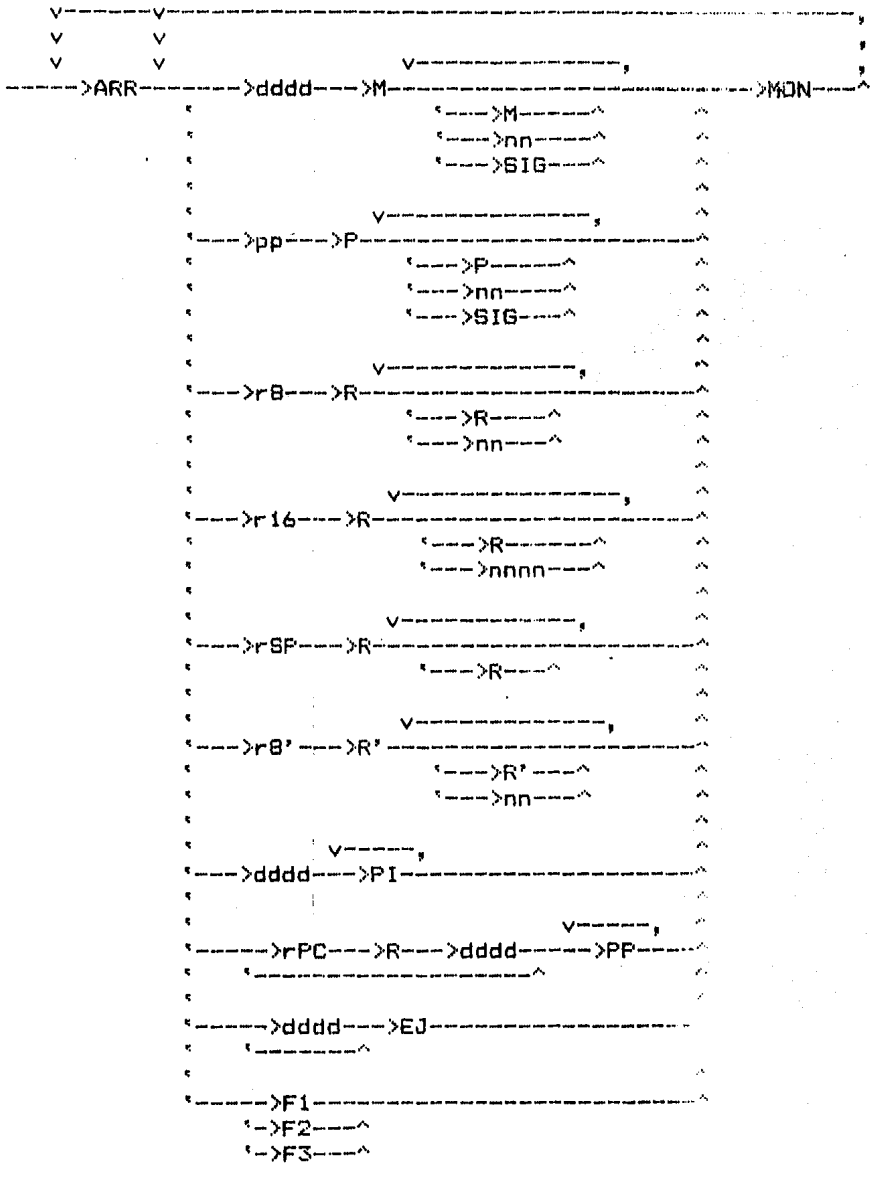


Fig. 4. Resumen Grafico de la Sintaxis de los Comandos.

III. Diseno de la EDH.

III.1. El Sistema de Bus. El Sistema EDH tiene una arquitectura en base a un Sistema de Bus. El Sistema de Bus esta definido en una Tarjeta Madre sobre la cual se encuentran soldados varios conectores que permiten conectar a las Tarjetas de los diferentes Modulos al Bus. El Sistema de Bus esta dividido en 3 Buses:

a. Bus Emulador.
b. Bus Libre.
c. Bus EDH.

El Bus Emulador permite la transferencia de senales para la prueba de Modulos. Este Bus es controlado por el Modulo del Emulador y ocupa 40 terminales del Sistema de Bus. Las senales de este Bus son las mismas del Microprocesador Z80, exeptuando la senal de energia que es de +8V.
El Bus Libre esta a disposicion del usuario. Cuando se usa el Modulo del Emulador, este Bus se usa para interconectar las 2 Tarjetas que forman el Modulo y, por tanto, no debe usarse para algun otro fin. Cuando no se conecta el Modulo del Emulador, el usuario puede definir libremente las senales que debe soportar este Bus.
El Bus EDH es la base de la arquitectura del Sistema EDH. Por este Bus se soportan todas las senales necesarias para la operacion del Sistema. Este Bus es controlado por el Modulo del Procesador y contiene todas las senales del Microprocesador Z80, exeptuando la senal de energia que es de +8V.

III.1.1. Requisitos Mecanicos. La Tarjeta Madre y sus conectores, estan disenados para aceptar Tarjetas de Modulo que contengan 100 terminales, 50 por lado, con un espaciamiento entre terminales del mismo lado de 0.125 de pulgada. Las Tarjetas de Modulo deberan tener 5.3 por 10 por .062 de pulgada (ancho, alto y grosor). Este tipo de Tarjetas son similares, en dimensiones, a las que usa el Bus S-100. Las terminales de las Tarjetas de Modulo estan numeradas en la siguiente forma:

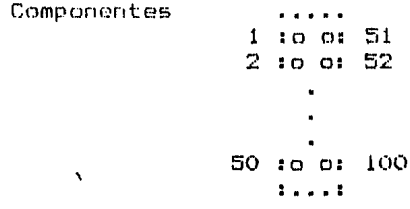


Fig. 5. Terminales de las Tarjetas de Modulo.

Cada Tarjeta de Modulo tiene un Lado de Componentes y un Lado de Pistas. El Lado de Componentes esta del lado de las Terminales 1 a 50, mientras que el Lado de Pistas se encuentra en el lado de las terminales 51 a 100.

La Tarjeta Madre, vista de atras, se observa de la siguiente manera:

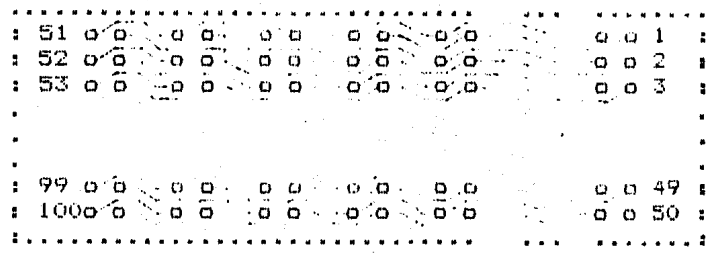


Fig. 6. Tarjeta Madre.

III.1.2. Requisitos Logicos. Los requisitos logicos del Bus definen la senal que existira en cada terminal.
En el Bus Emulador, la direccion de cada senal se ha definido con respecto al Modulo del Emulador, mientras que en el Bus EDH la direccion de cada senal se ha definido con respecto al Modulo del Procesador.
En el Bus Libre, la direccion de las senales depende del usuario, exepto cuando se conecta el Modulo del Emulador, ya que en este caso se encuentran definidas segun se indica en las tablas siguientes.
El Bus Emulador y el Bus EDH se dividen a su vez, desde el punto de vista logico, en 4 Busus:

Bus de Energia,
Bus de Datos,
Bus de Direcciones y
Bus de Control.

La descripcion de los requisitos logicos del Sistema de Bus, se muestra en las siguientes tablas.

REQUISITOS LOGICOS DEL BUS EMULADOR:

Lado de Componentes			Lado de Pistas		
T SENAL	DIR	DESCRIPCION	T SENAL	DIR	DESCRIPCION
Bus de Energia:					
1 +8V	E	+8 Volts DC			
Bus de Datos:					
2 D3	E/S	Dato 3	51 D7	E/S	Dato 7
3 D2	E/S	Dato 2	52 D6	E/S	Dato 6
4 D1	E/S	Dato 1	53 D5	E/S	Dato 5
5 D0	E/S	Dato 0	54 D4	E/S	Dato 4
Bus de Direcciones:					
6 A7	S	Lin Dir 7	55 A15	S	Lin Dir 15
7 A6	S	Lin Dir 6	56 A14	S	Lin Dir 14
8 A5	S	Lin Dir 5	57 A13	S	Lin Dir 13
9 A4	S	Lin Dir 4	58 A12	S	Lin Dir 12
10 A3	S	Lin Dir 3	59 A11	S	Lin Dir 11
11 A2	S	Lin Dir 2	60 A10	S	Lin Dir 10
12 A1	S	Lin Dir 1	61 A9	S	Lin Dir 9
13 A0	S	Lin Dir 0	62 A8	S	Lin Dir 8
Bus de Control:					
14 /WR	S	Ord de Escr	63 /RD	S	Ord de Lect
15 /IORQ	S	Sol de E/S	64 /MEMRQ	S	Sol de Mem
16 /REFRESH	S	Sen Refresh	65 /STATUS1	S	Ciclo Fetch
17 /BUSAK	S	Lib de Bus	66 /BUSRQ	E	Sol de Bus
18 /HALT	S	Sen Halt	67 /INTRQ	E	Sol de Int
19 /WAITRQ	E	Sol de Esp	68 /NMIRQ	E	Sol de NMI
			69 /CLOCK	S	Rel del Pro
			70 /SYSRESET	S	Res del Sis
Bus de Energia:					
20 0V	E	0 Volts DC			

REQUISITOS LOGICOS DEL BUS EDH:

Lado de Componentes			Lado de Pistas		
T SENAL	DIR	DESCRIPCION	T SENAL	DIR	DESCRIPCION
Bus de Energia:					
31 +8V	E	+8 Volts DC			
Bus de Datos:					
32 D3	E/S	Dato 3	81 D7	E/S	Dato 7
33 D2	E/S	Dato 2	82 D6	E/S	Dato 6
34 D1	E/S	Dato 1	83 D5	E/S	Dato 5
35 D0	E/S	Dato 0	84 D4	E/S	Dato 4
Bus de Direcciones:					
36 A7	S	Lin Dir 7	85 A15	S	Lin Dir 15
37 A6	S	Lin Dir 6	86 A14	S	Lin Dir 14
38 A5	S	Lin Dir 5	87 A13	S	Lin Dir 13
39 A4	S	Lin Dir 4	88 A12	S	Lin Dir 12
40 A3	S	Lin Dir 3	89 A11	S	Lin Dir 11
41 A2	S	Lin Dir 2	90 A10	S	Lin Dir 10
42 A1	S	Lin Dir 1	91 A9	S	Lin Dir 9
43 A0	S	Lin Dir 0	92 A8	S	Lin Dir 8
Bus de Control:					
44 /WR	S	Ord de Escr	93 /RD	S	Ord de Lect
45 /IORQ	S	Sol de E/S	94 /MEMRQ	S	Sol de Mem
46 /REFRESH	S	Sen Refresh	95 /STATUS1	S	Ciclo Fetch
47 /BUSAK	S	Lib de Bus	96 /BUSRQ	E	Sol de Bus
48 /HALT	S	Sen Halt	97 /INTRQ	E	Sol de Int
49 /WAITRQ	E	Sol de Esp	98 /NMIRQ	E	Sol de NMI
			99 /CLOCK	S	Rel del Pro
			100 /SYSRESET	S	Res del Sis
Bus de Energia:					
50 0V	E	0 Volts DC			

REQUISITOS LOGICOS DEL BUS LIBRE:
(Solo cuando esta conectado el Modulo del Emulador)

Lado de Componentes			Lado de Pistas		
T	SENAL	DIR DESCRIPCION	T	SENAL	DIR DESCRIPCION
Datos:					
21	D3	E/S Dato 3	71	D7	E/S Dato 7
22	D2	E/S Dato 2	72	D6	E/S Dato 6
23	D1	E/S Dato 1	73	D5	E/S Dato 5
24	D0	E/S Dato 0	74	D4	E/S Dato 4
Numero de Ciclo de Maquina:					
25	M1	A>B Lin Dir 1	75	M3	A>B Lin Dir 3
26	M0	A>B Lin Dir 0	76	M2	A>B Lin Dir 2
Senales de Control:					
27	/RFSH	A>B Sen Refresh	77	/CLOCK	B>A Rel del Pro
Activacion de Puertos:					
28	/E	B>A Puerto PE	78	/LC	B>A Puerto PLC
29	/LCC	B>A Puerto PLCC	79	/LCM	B>A Puerto PLCM
30	/WMD	B>A Puerto PWMD	80	/RMD	B>A Puerto PRMD

Las abreviaciones usadas en las Tablas anteriores, tienen el siguiente significado:

E	Entrada	Dir	Direccion
E/S	Entrada/Salida	Escr	Escritura
S	Salida	Lect	Lectura
Lin	Linea	Mem	Memoria
Ord	Orden	Int	Interrupcion
Sol	Solicitud	Esp	Espera
Sen	Senal	NMI	Int no mascarable
Lib	Liberacion	Sis	Sistema
Res	Reset	Pro	Procesador
Rel	Reloj		

A>B	Tarjeta A hacia Tarjeta B
B>A	Tarjeta B hacia Tarjeta A

III.1.3. Requisitos Electricos. Los requisitos electricos del Sistema de Bus definen los niveles de voltaje y corriente que se recomiendan para los Modulos que se conecten al Bus.

	Receptores del Bus	Emisores al Bus
Nivel Bajo:	0.8 Vmax a -0.36 mA	0.5 Vmax a 24 mA
Nivel Alto:	2.0 Vmin a 20 uA	2.4 Vmin a -15 mA
Tercer Estado:		+100 uA, -100 uA

III.1.4. Requisitos de Tiempo. Los requisitos de tiempo del Bus Emulador dependen de la Senal de Reloj (CLOCK) del Modulo del Emulador. El usuario tiene dos opciones para seleccionar esta senal: elegir la Senal /CLOCK del Bus EDH o elegir la Senal de Reloj del Modulo Bajo Prueba. Si el usuario elige la Senal /CLOCK del Bus EDH, los requisitos de tiempo del Bus son similares a los del Bus EDH. Si elige la Senal de Reloj del Modulo Bajo Prueba, los requisitos de tiempo dependen directamente de dicha Senal. La Senal /CLOCK del Bus EDH es de 1.9968 MHz.

Los requisitos de tiempo del Bus Libre dependen directamente de los requerimientos particulares del usuario.

*

III.2. El Modulo del Procesador. El Modulo del Procesador es el encargado de controlar al Bus EDH. Contiene al Microprocesador Z80 y la logica necesaria para enviar sus senales al Bus. Contiene ademas, el hardware necesario para generar las senales /CLOCK Y /RESET del Bus EDH.

Para fines de prueba y desarrollo de Modulos, contiene un Conector para cable de 40 terminales, para proporcionar todas las senales del Microprocesador Z80. Tales senales pueden ser enviadas y recibidas a traves del Bus EDH o a traves del Bus Emulador, dependiendo de la posicion del Conmutador de Bus. Existe tambien un Conmutador de Energia para proporcionar opcionalmente la senal de +5V al Conector.

La siguiente figura muestra un Diagrama a Bloques del Modulo:

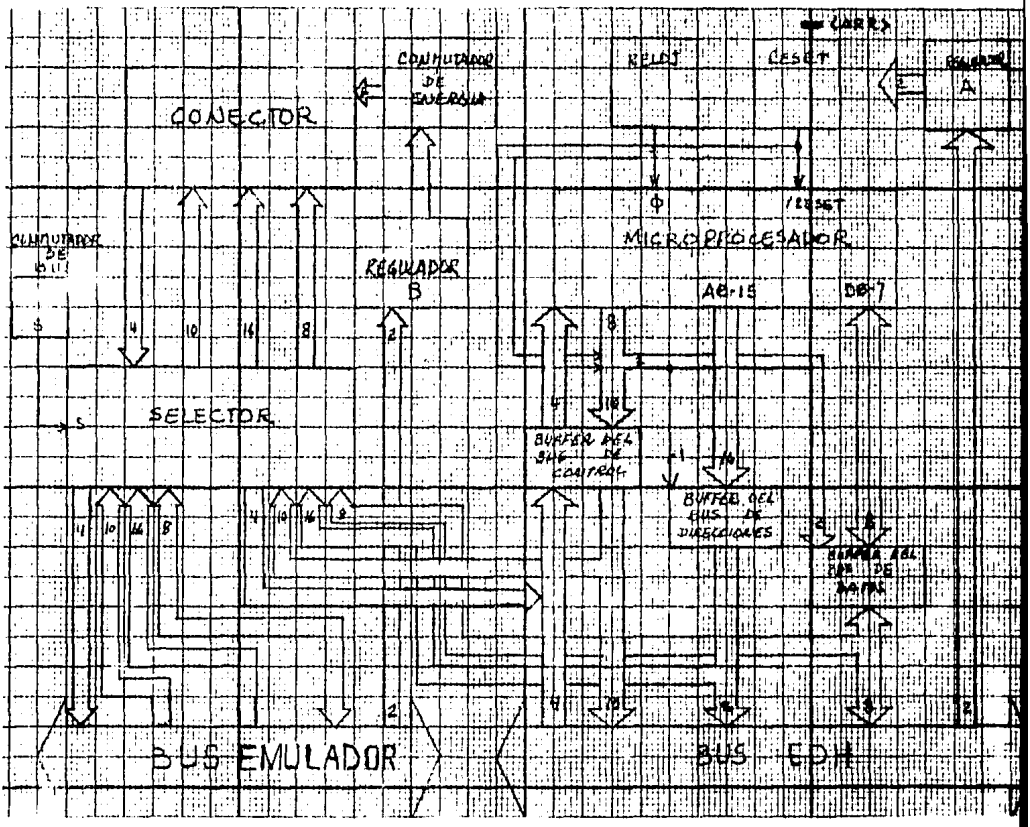


Fig. 7. Diagrama a Bloques del Modulo del Procesador.

Las siguientes figuras muestran el contenido de cada bloque.

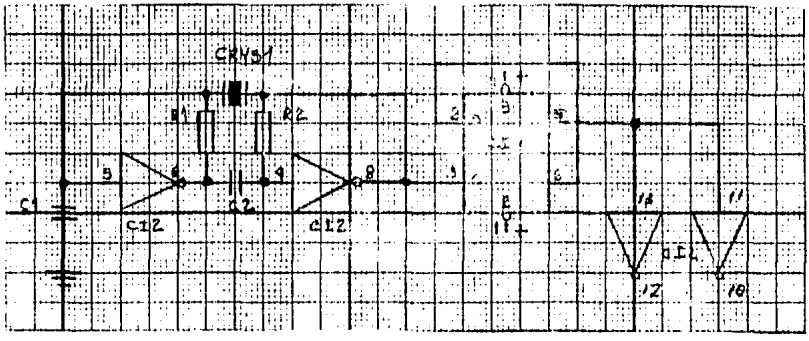


Fig. 8. Bloque del Reloj.

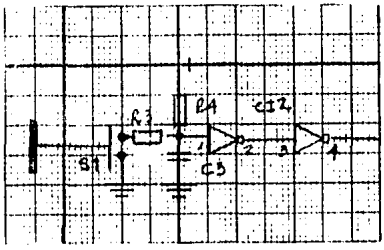


Fig. 9. Bloque del Reset.

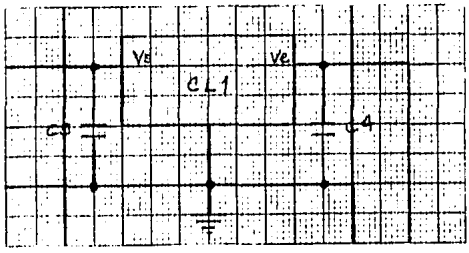


Fig. 10. Bloque del Regulador A.

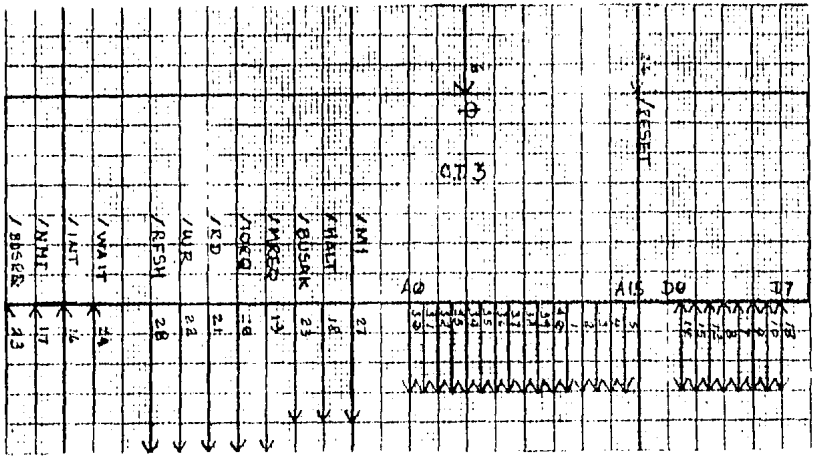


Fig. 11. Bloque del Microprocesador.

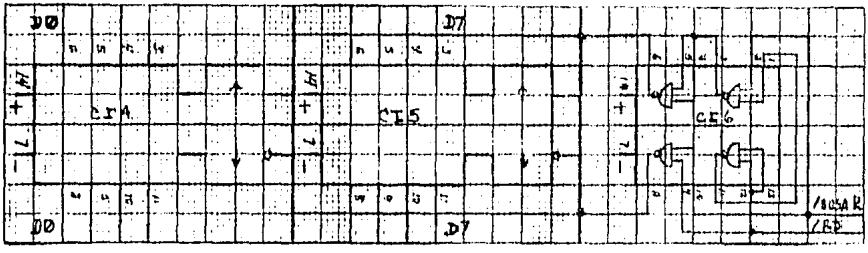


Fig. 12. Bloque del Buffer del Bus de Datos.

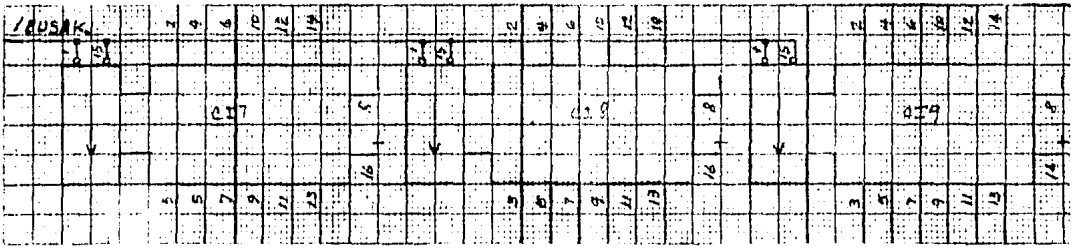


Fig. 13. Bloque del Buffer del Bus de Direcciones.

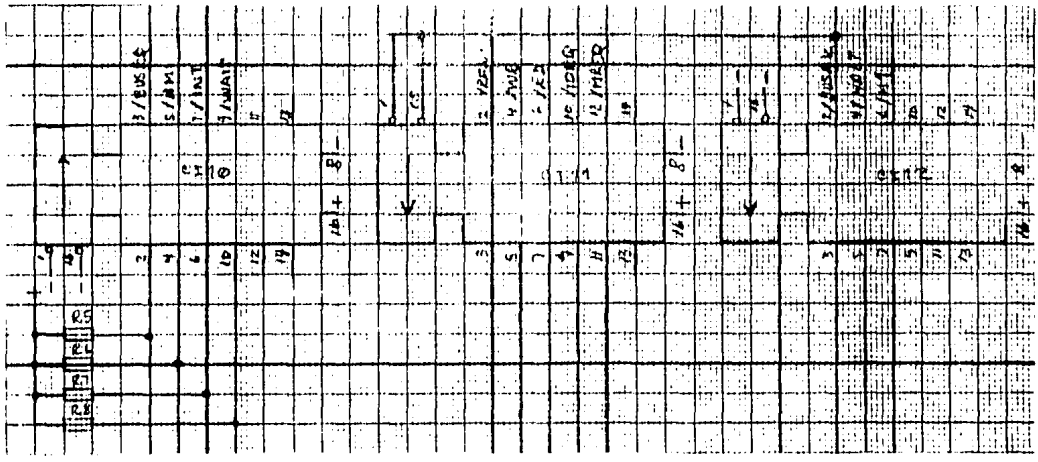


Fig. 14. Bloque del Buffer del Bus de Control.

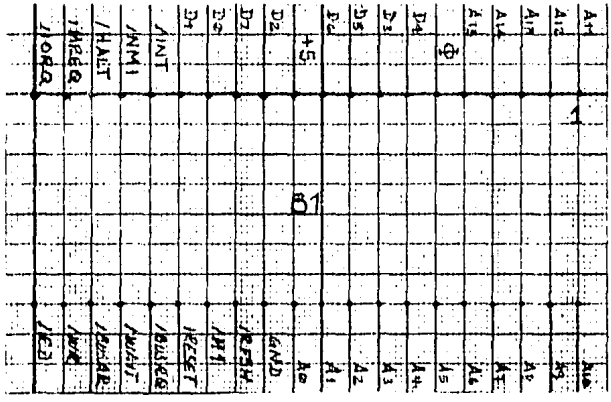


Fig. 15. Bloque del Conector.

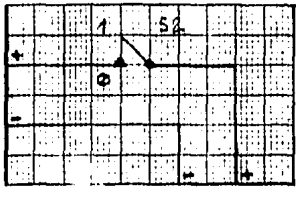


Fig. 16. Bloque del Conmutador de Energia.

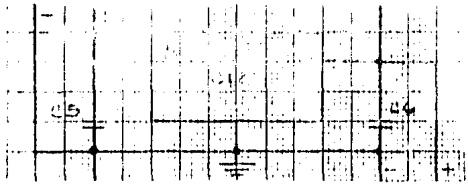


Fig. 17. Bloque del Regulador 8.

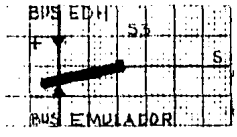


Fig. 18. Bloque del Conmutador de Bus.

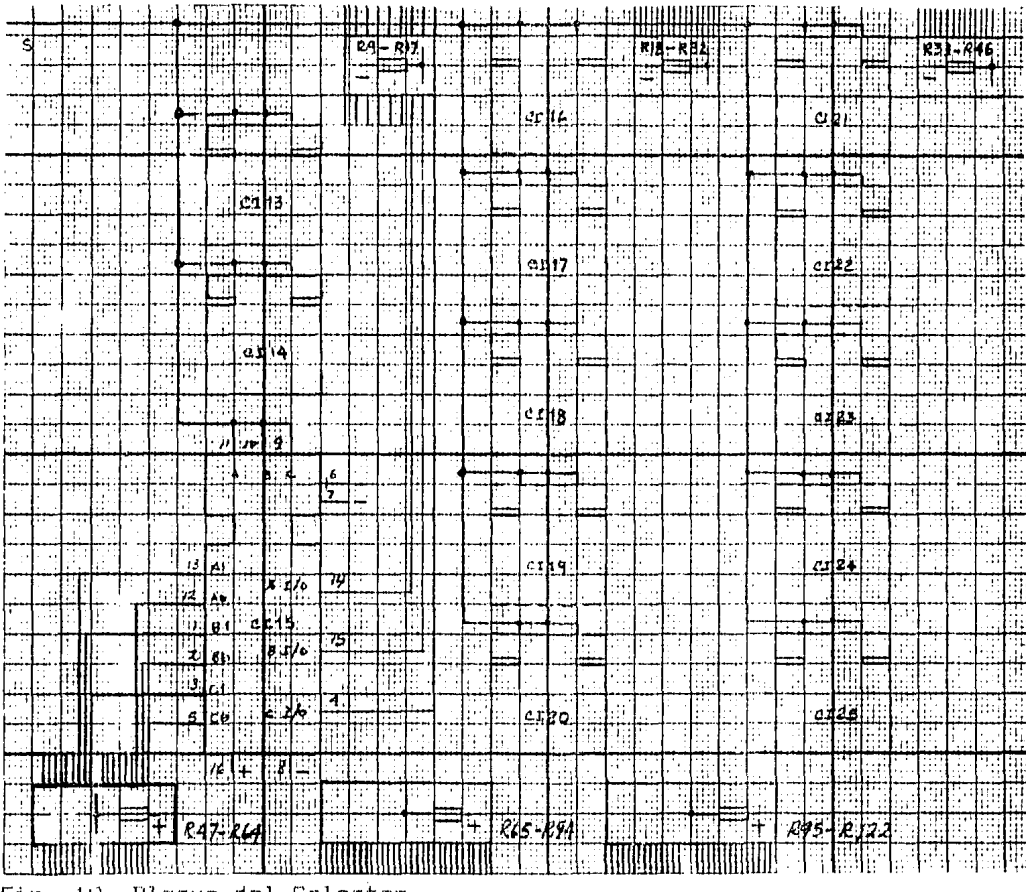


Fig. 19. Bloque del Selector.

III.2.1. Lista de Componentes.

Componente	Descripcion
CRYS1	Cristal a 3.9936 MHz.
R1-R2	Resistencias de 1 kilohm a 250 mW.
C1	Capacitor de 10 pF.
C2	Capacitor de .01 uF.
CI1	Circuito Integrado NOT 74LS04.
CI1	Circuito Integrado FFD 74LS74.
CI2	Circuito Integrado NOT 74LS04.
R3	Resistencia de 1 kilohm a 250 mW.
S1	Conmutador de Presion.
R4	Resistencia de 10 kilohms a 250 mW.
C3	Capacitor de 1 uF.
CL1	Regulador LM323K 5V/3A.
C3	Capacitor de 1uF.
C4	Capacitor de .1 uF.
CI3	Microprocesador Z80 2MHz.
CI4-CI5	Circuitos Integrados TRANS/REC 74LS243.
CI6	Circuito Integrado NAND 74LS00.
CI7-CI12	Circuitos Integrados BUFFER 74LS367.
R5-R8	Resistencias de 1 kilohm a 250 mW.
B1	Base para Conector de 40 Terminales.
S2	Conmutador de 2 Posiciones para 2 A.
CL2	Regulador LM323K 5V/3A.
C5	Capacitor de 1 uF.
C6	Capacitor de .1 uF.
S3	Conmutador de 2 Posiciones.
CI13-CI25	Circuitos Integrados SELECTOR CMOS 4066.
R9-R46	Resistencias de 1 kilohm a 250 mW.
R47-R122	Resistencias de 2.2 kilohm a 250 mW.

III.3. El Modulo de Memoria. El Modulo de Memoria del Sistema EDH esta diseñado para soportar circuitos integrados de memoria de solo lectura o de lectura/escritura, en combinaciones de 2 Kbytes.

El Modulo usa memorias de solo lectura y de lectura/escritura que son compatibles en cuanto a sus terminales, lo que permite una versatilidad importante al tener un Modulo de Memoria del tipo universal. Las memorias de lectura/escritura son del tipo estatico.
Los aspectos sobresalientes del Modulo de Memoria son:

a) 16 Kbytes de Memoria.
b) Seleccion del Rango de Direcionamiento.
c) Seleccion de Banco de 64 Kbytes.
d) Habilitacion o Deshabilitacion con /SYSRESET.
e) Activacion o Desactivacion por Programa.

La Memoria del Sistema se encuentra organizada en 2 Bancos. Cada Banco tiene una capacidad de 64 Kbytes, por lo que tiene una capacidad total de 128 Kbytes. Cada Banco esta integrado por 4 Modulos de Memoria de 16 Kbytes.

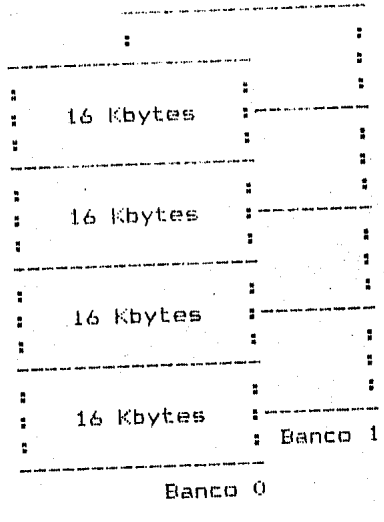


Fig. 20. Memoria del Sistema EDH.

III.3.1. Opciones de Funcionamiento. Antes de conectar el Modulo de Memoria en un conector del Bus del Sistema EDH, se deben elegir ciertas opciones. Las opciones se eligen por medio de conmutadores que se encuentran en la propia Tarjeta del Modulo de Memoria. Las opciones del Modulo de Memoria son:

a) Rango de Direccionamiento. Existen 4 posiciones dentro del rango de direccionamiento del Sistema que se pueden elegir para cada Modulo. Esta opcion de elige por medio de 2 conmutadores A15 y A14.

A15	A14	Rango de Direccionamiento
0	0	0000 H a 3FFF H
0	1	4000 H a 7FFF H
1	0	8000 H a BFFF H
1	1	C000 H a FFFF H

b) Banco de 64 Kbytes. Para definir los Bancos a los que pertenece cada Modulo se usan 2 conmutadores B0 y B1:

B1	B0	Banco de 64 Kbytes
0	0	Ningun Banco
0	1	Banco 0
1	0	Banco 1
1	1	Ambos Bancos

Cuando la Tarjeta del Modulo de Memoria esta operando en alguno de los bancos que los conmutadores B0 y B1 definen, enciende un diodo emisor de luz denominado - B. Esto permite saber que una determinada Tarjeta del Modulo de Memoria ha recibido el Primer Nivel de Seleccion o Seleccion de Banco, y que este coincide con lo definido por B0 y B1.

c) Habilitacion/Deshabilitacion con /SYSRESET. Esta opcion permite la Habilitacion o Deshabilitacion Logica del Modulo en cuestion despues de la senal /SYSRESET, en forma automatica. La opcion a elegir se realiza por medio del conmutador H.

H	Habilitacion/Deshabilitacion con /SYSRESET
0	Deshabilitacion
1	Habilitacion

d) Activacion/Desactivacion por Programa. Se explica en la parte de Direccionamiento (III.3.3).

III.3.2. Direccionamiento. Para direccionar una palabra en el Sistema EDH, se usan 2 niveles de seleccion: se selecciona un Banco de Memoria y se selecciona una palabra dentro del Banco.
El procesador puede direccionar directamente hasta 64 Kbytes de Memoria. Para poder direccionar mas alla de esta capacidad, usa a los Puertos para enviar informacion relativa al Banco que se requiere usar. Los Bancos por tanto, se activan y se desactivan bajo el control del programa, cuando el procesador envia una Palabra de Seleccion de Banco al Puerto 40 H que se encuentra en cada una de las Tarjetas de los Bancos.

Primer Nivel de Seleccion:

Puerto 40 H	Seleccion de Banco

Segundo Nivel de Seleccion:

A15-A14	Seleccion de Modulo de 16 Kbytes
A13-A0	Seleccion de la Palabra

Para activar o desactivar un Banco, el dato que se envia al Puerto 40 H de cada Tarjeta tiene el siguiente significado:

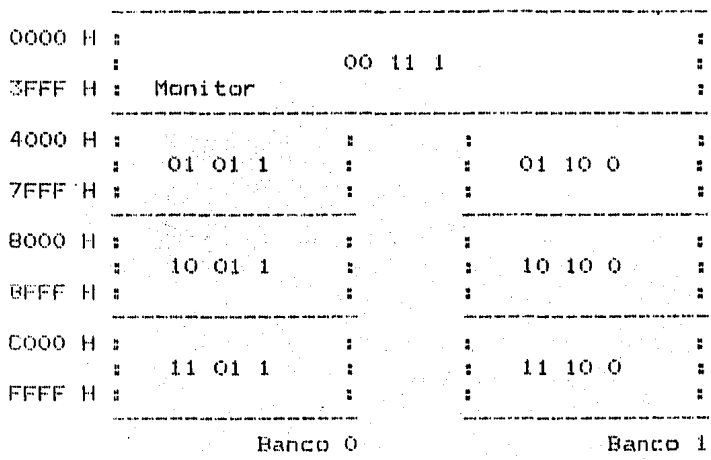
b7 a b2	b1	b0	Banco Activado
X	0	0	Ningun Banco
X	0	1	Banco 0
X	1	0	Banco 1
X	1	1	Ambos Bancos

Un Banco permanecera Activado o Desactivado hasta que reciba otra Palabra de Seleccion de Banco, por lo que generalmente, el procesador solo debe enviar el Segundo Nivel de Seleccion, a traves de las Lineas de Direccion A15-A0.

Debido a la facilidad de conmutacion de los Bancos, el usuario debe tener cuidado en mantener la continuidad en la ejecucion de su programa, esto es, asegurarse de que despues de que se ha conmutado de Banco, el procesador ejecutara la instruccion correcta. Debe observarse que la Activacion o Desactivacion de los Bancos se realiza en forma simultanea y que, por otra parte el procesador ignora las fronteras de los Bancos y simplemente incrementa su registro PC para direccionar a la siguiente instruccion.

III.3.3. Configuracion. Esta organizacion de la memoria, permite una gran flexibilidad en la arquitectura del Sistema EDH. Es posible, por ejemplo, convertir al Sistema Uniusuario en Multiusuario con la programatica adecuada, o ejecutar programas cuyos requerimientos de memoria alcancen los 128 Kbytes.

La configuracion estandar del Sistema EDH, usa un Modulo comun a ambos Bancos, en donde se localiza la informacion relacionada con la conmutacion de Bancos, ademas del Monitor.
La configuracion estandar del Sistema EDH se muestra en la siguiente figura:



Nota: Los digitos dentro de cada Modulo corresponden a los valores de los conmutadores A15, A14, B1, B0 y H.

Fig. 21. Configuracion Estandar de Memoria del EDH.

III.3.4. Construccion. La construccion del Modulo de Memoria se ha realizado tomando en cuenta las anteriores consideraciones de diseno. La siguiente figura muestra el Diagrama a Bloques del Modulo de Memoria.

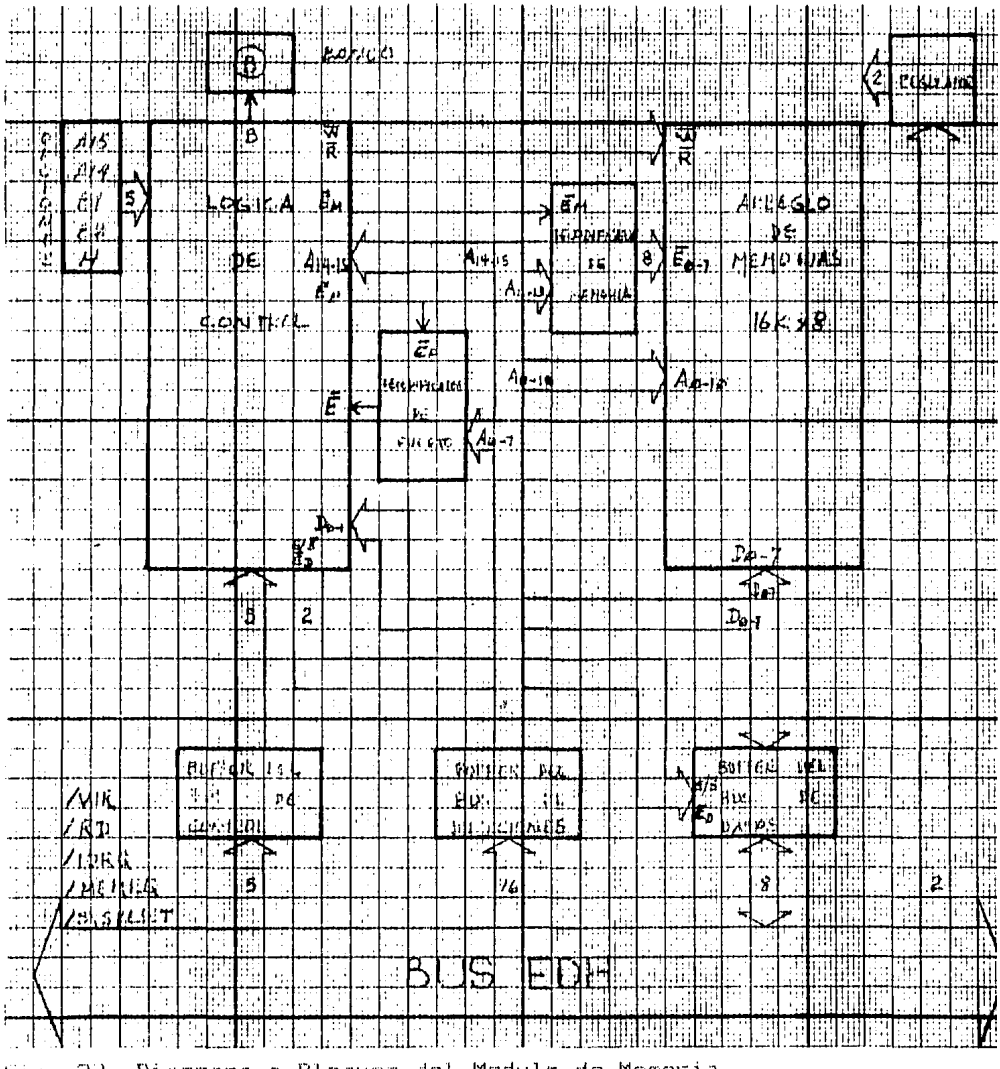


Fig. 22. Diagrama a Bloques del Modulo de Memoria.

Las siguientes figuras muestran el contenido de cada Bloque.

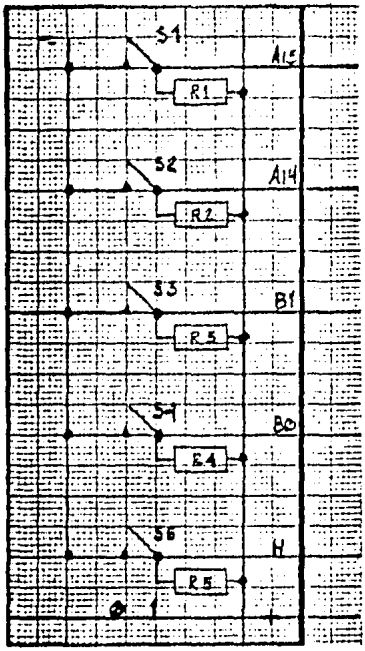


Fig. 23. Bloque de Opciones.

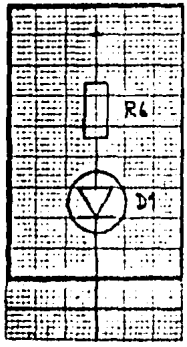


Fig. 24. Bloque del Banco.

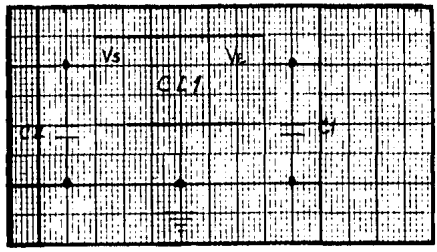


Fig. 25. Bloque del Regulador.

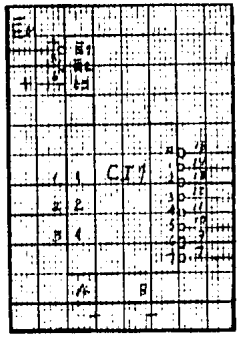


Fig. 26. Bloque del Decodificador de Memoria.

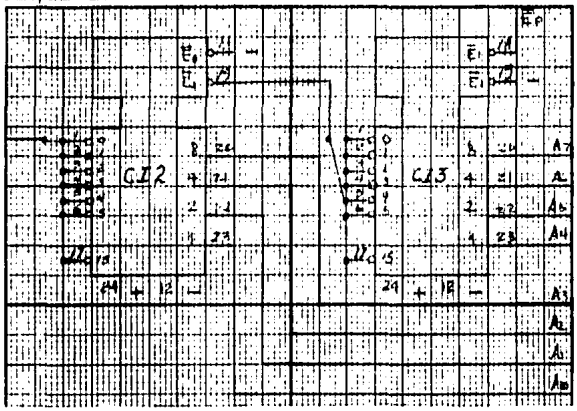


Fig. 27. Bloque del Decodificador de Puerto.

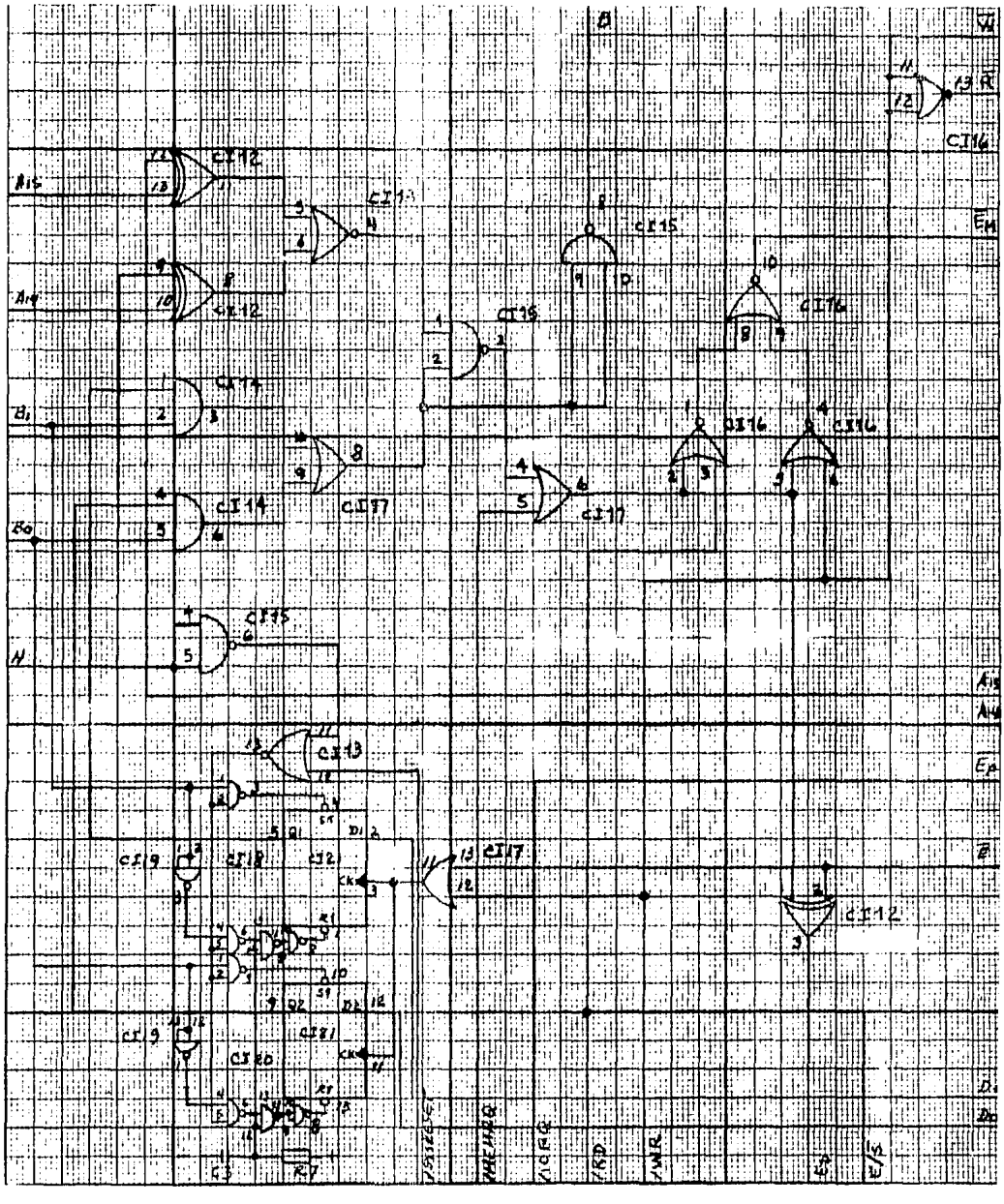


Fig. 28. Bloque de la Logica de Control.

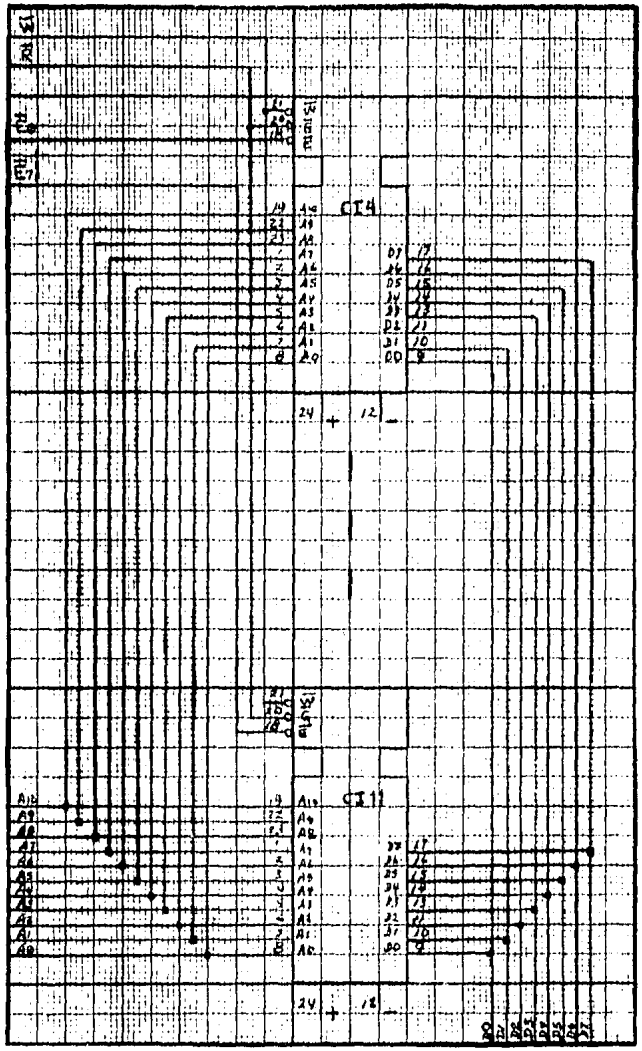


Fig. 2%. Bloque del Arreglo de Memorias.

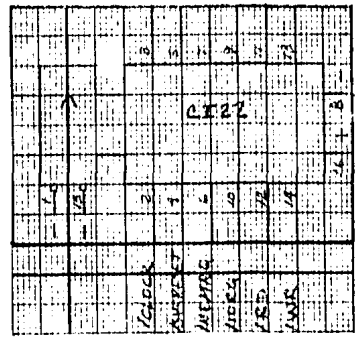


Fig. 30. Bloque del Buffer del Bus de Control.

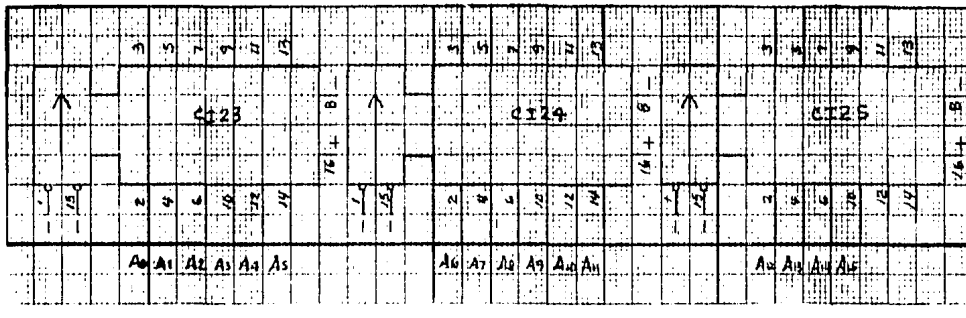


Fig. 31. Bloque del Buffer del Bus de Direcciones.

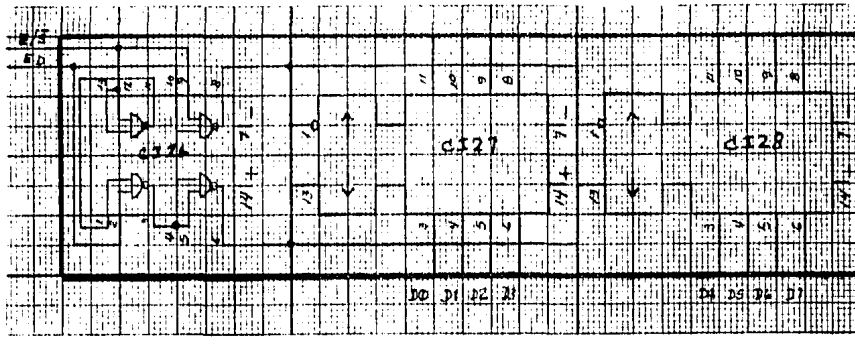


Fig. 32. Bloque del Buffer del Bus de Datos.

III.3.5. Lista de Componentes.

Componente	Descripcion
S1-S5	Conmutadores de 2 Posiciones.
R1-R5	Resistencias de 1 kilohm a 250 mW.
R6	Resistencia de 270 ohms a 250 mW.
D1	Diodo Emisor de Luz serie 1N4000.
CL1	Regulador LM323K 5V/3A.
C1	Capacitor de .1 uF.
C2	Capacitor de 1 uF.
CI1	Circuito Integrado DECOD 1/8 74LS138.
CI2-CI3	Circuitos Integrados DECOD 1/16 74LS154.
CI4-CI11	Memorias RWM 65116/PROM 2716.
CI12	Circuito Integrado OR EX 74LS86.
CI13	Circuito Integrado NOR 74LS02.
CI14	Circuito Integrado AND 74LS08.
CI15	Circuito Integrado NAND 74LS00.
CI16	Circuito Integrado NOR 74LS02.
CI17	Circuito Integrado OR 74LS32.
CI18-CI20	Circuitos Integrados NAND 74LS00.
C3	Capacitor de 22 uF.
R7	Resistencia de 1 kilohm a 250 mW.
CI21	Circuito Integrado FFD 74LS74.
CI22-CI25	Circuitos Integrados BUFFER 74LS367.
CI26	Circuito Integrado NAND 74LS00.
CI27-CI28	Circuitos Integrados TRANS/REC 74LS243.

*

III.4. El Modulo de Teclado/Unidad Visual. El Modulo de Teclado/Unidad Visual permite la comunicacion logica entre el usuario y el Sistema EDH. Este Modulo esta integrado de tres partes:

1. Modulo de Teclado/Unidad Visual. Contiene toda la logica necesaria para el funcionamiento del Modulo. Esta conectado directamente al Bus EDH.

2. Unidad de Teclado/Unidad Visual. Contiene las unidades visuales y el conjunto de teclas.

3. Cable Modulo-Unidad. Interconecta al Modulo de Teclado/Unidad Visual con la Unidad de Teclado/Unidad Visual.

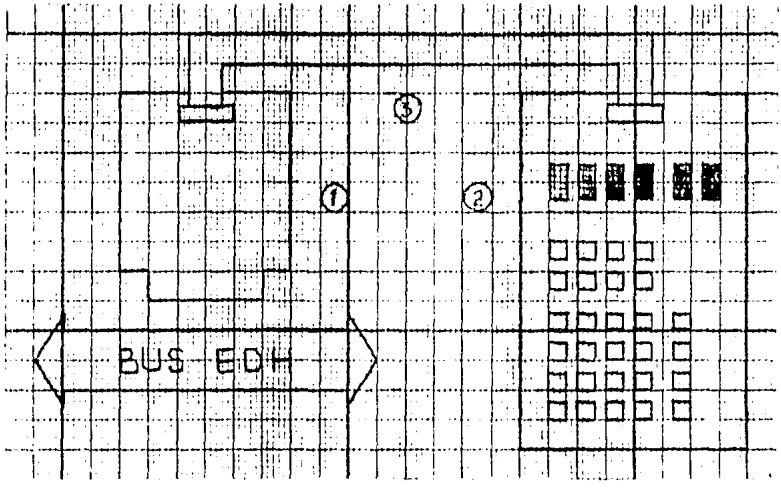


Fig. 33. Partes del Modulo de Teclado/Unidad Visual.

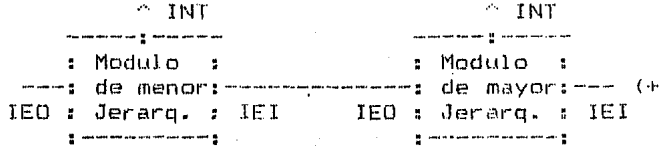
III.4.1. Modulo de Teclado/Unidad Visual. Esta parte permite la interconexion con el Bus del Sistema EDH y controla, a traves del Cable Modulo-Unidad, a la Unidad de Teclado/Unidad Visual. Esta parte contiene toda la logica de control, decodificacion, interconexion e interrupcion necesaria para el funcionamiento del Modulo bajo el Monitor del Sistema EDH.

*

El Monitor del Sistema EDH usa diferentes puertos para manejar a este Modulo. Cuatro puertos se destinan para el circuito integrado Z80-CTC (Counter Timer Circuit) que maneja las operaciones que requieren de interrupciones. Tres puertos se usan en la logica de interconexion para manejar a la Unidad de Teclado/Unidad Visual.

Puerto	Direccion	Uso
84 H	Ent/Sal	Canal 0 del Z80-CTC (Reserva).
85 H	Ent/Sal	Canal 1 del Z80-CTC (Reserva).
86 H	Ent/Sal	Canal 2 del Z80-CTC.
87 H	Ent/Sal	Canal 3 del Z80-CTC (Reserva).
88 H	Salida	Indica los segmentos a encender.
8C H	Salida	Indica el digito a encender y la fila de teclas a explorar.
90 H	Entrada	Informa sobre la tecla oprimida.

Para la ejecucion de los comandos FP y MON se requiere solicitar una interrupcion del tipo NMI (Interrupcion No Mascarable) al microprocesador. Esta solicitud de interrupcion NMI, la maneja la logica de interrupcion a traves del canal 2 del circuito Z80-CTC. Las capacidades restantes del Z80-CTC, tales como los canales 0, 1 y 3 y la solicitud de interrupcion INT, se han dejado disponibles al usuario.
La capacidad del Z80-CTC de solicitar interrupciones del tipo INT incluyen la logica necesaria para crear un sistema de interrupciones jerarquizado, que permita atender solicitudes de interrupcion de diferentes Modulos.
Esta jerarquia de solicitudes de interrupcion se maneja por medio de una Malla de Habilitaciones de Interrupcion. La Malla esta encadenada de tal manera que un dispositivo de mayor jerarquia de interrupcion emitira su solicitud de interrupcion antes que otro de menor jerarquia si ambos dispositivos requieren la interrupcion al mismo tiempo. La Malla de Habilitaciones de Interrupcion permite que un dispositivo dado emita su solicitud de interrupcion si su linea IEI (Entrada de la Habilitacion de Interrupcion) tiene un nivel alto. Cuando este dispositivo esta solicitando o ejecutando una interrupcion pone un nivel bajo en su linea IEO (Salida de la Habilitacion de Interrupcion). La Malla conecta la linea IEO de un dispositivo de mayor jerarquia con la linea IEI de un dispositivo de menor jerarquia.



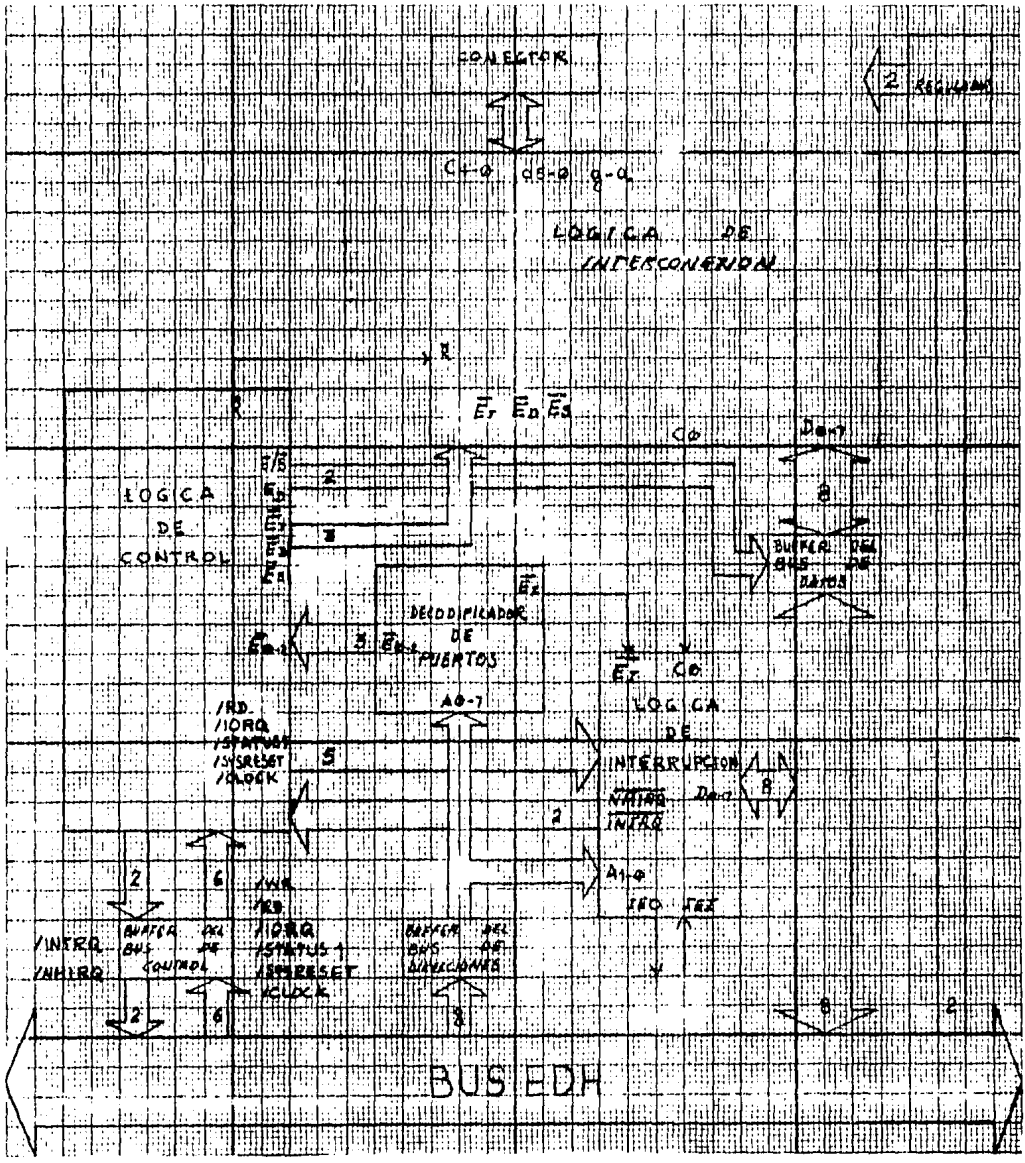


Fig. 34. Diagrama a Bloques del Modulo de Teclado/Unidad Visual.

III.4.2. Unidad de Teclado/Unidad Visual. La Unidad de Teclado/Unidad Visual contiene las 28 teclas que se requieren, ademas de la tecla <ARR>, para operar al Sistema EDH bajo el Monitor del Sistema que se ha descrito. Las teclas estan arregladas en forma matricial segun se puede observar en la Fig. 35. Ademas de las teclas, contiene seis unidades visuales de siete segmentos, en donde normalmente se muestra una direccion de memoria y su dato correspondiente. Vease tambien la Fig. 3.

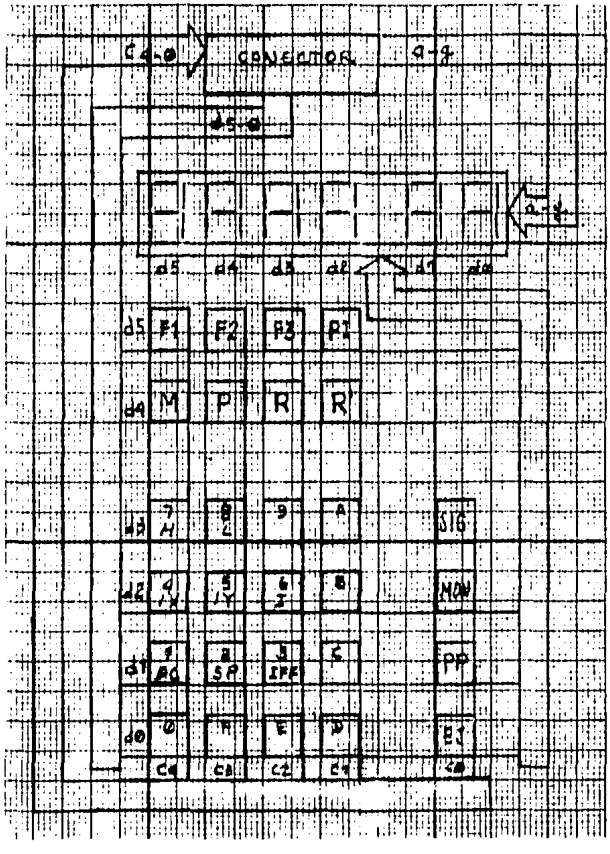


Fig. 35. Unidad de Teclado/Unidad Visual.

III.4.3. Cable Modulo-Unidad. Para interconectar al Modulo de Teclado/-- Unidad Visual con la Unidad de Teclado/Unidad Visual, se usa un cable plano de 24 hilos, 6 de los cuales son de reserva. Este - cable permite colocar a la Unidad de Teclado/Unidad Visual, en el lugar que le presente mayor comodidad para su operacion.

III.4.4. Construccion. La construccion de la Unidad de Teclado/Unidad -- Visual y del Cable Modulo-Unidad es directa de las figuras an-- teriores. La construccion del Modulo de Teclado/Unidad Visual - tiene como base su Diagrama a Bloques de la Fig. 34.

Las siguientes figuras muestran el contenido de cada Bloque.

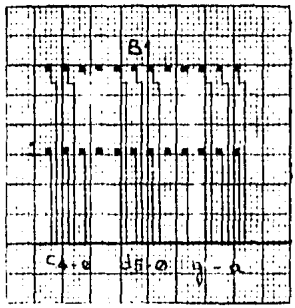


Fig. 36. Bloque del Conector.

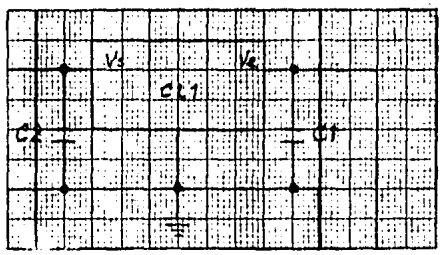


Fig. 37. Bloque del Regulador.

*

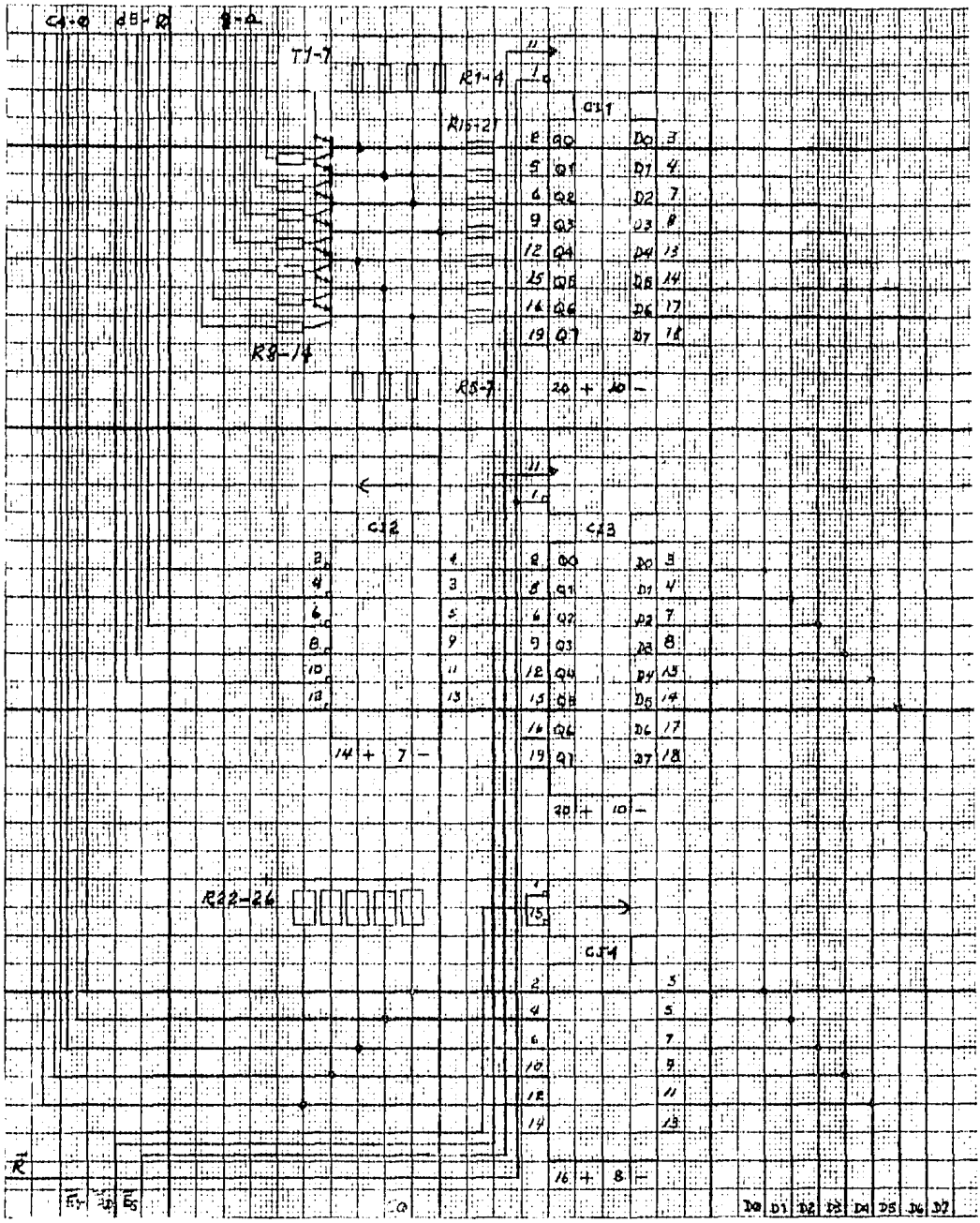


Fig. 3a. Bloque de la Logica de Interconexion.

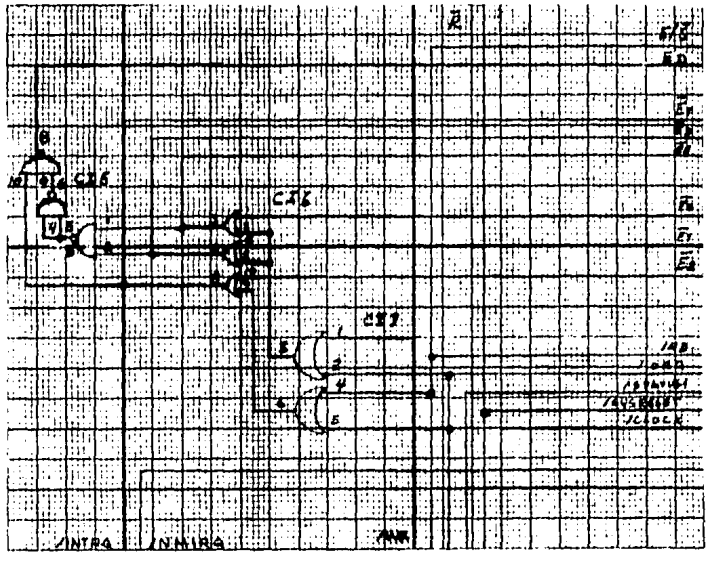


Fig. 39. Bloque de la Logica de Control.

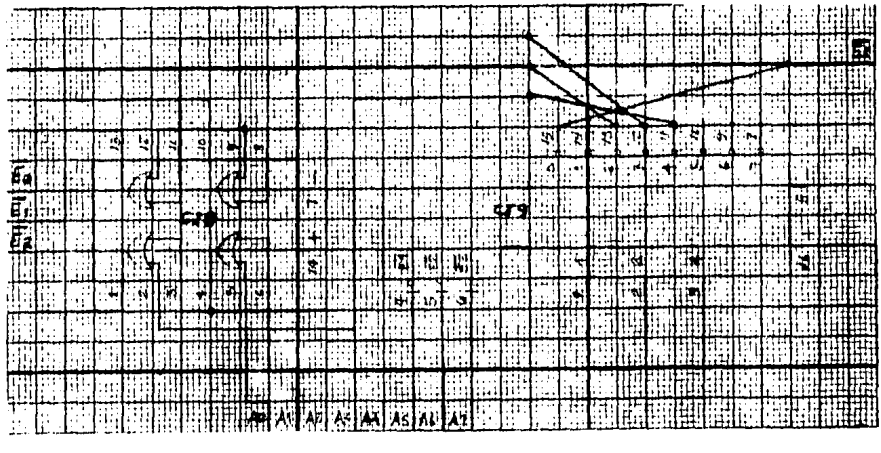


Fig. 40. Bloque del Decodificador de Puertos.

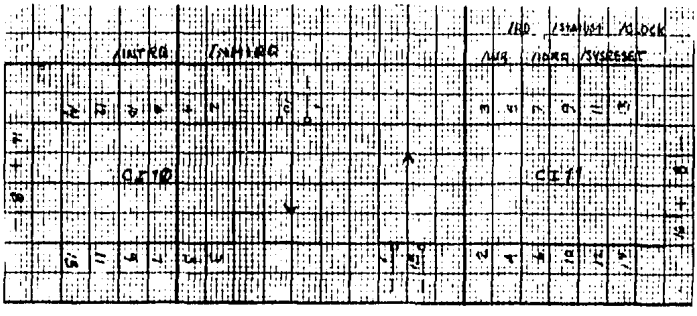


Fig. 41. Bloque del Buffer del Bus de Control.

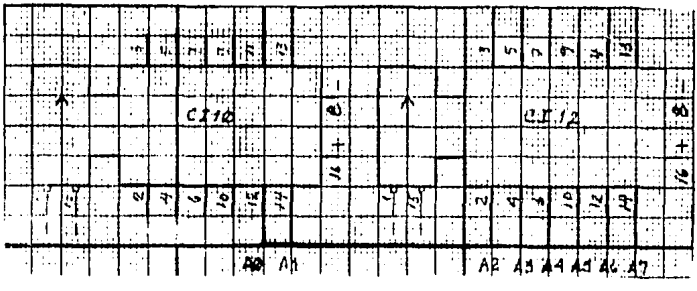


Fig. 42. Bloque del Buffer del Bus de Direcciones.

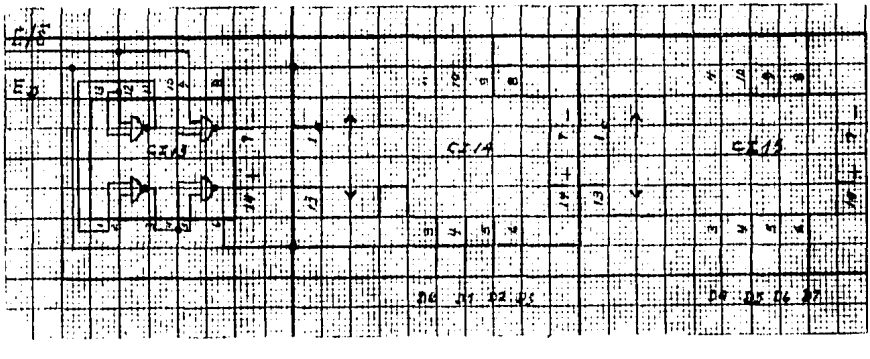


Fig. 43. Bloque del Buffer del Bus de Datos.

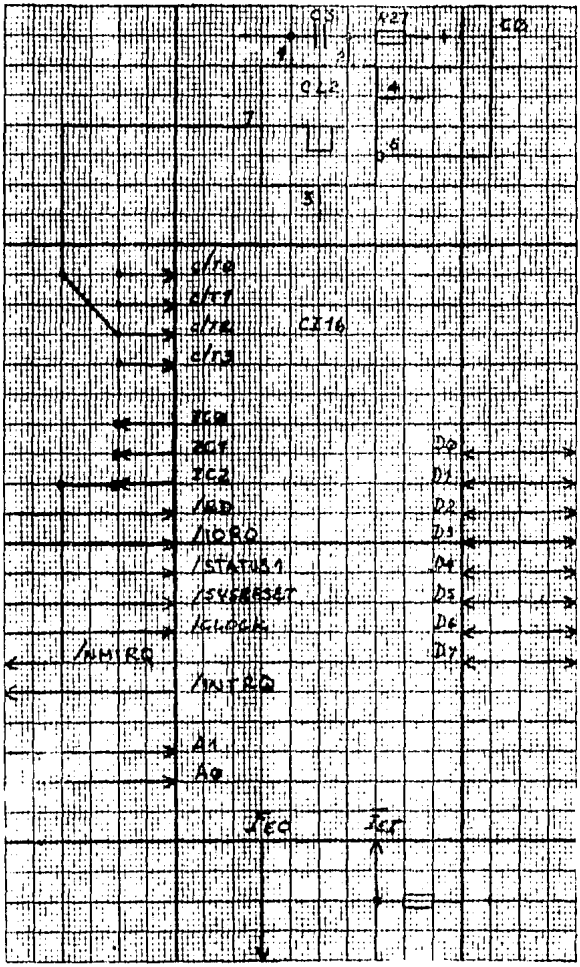


Fig. 44. Bloque de la Logica de Interrupcion.

III.4.5. Lista de Componentes.

Componente	Descripcion
B1	Base para Conector de 24 terminales.
CL1	Regulador LM323K 5V/3A.
C1	Capacitor de .1 uF.
C2	Capacitor de 1 uF.
T1-T7	Transistores PNP MPS 2907.
R1-R7	Resistencias de 10 kilohms a 250 mW.
R8-R14	Resistencias de 68 ohms a 250 mW.
R15-R21	Resistencias de 4.7 kilohms a 250 mW.
R22-R26	Resistencias de 10 kilohms a 250 mW.
CI1	Circuito Integrado FFD 74LS273.
CI2	Circuito Integrado NOT 74LS05.
CI3	Circuito Integrado FFD 74LS273.
CI4	Circuito Integrado BUFFER 74LS367.
CI5	Circuito Integrado NAND 74LS00.
CI6-CI8	Circuitos Integrados OR 74LS32.
CI9	Circuito Integrado DECOD 1/8 74LS138.
CI10-CI12	Circuitos Integrados BUFFER 74LS367.
CI13	Circuito Integrado NAND 74LS00.
CI14-CI15	Circuitos Integrados TRANS/REC 74LS243.
C3	Capacitor de 620 pF.
R27	Resistencia de 470 kilohms a 250 mW.
CL2	Monoestable MC 14538.
CI16	Circuito Integrado Z80-CTC.
R28	Resistencia de 1 kilohm a 250 mW.

*

IV. Aplicaciones.

IV.1. El Modulo del Emulador. El objetivo principal del Sistema EDH es convertirse en una herramienta util en el desarrollo de Modulos para Sistemas de Computo. Un Modulo que es frecuentemente necesario en la prueba y deteccion de faltas en Modulos de Memoria, en Modulos de Puertos de Entrada/Salida y en general en cualquier Modulo que requiera conectarse a un Microprocesador, es el Modulo del Emulador.

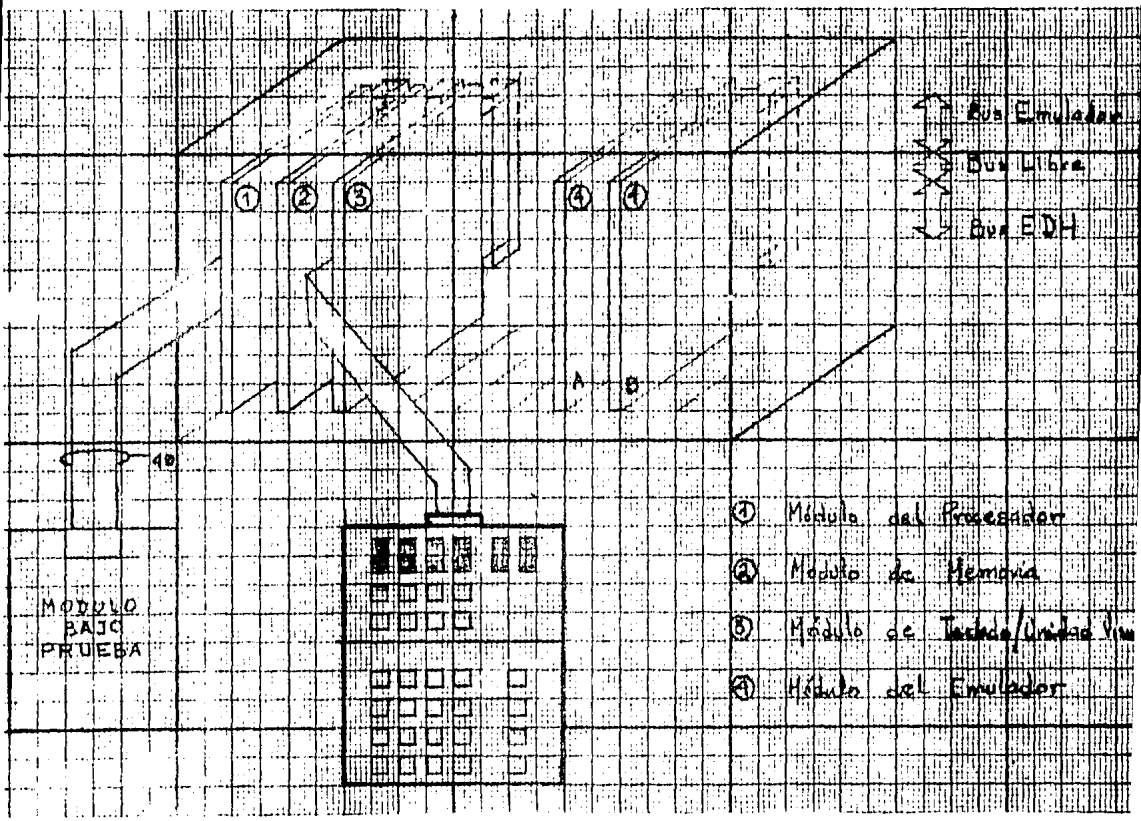


Fig. 15. Sistema EDH con el Modulo del Emulador.

Esencialmente, la funcion de un Emulador es reproducir, bajo control, las senales que genera el Microprocesador durante cada Operacion Basica o Ciclo de Maquina, dado que todas las Instrucciones que ejecuta el Microprocesador, son solamente una serie de Ciclos de Maquina.
Ademas de reproducir bajo control del usuario cada uno de los Ciclos de Maquina del Microprocesador Z80, el Emulador debe permitir que el usuario controle la Secuencia y Cantidad de Ciclos a Ejecutar, asi como la Emision y Recepcion de Datos involucrados en cada Ciclo de Maquina.
Las senales que genera bajo control de programa el Emulador, debe enviarlas al Bus Emulador.

IV.1.1. El Ciclo de Maquina. Un Microprocesador esta disenado para ejecutar un conjunto determinado de Instrucciones. Cada Instruccion o Ciclo de Instruccion, esta formado por una serie de Ciclos de Maquina (Mi). Los Ciclos de Maquina del Microprocesador Z80 son:

a. Ciclo Fetch.
b. Ciclo de Lectura o Escritura de Memoria.
c. Ciclo de Lectura o Escritura de Puerto.
d. Ciclo de Solicitud/Liberacion de Bus.
e. Ciclo de Solicitud/Aceptacion de Interrupcion.
f. Ciclo de Solicitud/Aceptacion de Interrupcion No Mascarable.
g. Ciclo de Salida de la Instruccion HALT.

Cada Ciclo de Maquina puede durar de 3 a 6 Periodos Basicos de Reloj o puede durar mas tiempo a fin de permitir la sincronizacion entre el Microprocesador y los dispositivos que a el se conectan. El Periodo Basico de Reloj se conoce como Estado T (Ti). La siguiente figura muestra un Ciclo de Instruccion con sus Ciclos de Maquina Mi y sus Estados Ti, donde se observa que el primer Ciclo de Maquina de un Ciclo de Instruccion, es el Ciclo Fetch.

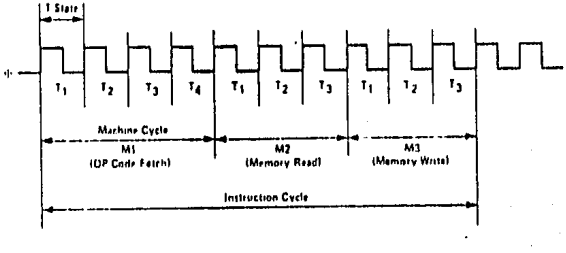


Fig. 46. Ciclo de Instruccion.

IV.1.2. El Ciclo Fetch. El Ciclo Fetch tambien conocido como Ciclo M1 se usa para obtener la Parte Operativa de la instruccion a ser ejecutada. La siguiente figura muestra el diagrama de tiempo que -- debe satisfacer el Emulador para el Ciclo Fetch, sin y con Estados de Espera (Tw).

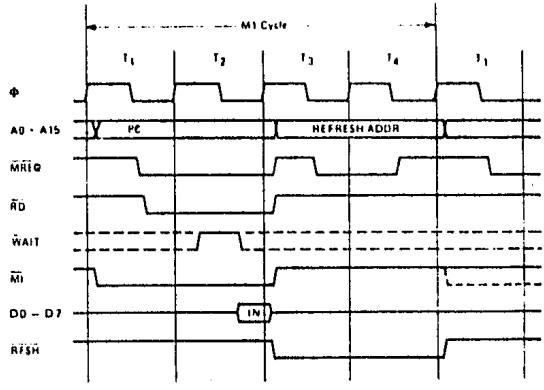


Fig. 47. Ciclo Fetch.

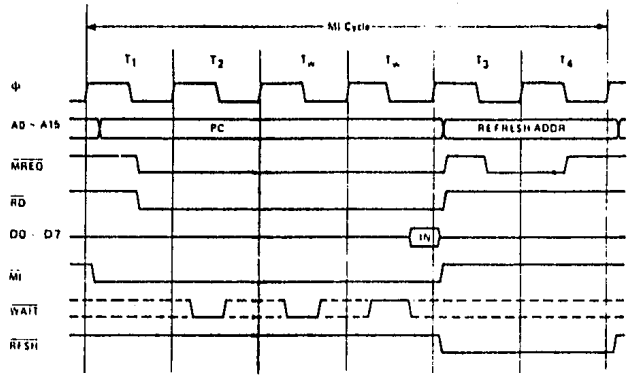


Fig. 48. Ciclo Fetch con Estados de Espera.

*

IV.1.3. Ciclo de Lectura o Escritura de Memoria. El Ciclo de Lectura o - Escritura de Memoria permite leer o escribir informacion en la - memoria.

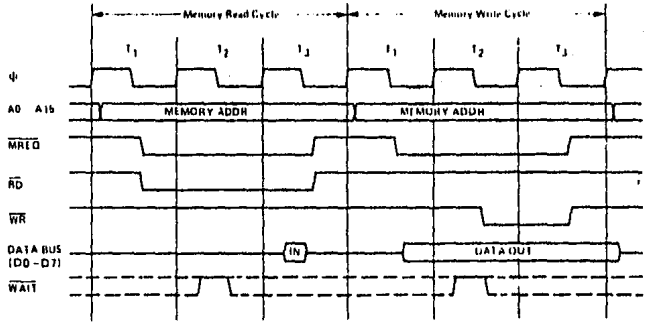


Fig. 49. Ciclo de Lectura o Escritura de Memoria.

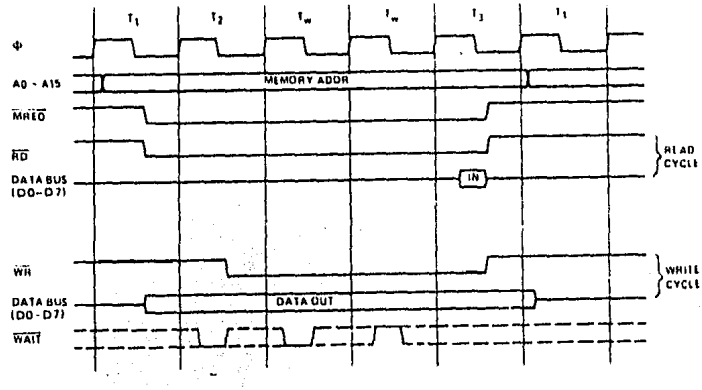


Fig. 50. Ciclo de Lectura o Escritura de Memoria con Estados de Espera.

IV.1.4. Ciclo de Lectura o Escritura de Puerto. Un Ciclo de Lectura o -- Escritura de Puerto permite emitir o recibir informacion de un - Puerto de Entrada/Salida. Durante este Ciclo se inserta automa-- ticamente un Estado de Espera.

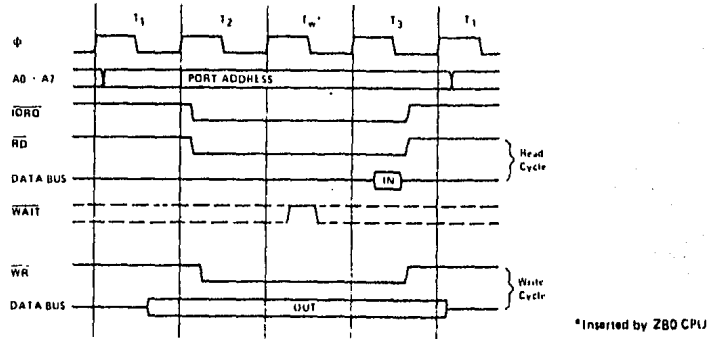


Fig. 51. Ciclo de Lectura o Escritura de Puerto.

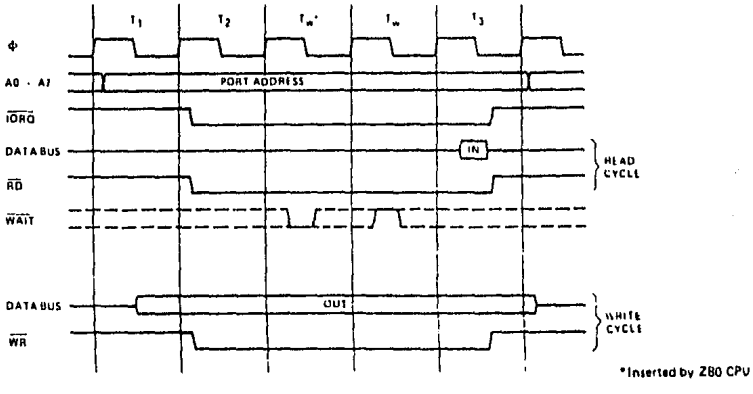


Fig. 52. Ciclo de Lectura o Escritura de Puerto con Estados de Espera.

IV.1.5. Ciclo de Solicitud/Liberacion de Bus. Este Ciclo muestra la forma en que el Microprocesador maneja una Solicitud de Bus a fin de permitir operaciones del tipo DMA (Acceso Directo a Memoria). Cuando el Microprocesador libera al Bus, el Bus de Direcciones, el Bus de Datos y las senales salientes de tercer estado del Bus de Control (/MREQ, /RD, /WR, /IORQ y /RFSH) presentan un estado de alta impedancia de tal manera que algun dispositivo externo pueda controlar al Bus para transferir datos entre la Memoria y los Puertos de Entrada/Salida. Durante la Liberacion del Bus, el Microprocesador no puede aceptar ningun tipo de interrupcion.

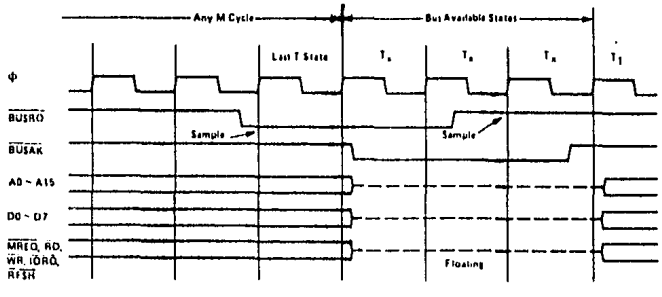


Fig. 53. Ciclo de Solicitud/Liberacion de Bus.

IV.1.6. Ciclo de Solicitud/Aceptacion de Interrupcion. El Microprocesador Z80 tiene 3 Modos de Interrupcion, siendo el Modo 0 con el que se inicializa al conectarle energia.
Cuando la senal /INT se activa y el Z80 esta en el Modo 0, el dispositivo que interrumpe puede colocar una instruccion en el Bus de Datos y el Z80 la ejecuta.
Si el Z80 esta en el Modo 1 cuando la senal /INT se activa, el contenido del Registro PC se guarda en el Stack y se ejecuta la subrutina que empieza en la direccion 0038 H.
Cuando el Z80 esta funcionando en el Modo 2 y se activa la senal /INT, el contenido del Registro PC se guarda en el Stack y se ejecuta la subrutina que empieza en la direccion que se forma con el contenido del Registro I para el byte mas significativo y un byte proporcionado por el dispositivo que interrumpe para el byte menos significativo de la direccion.
La senal /INT es aceptada por el Z80 siempre que el Flip Flop de Habilitacion de Interrupcion almacene un "1" y la senal /BUSRQ no este activa. Observese que se insertan automaticamente dos Estados de Espera.

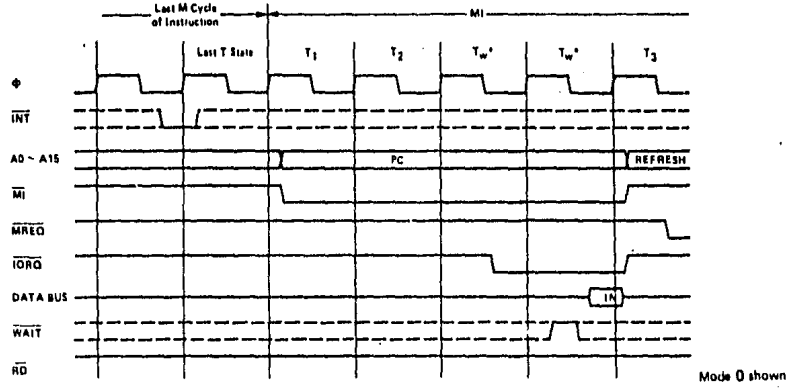


Fig. 54. Ciclo de Solicitud/Aceptacion de Interrupcion.

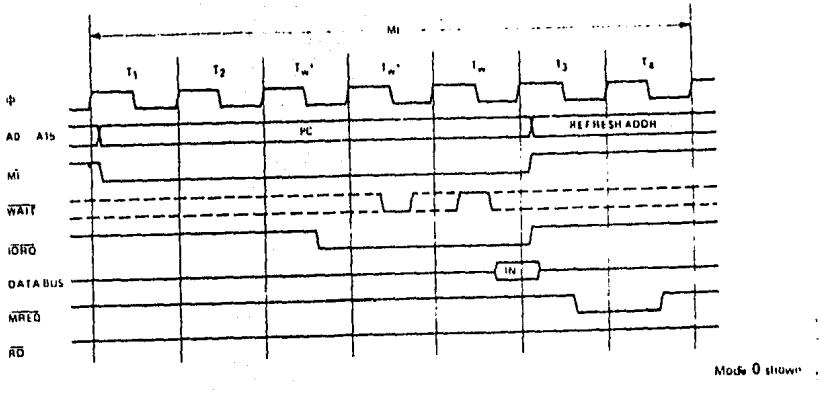


Fig. 55. Ciclo de Solicitud/Aceptacion de Interrupcion con Estados de Espera.

IV.1.7. Ciclo de Solicitud/Aceptacion de Interrupcion No Mascarable. Este Ciclo de Maquina se ejecuta cuando se activa la senal /NMI, la cual tiene mas prioridad que la senal /INT. Cuando se activa la senal /NMI, el contenido del Registro PC se guarda en el Stack y se ejecuta la subrutina que empieza en la direccion 0066 H.

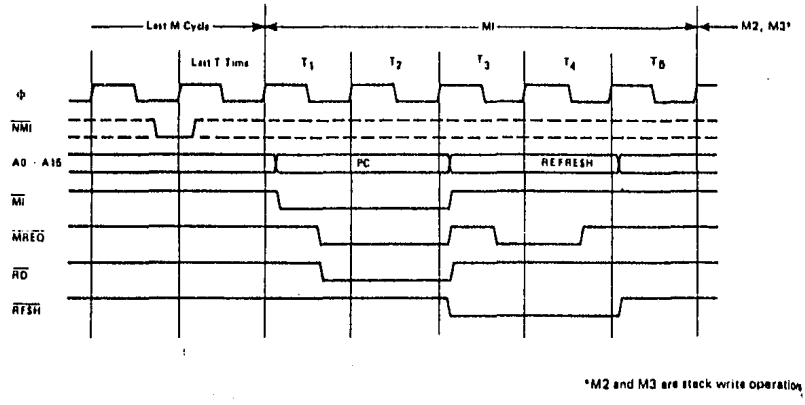


Fig. 56. Ciclo de Solicitud/Aceptacion de Interrupcion No Mascarable.

IV.1.8. Ciclo de Salida de la Instruccion HALT. Cuando el procesador recibe la instruccion HALT, ejecuta instrucciones NOP (No Operacion), hasta que recibe una interrupcion del tipo /NMI o del tipo /INT (si el Flip Flop de Habilitacion de Interrupcion almacena un "1"). El Ciclo de Maquina siguiente a la interrupcion, sera el Ciclo correspondiente a la interrupcion recibida, considerando que si se reciben ambas simultaneamente, /NMI tiene mayor prioridad.

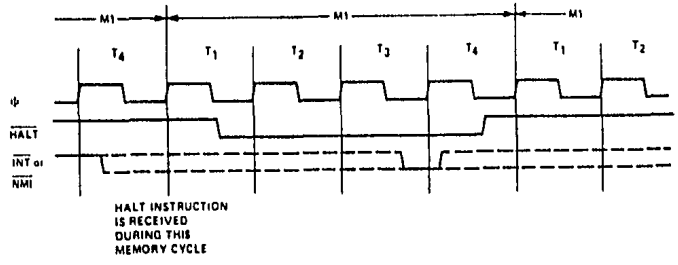


Fig. 57. Ciclo de Salida de la Instruccion HALT.

IV.I.9. Resumen de los Ciclos de Maquina. El Modulo del Emulador debe -- satisfacer los requerimientos indicados en cada Ciclo de Maquina a fin de emular correctamente al Microprocesador.
Enseguida se presenta un resumen de los Ciclos de Maquina, con-- siderando la siguiente simbologia:

0: Nivel Bajo.
1: Nivel Alto.
Z: Alta Impedancia.
TW: Inclusion de un Estado de Espera.
^Tw: Posible inclusion de Estados de Espera.
^Tx: Posible inclusion de Estados con Bus Libre.

1. Ciclo Fetch.

```
          T1....T4
CLOCK     10101010
             ^Tw

/HALT     11111111
/MREQ     10001001
/IORQ     11111111
/RD       10001111
/WR       11111111
/BUSAK    11111111
/M1       00001111
/RFSH     11110000
```

2. Ciclo de Lectura de Memoria.

```
          T1..T3
CLOCK     101010
             ^Tw

/HALT     111111
/MREQ     100001
/IORQ     111111
/RD       100001
/WR       111111
/BUSAK    111111
/M1       111111
/RFSH     111111
```


*

3. Ciclo de Escritura de Memoria.

```
         T1..T3
CLOCK    101010
            ^Tw

/HALT    111111
/MREQ    100001
/IORQ    111111
/RD      111111
/WR      111001
/BUSAK   111111
/M1      111111
/RFSH    111111
```

4. Ciclo de Lectura de Puerto.

```
         T1..TWT3
CLOCK    10101010
               ^Tw

/HALT    11111111
/MREQ    11111111
/IORQ    11000001
/RD      11000001
/WR      11111111
/BUSAK   11111111
/M1      11111111
/RFSH    11111111
```

5. Ciclo de Escritura de Puerto.

```
         T1..TWT3
CLOCK    10101010
               ^Tw

/HALT    11111111
/MREQ    11111111
/IORQ    11000001
/RD      11111111
/WR      11000001
/BUSAK   11111111
/M1      11111111
/RFSH    11111111
```

6. Ciclo de Solicitud/Liberacion de Bus.

```
         T1T2
CLOCK    1010
          ^Tx

/HALT    1111
/MREQ    ZZZZ
/IORQ    ZZZZ
/RD      ZZZZ
/WR      ZZZZ
/BUSAK   0001
/M1      1111
/RFSH    ZZZZ
```

7. Ciclo de Solicitud/Aceptacion de Interrupcion.

```
         T1..TWTW..T4
CLOCK    101010101010
               ^Tw

/HALT    111111111111
/MREQ    111111111001
/IORQ    111110001111
/RD      111111111111
/WR      111111111111
/BUSAK   111111111111
/M1      000000001111
/RFSH    111111110000
```

8. Ciclo de Solicitud/Aceptacion de Interrupcion
   No Mascarable.

```
         T1.....T5
CLOCK    1010101010

/HALT    1111111111
/MREQ    1000100111
/IORQ    1111111111
/RD      1000111111
/WR      1111111111
/BUSAK   1111111111
/M1      0000111111
/RFSH    1111000011
```

9. Ciclo de Salida de la Instruccion HALT.

```
          v--------,
        T1....T4  T1     T4
CLOCK   10101010  10     10
             '-interrup-^

/HALT   10000000  00     01
/MREQ   10001001  10     01
/IORQ   11111111  11     11
/RD     10001111  10     11
/WR     11111111  11     11
/BUSAK  11111111  11     11
/M1     00001111  00     11
/RFSH   11110000  11     00
```

Si se considera que /HALT corresponde al bit mas significativo y /RFSH al menos significativo, se puede presentar el Resumen de - los Ciclos de Maquina de la siguiente forma:

CLOCK	Ciclos de Maquina								
1/2 Periodo	1.	2.	3.	4.	5.	6.	7.	8.	9.
0	FD	FF	FF	FF	FF	FB	FD	FD	FD
1	AD	AF	BF	FF	FF	FB	FD	AD	2D
2	AD	AF	BF	CF	D7	FB	FD	AD	2D
3	AD	AF	B7	CF	D7	FF	FD	AD	2D
4	FE	AF	B7	CF	D7		FD	FE	7E
5	BE	FF	FF	CF	D7		DD	BE	3E
6	BE			CF	D7		DD	BE	3E
7	FE			FF	FF		DD	FE	7E
8							FE	FF	7D
9							BE	FF	2D
A							BE		3E
B							FE		FE
C									
D									
E									
F									

Las senales salientes del Bus de Control se han definido en los parrafos anteriores; las 6 senales entrantes del Bus de Control se han considerado de la siguiente manera:

CLOCK : Esta senal puede proporcionarse por el Emulador, el cual la toma del Bus EDH, o puede proporcionarse por el Modulo Bajo Prueba.

/INT : El nivel de esta senal se verifica por el Microprocesador con el flanco ascendente del ultimo Estado de cualquier Ciclo de Maquina. Si la senal esta activa y /BUSRQ e /NMI no lo estan, el siguiente Ciclo de Maquina debe ser el Ciclo de Solicitud/Aceptacion de Interrupcion.
Dado que se pretende disenar un Emulador totalmente programable a nivel de Ciclos de Maquina, la secuencia de tales Ciclos debe manejarse bajo software y no bajo hardware, de manera que los Ciclos de Maquina sean independientes entre si.

/NMI : Esta senal se maneja de manera similar a /INT.

/WAIT : Cuando en un Ciclo de Maquina el Microprocesador verifica el nivel de esta senal y la encuentra activa, inserta Estados de Espera Tw. El momento de la verificacion se ha indicado en cada Ciclo de Maquina con el simbolo ^Tw.

/BUSRQ : Esta senal se maneja de manera similar a /INT y /NMI, con una diferencia.
La diferencia es que el Microprocesador debe estar verificando periodicamente esta senal en el punto indicado ^Tx. Cuando el Modulo que ha solicitado al Bus y al cual se ha entregado su control, ya no lo requiere, desactiva la senal /BUSRQ. Al detectar esto, el Microprocesador termina el Ciclo de Solicitud/Liberacion de Bus, recuperando el control del Bus.

/RESET : Esta senal obliga que el Registro PC del Microprocesador almacene el valor 0000 H, e inicia un Ciclo Fetch, presentando en la Lineas de Direccion A15-A0, la direccion almacenada en PC.
Como se ha indicado anteriormente, si se pretende disenar un Emulador programable a nivel de Ciclos de Maquina, se debe mantener una independencia funcional entre los mismos, por lo que el efecto que produce esta senal, debe manejarse por programa.

En cuanto a las Lineas de Direccion A15-A0 del Microprocesador, se deduce de los Ciclos de Maquina, que el Emulador debe satisfacer tres casos, controlando bajo programa la informacion que se requiera en las Lineas de Direccion:

a. En las Lineas A15-A0 se presenta una Direccion de Memoria o de Puerto que se activa con el flanco ascendente del Estado T1.
b. En las Lineas A6-A0 se presenta una Direccion para Refrescamiento de las Memorias Dinamicas del Sistema, que se activa con la senal /RFSH y que corresponde al contenido del Registro R del Microprocesador, el cual se incrementa en +1 en cada operacion /RFSH; el bit A7 presenta un valor logico 0 y en A15-A8 se presenta el contenido del Registro I.
c. Las Lineas A15-A0 se van a un Estado de Alta Impedancia con la senal /BUSAK.

Las Lineas de Datos del Microprocesador son el medio que permite realizar transferencias de informacion entre los Registros del Microprocesador y los componentes externos. El Emulador debe poder emitir y recibir datos bajo programa con los mismos requerimientos de tiempo del Z80.

Las transferencias de informacion de los componentes externos al Microprocesador, se realizan en los siguientes Ciclos:
Con el flanco ascendente del Estado T3:
a. Ciclo Fetch.
b. Ciclo de Solicitud/Aceptacion de Interrupcion.
Con el flanco descendente del Estado T3:
c. Ciclo de Lectura de Memoria.
d. Ciclo de Lectura de Puerto.

Las transferencias de informacion del Microprocesador a los componentes externos se realizan presentando la informacion en las Lineas de Datos D7-D0 entre el flanco descendente del Estado T1 y el flanco ascendente del Estado T4 en los siguientes Ciclos:

a. Ciclo de Escritura de Memoria.
b. Ciclo de Escritura de Puerto.

Ademas de lo anterior, las Lineas de Datos deben presentar un Estado de Alta Impedancia cuando se activa la senal /BUSAK.

Las Lineas de Energia del Bus Emulador +8V y 0V difieren de las del Microprocesador Z80, +5V y 0V, para mantener la compatibilidad con el Bus EDH; sin embargo, en el Modulo del Procesador, existe un Conmutador para proporcionar, opcionalmente, la senal +5V al Modulo Bajo Prueba, a traves de cable.

IV.1.10. Operacion del Emulador. La operacion del Emulador esta enfocada a permitir al usuario concentrarse en la emulacion propiamente, y no en la operacion del Emulador. Para realizar una emulacion con el Modulo del Emulador, se siguen los siguientes pasos:

a. Conexion. Se conecta el Modulo del Emulador al Sistema EDH y el Modulo Bajo Prueba al Modulo del Procesador.
b. Datos. Se proporcionan al Sistema EDH los datos necesarios en la emulacion.
c. Ejecucion. Se ordena que se inicie la emulacion.
d. Analisis. Se analiza en tiempo real el comportamiento del Modulo Bajo Prueba.
e. Fin. Se detiene la emulacion y se recuperan los resultados.

El Paso a se realiza de acuerdo a la Fig. 36, tomando en cuenta que las 40 terminales del cable del Modulo del Procesador se encuentran en la misma posicion que en el Microprocesador Z80. La eleccion de la senal +5V se realiza en el Modulo del Procesador, mientras que la eleccion de la senal CLOCK se realiza en el Modulo del Emulador:

Conmutador CLOCK	Origen de la senal CLOCK
0	Modulo Bajo Prueba
1	Bus EDH

El Paso b, requiere preparar la siguiente informacion:

b.1. Secuencia de Ciclos de Maquina a emular.
b.2. Direcciones de Memoria o de Puerto involucradas en los Ciclos de Maquina.
b.3. Datos que se deben emitir en la emulacion en determinados Ciclos de Maquina.
b.4. Cantidad de Secuencias a emular.
b.5. Contenido inicial del Registro para Refrescamiento de las Memorias Dinamicas (R).
b.6. Contenido del Registro del Vector de Interrupcion (I).

El Emulador esta disenado para soportar una Secuencia de hasta 16 Ciclos de Maquina, lo que le permite emular cualquier instruccion del Microprocesador Z80. Para proporcionar la informacion anterior, se llena una tabla que se presenta en la pagina siguiente, empezando en el Ciclo de Maquina 0 y utilizando solo los Ciclos de Maquina necesarios. Los Codigos de los Ciclos de Maquina que se usan en el Emulador son:

Ciclo de Maquina	Codigo
Fetch	01 H
Lect. de Memoria	02 H
Escr. de Memoria	03 H
Lect. de Puerto	04 H
Escr. de Puerto	05 H
Sol/Lib de Bus	06 H
Sol/Acep de INT	07 H
Sol/Acep de NMI	08 H
Salida de HALT	09 H

SECUENCIA DE CICLOS DE MAQUINA A EMULAR

No	Dir Mem	CICLO Nombre	CICLO Codigo	Dir Mem	A15-A8	Dir Mem	A7-A0	Dir Mem	D7-D0 OUT
0	2200			2210		2220		2230	
1	2201			2211		2221		2231	
2	2202			2212		2222		2232	
3	2203			2213		2223		2233	
4	2204			2214		2224		2234	
5	2205			2215		2225		2235	
6	2206			2216		2226		2236	
7	2207			2217		2227		2237	
8	2208			2218		2228		2238	
9	2209			2219		2229		2239	
A	220A			221A		222A		223A	
B	220B			221B		222B		223B	
C	220C			221C		222C		223C	
D	220D			221D		222D		223D	
E	220E			221E		222E		223E	
F	220F			221F		222F		223F	

El ultimo Ciclo de Maquina a emular, dentro de la Secuencia, se debe indicar en la Direccion de Memoria 2250 H, considerando que los Ciclos de Maquina estan numerados del 0 al F:

Dir Mem	Ultimo Ciclo a Emular
2250 H	00 <= Ult Ciclo <= 0F

Si la Secuencia de Ciclos de Maquina se debe repetir indefinidamente o solo se debe ejecutar una sola vez, se debe indicar en la Direccion de Memoria 2251 H, de la siguiente manera:

Dir Mem	Repeticion de la Secuencia ?
2251 H	00 H = No, 01 H = Si.

Finalmente, se debe indicar el contenido inicial del Registro R en la Direccion 2252 H, y el contenido del Registro I, en la Direccion de Memoria 2253 H:

Dir Mem	Registro
2252 H	00 <= R <= 7F
2253 H	00 <= I <= FF

Todos los datos anteriores se deben introducir al Sistema EDH en las Direcciones de Memoria indicadas.

El Paso c se realiza usando el Comando Programable F1. Al oprimir la tecla correspondiente se ejecuta un programa que transfiere los datos que se han indicado en la Memoria del Sistema EDH al Emulador, e inicia la emulacion encendiendo el led E.

El Paso d permite al usuario analizar en tiempo real el comportamiento del Modulo Bajo Prueba. Debe hacerse notar que el Sistema EDH puede ejecutar otros programas mientras se realiza el analisis, ya que el control de la emulacion lo realiza totalmente el Modulo del Emulador.

El Paso e detiene la emulacion y recupera los datos que debio haber recibido el Emulador en cada Ciclo de Maquina a traves de las Lineas D7-D0. Este Paso se realiza por medio del Comando Programable F2. La informacion que recupera, la coloca en las siguientes Direcciones de Memoria para su analisis con el Comando M del Monitor:

Ciclo No	Dir Mem (D7-D0 IN)
0	2240
1	2241
...	....
F	224F

Cuando se detiene la emulacion, ya sea porque se hizo uso del Comando F2 o porque la secuencia no fue repetitiva, se apaga el led E.

V.1.11. Construccion. Considerando los requerimientos anteriores, se ha disenado el Modulo del Emulador. Primero se presenta un Diagrama a Bloques del Modulo y posteriormente se muestra el contenido de cada Bloque.
El Modulo del Emulador se ha disenado sobre 2 Tarjetas, A y B, las cuales se comunican a traves del Bus Libre.

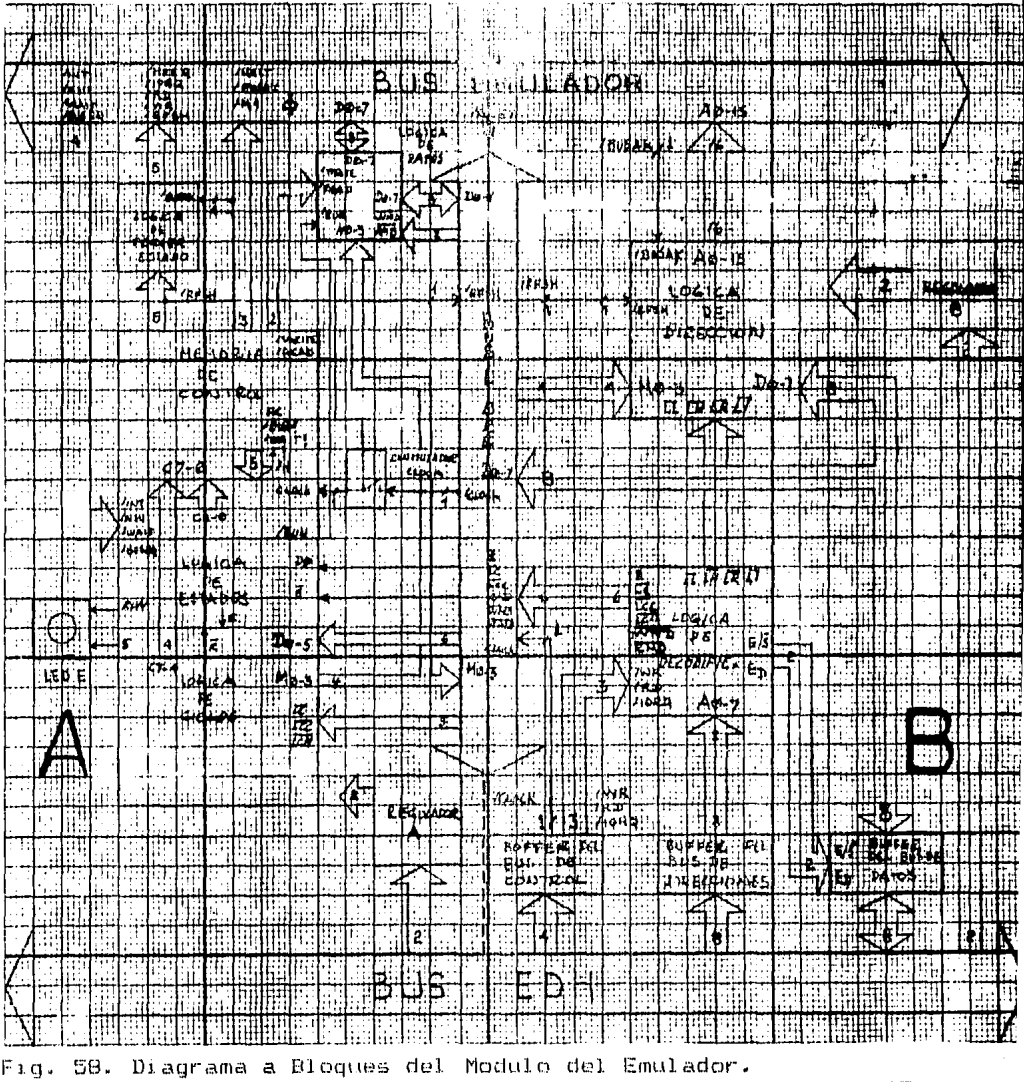


Fig. 58. Diagrama a Bloques del Modulo del Emulador.

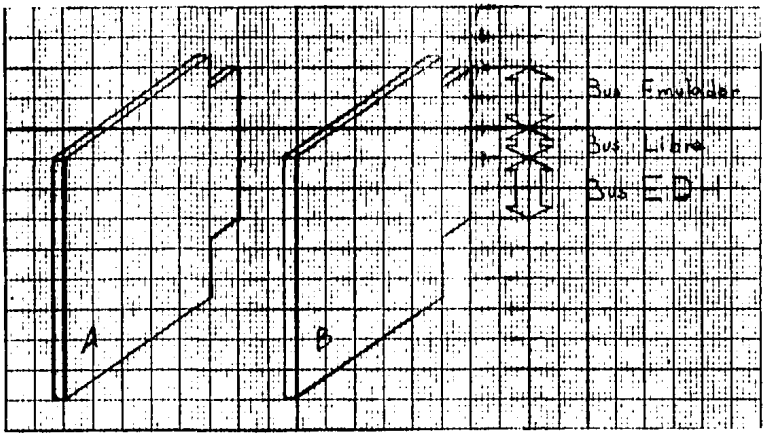


Fig. 59. Tarjetas A y B del Modulo del Emulador.

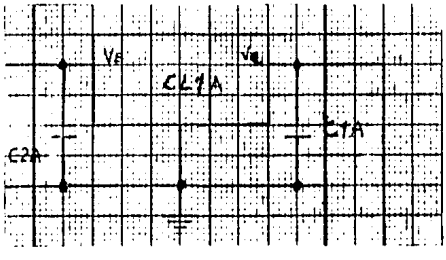


Fig. 60. Bloque del Regulador A.

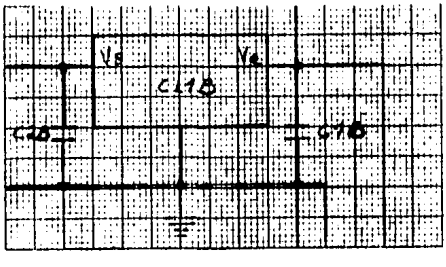


Fig. 61. Bloque del Regulador B.

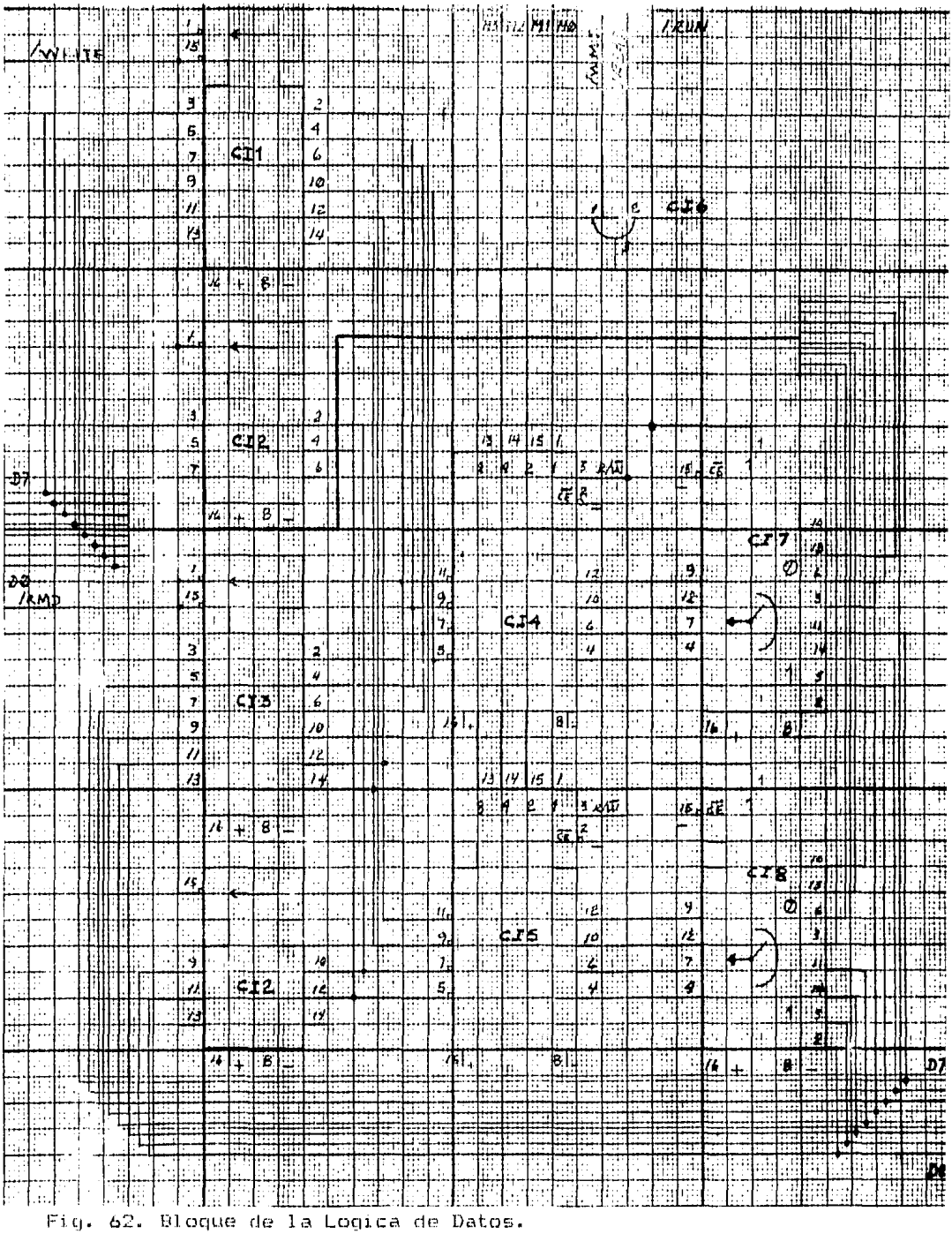


Fig. 62. Bloque de la Logica de Datos.

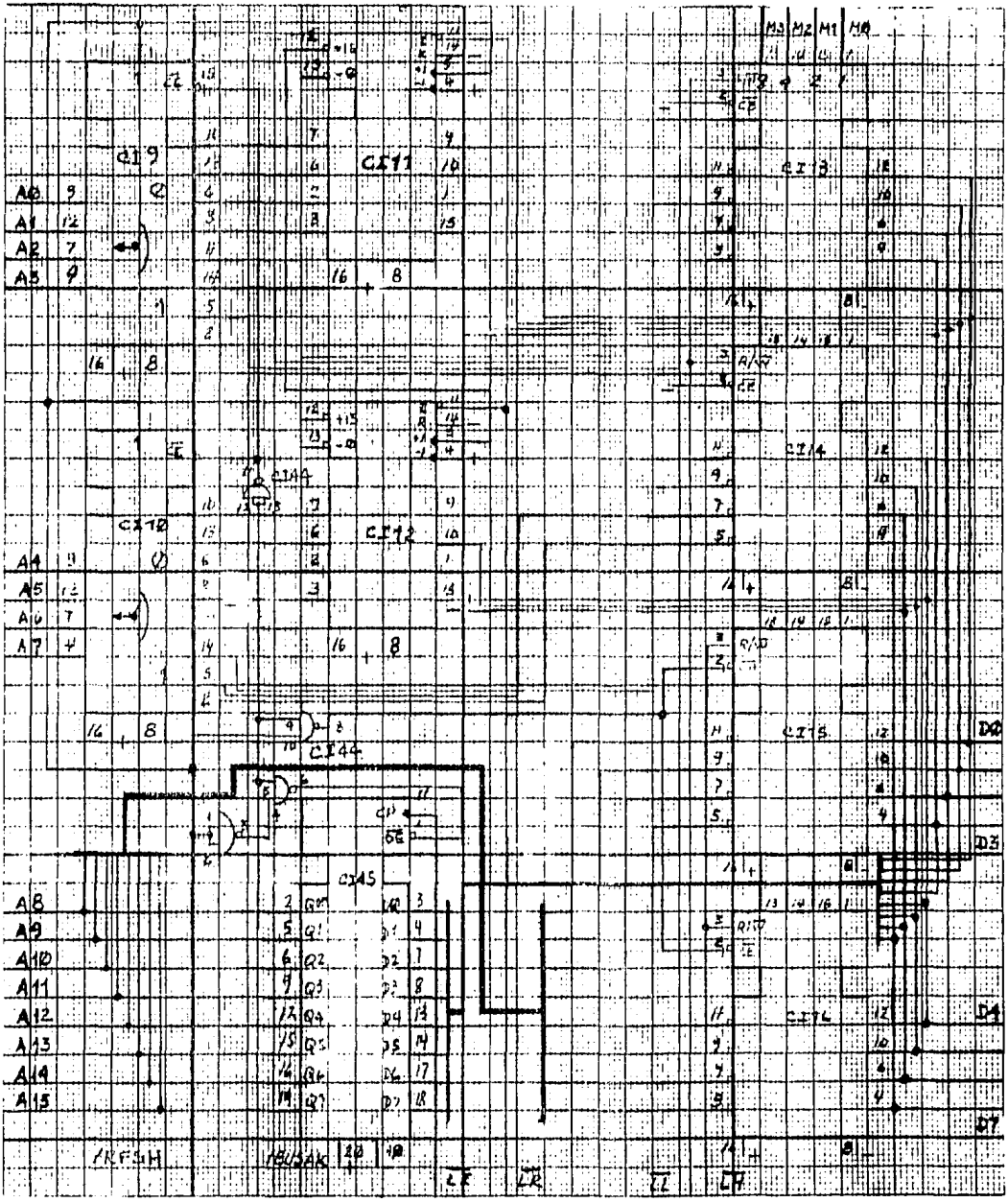


Fig. 63. Bloque de la Logica de Direccion.

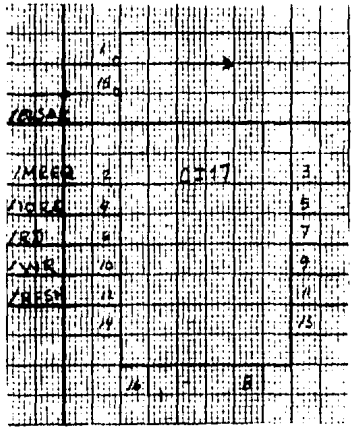


Fig. 64. Bloque de la Logica de Tercer Estado.

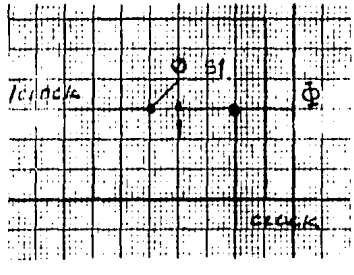


Fig. 65. Bloque del Conmutador CLOCK.

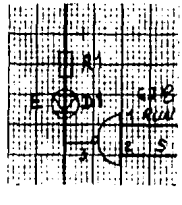


Fig. 66. Bloque del Led E.

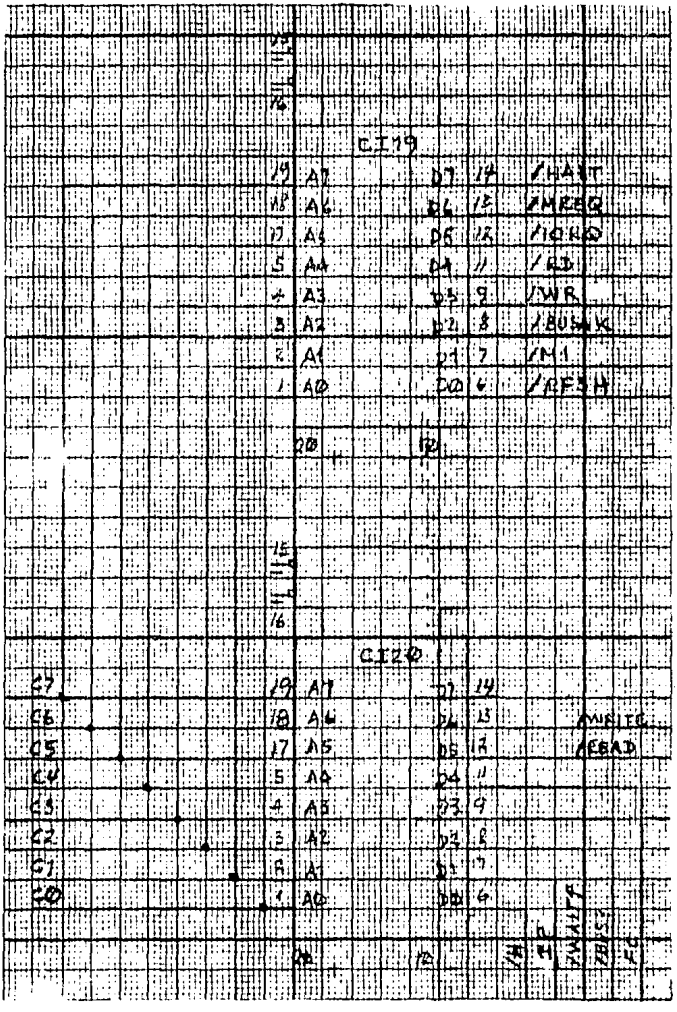


Fig. 67. Bloque de la Memoria de Control.

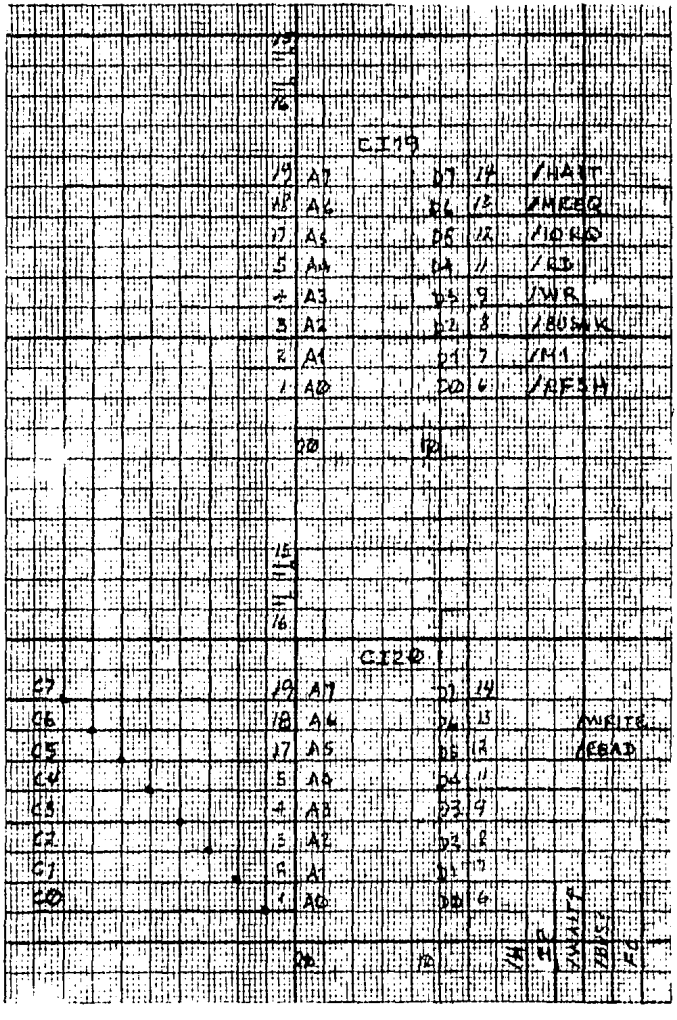


Fig. 67. Bloque de la Memoria de Control.

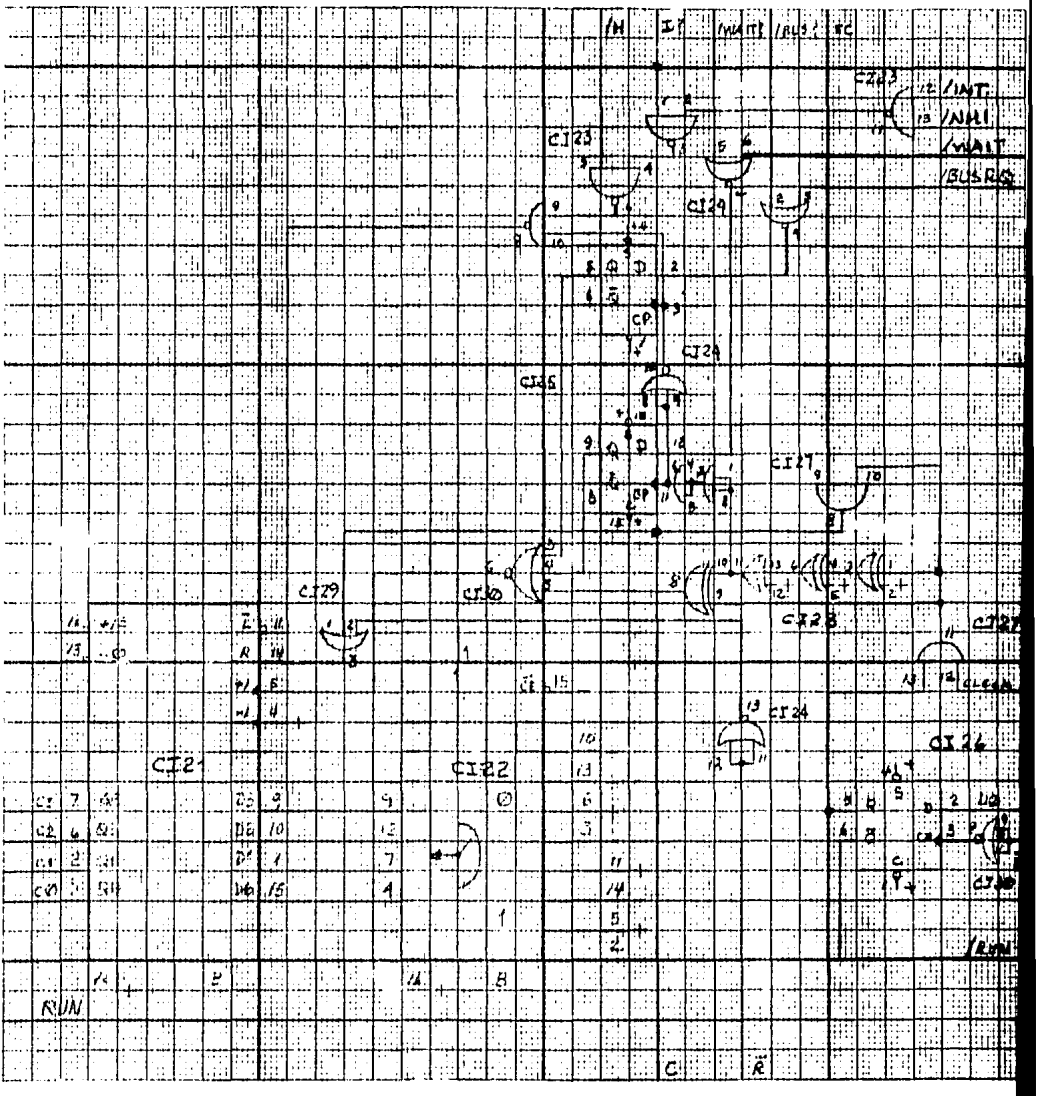


Fig. 6B. Bloque de la Logica de Estados.

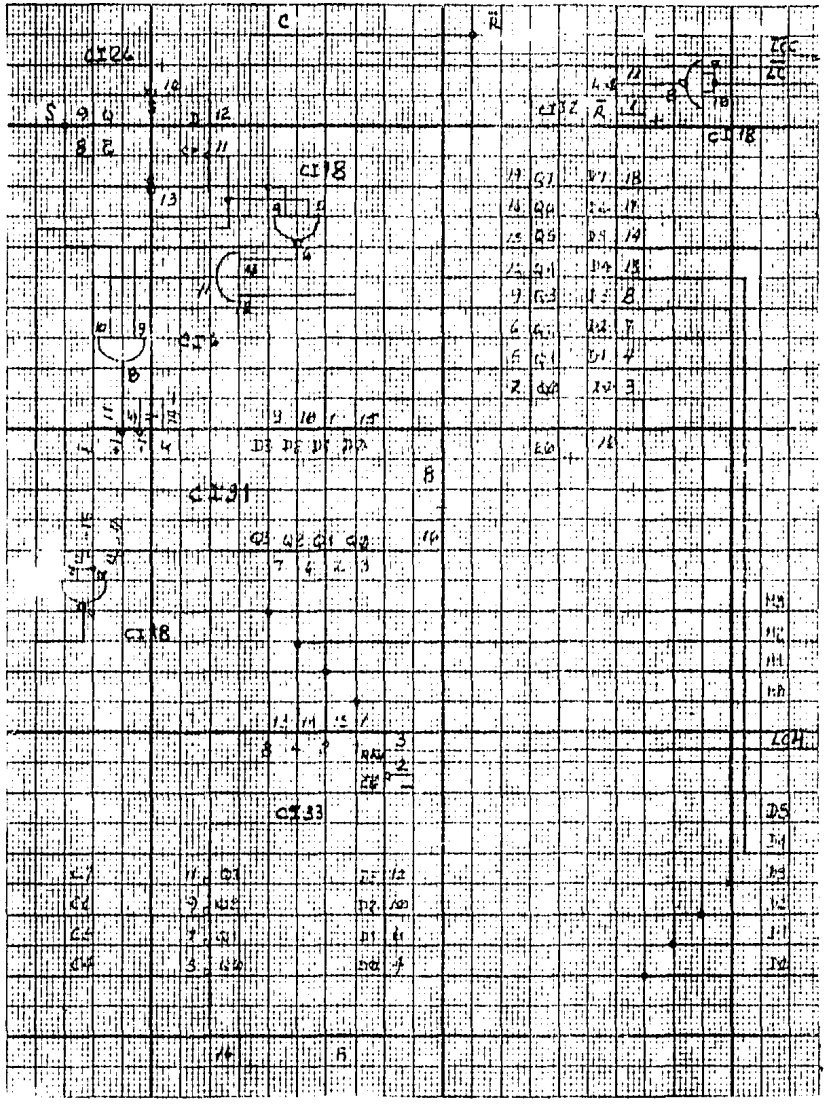


Fig. 69. Bloque de la Logica de Ciclos.

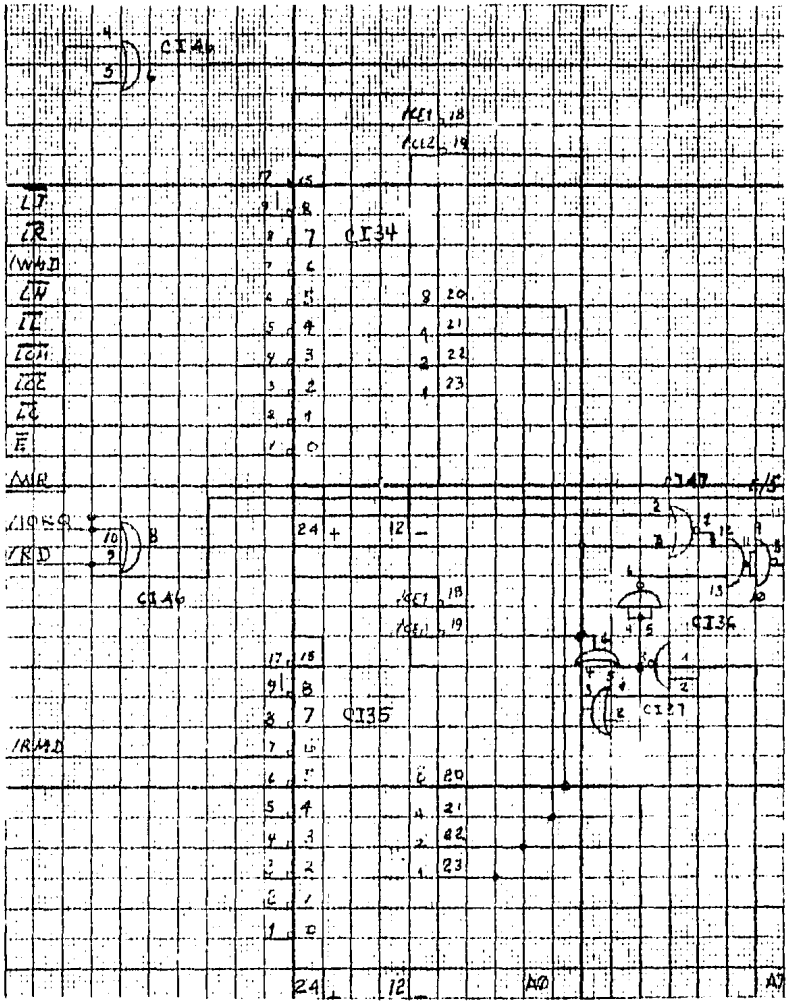


Fig. 70. Bloque de la Logica de Decodificacion.

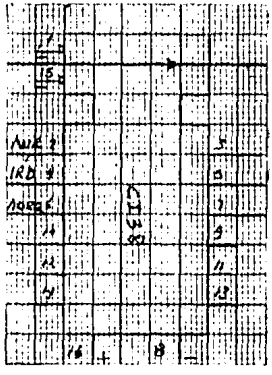


Fig. 71. Bloque del Buffer del Bus de Control.

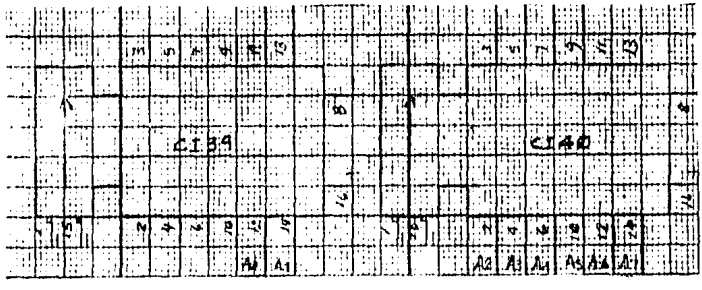


Fig. 72. Bloque del Buffer del Bus de Direcciones.

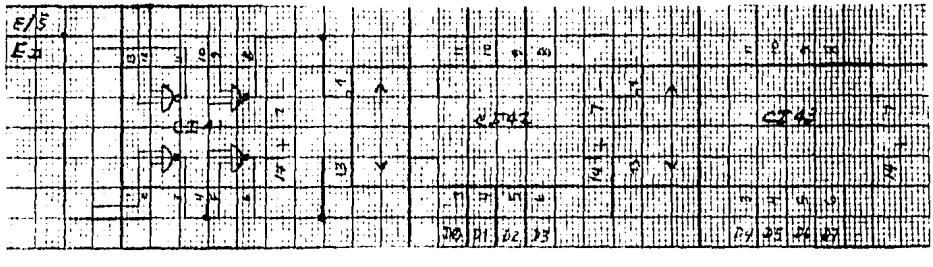


Fig. 73. Bloque del Buffer del Bus de Datos.

IV.1.12. Memoria de Control. El Bloque de la Memoria de Control consta de 2 Memorias PROM de 256 palabras con 8 bits por palabra cada una. Una de las Memorias se usa para generar las Lineas de Control que debe enviar el Emulador al Modulo Bajo Prueba, la otra Memoria se utiliza para generar Senales Auxiliares de Control para uso interno del Emulador.

Las Senales Auxiliares de Control son:

/WRITE	:	Indica el intervalo en que el Emulador envia informacion por las Lineas de Datos (D0-D7).
/READ	:	Indica el momento en que el Emulador recibe informacion por las Lineas de Datos (D0-D7).
FC	:	Indica el fin del Ciclo de Maquina.
/BUS?	:	Prueba si la senal /BUSRQ esta inactiva.
/WAIT?	:	Prueba si la senal /WAIT esta inactiva.
I?	:	Prueba si la senal /INT y/o la senal /NMI estan activas.
/H	:	Indica se continue en el Estado HALT.

Las Senales Auxiliares de Control deben estar activas segun se indica enseguida para cada Ciclo de Maquina, considerando la simbologia definida en IV.1.9.

1. Ciclo Fetch.

```
        T1....T4
CLOCK   10101010
           ^Tw

/WRITE  11111111
/READ   11101111
FC      00000001
/BUS?   11111111
/WAIT?  11001111
I?      00000000
/H      11111111
```

2. Ciclo de Lectura de Memoria.

```
        T1..T3
CLOCK   101010
           ^Tw

/WRITE  111111
/READ   111101
FC      000001
/BUS?   111111
/WAIT?  110011
I?      000000
/H      111111
```

3. Ciclo de Escritura de Memoria.

```
        T1..T3
CLOCK   101010
           ^Tw

/WRITE  100000
/READ   111111
FC      000001
/BUS?   111111
/WAIT?  110011
I?      000000
/H      111111
```

4. Ciclo de Lectura de Puerto.

```
        T1..TWT3
CLOCK   10101010
              ^Tw

/WRITE  11111111
/READ   11111101
FC      00000001
/BUS?   11111111
/WAIT?  11110011
I?      00000000
/H      11111111
```

5. Ciclo de Escritura de Puerto.

```
        T1..TWT3
CLOCK   10101010
             ^Tw

/WRITE  10000000
/READ   11111111
FC      00000001
/BUS?   11111111
/WAIT?  11110011
I?      00000000
/H      11111111
```

6. Ciclo de Solicitud/Liberacion de Bus.

```
        T1T2
CLOCK   1010
         ^Tx

/WRITE  1111
/READ   1111
FC      0001
/BUS?   1001
/WAIT?  1111
I?      0000
/H      1111
```

7. Ciclo de Solicitud/Aceptacion de Interrupcion.

```
        T1..TWTW..T4
CLOCK   101010101010
                ^Tw

/WRITE  111111111111
/READ   111111101111
FC      000000000001
/BUS?   111111111111
/WAIT?  111111001111
I?      000000000000
/H      111111111111
```

8. Ciclo de Solicitud/Aceptacion de Interrupcion
   No Mascarable.

```
        T1......T5
CLOCK   1010101010

/WRITE  1111111111
/READ   1111111111
FC      0000000001
/BUS?   1111111111
/WAIT?  1111111111
I?      0000000000
/H      1111111111
```

9. Ciclo de Salida de la Instruccion HALT.

```
           v--------,
         T1....T4   T1     T4
CLOCK    10101010   10     10
                 '-interrup-^

/WRITE   11111111   11     11
/READ    11111111   11     11
FC       00000000   00     01
/BUS?    11111111   11     11
/WAIT?   11111111   11     11
I?       00000110   00     00
/H       11111111   10     01
```

Si se considera que /WRITE corresponde al bit mas significativo y /H al menos significativo, se puede presentar un Resumen de las Senales Auxiliares de Control de la siguiente manera:

CLOCK	Ciclos de Maquina								
1/2 Periodo	1.	2.	3.	4.	5.	6.	7.	8.	9.
0	6D	6D	6D	6D	6D	6D	6D	6D	6D
1	6D	6D	2D	6D	2D	65	6D	6D	6D
2	69	69	29	6D	2D	65	6D	6D	6D
3	49	69	29	6D	2D	7D	6D	6D	6D
4	6D	4D	2D	69	29		6D	6D	6D
5	6D	7D	3D	69	29		6D	6D	6F
6	6D			4D	2D		69	6D	6F
7	7D			7D	3D		49	6D	6D
8							6D	6D	6D
9							6D	7D	6C
A							6D		6C
B							7D		7D

El contenido de la Memoria de las Senales Auxiliares de Control y de la Memoria de las Lineas de Control, se deduce de sus respectivos Resumenes.

Memoria de Senales Auxiliares de Control:

Direccion	Contenido
10 a 17 H	6D 6D 69 49 6D 6D 6D 7D
20 a 25 H	6D 6D 69 69 4D 7D
30 a 35 H	6D 2D 29 29 2D 3D
40 a 47 H	6D 6D 6D 6D 69 69 4D 7D
50 a 57 H	6D 2D 2D 2D 29 29 2D 3D
60 a 63 H	6D 65 65 7D
70 a 7B H	6D 6D 6D 6D 6D 6D 69 49 6D 6D 6D 7D
80 a 89 H	6D 6D 6D 6D 6D 6D 6D 6D 6D 7D
90 a 9B H	6D 6D 6D 6D 6D 6F 6F 6D 6D 6C 6C 7D

Memoria de Lineas de Control:

Direccion	Contenido
10 a 17 H	FD AD AD AD FE BE BE FE
20 a 25 H	FF AF AF AF AF FF
30 a 35 H	FF BF BF B7 B7 FF
40 a 47 H	FF FF CF CF CF CF CF FF
50 a 57 H	FF FF D7 D7 D7 D7 D7 FF
60 a 63 H	FB FB FB FF
70 a 7B H	FD FD FD FD FD DD DD DD FE BE BE FE
80 a 89 H	FD AD AD AD FE BE BE FE FF FF
90 a 9B H	FD 2D 2D 2D 7E 3E 3E 7E 7D 2D 3E FE

El contenido de las direcciones no indicadas en ambas memorias no interesa.

IV.1.13. Programa de Operacion. Cuando el usuario oprime la tecla F1, se inicia la ejecucion del programa COMF1 que traslada la informacion que se ha introducido en la Memoria, al Emulador. Al terminar de introducir la informacion y ordenar el inicio de la emulacion envia el siguiente mensaje informativo al usuario:

E _ _ r uc C

donde: r=0 indica secuencia no-repetitiva,
r=1 indica secuencia repetitiva,
0<=uc<=F indica el numero del ultimo ciclo a emular.

Al oprimir la tecla F2, se ejecuta el programa COMF2 el cual detiene la emulacion y recupera la informacion que durante la emulacion ha recibido el emulador. Esta informacion la traslada a las Direcciones de Memoria 2240 a 224F H, en correspondencia con los Ciclos de Maquina, a fin de que pueda ser analizada por medio del Comando M del Monitor. Al terminar de ejecutarse el programa envia el siguiente mensaje al usuario:

F _ _ _ _ _

El Programa de Operacion para el Modulo del Emulador se ha almacenado en las siguientes direcciones de memoria:

Programa	Direcciones
COMF1	0500 H a 0591 H
COMF2	05A0 H a 05EB H

El Monitor del Sistema EDH fue adaptado del Monitor del Sistema Z80 Starter Kit, el cual se ha incluido en el apendice en su forma original junto con la version modificada en codigo hexadecimal para el Monitor del Sistema EDH.
Para incluir los programas COMF1 y COMF2 fue necesario eliminar ciertas facilidades en el Monitor original relacionadas con las operaciones de lectura y escritura de cassettes y con la programacion de memorias EPROM. Esto se logro reprogramando las direcciones de memoria 04CA H a 0633 H. Las direcciones de memoria que no usan los programas COMF1 y COMF2, son instrucciones HALT.

Para que los Comandos Programables F1 y F2 iniciaran programas en las direcciones arriba indicadas, fue necesario reprogramar las direcciones 022D H y 022A H segun se observa en el Programa de la pagina siguiente. El Comando Programable F3, continua iniciando la ejecucion de un programa que empiece en la direccion 1800 H.

*

```
                        ;*****************************************************
                        ;
                        ;       MODULO      DEL      EMULADOR
                        ;
                        ;       programa    de     operacion
                        ;
                        ;*****************************************************
0500 =          CONF1   EQU     0500H           ; direcciones de programas
05A0 =          CONF2   EQU     05A0H
1800 =          CONF3   EQU     1800H

00A0 =          PE      EQU     0A0H            ; direcciones de puertos
00A1 =          PLC     EQU     0A1H            ; de salida
00A2 =          PLCC    EQU     0A2H
00A3 =          PLCM    EQU     0A3H
00A4 =          PLL     EQU     0A4H
00A5 =          PLH     EQU     0A5H
00A6 =          PWMD    EQU     0A6H
00A7 =          PLR     EQU     0A7H
00A8 =          PLI     EQU     0A8H

00A6 =          PRMD    EQU     0A6H            ; direcciones de puertos
                                                ; de entrada

00F4 =          DISUP   EQU     00F4H           ; direcciones de memoria
2200 =          DIRCIC  EQU     2200H
2240 =          DATOIN  EQU     2240H
2250 =          ULTCIC  EQU     2250H
2251 =          SECREP  EQU     2251H
2252 =          REGR    EQU     2252H
2253 =          REGI    EQU     2253H
23F7 =          DISMEM  EQU     23F7H
23F8 =          DSMEM1  EQU     23F8H
23F9 =          DSMEM2  EQU     23F9H
23FA =          DSMEM3  EQU     23FAH
23FB =          DSMEM4  EQU     23FBH
23FC =          DSMEM5  EQU     23FCH

0227            EDH:    ORG     0227H           ; direcciones de programas para
0227 C30018             JP      CONF3           ; F3, F2 y F1
022A C3A005             JP      CONF2
022D C30005             JP      CONF1

0500                    ORG     CONF1           ; programa para F1

0500 3E00       CONF1:  LD      A,00H           ; detiene clock
0502 D3A0               OUT     PE,A

0504 3A5022             LD      A,(ULTCIC)      ; calculo del numero del primer
0507 E60F               AND     A,0FH           ; ciclo
0509 32FB23             LD      (DSMEM4),A
050C 47                 LD      B,A
050D 3E0F               LD      A,0FH
050F 90                 SUB     A,B
0510 47                 LD      B,A
0511 4F                 LD      C,A

0512 110022             LD      DE,DIRCIC       ; inicializacion

0515 D3A1       CONF1A: OUT     PLC,A           ; numero del ciclo
0517 D3A2               OUT     PLCC,A

0519 1A                 LD      A,(DE)          ; codigo
051A 2F                 CPL
051B D3A3               OUT     PLCM,A

051D 3E10               LD      A,10H           ; direccion baja
051F 83                 ADD     A,E
0520 5F                 LD      E,A
0521 3E00               LD      A,00H
0523 8A                 ADC     A,D
0524 57                 LD      D,A
0525 1A                 LD      A,(DE)
0526 2F                 CPL
0527 D3A4               OUT     PLL,A
```

```
0529 3E10          LD    A,10H          ; direccion alta
052B 83            ADD   A,E
052C 5F            LD    E,A
052D 3E00          LD    A,00H
052F 8A            ADC   A,D
0530 57            LD    D,A
0531 1A            LD    A,(DE)
0532 2F            CPL
0533 D3A5          OUT   PLH,A

0535 3E10          LD    A,10H          ; dato a emitir
0537 83            ADD   A,E
0538 5F            LD    E,A
0539 3E00          LD    A,00H
053B 8A            ADC   A,D
053C 57            LD    D,A
053D 1A            LD    A,(DE)
053E 2F            CPL
053F D3A6          OUT   PWMD,A

0541 7B            LD    A,E            ; control de ciclos
0542 1E30          LD    E,30H
0544 93            SUB   A,E
0545 5F            LD    E,A
0546 7A            LD    A,D
0547 1600          LD    D,00H
0549 9A            SBC   A,D
054A 57            LD    D,A
054B 13            INC   DE
054C 04            INC   B
054D 78            LD    A,B
054E FE0F          CP    A,0FH
0550 FA1505        JP    M,COMF1A

0553 3E10          LD    A,10H          ; reset
0555 D3A1          OUT   PLC,A

0557 3A5122        LD    A,(SECREP)     ; secuencia repetitiva ?
055A E601          AND   A,01H
055C 32FA23        LD    (DSMEM3),A
055F 07            RLCA
0560 07            RLCA
0561 07            RLCA
0562 07            RLCA
0563 F6EF          OR    A,0EFH

0565 47            LD    B,A            ; indicacion de:
0566 79            LD    A,C            ; secuencia repetitiva ? y
0567 F6F0          OR    A,F0H          ; numero del primer ciclo
0569 A0            AND   A,B
056A 00            NOP
056B D3A1          OUT   PLC,A
056D D3A2          OUT   PLCC,A

056F 3A5222        LD    A,(REGR)       ; direccion inicial de
0572 D3A7          OUT   PLR,A          ; refrescamiento

0574 3A5322        LD    A,(REGI)       ; contenido del registro i
0577 D3A8          OUT   PLI,A

0579 3E01          LD    A,01H          ; inicio de la emulacion
057B D3A0          OUT   PE,A

057D 3E0E          LD    A,0EH          ; mensaje final
057F 32F723        LD    (DISMEM),A
0582 3E10          LD    A,10H
0584 32F823        LD    (DSMEM1),A
0587 32F923        LD    (DSMEM2),A
058A 3E0C          LD    A,0CH
058C 32FC23        LD    (DSMEM5),A

058F C3F400        JP    DISUP          ; fin
```

```
05A0                    ORG     CONF2           ; programa para F2

05A0 3E00       CONF2:  LD      A,00H           ; detiene clock
05A2 D3A0               OUT     PE,A

05A4 114022             LD      DE,DATOIN       ; borrado de memoria
05A7 0600               LD      B,00H
05A9 3EFF       CONF2A: LD      A,0FFH
05AB 12                 LD      (DE),A
05AC 13                 INC     DE
05AD 04                 INC     B
05AE 78                 LD      A,B
05AF FE0F               CP      A,0FH
05B1 FAA905             JP      M,CONF2A

05B4 3A5022             LD      A,(ULTCIC)      ; calculo del numero del primer
05B7 E60F               AND     A,0FH           ; ciclo
05B9 47                 LD      B,A
05BA 3E0F               LD      A,0FH
05BC 90                 SUB     A,B
05BD 47                 LD      B,A

05BE 114022             LD      DE,DATOIN       ; inicializacion
05C1 0600               LD      B,00H

05C3 D3A1       CONF2B: OUT     PLC,A           ; numero del ciclo
05C5 D3A2               OUT     PLCC,A

05C7 DBA6               IN      A,PRMD          ; dato recibido
05C9 2F                 CPL
05CA 12                 LD      (DE),A

05CB 13                 INC     DE              ; control de ciclos
05CC 04                 INC     B
05CD 78                 LD      A,B
05CE FE0F               CP      A,0FH
05D0 FAC305             JP      M,CONF2B

05D3 3E0F               LD      A,0FH           ; mensaje final
05D5 32F723             LD      (DISMEM),A
05D8 3E10               LD      A,10H
05DA 32F823             LD      (DSMEM1),A
05DD 32F923             LD      (DSMEM2),A
05E0 32FA23             LD      (DSMEM3),A
05E3 32FB23             LD      (DSMEM4),A
05E6 32FC23             LD      (DSMEM5),A

05E9 C3F400             JP      DISUP           ; fin
```

IV.1.14. Lista de Componentes.

Componentes	Descripcion
S1	Conmutador de 2 Posiciones.
CL1A-CL1B	Reguladores LM323K 5V/3A.
C1A-C1B	Capacitores de .1 uF.
C2A-C2B	Capacitores de 1 uF.
CI1-CI3	Circuitos Integrados BUFFER 74LS367.
CI4-CI5	Memorias RWM DM74LS189.
CI6	Circuito Integrado AND 74LS08.
CI7-CI10	Circ. Int. MULTIPLEXER 74LS257A.
CI11-CI12	Circ. Int. COUNTER 74LS193.
CI13-CI16	Memorias RWM DM74LS189.
CI17	Circuito Integrado BUFFER 74LS367.
R1	Resistencia de 270 ohms a 250 mW.
D1	Diodo Emisor de Luz serie 1N4000.
CI18	Circuito Integrado NAND 74LS00.
CI19-CI20	Memorias PROM DM74S471.
CI21	Circ. Int. COUNTER 74LS193.
CI22	Circ. Int. MULTIPLEXER 74LS257A.
CI23	Circuito Integrado NAND 74LS00.
CI24	Circuito Integrado NOR 74LS02.
CI25-CI26	Circuitos Integrados FFD 74LS74.
CI27	Circuito Integrado AND 74LS08.
CI28	Circuito Integrado OR EX 74LS86.
CI29	Circuito Integrado OR 74LS32.
CI30	Circuito Integrado NOR 74LS27.
CI31	Circ. Int. COUNTER 74LS193.
CI32	Circuito Integrado FFD 74LS273.
CI33	Memoria RWM DM74LS189.
CI34-CI35	Circuitos Integrados DECOD 1/16 74LS154.
CI36	Circuito Integrado NAND 74LS00.
CI37	Circuito Integrado OR 74LS32.
CI38-CI40	Circuitos Integrados BUFFER 74LS367.
CI41	Circuito Integrado NAND 74LS00.
CI42-CI43	Circuitos Integrados TRANS/REC 74LS243.
CI44	Circuito Integrado NAND 74LS00.
CI45	Circuito Integrado FFD 74LS374.
CI46	Circuito Integrado OR 74LS32.
CI47	Circuito Integrado NOR 74LS02.

*

V. Conclusiones.

El objetivo de esta Tesis fue disenar una Estacion para Desarrollo de Hardware (EDH), con la finalidad de convertirse en una herramienta -- util al Estudiante, al Ingeniero y al Profesor de Electronica, en la prueba y desarrollo de Modulos para Sistemas de Computo basados en el Microprocesador Z80.

Las caracteristicas de la Estacion para Desarrollo de Hardware debe-- rian ser apropiadas en cuanto a facilidades, confiabilidad y costo. - Su diseno deberia ser modular a fin de simplificar su construccion, - mantenimiento y futuras ampliaciones.

La Estacion para Desarrollo de Hardware cumple los objetivos que mo-- tivaron la presente Tesis.

Su arquitectura en base a un Sistema de Bus le brinda la flexibilidad necesaria para aprovechar totalmente la capacidad del Microprocesador Z80. Las tarjetas que usa para desarrollo de los Modulos tienen las - mismas dimensiones que las del Bus S-100, lo que permite ser facil y economicamente adquiribles en el mercado.

El Sistema EDH incluye un Modulo de Memoria del tipo universal, que - le permite crecer en pasos de 2 Kbytes. Este Modulo maneja opciones que le permiten soportar desarrollos de Hardware y Software mas ela-- borados; su capacidad maxima es de 128 Kbytes, lo cual se logra por - medio de la tecnica de Bancos de Memoria.

El Monitor del Sistema EDH es un programa de facil operacion. Incluye los comandos necesarios para manejar el Sistema y la opcion de usar - teclas programables.

El Sistema EDH se controla fundamentalmente por medio del Modulo de - Teclado/Unidad Visual, el cual constituye el medio basico de comuni-- cacion hombre-maquina.

El Modulo del Emulador permite generar bajo control de programa, las senales que emite el Microprocesador Z80 transfiriendolas a un Bus -- Emulador y, a traves de cable, a cualquier Modulo que lo requiera --- fuera del Sistema

Como Sistema de Computo que es, la Estacion para Desarrollo de Hard-- ware recibe, almacena, procesa y emite informacion, pero esta disena- da para funcionar como el nucleo de un Sistema que sirva como herra-- mienta en el desarrollo y prueba de Modulos para Sistemas de Computo basados en el Microprocesador Z80.

En este sentido, la Estacion para Desarrollo de Hardware es un siste- ma abierto, sin compromisos en su diseno que limiten el desarrollo -- del Hardware.

Las distintas etapas en el desarrollo del Sistema EDH, han brindado - la oportunidad de estudiar todos los aspectos basicos de un Sistema - de Computo.

*

VI. Apendices.

VI.1. Monitor del Sistema EDH. El Monitor del Sistema EDH fue adaptado - del Monitor del Sistema Z80 Starter Kit, conocido como ZBUG Moni-- tor Version 1.0. Este Monitor se ha incluido en forma completa en en las paginas siguientes.
El Monitor del Sistema EDH se presenta posteriormente en codigo -- hexadecimal.

*

LOAD MAP

```
DK1:RESTR  .OBJ[1,1]    REL.    BEG ADDR 0000    END ADDR 00F3
DK1:DISUP  .OBJ[1,1]    REL.    BEG ADDR 00F4    END ADDR 0122
DK1:DECKY  .OBJ[1,1]    REL.    BEG ADDR 0123    END ADDR 0633
DK1:UTIL   .OBJ[1,1]    REL.    BEG ADDR 0634    END ADDR 07F4
DK1:UTILR  .OBJ[1,1]    ABS     BEG ADDR 07F8    END ADDR 07FF


GLOBAL CROSS REFERENCE TABLE

SYMBOL ADDR  REFERENCES
ARFLG  23E0  06A4 03C5 02F0
BFLG   23F4  0681 04BF 04BB 049E 0275 00A9
BPTAB  23E4  067E 0250
CTC1P  07FA  04D5
CTC3L  07FE  0759
DJZOMS 064F  0172 0135
DECKY  0123  0121
DIG2   23F5  068F 036C 01A6
DIG4   23F6  0692 0498 0396 0235 01AC
DISMEM 23F7  06AA 0687 065B 0617 05BB 0470 0443 03B6 02F4 00F5 00E4
+            00B4 0080
DISUP  00F4  062E 05CB 04C8 04B3 03B2 0393 0369 02EB 02D6 01BC 01A3
+            014D 0132 00F2 00D0 009D
DSMEM1 23F8  017F 00B9
DSMEM2 23F9  061F 05C3 0420 0378 0340 0334 00EB 00BC
DSMEM3 23FA  037B 0187 00C3
DSMEM4 23FB  0627 03E2 03A4 0385 0357 034A 0327 0315 0301 02E5 00C6
DSMEM5 23FC  043A 00C9
DSMEM6 23FD  0363
DSMEM7 23FE  0434 0360 0192
FLG24  2319  0737 072D 0724 0714 0707 04D0
INCHR  0758  0591
INCHR4 079D  07FE
KEYPTR 23DB  068A 03AF 0390 0366 0351 032B 01B9 01B5 01A0 019C 0195
+            018A 017A
KYTBL  07B9  0160
MFLG   23E1  069E 03CC 039C 02A5
NMIS   01CB  0067
OTCHR  06F4
OTCHR1 06F7  050C
OTCHR4 0732  07FA
PFLG   23DE  069B 03B9 0373 02AC
PRFLG  23DA  0699 05D6 02B3
PUNHEN 23C2  04EE
PUNHSL 23C3  04F2
PUNHSH 23C0  0552 0529 04E6
PUNHSL 23C1  0556 0530 04EA
REGTB  07D5  047A
REGTBP 07E5  045D
REMBP2 007E  0208
RESTAR 009F
RESTR1 00AE  060A 05D1 057F 04B0 0482 0465 043F 03D0 02B8 02A2
RFLG   23DF  06A1 03DF 030B
RST16  23C4  0011
RST24  23C7  0019
RST32  23CA  0021
RST40  23CD  0029
RST48  23D0  0031
RST56  23D3  0039
SCRP1  07AA  0102
```

```
SSFLG   23F3   0695 027A 0202 01F9
STKPT   23F2   0485 0468 0346 033F 01FA 01F6 01CD 00A1 0083 004B 0043
+              000A
STKPT1  23E3   033C
UARIN   06R3
UBASC   06RB
UFGCR   0686   00AF
UFOR1   063C   084B 0837 081B 0519 03DA 04D8 03E3 03D7 038B 035D 00RB 00E7 009A

UFOR2   0659   05D9 04A3 03E9 03D3 039F 037E 02D9 02HE 023B
UFOR3   067C   04A6 0248 0205 0062
UFOR4   06A7   04CB
UIF     23DD   0437 035A 0295 01F2 00AC 005F
UIX3    0634   04B6 0260 007A
ULACC   06DB   05B3 05AC 05A5 05A1 059D 0599
UPACC   06C4
UPACCS  06D5   056R 0566 055F 054E 0549 0544 053C 0538 0534 052D 0526


GLOBAL SYMBOL TABLE

ARFLG   23E0   BFLG    23F4   BPTAR   23E4   CTC1P   07FA
CTC3L   07FE   D20MS   064F   DECKY   0123   DIG2    23F5
DIG4    23F6   DISMEM  23F7   DISUP   00F4   DSMEM1  23F8
DSMEM2  23F9   DSMEM3  23FA   DSMEM4  23FB   DSMEM5  23FC
DSMEM6  23FD   DSMEM7  23FE   FLG24   23D9   INCHR   0758
INCHR4  079D   KEYPTR  23DB   KYTBL   07B9   MFLG    23E1
NM1S    01CB   OTCHR   06F4   OTCHR1  06F7   OTCHR6  0732
PFLG    23DE   PRFLG   23DA   PUNHEH  23C2   PUNHEL  23C3
PUNHSH  23C0   PUNHSL  23C1   REGTR   07D5   REGTSP  07E5
REMBP2  007E   RESTAR  009F   RESTR1  00AE   RFLG    23DF
RST16   23C4   RST24   23C7   RST32   23CA   RST40   23CD
RST48   23D0   RST56   23D3   SEGPT   07A6   SSFLG   23F3
STKPT   23E2   STKPT1  23E3   UARIN   06R3   UBASC   06RB
UFGCR   0686   UFOR1   063C   UFOR2   0659   UFOR3   067C
UFOR4   06A7   UIF     23DD   UIX3    0634   ULACC   06DB
UPACC   06C4   UPACCS  06D5
```

```
RESTR  (C) 1978 MICRO DESIGN CONCEPTS    MOSTEK Z80 ASSEMBLER V2.0 PAGE 0001
 ADDR  OBJECT    ST # SOURCE STATEMENT      DATASET = DK0:RESTR.SRC

                 0002          NAME    RESTR
                 0003 ;RST,NMI AND BP ROUTINES
                 0004 ;VERSION 1.7  5/18/78
                 0005          PSECT   REL
                 0006          GLOBAL  DSMEM4
                 0007          GLOBAL  DSMEM1
                 0008          GLOBAL  DISUP
                 0009          GLOBAL  UIF
                 0010          GLOBAL  UIX3
                 0011          GLOBAL  STKPT
                 0012          GLOBAL  RST16
                 0013          GLOBAL  RST24
                 0014          GLOBAL  RST32
                 0015          GLOBAL  RST40
                 0016          GLOBAL  RST48
                 0017          GLOBAL  RST56
                 0018          GLOBAL  DSMEM5
                 0019          GLOBAL  DSMEM2
                 0020          GLOBAL  DISMEM
                 0021          GLOBAL  DSMEM3
                 0022          GLOBAL  UFGCR
                 0023          GLOBAL  UFOR1
                 0024          GLOBAL  NMIS
                 0025          GLOBAL  UFOR3
                 0026          GLOBAL  RESTAR
                 0027          GLOBAL  REMRP2
                 0028          GLOBAL  RESTR1
                 0029          GLOBAL  RFLG
>0000            0030          ORG     0000H
                 0031 ;****RESTART INSTRUCTION HANDLER****
                 0032 ;****RESET ENTRY POINT - RST0 ****
                 0033 ;***RST0 IS RESERVED FOR RESET AND RST8 IS RESERVED
                 0034 ;    FOR BREAKPOINTS.  ALL OTHER RST'S ARE MAPPED
                 0035 ;    TO LOCATIONS IN RAM WHERE THE USER CAN INSERT
                 0036 ;    A JUMP TO THE SERVICE ROUTINE IN RAM FOR
                 0037 ;    RST 8,16,24,32,40, AND 56.  THE USER'S REGISTERS
                 0038 ;    ARE STACKED UPON BREAKPOINT ENTRY AND ALL
                 0039 ;    BREAKPOINTS ARE REMOVED FROM THE USER'S PROGRAM.
                 0040 ;****RESET PUSHBUTTON ENTRY POINT - RST0
'0000  31C023    0041          LD      SP,23C0H
'0003  C39F00'   0042          JP      RESTAR  ;FINISH INITALIZATION
                 0043 ;****BREAKPOINT ENTRY - RST8
 >0008           0044          ORG     0008H   ;START RESTART TABLE
'0008  ED73FFFF  0045 BPENT:   LD      (STKPT),SP      ;SAVE USER'S SP
'000C  F5        0046          PUSH    AF      ;STACK USER REGISTERS
'000D  C5        0047          PUSH    BC
'000E  1803      0048          JR      BPENT2-$
                 0049 ;****USER ENTRY - RST16
'0010  C3FFFF    0050          JP      RST16   ;MAP INTO RAM
'0013  ED57      0051 BPENT2:  LD      A,I     ;I INTO A, IFF INTO F
'0015  F3        0052          DI
'0016  1803      0053          JR      BPENT3-$
                 0054 ;****USER ENTRY - RST24
>0018  C3FFFF    0055          JP      RST24
'001B  D5        0056 BPENT3:  PUSH    DE
'001C  E5        0057          PUSH    HL
'001D  F5        0058          PUSH    AF      ;SAVE I AND IFF
'001E  1803      0059          JR      BPENT4-$
```

```
RESTR  (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK Z-80 ASSEMBLER V2.0 PAGE 0002
 ADDR   OBJECT    ST # SOURCE STATEMENT          DATASET = DK0:RESTR . SRC

                 0060 ;****USER ENTRY - RST38
0020   C3FFFF    0061          JP      RST38
0023   08        0062 BPENT4:  EX      AF,AF'  ;GET ALT REGS
0024   D9        0063          EXX
0025   F5        0064          PUSH    AF
0026   1803      0065          JR      BPENT5-$
                 0066 ;USER ENTRY - RST40
0028   C3FFFF    0067          JP      RST40
002B   C5        0068 BPENT5:  PUSH    BC
002C   D5        0069          PUSH    DE
002D   E5        0070          PUSH    HL
002E   1803      0071          JR      BPENT6-$
                 0072 ;****USER ENTRY - RST48
0030   C3FFFF    0073          JP      RST48
0033   DDE5      0074 BPENT6:  PUSH    IX
0035   00        0075          NOP             ;SAVE A BYTE
0036   1803      0076          JR      BPENT7-$
                 0077 ;****USER ENTRY - RST56
0038   C3FFFF    0078          JP      RST56
003B   FDE5      0079 BPENT7:  PUSH    IY
003D   3E03      0080          LD      A,03H
003F   D386      0081          OUT     (CTC2),A        ;DISABLE KYBD INTR
0041   DD2A0A00' 0082          LD      IX,(STKPT)      ;GET USERS SP
0045   DD23      0083          INC     IX
0047   DD23      0084          INC     IX
0049   DD224300' 0085          LD      (STKPT),IX
004D   DD7EFE    0086          LD      A,(IX-2)        ;TEST PC LOW BYTE
0050   B7        0087          OR      A       ;SET STATUS
0051   2003      0088          JR      NZ,BPENT8-$
0053   DD35FF    0089          DEC     (IX-1)          ;DEC HIGH BYTE
0056   DD35FF    0090 BPENT8:  DEC     (IX-2)          ;DEC LOW BYTE
0059   DD7EF4    0091          LD      A,(IX-12)       ;IFF ON STACK
005C   E604      0092          AND     4               ;MASK OUT IFF
005E   32FFFF    0093          LD      (OIF),A         ;SAVE FOR LATER
0061   CDFFFF    0094          CALL    UFORS   ;GET TAR ADDR AND BPFLG
0064   1803      0095          JR      BPNT8A-$
0066   C3FFFF    0096          JP      NMIS
0069   2813      0097 BPNT8A:  JR      Z,REMBP2-$      ;NO BKPTS TO DISPLAY REGS
                 0098 ;*******BREAKPOINT REMOVAL*********
                 0099 ;REMOVE BREAKPOINTS WHILE IN ZBUG, THEY WILL BE RESTORED O
                 0100 ;A EXECUTE OR PROCEED COMMAND
006B   DD7E02    0101 REMBP:   LD      A,(IY+2)        ;GET OP CODE TO RESTORE
                 0102 ;CHECK FOR MULTI-DEFINED BREAKPOINTS
006E   FECF      0103          CP      0CFH
0070   2807      0104          JR      Z,REMBP1-$      ;BRANCH IF MULTI-DEFINED
0072   DD6E01    0105 REMBP0:  LD      L,(IX+1)
0075   DD6600    0106          LD      H,(IX+0)        ;GET ADDR OF BP
0078   77        0107          LD      (HL),A  ;RESTORE OP CODE
0079   CDFFFF    0108 REMBP1:  CALL    UIX3    ;GET NEXT POSITION AND DEC B
007C   20ED      0109          JR      NZ,REMBP-$      ;GO AGAIN
                 0110 ;DISPLAY PC AND A
007E   DD21FFFF  0111 REMBP2:  LD      IX,DISMEM       ;POINT TO DISMEM
0082   2A4800'   0112          LD      HL,(STKPT)      ;POINT TO STACK
0085   2B        0113          DEC     HL
0086   7E        0114          LD      A,(HL)  ;GET PCH
0087   CDFFFF    0115          CALL    UFOR1   ;WRITE TO DISMEM
008A   DD23      0116          INC     IX
008C   DD23      0117          INC     IX
```

```
RESTR   (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0003
 ADDR   OBJECT    ST # SOURCE STATEMENT        DATASET = DK0:RESTR.SRC

^0094  2B          0118          DEC     HL
^0095  7E          0119          LD      A,(HL)  ;GET PCL
^0096  CD8800^     0120          CALL    UFOR1   ;WRITE TO DISMEM
^0099  DD23        0121          INC     IX
^009B  DD23        0122          INC     IX
^009D  2B          0123          DEC     HL
^009E  7E          0124          LD      A,(HL)  ;GET A
^009F  CD9100^     0125          CALL    UFOR1   ;WRITE TO DISMEM
^00A2  C3FFFF      0126          JP      DISUP   ;GO DISPLAY
                   0127 ;*******FINISH RESTART ROUTINE***********
^00A5  ED738300^   0128 RESTAR:  LD      (STKPT),SP      ;INIT SP
^00A9  31A823      0129          LD      SP,23A8H        ;INIT SP (ZBUGS)
^00AC  3E00        0130          LD      A,00H
^00AE  32FFFF      0131          LD      (BFLG),A        ;CLEAR BFLG
^00B1  325F00^     0132          LD      (UIF),A         ;CLEAR INTR BIT
^00B4  CDFFFF      0133 RESTR1:  CALL    UFGCK   ;CLR FLAGS
^00B7  3E11        0134          LD      A,11H   ;PROMPT CHARACTER
^00B9  328000^     0135          LD      (DISMEM),A
^00BC  3E10        0136          LD      A,10H   ;B CHARACTER
^00BE  32FFFF      0137          LD      (DSMEM1),A
^00C1  32FFFF      0138          LD      (DSMEM2),A
^00C4  1802        0139          JR      RESTR2-$
^00C6  1813        0140          JR      RESTR3-$        ;RELATIVE OFFSET CALCULA
^00C8  32FFFF      0141 RESTR2:  LD      (DSMEM3),A
^00CB  32FFFF      0142          LD      (DSMEM4),A
^00CE  32FFFF      0143          LD      (DSMEM5),A
^00D1  DB90        0144          IN      A,(KBSEL)       ;SENSE SWITCH
^00D3  CB6F        0145          BIT     5,A     ;BIT TEST
^00D5  C2F100^     0146          JP      NZ,DISUP        ;GO TO ZBUG
^00D8  C30008      0147          JP      0800H   ;RESTART TO PROM
                   0148 ;RELATIVE OFFSET CALCULATION ROUTINE-AUTOMATICALLY CALC
                   0149 ;RELATIVE OFFSET AND PLACES IT IN MEMORY AND ON THE DIS
                   0150 ;HL = ADDRESS OF OPCODE AT DESTINATION OF RELATIVE JUMP
                   0151 ;DE = ADDRESS OF THE INSTRUCTION WHICH HAS THE OFFSET
^00DB  13          0152 RESTR3:  INC     DE      ;POINT TO OFFSET
^00DC  D5          0153          PUSH    DE
^00DD  DDE1        0154          POP     IX      ;SAVE DISPLACEMENT ADDRESS
^00DF  13          0155          INC     DE      ;POINT TO NEXT OPCODE
^00E0  7D          0156          LD      A,L     ;GET LOW BYTE
^00E1  93          0157          SUB     E       ;SUBTRACT
^00E2  6F          0158          LD      L,A     ;SAVE OFFSET
^00E3  DD7700      0159          LD      (IX+0),A        ;CHANGE MEMORY
^00E6  7C          0160          LD      A,H     ;GET HIGH BYTE
^00E7  9A          0161          SBC     A,D     ;SUBTRACT
^00E8  DD21B400^   0162          LD      IX,DISMEM
^00EC  CD9A00^     0163          CALL    UFOR1   ;WRITE OVERRANGE TO DISPLAY
^00EF  DD21BC00^   0164          LD      IX,DSMEM2
^00F3  7D          0165          LD      A,L
^00F4  CDE700^     0166          CALL    UFOR1   ;WRITE OFFSET TO DISPLAY
^00F7  C3D000^     0167          JP      DISUP   ;GO DISPLAY
 >0086             0168 CTC2:    EQU     86H     ;SS/PROM PGM
 >0090             0169 KBSEL:   EQU     90H     ;MONITOR/PROM1 SENSING
                   0170          END

ERRORS=0000
```

```
DISUP  (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0001
 ADDR  OBJECT     ST # SOURCE STATEMENT      ; DATASET = DK0:DISUP.SRC

                  0002          NAME    DISUP
                  0003 ;DISPLAY UPDATE ROUTINE
                  0004 ;VERSION 0.3  3/13/78
                  0005          PSECT   REL
                  0006          GLOBAL  DISMEM
                  0007          GLOBAL  SEGPT
                  0008          GLOBAL  DECKY
                  0009          GLOBAL  DISUP
                  0010          GLOBAL  DISUP
 >00F4            0011          ORG     00F4H
                  0012 ;*********DISPLAY UPDATE ROUTINE*********
                  0013 ;**FUNCTION:MOVES DATA FROM DISMEM TO DISPLAYS.
                  0014 ;**ENTRY:DATA TO BE DISPLAYED IN SIX MEMORY LOCATIONS
                  0015 ;          STARTING AT LOCATION DISMEM.
                  0016 ;**EXIT:  ALL SIX DIGITS REFRESHED - REGISTERS DISTROYED:AR
                  0017 ;         IX, A, B, D, E, H, L
                  0018 ;**REGISTER USAGE:HL IS POINTER TO CURRENT LOC IN DISMEM.
                  0019 ;                 IX IS POINTER TO SEG PATTERN TABLE
                  0020 ;                 B  IS CURRENT DIGIT POINTER
                  0021 ;                 E  HOLDS DIGIT DATA TO BE DISPLAYED
                  0022 ;                 A AND D ARE SCRATCH REGISTERS
^00F4  21FFFF     0023 DISUP:   LD      HL,DISMEM       ;POINT TO START OF BUFFER
^00F7  0620       0024          LD      B,020H  ;POINT TO LEFT DIGIT.
^00F9  5E         0025 DISUP1:  LD      E,(HL)  ;GET FIRST BYTE
^00FA  1600       0026          LD      D,0     ;CLR D
^00FC  3E00       0027          LD      A,0
^00FE  D38C       0028          OUT     (DIGLH),A       ;TURN OFF DISPLAY
^0100  DD21FFFF   0029          LD      IX,SEGPT        ;INDEX INTO TABLE
 0104  DD19      0030          ADD     IX,DE           ;FOR SEGMENT PATTERN
^0106  DD7E00     0031          LD      A,(IX+0)        ;GET PATTERN
^0109  D388       0032          OUT     (SEGLH),A       ;OUT TO SEG LATCH
^010B  78         0033          LD      A,B
^010C  D38C       0034          OUT     (DIGLH),A       ;OUT TO DIGIT LATCH
^010E  1E2D       0035          LD      E,45D   ;SET DELAY FOR 1 MS
^0110  1D         0036 DISUP2:  DEC     E
^0111  3E00       0037          LD      A,0     ;DELAY LOOP
^0113  BB         0038          CP      E
^0114  20FA       0039          JR      NZ,DISUP2-$
^0116  3E01       0040          LD      A,01H
^0118  B8         0041          CP      B       ;B=1?
^0119  2805       0042          JR      Z,DISUP3-$      ;YES,RE-INIT & EXIT
^011B  23         0043          INC     HL      ;NO,POINT TO NEXT DIGIT
^011C  CB38       0044          SRL     B       ;SHIFT TO NEXT DIGIT
^011E  18D9       0045          JR      DISUP1-$
^0120  C3FFFF     0046 DISUP3:  JP      DECKY   ;GO DECODE KEYS
 >008C            0047 DIGLH:   EQU     8CH     ;WRITE ONLY - DIGIT SELECT
 >0088            0048 SEGLH:   EQU     88H     ;WRITE ONLY - SEG PATTERN
                  0049          END


ERRORS=0000
```

```
DECKY  (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK F8P-8O ASSEMBLER V2.0 PAGE 0001
 ADDR  OBJECT    ST # SOURCE STATEMENT          DATASET = DKO:DECKY .SRC

>0123           0002          ORG     0123H
                0003 ;KEY DECODE AND EXECUTION
                0004 ;VERSION 2.1  5/17/78
                0005          PSECT   REL
                0006          NAME    DECKY
                0007          GLOBAL  UFOR4
                0008          GLOBAL  UFOR1
                0009          GLOBAL  UFOR2
                0010          GLOBAL  UFOR3
                0011          GLOBAL  REMBP2
                0012          GLOBAL  D20MS
                0013          GLOBAL  KEYPTR
                0014          GLOBAL  STKPT
                0015          GLOBAL  STKPT1
                0016          GLOBAL  SSFLG
                0017          GLOBAL  UPACC'S
                0018          GLOBAL  OTCHR1
                0019          GLOBAL  OTHCR6
                0020          GLOBAL  UFOR1
                0021          GLOBAL  RPTAB
                0022          GLOBAL  D1G4
                0023          GLOBAL  D1G2
                0024          GLOBAL  D1G8
                0025          GLOBAL  MFLG
                0026          GLOBAL  BFLG
                0027          GLOBAL  RFLG
                0028          GLOBAL  ARFLG
                0029          GLOBAL  PFLG
                0030          GLOBAL  PRFLG
                0031          GLOBAL  INCHR
                0032          GLOBAL  FUNHSH
                0033          GLOBAL  FUNHSL
                0034          GLOBAL  FUNHEH
                0035          GLOBAL  FUNHEL
                0036          GLOBAL  REGTR
                0037          GLOBAL  RESTAR
                0038          GLOBAL  RESTR1
                0039          GLOBAL  D1SUP
                0040          GLOBAL  REGTBP
                0041          GLOBAL  D1SMEM
                0042          GLOBAL  RFLG
                0043          GLOBAL  U1F
                0044          GLOBAL  USMEM1
                0045          GLOBAL  USMEM2
                0046          GLOBAL  USMEM3
                0047          GLOBAL  USMEM4
                0048          GLOBAL  USMEM5
                0049          GLOBAL  USMEM6
                0050          GLOBAL  USMEM7
                0051          GLOBAL  ULACC
                0052          GLOBAL  FLG24
                0053          GLOBAL  U1X3
                0054          GLOBAL  KYTBL
                0055          GLOBAL  CTC1F
                0056          GLOBAL  NM1S
                0057          GLOBAL  DECKY
                0058 ;FUNCTION: BLANKS DISPLAY, SCANS ENTIRE MATRIX FOR CLOSUR
                0059 ;            IF CLOSURE IS FOUND A 20 MS DELAY IS INVOKED
```

```
ECKY   (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0002
ATOR   OBJECT     ST # SOURCE STATEMENT          DATASET = DK0:DECKY .SRC

                  0060 ;          TO DEBOUNCE KEYS.  KEY MATRIX IS SCANNED ONE
                  0061 ;          ROW AT A TIME, CHECKING ALL COLUMNS.  KEY VALUE
                  0062 ;          IS FOUND AND IF IT IS A HEX DIGIT IT IS PLACED
                  0063 ;          IN THE NEXT ENPTY LOCATION OF DISMEM (POINTER=
                  0064 ;          KEYPTR).  FLAGS ARE SET WHEN 2 DIGITS (DIG2)
                  0065 ;          AND 4 DIGITS HAVE BEEN ENTERED (DIG4).  WHEN
                  0066 ;          8 DIGITS HAVE BEEN ENTERED A MEMORY CHANGE IS
                  0067 ;          CALLED.  IF A COMMAND KEY IS DECODED THE CORRECT
                  0068 ;          SERVICE ROUTINE IS FOUND IN A JUMP TABLE (JPTAB)
                  0069 ;          COMMAND KEY EXECUTION ROUTINES ARE ALSO INCLUDED
                  0070 ;          IN THIS MODULE.
                  0071 ;ENTRY:    NONE REQUIRED, DONE AFTER DISP UPDATE (DISUP)
                  0072 ;EXIT:     IF CMND KEY, CMND HAS BEEN EXECUTED.  IF HEX
                  0073 ;          KEY, DATA HAS BEEN ENTERED INTO DISMEM.
                  0074 ;REGISTERS USED:
                  0075 ;          A-SCRATCH-KEYSL VALUE
                  0076 ;          B-DIGLH VALUE
                  0077 ;          C-SCRATCH-UNIQUE WORD FROM ROW AND CLMN DATA
                  0078 ;          HL-POINTER INTO KYTBL
                  0079 ;          HL-POINTER INTO DISMEM (KYPTR)
                  0080 ;          BC-SCRATCH-USED TO COMPUTE OFFSET INTO DISMEM
                  0081 ;          FXX-USED BY PROM PROGRAMMER TO HOLD POINTERS
                  0082 ;          DURING ERROR CHECKING AND DISP USING NEXT KEY
'0123  3E7F       0083 DECKY:   LD      A,7FH
'0125  D338       0084          OUT     (SEGLH),A       ;TURN OFF DISPLAY
'0127  3E3F       0085          LD      A,3FH
'0129  D33C       0086          OUT     (DIGLH),A       ;OUTPUT ALL ROWS LOW
'012B  DB90       0087          IN      A,(KBSEL)       ;INPUT COLUMN DATA
'012D  E61F       0088          AND     1FH             ;MASK OUT OTHER INPUTS
 012F  FE1F       0089          CP      1FH             ;ANY KEY DOWN?
'0131  CAFFFF     0090          JP      Z,DISUP ;NO, RETRUN TO DISPLAY
'0134  CDFFFF     0091          CALL    D20MS           ;YES,WAIT FOR DEBOUNCE
'0137  0E8C       0092          LD      C,DIGLH
'0139  0601       0093          LD      B,01H
'013B  ED41       0094 KEYDN1:  OUT     (C),B   ;PLACE SELECTED ROW LOW
'013D  DB90       0095          IN      A,(KBSEL)       ;INPUT COLUMN DATA
'013F  E61F       0096          AND     1FH     ;MASK OUT D5,D6,D7
'0141  FE1F       0097          CP      1FH     ;KEY DOWN?
'0143  200A       0098          JR      NZ,KEYDN2-$     ;YES,DECODE KEY
'0145  CB20       0099          SLA     B       ;NO,SELECT NEXT ROW
'0147  3E40       0100          LD      A,40H
'0149  B8         0101          CP      B       ;ALL DONE?
'014A  20EF       0102          JR      NZ,KEYDN1-$     ;NO LOOP BACK
'014C  C33201'    0103          JP      DISUP   ;YES EXIT
                  0104 ;**ENTRY CONDITIONS TO KEYDECODE ARE ROW IN B AND
                  0105 ;  COLUMN (MASKED BY 1F) IN A.
'014F  0E00       0106 KEYDN2:  LD      C,00H
'0151  0D         0107 KEYDN3:  DEC     C
'0152  CB38       0108          SRL     B
'0154  20FB       0109          JR      NZ,KEYDN3-$     ;FALL THROUGH WHEN B=0
'0156  CB21       0110          SLA     C
'0158  CB21       0111          SLA     C
'015A  CB21       0112          SLA     C
'015C  CB21       0113          SLA     C       ;GET VALUE TO HIGH NIBBLE
 015E  81         0114          ADD     A,C     ;COMBINE WITH A
'015F  21FFFF     0115          LD      HL,KYTBL        ;SETUP POINTER TO NEXT DIG
'0162  BE         0116 KEYDN4:  CP      (HL)    ;A=TABLE FOUND
'0163  2804       0117          JR      Z,KEYDN5-$      ;YES,FOUND IT
```

```
DECKY   (C) 1978 MICRO DESIGN CONCEPTS   NOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0003
ADDR    OBJECT    ST # SOURCE STATEMENT          DATASET = MD:DECKY.SRC

                  0118            INC     HL
                  0119            INC     B               ;ADJ POINTERS
                  0120            JR      KEYDN4-$        ;LOOP BACK
                  0121 KEYDN5:    IN      A,(KBSEL)       ;CK FOR KEY RELEASE
                  0122            AND     01FH    ;MASK OTHER INPUTS
                  0123            CP      01FH
                  0124            JR      NZ,KEYDN5-$     ;LOOP BACK TILL KEY UP
                  0125            CALL    DZONS   ;DEBOUNCE
                  0126            LD      A,B     ;GET KEY VALUE
                  0127            CP      10H
                  0128            JR      NC,KEYDN6-$     ;COMMAND KEY GO DECODE
                  0129            LD      HL,(KEYPTR)     ;HEX KEY, SAVE IN DISMEM
                  0130            LD      (HL),B
                  0131            OR      A
                  0132            LD      BC,DSMEM1
                  0133            SBC     HL,BC   ;ENTERED DIGIT 2?
                  0134            JR      Z,KEYDN4-$
                  0135            OR      A       ;CLEAR CARRY
                  0136            LD      BC,DSMEM3       ;CLEAR CARRY
                  0137            LD      HL,(KEYPTR)
                  0138            SBC     HL,BC   ;ENTERED DIGIT 4?
                  0139            JR      Z,KEYDN8-$      ;YES, INC FLAG
                  0140            OR      A
                  0141            LD      BC,DSMEM7
                  0142            LD      HL,(KEYPTR)     ;HEX KEY, SAVE IN DISMEM
                  0143            SBC     HL,BC   ;8 DIGITS IN?
                  0144            JR      Z,KEYDN9-$      ;YES, INC FLAG
                  0145 KEYDN7:    LD      HL,(KEYPTR)     ;GET KEY POINTER
                  0146            INC     HL      ;INCREMENT IT
                  0147            LD      (KEYPTR),HL     ;SAVE HL
                  0148            JP      DISUP   ;EXIT
                  0149 KEYDN4:    LD      HL,DIG2
                  0150            INC     (HL)
                  0151            JR      KEYDN7-$
                  0152 KEYDN8:    LD      HL,DIG4
                  0153            INC     (HL)
                  0154            JR      KEYDN7-$
                  0155 KEYDN9:    CALL    ALTER   ;ENTER DATA TO MEM,PORT OR REG
                  0156            LD      HL,(KEYPTR)
                  0157            DEC     HL      ;BACKUP POINTER TO 6 DIGITS IN
                  0158            LD      (KEYPTR),HL     ;SAVE HL
                  0159            JP      DISUP
                  0160 ;FIND ADDRESS OF COMMAND KEY HANDLER-KEYVALUE IN A
                  0161 KEYDN6:    SUB     10H     ;KEY 16=0 KEY 17=3
                  0162            LD      C,A
                  0163            ADD     A,C     ;DOUBLE OFFSET
                  0164            ADD     A,C     ;TRIPLE OFFSET
                  0165            LD      C,A
                  0166            LD      B,00H
                  0167            LD      HL,JPTAB        ;GET TABLE ADDR
                  0168            ADD     HL,BC   ;ADD OFFSET
                  0169            JP      (HL)
                  0170 ;SERVICE ROUTINE FOR NM1 INTERRUPTS, SINGLE STEP OR ABORT
                  0171 NM1S:      LD      (STEPT),SP      ;PRESERVE USER'S SP
                  0172            PUSH    AF
                  0173            LD      A,03H
                  0174            OUT     (CTC2),A        ;SHUT DOWN CTC
                  0175            LD      A,1     ;GET 1 INTO A, IFF INTO F
```

```
UECKY   (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0004
 ADDR   OBJECT    ST # SOURCE STATEMENT          DATASET = DK0:DECKY.SRC

'01D6   C5          0176          PUSH    BC
'01D7   D5          0177          PUSH    DE        ;SAVE REGISTERS
'01D8   E5          0178          PUSH    HL
 01D9   F5          0179          PUSH    AF
'01DA   08          0180          EX      AF,AF'
'01DB   D9          0181          EXX
'01DC   F5          0182          PUSH    AF
'01DD   C5          0183          PUSH    BC
'01DE   D5          0184          PUSH    DE
'01DF   E5          0185          PUSH    HL
'01E0   DDE5        0186          PUSH    IX
'01E2   FDE5        0187          PUSH    IY
'01E4   DD2ACD01'   0188          LD      IX,(STKPT)        ;GET USER'S SP
'01E8   DD23        0189          INC     IX
'01EA   DD23        0190          INC     IX        ;ADJUST TO USER'S REAL SP
'01EC   DD7EF4      0191          LD      A,(IX-12)         ;IFF ON STACK
'01EF   E604        0192          AND     4         ;MASK IFF
'01F1   32FFFF      0193          LD      (UIF),A   ;SAVE
'01F4   DD22E601'   0194          LD      (STKPT),IX        ;SAVE USER'S SP
'01F8   3AFFFF      0195 NMIS1:   LD      A,(SSFLG)         ;IN SS MODE?
'01FB   B7          0196          OR      A         ;SET STATUS
'01FC   CA2301'     0197          JP      Z,DECKY   ;NO, ITS A KB INTR
'01FF   3E00        0198 NMIS2:   LD      A,00H
'0201   32F901'     0199          LD      (SSFLG),A         ;CLEAR FLAG
'0204   CDFFFF      0200          CALL    UFOR3     ;GET BP DATA
'0207   CAFFFF      0201          JP      Z,REMBP2          ;NO BP, DISP PC AND A
'020A   1842        0202          JR      CCS1C-$   ;INSTALL BP,ACTIVATE KB AND RET
                    0203 ;JUMP TABLE FOR COMMAND KEYS
'020C   C33002'     0204 JPTAB:   JP      CCS1      ;EXEC
 020F   C37202'     0205          JP      CCS2      ;SS
'0212   C3A102'     0206          JP      CCS3      ;NON
'0215   C3A402'     0207          JP      CCS4      ;NEXT
'0218   C3E102'     0208          JP      CCS5      ;REG' DISP
'021B   C30803'     0209          JP      CCS6      ;REG DISP
'021E   C34B03'     0210          JP      CCS7      ;PORT EXAM
'0221   C39503'     0211          JP      CCS8      ;MEM EXAM
'0224   C39704'     0212          JP      CCS9      ;BP
'0227   C3CA04'     0213          JP      CCS10     ;PUNCH TAPE
'022A   C38105'     0214          JP      CCS11     ;LOAD TAPE
'022D   C30305'     0215          JP      CCS12     ;PROG PROM
                    0216 ;EXEC KEY HANDLER
'0230   3E00        0217 CCS1:    LD      A,00H
'0232   D38C        0218          OUT     (DIGLH),A         ;CLEAR DISPLAY
'0234   3AAC01'     0219          LD      A,(DIG4)
'0237   B7          0220          OR      A         ;SET STATUS
'0238   280D        0221          JR      Z,CCS1A-$         ;4 DIGITS NOT IN PROCEED
'023A   CDFFFF      0222          CALL    UFOR2     ;GET STARTING ADDR IN HL
'023D   DD2AF601'   0223          LD      IX,(STKPT)        ;SET POINTER TO USER'S SP
'0241   DD75FE      0224          LD      (IX-2),L
'0244   DD74FF      0225          LD      (IX-1),H          ;CHANGE PC TO NEW ONE
                    0226 ;PROCEED MODE
'0247   CD0502'     0227 CCS1A:   CALL    UFOR3     ;CHECK IF BP'S ACTIVE
'024A   202B        0228          JR      NZ,CCS2A-$        ;YES SS ONCE
'024C   1816        0229          JR      CCS1D-$   ;NO, ENABLE KB AND RET
                    0230 ;INSTALL BREAKPOINTS AND SAVE USERS OP CODES
'024E   DD21FFFF    0231 CCS1C:   LD      IX,BPTAB          ;POINTER TO BP TAB
'0252   DD6600      0232 CCS1CA:  LD      H,(IX+0)
'0255   DD6E01      0233          LD      L,(IX+1)          ;GET ADDR OF OP CODE
```

ADDR OBJECT ST # SOURCE STATEMENT DATASET = DKO:DECKY .SRC

```
'0258 7E          0234          LD      A,(HL)  ;GET OP CODE
'0259 0ECF        0235          LD      C,0CFH  ;INSTALL RST7 OP CODE
'025B 71          0236          LD      (HL),C  ;AS A BP
'025C DD7702      0237          LD      (IX+2),A        ;SAVE USERS OP CODE
'025F CDFFFF      0238          CALL    U1X3    ;POINT TO NEXT LOCATION
'0262 20EE        0239          JR      NZ,CCS1CA-$     ;LOOP BACK FOR MORE
                  0240 ;ENABLE KEYBOARD FOR ABORT
'0264 3E04        0241 CCS1D:   LD      A,04H
'0266 D38C        0242          OUT     (D1GLH),A       ;ENABLE NMI INTERRUPT
'0268 3E45        0243          LD      A,45H
'026A D386        0244          OUT     (CTC2),A        ;CTC TO MAKE A ZC/TO
'026C 3E01        0245          LD      A,01H
'026E D386        0246          OUT     (CTC2),A        ;ON ONE NEG PULSE IN
'0270 1812        0247          JR      CCS2B-$ ;RESTORE USER'S REGISTERS
                  0248 ;SINGLE STEP KEY HANDLER
'0272 3E00        0249 CCS2:    LD      A,00H
'0274 32FFFF      0250          LD      (BFLG),A        ;CLEAR OUT ANY BP
'0277 3E01        0251 CCS2A:   LD      A,01H
'0279 320202'     0252          LD      (SSFLG),A       ;SET SS FLG
'027C 3E07        0253          LD      A,07H
'027E D386        0254          OUT     (CTC2),A        ;NO INTR RESET CTC2
'0280 3E0B        0255          LD      A,11D   ;INIT CTC FOR 176 COUNTS
'0282 D386        0256          OUT     (CTC2),A        ;UNTIL NMI INTR
                  0257 ;RESTORE REGISTERS
'0284 FDE1        0258 CCS2B:   POP     IY
'0286 DDE1        0259          POP     IX
'0288 E1          0260          POP     HL
'0289 D1          0261          POP     DE
'028A C1          0262          POP     BC      ;ALTERNATE REGISTERS
'028B F1          0263          POP     AF
'028C 08          0264          EX      AF,AF'
'028D D9          0265          EXX
'028E F1          0266          POP     AF      ;I AND IFF
'028F ED47        0267          LD      I,A     ;RESTORE I
'0291 F1          0268          POP     HL
'0292 D1          0269          POP     DE
'0293 C1          0270          POP     BC
'0294 3AF201'     0271          LD      A,(U1F) ;GET IFF
'0297 B7          0272          OR      A       ;SET STATUS
'0298 C29E02'     0273          JP      NZ,CCS2C        ;GO ENABLE INTR
'029B F1          0274          POP     AF
'029C F3          0275          DI              ;DISABLE INTERRUPTS
'029D C9          0276          RET             ;TO USER'S PROGRAM
'029E F1          0277 CCS2C:   POP     AF
'029F FB          0278          EI              ;ENABLE INTERRUPTS
'02A0 C9          0279          RET             ;RETURN TO USER PROGRAM
                  0280 ;MON KEY HANDLER - FORCES RESTART OF MONITOR
                  0281 ;AND SAVES USER'S REGISTERS
'02A1 C3FFFF      0282 CCS3:    JP      RESTR1  ;CLEAR FLAGS, DISP HDR
                  0283 ;NEXT KEY HANDLER
                  0284 ;USED TO SELECT NEXT MEMORY, NEXT PORT OR NEXT ERROR
                  0285 ;IN PROM PROGRAMMING ROUTINE
'02A4 3AFFFF      0286 CCS4:    LD      A,(MFLG)
'02A7 FE01        0287          CP      01H     ;MEMORY EXAM MODE ACTIVE
'02A9 2812        0288          JR      Z,CCS4B-$
'02AB 3AFFFF      0289          LD      A,(PFLG)
'02AE FE01        0290          CP      01H     ;PORT EXAM MODE ACTIVE?
'02B0 2826        0291          JR      Z,CCS4C-$
```

```
'02B2  3AFFFF    0292          LD      A,(PRFLG)
'02B5  FE01      0293          CP      01H       ;FROM PROG MODE ACTIVE?
'02B7  C2A202'   0294 CCS4A:   JP      NZ,RESTK1         ;KEY HAD NO MEANING
 02BA  C33006'   0295          JP      CCS12D    ;YES, FROM PROG MODE ACTIVE
                 0296 ;                          GO DISPLAY NEXT ERROR
'02BD  CD3B02'   0297 CCS4B:   CALL    UFOR2     ;FORMAT TO FOUR BYTES IN HL
'02C0  23        0298          INC     HL        ;SELECT NEXT MEMORY ADDR
'02C1  7C        0299          LD      A,H
'02C2  CDFFFF    0300          CALL    UFOR1     ;UPDATE FIRST TWO DIGITS
'02C5  DD23      0301          INC     IX        ;IX IS ALREADY PTING TO DISMEM
'02C7  DD23      0302          INC     IX
'02C9  7D        0303          LD      A,L
'02CA  CDC302'   0304          CALL    UFOR1     ;UPDATE SECOND TWO DIGITS
'02CD  DD23      0305          INC     IX
'02CF  DD23      0306          INC     IX
'02D1  7E        0307          LD      A,(HL)    ;READ NEW MEMORY
'02D2  CDCB02'   0308          CALL    UFOR1     ;UPDATE THIRD TWO DIGITS
'02D5  C3BC01'   0309          JP      DISUP     ;GO DISPLAY
'02D8  CD6E02'   0310 CCS4C:   CALL    UFOR2     ;FORMAT TO FOUR BYTES IN HL
'02DB  4C        0311          LD      C,H       ;ONLY H HAS MEANING
'02DC  0C        0312          INC     C         ;SELECT NEXT PORT
'02DD  ED78      0313          IN      A,(C)     ;READ PORT
'02DF  79        0314          LD      A,C       ;IX ALREADY POINTING TO DISMEM
'02E0  CDD302'   0315          CALL    UFOR1
'02E3  DD21FFFF  0316          LD      IX,DSMEM4         ;POINT TO LAST TWO DIGITS
'02E7  CD6102'   0317          CALL    UFOR1     ;WRITE INTO DISMEM
'02EA  C3BC02'   0318          JP      DISUP     ;GO DISPLAY
                 0319 ;ALTERNATE REGISTER DISPLAY
 02ED  3E01      0320 CCS5:    LD      A,01
'02EF  32FFFF    0321          LD      (ARFLG),A         ;SET ARFLG
'02F2  DD21FFFF  0322          LD      IX,DISMEM         ;POINT TO REG
'02F6  3E12      0323          LD      A,12H     ;PRIME CHAR
'02F8  DD7701    0324          LD      (IX+1),A          ;WRITE TO DISPLAY
'02FB  CD5504'   0325          CALL    ALTER5
'02FE  7E        0326          LD      A,(HL)    ;GET INTO A
'02FF  DD21E502' 0327          LD      IX,DSMEM4         ;PT TO LAST 2 DIGITS
'0303  CDE802'   0328          CALL    UFOR1     ;WRITE A TO DISMEM
'0306  18BA      0329          JR      CCS6C-$
                 0330 ;MAIN REGISTER DISPLAY
'0308  3E01      0331 CCS6:    LD      A,01H
'030A  32FFFF    0332          LD      (RFLG),A          ;SET RFLG
'030D  CD6E04'   0333          CALL    ALTER6
'0310  3804      0334          JR      C,CCS6A-$         ;SPEC REG LT 6?
'0312  7E        0335          LD      A,(HL)    ;GET INTO A
'0313  DD210103' 0336          LD      IX,DSMEM4         ;PT TO LAST 2 DIGITS
'0317  CD0403'   0337          CALL    UFOR1     ;WRITE A TO DSMEM4,5
'031A  1846      0338          JR      CCS6C-$
                 0339 ;HANDLE PC,SP,IX,IY
'031C  FE03      0340 CCS6A:   CP      3         ;IS IT KEY 3
'031E  2835      0341          JR      Z,CCS6B-$         ;YES, IFF
'0320  FE02      0342          CP      2         ;KEY 2?
'0322  2817      0343          JR      Z,CCS6A1-$        ;YES, STACK POINTER
'0324  7E        0344          LD      A,(HL)    ;LOW BYTE INTO A
'0325  DD211503' 0345          LD      IX,DSMEM4
'0329  DD22B901' 0346          LD      (KEYPTR),IX       ;PREPARE FOR FOUR DIGITS
'032D  CD1803'   0347          CALL    UFOR1     ;FIRST BYTE TO DSMEM2,3
'0330  23        0348          INC     HL
 0331  7E        0349          LD      A,(HL)    ;HIGH BYTE TO A
```

```
DECKY   (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0007
 ADDR   OBJECT    ST # SOURCE STATEMENT        DATASET = DKO:DECKY.SRC

0352   DD21FFFF  0350          LD      IX,DSMEM2       ;POINT TO FIRST TWO
0356   CD2F03    0351          CALL    UFOR1   ;SECOND BYTE TO DSMEM4,5
0359   1820      0352          JR      CCS6D-$ ;GO DISPLAY
035B   3AFFFF    0353 CCS6A1:  LD      A,(STKPT1)
035E   DD213403  0354          LD      IX,DSMEM2
0362   CD3703    0355          CALL    UFOR1   ;FIRST BYTE TO DISMEM
0365   3A3F02    0356          LD      A,(STKPT)
0368   DD212703  0357          LD      IX,DSMEM4
036C   CD4303    0358          CALL    UFOR1   ;SECOND BYTE TO DISMEM
036F   DD222B03  0359          LD      (KEYPTR),IX     ;PREPARE FOR 4 DIGITS IN
0373   1813      0360          JR      CCS6D-$
          0361 ;HANDLE IFF
0375   DD214A03  0362 CCS6B:  LD      IX,DSMEM4
0379   3A9502    0363          LD      A,(UIF) ;GET VALUE 4 OR 0
037C   CD4B03    0364          CALL    UFOR1   ;WRITE TO DISMEM
037F   219201    0365          LD      HL,DSMEM7
0382   21FFFF    0366 CCS6C:   LD      HL,DSMEM6       ;SETUP FOR REG CHG
0385   225103    0367          LD      (KEYPTR),HL
0388   C3E602    0368 CCS6D:   JP      DISUP
          0369 ;EXAMINE PORTS
038B   3AA601    0370 CCS7:    LD      A,(DIG2)
038E   FE01      0371          CP      1
0390   2020      0372          JR      NZ,CCS7A-$      ;VERIFY TWO DIGITS IN
0392   32AC02    0373          LD      (PFLG),A        ;SET PFLG
0395   3E10      0374          LD      A,10H
0397   324003    0375          LD      (DSMEM2),A
039A   327701    0376          LD      (DSMEM3),A      ;BLANK DIGITS 2 AND 3
039D   CDD902    0377          CALL    UFOR2   ;GET PORT ADDR INTO HL
03A0   4C        0378          LD      C,H     ;MOVE FIRST TWO DIGITS TO C
03A1   ED78      0379          IN      A,(C)   ;READ THE PORT
03A3   DD215703  0380          LD      IX,DSMEM4       ;PT TO DATA DIGITS
03A7   CD5B03    0381          CALL    UFOR1   ;WRITE INTO DISMEM
03AA   DD23      0382          INC     IX
03AC   DD23      0383          INC     IX      ;SETUP TO CHANGE PORTS
03AE   DD226A03  0384          LD      (KEYPTR),IX     ;SETUP TO CHG PORT
03B2   C3E703    0385 CCS7A:   JP      DISUP   ;GO DISPLAY
          0386 ;EXAMINE MEMORY
03B5   3A3502    0387 CCS8:    LD      A,(DIG4)
03B8   B7        0388          OR      A       ;VERIFY FOUR DIGITS IN
03B9   2816      0389          JR      Z,CCS8A-$
03BB   32A502    0390          LD      (MFLG),A        ;SET MFLG
03BE   CD7E03    0391          CALL    UFOR2   ;GET FOUR BYTE ADDR IN HL
03A1   7E        0392          LD      A,(HL)  ;GET MEMORY DATA
03A2   DD218503  0393          LD      IX,DSMEM4       ;POINT TO CORRECT DIGITS
03A6   CD8B03    0394          CALL    UFOR1   ;WRITE INTO DISMEM
03A9   DD23      0395          INC     IX
03AB   DD23      0396          INC     IX
03AD   DD229003  0397          LD      (KEYPTR),IX     ;SETUP FOR DATA CHG
03B1   C3930?    0398 CCS8A:   JP      DISUP
03B4   DD?1F402  0399 ALTER:   LD      IX,DISMEM
03B8   3A7303    0400          LD      A,(PFLG)
03BB   B7        0401          OR      A
03BC   202A      0402          JR      NZ,ALTER2-$     ;PORT CHNG
03BE   3A0B03    0403          LD      A,(RFLG)
03C1   B7        0404          OR      A
03C2   2035      0405          JR      NZ,ALTR3-$      ;REG CHANGE
03C4   3AF002    0406          LD      A,(ARFLG)
03C7   B7        0407          OR      A
```

```
DECKY   (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0008
 ADDR   OBJECT   ST # SOURCE STATEMENT          DATASET = DK0:DECKY.SRC

'03C8  C24104'  0408          JP      NZ,ALTR4          ;ALT REG CHANGE
'03CB  3A9C03'  0409          LD      A,(MFLG)
'03CE  B7       0410          OR      A
'03CF  CA8802'  0411          JP      Z,RESTR1          ;FALSE ALARM RESTART
'03D2  CD9F03'  0412          CALL    UFOR2   ;GET ADDR INTO HL
'03D5  DD7E06   0413          LD      A,(IX+6)          ;GET HIGH NIBBLE
'03D8  CD8E04'  0414          CALL    ALTER7
'03DB  DDB607   0415          OR      (IX+7)  ;OR WITH LOW NIB
'03DE  77       0416          LD      (HL),A  ;WRITE TO MEMORY
'03DF  7E       0417          LD      A,(HL)  ;READ IT BACK
'03E0  DD21A403'  0418 ALTER1: LD      IX,DSMEM4
'03E4  CDA703'  0419          CALL    UFOR1   ;WRITE NEW DATA INTO DISMEM
'03E7  C9       0420          RET
'03E8  CD0303'  0421 ALTER2: CALL     UFOR2   ;GET ADDRESS INTO HL
'03EB  DD7E06   0422          LD      A,(IX+6)          ;GET ONE NIB
'03EE  CD8E04'  0423          CALL    ALTER7
'03F1  DDB607   0424          OR      (IX+7)  ;OR WITH LOW NIB
'03F4  4C       0425          LD      C,H
'03F5  ED79     0426          OUT     (C),A   ;WRITE TO PORT
'03F7  18E7     0427          JR      ALTER1-$
               0428 ;MAIN REGISTER CHANGE
'03F9  CDAE04'  0429 ALTR3:   CALL    ALTER6
'03FC  380D     0430          JR      C,ALTR3A-$        ;SPEC REG LT 6?
'03FE  DD7E06   0431          LD      A,(IX+6)          ;GET HIGH NIBBLE
'0401  CD8E04'  0432          CALL    ALTER7
'0404  DDB607   0433          OR      (IX+7)  ;OR WITH HIGH NIBBLE
'0407  77       0434          LD      (HL),A  ;MODIFY REGS ON STACK
'0408  C3E003'  0435          JP      ALTER1
 040B  FE03     0436 ALTR3A:  CP      3       ;IS IT KEY 3?
'040D  2824     0437          JR      Z,ALTR3B-$        ;YES, IFF
'040F  FE02     0438          CP      2       ;IS IT KEY 2?
'0411  282A     0439          JR      Z,ALTR3C-$        ;YES, SP
'0413  DD7E04   0440          LD      A,(IX+4)          ;GET HIGH NIBBLE
'0416  CD8E04'  0441          CALL    ALTER7
'0419  DDB605   0442          OR      (IX+5)  ;OR WITH LOW NIBBLE
'041C  23       0443          INC     HL
'041D  77       0444          LD      (HL),A  ;MODIFY HGIH BYTE
'041E  DD21A803'  0445        LD      IX,DSMEM2
'0422  CD0503'  0446          CALL    UFOR1   ;WRITE DATA TO DISPLAY
'0425  DD7E04   0447          LD      A,(IX+4)          ;GET HIGH NIBBLE
'0428  CD8E04'  0448          CALL    ALTER7
'042B  DDB605   0449          OR      (IX+5)  ;OR WITH LOW NIBBLE
'042E  2B       0450          DEC     HL
'042F  77       0451          LD      (HL),A  ;MODIFY HIGH BYTE
'0430  C3E003'  0452          JP      ALTER1
'0433  3A6003'  0453 ALTR3B:  LD      A,(DSMEM7)        ;GET NEW VALUE
'0436  325A03'  0454          LD      (UIF),A ;MODIFY UIF
'0439  32FFFF   0455          LD      (DSMEM5),A        ;WRITE TO DISPLAY
'043C  C9       0456          RET
               0457 ;SP CHANGE IS ILLEGAL
'043D  E1       0458 ALTR3C:  POP     HL      ;DUMMY TO SIM RET
'043E  C3D003'  0459          JP      RESTR1
               0460 ;ALTERNATE REGISTER CHANGE
 441   DD21B603'  0461 ALTR4:  LD      IX,DISMEM         ;POINT TO REG
 0445  CD5504'  0462          CALL    ALTER5
'0448  DD7E06   0463          LD      A,(IX+6)          ;GET HIGH NIBBLE
'044B  CD8E04'  0464          CALL    ALTER7                                    99
'044E  DDB607   0465          OR      (IX+7)  ;OR WITH LOW NIBBLE
```

```
                0466                LD      (HL),A  ;MODIFY REG ON STK
'0451  77       0467                JP      ALTER1  ;GO DISPLAY
'0452  C3E003'  0468 ALTER5:        LD      E,(IX+0)        ;GET REGISTER
'0455  DD5E00   0469                LD      B,00H
'0458  0600     0470                LD      D,00H
'045A  1600     0471                LD      HL,REGTBP       ;POINT TO ALT TABLE
'045C  21FFFF   0472                ADD     HL,DE   ;ADD KEYVALUE OFFSET
'045F  19       0473                LD      C,(HL)  ;GET VALUE
'0460  4E       0474                LD      A,C
'0461  79       0475                CP      25D     ;ILLEGAL VALUE?
'0462  FE19     0476                JP      Z,RESTR1        ;YES, RESTART MONITOR
'0464  CA3F04'  0477                LD      HL,(STKPT)      ;POINT TO USER'S STACK
'0467  2A4603'  0478                OR      A       ;CLEAR CARRY
'046A  B7       0479                SBC     HL,BC   ;SUB OFFSET FROM SP
'046B  ED42     0480                RET
'046D  C9       0481 ALTER6:        LD      IX,DISMEM       ;POINT TO REGISTER
'046E  DD214304' 0482               LD      E,(IX)  ;GET REGISTER
'0472  DD5E00   0483                LD      B,00H
'0475  0600     0484                LD      D,00H
'0477  1600     0485                LD      HL,REGTB        ;POINT TO MAIN TABLE
'0479  21FFFF   0486                ADD     HL,DE   ;ADD KEYVALUE OFFSET
'047C  19       0487                LD      C,(HL)  ;GET VALUE
'047D  4E       0488                LD      A,C
'047E  79       0489                CP      25D     ;ILLEGAL VALUE?
'047F  FE19     0490                JP      Z,RESTR1        ;IF SO RESTART
'0481  CA3F04'  0491                LD      HL,(STKPT)      ;POINT TO USER'S STACK
'0484  2A4604'  0492                LD      A,E     ;GET KEYVALUE AGAIN
'0487  7B       0493                OR      A       ;CLEAR CARRY
'0488  B7       0494                SBC     HL,BC   ;SUB OFFSET FROM SP
'0489  ED42     0495                CP      6       ;SPEC REG?
'048B  FE06     0496                RET
'048D  C9       0497 ALTER7:        SLA     A
'048E  CB27     0498                SLA     A
'0490  CB27     0499                SLA     A
'0492  CB27     0500                SLA     A
'0494  CB27     0501                RET
'0496  C9       0502 ;BREAKPOINT INSTALLATION
                0503 ;MAKES AN ENTRY INTO BREAKPOINT TABLE IF ENOUGH SPACE
                0504 ;EXISTS.  THE ACTUAL BREAKPOINTS ARE PUT INTO RAM
                0505 ;ON THE EXECUTE COMMAND.
'0497  3A9603'  0506 CCS9:         LD      A,(DIG4)        ;GET FLAG
'049A  B7       0507                OR      A       ;SET STATUS
'049B  2005     0508                JR      NZ,CCS90-$      ;FOUR DIGITS IN?
'049D  32/502'  0509                LD      (BPLG),A        ;NO CLEAR ALL BP'S
'04A0  1810     0510                JR      CCS91-$
'04A2  CDE903'  0511 CCS90:        CALL    UFOR2   ;GET BP ADDR INTO HL
'04A5  CD4802'  0512                CALL    UFOR3   ;GET BP STATUS
'04A8  2810     0513                JR      Z,CCS9B-$       ;NO BP, INSTALL ONE
'04AA  78       0514                LD      A,B
'04AB  FE05     0515                CP      05      ;TABLE FULL?
'04AD  2006     0516                JR      NZ,CCS9A-$      ;NO GO ON
'04AF  C38204'  0517                JP      RESTR1  ;YES, CLEAR DISPLAY:
'04B2  C38203'  0518 CCS91:        JP      DISUP
                0519 ;FIND FIRST FREE SPACE IN TABLE
'04B5  CD6002'  0520 CCS9A:        CALL    UIX3    ;IX+3 AND B-1
'04B8  20FB     0521                JR      NZ,CCS9A-$      ;LOOP TIL FREE SPC
                0522 ;INSERT NEW BREAKPOINTS IN TABLE
'04BA  3A9E04'  0523 CCS9B:        LD      A,(BPLG)
```

```
LCKY   (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0010
ADDR   OBJECT    ST # SOURCE STATEMENT          DATASET = PRO:DECKY .SRC

04BD   3C        0524          INC     A
04BE   32B804'   0525          LD      (BFLG),A        ;INCREMENT FLAG
04C1   DD7400    0526          LD      (IX),H  ;WRITE TO TABLE
04C4   DD7501    0527          LD      (IX+1),L
04C7   C3B304'   0528          JP      DISUP
                 0529 ;PUNCH DATA TO CASSETTE INTERFACE IN KC FORMAT
                 0530 ;STARTING ADDRESS IN DE', ENDING ADDRESS IN HL',
                 0531 ;INTERRUPTS ARE LIVE DURING THIS ROUTINE.
04CA   CDFFFF    0532 CCS10.   CALL    UFOR4   ;CLEAR DISPLAY
04CD   3E01      0533          LD      A,1
04CF   32FFFF    0534          LD      (FLG24),A       ;SET FLG24 FOR MARKS
04D2   ED5E      0535          IM      2
04D4   01FFFF    0536          LD      BC,CTC1P
04D7   78        0537          LD      A,R
04D8   ED47      0538          LD      I,A
04DA   79        0539          LD      A,C
04DB   D384      0540          OUT     (CTC0),A        ;INTERRUPT VECTOR
04DD   3E85      0541          LD      A,85H
04DF   D385      0542          OUT     (CTC1),A        ;CTC1 TO INTR
04E1   3E1A      0543          LD      A,26D
04E3   D385      0544          OUT     (CTC1),A        ;WITH THIS TIME CONST
04E5   3AFFFF    0545          LD      A,(PUNHSH)
04E8   57        0546          LD      D,A
04E9   3AFFFF    0547          LD      A,(PUNHSL)      ;GET STARTING ADDR
04EC   5F        0548          LD      E,A
04ED   3AFFFF    0549          LD      A,(PUNHEH)
04F0   67        0550          LD      H,A
04F1   3AFFFF    0551          LD      A,(PUNHEL)      ;GET ENDING ADDR
04F4   6F        0552          LD      L,A
04F5   AF        0553          XOR     A       ;CLEAR CARRY
04F6   ED52      0554          SBC     HL,DE   ;CALCULATE BLOCK SIZE
04F8   23        0555          INC     HL
04F9   E5        0556          PUSH    HL      ;SAVE IT
04FA   210000    0557          LD      HL,0000H
04FD   0603      0558          LD      B,3D    ;SETUP FOR 40 SEC DELAY
04FF   FB        0559          EI
0500   76        0560 CCS10A   HALT            ;WAIT FOR CTC1 INTR
                 0561 ;******************
0501   2D        0562          DEC     L
0502   20FC      0563          JR      NZ,CCS10A-$
0504   25        0564          DEC     H
0505   20F9      0565          JR      NZ,CCS10A-$     ;DELAY LOOP
0507   10F7      0566          DJNZ    CCS10A-$        ;40 SEC FOR LEADER
0509   3E3A      0567 CCS10B:  LD      A,03AH  ;START WITH A COLON
050B   CDFFFF    0568          CALL    OTCHR1  ;OUTPUT COLON
050E   AF        0569          XOR     A       ;CLEAR CARRY, INTRPTS DISABLED
050F   011000    0570          LD      BC,10H
0512   E1        0571          POP     HL
0513   ED42      0572          SBC     HL,BC   ;COMPARE HL WITH 10
0515   3009      0573          JR      NC,CCS10C-$     ;NC=>HL>10H
0517   09        0574          ADD     HL,BC   ;RESTORE BLOCK SIZE<10H
0518   85        0575          ADD     A,L     ;SET STATUS IF L=0
0519   47        0576          LD      B,A
051A   2E00      0577          LD      L,0
051C   2838      0578          JR      Z,CCS10F-$
051E   1801      0579          JR      CCS10D-$
0520   41        0580 CCS10C:  LD      B,C     ;B=BLOCK SIZE
0521   E5        0581 CCS10D:  PUSH    HL      ;SAVE
```

```
DECKY   (C) 1978 MICRO DESIGN CONCEPTS    MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0011
 ADDR   OBJECT   ST # SOURCE STATEMENT          DATASET = DKO:DECKY .SRC

'0582  0E00       0582            LD      C,0         ;CLEAR CHECKSUM
'0584  78         0583            LD      A,B         ;PUNCH RECORD LENGTH
'0585  CDFFFF     0584            CALL    UPACCS
'0588  3AE604'    0585            LD      A,(PUNHSH)
'058B  67         0586            LD      H,A
'058C  CD2605'    0587            CALL    UPACCS
'058F  3AEA04'    0588            LD      A,(PUNHSL)          ;PUNCH HIGH BYTE FIRST
'0592  6F         0589            LD      L,A
'0593  CD2D05'    0590            CALL    UPACCS
'0596  AF         0591            XOR     A               ;ACC=0,PUNCH RECORD TYPE
'0597  CD3405'    0592            CALL    UPACCS
'059A  7E         0593 CCS10E:    LD      A,(HL)      ;PUNCH DATA
'059B  CD3805'    0594            CALL    UPACCS
'059E  23         0595            INC     HL
'059F  10F9       0596            DJNZ    CCS10E-$
'0541  97         0597            SUB     A
'0542  91         0598            SUB     C
'0543  CD3C05'    0599            CALL    UPACCS  ;PUNCH CKECKSUM
'0546  3E0D       0600            LD      A,0DH   ;CR
'0548  CD4405'    0601            CALL    UPACCS
'054B  3E0A       0602            LD      A,0AH   ;LF
'054D  CD4905'    0603            CALL    UPACCS
'0550  7C         0604            LD      A,H
'0551  322905'    0605            LD      (PUNHSH),A          ;SAVE HIGH BYTE
'0554  7D         0606            LD      A,L
'0555  323005'    0607            LD      (PUNHSL),A          ;SAVE LOW BYTE
'0558  18AF       0608            JR      CCS10B-$            ;PUNCH SOME MORE
'055A  0603       0609 CCS10F:    LD      B,3
'055C  AF         0610 CCS10G:    XOR     A
'055D  4F         0611            LD      C,A
'055E  CD4E05'    0612            CALL    UPACCS  ;PUNCH EOF
'0561  10F9       0613            DJNZ    CCS10G-$
'0563  3E01       0614            LD      A,1
'0565  CD5F05'    0615            CALL    UPACCS
'0568  97         0616            SUB     A
'0569  91         0617            SUB     C
'056A  CD6605'    0618            CALL    UPACCS  ;CKSUM FOR EOF
'056D  21FFFF     0619 CCS10H:    LD      HL,0FFFFH           ;SETUP 5 SEC TRAILER
'0570  FB         0620            EI                  ;ENABLE INTERRUPTS
'0571  76         0621            HALT
                  0622 ;**************
'0572  2D         0623 CCS10J:    DEC     L
'0573  20FD       0624            JR      NZ,CCS10J-$
'0575  25         0625            DEC     H
'0576  20FA       0626            JR      NZ,CCS10J-$
'0578  F3         0627            DI
'0579  3E03       0628            LD      A,03H
'057B  D365       0629            OUT     (CTC1),A            ;DISABLE CTC KILL TONE
'057D  F3         0630            DI                  ;DISABLE INTERRUPTS
'057E  C3B004'    0631            JP      RESTR1  ;RESTART MONITOR
                  0632 ;LOAD DATA FROM CASSETTE TAPE TO MEMORY (KC FORMAT)
'0581  ED5E       0633 CCS11:     IM      2           ;SELECT MODE TWO INTERRUPTS
'0583  21FF0F     0634            LD      HL,0FFFH            ;START 15 SEC DELAY
'0586  0620       0635            LD      B,20H
 0588  2D         0636 CCS11A:    DEC     L
'0589  20FD       0637            JR      NZ,CCS11A-$
'058B  25         0638            DEC     H
'058C  20FA       0639            JR      NZ,CCS11A-$
```

```
CKY   (C) 1978 MICRO DESIGN CONCEPTS   NOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0012
LOOP  OBJECT    ST # SOURCE STATEMENT          DATASET = DK0:DECKY .SRC

)58E  10F8      0640          DJNZ    CCS11A-$
              0641 ;NOW INTO LEADER - START MAIN LOOP
)590  CDFFFF    0642 CCS11G:  CALL    INCHR   ;GET FIRST CHARS
)593  D63A      0643          SUB     03AH    ;CHECK FOR COLON
)595  20F9      0644          JR      NZ,CCS11G-$     ;LOOP BACK
)597  4F        0645          LD      C,A     ;ZERO CHECKSUM
)598  CDFFFF    0646          CALL    ULACC   ;GET RECORD LENGTH AND CKSUM
)59B  47        0647          LD      B,A
)59C  CD9905'    0648          CALL    ULACC   ;GET HIGH BYTE OF ADDR
)59F  57        0649          LD      D,A
)5A0  CD9D05'    0650          CALL    ULACC   ;GET LOW BYTE OF ADDR
)5A3  5F        0651          LD      E,A
)5A4  CDA105'    0652          CALL    ULACC   ;GET RECORD TYPE
)5A7  3D        0653          DEC     A       ;RECORD TYPE=EOF?
)5A8  F5        0654          PUSH    AF
)5A9  2807      0655          JR      Z,CCS11J-$      ;JP IF YES
)5AB  CDA505'    0656 CCS11H:  CALL    ULACC   ;GET DATA
)5AE  12        0657          LD      (DE),A  ;STORE IT.
)5AF  13        0658          INC     DE
)5B0  10F9      0659          DJNZ    CCS11H-$        ;LOOP FOR MORE
)5B2  CDAC05'    0660 CCS11J:  CALL    ULACC   ;GET CHECKSUM
)5B5  AF        0661          XOR     A       ;CLEAR ACC
)5B6  81        0662          ADD     A,C
)5B7  2814      0663          JR      Z,CCS11K-$      ;CKSUM OK
              0664 ;ERROR REPORTING
)5B9  DD217004'  0665          LD      IX,DISMEM       ;POINT TO BUFFER
)5BD  7A        0666          LD      A,D
)5BE  CD2304'    0667          CALL    UFOR1   ;DISPLAY HIGH BYTE
)5C1  DD217004'  0668          LD      IX,DSMEM2
)5C5  7B        0669          LD      A,E
)5C6  CDBF05'    0670          CALL    UFOR1   ;DISPLAY LOW BYTE
)5C9  F1        0671          POP     AF      ;RESTORE SP
)5CA  C3C804'    0672          JP      DISUP   ;GO DISPLAY
)5CD  F1        0673 CCS11K:  POP     AF      ;TST FOR EOF
)5CE  20C0      0674          JR      NZ,CCS11G-$     ;GO LOOK OF NEXT RECORD
)5D0  C37F05'    0675          JP      RESTR1  ;RESTART MONITOR
              0676 ;PROM PROGRAMMER FOR 2758 AND 2716 PROMS
              0677 ;MOVES DATA FOR RAM STARTING AT 2000H TO PROM
              0678 ;STARTING AND 1000H.  NUMBER OF BYTES TO BE
              0679 ;TRANSFERED (EXPRESSED IN FOUR HEX DIGITS) IS
              0680 ;IN DISPLAY BUFFER (DISMEM).
)5D3  3E01      0681 CCS12:   LD      A,01H
)5D5  32B302'    0682          LD      (PRFLG),A       ;SET PROM PROG FLG
)5D8  CDA304'    0683          CALL    UFOR2   ;GET BYTE COUNT
)5DB  E5        0684          PUSH    HL      ;SAVE IT
)5DC  C1        0685          POP     BC
)5DD  E5        0686          PUSH    HL
)5DE  210020    0687          LD      HL,2000H        ;RAM SOURCE DATA
)5E1  110010    0688          LD      DE,1000H        ;PROM DEST ADDR
)5E4  3E25      0689 CCS12A:  LD      A,25H   ;CTC FOR 26MS ZC/TO2
)5E6  D386      0690          OUT     (CTC2),A        ;NO INTR
)5E8  3ECB      0691          LD      A,203D
)5EA  D386      0692          OUT     (CTC2),A        ;TIME CONST
)5EC  3E80      0693          LD      A,80H   ;CLEAR DISPLAY SET
)5EE  D38C      0694          OUT     (DIGLH),A       ;PROM PROG EN=1
)5F0  EDA0      0695          LDI             ;WAIT STATE INSERTED UNTIL
)5F2  3E00      0696          LD      A,00H   ;CTC2 TIMES OUT TWICE
)5F4  D38C      0697          OUT     (DIGLH),A       ;CLEAR PROM PROG EN
```

```
DECKY   (C) 1978 MICRO DESIGN CONCEPTS    MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0013
 ADDR   OBJECT    ST # SOURCE STATEMENT          DATASET = DK0:DECKY.SRC

'05F6   3E03      0698          LD      A,03H    ;RESET CTC2
'05F8   D386      0699          OUT     (CTC2),A
'05FA   EAE405'   0700          JP      PE,CCS12A        ;LOOPBACK IF BC-1 NE 0
                  0701 ;VERIFY PROM IS LIKE RAM
'05FD   C1        0702          POP     BC       ;RESTORE BYTE COUNT
'05FE   210020    0703          LD      HL,2000H         ;HL PTS TO RAM
'0601   110010    0704          LD      DE,1000H         ;DE PTS TO PROM
'0604   1A        0705 CCS12B:  LD      A,(DE)   ;GET PROM DATA
'0605   EDA1      0706          CPI                      ;COMPARE WITH RAM
'0607   2006      0707          JR      NZ,CCS12C-$      ;ERROR REPORT IT
'0609   E2D105'   0708          JP      PO,RESTR1        ;NO ERRORS - RETURN
'060C   13        0709          INC     DE       ;UPDATE DE
'060D   18F5      0710          JR      CCS12B-$         ;CHECK NEXT BYTE
                  0711 ;ERROR HANDLER
'060F   F5        0712 CCS12C:  PUSH    AF       ;PRESERVE ERROR DATA
'0610   C5        0713          PUSH    BC       ;PRESERVE BYTE COUNT
'0611   D5        0714          PUSH    DE       ;PRESERVE ERROR ADDR
'0612   D9        0715          EXX              ;PRESERVE MAIN REGISTERS
'0613   D1        0716          POP     DE       ;DE=ERROR ADDR IN PROM
'0614   C1        0717          POP     BC       ;BC=BYTE COUNT
'0615   DD21BB05' 0718          LD      IX,DISMEM
'0619   7A        0719          LD      A,D
'061A   CDC705'   0720          CALL    UFOR1    ;HIGH BYTE TO DISMEM
'061D   DD21C305' 0721          LD      IX,DSMEM2
'0621   7B        0722          LD      A,E
'0622   CD1306'   0723          CALL    UFOR1    ;LOW BYTE TO DISMEM
'0625   DD21E203' 0724          LD      IX,DSMEM4
'0629   F1        0725          POP     AF       ;RETRIEVE ERROR DATA
'062A   CD1306'   0726          CALL    UFOR1    ;ERROR DATA TO DISMEM
'062D   C3CD03    0727          JP      DISUP    ;DISPLAY IT
                  0728 ;NEXT KEY INPUT DURING ERROR DISPLAY
'0630   D9        0729 CCS12D:  EXX              ;GET BACK POINTERS
'0631   13        0730          INC     DE       ;SETUP FOR NEXT ERROR
'0632   1800      0731          JR      CCS12B-$         ;LOOP FOR MORE ERRS
                  0732 ;*****EQUATES
>008C             0733 DIGLH    EQU     8CH      ;WRITE ONLY DIGIT SEL
>0090             0734 KBSEL    EQU     90H      ;READ ONLY KB SEL
>0084             0735 CTC0:    EQU     84H      ;INTRPT VECTOR
>0085             0736 CTC1:    EQU     85H      ;CASSETTE INTERFACE-PUNCH
>0086             0737 CTC2:    EQU     86H      ;SS/PROM PROGRAMMER
>0087             0738 CTC3:    EQU     87H      ;CASSETTE INTERFACE-LOAD
>0088             0739 SEGLH:   EQU     88H      ;WRITE ONLY SEGMENT LATCH
                  0740          END

ERRORS=0000
```

```
L   (C) 1978 MICRO DESIGN CONCEPTS    SYSTEM : CP/80 ASSEMBLER V2.0 PAGE 0001
DR  OBJECT    ST # SOURCE STATEMENT          DATASET = DKO:UTIL  .SRC

                0002          NAME    UTIL
                0003 ;UTILITY PACKAGE
                0004 ;VERSION 1.5  5/13/79
 34             0005          ORG     0034H
                0006          PSECT   REL
                0007          GLOBAL  U1G2
                0008          GLOBAL  U1G4
                0009          GLOBAL  U1G8
                0010          GLOBAL  U1X3
                0011          GLOBAL  UFOR1
                0012          GLOBAL  U2OMS
                0013          GLOBAL  UFOR2
                0014          GLOBAL  UFOR3
                0015          GLOBAL  UFOR4
                0016          GLOBAL  BPTAB
                0017          GLOBAL  SSFLG
                0018          GLOBAL  UABIN
                0019          GLOBAL  UBASC
                0020          GLOBAL  UPACC
                0021          GLOBAL  UPACCS
                0022          GLOBAL  ULACC
                0023          GLOBAL  OTCHR
                0024          GLOBAL  KYTBL
                0025          GLOBAL  REGTB
                0026          GLOBAL  REGTBP
                0027          GLOBAL  DISMEM
                0028          GLOBAL  STEPT
                0029          GLOBAL  SSOPT
                0030          GLOBAL  KEYPTR
                0031          GLOBAL  OTCHR
                0032          GLOBAL  OTCHR1
                0033          GLOBAL  INCHR
                0034          GLOBAL  FLG24
                0035          GLOBAL  PFLG
                0036          GLOBAL  ANFLG
                0037          GLOBAL  BFLG
                0038          GLOBAL  FFLG
                0039          GLOBAL  CTC1P
                0040          GLOBAL  NFLG
                0041          GLOBAL  PRFLG
                0042          GLOBAL  CTC3L
                0043          GLOBAL  CTC1P
                0044          GLOBAL  UFOCR
                0045          GLOBAL  OTCHR6
                0046          GLOBAL  INCHR4
                0047 ;ADDS THREE TO IX AND DECREMENTS B
                0048 ;USED IN MANIPULATING THE TABLE HOLDING
                0049 ;BREAKPOINT ADDRESSES AND USER OP CODES
'0634  DD23     0050 U1X3:    INC     IX
'0636  DD23     0051          INC     IX
'0638  DD23     0052          INC     IX
'063A  05       0053          DEC     B
'063B  C9       0054          RET                   ;USER TO CHECK FOR BP OVRFLW
                0055 ;IX POINTS TO NEXT TWO LOCATIONS IN DISMEM TO BE
                0056 ;WRITTEN INTO.  A CONTAINS THE DATA AND UFOR1 DOES
                0057 ;THE WRITING
'063C  47       0058 UFOR1:   LD      B,A     ;SAVE A IN TEMP REG
'063D  E60F     0059          AND     00FH    ;MASK OUT HIGH NIBBLE
```

```
(C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0002
ADDR  OBJ CODE  ST # SOURCE STATEMENT          DATASET = DEO:UTIL  .SRC

0640  DD7701      0060          LD      (IX+1),A        ;PUT IN LOW NIBBLE
0643  78          0061          LD      A,B     ;GET FULL BYTE
0644  CB3F        0062          SRL     A
0646  CB3F        0063          SRL     A
0648  CB3F        0064          SRL     A
064A  CB3F        0065          SRL     A       ;HIGH NIBBLE TO LOW
064C  DD7700      0066          LD      (IX+0),A        ;PUT HIGH NIBBLE IN DISMEM
064F  C9          0067          RET
                  0068 ;DELAY 20 MILLISECONDS AND RETURN
0650  21F808      0069 DZOMS:   LD      HL,08FFH
0653  2B          0070 DZOMS1:  DEC     L
0654  20FD        0071          JR      NZ,DZOMS1-$
0656  25          0072          DEC     H
0657  20FA        0073          JR      NZ,DZOMS1-$
0659  C9          0074          RET
                  0075 ;FORMATS FIRST FOUR DIGITS IN DISMEM INTO AN ADDRESS IN
                  0076 ;HL, USED BY MEMORY AND PORT UPDATE COMMANDS.
065A  DD21FFFF    0077 UFOR2:   LD      IX,DISMEM       ;POINTER TO FIRST DIGIT
065E  DD7E00      0078          LD      A,(IX)  ;GET FIRST DIGIT
0661  CB27        0079          SLA     A
0663  CB27        0080          SLA     A
0665  CB27        0081          SLA     A
0667  CB27        0082          SLA     A       ;MOVE TO HIGH NIBBLE IN A
0669  DDB601      0083          OR      (IX+1)  ;BRING IN LOW NIBBLE
066C  67          0084          LD      H,A     ;SAVE
066D  DD7E02      0085          LD      A,(IX+2)        ;SECOND BYTE HIGH NIBBLE
0670  CB27        0086          SLA     A
0672  CB27        0087          SLA     A
0674  CB27        0088          SLA     A
0676  CB27        0089          SLA     A
0678  DDB603      0090          OR      (IX+3)  ;BRING IN LOW NIBBLE
067B  6F          0091          LD      L,A     ;COMPLETE POINTER
067C  C9          0092          RET
                  0093 ;GETS TABLE ADDRESS INTO IX AND BFLG INTO B
067D  DD21FFFF    0094 UFOR3:   LD      IX,BPTAB        ;POINT IX TO START OF TAB
0681  3AFFFF      0095          LD      A,(BFLG)        ;NO. OF BP ACTIVE
0684  B7          0096          OR      A       ;SET STATUS
0685  47          0097          LD      B,A
0686  C9          0098          RET             ;USER TESTS STATUS
                  0099 ;FLAG INITALAZATION ROUTINE
0687  2158067     0100 UFGCR:   LD      HL,DISMEM
068A  22FFFF      0101          LD      (KEYPTR),HL     ;POINT INTO DISMEM
068D  3E00        0102          LD      A,00H
068F  32FFFF      0103          LD      (DIG2),A        ;ZERO FLAGS
0692  32FFFF      0104          LD      (DIG4),A
0695  32FFFF      0105          LD      (SSFLG),A
0698  32FFFF      0106          LD      (PRFLG),A
069B  32FFFF      0107          LD      (PFLG),A
069E  32FFFF      0108          LD      (MFLG),A
06A1  32FFFF      0109          LD      (RFLG),A
06A4  32FFFF      0110          LD      (ARFLG),A
06A7  C9          0111          RET
                  0112 ;PUTS BLANKS INTO DISPLAY MEMORY - DISMEM
06A8  0608        0113 UFOR4:   LD      B,8
06AA  2158067     0114          LD      HL,DISMEM
06AD  3E10        0115          LD      A,10H   ;BLANK CODE
06AF  77          0116 UFOR4A:  LD      (HL),A
06B0  23          0117          INC     HL
```

```
ADDR  OBJECT    ST #  SOURCE STATEMENT          DATASET = INFO:UTIL  .SRC

06B0  10FC      0118          DJNZ    UPDR4A-$          ;LOOP TILL DONE
06B2  C9        0119          RET
                0120 ;CONVERTS ONE ASCII DIGIT IN ACC TO BINARY
06B3  D630      0121 UABIN:   SUB     030H
06B5  FE0A      0122          CP      10
06B7  F8        0123          RET     M
06B8  D607      0124          SUB     7
06BA  C9        0125          RET
                0126 ;CONVERTS ONE DIGIT OF BINARY IN ACC TO ASCII
06BB  E60F      0127 UBASC:   AND     0FH       ;MASK OUT HIGH DIGIT
06BD  C690      0128          ADD     A,90H
06BF  27        0129          DAA
06C0  CE40      0130          ADC     A,40H
06C2  27        0131          DAA
06C3  C9        0132          RET
                0133 ;PUNCHS TWO CHARACTERS IN ACCUMULATOR-HIGH NIBBLE FIRST
06C4  D9        0134 UPACC:   EXX               ;USE ALT REGS IS OTCHR
06C5  F5        0135          PUSH    AF        ;SAVE ACC
06C6  0F        0136          RRCA
06C7  0F        0137          RRCA
06C8  0F        0138          RRCA
06C9  0F        0139          RRCA              ;GET UPPER DIGIT
06CA  CDF406'   0140          CALL    OTCHR
06CD  F1        0141          POP     AF
06CE  E60F      0142          AND     0FH       ;MASK OUT HIGH NIBBLE
06D0  CDF406'   0143          CALL    OTCHR
06D3  D9        0144          EXX               ;RESTORE REGISTERS
06 4  C9        0145          RET
                0146 ;PUNCHS TWO CHARACTERS IN ACCUMULATOR AND ADDS CHECKSUM
06D5  F5        0147 UPACCS:  PUSH    AF        ;SAVE ACC
06D6  81        0148          ADD     A,C       ;SUM CHECKSUM
06D7  4F        0149          LD      C,A       ;SAVE IN C
06D8  F1        0150          POP     AF
06D9  18E9      0151          JR      UPACC-$   ;PUNCH ACC
                0152 ;READS TWO CHARACTERS FROM TAPE AND CONVERTS TO BINARY
                0153 ;DATA RETURNED IN ACC AND CHECKSUM KEPT IN C
06DB  C5        0154 ULACC    PUSH    BC        ;SAVE C REG (CHECKSUM)
06DC  CD5807'   0155          CALL    INCHR     ;READ CHAR FIRST DIGIT
06DF  CDB306'   0156          CALL    UABIN     ;CONVERT TO BINARY
06E2  07        0157          RLCA
06E3  07        0158          RLCA
06E4  07        0159          RLCA
06E5  07        0160          RLCA              ;SHIFT TO HIGH NIBBLE
06E6  4F        0161          LD      C,A       ;SAVE FIRST DIGIT
06E7  CD5807'   0162          CALL    INCHR     ;GET SECOND DIGIT
06EA  CDB306'   0163          CALL    UABIN     ;CONVERT
06ED  B1        0164          OR      C         ;MERGE NIBBLES
06EE  C1        0165          POP     BC        ;RESTORE CHECKSUMS
06EF  F5        0166          PUSH    AF        ;SAVE ACC
06F0  81        0167          ADD     A,C       ADD TWO DIGITS TO CKSUM
06F1  4F        0168          LD      C,A       ;SAVE IN C
06F2  F1        0169          POP     AF        ;RESTORE ACC
06F3  C9        0170          RET
                0171 ;ROUTINE USES CTC CHANNEL 1 AS BASIC TIME BASE (4800HZ
                0172 ;OR 2400HZ).  ENTRY IS WITH ONE BINARY CHARACTER IN THE
                0173 ;LOW NIBBLE OF A.  THIS ROUTINE WILL CONVERT TO ASCII AND
                0174 ;THE SHIFT THE CHARACTER OUT WITH ONE START AND TWO STOP
                0175 ;BITS.  OUTPUTS ARE PULSES (ZC/TO) FROM CTC1 AT TWICE THE
```

```
UTIL    (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0004
 ADDR   OBJECT    ST # SOURCE STATEMENT          DATASET = DK0:UTIL   .SRC

                 0176 ;KANSAS CITY STANDARD RATES (1=4800HZ, 0=2400HZ).
'06F4  CD8B06'   0177 OTCHR:   CALL    UBASC    ;CONVERT TO ASCII
'06F7  FB        0178 OTCHR1:  EI
'06F8  57        0179          LD      D,A      ;A TO D
'06F9  3E10      0180          LD      A,10H    ;INIT A FOR CTC INTR
'06FB  2E0A      0181          LD      L,0AH    ;BIT COUNT OF WORD
'06FD  CB12      0182          RL      D        ;CARRY TO D0
'06FF  FE01      0183 OTCHR2:  CP      01D      ;A=1? WAIT UNTIL
'0701  20FC      0184          JR      NZ,OTCHR2-$      ;CTC ON NEXT TO LAST CNT
'0703  47        0185          LD      B,A      ;SAVE A
'0704  3E00      0186          LD      A,0
'0706  32FFFF    0187          LD      (FLG24),A        ;CLEAR FLG24-START BIT
'0709  78        0188          LD      A,B
'070A  76        0189 OTCHR3:  HALT
                 0190 ;*********
'070B  37        0191          SCF              ;SET CARRY
'070C  CB1A      0192          RR      D        ;SHIFT D ONE RIGHT
'070E  2D        0193          DEC     L
'070F  2007      0194          JR      NZ,OTCHR4-$
'0711  3E01      0195          LD      A,1
'0713  320707'   0196          LD      (FLG24),A        ;WORD OUT-MARK LINE
'0716  F3        0197          DI               ;EXIT WITH INTERRUPTS DISABLED
'0717  C9        0198          RET
'0718  FE01      0199 OTCHR4:  CP      1        ;NEXT TO LAST COUNT
'071A  20FC      0200          JR      NZ,OTCHR4-$      ;NO WAIT
'071C  CB42      0201          BIT     0,D      ;TEST NEXT BIT
'071E  2009      0202          JR      NZ,OTCHR5-$      ;NEXT BIT IS A ONE
'0720  47        0203          LD      B,A
'0721  3E00      0204          LD      A,0      ;NEXT BIT IS A ZERO
'0723  321407'   0205          LD      (FLG24),A        ;CLEAR FLG24
'0726  78        0206          LD      A,B
'0727  18E1      0207          JR      OTCHR3-$
'0729  47        0208 OTCHR5:  LD      B,A
'072A  3E01      0209          LD      A,1
'072C  322407'   0210          LD      (FLG24),A        ;SET FLG24
'072F  78        0211          LD      A,B              ;RESTORE A
'0730  18D8      0212          JR      OTCHR3-$         ;WAIT FOR END OF CHAR
                 0213 ;CTC1 INTERRUPT SERVICE ROUTINE DURING PUNCH
'0732  3D        0214 OTCHR6:  DEC     A        ;A IS CYCLE COUNTER
'0733  2020      0215          JR      NZ,OTCHR8-$      ;NOT LDAT COUNT,RETURN
'0735  DD212D07' 0216          LD      IX,FLG24
'0739  DDCB0046  0217          BIT     0,(IX+0)         ;TEST FLG24
'073D  200C      0218          JR      NZ,OTCHR7-$
'073F  3E85      0219          LD      A,85H    ;FLG24 IS CLR=ZERO
'0741  D385      0220          OUT     (CTC1),A         ;NEW CONTROL WORD
'0743  3E34      0221          LD      A,52D    ;INTERRUPTS LIVE
'0745  D385      0222          OUT     (CTC1),A         ;TIME CONST FOR 1200HZ
'0747  3E08      0223          LD      A,8      ;COUNT 8 CYCLES
'0749  180A      0224          JR      OTCHR8-$         ;RETURN
'074B  3E85      0225 OTCHR7:  LD      A,85H
'074D  D385      0226          OUT     (CTC1),A         ;NEW CONTROL WORD
'074F  3E1A      0227          LD      A,26D
'0751  D385      0228          OUT     (CTC1),A         ;TIME CONST FOR 2400HZ
'0753  3E10      0229          LD      A,16D    ;SETUP FOR 16 COUNTS
'0755  FB        0230 OTCHR8:  EI
'0756  ED4D      0231          RETI
                 0232 ;INPUT BIT RATE HAS BEEN AVERAGED AND IS IN (BITRT)
                 0233 ;ON EXIT CHARACTER IS IN A
```

```
TIL    (C) 1978 MICRO DESIGN CONCEPTS    MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0005
ADDR   OBJECT    ST # SOURCE STATEMENT          DATASET = DK0:UTIL  .SRC

                 0234 ;REGISTERS USED ARE A,B,H
                 0235 ;INTERRUPTS ARE ENABLED AND CTC1 INTERRUPTS AT THE BIT RAT
 758   21FFFF    0236 INCHR:   LD       HL,CTC3L
075B   7C        0237          LD       A,H
075C   ED47      0238          LD       I,A       ;SETUP I REGISTER
075E   7D        0239          LD       A,L
075F   D384      0240          OUT      (CTC0),A          ;CTC INTERRUPT VECTOR
0761   0608      0241 INCHR1:  LD       B,8D      ;BIT COUNT FOR A WORD
0763   FB        0242          EI
0764   3E00      0243          LD       A,0
0766   2600      0244          LD       H,0       ;CLEAR WORDS
0768   DB90      0245 INCHR1A: IN       A,(KBSEL)         ;GET INPUT DATA
076A   CB7F      0246          BIT      7,A
076C   20FA      0247          JR       NZ,INCHR1A-$      ;LOOP BACK FOR START BIT
076E   3EA5      0248          LD       A,0A5H   ;INTR-256 PRESCALER
0770   D387      0249          OUT      (CTC3),A          ;CTC3 CONTROL WORD
0772   3E0D      0250          LD       A,0DH    ;DIVIDE BY 2 FOR MID OF BIT
0774   D387      0251          OUT      (CTC3),A          ;CTC3 TIME CONSTANT
0776   3EA5      0252          LD       A,0A5H
0778   D387      0253          OUT      (CTC3),A
077A   3E1A      0254          LD       A,01AH
077C   D387      0255          OUT      (CTC3),A          ;NEXT ONE FULL BIT WIDTH
077E   76        0256          HALT
                 0257 ;*****************
077F   CB7F      0258          BIT      7,A
0781   2014      0259          JR       NZ,INCHR3-$       ;START BIT GONE-FALSE ST
0783   76        0260 INCHR2:  HALT              ;WAIT FOR FIRST BIT
                 0261 ;*****************
 784   E680      0262          AND      80H       ;MASK OUT OTHER INPUTS
0786   B4        0263          OR       H
0787   67        0264          LD       H,A       ;SAVE
0788   1018      0265          DJNZ )NCHR5-$
078A   CB7F      0266          BIT      7,A
078C   2809      0267          JR       Z,INCHR3-$        ;FRAMING ERROR,RESTART
078E   F3        0268          DI
078F   3E03      0269          LD       A,03H
0791   D387      0270          OUT      (CTC3),A          ;RESET CTC RTN WITH DATA
0793   7C        0271          LD       A,H
0794   E67F      0272          AND      7FH       ;MASK OUT START BIT
0796   C9        0273          RET
0797   3E03      0274 INCHR3:  LD       A,03H
0799   D387      0275          OUT      (CTC3),A          ;RESET CTC3,FRAMING ERROR
079B   18C4      0276          JR       INCHR1-$          ;GO LOOK FOR ANOTHER CHAR
                 0277 ;CTC INTERRUPT SERVICE ROUTINE DURING LOAD
079D   DB90      0278 INCHR4:  IN       A,(KBSEL)         ;GET DATA
079F   FB        0279          EI
07A0   ED4D      0280          RETI
07A2   CB0C      0281 INCHR5:  RRC      H
07A4   18DD      0282          JR       INCHR2-$          ;WAIT FOR NEXT BIT
                 0283 ;***********EQUATES AND TABLES**********
                 0284 ;CTC READ ACCESSES DOWN COUNTER,WRITE SETS UP COUNTER
 0084            0285 CTC0:    EQU      84H       ;RESERVED FOR USER
 0085            0286 CTC1:    EQU      85H       ;AUDIO CASSETTE-PUNCH
  086            0287 CTC2:    EQU      86H       ;SINGLE STEP AND PROM PROGRAMMER
 0087            0288 CTC3:    EQU      87H       ;AUDIO CASSETTE-LOAD
 0090            0289 KBSEL:   EQU      90H       ;CASSETTE DATA
                 0290 ;***SEVEN SEGMENT DISPLAY PATTERNS
07A6   40        0291 SEGPT:   DEFB     40H       ;0
```

```
*07AF  79          0292          DEFB    79H       ;1
*07B0  24          0293          DEFB    24H       ;2
*07B1  30          0294          DEFB    30H       ;3
*07B2  19          0295          DEFB    19H       ;4
*07B3  12          0296          DEFB    12H       ;5
*07B4  02          0297          DEFB    02H       ;6
*07B5  78          0298          DEFB    78H       ;7
*07B6  00          0299          DEFB    00H       ;8
*07B7  18          0300          DEFB    18H       ;9
*07B8  08          0301          DEFB    08H       ;A
*07B9  03          0302          DEFB    03H       ;B LOWER CASE
*07BA  46          0303          DEFB    46H       ;C
*07BB  21          0304          DEFB    21H       ;D LOWER CASE
*07BC  06          0305          DEFB    06H       ;E
*07BD  0E          0306          DEFB    0EH       ;F
*07BE  7F          0307          DEFB    7FH       ;BLANK
*07BF  3F          0308          DEFB    3FH       ;PROMPT
*07C0  7D          0309          DEFB    7DH       ;PRIME MARK
                   0310 ;***KEY VALUE LOOKUP TABLE
*07C1  FF          0311 KYTBL:   DEFB    0FFH      ;0 B=01,A=0F
*07C2  EF          0312          DEFB    0EFH      ;1 B=02,A=0F
*07C3  F7          0313          DEFB    0F7H      ;2 B=02,A=17
*07C4  FB          0314          DEFB    0FBH      ;3 B=02,A=1B
*07C5  DF          0315          DEFB    0DFH      ;4 B=04,A=0F
*07C6  E7          0316          DEFB    0E7H      ;5 B=04,A=17
*07C7  EB          0317          DEFB    0EBH      ;6 B=04,A=1B
*07C8  CF          0318          DEFB    0CFH      ;7 B=08,A=0F
*07C9  D7          0319          DEFB    0D7H      ;8 B=08,A=17
 07CA  DB          0320          DEFB    0DBH      ;9 B=08,A=1B
*07CB  DD          0321          DEFB    0DDH      ;A B=08,A=1D
*07CC  ED          0322          DEFB    0EDH      ;B B=04,A=1D
*07CD  FD          0323          DEFB    0FDH      ;C B=02,A=1D
*07CE  0D          0324          DEFB    00DH      ;D B=01,A=1D
*07CF  0B          0325          DEFB    00BH      ;E B=01,A=1B
*07D0  07          0326          DEFB    07H       ;F B=01,A=17
*07D1  0E          0327          DEFB    0EH       ;EXEC B=01,A=1E
*07D2  FE          0328          DEFB    0FEH      ;SS B=02,A=1E
*07D3  EE          0329          DEFB    0EEH      ;MON    B=04,A=1E
*07D4  DE          0330          DEFB    0DEH      ;NEXT B=08,A=1E
*07D5  CD          0331          DEFB    0CDH      ;REG' DISP B=10,A=1D
*07D6  CB          0332          DEFB    0CBH      ;REG DISP B=10,A=1B
*07D7  C7          0333          DEFB    0C7H      ;PORT EXAM B=10,A=17
*07D8  BF          0334          DEFB    0BFH      ;MEM EXAM B=10,A=0F
*07D9  BD          0335          DEFB    0BDH      ;BP B=20,A=1D
*07DA  BB          0336          DEFB    0BBH      ;PUNCH B=20,A=1B
*07DB  B7          0337          DEFB    0B7H      ;LOAD B=20H,A=17
*07DC  AF          0338          DEFB    0AFH      ;PROG B=20,A=0F
                   0339 ;***REGTAB-RELATES KEYVALUE TO POSITION OF REGISTER ON STK
*07DD  19          0340 REGTB:   DEFB    25D       ;KEY 0 NU NO DISPLAY
*07DE  02          0341          DEFB    02D       ;KEY 1=PC
*07DF  02          0342          DEFB    2D        ;KEY 2=SP
*07E0  0C          0343          DEFB    12D       ;KEY 3=1FF DISPLAY UIF
*07E1  16          0344          DEFB    22D       ;KEY 4=IX
*07E2  18          0345          DEFB    24D       ;KEY 5=IY
 07E3  0B          0346          DEFB    11D       ;KEY 6=I
*07E4  09          0347          DEFB    9D        ;KEY 7=H
*07E5  0A          0348          DEFB    10D       ;KEY 8=L
*07E6  19          0349          DEFB    25D       ;KEY 9=NU NO DISP
```

```
IIL   (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0007
ADDR  OBJECT    ST # SOURCE STATEMENT          DATASET = DEV:UTIL  .SRC

07DF  03        0350          DEFB    3D       ;KEY A=A
07E0  05        0351          DEFB    5D       ;KEY B=B
 7E1  06        0352          DEFB    6D       ;KEY C=C
 7E2  07        0353          DEFB    7D       ;KEY D=D
07E3  08        0354          DEFB    8D       ;KEY E=E
07E4  04        0355          DEFB    4D       ;KEY F=F
                0356 ;***ALTERNATE REGISTER SET
07E5  19        0357 REGTBP:  DEFB    25D      ;KEY 0 NU NO DISPLAY
07E6  19        0358          DEFB    25D      ;KEY 1 NU NO DISPLAY
07E7  19        0359          DEFB    25D      ;KEY 2 NU NO DISPLAY
07E8  19        0360          DEFB    25D      ;KEY 3 NU NO DISPLAY
07E9  19        0361          DEFB    25D      ;KEY 4 NU NO DISPLAY
07EA  19        0362          DEFB    25D      ;KEY 5 NU NO DISPLAY
07EB  19        0363          DEFB    25D      ;KEY 6 NU NO DISPLAY
07EC  13        0364          DEFB    19D      ;KEY 7=H'
07ED  14        0365          DEFB    20D      ;KEY 8=L'
07EE  19        0366          DEFB    25D      ;KEY 9=NU NO DISPLAY
07EF  0D        0367          DEFB    13D      ;KEY A=A'
07F0  0F        0368          DEFB    15D      ;KEY B=B'
07F1  10        0369          DEFB    16D      ;KEY C=C'
07F2  11        0370          DEFB    17D      ;KEY D=D'
07F3  12        0371          DEFB    18D      ;KEY E=E'
07F4  0E        0372          DEFB    14D      ;KEY F=F'
                0373          END


ERRORS=0000
```

```
ILR    (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0001
ADDR   OBJECT    ST # SOURCE STATEMENT          DATASET = DK0:UTILR .SRC

                 0002          NAME    UTILR
                 0003 ;UTILTIY RAM AND CONSTANTS
                 0004 ;VERSION 1.2  5/13/78
07F8             0005          ORG     07F8H
                 0006          PSECT ABS
                 0007          GLOBAL  CTC0L
                 0008          GLOBAL  CTC1P
                 0009          GLOBAL  OTCHR6
                 0010          GLOBAL  INCHR4
                 0011 ;CTC INTERRUPT VECTOR TABLE
07F8  D623       0012 CTC0:    DEFW    CTC0V   ;MAP TO RAM
07FA  FFFF       0013 CTC1P:   DEFW    OTCHR6  ;PUNCH VECTOR FA
07FC             0014 CTC2:    DEFS    2       ;NOT USED FOR INTR
07FE  FFFF       0015 CTC3L:   DEFW    INCHR4  ;LOAD VECTOR FE
23C0             0016          ORG     23C0H
                 0017          GLOBAL  DIG2
                 0018          GLOBAL  DIG4
                 0019          GLOBAL  DIG8
                 0020          GLOBAL  RPTAB
                 0021          GLOBAL  SSFLG
                 0022          GLOBAL  UIF
                 0023          GLOBAL  DSMEM
                 0024          GLOBAL  DSMEM1
                 0025          GLOBAL  DSMEM2
                 0026          GLOBAL  DSMEM3
                 0027          GLOBAL  DSMEM4
                 0028          GLOBAL  DSMEM5
                 0029          GLOBAL  DSMEM6
                 0030          GLOBAL  DSMEM7
                 0031          GLOBAL  STKPT
                 0032          GLOBAL  STKPT1
                 0033          GLOBAL  RST16
                 0034          GLOBAL  RST24
                 0035          GLOBAL  RST32
                 0036          GLOBAL  RST40
                 0037          GLOBAL  RST48
                 0038          GLOBAL  RST56
                 0039          GLOBAL  KEYPTR
                 0040          GLOBAL  FLG24
                 0041          GLOBAL  RFLG
                 0042          GLOBAL  ARFLG
                 0043          GLOBAL  JFLG
                 0044          GLOBAL  PFLG
                 0045          GLOBAL  MFLG
                 0046          GLOBAL  PRFLG
                 0047          GLOBAL  PUNHSH
                 0048          GLOBAL  PUNHSL
                 0049          GLOBAL  PUNHEH
                 0050          GLOBAL  PUNHEL
                 0051 ;***RAM VARIABLES
23C0             0052 PUNHSH:  DEFS    1       ;DUMP STARTING ADDR-HIGH BYTE
23C1             0053 PUNHSL:  DEFS    1       ;DUMP STARTING ADDR-LOW BYTE
23C2             0054 PUNHEH:  DEFS    1       ;DUMP ENDING ADDR-LOW BYTE
23C3             0055 PUNHEL:  DEFS    1       ;DUMP ENDING ADDR-LOW BYTE
23C4             0056 RST16:   DEFS    3       ;USER INSERTS JUMPS TO
23C7             0057 RST24:   DEFS    3       ;HANDLE RESTART 16-56
23CA             0058 RST32:   DEFS    3
23CD             0059 RST40:   DEFS    3
```

```
ILR  (C) 1978 MICRO DESIGN CONCEPTS   MOSTEK FLP-80 ASSEMBLER V2.0 PAGE 0002
DDR  OBJECT     ST # SOURCE STATEMENT          DATASET = DK0:UTILR.SRC

>23D0            0060 RST48:   DEFS    3
>23D3            0061 RST56:   DEFS    3
                 0062 ;***USER INSERTED JUMP FOR CTC0 INTERRUPT
 23D6            0063 CTC0V:   DEFS    3        ;CTC0 INTR WILL BE VECTORED HERE
                 0064 ;***USER INSERTED JUMP FOR CTC3 INTERRUPT
>23D9            0065 FLG24:   DEFS    1        ;FLAG FOR MARK (1) OUT (PUNCH)
>23DA            0066 PRFLG:   DEFS    1        ;PROM PROGRAMMER FLAG
>23DB            0067 KEYPTR:  DEFS    2        ;PTR FOR NEXT WRITE INTO DISMEM
>23DD            0068 UIF      DEFS    1        ;USER'S IFF2
>23DE            0069 PFLG     DEFS    1        ;PORT EXAMINE FLAG
>23DF            0070 RFLG     DEFS    1        ;REGISTER EXAMINE FLAG
>23E0            0071 ARFLG    DEFS    1        ;REGISTER EXAMINE FLAG (ALT)
>23E1            0072 MFLG:    DEFS    1        ;MEMORY EXAMINE FLAG
>23E2            0073 STKPT:   DEFS    1        ;USER'S STACK POINTER-HIGH BYTE
>23E3            0074 STKPT1:  DEFS    1        ;USER'S STACK POINTER-LOW BYTE
                 0075 ;**BREAKPOINT TABLE ORGANIZED AS TWO BYTES OF ADDR (H,L)
                 0076 ;**WHERE BREAKPOINT IS INSTALLED FOLLOWED BY THE ONE BYTE
                 0077 ;**OF THE OP CODE REPLACED BY THE RST 8 INSTRUCTION
>23E4            0078 BPTAB:   DEFS    15       ;BP ADDR AND OP CODE REMOVED
>23F3            0079 SSFLG    DEFS    1        ;SINGLE STEP MODE FLAG
>23F4            0080 BFLG     DEFS    1        ;NO OF BREAKPOINTS INSTALLED
>23F5            0081 DIG2     DEFS    1        ;2 DIGITS ENTERED FLAG
>23F6            0082 DIG4     DEFS    1        ;4 DIGITS ENTERED FLAG
>23F7            0083 DISMEM   DEFS    1        ;DISPLAY MEMORY BUFFER
>23F8            0084 DSMEM1   DEFS    1
>23F9            0085 DSMEM2   DEFS    1
>23FA            0086 DSMEM3   DEFS    1
>23FB            0087 DSMEM4   DEFS    1
 23FC            0088 DSMEM5   DEFS    1
>23FD            0089 DSMEM6   DEFS    1
>23FE            0090 DSMEM7   DEFS    1
                 0091          END

ERRORS=0000
```

```
A>DUMP B:MONITOR1.ECH
DUMP (DUMP FILE) VERSION 00.07

        RECORD 0
0000  21 C0 23 C3 3F 00 FF FF-ED 73 E3 23 F3 C3 18 03
0010  C3 C1 23 ED 57 F3 18 03-C3 C7 23 DD E5 F5 18 03
0020  C3 CA 23 08 D9 F5 18 03-C3 CD 23 C5 D5 E5 18 02
0030  C3 D0 23 DD E5 00 18 03-C3 D3 23 FD E5 3E 03 D3
0040  66 DD CA E2 23 DD 23 DD-23 DD 23 E3 23 DD 7E FE
0050  B7 20 03 DD 35 FF DD 35-FE DD 7E F4 E6 04 32 DD
0060  3D CD 7C 06 18 03 C3 CB-01 38 13 DD 7E 02 FE CF
0070  38 07 DD 6E 01 DD 66 00-77 CD D4 06 20 ED DD 21

        RECORD 1
0080  F7 33 3A E2 23 2B 7E CD-3C 06 DD 23 DD 23 2B 7E
0090  CD 3C 06 D0 23 D0 23 2B-7E CD 3C 06 C3 F4 00 ED
00A0  73 E2 23 31 A6 03 3E 00-32 F4 23 32 DD 23 CD 66
00B0  06 3E 11 32 F7 03 3E 10-32 F8 23 32 F9 23 18 02
00C0  18 13 32 FA 23 32 FB 23-32 FC 23 DB 90 CB 6F C2
00D0  F4 00 C3 00 08 13 D0 DD-E1 13 7D 93 6F DD 77 00
00E0  7C 9A DD 21 F7 23 CD 3C-06 DD 21 F9 23 7D CD 3C
00F0  06 C3 F4 00 21 F7 23 06-20 3E 16 00 3E 00 D3 3C

        RECORD 2
0100  DD 21 A6 07 DD 19 DD 7E-00 D3 88 78 D3 3C 1E 3D
0110  1D 3E 00 BB 20 FA 3E 01-B8 38 05 23 CB 38 18 D9
0120  C3 2D 01 3E 7F D3 88 3E-7F D3 3C DB 90 E6 1F FE
0130  1F CA F4 00 CD 4F 06 0E-8C 06 01 ED 41 DB 90 E6
0140  1F FE 1F 20 0A CB 20 3E-40 88 20 EF C3 F4 00 0E
0150  00 0D CB 38 20 F6 CB 21-CB 21 CB 21 CB 21 81 21
0160  B8 07 BE 38 01 23 04 18-F3 DB 90 E6 1F FE 1F 20
0170  F8 CD 4F 06 78 FE 10 30-4D 3A D8 23 70 B7 01 F8

        RECORD 3
0180  23 ED 42 28 20 87 01 FA-23 3A DB 23 ED 42 28 16
0190  B7 01 FE 23 2A DB 23 ED-42 23 16 2A DB 23 23 22
01A0  DB 23 C3 F4 00 21 F5 23-34 18 F0 21 F6 23 34 18
01B0  EA CD F4 03 2A DA 23 2B-22 DA 23 C3 F4 00 D6 10
01C0  4F 81 91 4F 06 00 21 0C-07 09 E9 ED 72 E2 23 F5
01D0  3E 03 D3 86 ED 57 C6 D5-E5 F5 08 D9 F5 C5 D5 E5
01E0  D5 E5 FD E5 DD 2A E2 23-DD 23 DD 23 DD 7E F4 E6
01F0  04 3C DD 23 DD 23 E2 23-3A F3 23 B7 CA 33 01 3E

        RECORD 4
0200  00 32 F3 23 CD 7C 0A CA-7E 00 18 12 C3 30 03 C3
0210  72 0A C3 A1 02 C3 A4 02-C3 E0 0C C3 08 03 C3 5B
0220  03 C3 95 03 C3 97 04 C3-00 18 C3 A0 05 C3 00 05
0230  3E 00 D3 8C 3A F6 23 B7-28 60 CD 59 06 DD 2A E7
0240  23 DD 73 FE DD 74 FF CD-7C 06 30 3B 18 16 DD 21
0250  E4 23 DD 66 00 DD 6E 01-7E 0A CF 71 DD 77 02 CD
0260  D4 06 20 EE 3E 04 D3 80-3E 40 D3 86 3E 01 D3 86
0270  18 12 3E 00 32 F4 23 3E-01 32 F3 23 3E 07 D3 86
```

```
         RECORD 5
0280  3E 05 D3 [illegible] FD E1 C1 E1-F1 C1 C1 F1 C3 C9 F1 ED
0290  47 E1 D1 C1 3A C0 [illegible] 57-[illegible] 3F [illegible] F1 F3 C9 F1 F5
02A0  C9 CD AE 00 3A E1 [illegible] FE-[illegible] [illegible] [illegible] 3A 0E [illegible] FE 01
02B0  C8 36 DA 0A 33 FE 01 [illegible]-A2 [illegible] [illegible] [illegible] C6 CD 79 06
02C0  23 7C CC DC 06 CD 23 [illegible]-[illegible] [illegible] [illegible] DC 06 CD 23 DD
02D0  23 7E CD DC 06 CD F4 00-[illegible] [illegible] [illegible] 40 00 ED 78 79
02E0  CD DC 06 DD 21 F5 03 [illegible]-[illegible] [illegible] [illegible] F4 00 3E 01 32
02F0  E0 03 DD 21 F7 03 3E 10-[illegible] 77 01 CD 3B 04 7E DD

         RECORD 6
0300  21 FB 03 CD DC 06 18 3A-2E 01 32 DF 03 CD 8E 04
0310  38 0A 7E DD 21 FE 03 CD-[illegible] [illegible] 12 16 FE 03 28 3D
0320  FE 02 28 17 7E DD 21 F5-[illegible] [illegible] [illegible] 0E 23 CD DC 06
0330  23 7E DD 21 F9 03 CD DC-06 18 [illegible] 3A E3 03 DD 21
0340  F9 03 CD DC 06 3A E2 03-[illegible] 21 F8 03 CD DC 06 DD
0350  22 D8 03 13 13 DD 21 F8-[illegible] 3A [illegible] 03 CD DC 06 21
0360  FE 03 21 FD 03 22 0E 03-[illegible] F4 [illegible] 3A F3 03 FE 01
0370  20 28 32 0E 23 3E 10 32-[illegible] [illegible] 32 FA 03 CD 3F 06

         RECORD 7
0380  4C ED 78 DD 21 F5 03 CD-3C 06 [illegible] 23 DD 23 [illegible] 22
0390  0E 23 CD F4 00 3A [illegible] 13-E7 28 [illegible] 32 E1 03 [illegible] C9
03A0  06 7E DD 21 FE 03 CD 3C-06 [illegible] [illegible] [illegible] [illegible] 00 22 0E
03B0  23 CD F4 00 DD 21 F7 03-DF 0E [illegible] 57 28 2A 3A DF
03C0  23 B7 28 35 3A E0 03 B7-C0 [illegible] [illegible] 3A E1 03 B7 CA
03D0  AE 00 CD 39 06 CD 7E 06-C1 3E [illegible] [illegible] E6 07 77 7E
03E0  DD 21 FE 03 CD 3C 06 C9-CD [illegible] [illegible] [illegible] 7E 06 CD 8E
03F0  04 [illegible] E6 07 4C ED 79 18-E7 [illegible] [illegible] 04 36 00 DD 7E

         RECORD 8
0400  06 CD 8E 04 DD E6 07 77-[illegible] 20 [illegible] FE CD 28 24 FE
0410  02 28 2A DD 7E 04 CD 8E-04 DD E6 [illegible] [illegible] 77 DD 21
0420  F9 03 CD 3C 06 DD 7E 04-CD 8E 04 DD E6 [illegible] 28 77
0430  CD E0 03 3A FE 03 32 [illegible]-[illegible] [illegible] FC [illegible] C9 E1 CD AE
0440  00 DD 21 F7 03 CD 3C 04-[illegible] 7E [illegible] CD 8E 04 DD E6
0450  07 77 CD E0 03 DD 7E 00-[illegible] [illegible] [illegible] 00 21 E3 07 19
0460  4E 79 FE 19 CA AE 00 3A-E[illegible] [illegible] 37 ED 42 C9 DD 21
0470  F7 23 DD 5E 00 06 00 16-00 21 D5 07 19 4E 79 FE

         RECORD 9
0480  19 CA AE 00 2A E2 03 7B-B7 ED 42 FE 06 C9 CB 27
0490  CB 27 CB 27 CB 27 C9 3A-F6 03 B7 28 05 32 F4 03
04A0  18 10 CD 39 06 CD 7C 06-28 10 79 FE 05 28 05 C3
04B0  AE 00 CD F4 00 CD 34 06-28 F8 3A F4 03 3C 32 F4
04C0  03 DD 74 00 DD 75 01 C3-F4 00 76 76 76 76 76 76
04D0  76 76 76 76 76 76 76 76-76 76 76 76 76 76 76 76
04E0  76 76 76 76 76 76 76 76-76 76 76 76 76 76 76 76
04F0  [illegible] 76 76 76 76 76 76 76-76 76 76 76 76 76 76 76

         RECORD 10
0500  3E [illegible] D3 A8 3A 50 22 E6-0F 32 FE 03 47 3E 0F 90
0510  47 4F 11 03 22 D3 A1 D3-A0 1A 2F 13 A3 3E 10 D3
0520  3F 3E 00 3A 57 1A 2F D3-A1 3E 10 D3 3F 3E 00 3A
0530  57 1A 2F D3 A0 3E 10 D3-3F 3E 00 3A 57 1A 2F D3
0540  A6 7B 1E D8 93 3F 7A 16-00 9A 37 1D 04 78 FE 0F
0550  FA 15 05 3E 10 D3 A1 3A-01 22 E6 01 32 FA 03 07
0560  07 07 07 F6 EF 47 79 F6-F0 A0 [illegible] D3 A1 D3 A0 3A
0570  50 22 D3 A7 3A 53 22 D3-A3 3E 01 D3 A0 3E 0E D3
```

```
        RECORD 11
0580  F7 CD 3E 10 32 F6 C3 32-F9 C3 3E 00 32 F0 C3 C3
0590  F4 00 76 76 76 76 76 76-76 76 76 76 76 76 76 76
05A0  3E 00 D3 A0 11 40 00 06-00 1B FF 10 FD 04 78 FE
05B0  0F FA A9 0C 3A 00 00 E6-0F 47 3E 0F 90 47 11 40
05C0  00 06 00 DD A1 DD A0 DE-A0 CF 10 1C 04 78 FE 0F
05D0  FA CD 0C 3E 0F 32 F7 C3-3E 10 32 F8 C3 32 F9 C3
05E0  32 FA C3 32 FB C3 32 F0-C3 C3 F4 00 76 76 76 76
05F0  76 76 76 76 76 76 76 76-76 76 76 76 76 76 76 76

        RECORD 12
0600  76 76 76 76 76 76 76 76-76 76 76 76 76 76 76 76
0610  76 76 76 76 76 76 76 76-76 76 76 76 76 76 76 76
0620  76 76 76 76 76 76 76 76-76 76 76 76 76 76 76 76
0630  76 76 76 76 DD 23 DD 03-DD 23 0C C9 47 E6 0F DD
0640  77 01 78 C8 3F C8 3F C8-3F C8 3F D0 77 00 C9 21
0650  FF C8 CD 00 FD CD 30 FA-C9 D0 21 F7 C3 D0 7E 00
0660  C8 27 C8 27 C8 27 C8 27-C0 86 01 27 D0 7E 00 C6
0670  27 C8 27 C8 27 C8 27 D0-86 01 6F C9 DD 21 E4 C3

        RECORD 13
0680  DA F4 C3 B7 47 C9 21 F7-C3 CC C8 CD 3E 00 32 F5
0690  C3 32 F6 C3 32 F7 C3 32-FA C3 32 0E C3 32 E1 C3
06A0  32 0F C3 32 E0 C3 C9 06-00 21 F7 C3 3E 10 77 23
06B0  10 FC C9 06 D0 FE 0A F8-06 07 C9 E6 0F 26 00 27
06C0  CE 40 27 C9 D8 FC 0F 0F-0F 0F C0 F4 06 F1 E6 0F
06D0  C0 F4 06 09 C9 FC 91 4F-F1 18 E9 C5 CD C8 07 CD
06E0  E3 32 07 07 07 07 4F CD-C8 07 CD E3 06 E1 C1 F5
06F0  81 4F F1 C9 CD E8 06 FE-07 3E 10 3E 0A C6 1C FE

        RECORD 14
0700  81 1C FC 47 3E 00 32 09-C3 78 76 37 C8 1A 3D 20
0710  07 16 01 32 09 C3 F3 C9-FE 01 16 FC CB 42 C8 09
0720  47 3E 00 32 09 C3 78 18-E1 47 3E 01 32 09 C3 78
0730  18 06 3D C8 00 00 21 03-C3 00 16 06 46 C0 00 3E
0740  80 00 20 3E 04 01 80 3E-06 18 0F 3E 80 01 80 3E
0750  1A 03 80 3E 10 F8 E6 40-21 FE 07 7C E0 47 7D 03
0760  84 06 08 FE 3E 00 3E 00-18 90 C9 7F CD FA 3E AC
0770  DD 87 3E 0D DD 87 3E AC-DD 87 3E 1A DD 87 76 CB

        RECORD 15
0780  7F 00 14 76 E6 00 84 67-10 18 C8 7F 00 09 FD 3E
0790  00 03 87 7D E6 7F C8 3E-80 00 07 16 04 D6 90 FB
07A0  ED 4D C8 00 18 D0 18 79-14 3E 19 12 00 78 00 18
07B0  00 0D 46 21 06 0E 7F 2F-7C FF EF F7 FB DF E7 EB
07C0  CF 07 08 00 ED F0 00 08-07 0E FE EE 1E CD C8 C7
07D0  BF 0D 0B 07 AF 19 0C 00-00 19 19 08 09 0A 19 0D
07E0  0C 06 07 00 04 19 19 19-19 19 19 19 1D 14 19 00
07F0  0F 10 11 12 0E 00 00 00-0E 00 00 07 00 23 90 07
```

VI.2. Datos para el Emulador.

SECUENCIA DE CICLOS DE MAQUINA A EMULAR

No	Dir Mem	CICLO Nombre	CICLO Codigo	Dir Mem	A15-A8	Dir Mem	A7-A0	Dir Mem	D7-D0 OUT
0	2200	_____	_____	2210	_____	2220	_____	2230	_____
1	2201	_____	_____	2211	_____	2221	_____	2231	_____
2	2202	_____	_____	2212	_____	2222	_____	2232	_____
3	2203	_____	_____	2213	_____	2223	_____	2233	_____
4	2204	_____	_____	2214	_____	2224	_____	2234	_____
5	2205	_____	_____	2215	_____	2225	_____	2235	_____
6	2206	_____	_____	2216	_____	2226	_____	2236	_____
7	2207	_____	_____	2217	_____	2227	_____	2237	_____
8	2208	_____	_____	2218	_____	2228	_____	2238	_____
9	2209	_____	_____	2219	_____	2229	_____	2239	_____
A	220A	_____	_____	221A	_____	222A	_____	223A	_____
B	220B	_____	_____	221B	_____	222B	_____	223B	_____
C	220C	_____	_____	221C	_____	222C	_____	223C	_____
D	220D	_____	_____	221D	_____	222D	_____	223D	_____
E	220E	_____	_____	221E	_____	222E	_____	223E	_____
F	220F	_____	_____	221F	_____	222F	_____	223F	_____

Ciclo de Maquina	Codigo
Fetch	01 H
Lect. de Memoria	02 H
Escr. de Memoria	03 H
Lect. de Puerto	04 H
Escr. de Puerto	05 H
Sol/Lib de Bus	06 H
Sol/Acep de INT	07 H
Sol/Acep de NMI	08 H
Salida de HALT	09 H

Dir Mem	Rango	Observaciones
2250 H _____	00 H <=Ult Ciclo<= 0F H	Ultimo Ciclo a Emular
2251 H _____	00 H = No, 01 H = Si	Repeticion de la Sec. ?
2252 H _____	00 H <= R <= 7F H	Registro R
2253 H _____	00 H <= I <= FF H	Registro I

II. Bibliografia.

1. 64 K Memory.
   Cromemco Inc., California USA, 1980.

2. AIM 65 User's Guide.
   Rockwell International Corp., California USA, 1979.

3. Cmos Cookbook.
   Don Lancaster.
   Howard W. Sams & Co. Inc., Indianapolis USA, 1977.

4. Computer System Architecture.
   Morris Mano.
   Prentice-Hall Inc., Englewood Cliffs USA, 1976.

5. Digital Technique.
   Naslund, Hansson, Bilkenroth and Raid.
   Telefonaktiebolaget LM Ericsson, Estocolmo Suecia, 1974.

6. Digital Electronics.
   George Young.
   Hayden Book Company Inc., New Jersey USA, 1980.

7. Low Power Schottky TTL Integrated Circuits.
   Motorola Semiconductor Products Inc., Phoenix USA, 1977.

8. Memory Data Book.
   National Semiconductor Corp., California USA, 1977.

9. Microcomputer Operating Systems.
   Mark Dahmke.
   Byte Publications Inc., Peterborough USA, 1982.

10. Microcomputer Data Book.
    Mostek Corp., Carroliton USA, 1979.

11. MCS 85 User's Manual.
    Intel Corp., California USA, 1978.

12. Principios Fundamentales de la Tecnica SPC.
    Bilkenroth, Wahlberg and Raid.
    Telefonaktiebolaget LM Ericsson, Estocolmo Suecia, 1978.

13. TRS-80 Microcomputer Technical Reference Handbook.
    Radio Shack, Fort Worth USA, 1978.

14. Z80 Microprocessor.
Nichols, Nichols and Rony.
Howard W. Sams & Co. Inc., Indianapolis USA, 1980.

15. Z80 Starter System.
S. D. Systems, Dallas USA, 1979.

16. Z80 Users Manual.
Joseph J. Carr.
Reston Publishing Company Inc., Reston USA, 1980.